Solaris のシステム管理 (ネットワークサービス)

Part No: 819–0381–19
2010 年 9 月

ORACLE

100908@24661

目次

図目次

表目次

例目次

はじめに

『Solaris のシステム管理 (ネットワークサービス)』は、Solaris のシステム管理マニュアルの一部です。このマニュアルでは、SunOS 5.10 オペレーティングシステムがすでにインストールされており、使用する予定のネットワークソフトウェアが設定済みであることを前提としています。Solaris 10 製品ファミリには、SunOS 5.10 オペレーティングシステムのほか、多くの機能が組み込まれています。

注 – この Solaris のリリースでは、SPARC および x86 系列のプロセッサアーキテクチャーをサポートしています。サポートされるシステムについては、Solaris OS: Hardware Compatibility Lists (`http://www.sun.com/bigadmin/hcl`) を参照してください。本書では、プラットフォームにより実装が異なる場合は、それを特記します。

本書の x86 に関連する用語については、以下を参照してください。

- 「x86」は、64 ビットおよび 32 ビットの x86 互換製品系列を指します。
- 「x64」は、具体的には 64 ビット x86 互換 CPU を指します。
- 「32 ビット x86」は、x86 をベースとするシステムに関する 32 ビット特有の情報を指します。

サポートされるシステムについては、Solaris OS: Hardware Compatibility List を参照してください。

対象読者

このマニュアルは、Solaris 10 リリースが稼働しているシステムの管理者を対象としています。このマニュアルを活用するには、1、2 年程度の UNIX システムの管理経験が必要です。UNIX システム管理のトレーニングコースに参加することも役に立ちます。

Solaris システム管理マニュアルセットの構成

システム管理マニュアルセットに含まれる各マニュアルとその内容は、次のとおりです。

マニュアルのタイトル	トピック
『Solaris のシステム管理 (基本編)』	ユーザーアカウントとグループ、サーバーとクライアントのサポート、システムのシャットダウンとブート、およびサービスの管理
『Solaris のシステム管理 (上級編)』	端末とモデムの設定、システムリソースの管理 (ディスク割り当て、アカウンティング、および crontab ファイルの管理)、システムプロセスの管理、および Oracle Solaris ソフトウェアの障害追跡
『Solaris のシステム管理 (デバイスとファイルシステム)』	リムーバブルメディア、ディスクとデバイス、ファイルシステム、およびデータのバックアップと復元
『Solaris のシステム管理 (IP サービス)』	TCP/IP ネットワーク管理、IPv4 と IPv6 アドレス管理、DHCP、IPsec、IKE、Solaris IP フィルタ、モバイル IP、IP ネットワークマルチのパス化 (IPMP)、および IPQoS
『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』	DNS、NIS、および LDAP のネーミングとディレクトリサービス (NIS から LDAP への移行、および NIS+ から LDAP への移行を含む)
『Solaris のシステム管理 (ネーミングとディレクトリサービス : NIS+ 編)』	NIS+ のネーミングとディレクトリサービス
『Solaris のシステム管理 (ネットワークサービス)』	Web キャッシュサーバー、時間関連サービス、ネットワークファイルシステム (NFS と Autofs)、メール、SLP、および PPP
『Solaris のシステム管理 (印刷)』	印刷に関するトピックや、サービス、ツール、プロトコル、およびテクノロジを使って印刷サービスおよびプリンタを設定および管理する方法
『Solaris のシステム管理 (セキュリティサービス)』	監査、デバイス管理、ファイルセキュリティー、BART、Kerberos サービス、PAM、Solaris 暗号化フレームワーク、特権、RBAC、SASL、および Solaris Secure Shell
『Oracle Solaris のシステム管理 (Oracle Solaris コンテナ : 資源管理と Oracle Solaris ゾーン)』	リソース管理に関連する計画と作業、拡張アカウンティング、リソース制御、フェアシェアスケジューラ (FSS)、資源上限デーモン (rcapd) による物理メモリーの制御、および資源プール (Solaris Zones ソフトウェア区分技術と lx ブランドゾーンによる仮想化)

マニュアルのタイトル	トピック
『Oracle Solaris ZFS 管理ガイド』	ZFS ストレージプールおよびファイルシステムの作成と管理、スナップショット、クローン、バックアップ、アクセス制御リスト (ACL) による ZFS ファイルの保護、ゾーンがインストールされた Solaris システム上での ZFS の使用、エミュレートされたボリューム、およびトラブルシューティングとデータ回復
『Oracle Solaris Trusted Extensions 管理の手順』	Oracle Solaris Trusted Extensions 機能固有のシステム管理
『Oracle Solaris Trusted Extensions 構成ガイド』	Solaris 10 5/08 リリース以降での、Oracle Solaris Trusted Extensions 機能の計画、有効化、および初期設定の方法

関連情報

次に、このマニュアルで参照している関連書籍を示します。

- 『Solaris のシステム管理 (上級編)』
- 『Solaris のシステム管理 (基本編)』
- 『Solaris のシステム管理 (IP サービス)』
- 『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』
- 『Solaris のシステム管理 (ネーミングとディレクトリサービス : NIS+ 編)』
- 『Oracle Solaris のシステム管理 (Oracle Solaris コンテナ : 資源管理と Oracle Solaris ゾーン)』
- 『Solaris のシステム管理 (セキュリティサービス)』
- Anderson、Bart、Bryan Costales、Harry Henderson 著、『UNIX Communications』、Howard W. Sams & Company、1987
- Costales、Bryan 著、『sendmail, Third Edition』、O'Reilly & Associates, Inc.、2002
- Frey、Donnalyn、Rick Adams 著、『!%@:: A Directory of Electronic Mail Addressing and Networks』、O'Reilly & Associates, Inc., 1993
- Krol、Ed 著、『The Whole Internet User's Guide and Catalog』、O' Reilly & Associates, Inc.、発行 1993 年
- O' Reilly、Tim、Grace Todino 著、『Managing UUCP and Usenet』、O' Reilly & Associates, Inc.、1992

関連情報

PPPoE の使用許諾権については、次の各ファイルを参照してください。

/var/sadm/pkg/SUNWpppd/install/copyright

/var/sadm/pkg/SUNWpppdu/install/copyright

/var/sadm/pkg/SUNWpppg/install/copyright

マニュアル、サポート、およびトレーニング

追加リソースについては、次の Web サイトを参照してください。

- マニュアル (http://docs.sun.com)
- サポート (http://www.oracle.com/us/support/systems/index.html)
- トレーニング (http://education.oracle.com) – 左のナビゲーションバーで「Sun」のリンクをクリックします。

Oracle へのご意見

Oracle はドキュメントの品質向上のために、お客様のご意見やご提案をお待ちしています。誤りを見つけたり、改善に向けた提案などがある場合は、http://docs.sun.com で「Feedback」をクリックしてください。可能な場合には、ドキュメントのタイトルやパート番号に加えて、章、節、およびページ番号を含めてください。返信を希望するかどうかもお知らせください。

Oracle Technology Network (http://www.oracle.com/technetwork/index.html) では、Oracle ソフトウェアに関する広範なリソースが提供されています。

- ディスカッションフォーラム (http://forums.oracle.com) で技術的な問題や解決策を話し合う。
- Oracle By Example (http://www.oracle.com/technology/obe/start/index.html) のチュートリアルで、手順に従って操作を体験する。
- サンプルコード (http://www.oracle.com/technology/sample_code/index.html) をダウンロードする。

表記上の規則

このマニュアルでは、次のような字体や記号を特別な意味を持つものとして使用します。

表 P–1 表記上の規則

字体または記号	意味	例
AaBbCc123	コマンド名、ファイル名、ディレクトリ名、画面上のコンピュータ出力、コード例を示します。	.login ファイルを編集します。 ls -a を使用してすべてのファイルを表示します。 system%
AaBbCc123	ユーザーが入力する文字を、画面上のコンピュータ出力と区別して示します。	system% **su** password:
AaBbCc123	変数を示します。実際に使用する特定の名前または値で置き換えます。	ファイルを削除するには、rm *filename* と入力します。
『』	参照する書名を示します。	『コードマネージャ・ユーザーズガイド』を参照してください。
「」	参照する章、節、ボタンやメニュー名、強調する単語を示します。	第 5 章「衝突の回避」を参照してください。 この操作ができるのは、「スーパーユーザー」だけです。
\	枠で囲まれたコード例で、テキストがページ行幅を超える場合に、継続を示します。	sun% **grep '^#define ** **XV_VERSION_STRING'**

Oracle Solaris OS に含まれるシェルで使用する、UNIX のデフォルトのシステムプロンプトとスーパーユーザープロンプトを次に示します。コマンド例に示されるデフォルトのシステムプロンプトは、Oracle Solaris のリリースによって異なります。

- C シェル

  machine_name% **command y|n** [*filename*]

- C シェルのスーパーユーザー

  machine_name# **command y|n** [*filename*]

- Bash シェル、Korn シェル、および Bourne シェル

  $ **command y|n** [*filename*]

- Bash シェル、Korn シェル、および Bourne シェルのスーパーユーザー

```
# command y|n [filename]
```

[ ] は省略可能な項目を示します。上記の例は、*filename* は省略してもよいことを示しています。

| は区切り文字 (セパレータ) です。この文字で分割されている引数のうち 1 つだけを指定します。

キーボードのキー名は英文で、頭文字を大文字で示します (例: Shift キーを押します)。ただし、キーボードによっては Enter キーが Return キーの動作をします。

ダッシュ (-) は 2 つのキーを同時に押すことを示します。たとえば、Ctrl-D は Control キーを押したまま D キーを押すことを意味します。

パート I

ネットワークサービス(トピック)

このパートでは、このマニュアルの概要、およびNCAサービスとNTPサービスの概要、作業、リファレンス情報について説明します。

◆ ◆ ◆ 第 1 章

ネットワークサービス (概要)

この章では、このマニュアルで説明する主なトピックの一覧を示します。また、このリリースに含まれる Perl コマンドについても説明します。

Solaris 10 リリースのトピック

このマニュアルでは、次のサービスとユーティリティーについて説明します。

44 ページの「Perl 5」
: システム管理作業を簡略化するためのスクリプトを生成するのに使用する Perl (Practical Extraction and Report Language) について説明します。

第 2 章「Web キャッシュサーバーの管理」
: Web ページのキャッシングにより、Web サーバーのパフォーマンスを向上させる NCA について説明します。

第 3 章「システムの時刻関連サービス」
: 多くのシステムで時間の同期をとるために使用できる、NTP などの時間に関するユーティリティーについて説明します。

第 4 章「ネットワークファイルシステムの管理 (概要)」
: リモートホストからファイルシステムへのアクセスを可能にする NFS プロトコルについて説明します。

第 7 章「SLP (概要)」
: 動的サービス発見プロトコルである SLP について説明します。

第 12 章「メールサービス (概要)」
: ネットワークに応じたルーティングにより、だれにでもメッセージを送信できるメールサービス機能について説明します。

第 15 章「Solaris PPP 4.0 (概要)」
リモートホスト間にポイントツーポイント接続を提供する PPP プロトコルについて説明します。

第 24 章「UUCP (概要)」
ホスト間のファイル交換を可能にする UUCP について説明します。

第 27 章「リモートシステムの利用 (概要)」
リモートシステムからファイルにアクセスするために使用するコマンド、`ftp`、`rlogin`、および `rcp` について説明します。

Perl 5

この Solaris リリースには、Perl (Practical Extraction and Report Language) 5.8.4 が付属しています。この強力な汎用プログラミング言語は、一般にフリーソフトウェアとして入手可能です。Perl はプロセス、ファイル、およびテキスト処理機能に優れ、複雑なシステム管理作業を行う際の標準的な開発ツールとして広く使用されています。

Perl 5 には、動的にロード可能なモジュールフレームワークが含まれています。このモジュールフレームワークを使用すると、特定の作業に新しい機能を追加することができます。多くのモジュールが http://www.cpan.org の Comprehensive Perl Archive Network (CPAN) から自由に入手できます。`gcc` を使用して CPAN のアドオンモジュールを構築してインストールするには、`/usr/perl5/5.8.4/bin/perlgcc` スクリプトを使用します。詳細は、`perlgcc`(1) のマニュアルページを参照してください。

Perl ドキュメントへのアクセス

この Solaris リリースには、Perl に関する情報ソースも含まれています。次に同じ情報へアクセスするための 2 通りの方法を示します。

MANPATH 環境変数に `/usr/perl5/man` を設定すると、マニュアルページにアクセスできます。次の例は Perl の概要を表示します。

```
% setenv MANPATH ${MANPATH}:/usr/perl5/man
% man perl
```

追加ドキュメントには、`perldoc` ユーティリティーを使用してアクセスできます。次の例も同じ概要を表示します。

```
% /usr/perl5/bin/perldoc perl
```

`perl` 概要ページには、このリリースに含まれているすべてのドキュメントの一覧が示されています。

Perl の互換性について

一般に、version 5.8.4 の Perl は以前のバージョンと互換性があるため、スクリプトの再作成や再コンパイルは必要ありません。ただし、XSUB ベースのモジュール (.xs) はすべて、再コンパイルおよび再インストールする必要があります。

Solaris 版 Perl の変更点

Solaris 版 Perl は 64 ビット整数、malloc システムコール、および大規模ファイルをサポートするようにコンパイルされています。また、必要なパッチも適用済みです。すべての構成情報の一覧については、次のコマンドの出力を参照してください。

```
% /usr/perl5/bin/perlbug -dv
---
Flags:
    category=
    severity=
---
Site configuration information for perl v5.8.4:
 .
 .
```

`perl -V` と入力すると、構成の要約リストを生成できます。

◆ ◆ ◆ 第 2 章

Web キャッシュサーバーの管理

この章では、Solaris NCA (ネットワークキャッシュとアクセラレータ) の概要について説明します。NCA を使用するための手順と NCA に関する参考資料を示します。さらに、Solaris 10 6/06 リリース用として、SSL (Secure Sockets Layer) の使用法の概要と、SSL カーネルプロキシを使って SSL パケット処理のパフォーマンスを改善するための手順が追加されました。

NCA (ネットワークキャッシュとアクセラレータ) (概要)

Solaris NCA (ネットワークキャッシュとアクセラレータ) は、HTTP 要求時にアクセスされる Web ページのカーネル内キャッシュを保持することにより、Web サーバーのパフォーマンスを向上させます。このカーネル内キャッシュはシステムメモリーを使用するため、通常は Web サーバーによって処理される HTTP 要求のパフォーマンスを、大幅に向上させます。HTTP 要求時に Web ページがシステムメモリー内に保持されているため、カーネルと Web サーバー間のオーバーヘッドが減少し、Web サーバーのパフォーマンスが向上します。NCA にはソケットインタフェースが用意されており、どのような Web サーバーでも最小限の変更で NCA と通信できます。

要求されたページがカーネル内キャッシュから取得された場合 (キャッシュヒット時) は、パフォーマンスが飛躍的に向上します。要求されたページがキャッシュ内になく、Web サーバーから取得する必要がある場合 (キャッシュミス時) でも、パフォーマンスは大幅に改善されます。

この製品は、専用の Web サーバー上で実行するようにします。NCA が動作するサーバー上で他の大きいプロセスを実行すると、問題が起きることがあります。

NCA はすべてのキャッシュヒットを記録するロギング機能を提供します。ログはパフォーマンスを向上させるためにバイナリ形式で格納されます。`ncab2clf` コマンドを使用すると、バイナリ形式のログを共通ログ形式 (CLF) に変換できます。

この Solaris リリースには、次のような機能強化が行われています。

- ソケットインタフェースの提供。
- AF_NCA サポートを可能にするベクトル化 sendfile システムコールの提供。詳細は、`sendfilev(3EXT)` のマニュアルページを参照してください。
- `ncab2clf` コマンド用の 2 つの新しいオプション、具体的には、選択された日付以前のレコードをスキップするための `-s` オプションと、指定された数のレコードを処理するための `-n` オプションの追加。
- `ncalogd.conf` ファイル内の `logd_path_name` を用いて raw デバイス、ファイル、または両者の組み合わせを指定できます。
- 1 つの Web サーバーによる複数の AF_NCA ソケットのオープンをサポート。複数のソケットを使用すると、1 つのサーバーで複数の Web サーバーを実行できます。
- `/etc/nca/ncaport.conf` という名前の新しい設定ファイル。このファイルによって、NCA で使用する IP アドレスやポートを管理することができます。Web サーバーによっては、AF_NCA ソケットを直接にサポートしない場合があります。AF_NCA ソケットをサポートしないサーバーでは、この /etc/nca/ncaport.conf ファイルと NCA ソケットユーティリティーライブラリを使って、AF_INET ソケットを AF_NCA ソケットに変換します。

SSL (Secure Sockets Layer) プロトコルを使用する Web サーバー

Solaris 10 6/06 リリースでは、SSL (Secure Sockets Layer) プロトコルを使用するように、Apache 2.0 と Sun Java System Web Server を設定できるようになりました。このプロトコルを使えば、2 つのアプリケーションの間で、機密性、メッセージの完全性、およびエンドポイント認証を実現できます。Solaris のカーネルに変更が加えられ、SSL トラフィック処理の高速化が図られました。

SSL カーネルプロキシは、SSL プロトコルのサーバー側を実装します。このプロキシによって、Web サーバーのようなサーバーアプリケーションの SSL パフォーマンスが向上します。これはユーザーレベルの SSL ライブラリを使用したアプリケーションを通じて達成されます。アプリケーションのワークロードに応じて、パフォーマンスは最大 35% 向上する可能性があります。

SSL カーネルプロキシは、多くの一般的な暗号化スイートのほかに、SSL 3.0 プロトコルと TLS 1.0 プロトコルをサポートします。完全な一覧については、`ksslcfg(1M)` の

マニュアルページを参照してください。このプロキシは、サポートされないすべての暗号化スイートに対して、ユーザーレベルの SSL サーバーへのフォールバックを行うよう構成することができます。

SSL カーネルプロキシを使用するようにサーバーを構成するための手順を示します。

- 56 ページの「SSL カーネルプロキシを使用するように Apache 2.0 Web サーバーを設定する方法」
- 58 ページの「SSL カーネルプロキシを使用するように Sun Java System Web Server を設定する方法」
- 60 ページの「ゾーン内での SSL カーネルプロキシの使用」

Web キャッシュサーバーの管理 (作業マップ)

次の表に、NCA または SSL を使用するために必要な手順を示します。

作業	説明	参照先
NCA の利用を計画する	NCA を有効にする前に解決すべき事項のリスト。	50 ページの「NCA の利用を計画する」
NCA を有効にする	Web サーバー上の Web ページのカーネル内キャッシュを有効にするための手順。	51 ページの「Web ページのキャッシングを有効にする方法」
NCA を無効にする	Web サーバー上の Web ページのカーネル内キャッシュを無効にするための手順。	54 ページの「Web ページのキャッシングを無効にする方法」
NCA ロギングを管理する	NCA ロギング処理を有効または無効にするための手順。	54 ページの「NCA ロギングを有効または無効にする方法」
NCA ソケットライブラリをロードする	AF_NCA ソケットがサポートされていない場合に NCA を使用するための手順。	55 ページの「NCA 用のソケットユーティリティーライブラリを読み込む方法」
Apache 2.0 Web サーバーで SSL カーネルプロキシを使用する	特定の Web サーバーで SSL カーネルプロキシを使って SSL パケット処理を改善するための手順。	56 ページの「SSL カーネルプロキシを使用するように Apache 2.0 Web サーバーを設定する方法」
Sun Java System Web Server で SSL カーネルプロキシを使用する	特定の Web サーバーで SSL カーネルプロキシを使って SSL パケット処理を改善するための手順。	58 ページの「SSL カーネルプロキシを使用するように Sun Java System Web Server を設定する方法」
局所ゾーン内の Web サーバーで SSL カーネルプロキシを使用する	局所ゾーン内の Web サーバーで SSL カーネルプロキシを使用するための手順。	60 ページの「ゾーン内での SSL カーネルプロキシの使用」

NCA の利用を計画する

この節では、NCA サービスを開始する前に解決しておく必要のある事項について説明します。

NCA を使用するためのシステム要件

NCA をサポートするには、システムは次の要件を満たす必要があります。

- 256M バイトの RAM がインストールされている。
- Solaris 10、9 リリース、または Solaris 8 アップグレードリリースのいずれかがインストールされている。
- NCA のソケットユーティリティーライブラリを使用するように起動スクリプトが変更されている NCA または Web サーバーを直接サポートする、次の Web サーバーをサポートしている。
  - Apache Web サーバー。Solaris 8 アップグレード、Solaris 9、および Solaris 10 リリースに同梱されています
  - Sun Java System Web Server
  - Zeus Technology の Zeus Web サーバー、`http://www.zeus.com`

この製品は、専用の Web サーバー上で実行するようにします。NCA を実行しているサーバー上で別の大きいプロセスを実行すると、問題が生じることがあります。

NCA ロギング

NCA サービスでは、Web アクティビティーを記録するように設定できます。通常、Web サーバーのロギングが有効になっているときには NCA のロギングも有効にします。

ライブラリ置き換えによる door サーバーデーモンのサポート

多くの Web サーバーが AF_INET ソケットを使用しています。デフォルトでは、NCA は AF_NCA ソケットを使用します。この状況に対応するために、置き換え用のライブラリが用意されています。新しいライブラリは標準ソケットライブラリ `libsocket.so` の前にロードされます。ライブラリ呼び出し `bind()` は、新しいライブラリ `ncad_addr.so` によって置き換えられます。`/etc/nca/ncakmod.conf` 内で状態が有効に設定されているとします。Solaris 9 および Solaris 10 リリースに付属している Apache は、このライブラリを呼び出すように設定されています。IWS または Netscape サーバーで新しいライブラリを使用する場合は、55 ページの「NCA 用のソケットユーティリティーライブラリを読み込む方法」を参照してください。

複数インスタンスのサポート

NCA がインストールされているシステムでは、複数の Web サーバーを実行することがよくあります。たとえば、1 つのサーバーで、外部からのアクセス用の Web サーバーと Web 管理サーバーの両方をサポートする場合があります。これらのサーバーを別にするには、それぞれのサーバーが別のポートを使用するように設定します。

Web ページのキャッシュ管理 (手順)

この節では、サービスを有効または無効にするための手順を示します。

▼ Web ページのキャッシングを有効にする方法

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 インタフェースを登録します。

`/etc/nca/nca.if` ファイルに各物理インタフェースの名前を指定します。詳細は、`nca.if(4)` のマニュアルページを参照してください。

```
# cat /etc/nca/nca.if
hme0
hme1
```

インタフェースごとに、対応する `hostname.` *interface-name* ファイルが必要です。また、`/etc/hosts` ファイル内に `hostname.`*interface-name* の内容と一致するエントリが必要です。すべてのインタフェースで NCA 機能を使用可能にするには、`nca.if` ファイル内でアスタリスク (*) を指定します。

3 `ncakmod` カーネルモジュールを有効にします。

`/etc/nca/ncakmod.conf` 内の `status` エントリを `enabled` に変更します。

```
# cat /etc/nca/ncakmod.conf
#
# NCA Kernel Module Configuration File
#
status=enabled
httpd_door_path=/var/run/nca_httpd_1.door
nca_active=disabled
```

詳細は、`ncakmod.conf(4)` のマニュアルページを参照してください。

4 **(省略可能) NCA ロギングを有効にします。**

/etc/nca/ncalogd.conf 内の status エントリを enabled に変更します。

```
# cat /etc/nca/ncalogd.conf
#
# NCA Logging Configuration File
#
status=enabled
logd_path_name="/var/nca/log"
logd_file_size=1000000
```

logd_path_name エントリに示されているパスを変更すると、ログファイルの格納場所を変更できます。ログファイルには raw デバイスとファイルのどちらでも指定できます。次に、NCA ログファイルのパスの例を示します。この設定ファイルの詳細は、ncalogd.conf(4) のマニュアルページを参照してください。

5 **(省略可能) 複数インスタンスのサポートのためのポートを定義します。**

/etc/nca/ncaport.conf ファイルにポート番号を追加します。次の例では、NCA はすべての設定済み IP アドレスについて、ポート 80 を監視します。

```
# cat /etc/nca/ncaport.conf
#
# NCA Kernel Module Port Configuration File
#
  .
  .
ncaport=*/80
```

6 **x86 のみ: 仮想メモリーサイズを増やします。**

eeprom コマンドを使用して、システムの kernelbase を設定します。

```
# eeprom kernelbase=0x90000000
# eeprom kernelbase
kernelbase=0x90000000
```

2 行目の eeprom コマンドを実行すると、パラメータが設定済みかどうかを確認できます。

注 – kernelbase を設定すると、ユーザープロセスが使用できる仮想メモリー領域が 3G バイト未満に減少します。このため、システムは ABI に準拠しなくなります。システムをブートすると、そのことを警告するメッセージがコンソールに表示されます。ほとんどのプログラムは、実際には 3G バイトの仮想アドレス空間を必要としません。3G バイト以上の仮想アドレス空間を必要とするプログラムは、NCA を無効に設定したシステム上で実行する必要があります。

---

7 **サーバーを再起動します。**

例 2–1 NCA ログファイルとして raw デバイスを使用する

ncalogd.conf ファイル内の logd_path_name 文字列で、NCA ログファイルの格納先として raw デバイスを指定できます。raw デバイスを使用する利点としては、アクセス時のオーバーヘッドが小さいため、サービスを高速に実行できることが挙げられます。

NCA サービスはファイル内に記述されているすべての raw デバイスに対して、対応するファイルシステムがないことを確認します。このテストは、アクティブなファイルシステムを誤って上書きしてしまわないように実行されます。

このテストでファイルシステムが検出されないようにするには次のコマンドを実行します。このコマンドは、ファイルシステムとして構成されている任意のディスクパーティション上のファイルシステム部分を破棄します。この例では、/dev/rdsk/c0t0d0s7 が古いファイルシステムを持つ raw デバイスです。

```
# dd if=/dev/zero of=/dev/rdsk/c0t0d0s7 bs=1024 count=1
```

上記の dd コマンドを実行すると、ncalogd.conf ファイルに raw デバイスを追加できるようになります。

```
# cat /etc/nca/ncalogd.conf
#
# NCA Logging Configuration File
#
status=enabled
logd_path_name="/dev/rdsk/c0t0d0s7"
logd_file_size=1000000
```

例 2–2 NCA ロギング用に複数のファイルを使用する

ncalogd.conf ファイル内の logd_path_name 文字列で、NCA ログファイルの格納先として複数のファイルを指定できます。最初のファイルが満杯になると、二番目のファイルが使用されます。次の例では、最初に /var/nca/log ファイルを書き込みに使用し、次に raw パーティションを使用する方法を示します。

```
# cat /etc/nca/ncalogd.conf
#
# NCA Logging Configuration File
#
status=enabled
logd_path_name="/var/nca/log /dev/rdsk/c0t0d0s7"
logd_file_size=1000000
```

▼ Web ページのキャッシングを無効にする方法

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 ncakmod カーネルモジュールを無効にします。

/etc/nca/ncakmod.conf 内の status エントリを disabled に変更します。

```
# cat /etc/nca/ncakmod.conf
# NCA Kernel Module Configuration File
#
status=disabled
httpd_door_path=/var/run/nca_httpd_1.door
nca_active=disabled
```

詳細は、ncakmod.conf(4) のマニュアルページを参照してください。

3 NCA ロギングを無効にします。

/etc/nca/ncalogd.conf 内の status エントリを disabled に変更します。

```
# cat /etc/nca/ncalogd.conf
#
# NCA Logging Configuration File
#
status=disabled
logd_path_name="/var/nca/log"
logd_file_size=1000000
```

詳細は、ncalogd.conf(4) のマニュアルページを参照してください。

4 サーバーを再起動します。

▼ NCA ロギングを有効または無効にする方法

NCA が有効になっている場合、必要に応じて NCA のログ処理のオン / オフを切り換えることができます。詳細は、51 ページの「Web ページのキャッシングを有効にする方法」を参照してください。

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 NCA ロギングのオン/オフを切り換えます。

ロギングを恒久的に無効にする場合は、/etc/nca/ncalogd.conf 内の status を disabled に変更し、システムをリブートする必要があります。詳細は、ncalogd.conf(4) のマニュアルページを参照してください。

a. ロギングを停止します。

```
# /etc/init.d/ncalogd stop
```

b. ロギングを開始します。

```
# /etc/init.d/ncalogd start
```

NCA 用のソケットユーティリティーライブラリを読み込む方法

この手順は、AF_NCA ソケットを直接にサポートしていない Web サーバーに対してのみ使用します。

Web サーバーの起動スクリプトに、ライブラリをプリロードするための 1 行を追加します。次のような行を追加します。

```
LD_PRELOAD=/usr/lib/ncad_addr.so /usr/bin/httpd
```

▼ NCA サービスに新しいポートを追加する方法

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 新しいポートを追加します。

/etc/nca/ncaport.conf に、新しいポートのエントリを追加します。次の例では、IP アドレス 192.168.84.71 に対してポート 8888 を追加しています。詳細は、ncaport.conf(4) のマニュアルページを参照してください。

```
# cat /etc/nca/ncaport.conf
#
# NCA Kernel Module Port Configuration File
#
  .
  .
ncaport=*/80
ncaport=192.168.84.71/8888
```

3 新しい Web インスタンスを起動します。

Web サーバーが NCA でアドレスを使用するには、先にそのアドレスが NCA のポート設定のファイルに入っている必要があります。Web サーバーが実行中である場合は、新しいアドレスの定義後にその Web サーバーを再起動する必要があります。

▼ SSL カーネルプロキシを使用するように Apache 2.0 Web サーバーを設定する方法

Apache 2.0 Web サーバー上で SSL パケット処理のパフォーマンスを改善するには、次の手順を使用してください。

始める前に 次の手順を使用するには、Apache 2.0 Web サーバーのインストールと設定が完了している必要があります。Apache 2.0 Web サーバーは、Solaris 10 リリースに含まれています。

SSL カーネルプロキシを使用するには、サーバーの非公開鍵と証明書が単一のファイル内に存在している必要があります。`ssl.conf` ファイル内に `SSLCertificateFile` パラメータだけが指定されている場合、そこに指定されたファイルを直接、カーネル SSL 用として使用できます。`SSLCertificateKeyFile` パラメータも指定されている場合、証明書ファイルと非公開鍵ファイルを 1 つにまとめる必要があります。証明書ファイルと鍵ファイルを 1 つにまとめる方法の 1 つは、次のコマンドを実行することです。

```
# cat cert.pem key.pem >cert-and-key.pem
```

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。`ksslcfg` コマンドは、`Network Security` プロファイルに含まれています。

2 Web サーバーを停止します。

次のコマンドは、Web サーバーが SMF を使って実行されるように設定されているシステム上の Web サーバーを停止します。

```
# svcadm disable svc:/network/http:apache2
```

サービスがまだ変換されていない場合には、次のコマンド構文を使ってサービスを停止します。 `/usr/apache2/bin/apachectl stop`

3 `ksslcfg` コマンドで使用するパラメータを決定します。

オプションの完全な一覧については、ksslcfg(1M)情報を指定する必要のあるパラメータは、次のとおりです。

- key-format – -f オプションとともに使用し、証明書と鍵の形式を定義します。SSL カーネルプロキシの場合、pem、pkcs12 のいずれかの値を指定してください。
- key-and-certificate-file – -i オプションとともに使用し、サーバーの鍵と証明書の格納先となるファイルの場所を設定します。
- password-file– -p オプションとともに使用し、非公開鍵の暗号化に使用するパスワードが含まれているファイルの場所を選択します。このパスワードは、自動再起動を実現するために使用されます。このファイルのアクセス権は、**0400** にしてください。
- proxy-port – -x オプションとともに使用し、SSL プロキシポートを設定します。標準ポート **80** とは別のポートを選択します。Web サーバーは SSL プロキシポートを待機します。
- ssl-port – SSL カーネルプロキシが待機するポートを選択します。これは通常、**443** に設定されます。

注 – ssl-port と proxy-port の値を NCA 用として設定することはできません。なぜなら、これらのポートは SSL カーネルプロキシ専用として使用されるからです。通常、ポート 80 が NCA 用として、ポート 8443 が proxy-port 用として、443 が ssl-port 用として、それぞれ使用されます。

---

4 サービスインスタンスを作成します。

ksslcfg コマンドで、SSL プロキシポートと関連パラメータを指定します。

```
ksslcfg create -f key-format -i key-and-certificate-file -p password-file -x proxy-port ssl-port
```

5 インスタンスが正しく作成されたことを確認します。

次のコマンドによって報告されるサービスの状態は「online」です。

```
# svcs svc:/network/ssl/proxy
```

6 **SSL** プロキシポート上で待機するように **Web** サーバーを設定します。

/etc/apache2/http.conf ファイルを編集し、SSL プロキシポートを定義するための行を 1 行追加します。サーバーの IP アドレスを使用した場合、Web サーバーはそのインタフェース上でのみ待機します。この行は次のようになります。

```
Listen 0.0.0.0:proxy-port
```

7 **Web** サーバーの **SMF** 依存関係を設定します。

SSL カーネルプロキシインスタンスの起動後に Web サーバーが起動されるようにすべきです。次のコマンドは、そうした依存関係を確立します。

```
# svccfg -s svc:/network/http:apache2
svc:/network/http:apache2> addpg kssl dependency
svc:/network/http:apache2> setprop kssl/entities = fmri:svc:/network/ssl/proxy:kssl-INADDR_ANY-443
```

```
svc:/network/http:apache2> setprop kssl/grouping = astring: require_all
svc:/network/http:apache2> setprop kssl/restart_on = astring: refresh
svc:/network/http:apache2> setprop kssl/type = astring: service
svc:/network/http:apache2> end
```

8 Web サーバーを有効にします。

```
# svcadm enable svc:/network/http:apache2
```

SMF を使ってサービスが起動されない場合は、次のコマンドを使用します。

```
/usr/apache2/bin/apachectl startssl
```

例 2–3 SSL カーネルプロキシを使用するように Apache 2.0 Web サーバーを設定する

次のコマンドは、pem 鍵形式を使ってインスタンスを作成します。

```
# ksslcfg create -f pem -i cert-and-key.pem -p file -x 8443 443
```

▼ SSL カーネルプロキシを使用するように Sun Java System Web Server を設定する方法

Sun Java System Web Server 上で SSL パケット処理のパフォーマンスを改善するには、次の手順を使用してください。この Web サーバーについては、『Sun Java System Web Server 6.1 2005Q4 SP4 管理者ガイド』を参照してください。

始める前に 次の手順を使用するには、Sun Java System Web Server のインストールと設定が完了している必要があります。

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。ksslcfg コマンドは、Network Security プロファイルに含まれています。

2 Web サーバーを停止します。

管理者の Web インタフェースを使ってサーバーを停止します。詳細は、『Sun Java System Web Server 6.1 2005Q4 SP4 管理者ガイド』の「サーバーの起動と停止」を参照してください。

3 暗号化フレームワークのメタスロットを無効にします。

この手順は、カーネル SSL サービスのインスタンスが作成されるときに確実にメタスロットが無効になるようにするために必要です。

```
# cryptoadm disable metaslot
```

4 ksslcfg コマンドで使用するパラメータを決定します。

オプションの完全な一覧については、ksslcfg(1M)情報を指定する必要のあるパラメータは、次のとおりです。

- key-format – -f オプションとともに使用し、証明書と鍵の形式を定義します。
- token-label – -T オプションとともに使用し、PKCS#11 トークンを指定します。
- certificate-label – -C オプションとともに使用し、PKCS#11 トークン内の証明書オブジェクトに含まれるラベルを選択します。
- password-file – -p オプションとともに使用し、Web サーバーが使用する PKCS#11 トークンにユーザーをログインさせるためのパスワードが含まれているファイルの場所を選択します。このパスワードは、自動再起動を実現するために使用されます。このファイルのアクセス権は、0400 にしてください。
- proxy-port – -x オプションとともに使用し、SSL プロキシポートを設定します。標準ポート 80 とは別のポートを選択します。Web サーバーは SSL プロキシポートを待機します。
- ssl-port – SSL カーネルプロキシが待機するポートを定義します。この値は通常、443 に設定されます。

注 – ssl-port と proxy-port の値を NCA 用として設定することはできません。なぜなら、これらのポートは SSL カーネルプロキシ専用として使用されるからです。通常、ポート 80 が NCA 用として、ポート 8443 が proxy-port 用として、443 が ssl-port 用として、それぞれ使用されます。

---

5 サービスインスタンスを作成します。

ksslcfg コマンドで、SSL プロキシポートと関連パラメータを指定します。

```
ksslcfg create -f key-format -T PKCS#11-token -C certificate-label -p password-file -x proxy-port ssl-port
```

6 暗号化フレームワークのメタスロットを有効にします。

```
# cryptoadm enable metaslot
```

7 インスタンスが正しく作成されたことを確認します。

次のコマンドによって報告されるサービスの状態は「online」です。

```
# svcs svc:/network/ssl/proxy
```

8 SSL プロキシポート上で待機するように Web サーバーを設定します。

詳細は、『Sun Java System Web Server 6.1 2005Q4 SP4 管理者ガイド』の「待機ソケットの追加と編集」を参照してください。

9 Web サーバーを起動します。

例 2–4 SSL カーネルプロキシを使用するように Sun Java System Web Server を設定する

次のコマンドは、pkcs11 鍵形式を使ってインスタンスを作成します。

```
# ksslcfg create -f pkcs11 -T "Sun Software PKCS#11 softtoken" -C "Server-Cert" -p file -x 8443 443
```

ゾーン内での SSL カーネルプロキシの使用

SSL カーネルプロキシはゾーン内でも動作しますが、その際には次の制限があります。

- カーネル SSL の管理はすべて、大域ゾーンから行う必要があります。大域ゾーンの管理者は、局所ゾーン内の証明書や鍵のファイルにアクセスできる必要があります。大域ゾーンでの ksslcfg コマンドによるサービスインスタンスの設定が完了すると、局所ゾーンで Web サーバーを起動できるようになります。
- ksslcfg コマンドを実行してインスタンスを設定する際に、特定のホスト名または IP アドレスを指定する必要があります。特に、インスタンスは INADDR_ANY を使用できません。

例 2–5 SSL カーネルプロキシを使用するように局所ゾーン内の Apache Web サーバーを設定する

まず、局所ゾーン内で Web サーバーを停止します。大域ゾーン内で、サービスを設定するための手順をすべて実行します。apache-zone という名前の局所ゾーンに対するインスタンスを作成するには、次のコマンドを使用します。

```
# ksslcfg create -f pem -i /zone/apache-zone/root/keypair.pem -p /zone/apache-zone/root/pass \
    -x 8443 apache-zone 443
```

局所ゾーン内で、次のコマンドを実行してサービスインスタンスを有効にします。

```
# svcadm enable svc:/network/http:apache2
```

Web ページのキャッシング (リファレンス)

この節では、NCA を使用するために必要なファイルとコンポーネントについて説明します。また、NCA が Web サーバーと通信する方法についても説明します。

NCA ファイル

NCA 機能をサポートするには、いくつかのファイルが必要です。ほとんどのファイルは ASCII 形式ですが、バイナリ形式のファイルもあります。次の表に必要なファイルの一覧を示します。

表 2–1 NCA ファイル

ファイル名	機能
/dev/nca	NCA デバイスのパス名。
/etc/hostname. *	サーバー上で構成されているすべての物理インタフェースについてホスト名が記述されているファイル。
/etc/hosts	サーバーに対応付けられるすべてのホスト名が記述されているファイル。NCA が機能するには、このファイルの各エントリが、対応する /etc/hostname. * ファイル内のエントリと一致していなければなりません。
/etc/init.d/ncakmod	NCA サーバーを起動するスクリプト。このスクリプトは、サーバーのブート時に実行されます。
/etc/init.d/ncalogd	NCA ロギングを開始するスクリプト。このスクリプトは、サーバーのブート時に実行されます。
/etc/nca/nca.if	NCA が実行されるすべてのインタフェースが記述されているファイル。詳細は、nca.if(4) のマニュアルページを参照してください。
/etc/nca/ncakmod.conf	NCA 用のすべての構成パラメータが記述されているファイル。詳細は、ncakmod.conf(4) のマニュアルページを参照してください。
/etc/nca/ncalogd.conf	NCA ロギング用のすべての構成パラメータが記述されているファイル。詳細は、ncalogd.conf(4) のマニュアルページを参照してください。
/etc/nca/ncaport.conf	NCA で使用する IP アドレスとポートが記述されているファイル。詳細は、ncaport.conf(4) のマニュアルページを参照してください。
/usr/bin/ncab2clf	ログファイル内のデータを共通ログ形式に変換するために使用されるコマンド。詳細は、ncab2clf(1) のマニュアルページを参照してください。
/usr/lib/net/ncaconfd	ブート時に複数のインタフェース上で NCA が実行するように設定するために使用されるコマンド。詳細は、ncaconfd(1M) のマニュアルページを参照してください。

表 2–1 NCA ファイル (続き)

ファイル名	機能
/usr/lib/nca_addr.so	AF_INET ソケットの代わりに AF_NCA ソケットを使用するライブラリ。このライブラリは AF_INET ソケットを使用する Web サーバー上で使用します。詳細は、ncad_addr(4) のマニュアルページを参照してください。
/var/nca/log	ログファイルのデータを保持するファイル。バイナリ形式のファイルなので編集できません。
/var/run/nca_httpd_1.door	ドアパス名。

NCA アーキテクチャー

NCA が機能するためには、次のコンポーネントが必要です。

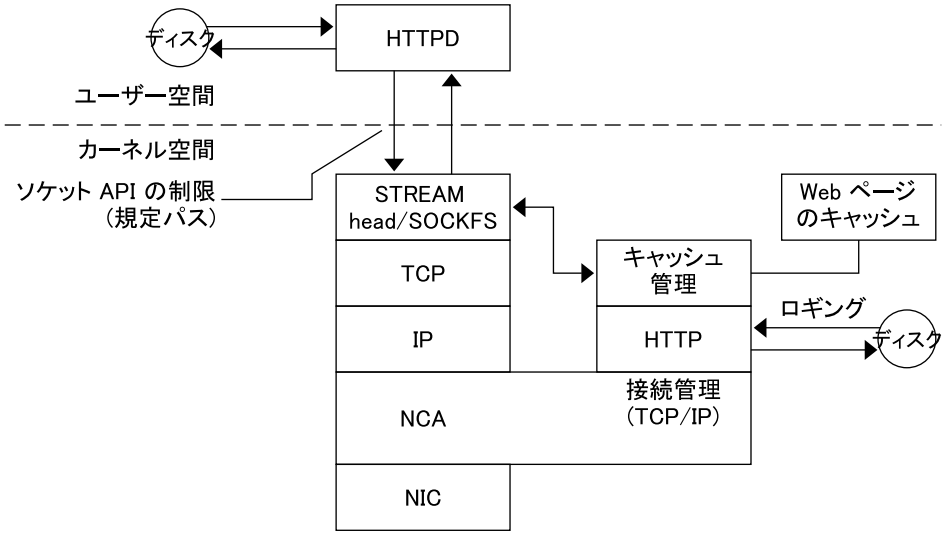
- カーネルモジュール: ncakmod
- Web サーバー: httpd

カーネルモジュール ncakmod は、Web ページのキャッシュをシステムメモリー内に保持します。このモジュールは、ソケットインタフェースを介して Web サーバー httpd と通信します。プロトコルファミリータイプは PF_NCA です。

また、カーネルモジュールは、すべての HTTP キャッシュヒットを記録するログ機能も備えています。NCA ロギングは、HTTP データをバイナリ形式でディスクに書き込みます。NCA には、バイナリログファイルを共通ログ形式 (CLF) に変換するユーティリティーが用意されています。

次の図に、通常のデータフローと、NCA が有効になっている場合のデータフローを示します。

図 2–1　NCA サービスのデータフロー



NCA から httpd への要求フロー

次に、クライアントと Web サーバー間の要求フローを示します。

1. クライアントから Web サーバーに対して HTTP 要求が発行されます。
2. ページがキャッシュ内にある場合は、カーネル内キャッシュの Web ページが返されます。
3. ページがキャッシュ内にない場合は、Web サーバーに要求が送信され、ページが取得または更新されます。
4. ページがキャッシュされているかどうかは、HTTP 応答で使用される HTTP プロトコルのセマンティクスによって異なります。そのあと、ページがクライアントに返されます。HTTP 要求ヘッダーに Pragma:No-cache が含まれている場合、ページはキャッシュされません。

◆ ◆ ◆ 第 3 章

3 システムの時刻関連サービス

多くのデータベースと認証サービスでは、ネットワーク内でシステムクロックを同期させる必要があります。この章の内容は次のとおりです。

時刻の同期 (概要)

Solaris 2.6 以降、Solaris ソフトウェアには Delaware 大学の NTP (Network Time Protocol) 公開ドメインソフトウェアが添付されています。`xntpd` デーモンは、UNIX システムの時刻をインターネット標準時刻サーバーの時刻と合うように調整し、保守します。`xntpd` デーモンは、RFC 1305 に規定されている NTP version 3 標準に完全に準拠して実装されています。

`xntpd` デーモンは、システムの起動時に `/etc/inet/ntp.conf` ファイルを読み込みます。構成オプションの詳細は、`xntpd(1M)` のマニュアルページを参照してください。

ネットワーク内で NTP を使用するときには、次のことを考慮してください。

- `xntpd` デーモンは最小限のシステム資源しか使用しません。
- NTP クライアントは起動時に、自動的に NTP サーバーと同期を取ります。クライアントは同期の取れていない状態になった場合、タイムサーバーと通信したときに再同期を取ります。

`cron` を使用して `rdate` コマンドを実行することにより、時刻の同期を取ることもできます。

NTP の管理 (作業)

NTP サービスを設定および使用するための手順を示します。

▼ NTP サーバーを設定する方法

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 `ntp.conf` ファイルを作成します。

`xntpd` デーモンを正しく実行するには、最初に `ntp.conf` ファイルを作成する必要があります。`ntp.server` ファイルをテンプレートとして使用できます。

```
# cd /etc/inet
# cp ntp.server ntp.conf
```

3 `xntpd` デーモンを起動します。

```
# svcadm enable network/ntp
```

▼ NTP クライアントを設定する方法

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 `ntp.conf` ファイルを作成します。

`xntpd` デーモンを有効にするには、最初に `ntp.conf` ファイルを作成する必要があります。

```
# cd /etc/inet
# cp ntp.client ntp.conf
```

3 `xntpd` デーモンを起動します。

```
# svcadm enable network/ntp
```

他の時刻関連コマンドの使用 (作業)

次の手順を使用すると、NTP を設定しなくても、必要に応じて現在の時刻を更新できます。

▼ 他のシステムの日時と同期させる方法

1. スーパーユーザーになるか、同等の役割を引き受けます。

   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2. `rdate` コマンドを使用して、日付と時刻を設定し直し、他のシステムとの同期を取ります。

   ```
   # rdate another-system
   ```

   another-system　　ほかのシステムの名前

3. `date` コマンドを使用して、システムの日時が正しく設定し直されていることを確認してください。

   出力は、指定したシステムと同じ日付と時刻を示します。

例 3–1 他のシステムの日時と同期させる方法

次の例は、`rdate` を使用してシステムの日時を別のシステムの日時と同期させる方法を示します。次の例は、数時間遅れていたシステム `earth` の日付と時刻をサーバー `starbug` の日付と時刻に一致させます。

```
earth# date
Tue Jun  5 11:08:27 MDT 2001
earth# rdate starbug
Tue Jun  5 14:06:37 2001
earth# date
Tue Jun  5 14:06:40 MDT 2001
```

NTP (リファレンス)

NTP サービスを実行するには、次のファイルが必要です。

表 3–1 NTP ファイル

ファイル名	機能
/etc/inet/ntp.conf	NTP 用のすべての構成オプションが記述されているファイル。
/etc/inet/ntp.client	NTP クライアント用のサンプル構成ファイル。
/etc/inet/ntp.server	NTP サーバー用のサンプル構成ファイル。
/etc/inet/ntp.keys	NTP 認証キーを含むファイル。
/usr/lib/inet/xntpd	NTP デーモン。詳細は、xntpd(1M) のマニュアルページを参照してください。
/usr/sbin/ntpdate	NTP に基づいてローカルな日付と時刻を設定するユーティリティー。詳細は、ntpdate(1M) のマニュアルページを参照してください。
/usr/sbin/ntpq	NTP 照会プログラム。詳細は、ntpq(1M) のマニュアルページを参照してください。
/usr/sbin/ntptrace	マスターの NTP サーバーまで NTP ホストを追跡するプログラム。詳細は、ntptrace(1M) のマニュアルページを参照してください。
/usr/sbin/xntpdc	xntpd デーモン用の NTP 照会プログラム。詳細は、xntpdc(1M) のマニュアルページを参照してください。
/var/ntp/ntpstats	NTP の統計情報を保持するディレクトリ。
/var/ntp/ntp.drift	NTP サーバー上で初期周波数オフセットを設定するファイル。

パート II

ネットワークファイルシステムへのアクセス(トピック)

このパートでは、NFS サービスの概要、作業、およびリファレンス情報について説明します。

◆ ◆ ◆ 第 4 章

ネットワークファイルシステムの管理 (概要)

この章では、ネットワーク経由でファイルシステムにアクセスするために使用する NFS サービスの概要を説明します。また、NFS サービスを理解するために必要な概念、および NFS と autofs の最新の機能についても説明します。

注 – システムでゾーンが有効なときに非大域ゾーンでこの機能を使用するには、『Oracle Solaris のシステム管理 (Oracle Solaris コンテナ : 資源管理と Oracle Solaris ゾーン)』を参照してください。

NFS サービスの新機能

この節では、Solaris OS の各リリースの新機能に関する情報を提供します。

Solaris 10 11/06 リリースでの変更点

Solaris 10 11/06 リリースは、ファイルシステム監視ツールをサポートします。次を参照してください。

- 説明や例については、158 ページの「fsstat コマンド」
- 詳細は、fsstat(1M) のマニュアルページ

また、このマニュアルには、nfsmapid デーモンの詳細も記載されています。nfsmapid の詳細は、次を参照してください。

- 148 ページの「nfsmapid デーモン」
- nfsmapid(1M) のマニュアルページ

新機能の完全な一覧については、『Oracle Solaris 10 9/10 の新機能』を参照してください。

Solaris 10 リリースでの変更点

Solaris 10 以降のリリースでは、NFS のデフォルトは version 4 です。NFS version 4 の機能とその他の変更については、次を参照してください。

- 112 ページの「CacheFS を使用して NFS ファイルシステムにアクセスする」
- 141 ページの「/etc/default/autofs ファイル」
- 142 ページの「/etc/default/nfs ファイルのキーワード」
- 146 ページの「lockd デーモン」
- 147 ページの「nfs4cbd デーモン」
- 148 ページの「nfsmapid デーモン」
- 160 ページの「NFS ファイルシステム用の mount オプション」
- 181 ページの「RDMA 経由の NFS」
- 183 ページの「NFS におけるバージョンのネゴシエーション」
- 184 ページの「NFS version 4 における機能」
- 218 ページの「autofs がクライアント用のもっとも近い読み取り専用ファイルを選択する方法 (複数ロケーション)」

また、次も参照してください。

- 作業については、94 ページの「NFS サービスの設定」
- 新機能の完全な一覧については、『Oracle Solaris 10 9/10 の新機能』

さらに、NFS サービスは、サービス管理機能で管理されます。このサービスに関する有効化、無効化、再起動などの管理アクションは svcadm コマンドを使用して実行できます。サービスの状態は、svcs コマンドを使用して照会できます。サービス管理機能の詳細は、smf(5) のマニュアルページおよび『Solaris のシステム管理 (基本編)』の第 18 章「サービスの管理 (概要)」を参照してください。

NFS の用語

ここでは、NFS サービスを使用するために必要な基本用語について説明します。NFS サービスの詳細は、第 6 章「ネットワークファイルシステムへのアクセス (リファレンス)」で説明します。

NFS サーバーとクライアント

「クライアント」と「サーバー」という用語は、コンピュータがファイルシステムを共有するときの役割を示すものです。ネットワークを介してファイルシステムを提供するコンピュータは、サーバーの役割を果たします。そのファイルシステムにアクセスしているコンピュータをクライアントと呼びます。NFS を使用することによって、どのコンピュータからも他のコンピュータのファイルシステムにアクセスでき、同時に自分のファイルシステムへのアクセスも可能になります。ネットワーク上では 1 台のコンピュータがクライアントかサーバー、またはその両方の役割として動作することができます。

クライアントは、サーバーの共有ファイルシステムをマウントすることによってサーバーのファイルにアクセスします。クライアントがリモートファイルシステムをマウントしたとき、ファイルシステムがコピーされるのではありません。マウントプロセスでは一連の遠隔手続き呼び出しによって、クライアントからサーバーのディスク上にあるファイルシステムに透過的にアクセスできるようになります。マウントはローカルマウントのように行われます。ユーザーはファイルシステムがローカルにあるのと同じようにコマンドを入力します。ファイルシステムをマウントする方法については、88 ページの「ファイルシステムのマウント」を参照してください。

サーバーのファイルシステムは、NFS オペレーションによって共有すると、クライアントからアクセスできるようになります。NFS ファイルシステムは、autofs を使用すると自動的にマウントできます。share コマンドと autofs に関連する作業については、84 ページの「ファイルシステムの自動共有」 and 105 ページの「autofs 管理作業の概要」を参照してください。

NFS ファイルシステム

NFS サービスで共有できるオブジェクトは、ディレクトリツリー、つまり、あるファイル階層の全体またはその一部であり、これにはファイルが 1 つのみの場合も含まれます。すでに共有しているファイル階層と重複するファイル階層は共有できません。モデムやプリンタなどの周辺機器も共有できません。

多くの UNIX システム環境で共有されるファイル階層構造は、1 つのファイルシステム、またはその一部です。しかし NFS サポートは複数のオペレーティングシステムにまたがって動作しますが、ファイルシステムという概念は UNIX 以外の環境では

意味がないかもしれません。したがって、「ファイルシステム」という語は、NFS での共有およびマウントが可能なファイルまたはファイル階層構造を指します。

NFS サービスについて

NFS サービスとは、アーキテクチャーが異なり、別のオペレーティングシステムで動作しているコンピュータが、ネットワークを通じてファイルシステムを共有できるようにするサービスのことです。NFS サポートは、MS-DOS から VMS オペレーティングシステムまで多くのプラットフォームに実装されています。

NFS 環境は、異なるオペレーティングシステムで実現できます。NFS はアーキテクチャーの仕様を定義するのではなく、ファイルシステムの抽象モデルを定義しているためです。それぞれのオペレーティングシステムでは、ファイルシステムセマンティクスに NFS 抽象モデルを適用します。このモデルにより、書き込みや読み取りのようなファイルシステムオペレーションが、ローカルファイルにアクセスするように機能することになります。

NFS サービスには次の利点があります。

- 複数のコンピュータで同一のファイルを使用するため、ネットワーク上のだれもが同じデータにアクセスできる
- 各ユーザーアプリケーションがローカルのディスクスペースを占めるのではなく、複数のコンピュータでアプリケーションを共有するため、記憶領域を有効利用できる
- すべてのユーザーが同一セットのファイルを読み取るので、データの整合性と信頼性が向上する
- ファイルシステムをユーザーに透過的な形でマウントできる
- リモートファイルに透過的にアクセスできる
- さまざまな環境をサポートする
- システム管理の手間を省ける

NFS サービスを使用すると、ファイルシステムの実際の場所をユーザーとは無関係に決めることができます。ユーザーは場所を気にすることなく、すべての適切なファイルにアクセスできるということです。NFS サービスでは、共通して使用するファイルのコピーをすべてのシステムに置くのではなく、コピーを 1 つのコンピュータのディスクに置き、他のシステムからネットワークを通じてアクセスできるようにします。NFS オペレーションでは、リモートファイルシステムとローカルファイルシステムの区別がありません。

autofs について

NFS サービスで共有されるファイルシステムは、「自動マウント」と呼ばれる方法によってマウントできます。クライアント側のサービスである autofs は、自動マウントを実現するファイルシステム構造です。autofs のファイルシステムは、automount で作成されます。automount は、システムをブートすると自動的に実行されます。automountd という常駐型の自動マウントデーモンが、必要に応じてリモートディレクトリのマウントとアンマウントを行います。

automountd を実行しているクライアントコンピュータがリモートのファイルまたはディレクトリにアクセスしようとすると、リモートのファイルシステムがデーモンによってマウントされます。このリモートファイルシステムは、必要な間はマウントされたままです。リモートファイルシステムが一定時間アクセスされないと、自動的にアンマウントされます。

ブート時にマウントする必要はなく、ユーザーはディレクトリをマウントするためにスーパーユーザーのパスワードを知る必要はありません。ユーザーが mount と umount コマンドを使用する必要もありません。autofs は、ユーザーの介入なしに、必要に応じてファイルシステムをマウントまたはアンマウントします。

automountd によって一部のファイル階層をマウントするということは、mount によって他の階層をマウントしないということではありません。ディスクレスコンピュータは、mount コマンドと /etc/vfstab ファイルを使用して / (ルート)、/usr、および /usr/kvm をマウントしなければなりません。

autofs サービスについては、105 ページの「autofs 管理作業の概要」 と 213 ページの「autofs のしくみ」で詳しく説明します。

NFS サービスの機能

ここでは、NFS サービスの重要な機能について説明します。

NFS version 2 プロトコル

version 2 は、一般に広く使用された初めての NFS プロトコルです。version 2 は、引き続き広範囲のプラットフォームで使用できます。Solaris のすべてのリリースが NFS プロトコルの version 2 をサポートし、Solaris 2.5 より以前のリリースは version 2 だけをサポートします。

NFS version 3 プロトコル

NFS version 3 のプロトコルは、Solaris 2.5 で新機能として追加されたものです。相互運用性とパフォーマンスを向上させるために、いくつかの変更が行われました。これらをすべて有効に利用するには、NFS サーバーとクライアントの両方で、version 3 プロトコルを使用する必要があります。

NFS version 2 プロトコルとは異なり、NFS version 3 プロトコルは 2G バイト以上のファイルを扱えます。以前の制限はなくなりました。79 ページの「NFS 大規模ファイルのサポート」を参照してください。

NFS version 3 では、サーバーで非同期の書き込みが可能になります。サーバーがクライアントの書き込み要求をメモリーに保存するので、効率が向上しました。クライアントは、サーバーが変更内容をディスクに反映させるのを待つ必要がないため、応答時間が短縮されます。サーバーは要求をバッチ処理することもできるので、サーバー上の応答時間も短縮されました。

Solaris NFS version 3 の多くの操作では、ローカルキャッシュに保存されているファイル属性が返されます。キャッシュの更新頻度が増えたため、ローカルキャッシュのデータを更新する操作を独立して行う必要性が少なくなります。したがってサーバーに対する RPC コールの回数が減少し、パフォーマンスが向上します。

ファイルアクセス権の確認処理も改善されました。version 2 では、ユーザーが適切なアクセス権を持っていないリモートファイルをコピーしようとすると、「書き込みエラー」や「読み取りエラー」というメッセージが出力されました。version 3 では、ファイルを開く前にアクセス権がチェックされるため、「オープンエラー」というメッセージが出力されます。

NFS version 3 プロトコルでは、8K バイトの転送サイズ制限が解除されました。クライアントとサーバーは、version 2 の 8K バイトの制限を受けることなく、サポートされている転送サイズをネゴシエートできます。Solaris 2.5 では、デフォルトで、転送サイズが 32K バイトに設定されていることに注意してください。Solaris 10 以降のリリースでは、書き込み転送サイズの制限が緩和されました。使用するトランスポートプロトコルに基づいて転送サイズが決定されるようになりました。

NFS version 4 プロトコル

NFS version 4 は、以前のバージョンでは使用できない機能を備えています。

NFS version 4 プロトコルでは、ユーザー ID とグループ ID が文字列として表されます。`nfsmapid` は、次の目的でクライアントとサーバーが使用します。

- version 4 のこれらの ID 文字列をローカルの数値 ID に割り当てる
- ローカルの数値 ID を version 4 の ID 文字列に割り当てる

詳細は、148 ページの「`nfsmapid` デーモン」を参照してください。

NFS version4 では、ID マッパー nfsmapid を使用して、サーバー上の ACL エントリ内のユーザーまたはグループ ID を、クライアント上の ACL エントリ内のユーザーまたはグループ ID にマッピングします。逆も同じです。詳細は、192 ページの「NFS version 4 での ACL と nfsmapid」を参照してください。

NFS version 4 では、ファイルシステムの共有を解除するとき、そのファイルシステムにあるオープンファイルまたはファイルロックの状態がすべて削除されます。NFS version 3 では、ファイルシステムが共有解除される前に、サーバーはクライアントが取得したロックを保持しました。詳細は、184 ページの「NFS version 4 におけるファイルシステムの共有解除と再共有」を参照してください。

NFS version 4 のサーバーは擬似ファイルシステムを使用して、クライアントがサーバーにエクスポートされたオブジェクトにアクセスできるようにします。NFS version 4 以前のバージョンには、擬似ファイルシステムがありません。詳細は、185 ページの「NFS version 4 におけるファイルシステムの名前空間」を参照してください。

NFS version 2 と version 3 では、サーバーは持続的ファイルハンドルを返しました。NFS version 4 は、揮発性ファイルハンドルをサポートします。詳細は、186 ページの「NFS version 4 における揮発性ファイルハンドル」を参照してください。

委託とは、サーバーがファイルの管理をクライアントに委託するテクニックです。委託は、クライアントとサーバーの両方でサポートされます。たとえば、サーバーは、読み取り委託または書き込み委託のいずれかをクライアントに付与できます。詳細は、190 ページの「NFS version 4 における委託」を参照してください。

Solaris 10 以降のリリースでは、NFS version 4 は LIPKEY/SPKM セキュリティー方式をサポートしません。

また、NFS version 4 は次のデーモンを使用しません。

- mountd
- nfslogd
- statd

NFS version 4 での機能の一覧は、184 ページの「NFS version 4 における機能」を参照してください。

NFS version 4 の使用に関する手順については、94 ページの「NFS サービスの設定」を参照してください。

NFS バージョンの制御

/etc/default/nfs ファイルには、クライアントとサーバーで使用される NFS プロトコルを制御するためのキーワードがあります。たとえば、キーワードを使用し

て、バージョンネゴシエーションを管理します。詳細は、142 ページの「/etc/default/nfs ファイルのキーワード」、または nfs(4) のマニュアルページを参照してください。

NFS ACL サポート

Solaris 2.5 で、アクセス制御リスト (ACL) サポートが追加されました。 ACL では、ファイルアクセス権を通常の UNIX よりも厳密に設定します。この追加機能では効率は改善されませんが、ファイルへのアクセスがより厳密に制限されるので、セキュリティーが向上します。

NFS version 2 と version 3 プロトコルは、旧 POSIX ドラフトスタイルの ACL をサポートします。 POSIX ドラフト ACL は、UFS によりネイティブでサポートされます。UFS ACL の詳細は、『Solaris のシステム管理 (セキュリティサービス)』の「アクセス制御リストによる UFS ファイルの保護」を参照してください。

NFS version 4 プロトコルは、新しい NFSv4スタイルの ACL をサポートします。 NFSv4 ACL は、ZFS によりネイティブでサポートされます。NFSv4 ACL の全機能を利用するには、NFSv4 サーバーの基盤となるファイルシステムとして ZFS を使用する必要があります。NFSv4 ACL は、豊富な継承プロパティーセット、および標準の読み取り、書き込み、実行を超えたアクセス権ビットセットを備えています。新しい ACL の概要については、『Oracle Solaris ZFS 管理ガイド』の第 8 章「ACL による Oracle Solaris ZFS ファイルの保護」を参照してください。NFS version 4 での ACL のサポートの詳細は、192 ページの「NFS version 4 での ACL と nfsmapid」を参照してください。

TCP 経由の NFS

NFS プロトコルのデフォルトのトランスポートプロトコルは、Solaris 2.5 で TCP (Transport Control Protocol) に変更されました。TCP は、低速ネットワークとワイドエリアネットワークのパフォーマンスの向上に役立ちます。TCP には、トラフィック抑制機能とエラー回復機能もあります。TCP を利用した NFS は、version 2、version 3、および version 4 で動作します。Solaris 2.5 より前のリリースでは、NFS のデフォルトプロトコルはユーザーデータグラムプロトコル （UDP) でした。

UDP 経由の NFS

Solaris 10 以降のリリースでは、NFS クライアントで余分な UDP ポートが使用されなくなりました。これまで、UDP 経由の NFS 転送では、未処理の要求ごとに別々の UDP ポートが使用されていました。これからはデフォルトで、予約済みの UDP ポートが 1 つだけ使用されるようになりました。ただし、このサポートは設定可能です。複数のポートを同時に使用したほうがスケーラビリティーが高まり、結果的にシステムのパフォーマンスが向上するような場合には、複数のポートを使用する

ようにシステムを設定できます。なお、この機能は、TCP 経由の NFS に最初から備わっていた同種の設定可能なサポートを UDP に移植したものです。詳細は、『Oracle Solaris カーネルのチューンアップ・リファレンスマニュアル』を参照してください。

注 – NFS version 4 は、UDP を使用しません。`proto=udp` オプションを使用してファイルシステムをマウントする場合は、NFS version 3 が version 4 の代わりに使用されます。

RDMA 経由の NFS の概要

Solaris 10 リリースでは、RDMA (Remote Direct Memory Access) プロトコルが導入されています。RDMAは、高速ネットワーク上でデータのメモリー間転送を行うテクノロジです。特に、RDMA により、CPU の介入なしでメモリーに遠隔データ転送を直接行えます。この機能を提供するために、RDMA は、SPARC プラットフォーム上の InfiniBand のインターコネクト I/O テクノロジと Solaris オペレーティングシステムを組み合わせます。詳細は、181 ページの「RDMA 経由の NFS」を参照してください。

ネットワークロックマネージャーと NFS

Solaris 2.5 からネットワークロックマネージャーの改良版も含まれています。このため NFS ファイルに対して UNIX のレコードロックと PC のファイル共有を使用できます。NFS ファイルのロックメカニズムの信頼性の向上により、ロックを使用するコマンドのハングアップが起こりにくくなりました。

注 – ネットワークロックマネージャーは、NFS version 2 と version 3 のマウントでのみ使用されます。ファイルロックは、NFS version 4 プロトコルに組み込まれています。

NFS 大規模ファイルのサポート

Solaris 2.6 の NFS version 3 プロトコルから、2G バイトを超えるサイズのファイル (大規模ファイル) も正しく処理できるようになりました。NFS version 2 プロトコル、および Solaris 2.5 に実装されている version 3 プロトコルでは 2G バイトを超えるサイズのファイルは処理できませんでした。

NFS クライアントのフェイルオーバー機能

Solaris 2.6 では、読み取り専用ファイルシステムの動的フェイルオーバー機能が追加されました。フェイルオーバーによって、マニュアルページ、その他のドキュメント、共有バイナリなどのあらかじめ複製されている読み取り専用リソースを高度に

利用できます。フェイルオーバー機能は、ファイルシステムがマウントされた後ならばいつでも実行可能です。手動マウントでは、今までのリリースのオートマウンタのように複数の複製を一覧表示できるようになりました。オートマウンタは、フェイルオーバーの際にファイルシステムが再マウントされるまで待つ必要がなくなったこと以外は変更されていません。詳細は、92 ページの「クライアント側フェイルオーバーを使用する方法」 と 197 ページの「クライアント側フェイルオーバー機能」を参照してください。

NFS サービスのための Kerberos のサポート

Solaris 2.0 では、Kerberos V4 クライアントがサポートされていました。Solaris 2.6 では、`mount` と `share` コマンドが Kerberos V5 認証を使用する NFS version 3 のマウントをサポートするように変更されました。`share` コマンドもクライアントごとに異なる複数の認証機能を使用できるように変更されました。セキュリティー方式に関連する変更の詳細は、80 ページの「RPCSEC_GSS セキュリティー方式」を参照してください。Kerberos V5 認証の詳細は、『Solaris のシステム管理 (セキュリティサービス)』の「Kerberos NFS サーバーの構成」を参照してください。

WebNFS のサポート

Solaris 2.6 には、インターネット上のファイルシステムにファイアウォール経由でアクセスできるようにする機能も追加されました。この機能は、NFS プロトコルの拡張機能によって実現しました。インターネットアクセスに WebNFS プロトコルを使用する利点の 1 つは、信頼性が高いことです。このサービスは、NFS version 3 と version 2 プロトコルの拡張として構築されています。さらに、WebNFS ではそうしたファイルを共有しても匿名 `ftp` サイトを管理するオーバーヘッドが生じません。WebNFS サービスに関連する変更の詳細は、81 ページの「WebNFS サービスのセキュリティーネゴシエーション」を参照してください。作業の詳細は、102 ページの「WebNFS の管理作業」を参照してください。

注 – NFS version 4 プロトコルは、WebNFS サービスに優先します。NFS version 4 は、MOUNT プロトコルと WebNFS サービスに追加されたすべてのセキュリティーネゴシエーションを完全に統合します。

RPCSEC_GSS セキュリティー方式

Solaris 7 から、RPCSEC_GSS と呼ばれるセキュリティー方式がサポートされています。この方式では、標準的な GSS-API インタフェースを使用して、認証、一貫性、機密性を実現し、複数のセキュリティーメカニズムをサポートしています。Kerberos V5 認証のサポートについての詳細は、80 ページの「NFS サービスのための Kerberos のサポート」を参照してください。GSS-API についての詳細は、『Oracle Solaris セキュリティーサービス開発ガイド』を参照してください。

Solaris 7 の NFS に対する拡張機能

Solaris 7 で、`mount` コマンドと `automountd` コマンドが拡張され、マウント要求で MOUNT プロトコルの代わりに公開ファイルハンドルも使用できるようになりました。MOUNT プロトコルは、WebNFS サービスが使用するアクセス方法と同じです。公開ファイルハンドルを使用すると、ファイアウォールを越えたマウントが可能です。さらに、サーバーとクライアント間のトランザクションが少なくて済むため、マウントにかかる時間が短縮されます。

この拡張機能で、標準のパス名の代わりに NFS URL を使用することもできるようになりました。また、`mount` コマンドとオートマウンタのマップに `public` オプションを指定すると、必ず公開ファイルハンドルを使用するようになります。WebNFS サービスの変更の詳細は、80 ページの「WebNFS のサポート」を参照してください。

WebNFS サービスのセキュリティーネゴシエーション

Solaris 8 で、WebNFS クライアントが NFS サーバーとセキュリティーメカニズムをネゴシエートするための新しいプロトコルが追加されました。このプロトコルの追加により、WebNFS サービスの使用時に、セキュリティー保護されたトランザクションを使用できます。詳細は、202 ページの「WebNFS セキュリティーネゴシエーション機能のしくみ」を参照してください。

NFS サーバーロギング

Solaris 8 で、NFS サーバーはロギングによって、ファイルシステムで実行されたファイル操作の記録を提供できるようになりました。この記録には、どのファイルが、いつ、だれによってアクセスされたかという情報が含まれています。一連の構成オプションを使用して、これらの情報を含むログの場所を指定することができます。また、これらのオプションを使用して、ログに記録する処理を選択することもできます。この機能は、NFS クライアントや WebNFS クライアントで匿名 ftp を利用するサイトで特に便利です。詳細は、87 ページの「NFS サーバーログを有効にする方法」を参照してください。

注 – NFS version 4 は、サーバーロギングをサポートしません。

---

autofs の機能

autofs は、ローカルの名前空間に指定されているファイルシステムで動作します。この情報は、NIS、NIS+、およびローカルファイルに保持されます。

Solaris 2.6 から、完全にマルチスレッド化された automountd が組み込まれています。この拡張によって autofs はさらに信頼性が高まりました。また、複数のマウントを並行してサービスできるようになったため、あるサーバーが使用できないときにサービスが停止することも避けられます。

この新しい automountd には、改善されたオンデマンドマウント機能も備わっています。Solaris 2.6 より前のリリースでは、階層に含まれるすべてのファイルシステムがマウントされていました。現在は、いちばん上のファイルシステムしかマウントされません。そのマウントポイントに関係する他のファイルシステムは、必要に応じてマウントされます。

autofs サービスで、間接マップを表示できるようになりました。これによりユーザーは、どのディレクトリがマウントできるかを確認するために各ファイルシステムを実際にマウントする必要がなくなります。autofs マップに -nobrowse オプションが追加されたので、/net や /home などの大きなファイルが自動的に表示されることはありません。また、-automount で n オプションを使用することによって、autofs のブラウズ機能をクライアントごとにオフにすることもできます。詳細は、120 ページの「autofs のブラウズ機能を無効にする」を参照してください。

◆ ◆ ◆ 第 5 章

ネットワークファイルシステムの管理 (手順)

この章では、NFS サービスの設定、共有する新規ファイルシステムの追加、ファイルシステムのマウントなど、NFS の管理作業の実行方法について説明します。また、Secure NFS システムおよび WebNFS の機能の使用方法についても説明します。章の最後ではトラブルシューティングの手順を説明し、NFS のいくつかのエラーメッセージとその意味を示します。

NFS 管理者の責任は、サイトの要求やネットワーク上に存在するコンピュータの役割によって変わります。管理者がローカルネットワークのコンピュータすべてに責任を持つこともありえます。そのような場合は、次の設定事項について判断する必要があります。

- サーバー専用にするコンピュータの決定
- サーバーとクライアントの両方として動作するコンピュータの決定
- クライアントとしてのみ動作するコンピュータの決定

設定が完了したサーバーの保守には、次の作業が必要です。

- ファイルシステムの共有開始と共有解除
- 管理ファイルを修正し、コンピュータが共有したり、自動的にマウントしたファイルシステムのリストを更新したりすること
- ネットワークの状態のチェック
- NFS に関連した問題の診断と解決

- autofs のマップの設定

コンピュータは、サーバーとクライアントのどちらにもなれることに注意してください。つまり、ローカルファイルシステムをリモートコンピュータと共有したり、リモートファイルシステムをマウントしたりできます。

注 – システムでゾーンが有効なときに非大域ゾーンでこの機能を使用するには、『Oracle Solaris のシステム管理 (Oracle Solaris コンテナ : 資源管理と Oracle Solaris ゾーン)』を参照してください。

ファイルシステムの自動共有

NFS 環境でファイルシステムを共有することにより、サーバーのファイルシステムにアクセスできるようになります。共有するファイルシステムは、`share` コマンドまたは `/etc/dfs/dfstab` ファイルで指定します。

`/etc/dfs/dfstab` ファイル中のエントリは、NFS サーバーオペレーションを起動したときに自動的に共有されます。同じファイルシステムを定期的に共有する必要がある場合は、自動共有を設定するようにしてください。たとえばサーバーがホームディレクトリをサポートしている場合、ホームディレクトリを常に使用できるようにしておく必要があります。ファイルシステムの共有はほとんどが自動的に行われます。共有を手動で実行するのは、テストまたはトラブルシューティングの場合だけです。

`dfstab` ファイルには、サーバーがクライアントと共有しているすべてのファイルシステムが一覧表示されています。このファイルを使用して、ファイルシステムをマウントできるクライアントを制御します。`dfstab` ファイルを変更して、ファイルシステムを追加または削除したり、共有方法を変更したりできます。その場合は、`vi` などのサポートされているテキストエディタを使って dfstab ファイルを編集します。コンピュータが次に実行レベル 3 に入ったときに、更新された `dfstab` ファイルが読み込まれ、共有するファイルシステムが自動的に判断されます。

`dfstab` ファイルの各行は、`share` コマンドで構成されています。このコマンドは、コマンド行プロンプトに入力してファイルシステムを共有するのと同じコマンドです。`share` コマンドは、`/usr/sbin` に保存されています。

表 5–1 ファイルシステムの共有 (作業マップ)

作業	説明	参照先
ファイルシステムの自動共有を確立します	サーバーのリブート時、ファイルシステムが自動的に共有されるようにサーバーを設定する手順	85 ページの「ファイルシステム自動共有を設定する方法」

表 5–1 ファイルシステムの共有 (作業マップ)　(続き)

作業	説明	参照先
WebNFS を有効にします	ユーザーが WebNFS でファイルにアクセスできるようにサーバーを設定する手順	86 ページの「WebNFS アクセスを有効にする方法」
NFS サーバーログを有効にします	NFS ログが選択したファイルシステム上で動作するようにサーバーを設定する手順	87 ページの「NFS サーバーログを有効にする方法」

▼ ファイルシステム自動共有を設定する方法

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 共有する対象の各ファイルシステムに関してエントリを追加します。

`/etc/dfs/dfstab` を編集します。自動的に共有する各ファイルシステムのファイルにエントリを 1 つ追加します。各エントリは、ファイル中に 1 行で記述する必要があり、次のような構文を使用します。

```
share [-F nfs] [-o specific-options] [-d description] pathname
```

`/etc/dfs/dfstab` については `dfstab(4)` のマニュアルページを、オプションの完全な一覧については `share_nfs(1M)` のマニュアルページを、それぞれ参照してください。

3 ファイルシステムを共有します。

エントリを `/etc/dfs/dfstab` に追加したあと、システムをリブートするか、`shareall` コマンドを使用して、ファイルシステムを共有可能にします。

```
# shareall
```

4 情報が正しいことを確認します。

`share` コマンドを実行し、適切なオプションが表示されていることを確認します。

```
# share
-       /export/share/man   ro   ""
-       /usr/src     rw=eng   ""
-       /export/ftp    ro,public  ""
```

参照　次の手順では、サーバー上で共有したファイルシステムにクライアントがアクセスできるように autofs マップを設定します。105 ページの「autofs 管理作業の概要」を参照してください。

▼ WebNFS アクセスを有効にする方法

Solaris 2.6 リリース以降ではデフォルトで、NFS マウントに利用可能なすべてのファイルシステムが、WebNFS アクセス用として自動的に利用可能となります。この手順を使用する必要があるのは、次のいずれかの場合だけです。

- NFS マウントがその時点で利用可能になっていないサーバーで NFS マウントができるようにする場合
- `public` オプションを使用することで、公開ファイルハンドルをリセットして NFS URL を短くする場合
- `index` オプションを使用することで、特定の HTML ファイルが強制的に読み込まれるようにする場合

WebNFS サービスを起動する際の注意事項については、103 ページの「WebNFS アクセスの計画」を参照してください。

1. スーパーユーザーになるか、同等の役割を引き受けます。

   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2. **WebNFS サービスを使用して、共有する各ファイルシステムのエントリを追加します。**

   `/etc/dfs/dfstab` を編集します。各ファイルシステムごとにエントリを 1 つ追加します。次の例の `public` タグおよび `index` タグは省略できます。

   ```
   share -F nfs -o ro,public,index=index.html /export/ftp
   ```

   `/etc/dfs/dfstab` については `dfstab(4)` のマニュアルページを、オプションの完全な一覧については `share_nfs(1M)` のマニュアルページを、それぞれ参照してください。

3. ファイルシステムを共有します。

   エントリを `/etc/dfs/dfstab` に追加したあと、システムをリブートするか、`shareall` コマンドを使用して、ファイルシステムを共有可能にします。

   ```
   # shareall
   ```

4. 情報が正しいことを確認します。

   `share` コマンドを実行し、適切なオプションが表示されていることを確認します。

   ```
   # share
   -        /export/share/man   ro   ""
   -        /usr/src     rw=eng   ""
   -        /export/ftp    ro,public,index=index.html  ""
   ```

▼ NFS サーバーログを有効にする方法

1. スーパーユーザーになるか、同等の役割を引き受けます。

   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2. (省略可能) ファイルシステム構成の設定を変更します。

   `/etc/nfs/nfslog.conf` で設定を変更する方法は 2 つあります。すべてのファイルシステムについてデフォルトの設定を編集するには、`global` タグに関連するデータを変更します。または、このファイルシステムについて新しいタグを追加します。これらの変更が必要でない場合には、このファイルを変更する必要はありません。`/etc/nfs/nfslog.conf` の書式については、`nfslog.conf(4)` のマニュアルページを参照してください。

3. NFS サーバーログを使用して、共有する各ファイルシステムについてエントリを追加します。

   `/etc/dfs/dfstab` を編集します。NFS サーバーログを有効にするファイルシステムについてエントリを 1 つ追加します。`log=`*tag* オプションとともに使用するタグは、`/etc/nfs/nfslog.conf` にも記述する必要があります。次の例では、`global` タグ内のデフォルト設定を使用しています。

   ```
   share -F nfs -o ro,log=global /export/ftp
   ```

   `/etc/dfs/dfstab` については `dfstab(4)` のマニュアルページを、オプションの完全な一覧については `share_nfs(1M)` のマニュアルページを、それぞれ参照してください。

4. ファイルシステムを共有します。

   エントリを `/etc/dfs/dfstab` に追加したあと、システムをリブートするか、`shareall` コマンドを使用して、ファイルシステムを共有可能にします。

   ```
   # shareall
   ```

5. 情報が正しいことを確認します。

   `share` コマンドを実行し、適切なオプションが一覧表示されることを確認します。

   ```
   # share
   -        /export/share/man   ro   ""
   -        /usr/src     rw=eng   ""
   -        /export/ftp    ro,log=global  ""
   ```

6. NFS ログデーモン `nfslogd` が動作していることを確認します。

   ```
   # ps -ef | grep nfslogd
   ```

7 (省略可能) 動作していない場合は、nfslogd を起動します。

- (省略可能) /etc/nfs/nfslogtab が存在している場合は、次のように入力して、**NFS** ログデーモンを起動します。

```
# svcadm restart network/nfs/server:default
```

- (省略可能) /etc/nfs/nfslogtab が存在していない場合は、ファイルを作成するために任意の share コマンドを実行してから、デーモンを起動します。

```
# shareall
# svcadm restart network/nfs/server:default
```

ファイルシステムのマウント

ファイルシステムをマウントするには、いくつかの方法があります。システムをブートするときに自動的にマウントされるようにするか、コマンド行から必要に応じてマウントするか、オートマウンタを使用します。オートマウンタには、ブート時のマウントやコマンド行からのマウントに比較していくつもの利点がありますが、状況によってこの 3 つの方法を組み合わせる必要があります。また、ファイルシステムのマウント時に使用するオプションに応じて、プロセスを有効または無効にする方法がいくつかあります。ファイルシステムのマウントに関するすべての作業のリストについては、次の表を参照してください。

表 5–2 ファイルシステムのマウントの作業マップ

作業	説明	参照先
ブート時にファイルシステムをマウントします	システムがリブートされるときに必ずファイルシステムがマウントされるようにする手順。	89 ページの「ブート時にファイルシステムにマウントする方法」
コマンドを使用してファイルシステムをマウントします	システムの動作時にファイルシステムをマウントする手順。この手順はテストに有効です。	90 ページの「コマンド行からファイルシステムをマウントする方法」
オートマウンタによりマウントします	コマンド行を使用せずに、要求に応じてファイルシステムにアクセスする手順。	90 ページの「オートマウンタによるマウント」
大規模ファイルを避けます	ファイルシステム上に大規模ファイルが作成されないようにする手順。	91 ページの「NFS サーバー上で大規模ファイルを無効にする方法」
クライアント側フェイルオーバーを開始します	サーバーの不良時、動作中のファイルシステムへの自動切り換えを有効にする手順。	92 ページの「クライアント側フェイルオーバーを使用する方法」
クライアントに対するマウントアクセスを無効にします	任意のクライアントがリモートシステムにアクセスする機能を無効にする手順。	92 ページの「1 つのクライアントに対するマウントアクセスを無効にする方法」

表 5–2 ファイルシステムのマウントの作業マップ (続き)

作業	説明	参照先
ファイアウォールを越えてファイルシステムにアクセスを提供します	WebNFS プロトコルでファイアウォールを越えてファイルシステムへのアクセスを許可する手順。	93 ページの「ファイアウォールを越えて NFS ファイルシステムをマウントする方法」
NFS URL を使ってファイルシステムをマウントします	NFS URL を使ってファイルシステムへのアクセスを許可する手順。このプロセスによって、MOUNT プロトコルを使用しないでファイルシステムへのアクセスが可能になります。	94 ページの「NFS URL を使用して NFS ファイルシステムをマウントする方法」

▼ ブート時にファイルシステムにマウントする方法

autofs マップを使用するのではなく、ブート時にファイルシステムをマウントするには、次の手順に従います。リモートファイルシステムにアクセスするクライアントごとに、この手順を行う必要があります。

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 ファイルシステムに関するエントリを /etc/vfstab に追加します。

/etc/vfstab ファイルのエントリ構文は、次のとおりです。

```
special  fsckdev  mountp  fstype  fsckpass  mount-at-boot  mntopts
```

詳細は、vfstab(4) のマニュアルページを参照してください。

注意 – NFS クライアントの vfstab エントリも持つ NFS サーバーでは、リブート時のハングアップを避けるために、常に bg オプションを指定する必要があります。詳細は、160 ページの「NFS ファイルシステム用の mount オプション」を参照してください。

例 5–1 クライアントの vfstab ファイル内のエントリ

wasp サーバーの /var/mail ディレクトリをクライアントマシンにマウントさせたいとします。それには、そのファイルシステムをクライアント上の /var/mail としてマウントし、読み取りと書き込みの両方ができるようにします。この場合は、次の項目をクライアントの vfstab ファイルに追加します。

```
wasp:/var/mail - /var/mail nfs - yes rw
```

▼ コマンド行からファイルシステムをマウントする方法

新規マウントポイントをテストするために、コマンド行からファイルシステムをマウントすることがあります。このようにしてマウントすると、オートマウンタでアクセスできないファイルシステムに、一時的にアクセスすることができます。

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 ファイルシステムをマウントします。

次のコマンドを入力します。

```
# mount -F nfs -o ro bee:/export/share/local /mnt
```

上の例では、bee サーバーの /export/share/local ファイルシステムが、ローカルシステムの /mnt に読み取り専用でマウントされます。コマンド行からこのようにマウントすることにより、ファイルシステムを一時的に表示することができます。umount を実行するかローカルホストをリブートすると、このマウントは解除されます。

注意 – Solaris 2.6 およびそれ以降に出たパッチに置き換えられた mount コマンドでは、無効なオプションを指定しても警告されません。解釈できないオプションがあると無視されるだけです。予想外の結果が生じるのを避けるために、使用するオプションはすべて確認してください。

オートマウンタによるマウント

105 ページの「autofs 管理作業の概要」では、オートマウンタによるマウントの確立とサポートについて詳細に説明します。通常のシステムに変更を加えることなく、リモートファイルシステムが /net マウントポイントでアクセスできるようになります。前述の例の /export/share/local ファイルシステムをマウントする場合は、次のように入力します。

```
% cd /net/bee/export/share/local
```

オートマウンタでは、すべてのユーザーがファイルシステムをマウントできるので、root としてアクセスする必要はありません。またファイルシステムのマウントを自動的に解除できるので、作業の終了後、ファイルシステムのマウントを解除する必要はありません。

▼ NFS サーバー上で大規模ファイルを無効にする方法

2G バイト超のファイルを処理できないクライアントをサポートするサーバーでは、大規模ファイル作成機能の無効化が必要になることがあります。

注 – Solaris 2.6 より前の動作環境では、大規模ファイルは使用できません。クライアントが大規模ファイルにアクセスする必要がある場合には、NFS サーバーのクライアントが Solaris 2.6 以降のリリースで動作していることを確認してください。

---

1. スーパーユーザーになるか、同等の役割を引き受けます。

   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2. ファイルシステム上に大規模ファイルが存在していないことを確認してください。

   次に例を示します。

   ```
   # cd /export/home1
   # find . -xdev -size +2000000 -exec ls -l {} \;
   ```

   システム上に大規模ファイルが存在する場合には、削除するか、他のファイルシステムに移動する必要があります。

3. ファイルシステムをマウント解除します。

   ```
   # umount /export/home1
   ```

4. `largefiles` を使用してファイルシステムがマウントされている場合は、ファイルシステムの状態をリセットします。

   `fsck` は、ファイルシステム上に大規模ファイルが存在しない場合に、ファイルシステムの状態をリセットします。

   ```
   # fsck /export/home1
   ```

5. `nolargefiles` を使用して、ファイルシステムをマウントします。

   ```
   # mount -F ufs -o nolargefiles /export/home1
   ```

   コマンド行からマウントすることができますが、オプションを常時使用するようにするには、`/etc/vfstab` に次のようなエントリを追加してください。

   ```
   /dev/dsk/c0t3d0s1 /dev/rdsk/c0t3d0s1 /export/home1  ufs  2  yes  nolargefiles
   ```

▼ クライアント側フェイルオーバーを使用する方法

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 **NFS** クライアント上で、`ro` オプションを使用してファイルシステムをマウントします。

コマンド行からも、オートマウンタを使用しても、また`/etc/vfstab` ファイルに次のようなエントリを追加することによってもマウントできます。

```
bee,wasp:/export/share/local  -  /usr/local  nfs  -  no  ro
```

この構文はオートマウンタでも指定できました。しかし、フェイルオーバー機能が使用できるのは単一のサーバーが選択されているときだけで、ファイルシステムがマウントされている間は使用できませんでした。

注 – 異なるバージョンの NFS プロトコルを実行しているサーバーを、コマンド行や `vfstab` のエントリに混在させないでください。NFS version 2、version 3、または version 4 のプロトコルをサポートしているサーバーを混在して使用できるのは、autofs を使用する場合だけです。autofs では、version 2 、version 3、または version 4 のサーバーの最適なサブセットが使用されます。

▼ 1 つのクライアントに対するマウントアクセスを無効にする方法

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 `/etc/dfs/dfstab` にエントリを追加します。

最初の例では、`rose` という名前のホストを除き、`eng` ネットグループ内のすべてのクライアントへのマウントアクセスを許可しています。2 つめの例では、`rose` を除き、`eng.sun.com` DNS ドメイン内にあるすべてのクライアントへのマウントアクセスを許可しています。

```
share -F nfs -o ro=-rose:eng /export/share/man
share -F nfs -o ro=-rose:.eng.example.com /export/share/man
```

アクセスリストに関する補足情報については、171 ページの「share コマンドを使ってアクセスリストを設定する」を参照してください。/etc/dfs/dfstab については、dfstab(4) のマニュアルページを参照してください。

3 ファイルシステムを共有します。

/etc/dfs/dfstab への変更は、このファイルシステムがもう一度共有されるかサーバーがリブートされるまでは NFS サーバーに反映されません。

```
# shareall
```

▼ ファイアウォールを越えて NFS ファイルシステムをマウントする方法

ファイアウォールを越えてファイルシステムにアクセスするには、次の手順を実行します。

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 次のコマンドを使用して、ファイルシステムを手動でマウントします。

```
# mount -F nfs bee:/export/share/local /mnt
```

この例では、/export/share/local というファイルシステムは、公開ファイルハンドルを使ってローカルクライアントにマウントしています。標準のパス名の代わりに、NFS URL を使用することができます。ただし bee サーバーで公開ファイルハンドルがサポートされていないと、マウント操作は失敗します。

注 – この手順では、NFS サーバーのファイルシステムを public オプションで共有する必要があります。また、クライアントとサーバー間のファイアウォールでは、ポート 2049 で TCP 接続できるようにする必要があります。Solaris 2.6 以降のリリースでは、共有しているすべてのファイルシステムに、公開ファイルハンドルでアクセスできます。そのため、デフォルトでは、public オプションが適用されています。

▼ NFS URL を使用して NFS ファイルシステムをマウントする方法

1. スーパーユーザーになるか、同等の役割を引き受けます。
   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2. (省略可能) NFS version 2 または version 3 を使用している場合、次のコマンドを使用して、ファイルシステムを手動でマウントします。

   ```
   # mount -F nfs nfs://bee:3000/export/share/local /mnt
   ```

   この例では、サーバー bee の /export/share/local というファイルシステムが、NFS ポート番号 3000 を使ってマウントされます。ポート番号を指定する必要はありません。その場合、デフォルトの NFS ポート番号である 2049 が使用されます。NFS URL に、public オプションを含めるかどうかを選択できます。public オプションを指定しない場合、サーバーが公開ファイルハンドルをサポートしていなければ、MOUNT プロトコルが使用されます。public オプションを指定すると、必ず公開ファイルハンドルを使用するように指定され、公開ファイルハンドルがサポートされていないとマウントは失敗します。

3. (省略可能) NFS version 4 を使用している場合、次のコマンドを使用して、ファイルシステムを手動でマウントします。

   ```
   # mount -F nfs -o vers=4 nfs://bee:3000/export/share/local /mnt
   ```

NFS サービスの設定

この節では、次のことを行うために必要な作業を説明します。

- NFS サーバーを起動および停止する
- オートマウンタを起動および停止する
- 異なるバージョンの NFS を選択する

注 – Solaris 10 以降のリリースでは、NFS のデフォルトは version 4 です。

表 5–3 NFS サービスの作業マップ

作業	説明	参照先
NFS サーバーを起動します	NFS サービスが自動的に起動されていない場合に、NFS サービスを起動する手順。	95 ページの「NFS サービスを起動する方法」
NFS サーバーを停止します	NFS サービスを停止する手順。通常は、サービスを停止する必要はありません。	96 ページの「NFS サービスを停止する方法」

表 5–3 NFS サービスの作業マップ (続き)

作業	説明	参照先
オートマウンタを起動します	オートマウンタを起動する手順。オートマウンタマップが変更された場合、この手順が必要です。	96 ページの「オートマウンタを起動する方法」
オートマウンタを停止します	オートマウンタを停止する手順。オートマウンタマップが変更された場合、この手順が必要です。	96 ページの「オートマウンタを停止する方法」
サーバー上で異なるバージョンの NFS を選択します	サーバー上で異なるバージョンの NFS を選択する手順。NFS version 4 を使用しない場合、この手順を使用します。	97 ページの「サーバー上で異なるバージョンの NFS を選択する方法」
クライアント上で異なるバージョンの NFS を選択します	`/etc/default/nfs` ファイルを変更して、クライアント上で異なるバージョンの NFS を選択する手順。NFS version 4 を使用しない場合、この手順を使用します。	98 ページの「`/etc/default/nfs` ファイルを変更することで、クライアント上で異なるバージョンの NFS を選択する方法」
	コマンド行を使用して、クライアント上で異なるバージョンの NFS を選択する代替手順。NFS version 4 を使用しない場合、この代替手順を使用します。	99 ページの「コマンド行を使用して、クライアント上で異なるバージョンの NFS を選択する方法」

▼ NFS サービスを起動する方法

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 サーバー上で **NFS** サービスを有効にします。

次のコマンドを入力します。

```
# svcadm enable network/nfs/server
```

このコマンドを実行すると、NFS サービスが有効になります。

注 – Solaris 9 を起動すると、システムのブート時に NFS サーバーは自動的に起動します。さらに、システムのブート以降は、NFS ファイルシステムを共有すると NFS サービスデーモンが自動的に有効になります。85 ページの「ファイルシステム自動共有を設定する方法」を参照してください。

▼ NFS サービスを停止する方法

1. スーパーユーザーになるか、同等の役割を引き受けます。
   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。
2. サーバー上で **NFS** サービスを無効にします。
   次のコマンドを入力します。

```
# svcadm disable network/nfs/server
```

▼ オートマウンタを起動する方法

1. スーパーユーザーになるか、同等の役割を引き受けます。
   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。
2. **autofs** デーモンを有効にします。
   次のコマンドを入力します。

```
# svcadm enable system/filesystem/autofs
```

▼ オートマウンタを停止する方法

1. スーパーユーザーになるか、同等の役割を引き受けます。
   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。
2. **autofs** デーモンを無効にします。
   次のコマンドを入力します。

```
# svcadm disable system/filesystem/autofs
```

▼ サーバー上で異なるバージョンの NFS を選択する方法

NFS version 4 を使用しない場合、この手順を使用します。

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 `/etc/default/nfs` ファイルを編集します。

たとえば、サーバーが version 3 だけを使用するようにするには、NFS_SERVER_VERSMAX と NFS_SERVER_VERSMIN の値を 3 に設定します。キーワードとその値の一覧については、142 ページの「`/etc/default/nfs` ファイルのキーワード」を参照してください。

```
NFS_SERVER_VERSMAX=value
NFS_SERVER_VERSMIN=value
```

value　バージョン番号を指定します。

注 – これらの行は、デフォルトでコメントになっています。ポンド記号 (#) を削除することも忘れないでください。

---

3 **SMF パラメータを変更して、NFS のバージョン番号を設定します。**

たとえば、サーバーが version 3 だけを使用するようにするには、次のようにして `server_vermax` と `server_versmin` の両方を 3 に設定します。

```
# sharectl set -p server_vermax=3 nfs
# sharectl set -p server_vermin=3 nfs
```

4 (省略可能) サーバー委託を無効にする場合、`/etc/default/nfs` ファイルに次の行を追加します。

```
NFS_SERVER_DELEGATION=off
```

注 – NFS version 4 では、サーバー委託は、デフォルトで有効になっています。詳細は、190 ページの「NFS version 4 における委託」を参照してください。

---

5 (省略可能) クライアントとサーバーの共通ドメインを設定する場合は、`/etc/default/nfs` ファイルに次の行を追加します。

```
NFSMAPID_DOMAIN=my.comany.com
```

my.comany.com　共通ドメインを指定します

詳細は、148 ページの「`nfsmapid` デーモン」を参照してください。

6 **NFS サービスがサーバー上で動作していることを確認します。**

次のコマンドを入力します。

```
# svcs network/nfs/server
```

このコマンドは、NFS サーバーサービスがオンラインか、または無効かをレポートします。

7 **(省略可能) 必要に応じて、NFS サービスを無効にします。**

NFS サービスがオンラインであることを前の手順で検出した場合、次のコマンドを入力して、サービスを無効にします。

```
# svcadm disable network/nfs/server
```

注 – NFS サービスを構成する必要がある場合は、85 ページの「ファイルシステム自動共有を設定する方法」を参照してください。

---

8 **NFS サービスを有効にします。**

次のコマンドを入力して、サービスを有効にします。

```
# svcadm enable network/nfs/server
```

参照 183 ページの「NFS におけるバージョンのネゴシエーション」

▼ /etc/default/nfs ファイルを変更することで、クライアント上で異なるバージョンの NFS を選択する方法

次の手順は、`/etc/default/nfs` ファイルを変更して、クライアント上で使用される NFS のバージョンを制御する方法を示しています。コマンド行を使用する場合は、99 ページの「コマンド行を使用して、クライアント上で異なるバージョンの NFS を選択する方法」を参照してください。

1 **スーパーユーザーになるか、同等の役割を引き受けます。**

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 /etc/default/nfs ファイルを編集します。

たとえば、クライアント上で version 3 だけを使用するようにするには、NFS_CLIENT_VERSMAX と NFS_CLIENT_VERSMIN の値を 3 に設定します。キーワードとその値の一覧については、142 ページの「/etc/default/nfs ファイルのキーワード」を参照してください。

```
NFS_CLIENT_VERSMAX=value
NFS_CLIENT_VERSMIN=value
```

value バージョン番号を指定します。

注 – これらの行は、デフォルトでコメントになっています。ポンド記号 (#) を削除することも忘れないでください。

---

3 クライアント上で **NFS** をマウントします。

次のコマンドを入力します。

```
# mount server-name:/share-point /local-dir
```

server-name サーバーの名前を指定します。

/share-point 共有するリモートディレクトリのパスを指定します。

/local-dir ローカルマウントポイントのパスを指定します。

参照 183 ページの「NFS におけるバージョンのネゴシエーション」

▼ コマンド行を使用して、クライアント上で異なるバージョンの NFS を選択する方法

次の手順は、コマンド行を使用して、クライアントで特定のマウントに使用される NFS のバージョンを制御する方法を示しています。/etc/default/nfs ファイルを変更する場合は、98 ページの「/etc/default/nfs ファイルを変更することで、クライアント上で異なるバージョンの NFS を選択する方法」を参照してください。

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 クライアント上で、目的のバージョンの **NFS** をマウントします。

次のコマンドを入力します。

```
# mount -o vers=value server-name:/share-point /local-dir
```

value バージョン番号を指定します。

server-name サーバーの名前を指定します。

/share-point 共有するリモートディレクトリのパスを指定します。

/local-dir ローカルマウントポイントのパスを指定します。

注 – このコマンドは、NFS プロトコルを使用して、リモートディレクトリをマウントし、`/etc/default/nfs` ファイルのクライアント設定を上書きします。

参照 183 ページの「NFS におけるバージョンのネゴシエーション」

Secure NFS システムの管理

Secure NFS システムを使用するには、関与するすべてのコンピュータにドメイン名が必要です。通常、ドメインとは、複数のコンピュータから構成される管理上のエンティティーのことであり、大規模なネットワークの一部です。ネームサービスを実行している場合、そのドメインに対してネームサービスを設定するようにしてください。『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』を参照してください。

NFS サービスでは、Kerberos version 5 認証もサポートされています。Kerberos サービスについては、『Solaris のシステム管理 (セキュリティサービス)』の第 21 章「Kerberos サービスについて」を参照してください。

Secure NFS 環境は、Diffie-Hellman 認証を使用するようにも設定できます。この認証サービスについては、『Solaris のシステム管理 (セキュリティサービス)』の第 16 章「認証サービスの使用 (手順)」を参照してください。

▼ DH 認証を使用して Secure NFS 環境を設定する方法

1 ドメインにドメイン名を割り当て、そのドメイン名をドメイン内の各コンピュータに知らせます。

NIS+ をネームサービスとして使用している場合は、『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』を参照してください。

2 newkey コマンドまたは nisaddcred コマンドを使用して、クライアントのユーザーの公開鍵と秘密鍵を設定します。chkey コマンドを使用して、各ユーザーに独自の **Secure RPC** パスワードを設定してもらいます。

注 – これらのコマンドについての詳細は、newkey(1M)、nisaddcred(1M)、および chkey(1) のマニュアルページを参照してください。

公開鍵と秘密鍵が生成されると、公開鍵と暗号化された秘密鍵が publickey データベースに格納されます。

3 ネームサービスが応答していることを確認します。

NIS+ を実行している場合は、次のように入力してください。

```
# nisping -u
Last updates for directory eng.acme.com. :
Master server is eng-master.acme.com.
        Last update occurred at Mon Jun  5 11:16:10 1995

Replica server is eng1-replica-replica-58.acme.com.
        Last Update seen was Mon Jun  5 11:16:10 1995
```

NIS を実行している場合は、ypbind デーモンが動作していることを確認してください。

4 キーサーバーの keyserv デーモンが動作していることを確認します。

次のコマンドを入力します。

```
# ps -ef | grep keyserv
root    100      1  16    Apr 11 ?        0:00 /usr/sbin/keyserv
root   2215   2211   5  09:57:28 pts/0    0:00 grep keyserv
```

デーモンが動作していない場合は、次のように入力してキーサーバーを起動します。

```
# /usr/sbin/keyserv
```

5 秘密鍵の復号化と保存を実行します。

通常、ログインパスワードはネットワークパスワードと同じです。この場合、keylogin は不要です。ログインパスワードとネットワークパスワードが異なる場合、ユーザーはログインしてから keylogin を実行しなければなりません。また、keylogin -r コマンドを root として実行し、復号化した秘密鍵を /etc/.rootkey に保存する必要があります。

注 – keylogin -r は、root の秘密鍵が変更されたか、/etc/.rootkey が損失した場合に、実行する必要があります。

---

6 ファイルシステムに対するマウントオプションを更新します。

Diffie-Hellman 認証を使用するには、/etc/dfs/dfstab ファイルを編集し、該当するエントリに sec=dh オプションを追加します。

```
share -F nfs -o sec=dh /export/home
```

/etc/dfs/dfstab については、dfstab(4) のマニュアルページを参照してください。

7 ファイルシステムに対するオートマウンタマップを更新します。

auto_master データを編集し、Diffie-Hellman 認証の適切なエントリ内にマウントオプションとして sec=dh を含めます。

```
/home     auto_home     -nosuid,sec=dh
```

注 – Solaris 2.5 以前のリリースでは、その機能が制限されています。クライアントが、セキュリティー保護されている共有ファイルシステムにセキュリティーモードでマウントしない場合、ユーザーは、そのユーザー自身ではなく、nobody ユーザーとしてアクセスすることになります。Solaris 2.5 よりあとの NFS version 2 では、セキュリティーモードが一致しないと、share コマンド行に -sec=none が指定されていないかぎり、NFS サーバーによってアクセスが拒否されます。NFS の version 3 では、セキュリティー保護されていることを示すモードが NFS サーバーから引き継がれるので、クライアントが sec=dh を指定する必要はありません。ユーザーは、そのユーザー自身としてファイルにアクセスできます。

コンピュータを設置し直したり、移設したり、アップグレードしたりするときに、新しい鍵を設定せず、root 用の鍵も変更しない場合は、必ず /etc/.rootkey を保存してください。/etc/.rootkey を削除するには、通常、次のコマンドを入力します。

```
# keylogin -r
```

WebNFS の管理作業

この節では、WebNFS システムを管理する方法について説明します。次の表に、関連する作業を示します。

表 5–4 WebNFS 管理の作業マップ

作業	説明	参照先
WebNFS に関する計画を作成します	WebNFS サービスを有効にする前に考慮する項目。	103 ページの「WebNFS アクセスの計画」
WebNFS を有効にします	WebNFS プロトコルを使用して NFS ファイルシステムのマウントを有効にする手順。	86 ページの「WebNFS アクセスを有効にする方法」

表 5–4 WebNFS 管理の作業マップ (続き)

作業	説明	参照先
ファイアウォール経由で WebNFS を有効にします	WebNFS プロトコルを使用して、ファイアウォール経由でファイルへのアクセスを許可する手順。	105 ページの「ファイアウォール経由で WebNFS アクセスを有効にする方法」
NFS URL を使ってブラウズします	Web ブラウザ内での NFS URL の使用についての説明。	104 ページの「NFS URL を使ってブラウズする方法」
autofs で公開ファイルハンドルを使用します	オートマウンタでファイルシステムをマウントする場合に、公開ファイルハンドルの使用を強制するための手順。	120 ページの「autofs で公開ファイルハンドルを使用する方法」
autofs で NFS URL を使用します	オートマウンタマップに NFS URL を追加するための手順。	120 ページの「autofs で NFS URL を使用する方法」
ファイアウォールを越えてファイルシステムにアクセスを提供します	WebNFS プロトコルでファイアウォールを越えてファイルシステムへのアクセスを許可する手順。	93 ページの「ファイアウォールを越えて NFS ファイルシステムをマウントする方法」
NFS URL を使ってファイルシステムをマウントします	NFS URL を使ってファイルシステムへのアクセスを許可する手順。このプロセスによって、MOUNT プロトコルを使用しないでファイルシステムへのアクセスが可能になります。	94 ページの「NFS URL を使用して NFS ファイルシステムをマウントする方法」

WebNFS アクセスの計画

WebNFS を使用するにはまず、`nfs://server/path` のような NFS URL を実行し、読み込めるアプリケーションが必要です。次に、WebNFS アクセスのためにエクスポートするファイルシステムを選択します。アプリケーションが Web ブラウザの場合は、Web サーバーの文書のルートがよく使用されます。WebNFS アクセスのためにエクスポートするファイルシステムを選択するときは、次の事項を検討する必要があります。

1. サーバーには公開ファイルハンドルが 1 つずつあり、このハンドルはデフォルトではサーバーのルートファイルシステムに関連付けられています。NFS URL に示されたパスは、この公開ファイルハンドルが関連付けられているディレクトリからの相対パスとして評価されます。その結果としてパスが示す先のファイルまたはディレクトリが、エクスポートされたファイルシステムの中にあると、サーバーによってアクセスが実現されます。`share` コマンドの `public` オプションを使用すると、エクスポートされる特定のディレクトリにこの公開ファイルハンドルを関連付けることができます。このオプションを使用すると、URL はサーバーのルートファイルシステムではなく公開ファイルシステムからの相対パスになります。ルートファイルシステムを共有しないと、ルートファイルシステムへの Web アクセスはできません。

2. WebNFS 環境では、すでにマウント権限を持っているユーザーは、ブラウザからファイルにアクセスできます。ファイルシステムが public オプションを使ってエクスポートされているかどうかには関係ありません。ユーザーは NFS の設定によってファイルへのアクセス権を持っているため、ブラウザからのアクセスを許すことによって新たにセキュリティーが損なわれる恐れはありません。ファイルシステムをマウントできないユーザーは、public オプションを使ってファイルシステムを共有するだけで、WebNFS アクセスを使用できるようになります。
3. すでに公開されているファイルシステムは、public オプションを使用するのに適しています。たとえば、ftp アーカイブの最上位のディレクトリや Web サイトのメイン URL ディレクトリなどです。
4. share コマンドで index オプションを使用すると、HTML ファイルを強制的に読み込むことができます。そうしない場合は、NFS URL がアクセスされたときにディレクトリが一覧表示されます。

   ファイルシステムを選択したらファイルを確認し、必要に応じてファイルやディレクトリの表示を制限するようにアクセス権を設定します。アクセス権は、共有される NFS ファイルシステムに合わせて設定します。多くのサイトでは、ディレクトリに対しては 755、ファイルに対しては 644 が適切なアクセスレベルです。

   また、NFS と HTTP URL の両方を使用して 1 つの Web サイトにアクセスする場合は、その他の事項も検討する必要があります。これについては、203 ページの「Web ブラウザの使用と比較した場合の WebNFS の制約」で説明します。

NFS URL を使ってブラウズする方法

ブラウザが WebNFS サービスをサポートしている場合は、次のような NFS URL にアクセスできます。

nfs://*server*<:*port*>/*path*

server　ファイルサーバー名

port　使用するポート番号 (デフォルト値は 2049)

path　公開ファイルハンドルまたはルートファイルシステムに関連するファイルへのパス

注 – ほとんどのブラウザでは、前のトランザクションで使用した URL サービスのタイプ (nfs や http など) を次のトランザクションでも使用できます。ただし、異なるタイプのサービスを含む URL を読み込んだ場合に例外があります。NFS URL を使用したあとに、HTTP URL に対する参照が読み込まれたとします。その場合、次に続くページは NFS プロトコルではなく HTTP プロトコルを使って読み込まれます。

ファイアウォール経由で WebNFS アクセスを有効にする方法

ローカルのサブネットに属していないクライアントに対して WebNFS アクセスを有効にするには、ポート 2049 での TCP 接続を許可するようにファイアウォールを設定します。httpd に対してアクセスを許可するだけでは、NFS URL が使えるようにはなりません。

autofs 管理作業の概要

この節では、ユーザー自身の環境で遭遇する可能性のあるもっとも一般的な作業について説明します。各シナリオについて、ユーザーのクライアントで必要とする条件に最も適合するように autofs を設定するために推奨される手順も示します。この節で説明する作業を実行するには、Solaris 管理コンソールツールを使用するか、『Solaris のシステム管理 (ネーミングとディレクトリサービス : NIS+ 編)』を参照してください。

注 – Solaris 10 以降のリリースでは、/etc/default/autofs ファイルを使用して autofs 環境を設定することもできます。作業の詳細は、107 ページの「/etc/default/autofs ファイルを使用して autofs 環境を設定する」を参照してください。

---

autofs 管理の作業マップ

次の表に、autofs に関連する作業についての説明と参照箇所を示します。

表 5–5　autofs 管理の作業マップ

作業	説明	参照先
autofs を起動します	システムをリブートすることなく自動マウントサービスを起動します	96 ページの「オートマウンタを起動する方法」
autofs を停止します	他のネットワークサービスを使用不可にすることなく自動マウントサービスを停止します	96 ページの「オートマウンタを停止する方法」
/etc/default/autofs ファイルを使って autofs 環境を設定します	/etc/default/autofs ファイル内のキーワードに値を割り当てます	107 ページの「/etc/default/autofs ファイルを使用して autofs 環境を設定する」
autofs でファイルシステムにアクセスします	自動マウントサービスを使ってファイルシステムにアクセスします	90 ページの「オートマウンタによるマウント」

表 5–5 autofs 管理の作業マップ (続き)

作業	説明	参照先
autofs マップを修正します	他のマップを一覧表示するために使用されるマスターマップの修正を行う手順	109 ページの「マスターマップを修正する方法」
	ほとんどのマップに対して使用される間接マップの修正を行う手順	110 ページの「間接マップを修正する方法」
	クライアント上のマウントポイントとサーバー間の直接の関係が必要な場合に使用される直接マップの修正を行う手順	110 ページの「直接マップを修正する方法」
非 NFS ファイルシステムにアクセスするために autofs マップを修正します	CD-ROM アプリケーション用のエントリで autofs マップを設定する手順	111 ページの「autofs で CD-ROM アプリケーションにアクセスする方法」
	PC-DOS フロッピーディスク用のエントリで autofs マップの設定を行う手順	112 ページの「autofs で PC-DOS データフロッピーディスクにアクセスする方法」
	autofs を使用して CacheFS ファイルシステムにアクセスする手順	113 ページの「CacheFS を使用して NFS ファイルシステムにアクセスする方法」
/home を使用します	共通の /home マップの設定方法の例	114 ページの「/home の共通表示の設定」
	複数のファイルシステムを参照する /home マップを設定する手順	114 ページの「複数のホームディレクトリファイルシステムで /home を設定する方法」
新しい autofs マウントポイントを使用します	プロジェクト関連の autofs マップを設定する手順	115 ページの「/ws 下のプロジェクト関連ファイルを統合する方法」
	異なるクライアントアーキテクチャーをサポートする autofs マップを設定する手順	117 ページの「共有名前空間にアクセスするために異なるアーキテクチャーを設定する方法」
	異なるオペレーティングシステムをサポートする autofs マップを設定する手順	118 ページの「非互換のクライアントオペレーティングシステムのバージョンをサポートする方法」
autofs でファイルシステムを複製します	フェイルオーバーしたファイルシステムへのアクセスを提供します	118 ページの「複数のサーバーを通じて共用ファイルを複製する方法」
autofs でセキュリティー制限を使用します	ファイルへのリモート root アクセスを制限する一方でファイルシステムへのアクセスを提供します	119 ページの「autofs セキュリティー制限を適用する方法」
autofs で公開ファイルハンドルを使用します	ファイルシステムのマウント時に公開ファイルハンドルの使用を強制します	120 ページの「autofs で公開ファイルハンドルを使用する方法」

表 5–5　autofs 管理の作業マップ　　(続き)

作業	説明	参照先
autofs で NFS URL を使用します	オートマウンタが使用できるように、NFS URL を追加します	120 ページの「autofs で NFS URL を使用する方法」
autofs のブラウズ機能を無効にします	autofs マウントポイントが 1 つのクライアント上で自動的に生成されないように、ブラウズ機能を無効にする手順	121 ページの「1 つの NFS クライアントの autofs ブラウズ機能を完全に無効にする方法」
	autofs マウントポイントがすべてのクライアント上で自動的に生成されないように、ブラウズ機能を無効にする手順	121 ページの「すべてのクライアントの autofs ブラウズ機能を無効にする方法」
	特定の autofs マウントポイントがある 1 つのクライアント上で自動的に生成されないように、ブラウズ機能を無効にする手順	121 ページの「選択したファイルシステムの autofs ブラウズ機能を無効にする方法」

/etc/default/autofs ファイルを使用して autofs 環境を設定する

Solaris 10 以降のリリースでは、/etc/default/autofs ファイルを使用して autofs 環境を設定することができます。特に、このファイルにより、autofs コマンドおよび autofs デーモンを設定する方法が追加されました。コマンド行と同じように、この設定ファイルで指定できます。指定するには、キーワードに値を割り当てます。詳細は、141 ページの「/etc/default/autofs ファイル」を参照してください。

次の手順は、/etc/default/autofs ファイルの使用方法を示しています。

▼ /etc/default/autofs ファイルを使用する方法

1. スーパーユーザーになるか、同等の役割を引き受けます。

   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2. /etc/default/autofs ファイル内でエントリを追加または変更します。

   たとえば、すべての autofs マウントポイントの表示をオフに設定するには、次の行を追加します。

   AUTOMOUNTD_NOBROWSE=ON

   このキーワードは、-automountd コマンドの n 引数と同等です。キーワードの一覧については、141 ページの「/etc/default/autofs ファイル」を参照してください。

3 **autofs デーモンを再起動します。**

次のコマンドを入力します。

```
# svcadm restart system/filesystem/autofs
```

マップの管理作業

次の表は、autofs マップの管理時に認識しておく必要のある事項について示しています。選択したマップのタイプおよびネームサービスにより、autofs マップへの変更を行うために使用する必要があるメカニズムが異なります。

次の表に、マップのタイプとその使用方法を示します。

表 5–6 autofs マップのタイプとその使用方法

マップのタイプ	用途
マスター	ディレクトリをマップに関連付けます
直接	autofs を特定のファイルシステム向けにします
間接	autofs をリファレンス指向のファイルシステム向けにします

次の表では、使用しているネームサービスごとの、autofs 環境の変更方法を示しています。

表 5–7 マップの保守

ネームサービス	メソッド
ローカルファイル	テキストエディタ
NIS	`make` ファイル
NIS+	`nistbladm`

次の表に、マップのタイプに対して行なった修正に応じた `automount` コマンドの実行について示します。たとえば、直接 (direct) マップに対する追加または削除を行なった場合、ローカルシステム上で `automount` コマンドを実行する必要があります。automount コマンドを実行すると、変更が反映されます。ただし、既存のエントリを修正した場合は、変更を反映するために `automount` コマンドを実行する必要はありません。

表 5–8 automount コマンドを実行する場合

マップのタイプ	automount を再実行するか否か	
	追加または削除	修正
auto_master	Y	Y
direct	Y	N
indirect	N	N

マップの修正

次の手順は、複数の種類のオートマウンタマップを更新する方法を示します。ネームサービスとして NIS+ を使用する必要があります。

▼ マスターマップを修正する方法

1. マップを変更する権限を持つユーザーとしてログインします。
2. nistbladm コマンドを使用して、マスターマップへの変更を行います。
   『Solaris のシステム管理 (ネーミングとディレクトリサービス : NIS+ 編)』を参照してください。
3. 各クライアントで、スーパーユーザーになるか、それと同等の役割になります。
   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。
4. 各クライアントで、automount コマンドを実行し、変更が反映されるようにします。
5. マップを変更したことを他のユーザーに通知します。
   他のユーザーがコンピュータ上でスーパーユーザーとして automount コマンドを実行できるように、通知が必要になります。automount コマンドは、実行時にマスターマップから情報を収集することに注意してください。

▼ 間接マップを修正する方法

1. マップを変更する権限を持つユーザーとしてログインします。
2. nistbladm コマンドを使用して、間接マップへの変更を行います。
   『Solaris のシステム管理 (ネーミングとディレクトリサービス : NIS+ 編)』を参照してください。変更は、マップを次に使用する時、つまり次回のマウント実行時に反映されることに注意してください。

▼ 直接マップを修正する方法

1. マップを変更する権限を持つユーザーとしてログインします。
2. nistbladm コマンドを使用して、直接マップに対する変更点の追加または削除を行います。
   『Solaris のシステム管理 (ネーミングとディレクトリサービス : NIS+ 編)』を参照してください。
3. マップを変更したことを他のユーザーに通知します。
   必要に応じ、他のユーザーがコンピュータ上でスーパーユーザーとして automount コマンドを実行できるように、通知が必要になります。

注 – 既存の直接マップエントリの内容の変更だけを行なった場合は、automount コマンドを実行する必要はありません。

たとえば、異なるサーバーから /usr/src ディレクトリがマウントされるように auto_direct マップを修正するとします。/usr/src がその時点でマウントされていない場合、/usr/src にアクセスするとすぐにその新しいエントリが反映されます。/usr/src がその時点でマウントされている場合、オートアンマウントが実行されるまで待ちます。その後、アクセスが可能になります。

注 – できるだけ間接マップを使用してください。間接マップは構築が容易であり、コンピュータのファイルシステムへの要求が少なくて済みます。また、間接マップは直接マップよりもマウントテーブル内のスペースを必要としません。

マウントポイントの重複回避

/src 上にマウントされたローカルなディスクパーティションがあり、ほかのソースディレクトリのマウントにもその autofs サービスを使用する場合、問題が発生する

可能性があります。マウントポイント /src を指定した場合、ユーザーがローカルパーティションにアクセスしようとするたびに、NFS サービスはそのローカルパーティションを非表示にします。

たとえば /export/src などの他の場所に、パーティションをマウントする必要があります。その後、次のようなエントリを /etc/vfstab に含める必要があります。

```
/dev/dsk/d0t3d0s5 /dev/rdsk/c0t3d0s5 /export/src ufs 3 yes -
```

このエントリは、auto_src にも必要です。

```
terra        terra:/export/src
```

terra はコンピュータ名です。

非 NFS ファイルシステムへのアクセス

autofs は NFS ファイル以外のファイルシステムもマウントすることができます。autofs は、フロッピーディスクや CD-ROM など、削除可能な媒体上のファイルをマウントします。通常は、Volume Manager を使って削除可能な媒体上のファイルをマウントすることになります。次の例では、autofs を利用してこのマウントがどのように行われるかを示します。Volume Manager と autofs は同時に動作することができないため、まず Volume Manager を終了してから次に示すエントリを使用する必要があります。

サーバーからファイルシステムのマウントを行う代わりに、ドライブに媒体を配置してマップから参照します。autofs を使用し非 NFS ファイルシステムにアクセスを行う場合は、次の手順を参照してください。

▼ autofs で CD-ROM アプリケーションにアクセスする方法

注 – ボリュームマネージャーを使用していない場合に、この手順を行なってください。

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 **autofs** マップを更新します。

次のような CD-ROM のファイルシステム用のエントリを追加します。

```
hsfs     -fstype=hsfs,ro     :/dev/sr0
```

マウントする CD-ROM 装置の名前が、コロンのあとに続けて表示されます。

▼ autofs で PC-DOS データフロッピーディスクにアクセスする方法

注 – ボリュームマネージャーを使用していない場合に、この手順を行なってください。

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 **autofs** マップを更新します。

次のようなフロッピーディスクのファイルシステム用のエントリを追加します。

```
pcfs     -fstype=pcfs     :/dev/diskette
```

CacheFS を使用して NFS ファイルシステムにアクセスする

キャッシュファイルシステム (CacheFS) は、汎用不揮発性キャッシュメカニズムで、小型で高速なローカルディスクを利用して、特定のファイルシステムのパフォーマンスを向上させます。たとえば、CacheFS を使用すると、NFS 環境のパフォーマンスを改善できます。

CacheFS は、異なるバージョンの NFS では違った動作をします。たとえば、クライアントとバックファイルシステムで NFS version 2 または version 3 が動作している場合、ファイルはクライアントのアクセス用にフロントファイルシステムにキャッシュされます。ただし、クライアントとサーバーの両方で NFS version 4 が動作している場合は、次のように機能します。クライアントが CacheFS のファイルへのアクセスを初めて要求するとき、要求は、フロント (またはキャッシュされた) ファイルシステムを省略して、バックファイルシステムに直接送られます。NFS version 4 では、ファイルはフロントファイルシステムにキャッシュされなくなりました。すべてのファイルアクセスは、バックファイルシステムから提供されます。また、ファイルはフロントファイルシステムにキャッシュされていないため、フロントファイルシステムに反映する CacheFS 固有のマウントオプションは無視されます。CacheFS 固有のマウントオプションはバックファイルシステムに適用しません。

注 – 初めてシステムを NFS version 4 に構成すると、キャッシュが動作しないことを示す警告がコンソールに表示されます。

▼ CacheFS を使用して NFS ファイルシステムにアクセスする方法

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 `cfsadmin` コマンドを実行して、ローカルディスク上にキャッシュディレクトリを作成します。

```
# cfsadmin -c /var/cache
```

3 適切なオートマウンタマップに `cachefs` エントリを追加します。

たとえば、次に示すエントリをマスターマップに追加すると、すべてのホームディレクトリがキャッシュされます。

```
/home auto_home -fstype=cachefs,cachedir=/var/cache,backfstype=nfs
```

次のエントリを `auto_home` マップに追加すると、`rich` という名称のユーザーのホームディレクトリのキャッシュだけが行われます。

```
rich -fstype=cachefs,cachedir=/var/cache,backfstype=nfs dragon:/export/home1/rich
```

注 – あとから検索されるマップ内のオプションは、先に検索されたマップ内のオプションを無効にします。そのため、最後に検出されたオプションが使用されます。前述の例では、`auto_home` マップに追加されたエントリにマスターマップのオプションを含むのは、変更が必要なオプションがあった場合だけです。

オートマウンタのカスタマイズ

オートマウンタマップの設定方法はいくつかあります。次に、オートマウンタマップをカスタマイズして簡単に使用できるディレクトリ構造を実現する方法について詳細に説明します。

/home の共通表示の設定

すべてのネットワークユーザーにとっての理想は、自分自身のホームディレクトリ、または他の人のホームディレクトリを /home の下に配置できるようにすることです。この表示方法は通常、クライアントでもサーバーでも、すべてのコンピュータを通じて共通です。

Solaris をインストールすると、常にマスターマップ /etc/auto_master もインストールされます。

```
# Master map for autofs
#
+auto_master
/net     -hosts     -nosuid,nobrowse
/home    auto_home  -nobrowse
```

auto_home 用のマップも、/etc の下にインストールされます。

```
# Home directory map for autofs
#
+auto_home
```

外部 auto_home マップに対する参照を除き、このマップは空になります。/home 下のディレクトリをすべてのコンピュータに対して共通にする場合、この /etc/auto_home マップは修正しないでください。すべてのホームディレクトリのエントリは、NIS または NIS+ のネームサービスファイルで表示されなくてはなりません。

注 – ユーザーは、各ホームディレクトリから setuid 実行可能ファイルを実行することが許可されていません。この制限がないと、すべてのユーザーがすべてのコンピュータ上でスーパーユーザーの権限を持つことになります。

▼ 複数のホームディレクトリファイルシステムで /home を設定する方法

1. スーパーユーザーになるか、同等の役割を引き受けます。

   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2. /export/home の下にホームディレクトリパーティションをインストールします。

   システムに複数のパーティションがある場合は、/export/home1、/export/home2 のように、別のディレクトリにそれぞれインストールを行います。

3 **Solaris 管理コンソールツールを使用して、auto_home マップを作成して維持します。**

新しいユーザーアカウントを作成する場合は、そのユーザーのホームディレクトリの場所を auto_home マップに入力します。マップのエントリは、次のように単純な形式にすることができます。

```
rusty        dragon:/export/home1/&
gwenda       dragon:/export/home1/&
charles      sundog:/export/home2/&
rich         dragon:/export/home3/&
```

マップキーの代替となる & (アンパサンド) の使い方に注意してください。このアンパサンドは、次の例の 2 つ目の rusty の使用を省略した形式です。

```
rusty         dragon:/export/home1/rusty
```

auto_home マップを配置すると、ユーザーは、/home/*user* というパスを使用して、ユーザー自身のホームディレクトリを含むあらゆるホームディレクトリを参照できます。*user* はログイン名で、マップ内でのキーになります。すべてのホームディレクトリを共通に表示するしくみは、他のユーザーのコンピュータにログインする場合に便利です。autofs は、ユーザー自身のホームディレクトリをマウントします。同様に、他のコンピュータ上でリモートのウィンドウシステムクライアントを実行するとウィンドウシステムクライアントと同じ /home ディレクトリが表示されます。

この共通表示は、サーバーにも拡張されています。前の例を使用すれば、rusty がサーバー dragon にログインする場合、autofs は、/export/home1/rusty を /home/rusty にループバックマウントすることにより、ローカルディスクへの直接アクセスを提供します。

ユーザーは、各ホームディレクトリの実際の位置を意識する必要はありません。rusty がさらにディスク容量を必要とし、自身のホームディレクトリを他のサーバーに再配置する必要がある場合には、単純な変更で十分です。新しい場所を反映するように auto_home マップ内の rusty のエントリを変更することだけが必要になります。他のユーザーは、/home/rusty パスを継続して使用することができます。

▼ /ws 下のプロジェクト関連ファイルを統合する方法

大規模なソフトウェア開発プロジェクトの管理者を想定してください。そこで、プロジェクト関連のファイルをすべて /ws というディレクトリの下で利用できるようにすると仮定します。このようなディレクトリは、そのサイトのすべてのワークステーションで共通である必要があります。

1 **/ws ディレクトリに対するエントリを、サイトの NIS または NIS+ の auto_master マップに追加します。**

```
/ws      auto_ws      -nosuid
```

auto_ws マップが、/ws ディレクトリの内容を決定します。

2 -nosuid オプションを用心のために追加しておきます。

このオプションは、すべての作業空間に存在する可能性のある setuid プログラムをユーザーが実行できないようにします。

3 auto_ws マップにエントリを追加します。

auto_ws マップは、各エントリがサブプロジェクトを記述するように構成されています。最初の操作により、マップが次のようになります。

```
compiler   alpha:/export/ws/&
windows    alpha:/export/ws/&
files      bravo:/export/ws/&
drivers    alpha:/export/ws/&
man        bravo:/export/ws/&
tools      delta:/export/ws/&
```

各エントリの最後のアンパサンド (&) は、エントリキーを省略したものです。たとえば、最初のエントリは次のエントリと同じ意味です。

```
compiler        alpha:/export/ws/compiler
```

この最初の操作により、マップはシンプルなものになりますが、このマップでは不十分です。プロジェクトのオーガナイザーが、man エントリ内のドキュメントを各サブプロジェクトの下のサブディレクトリとして提供しようとしているとします。さらに、各サブプロジェクトは、ソフトウェアの複数のバージョンを記述するために、複数のサブディレクトリを必要とします。この場合、サーバー上のディスクパーティション全体に対して、これらのサブディレクトリをそれぞれ割り当てる必要があります。

次のように、マップ内のエントリを修正してください。

```
compiler \
    /vers1.0    alpha:/export/ws/&/vers1.0 \
    /vers2.0    bravo:/export/ws/&/vers2.0 \
    /man        bravo:/export/ws/&/man
windows \
    /vers1.0    alpha:/export/ws/&/vers1.0 \
    /man        bravo:/export/ws/&/man
files \
    /vers1.0    alpha:/export/ws/&/vers1.0 \
    /vers2.0    bravo:/export/ws/&/vers2.0 \
    /vers3.0    bravo:/export/ws/&/vers3.0 \
    /man        bravo:/export/ws/&/man
drivers \
    /vers1.0    alpha:/export/ws/&/vers1.0 \
    /man        bravo:/export/ws/&/man
tools \
    /           delta:/export/ws/&
```

現在のマップはかなり長くなっていますが、まだ 5 つのエントリを含んでいるだけです。各エントリは、複数のマウントがあるために長くなっています。たとえば、/ws/compiler に対する参照は、vers1.0、vers2.0、および man ディレクトリ用に

3 つのマウントを必要とします。各行の最後のバックスラッシュは、エントリが次の行まで続いていることを autofs に伝えるものです。実際、エントリは 1 つの長い行となっていますが、行ブレークやインデントのいくつかはエントリを読みやすくする目的で使用されています。tools ディレクトリには、すべてのサブプロジェクトに対するソフトウェア開発ツールが含まれているため、同じサブディレクトリ構造の対象とはなっていません。tools ディレクトリは単一のマウントのままです。

この配置は、システムの管理者に大きな柔軟性を提供します。ソフトウェアプロジェクトでは、非常に大きなディスクスペースを消費します。プロジェクトのすべての過程を通じて、さまざまなディスクパーティションを再配置し、拡張することになる可能性もあります。このような変更が auto_ws マップに反映される場合は、/ws 下のディレクトリ階層構造が変更されることもなく、ユーザーに対する通知の必要はありません。

サーバー alpha と bravo が同一の autofs マップを参照するため、それらのコンピュータにログインするすべてのユーザーは期待通りに /ws 名前空間を確認できます。このようなユーザーには、NFS マウントではなく、ループバックマウントを通じてのローカルファイルへの直接アクセスが提供されます。

▼ 共有名前空間にアクセスするために異なるアーキテクチャーを設定する方法

表計算アプリケーションやワードプロセッサパッケージのようなローカルの実行可能ファイルやアプリケーションについて、共有名前空間を作成する必要があります。この名前空間のクライアントは、異なる実行可能フォーマットを必要とする複数の異なるワークステーションアーキテクチャーを使用します。また、ワークステーションには、異なるリリースのオペレーティングシステムを使用するものもあります。

1. auto_local マップを作成します。

   『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』を参照してください。

2. 共有名前空間について、サイト固有の名称を 1 つ選択します。

   この名称により、その名前空間に属するファイルとディレクトリが簡単に識別できるようになります。たとえば、その名称として /usr/local を選択した場合、/usr/local/bin パスは明らかにこの名前空間の一部です。

3. ユーザーのコミュニティー識別を簡単にするため、**autofs** 間接マップを作成します。

   autofs 間接マップを /usr/local にマウントします。NIS の auto_master マップ内で、次のエントリを設定します。

   ```
   /usr/local      auto_local      -ro
   ```

なお、-ro マウントオプションは、クライアントがファイルやディレクトリのすべてに対して書き込みができないことを示してます。

4 サーバー上の任意のディレクトリをエクスポートします。

5 **auto_local** マップ内に bin エントリを 1 つ含めます。

ディレクトリ構造は、次のようになります。

```
bin      aa:/export/local/bin
```

6 (省略可能) 異なるアーキテクチャーのクライアントを処理するため、**autofs** CPU 変数を加えて、エントリの変更を行います。

```
bin      aa:/export/local/bin/$CPU
```

- SPARC クライアント – 実行可能ファイルを /export/local/bin/sparc に配置します。
- x86 クライアント – 実行可能ファイルを /export/local/bin/i386 に配置します。

▼ 非互換のクライアントオペレーティングシステムのバージョンをサポートする方法

1 クライアントのオペレーティングシステムのタイプを決定する変数と、アーキテクチャータイプを結合します。

autofs OSREL 変数と CPU 変数を結合して、CPU タイプと OS リリースの両方を示す名前を作成することができます。

2 次のようなマップエントリを作成します。

```
bin      aa:/export/local/bin/$CPU$OSREL
```

SunOS 5.6 を動作させているクライアントについて、次のファイルシステムをエクスポートします。

- SPARC クライアント – /export/local/bin/sparc5.6 をエクスポートします。
- x86 クライアント – /export/local/bin/i3865.6 に実行可能ファイルを配置します。

▼ 複数のサーバーを通じて共用ファイルを複製する方法

読み取り専用の複製されたファイルシステムを共有する最良の方法は、フェイルオーバーの利用です。フェイルオーバーについての説明は、197 ページの「クライアント側フェイルオーバー機能」を参照してください。

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 **autofs** マップ内のエントリを修正します。

すべての複製サーバーのリストを、コンマ区切りのリストとして、次のように作成します。

```
bin      aa,bb,cc,dd:/export/local/bin/$CPU
```

autofs は、最も近いサーバーを選択します。サーバーが複数のネットワークインタフェースを持っている場合は、各インタフェースのリストを作成してください。autofs はクライアントに最も近接したインタフェースを選択し、NFS トラフィックの不必要なルーティングを避けるようにしています。

▼ autofs セキュリティー制限を適用する方法

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 **NIS** または **NIS+** のネームサービス `auto_master` ファイル内に次のようなエントリを作成します。

```
/home      auto_home      -nosuid
```

`nosuid` オプションは、`setuid` または `setgid` ビットを設定したファイルをユーザーが作成できないようにします。

このエントリは、汎用ローカルファイル `/etc/auto_master` 内の `/home` のエントリを無効にします。前述の例を参照してください。これは、`+auto_master` が、ファイル内の `/home` エントリより先に、外部のネームサービスマップを参照するためです。`auto_home` マップ内のエントリにマウントオプションがある場合、`nosuid` オプションは無効になります。そのため、`auto_home` マップ内でオプションを使用しないようにするか、`nosuid` オプションを各エントリに含める必要があります。

注 – サーバー上の `/home` またはその下に、ホームディレクトリのディスクパーティションをマウントしないでください。

▼ autofs で公開ファイルハンドルを使用する方法

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 **autofs** マップに、次のようなエントリを作成します。

```
/usr/local      -ro,public     bee:/export/share/local
```

`public` オプションは、公開ハンドルの使用を強制します。NFS サーバーが公開ファイルハンドルをサポートしない場合、マウントは失敗します。

▼ autofs で NFS URL を使用する方法

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 次のような **autofs** エントリを作成します。

```
/usr/local      -ro     nfs://bee/export/share/local
```

サービスは、NFS サーバー上で公開ファイルハンドルの使用を試みます。サーバーが公開ファイルハンドルをサポートしない場合、MOUNT プロトコルが使用されます。

autofs のブラウズ機能を無効にする

Solaris 2.6 から、インストールされる `/etc/auto_master` のデフォルトバージョンには、`-/home` と `/net` 用のエントリに追加された `nobrowse` オプションが含まれます。さらに、アップグレード手順により、`/home` と `/net` のエントリが修正されていない場合は、`-nobrowse` オプションがそれらのエントリに追加されます。ただし、このような変更を手動で加えるか、あるいはインストール後にサイト固有の autofs マウントポイントに対するブラウズ機能をオフにすることが必要な場合もあります。

ブラウズ機能をオフにする方法はいくつかあります。`automountd` デーモンに対してコマンド行オプションを使用してブラウズ機能を無効にすると、そのクライアントに対する autofs ブラウズ機能は完全に無効になります。あるいは、NIS 名前空間または NIS+ 名前空間の autofs マップを使用して、すべてのクライアントにおける各マップエントリのブラウズ機能を無効にします。また、ネットワーク規模の名前空間を使用していない場合は、ローカルな autofs を使用して、各クライアントにおける各マップエントリのブラウズ機能を無効にすることができます。

▼ 1 つの NFS クライアントの autofs ブラウズ機能を完全に無効にする方法

1 NFS クライアント上で、スーパーユーザー、またはそれと同等の役割になります。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 /etc/default/autofs ファイルを編集して、次のキーワードと値を追加します。

```
AUTOMOUNTD_NOBROWSE=TRUE
```

3 autofs サービスを再起動します。

```
# svcadm restart system/filesystem/autofs
```

▼ すべてのクライアントの autofs ブラウズ機能を無効にする方法

すべてのクライアントに対するブラウズ機能を無効にするには、NIS または NIS+ のようなネームサービスを使用する必要があります。それ以外の場合には、各クライアント上でオートマウンタマップを手動で編集する必要があります。この例では、/home ディレクトリのブラウズ機能が無効にされています。無効にする必要がある各間接 autofs ノードに対して、この手順を実行してください。

1 ネームサービス auto_master ファイル内の /home エントリに -nobrowse オプションを追加します。

```
/home     auto_home     -nobrowse
```

2 すべてのクライアント上で、automount コマンドを実行します。
新規の動作は、クライアントシステム上で automount コマンドを実行した後、またはリブートした後に反映されます。

```
# /usr/sbin/automount
```

▼ 選択したファイルシステムの autofs ブラウズ機能を無効にする方法

この例では、/net ディレクトリのブラウズ機能を無効にします。/home または他の autofs マウントポイントにも、同じ手順を使用できます。

1. automount エントリが /etc/nsswitch.conf にあることを確認します。

   優先するローカルファイルエントリについては、ネームサービススイッチファイル内のエントリがネームサービスの前に files を一覧表示する必要があります。次に例を示します。

   ```
   automount:  files nis
   ```

   これは、標準的な Solaris にインストールされるデフォルトの構成を示します。

2. /etc/auto_master 内の +auto_master エントリの位置を確認します。

   名前空間内のエントリに優先するローカルファイルへの追加については、+auto_master エントリが /net の下に移動されている必要があります。

   ```
   # Master map for automounter
   #
   /net    -hosts     -nosuid
   /home   auto_home
   /xfn    -xfn
   +auto_master
   ```

   標準的な構成では、+auto_master エントリがファイルの先頭に配置されます。このように配置することにより、ローカルな変更が使用されなくなります。

3. /etc/auto_master ファイル内の /net エントリに nobrowse オプションを追加します。

   ```
   /net     -hosts     -nosuid,nobrowse
   ```

4. すべてのクライアント上で、automount コマンドを実行します。

   新規の動作は、クライアントシステム上で automount コマンドを実行した後、またはリブートした後に反映されます。

   ```
   # /usr/sbin/automount
   ```

NFS のトラブルシューティングの方法

NFS の問題を追跡するときは、問題が発生する可能性があるのは主に、サーバー、クライアント、およびネットワークであることを覚えておいてください。この節で説明するのは、個々の構成要素を切り離して、正常に動作しない部分を見つけ出そうというものです。リモートマウントを正常に実行するには、サーバー上で mountd デーモンと nfsd デーモンが動作している必要があります。

デフォルトでは、すべてのマウントに -intr オプションが設定されます。プログラムが「server not responding」(サーバーが応答しません) というメッセージを出してハングアップした場合、キーボード割り込み (Ctrl-C) で終了できます。

ネットワークまたはサーバーに問題がある場合、ハードマウントされたリモートファイルにアクセスするプログラムの障害と、ソフトマウントされたリモートファイルにアクセスするプログラムの障害とは異なります。ハードマウントされたリモートファイルシステムの場合、クライアントのカーネルは、サーバーがふたた

び応答するまで要求を再試行します。ソフトマウントされたリモートファイルシステムの場合、クライアントのシステムコールは、しばらく試行した後にエラーを返します。このエラーによって予想外のアプリケーションエラーやデータ破壊が発生する恐れがあるため、ソフトマウントは行わないでください。

ファイルシステムがハードマウントされていると、サーバーが応答に失敗した場合は、これにアクセスしようとするプログラムはハングアップします。この場合、NFS は次のメッセージをコンソールに表示します。

```
NFS server hostname not responding still trying
```

サーバーが少し後に応答すると、次のメッセージがコンソールに表示されます。

```
NFS server hostname ok
```

サーバーが応答しないような、ソフトマウントされたファイルシステムにアクセスしているプログラムは、次のメッセージを表示します。

```
NFS operation failed for server hostname: error # (error-message)
```

注 – 読み取りと書き込みをするデータを持つファイルシステム、または実行可能ファイルを持つファイルシステムは、ソフトマウントしないでください。エラーが発生する可能性があります。アプリケーションがそのようなソフトエラーを無視すれば、書き込み可能なデータが破壊される恐れがあります。またマウントされた実行可能ファイルが正常にロードされず、動作も正常に行われない可能性があります。

NFS のトラブルシューティングの手順

NFS サービスがエラーになった場所を判断するには、いくつかの手順を踏まなければなりません。次の項目をチェックしてください。

- クライアントがサーバーに到達できるかどうか
- クライアントがサーバー上の NFS サービスを受けられるかどうか
- NFS サービスがサーバー上で動作しているかどうか

上記の項目をチェックする過程で、ネットワークのほかの部分が機能していないことに気付く場合があります。たとえば、ネームサービスやネットワークのハードウェアが機能していない場合があります。複数のネームサービスでのデバッグ手順については、『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』で説明しています。また、上記の項目をチェックする過程で、クライアント側には問題がないことが判明することもあります。たとえば、作業領域のすべてのサブネットから、少なくとも 1 つの障害が発生したことが通知さ

れた場合などです。このような場合は、問題がサーバーかサーバー周辺のネットワークハードウェアで発生しているとみなし、クライアントではなく、サーバーでデバッグを開始する必要があります。

▼ NFS クライアントの接続性を確認する方法

1 クライアントから NFS サーバーに到達できることを確認します。クライアントで次のコマンドを入力します。

```
% /usr/sbin/ping bee
bee is alive
```

コマンドを入力した結果、サーバーが動作していることがわかったら、NFS サーバーをリモートで確認します。125 ページの「NFS サーバーをリモートで確認する方法」を参照してください。

2 クライアントからサーバーに到達できない場合は、ローカルネームサービスが動作していることを確認します。

NIS+ クライアントで次のコマンドを入力します。

```
% /usr/lib/nis/nisping -u
Last updates for directory eng.acme.com. :
Master server is eng-master.acme.com.
        Last update occurred at Mon Jun  5 11:16:10 1995

Replica server is eng1-replica-58.acme.com.
        Last Update seen was Mon Jun  5 11:16:10 1995
```

3 ネームサービスが実行されている場合は、クライアントが正しいホスト情報を受け取るために次のように入力します。

```
% /usr/bin/getent hosts bee
129.144.83.117    bee.eng.acme.com
```

4 ホスト情報に誤りがなく、クライアントからサーバーに接続できない場合は、別のクライアントから ping コマンドを実行します。

別のクライアントから実行したコマンドが失敗したら、126 ページの「サーバーで NFS サービスを確認する方法」を参照してください。

5 別のクライアントとサーバーがソフトウェア的に接続されている場合は、ping コマンドを使用して元のクライアントとローカルネット上の他のシステムとの接続性を確認します。

このコマンドが失敗する場合は、そのクライアントのネットワークソフトウェアの構成を確認します (/etc/netmasks、/etc/nsswitch.conf など)。

6 (省略可能) rpcinfo コマンドの出力を確認します。

rpcinfo コマンドを使用しても「program 100003 version 4 ready and waiting」と表示されない場合は、NFS version 4 がサーバー上で有効になっていません。NFS version 4 の有効化については、表 5–3 を参照してください。

7 ソフトウェアに問題がない場合は、ネットワークハードウェアを確認します。

クライアントをネットワークの別の場所へ移動して確認します。

▼ NFS サーバーをリモートで確認する方法

NFS version 4 のサーバーを使用している場合は、UDP と MOUNT プロトコルをサポートする必要がないことに注意してください。

1 NFS サーバーで NFS サービスが実行されていることを、次のコマンドを入力して確認します。

```
% rpcinfo -s bee|egrep 'nfs|mountd'
 100003  3,2    tcp,udp,tcp6,upd6                nfs     superuser
 100005  3,2,1  ticots,ticotsord,tcp,tcp6,ticlts,udp,upd6  mountd  superuser
```

デーモンが起動していない場合は、127 ページの「NFS サービスを再起動する方法」を参照してください。

2 サーバーで nfsd プロセスが応答することを確認します。

クライアント上で、次のコマンドを入力し、サーバーからの UDP NFS 接続をテストします。

```
% /usr/bin/rpcinfo -u bee nfs
program 100003 version 2 ready and waiting
program 100003 version 3 ready and waiting
```

注 – NFS version 4 は、UDP をサポートしません。

サーバーが動作している場合、プログラムとバージョン番号が表示されます。-t オプションを使用すると、TCP 接続を検査できます。上記コマンドでエラーになる場合は、126 ページの「サーバーで NFS サービスを確認する方法」に進んでください。

3 サーバーで mountd が応答することを、次のコマンドを入力して確認します。

```
% /usr/bin/rpcinfo -u bee mountd
program 100005 version 1 ready and waiting
program 100005 version 2 ready and waiting
program 100005 version 3 ready and waiting
```

サーバーが動作している場合は、UDP プロトコルに関連しているプログラムとそのバージョン番号が出力されます。-t オプションを使用すると、TCP 接続を検査できます。エラーになる場合は、126 ページの「サーバーで NFS サービスを確認する方法」に進んでください。

4 ローカル **autofs** サービスを使用していた場合は、そのサービスを確認します。

```
% cd /net/wasp
```

/net か /home マウントポイントのうち、適切に動作する方を確認します。エラーになる場合は、次のコマンドを root としてクライアントから入力し、autofs サービスを再起動します。

```
# svcadm restart system/filesystem/autofs
```

5 サーバーのファイルシステムの共有が正常に行えることを確認します。

```
% /usr/sbin/showmount -e bee
/usr/src                                    eng
/export/share/man                        (everyone)
```

サーバーの項目とローカルマウントエントリにエラーがないことをチェックします。名前空間も確認します。この例で最初のクライアントが eng ネットグループの中にない場合、/usr/src ファイルシステムはマウントできません。

すべてのローカルファイルを調べて、マウント情報を含むエントリをすべて検査します。リストには、/etc/vfstab とすべての /etc/auto_* ファイルが含まれています。

▼ サーバーで NFS サービスを確認する方法

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 サーバーがクライアントに到達できることを確認します。

```
# ping lilac
lilac is alive
```

3 サーバーからクライアントに到達できない場合は、ローカルネームサービスが動作していることを確認します。

NIS+ クライアントで次のコマンドを入力します。

```
% /usr/lib/nis/nisping -u
Last updates for directory eng.acme.com. :
Master server is eng-master.acme.com.
        Last update occurred at Mon Jun  5 11:16:10 1995

Replica server is eng1-replica-58.acme.com.
        Last Update seen was Mon Jun  5 11:16:10 1995
```

4 ネームサービスが動作している場合は、サーバーにあるネットワークソフトウェアの構成を確認します (/etc/netmasks、/etc/nsswitch.conf など)。

5 次のコマンドを入力し、rpcbind デーモンが動作していることを確認します。

```
# /usr/bin/rpcinfo -u localhost rpcbind
program 100000 version 1 ready and waiting
program 100000 version 2 ready and waiting
program 100000 version 3 ready and waiting
```

サーバーが動作している場合は、UDP プロトコルに関連しているプログラムとそのバージョン番号が出力されます。rpcbind がハングアップしたと思われる場合は、サーバーをリブートしてください。

6 次のコマンドを入力して、nfsd デーモンが動作していることを確認します。

```
# rpcinfo -u localhost nfs
program 100003 version 2 ready and waiting
program 100003 version 3 ready and waiting
# ps -ef | grep nfsd
root    232      1  0  Apr 07     ?     0:01 /usr/lib/nfs/nfsd -a 16
root   3127   2462  1  09:32:57  pts/3  0:00 grep nfsd
```

注 – NFS version 4 は、UDP をサポートしません。

サーバーが動作している場合は、UDP プロトコルに関連しているプログラムとそのバージョン番号が出力されます。rpcinfo に -t オプションを指定し、TCP 接続も確認します。これらのコマンドを使用するとエラーになる場合は、NFS サービスを再起動します。127 ページの「NFS サービスを再起動する方法」を参照してください。

7 次のコマンドを入力して、mountd デーモンが動作していることを確認します。

```
# /usr/bin/rpcinfo -u localhost mountd
program 100005 version 1 ready and waiting
program 100005 version 2 ready and waiting
program 100005 version 3 ready and waiting
# ps -ef | grep mountd
root    145      1 0 Apr 07  ?     21:57 /usr/lib/autofs/automountd
root    234      1 0 Apr 07  ?     0:04  /usr/lib/nfs/mountd
root   3084 2462 1 09:30:20 pts/3  0:00  grep mountd
```

サーバーが動作している場合は、UDP プロトコルに関連しているプログラムとそのバージョン番号が出力されます。rpcinfo に -t オプションを指定し、TCP 接続も確認します。これらのコマンドを使用するとエラーになる場合は、NFS サービスを再起動します。127 ページの「NFS サービスを再起動する方法」を参照してください。

▼ NFS サービスを再起動する方法

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 サーバー上で **NFS** サービスを再起動します。

次のコマンドを入力します。

```
# svcadm restart network/nfs/server
```

NFS ファイルサービスを提供しているホストを確認する方法

-m オプションを指定して nfsstat コマンドを実行し、最新の NFS 情報を取得します。現在のサーバー名は、「currserver=」のあとに表示されます。

```
% nfsstat -m
/usr/local from bee,wasp:/export/share/local
 Flags: vers=3,proto=tcp,sec=sys,hard,intr,llock,link,synlink,
        acl,rsize=32768,wsize=32678,retrans=5
 Failover: noresponse=0, failover=0, remap=0, currserver=bee
```

▼ mount コマンドに使用されたオプションを確認する方法

Solaris 2.6 およびそれ以降に出たパッチに置き換えられた mount コマンドでは、無効なオプションを指定しても警告されません。コマンド行に入力したオプション、または /etc/vfstab から指定したオプションが有効であるかどうかを判断するには、次の手順に従います。

たとえば、次のコマンドが実行されたとします。

```
# mount -F nfs -o ro,vers=2 bee:/export/share/local /mnt
```

1 次のコマンドを実行し、オプションを確認します。

```
% nfsstat -m
/mnt from bee:/export/share/local
Flags:  vers=2,proto=tcp,sec=sys,hard,intr,dynamic,acl,rsize=8192,wsize=8192,
        retrans=5
```

bee からマウントされたファイルシステムは、プロトコルのバージョンが 2 に設定されています。nfsstat コマンドを使用しても、一部のオプションの情報は表示されませんが、オプションを確認するにはこれが最も正確な方法です。

2 /etc/mnttab でエントリを確認します。

mount コマンドは、無効なオプションをマウントテーブルに追加することができません。そのため、mnttab ファイルに記述されているオプションとコマンド行のオプションが一致していることを確認してください。このようにすると、nfsstat コマンドにより報告されなかったオプションを特定することができます。

```
# grep bee /etc/mnttab
bee:/export/share/local /mnt nfs     ro,vers=2,dev=2b0005e 859934818
```

autofs のトラブルシューティング

autofs の使用時に、問題の発生することがあります。この節では、問題解決プロセスについてわかりやすく説明します。この節は、2 つのパートに分かれています。

この節では、autofs が生成するエラーメッセージのリストを示します。このリストは、2 つのパートに分かれています。

- automount の詳細形式 (-v) オプションにより生成されるエラーメッセージ
- 通常表示されるエラーメッセージ

各エラーメッセージの後には、そのメッセージの説明と考えられる原因が続きます。

トラブルシューティング時には、詳細形式 (-v) オプションで autofs プログラムを開始します。そうしないと、原因がわからないまま問題に遭遇することになります。

次の節は、autofs のエラー時に表示されがちなエラーメッセージと、生じうる問題についての説明です。

automount -v により生成されるエラーメッセージ

`bad key` *key* `in direct map` *mapname*
説明: 直接マップのスキャン中、autofs が接頭辞 / のないエントリキーを発見しました。

対処方法: 直接マップ内のキーは、フルパス名でなくてはなりません。

`bad key` *key* `in indirect map` *mapname*
説明: 間接マップのスキャン中、autofs が / を含むエントリキーを発見しました。

対処方法: 間接マップのキーは、パス名ではなく、単なる名称でなくてはなりません。

`can't mount` *server*`:` *pathname: reason*
説明: サーバー上のマウントデーモンが、*server*:*pathname* のファイルハンドルの提供を 拒否しました。

対処方法: サーバー上のエクスポートテーブルを確認してください。

`couldn't create mount point` *mountpoint*`:` *reason*
説明: autofs は、マウントに必要なマウントポイントを作成することができませんでした。この問題は、すべてのサーバーのエクスポートされたファイルシステムを階層的にマウントしようとする場合に頻繁に生じます。

対処方法: 必要なマウントポイントは、マウントできないファイルシステム内にだけ存在するため、ファイルシステムはエクスポートできません。エクスポートされる親ファイルシステムは、読み取り専用でエクスポートされるため、マウントポイントを作成できません。

`leading space in map entry` *entry* `text in` *mapname*
説明: autofs は自動マウントマップ内に先頭にスペースを含むエントリを発見しました。この問題は、通常、マップエントリが不当である場合に発生します。次に例を示します。

```
fake
/blat           frobz:/usr/frotz
```

対処方法: この例では、autofs が 2 つめの行を検出した場合に警告が生成されます。これは、最初の行がバックスラッシュ (\) で終端されていないためです。

mapname`: Not found`
説明: 必要とされるマップが配置されていません。このメッセージは、`-v` オプションが使用されている場合にだけ生成されます。

対処方法: マップ名のスペルとパス名を確認してください。

`remount` *server*`:` *pathname* `on` *mountpoint* `: server not responding`
説明: autofs が、アンマウントしたファイルシステムの再マウントに失敗しました。

対処方法: サポートが必要な場合は、ご購入先に連絡してください。このエラーメッセージが出力されることはほとんどなく、直接的な解決策はありません。

`WARNING:` *mountpoint* `already mounted on`
説明: autofs が、既存のマウントポイント上にマウントしようとしました。このメッセージは、autofs 内で内部エラー (異常) が生じたことを意味しています。

対処方法: サポートが必要な場合は、ご購入先に連絡してください。このエラーメッセージが出力されることはほとんどなく、直接的な解決策はありません。

その他のエラーメッセージ

`dir` *mountpoint* `must start with '/'`
対処方法: オートマウンタのマウントポイントは、フルパス名で指定しなくてはなりません。マウントポイントのスペルとパス名を確認してください。

`hierarchical mountpoints: pathname1 and pathname2`
対処方法: autofsは、マウントポイントが階層的な関係を持つことを許可しません。autofsマウントポイントは、他の自動マウントされたファイルシステムに含まれていてはなりません。

`host server not responding`
説明: autofsが、*server*で示されるサーバーにコンタクトしようとしましたが、応答がありません。

対処方法: NFSサーバーの状態を確認してください。

hostname`: exports: rpc-err`
説明: このエラーは、*hostname*からエクスポートリストを取得する場合に発生します。このメッセージは、サーバーまたはネットワークに問題があることを示します。

対処方法: NFSサーバーの状態を確認してください。

`map mapname, key key: bad`
説明: マップエントリが不適切な形式であり、autofsが処理できません。

対処方法: そのエントリを再確認してください。そのエントリに、エスケープする必要のある文字が含まれている可能性があります。

mapname`: nis-err`
説明: このエラーは、NISマップのエントリを参照する場合に発生します。このメッセージは、NISに問題がある可能性があることを示しています。

対処方法: NISサーバーの状態を確認してください。

`mount of server: pathname on mountpoint:reason`
説明: autofsがマウントに失敗しました。サーバーまたはネットワークに問題のある可能性があります。*reason*の文字列によって、問題が特定されます。

対処方法: サポートが必要な場合は、ご購入先に連絡してください。このエラーメッセージが出力されることはほとんどなく、直接的な解決策はありません。

mountpoint`: Not a directory`
説明: autofsは、ディレクトリではない*mountpoint*に示される場所に自分自身をマウントすることができません。

対処方法: マウントポイントのスペルとパス名を確認してください。

`nfscast: cannot send packet: reason`
説明: autofs が、複製されたファイルシステムの場所を示すリスト内にあるサーバーへの照会パケットを送信できません。*reason* の文字列によって、問題が特定されます。

対処方法: サポートが必要な場合は、ご購入先に連絡してください。このエラーメッセージが出力されることはほとんどなく、直接的な解決策はありません。

`nfscast: cannot receive reply: reason`
説明: autofs が、複製されたファイルシステムの場所を示すリスト内にあるいずれのサーバーからも応答を受けられません。*reason* の文字列によって、問題が特定されます。

対処方法: サポートが必要な場合は、ご購入先に連絡してください。このエラーメッセージが出力されることはほとんどなく、直接的な解決策はありません。

`nfscast: select: reason`
説明: このようなエラーメッセージはすべて、複製されたファイルシステムのサーバーに対して確認を実行した際に問題が発生したことを示します。このメッセージは、ネットワークに問題がある可能性があることを示しています。*reason* の文字列によって、問題が特定されます。

対処方法: サポートが必要な場合は、ご購入先に連絡してください。このエラーメッセージが出力されることはほとんどなく、直接的な解決策はありません。

`pathconf: no info for server:pathname`
説明: autofs が、パス名に関する `pathconf` 情報の取得に失敗しました。

対処方法: `fpathconf(2)` のマニュアルページを参照してください。

`pathconf: server : server not responding`
説明: autofs が、`pathconf()` に情報を提供する *server* に示されるサーバー上のマウントデーモンにコンタクトできませんでした。

対処方法: このサーバーで `POSIX` マウントオプションを使用しないでください。

autofs のその他のエラー

`/etc/auto*` ファイルが実行ビットセットを持っている場合、オートマウンタは次のようなメッセージを生成するマップの実行を試みます。

```
/etc/auto_home: +auto_home: not found
```

この場合、auto_home ファイルは不適切な権限をもつことになります。このファイル内の各エントリは、よく似たエラーメッセージを生成します。ファイルへのこのような権限は、次のコマンドを入力することにより取り消す必要があります。

```
# chmod 644 /etc/auto_home
```

NFS のエラーメッセージ

この節では、エラーメッセージとそのエラーを発生させる原因となった状態について説明し、1 つ以上の解決策を提供しています。

`Bad argument specified with index option - must be a file`
: **対処方法:** index オプションにはファイル名を指定する必要があります。ディレクトリ名は使用できません。

`Cannot establish NFS service over /dev/` *tcp*`: transport setup problem`
: **説明:** このメッセージは、名前空間の中のサービス情報が更新されなかったときによく出力されます。またこのメッセージは、UDP の状態を示すことがあります。

  対処方法: この問題を解決するには、名前空間の中のサービスデータを更新します。

  NIS+ の場合、エントリは次のとおりです。

```
nfsd nfsd tcp 2049 NFS server daemon
nfsd nfsd udp 2049 NFS server daemon
```

  NIS と /etc/services の場合、エントリは次のとおりです。

```
nfsd    2049/tcp    nfs    # NFS server daemon
nfsd    2049/udp    nfs    # NFS server daemon
```

`Cannot use index option without public option`
: **対処方法:** share コマンドの index オプションに public オプションを指定してください。index オプションを使用するには、公開ファイルハンドルを定義する必要があります。

注 – Solaris 2.5.1 では、share コマンドを使って公開ファイルハンドルを設定する必要があります。Solaris 2.6 リリースにおいて、公開ファイルハンドルはデフォルトでルート (/) に設定されるように変更されました。このエラーメッセージは出力されません。

`Could not start` *daemon* `:` *error*
: **説明:** このメッセージは、デーモンが異常終了するか、システムコールにエラーが発生した場合に表示されます。*error* の文字列によって、問題が特定されます。

対処方法: サポートが必要な場合は、ご購入先に連絡してください。このエラーメッセージが出力されることはほとんどなく、直接的な解決策はありません。

`Could not use public filehandle in request to` *server*
説明: このメッセージは、`public` オプションが指定されているにもかかわらず NFS サーバーが公開ファイルハンドルをサポートしていない場合に表示されます。この場合、マウントが失敗します。

対処方法: この問題を解決するには、公開ファイルハンドルを使用しないでマウント要求を行うか、NFS サーバーが公開ファイルハンドルをサポートするように設定し直します。

daemon `running already with pid` *pid*
説明: デーモンがすでに実行されています。

対処方法: 新たにデーモンを実行する場合は、現在のデーモンを終了し、新しいデーモンを開始します。

`error locking` *lock file*
説明: このメッセージは、デーモンに関連付けられている *lock file* を正しくロックできなかった場合に表示されます。

対処方法: サポートが必要な場合は、ご購入先に連絡してください。このエラーメッセージが出力されることはほとんどなく、直接的な解決策はありません。

`error checking` *lock file* `:` *error*
説明: このメッセージは、デーモンに関連付けられている *lock file* を正しく開くことができなかった場合に表示されます。

対処方法: サポートが必要な場合は、ご購入先に連絡してください。このエラーメッセージが出力されることはほとんどなく、直接的な解決策はありません。

`NOTICE: NFS3: failing over from` *host1* `to` *host2*
説明: このメッセージは、フェイルオーバーが発生するとコンソールに表示されます。報告のためだけのメッセージです。

対処方法: 何もする必要はありません。

filename`: File too large`
説明: NFS version 2 のクライアントが、2G バイトを超えるサイズのファイルにアクセスしようとしています。

対処方法: NFS version 2 を使用しないでください。 version 3 または version 4 を使用してファイルシステムをマウントします。`nolargefiles` オプションについては、160 ページの「NFS ファイルシステム用の `mount` オプション」を参照してください。

`mount: ... server not responding:RPC_PMAP_FAILURE - RPC_TIMED_OUT`
説明: 実行レベルの誤りか、`rpcbind` の停止かハングアップのため、マウント先のファイルシステムを共有しているサーバーがダウンしているかまたはそこに到達できません。

対処方法: サーバーがリブートするまで待機します。サーバーがハングアップしている場合は、サーバーをリブートします。

`mount: ... server not responding: RPC_PROG_NOT_REGISTERED`
説明: マウント要求が `rpcbind` によって登録されているにもかかわらず、NFS マウントデーモン (`mountd`) が登録されていません。

対処方法: サーバーがリブートするまで待機します。サーバーがハングアップしている場合は、サーバーをリブートします。

`mount: ... No such file or directory`
説明: リモートディレクトリもローカルディレクトリも存在しません。

対処方法: ディレクトリ名のスペルをチェックします。両方のディレクトリで `ls` コマンドを実行します。

`mount: ...: Permission denied`
説明: コンピュータ名が、クライアントのリストに載っていないか、マウントするファイルシステムにアクセスできるネットグループに含まれていません。

対処方法: `showmount -e` を実行し、アクセスリストを確認してください。

`NFS file temporarily unavailable on the server, retrying ...`
説明: NFS version 4 サーバーでは、ファイルの管理をクライアントに委託できます。このメッセージは、クライアントからの要求と重複するほかのクライアントへの委託を、サーバーが再発信していることを示します。

対処方法: サーバーがクライアントの要求を処理する前に、再発信が行われる必要があります。委託の詳細は、190 ページの「NFS version 4 における委託」を参照してください。

`NFS fsstat failed for server` *hostname*`: RPC: Authentication error`
説明: さまざまな状況で発生するエラーです。もっともデバッグが困難なのは、ユーザーの属しているグループが多すぎる場合です。現在、ユーザーは最大 16 個のグループに属すことができますが、NFS マウントでファイルにアクセスしている場合は、それよりも少なくなります。

対処方法: ただし、ユーザーが 17 個以上のグループに所属する必要がある場合の方法もあります。NFS サーバーおよび NFS クライアントで Solaris 2.5 リリース以降が動作している場合は、アクセス制御リストを使用して、必要なアクセス特権を与えることができます。

`nfs mount: ignoring invalid option “ -option”`
説明: *-option* フラグが無効です。

対処方法: 必要な構文を確認するには、`mount_nfs(1M)` のマニュアルページを参照してください。

注 – このエラーメッセージは、Solaris 2.6 以降、または Solaris 2.6 より前のバージョンにパッチを適用した状態で `mount` コマンドを実行したときには表示されません。

`nfs mount: NFS can’t support “nolargefiles”`
説明: NFS クライアントが、`-nolargefiles` オプションを使用して NFS サーバーからファイルシステムをマウントしようとしました。

対処方法: このオプションは、NFS ファイルシステムタイプに対してはサポートされていません。

`nfs mount: NFS V2 can’t support “largefiles”`
説明: NFS version 2 プロトコルでは、大規模ファイルを処理できません。

対処方法: 大規模ファイルを扱う必要がある場合は、version 3 または version 4 を使用してください。

`NFS server` *hostname* `not responding still trying`
説明: ファイル関連の作業中にプログラムがハングアップすると、NFS サーバーに障害が発生する可能性があります。このメッセージは、NFS サーバー (*hostname*) がダウンしているか、サーバーかネットワークに問題があることを示すものです。

対処方法: フェイルオーバー機能を使用している場合、*hostname* はサーバー名のリストになります。124 ページの「NFS クライアントの接続性を確認する方法」を参照してください。

`NFS server recovering`
説明: NFS version 4 サーバーのリブート中に、一部の操作が許可されませんでした。このメッセージは、サーバーがこの操作の続行を許可するまで、クライアントが待機していることを示します。

対処方法: 何もする必要はありません。サーバーが操作を許可するまで待機します。

`Permission denied`
説明: このメッセージは、次の理由により、`ls -l`、`getfacl`、および `setfacl` コマンドによって表示されます。

- NFS version 4 サーバー上のアクセス制御リスト (ACL) エントリ内に存在するユーザーまたはグループを、NFS version 4 クライアント上の有効なユーザーまたはグループにマッピングできない場合、ユーザーはクライアント上の ACL を読み取ることができない。
- NFS version 4 クライアント上で設定されている ACL エントリ内に存在するユーザーまたはグループを、NFS version 4 サーバー上の有効なユーザーまたはグループにマッピングできない場合、ユーザーはクライアント上の ACL に書き込みや変更を行うことができない。
- NFS version 4 のクライアントとサーバーで NFSMAPID_DOMAIN の値が一致しない場合、ID マッピングが失敗する。

詳細は、192 ページの「NFS version 4 での ACL と `nfsmapid`」を参照してください。

対処方法: 次の手順を実行してください。

- ACL エントリ内のすべてのユーザーおよびグループ ID がクライアントとサーバーの両方に存在することを確認します。
- NFSMAPID_DOMAIN の値が `/etc/default/nfs` ファイル内で正しく設定されていることを確認します。詳細は、142 ページの「`/etc/default/nfs` ファイルのキーワード」を参照してください。

ユーザーまたはグループをサーバーまたはクライアント上でマッピングできるかどうかを判断するには、194 ページの「ACL エントリ内のすべてのユーザーおよびグループ ID が NFS version 4 のクライアントとサーバーの両方に存在することを確認します。」にあるスクリプトを使用します。

`port` *number* `in nfs URL not the same as port` *number* `in port option`
説明: NFS URL のポート番号は、マウントの `-port` オプションのポート番号と一致していなければなりません。一致していないと、マウントは失敗します。

対処方法: 同じポート番号にしてコマンドを再実行するか、ポート番号の指定を省略してください。通常は、NFS URL と `-port` オプションの両方にポート番号を指定する必要はありません。

`replicas must have the same version`
説明: NFS フェイルオーバー機能が正しく機能するためには、複製の NFS サーバーが同じバージョンの NFS プロトコルをサポートしていなければなりません。

対処方法: 複数のバージョンが混在することは許されません。

`replicated mounts must be read-only`
説明: NFS フェイルオーバー機能は、読み書き可能としてマウントされたファイルシステムでは動作しません。ファイルシステムを読み書き可能としてマウントすると、ファイルが変更される可能性が高くなるためです。

対処方法: NFS のフェイルオーバー機能は、ファイルシステムがまったく同じであることが前提です。

`replicated mounts must not be soft`
説明: 複製されるマウントの場合、フェイルオーバーが発生するまでタイムアウトを待つ必要があります。

対処方法: `soft` オプションを指定すると、タイムアウトが開始してすぐにマウントが失敗するため、複製されるマウントには `-soft` オプションは指定できません。

`share_nfs: Cannot share more than one filesystem with 'public' option`
対処方法: `/etc/dfs/dfstab` ファイルを調べて、`-public` オプションによって共有するファイルシステムを複数選択していないか確認してください。公開ファイルハンドルは、サーバーあたり 1 つしか設定できません。したがって、public オプションで共有できるファイルシステムは 1 つだけです。

`WARNING: No network locking on` *hostname*`:` *path*`: contact admin to install server change`
説明: NFS クライアントが、NFS サーバー上のネットワークロックマネージャーと接続を確立できませんでした。この警告は、マウントできなかったことを知らせるためではなく、ロックが機能しないことを警告するために出力されます。

対処方法: サーバーを、ロックマネージャーを完全にサポートする新しいバージョンの OS にアップグレードします。

◆ ◆ ◆ 第 6 章

ネットワークファイルシステムへのアクセス(リファレンス)

この章では、NFS コマンドについて説明します。また、NFS 環境のさまざまな部分とそれらが互いにどのように連係するかについても説明します。

注 – システムでゾーンが有効なときに非大域ゾーンでこの機能を使用するには、『Oracle Solaris のシステム管理 (Oracle Solaris コンテナ : 資源管理と Oracle Solaris ゾーン)』を参照してください。

NFS ファイル

ファイルによっては、いずれのコンピュータ上でも NFS アクティビティーをサポートする必要があるファイルがあります。その多くは ASCII ファイルで、いくつかはデータファイルです。表 6–1 にこのようなファイルとその機能をまとめます。

表 6–1 NFS ファイル

ファイル名	機能
`/etc/default/autofs`	autofs 環境の構成情報を示します。
`/etc/default/fs`	ローカルファイルシステムにおけるデフォルトファイルシステムのタイプを示します。

表 6–1 NFS ファイル (続き)

ファイル名	機能
/etc/default/nfs	lockd および nfsd の構成情報を示します。詳細は、142 ページの「/etc/default/nfs ファイルのキーワード」および nfs(4) のマニュアルページを参照してください。
/etc/default/nfslogd	NFS サーバーログデーモン (nfslogd) の構成情報を示します。
/etc/dfs/dfstab	共有するローカルリソースを示します。
/etc/dfs/fstypes	リモートファイルシステムにおけるデフォルトファイルシステムのタイプを示します。
/etc/dfs/sharetab	共有されるローカルとリモートのリソースを示します。sharetab(4) のマニュアルページを参照してください。このファイルは編集しないでください。
/etc/mnttab	自動マウントしたディレクトリを含む、現在マウントしているファイルシステムを示します。mnttab(4) のマニュアルページを参照してください。このファイルは編集しないでください。
/etc/netconfig	トランスポートプロトコルを示します。このファイルは編集しないでください。
/etc/nfs/nfslog.conf	NFS サーバーログのための一般的な構成情報を示します。
/etc/nfs/nfslogtab	nfslogd によるログ後処理のための情報を示します。このファイルは編集しないでください。
/etc/nfssec.conf	NFS のセキュリティーサービスを示します。
/etc/rmtab	NFS クライアントがリモートでマウントしたファイルシステムを示します。rmtab(4) のマニュアルページを参照してください。このファイルは編集しないでください。
/etc/vfstab	ローカルにマウントするファイルシステムを定義します。vfstab(4) のマニュアルページを参照してください。

/etc/dfs/fstypes の最初のエントリは、リモートファイルシステムにおけるデフォルトファイルシステムのタイプとして利用されることがよくあります。このエントリは、NFS ファイルシステムのタイプをデフォルトとして定義します。

/etc/default/fs には、エントリが 1 つしかありません。ローカルディスクにおけるデフォルトファイルシステムのタイプです。クライアントやサーバーでサポートするファイルシステムのタイプは、/kernel/fs のファイルを確認して決定することができます。

/etc/default/autofs ファイル

Solaris 10 以降のリリースでは、/etc/default/autofs ファイルを使用して autofs 環境を設定することができます。特に、このファイルにより、autofs コマンドおよび autofs デーモンを設定する方法が追加されました。コマンド行と同じように、この設定ファイルで指定できます。ただし、コマンド行とは異なり、オペレーティングシステムのアップグレード中にも、このファイルは指定を保持します。さらに、autofs 環境の既存の動作が保持されているかを確認するために、クリティカルな起動ファイルを更新する必要がなくなります。指定を行うには、次のキーワードに値を割り当てます。

AUTOMOUNT_TIMEOUT
: ファイルシステムがアンマウントされるまでアイドル状態を持続する時間を設定します。このキーワードは、automount の -t 引数と同等です。デフォルト値は 600 です。

AUTOMOUNT_VERBOSE
: マウント、アンマウント、およびその他の重要でないイベントを通知します。このキーワードは、-automount の v 引数と同等です。デフォルトの値は FALSE です。

AUTOMOUNTD_VERBOSE
: 状態メッセージをコンソールに記録します。このキーワードは automountd デーモンの -v 引数と同等です。デフォルトの値は FALSE です。

AUTOMOUNTD_NOBROWSE
: すべての autofs マウントポイントのブラウズをオンまたはオフにします。このキーワードは -automountd の n 引数と同等です。デフォルトの値は FALSE です。

AUTOMOUNTD_TRACE
: 各遠隔手続き呼び出し (RPC) を拡張し、拡張された RPC を標準出力に表示します。このキーワードは、-automountd の T 引数と同等です。デフォルト値は 0 です。値の範囲は 0 から 5 です。

AUTOMOUNTD_ENV
: さまざまな値をさまざまな環境に割り当てることを許可します。このキーワードは、-automountd の D 引数と同等です。AUTOMOUNTD_ENV キーワードは、何度でも使用できます。ただし、環境割り当てごとに行を分けて使用する必要があります。

詳細は、automount(1M) および automountd(1M) のマニュアルページを参照してください。手順については、107 ページの「/etc/default/autofs ファイルを使用する方法」を参照してください。

/etc/default/nfs ファイルのキーワード

NFS version 4 では、次のキーワードを /etc/default/nfs ファイルに設定できます。これらのキーワードは、クライアントとサーバーの両方で使用される NFS プロトコルを制御します。

NFS_SERVER_VERSMIN
: サーバーが登録し提供する最小バージョンの NFS プロトコルを設定します。Solaris 10 以降のリリースでは、デフォルトは 2 です。有効な値はほかに 3 と 4 があります。94 ページの「NFS サービスの設定」を参照してください。

NFS_SERVER_VERSMAX
: サーバーが登録し提供する最大バージョンの NFS プロトコルを設定します。Solaris 10 以降のリリースでは、デフォルトは 4 です。有効な値はほかに 2 と 3 があります。94 ページの「NFS サービスの設定」を参照してください。

NFS_CLIENT_VERSMIN
: NFS クライアントが使用する最小バージョンの NFS プロトコルを設定します。Solaris 10 以降のリリースでは、デフォルトは 2 です。有効な値はほかに 3 と 4 があります。94 ページの「NFS サービスの設定」を参照してください。

NFS_CLIENT_VERSMAX
: NFS クライアントが使用する最大バージョンの NFS プロトコルを設定します。Solaris 10 以降のリリースでは、デフォルトは 4 です。有効な値はほかに 2 と 3 があります。94 ページの「NFS サービスの設定」を参照してください。

NFS_SERVER_DELEGATION
: NFS version 4 の委託機能をサーバーで有効にするかどうかを制御します。この機能が有効な場合、サーバーは NFS version 4 のクライアントに委託しようとします。デフォルトでは、サーバー委託は有効になっています。サーバー委託を無効にするには、97 ページの「サーバー上で異なるバージョンの NFS を選択する方法」を参照してください。詳細は、190 ページの「NFS version 4 における委託」を参照してください。

NFSMAPID_DOMAIN
: クライアントとサーバーに共通のドメインを設定します。ローカル DNS ドメイン名を使用するデフォルトの動作は無効になります。作業の詳細は、94 ページの「NFS サービスの設定」を参照してください。また、148 ページの「nfsmapid デーモン」も参照してください。

/etc/default/nfslogd ファイル

このファイルは、NFS サーバーログ機能を使用するときに使用されるいくつかのパラメータを定義します。次のパラメータを定義することができます。

CYCLE_FREQUENCY
ログファイルを循環させる前に経過すべき時間数を決定するパラメータです。デフォルト値は 24 時間です。このパラメータはログファイルが大きくなり過ぎないように使用します。

IDLE_TIME
`nfslogd` が、バッファーファイル内のさらなる情報を検査する前にスリープすべき秒数を決定するパラメータです。このパラメータは、構成ファイルの検査頻度も決定します。このパラメータと MIN_PROCESSING_SIZE によりバッファーファイルの処理頻度が決まります。デフォルト値は 300 秒です。この数値を増加させると、検査の回数が減ってパフォーマンスが向上します。

MAPPING_UPDATE_INTERVAL
ファイルハンドルパスマッピングテーブル内でレコードを更新する間隔を秒数で指定します。デフォルト値は 86400 秒つまり 1 日です。このパラメータを使用すると、ファイルハンドルパスマッピングテーブルを常時更新しないで最新の状態に保つことができます。

MAX_LOGS_PRESERVE
保存するログファイル数を決めます。デフォルト値は 10 です。

MIN_PROCESSING_SIZE
バッファーファイルが処理してログファイルに書き込むための最小限のバイト数を設定します。このパラメータと IDLE_TIME によりバッファーファイルの処理頻度が決まります。デフォルト値は 524,288 バイトです。この数値を大きくするとバッファーファイルの処理回数が減ってパフォーマンスが向上します。

PRUNE_TIMEOUT
ファイルハンドルパスマッピングレコードを中断して削減できるようになるまでに経過しなければならない時間数を選択するパラメータです。デフォルト値は 168 時間、つまり 7 日間です。

UMASK
`nfslogd` によって作成されるログファイルのファイルモード生成マスクを指定します。デフォルト値は 0137 です。

/etc/nfs/nfslog.conf ファイル

このファイルは `nfslogd` で使用するログのパス、ファイル名、およびタイプを定義します。各定義はタグと関連づけられています。NFS サーバーのログを開始するためには、各ファイルシステムについてタグを付ける必要があります。広域タグはデフォルト値を定義します。必要に応じて、各タグに、次のパラメータを使用することができます。

defaultdir=*path*
ログファイルのデフォルトのディレクトリパスを指定するパラメータです。特に指定しないかぎり、デフォルトのディレクトリは `/var/nfs` です。

log=*path/filename*
ログファイルのパスとファイル名を指定するパラメータです。デフォルトは /var/nfs/nfslog です。

fhtable=*path/filename*
ファイルハンドルパスデータベースのパスとファイル名を選択するパラメータです。デフォルトは /var/nfs/fhtable です。

buffer=*path/filename*
バッファーファイルのパスとファイル名を決定するパラメータです。デフォルトは /var/nfs/nfslog_workbuffer です。

logformat=*basic*|*extended*
ユーザーから読み取り可能なログファイルを作成するときに使用するフォーマットを選択します。基本フォーマットでは、ftpd デーモンに似たログファイルが作成されます。拡張フォーマットは、より詳細に表示されます。

パスが指定されていない場合は、defaultdir が定義するパスが使用されます。絶対パスを使用すると defaultdir を無効にすることができます。

ファイルを識別しやすくするために、ファイルを別々のディレクトリに入れておきます。次に、必要な変更の例を示します。

```
% cat /etc/nfs/nfslog.conf
#ident  "@(#)nfslog.conf        1.5     99/02/21 SMI"
#
  .
  .
# NFS server log configuration file.
#

global  defaultdir=/var/nfs \
        log=nfslog fhtable=fhtable buffer=nfslog_workbuffer

publicftp log=logs/nfslog fhtable=fh/fhtables buffer=buffers/workbuffer
```

この例では、log=publicftp と共有するファイルシステムはすべて、次の値を使用します。

- デフォルトのディレクトリは /var/nfs です。
- ログファイルは、/var/nfs/logs/nfslog* に保存されます。
- ファイルハンドルパスデータベーステーブルは、/var/nfs/fh/fhtables に保存されます。
- バッファーファイルは、/var/nfs/buffers/workbuffer に保存されます。

手順については、87 ページの「NFS サーバーログを有効にする方法」を参照してください。

NFS デーモン

NFS アクティビティーをサポートするために、システムが実行レベル 3 またはマルチユーザーモードで稼動し始めたときに、複数のデーモンが起動されます。mountd デーモンおよび nfsd デーモンは、サーバーであるシステム上で実行されます。サーバーデーモンの自動起動は、NFS ファイルシステムのタイプでラベル付けされたエントリが /etc/dfs/sharetab に存在するかどうかで変わります。lockd デーモンおよび statd デーモンは、NFS クライアントおよび NFS サーバー上で実行され、NFS のファイルロックをサポートします。ただし、以前のバージョンの NFS とは異なり、NFS version 4 では、デーモン lockd、statd、mountd、および nfslogd は使用されません。

この節では、次のデーモンについて説明します。

automountd デーモン

このデーモンは autofs サービスからのマウントおよびアンマウント要求を処理します。このコマンドの構文は次のとおりです。

```
automountd [ -Tnv ] [ -D name= value ]
```

このコマンドは、次のように動作します。

- -T は、トレースを有効にします。
- -n は、すべての autofs ノード上で、ブラウズを無効にします。
- -v は、コンソールへのすべての状態メッセージを記録します。
- -D *name*=*value* は、*name* によって示された自動マウントマップ変数の値を *value* に置き換えます。

自動マウントマップのデフォルト値は /etc/auto_master です。トラブルシューティングには -T オプションを使用してください。

lockd デーモン

このデーモンは NFS ファイルのレコードロックをサポートします。lockd デーモンは、ネットワークロックマネージャー (NLM) プロトコルについて、クライアントとサーバー間の RPC 接続を管理します。通常は、パラメータを指定しないで起動します。使用できるオプションは 3 つあります。lockd(1M) のマニュアルページを参照してください。これらのオプションは、コマンド行からも、/etc/default/nfs 内の適切な文字列を編集することによっても使用することができます。次は、/etc/default/nfs ファイルに設定可能なキーワードの説明です。

注 – Solaris 10 以降のリリースでは、LOCKD_GRACE_PERIOD キーワードと -g オプションの使用は推奨されていません。推奨されていないキーワードは、新しいキーワード GRACE_PERIOD に置き換えられています。両キーワードが設定されている場合、GRACE_PERIOD の値は、LOCKD_GRACE_PERIOD の値に優先します。次の GRACE_PERIOD の説明を参照してください。

LOCKD_GRACE_PERIOD 同様、/etc/default/nfs に *GRACE_PERIOD*= graceperiod を追加すると、クライアントがサーバーのリブート後に NFS version 3 のロック (NLM が提供) と version 4 のロックを再要求する秒数を設定できます。つまり、GRACE_PERIOD の値は、NFS version 3 と NFS version 4 に対するロックリカバリの猶予期間を制御します。

/etc/default/nfs に *LOCKD_RETRANSMIT_TIMEOUT*=timeout パラメータを追加すると、ロック要求をリモートサーバーに再転送するまでの秒数を選択できます。このオプションは NFS クライアントのサービスに関係します。デフォルト値は 15 秒です。この値を小さくすると、トラフィックの多いネットワーク上の NFS クライアントに対する応答時間を改善できます。ただし、ロック要求が増えることによってサーバーの負荷が増す可能性があります。デーモンに -t *timeout* オプションを指定して開始すると、コマンド行から同じパラメータを使用できます。

/etc/default/nfs に *LOCKD_SERVERS*=nthreads パラメータを追加すると、サーバーが同時に処理できる各接続ごとのスレッドの最大数を指定できます。この値は、NFS サーバーに対して予想される負荷に基づいて決定してください。デフォルト値は 20 です。TCP を使用する各 NFS クライアントは、NFS サーバーとの間で 1 つの接続を使用するため、各クライアントは、サーバー上で、最大 20 のスレッドを同時に使用することができます。

UDP を使用するすべての NFS クライアントは、NFS サーバーと 1 つの接続を共有します。その場合、UDP 接続が使用できるスレッドの数を増やさなければならないことがあるかもしれません。各 UDP クライアントには、少なくとも 2 つのスレッドを許可します。ただし、この数は、クライアントの負荷により違います。そのため、クライアントごとに 2 つのスレッドを許可しても、十分ではない場合があります。多くのスレッドを使用する場合の不利な点は、これらのスレッドを使用すると、NFS サーバー上で使用するメモリーの容量が増えるという点です。ただし、ス

レッドを使用しない場合は、*nthreads* の値を増やしても影響がありません。デーモンに nthreads オプションを指定して開始すると、コマンド行から同じパラメータを使用できます。

mountd デーモン

このデーモンは、リモートシステムからのファイルシステムマウント要求を処理して、アクセス制御を行います。mountd デーモンは、/etc/dfs/sharetab を調べて、リモートマウントに使用可能なファイルシステムと、リモートマウントを実行できるシステムを判断します。このコマンドでは、-v オプションと -r オプションを使用できます。mountd(1M) のマニュアルページを参照してください。

-v オプションは、コマンドを冗長モードで実行します。クライアントが付与されるアクセス権を NFS サーバーが決定するたびに、コンソールにメッセージが表示されます。この情報は、クライアントがファイルシステムにアクセスできない理由を調べるときに役立ちます。

-r オプションは、その後のクライアントからのマウント要求をすべて拒絶します。このオプションを指定しても、すでにファイルシステムがマウントされているクライアントには影響しません。

注 – NFS version 4 は、このデーモンを使用しません。

---

nfs4cbd デーモン

NFS version 4 クライアントの排他使用のための nfs4cbd は、NFS version 4 コールバックプログラムでの通信の終端を管理します。デーモンには、ユーザーがアクセス可能なインタフェースがありません。詳細は、nfs4cbd(1M) のマニュアルページを参照してください。

nfsd デーモン

これは、他のクライアントからのファイルシステム要求を処理するデーモンです。このコマンドに対してはいくつかのオプションを指定できます。オプションをすべて確認するには、nfsd(1M) のマニュアルページを参照してください。これらのオプションは、コマンド行からも、/etc/default/nfs 内の適切な文字列を編集することによっても使用することができます。

/etc/default/nfs に NFSD_LISTEN_BACKLOG=*length* パラメータを追加すると、接続型トランスポートを使用した NFS および TCP の接続キューの長さを設定できます。デフォルト値は 32 エントリです。nfsd に -l オプションを指定して開始すると、コマンド行から同じ項目を選択できます。

/etc/default/nfs に NFSD_MAX_CONNECTIONS=*#-conn* パラメータを追加すると、接続型トランスポートごとの最大接続数を選択できます。*#-conn* のデフォルト値はありません。コマンド行から -c *#-conn* オプションを指定してデーモンを開始すると、同じパラメータを使用できます。

/etc/default/nfs に NFSD_SERVER=*nservers* パラメータを追加すると、サーバーが一度に処理する要求の最大数を選択できます。*nservers* のデフォルト値は 16 です。コマンド行から nservers オプションを指定して nfsd を起動すると、同じように最大数を選択できます。

以前のバージョンの nfsd デーモンとは異なり、現在のバージョンの nfsd では複数のコピーを作成して要求を同時に処理することはありません。処理テーブルを ps でチェックすると、動作しているデーモンのコピーが 1 つしかないことがわかります。

nfslogd デーモン

このデーモンは実行された処理のログ機能を提供します。サーバーに対して記録される NFS 操作は、/etc/default/nfslogd に定義されている構成オプションに基づくものです。NFS サーバーのログ機能がオンになると、選択されたファイルシステム上でのすべての RPC 操作の記録がカーネルによりバッファーファイルに書き込まれます。次に nfslogd がこれらの要求を後処理します。ログインおよび IP アドレスへの UID をホスト名に割り当てやすくするために、ネームサービススイッチが使用されます。識別されたネームサービスで一致するものが見つからない場合は、その番号が記録されます。

パス名へのファイルハンドルの割り当ても nfslogd により行われます。このデーモンは、ファイルハンドルパスマッピングテーブル内でこれらの割り当てを追跡します。/etc/nfs/nfslogd で識別される各タグについて 1 つのマッピングテーブルが存在します。後処理の後に、レコードが ASCII ログファイルに書き込まれます。

注 – NFS version 4 は、このデーモンを使用しません。

---

nfsmapid デーモン

version 4 の NFS プロトコル (RFC3530) では、クライアントとサーバーの間でユーザー識別子またはグループ識別子を交換する方法が変更されました。このプロトコルでは、NFS version 4 クライアントと NFS version 4 サーバーとの間で、ファイルの所有者とグループの属性をそれぞれ user@nfsv4_domain、group@nfsv4_domain の形式で文字列として交換する必要があります。

たとえば、known_user ユーザーに完全指定のホスト名が system.example.com である NFS version 4 クライアント上に UID 123456 が割り当てられているとします。このクライアントが NFS version 4 サーバーに要求を行うには、UID 123456 を

known_user@example.com に割り当ててから、この属性を NFS version 4 サーバーに送信する必要があります。NFS version 4 サーバーは、ユーザーとグループのファイル属性を user_or_group@nfsv4_domain 形式で受信することを予期します。サーバーがクライアントから known_user@example.com を受信すると、サーバーはこの文字列をローカルの UID 123456 に割り当て、配下のファイルシステムがこれを認識します。この機能では、ネットワーク上のすべての UID と GID が一意であること、およびクライアント上の NFS version 4 のドメインがサーバー上の NFS version 4 のドメインと一致していることを前提としています。

注 – NFS version 4 のドメインが一致している場合でも、渡されたユーザー名またはグループ名をサーバーが認識しない場合、そのサーバーはそのユーザー名またはグループ名を一意の ID (整数値) に割り当てることができません。そのような場合は、サーバーは着信ユーザー名または着信グループ名を nobody ユーザーに割り当てます。そうした状況が発生することを避けるために、管理者は NFS version 4 クライアントだけに存在する特別なアカウントを作成しないようにしてください。

NFS version 4 のクライアントとサーバーは、整数から文字列への変換と文字列から整数への変換に対応しています。たとえば、NFS version 4 サーバーが GETATTR 処理を受け取ると、配下のファイルシステムから取得した UID および GID をそれぞれの文字列表現に割り当てたうえで、この情報をクライアントに送信します。またクライアントでも、UID と GID を文字列表現に割り当てる必要があります。たとえば、クライアントが chown コマンドを受け取ると、新しい UID および GID を文字列表現に割り当ててから、SETATTR 処理をサーバーに送信します。

ただし、クライアントとサーバーでは、文字列が認識されない場合の対処が異なることに注意してください。

- ユーザーがサーバー上に存在しない場合、特に同じ NFS version 4 ドメイン構成の中に存在しない場合には、サーバーは遠隔手続き呼び出し(RPC) を拒否し、クライアントにエラーメッセージを返します。このような場合は、リモートユーザーが実行できる操作が制限されます。
- ユーザーがクライアント上とサーバー上に存在している場合でも、そのドメインが一致しない場合には、サーバーが受け取った属性変更処理のうち、着信ユーザー文字列を整数値に割り当てて配下のファイルシステムが認識できるようにする必要がある処理 (SETATTR など) については、サーバーで拒否されます。NFS version 4 のクライアントとサーバーが正常に機能するには、それらの NFS version 4 ドメイン (文字列のうち、@ 記号のあとの部分) が一致しているべきです。
- NFS version 4 クライアントがサーバーから送信されたユーザー名またはグループ名を認識しない場合には、クライアントはその文字列を一意の ID (整数値) に割り当てることができません。そのような場合は、クライアントは着信ユーザー文字列または着信グループ文字列を nobody ユーザーに割り当てます。nobody に割り当てられると、さまざまなアプリケーションでさまざまな問題が発生します。NFS version 4 の機能では、ファイル属性を変更する処理は失敗します。

構成ファイルと nfsmapid

次に、nfsmapid デーモンが /etc/nsswitch.conf ファイルと /etc/resolv.conf ファイルをどのように使用するかについて説明します。

- nfsmapid は、標準の C ライブラリ関数を使用して、バックエンドネームサービスにパスワードおよびグループ情報を要求します。これらのネームサービスは、/etc/nsswitch.conf ファイルの設定によって制御されます。nsswitch.conf ファイルになんらかの変更を加えた場合には、nfsmapid 処理に影響があります。nsswitch.conf ファイルの詳細は、nsswitch.conf(4) のマニュアルページを参照してください。
- NFS version 4 クライアントがさまざまなドメインのファイルシステムを確実にマウントできるように、nfsmapid は DNS TXT リソースレコード (RR) _nfsv4idmapdomain の設定に依存しています。_nfsv4idmapdomain リソースレコードの設定の詳細については、151 ページの「nfsmapid と DNS TXT レコード」を参照してください。また、次の点にも注意してください。
  - DNS TXT RR は、必要なドメイン情報を使って、DNS サーバー上で明示的に設定するようにしてください。
  - /etc/resolv.conf ファイルは、resolver が DNS サーバーを見つけてクライアントとサーバーの NFS version 4 ドメインの TXT レコードを検索できるように、必要なパラメータを使って設定するようにしてください。

  詳細については、次を参照してください。

  - 150 ページの「優先ルール」
  - 153 ページの「NFS version 4 のデフォルトドメインを設定する」
  - resolv.conf(4) のマニュアルページ

優先ルール

nfsmapid が正しく動作するには、NFS version 4 のクライアントとサーバーが同じドメインに割り当てられている必要があります。NFS version 4 ドメインが確実に一致するように、nfsmapid は次の厳密な優先ルールに従って動作します。

1. デーモンは、NFSMAPID_DOMAIN キーワードに割り当てられた値を /etc/default/nfs ファイルで最初に確認します。値が検出された場合、その割り当てられている値は他の設定よりも優先されます。割り当てられている値は、発信属性文字列に追加され、着信属性文字列と比較されます。/etc/default/nfs ファイル内のキーワードの詳細は、142 ページの「/etc/default/nfs ファイルのキーワード」を参照してください。手順については、94 ページの「NFS サービスの設定」を参照してください。

注 – NFSMAPID_DOMAIN 設定を使用する方法はスケーラブルではないため、大規模な配備を行う場合には推奨されません。

---

2. 値が NFSMAPID_DOMAIN に割り当てられていない場合、デーモンは DNS TXT RR でドメイン名を確認します。nfsmapid は、resolver の一連のルーチンによって使用される /etc/resolv.conf ファイル内の指令に依存します。resolver は、設定されている DNS サーバーから _nfsv4idmapdomain TXT RR を検索します。DNS TXT レコードを使用する方がよりスケーラブルです。このため、/etc/default/nfs ファイルにキーワードを設定するよりも、TXT レコードを継続して使用する方がよいでしょう。
3. ドメイン名を提供する DNS TXT レコードが設定されていない場合、nfsmapid デーモンは /etc/resolv.conf ファイル内の domain または search 指令で指定された値を使用します。このとき、最後に指定された指令が優先されます。

   次の例では、domain および search の両方の指令が使用されています。nfsmapid デーモンは、search 指令のあとに最初に記載されているドメイン名である company.com を使用します。

   ```
   domain example.company.com
   search company.com foo.bar.com
   ```
4. /etc/resolv.conf ファイルが存在しない場合、nfsmapid は domainname コマンドの動作に従って NFS version 4 ドメインの名前を取得します。より詳しく説明すると、/etc/defaultdomain ファイルが存在する場合には、nfsmapid は NFS version 4 ドメインのためにそのファイルの内容を使用します。/etc/defaultdomain ファイルが存在しない場合には、nfsmapid はネットワークに設定されているネームサービスから渡されるドメイン名を使用します。詳細は、domainname(1M) のマニュアルページを参照してください。

nfsmapid と DNS TXT レコード

DNS は汎用性が高いので、NFS version 4 のドメイン名を格納して配布するための効率的な機構です。また、DNS は本質的にスケーラブルなので、DNS TXT リソースレコードを使用する方法は、大規模な配備の NFS version 4 のドメインを設定するうえで、もっとも推奨される方法です。エンタープライズレベルの DNS サーバーでは、_nfsv4idmapdomain TXT レコードを設定するようにしてください。このように設定すれば、NFS version 4 のクライアントまたはサーバーは DNS ツリーをたどることによって NFS version 4 ドメインを見つけることができます。

DNS サーバーから NFS version 4 のドメイン名を提供するように設定するときは、次の例のように入力することをお勧めします。

```
_nfsv4idmapdomain       IN      TXT             "foo.bar"
```

この例では、設定されるドメイン名は、二重引用符で囲まれている値です。ttl フィールドが指定されていないことと、ドメインが owner フィールドの値である _nfsv4idmapdomain に追加されていないことに注意してください。この設定により、TXT レコードで、Start-Of-Authority (SOA) レコードのゾーンの ${ORIGIN} エントリを使用できるようになります。たとえば、さまざまなレベルのドメイン名前空間で、レコードは次のように読み取ることができます。

```
_nfsv4idmapdomain.subnet.yourcorp.com.    IN    TXT    "foo.bar"
_nfsv4idmapdomain.yourcorp.com.           IN    TXT    "foo.bar"
```

この設定では、DNS クライアントが DNS ツリー階層を検索するときに、`resolv.conf` ファイルを使用して柔軟に検索することができます。`resolv.conf(4)` のマニュアルページを参照してください。この機能により、TXT レコードの検索での確率がより高くなります。柔軟性の向上により、低いレベルの DNS サブドメインが、自身の DNS TXT リソースレコード (RR) を定義できるようになりました。この機能により、低いレベルの DNS サブドメインを、高いレベルの DNS ドメインの定義した TXT レコードに優先させることができます。

注 – TXT レコードで指定したドメインには、任意の文字列を使用できます。この文字列は、NFS version 4 を使用するクライアントとサーバーの DNS ドメインと同じである必要はありません。 NFS version 4 データをほかの DNS ドメインと共有しないようにするオプションがあります。

NFS version 4 のドメインを確認する

ネットワークの NFS version 4 ドメインの値を割り当てる前に、ネットワークに NFS version 4 ドメインがすでに設定されているかどうかを確認します。次の例は、ネットワークの NFS version 4 ドメインを確認する方法を示します。

- NFS version 4 ドメインを DNS TXT RR で確認するには、`nslookup` コマンドまたは `dig` コマンドを使用します。

  `nslookup` コマンドの出力例を次に示します。

  ```
  # nslookup -q=txt _nfsv4idmapdomain
  Server:         10.255.255.255
  Address:        10.255.255.255#53

  _nfsv4idmapdomain.example.company.com text = "company.com"
  ```

  `dig` コマンドの出力例を次に示します。

  ```
  # dig +domain=example.company.com -t TXT _nfsv4idmapdomain
  ...
  ;; QUESTION SECTION:
  ;_nfsv4idmapdomain.example.company.com. IN    TXT

  ;; ANSWER SECTION:
  _nfsv4idmapdomain.example.company.com. 21600 IN TXT   "company.com"

  ;; AUTHORITY SECTION:
  ...
  ```

  DNS TXT RR の設定方法については、151 ページの「`nfsmapid` と DNS TXT レコード」を参照してください。

- ネットワークに NFS version 4 の DNS TXT RR が設定されていない場合は、次のコマンドを使用して、NFS version 4 ドメインを DNS ドメイン名で確認します。

```
# egrep domain /etc/resolv.conf
domain example.company.com
```

- `/etc/resolv.conf` ファイルがクライアントの DNS ドメイン名を提供するように設定されていない場合は、次のコマンドを使用して、ネットワークの NFS version 4 ドメイン構成でドメインを確認します。

```
# cat /var/run/nfs4_domain
company.com
```

- NIS などの別のネームサービスを使用している場合は、次のコマンドを使用して、ネットワークに構成されているネームサービスでドメインを確認します。

```
# domainname
it.example.company.com
```

詳細は、次のマニュアルページを参照してください。

- `nslookup(1M)`
- `dig(1M)`
- `resolv.conf(4)`
- `domainname(1M)`

NFS version 4 のデフォルトドメインを設定する

この節では、ネットワークがどのようにして目的のデフォルトドメインを取得するかについて説明します。

- Solaris Express 5/06 リリースの場合は、153 ページの「Solaris Express 5/06 リリースで NFS version 4 のデフォルトドメインを設定する」を参照してください。
- 初期 Solaris 10 リリースの場合は、155 ページの「Solaris 10 リリースで NFS version 4 のデフォルトドメインを設定する」を参照してください。

Solaris Express 5/06 リリースで NFS version 4 のデフォルトドメインを設定する

初期 Solaris 10 リリースでは、OS インストール後の初回システムリブート中に、ドメインの定義が行われていました。Solaris Express 5/06 リリースでは、OS のインストール中に NFS version 4 ドメインの定義が行われます。この機能を提供するために、次の機能が追加されました。

- `sysidtool` コマンドに `sysidnfs4` プログラムが含まれています。このプログラムは、ネットワークの NFS version 4 ドメインが設定済みかどうかを判定するために、インストール処理中に実行されます。`sysidtool(1M)` および `sysidnfs4(1M)` のマニュアルページを参照してください。
- `sysidcfg` ファイルに新しいキーワード `nfs4_domain` が追加されています。このキーワードを使えば、NFS version 4 のドメインを定義できます。`sysidcfg` ファイルにはほかのキーワードも定義できます。`sysidcfg(4)` のマニュアルページを参照してください。

次に、この機能の動作手順を説明します。

1. sysidnfs4 プログラムは /etc/.sysIDtool.state ファイルをチェックし、NFS version 4 ドメインが特定されているかどうかを判定します。
   - .sysIDtool.state ファイルから、ネットワークの NFS version 4 ドメインが設定されていることが判明すると、sysidnfs4 プログラムはそれ以上のチェックを行いません。次の .sysIDtool.state ファイルの例を参照してください。

     ```
     1       # System previously configured?
     1       # Bootparams succeeded?
     1       # System is on a network?
     1       # Extended network information gathered?
     1       # Autobinder succeeded?
     1       # Network has subnets?
     1       # root password prompted for?
     1       # locale and term prompted for?
     1       # security policy in place
     1       # NFSv4 domain configured
     xterms
     ```

     # NFSv4 domain configured の前に 1 が表示されていれば、NFS version 4 ドメインが設定されています。
   - .sysIDtool.state ファイルから、ネットワークの NFS version 4 ドメインが設定されていないことが判明した場合、sysidnfs4 プログラムはさらなるチェックを行う必要があります。次の .sysIDtool.state ファイルの例を参照してください。

     ```
     1       # System previously configured?
     1       # Bootparams succeeded?
     1       # System is on a network?
     1       # Extended network information gathered?
     1       # Autobinder succeeded?
     1       # Network has subnets?
     1       # root password prompted for?
     1       # locale and term prompted for?
     1       # security policy in place
     0       # NFSv4 domain configured
     xterms
     ```

     # NFSv4 domain configured の前に 0 が表示されていれば、NFS version 4 ドメインは設定されていません。
2. NFS version 4 ドメインが特定されていない場合、sysidnfs4 プログラムは sysidcfg ファイル内の nfs4_domain キーワードをチェックします。
   - nfs4_domain の値が存在する場合は、その値が /etc/default/nfs ファイル内の NFSMAPID_DOMAIN キーワードに設定されます。NFSMAPID_DOMAIN に値が設定されると、その値が何であれ、それが nfsmapid デーモンの動的ドメイン選択機能よりも優先されます。nfsmapid の動的ドメイン選択機能の詳細については、150 ページの「優先ルール」を参照してください。
   - nfs4_domain の値が存在しない場合、sysidnfs4 プログラムは、オペレーティングシステムの設定済みネームサービスから nfsmapid によって派生されるドメインを特定します。この派生値はデフォルトドメインとして対話プロンプトに

表示されますが、ユーザーは、そのデフォルト値を受け入れるか、異なる NFS version 4 ドメインを割り当てるかを選択できます。

この機能により、次のものが廃止になります。

- 初期 Solaris 10 のメディアディストリビューションに付属していたサンプルの JumpStart スクリプト `set_nfs4_domain` が必要でなくなり、非推奨となった。
- `sysidnfs4` プログラムの以前の実装によって作成されていた `/etc/.NFS4inst_state.domain` ファイルが、必要でなくなった。

注 – DNS 特有のユビキタスでスケーラブルな性質のため、大規模な NFS version 4 配備のドメイン設定には DNS TXT レコードを引き続き使用することを強く推奨します。151 ページの「`nfsmapid` と DNS TXT レコード」を参照してください。

Solaris のインストールプロセスに関する具体的な情報については、次を参照してください。

- 『Oracle Solaris 10 9/10 インストールガイド (基本編)』
- 『Oracle Solaris 10 9/10 インストールガイド (ネットワークインストール)』

Solaris 10 リリースで NFS version 4 のデフォルトドメインを設定する

初期 Solaris 10 リリースの NFS version 4 では、ネットワーク内に複数の DNS ドメインが存在しているにもかかわらず、単一の UID および GID 名前空間しかない場合、すべてのクライアントが NFSMAPID_DOMAIN に対して単一の値を使用する必要があります。DNS を使用するサイトでは、`nfsmapid` が、`_nfsv4idmapdomain` に割り当てられた値からドメイン名を取得して、この問題を解決します。詳細は、151 ページの「`nfsmapid` と DNS TXT レコード」を参照してください。ネットワークが DNS を使用する構成になっていない場合は、Solaris オペレーティングシステムの最初のブート時に、`sysidconfig(1M)` ユーティリティーによって NFS version 4 のドメイン名に関する次のプロンプトが表示されます。

```
This system is configured with NFS version 4, which uses a
domain name that is automatically derived from the system's
name services. The derived domain name is sufficient for most
configurations. In a few cases, mounts that cross different
domains might cause files to be owned by nobody due to the
lack of a common domain name.

Do you need to override the system's default NFS verion 4 domain
name (yes/no)? [no]
```

デフォルトの応答は `[no]` です。`[no]` を選択すると、次のプロンプトが表示されます。

```
For more information about how the NFS version 4 default domain name is
derived and its impact, refer to the man pages for nfsmapid(1M) and
nfs(4), and the System Administration Guide: Network Services.
```

[yes] を選択すると、次のプロンプトが表示されます。

```
Enter the domain to be used as the NFS version 4 domain name.
NFS version 4 domain name []:
```

注 – NFSMAPID_DOMAIN の値が /etc/default/nfs に存在する場合は、指定した [domain_name] が優先されます。

---

nfsmapid の追加情報

nfsmapid の詳細は、次を参照してください。

- nfsmapid(1M) のマニュアルページ
- nfs(4) のマニュアルページ
- http://www.ietf.org/rfc/rfc1464.txt
- 192 ページの「NFS version 4 での ACL と nfsmapid」

statd デーモン

lockd とともに動作し、ロックマネージャーにクラッシュ/回復機能を提供します。statd デーモンは、NFS サーバーにロックを保持するクライアントを追跡します。サーバーがクラッシュした場合は、サーバーのリブート中に、サーバー側 statd がクライアント側 statd にコンタクトします。次にクライアント側 statd は、サーバー上のすべてのロックを再要求します。クライアント側 statd は、サーバー上のクライアントのロックがクリアされるように、サーバー側 statd にクライアントがいつクラッシュしたかを通知します。このデーモンにオプションはありません。詳細は、statd(1M) のマニュアルページを参照してください。

Solaris 7 で、statd がクライアントを追跡する方法が改善されました。Solaris 7 より前のリリースの statd では、クライアントごとにそのクライアントの修飾されていないホスト名を使用して、/var/statmon/sm にファイルが作成されました。そのため、同じホスト名の 2 つのクライアントが異なるドメインに存在する場合や、クライアントが NFS サーバーと異なるドメインに存在する場合に、このファイルのネーミングが原因となり問題が発生していました。修飾されていないホスト名にはドメインや IP アドレスの情報がないため、このようなクライアントを区別する方法がありませんでした。これに対処するため、Solaris 7 の statd では、修飾されていないホスト名に対してクライアントの IP アドレスを使用して /var/statmon/sm にシンボリックリンクを作成します。このリンクは、次のようになります。

```
# ls -l /var/statmon/sm
lrwxrwxrwx   1 daemon          11 Apr 29 16:32 ipv4.192.168.255.255 -> myhost
lrwxrwxrwx   1 daemon          11 Apr 29 16:32 ipv6.fec0::56:a00:20ff:feb9:2734 -> v6host
--w-------   1 daemon          11 Apr 29 16:32 myhost
--w-------   1 daemon          11 Apr 29 16:32 v6host
```

この例では、クライアントのホスト名は myhost で、クライアントの IP アドレスは 192.168.255.255 です。ほかのホストが myhost という名前を持ち、ファイルシステムをマウントしていると、myhost というホスト名に対するシンボリックリンクは 2 つ作成されます。

注 – NFS version 4 は、このデーモンを使用しません。

NFS コマンド

次のコマンドは、root として実行しないと、十分な効果が得られません。ただし、情報の要求は、すべてのユーザーが行うことができます。

- 157 ページの「automount コマンド」
- 158 ページの「clear_locks コマンド」
- 158 ページの「fsstat コマンド」
- 159 ページの「mount コマンド」
- 166 ページの「mountall コマンド」
- 175 ページの「setmnt コマンド」
- 167 ページの「share コマンド」
- 173 ページの「shareall コマンド」
- 174 ページの「showmount コマンド」
- 165 ページの「umount コマンド」
- 167 ページの「umountall コマンド」
- 173 ページの「unshare コマンド」
- 174 ページの「unshareall コマンド」

automount コマンド

このコマンドは autofs マウントポイントをインストールし、オートマスターファイル内の情報を各マウントポイントに関連付けます。このコマンドの構文は次のとおりです。

```
automount [ -t duration ] [ -v ]
```

-t *duration* はファイルシステムがマウントされた状態でいる時間 (秒) を設定し、-v は冗長モードを選択します。冗長モードでこのコマンドを実行するとトラブルシューティングが容易になります。

継続時間の値は、特に設定しないと 5 分に設定されます。通常はこの値が適切です。しかし、自動マウントされたファイルシステムの多いシステムでは、この値を増やす必要がある場合もあります。特に、サーバーを多くのユーザーが使用中の場合は、自動マウントされたファイルシステムを 5 分ごとにチェックするのは能率的

でない場合があります。autofs ファイルシステムは 1800 秒 (30 分) ごとにチェックする方が適しています。5 分おきにファイルシステムマウントを解除しないと、/etc/mnttab が大きくなることがあります。df が /etc/mnttab にある各エントリをチェックしたときの出力を減らすには、-F オプション (df(1M) のマニュアルページを参照) または egrep を使用して、df の出力にフィルタをかけます。

この継続時間を調節すると、オートマウンタマップへの変更が反映される速さを変更できるということも考慮すべきです。変更はファイルシステムがアンマウントされるまでは見ることができません。オートマウンタマップの変更方法については、109 ページの「マップの修正」を参照してください。

clear_locks コマンド

このコマンドを使用すると、ある NFS クライアントのファイル、レコード、または共有のロックをすべて削除できます。このコマンドを実行するには、スーパーユーザーでなければなりません。NFS サーバーから、特定のクライアントに対するロックを解除できます。また、NFS クライアントから、特定のサーバーにおけるそのクライアントに対するロックを解除できます。次の例では、現在のシステム上の tulip という NFS クライアントに対するロックが解除されます。

```
# clear_locks tulip
```

-s オプションを指定すると、どの NFS ホストからロックを解除するかを指定できます。このオプションは、ロックを作成した NFS クライアントで実行する必要があります。次の場合、クライアントによるロックが bee という名前の NFS サーバーから解除されます。

```
# clear_locks -s bee
```

注意 – このコマンドは、クライアントがクラッシュしてロックを解除できないとき以外には使用しないでください。データが破壊されるのを避けるため、使用中のクライアントに関するロックは解除しないでください。

fsstat コマンド

Solaris 10 11/06 以降のリリースでは、fsstat ユーティリティーを使用して、ファイルシステムの種類およびマウントポイントごとに、ファイルシステムオペレーションを監視できます。出力のカスタマイズを可能にするオプションが多数用意されています。次に例を示します。

次の例では、NFS version 3、version 4、およびルートマウントポイントに対する出力を表示しています。

```
% fsstat nfs3 nfs4 /
  new     name   name    attr    attr   lookup   rddir   read   read   write   write
 file    remov   chng     get     set      ops     ops    ops  bytes     ops   bytes
3.81K       90  3.65K   5.89M   11.9K    35.5M   26.6K   109K   118M   35.0K   8.16G  nfs3
  759      503    457   93.6K   1.44K     454K   8.82K  65.4K   827M     292    223K  nfs4
25.2K    18.1K  1.12K   54.7M    1017     259M   1.76M  22.4M  20.1G   1.43M   3.77G  /
```

次の例では、-i オプションを使って NFS version 3、version 4、およびルートマウントポイントの入出力操作に関する統計を提供しています。

```
% fsstat -i nfs3 nfs4 /
 read    read    write   write   rddir   rddir   rwlock   rwulock
  ops   bytes      ops   bytes     ops   bytes      ops       ops
 109K    118M    35.0K   8.16G   26.6K   4.45M     170K      170K  nfs3
65.4K    827M      292    223K   8.82K   2.62M    74.1K     74.1K  nfs4
22.4M   20.1G    1.43M   3.77G   1.76M   3.29G    25.5M     25.5M  /
```

次の例では、-n オプションを使って NFS version 3、version 4、およびルートマウントポイントの命名操作に関する統計を提供しています。

```
% fsstat -n nfs3 nfs4 /
lookup   creat   remov  link   renam  mkdir  rmdir   rddir  symlnk  rdlnk
 35.5M   3.79K      90     2   3.64K      5      0   26.6K      11   136K  nfs3
  454K     403     503     0     101      0      0   8.82K     356  1.20K  nfs4
  259M   25.2K   18.1K   114    1017     10      2   1.76M      12  8.23M  /
```

詳細は、fsstat(1M) のマニュアルページを参照してください。

mount コマンド

このコマンドを使用すると、指定したファイルシステムをローカルまたはリモートで、指定したマウントポイントにマウントできます。詳細は、mount(1M) のマニュアルページを参照してください。引数を指定しないで mount を使用すると、現在コンピュータにマウントされているファイルシステムのリストが表示されます。

Solaris の標準インストールには、さまざまな種類のファイルシステムが含まれています。ファイルシステムの種類ごとにマニュアルページがあり、その種類に対して mount を実行するときに使用可能なオプションのリストが示されています。NFS ファイルシステムについては、mount_nfs(1M) のマニュアルページを参照してください。UFS ファイルシステムについては、mount_ufs(1M) のマニュアルページを参照してください。

Solaris 7 で、server:/pathname という標準の構文の代わりに NFS URL を使用して NFS サーバー上のマウントするパス名を指定することが可能になりました。詳細は、94 ページの「NFS URL を使用して NFS ファイルシステムをマウントする方法」を参照してください。

注意 – Solaris 2.6 以降の mount コマンドでは、無効なオプションがあっても警告されません。解釈できないオプションがあると無視されるだけです。予想外の結果が生じるのを避けるために、使用するオプションはすべて確認してください。

NFS ファイルシステム用の mount オプション

NFS ファイルシステムのマウント時に -o フラグのあとに指定できるオプションの一部を、次に示します。オプションの完全な一覧については、mount_nfs(1M) のマニュアルページを参照してください。

bg|fg
: これらのオプションは、マウントが失敗したときの再試行の方法を選択するオプションです。bg オプションの場合はバックグラウンドで、fg オプションの場合はフォアグラウンドでマウントが試みられます。デフォルトは fg です。常に使用可能にしておく必要のあるファイルシステムに対しては fg が適しています。fg オプションを指定すると、マウントが完了するまで、次の処理は行われません。bg は、マウント要求が完了しなくてもクライアントはほかの処理を実行できるため、クリティカルでないファイルシステムの処理に適しています。

forcedirectio
: このオプションは、大規模の連続したデータ転送のパフォーマンスを向上させます。データは直接ユーザーバッファーにコピーされます。クライアント上のカーネル内ではキャッシュへの書き込みは行われません。この機能はデフォルトではオフです。

  これまで、書き込み要求はすべて、NFS クライアントと NFS サーバーの両方で直列化されていました。今回の NFS クライアントの変更により、単一ファイルに対する並行書き込み、並行読み取り/書き込みを、アプリケーションから実行できるようになりました。この機能をクライアント上で有効にするには、mount コマンドオプション forcedirectio を使用します。このオプションを使用した場合、マウントされたファイルシステム内のすべてのファイルに対して、この機能が有効になります。この機能をクライアントの単一のファイルに対してのみ有効にするには、directio() インタフェースを使用します。この機能を有効にしないかぎり、ファイルへの書き込みは直列化されます。また、並行書き込みや並行読み取り/書き込みが実行されると、そのファイルに関しては、POSIX のセマンティクスはサポートされなくなります。

  このオプションの使用例については、163 ページの「mount コマンドの使用」を参照してください。

largefiles
: このオプションを使用すると、Solaris 2.6 が稼働しているサーバー上に置かれた 2G バイトを超えるサイズのファイルにアクセスできるようになります。大規模ファイルにアクセスできるかどうかは、サーバーでしか制御できません。したがって、このオプションは NFS version 3 のマウントでは無視されます。デフォル

トでは、Solaris 2.6 以降のリリースの UFS ファイルシステムはすべて largefiles オプション付きでマウントされます。NFS version 2 プロトコルを使用したマウントで largefiles オプションを指定すると、エラーが発生してマウントできません。

nolargefiles
: この UFS マウント用のオプションを指定すると、ファイルシステム上に大規模ファイルが存在できないことが保証されます。mount_ufs(1M) のマニュアルページを参照してください。大規模ファイルの存在は NFS サーバー上でのみ制御できるため、NFS マウントを使用している場合は、nolargefiles オプションを指定できません。このオプションを指定してファイルシステムを NFS マウントしようとすると、エラーが発生して拒否されます。

nosuid|suid
: Solaris 10 以降のリリースでは、nosuid オプションは、nodevices オプションを nosetuid オプションと同時に指定することと同等です。nodevices オプションが指定されている場合、マウントされたファイルシステム上のデバイス特殊ファイルを開くことができません。nosetuid オプションが指定されている場合、ファイルシステム上に置かれたバイナリファイルの setuid ビットと setgid ビットは無視されます。プロセスは、バイナリファイルを実行するユーザーの特権で実行します。

  suid オプションは、devices オプションを setuid オプションと同時に指定することと同等です。devices オプションが指定されている場合、マウントされたファイルシステムのデバイス特殊ファイルを開くことができます。setuid オプションが指定されている場合、ファイルシステムに置かれたバイナリファイルの setuid ビットと setgid ビットは、カーネルが引き受けます。

  いずれのオプションも指定されていない場合、デフォルトのオプションは suid になります。これにより、devices オプションを setuid オプションと同時に指定するデフォルトの動作になります。

  次の表は、nosuid または suid を devices または nodevices、および setuid または nosetuid と組み合わせることによる結果を示しています。オプションの各組み合わせでは、もっとも制限の高いオプションが動作を決定します。

オプションの組み合わせによる動作	オプション	オプション	オプション
nosetuid と nodevices の同時指定と同等	nosuid	nosetuid	nodevices
nosetuid と nodevices の同時指定と同等	nosuid	nosetuid	devices
nosetuid と nodevices の同時指定と同等	nosuid	setuid	nodevices

オプションの組み合わせによる動作	オプション	オプション	オプション
nosetuid と nodevices の同時指定と同等	nosuid	setuid	devices
nosetuid と nodevices の同時指定と同等	suid	nosetuid	nodevices
nosetuid と devices の同時指定と同等	suid	nosetuid	devices
setuid と nodevices の同時指定と同等	suid	setuid	nodevices
setuid と devices の同時指定と同等	suid	setuid	devices

nosuid オプションを指定すると、信頼できないサーバーにアクセスする可能性のある NFS クライアントのセキュリティーを向上できます。このオプションを使用してリモートファイルシステムのマウントを行うと、信頼できないデバイスのインポートまたは信頼できない setuid バイナリファイルのインポートによって特権が拡大する可能性を減らすことができます。これらのオプションはすべて、Solaris ファイルシステム全体で使用可能です。

public
このオプションを指定すると、NFS サーバーにアクセスするときに必ず公開ファイルハンドルを使用するようになります。NFS サーバーが公開ファイルハンドルをサポートしていれば、MOUNT プロトコルが使用されないため、マウント操作は短時間で行われます。また、MOUNT プロトコルを使用しないため、ファイアウォールを越えたマウントが可能です。

rw|ro
- rw オプションと - ro オプションは、ファイルシステムが読み書き可能と読み取り専用のどちらでマウントされるかを示します。デフォルトは読み書き可能で、これはリモートホームディレクトリやメールスプールディレクトリなどの、ユーザーによる変更が必要なファイルシステムに適しています。読み取り専用オプションは、ユーザーが変更してはいけないディレクトリに適しています。具体的には、マニュアルページの共有コピーなどです。

sec=*mode*
このオプションは、マウント時に使用される認証メカニズムを指定します。*mode* の値は、次のいずれかです。

- Kerberos version 5 認証サービス用の krb5 を使用する。
- 整合性を指定する Kerberos version 5 用の krb5i を使用する。
- 機密性を指定する Kerberos version 5 用の krb5p を使用する。
- 認証なしの none を使用する。
- Diffie-Hellman (DH) 認証用の dh を使用する。

- UNIX 標準認証用の sys を使用する。

モードは、/etc/nfssec.conf ファイルにも定義されます。

soft|hard
: soft オプションを指定してマウントされた NFS ファイルシステムは、サーバーが応答しなくなるとエラーを返します。hard オプションが指定されていると、サーバーが応答するまで続けて再試行が行われます。デフォルトは hard です。ほとんどのファイルシステムには hard を使用します。ソフトマウントされたファイルシステムからの値を検査しないアプリケーションが多いので、アプリケーションでエラーが発生してファイルが破壊される恐れがあるためです。アプリケーションが戻り値を確認する場合は、soft が使用されているとルーティングの問題などによってアプリケーションが正しく判断できず、ファイルが破壊されることがあります。原則として、soft は使用しないでください。hard オプションを指定した場合にファイルシステムが使用できなくなると、そのファイルシステムを使用するアプリケーションはファイルシステムが復旧するまでハングアップする可能性があります。

mount コマンドの使用

次の例を参照してください。

- NFS version 2 または version 3 では、次のコマンドはどちらもサーバー bee から NFS ファイルシステムを読み取り専用としてマウントします。

```
# mount -F nfs -r bee:/export/share/man /usr/man
```

```
# mount -F nfs -o ro bee:/export/share/man /usr/man
```

  NFS version 4 では、次のコマンド行で同じマウントを行えます。

```
# mount -F nfs -o vers=4 -r bee:/export/share/man /usr/man
```

- NFS version 2 または version 3 では、次のコマンドは -O オプションを使用しているため、すでに /usr/man がマウントされている場合でも、強制的にマニュアルページをサーバー bee からローカルシステムにマウントします。次を参照してください。

```
# mount -F nfs -O bee:/export/share/man /usr/man
```

  NFS version 4 では、次のコマンド行で同じマウントを行えます。

```
# mount -F nfs -o vers=4 -O bee:/export/share/man /usr/man
```

- NFS version 2 または version 3 では、次のコマンドはクライアント側フェイルオーバー機能を使用します。

```
# mount -F nfs -r bee,wasp:/export/share/man /usr/man
```

  NFS version 4 では、次のコマンド行はクライアント側フェイルオーバー機能を使用します。

```
# mount -F nfs -o vers=4 -r bee,wasp:/export/share/man /usr/man
```

注 – コマンド行から使用する場合、リスト内のサーバーがサポートしている NFS プロトコルは同じバージョンでなければなりません。コマンド行から mount を実行するときは、version 2 と version 3 のサーバーを同時に使用しないでください。autofs を実行するときは、両サーバーを同時に使用することができます。autofs により、version 2 または version 3 のサーバーの最適な組み合わせが自動的に選択されます。

- 次に、NFS version 2 または version 3 において、mount コマンドに NFS URL を使用する例を示します。

```
# mount -F nfs nfs://bee//export/share/man /usr/man
```

次に、NFS version 4 において、mount コマンドに NFS URL を使用する例を示します。

```
# mount -F nfs -o vers=4 nfs://bee//export/share/man /usr/man
```

- forcedirectio マウントオプションを使用すると、ファイルに対してクライアントが、並行書き込みと並行読み取り/書き込みを行えるようになります。次に例を示します。

```
# mount -F nfs -o forcedirectio bee:/home/somebody /mnt
```

この例では、サーバー bee からの NFS ファイルシステムがマウントされ、ディレクトリ /mnt にあるファイルごとに並行読み取り/書き込みが有効になります。並行読み取り/書き込みのサポートを有効にすると、次のことが発生します。

  - クライアントは、ファイルへの並列した書き込みをアプリケーションに許可します。
  - クライアントでのキャッシュが無効になります。その結果、読み取りと書き込みのデータはサーバー上に保持されます。つまり、クライアントは読み取られたデータまたは書き込まれたデータをキャッシュに書き込まないため、アプリケーションがキャッシュに書き込んでいないデータはサーバーから読み取られます。クライアントのオペレーティングシステムは、このデータのコピーを持ちません。通常、NFS クライアントは、アプリケーションが使用するカーネルにデータをキャッシュします。

    クライアント側でキャッシュが無効になっているため、先読みと後書きプロセスが無効になります。先読みプロセスは、アプリケーションが次に要求する可能性のあるデータをカーネルが予測したときに、発生します。次に、カーネルはあらかじめデータを収集するプロセスを開始します。カーネルの目標は、アプリケーションがデータを要求する前にそのデータを準備しておくことです。

    クライアントは、書き込みのスループットを向上する後書きプロセスを使用します。アプリケーションがデータをファイルに書き込むたびに、入出力操作をただちに開始する代わりに、データはメモリー内にキャッシュされます。のちに、データはディスクに書き込まれます。

後書きプロセスにより、データがより大きな領域に書き込まれたり、アプリケーションから非同期で書き込まれたりする可能性があります。通常、より大きな領域を使用するとスループットが向上します。非同期の書き込みにより、アプリケーション処理と入出力処理間でオーバーラップができるようになります。また、ストレージサブシステムが、より優れた入出力処理を行うことで入出力を最適化できるようになります。同期の書き込みは、最適化されていないストレージサブシステムでの入出力を強制的に処理します。

- アプリケーションでキャッシュされていないデータのセマンティクスを処理する準備ができていない場合、著しくパフォーマンスが低下する可能性があります。マルチスレッド化されたアプリケーションは、この問題を回避します。

注 – 並行書き込みのサポートが有効にされていない場合、すべての書き込み要求は直列化されます。要求が直列化されると、次のことが発生します。ある書き込み要求が進行中のとき、2 番目の書き込み要求は、最初の処理が完了するのを待ってから処理を続行する必要があります。

- `mount` コマンドに引数を指定しないと、クライアントにマウントされたファイルシステムが表示されます。次を参照してください。

```
% mount
/ on /dev/dsk/c0t3d0s0 read/write/setuid on Wed Apr 7 13:20:47 2004
/usr on /dev/dsk/c0t3d0s6 read/write/setuid on Wed Apr 7 13:20:47 20041995
/proc on /proc read/write/setuid on Wed Apr 7 13:20:47 2004
/dev/fd on fd read/write/setuid on Wed Apr 7 13:20:47 2004
/tmp on swap read/write on Wed Apr 7 13:20:51 2004
/opt on /dev/dsk/c0t3d0s5 setuid/read/write on Wed Apr 7 13:20:51 20041995
/home/kathys on bee:/export/home/bee7/kathys
  intr/noquota/nosuid/remote on Wed Apr 24 13:22:13 2004
```

umount コマンド

このコマンドにより、現在マウントされているリモートファイルシステムが解除されます。`umount` コマンドは、テストのために `-V` オプションをサポートしています。また、`-a` オプションを使用することによって 1 度に複数のファイルシステムをアンマウントできます。`-a` オプションに *mount-points* を指定すると、そのファイルシステムがアンマウントされます。マウントポイントを指定しないと、`/etc/mnttab` のリストにあるファイルシステムのうち必須でないものすべてのアンマウントが試みられます。必須のファイルシステムとは、`/`、`/usr`、`/var`、`/proc`、`/dev/fd`、`/tmp` などです。 ファイルシステムがすでにマウントされていて、`/etc/mnttab` に項目が指定されている場合、ファイルシステムのタイプのフラグを指定する必要はありません。

`-f` オプションを指定すると、使用中のファイルシステムが強制的にアンマウントされます。このオプションを使用して、マウントできないファイルシステムのマウントを試みた最中にハングアップしたクライアントを復帰させることが可能です。

注意 – ファイルシステムを強制的にアンマウントすると、ファイルへの書き込み中だった場合には、データを損失することがあります。

次に例を示します。

例 6–1 ファイルシステムをアンマウントする

次の例は、/usr/man にマウントしたファイルシステムをアンマウントします。

```
# umount /usr/man
```

例 6–2 umount でオプションを使用する

次の例では、umount -a -V の実行結果が表示されます。

```
# umount -a -V
umount /home/kathys
umount /opt
umount /home
umount /net
```

このコマンドでは、ファイルシステムのアンマウント自体は実行されないことに注意してください。

mountall コマンド

このコマンドを使用すると、ファイルシステムテーブルに一覧表示されたすべてのファイルシステム、または特定グループのファイルシステムをマウントできます。このコマンドを実行すると、次の操作を実行することができます。

- -F *FSType* オプションを使用して、ファイルシステムのタイプを選択する
- -r オプションを使用して、ファイルシステムテーブル中に一覧表示されたリモートファイルシステムをすべて選択する
- -l オプションを使用して、ローカルファイルシステムをすべて選択する

NFS ファイルシステムタイプと指定されているファイルシステムはすべてリモートファイルシステムなので、これらのオプションは余分な指定になることがあります。詳細は、mountall(1M) のマニュアルページを参照してください。

次の 2 つのユーザー入力例では、同じ結果が得られます。

```
# mountall -F nfs

# mountall -F nfs -r
```

umountall コマンド

このコマンドを使用すると、ファイルシステムのグループをアンマウントできます。-k オプションは、*mount-point* に関連付けられているプロセスを終了させるために fuser -k *mount-point* コマンドを実行します。-s オプションは、アンマウントを並行処理しないことを示します。-l は、ローカルファイルシステムだけを使用することを、-r はリモートファイルシステムだけを使用することを示します。-h *host* オプションは、指定されたホストのファイルシステムをすべてアンマウントすることを指定します。-h オプションは、-l または -r と同時に指定できません。

次の例では、リモートホストからマウントしたすべてのファイルシステムがアンマウントされます。

```
# umountall -r
```

次の例では、bee サーバーからマウントしたすべてのファイルシステムがアンマウントされます。

```
# umountall -h bee
```

share コマンド

このコマンドを使用すると、NFS サーバーのローカルファイルシステムをマウントできるようになります。また、システム上のファイルシステムのうち、現在共有しているもののリストを表示します。NFS サーバーが動作していないと、share コマンドは使用できません。NFS サーバーソフトウェアは、/etc/dfs/dfstab にエントリがある場合、起動の途中で自動的に起動されます。NFS サーバーソフトウェアが動作していない場合、このコマンドはエラーを報告しません。そのため、ソフトウェアが動作していることを確認する必要があります。

すべてのディレクトリツリーは共有できるオブジェクトです。ただし、各ファイルシステムの階層構造は、そのファイルシステムが位置するディスクスライスやパーティションで制限されます。たとえば、ルート (/) ファイルシステムを共有しても、/usr が同じディスクパーティションかスライスに存在しなければ、/usr を共有することはできません。通常、ルートはスライス 0 に、/usr はスライス 6 にインストールされます。また、/usr を共有しても、/usr のサブディレクトリにマウントされているローカルディスクパーティションは共有できません。

すでに共有している大規模なファイルシステムの一部であるファイルシステムを共有することはできません。たとえば、/usr および /usr/local が同じディスクスライスにある場合は、/usr または /usr/local を共有できます。ただし、異なる共有オプションを指定してこれら両方のディレクトリを共有するには、/usr/local を別のディスクスライスに移動する必要があります。

読み取り専用で共有しているファイルシステムに、読み取りと書き込みが可能な状態で共有しているファイルシステムのファイルハンドルでアクセスすることができます。ただし、両方のファイルシステムが同じディスクスライスにある必要があります。より安全にこれらのファイルシステムを使用するには、読み取りと書き込みが設定されているファイルシステムを、読み取り専用で共有しているファイルシステムとは別のパーティションまたはディスクスライスに配置します。

注 – ファイルシステムの共有を解除してから再度共有するとき、NFS version 4 がどのように動作するかについては、184 ページの「NFS version 4 におけるファイルシステムの共有解除と再共有」を参照してください。

非ファイルシステム用 share オプション

`-o` フラグに指定できるオプションの一部を次に示します。

rw|ro
: *pathname* に指定したファイルシステムを、すべてのクライアントに対して読み取りと書き込みの両方が可能な状態で共有するか、読み取り専用で共有するかを指定します。

rw=*accesslist*
: ファイルシステムは、リストに示されているクライアントに対してだけ、読み取りと書き込みの両方が可能な状態で共有されます。それ以外の要求は拒否されます。*accesslist* に定義されるクライアントのリストは、Solaris 2.6 から拡張されました。詳細については、171 ページの「share コマンドを使ってアクセスリストを設定する」を参照してください。このオプションは `-ro` オプションよりも優先されます。

NFS 用 share オプション

NFS ファイルシステムで指定できるオプションは、次のとおりです。

aclok
: このオプションを指定すると、NFS version 2 プロトコルをサポートしている NFS サーバーが NFS version 2 クライアントのアクセス制御を行うように設定できます。このオプションを指定しないと、すべてのクライアントは最小限のアクセスしかできません。指定すると、最大限のアクセスができるようになります。たとえば `-aclok` オプションを指定して共有したファイルシステムでは、1 人のユーザーが読み取り権を持っていれば全員が読み取りを許可されます。このオプションを指定しないと、アクセス権を持つべきクライアントからのアクセスが拒否される可能性があります。ユーザーに与えるアクセス権は、既存のセキュリティーシステムによって決定します。アクセス制御リスト (ACL) の詳細は、『Solaris のシステム管理 (セキュリティサービス)』の「アクセス制御リストによる UFS ファイルの保護」を参照してください。

注 – アクセス制御リスト (ACL) を使用するには、クライアントとサーバーが、NFS version 3 プロトコルおよび NFS_ACL プロトコルをサポートしているソフトウェアを実行している必要があります。NFS version 3 プロトコルしかサポートしていないソフトウェアの場合、クライアントは正しいアクセス権を取得できますが、ACL を操作することはできません。NFS_ACL プロトコルをサポートしていれば、正しいアクセス権を取得した上で ACL の操作も可能です。この両方をサポートしているのは、Solaris 2.5 およびその互換バージョンです。

anon=*uid*
: *uid* は、認証されていないユーザーのユーザー ID を選択するために使用します。*uid* を `-1` に設定すると、認証されていないユーザーからのアクセスは拒否されます。`anon=0` とするとルートアクセス権を与えることができますが、これのオプションを指定すると、認証されていないユーザーにルートアクセス権を与えることになるため、代わりに `root` オプションを使用してください。

index=*filename*
: `-index=`*filename* オプションを使用すると、ユーザーが NFS URL にアクセスすると、ディレクトリのリストが表示されるのではなく、HTML (HyperText Markup Language) ファイルが強制的に読み込まれます。これは、HTTP URL がアクセスしているディレクトリに `index.html` ファイルが見つかるとブラウザのような動作をするというものです。このオプションを設定することは、`httpd` に対して `DirectoryIndex` オプションを指定するのと同じ意味です。たとえば、`dfstab` ファイルに、次のようなエントリがあるとします。

```
share -F nfs -o ro,public,index=index.html /export/web
```

このとき、次の URL によって表示される情報はすべて同じです。

```
nfs://<server>/<dir>
nfs://<server>/<dir>/index.html
nfs://<server>//export/web/<dir>
nfs://<server>//export/web/<dir>/index.html
http://<server>/<dir>
http://<server>/<dir>/index.html
```

log=*tag*
: このオプションは、ファイルシステム用の NFS サーバーログ構成情報の入った `/etc/nfs/nfslog.conf` 内のタグを指定します。NFS サーバーログ機能を使用可能にするにはこのオプションを選択する必要があります。

nosuid
: このオプションを使用すると、`setuid` モードまたは `setgid` モードを有効にしようとしても無視されます。NFS クライアントは、`setuid` か `setgid` のビットがオンの状態ではファイルを作成できません。

public
: -public オプションは、WebNFS ブラウズのために追加されました。このオプションで共有できるのは、1 台のサーバーにつき 1 つのファイルシステムだけです。

root=*accesslist*
: サーバーが、リスト上のホストに対してルートアクセス権を与えます。デフォルトでは、サーバーはどのリモートホストにもルートアクセス権は与えません。選択されているセキュリティーモードが -sec=sys 以外だと、*accesslist* に指定できるのはクライアントホスト名だけです。*accesslist* に定義されたクライアントのリストは、Solaris 2.6 で拡張されました。詳細については、171 ページの「share コマンドを使ってアクセスリストを設定する」を参照してください。

注意 – ほかのホストにルートアクセス権を与えるには、広い範囲でセキュリティーが保証されていることが前提です。-root= オプションは十分慎重に使用してください。

root=*client-name*
: *client-name* の値は、AUTH_SYS 認証で、exportfs(1B) で取得されたアドレスのリストにクライアントの IP アドレスが含まれているかどうかを検査するために使用します。リストに一致する IP アドレスが見つかった場合、クライアントに共有ファイルシステムへのルートアクセス権が与えられます。

root=*host-name*
: AUTH_SYS または RPCSEC_GSS などのセキュリティー保護された NFS モードの場合、サーバーは、アクセスリストから派生したホストベースの主体名のリストに、クライアントの主体名が含まれているかどうかを検査します。クライアント主体名の汎用構文は root@*hostname* です。Kerberos V の場合、構文は root/*hostname.fully.qualified*@REALM です。*host-name* の値を使用する場合、アクセスリスト上のクライアントには主体名の資格が必要になります。Kerberos V の場合、クライアントには root/*hostname.fully.qualified*@REALM の主体名の有効な keytab エントリが必要です。詳細は、『Solaris のシステム管理 (セキュリティサービス)』の「Kerberos クライアントの構成」を参照してください。

sec=*mode*[:*mode*]
: *mode* は、ファイルシステムへのアクセス権を取得するために必要なセキュリティーモードです。デフォルトのセキュリティーモードは、UNIX の認証です。モードは複数指定できますが、コマンド行に指定するときは 1 行につき 1 つのセキュリティーモードだけにしてください。各 -mode オプションはほかの -mode が検出されるまで、後続のすべての -rw、-ro、-rw=、- ro=、-root=、および -window= オプションに適用されます。-sec=none とすると、すべてのユーザーがユーザー nobody にマップされます。

window=*value*
value は、NFS サーバーで資格が有効な時間の上限です。デフォルトは 30000 秒 (8.3 時間) です。

share コマンドを使ってアクセスリストを設定する

2.6 より前の Solaris リリースでは、share コマンドの -ro=、-rw=、または -root= オプションに含まれる *accesslist* は、ホスト名またはネットグループ名のリストに限定されていました。Solaris 2.6 以降では、このアクセス制御リストにドメイン名、サブネット番号、およびアクセス権を与えないエントリも指定できます。この拡張により、名前空間を変更したり多数のクライアントを定義したリストを使用することなく、ファイルアクセス制御を単一のサーバーで簡単に管理できます。

次のコマンドは、ほとんどのシステムに読み取り専用アクセスを提供しますが、rose と lilac には読み取りと書き込みのアクセスを許可します。

```
# share -F nfs -o ro,rw=rose:lilac /usr/src
```

次の例では、eng ネットグループのすべてのホストで読み取りだけができるようになります。rose クライアントでは、読み取りと書き込みの両方ができます。

```
# share -F nfs -o ro=eng,rw=rose /usr/src
```

注 – 引数なしで rw と ro の両方を指定できません。読み書き可能オプションを指定しないと、デフォルトによってすべてのクライアントが読み書き可能になります。

複数のクライアントが 1 つのファイルシステムを共有するには、同じ行にすべてのオプションを入力する必要があります。同じオブジェクトに対して share コマンドを何度も実行しても、最後に実行されたコマンドだけが有効になります。次のコマンドでは、3 つのクライアントシステムで読み取りと書き込みができますが、rose と tulip では、ファイルシステムに root でアクセスできます。

```
# share -F nfs -o rw=rose:lilac:tulip,root=rose:tulip /usr/src
```

複数の認証メカニズムを使用するファイルシステムを共有する場合は、正しいセキュリティーモードの後に -ro、-ro=、-rw、-rw=、-root、および -window オプションを必ず含めるようにしてください。この例では、eng というネットグループ内のすべてのホストに対して UNIX 認証が選択されています。これらのホストは、ファイルシステムを読み取り専用モードでしかマウントできません。ホスト tulip と lilac は、Diffie-Hellman (DH) 認証を使用すれば読み書き可能でファイルシステムをマウントできます。これらのオプションを指定すると、tulip および lilac は、DH 認証を使用していない場合でも、ファイルシステムを読み取り専用でマウントすることができます。ただし、ホスト名が eng ネットグループに含まれている必要があります。

```
# share -F nfs -o sec=dh,rw=tulip:lilac,sec=sys,ro=eng /usr/src
```

デフォルトのセキュリティーモードは UNIX 認証ですが、-sec オプションを使用している場合、この UNIX 認証は含まれなくなります。そのため、UNIX 認証をほかの認証メカニズムとともに使用する場合は、-sec=sys オプションを指定する必要があります。

実際のドメイン名の前にドットを付けると、アクセスリスト中で DNS ドメイン名を使用できます。ドットの後の文字列はドメイン名です。完全指定のホスト名ではありません。次のエントリは、マウントから eng.example.com ドメイン内のすべてのホストへのアクセスを許可するためのものです。

```
# share -F nfs -o ro=.:.eng.example.com /export/share/man
```

この例で、「.」はそれぞれ NIS または NIS+ 名前空間を通じて一致するすべてのホストに対応します。ネームサービスから返される結果にはドメイン名は含まれません。「.eng.example.com」というエントリは、名前空間の解決に DNS を使用するすべてのホストに一致します。DNS が返すホスト名は必ず完全指定の名前になるので、DNS と他の名前空間を組み合わせると長いエントリが必要です。

実際のネットワーク番号かネットワーク名の前に「@」を指定すると、アクセスリストの中でサブネット番号を使用できます。この文字は、ネットワーク名をネットグループ名や完全指定のホスト名と区別するためです。サブネットは、/etc/networks の中か NIS または NIS+ 名前空間の中で識別できなければなりません。次のエントリは、サブネット 192.168 が eng ネットワークと識別されている場合、すべて同じ意味を持ちます。

```
# share -F nfs -o ro=@eng /export/share/man
# share -F nfs -o ro=@192.168 /export/share/man
# share -F nfs -o ro=@192.168.0.0 /export/share/man
```

2 番目と 3 番目のエントリは、ネットワークアドレス全体を指定する必要がないことを表しています。

ネットワークアドレスの先頭部分がバイトによる区切りでなく、CIDR (Classless Inter-Domain Routing) のようになっている場合には、マスクの長さをコマンド行で具体的に指定できます。この長さは、ネットワーク名かネットワーク番号の後ろにスラッシュで区切ってアドレスの接頭辞に有効ビット数として指定します。次に例を示します。

```
# share -f nfs -o ro=@eng/17 /export/share/man
# share -F nfs -o ro=@192.168.0/17 /export/share/man
```

この例で、「/17」はアドレスの先頭から 17 ビットがマスクとして使用されることを表します。CIDR の詳細は、RFC 1519 を参照してください。

また、エントリの前に「-」を指定することでアクセスの拒否を示すこともできます。エントリは左から右に読み込まれるため、アクセス拒否のエントリは次のようにそのエントリを適用するエントリの前に置く必要があることに注意してください。

```
# share -F nfs -o ro=-rose:.eng.example.com /export/share/man
```

この例では、eng.example.com ドメイン内のホストのうち、rose を除いたすべてに対してアクセスが許可されます。

unshare コマンド

このコマンドを使用すると、以前に使用可能な状態になっていたファイルシステムを、クライアントがマウントできないようにします。unshare コマンドを使用すると、share コマンドで共有したファイルシステムや、/etc/dfs/dfstab で自動的に共有しているファイルシステムが共有できないようになります。unshare コマンドを使って dfstab ファイルを使って共有していたファイルシステムの共有を解除する場合は、注意が必要です。一度実行レベル 3 を終了し再度実行すると、ファイルシステムは再度共有されます。実行レベル 3 を終了しても変更内容を継続させるには、そのファイルシステムを dfstab ファイルから削除する必要があります。

NFS ファイルシステムの共有を解除している場合、クライアントから既存マウントへのアクセスは禁止されます。クライアントにはファイルシステムがまだマウントされている可能性がありますが、ファイルにはアクセスできません。

注 – ファイルシステムの共有を解除してから再度共有するとき、NFS version 4 がどのように動作するかについては、184 ページの「NFS version 4 におけるファイルシステムの共有解除と再共有」を参照してください。

次の例では、指定したファイルシステムの共有が解除されます。

```
# unshare /usr/src
```

shareall コマンド

このコマンドを使用すると、複数のファイルシステムを共有することができます。オプションなしで使用すると、/etc/dfs/dfstab 内のすべてのエントリが共有されます。share コマンドを並べたファイルの名前を指定することができます。ファイル名を指定しないと、/etc/dfs/dfstab の内容が検査されます。「-」を使ってファイル名を置き換えれば、標準入力から share コマンドを入力できます。

次の例では、ローカルファイルに一覧表示されているすべてのファイルシステムが共有されます。

```
# shareall /etc/dfs/special_dfstab
```

unshareall コマンド

このコマンドを使用すると、現在共有されているリソースがすべて使用できなくなります。`-F` *FSType* オプションによって、`/etc/dfs/fstypes` に定義されているファイルシステムタイプのリストを選択します。このフラグによって、特定のタイプのファイルシステムだけを共有解除できます。デフォルトのファイルシステムタイプは、`/etc/dfs/fstypes` に定義されています。特定のファイルシステムを選択するには、`unshare` コマンドを使います。

次の例では、NFS タイプのファイルシステムの共有がすべて解除されます。

```
# unshareall -F nfs
```

showmount コマンド

このコマンドは、次のいずれかを表示します。

- NFS サーバーから共有している、リモートマウントされたファイルシステムを持つすべてのクライアント
- クライアントによってマウントされたファイルシステムのみ
- 共有されたファイルシステムおよびクライアントのアクセス情報

注 – `showmount` コマンドを使用すると、NFS version 2 と version 3 のエクスポートだけが表示され、NFS version 4 のエクスポートは表示されません。

コマンドは、次のような構文になります。

`showmount [ -ade ] [` *hostname* `]`

`-a` すべてのリモートマウントのリストを出力します。各エントリには、クライアント名とディレクトリが含まれます。

`-d` クライアントがリモートマウントしたディレクトリのリストを表示します。

`-e` 共有されているファイル、またはエクスポートされたファイルのリストを表示します。

hostname 表示する情報の取得元 NFS サーバーを指定します。

hostname を指定しない場合、ローカルホストの情報が表示されます。

次のコマンドでは、すべてのクライアント、およびマウントしたローカルディレクトリが表示されます。

```
# showmount -a bee
lilac:/export/share/man
lilac:/usr/src
rose:/usr/src
tulip:/export/share/man
```

次のコマンドでは、マウントしたディレクトリが表示されます。

```
# showmount -d bee
/export/share/man
/usr/src
```

次のコマンドでは、共有しているファイルシステムが表示されます。

```
# showmount -e bee
/usr/src                                (everyone)
/export/share/man                    eng
```

setmnt コマンド

このコマンドを使用すると、/etc/mnttab テーブルが作成されます。このテーブルは、mount コマンドと umount コマンドで参照されます。通常、このコマンドは、システムのブート時に自動的に実行されるため、手動で実行する必要はありません。

NFS のトラブルシューティング用のコマンド

NFS のトラブルシューティングには次のコマンドを使用します。

nfsstat コマンド

このコマンドを使用すると、NFS と RPC 接続について統計情報を収集できます。このコマンドの構文は次のとおりです。

nfsstat [-cmnrsz]

-c　クライアント側の情報を表示します

-m　NFS マウントされた各ファイルシステムの統計を表示します

-n　クライアント側とサーバー側の両方で、NFS の情報が表示されるように指定します

-r　RPC 統計を表示します

-s　サーバー側の情報を表示します

-z　統計をゼロに設定するように指定します

コマンド行にオプションを指定しないと、-cnrs が使用されます。

新しいソフトウェアやハードウェアを処理環境に追加した場合、サーバー側の統計を収集することが、デバッグにたいへん役立ちます。このコマンドを週に最低 1 度は実行し、履歴を作成するようにしてください。統計を保存しておくと、以前のパフォーマンスの有効な記録となります。

次の例を参照してください。

```
# nfsstat -s

Server rpc:
Connection oriented:
calls      badcalls   nullrecv   badlen     xdrcall    dupchecks  dupreqs
719949194  0          0          0          0          58478624   33
Connectionless:
calls      badcalls   nullrecv   badlen     xdrcall    dupchecks  dupreqs
73753609   0          0          0          0          987278     7254

Server nfs:
calls                badcalls
787783794            3516
Version 2: (746607 calls)
null       getattr    setattr    root       lookup     readlink   read
883 0%     60 0%      45 0%      0 0%       177446 23% 1489 0%    537366 71%
wrcache    write      create     remove     rename     link       symlink
0 0%       1105 0%    47 0%      59 0%      28 0%      10 0%      9 0%
mkdir      rmdir      readdir    statfs
26 0%      0 0%       27926 3%   108 0%
Version 3: (728863853 calls)
null          getattr       setattr       lookup        access
1365467 0%    496667075 68% 8864191 1%    66510206 9%   19131659 2%
readlink      read          write         create        mkdir
414705 0%     80123469 10%  18740690 2%   4135195 0%    327059 0%
symlink       mknod         remove        rmdir         rename
101415 0%     9605 0%       6533288 0%    111810 0%     366267 0%
link          readdir       readdirplus   fsstat        fsinfo
2572965 0%    519346 0%     2726631 0%    13320640 1%   60161 0%
pathconf      commit
13181 0%      6248828 0%
Version 4: (54871870 calls)
null               compound
266963 0%          54604907 99%
Version 4: (167573814 operations)
reserved           access             close              commit
0 0%               2663957 1%         2692328 1%         1166001 0%
create             delegpurge         delegreturn        getattr
167423 0%          0 0%               1802019 1%         26405254 15%
getfh              link               lock               lockt
11534581 6%        113212 0%          207723 0%          265 0%
locku              lookup             lookupp            nverify
230430 0%          11059722 6%        423514 0%          21386866 12%
open               openattr           open_confirm       open_downgrade
```

```
2835459 1%          4138 0%             18959 0%            3106 0%
putfh               putpubfh            putrootfh           read
52606920 31%        0 0%                35776 0%            4325432 2%
readdir             readlink            remove              rename
606651 0%           38043 0%            560797 0%           248990 0%
renew               restorefh           savefh              secinfo
2330092 1%          8711358 5%          11639329 6%         19384 0%
setattr             setclientid         setclientid_confirm verify
453126 0%           16349 0%            16356 0%            2484 0%
write               release_lockowner   illegal
3247770 1%          0 0%                0 0%

Server nfs_acl:
Version 2: (694979 calls)
null        getacl      setacl      getattr     access      getxattrdir
0 0%        42358 6%    0 0%        584553 84%  68068 9%    0 0%
Version 3: (2465011 calls)
null        getacl      setacl      getxattrdir
0 0%        1293312 52% 1131 0%     1170568 47%
```

このリストは、NFS サーバーの統計の例です。最初の 5 行は RPC に関するもので、残りの部分は NFS のアクティビティーのレポートです。どちらの統計でも、`badcalls` または `calls` の平均値、および各週の calls の数がわかるので、問題を特定するのに役立ちます。`badcalls` 値は、クライアントからの不良メッセージ数を示しています。この値は、ネットワークのハードウェアに問題が発生したことを示す場合があります。

いくつかの接続では、ディスクに対する書き込みアクティビティーが発生します。この数値の急激な上昇は障害の可能性を示すものなので、調査が必要です。NFS version 2 の統計で注意が必要なのは、`setattr`、`write`、`create`、`remove`、`rename`、`link`、`symlink`、`mkdir`、および `rmdir` です。NFS version 3 と version 4 では、`commit` の値に特に注意します。ある NFS サーバーの `commit` レベルが、それと同等のサーバーと比較して高い場合は、NFS クライアントに十分なメモリーがあるかどうかを確認してください。サーバーの `commit` オペレーションの数は、クライアントにリソースがない場合に上昇します。

pstack コマンド

このコマンドを使用すると、各プロセスのスタックトレースが表示されます。`pstack` コマンドは、必ずプロセスの所有者、または `root` として実行してください。`pstack` を使用して、プロセスがハングアップした場所を判断します。使用できるオプションは、確認するプロセスの `PID` だけです。proc(1) のマニュアルページを参照してください。

次の例では、実行中の `nfsd` プロセスを確認しています。

```
# /usr/bin/pgrep nfsd
243
```

```
# /usr/bin/pstack 243
243:    /usr/lib/nfs/nfsd -a 16
 ef675c04 poll     (24d50, 2, ffffffff)
 000115dc ???????? (24000, 132c4, 276d8, 1329c, 276d8, 0)
 00011390 main     (3, effff14, 0, 0, ffffffff, 400) + 3c8
 00010fb0 _start   (0, 0, 0, 0, 0, 0) + 5c
```

この例では、プロセスが新規の接続要求を持っていることが示されています。これは正常な反応です。要求が行われた後でもプロセスがポーリングしていることがスタックからわかった場合、そのプロセスはハングアップしている可能性があります。127 ページの「NFS サービスを再起動する方法」の指示に従って問題を解決してください。ハングアップしたプログラムによって問題が発生しているかどうかを確実に判断するには、123 ページの「NFS のトラブルシューティングの手順」を参照してください。

rpcinfo コマンド

このコマンドは、システムで動作している RPC サービスに関する情報を生成します。RPC サービスの変更にも使用できます。このコマンドには、たくさんのオプションがあります。rpcinfo(1M) のマニュアルページを参照してください。次は、このコマンドで使用できるオプションの構文です。

`rpcinfo [ -m | -s ] [ `*hostname*` ]`

`rpcinfo -T `*transport hostname*` [ `*progname*` ]`

`rpcinfo [ -t | -u ] [ `*hostname*` ] [ `*progname*` ]`

`-m`	rpcbind 処理の統計テーブルを表示します
`-s`	登録されているすべての RPC プログラムを簡易リストで表示します
`-T`	特定のトランスポートまたはプロトコルを使用するサービスの情報を表示します
`-t`	TCP を使用する RPC プログラムを検索します
`-u`	UDP を使用する RPC プログラムを検索します
transport	サービスに使用するトランスポートまたはプロトコルを選択します
hostname	必要な情報の取得元のサーバーのホスト名を選択します
progname	情報の取得対象の RPC プログラムを選択します

hostname を指定しないと、ローカルホスト名が使用されます。*progname* の代わりに RPC プログラム番号が使用できますが、ユーザーが覚えやすいのは番号よりも名前です。NFS version 3 が実行されていないシステムでは、`-s` オプションの代わりに `-p` オプションを使用できます。

このコマンドを実行すると、次の項目を含むデータを生成することができます。

- RPC プログラム番号
- 特定プログラムのバージョン番号
- 使用されているトランスポートプロトコル
- RPC サービス名
- RPC サービスの所有者

次の例では、サーバーで実行されている RPC サービスに関する情報を収集しています。生成されたテキストには `sort` コマンドのフィルタをかけ、より読みやすくしています。この例では、RPC サービスの数行を省略しています。

```
% rpcinfo -s bee |sort -n
   program version(s) netid(s)                         service     owner
    100000  2,3,4     udp6,tcp6,udp,tcp,ticlts,ticotsord,ticots rpcbind     superuser
    100001  4,3,2     ticlts,udp,udp6                   rstatd      superuser
    100002  3,2       ticots,ticotsord,tcp,tcp6,ticlts,udp,udp6 rusersd     superuser
    100003  3,2       tcp,udp,tcp6,udp6                 nfs         superuser
    100005  3,2,1     ticots,ticotsord,tcp,tcp6,ticlts,udp,udp6 mountd      superuser
    100007  1,2,3     ticots,ticotsord,ticlts,tcp,udp,tcp6,udp6 ypbind      superuser
    100008  1         ticlts,udp,udp6                   walld       superuser
    100011  1         ticlts,udp,udp6                   rquotad     superuser
    100012  1         ticlts,udp,udp6                   sprayd      superuser
    100021  4,3,2,1   tcp,udp,tcp6,udp6                 nlockmgr    superuser
    100024  1         ticots,ticotsord,ticlts,tcp,udp,tcp6,udp6 status      superuser
    100029  3,2,1     ticots,ticotsord,ticlts           keyserv     superuser
    100068  5         tcp,udp                           cmsd        superuser
    100083  1         tcp,tcp6                          ttdbserverd superuser
    100099  3         ticotsord                         autofs      superuser
    100133  1         ticots,ticotsord,ticlts,tcp,udp,tcp6,udp6 -           superuser
    100134  1         ticotsord                         tokenring   superuser
    100155  1         ticots,ticotsord,tcp,tcp6         smserverd   superuser
    100221  1         tcp,tcp6                          -           superuser
    100227  3,2       tcp,udp,tcp6,udp6                 nfs_acl     superuser
    100229  1         tcp,tcp6                          metad       superuser
    100230  1         tcp,tcp6                          metamhd     superuser
    100231  1         ticots,ticotsord,ticlts           -           superuser
    100234  1         ticotsord                         gssd        superuser
    100235  1         tcp,tcp6                          -           superuser
    100242  1         tcp,tcp6                          metamedd    superuser
    100249  1         ticots,ticotsord,ticlts,tcp,udp,tcp6,udp6 -           superuser
    300326  4         tcp,tcp6                          -           superuser
    300598  1         ticots,ticotsord,ticlts,tcp,udp,tcp6,udp6 -           superuser
    390113  1         tcp                               -           unknown
 805306368  1         ticots,ticotsord,ticlts,tcp,udp,tcp6,udp6 -           superuser
1289637086  1,5       tcp                               -           26069
```

次の例では、サーバーの特定トランスポートを選択して、RPC サービスの情報を収集する方法について説明しています。最初の例では、TCP で実行されている `mountd` サービスをチェックしています。2 番目の例では、UDP で実行されている NFS サービスをチェックしています。

```
% rpcinfo -t bee mountd
program 100005 version 1 ready and waiting
program 100005 version 2 ready and waiting
program 100005 version 3 ready and waiting
% rpcinfo -u bee nfs
program 100003 version 2 ready and waiting
program 100003 version 3 ready and waiting
```

snoop コマンド

このコマンドは、ネットワーク上のパケットの監視によく使用されます。snoop コマンドは、root で実行する必要があります。このコマンドは、クライアントとサーバーの両方で、ネットワークハードウェアが機能しているかどうかを確認する方法としてよく使用されます。使用できるオプションは多数あります。snoop(1m) のマニュアルページを参照してください。次で、このコマンドの概要を説明します。

snoop [-d *device*] [-o *filename*] [host *hostname*]

-d *device*　　ローカルネットワークのインタフェースを指定します

-o *filename*　　受信したすべてのパケットを指定したファイルに保存します

hostname　　特定のホストが送受信したパケットを表示します

-d *device* オプションは、複数のネットワークインタフェースがあるサーバーで特に有効です。ホストの設定以外にも、使用できる式が多数あります。コマンド正規表現を grep で組み合わせることでも、十分に使用できるデータを生成できます。

トラブルシューティングをする場合は、パケットの発信元と送信先のホストが正しいことを確認してください。また、エラーメッセージも調べてください。パケットをファイルに保存すると、データを簡単に参照することができます。

truss コマンド

このコマンドを使用すると、プロセスがハングアップしたかどうかを確認できます。truss コマンドは、必ずプロセスの所有者、または root として実行してください。このコマンドに指定できるオプションは多数あります。truss(1) のマニュアルページを参照してください。次で、このコマンドの構文を説明します。

truss [-t *syscall*] -p *pid*

-t *syscall*　　追跡するシステムコールを選択します

-p *pid*　　追跡するプロセスの PID を指定します

syscall には、追跡するシステムコールをコンマで区切って指定することもできます。また、*syscall* の指定を ! で始めると、そのシステムコールは追跡されなくなります。

次の例は、プロセスが新しいクライアントからの接続要求を待っていることを示しています。

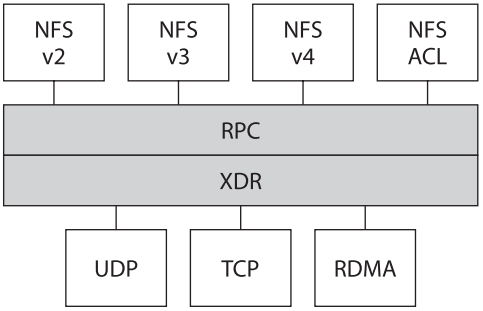
```
# /usr/bin/truss -p 243
poll(0x00024D50, 2, -1)         (sleeping...)
```

これは正常な反応です。新規接続の要求が行われた後でも反応が変わらない場合、そのプロセスはハングアップしている可能性があります。127 ページの「NFS サービスを再起動する方法」の指示に従ってハングアップの問題を解決してください。ハングアップしたプログラムによって問題が発生しているかどうかを確実に判断するには、123 ページの「NFS のトラブルシューティングの手順」を参照してください。

RDMA 経由の NFS

Solaris 10 リリースでは、RDMA (Remote Direct Memory Access) プロトコルが導入されています。RDMAは、高速ネットワーク上でデータのメモリー間転送を行うテクノロジです。特に、RDMA により、CPU の介入なしでメモリーに遠隔データ転送を直接行えます。また、データを直接配置できます。これは、データのコピーを省略し、さらに CPU の介入も省略します。このように、RDMA はホストの CPU を解放するだけでなく、ホストのメモリーと入出力バスの接続も減らします。この機能を提供するために、RDMA は、SPARC プラットフォーム上の InfiniBand のインターコネクト入出力テクノロジと Solaris オペレーティングシステムを組み合わせます。次の図は、UDP や TCP など、その他のプロトコルとの RDMA の関係を示します。

図 6–1 その他のプロトコルとの RDMA の関係



NFS は RPC の上に重ねて階層化したプロトコル群です。
XDR (eXternal Data Representation) 層は、RPC の引数や結果を、UDP、TCP、RDMA などいくつかの RPC トランスポートの 1 つにエンコードします。

RDMA トランスポートをクライアントとサーバーで使用できない場合、TCP トランスポートが初期フォールバックになります。TCP が使用できない場合は UDP がフォールバックになります。ただし、`proto=rdma` マウントオプションを使用する場合、NFS マウントは強制的に RDMA だけになることに注意してください。

NFS マウントオプションの詳細は、`mount_nfs(1M)` のマニュアルページおよび 159 ページの「`mount` コマンド」を参照してください。

注 – InfiniBand の RDMA は、IP アドレス指定形式および IP ルックアップインフラストラクチャーを使用して、ピアを指定します。ただし、RDMA は、独立したプロトコルスタックであるため、すべての IP のセマンティクスを完全には実装しません。たとえば、RDMA はピアと通信するための IP アドレス指定を使用しません。したがって、RDMA は、IP アドレスに基づいたさまざまなセキュリティーポリシーの設定を省略することがあります。ただし、`mount` 制限や Secure RPC などの NFS と RPC の管理ポリシーは省略されません。

NFS サービスのしくみ

次の節では、NFS の複雑な機能をいくつか紹介します。この節で紹介する機能のいくつかは、NFS version 4 専用であることに注意してください。

- 183 ページの「NFS におけるバージョンのネゴシエーション」
- 184 ページの「NFS version 4 における機能」
- 194 ページの「UDP と TCP のネゴシエーション」

注 – システムでゾーンが有効なときに非大域ゾーンでこの機能を使用するには、『Oracle Solaris のシステム管理 (Oracle Solaris コンテナ : 資源管理と Oracle Solaris ゾーン)』を参照してください。

NFS におけるバージョンのネゴシエーション

NFS 起動プロセスには、サーバーとクライアントのプロトコルレベルのネゴシエーションが含まれています。バージョンのレベルを指定しない場合、デフォルトにより最適なレベルが選択されます。たとえば、クライアントとサーバーの両方が version 3 をサポートしていると、version 3 が使用されます。クライアントまたはサーバーが version 2 しかサポートしていないと、version 2 が使用されます。

Solaris 10 以降のリリースでは、`/etc/default/nfs` ファイルにキーワード NFS_CLIENT_VERSMIN、NFS_CLIENT_VERSMAX、NFS_SERVER_VERSMIN、NFS_SERVER_ を設定できます。これらのキーワードのデフォルト値に代わって、クライアントとサーバーに最小値と最大値を指定できます。クライアントとサーバーの最小値は 2 がデフォルトで、最大値は 4 がデフォルトです。142 ページの「`/etc/default/nfs` ファイルのキーワード」 を参照してください。サーバーがサポートするバージョンを検出するために、NFS クライアントは NFS_CLIENT_VERSMAX の設定から始めて、NFS_CLIENT_VERSMIN のバージョン設定に到るまで各バージョンを試行し続けます。サポートされるバージョンが検出されるとすぐに、この処理は終了します。たとえば、NFS_CLIENT_VERSMAX=4 および NFS_CLIENT_VERSMIN=2 と設定すると、クライアントは version 4、version 3、version 2 の順に試行します。NFS_CLIENT_VERSMIN と NFS_CLIENT_VERSMAX が同じ値に設定されていると、クライアントは常にこの設定されたバージョンを使用し、その他のバージョンは試行しません。サーバーがこのバージョンをサポートしていない場合、マウントは失敗します。

注 – ネゴシエーションによって決まった値を変更するには、`mount` コマンドで `vers` オプションを使用します。`mount_nfs(1M)` のマニュアルページを参照してください。

手順については、94 ページの「NFS サービスの設定」を参照してください。

NFS version 4 における機能

version 4 の NFS は大幅に変更が行われました。この節では、これらの新しい機能を説明します。

注 – Solaris 10 以降のリリースでは、NFS version 4 は LIPKEY/SPKM セキュリティー方式をサポートしません。また、`mountd`、`nfslogd`、および `statd` デーモンを使用しません。

NFS version 4 の使用に関する手順については、94 ページの「NFS サービスの設定」を参照してください。

NFS version 4 におけるファイルシステムの共有解除と再共有

NFS version 3 と version 4 では、クライアントが共有を解除されたファイルシステムにアクセスしようとすると、サーバーはエラーコードを返します。ただし、NFS version 3 では、ファイルシステムが共有されなくなる前に、サーバーはクライアントが取得したロックを保持します。したがって、ファイルシステムが再度共有されるとき、NFS version 3 クライアントは、そのファイルシステムが共有解除されなかったかのように、ファイルシステムにアクセスできます。

NFS version 4 では、ファイルシステムの共有を解除するとき、そのファイルシステムにあるオープンファイルまたはファイルロックの状態がすべて削除されます。クライアントは、これらのファイルにアクセスしようとしたりロックしようとしたりすると、エラーを受け取ります。通常、このエラーは、アプリケーションに対する入出力エラーとして報告されます。ただし、オプションを変更するために現在共有されているファイルシステムを再共有しても、サーバーの状態は削除されません。

関連情報については、188 ページの「NFS version 4 におけるクライアント回復」を参照するか、`unshare_nfs(1M)` のマニュアルページを参照してください。

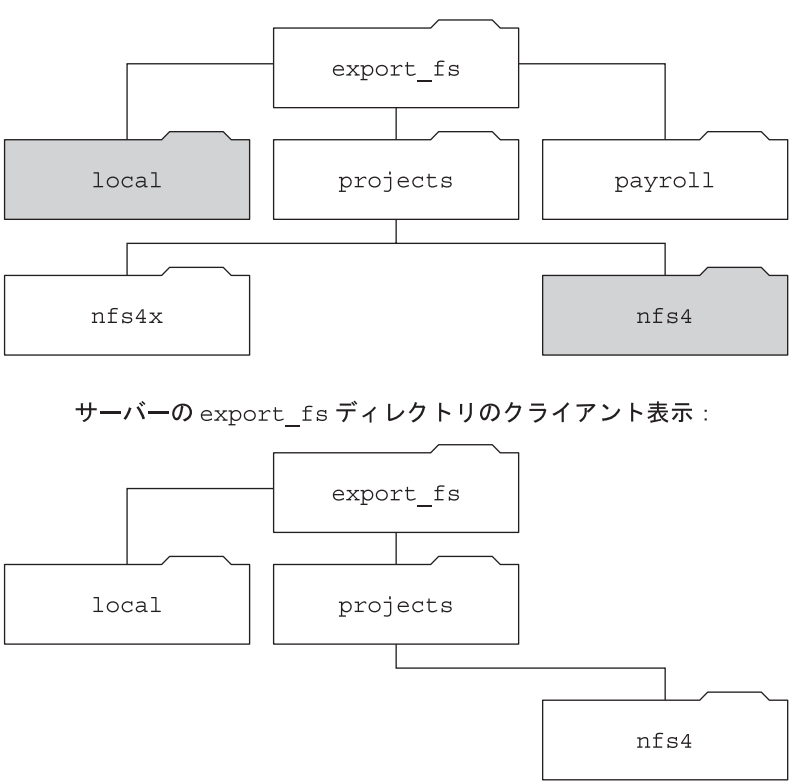
NFS version 4 におけるファイルシステムの名前空間

NFS version 4 サーバーは、擬似ファイルシステムを作成し管理します。擬似ファイルシステムにより、クライアントは、サーバー上のエクスポートされた全ファイルにシームレスにアクセスできます。NFS version 4 より前のバージョンには、擬似ファイルシステムがありません。クライアントは、アクセスする各共有サーバーのファイルシステムに強制的にマウントされます。次のような例を考えます。

図 6–2 サーバーのファイルシステムとクライアントのファイルシステムの表示

サーバーエクスポート：
/export_fs/local
/export_fs/projects/nfs4

サーバーファイルシステム：
/
/export_fs



クライアントには payroll ディレクトリと nfs4x ディレクトリは表示されません。これらのディレクトリはエクスポートされておらず、エクスポートされたディレクトリには通じていないためです。ただし、local ディレクトリは、エクスポートされたディレクトリであるため、クライアントに表示されます。projects ディレクトリは、エクスポートされたディレクトリ nfs4 に通じているため、クライアントに表示されます。このように、明示的にエクスポートされていないサーバーの名前空間の部分は、擬似ファイルシステムで橋渡しされます。この擬似ファイルシステムは、エクスポートされたディレクトリ、およびサーバーのエクスポートに通じるディレクトリだけを表示します。

擬似ファイルシステムは、ディレクトリだけを含む構造で、サーバーによって作成されます。擬似ファイルシステムにより、クライアントはエクスポートされたファイルシステムの階層を検索できるようになります。このようにして、クライアントの擬似ファイルシステムの表示は、エクスポートされたファイルシステムに通じるパスに限定されます。

以前のバージョンの NFS では、クライアントは、サーバーのファイルシステムを検索するには、各ファイルシステムをマウントする必要がありました。しかし、NFS version 4 では、サーバーの名前空間が次のことを行います。

- クライアントのファイルシステム表示を、サーバーのエクスポートに通じるディレクトリに限定します。
- クライアントが配下の各ファイルシステムをマウントしなくても、サーバーのエクスポートにシームレスにアクセスできるようにします。前述の例を参照してください。オペレーティングシステムが異なるとき、クライアントはサーバーの各ファイルシステムをマウントする必要のある場合があります。

POSIX に関連する理由により、Solaris NFS version 4 クライアントは、サーバーのファイルシステムの境界を越えません。境界を越えようとすると、クライアントはディレクトリを空のように見せます。この状況に対処するには、サーバーのファイルシステムごとにマウントを行う必要があります。

NFS version 4 における揮発性ファイルハンドル

ファイルハンドルは、サーバー上で作成され、ファイルとディレクトリを一意に識別する情報を持ちます。NFS version 2 と version 3 では、サーバーは持続的ファイルハンドルを返しました。したがって、クライアントは、サーバーが常に同じファイルを参照するファイルハンドルを生成することを保証できました。次に例を示します。

- ファイルが削除され同じ名前のファイルに置き換えられた場合、サーバーは必ず新しいファイルの新しいファイルハンドルを生成する。クライアントが古いファイルハンドルを使用していた場合、サーバーはファイルハンドルが無効であることを示すエラーを返す。
- ファイル名が変更されている場合、ファイルハンドルは変更されない。

- サーバーをリブートする必要があった場合、ファイルハンドルは変更されない。

このように、サーバーがファイルハンドルを含むクライアントからの要求を受け取った場合、解決策は単純であり、ファイルハンドルは常に正しいファイルを参照します。

NFS を操作するためのファイルとディレクトリを識別するこの方法は、多くの UNIX ベースのサーバーに適しています。ただし、この方法は、ファイルのパス名などほかの識別方法を使用するサーバー上では実装できません。この問題を解決するために、NFS version 4 プロトコルは、サーバーがそのファイルハンドルが揮発性であることを宣言できるようにします。したがって、ファイルハンドルが変更されます。ファイルハンドルが変更された場合、クライアントは新しいファイルハンドルを検出する必要があります。

NFS version 2 と 3 のように、Solaris NFS version 4 サーバーは常に持続的ファイルハンドルを提供します。ただし、Solaris NFS version 4 以外のサーバーにアクセスする Solaris NFS version 4 クライアントは、そのサーバーが揮発性ファイルハンドルを使用する場合、揮発性ファイルハンドルをサポートする必要があります。特に、サーバーがクライアントにファイルハンドルが揮発性であることを知らせている場合は、クライアントはパス名とファイルハンドル間のマッピングをキャッシュする必要があります。クライアントは、期限切れになるまで、揮発性ファイルハンドルを使用します。期限が切れたとき、クライアントは次を実行します。

- そのファイルハンドルを参照するキャッシュされた情報をフラッシュする
- そのファイルの新しいファイルハンドルを検索する
- 操作をもう一度実行する

注 – サーバーは、どのファイルハンドルが持続的あるいは揮発性かを、クライアントに常に知らせます。

揮発性ファイルハンドルは、次のいずれかの理由により期限切れになります。

- ファイルを閉じたとき
- ファイルハンドルのファイルシステムが移行するとき
- クライアントがファイル名を変更するとき
- サーバーがリブートするとき

クライアントが新しいファイルハンドルを検索できない場合、エラーメッセージが `syslog` ファイルに追加されます。このファイルにアクセスしようとすると、入出力エラーで失敗します。

NFS version 4 におけるクライアント回復

NFS version 4 プロトコルは、ステートフルプロトコルです。クライアントとサーバーが次の項目に関する現在の情報を管理するとき、プロトコルはステートフルです。

- オープンファイル
- ファイルロック

サーバーのクラッシュなどの障害が発生したとき、クライアントとサーバーは連携して、障害が発生する前のオープン状態とロック状態を再度確立します。

サーバーがクラッシュしてリブートしたとき、サーバーの状態は消失します。クライアントは、サーバーがリブートしたことを検出して、サーバーの状態の再構築を支援するプロセスを開始します。このプロセスは、クライアントがプロセスを指示するため、クライアント回復として知られています。

クライアントは、サーバーがリブートしたことを検出すると、ただちに現在の動作を停止して、クライアント回復のプロセスを開始します。回復プロセスが開始されたとき、次のようなメッセージが、システムエラーログ `/var/adm/messages` に表示されます。

```
NOTICE: Starting recovery server basil.example.company.com
```

回復プロセスの間、クライアントは、クライアントの以前の状態に関するサーバー情報を送信します。ただし、この間、クライアントはサーバーに新しい要求を送信しません。ファイルのオープンやファイルロックの設定の新しい要求は、サーバーが回復を完了するのを待ってから続行する必要があります。

クライアント回復プロセスが完了したとき、次のメッセージがシステムエラーログ `/var/adm/messages` に表示されます。

```
NOTICE: Recovery done for server basil.example.company.com
```

クライアントは、サーバーへの状態情報の送信を正常に完了しました。ただし、クライアントがこのプロセスを完了しても、その他のクライアントがサーバーに状態情報を送信するプロセスを完了していない可能性があります。したがって、しばらくの間、サーバーはオープンまたはロック要求を受け付けません。この期間は猶予期間として知られており、すべてのクライアントが回復を完了できるように指定されています。

猶予期間中に、クライアントが新しいファイルを開こうとしたり、新しいロックを確立しようとしたりすると、サーバーは `GRACE` エラーコードで要求を拒否します。このエラーを受け取ったとき、クライアントは猶予期間が終わるのを待ってから、要求をサーバーに再送信します。猶予期間中は、次のメッセージが表示されます。

```
NFS server recovering
```

猶予期間中、ファイルを開いたりファイルロックを設定したりしないコマンドは処理できることに注意してください。たとえば、コマンド ls と cd はファイルを開いたりファイルロックを設定したりしません。したがって、これらのコマンドは中断されません。ただし、ファイルを開く cat などのコマンドは、猶予期間が終わるまで中断されます。

猶予期間が終了すると、次のメッセージが表示されます。

```
NFS server recovery ok.
```

クライアントは、サーバーに新しいオープン要求またはロック要求を送信できるようになります。

クライアント回復は、さまざまな理由により失敗することがあります。たとえば、サーバーのリブート後にネットワークパーティションが存在する場合、クライアントは、猶予期間が終了する前にサーバーとの状態を再度確立できません。猶予期間が終了すると、新しい状態操作により競合が発生するため、サーバーはクライアントに状態の再確立を許可しません。たとえば、新しいファイルロックは、クライアントが回復しようとしている古いファイルロックと競合します。このような状況が発生すると、サーバーは NO_GRACE エラーコードをクライアントに返します。

特定のファイルに対するオープン操作の回復が失敗すると、クライアントはファイルを使用不可能としてマークし、次のメッセージが表示されます。

```
WARNING: The following NFS file could not be recovered and was marked dead
(can't reopen:  NFS status 70):  file :  filename
```

番号 70 は 1 つの例です。

回復中にファイルロックの再確立が失敗した場合、次のエラーメッセージが送信されます。

```
NOTICE: nfs4_send_siglost:  pid PROCESS-ID lost
lock on server SERVER-NAME
```

この場合、SIGLOST シグナルがプロセスに送信されます。SIGLOST シグナルのデフォルトの動作は、プロセスを中断することです。

この状態から回復するには、障害発生時にファイルを開いていたすべてのアプリケーションを再起動する必要があります。次のことに注意してください。

- ファイルを再度開くことができない一部のプロセスは入出力エラーを受け取ります。
- ファイルを再度開いたり、または回復の失敗後にオープン操作を実行したその他のプロセスは、問題なくファイルにアクセスできます。

このように、特定のファイルにアクセスできるプロセスとアクセスできないプロセスがあります。

NFS version 4 における OPEN 共有サポート

NFS version 4 プロトコルには、クライアントがほかのクライアントによるファイルアクセスの制御に使用するファイル共有モードがいくつかあります。クライアントは、次のように指定できます。

- DENY_NONE モードを指定すると、ほかのクライアントはファイルへの読み取りと書き込みアクセスを許可されます。
- DENY_READ モードを指定すると、ほかのクライアントはファイルへの読み取りアクセスを拒否されます。
- DENY_WRITE モードを指定すると、ほかのクライアントはファイルへの書き込みアクセスを拒否されます。
- DENY_BOTH モードを指定すると、ほかのクライアントはファイルへの読み取りと書き込みアクセスを拒否されます。

Solaris NFS version 4 サーバーは、これらのファイル共有モードを完全に実装します。したがって、クライアントが現在の共有モードと矛盾する方法でファイルを開こうとすると、サーバーは操作を失敗させて、その試行を拒否します。このような試行が、ファイルのオープン操作または作成操作の開始に失敗すると、Solaris NFS version 4 クライアントはプロトコルエラーを受け取ります。このエラーは、アプリケーションエラー EACCES にマップされます。

プロトコルにはいくつかの共有モードがありますが、現在のところ、Solaris でのオープン操作では、複数の共有モードを提供していません。ファイルを開くとき、Solaris NFS version 4 クライアントは、DENY_NONE モードだけを使用します。

また、Solaris fcntl システムコールには、ファイルの共有を制御する F_SHARE コマンドがありますが、fcntl コマンドは NFS version 4 では正しく実装されません。NFS version 4 クライアントで fcntl コマンドを使用すると、クライアントはアプリケーションに EAGAIN エラーを返します。

NFS version 4 における委託

NFS version 4 は、委託のクライアントサポートとサーバーサポートを提供します。委託とは、サーバーがファイルの管理をクライアントに委託するテクニックです。たとえば、サーバーは、読み取り委託または書き込み委託のいずれかをクライアントに付与できます。読み込み委託は互いに競合しないため、複数のクライアントに同時に付与できます。書き込み委託はほかのクライアントのファイルアクセスと競合するため、1 つのクライアントにだけ付与できます。書き込み委託を保持している間、クライアントは、ファイルへの排他的アクセスを保証されているために、さまざまな操作をサーバーに送信しません。同様に、読み込み委託を保持している

間、クライアントはさまざまな操作をサーバーに送信しません。クライアントが書き込みモードでファイルを開けないことをサーバーが保証するためです。委託により、委託されたファイルに対するサーバーとクライアントの相互作用を大幅に減少することができます。したがって、ネットワークトラフィックが減少し、クライアントとサーバーのパフォーマンスが向上します。ただし、パフォーマンス向上の度合いは、アプリケーションが使用するファイルの相互作用の種類およびネットワークとサーバー輻輳の量によって異なります。

委託を付与するかどうかの決定は、サーバーがすべて行います。クライアントは、委託を要求しません。サーバーは、ファイルに対するアクセスパターンに基づいて、委託を付与するかどうかを決定します。複数の異なるクライアントから書き込みモードで、ファイルが最近アクセスされた場合、サーバーは委託を付与しないことがあります。このアクセスパターンは将来競合する可能性があることを示しているためです。

競合は、ファイルに付与されている委託と一致しない方法でクライアントがそのファイルにアクセスするときに発生します。たとえば、あるクライアントがファイルの書き込み委託を保持しており、2 番目のクライアントが読み取りまたは書き込みアクセス用にそのファイルを開くとサーバーは最初のクライアントの書き込み委託を再呼び出しします。同様に、あるクライアントが読み取り委託を保持しており、別のクライアントが書き込み用に同じファイルを開くと、サーバーは読み取り委託を再呼び出しします。どちらの場合も、競合が存在しているため、2 番目のクライアントは委託を付与されません。競合が発生すると、サーバーはコールバックメカニズムを使用して、委託を保持しているクライアントと連絡をとります。このコールバックを受信すると、クライアントはファイルの更新された状態をサーバーに送信し、委託を返します。クライアントが再呼び出しに対する応答に失敗すると、サーバーは委託を取り消します。こうした場合、サーバーはこのファイルに対するクライアントの操作をすべて拒否し、クライアントは要求された操作を失敗として報告します。一般的に、これらの失敗は入出力エラーとしてアプリケーションに報告されます。これらのエラーから回復するには、ファイルを閉じてから再度開く必要があります。取り消された委託による失敗は、クライアントが委託を保持している間にクライアントとサーバー間にネットワークパーティションが存在しているときに発生します。

サーバーは、別のサーバーに格納されているファイルに対するアクセスの競合を解決できません。つまり、NFS サーバーは、格納しているファイルに対する競合だけを解決します。さらに、さまざまなバージョンの NFS を実行しているクライアントによって発生する競合に対して、NFS サーバーは NFS version 4 を実行しているクライアントにだけ再呼び出しを開始します。以前のバージョンの NFS を実行しているクライアントに再呼び出しを開始できません。

競合を検出するプロセスはさまざまです。たとえば、NFS version 4 とは異なり、version 2 と version 3 にはオープン手順がないため、クライアントがファイルの読み取り、書き込み、またはロックを試行したあとでのみ、競合が検出されます。これらの競合に対するサーバーの応答もさまざまです。次に例を示します。

- NFS version 3 では、サーバーは JUKEBOX エラーを返します。これにより、クライアントはアクセス要求を停止し、あとで再試行します。クライアントは、`File unavailable` というメッセージを出力します。
- NFS version 2 では、JUKEBOX エラーと同等のエラーが存在しないため、サーバーは応答しません。これにより、クライアントは待機してから再試行します。クライアントは、`NFS server not responding` というメッセージを出力します。

これらの状態は、委託の競合が解決されたときにクリアされます。

デフォルトでは、サーバー委託は有効になっています。`/etc/default/nfs` ファイルを変更すると、委託を無効にできます。手順については、97 ページの「サーバー上で異なるバージョンの NFS を選択する方法」を参照してください。

クライアントの委託にキーワードは必要ありません。NFS version 4 コールバックデーモン `nfs4cbd` により、クライアント上のコールバックサービスが提供されます。このデーモンは、NFS version 4 のマウントが有効になると自動的に起動されます。デフォルトで、クライアントは、`/etc/netconfig` システムファイルに一覧表示されているすべてのインターネット転送に必要なコールバック情報を提供します。クライアントで IPv6 が有効であり、クライアントの名前の IPv6 アドレスが指定されている場合、コールバックデーモンは IPv6 接続を受け入れます。

コールバックデーモンは、一時的なプログラム番号と動的に割り当てられたポート番号を使用します。この情報は、サーバーに提供され、サーバーは委託を付与する前にコールバックパスをテストします。コールバックパスが正常にテストされない場合、サーバーは委託を付与しません。外部から見ることのできる動作だけになります。

コールバック情報は NFS version 4 要求に組み込まれているため、サーバーは、NAT (Network Address Translation) を使用するデバイスを通してクライアントと連絡を取ることができません。また、コールバックデーモンは、動的ポート番号も使用します。したがって、ファイアウォールがポート 2049 上で通常の NFS トラフィックを有効にしている場合でも、サーバーがファイアウォールを検索できない場合があります。この場合、サーバーは委託を付与しません。

NFS version 4 での ACL と `nfsmapid`

アクセス制御リスト (ACL) は、ファイルの所有者が、ファイル所有者、グループ、そのほかの固有のユーザーおよびグループに関するファイルアクセス権を定義できるようにすることで、ファイルのセキュリティーを高めます。ACL は、`setfacl` コマンドを使用することで、サーバーおよびクライアント上で設定され

ます。詳細については、setfacl(1) のマニュアルページを参照してください。NFS version 4 では、ID マッパー nfsmapid を使用して、サーバー上の ACL エントリ内のユーザーまたはグループ ID を、クライアント上の ACL エントリ内のユーザーまたはグループ ID にマッピングします。逆も同じです。ACL エントリのユーザーおよびグループ ID は、クライアントとサーバーの両方に存在する必要があります。

ID マッピングが失敗する理由

次の状態は、ID マッピングが失敗する原因になる可能性があります。

- サーバー上の ACL エントリ内に存在するユーザーまたはグループをクライアント上の有効なユーザーまたはグループにマッピングできない場合、ユーザーはクライアント上の ACL を読み取ることができません。

  たとえば、ls -lv や ls -lV コマンドを発行した場合、サーバーからクライアントにマッピングできないユーザーまたはグループ ID ACL エンティティーを含むファイルに対して、Permission denied エラーメッセージが表示されます。ID マッパーは ACL 内のユーザーまたはグループをマッピングできません。ID マッパーがユーザーまたはグループをマッピングできた場合、ls -l により生成されるファイルリストのアクセス権のあとにはプラス (+) 記号が表示されています。次に例を示します。

  ```
  % ls -l
  -rw-r--rw-+   1 luis    staff      11968 Aug 12  2005 foobar
  ```

  同様に、同じ理由により getfacl コマンドは Permission denied エラーメッセージを返すことができます。このコマンドの詳細は、getfacl(1) のマニュアルページを参照してください。
- クライアント上で設定されている ACL エントリ内のユーザーまたはグループ ID をサーバー上の有効なユーザーまたはグループ ID にマッピングできない場合、setfacl や chmod コマンドが失敗し、 Permission denied エラーメッセージを返す可能性があります。
- クライアントとサーバーで NFSMAPID_DOMAIN の値が一致しない場合、ID マッピングは失敗します。詳細は、142 ページの「/etc/default/nfs ファイルのキーワード」を参照してください。

ACL を使用した ID マッピングの問題を回避する

ID マッピングの問題を回避するには、次の処置を行います。

- NFSMAPID_DOMAIN の値が /etc/default/nfs ファイル内で正しく設定されていることを確認します。
- ACL エントリ内のすべてのユーザーおよびグループ ID が NFS version 4 のクライアントとサーバーの両方に存在することを確認します。

ACL エントリ内のすべてのユーザーおよびグループ ID が NFS version 4 のクライアントとサーバーの両方に存在することを確認します。

サーバーまたはクライアント上でユーザーまたはグループをマッピングできるかどうかを判別するには、次のスクリプトを使用します。

```
#! /usr/sbin/dtrace -Fs

sdt:::nfs4-acl-nobody
{
     printf("validate_idmapping: (%s) in the ACL could not be mapped!",
stringof(arg0));
}
```

注 – このスクリプトで使用されているプローブ名は、将来変更される可能性があるインタフェースです。詳細については、『Solaris 動的トレースガイド』の「安定性レベル」を参照してください。

ACL または nfsmapid の追加情報

次を参照してください。

- 『Solaris のシステム管理 (セキュリティサービス)』の「ACL による UFS ファイルの保護 (作業マップ)」
- 『Oracle Solaris ZFS 管理ガイド』の第 8 章「ACL による Oracle Solaris ZFS ファイルの保護」
- 148 ページの「nfsmapid デーモン」

UDP と TCP のネゴシエーション

開始時には、トランスポートプロトコルもネゴシエートされます。デフォルトでは、クライアントとサーバーの両方がサポートしているコネクション型トランスポートの中で最初に見つかったものが選択されます。それが見つからない場合は、コネクションレス型トランスポートプロトコルの中で最初に見つかったものが使用されます。システムでサポートされているトランスポートプロトコルのリストは、/etc/netconfig にあります。TCP はコネクション型トランスポートプロトコルで、Solaris 2.6 からサポートされています。UDP はコネクションレス型トランスポートプロトコルです。

NFS プロトコルのバージョンとトランスポートプロトコルが両方ともネゴシエーションによって決まった場合は、NFS プロトコルのバージョンがトランスポートプロトコルよりも優先されます。UDP を使用する NFS version 3 プロトコルの方が、TCP を使用する NFS version 2 プロトコルよりも優先されます。mount コマンドでは NFS プロトコルのバージョンもトランスポートプロトコルも手動で選択できま

す。mount_nfs(1M) のマニュアルページを参照してください。ほとんどの場合、ネゴシエーションによって選択されるオプションの方が適切です。

ファイル転送サイズのネゴシエーション

ファイル転送サイズは、クライアントとサーバーの間でデータを転送するときに使用されるバッファーのサイズです。原則として、ファイル転送サイズが大きいほどパフォーマンスが向上します。NFS version 3 には転送サイズに上限はありませんが、Solaris 2.6 以降がデフォルトで提示するバッファーサイズは 32K バイトです。クライアントは、必要であればマウント時にこれより小さい転送サイズを提示することができますが、ほとんどの場合その必要はありません。

転送サイズは、NFS version 2 を使用しているシステムとはネゴシエートされません。このとき、ファイル転送サイズの上限は 8K バイトに設定されます。

mount コマンドに対して -rsize オプションと -wsize オプションを使用すると、転送サイズを手動で設定できます。PC クライアントの一部では転送サイズを小さくする必要があります。また、NFS サーバーが大きなファイル転送サイズに設定されている場合は、転送サイズを大きくすることができます。

注 – Solaris 10 以降のリリースでは、書き込み転送サイズの制限が緩和されました。使用するトランスポートプロトコルに基づいて転送サイズが決定されるようになりました。たとえば、UDP 使用時の NFS 転送の上限は、以前と同じく 32K バイトです。これに対し、TCP は UDP のようなデータグラム制限を持たないストリーミングプロトコルであるため、TCP 使用時の最大転送サイズは、1M バイトまで拡張されています。

ファイルシステムがどのようにマウントされるか

次の説明は、NFS version 3 のマウントに適用されます。NFS version 4 のマウントプロセスは、ポートマップサービスおよび MOUNT プロトコルを含みません。

クライアントがサーバーからファイルシステムをマウントするとき、クライアントはサーバーからファイルハンドルを取得する必要があります。ファイルハンドルは、そのファイルシステムに対応していなければなりません。そのためには、クライアントとサーバーの間でいくつかのトランザクションが発生します。この例では、クライアントはサーバーから /home/terry をマウントします。snoop によって追跡したトランザクションは、次のとおりです。

```
client -> server PORTMAP C GETPORT prog=100005 (MOUNT) vers=3 proto=UDP
server -> client PORTMAP R GETPORT port=33492
client -> server MOUNT3 C Null
```

```
server -> client MOUNT3 R Null
client -> server MOUNT3 C Mount /export/home9/terry
server -> client MOUNT3 R Mount OK FH=9000 Auth=unix
client -> server PORTMAP C GETPORT prog=100003 (NFS) vers=3 proto=TCP
server -> client PORTMAP R GETPORT port=2049
client -> server NFS C NULL3
server -> client NFS R NULL3
client -> server NFS C FSINFO3 FH=9000
server -> client NFS R FSINFO3 OK
client -> server NFS C GETATTR3 FH=9000
server -> client NFS R GETATTR3 OK
```

この追跡結果では、クライアントがまずマウントポート番号を NFS サーバーの portmap サービスに要求します。クライアントが取得したマウントポート番号 (33492) は、サーバーでサービスが使用可能かどうかをテストするために使用されます。このポート番号でサービスが実行中であることが確認できると、クライアントはマウントを要求します。サーバーはこの要求に応答するときに、マウントするファイルシステムのファイルハンドル (9000) を指定します。これに対してクライアントは、NFS ポート番号を要求します。クライアントはサーバーからポート番号を受け取ると、NFS サービス (nfsd) が使用可能かどうかをテストします。また、そのファイルハンドルを使うファイルシステムに関する NFS 情報を要求します。

次の追跡結果では、クライアントは public オプションを使ってファイルシステムをマウントしています。

```
client -> server NFS C LOOKUP3 FH=0000 /export/home9/terry
server -> client NFS R LOOKUP3 OK FH=9000
client -> server NFS C FSINFO3 FH=9000
server -> client NFS R FSINFO3 OK
client -> server NFS C GETATTR3 FH=9000
server -> client NFS R GETATTR3 OK
```

デフォルトの公開ファイルハンドル (0000) を使用しているために、すべてのトランザクションにポートマップサービスから情報が与えられ、NFS ポート番号を決定するためのトランザクションはありません。

注 – NFS version 4 は、揮発性ファイルハンドルをサポートします。詳細は、186 ページの「NFS version 4 における揮発性ファイルハンドル」を参照してください。

マウント時の -public オプションと NFS URL の意味

-public オプションを使用すると、マウントが失敗することがあります。NFS URL を組み合わせると、状況がさらに複雑になる可能性があります。これらのオプションを使用した場合にファイルシステムがどのようにマウントされるかは、次のとおりです。

public オプションと NFS URL – 公開ファイルハンドルが使用されます。公開ファイルハンドルがサポートされていないと、マウントは失敗します。

public オプションと通常のパス – 公開ファイルハンドルが使用されます。公開ファイルハンドルがサポートされていないと、マウントは失敗します。

NFS URL のみ – NFS サーバーでサポートされていれば、公開ファイルハンドルを使用します。公開ファイルハンドルを使用してマウントが失敗する場合は、MOUNT プロトコルを使ってマウントします。

通常のパスのみ – 公開ファイルハンドルは使用しないでください。MOUNT プロトコルが使用されます。

クライアント側フェイルオーバー機能

クライアント側のフェイルオーバー機能を使用すると、NFS クライアントは同じデータを利用できる複数のサーバーを知ることができるため、現在のサーバーが使用不能になっても、ほかのサーバーに切り替えることができます。ファイルシステムが使用不能になる原因には次のものがあります。

- ファイルシステムが、クラッシュしているサーバーに接続している
- サーバーの過負荷
- ネットワーク障害

通常、このような場合のフェイルオーバー機能はユーザーにはわかりません。つまり、フェイルオーバー機能はクライアント上のプロセスを中断することなく実行されます。

フェイルオーバー機能が行われるためには、ファイルシステムが読み取り専用でマウントされている必要があります。また、ファイルシステムが完全に同じでないとフェイルオーバー機能は成功しません。ファイルシステムが同一になる条件については、198 ページの「複製されたファイルシステムとは」を参照してください。フェイルオーバー機能の候補としては、静的なファイルシステム、または変更の少ないファイルシステムが適しています。

同じ NFS マウント上では、CacheFS 機能とクライアント側のフェイルオーバー機能の両方は使用できません。CacheFS ファイルシステムは、それぞれについて追加情報が

格納されています。この情報はフェイルオーバーの際に更新できないため、ファイルシステムをマウントするときにはフェイルオーバー機能と CacheFS のどちらか片方の機能しか使用できません。

各ファイルシステムについて用意すべき複製の数を決める要素はさまざまです。理想的には、サーバーを 2 台以上設置します。それぞれのサーバーが複数のサブネットをサポートする必要があります。これは、各サブネットに一意のサーバーを設置するよりもよい方法です。フェイルオーバー処理の際にはリストにある各サーバーが確認されます。そのため、サーバーの台数を増やすと、それぞれのマウント処理が遅くなります。

フェイルオーバー機能に関する用語

フェイルオーバー機能のプロセスを完全に理解するには、次の 2 つの用語を理解する必要があります。

- フェイルオーバー – 複製されたファイルシステムをサポートしているサーバーのリストから、1 つのサーバーを選択するプロセス。通常、ソートされたリストの順番を元に、次のサーバーが応答するならばそのサーバーが使用されます。
- 再マッピング – 新しいサーバーを使用すること。クライアントは、正常な状態のときにリモートファイルシステム上のアクティブなファイルのそれぞれのパス名を格納します。再マッピング時には、そのパス名に基づいて新しいサーバー上のファイルを検出します。

複製されたファイルシステムとは

フェイルオーバー機能に関して、あるファイルシステムのすべてのファイルが元のファイルシステムのファイルとサイズもファイルタイプも同じ場合に、そのファイルシステムを「複製」といいます。アクセス権、作成日付などのファイル属性は関係ありません。ファイルサイズまたはファイルタイプが異なると再マッピングは失敗し、元のサーバーが再び使用可能になるまでプロセスはハングアップします。NFS version 4 では、動作が異なります。199 ページの「NFS version 4 におけるクライアント側フェイルオーバー機能」を参照してください。

複製されたファイルシステムを保守するには、`rdist` や `cpio` などのファイル転送メカニズムを使います。複製されたファイルシステムを更新すると不一致が発生するため、最良の結果を得るには次の予防策を考慮してください。

- 新しいバージョンのファイルをインストールするときは、あらかじめ古いバージョンのファイル名を変更する
- クライアントがほとんど使用しない夜間に更新を実行する
- 更新は小規模にとどめる
- コピーの数を最小限にする

フェイルオーバー機能と NFS ロック

ソフトウェアパッケージの一部は、ファイルに読み取りロックをかける必要があります。そのようなソフトウェアが正常に動作できるようにするため、読み取り専用ファイルシステムに対しても読み取りロックがかけられるようになっています。ただし、これはクライアント側でしか認識されません。サーバー側で意識されないため、再マッピングされてもロックはそのまま残ります。ファイルはもともと変更が許されないので、サーバー側でファイルをロックする必要はありません。

NFS version 4 におけるクライアント側フェイルオーバー機能

NFS version 4 では、ファイルサイズが違うまたはファイルタイプが同じでないために複製が確立されない場合、次のことが起こります。

- ファイルが使用不能とマークされる。
- 警告が出力される。
- アプリケーションがシステムコールの失敗を受け取る。

注 – アプリケーションを再起動して、ファイルに再度アクセスすると、正常にアクセスできます。

NFS version 4 では、サイズが異なるディレクトリの複製エラーを受け取ることはありません。以前のバージョンの NFS では、この状態はエラーとして扱われ、再マッピングプロセスを妨げました。

さらに、NFS version 4 では、ディレクトリ読み取り操作が正常に行われない場合、次に一覧表示されたサーバーによって操作が行われます。以前のバージョンの NFS では、正常でない読み取り操作により、再マッピングが失敗し、プロセスは元のサーバーが使用可能になるまで停止しました。

大規模ファイル

Solaris 2.6 以降、Solaris OS は 2G バイトを超えるファイルをサポートします。デフォルトでは、UFS ファイルシステムはこの新機能をサポートするために `-largefiles` オプション付きでマウントされます。以前のリリースでは、2G バイトを超えるファイルは扱えません。具体的な方法については、91 ページの「NFS サーバー上で大規模ファイルを無効にする方法」を参照してください。

`-largefiles` オプションを使ってサーバー上のファイルシステムをマウントする場合、大規模ファイルにアクセスするために Solaris 2.6 NFS クライアントを変更する必要はありません。ただし、Solaris 2.6 のコマンドすべてで大規模ファイルを扱えるわけではありません。大規模ファイルを扱えるコマンドについては、`largefile(5)` のマニュアルページを参照してください。大規模ファイル用機能拡張を備えた NFS

version 3 プロトコルをサポートしていないクライアントは、大規模ファイルには一切アクセスできません。Solaris 2.5 クライアントでは、NFS version 3 プロトコルを使用することはできますが、大規模ファイルを扱う機能は含まれていません。

NFS サーバーログ機能のしくみ

NFS サーバーログ機能は NFS の読み取りと書き込み、およびこのファイルシステムを変更する操作の記録を提供します。このデータは情報へのアクセスを追跡するのに利用できます。さらに、この記録は、情報へのアクセスを測定する定量的な方法を提供します。

ログ機能が有効になっているファイルシステムにアクセスすると、カーネルが raw データをバッファーファイルに書き込みます。このデータには、次の内容が含まれています。

- タイムスタンプ
- クライアントの IP アドレス
- 要求者の UID
- アクセスされているファイルまたはディレクトリオブジェクトのファイルハンドル
- 発生した処理のタイプ

`nfslogd` デーモンはこの raw データを、ログファイルに保存される ASCII レコードに変換します。使用可能なネームサービス機能が一致しているものを見つけると、その変換中に IP アドレスはホスト名に変更され、UID はログインに変更されます。ファイルハンドルはパス名にも変換されます。デーモンはファイルハンドルを追跡し、情報を別のファイルハンドルパステーブルに保存して、変換を完了します。このようにすると、ファイルハンドルにアクセスされるたびに、パスを識別し直す必要がなくなります。`nfslogd` をオフにするとファイルハンドルパステーブルのマッピングが変更されなくなるため、デーモンは常に実行させておく必要があります。

注 – サーバーロギングは NFS version 4 ではサポートされません。

WebNFS サービスのしくみ

WebNFS サービスとは、あるディレクトリに置かれたファイルを、公開ファイルハンドルを使ってクライアントからアクセスできるようにするものです。ファイルハンドルは、NFS クライアントがファイルを識別できるようにカーネルが生成するアドレスです。公開ファイルハンドルの値はあらかじめ決まっているため、サーバーがクライアントに対してファイルハンドルを生成する必要はありませ

ん。定義済みのファイルハンドルを使用するというこの機能によって、MOUNT プロトコルが不要になってネットワークトラフィックが減り、クライアントにとってはプロセスが高速化します。

デフォルトでは、NFS サーバーの公開ファイルハンドルはルートファイルシステムに対して設定されます。このデフォルトのため、サーバーに対してマウント権限を持っているすべてのクライアントに対して WebNFS アクセス権が与えられます。公開ファイルハンドルは、share コマンドによって任意のファイルシステムに切り替えることができます。

あるファイルシステムに対するファイルハンドルをクライアントが持っているとき、アクセスするファイルに対応するファイルハンドルを知るには LOOKUP を実行します。NFS プロトコルでは、パス名の構成要素を 1 度に 1 つしか評価できません。したがって、ディレクトリ階層のレベルが 1 つ増えるたびに 1 回ずつ LOOKUP を実行します。公開ファイルハンドルからの相対パスに対して LOOKUP を実行する場合には、WebNFS サーバーはマルチコンポーネントルックアップによって 1 度にパス名全体を評価できます。マルチコンポーネントルックアップにより、WebNFS サーバーはパス名の中のディレクトリレベルを 1 つずつファイルハンドルに変換しなくても目的のファイルに対するファイルハンドルを配信できます。

また、NFS クライアントは、単一の TCP 接続を介して、複数のファイルを同時にダウンロードすることができます。このようにして接続すると、サーバーに複数の接続を設定することによる負荷をかけることなく、すばやくアクセスすることができます。Web ブラウザアプリケーションも複数ファイルを同時にダウンロードできますが、それぞれのファイルに独自の接続が確立されます。WebNFS ソフトウェアは接続を 1 つしか使用しないため、サーバーに対するオーバーヘッドを軽減できます。

パス名の中の最後の構成要素が他のファイルシステムに対するシンボリックリンクである場合、通常の NFS アクティビティーによってあらかじめそのファイルへのアクセス権を持っていれば、クライアントはそのファイルにアクセスできます。

通常、NFS URL は公開ファイルハンドルからの相対位置として評価されます。パスの先頭にスラッシュを 1 つ追加すると、サーバーのルートファイルシステムからの相対位置に変更できます。次の例では、公開ファイルハンドルが /export/ftp ファイルシステムに設定されていればこの 2 つの NFS URL は同等です。

```
nfs://server/junk
nfs://server//export/ftp/junk
```

注 – NFS version 4 プロトコルは、WebNFS サービスに優先します。NFS version 4 は、MOUNT プロトコルと WebNFS サービスに追加されたすべてのセキュリティーネゴシエーションを完全に統合します。

WebNFS セキュリティーネゴシエーション機能のしくみ

Solaris 8 リリースから、WebNFS クライアントが WebNFS サーバーと、選択されたセキュリティーメカニズムについてネゴシエーションできるようにする新しいプロトコルがあります。この新しいプロトコルは、セキュリティーネゴシエーションマルチコンポーネントルックアップを使用しています。これは、WebNFS プロトコルの以前のバージョンで使用されていたマルチコンポーネントルックアップの拡張版です。

WebNFS クライアントは、公開ファイルハンドルを使って通常のマルチコンポーネントルックアップ要求を行うことにより、このプロセスを開始します。このクライアントには、サーバーがどのようにしてこのパスを保護しているかについての知識がないため、デフォルトのセキュリティーメカニズムが使用されます。デフォルトのセキュリティーメカニズムでは不十分な場合は、サーバーは `AUTH_TOOWEAK` エラーを返します。このメッセージは、そのデフォルトメカニズムが有効ではなく、クライアントはより強力なメカニズムを使用する必要があることを意味しています。

クライアントは、`AUTH_TOOWEAK` エラーを受信すると、サーバーに対してどのセキュリティーメカニズムが必要か決定するように要求します。この要求が成功すると、サーバーは、指定されたパスに必要なセキュリティーメカニズムの配列を返します。このセキュリティーメカニズムの配列のサイズによっては、クライアントは完全な配列を得るためにさらに要求を出さなければならない場合があります。サーバーが WebNFS セキュリティーネゴシエーションをサポートしていない場合は、この要求は失敗します。

要求が成功すると、WebNFS クライアントは、クライアントがサポートしている最初のセキュリティーメカニズムを配列から選択します。その後、クライアントは、選択したセキュリティーメカニズムを使用して、通常のマルチコンポーネントルックアップ要求を発行し、ファイルハンドルを獲得します。この後に続くすべての NFS 要求は、選択されたセキュリティーメカニズムとファイルハンドルを使って出されます。

注 – NFS version 4 プロトコルは、WebNFS サービスに優先します。NFS version 4 は、MOUNT プロトコルと WebNFS サービスに追加されたすべてのセキュリティーネゴシエーションを完全に統合します。

Web ブラウザの使用と比較した場合の WebNFS の制約

HTTP を使用する Web サイトで実現可能な機能のいくつかは、WebNFS ではサポートされていません。この違いは、NFS サーバーはファイルを送るだけであるため、特別な処理はすべてクライアントで行う必要があることが原因です。ある Web サイトを WebNFS と HTTP 両方のアクセスに対応させるには、次を考慮してください。

- NFS によるブラウズでは CGI スクリプトは実行されません。したがって、CGI スクリプトを多用している Web サイトを含むファイルシステムは、NFS によるブラウズに適していない可能性があります。
- ブラウザからは、形式の異なるファイルを扱うために別のビューアが起動されることがあります。NFS URL からそうしたファイルにアクセスすると、ファイル名からファイルタイプが判別できるならば外部のビューアが起動されます。ブラウザは、NFS URL が使用されている場合、標準の MIME タイプで決まっているファイル名拡張子をすべて認識します。WebNFS ソフトウェアは、ファイルの内容からファイルタイプを判別しません。したがって、ファイルタイプはファイル名の拡張子だけから判別されます。
- NFS によるブラウズでは、サーバー側のイメージマップ (クリック可能なイメージ) は使用できません。ただし、クライアント側のイメージマップ (クリック可能なイメージ) は、場所とともに URL が定義されているため使用できます。文書サーバーからの応答は不要です。

Secure NFS システム

NFS 環境は、アーキテクチャーやオペレーティングシステムの異なるコンピュータから構成されるネットワーク上でファイルシステムを共有するために、有力で使いやすい手段です。しかし、NFS の操作によるファイルシステムの共有を便利にする機能が、一方ではセキュリティー上の問題につながっています。今まで、NFS はほとんどのバージョンで UNIX (AUTH_SYS) 認証を使用してきましたが、現在では AUTH_DH のようなより強力な認証方式も使用可能です。UNIX 認証を使用している場合、NFS サーバーは、要求をしたユーザーではなくコンピュータを認証して、ファイル要求を認証します。そのため、クライアントユーザーは、su を実行してファイルの所有者を装ったりすることができます。DH 認証では、NFS サーバーはユーザーを認証するため、このような操作が困難になります。

スーパーユーザーのアクセス権とネットワークプログラミングについての知識があれば、だれでも任意のデータをネットワークに取り入れたり、ネットワークから取り出したりできます。ネットワークに対するもっとも危険な攻撃は、データをネットワークに持ち込むような攻撃です。たとえば、有効なパケットを生成したり、または「対話」を記録し後で再生することによってユーザーを装うなどの手段

があります。これらはデータの整合性に影響を与えます。ユーザーを装わず、単にネットワークトラフィックを傍受するための盗聴が行われる攻撃であれば、データの整合性が損なわれることはないため、それほど危険ではありません。ネットワーク上でやりとりされるデータを暗号化すると、機密情報のプライバシを保護できます。

ネットワークのセキュリティー問題に対する共通の対処方法は、解決策を各アプリケーションにゆだねることです。さらに優れた手法としては、すべてのアプリケーションを対象として、標準の認証システムを導入することです。

Solaris オペレーティングシステムには、NFS の操作が構築されるメカニズムである遠隔手続き呼び出し (RPC) のレベルで、認証システムが組み込まれています。このシステムは Secure RPC と呼ばれ、ネットワーク環境のセキュリティーを大幅に向上させるとともに、NFS のセキュリティーを強化します。Secure RPC の機能を利用した NFS システムを Secure NFS システムといいます。

Secure RPC

Secure RPC は Secure NFS システムの基本となるメカニズムです。Secure RPC の目標は、少なくともタイムシェアリングシステム程度に安全なシステムを構築することです。タイムシェアリングシステムでは、すべてのユーザーが 1 台のコンピュータを共有します。タイムシェアリングシステムはログインパスワードによりユーザーを認証します。データ暗号化規格 (DES) 認証でも、同じ認証処理が実行されます。ユーザーは、ローカル端末の場合と同じように、任意のリモートコンピュータにログインできます。ユーザーのログインパスワードは、ネットワークセキュリティーへの保証です。タイムシェアリングでは、システム管理者は信頼のおける人で、パスワードを変更してだれかを装うようなことはしないという道徳上の義務を負います。Secure RPC では、ネットワーク管理者は「公開鍵」を格納するデータベースのエントリを変更しないという前提で信頼されています。

RPC 認証システムを理解するには、「資格 (credential)」と「ベリファイア」という 2 つの用語を理解する必要があります。ID バッジを例にとれば、資格とは、名前、住所、誕生日など個人を識別するものです。ベリファイアとはバッジに添付された写真です。バッジの写真をその所持者と照合することによって、そのバッジが盗まれたものではないことを確認できます。RPC では、クライアントプロセスは RPC 要求のたびに資格とベリファイアの両方をサーバーに送信します。クライアントはサーバーの資格をすでに知っているため、サーバーはベリファイアだけを送り返します。

RPC の認証機能は拡張が可能で、UNIX、DH、および KERB などのさまざまな認証システムを組み込むことができます。

ネットワークサービスで UNIX 認証を使用する場合、資格にはクライアントのホスト名、UID、GID、グループアクセスリストが含まれ、ベリファイアには何も含まれ

ません。ベリファイアが存在しないため、root ユーザーは su などのコマンドを使用して、適切な資格を偽ることができます。UNIX 認証でのもう 1 つの問題は、ネットワーク上のすべてのコンピュータを UNIX コンピュータと想定していることです。UNIX 認証を異機種ネットワーク内の他のオペレーティングシステムに適用した場合、これは正常に動作しません。

UNIX 認証の問題を克服するために、Secure RPC では DH 認証を使用します。

DH 認証

DH 認証は、Data Encryption Standard (DES) と Diffie-Hellman 公開鍵暗号手法を使ってネットワーク上のユーザーとコンピュータの両方を認証します。DES は、標準の暗号化メカニズムです。Diffie-Hellman 公開鍵暗号手法は、2 つの鍵、つまり公開鍵と秘密鍵を持つ暗号方式です。 公開鍵と秘密鍵は名前空間に格納されます。NIS では、これらのキーは public-key マップに保存されています。これらのマップにはすべての認証の候補ユーザーの公開鍵と秘密鍵が入っています。このマップの設定方法については、『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』を参照してください。

DH 認証のセキュリティーは、送信側が現在時刻を暗号化する機能に基づいていて、受信側はこれを復号化して、自分の時刻と照合します。タイムスタンプは DES を使用して暗号化されます。この方式が機能するには次の条件が必要です。

- 2 つのエージェントの現在時刻が一致している。
- 送信側と受信側が同じ暗号化鍵を使用する。

ネットワークが時間同期プログラムを実行する場合、クライアントとサーバー上の時間は自動的に同期がとられます。時間同期プログラムを使用できない場合、ネットワーク時間ではなく、サーバーの時間を使ってタイムスタンプを計算できます。クライアントは、RPC セッションを開始する前にサーバーに時間を要求し、自分のクロックとサーバーのクロックとの時間差を計算します。タイムスタンプを計算するときには、この差を使ってクライアントのクロックを補正します。クライアントとサーバーのクロックが同期していないと、サーバーはクライアントの要求を拒否します。その場合、クライアントの DH 認証システムはサーバーとの間で再び同期をとります。

クライアントとサーバーは、ランダムな「対話鍵」(「セッション鍵」とも呼ばれる)を生成したあと公開鍵暗号方式を使って「共通鍵」を推理することによって、同一の暗号化鍵に到達します。この共通鍵は、クライアントとサーバーだけが推理できる鍵です。対話鍵は、クライアントのタイムスタンプを暗号化および復号化するために使用されます。共通鍵は、この対話鍵を暗号化および復号化するために使用されます。

KERB 認証

Kerberos は、マサチューセッツ工科大学 (MIT) で開発された認証システムです。Kerberos は、DES を含むさまざまな暗号化タイプを提供します。Kerberos サポートは Secure RPC の一部としては提供されなくなりましたが、Solaris 9 以降のリリースでは、サーバー側とクライアント側に実装されています。Kerberos 認証の実装に関する詳細については、『Solaris のシステム管理 (セキュリティサービス)』の第 21 章「Kerberos サービスについて」を参照してください。

NFS での Secure RPC の使用

Secure RPC を使用する場合は、次の点に注意してください。

- サーバーがクラッシュしたとき周囲にだれもいない場合 (停電のあとなど) には、システムに格納されていた秘密鍵はすべて削除されます。そのためどのプロセスからも、セキュリティー保護されたネットワークサービスにアクセスしたり NFS ファイルシステムをマウントしたりできません。リブート中の重要な処理は、通常 `root` として実行されます。そのため、root の秘密鍵を別に保存していればこれらのプロセスを実行できますが、その秘密鍵を復号化するパスワードを入力することはできません。`keylogin -r` を使用すると `root` の秘密鍵がそのまま `/etc/.rootkey` に格納され、`keyserv` がそれを読み取ります。
- システムによっては、シングルユーザーモードで起動し、コンソールには root のログインシェルが表示されてパスワードの入力が要求されないことがあります。このような場合は、物理的なセキュリティーが不可欠です。
- ディスクレスコンピュータのブートは、完全に安全とはいえません。ブートサーバーになりすましてリモートコンピュータに対する秘密鍵の入力を記録するような、不正なカーネルをだれかがブートすることが考えられます。Secure NFS システムによって保護されているのはカーネルとキーサーバーが起動した後だけです。そうでないと、ブートサーバーからの応答を認証することができません。このような制限は重大な問題につながる可能性がありますが、この部分を攻撃するにはカーネルのソースコードを使用した高度な技術が必要です。また、不法行為の痕跡が残ります。つまり、ネットワークを通じてブートサーバーにポーリングすれば、不正なブートサーバーの場所がわかります。
- 多くの setuid プログラムは `root` が所有者です。`root` の秘密鍵が `/etc/.rootkey` に格納されていれば、これらのプログラムは正常に動作します。しかし、ユーザーが所有者である setuid プログラムは動作しない可能性があります。たとえば、ある setuid プログラムの所有者が `dave` であり、ブート後 `dave` が 1 度もログインしていないとします。このプログラムはセキュリティー保護されたネットワークサービスにはアクセスできません。
- リモートコンピュータに (`login`、`rlogin`、または `telnet` を使用して) ログインし、`keylogin` を使ってアクセスすると自分のアカウントへのアクセスを許したことになります。これは、秘密鍵が相手側のコンピュータのキーサーバーに渡され、キーサーバーがその秘密鍵を格納したためです。このプロセスが問題になる

のは、相手側のリモートコンピュータを信用できない場合だけです。しかし、疑いがある場合は、パスワードを要求するリモートコンピュータにはログインしないでください。代わりに NFS 環境を使用して、そのリモートコンピュータから共有されているファイルシステムをマウントします。または、keylogout を使ってキーサーバーから秘密鍵を消去します。

- ホームディレクトリが共有されていて -o sec=dh 指定されていると、リモートログインによって問題が生じる可能性があります。/etc/hosts.equiv ファイルまたは ~/.rhosts ファイルに、パスワードを要求するように設定されていない場合は、ログインが成功します。ただし、ローカルで認証されていないため、ユーザーは自分のホームディレクトリにアクセスできません。パスワードを要求され、入力したパスワードがネットワークパスワードと一致すれば、自分のホームディレクトリにアクセスできます。

autofs マップ

autofs は 3 種類のマップを使用します。

- マスターマップ
- 直接マップ
- 間接マップ

autofs マスターマップ

auto_master マップでは、ディレクトリからマップへの関連付けを行います。このマップは、すべてのマップを指定するマスターリストであり、autofs が参照します。auto_master ファイルの内容の例を次に示します。

例 6–3 /etc/auto_master ファイルの例

```
# Master map for automounter
#
+auto_master
/net            -hosts          -nosuid,nobrowse
/home           auto_home       -nobrowse
/-              auto_direct     -ro
```

この例では、汎用の auto_master ファイルに auto_direct マップのための追加が行われています。マスターマップ /etc/auto_master の各行は、次の構文に従っています。

mount-point map-name [*mount-options*]

mount-point *mount-point* は、ディレクトリのフル (絶対) パス名です。このディレクトリが存在しない場合、可能ならば autofs はこのディレクトリを作成します。このディレクトリが存在し、しかも空ではない場合、マウントすることによってその内容が隠されます。この場合、autofs は警告を出します。

マウントポイントとして `/-` を指定すると、この特定のマップが直接マップであり、マップに関連付けられている特定のマウントポイントがないことを表します。

map-name *map-name* 名は、位置に対する指示またはマウント情報を検出するために、autofs が使用するマップです。この名前がスラッシュ (/) で始まる場合、autofs はこの名前をローカルファイルとして解釈します。そうでない場合、autofs はネームサービススイッチ構成ファイル (`/etc/nsswitch.conf`) で指定される検索によりマウント情報を検索します。また、`/net` には、特別なマップを使用します。詳細は、209 ページの「`/net` マウントポイント」を参照してください。

mount-options *mount-options* は省略できます。map-name のエントリにほかのオプションがある場合を除き、map-name で指定されたエントリのマウントに適用されるオプションをコンマで区切って並べます。特定のファイルシステムのマウントオプションについては、各ファイルシステムについてのマニュアルページを参照してください。たとえば、NFS に固有のマウントオプションについては、`mount_nfs(1M)` のマニュアルページを参照してください。NFS 固有のマウントポイントの場合、`bg` (バックグラウンド) オプションと `fg` (フォアグラウンド) オプションは適用されません。

で始まる行はコメント行です。その行のテキストの最後まですべて無視されます。

長い行を短い行に分割するには、行末にバックスラッシュ (\) を入力します。入力できる文字数の上限は 1024 です。

注 – 2 つのエントリで同じマウントポイントが使用されるときは、1 番目のエントリは `automount` コマンドが使用します。2 番目のエントリは無視されます。

`/home` マウントポイント

`/home` マウントポイントは、`/etc/auto_home` (間接マップ) に記述されたエントリがマウントされるディレクトリです。

注 – autofs はすべてのコンピュータで動作し、デフォルトでは /net と /home (自動マウントされるホームディレクトリ) をサポートします。このデフォルトは、NIS ならば auto.master マップ、NIS+ ならば auto_master テーブルを使用して、またはローカルの /etc/auto_master ファイルを編集することによって変更できます。

/net マウントポイント

autofs は、特別なマップ -hosts 内の全エントリを /net ディレクトリの下にマウントします。これは hosts データベースだけを使用する組み込みマップです。たとえば、hosts データベースにあるコンピュータ gumbo が、ファイルシステムのどれかをエクスポートするとします。次のコマンドを入力すると、現在のディレクトリがコンピュータ gumbo のルートディレクトリに変更されます。

```
% cd /net/gumbo
```

なお、autofs はホスト gumbo のエクスポートされたファイルシステムだけをマウントできます。つまり、ローカルディスク上のファイルシステムではなく、ネットワークユーザーが使用できるサーバー上のファイルシステムです。したがって、gumbo にあるすべてのファイルとディレクトリは、/net/gumbo では利用できない場合があります。

/net を使用したアクセスでは、サーバー名はパスの中に指定されるため、位置に依存します。したがって、エクスポートされるファイルシステムを別のサーバーに移動すると、そのパスは使用できなくなります。このような場合は /net を使用しないで、そのファイルシステムに対応するエントリをマップの中に設定します。

注 – autofs はマウント時だけサーバーのエクスポートリストを調べます。サーバーのファイルシステムが一度マウントされると、そのファイルシステムがアンマウントされ、次にマウントされるまで autofs はそのサーバーをチェックしません。したがって、新たにエクスポートされたファイルシステムは、それがサーバーからアンマウントされ、再度マウントされるまでは見えません。

直接マップ

直接マップは自動マウントポイントです。つまり、直接マップによって、クライアント上のマウントポイントとサーバー上のディレクトリが直接対応付けられます。直接マップにはフルパス名があり、明示的に関係を示します。次に一般的な /etc/auto_direct マップを示します。

```
/usr/local          -ro \
   /bin                    ivy:/export/local/sun4 \
   /share                  ivy:/export/local/share \
```

```
   /src                    ivy:/export/local/src
/usr/man            -ro    oak:/usr/man \
                           rose:/usr/man \
                           willow:/usr/man
/usr/games          -ro    peach:/usr/games
/usr/spool/news     -ro    pine:/usr/spool/news \
                           willow:/var/spool/news
```

直接マップの行は、次の構文に従っています。

key [*mount-options*] *location*

key *key* は直接マップでのマウントポイントのパス名です。

mount-options *mount-options* は、このマウントに適用するオプションです。これらのオプションが必要なのは、マップのデフォルトと異なる場合だけです。特定のファイルシステムのマウントオプションについては、各ファイルシステムについてのマニュアルページを参照してください。たとえば、CacheFS に固有のマウントオプションについては、`mount_cachefs(1M)` のマニュアルページを参照してください。異なるバージョンの NFS での CacheFS オプションの使用については、112 ページの「CacheFS を使用して NFS ファイルシステムにアクセスする」を参照してください。

location *location* はファイルシステムの位置を示します。1 つまたは複数のファイルシステムを *server*:*pathname* (NFS ファイルシステムの場合)、または :*devicename* (High Sierra ファイルシステム (HSFS) の場合) で指定します。

注 – *pathname* に自動マウントされたマウントポイントを含めることはできません。*pathname* は、ファイルシステムの実際の絶対パスにするようにしてください。たとえば、ホームディレクトリの位置は、*server*:`/home/`*username* ではなく、*server*:`/export/home/`*username* として表示する必要があります。

マスターマップと同様、`#` で始まる行はコメントです。その行のテキストの最後まですべて無視されます。長い行を短い行に分割するには、行の最後にバックスラッシュを入力します。

すべてのマップにおいて、直接マップ内のエントリは、`/etc/vfstab` 内の対応するエントリにもっともよく似ています。`/etc/vfstab` のエントリは、次のようになっているとします。

```
dancer:/usr/local - /usr/local/tmp nfs - yes ro
```

直接マップ内では、同じエントリが次のようになります。

```
/usr/local/tmp     -ro     dancer:/usr/local
```

注 – オートマウンタマップの間では、オプションの連結はされません。オートマウンタマップに追加されたどのオプションも、前に検索されたマップに表示されているすべてのオプションを上書きします。たとえば、auto_master マップに指定されているオプションは、他のマップの中の対応するエントリによって上書きされます。

この種類のマップについては、これ以外にも重要な機能があります。218 ページの「autofs がクライアント用のもっとも近い読み取り専用ファイルを選択する方法 (複数ロケーション)」を参照してください。

/– マウントポイント

例 6–3 にある /- というマウントポイントは、auto_direct の中のエントリを具体的なマウントポイントに関連付けないように autofs に指示します。間接マップの場合は、auto_master ファイルに定義されたマウントポイントを使います。直接マップの場合は、名前付きマップ内で指定したマウントポイントを使用します。直接マップ内では、キー、つまりマウントポイントはフルパス名であることに注意してください。

NIS または NIS+ の auto_master ファイルには、直接マップのエントリは 1 つしか存在できません。マウントポイントは 1 つの名前空間の中で一意でなければならないためです。auto_master がローカルファイルならば、重複しないかぎり直接マップのエントリがいくつあってもかまいません。

間接マップ

間接マップは、キーの置換値を使ってクライアント上のマウントポイントとサーバー上のディレクトリとを対応させます。間接マップは、ホームディレクトリなどの特定のファイルシステムをアクセスするのに便利です。auto_home マップは間接マップの一例です。

間接マップ内の行は次の一般的な構文になります。

key [*mount-options*] *location*

key — *key* は間接マップでの単純名 (スラッシュなし) です。

mount-options — *mount-options* は、このマウントに適用するオプションです。これらのオプションが必要なのは、マップのデフォルトと異なる場合だけです。特定のファイルシステムのマウントオプションについては、各ファイルシステムについてのマニュアルページを参照してく

ください。たとえば、NFS に固有のマウントオプションについては、mount_nfs(1M) のマニュアルページを参照してください。

location *location* はファイルシステムの位置を示します。1 つまたは複数のファイルシステムを *server*:*pathname* で指定します。

注 – *pathname* に自動マウントされたマウントポイントを含めることはできません。*pathname* は、ファイルシステムの実際の絶対パスにするようにしてください。たとえば、ディレクトリの位置は、*server*:/net/*server*/usr/local ではなく、*server* :/usr/local として指定する必要があります。

マスターマップと同様、# で始まる行はコメントです。その行のテキストの最後まですべて無視されます。長い行を短い行に分割するには、行の最後にバックスラッシュ (\) を入力します。例 6–3 に、次のエントリを含む auto_master マップを示します。

```
/home      auto_home        -nobrowse
```

auto_home は、/home のもとでマウントされるエントリを含む間接マップの名前です。通常、auto_home マップには、次のパスが含まれています。

```
david                   willow:/export/home/david
rob                     cypress:/export/home/rob
gordon                  poplar:/export/home/gordon
rajan                   pine:/export/home/rajan
tammy                   apple:/export/home/tammy
jim                     ivy:/export/home/jim
linda    -rw,nosuid     peach:/export/home/linda
```

例として、前のマップがホスト oak にあると想定します。パスワードデータベースに、ユーザー linda のホームディレクトリが /home/linda であることを示すエントリがあるとします。linda がコンピュータ oak にログインするたびに、autofs は、コンピュータ peach にあるディレクトリ /export/home/linda をマウントします。彼女のホームディレクトリは、読み書き可能な nosuid にマウントされます。

次のような状況が発生したと想定してください。 ユーザー linda のホームディレクトリがパスワードデータベースに、/home/linda として表示されます。Linda も含めだれでも、前の例のマップを参照するマスターマップで設定されたどのコンピュータからでも、このパスにアクセスできます。

こうした状況のもとでは、ユーザー linda はこれらのどのコンピュータでも login や rlogin を実行し、代わりに彼女用のホームディレクトリをマウントさせることができます。

さらに、これで linda は次のコマンドも入力できます。

```
% cd ~david
```

autofs は彼女のために David のホームディレクトリをマウントします (すべてのアクセス権で許可されている場合)。

注 – オートマウンタマップの間では、オプションの連結はされません。オートマウンタマップに追加されたどのオプションも、前に検索されたマップに表示されているすべてのオプションを上書きします。たとえば、auto_master マップに含まれているオプションは、他のいずれかのマップの対応するエントリによって上書きされます。

ネームサービスのないネットワークで、Linda が自分のファイルにアクセスするには、ネットワーク上のすべてのシステムで、すべての関連ファイル (/etc/passwd など) を変更する必要があります。NIS では、NIS マスターサーバーで変更を行い、関連するデータベースをスレーブのデータベースに伝達します。NIS+ を稼働中のネットワークでは、変更後に関連データベースがスレーブサーバーに自動的に伝達されます。

autofs のしくみ

autofs は、自動的に適切なファイルシステムをマウントするためのクライアント側のサービスです。自動マウントを行うのに、次のコンポーネントが相互に動作します。

- automount コマンド
- autofs ファイルシステム
- automountd デーモン

自動マウントサービス svc:/system/filesystem/autofs は、システムの起動時に呼び出され、マスターマップファイル auto_master を読み取って、autofs マウントの最初のセットを作成します。これらの autofs のマウントは起動時に自動的にはマウントされません。後でファイルシステムがマウントされるポイントです。このようなポイントをトリガーノードと呼ぶこともあります。

autofs マウントが設定されると、要求があったときにファイルシステムをマウントすることができます。たとえば、autofs が、現在マウントされていないファイルシステムをアクセスする要求を受け取ると、automountd を呼び出して要求されたファイルシステムを実際にマウントさせます。

最初に autofs マウントをマウントしたあとは、必要に応じて automount コマンドを実行し、autofs マウントを更新します。このコマンドは、auto_master マップにあるマウントのリストと、マウントテーブルファイル /etc/mnttab (前のバージョンでは /etc/mtab) にあるマウントされたファイルシステムのリストを比較します。その

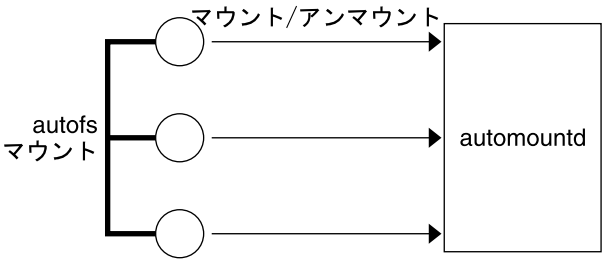
後、automount によって、適切な変更が加えられます。このプロセスにより、システム管理者は auto_master 内のマウント情報を変更し、autofs デーモンを停止したり再起動したりすることなく、それらの変更結果を autofs プロセスに使用させることができます。ファイルシステムがマウントされれば、以後のアクセスに automountd は不要になります。次に automountd が必要になるのは、ファイルシステムが自動的にアンマウントされたときです。

mount とは異なり、automount はマウントすべきファイルシステムを調べるために /etc/vfstab ファイル (各コンピュータごとに異なる) を参照しません。automount コマンドは、ドメイン内とコンピュータ上で名前空間とローカルファイルを通して制御されます。

次の図では、autofs のしくみの概要を簡単に説明します。

自動マウントデーモンである automountd は、ブート時にサービス svc:/system/filesystem/autofs によって起動されます。図 6–3 を参照してください。このサービスは automount コマンドも実行します。このコマンドはマスターマップを読み取り、autofs のマウントポイントをインストールします。詳細は、215 ページの「autofs のナビゲーションプロセス開始法 (マスターマップ)」を参照してください。

図 6–3　svc:/system/filesystem/autofs サービスによる automount の起動



autofs は、自動マウント操作とアンマウント操作をサポートするカーネルファイルシステムの 1 つです。

autofs マウントポイントで、ファイルシステムへのアクセスが要求された場合は、次の動作が行われます。

1. autofs がその要求に介入します。
2. autofs は要求されたファイルシステムをマウントするよう、automountd にメッセージを送信します。
3. automountd がマップからファイルシステム情報を見つけ、マウントを実行します。

4. autofs は、介入した要求の実行を続行させます。
5. 一定時間そのファイルシステムがアクセスされないと、autofs はそのファイルシステムをアンマウントします。

注 – autofs サービスによって管理されるマウントは、手動でマウントまたはアンマウントは行わないでください。たとえこの操作がうまくいったとしても、autofs サービスはオブジェクトがアンマウントされたことを認識しないので、一貫性が損なわれる恐れがあります。リブートによって、autofs のマウントポイントがすべて消去されます。

---

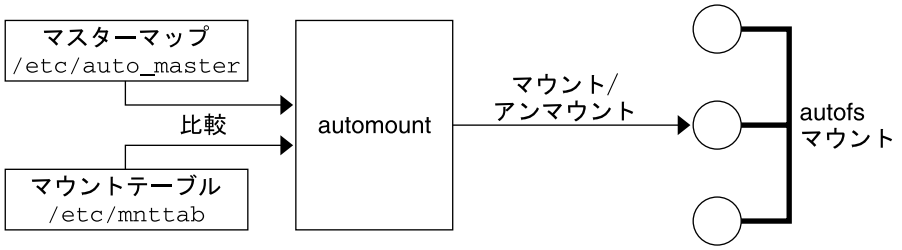
autofs のネットワークナビゲート (マップ)

autofs は一連のマップを探索することによって、ネットワークをナビゲートします。マップは、ネットワーク上の全ユーザーのパスワードエントリや、ネットワーク上の全ホストコンピュータの名前などの情報を含むファイルです。マップには UNIX の管理ファイルに相当するネットワーク規模の管理ファイルも含まれています。マップはローカルに使用するか、あるいは NIS や NIS+ のようなネットワークネームサービスを通じて使用できます。ユーザーは Solaris 管理コンソールツールを使用して、自分の環境ニーズに適合するマップを作成します。224 ページの「autofs のネットワークナビゲート法の変更 (マップの変更)」を参照してください。

autofs のナビゲーションプロセス開始法 (マスターマップ)

`automount` コマンドはシステムの起動時にマスターマップを読み取ります。図 6–4 に示すように、マスターマップ内の各エントリは、直接または間接のマップ名、そのパス、およびそのマウントオプションです。エントリの順序は重要ではありません。`automount` は、マスターマップ内のエントリとマウントテーブル内のエントリを比較して、現在のリストを生成します。

図 6–4 マスターマップによるナビゲーション



autofs マウントプロセス

マウント要求が発生したときに autofs サービスが何を実行するかは、オートマウンタマップの設定によって異なります。マウントプロセスの基本はすべてのマウントで同じですが、指定されているマウントポイントとマップの複雑さによって結果が変わります。Solaris 2.6 ではマウントプロセスも変更され、トリガーノードも作成されるようになりました。

単純な autofs マウント

autofs マウントプロセスの説明のために、次のファイルがインストールされていると仮定します。

```
$ cat /etc/auto_master
# Master map for automounter
#
+auto_master
/net        -hosts        -nosuid,nobrowse
/home       auto_home     -nobrowse
/share      auto_share
$ cat /etc/auto_share
# share directory map for automounter
#
ws          gumbo:/export/share/ws
```

/share ディレクトリがアクセスされると、autofs サービスは /share/ws に対するトリガーノードを作成します。これは、/etc/mnttab の中では次のようなエントリになります。

```
-hosts  /share/ws     autofs  nosuid,nobrowse,ignore,nest,dev=###
```

/share/ws ディレクトリがアクセスされると、autofs サービスは次の手順を実行します。

1. サーバーのマウントサービスが使用可能かどうかを確認します。
2. 要求されたファイルシステムを、/share の下にマウントします。これで、/etc/mnttab ファイルには次のエントリが追加されます。

```
-hosts  /share/ws     autofs  nosuid,nobrowse,ignore,nest,dev=###
gumbo:/export/share/ws /share/ws   nfs   nosuid,dev=####    #####
```

階層型マウント

オートマウンタファイルに複数の層が定義されていると、マウントプロセスはさらに複雑になります。前の例の /etc/auto_shared ファイルを拡張して、次の行を追加したとします。

```
# share directory map for automounter
#
ws        /       gumbo:/export/share/ws
          /usr    gumbo:/export/share/ws/usr
```

この場合、/share/ws マウントポイントがアクセスされたときのマウントプロセスは基本的に最初の例と同じです。また、/share/ws ファイルシステムの中に次のレベル (/usr) へのトリガーノードを作成することにより、そのレベルがアクセスされたときにマウントできるようにします。この例でトリガーノードが作成されるためには、NFS に /export/share/ws/usr が存在している必要があります。

注意 – 階層的にマウントを指定する場合は、-soft オプションは使用しないでください。この制限についての説明は、217 ページの「autofs アンマウント」を参照してください。

autofs アンマウント

一定時間アクセスがないためにアンマウントされるときは、マウントと逆の順序で実行されます。あるディレクトリより上位のディレクトリが使用中であれば、それより下のディレクトリだけがアンマウントされます。アンマウントすると、トリガーノードがすべて削除され、ファイルシステムがアンマウントされます。ファイルシステムが使用中であれば、アンマウントは失敗してトリガーノードは再インストールされます。

注意 – 階層的にマウントを指定する場合は、-soft オプションは使用しないでください。-soft オプションを使用すると、トリガーノードを再インストールする要求がタイムアウトすることがあります。トリガーノードを再インストールできないと、マウントの次の階層にアクセスできません。この問題を解決するには、オートマウンタを使用して、階層にあるすべてのコンポーネントのマウントを解除します。オートマウンタでアンマウントするには、ファイルシステムが自動的にアンマウントされるのを待つか、システムをリブートします。

autofs がクライアント用のもっとも近い読み取り専用ファイルを選択する方法 (複数ロケーション)

次は、直接マップの例です。

```
/usr/local          -ro \
   /bin                   ivy:/export/local/sun4\
   /share                 ivy:/export/local/share\
   /src                   ivy:/export/local/src
/usr/man            -ro   oak:/usr/man \
                          rose:/usr/man \
                          willow:/usr/man
/usr/games          -ro   peach:/usr/games
/usr/spool/news     -ro   pine:/usr/spool/news \
                          willow:/var/spool/news
```

マウントポイント /usr/man および /usr/spool/news には複数の場所があり、/usr/man のマウントポイントは 3 つ、/usr/spool/news のマウントポイントは 2 つの場所が記述されています。複製された場所のどこからマウントしてもユーザーは同じサービスを受けられます。ユーザーの書き込みまたは変更が可能ならば、その変更をロケーション全体で管理しなければならなくなるので、この手順は、読み取り専用のファイルシステムをマウントするときにだけ意味があります。あるときに、あるサーバー上のファイルを変更し、そのすぐあとに別のサーバー上で「同じ」ファイルを変更するといった作業は避けたいものです。この利点は、もっとも利用しやすいサーバーが、そのユーザーの手をまったく必要としないで自動的にマウントされるということです。

ファイルシステムを複製として設定してあると (198 ページの「複製されたファイルシステムとは」を参照)、クライアントはフェイルオーバー機能を使用できます。最適なサーバーが自動的に決定されるだけでなく、そのサーバーが使用できなくなるとクライアントは自動的に 2 番目に適したサーバーを使います。フェイルオーバー機能は、Solaris 2.6 の新機能です。

複製として設定するのに適しているファイルシステムの例は、マニュアルページです。大規模なネットワークでは、複数のサーバーがマニュアルページをエクスポートできます。どのサーバーからマニュアルページをマウントしても、そのサーバーが動作しており、しかもそのファイルシステムをエクスポートしているかぎり、問題ありません。上の例では、複数のマウント位置は、マップエントリ内のマウント位置のリストになっています。

```
/usr/man -ro oak:/usr/man rose:/usr/man willow:/usr/man
```

この例では、サーバー oak、rose、willow のどれからでもマニュアルページをマウントできます。どのサーバーが最適であるかは、次のいくつかの要素によって決まります。

- 特定レベルの NFS プロトコルをサポートしているサーバーの数
- サーバーとの距離
- 重み付け

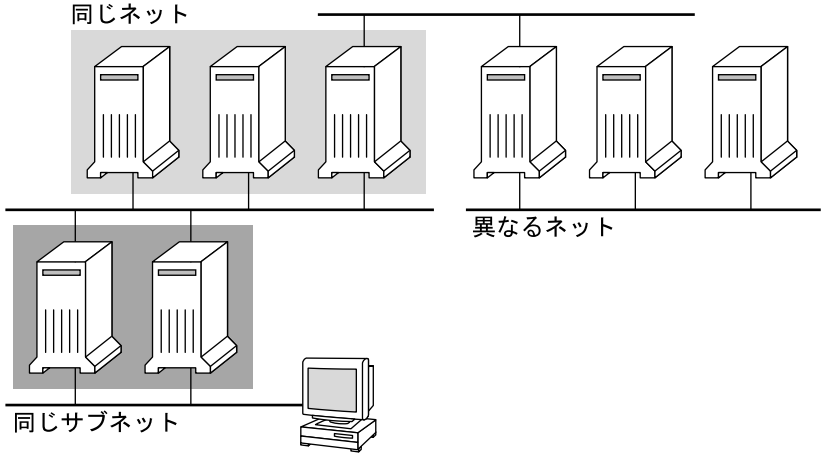
順位を決定するときには、各バージョンの NFS プロトコルをサポートしているサーバーの数が数えられます。サポートしているサーバーの数が多いプロトコルがデフォルトになります。これによって、クライアントにとっては利用できるサーバーの数が最大になります。

プロトコルが同じバージョンのサーバーの組の中で数がもっとも多いものがわかると、サーバーのリストが距離によってソートされます。距離を判定するために、IPv4 アドレスが調査されます。IPv4 アドレスは、どのサーバーが各サブネットにあるかを示します。ローカルサブネット上のサーバーには、リモートサブネット上のサーバーよりも高い優先順位が付けられます。もっとも近いサーバーが優先されることにより、待ち時間とネットワークトラフィックが軽減されます。

注 – IPv6 アドレスを使用している複製に対しては、距離を判定できません。

図 6–5 に、サーバーとの距離を示します。

図 6–5　サーバーとの距離



ローカルサブネット上に同じプロトコルをサポートしているサーバーが複数あるときは、それぞれのサーバーに接続する時間が計測され、速いものが使用されます。優先順位には、重み付けも関係します (221 ページの「autofs と重み付け」を参照してください)。

たとえば、version 4 サーバーの方が多いと、version 4 がデフォルトで使用されるプロトコルになります。ただし、優先順位の決定は複雑になります。次に、優先順位の決定の例をいくつか示します。

- ローカルサブネット上のサーバーには、リモートサブネット上のサーバーよりも高い優先順位が付けられます。ローカルサブネットに version 3 サーバーがあり、もっとも近い version 4 サーバーがリモートサブネット上にあると、version 3 サーバーが優先されます。同様に、ローカルサブネットが version 2 サーバーで構成されていると、version 3 と version 4 サーバーを使用するリモートサブネットよりも優先されます。
- ローカルサブネットがさまざまな数の version 2、version 3、および version 4 サーバーで構成されていると、さらに優先順位付けが必要になります。オートマウンタは、ローカルサブネット上でもっとも高いバージョンを優先します。この場合、version 4 がもっとも高いバージョンです。ただし、ローカルサブネットに、version 4 サーバーよりも version 3 または version 2 サーバーの方が多い場合、オートマウンタはローカルサブネットのもっとも高いバージョンから 1 つ下のバージョンを選択します。たとえば、ローカルサブネットに、version 4 サーバーが 3 台、version 3 サーバーが 3 台、version 2 サーバーが 10 台ある場合、version 3 サーバーが選択されます。
- 同じように、ローカルサブネットがさまざまな数の version 2 と version 3 サーバーで構成されていると、最初にオートマウンタは、どのバージョンがローカルサブネットでもっとも高いバージョンかを見つけます。次に、オートマウンタは各バージョンを実行するサーバーの数を数えます。ローカルサブネット上でもっとも高いバージョンが、同時にもっとも多いサーバーの場合、もっとも高いバージョンが選択されます。低いバージョンのサーバーの数が多い場合、オートマウンタはローカルサブネットのもっとも高いバージョンから 1 つ下のバージョンを選択します。たとえば、ローカルサブネット上で version 2 サーバーの方が version 3 サーバーよりも多い場合、version 2 サーバーが選択されます。

注 – また、重み付けも `/etc/default/nfs` ファイル内のキーワード値に影響されます。特に、NFS_SERVER_VERSMIN、NFS_CLIENT_VERSMIN、NFS_SERVER_VERSMAX、および NFS_CLIENT_VERSMAX の値により、いくつかのバージョンを優先順位の決定から除外することができます。これらのキーワードについての詳細は、142 ページの「`/etc/default/nfs` ファイルのキーワード」を参照してください。

フェイルオーバー機能を指定していると、この優先順位はサーバーが選択されるマウント時に確認されます。複数の場所を指定しておくと、個々のサーバーが一時的にファイルシステムをエクスポートできないときに便利です。

多くのサブネットを持つ大規模ネットワークでは、フェイルオーバーは特に便利です。autofs は適切なサーバーを選択して、ネットワークトラフィックをローカルネットワークのセグメントに限定することができます。サーバーが複数のネットワークインタフェースを持つ場合は、それぞれのインタフェースが別々のサーバーであるとみなして、各ネットワークインタフェースに対応付けられているホスト名を指定します。autofs はそのクライアントにいちばん近いインタフェースを選択します。

注 – 手動によるマウントでは、重み付けと距離の確認は行われません。`mount` コマンドは、左から右へ一覧表示されるサーバーの優先順位を付けます。

詳細は、`automount(1M)` のマニュアルページを参照してください。

autofs と重み付け

距離のレベルが同じサーバーから 1 つを選択するために、autofs マップに重み付けの値を追加することができます。次に例を示します。

```
/usr/man -ro oak,rose(1),willow(2):/usr/man
```

括弧内の数値が重み付けを表します。重み付けのないサーバーの値はゼロであり、選択される可能性が最高になります。重み付けの値が大きいほど、そのサーバーが選択される可能性は低くなります。

注 – 重み付けは、サーバーの選択に関係する要素の中でもっとも小さい影響力しかありません。ネットワーク上の距離が同じサーバーの間で選択を行う場合に考慮されるだけです。

マップエントリ内の変数

変数名の前にドル記号 ($) を付けることによって、クライアント固有の変数を作成できます。この変数は、同じファイルシステムの位置にアクセスする異なるアーキテクチャータイプの調整に役立ちます。変数名を括弧でくくることで、その後に続く文字や数字と変数とを区切ることができます。表 6–2 に定義済みのマップ変数を示します。

表 6–2 定義済みのマップ変数

変数	意味	提供元	例
ARCH	アーキテクチャータイプ	uname -m	sun4
CPU	プロセッサタイプ	uname -p	sparc
HOST	ホスト名	uname -n	dinky
OSNAME	オペレーティングシステム名	uname -s	SunOS
OSREL	オペレーティングシステムのリリース	uname -r	5.8
OSVERS	オペレーティングシステムのバージョン (リリースのバージョン)	uname -v	GENERIC

キーとして使用する場合を除いて、変数はエントリ行内のどこにでも使用できます。たとえば、/usr/local/bin/sparc および /usr/local/bin/x86 から、SPARC アーキテクチャーと x86 アーキテクチャーのバイナリをそれぞれエクスポートするファイルサーバーがあるとします。クライアントは、次のようなマップエントリを使ってマウントすることができます。

```
/usr/local/bin        -ro     server:/usr/local/bin/$CPU
```

これで、すべてのクライアントの同じエントリがすべてのアーキテクチャーに適用されます。

注 – どの sun4 アーキテクチャー向けに書かれたアプリケーションでも、ほとんどはすべての sun4 プラットフォームで実行できます。-ARCH 変数は、sun4 にハードコードされています。

他のマップを参照するマップ

ファイルマップで使用されたマップエントリ +*mapname* により、automount は指定されたマップを、あたかも現在のマップに含まれているかのように読み取ります。*mapname* の前にスラッシュがない場合、autofs はそのマップ名を文字列として扱い、ネームサービススイッチ方式を使用してマップ名を検出します。パス名が絶対パス名の場合、automount はその名前のローカルマップを検索します。マップ名がダッシュ (-) で始まる場合、automount は hosts などの適切な組み込みマップを参照します。

このネームサービススイッチファイルには、automount と指定された autofs 用のエントリが収められています。そしてそのエントリには、ネームサービスが検索される順序が収められています。ネームサービススイッチファイルの例を次に示します。

```
#
# /etc/nsswitch.nis:
#
# An example file that could be copied over to /etc/nsswitch.conf;
# it uses NIS (YP) in conjunction with files.
#
# "hosts:" and "services:" in this file are used only if the /etc/netconfig
# file contains "switch.so" as a nametoaddr library for "inet" transports.
# the following two lines obviate the "+" entry in /etc/passwd and /etc/group.
passwd:         files nis
group:          files nis

# consult /etc "files" only if nis is down.
hosts:          nis [NOTFOUND=return] files
networks:       nis [NOTFOUND=return] files
protocols:      nis [NOTFOUND=return] files
rpc:            nis [NOTFOUND=return] files
ethers:         nis [NOTFOUND=return] files
netmasks:       nis [NOTFOUND=return] files
bootparams:     nis [NOTFOUND=return] files
publickey:      nis [NOTFOUND=return] files
netgroup:       nis
automount:      files nis
aliases:        files nis
# for efficient getservbyname() avoid nis
services:       files nis
```

この例では、ローカルマップが NIS マップよりも先に検索されます。そのため、ローカルマップ /etc/auto_home に、もっとも頻繁にアクセスするホームディレクトリ用のエントリを含めることができます。他のエントリについては、スイッチを使用して NIS マップにフォールバックすることができます。

```
bill              cs.csc.edu:/export/home/bill
bonny             cs.csc.edu:/export/home/bonny
```

組み込まれたマップを参照したあと、一致するものがなければ、automount は現在のマップの走査を続けます。そのため、+ エントリの後にさらにエントリを追加できます。

```
bill              cs.csc.edu:/export/home/bill
bonny             cs.csc.edu:/export/home/bonny
+auto_home
```

組み込まれたマップは、ローカルファイルまたは組み込みマップとすることができます。ローカルファイルだけが + エントリを持つことができることに注意してください。

```
+auto_home_finance      # NIS+ map
+auto_home_sales        # NIS+ map
+auto_home_engineering  # NIS+ map
+/etc/auto_mystuff      # local map
+auto_home              # NIS+ map
+-hosts                 # built-in hosts map
```

注 – NIS+ または NIS のマップでは「+」エントリを使用できません。

実行可能な autofs マップ

autofs マウントポイントを生成するコマンドを実行する autofs マップを作成することもできます。データベースやフラットファイルから autofs 構造を作成しなければならない場合は、実行可能な autofs マップが有効なことがあります。短所は、マップをすべてのホストにインストールしなければならないことです。実行可能なマップは、NIS と NIS+ のどちらのネームサービスにも含めることができません。

実行可能マップは、auto_master ファイルにエントリが必要です。

```
/execute    auto_execute
```

実行可能マップの例を示します。

```
#!/bin/ksh
#
# executable map for autofs
#

case $1 in
            src)  echo '-nosuid,hard bee:/export1' ;;
esac
```

この例が機能するためには、ファイルが /etc/auto_execute としてインストールされ、実行可能ビットがオンになっている必要があります。アクセス権は 744 に設定します。この場合、次のコマンドを実行すると、bee のファイルシステム /export1 がマウントされます。

```
% ls /execute/src
```

autofs のネットワークナビゲート法の変更 (マップの変更)

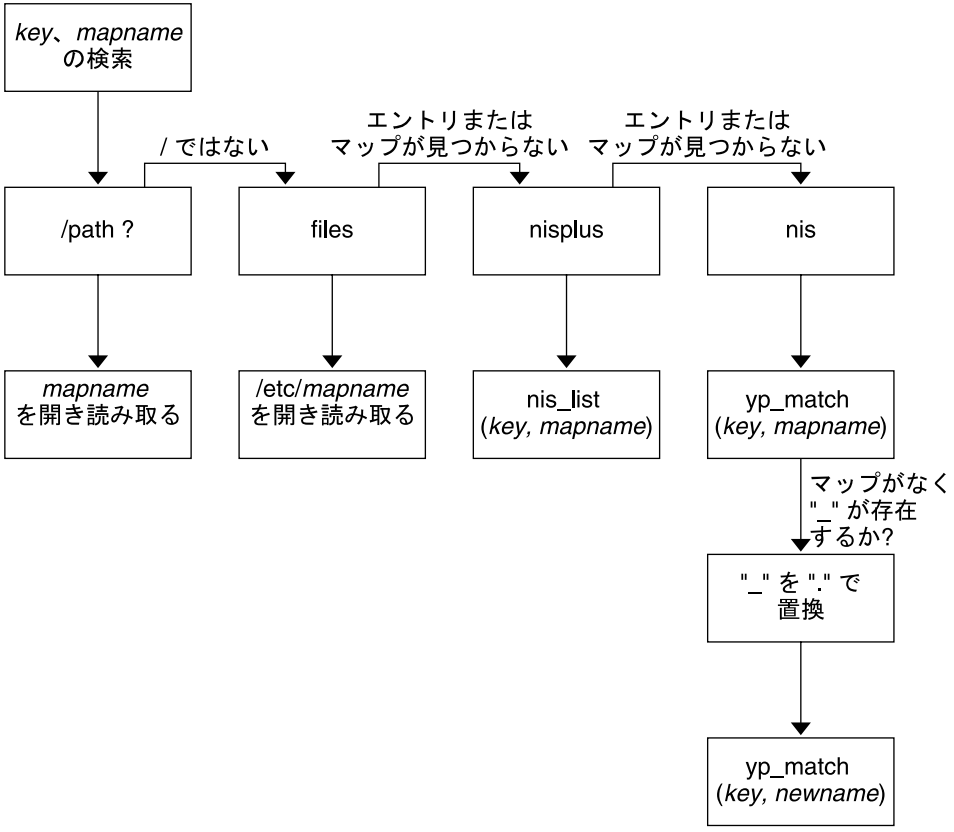
マップへのエントリを変更、削除、または追加して、ユーザーの環境ニーズに合わせることができます。ユーザーが必要とするアプリケーションやその他のファイルシステムがその位置を変更すると、マップはこれらの変更を反映しなければなりません。autofs のマップは、いつでも変更できます。automountd が次にファイルシステムをマウントしたときにその変更内容が有効になるかどうかは、変更したマップと変更内容によって決まります。

ネームサービスに対する autofs のデフォルトの動作

ブート時に、autofs は、サービス `svc:/system/filesystem/autofs` によって起動され、マスターマップ `auto_master` を確認します。次に説明する規則が適用されます。

autofs は、`/etc/nsswitch.conf` ファイルの自動マウントエントリで指定されたネームサービスを使用します。ローカルファイルや NIS ではなく NIS+ が指定された場合、マップ名はすべてそのまま使用されます。NIS を選択し、autofs が必要なマップを検出できない場合で、1 つまたは複数の下線を含むマップ名を検出したときには、それらの下線がドットに変更されます。こうすることにより、NIS の古いファイル名を利用することができます。次に autofs はもう 1 度マップを調べます。この手順を図 6–6 に示します。

図 6–6　autofs によるネームサービスの使用



このセッションでは、画面は次の例のようになります。

```
$ grep /home /etc/auto_master
/home           auto_home

$ ypmatch brent auto_home
Can't match key brent in map auto_home.  Reason: no such map in
server's domain.

$ ypmatch brent auto.home
diskus:/export/home/diskus1/&
```

ネームサービスとして「ファイル」が選択された場合、すべてのマップは /etc ディレクトリ内のローカルファイルとみなされます。autofs は、使用するネームサービスとは無関係に、スラッシュ (/) で始まるマップ名をローカルとして解釈します。

autofs リファレンス

これ以降の節では、autofs の高度な機能を取り上げます。

autofs とメタキャラクタ

autofs は一部の文字を、特別な意味を持つものとして認識します。置き換えに使用する文字や、autofs のマップ構文解析機能から他の文字を保護するために使用する文字もあります。

アンパサンド (&)

たとえば、次のように、多数のサブディレクトリを指定したマップがある場合は、文字列置換を使用できます。

```
john        willow:/home/john
mary        willow:/home/mary
joe         willow:/home/joe
able        pine:/export/able
baker       peach:/export/baker
```

この場合、アンパサンド文字 (&) を使用して、任意の位置に記述されたこのキーを置換することができます。アンパサンド文字を使用すると、前述のマップは次のようになります。

```
john        willow:/home/&
mary        willow:/home/&
joe         willow:/home/&
able        pine:/export/&
baker       peach:/export/&
```

キー置換はまた、次のような直接マップでも使用できます。

```
/usr/man                        willow,cedar,poplar:/usr/man
```

また、このエントリは、次のようにさらに簡単にすることができます。

```
/usr/man                        willow,cedar,poplar:&
```

アンパサンド文字による置換では、キー文字列全体を使用していることに注意してください。そのため、直接マップ内のキーの最初の文字が / である場合は、そのスラッシュが置換に含まれます。したがって、次のように指定することはできません。

```
/progs                &1,&2,&3:/export/src/progs
```

これは、autofs が、この例を次のように解釈するためです。

```
/progs                 /progs1,/progs2,/progs3:/export/src/progs
```

アスタリスク (*)

任意のキーを一致させるのに、任意の文字を表す置換文字であるアスタリスク (*) を使用できます。このマップエントリを使用して、すべてのホストから /export ファイルシステムをマウントできます。

```
*                       &:/export
```

ここでは、各アンパサンドは特定のキーの値によって置換されています。autofs はこのアスタリスクをファイルの終わりとして解釈します。

autofs と特殊文字

特殊文字が含まれているマップエントリがある場合に、autofs のマップ構文解析機能を混乱させる名前のディレクトリをマウントする必要があるかもしれません。autofs の構文解析機能は、名前に含まれるコロン、コンマ、スペースなどを認識します。これらの名前は二重引用符で囲んでください。

```
/vms    -ro    vmsserver: -  -  - "rc0:dk1 - "
/mac    -ro    gator:/ - "Mr Disk - "
```

パート III

SLP (トピック)

このパートでは、サービスロケーションプロトコル (SLP) サービスの概要、計画、作業、およびリファレンス情報について説明します。

◆ ◆ ◆ 第 7 章

7 SLP (概要)

サービスロケーションプロトコル (SLP) は、SLP が使用できるネットワークサービスを検出しそれに対応するための、移植性が高くプラットフォームに依存しないフレームワークを提供します。この章では、SLP のアーキテクチャーの概要と、IP イントラネットに対応する SLP の Solaris での実装について説明します。

SLP のアーキテクチャー

この節では、SLP の基本的な処理を示し、SLP の管理で使用されるエージェントとプロセスについて説明します。

SLP は、次のサービスを自動的に行い、設定はほとんどあるいはまったく必要ありません。

- クライアントアプリケーションがサービスへのアクセスに必要な情報を要求する
- プリンタ、ファイルサーバー、ビデオカメラ、HTTP サーバーなどのネットワークのハードウェアデバイスやソフトウェアサーバーにサービスを通知する
- 主サーバーの障害からの管理された回復

また、SLP の動作を管理、調整するために、必要に応じて次のことを実行できます。

- サービスとユーザーを論理グループや機能グループから構成されるスコープに編成する
- SLP のロギングを有効にして、ネットワーク上の SLP 動作の監視とトラブルシューティングを行う
- SLP のタイミングパラメータを調整して、パフォーマンスの向上とスケーラビリティーの拡張を行う

- SLP がマルチキャストルーティングに対応していないネットワークに配置されている場合、マルチキャストメッセージの送信や処理を行わないように SLP を構成する
- SLP のディレクトリエージェントを配置して、スケーラビリティーとパフォーマンスを改善する

SLP 設計の概要

SLP ライブラリは、サービスをネットワークで検出するための情報を、サービスを通知するネットワーク対応のエージェントに与えます。SLP エージェントは、サービスの種類と場所に関する最新情報を保持します。これらのエージェントはプロキシ登録を使用することで、SLP が直接使用できないサービスを通知することもできます。詳細は、第 10 章「レガシーサービスの組み込み」を参照してください。

クライアントアプリケーションは、SLP ライブラリに依頼して、サービスを通知するエージェントに直接要求を出してもらいます。

SLP エージェントとプロセス

次の表では、SLP エージェントについて説明します。ここで使用する用語の詳細な定義は、用語集を参照してください。

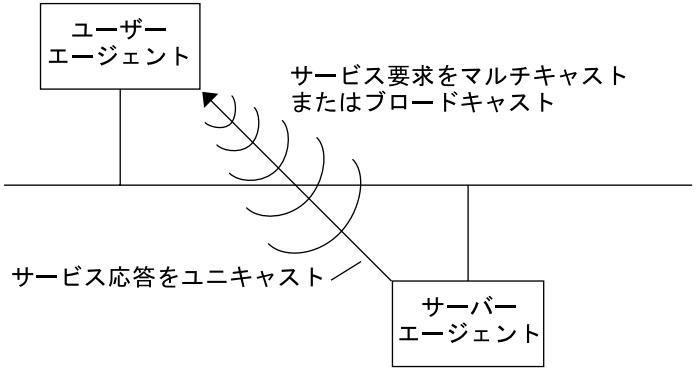
表 7–1 SLP エージェント

SLP エージェント	説明
ディレクトリエージェント (DA)	サービスエージェント (SA) が登録する SLP 通知をキャッシュするプロセス。DA は、要求に応じて、サービス通知をユーザーエージェント (UA) に転送します。
サービスエージェント (SA)	サービス通知を配信するためやサービスをディレクトリエージェント (DA) に登録するために、サービスの代理として動作する SLP エージェント。
ユーザーエージェント (UA)	サービス通知情報を取得するために、ユーザーやアプリケーションの代理として動作する SLP エージェント。
スコープ	サービスに対する管理上または論理上のグループ。

次の図は、SLP アーキテクチャーを実装する、基本的なエージェントおよびプロセスを示しています。図は、SLP のデフォルトの配置を表しています。特別な構成はまったく行われていません。UA と SA の 2 つのエージェントだけが必要です。SLP フレームワークでは、UA がサービス要求を SA にマルチキャストすることを許可しています。SA は、UA に対して応答をユニキャストします。たとえば、UA がサービ

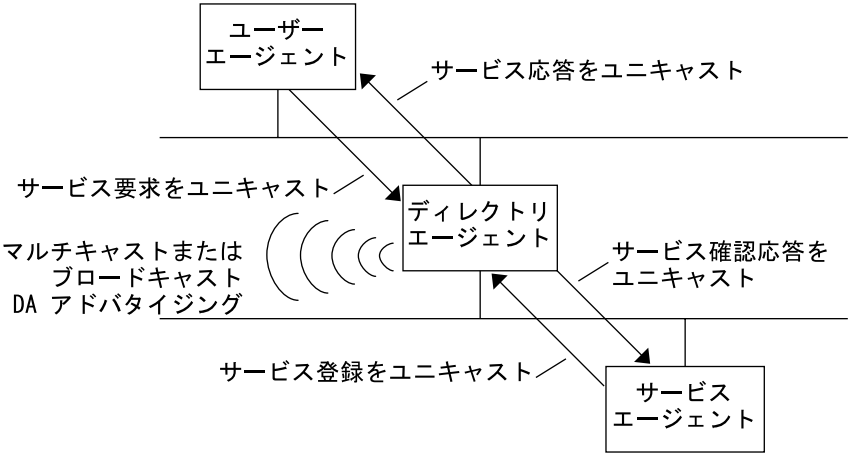
ス要求メッセージを送信すると、SA はサービス応答メッセージを返します。サービス応答には、クライアントの要求と一致するサービスの場所が含まれています。属性やサービスタイプに関する要求や応答も可能です。詳細は、第 11 章「SLP (リファレンス)」を参照してください。

図 7–1　SLP の基本的なエージェントとプロセス



次の図は、フレームワークに DA が配置された場合の、SLP アーキテクチャーを実装する基本的なエージェントとプロセスを示しています。

図 7–2　DA を使って実装される SLP アーキテクチャーのエージェントとプロセス



DA を配置すると、ネットワークにはより少ないメッセージが送られるので、UA は情報をすばやく受け取ることができます。DA は、ネットワークのサイズが増大する場合やマルチキャストルーティングがサポートされていない場合に必要です。DA は

登録されたサービス通知のキャッシュの役割を果たします。SA は DA に対して、通知するすべてのサービスを一覧表示した登録メッセージ (SrvReg) を送り、その応答として確認応答 (SrvAck) を受け取ります。サービス通知は DA によって更新されるか、通知に設定された有効期限に従って期限切れになります。UA が DA を検出すると、UA は要求を SA にマルチキャストするのではなく、DA にユニキャストします。

Solaris SLP メッセージの詳細は、第 11 章「SLP (リファレンス)」を参照してください。

SLP の実装

Solaris SLP の実装では、表 7–1 にある SLP の SA、UA、DA、SA サーバー、スコープなどのアーキテクチャーコンポーネントが一部は slpd に、一部はアプリケーションプロセスに割り当てられます。SLP デーモン (slpd) は、特定のオフホストの SLP 相互作用を構成して、次のことを実行します。

- ネットワーク上のすべての DA に対し、ディレクトリエージェントの受動的検出と能動的検出を使用する
- ローカルホスト上の UA と SA が使用するために DA の更新テーブルを保持する
- レガシーサービス通知に対してプロキシ SA サーバーとして機能する (プロキシ登録)

net.slpisDA プロパティーを設定し、slpd が DA として機能するように構成することもできます。第 9 章「SLP の管理 (手順)」を参照してください。

SLP デーモンの詳細は、slpd(1M) のマニュアルページを参照してください。

slpd の他に、C/C++ クライアントライブラリと Java クライアントライブラリ (libslp.so および slp.jar) が、UA クライアントと SA クライアントに SLP のフレームワークへのアクセス権を提供します。クライアントライブラリは、次の機能を提供します。

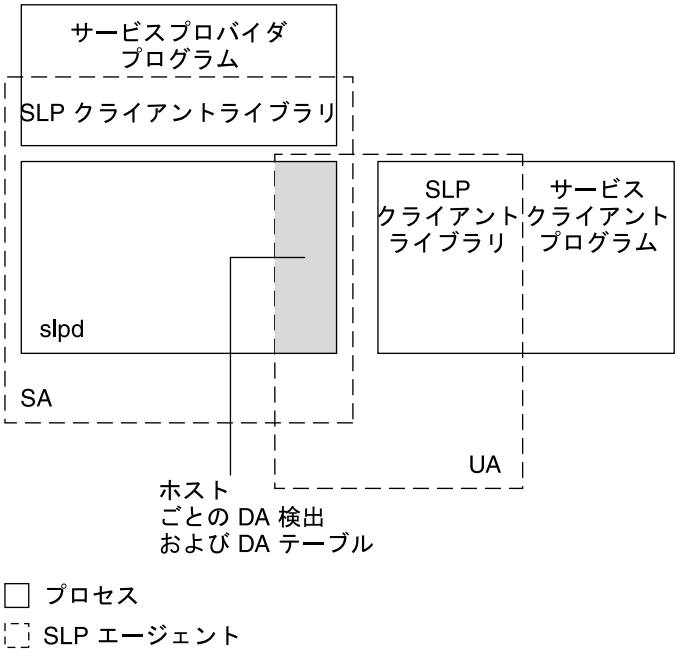
- サービス通知の登録と登録解除が可能なネットワークサービスを提供するソフトウェア
- サービス通知にクエリーを発行することによってサービスを要求できるクライアントソフトウェア
- 登録と要求に使用できる SLP スコープのリスト

slpd とクライアントライブラリ (前述のサービスを提供する) 間のプロセス間通信を可能にするには、特別な構成は必要ありません。ただし、ライブラリが機能するように、先に slpd プロセスを実行してからクライアントライブラリをロードする必要があります。

次の図で、サービスプロバイダプログラム内の SLP クライアントライブラリは、SA の機能を使用します。サービスプロバイダプログラムは SLP クライアントライブラ

リを使用して、サービスを slpd に登録または登録解除します。サービスクライアントプログラムの SLP クライアントライブラリは、UA の機能を使用します。サービスクライアントプログラムは SLP クライアントライブラリを使用して、要求を出します。SLP クライアントライブラリは、SA に要求をマルチキャストするか、DA に要求をユニキャストします。この通信はアプリケーションから見て透過です。ただし、ユニキャスト方式の要求発行はより高速になります。クライアントライブラリの動作は、SLP のさまざまな構成プロパティーの設定によって影響を受けます。詳細は、第 9 章「SLP の管理 (手順)」を参照してください。slpd プロセスは、マルチキャスト要求への応答、DA への登録など、SA の全機能を処理します。

図 7–3　SLP の実装



SLP の参考資料

SLP の詳細は、次の文書を参照してください。

- Kempf、James、Pete St. Pierre 著、『Service Location Protocol for Enterprise Networks』、John Wiley & Sons、Inc. (ISBN 番号: 0–471–31587–7)。
- 『Authentication Management Infrastructure Administration Guide』(Part No: 805–1139–03)。

- Guttman、 Erik、 Charles Perkins、 John Veizades、 Michael Day 著、 『Service Location Protocol, Version 2, RFC 2608』 、 Internet Engineering Task Force (IETF) 。 [http://www.ietf.org/rfc/rfc2608.txt]
- Kempf、 James、 Erik Guttman 著、 『An API for Service Location, RFC 2614』 、 Internet Engineering Task Force (IETF) 。 [http://www.ietf.org/rfc/rfc2614.txt]

◆ ◆ ◆ 第 8 章

SLP の計画と有効化 (手順)

この章では、SLP の計画と有効化について説明します。次の節では、SLP の構成と SLP を有効にするためのプロセスを取り上げています。

SLP 構成の検討事項

SLP デーモンはデフォルトのプロパティーで構成済みです。デフォルトの設定で正しく動作する場合、SLP の配置において、ほとんど管理は必要ありません。

ただし場合によっては、デフォルトの SLP プロパティーを変更して、SLP のネットワーク動作を調整することや各種の SLP 機能を有効にすることが必要になります。たとえば、いくつかの構成を変更して、SLP のロギングを有効にすることができます。SLP のログ情報と `snoop` トレースの情報によって、追加の構成が必要かどうかを判断できます。

SLP 構成プロパティーは、`/etc/inet` ディレクトリ内の `slp.conf` ファイルにあります。デフォルトのプロパティー設定を変更する場合は、第 9 章「SLP の管理 (手順)」の該当する手順を参照してください。

SLP 構成プロパティーの設定を変更する前に、ネットワーク管理で大切な次のことがらを検討してください。

- 動作しているネットワーク技術の種類
- ネットワーク技術が円滑に処理できるトラフィック量
- ネットワークで使用できるサービスの数と種類
- ネットワーク上のユーザー数、ユーザーが必要とするサービス、もっとも頻繁にアクセスするサービスに関係するユーザーの場所

再構成の判断

SLP 対応の snoop ユーティリティーと SLP ログユーティリティーを使用して、再構成が必要かどうかや、変更する必要があるプロパティーを判断できます。たとえば、次の目的のために特定のプロパティーを再構成する場合があります。

- 各種の待ち時間および帯域幅の性質が混在するネットワークメディアを調整する
- ネットワークの障害または計画されていないパーティション分割から回復させる
- DA を追加して SLP マルチキャストの急増を軽減する
- 新規のスコープを実装して、もっとも頻繁にアクセスするサービスにユーザーを編成する

snoop を使用して SLP 動作を監視する

snoop ユーティリティーは受動的に機能する管理ツールで、ネットワークのトラフィック情報を提供します。ユーティリティー自身が発するトラフィックは最小限で、ネットワーク上のすべての動作を監視できます。

snoop ユーティリティーは、実際の SLP メッセージトラフィックのトレースを行います。たとえば、snoop に slp コマンド行引数を付けて実行すると、登録および登録解除の SLP トレース情報が表示されます。このトレース情報を使用して、登録されているサービスの種類および登録動作の量をチェックできるので、ネットワークの負荷を測定できます。

snoop ユーティリティーは、SLP ホスト間のトラフィックフローの監視にも役立ちます。snoop に slp コマンド行引数を付けて実行し、次の種類の SLP 動作を監視することで、ネットワークまたはエージェントの再構成が必要かどうかを判断できます。

- 特定の DA を使用しているホスト数。この情報により、負荷を均等にするために DA をさらに追加して配置するかどうかを判断できます。
- 特定の DA を使用しているホスト数。この情報により、特定のホストに新規または別のスコープを構成すべきかどうかを判断できます。
- UA がタイムアウトを要求しているか、あるいは DA の確認応答が遅いかどうか。UA のタイムアウトや再伝送を監視することで、DA が過負荷になっているかどうかを判断できます。DA が SA に登録の確認応答を送るのに数秒以上かかっているかどうかも確認できます。この情報により、必要に応じて、DA を追加したりスコープの構成を変更したりして、DA にかかるネットワーク負荷を調整します。

snoop に -V (詳細) コマンド行引数を付けて実行すると、登録の有効期限や SrvReg の新規フラグの値を得ることができるので、再登録の数を削減すべきかどうかを判断できます。

snoop を使用して、次のような別の種類の SLP トラフィックをトレースすることもできます。

- UA クライアントと DA 間のトラフィック
- UA クライアントのマルチキャストとそれに対する SA の応答との間のトラフィック

snoop の詳細は、snoop(1m) のマニュアルページを参照してください。

ヒント – トラフィックおよび輻輳の統計情報を表示するには、netstat コマンドを snoop と併せて使用します。netstat の詳細は、netstat(1M) のマニュアルページを参照してください。

▼ snoop を使用して SLP トレースを実行する方法

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 snoop に slp コマンド行引数を付けて実行します。

```
Brief Mode:
# snoop slp
```

snoop をデフォルトの簡易モードで実行すると、進行中の出力が画面に表示されます。SLP メッセージは SLP トレースあたり 1 行に収まるように切り捨てられます。

```
Verbose Mode:
# snoop -v slp
```

snoop を「詳細」モードで実行すると、進行中の出力がすべて画面に表示されます。出力される情報は次のとおりです。

- サービス URL の完全なアドレス
- すべてのサービス属性
- 登録の有効期限
- すべてのセキュリティーパラメータとフラグ (存在する場合)

注 – slp コマンド行引数をほかの snoop オプションとともに使用できます。

---

snoop slp トレースの分析

次の例では、slpd は *slphost1* 上で SA サーバーとしてデフォルトモードで動作しています。SLP デーモンは、*slphost2* をエコーサーバーとして初期化して登録しています。その後、snoop slp プロセスが *slphost1* 上で呼び出されます。

注 – トレース結果を説明しやすくするために、次の snoop からの出力結果にトレース行番号を付けています。

```
(1) slphost1 -> 239.255.255.253 SLP V@ SrvRqst [24487] service:directory-agent []
(2) slphost2 -> slphost1 SLP V2 DAAdvert [24487] service:directory-agent://129
(3) slphost1 -> 239.255.255.253 SLP V2 SrvRqst [24487] service:directory-agent []
(4) slphost1 -> 239.255.255.253 SLP V2 SrvRqst [24487] service:directory-agent []
(5) slphost1 -> slphost2 SLP V2 SrvReg [24488/tcp]service:echo.sun:tcp://slphost1:
(6) slphost2 -> slphost1 SLP V2 SrvAck [24488/tcp] ok
(7) slphost1 -> slphost2 SLP V2 SrvDereg [24489/tcp] service:echo.sun:tcp://slphost1:
(8) slphost2 -> slphost1 SLP V2 SrvAck [24489/tcp] ok
```

1. *slphost1* 上の slpd が、ディレクトリエージェントを探すために SLP マルチキャストグループアドレスにマルチキャストして、ディレクトリエージェントを能動検出していることを示しています。能動検出に対するメッセージ番号 (24487) は、トレース表示では角括弧内に示されます。
2. トレース 1 からの能動検出要求 24487 に対し、ホスト slphost2 上で DA として動作している *slpd* が応答したことを示します。*slphost2* からのサービス URL は 1 行に収まるように切り捨てられています。トレース 1 および 2 のメッセージ番号が一致していることからわかるように、DA はマルチキャストディレクトリエージェント検出メッセージに応答して、DA 通知を送っています。
3. 追加の DA に対する *slphost1* 上の UA からのマルチキャストを示します。*slphost2* はすでに要求に応答しているため、ふたたび応答することはなく、他のどの DA も応答しません。
4. 前の行で示したマルチキャストを繰り返しています。
5. slphost1 上の *slpd* は、SA クライアントが作成した登録をホスト *slphost2* 上の DA に転送します。エコーサーバーに対するユニキャストサービス登録 (SrvReg) が、*slphost1* によって *slphost2* 上の DA に行われています。
6. *slphost2* が *slphost1* の SrvReg に対してサービス確認応答 (SrvAck) で応答していることを表し、登録が完了したことを示しています。

   SA クライアントを稼働しているエコーサーバーと *slphost1* 上の SLP デーモンとの間のトラフィックは、snoop トレースでは表示されません。表示されないのは、snoop 動作がネットワークループバック上で実行されているからです。
7. *slphost1* 上のエコーサーバーが、エコーサービス通知の登録を解除します。*slphost1* 上の SLP デーモンは、登録解除を *slphost2* 上の DA に転送します。

8. *slphost2* が *slphost1* に対してサービス確認応答 (SrvAck) で応答していることを表し、登録解除が完了したことを示しています。

   トレース行 5、6、7、8 のメッセージ番号に追加されている /tcp パラメータは、メッセージ交換が TCP で発生したことを示しています。

次に進む手順

SLP トラフィックを監視後、snoop トレースから集められた情報を使用して、SLP デフォルトの再構成が必要かどうかを判断できます。SLP プロパティー値の設定については、第 9 章「SLP の管理 (手順)」を参照してください。SLP メッセージとサービス登録については、第 11 章「SLP (リファレンス)」を参照してください。

◆ ◆ ◆ 第 9 章

9 SLP の管理 (手順)

この章では、SLP のエージェントとプロセスを構成するための情報と作業手順について説明します。

SLP プロパティーの構成

SLP 構成プロパティーは、ネットワークの相互作用、SLP エージェントの特性、状態、およびログを制御します。ほとんどの場合、これらのプロパティーのデフォルトの構成は変更する必要がありません。ただし、ネットワークの媒体またはトポロジが変更されて、次のことを行うためには、この章の手順を使用します。

- ネットワークの待ち時間を補正する
- ネットワークの輻輳を軽減する
- エージェントの追加、または IP アドレスの再割り当てを行う
- SLP ログを起動する

SLP 構成ファイル `/etc/inet/slp.conf` を編集すると、次の表に示す処理を行うことができます。

表 9–1 SLP 構成の操作

操作	説明
slpd が DA サーバーと SA サーバーのどちらで機能するかを指定します。SA サーバーがデフォルトです。	net.slpisDA プロパティーに True を設定します。
マルチキャストメッセージのタイミングを設定します。	net.slp.DAHeartBeat プロパティーを設定して、非要請 DA 通知を DA がマルチキャストする回数を制御します。
DA ロギングを使用可能にしてネットワークトラフィックを監視します。	net.slp.traceDATraffic プロパティーに True を設定します。

SLP 構成ファイルの基本要素

/etc/inet/slp.conf ファイルは、SLP デーモンを再起動するたびにすべての SLP 動作を定義して起動します。構成ファイルは次の要素から成ります。

- 構成プロパティー
- コメント行と注釈

設定プロパティー

net.slp.isDA や net.slp.DAHeartBeat などのすべての基本的な SLP プロパティーは、次の書式で名前が付けられています。

```
net.slp.<keyword>
```

SLP の動作は、slp.conf ファイル内のプロパティーの値またはプロパティーの組み合わせによって定義されます。プロパティーは、SLP 構成ファイル内でキーと値の対で構成されています。次の例に示すように、キーと値の対は、プロパティー名とその設定値で構成されています。

```
<property name>=<value>
```

各プロパティーのキーはプロパティー名です。値はプロパティーに、数値 (間隔または時間)、真偽の状態、または文字列値のパラメータを設定します。プロパティーの値は次のデータ型の 1 つで構成されます。

- 真偽設定 (ブール型)
- 整数
- 整数のリスト
- 文字列
- 文字列のリスト

定義した値が許可されていない場合は、そのプロパティー名のデフォルト値が使用されます。さらに、syslog を使用してエラーメッセージが記録されます。

コメント行と注釈

slp.conf ファイルにコメントを追加して、その行の性質および機能を説明できます。コメント行はファイルに任意に書き込めるので、管理する上で役立ちます。

注 – 構成ファイル内の設定には、大文字と小文字の区別がありません。詳細は、Guttman、Erik、James Kempf、Charles Perkins 著、 Internet Engineering Task Force (IETF) 発行の『Service Templates and service: scheme RFC 2609』を参照してください。[http://www.ietf.org/rfc/rfc2609.txt]

▼ SLP 構成の変更方法

SLP 構成ファイルのプロパティー設定を変更するには、次の手順を実行します。SLP を使用できるクライアントまたはサービスソフトウェアは、SLP API を使用して、SLP 構成も変更できます。API については、Internet Engineering Task Force (IETF) 発行の『An API for Service Location, RFC 2614』を参照してください。[http://www.ietf.org/rfc/rfc2614.txt]

1. スーパーユーザーになるか、同等の役割を引き受けます。

   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2. ホスト上の slpd とすべての **SLP** 動作を停止します。

   ```
   # svcadm disable network/slp
   ```

3. 構成の設定を変更する前に、デフォルトの /etc/inet/slp.conf ファイルのバックアップをとります。

4. 必要に応じて、/etc/inet/slp.conf ファイルのプロパティー設定を編集します。

   SLP プロパティーの設定については、244 ページの「設定プロパティー」を参照してください。slp.conf プロパティーを変更する別の例については、後述の各節を参照してください。slp.conf(4) のマニュアルページを参照してください。

5. 変更を保存し、ファイルを閉じます。

6. 変更を反映するには、slpd を再起動します。

   ```
   # svcadm enable network/slp
   ```

注 – slpd を停止または起動するとき、SLP デーモンは構成ファイルから情報を取得します。

例 9–1 slpd が DA サーバーとして動作するように設定する

slpd.conf ファイルの net.slp.isDA プロパティーに True を設定して、slpd が DA サーバーとして動作するように SA サーバーのデフォルトを変更できます。

```
net.slp.isDA=True
```

各領域で、各種のプロパティーが構成の異なる場合を制御します。以降の各節では、SLP 構成で使用するデフォルトのプロパティー設定を変更するさまざまなシナリオについて説明します。

DA 通知と検出頻度の変更

次のような場合は、DA 通知と検出要求のタイミングを制御するプロパティーを変更できます。

- SA または UA が slp.conf ファイルの net.slp.DAAddresses プロパティーから静的に DA 構成情報を取得するように設定する場合は、DA 検出を無効にできます。
- ネットワークが頻繁にパーティション分割を行う場合は、受動的な通知および定期的な能動的検出の頻度を変更できます。
- UA と SA クライアントがダイアルアップ接続の一方の側で DA にアクセスしている場合は、DA のハートビート頻度と能動的検出の間隔を減らすことで、ダイアルアップ回線の起動回数を少なくできます。
- ネットワークが輻輳している場合は、マルチキャストを制限できます。

この節の手順では、次のプロパティーを変更する方法について説明します。

表 9–2 DA 通知タイミングと検出要求のプロパティー

プロパティー	説明
net.slp.passiveDADetection	非要請 DA 通知を slpd が待機するかどうかを示すブール値
net.slp.DAActiveDiscoveryInterval	新しい DA に対して slpd が DA の能動的検出を実行する頻度を示す値
net.slp.DAHeartBeat	非要請 DA 通知を DA がマルチキャストする頻度を示す値

UA と SA を静的に構成された DA に限定する

UA と SA が slp.conf ファイル内の静的な構成情報から DA アドレスだけを取得するように制限することが必要な場合があります。次の手順では、slpd が net.slp.DAAddresses プロパティーから DA 情報だけを取得するように 2 つのプロパティーを変更できます。

▼ UA と SA を静的に構成された DA に限定する方法

次の手順に従って、net.slp.passiveDADetection および net.slp.DAActiveDiscoveryInterval プロパティーを変更します。

注 – この手順は、静的な構成を使用するように制限されている UA と SA を実行するホストにだけ使用してください。

---

1. スーパーユーザーになるか、同等の役割を引き受けます。

   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2. ホスト上の slpd とすべての **SLP** 動作を停止します。

   ```
   # svcadm disable network/slp
   ```

3. 構成の設定を変更する前に、デフォルトの /etc/inet/slp.conf ファイルのバックアップをとります。

4. slp.conf ファイル内の net.slp.passiveDADetection プロパティーに False を設定して、受動的検出を無効にします。この設定により、slpd は非要請 **DA** 通知を無視します。

   ```
   net.slp.passiveDADetection=False
   ```

5. net.slp.DAActiveDiscoveryInterval に -1 を設定して、初期および定期の能動的検出を無効にします。

   ```
   net.slp.DAActiveDiscoveryInterval=-1
   ```

6. 変更を保存し、ファイルを閉じます。

7. 変更を反映するには、slpd を再起動します。

   ```
   # svcadm enable network/slp
   ```

ダイアルアップネットワークに対する DA 検出の構成

UA または SA がダイアルアップネットワークによって DA から切り離されている場合は、DA 検出を構成して、検出要求と DA 通知の数を削減するか、完全になくすことができます。ダイアルアップネットワークでは、通常起動時に課金されます。余分な通話を最小限に抑えることにより、ダイアルアップネットワークの使用コストを削減できます。

注 – 247 ページの「UA と SA を静的に構成された DA に限定する」で説明している方法で、DA 検出を完全に無効にすることができます。

▼ ダイアルアップネットワークに対する DA 検出の構成方法

次の手順に従って、DA ハートビートの期間と能動的検出の間隔を長くすることで、非要請 DA 通知と能動的検出を削減できます。

1. スーパーユーザーになるか、同等の役割を引き受けます。

   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2. ホスト上の `slpd` とすべての **SLP** 動作を停止します。

   ```
   # svcadm disable network/slp
   ```

3. 構成の設定を変更する前に、デフォルトの `/etc/inet/slp.conf` ファイルのバックアップをとります。

4. `slpd.conf` ファイル内の `net.slp.DAHeartbeat` プロパティーの値を大きくします。

   `net.slp.DAHeartbeat=`*value*

   value　DA 通知の受動的ハートビートに対して秒数を設定する、32 ビットの整数

   デフォルト値は、10800 秒 (3 時間) です

   値の範囲は、2000 から 259200000 秒です

   たとえば、DA を実行しているホストに対して、DA のハートビートを約 18 時間に設定できます。

   ```
   net.slp.DAHeartbeat=65535
   ```

5 slpd.conf ファイル内の net.slp.DAActiveDiscoveryInterval プロパティーの値を大きくします。

```
net.slp.DAActiveDiscoveryInterval value
```

value DA の能動的検出クエリーに対して秒数を設定する、32 ビットの整数

デフォルトの値は、900 秒 (15 分) です

値の範囲は、300 から 10800 秒です

たとえば、UA と SA を実行しているホストに対して、DA の能動的検出の間隔を 18 時間に設定できます。

```
net.slp.DAActiveDiscoveryInterval=65535
```

6 変更を保存し、ファイルを閉じます。

7 変更を反映するには、slpd を再起動します。

```
# svcadm enable network/slp
```

頻繁なパーティション分割に対する DA のハートビートの構成

スコープをサポートするすべての DA に登録するには、SA が必要です。DA は、slpd が能動的検出を行なったあとで現れることがあります。DA が slpd スコープをサポートする場合、SLP デーモンはホスト上のすべての通知を DA に登録します。

slpd が DA を検出する 1 つの方法は、起動時に DA が送り出す初期の非要請通知を使用します。SLP デーモンは定期的な非要請通知 (ハートビート) を使用して、DA がまだアクティブであるかどうかを判断します。ハートビートが出現しない場合、SLP デーモンは自分が使用する DA を削除し、これを UA に申し出ます。

最後に、DA にシャットダウン要求が出されると、DA は特別な DA 通知を転送して、受信中の SA サービスに DA がサービスから抜け出すことを知らせます。SLP デーモンもこの特別な通知を使用して、キャッシュからアクティブでない DA を削除します。

ネットワークが頻繁にパーティション分割を行い、SA の期限が長い場合、ハートビートの通知を受けなければ、slpd はパーティションの分割中に DA をキャッシュから削除できます。ハートビートの頻度を減らすことにより、使用中止になった DA がパーティションの修正後にキャッシュに復元されるまでの遅延時間を縮小できます。

▼ 頻繁なパーティション分割に対して DA のハートビートを構成する方法

次の手順に従って、net.slp.DAHeartBeat プロパティーを変更し、DA のハートビート期間を短くします。

注 – DA 検出が完全に無効になっている場合、UA と SA を実行しているホストが正しい DA にアクセスするように、そのホストの slp.conf の net.slp.DAAddresses プロパティーを設定する必要があります。

---

1. スーパーユーザーになるか、同等の役割を引き受けます。

   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2. ホスト上の slpd とすべての **SLP** 動作を停止します。

   ```
   # svcadm disable network/slp
   ```

3. 構成の設定を変更する前に、デフォルトの /etc/inet/slp.conf ファイルのバックアップをとります。

4. net.slp.DAHeartBeat の値を **1** 時間 **(3600** 秒**)** に短縮します。デフォルトでは、**DA** のハートビート期間は **3** 時間 **(10800** 秒**)** に設定されています。

   ```
   net.slp.DAHeartBeat=3600
   ```

5. 変更を保存し、ファイルを閉じます。

6. 変更を反映するには、slpd を再起動します。

   ```
   # svcadm enable network/slp
   ```

ネットワーク輻輳の軽減

ネットワークが非常に混雑している場合、マルチキャストの量を制限できます。ネットワークに DA を配置していない場合は、DA を配置すると SLP 関連のマルチキャストの量を大幅に削減できます。

ただし、DA の配置後でも DA 検出のためのマルチキャストは必要です。DA 検出に必要なマルチキャストの量は、248 ページの「ダイアルアップネットワークに対する DA 検出の構成方法」で説明している方法で削減できます。247 ページの「UA と SA を静的に構成された DA に限定する」で説明している方法で、DA 検出のためのマルチキャストを完全になくすことができます。

異なるネットワーク媒体、トポロジ、または構成の調整

この節では、次のプロパティーを変更して SLP のパフォーマンスを調整する場合の可能なシナリオについて説明します。

表 9–3 SLP パフォーマンスのプロパティー

プロパティー	説明
net.slp.DAAttributes	DA が通知を受け取る最短の更新間隔。
net.slp.multicastTTL	マルチキャストパケットの有効期限。
net.slp.MTU	ネットワークパケットのサイズ (バイト)。サイズには、IP と TCP または UDP の各ヘッダーが含まれています。
net.slp.isBroadcastOnly	ブロードキャストを DA サービス検索および DA ベースでないサービス検索に使用する必要があるかどうかを示すために設定されるブール値。

SA 再登録の削減

SA は、期限が切れる前に定期的にサービス通知を更新する必要があります。DA が多くの UA および SA から非常に重い負荷を受けている場合は、頻繁な更新により DA が過負荷になることがあります。DA が過負荷になると、UA の要求がタイムアウトして欠落します。UA 要求のタイムアウトには多くの原因が考えられます。DA の過負荷が問題であると判断する前に、snoop トレースを使ってサービス登録に登録されているサービス通知の有効期限を確認してください。有効期限が短く、再登録が頻繁に発生している場合は、再登録が頻繁すぎることがタイムアウトの原因と考えられます。

注 – サービス登録は、FRESH フラグが設定されていなければ再登録になります。サービス登録メッセージについては、第 11 章「SLP (リファレンス)」を参照してください。

▼ SA 再登録を削減する方法

次の手順に従って、SA の最小更新間隔を長くすることで、再登録回数を削減します。

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 ホスト上の slpd とすべての **SLP** 動作を停止します。

```
# svcadm disable network/slp
```

3 構成の設定を変更する前に、デフォルトの /etc/inet/slp.conf ファイルのバックアップをとります。

4 net.slp.DAAttributes プロパティーの min-refresh-interval 属性の値を大きくします。

デフォルトの最短再登録期間はゼロ (0) です。デフォルトのゼロである場合、SA はいつでも自由に再登録できます。次の例では、間隔は 3600 秒 (1 時間) に増やしています。

```
net.slp.DAAttributes(min-refresh-interval=3600)
```

5 変更を保存し、ファイルを閉じます。

6 変更を反映するには、slpd を再起動します。

```
# svcadm enable network/slp
```

マルチキャストの有効期限プロパティーの構成

マルチキャストの有効期限プロパティー (net.slp.multicastTTL) によって、マルチキャストパケットがイントラネット内で伝達される範囲が決まります。マルチキャスト TTL は net.slp.multicastTTL プロパティーを 1 から 255 までの整数に設定することにより構成されます。マルチキャスト TTL のデフォルト値は 255 で、これは理論的にはパケット経路が無制限であることを意味します。しかし、TTL を 255 とすると、マルチキャストパケットがイントラネットを超えて管理ドメインの端にある境界ルーターまで進む原因になります。マルチキャストパケットがインターネットのマルチキャストバックボーンまたは ISP に漏れないようにするには、境界ルーター上のマルチキャストが正しく構成されている必要があります。

マルチキャスト TTL のスコープ設定は、TTL 比較が行われることを除いて、標準的な IP の TTL と似ています。マルチキャストを実行できるルーター上の各インタフェースには、TTL 値が割り当てられています。マルチキャストパケットが着信すると、ルーターはパケットの TTL をインタフェースの TTL と比較します。パケットの TTL がインタフェースの TTL 値と同じかそれより大きい場合は、標準的な IP の TTL の場合と同じように、パケットの TTL を 1 減らします。TTL がゼロになると、そのパケットは破棄されます。SLP マルチキャストに TTL スコープを使用する場合、パケットをイントラネットの特定のサブセクションに限定するために、ルーターが正しく構成されている必要があります。

▼ マルチキャストの有効期限プロパティーの構成方法

次の手順に従って、net.slp.multicastTTL プロパティーを設定し直します。

1. スーパーユーザーになるか、同等の役割を引き受けます。
   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2. ホスト上の slpd とすべての **SLP** 動作を停止します。

   ```
   # svcadm disable network/slp
   ```

3. 構成の設定を変更する前に、デフォルトの /etc/inet/slp.conf ファイルのバックアップをとります。

4. slpd.conf ファイル内の net.slp.multicastTTL プロパティーを変更します。

   net.slp.multicastTTL=*value*

   value　マルチキャスト TTL を定義する 255 以下の正の整数

   注 – TTL 値を減らしてマルチキャストの伝達範囲を縮小することができます。TTL の値が 1 の場合、パケットはそのサブネットに限定されます。TTL の値が 32 の場合は、パケットはそのサイトに限定されます。「サイト」は、マルチキャスト TTL について記述されている RFC 1075 では定義されていません。32 以上の値は、インターネット上の論理的な経路を指すので使用しないでください。32 未満の値は、各ルーターが TTL で正しく構成されていれば、マルチキャストをアクセス可能なサブネットのセットに限定するために使用できます。

   ---

5. 変更を保存し、ファイルを閉じます。

6. 変更を反映するには、slpd を再起動します。

   ```
   # svcadm enable network/slp
   ```

パケットサイズの構成

SLP のデフォルトのパケットサイズは 1400 バイトです。ほとんどのローカルエリアネットワークにはこのサイズで十分です。無線ネットワークまたは広域ネットワークの場合は、メッセージの断片化を防いだりネットワークのトラフィックを削減したりするために、パケットサイズを縮小できます。より大きなパケットを持つローカルエリアネットワークの場合は、パケットサイズを大きくするとパフォーマ

ンスが向上します。ネットワークの最小パケットサイズを確認して、パケットサイズの縮小が必要かどうかを判断できます。ネットワーク媒体のパケットサイズがより小さい場合は、それに合わせて net.slp.MTU の値を小さくできます。

ネットワーク媒体のパケットサイズがより大きい場合は、それに合わせて値を大きくできます。ただし、SA からのサービス通知または UA からのクエリーが頻繁にデフォルトのパケットサイズをオーバーフローするのでなければ、net.slp.MTU の値を変更する必要はありません。snoop を使用して、UA 要求がデフォルトのパケットサイズを頻繁にオーバーフローし、UDP ではなく TCP を使用するためにロールオーバーしているかどうかを判断できます。

net.slp.MTU プロパティーは、リンク層ヘッダー、IP ヘッダー、UDP または TCP ヘッダー、SLP メッセージを含めた、IP パケットの全体サイズを測定します。

▼ パケットサイズの構成方法

次の手順に従って、net.slp.MTU プロパティーを調整することで、デフォルトのパケットサイズを変更します。

1. スーパーユーザーになるか、同等の役割を引き受けます。

   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2. ホスト上の slpd とすべての **SLP** 動作を停止します。

   ```
   # svcadm disable network/slp
   ```

3. 構成の設定を変更する前に、デフォルトの /etc/inet/slp.conf ファイルのバックアップをとります。

4. slpd.conf ファイル内の net.slp.MTU プロパティーを変更します。

   net.slp.MTU=*value*

   value　ネットワークのパケットサイズ (バイト単位) を指定する、16 ビットの整数

   デフォルト値は、1400

   値の範囲は、128 から 8192

5. 変更を保存し、ファイルを閉じます。

6. 変更を反映するには、slpd を再起動します。

   ```
   # svcadm enable network/slp
   ```

ブロードキャスト専用ルーティングの構成

SLP は、DA が存在しない場合のサービス検出や DA 検出を、マルチキャストを使って行うように設計されています。使用するネットワークが、マルチキャストルーティングを配置しない場合は、`net.slp.isBroadcastOnly` プロパティーに `True` を設定することで、SLP がブロードキャストを使用するように構成できます。

マルチキャストと異なり、ブロードキャストパケットはデフォルトでサブネットを越えて伝達しません。このため、マルチキャストを行わないネットワークでは、DA を使用しないサービス検出は、単一のサブネット上でしか機能しません。さらに、ブロードキャストが使用されているネットワークに DA およびスコープを配置する場合は、特別な考慮が求められます。マルチホームホスト上の DA は、マルチキャストが使用できない複数のサブネット間でサービス検出をブリッジできます。マルチホームホスト上の DA の配置については、271 ページの「DA の配置とスコープ名の割り当て」を参照してください。

▼ ブロードキャスト専用ルーティングの構成方法

次の手順に従って、`net.slp.isBroadcastOnly` プロパティーを `True` に変更します。

1. スーパーユーザーになるか、同等の役割を引き受けます。

   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2. ホスト上の `slpd` とすべての **SLP** 動作を停止します。

   ```
   # svcadm disable network/slp
   ```

3. 構成の設定を変更する前に、デフォルトの `/etc/inet/slp.conf` ファイルのバックアップをとります。

4. `slpd.conf` ファイル内の `net.slp.isBroadcastOnly` プロパティーを `True` に変更します。

   ```
   net.slp.isBroadcastOnly=True
   ```

5. 変更を保存し、ファイルを閉じます。

6. 変更を反映するには、`slpd` を再起動します。

   ```
   # svcadm enable network/slp
   ```

SLP 検出要求のタイムアウトの変更

SLP 検出要求のタイムアウトを変更する必要があるのは、次の 2 つの場合です。

- SLP エージェントが複数のサブネット、ダイアルアップ回線、または別の WAN によって切り離されている場合は、ネットワークの待ち時間が長く、デフォルトのタイムアウトでは要求や登録を完了できないことがあります。逆に、ネットワークの待ち時間が短い場合は、タイムアウトを短くすることにより、パフォーマンスが向上することがあります。
- トラフィックが多いネットワークまたは衝突率の高いネットワークの場合、SA および UA がメッセージを送る前に待たなければならない最長の時間が不足して、衝突のないトランザクションを確保できない場合があります。

デフォルトのタイムアウトの変更

ネットワークの待ち時間が長いと、UA および SA が要求と登録を行う場合、応答を受け取る前にタイムアウトになる原因になります。複数のサブネット、ダイアルアップ回線、または WAN によって UA が SA から切り離されている場合、または UA と SA の両方が DA から切り離されている場合、待ち時間が問題となることがあります。待ち時間が問題であるかどうかを判断するには、UA および SA の要求と登録でタイムアウトが起こったために SLP 要求が失敗しているかどうかを確認します。`ping` コマンドを使って実際の待ち時間を測定することもできます。

次の表は、タイムアウトを制御する構成プロパティーを示します。この節で説明する手順で、これらのプロパティーを変更できます。

表 9–4 タイムアウトプロパティー

プロパティー	説明

`net.slp.multicastTimeouts`
`net.slp.DADiscoveryTimeouts`
`net.slp.datagramTimeouts`
	これらのプロパティーは、メッセージ転送が中止されるまで、マルチキャストやユニキャストが繰り返し実行する UDP メッセージの転送に使用できるタイムアウトのリストを制御します。

`net.slp.multicastMaximumWait`	このプロパティーは、マルチキャストメッセージが中止されるまで、転送される最長時間を制御します。
`net.slp.datagramTimeouts`	このプロパティーに一覧表示される値の合計を示す DA タイムアウトの上限。UDP ダイアグラムは、応答を受け取るかタイムアウトの上限になるまで、DA に繰り返し送られます。

マルチキャストサービスの検出中または DA の検出中に頻繁にタイムアウトが発生する場合は、net.slp.multicastMaximumWait プロパティーをデフォルト値の 15000 ミリ秒 (15 秒) から増やしてください。最大待ち時間を長くすることにより、待ち時間の長いネットワーク上で要求に対してより長い時間が許可されます。net.slp.multicastMaximumWait プロパティーの値を増やしたあとは、net.slp.multicastTimeouts と net.slp.DADiscoveryTimeouts も変更する必要があります。これらのプロパティーのタイムアウト値の合計が net.slp.multicastMaximumWait 値と等しくなるようにしてください。

▼ デフォルトのタイムアウトの変更方法

次の手順に従って、タイムアウトを制御する SLP プロパティーを変更します。

1. スーパーユーザーになるか、同等の役割を引き受けます。

   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2. ホスト上の slpd とすべての **SLP** 動作を停止します。

   ```
   # svcadm disable network/slp
   ```

3. 構成の設定を変更する前に、デフォルトの /etc/inet/slp.conf ファイルのバックアップをとります。

4. slpd.conf ファイル内の net.slp.multicastMaximumWait プロパティーを変更します。

   net.slp.multicastMaximumWait=*value*

   value net.slp.multicastTimeouts と net.slp.DADiscoveryTimeouts に設定する値の合計値を示す、32 ビットの整数

   デフォルト値は、15000 ミリ秒 (15 秒) です

   値の範囲は、1000 から 60000 ミリ秒です

   たとえば、マルチキャスト要求で 20 秒 (20000 ミリ秒) 必要だと判断したら、net.slp.multicastTimeouts プロパティーと net.slp.DADiscoveryTimeouts プロパティーに一覧表示されている値の合計が 20000 ミリ秒になるように調整します。

   ```
   net.slp.multicastMaximumWait=20000
   net.slp.multicastTimeouts=2000,5000,6000,7000
   net.slp.DADiscoveryTimeouts=3000,3000,6000,8000
   ```

5. slpd.conf ファイル内の net.slp.datagramTimeouts プロパティーを必要に応じて変更します。

   net.slp.datagramTimeouts=*value*

value ユニキャストのデータグラム転送を DA に実行するためのタイムアウト (ミリ秒) を指定する、32 ビット整数のリスト

デフォルト値は、3000,3000,3000 です

たとえば、頻繁なタイムアウトの発生を回避するために、データグラムのタイムアウトを 20000 ミリ秒に増やすことができます。

```
net.slp.datagramTimeouts=2000,5000,6000,7000
```

高パフォーマンスのネットワークでは、逆に UDP データグラム転送のマルチキャストまたはユニキャストのタイムアウトの上限を小さくできます。タイムアウトの上限を小さくすることで、SLP 要求を満たすために必要な待ち時間を短縮できます。

6 変更を保存し、ファイルを閉じます。

7 変更を反映するには、`slpd` を再起動します。

```
# svcadm enable network/slp
```

ランダム待ち時間の上限の構成

トラフィックの重いネットワークや衝突率の高いネットワークでは、DA との通信が影響を受けることがあります。衝突率が高い場合、送信エージェントは、UDP データグラムを再転送する必要があります。再転送が発生しているかどうかは、`snoop` を使用して、SA サーバーとして `slpd` を実行しているホスト、および DA サーバーとして `slpd` を実行しているホストのネットワークトラフィックを監視することにより判断できます。SA サーバーとして `slpd` を実行しているホストから同じサービスについて複数のサービス登録メッセージが `snoop` トレースに現れる場合は、衝突の問題があると考えられます。

衝突は、ブート時の主要な問題となる場合があります。DA が最初に起動されると、DA は非要請通知を送り出し、SA はそれらの登録に応答します。SLP は、DA 通知を受け取ってから応答するまでにランダムな時間だけ、SA を待たせます。このランダムな待ち時間は、`net.slp.randomWaitBound` によって制御される最大値を使って均等に分散されます。デフォルトのランダム待ち時間の上限は 1000 ミリ秒 (1 秒) です。

▼ ランダム待ち時間の上限の構成方法

次の手順に従って、`slp.conf` ファイルの `net.slp.RandomWaitBound` プロパティーを変更します。

1. スーパーユーザーになるか、同等の役割を引き受けます。

   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2. ホスト上の `slpd` とすべての **SLP** 動作を停止します。

   ```
   # svcadm disable network/slp
   ```

3. 構成の設定を変更する前に、デフォルトの `/etc/inet/slp.conf` ファイルのバックアップをとります。

4. `slpd.conf` ファイル内の `net.slp.RandomWaitBound` プロパティーを変更します。

   `net.slp.RandomWaitBound=`*value*

   value　DA に接続するまでのランダム待ち時間の計算に使用される上限

   デフォルト値は、1000 ミリ秒 (1 秒) です

   値の範囲は、1000 から 3000 ミリ秒です

   たとえば、ランダム待ち時間を 2000 ミリ秒 (2 秒) に延長できます。

   ```
   net.slp.randomWaitBound=2000
   ```

   ランダム待ち時間の上限を長くすると、登録で遅延が長くなります。SA は新しく検出された DA をより時間をかけて登録できるので、衝突とタイムアウトを回避することができます。

5. `slpd.conf` ファイル内の `net.slp.datagramTimeouts` プロパティーを必要に応じて変更します。

   `net.slp.datgramTimeouts=`*value*

   value　ユニキャストのデータグラム転送を DA に実行するためのタイムアウト (ミリ秒) を指定する、32 ビット整数のリスト

   デフォルト値は、3000,3000,3000 です

   たとえば、頻繁なタイムアウトの発生を回避するために、データグラムのタイムアウトを 20000 ミリ秒に増やすことができます。

   ```
   net.slp.datagramTimeouts=2000,5000,6000,7000
   ```

   高パフォーマンスのネットワークでは、逆に UDP データグラム転送のマルチキャストまたはユニキャストのタイムアウトの上限を小さくできます。この設定により、SLP 要求を満たす際に、待ち時間を短縮できます。

6. 変更を保存し、ファイルを閉じます。

7 変更を反映するには、slpd を再起動します。

```
# svcadm enable network/slp
```

スコープの配置

スコープを使用すると、論理的、物理的、および管理上のユーザーのグループによるサービスへの対応が可能です。スコープを使用することで、サービス通知へのアクセスの管理が可能になります。

net.slp.useScopes プロパティーを使用すると、スコープを作成できます。たとえば、次のように構成すると、ホスト上の /etc/inet/slp.conf ファイルに、newscope という名前の新規のスコープが追加されます。

```
net.slp.useScopes=newscope
```

たとえば、プリンタや FAX などのネットワーク接続されたオフィス機器の小部屋が、会社の 6号棟 2 階の南側の廊下の突き当たりにあるとします。これらのオフィス機器は 2 階のすべてのユーザーに提供されている場合や、使用が特定の部署のメンバーに限定する場合があります。スコープはこれらの機器に対するサービス通知へのアクセスに対応する手段を提供します。

オフィス機器をマーケティング部専用にすると、mktg という名前のスコープを作成することができます。別の部署に所属しているオフィス機器は、別のスコープ名で構成できます。

また、部署が分散している場合もあります。たとえば、機械工学部門と CAD/CAM 部門が 1 階と 2 階に分かれているとします。この場合でも、両者に同じスコープを割り当てることにより、1 階と 2 階にあるホストに 2 階のマシンを提供できます。ネットワークとユーザーに都合よく動作するように、スコープはどのように配置してもかまいません。

注 – 特定のスコープを持つ UA は、別のスコープで通知されたサービスを実際に使用できないわけではありません。スコープの構成は、UA が検出するサービス通知を制御するだけです。サービス自体が、なんらかのアクセス制御の制限を行う必要があります。

スコープを構成する場合

SLP はスコープ構成をまったく行わなくても十分機能します。Solaris オペレーティング環境では、SLP のデフォルトのスコープは default です。構成されているスコープがない場合は、default がすべての SLP メッセージのスコープになります。

次の環境のどれかに当てはまれば、スコープを構成できます。

- サポートしている組織が、所属メンバーに対するサービス通知アクセスを制限する場合。
- サポートしている組織が、特定のユーザーが特定領域のサービスにアクセスするように物理的に配置されている場合。
- ユーザーが認識できるサービス通知を分割する必要がある場合。

最初の場合の例を248 ページの「ダイアルアップネットワークに対する DA 検出の構成」に挙げました。2 番目の例は、組織が 2 つの建物に分かれていて、1 つの建物のユーザーはその建物のローカルサービスにアクセスするようにする場合です。ビルディング 1 のユーザーはスコープ `B1` を使用して、ビルディング 2 のユーザーはスコープ `B2` を使って構成できます。

スコープを構成する場合の検討事項

`slpd.conf` ファイル内の `net.slp.useScopes` プロパティーを変更する場合は、ホスト上のすべてのエージェントにスコープを構成します。ホストが SA を実行している場合や DA として機能している場合に、その SA と DA に `default` 以外のスコープを構成するには、このプロパティーを構成する必要があります。UA だけがマシン上で動作し、UA が、`default` 以外のスコープをサポートしている SA と DA を検出する必要がある場合は、UA が使用するスコープを制限するのでなければ、プロパティーを構成する必要はありません。プロパティーを構成しない場合、UA は、`slpd` を通じて、使用可能な DA とスコープを自動的に検出します。SLP デーモンは、能動的および受動的 DA 検出を使用して DA を見つけるか、DA が動作していない場合は SA 検出を使用して DA を見つけます。プロパティーを構成する場合、UA は構成されたスコープを使用するだけで、構成されたスコープを破棄することはありません。

スコープを構成することを決定した場合は、ネットワーク内のすべての SA にスコープが構成されていることが確実でないかぎり、構成されたスコープのリスト上で `default` スコープを維持することを検討してください。構成されていない SA があると、構成されたスコープを持つ UA はそれらの SA を見つけることができません。これは、構成されていない SA が自動的に `default` スコープを持つのに対し、UA は構成されたスコープを持つためです。

`net.slp.DAAddresses` プロパティーを設定して DA も構成しようとする場合は、構成される DA によってサポートされるスコープが、`net.slp.useScopes` プロパティーで構成したスコープと同じであることを確認してください。スコープが同じでない場合は、再起動時に `slpd` がエラーメッセージを出力します。

▼ スコープの構成方法

次の手順に従って、スコープ名を `slp.conf` ファイルの `net.slp.useScopes` プロパティーに追加します。

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 ホスト上の `slpd` とすべての **SLP** 動作を停止します。

```
# svcadm disable network/slp
```

3 構成の設定を変更する前に、デフォルトの `/etc/inet/slp.conf` ファイルのバックアップをとります。

4 `slpd.conf` ファイル内の `net.slp.useScopes` プロパティーを変更します。

`net.slp.useScopes=<`*scope names*`>`

scope names 文字列のリストで、DA または SA が要求時に使用を許されるスコープを示すか、DA がサポートする必要があるスコープを示す

デフォルトの値は、SA と DA の場合は Default、UA の場合は未設定

注 –

スコープ名は、次の文法上のガイドラインに従って構成します。

- 大文字または小文字の英数字
- 句読点 (", \, !, <, =, >, および ~ を除く)
- 名前の一部と考えられるスペース
- 非 ASCII 文字

  ASCII でない文字をエスケープするには、バックスラッシュを使用します。たとえば、UTF-8 コード体系では、フランス語の *aigue* アクセントのある文字 *e* を表すために、16 進コード `0xc3a9` を使用します。プラットフォームが UTF-8 をサポートしていない場合は、UTF-8 の 16 進コード `\c3\a9` をエスケープシーケンスとして使用します。

たとえば、`bldg6` で `eng` および `mktg` グループ用のスコープを指定するには、`net.slp.useScopes` 行を次のように変更します。

```
net.slp.useScopes=eng,mktg,bldg6
```

5 変更を保存し、ファイルを閉じます。

6 変更を反映するには、slpd を再起動します。

```
# svcadm enable network/slp
```

DA の配置

この節では、SLP を実行しているネットワークでの計画的な DA の配置について説明します。

配置された DA または構成されたスコープがなくても、基本のエージェントである UA と SA だけで SLP は十分機能します。特定の構成を持たないすべてのエージェントは自動的に default スコープを使用します。DA はサービス通知のキャッシュとして機能します。DA を配置すると、ネットワークに送られるメッセージ数が削減されるため、メッセージ応答の受け取りに必要な時間も短縮されます。これにより、SLP をより大規模なネットワークに対応させることができます。

SLP DA を配置する理由

DA を配置する主な目的は、サービス検出によって生じるマルチキャストトラフィックの量とユニキャスト応答の収集に関係する遅延を削減することです。多くの UA および SA を持つ大規模なネットワークでは、サービス検出によって生じるマルチキャストの量が非常に大きくなるので、ネットワークのパフォーマンスが下がります。1 つまたは複数の DA を配置すると、UA はサービスについて DA にユニキャストし、SA はユニキャストを使用して DA に登録する必要があります。DA を使用したネットワークでは、SLP 登録されたマルチキャストは、能動的および受動的 DA 検出のマルチキャストだけです。

SA は、マルチキャストのサービス要求を受け取るのではなく、共通のスコープのセット内で検出した任意の DA に自動的に登録します。ただし、DA がサポートしていないスコープ内のマルチキャスト要求には、SA が直接応答します。

UA から出されたサービス要求は、UA のスコープ内に DA が配置されている場合は、ネットワーク上へのマルチキャストではなく DA に対するユニキャストです。そのため、UA のスコープ内に DA を配置すると、マルチキャストが削減されます。通常の UA 要求を行うマルチキャストをなくすことにより、クエリー応答の受け取りに必要な時間が秒単位からミリ秒単位に大幅に縮小します。

DA は SA および UA の動作の中心として機能します。スコープの集合に対して 1 つまたは複数の DA を配置することにより、SLP の動作を監視するための集中的なポイントが提供されます。DA ログを起動することにより、ネットワークに散在している複

数の SA から取り寄せたログをチェックするよりも、登録および要求の監視が容易になります。負荷を均等にする必要に合わせて、1 つまたは複数の特定のスコープに対して DA をいくつでも配置できます。

マルチキャストルーティングが使用できないネットワークでは、SLP がブロードキャストを使用するように構成できます。しかし、ブロードキャストは各ホストにメッセージを処理するように要求するため、非常に効率が悪くなります。また、ブロードキャストは通常、ルーターを超えて伝達されません。この結果、マルチキャストルーティングに対応していないネットワークでは、同じサブネットでしかサービスを検出できません。マルチキャストルーティングに一部しか対応していない場合は、ネットワーク上でサービスを検出する機能に矛盾が生じます。マルチキャストメッセージは DA の検出に使用されます。したがって、マルチキャストルーティングに一部しか対応していない場合は、UA と SA はサービスを SA のスコープ内にある既知の DA に登録することが暗黙の了解になっています。たとえば、UA が `DA1` と呼ばれる DA にクエリーを出し、SA がサービスを `DA2` に登録している場合、UA はサービスの検出に失敗します。マルチキャストが使用できないネットワーク上の SLP の配置については、255 ページの「ブロードキャスト専用ルーティングの構成」を参照してください。

サイト全体がマルチキャストルーティングに対応していないネットワークでは、`net.slp.DAAdresseses` プロパティーを使用して、SLP の UA と SA が DA 位置に関して矛盾のないリストを持つように構成する必要があります。

最後に、Solaris SLPv2 の DA は SLPv1 との相互運用性をサポートしています。SLPv1 相互運用性は、Solaris DA ではデフォルトにより有効になっています。ネットワークにプリンタなどの SLPv1 デバイスが接続されている場合、またはサービス検出で SLPv1 を使用している Novell Netware 5 と相互運用する必要がある場合、DA を配置する必要があります。DA が配置されていないと、Solaris SLP の UA は SLPv1 によって通知されたサービスを見つけることができません。

DA を配置する場合

次の条件のどれかに当てはまる場合は、エンタープライズに DA を配置します。

- `snoop` で測定した、ネットワーク上での SLP のマルチキャストのトラフィックが帯域幅の 1% を超える。
- UA クライアントがサービス要求のマルチキャスト中に長時間遅延またはタイムアウトする。
- 1 台または複数台のホスト上にある特定のスコープに対して、SLP サービス通知の監視を集中する。
- ネットワークが、サービスを共有する複数のサブネットから構成され、マルチキャストに対応していない。

- ネットワークが前バージョンの SLP (SLPv1) をサポートするデバイスを使用している、または SLP サービス検出で Novell Netware 5 と相互運用したい。

▼ DA を配置する方法

次の手順に従って、`slp.conf` ファイルの `net.slp.isDA` プロパティーに `True` を設定します。

注 – 1 台のホストにつき 1 つの DA だけが割り当てられます。

---

1. スーパーユーザーになるか、同等の役割を引き受けます。

   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2. ホスト上の `slpd` とすべての **SLP** 動作を停止します。

   ```
   # svcadm disable network/slp
   ```

3. 構成の設定を変更する前に、デフォルトの `/etc/inet/slp.conf` ファイルのバックアップをとります。

4. `slpd.conf` ファイル内の `net.slp.isDA` プロパティーに `True` を設定します。

   ```
   net.slp.isDA=True
   ```

5. 変更を保存し、ファイルを閉じます。

6. 変更を反映するには、`slpd` を再起動します。

   ```
   # svcadm enable network/slp
   ```

DA を配置する場所

この節は、DA を配置する場所について状況ごとにヒントを示します。

- マルチキャストルーティングが使用できず、DA がサブネット間のサービス検出をブリッジする必要がある場合

  この場合は、インタフェースとサービスを共有するすべてのサブネットを持つホスト上に DA を配置してください。IP パケットがインタフェースの間を経路指定されない場合を除き、`net.slp.interfaces` 構成プロパティーを設定する必要はありません。`net.slp.interfaces` プロパティーの設定については、267 ページの「SLP に対するマルチホームの構成」を参照してください。

- DA が拡張に備えて配置されており、考慮すべき主要な事柄がエージェントのアクセスの最適化である場合

  UA は通常、DA に対してサービスを大量に要求します。SA がサービスを DA に登録すると、SA は通知を定期的に適切な頻度で更新できます。その結果、UA から DA へのアクセスの方が SA のアクセスよりはるかに頻繁になります。通常、サービス通知の数も要求の数より小さくなります。このため、UA のアクセスに対して DA の配置が最適化されている場合、多くの DA を配置することは効率化をうながします。
- UA のアクセスを最適化するために、ネットワーク上でトポロジ的に UA の近くになるように DA を配置する場合

  UA クライアントと SA クライアントの両方が共有しているスコープを使用して、DA を構成してください。

複数の DA を配置して負荷を均等にする

負荷を均等にする手段として、同じスコープの集合体について複数の DA を配置できます。次の状況のどれかに当てはまれば、DA を配置できます。

- DA に対する UA 要求がタイムアウトしているか、あるいは `DA_BUSY_NOW` エラーが返っている。
- DA ログが、多くの SLP 要求が欠落していることを示す。
- スコープ内でサービスを共有しているユーザーのネットワークが、複数の建物や物理的なサイトに渡っている。

SLP トラフィックの `snoop` トレースを行うことによって、どれくらいの UA 要求で `DA_BUSY_NOW` エラーが返されるかを判断することができます。返される UA 要求の数が多い場合は、DA から物理的およびトポロジ的に離れている建物内の UA は、応答が遅かったり過度にタイムアウトしたりすることがあります。このような場合、建物内の UA クライアントの応答を改善するために、建物ごとに DA を配置できます。

建物間を接続しているリンクは、建物内のローカルエリアネットワークよりも遅いことがあります。ネットワークが複数の建物または物理的なサイトに渡っている場合は、`/etc/inet/slp.conf` ファイル内の `net.slp.DAAddresses` プロパティーを特定のホスト名またはアドレスのリストに設定して、指定した DA だけに UA がアクセスするようにします。

特定の DA がサービス登録に対して大量のホストメモリーを消費している場合は、DA がサポートするスコープ数を減らすことによって、SA 登録の数を削減します。登録数の多いスコープを、たとえば 2 つのスコープに分けることができます。次に、片方の DA を別のホストに配置することによって、もう片方のスコープだけをサポートするようにできます。

SLP とマルチホーム

マルチホームサーバーは、複数の IP サブネット上でホストとして機能します。そのようなサーバーに複数のネットワークインタフェースカードが装着されると、ルーターとして機能できます。マルチキャストパケットを含む IP パケットは、このインタフェース間を経路指定されます。場合によっては、インタフェース間の経路指定ができないことがあります。この節では、そのような場合に SLP を構成する方法について説明します。

SLP に対するマルチホームの構成

構成を行わない場合、`slpd` はデフォルトのネットワークインタフェース上でマルチキャストと UDP/TCP ユニキャストに対して待機しています。ユニキャストルーティングとマルチキャストルーティングがマルチホームマシンのインタフェース間で使用できる場合は、追加の構成を行う必要はありません。追加の構成が必要ないのは、別のインタフェースに到達するマルチキャストパケットがデフォルトで正確に経路指定されているからです。その結果、DA またはほかのサービス通知のマルチキャスト要求は、`slpd` に届きます。経路指定がなんらかの理由で調整されていない場合は、構成が必要です。

経路指定されていない複数のネットワークインタフェースに対して構成を行う場合

マルチホームマシンの構成が必要と考えられるのは、主に次の場合です。

- ユニキャストルーティングはインタフェース間で使用できるが、マルチキャストルーティングは使用できない。
- ユニキャストルーティングとマルチキャストルーティングの両方がインタフェース間で使用できない。

マルチキャストルーティングがインタフェース間で使用できない場合は、通常、マルチキャストがネットワークに配置されていないことが原因です。この場合は通常、それぞれのサブネット上の DA ベースでないサービス検出および DA 検出にブロードキャストが使用されます。ブロードキャストは、`net.slp.isBroadcastOnly` プロパティーを `True` に設定することによって構成します。

経路指定されていない複数のネットワークインタフェースの構成 (作業マップ)

表 9–5 経路指定されていない複数のネットワークインタフェースの構成

作業	説明	参照先
net.slp.interfaces プロパティーを構成します	このプロパティーを設定することで、slpd は、指定されたインタフェース上でユニキャストとマルチキャスト/ブロードキャストの SLP 要求を待機できます。	268 ページの「net.slp.interfaces プロパティーの構成」
サブネット上の UA が到達可能なアドレスを持つサービス URL を取得できるように、プロキシサービス通知を配置します	マルチホームホストではなく単一のサブネットに接続された slpd を実行しているマシンにプロキシ通知を限定します。	270 ページの「マルチホームホスト上のプロキシ通知」
UA と SA 間で確実に到達できるように DA を配置してスコープを構成します	マルチホーム上の net.slp.interfaces プロパティーを単一インタフェースのホスト名またはアドレスで構成します。 マルチホームホスト上で DA を実行し、各サブネット上の SA と UA は別のホストを使用するように構成します。	271 ページの「DA の配置とスコープ名の割り当て」

net.slp.interfaces プロパティーの構成

net.slp.interfaces プロパティーが設定されている場合、slpd は、ユニキャストとマルチキャスト/ブロードキャストの SLP 要求を、デフォルトのインタフェース上ではなく、プロパティーに一覧表示されたインタフェース上で待機します。

通常、net.slp.interfaces プロパティーを設定すると同時に、net.slp.isBroadcastOnly プロパティーも設定することでブロードキャストを有効にします。ただし、マルチキャストは配置されているが、この特定のマルチホームホスト上で経路指定されていない場合、マルチキャスト要求は、複数のインタフェースから slpd に到達できます。このような状況は、パケットの経路指定が、別のマルチホームホストまたはインタフェースからサービスを受けるサブネットに接続されているルーターによって制御されている場合に起こります。

この場合、SA サーバーまたは要求を送っている UA は、マルチホームホストの slpd から 2 つの応答を受け取ります。これらの応答はクライアントライブラリによってフィルタにかけられて除かれるので、クライアントには見えません。ただし、この応答は、snoop トレースで見ることができます。

注 –

ユニキャストルーティングがオフになっている場合、マルチホームホスト上の SA クライアントによるサービス通知がすべてのサブネットに到達できないことがあります。サービスが到達できない場合、SA クライアントは次のことを実行できます。

- 個々のサブネットにつき 1 つのサービス URL を通知する。
- 特定のサブネットからの要求が到達可能な URL で確実に応答されるようにする。

SA クライアントライブラリには、到達可能な URL が確実に通知されるようにするためのしくみはありません。したがって、到達可能な URL が確実に通知されるようにするには、経路指定のないマルチホームホストを処理できるかどうかにかかわらず、サービスプログラムに任せる必要があります。

ユニキャストルーティングが無効なマルチホームホストにサービスを配置する前に、`snoop` を使ってサービスが複数のサブネットから出された要求を正確に処理しているかどうかを判断してください。さらに、マルチホームホストに DA を配置することを計画している場合は、271 ページの「DA の配置とスコープ名の割り当て」を参照してください。

▼ `net.slp.interfaces` プロパティーの構成方法

次の手順に従って、`slp.conf` ファイルの `net.slp.interfaces` プロパティーを変更します。

1. スーパーユーザーになるか、同等の役割を引き受けます。

   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2. ホスト上の `slpd` とすべての **SLP** 動作を停止します。

   ```
   # svcadm disable network/slp
   ```

3. 構成の設定を変更する前に、デフォルトの `/etc/inet/slp.conf` ファイルのバックアップをとります。

4. `slpd.conf` ファイル内の `net.slp.interfaces` プロパティーを変更します。

   `net.slp.interfaces=`*value*

   value　IPv4 アドレスまたはネットワークインタフェースカードのホスト名のリストで、そこに存在する DA や SA はポート 427 上でマルチキャスト、ユニキャスト UDP、および TCP の各メッセージを待機する必要がある

   たとえば、3 枚のネットワークカードを持ち、マルチキャストルーティングがオフになっているサーバーが、3 つのサブネットに接続されているとします。その 3 つのネットワークインタフェースの IP アドレスは `192.147.142.42`、`192.147.143.42`、および `192.147.144.42` です。サブネットマスクは `255.255.255.0` です。次のプロパ

ティーの設定を行うと、slpd はユニキャストおよびマルチキャストまたはブロードキャストのメッセージについて、3 つすべてのインタフェース上のものに対して待機します。

```
net.slp.interfaces=192.147.142.42,192.147.143.42,192.147.144.42
```

注 – net.slp.interfaces プロパティーには、IP アドレスまたは解決可能なホスト名を設定できます。

---

5 変更を保存し、ファイルを閉じます。

6 変更を反映するには、slpd を再起動します。

```
# svcadm enable network/slp
```

マルチホームホスト上のプロキシ通知

複数のインタフェースを持つホストが slpd およびプロキシ登録を使ってサービスを通知する場合は、slpd によって通知されるサービス URL に到達可能なホスト名またはアドレスが含まれている必要があります。インタフェース間でユニキャストルーティングが有効な場合は、すべてのサブネット上のホストは別のサブネット上のホストに到達できます。任意のサブネット上のサービスに対してプロキシ登録も行うことができます。ただし、ユニキャストルーティングが無効な場合は、1 つのサブネット上のサービスクライアントはマルチホームホストを通じて別のサブネット上のサービスに到達することはできません。ただし、これらのクライアントが別のルーターを通じてサービスに到達できる可能性はあります。

たとえば、デフォルトのホスト名が bigguy のホストが、経路指定されていない異なる 3 つのサブネット上に 3 枚のインタフェースカードを持っているとします。これらのサブネット上のホスト名は、IP アドレス 192.147.142.42 を持つ bigguy、IP アドレス 192.147.143.42 を持つ bigguy1、IP アドレス 192.147.144.42 を持つ bigguy2 です。ここで、レガシープリンタ oldprinter がサブネット 143 に接続され、すべてのインタフェース上で待機するために、URL service:printing:lpr://oldprinter/queue1 が net.slp.interfaces で構成されているとします。oldprinter の URL はすべてのインタフェース上でプロキシ通知されます。サブネット 142 と 144 上のマシンは、サービス要求に対する応答でこの URL を受信しますが、oldprinter サービスにアクセスすることはできません。

この問題の解決方法は、マルチホームホスト上ではなく、サブネット 143 だけに接続されたマシン上で動作している slpd を使ってプロキシ通知を行うことです。サブネット 143 上のホストだけがサービス要求に対する応答でこの通知を取得できます。

DA の配置とスコープ名の割り当て

マルチホームホストを持つネットワーク上で DA の配置とスコープ名の割り当てを行う場合は、クライアントがアクセス可能なサービスを確実に取得できるように注意してください。経路指定が無効で net.slp.interfaces プロパティーが構成されている場合は特に注意してください。また、マルチホームマシン上のインタフェース間でユニキャストルーティングが有効な場合は、特別な DA やスコープを構成する必要はありません。これは、DA にキャッシュされている通知が任意のサブネットからアクセス可能なサービスを識別するためです。ただし、ユニキャストルーティングが無効な場合は、DA をうまく配置しないと問題になることがあります。

前述の例で何が問題になりうるかを見るために、bigguy が DA を実行し、すべてのサブネット上のクライアントが同じスコープを持つ場合に何が起こるかを考えてみます。サブネット 143 上の SA はサービス通知を DA に登録します。サブネット 144 上の UA は、サブネット 143 上のホストに到達できなくても、それらのサービス通知を入手できます。

この問題の 1 つの解決方法は、マルチホームホスト上ではなく、各サブネット上で DA を実行することです。この場合は、マルチホームホスト上の net.slp.interfaces プロパティーを、単一のインタフェースホスト名またはアドレスを使って構成するか、構成しないでそのままにし、強制的にデフォルトのインタフェースを使用するようにします。この解決方法の欠点は、通常大規模なマシンであり、DA として高機能であるマルチホームホストを DA に設定できないことです。

もう 1 つの解決方法は、マルチホームホスト上で DA を実行するが、各サブネット上の SA および UA が異なるスコープを持つようにスコープを構成することです。たとえば、前述の場合、142 サブネット上の UA と SA がスコープ scope142 を持つようにスコープを構成することができます。143 サブネット上の UA と SA は、scope143 という別のスコープを持ち、144 サブネット上の UA と SA は scope144 という別のスコープを持つように構成することができます。3 つのインタフェースを持つ bigguy 上の net.slp.interfaces プロパティーを構成して、DA を 3 つのサブネット上の 3 つのスコープに作用させることができます。

経路指定されていない複数のネットワークインタフェースを構成する場合の検討事項

net.slp.interfaces プロパティーを構成すると、マルチホームホスト上の DA がサブネット間のサービス通知をブリッジできるように設定できます。このような構成は、ネットワークでマルチキャストルーティングがオフで、マルチホームホスト上のインタフェース間でユニキャストルーティングが有効な場合に便利です。ユニキャストはインタフェース間を経路指定しているため、サービスが置かれているサブネットと異なるサブネット上のホストは、サービス URL を受信すればそのサービ

スに接続することができます。DA がない場合は、特定のサブネット上の SA サーバーが同じサブネット上に出されたブロードキャストだけを受信するので、そのサブネット以外にサービスを置くことはできません。

`net.slp.interfaces` プロパティーの構成が必要な場合は、マルチキャストがネットワークに配置されておらず、代わりにブロードキャストが使用されている場合です。その他の場合は、不必要な応答の重複や到達できないサービスを避けるために、入念に検討および計画を行なってください。

◆ ◆ ◆ 第 10 章

10 レガシーサービスの組み込み

レガシーサービスとは、SLP の開発および実装が旧式になっているネットワークサービスのことです。たとえば、ラインプリンタデーモン (`lpsched`)、NFS ファイルサービス、NIS や NIS+ ネームサービスなどの Solaris サービスは、SLP で使用する内部 SA を持っていません。この章では、レガシーサービスを通知する場合とその方法について説明します。

レガシーサービスを通知する場合

レガシーサービス通知では、SLP UA を使用可能にすることで、ネットワーク上の次のようなデバイスやサービスを検出できます。SLP SA を含まないハードウェアデバイスやソフトウェアサービスを検出できます。たとえば、SLP UA を持つアプリケーションが、SLP SA を含まないプリンタやデータベースを検出する必要がある場合、レガシー通知が必要になります。

レガシーサービスの通知

レガシーサービスは、次の方法で通知できます。

- SLP SA を組み込むようにサービスを変更する。
- SLP が有効でないサービスの代わりにサービスを通知する小さなプログラムを書く。
- プロキシ通知を使用して、`slpd` にサービスを通知させる。

サービスの変更

ソフトウェアサーバーのソースコードを使用できる場合は、SLP SA を組み込むことができます。SLP 用の C 言語の API と Java の API は比較的簡単に使用できます。C 言語の API のマニュアルページと Java の API のマニュアルを参照してください。サービスがハードウェアデバイスの場合は、製造元が SLP を組み込む PROM を更新していることがあります。詳細は、デバイスの製造元に問い合わせてください。

SLP が使用できないサービスの通知

ソースコードや更新された SLP を含む PROM が使用できない場合は、SLP クライアントライブラリを使ってサービスを通知する小さなアプリケーションを書くことができます。このアプリケーションは小さなデーモンとして機能し、サービスの起動・停止に使用する場合と同じシェルスクリプトで起動・停止します。

SLP プロキシ登録

Solaris の `slpd` は、プロキシ登録ファイルを使用したレガシーサービスの通知をサポートしています。プロキシ通知ファイルは、移植性のあるフォーマットで書かれたサービス通知のリストです。

▼ SLP プロキシ登録を有効にする方法

1. ホストのファイルシステムまたは **HTTP** でアクセス可能なネットワーク上の任意のディレクトリに、プロキシ登録ファイルを作成します。

2. サービスについてサービスタイプのテンプレートが存在するかどうかを確認します。

   テンプレートは、サービスタイプのサービス URL と属性を記述したものです。テンプレートを使用して、特定のサービスタイプについて通知の構成要素を定義します。

   - サービスタイプテンプレートが存在する場合は、そのテンプレートを使ってプロキシ登録を構成してください。サービスタイプテンプレートについては、RFC 2609 を参照してください。
   - サービスについてサービスタイプテンプレートを使用できない場合は、サービスを正確に記述する属性の集合体を選択してください。通知に対して、デフォルト以外の命名権限を使用してください。デフォルトの命名権限は標準化されたサービスタイプについてだけ許可されています。命名権限については、RFC 2609 を参照してください。

たとえば、*BizApp* という会社にソフトウェアバグの追跡に使用されるローカルデータベースがあるとします。データベースを通知するために、この会社は、サービスタイプ `service:bugdb.bizapp` を持つ URL を使用します。この場合、命名権限は `bizapp` になります。

3 前の手順で作成した登録ファイルの場所を使用して、`/etc/inet/slp.conf` ファイルの `net.slp.serializedRegURL` プロパティーを構成するには、次の手順に従います。

4 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

5 ホスト上の `slpd` とすべての **SLP** 動作を停止します。

```
# svcadm disable network/slp
```

6 構成の設定を変更する前に、デフォルトの `/etc/inet/slp.conf` ファイルのバックアップをとります。

7 `/etc/inet/slp.conf` ファイルの `net.slp.serializedRegURL` プロパティーにプロキシ登録の場所を指定します。

`net.slp.net.slp.serializedRegURL=`*proxy registration file URL*

たとえば、直列化登録ファイルが `/net/inet/slp.reg` である場合、プロパティーを次に示すように構成します。

```
net.slp.serializedRegURL=file:/etc/inet/slp.reg
```

8 変更を保存し、ファイルを閉じます。

9 変更を反映するには、`slpd` を再起動します。

```
# svcadm enable network/slp
```

SLP プロキシ登録による通知

サービス通知は、サービス URL を特定する行、オプションのスコープ行、一連の属性の定義から構成されます。SLP デーモンはファイルからプロキシ通知を読み、その通知を登録し、SA クライアントと同じようにそれらを保持します。次のリストは、プロキシ登録ファイルの例を示します。

この例では、LPR プロトコルをサポートするレガシープリンタと ftp サーバーが通知されています。行番号は説明のために付け加えたもので、実際のファイルには記述されていません。

```
(1) #Advertise legacy printer.
(2)
(3) service:lpr://bizserver/mainspool,en,65535
(4) scope=eng,corp
(5) make-model=Laserwriter II
(6) location-description=B16-2345
(7) color-supported=monochromatic
(8) fonts-supported=Courier,Times,Helvetica 9 10
(9)
(10) #Advertise FTP server
(11)
(12) ftp://archive/usr/src/public,en,65535,src-server
(13) content=Source code for projects
(14)
```

注 – プロキシ登録ファイルは、ASCII でない文字のエスケープに、構成ファイルと同じ取り決めを使用します。プロキシ登録ファイルのフォーマットについては、RFC 2614 を参照してください。

表 10–1 SLP プロキシ登録ファイルの説明

行番号	説明
1 と 10	シャープ記号 (#) で始まるコメント行で、ファイルの動作には影響しません。コメント行の最後まですべての文字が無視されます。
2、9、14	通知の区切りを示す空行。
3、12	3 つの必須フィールドと 1 つのオプションフィールドがコンマで区切られたサービス URL。 ■ 一般的な URL または `service: URL` が通知されます。service: URL の指定方法の仕様については、 RFC 2609 を参照してください。 ■ 通知の言語を指定します。前述の例では、フィールドは英語 *en* を指定しています。この言語は RFC 1766 の言語タグです。 ■ 登録の有効期限を秒単位で規定します。有効期限は符号なしの 16 ビット整数に限定されます。有効期限が最大値 65535 より小さい場合、`slpd` は通知をタイムアウトします。有効期限が 65535 の場合、`slpd` は定期的に通知を更新し、`slpd` が存在するかぎり有効期限は永続するとされます。 ■ サービスタイプフィールド (省略可能) – サービスタイプの定義に使用します。サービス URL が定義されている場合は、URL が通知されるサービスタイプを変更できます。前述のプロキシ登録ファイルの例では、12 行目に一般的な FTP URL が含まれています。オプションのタイプフィールドを使用して、この URL をサービスタイプ名 *src-server* で通知できます。デフォルトでは `service` 接頭辞はタイプ名には付きません。

表 10–1 SLP プロキシ登録ファイルの説明 (続き)

行番号	説明
4	スコープの指定。 オプション行はトークン scope と等号、およびコンマで区切られたスコープ名のリストで構成されます。このスコープ名は、net.slp.useScopes 構成プロパティーで定義されています。ホストに構成されたスコープだけが、このスコープリストに表示されます。スコープ行がない場合は、slpd が構成されているすべてのスコープに登録が行われています。スコープ行は URL 行のすぐあとになければなりません。その他の場所にある場合、スコープ名は属性として認識されます。
5 から 8	属性の定義。 オプションのスコープ行のあとは、サービス通知の大部分は属性と値リストのペアの行で構成されます。各ペアは属性タグ、等号、コンマで区切られた属性値のリスト (属性が単一値の場合は単一値) で構成されます。前述のプロキシ登録ファイルの例では、8 行目が複数の値を持つ属性リストを示しています。これ以外の値リストはすべて単一値を持っています。属性名および値のフォーマットは、ネットワークを通過する SLP メッセージと同じです。

レガシーサービスを通知する場合の検討事項

通常、SLP を追加する場合、ほかのサービスの代理として SLP API で通知する SLP 対応のサービスを書くよりも、ソースコードを変更する方が望ましい方法です。ソースコードの変更は、プロキシ登録を使用するよりも望ましい方法です。ソースコードを変更する場合、サービス固有の機能を追加したり、サービスの使用可否を綿密に追跡したりできます。ソースコードが使用できない場合は、プロキシ登録を使用するよりほかのサービスの代理として通知する SLP 対応のヘルパーサービスを書く方が望ましい方法です。このヘルパーサービスを、起動と停止の制御に使用されるサービスの開始または停止手順に組み込むことをお勧めします。プロキシ通知は通常、ソースコードが使用できず、スタンドアロンの SA を書くことが実際的ではない場合の 3 番目の選択肢です。

プロキシ通知は、プロキシ登録ファイルを読み取る slpd が動作している間だけ保持されます。プロキシ通知とサービスの間には直接的な関係はありません。通知がタイムアウトしたり slpd が停止したりすると、プロキシ通知は使用できなくなります。

サービスが停止した場合は、slpd を停止する必要があります。直列化登録ファイルを編集してプロキシ通知をコメントにするか削除し、slpd を再起動してください。サービスを再起動または再インストールしたときは同じ手順に従ってください。プロキシ通知とサービスの間に関係のないことがプロキシ通知の主な欠点です。

◆ ◆ ◆ 第 11 章

11 SLP (リファレンス)

この章では、SLP のステータスコードとメッセージタイプについて説明します。SLP のメッセージタイプは、省略形と機能コードを示します。SLP のステータスコードは、説明と機能コードを示します。ステータスコードは、該当する要求を受信しているか (コード 0)、受信側がビジーであるかを示します。

注 – SLP デーモン (`slpd`) は、ユニキャストメッセージに対してだけステータスコードを返します。

SLP のステータスコード

表 11–1 SLP のステータスコード

ステータスのタイプ	ステータスコード	説明
No Error	0	要求はエラーなしで処理されました。
LANGUAGE_NOT_SUPPORTED	1	AttrRqst または SrvRqst について、スコープ内にサービスタイプのデータがありますが、指定された言語ではありません。
PARSE_ERROR	2	メッセージが SLP 構文に従っていません。
INVALID_REGISTRATION	3	SrvReg に問題があります。たとえば、有効期限がゼロである、言語タグが欠けているなど。
SCOPE_NOT_SUPPORTED	4	SLP メッセージが、要求に応える SA または DA がサポートするスコープリスト内のスコープを含んでいませんでした。
AUTHENTICATION_UNKNOWN	5	DA または SA がサポートしていない SLP SPI に対する要求を受信しました。

表 11–1 SLP のステータスコード (続き)

ステータスのタイプ	ステータスコード	説明
AUTHENTICATION_ABSENT	6	UA または DA が SrvReg において URL および属性認証を要求しましたが受信しませんでした。
AUTHENTICATION_FAILED	7	UA または DA が認証ブロックにおいて認証エラーを検出しました。
VER_NOT_SUPPORTED	9	メッセージでサポートしていないバージョン番号。
INTERNAL_ERROR	10	DA または SA で未知のエラーが発生しました。たとえば、オペレーティングシステムがファイルスペースを使い果たしたなど。
DA_BUSY_NOW	11	UA または SA は、急増するバックオフを使用して再試行する必要があります。DA が他のメッセージの処理でビジー状態です。
OPTION_NOT_UNDERSTOOD	12	DA または SA が必須の範囲から未知のオプションを受信しました。
INVALID_UPDATE	13	DA が登録されていないサービスに対して、FRESH 設定なしで、あるいは矛盾するサービスタイプで、SrvReg を受信しました。
MSG_NOT_SUPPORTED	14	SA が AttrRqst または SrvTypeRqst を受信しましたが、サポートしていません。
REFRESH_REJECTED	15	SA が DA に対して、DA の最短更新間隔よりも頻繁に SrvReg または SrvDereg の一部を送りました。

SLP のメッセージタイプ

表 11–2 SLP のメッセージタイプ

メッセージタイプ	略語	機能コード	説明
サービス要求	`SrvRqst`	1	サービスを検出するために UA が発行します。あるいは、能動的 DA 検出において UA あるいは SA サーバーが発行します。
サービス応答	`SrvRply`	2	DA あるいは SA がサービス要求に対して応答します。
サービス登録	`SrvReg`	3	SA が新規の通知を登録したり、既存の通知を新規の属性および変更された属性で更新したり、URL の有効期限を更新できるようにしたりします。

表 11–2 SLP のメッセージタイプ (続き)

メッセージタイプ	略語	機能コード	説明
サービス登録解除	SrvDereg	4	表しているサービスが無効になった場合にその通知の登録を解除するために SA が使用します。
確認応答	SrvAck	5	SA のサービス要求またはサービス登録解除メッセージに対する DA の応答。
属性要求	AttrRqst	6	URL またはサービスタイプが作成し、属性のリストを要求します。
属性応答	AttrRply	7	属性のリストを返す場合に使用されます。
DA 通知	DAAdvert	8	サービス要求をマルチキャストするための DA の応答。
サービスタイプ要求	SrvTypeRqst	9	特定の命名権限を持ち、特定のスコープセットにある登録されたサービスタイプについて問い合わせるために使用されます。
サービスタイプ応答	SrvTypeRply	10	サービスタイプ要求に対する応答として返されるメッセージ。
SA 通知	SAAdvert	11	DA が配置されていないネットワークで、UA は SAAdvert を使用して SA およびそのスコープを検出します。

パート IV

メールサービス(トピック)

このパートでは、メールサービスの概要、作業、およびリファレンス情報について説明します。

◆ ◆ ◆ 第 12 章

12 メールサービス (概要)

電子メールサービスの設定と維持管理には、ネットワークの日常の運用にとって不可欠な、複雑な作業が伴います。ネットワーク管理者は、既存のメールサービスを拡張しなければならない場合があります。または、新しいネットワークまたはサブネット上でメールサービスを設定しなければならないこともあります。メールサービスに関する各章では、ネットワークでメールサービスを計画したり設定したりするために必要な情報を提供します。この章では、sendmail の新機能の説明へのリンクを用意し、参考資料を紹介します。この章ではまた、メールサービスを確立するために必要なソフトウェアおよびハードウェアコンポーネントの概要を説明します。

第 13 章「メールサービス (手順)」では、メールサービスの設定および管理方法の手順を説明します。詳細は、291 ページの「メールサービス (作業マップ)」を参照してください。

メールサービスのコンポーネントについての詳細は、第 14 章「メールサービス (リファレンス)」を参照してください。この章では、メールサービスのプログラムとファイル、メールルーティング処理、ネームサービスを使った sendmail の対話型操作、sendmail version 8.13 の機能についても説明します。380 ページの「sendmail の version 8.13 での変更点」を参照してください。

メールサービスの新機能

この節では、さまざまな Solaris リリースの新機能について説明します。

このリリースでの変更点

Solaris 10 7/10 リリース リリースでは、次の変更点が加えられました。

- sendmail のデフォルトバージョンが 8.14 に更新されました。
- 従来のデーモン (`svc:/network/smtp:sendmail`) およびクライアントキューランナー (`svc:/network/smtp:sendmail-client`) の管理を改善するため、sendmail インスタンスが 2 つのインスタンスに分割されました。
- `sendmail.cf` および `submit.mc` 構成ファイルが自動的に再構築されるように、システムを構成可能になりました。必要な手順については、307 ページの「構成ファイルを自動的に再構築する方法」を参照してください。
- デフォルトでは、sendmail デーモンは新しいローカルデーモンモードで動作します。ローカル専用モードでは、ローカルホストやループバック SMTP 接続からの着信メールだけを受信します。たとえば、cron ジョブからのメールやローカルユーザー間のメールを受信します。発信メールの経路は変更されず、着信メールだけが変更されます。ローカル専用モードを選択する場合には、`-bl` (Become Local モードの略) オプションを使用します。このモードの詳細は、`sendmail(1M)` のマニュアルページを参照してください。`-bd` (Become Daemon モード) に戻す方法については、308 ページの「オープンモードで `sendmail` を使用する方法」を参照してください。

Solaris 10 1/06 リリースでの変更点

Solaris 10 1/06 以降のリリースでは、`sendmail` は Transport Layer Security (TLS) を使用した SMTP をサポートしています。詳細については、次を参照してください。

- 381 ページの「`sendmail` の version 8.13 で TLS を使用して SMTP を実行するためのサポート」
- 309 ページの「TLS を使用するよう SMTP を構成する」

Solaris 06 10/9 リリースに含まれる新機能の完全な一覧については、『Oracle Solaris 10 9/10 の新機能』を参照してください。

Solaris 10 リリースでの変更点

Solaris 10 以降のリリースでは、`sendmail` version 8.13 がデフォルトになっています。version 8.13 に関する情報とほかの変更点については、次を参照してください。

- 342 ページの「`sendmail` のコンパイルに使用できるフラグと使用できないフラグ」
- 343 ページの「`MILTER` (`sendmail` のメールフィルタ API)」
- 344 ページの「構成ファイルのバージョン」
- 356 ページの「`vacation` ユーティリティーの拡張機能」

- 359 ページの「/etc/mail/cf ディレクトリの内容」
- 380 ページの「sendmail の version 8.13 での変更点」
- 390 ページの「sendmail の version 8.12 からの TCP ラッパーのサポート」

さらに、メールサービスは、サービス管理機能によって管理されています。このサービスに関する有効化、無効化、再起動などの管理アクションは svcadm コマンドを使用して実行できます。サービスの状態は、svcs コマンドを使用して照会できます。サービス管理機能の詳細は、smf(5) のマニュアルページおよび『Solaris のシステム管理 (基本編)』の第 18 章「サービスの管理 (概要)」を参照してください。

sendmail のその他の情報

次に、上記以外の sendmail 関連の参考資料を示します。

- Costales、Bryan 著、『sendmail, Third Edition』、O'Reilly & Associates, Inc.、2002
- sendmail 関連のホームページ – http://www.sendmail.org
- sendmail 関連の FAQ – http://www.sendmail.org/faq
- 新しい sendmail 構成ファイルの README – http://www.sendmail.org/m4/readme.html
- sendmail の最新 Sun バージョンへの移行ガイド – http://www.sendmail.org/vendor/sun/

メールサービスのコンポーネントの概要

メールサービスを確立するためには、多くのソフトウェアコンポーネントおよびハードウェアコンポーネントが必要になります。次では、これらのコンポーネントについて簡単に紹介します。コンポーネントの説明で使用する用語についても紹介します。

最初の節 287 ページの「ソフトウェアコンポーネントの概要」では、メール配信システムのソフトウェア部分を説明するのに使用する用語を定義します。その次の節 288 ページの「ハードウェアコンポーネントの概要」では、メール構成におけるハードウェアシステムの機能について取り上げます。

ソフトウェアコンポーネントの概要

次の表にメールシステムのソフトウェアコンポーネントを示します。ソフトウェアコンポーネントすべてに関する詳細については、345 ページの「ソフトウェアコンポーネント」を参照してください。

コンポーネント	説明
`.forward` ファイル	ユーザーのホームディレクトリ内で設定して、メールを自動的にリダイレクトしたり、プログラムに送ったりすることができるファイル
メールボックス	メールサーバー上にあり、電子メールメッセージの最終受信先であるファイル
メールアドレス	メールメッセージが配信される受信者とシステムの名称を含むアドレス
メール別名	メールアドレス内で使用されている代替名
メールキュー	メールサーバーによる処理を必要とするメールメッセージの集まり
ポストマスター	メールサービスについての問題を報告し質問を出すために使用される特別なメール別名
sendmail 構成ファイル	メールのルーティングに必要なすべての情報の入ったファイル

ハードウェアコンポーネントの概要

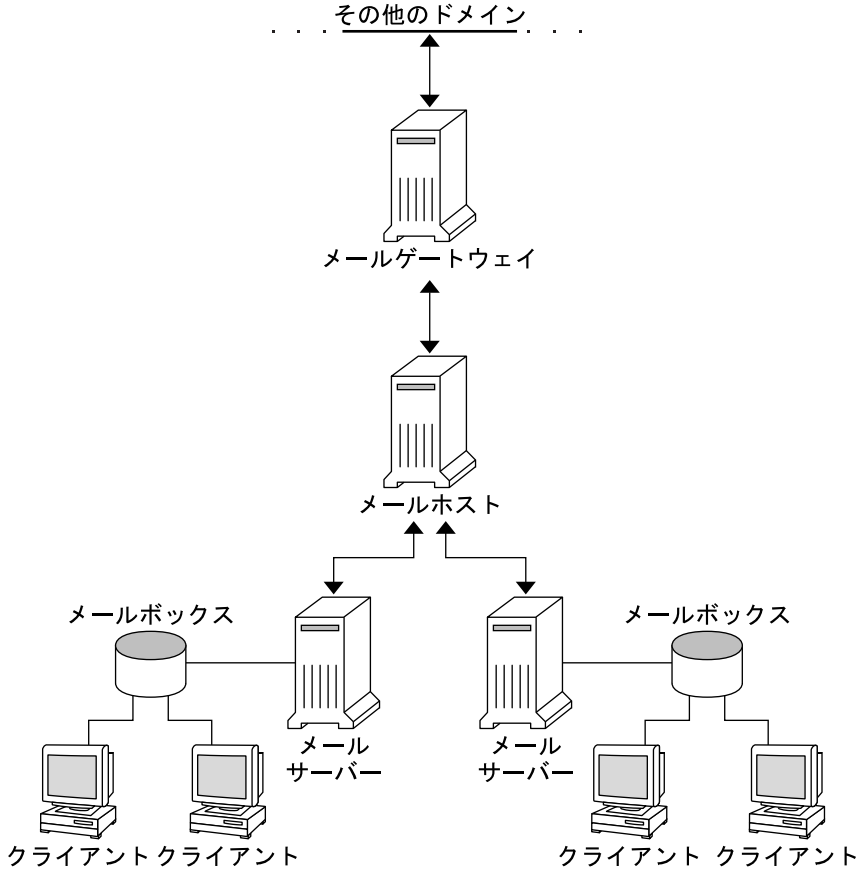
メール構成では次の 3 つの要素が必要ですが、これらは同じシステムで組み合わせることも、別のシステムで提供することもできます。

- メールホスト – 解釈処理が困難なメールアドレスを扱うように構成されたシステム
- 少なくとも 1 台のメールサーバー – 1 つまたは複数のメールボックスを保持するように構成されたシステム
- メールクライアント – メールサーバーからメールにアクセスするシステム

ユーザーがドメイン外のネットワークと通信をするためには、4 番目の要素であるメールゲートウェイを追加する必要があります。

図 12–1 には、一般的な電子メール構成を示しますが、ここでは基本的な 3 つのメール要素とメールゲートウェイが使用されています。

図 12–1　一般的な電子メール構成



各要素については、353 ページの「ハードウェアコンポーネント」を参照してください。

◆ ◆ ◆ 第 13 章

メールサービス (手順)

この章ではメールサービスを設定し、管理する方法について説明します。メールサービスの管理に詳しくない場合は、メールサービスのコンポーネントを紹介している第 12 章「メールサービス (概要)」を参照してください。この章では、一般的なメールサービス構成についても説明しています (図 12–1 を参照)。この章では、次の関連作業について説明します。

- 291 ページの「メールサービス (作業マップ)」
- 296 ページの「メールサービスの設定 (作業マップ)」
- 305 ページの「sendmail 構成の変更 (作業マップ)」
- 315 ページの「メール別名ファイルの管理 (作業マップ)」
- 327 ページの「キューディレクトリの管理 (作業マップ)」
- 330 ページの「.forward ファイルの管理 (作業マップ)」
- 333 ページの「メールサービスの障害対処とヒント (作業マップ)」

メールサービスのコンポーネントについての詳細は、第 14 章「メールサービス (リファレンス)」を参照してください。また、この章では、メールサービスのプログラムとファイル、メールルーティング処理、ネームサービスを使った sendmail の対話型操作、sendmail(1M) のマニュアルページで十分に説明されていない sendmail の version 8.13 での機能についても説明します。

メールサービス (作業マップ)

次の表から、具体的な一連の手順を扱っているほかの作業マップがわかります。

作業	説明	参照先
メールサービスの設定	メールサービスの各コンポーネントを設定する手順。メールサーバー、メールクライアント、メールホスト、およびメールゲートウェイの設定方法について説明します。`sendmail` で DNS を利用する方法について説明します。	296 ページの「メールサービスの設定 (作業マップ)」
sendmail 構成ファイルの変更	構成ファイルまたはサービスプロパティーを変更する手順。	305 ページの「sendmail 構成の変更 (作業マップ)」
メール別名ファイルの管理	ネットワークで別名を提供するための手順。NIS+ テーブルのエントリの管理方法を説明します。また、NIS マップ、ローカルメール別名、キー付きマップファイル、およびポストマスター別名の設定方法も説明します。	315 ページの「メール別名ファイルの管理 (作業マップ)」
メールキューの管理	スムーズなキュー処理を提供するための手順。メールキューを表示したり移動したりする方法、強制的なメールキュー処理方法、およびメールキューのサブセットの実行方法について説明します。古いメールキューの実行方法についても説明します。	327 ページの「キューディレクトリの管理 (作業マップ)」
`.forward` ファイルの管理	`.forward` ファイルを無効にしたり、`.forward` ファイルの検索パスを変更したりする手順。`/etc/shells` を作成し生成することにより、`.forward` ファイルの使用をユーザーに許可する方法も説明します。	330 ページの「`.forward` ファイルの管理 (作業マップ)」
メールサービスのトラブルシューティング手順とヒント	メールサービスで発生した問題を解決するための手順とヒント。メール構成のテスト、メール別名の確認、`sendmail` ルールセットのテスト、ほかのシステムへの接続の確認、メッセージの記録などの方法について学びます。ほかのメール診断情報の情報源も紹介します。	333 ページの「メールサービスの障害対処とヒント (作業マップ)」
エラーメッセージの解釈処理	メール関連のエラーメッセージを解釈処理するための情報。	338 ページの「エラーメッセージの解釈」

メールシステムの計画

次に、メールシステムを計画するときに考慮すべき点を挙げます。

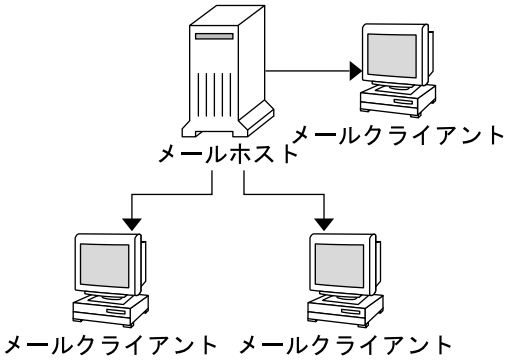
- 必要に応じてメール構成のタイプを決定します。この節では、メール構成の基本の 2 タイプについて説明し、各構成を設定するために必要なことがらについて簡単に説明します。新しいメールシステムを設定する必要がある場合、あるいは既存のメールシステムを拡張する場合は、この節の内容が役立つでしょう。293 ページの「ローカルメール専用」では 1 番目の構成タイプについて、294 ページの「ローカルメールとリモート接続」では 2 番目の構成タイプについて説明します。
- 必要に応じてメールサーバー、メールホスト、およびメールゲートウェイとして動作するシステムを選択します。
- サービスを提供するすべてのメールクライアントのリストを作成し、メールボックスの場所も含めます。このリストは、ユーザーのメール別名を作成するときに役立ちます。
- 別名の更新方法とメールメッセージの転送方法を決めます。ユーザーがメールの転送要求を送る場所として、`aliases` メールボックスを設定できます。ユーザーはこのメールボックスを使って、デフォルトのメール別名の変更要求を送ることもできます。システムで NIS または NIS+ を使用する場合、メール転送の管理は、ユーザー自身ではなく、管理者が行うこともできます。315 ページの「メール別名ファイルの管理 (作業マップ)」に、別名に関連する作業の一覧があります。330 ページの「`.forward` ファイルの管理 (作業マップ)」に、`.forward` ファイルの管理に関連する作業の一覧があります。

メールシステムの計画を立てたら、サイトにシステムを設定し、296 ページの「メールサービスの設定 (作業マップ)」で説明する機能を実行します。ほかの作業については、291 ページの「メールサービス (作業マップ)」を参照してください。

ローカルメール専用

図 13–1 に示すように、もっとも単純なメール構成は、1 台のメールホストに 2 台以上のワークステーションが接続されている場合です。メールは完全にローカルです。すべてのクライアントがローカルのディスクにメールを格納し、すべてのクライアントがメールサーバーとして機能します。メールアドレスは `/etc/mail/aliases` ファイルを使って構文解析されます。

図 13–1 ローカルメール構成



この種類のメール構成を設定するには、次が必要です。

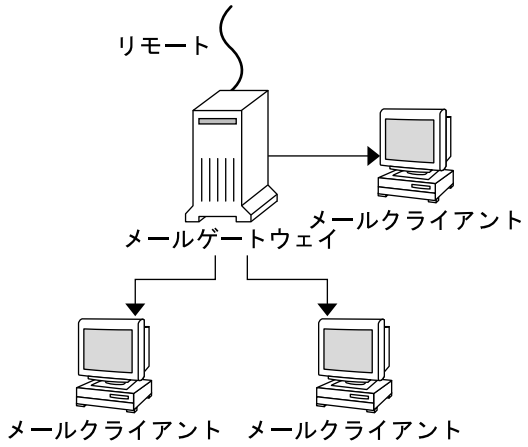
- 各メールクライアントシステム上に、デフォルトの `/etc/mail/sendmail.cf` ファイル。編集は不要です。
- メールホストとして指定されたサーバー。NIS または NIS+ を実行している場合に、この指定を行うには、メールホスト上の `/etc/hosts` ファイルに `mailhost.` *domain-name* を追加します。DNS や LDAP など、別のネームサービスを実行している場合は、`/etc/hosts` ファイルに追加情報を入力します。301 ページの「メールホストを設定する方法」を参照してください。
- NIS や NIS+ 以外のネームサービスを使用している場合は、ローカルメールボックスのあるすべてのシステム上に、対応する `/etc/mail/aliases` ファイルが必要です。
- 各メールクライアントシステムの `/var/mail` に、メールボックスを格納できるだけの十分な領域。

メールサービスの設定の詳細については、296 ページの「メールサービスを設定する」を参照してください。メールサービスの設定に関する特定の手順については、296 ページの「メールサービスの設定 (作業マップ)」を参照してください。

ローカルメールとリモート接続

小規模のネットワークにおけるもっとも一般的なメール構成を図 13–2 に示します。1 つのシステムが、メールサーバー、メールホスト、およびリモート接続を行うメールゲートウェイを兼ねています。メールは、メールゲートウェイ上の `/etc/mail/aliases` ファイルを使って配布されます。ネームサービスは必要ありません。

図 13–2　UUCP 接続を使ったローカルメール構成



この構成では、メールクライアントがメールホスト上の /var/mail からメールファイルをマウントすると想定できます。この種類のメール構成を設定するには、次が必要です。

- 各メールクライアントシステム上に、デフォルトの /etc/mail/sendmail.cf ファイル。このファイルは編集不要です。
- メールホストとして指定されたサーバー。NIS または NIS+ を実行している場合に、この指定を行うには、メールホスト上の /etc/hosts ファイルに mailhost. *domain-name* を追加します。DNS や LDAP など、別のネームサービスを実行している場合は、/etc/hosts ファイルに追加情報を入力します。301 ページの「メールホストを設定する方法」を参照してください。
- NIS や NIS+ 以外のネームサービスを使用している場合は、ローカルメールボックスのあるすべてのシステム上に、対応する /etc/mail/aliases ファイルが必要です。
- メールサーバーの /var/mail に、クライアントのメールボックスを格納できるだけの十分な領域。

メールサービスの設定の詳細については、296 ページの「メールサービスを設定する」を参照してください。メールサービスの設定に関する特定の手順については、296 ページの「メールサービスの設定 (作業マップ)」を参照してください。

メールサービスの設定 (作業マップ)

次の表では、メールサービスの設定の手順を説明します。

作業	説明	参照先
メールサーバーを設定する	サーバーがメールを経路指定できるようにする手順	297 ページの「メールサーバーを設定する方法」
メールクライアントを設定する	ユーザーがメールを受信できるようにする手順	299 ページの「メールクライアントを設定する方法」
メールホストを設定する	電子メールアドレスを解釈処理できるメールホストを確立する手順	301 ページの「メールホストを設定する方法」
メールゲートウェイを設定する	ドメイン外のネットワークとの通信を管理する手順	302 ページの「メールゲートウェイを設定する方法」
`sendmail` で DNS を使用する	DNS ホストルックアップ機能を有効にする手順	304 ページの「`sendmail` で DNS を使用する方法」

メールサービスを設定する

サイトが企業外の電子メールサービスに接続していないか、あるいは企業が 1 つのドメイン内にある場合は、メールサービスを比較的容易に設定できます。

ローカルメール用に 2 つのタイプの構成が必要です。これらの構成については、図 13–1の293 ページの「ローカルメール専用」 を参照してください。ドメイン外のネットワークと通信するためには、さらに 2 つのタイプの構成が必要です。これらの構成については、図 12–1の288 ページの「ハードウェアコンポーネントの概要」 または 図 13–2の294 ページの「ローカルメールとリモート接続」 を参照してください。これらの構成は、同じシステムで組み合わせるか、または別のシステムで提供できます。たとえば、同じシステムにメールホストとメールサーバーの機能を持たせる場合は、この節の説明に従って、まずそのシステムをメールホストとして設定します。次に、この節の説明に従って、同じシステムをメールサーバーとして設定します。

注 – 次のメールサーバーとメールクライアントの設定の手順は、メールボックスが NFS でマウントされているときに適用されます。ただし、メールボックスは通常、ローカルにマウントされた `/var/mail` ディレクトリで維持されるので、次の手順は必要ありません。

▼ メールサーバーを設定する方法

メールサーバーはローカルユーザーにメールサービスを提供するだけなので、設定には特別な手順は必要ありません。ユーザーはパスワードファイルまたは名前空間にエントリが必要です。さらに、メールが配信されるためには、ユーザーはローカルのホームディレクトリを用意して、`~/.forward` ファイルを確認する必要があります。このため、ホームディレクトリサーバーがしばしばメールサーバーとして設定されます。メールサーバーについては、353 ページの「ハードウェアコンポーネント」の 第 14 章「メールサービス (リファレンス)」でさらに詳しく説明します。

メールサーバーは、メールクライアント宛てにメールを経路指定します。このタイプのメールサーバーは、クライアントのメールボックス用に十分なスプール空間が必要です。

注 – `mail.local` プログラムは、メッセージがはじめて配信された時に `/var/mail` ディレクトリでメールボックスを自動的に作成します。メールクライアントの個々のメールボックスを作成する必要はありません。

クライアントが自分のメールボックスにアクセスするには、`/var/mail` ディレクトリをリモートマウントに利用できなければなりません。または、POP (Post Office Protocol)、IMAP (Internet Message Access Protocol) などのサービスをサーバーから利用できなければなりません。次では、`/var/mail` ディレクトリを使ってメールサーバーを設定する方法を示します。このマニュアルでは、POP または IMAP の構成方法については説明しません。

次の作業のために、`/var/mail` ディレクトリがエクスポートされていることを `/etc/dfs/dfstab` ファイルで確認します。

1. スーパーユーザーになるか、同等の役割を引き受けます。

   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2. `sendmail` を停止します。

   ```
   # svcadm disable -t network/smtp:sendmail
   ```

3. `/var/mail` ディレクトリをリモートアクセスに使用できるかどうかを確認します。

   ```
   # share
   ```

   `/var/mail` ディレクトリが表示された場合は、手順 5 に進みます。

/var/mail ディレクトリが表示されない場合、あるいはリストが表示されない場合は、該当する手順に進みます。

a. (省略可能) リストが表示されない場合は、**NFS** サービスを起動します。

85 ページの「ファイルシステム自動共有を設定する方法」の手順に
従って、/var/mail ディレクトリを使用して NFS サービスを起動します。

b. (省略可能) /var/mail ディレクトリがリストに含まれていない場合は、/var/mail ディレクトリを **/etc/dfs/dfstab** に追加します。

/etc/dfs/dfstab ファイルに次のコマンド行を追加します。

```
share -F nfs -o rw /var/mail
```

4 ファイルシステムをマウントできるようにします。

```
# shareall
```

5 ネームサービスが起動されていることを確認します。

a. (省略可能) **NIS** を実行している場合は、次のコマンドを使用します。

```
# ypwhich
```

詳細は、ypwhich(1) のマニュアルページを参照してください。

b. (省略可能) **NIS+** を実行している場合は、次のコマンドを使用します。

```
# nisls
```

詳細は、nisls(1) のマニュアルページを参照してください。

c. (省略可能) **DNS** を実行している場合は、次のコマンドを使用します。

```
# nslookup hostname
```

hostname　　ホスト名を指定します。

詳細は、nslookup(1M) のマニュアルページを参照してください。

d. (省略可能) **LDAP** を実行している場合は、次のコマンドを使用します。

```
# ldaplist
```

詳細は、ldaplist(1) のマニュアルページを参照してください。

6 sendmail を再起動します。

```
# svcadm enable network/smtp:sendmail
```

▼ メールクライアントを設定する方法

メールクライアントは、メールサーバー上にメールボックスを持っている、メールサービスのユーザーです。メールクライアントにはさらに、/etc/mail/aliases ファイルで、メールボックスの位置を示すメール別名が設定されています。

注 – POP (Post Office Protocol) または IMAP (Internet Message Access Protocol) のようなサービスを使ってメールクライアントを設定することもできます。ただし、POP または IMAP の構成方法については、このマニュアルでは説明していません。

---

1 メールクライアントシステム上でスーパーユーザーになるか、同等の役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 sendmail を停止します。

```
# svcadm disable -t network/smtp:sendmail
```

3 メールクライアントのシステムで /var/mail マウントポイントがあることを確認します。

マウントポイントは、インストール過程で作成されています。ls を使用すると、ファイルシステムが存在するかどうかを確認できます。次の例はファイルシステムが作成されていない場合に受け取る応答を示しています。

```
# ls -l /var/mail
/var/mail not found
```

4 /var/mail ディレクトリにファイルが何もないことを確認します。

メールファイルがこのディレクトリにある場合は、それらのファイルを移動させ、サーバーから /var/mail ディレクトリがマウントされるときにその対象とならないようにします。

5 メールサーバーから /var/mail ディレクトリをマウントします。

メールディレクトリは自動的にマウントすることも、ブート時にマウントすることもできます。

a. (省略可能) /var/mail を自動的にマウントします。

次のようなエントリを /etc/auto_direct ファイルに追加します。

```
/var/mail -rw,hard,actimeo=0 server:/var/mail
```

server 割り当てられているサーバー名を指定します。

b. (省略可能) ブート時に /var/mail をマウントします。

/etc/vfstab ファイルに次のエントリを追加します。このエントリにより、指定されたメールサーバー上の /var/mail ディレクトリがローカルの /var/mail ディレクトリをマウントできます。

server:/var/mail - /var/mail nfs - no rw,hard,actimeo=0

システムをリブートするたびに、クライアントのメールボックスが自動的にマウントされます。システムをリブートしない場合は、次のコマンドを入力すれば、クライアントのメールボックスをマウントできます。

```
# mountall
```

注意 – メールボックスのロックとメールボックスへのアクセスが適切に動作するには、NFS サーバーからメールをマウントするときに actimeo=0 オプションを入れる必要があります。

6 /etc/hosts を更新します。

/etc/hosts ファイルを編集し、メールサーバーのエントリを追加します。ネームサービスを使用する場合、この手順は必要ありません。

```
# cat /etc/hosts
#
# Internet host table
#
..
IP-address      mailhost   mailhost   mailhost.example.com
```

IP-address 割り当てられている IP アドレスを指定します。

example.com 割り当てられているドメインを指定します。

mailhost 割り当てられているメールホストを指定します。

詳細は、hosts(4) のマニュアルページを参照してください。

7 別名ファイルの 1 つにクライアントのエントリを追加します。

メール別名ファイルの管理に関する作業マップについては、315 ページの「メール別名ファイルの管理 (作業マップ)」を参照してください。mail.local プログラムは、メッセージがはじめて配信された時に /var/mail ディレクトリでメールボックスを自動的に作成します。メールクライアントの個々のメールボックスを作成する必要はありません。

8 sendmail を再起動します。

```
# svcadm enable network/smtp:sendmail
```

▼ メールホストを設定する方法

メールホストは、電子メールアドレスを解決し、ドメイン内でメールを再度ルーティングします。メールホストに適しているのは、ネットワークにリモート接続を提供するシステム、または親ドメインにネットワークを接続するシステムです。次に、メールホストを設定する手順を示します。

1 メールホストシステム上でスーパーユーザーになるか、同等の役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 `sendmail` を停止します。

```
# svcadm disable -t network/smtp:sendmail
```

3 ホスト名の構成を確認します。

次のように `check-hostname` スクリプトを実行し、`sendmail` が、このサーバーの完全指定のホスト名を識別できるかどうかを確認します。

```
% /usr/sbin/check-hostname
hostname phoenix OK: fully qualified as phoenix.example.com
```

このスクリプトで完全指定ホスト名が識別できなかった場合は、完全指定ホスト名をホストの最初の別名として `/etc/hosts` 内に追加する必要があります。

4 `/etc/hosts` ファイルを更新します。

次から、適切な手順を選択します。

a. **(省略可能) NIS または NIS+ を使用している場合は、新しいメールホストとなるシステムの `/etc/hosts` ファイルを編集します。**

メールホストシステムの IP アドレスとシステム名のあとに `mailhost` と `mailhost.`*domain* を追加します。

IP-address *mailhost* `mailhost mailhost.`*domain* `loghost`

IP-address 割り当てられている IP アドレスを指定します。

mailhost メールホストシステムのシステム名を指定します。

domain 拡張ドメイン名を指定します。

これで、このシステムはメールホストとして指定されます。*domain* は、次のコマンドの出力にサブドメイン名として指定されている文字列と同じにする必要があります。

```
% /usr/lib/sendmail -bt -d0 </dev/null
Version 8.13.1+Sun
 Compiled with: LDAPMAP MAP_REGEX LOG MATCHGECOS MIME7TO8 MIME8TO7
```

```
                NAMED_BIND NDBM NETINET NETINET6 NETUNIX NEWDB NIS
                NISPLUS QUEUE SCANF SMTP USERDB XDEBUG

============ SYSTEM IDENTITY (after readcf) ============
      (short domain name) $w = phoenix
  (canonical domain name) $j = phoenix.example.com
         (subdomain name) $m = example.com
               (node name) $k = phoenix
========================================================
```

以上の変更を行なったあとの hosts ファイルの例を次に示します。

```
# cat /etc/hosts
#
# Internet host table
#
172.31.255.255   localhost
192.168.255.255  phoenix mailhost mailhost.example.com loghost
```

b. **(省略可能) NIS または NIS+ を使用しない場合は、ネットワーク内の各システムにある /etc/hosts ファイルを編集します。**

次のようなエントリを作成します。

IP-address mailhost mailhost *mailhost.domain* loghost

5 sendmail を再起動します。

```
# svcadm enable network/smtp:sendmail
```

6 メール構成をテストします。

手順については、334 ページの「メール構成をテストする方法」を参照してください。

注 – メールホストの詳細は、353 ページの「ハードウェアコンポーネント」の第 14 章「メールサービス (リファレンス)」を参照してください。

▼ メールゲートウェイを設定する方法

メールゲートウェイは、ドメイン外のネットワークとの通信を管理します。送信側メールゲートウェイ上のメールプログラムは、受信側システムのメールプログラムと同じでなければなりません。

メールゲートウェイに適しているのは、Ethernet および電話回線に接続されているシステムです。インターネットへのルーターとして設定されているシステムも適しています。メールホストをメールゲートウェイとして設定するか、あるいは別のシステムをメールゲートウェイとして設定できます。複数のメールゲートウェイを自分のドメイン用として設定できます。UUCP (UNIX-to-UNIX Copy Program) 接続がある場合は、メールゲートウェイとして UUCP 接続を使ってシステムを構成します。

1. メールゲートウェイ上でスーパーユーザーになるか、同等の役割になります。

   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2. sendmail を停止します。

   ```
   # svcadm disable -t network/smtp:sendmail
   ```

3. ホスト名の構成を確認します。

   次のように check-hostname スクリプトを実行し、sendmail が、このサーバーの完全指定のホスト名を識別できるかどうかを確認します。

   ```
   # /usr/sbin/check-hostname
   hostname phoenix OK: fully qualified as phoenix.example.com
   ```

   このスクリプトで完全指定ホスト名が識別できなかった場合は、完全指定ホスト名をホストの最初の別名として /etc/hosts 内に追加する必要があります。この手順の詳細は、手順 4の301 ページの「メールホストを設定する方法」を参照してください。

4. ネームサービスが起動されていることを確認します。

   a. (省略可能) NIS を実行している場合は、次のコマンドを使用します。

      ```
      # ypwhich
      ```

      詳細は、ypwhich(1) のマニュアルページを参照してください。

   b. (省略可能) NIS+ を実行している場合は、次のコマンドを使用します。

      ```
      # nisls
      ```

      詳細は、nisls(1) のマニュアルページを参照してください。

   c. (省略可能) DNS を実行している場合は、次のコマンドを使用します。

      ```
      # nslookup hostname
      ```

      hostname　　ホスト名を指定します。

      詳細は、nslookup(1M) のマニュアルページを参照してください。

   d. (省略可能) LDAP を実行している場合は、次のコマンドを使用します。

      ```
      # ldaplist
      ```

      詳細は、ldaplist(1) のマニュアルページを参照してください。

5. sendmail を再起動します。

   ```
   # svcadm enable network/smtp:sendmail
   ```

6 メール構成をテストします。

手順については、334 ページの「メール構成をテストする方法」を参照してください。

注 – メールゲートウェイの詳細は、353 ページの「ハードウェアコンポーネント」の第 14 章「メールサービス (リファレンス)」を参照してください。

▼ sendmail で DNS を使用する方法

DNS ネームサービスは、個別の別名をサポートしません。このネームサービスは、MX (メール交換局) レコードおよび CNAME レコードを使用するホストまたはドメインの別名をサポートします。ホスト名とドメイン名は両方またはいずれか一方を DNS データベースで指定できます。sendmail と DNS の詳細は、375 ページの「sendmail とネームサービスの相互作用」の第 14 章「メールサービス (リファレンス)」、または『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』を参照してください。

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 **DNS ホストルックアップ機能を有効にします (NIS+ のみ)。**

/etc/nsswitch.conf ファイルを編集し、dns フラグを含む hosts の定義から # を削除します。 DNS ホスト別名を使用するには、次の例に示すように、ホストエントリに dns フラグが含まれている必要があります。

```
# grep hosts /etc/nsswitch.conf
#hosts:     nisplus [NOTFOUND=return] files
hosts:      dns nisplus [NOTFOUND=return] files
```

3 mailhost と mailhost.*domain* エントリを確認します。

nslookup を使用して、mailhost と mailhost. *domain* のエントリが DNS データベースに存在することを確認します。詳細は、nslookup(1M) のマニュアルページを参照してください。

sendmail 構成の変更 (作業マップ)

作業	説明	参照先
sendmail 構成ファイルの構築	sendmail.cf ファイルを変更する手順。例としてドメインマスカレードを有効にする方法を取り上げます。	305 ページの「新しい sendmail.cf ファイルを構築する方法」
仮想ホストの設定	メールが複数のドメインに受け入れられるように sendmail を設定する手順。	307 ページの「仮想ホストを設定する」
sendmail 構成ファイルの自動再構築の設定	アップグレード後に sendmail.cf および submit.mc 構成ファイルが自動的に再構築されるように sendmail サービスを変更する手順。	307 ページの「構成ファイルを自動的に再構築する方法」
sendmail のオープンモードでの実行	オープンモードが有効になるように sendmail サービスのプロパティーを変更する手順。	308 ページの「オープンモードで sendmail を使用する方法」
Transport Layer Security (TLS) を使用する SMTP の設定	SMTP を有効にして TLS との接続をセキュリティー保護する手順。	309 ページの「TLS を使用するよう SMTP を構成する」
代替構成を使用したメール配信の管理	マスターデーモンが無効な場合に発生する可能性があるメール配信上の問題を防ぐための手順。	314 ページの「sendmail.cf の代替構成を使ってメール配信を管理する方法」

sendmail 構成を変更する

305 ページの「新しい sendmail.cf ファイルを構築する方法」で、構成ファイルの構築方法について説明します。sendmail.cf ファイルの以前のバージョンも引き続き使用できますが、新しい形式を使用することをお薦めします。

詳細は、次を参照してください。

- /etc/mail/cf/README。構成手順の詳細な説明です。
- http://www.sendmail.org。sendmail 構成に関するオンライン情報です。
- 344 ページの「構成ファイルのバージョン」の 367 ページの「sendmail 構成ファイル」と第 14 章「メールサービス (リファレンス)」。いくつかのガイダンスを示します。
- 396 ページの「sendmail の version 8.12 から追加または改訂された m4 構成マクロ」

▼ 新しい sendmail.cf ファイルを構築する方法

次に、新しい構成ファイルを構築する手順を示します。

注 – /usr/lib/mail/cf/main-v7sun.mc は、/etc/mail/cf/cf/sendmail.mc になりました。

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 sendmail を停止します。

```
# svcadm disable -t network/smtp:sendmail
```

3 変更しようとする構成ファイルのコピーを作成します。

```
# cd /etc/mail/cf/cf
# cp sendmail.mc myhost.mc
```

myhost .mc ファイルの新しい名前を指定します。

4 必要に応じて、新しい構成ファイル (たとえば、*myhost*.mc) を編集します。

たとえば、ドメインマスカレードを有効にするには、次のコマンド行を追加します。

```
# cat myhost.mc
..
MASQUERADE_AS(‘host.domain’)
```

host.domain 目的のホスト名とドメイン名を指定します。

この例では、MASQUERADE_AS は、送信されたメールに、$j ではなく *host.domain* から送信されたものとしてラベルを付けます。

5 m4 を使って構成ファイルを構築します。

```
# /usr/ccs/bin/make myhost.cf
```

6 -C オプションを使用して、新しい構成ファイルをテストし、新しいファイルを指定します。

```
# /usr/lib/sendmail -C myhost.cf -v testaddr </dev/null
```

このコマンドはメッセージを表示するとともに、メッセージを testaddr に送信します。システム上で sendmail サービスを再起動せずに、送信メールだけがテストできます。まだメールを処理していないシステムでは、334 ページの「メール構成をテストする方法」で説明する完全なテスト手順を使用してください。

7 オリジナルのコピーを作成したあと、新しい構成ファイルをインストールします。

```
# cp /etc/mail/sendmail.cf /etc/mail/sendmail.cf.save
# cp myhost.cf /etc/mail/sendmail.cf
```

8 sendmail サービスを再起動します。

```
# svcadm enable network/smtp:sendmail
```

仮想ホストを設定する

ホストに複数の IP アドレスを割り当てる必要がある場合は、http://www.sendmail.org/tips/virtualHosting の Web サイトを参照してください。このサイトでは、sendmail を使って仮想ホストを設定する方法を詳しく説明しています。ただし、「Sendmail Configuration」の節では、次に示す手順 3b は実行しないでください。

```
# cd sendmail-VERSION/cf/cf
# ./Build mailserver.cf
# cp mailserver.cf /etc/mail/sendmail.cf
```

代わりに、Solaris オペレーティングシステムでは、次の手順を実行してください。

```
# cd /etc/mail/cf/cf
# /usr/ccs/bin/make mailserver.cf
# cp mailserver.cf /etc/mail/sendmail.cf
```

mailserver .cf ファイルの名前を指定します。

305 ページの「sendmail 構成を変更する」では、構築手順の一部として、これと同じ 3 つの手順を説明しています。

/etc/mail/sendmail.cf ファイルを生成したら、次の手順に進み、仮想ユーザーテーブルを作成できます。

▼ 構成ファイルを自動的に再構築する方法

sendmail.cf または submit.cf のコピーを独自に構築済みであれば、アップグレード時に構成ファイルが置き換えられることはありません。次の手順は、sendmail.cf ファイルが自動的に再構築されるように sendmail サービスのプロパティーを構成する方法を示します。submit.cf 構成ファイルを自動的に再構築する方法については、例 13–1 を参照してください。両方のファイルの構築が必要な場合には、これらの手順を組み合わることもできます。

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 **sendmail プロパティーを設定します。**

```
# svccfg -s sendmail
svc:/network/smtp:sendmail> setprop config/path_to_sendmail_mc=/etc/mail/cf/cf/myhost.mc
svc:/network/smtp:sendmail> quit
```

3 **sendmail サービスの再表示と再起動を行います。**

最初のコマンドは、変更を実行中のスナップショット内に転送します。2 番目のコマンドは、新しいオプションを使って sendmail サービスを再起動します。

```
# svcadm refresh svc:/network/smtp:sendmail
# svcadm restart svc:/network/smtp:sendmail
```

例 13–1 submit.cf の自動再構築を設定する

この手順では、submit.mc 構成ファイルが自動的に再構築されるように sendmail サービスを構成します。

```
# svccfg -s sendmail-client:default
svc:/network/smtp:sendmail> setprop config/path_to_submit_mc=/etc/mail/cf/cf/submit-myhost.mc
svc:/network/smtp:sendmail> exit
# svcadm refresh svc:/network/sendmail-client
# svcadm restart svc:/network/sendmail-client
```

▼ オープンモードで sendmail を使用する方法

Solaris 10 リリースでは、sendmail サービスがデフォルトでローカル専用モードで実行するように変更されました。ローカル専用モードとは、ローカルホストからのメールだけが受け入れられることを意味します。その他のシステムからのメッセージはすべて拒否されます。Solaris の以前のリリースは、すべてのリモートシステムからのメールを受け入れるように構成されていました。これはオープンモードとして知られています。オープンモードを使用するには、次の手順に従います。

注意 – ローカル専用モードでの sendmail の実行は、オープンモードでの実行よりもはるかに安全です。潜在的なセキュリティーの問題を確実に認識した上で、この手順を実行してください。

1 **スーパーユーザーになるか、同等の役割を引き受けます。**

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 **sendmail** プロパティーを設定します。

```
# svccfg -s sendmail
svc:/network/smtp:sendmail> setprop config/local_only = false
svc:/network/smtp:sendmail> quit
```

3 **sendmail** サービスの再表示と再起動を行います。

```
# svcadm refresh svc:/network/smtp:sendmail
# svcadm restart svc:/network/smtp:sendmail
```

▼ TLS を使用するよう SMTP を構成する

Solaris 10 1/06 以降のリリースでは、SMTP は sendmail の version 8.13 で Transport Layer Security (TLS) を使用できます。SMTP サーバーおよびクライアントに対するこのサービスは、インターネット上での機密性の高い認証された通信だけでなく、盗聴や攻撃からの保護も実現します。このサービスは、デフォルトでは有効になっていないことに注意してください。

次の手順では、サンプルデータを使用して、sendmail が TLS を使用できるようにする証明書を設定する方法を示します。詳細については、381 ページの「sendmail の version 8.13 で TLS を使用して SMTP を実行するためのサポート」を参照してください。

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 sendmail を停止します。

```
# svcadm disable -t network/smtp:sendmail
```

3 sendmail が **TLS** を使用できるようにする証明書を設定します。

a. 次の手順を行います。

```
# cd /etc/mail
# mkdir -p certs/CA
# cd certs/CA
# mkdir certs crl newcerts private
# echo "01" > serial
# cp /dev/null index.txt
# cp /etc/sfw/openssl/openssl.cnf .
```

b. 任意のテキストエディタを使用して openssl.cnf ファイルの dir の値を /etc/sfw/openssl から /etc/mail/certs/CA に変更します。

c. openssl コマンド行ツールを使用して **TLS** を実装します。

次のコマンド行は対話型テキストを生成することに注意してください。

```
# openssl req -new -x509 -keyout private/cakey.pem -out cacert.pem -days 365 \
-config openssl.cnf
Generating a 1024 bit RSA private key
.....................................++++++
.....................................++++++
writing new private key to 'private/cakey.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) []:US
State or Province Name (full name) []:California
Locality Name (eg, city) []:Menlo Park
Organization Name (eg, company) [Unconfigured OpenSSL Installation]:Sun Microsystems
Organizational Unit Name (eg, section) []:Solaris
Common Name (eg, YOUR name) []:somehost.somedomain.example.com
Email Address []:someuser@example.com
```

req	このコマンドは証明書要求を作成し、処理します。
-new	この req オプションを選択すると、新しい証明書要求が作成されます。
-x509	この req オプションを選択すると、自己署名付き証明書が作成されます。
-keyout private/cakey.pem	この req オプションを選択すると、新しく作成された秘密鍵のファイル名として private/cakey.pem を割り当てることができます。
-out cacert.pem	この req オプションを選択すると、出力ファイルとして cacert.pem を割り当てることができます。
-days 365	この req オプションを選択すると、証明書を 365 日間証明することができます。デフォルト値は 30 です。
-config openssl.cnf	この req オプションを選択すると、構成ファイルとして openssl.cnf を指定することができます。

このコマンドは、次の内容を指定する必要があります。

- Country Name (US など)。
- State or Province Name (California など)。

- Locality Name (Menlo Park など)。
- Organization Name (Sun Microsystems など)。
- Organizational Unit Name (Solaris など)。
- Common Name (マシンの完全指定のホスト名)。詳細は、check-hostname(1M) のマニュアルページを参照してください。
- Email Address (someuser@example.com など)。

4 (省略可能) セキュリティー保護された新しい接続が必要である場合、新しい証明書を作成し、認証局を使用して新しい証明書に署名します。

a. 新しい証明書を作成します。

```
# cd /etc/mail/certs/CA
# openssl req -nodes -new -x509 -keyout newreq.pem -out newreq.pem -days 365 \
-config openssl.cnf
Generating a 1024 bit RSA private key
..............++++++
..............++++++
writing new private key to 'newreq.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) []:US
State or Province Name (full name) []:California
Locality Name (eg, city) []:Menlo Park
Organization Name (eg, company) [Unconfigured OpenSSL Installation]:Sun Microsystems
Organizational Unit Name (eg, section) []:Solaris
Common Name (eg, YOUR name) []:somehost.somedomain.example.com
Email Address []:someuser@example.com
```

このコマンドでは、手順 3c で指定した情報と同じ情報を指定する必要があります。

この例では、証明書と秘密鍵はファイル newreq.pem 内にあることに注意してください。

b. 認証局を使用して新しい証明書に署名します。

```
# cd /etc/mail/certs/CA
# openssl x509 -x509toreq -in newreq.pem -signkey newreq.pem -out tmp.pem
Getting request Private Key
Generating certificate request
# openssl ca -config openssl.cnf -policy policy_anything -out newcert.pem -infiles tmp.pem
Using configuration from openssl.cnf
Enter pass phrase for /etc/mail/certs/CA/private/cakey.pem:
Check that the request matches the signature
Signature ok
```

```
Certificate Details:
        Serial Number: 1 (0x1)
        Validity
            Not Before: Jun 23 18:44:38 2005 GMT
            Not After : Jun 23 18:44:38 2006 GMT
        Subject:
            countryName               = US
            stateOrProvinceName       = California
            localityName              = Menlo Park
            organizationName          = Sun Microsystems
            organizationalUnitName    = Solaris
            commonName                = somehost.somedomain.example.com
            emailAddress              = someuser@example.com
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
                93:D4:1F:C3:36:50:C5:97:D7:5E:01:E4:E3:4B:5D:0B:1F:96:9C:E2
            X509v3 Authority Key Identifier:
                keyid:99:47:F7:17:CF:52:2A:74:A2:C0:13:38:20:6B:F1:B3:89:84:CC:68
                DirName:/C=US/ST=California/L=Menlo Park/O=Sun Microsystems/OU=Solaris/\
                CN=someuser@example.com/emailAddress=someuser@example.com
                serial:00

Certificate is to be certified until Jun 23 18:44:38 2006 GMT (365 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
# rm -f tmp.pem
```

この例では、ファイル newreq.pem には未署名の証明書と秘密鍵が含まれています。ファイル newcert.pem には署名済みの証明書が含まれています。

x509 ユーティリティー	証明書の情報を表示し、証明書をさまざまな形式に変換し、証明書要求に署名します。
ca アプリケーション	さまざまな形式の証明書要求の署名と、CRL (Certificate Revocation List) の生成に使用されます。

5 .mc ファイルに次の行を追加することにより、sendmail が証明書を使用できるようにします。

```
define(‘confCACERT_PATH’, ‘/etc/mail/certs’)dnl
define(‘confCACERT’, ‘/etc/mail/certs/CAcert.pem’)dnl
define(‘confSERVER_CERT’, ‘/etc/mail/certs/MYcert.pem’)dnl
define(‘confSERVER_KEY’, ‘/etc/mail/certs/MYkey.pem’)dnl
define(‘confCLIENT_CERT’, ‘/etc/mail/certs/MYcert.pem’)dnl
define(‘confCLIENT_KEY’, ‘/etc/mail/certs/MYkey.pem’)dnl
```

詳細は、382 ページの「TLS を使用して SMTP を実行するための構成ファイルのオプション」を参照してください。

6 /etc/mail ディレクトリで sendmail.cf ファイルを再構築し、インストールします。
手順の詳細は、305 ページの「sendmail 構成を変更する」を参照してください。

7 openssl を使用して作成したファイルから、.mc ファイルで定義したファイルへの、シンボリックリンクを作成します。

```
# cd /etc/mail/certs
# ln -s CA/cacert.pem CAcert.pem
# ln -s CA/newcert.pem MYcert.pem
# ln -s CA/newreq.pem MYkey.pem
```

8 セキュリティーを高めるには、MYkey.pem に関して、グループなどに対して読み取り権を許可しないでください。

```
# chmod go-r MYkey.pem
```

9 シンボリックリンクを使用して、confCACERT_PATH に割り当てられているディレクトリで **CA** 証明書をインストールします。

```
# C=CAcert.pem
# ln -s $C ‘openssl x509 -noout -hash < $C‘.0
```

10 そのほかのホストとのメールのセキュリティーを保護するには、ホストの証明書をインストールします。

a. ほかのホストの confCACERT オプションにより定義されたファイルを、/etc/mail/certs/*host.domain*.cert.pem にコピーします。
host.domain を、ほかのホストの完全指定のホスト名に置き換えます。

b. シンボリックリンクを使用して、confCACERT_PATH に割り当てられているディレクトリで **CA** 証明書をインストールします。

```
# C=host.domain.cert.pem
# ln -s $C ‘openssl x509 -noout -hash < $C‘.0
```

host.domain を、ほかのホストの完全指定のホスト名に置き換えます。

11 sendmail を再起動します。

```
# svcadm enable network/smtp:sendmail
```

例 13–2 Received: メールヘッダー

次に、TLS を使用したセキュリティー保護されたメールの Received: ヘッダーの例を示します。

```
Received: from his.example.com ([IPv6:2001:db8:3c4d:15::1a2f:1a2b])
        by her.example.com (8.13.4+Sun/8.13.4) with ESMTP id j2TNUB8i242496
        (version=TLSv1/SSLv3 cipher=DHE-RSA-AES256-SHA bits=256 verify=OK)
        for <janepc@her.example.com>; Tue, 29 Mar 2005 15:30:11 -0800 (PST)
Received: from her.example.com (her.city.example.com [192.168.0.0])
```

```
        by his.example.com (8.13.4+Sun/8.13.4) with ESMTP id j2TNU7cl571102
        version=TLSv1/SSLv3 cipher=DHE-RSA-AES256-SHA bits=256 verify=OK)
        for <janepc@her.example.com>; Tue, 29 Mar 2005 15:30:07 -0800 (PST)
```

verify の値が OK である、つまり認証が成功したことに注意してください。詳細は、384 ページの「TLS を使用して SMTP を実行するためのマクロ」を参照してください。

参照　次の OpenSSL のマニュアルページも参照してください。

- openssl(1) (http://www.openssl.org/docs/apps/openssl.html)
- req(1) (http://www.openssl.org/docs/apps/req.html)
- x509(1) (http://www.openssl.org/docs/apps/x509.html)
- ca(1) (http://www.openssl.org/docs/apps/ca.html)

▼ sendmail.cf の代替構成を使ってメール配信を管理する方法

送受信されるメールの転送を容易にするため、sendmail の新しいデフォルトの構成は、デーモンとクライアントキューランナーを使用します。クライアントキューランナーは、ローカルの SMTP ポートのデーモンにメールを送信できなければなりません。デーモンが SMTP ポート上で待機していない場合、メールはキューに留まります。この問題を避けるには、次の作業を行います。デーモンとクライアントキューランナーについての詳細、およびこの代替構成を使用する必要性を理解するには、390 ページの「sendmail の version 8.12 からの submit.cf 構成ファイル」を参照してください。

この手順を実行すると、デーモンは、ローカルホストからの接続を受け付けるためだけに動作するようになります。

1. スーパーユーザーになるか、同等の役割を引き受けます。

   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2. sendmail クライアントサービスを停止します。

   ```
   # svcadm disable -t sendmail-client
   ```

3. 変更しようとする構成ファイルのコピーを作成します。

   ```
   # cd /etc/mail/cf/cf
   # cp submit.mc submit-myhost.mc
   ```

   myhost　.mc ファイルの新しい名前を指定します。

4 新しい構成ファイル (たとえば、submit- *myhost*.mc) を編集します。
待機中のホスト IP アドレスを msp 定義に変更します。

```
# grep msp submit-myhost.mc
FEATURE('msp', '[#.#.#.#]')dnl
```

5 m4 を使って構成ファイルを構築します。

```
# /usr/ccs/bin/make submit-myhost.cf
```

6 オリジナルのコピーを作成したあと、新しい構成ファイルをインストールします。

```
# cp /etc/mail/submit.cf /etc/mail/submit.cf.save
# cp submit-myhost.cf /etc/mail/submit.cf
```

7 sendmail クライアントサービスを再起動します。

```
# svcadm enable sendmail-client
```

メール別名ファイルの管理 (作業マップ)

次の表では、メール別名ファイルの管理の手順を説明します。このトピックの詳細は、368 ページの「メール別名ファイル」の 第 14 章「メールサービス (リファレンス)」を参照してください。

作業	説明	参照先
NIS+ mail_aliases テーブルでの別名のエントリの管理	ネームサービスが NIS+ である場合に、mail_aliases テーブルの内容を管理する手順。 NIS+ mail_aliases テーブルを作成します。	317 ページの「NIS+ mail_aliases テーブルを作成する方法」
	NIS+ mail_aliases テーブルの内容を表示します。 この手順には、個々のエントリを表示する方法と部分一致エントリを表示する方法の例が含まれています。	317 ページの「NIS+ mail_aliases テーブルの内容を表示する方法」
	コマンド行から NIS+ mail_aliases テーブルへ別名を追加します。	318 ページの「コマンド行から NIS+ mail_aliases テーブルへ別名を追加する方法」
	NIS+ mail_aliases テーブルを編集してエントリを追加します。	319 ページの「NIS+ mail_aliases テーブルを編集してエントリを追加する方法」

作業	説明	参照先
	NIS+ mail_aliases テーブルでエントリを編集します。 この手順には、エントリを削除する方法の例が含まれています。	320 ページの「NIS+ mail_aliases テーブルのエントリを編集する方法」
NIS mail.aliases マップの設定	ネームサービスが NIS の場合に、mail.aliases マップを使って別名を設定する手順。	321 ページの「NIS mail.aliases マップを設定する方法」
ローカルのメール別名ファイルの設定	NIS や NIS+ などのネームサービスを使用していない場合に、/etc/mail/aliases ファイルを使って別名を設定する手順。	322 ページの「ローカルメール別名ファイルを設定する方法」
キー付きマップファイルの作成	キー付きマップファイルを使って別名を設定する手順。	323 ページの「キー付きマップファイルの作成方法」
postmaster 別名の設定	postmaster 別名を管理する手順。この別名は必須です。	324 ページの「postmaster 別名の管理」

メール別名ファイルを管理する

メール別名はドメイン独自にする必要があります。この節では、メール別名ファイルを管理する手順を説明します。また、Solaris 管理コンソールの「メーリングリスト」機能を使って別名データベース上でこれらの作業を実行することもできます。

その他に、makemap を使ってローカルメールホストにデータベースファイルを作成することもできます。makemap(1M) のマニュアルページを参照してください。ローカルのデータベースファイルを使用しても、NIS や NIS+ のようなネームサービスを使用するほどの利点は得られません。しかし、ネットワークのルックアップは必要ないため、ローカルのデータベースファイルからの方がより早くデータを取り出すことができます。詳細は、375 ページの「sendmail とネームサービスの相互作用」の 368 ページの「メール別名ファイル」および第 14 章「メールサービス (リファレンス)」を参照してください。

次の操作を行うことができます。

- 317 ページの「NIS+ mail_aliases テーブルを作成する方法」
- 317 ページの「NIS+ mail_aliases テーブルの内容を表示する方法」
- 318 ページの「コマンド行から NIS+ mail_aliases テーブルへ別名を追加する方法」
- 319 ページの「NIS+ mail_aliases テーブルを編集してエントリを追加する方法」
- 320 ページの「NIS+ mail_aliases テーブルのエントリを編集する方法」
- 321 ページの「NIS mail.aliases マップを設定する方法」
- 322 ページの「ローカルメール別名ファイルを設定する方法」

- 323 ページの「キー付きマップファイルの作成方法」

▼ NIS+ mail_aliases テーブルを作成する方法

aliasadm コマンドを使用して、NIS+ テーブルのエントリを管理することができます。テーブルを作成するには、次の手順に従います。詳細は、aliasadm(1M) のマニュアルページを参照してください。

1. テーブルを所有する NIS+ グループのメンバーになるか、メールサーバーのスーパーユーザーになるか、同等の役割になります。

   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2. NIS+ テーブルを作成します。

   ```
   # aliasadm -I
   ```

3. テーブルにエントリを追加します。

   - 2 つまたは 3 つの別名を追加する方法については、318 ページの「コマンド行から NIS+ mail_aliases テーブルへ別名を追加する方法」を参照してください。
   - 多数の別名を追加する方法については、319 ページの「NIS+ mail_aliases テーブルを編集してエントリを追加する方法」を参照してください。

▼ NIS+ mail_aliases テーブルの内容を表示する方法

テーブルの全内容を表示するには、次の手順に従います。

1. テーブルを所有する NIS+ グループのメンバーになるか、メールサーバーのスーパーユーザーになるか、同等の役割になります。

   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2. 別名のアルファベット順に全エントリを表示します。

   ```
   # aliasadm -l
   ```

   詳細は、aliasadm(1M) のマニュアルページを参照してください。

例 13–3 NIS+ mail_aliases テーブルの個々のエントリを表示する

また、aliasadm コマンドを使用して、個々のエントリを表示することもできます。この手順の最初の手順が完了したら、次のように入力します。

```
# aliasadm -m ignatz
ignatz: ignatz@saturn # Alias for Iggy Ignatz
```

このコマンドは、完全に一致する別名のみ表示し、部分的に一致するエントリは表示しません。aliasadm -m オプションでは、メタキャラクタ (*、? など) は使用できません。

例 13–4 NIS+ mail_aliases テーブル内の部分一致エントリを表示する

また、aliasadm コマンドを使用して、部分一致エントリを表示することもできます。この手順の最初の手順が完了したら、次のように入力します。

```
# aliasadm -l | grep partial-string
```

partial-string は、検索に使用する文字列で置き換えます。

▼ コマンド行から NIS+ mail_aliases テーブルへ別名を追加する方法

2 つまたは 3 つの別名をテーブルに追加するには、次の手順に従います。多数の別名を追加する場合は、319 ページの「NIS+ mail_aliases テーブルを編集してエントリを追加する方法」を参照してください。

1. メールクライアント、メールボックスの場所、およびメールサーバーシステムの名前の各リストをコンパイルします。
2. テーブルを所有する NIS+ グループのメンバーになるか、メールサーバーのスーパーユーザーになるか、同等の役割になります。

   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。
3. (省略可能) 必要な場合は、NIS+ テーブルを作成します。

   まったく新しい NIS+ mail_aliases テーブルを作成する場合は、最初に NIS+ テーブルを初期設定しなければなりません。テーブルの作成方法については、317 ページの「NIS+ mail_aliases テーブルを作成する方法」を参照してください。
4. テーブルに別名を追加します。

   次に、一般的なエントリの例を示します。

   ```
   # aliasadm -a iggy iggy.ignatz@saturn "Iggy Ignatz"
   ```

   上記の例の入力内容を次に説明します。

   -a　　別名を追加するためのオプション

`iggy`	簡略別名
`iggy.ignatz@saturn`	拡張別名
`"Iggy Ignatz"`	引用符で囲んだ別名

5 作成したエントリを表示し、エントリに間違いがないことを確認します。

```
# aliasadm -m alias
```

alias 作成したエントリ

詳細は、`aliasadm(1M)` のマニュアルページを参照してください。

▼ NIS+ `mail_aliases` テーブルを編集してエントリを追加する方法

`aliasadm` コマンドを使用して、NIS+ テーブルのエントリを管理することができます。多数の別名をテーブルに追加するには、次の手順に従います。

1 メールクライアント、メールボックスの場所、およびメールサーバーシステムの名前の各リストをコンパイルします。

2 テーブルを所有する **NIS+** グループのメンバーになるか、メールサーバーのスーパーユーザーになるか、同等の役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

3 別名テーブルを表示して編集します。

```
# aliasadm -e
```

このコマンドは、テーブルを表示し、テーブルの編集を可能にします。使用するエディタは、`$EDITOR` 環境変数で設定されています。この変数が設定されていない場合、`vi` がデフォルトのエディタになります。

4 次の形式で、**1** 行に **1** 別名ずつ入力します。

alias: *expanded-alias* `#` [`"`*option*`"` `#` `"`*comments*`"`]

alias	この列には、簡略別名を入力します。
expanded-alias	この列には、拡張別名を入力します。
option	この列は、将来の拡張のために予約されています。
comments	この列は、別名など、個々の別名に関するコメントに使用します。

オプション列をブランクにする場合は、空の引用符 2 つ ("") を入力し、そのあとにコメントを追加します。

NIS+ `mail_aliases` テーブルでは、エントリの順序は重要ではありません。`aliasadm -l` コマンドがリストをソートし、エントリをアルファベット順に表示します。

詳細は、368 ページの「メール別名ファイル」および `aliasadm(1M)` のマニュアルページを参照してください。

▼ NIS+ `mail_aliases` テーブルのエントリを編集する方法

テーブル内のエントリを編集するには、次の手順に従います。

1. テーブルを所有する NIS+ グループのメンバーになるか、メールサーバーのスーパーユーザーになるか、同等の役割になります。

   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2. 別名エントリを表示します。

   ```
   # aliasadm -m alias
   ```

   alias は、割り当てられている別名で置き換えます。

3. 必要に応じて別名エントリを編集します。

   ```
   # aliasadm -c alias expanded-alias [options comments]
   ```

alias	必要な場合は、別名を編集します。
expanded-alias	必要な場合は、拡張別名を編集します。
options	必要な場合は、オプションを編集します。
comments	必要な場合は、このエントリのコメントを編集します。

   詳細は、`aliasadm(1M)` のマニュアルページおよび368 ページの「メール別名ファイル」を参照してください。

4. 編集したエントリを表示し、エントリに間違いがないことを確認します。

   ```
   # aliasadm -m alias
   ```

   詳細は、`aliasadm(1M)` のマニュアルページを参照してください。

例 13–5 NIS+ mail_aliases テーブルからエントリを削除する

テーブルからエントリを削除するには、この手順の最初の手順の完了後、次の構文を入力します。

```
# aliasadm -d alias
```

alias は、削除するエントリの別名で置き換えます。

▼ NIS mail.aliases マップを設定する方法

次の手順によって、NIS の mail.aliases マップを使って別名の設定を容易に行うことができます。

1. メールクライアント、メールボックスの場所、およびメールサーバーシステムの名前の各リストをコンパイルします。
2. NIS マスターサーバーのスーパーユーザーになるか、同等の役割になります。

   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。
3. /etc/mail/aliases ファイルを編集し、次のようなエントリを作成します。

   a. メールクライアントごとにエントリを追加します。

   ```
   # cat /etc/mail/aliases
   ..
   alias:expanded-alias
   ```

alias	簡略別名を指定します。
expanded-alias	拡張別名 (user@host.domain.com) を指定します。

   b. Postmaster: root エントリがあることを確認します。

   ```
   # cat /etc/mail/aliases
   ..
   Postmaster: root
   ```

   c. root の別名を追加します。ポストマスターとして指定された個人のメールアドレスを使用します。

   ```
   # cat /etc/mail/aliases
   ..
   root: user@host.domain.com
   ```

user@host.domain.com	指定されたポストマスターに割り当てられているアドレスを指定します。

4 NIS マスターサーバーがネームサービスを実行中で、各メールサーバーのホスト名を解釈処理できることを確認します。

5 /var/yp ディレクトリに移動します。

```
# cd /var/yp
```

6 make コマンドを適用します。

```
# make
```

/etc/hosts および /etc/mail/aliases ファイルの変更は、NIS スレーブシステムに伝達されます。変更は、遅くとも数分後には有効になります。

▼ ローカルメール別名ファイルを設定する方法

ローカルメール別名ファイルで別名を解釈処理するには、次の手順に従います。

1 ユーザーとメールボックスの場所の各リストをコンパイルします。

2 メールサーバーのスーパーユーザーになるか、同等の役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

3 /etc/mail/aliases ファイルを編集し、次のようなエントリを作成します。

a. ユーザーごとにエントリを追加します。

user1: *user2@host.domain*

user1	新しい別名を指定します。
user2@host.domain	新しい別名の実際のアドレスを指定します。

b. Postmaster: root エントリがあることを確認します。

```
# cat /etc/mail/aliases
..
Postmaster: root
```

c. root の別名を追加します。ポストマスターとして指定された個人のメールアドレスを使用します。

```
# cat /etc/mail/aliases
..
root: user@host.domain.com
```

user@host.domain.com	指定されたポストマスターに割り当てられているアドレスを指定します。

4 別名データベースを再構築します。

```
# newaliases
```

/etc/mail/sendmail.cf の AliasFile オプションの構成によって、このコマンドがバイナリ形式で、/etc/mail/aliases.db ファイルを 1 つ生成するか、または /etc/mail/aliases.dir と /etc/mail/aliases.pag の 1 組のファイルを生成するかが決まります。

5 次の手順のどちらかを実行して、生成されたファイルをコピーします。

a. (省略可能) /etc/mail/aliases、/etc/mail/aliases.dir、および /etc/mail/aliases.pag ファイルをほかの各システムにコピーします。

rcp または rdist コマンドを使用して 3 つのファイルをコピーできます。詳細は、rcp(1) のマニュアルページまたは rdist(1) のマニュアルページを参照してください。また、この目的のためのスクリプトを作成することもできます。

これらのファイルをコピーしたら、newaliases コマンドをほかの各システムで実行する必要はありません。ただし、メールクライアントを追加または削除するたびにすべての /etc/mail/aliases ファイルを更新する必要があるので注意してください。

b. (省略可能) /etc/mail/aliases および /etc/mail/aliases.db ファイルをほかの各システムにコピーします。

rcp または rdist コマンドを使用してこれらのファイルをコピーできます。詳細は、rcp(1) のマニュアルページまたは rdist(1) のマニュアルページを参照してください。また、この目的のためのスクリプトを作成することもできます。

これらのファイルをコピーしたら、newaliases コマンドをほかの各システムで実行する必要はありません。ただし、メールクライアントを追加または削除するたびにすべての /etc/mail/aliases ファイルを更新する必要があるので注意してください。

▼ キー付きマップファイルの作成方法

キー付きマップファイルを作成するには、次の手順に従います。

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 入力ファイルを作成します。

エントリには、次の構文を使用できます。

old-name@newdomain.com *new-name@newdomain.com*
old-name@olddomain.com `error:nouser No such user here`
@olddomain.com `%1@`*newdomain.com*

old_name@newdomain.com 新たに割り当てたドメインでこれまで割り当てられていたユーザー名を指定します。

new_name@newdomain.com 新たに割り当てるアドレスを指定します。

old_name@olddomain.com これまで割り当てられていたドメインでこれまで割り当てられていたユーザー名を指定します。

olddomain.com これまで割り当てられていたドメインを指定します。

newdomain.com 新たに割り当てるドメインを指定します。

1 番目のエントリにより、メールは新しい別名に転送されます。2 番目のエントリにより、不適切な別名が使用された時にメッセージが作成されます。最後のエントリにより、すべての着信メールは `olddomain` から `newdomain` へ転送されます。

3 データベースファイルを作成します。

```
# /usr/sbin/makemap maptype newmap < newmap
```

maptype `dbm`、`btree`、`hash` などのデータベースタイプを選択します。

newmap 入力ファイル名とデータベースファイル名の最初の部分を指定します。`dbm` データベースタイプを選択すると、データベースファイルは接尾辞に `.pag` または `.dir` を使って作成されます。ほかの 2 つのデータベースタイプの場合、ファイル名には `.db` が付きます。

postmaster 別名の管理

各システムは `postmaster` メールボックスにメールを送信できなければなりません。`postmaster` の NIS または NIS+ 別名を作成できます。あるいは、ローカルの `/etc/mail/aliases` ファイルそれぞれに別名を作成することもできます。次の手順を参照してください。

- 325 ページの「ローカルの各 `/etc/mail/aliases` ファイルに `postmaster` 別名を作成する方法」
- 325 ページの「`postmaster` 用に別のメールボックスを作成する方法」
- 326 ページの「`postmaster` メールボックスを `/etc/mail/aliases` ファイルの別名に追加する方法」

▼ ローカルの各 /etc/mail/aliases ファイルに postmaster 別名を作成する方法

postmaster 別名をローカルの各 /etc/mail/aliases ファイルに作成する場合は、次の手順に従います。

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 /etc/mail/aliases エントリを表示します。

```
# cat /etc/mail/aliases
# Following alias is required by the mail protocol, RFC 2821
# Set it to the address of a HUMAN who deals with this system's
# mail problems.
Postmaster: root
```

3 各システムの /etc/mail/aliases ファイルを編集します。

root をポストマスターに指定する個人のメールアドレスに変更します。

Postmaster: *mail-address*

mail-address ポストマスターとして指定された個人に割り当てられたアドレスを使用します。

4 (省略可能) ポストマスター用に別のメールボックスを作成します。

ポストマスターがポストマスターメールと個人メールとを区別するために、別のメールボックスを作成できます。別のメールボックスを作成する場合は、/etc/mail/aliases ファイルを編集するときに、ポストマスターの個人メールアドレスではなくメールボックスアドレスを使用してください。詳細は、325 ページの「postmaster 用に別のメールボックスを作成する方法」を参照してください。

▼ postmaster 用に別のメールボックスを作成する方法

postmaster 用に別のメールボックスを作成する場合は、次の手順に従います。

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 postmaster として指定された個人のアカウントを作成します。パスワードフィールドにアスタリスク (*) を入力します。

ユーザーアカウントの追加の詳細については、『Solaris のシステム管理 (基本編)』の第 5 章「ユーザーアカウントとグループの管理 (手順)」を参照してください。

3 メールが配信されたら、mail プログラムがメールボックス名に読み書きできるようにします。

```
# mail -f postmaster
```

postmaster　割り当てられているアドレスを指定します。

▼ postmaster メールボックスを /etc/mail/aliases ファイルの別名に追加する方法

postmaster メールボックスを /etc/mail/aliases ファイル内の別名に追加する場合は、次の手順に従います。

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 root の別名を追加します。ポストマスターとして指定された個人のメールアドレスを使用します。

```
# cat /etc/mail/aliases
..
root: user@host.domain.com
```

user@host.domain.com　ポストマスターとして指定された個人に割り当てられたアドレスを使用します。

3 ポストマスターのローカルシステムで、/etc/mail/aliases ファイルに別名の名前を定義するエントリを作成します。sysadmin が 1 例です。ローカルメールボックスへのパスも指定します。

```
# cat /etc/mail/aliases
..
sysadmin: /usr/somewhere/somefile
```

sysadmin　新しい別名の名前を作成します。

/usr/somewhere/somefile　ローカルメールボックスのパスを指定します。

4 別名データベースを再構築します。

```
# newaliases
```

キューディレクトリの管理 (作業マップ)

次の表では、メールキューの管理の手順を説明します。

作業	説明	参照先
メールキュー /var/spool/mqueue の内容の表示	キューにあるメッセージの数とそれらのメッセージがキューから消去されるのにかかる時間を表示する手順。	328 ページの「メールキュー /var/spool/mqueue の内容を表示する方法」
メールキュー /var/spool/mqueue の強制処理	以前にメッセージを受信できなかったシステムへのメッセージを処理する手順。	328 ページの「メールキュー /var/spool/mqueue でメールキューを強制処理する方法」
メールキュー /var/spool/mqueue のサブセットの実行	ホスト名など、アドレスの部分文字列を強制的に処理する手順。さらに、特定のメッセージをキューから強制的に処理する手順。	329 ページの「メールキュー /var/spool/mqueue のサブセットを実行する方法」
メールキュー /var/spool/mqueue の移動	メールキューを移動する手順。	329 ページの「メールキュー /var/spool/mqueue を移動する方法」
古いメールキュー /var/spool/omqueue の実行	古いメールキューを実行する手順。	330 ページの「古いメールキュー /var/spool/omqueue を実行する方法」

キューディレクトリの管理

この節では、キューの管理に役立つ作業について説明します。クライアント専用のキューの詳細については、390 ページの「sendmail の version 8.12 からの submit.cf 構成ファイル」を参照してください。ほかの関連情報については、402 ページの「sendmail の version 8.12 から追加されたキューの機能」を参照してください。

次を参照してください。

- 328 ページの「メールキュー /var/spool/mqueue の内容を表示する方法」
- 328 ページの「メールキュー /var/spool/mqueue でメールキューを強制処理する方法」
- 329 ページの「メールキュー /var/spool/mqueue のサブセットを実行する方法」
- 329 ページの「メールキュー /var/spool/mqueue を移動する方法」
- 330 ページの「古いメールキュー /var/spool/omqueue を実行する方法」

▼ メールキュー /var/spool/mqueue の内容を表示する方法

- キューにあるメッセージの数とそれらのメッセージがキューから消去されるのにかかる時間を表示します。

  次の行を入力します。

  ```
  # /usr/bin/mailq | more
  ```

  このコマンドは、次の情報を表示します。

  - キュー ID
  - メッセージのサイズ
  - メッセージがキューに入った日付
  - メッセージの状態
  - 送信者と受信者

  さらに、このコマンドは、承認属性 `solaris.admin.mail.mailq` を確認します。確認が取れると、`sendmail` で `-bp` フラグを指定するのと同じ処理が実行されます。確認ができない場合は、エラーメッセージが表示されます。デフォルトでは、この承認属性はすべてのユーザーで使用できるようになっています。承認属性は、`prof_attr` 内のユーザーエントリを変更することにより無効にできます。詳細は、`prof_attr(4)` および `mailq(1)` のマニュアルページを参照してください。

▼ メールキュー /var/spool/mqueue でメールキューを強制処理する方法

たとえば、以前にメッセージを受信できなかったシステムへのメッセージを処理するには、次の手順に従います。

1. スーパーユーザーになるか、同等の役割を引き受けます。

   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2. キューを強制処理し、キューが消去されるとジョブの進捗状況を表示します。

   ```
   # /usr/lib/sendmail -q -v
   ```

▼ メールキュー /var/spool/mqueue のサブセットを実行する方法

たとえば、ホスト名など、アドレスの部分文字列を強制的に処理するには、次の手順に従います。また、特定のメッセージをキューから強制的に処理するにも、次の手順に従います。

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 -qR*string* を使用して、いつでもメールキューのサブセットを実行できます。

```
# /usr/lib/sendmail -qRstring
```

string　受信者の別名または *user@host.domain* の部分文字列 (ホスト名など) を指定

代わりに、-qI*nnnnn* を使ってメールキューのサブセットを実行することもできます。

```
# /usr/lib/sendmail -qInnnnn
```

nnnnn　キュー ID を指定します。

▼ メールキュー /var/spool/mqueue を移動する方法

メールキューを移動する場合は、次の手順に従います。

1 メールホストのスーパーユーザーになるか、同等の役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 sendmail デーモンを終了します。

```
# svcadm disable network/smtp:sendmail
```

これで、sendmail はキューディレクトリを処理しなくなります。

3 /var/spool ディレクトリに移動します。

```
# cd /var/spool
```

4 mqueue ディレクトリとディレクトリ内のすべての内容を omqueue ディレクトリに移動します。次に、mqueue という名前の新しい空のディレクトリを作成します。

```
# mv mqueue omqueue; mkdir mqueue
```

5 ディレクトリのアクセス権を所有者は読み取り/書き込み/実行に、またグループは読み取り/実行に設定します。また、所有者とグループを daemon に設定します。

```
# chmod 750 mqueue; chown root:bin mqueue
```

6 sendmail を起動します。

```
# svcadm enable network/smtp:sendmail
```

▼ 古いメールキュー /var/spool/omqueue を実行する方法

古いメールキューを実行するには、次の手順に従います。

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 古いメールキューを実行します。

```
# /usr/lib/sendmail -oQ/var/spool/omqueue -q
```

-oQ フラグで、代替キューディレクトリを指定します。-q フラグで、キューのすべてのジョブを実行するように指示します。画面に冗長出力を表示する場合は、-v フラグを使用します。

3 空のディレクトリを削除します。

```
# rmdir /var/spool/omqueue
```

.forward ファイルの管理 (作業マップ)

次の表では、.forward ファイルを管理するための手順を説明します。詳細は、371 ページの「.forward ファイル」の 第 14 章「メールサービス (リファレンス)」を参照してください。

作業	説明	参照先
.forward ファイルを無効にする	たとえば、自動転送を禁止する場合に実行する手順。	331 ページの「.forward ファイルを無効にする方法」
.forward ファイルの検索パスを変更する	たとえば、すべての .forward ファイルを共通ディレクトリに移動させる場合に実行する手順。	332 ページの「.forward ファイルの検索パスを変更する方法」

作業	説明	参照先
/etc/shells を作成し生成する	メールをプログラムまたはファイルに転送するために、ユーザーが .forward ファイルを使用できるようにする手順。	332 ページの「/etc/shells の作成および生成方法」

.forward ファイルを管理する

この節では、.forward ファイルの管理に関する複数の手順を説明します。これらのファイルはユーザーが編集できるので、ファイルが問題の原因になる場合があります。詳細は、371 ページの「.forward ファイル」の 第 14 章「メールサービス (リファレンス)」を参照してください。

次を参照してください。

- 331 ページの「.forward ファイルを無効にする方法」
- 332 ページの「.forward ファイルの検索パスを変更する方法」
- 332 ページの「/etc/shells の作成および生成方法」

▼ .forward ファイルを無効にする方法

自動転送を禁止し、特定のホストの .forward ファイルを無効にするには、次の手順に従います。

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 /etc/mail/cf/domain/solaris-generic.m4 またはサイト固有のドメイン **m4** ファイルのコピーを作成します。

```
# cd /etc/mail/cf/domain
# cp solaris-generic.m4 mydomain.m4
```

mydomain　選択するファイル名を指定します。

3 次の行を作成したファイルに追加します。

```
define('confFORWARD_PATH','')dnl
```

m4 ファイルに confFORWARD_PATH の値がすでに存在する場合は、NULL 値に置き換えます。

4 新しい構成ファイルを構築してインストールします。

この手順の詳細については、305 ページの「新しい sendmail.cf ファイルを構築する方法」を参照してください。

注 – .mc ファイルを編集する場合は、忘れずに、DOMAIN(‘solaris-generic’) を DOMAIN(‘*mydomain* ’) に変更してください。

▼ .forward ファイルの検索パスを変更する方法

たとえば、すべての .forward ファイルを共通ディレクトリに入れる場合は、次の手順に従います。

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 /etc/mail/cf/domain/solaris-generic.m4 またはサイト固有のドメイン **m4** ファイルのコピーを作成します。

```
# cd /etc/mail/cf/domain
# cp solaris-generic.m4 mydomain.m4
```

mydomain 選択するファイル名を指定します。

3 次の行を作成したファイルに追加します。

```
define(‘confFORWARD_PATH’,‘$z/.forward:/var/forward/$u’)dnl
```

m4 ファイルに confFORWARD_PATH の値がすでに存在する場合は、新しい値に置き換えます。

4 新しい構成ファイルを構築してインストールします。

この手順の詳細については、305 ページの「新しい sendmail.cf ファイルを構築する方法」を参照してください。

注 – .mc ファイルを編集する場合は、忘れずに、DOMAIN(‘solaris-generic’) を DOMAIN(‘*mydomain* ’) に変更してください。

▼ /etc/shells の作成および生成方法

このファイルは標準リリースには含まれていません。.forward ファイルを使用してプログラムまたはファイルにメールを転送することをユーザーに許可する場合

は、このファイルを追加する必要があります。grep を使用して、パスワードファイルに一覧表示されたすべてのシェルを特定し、ファイルを手動で作成することができます。これにより、シェルをファイルに入力できます。しかし、次に示す、ダウンロード可能なスクリプトを使用する手順の方が簡単です。

1 スクリプトをダウンロードします。

http://www.sendmail.org/vendor/sun/gen-etc-shells.html

2 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

3 シェルのリストを作成するために、gen-etc-shells を実行します。

```
# ./gen-etc-shells.sh > /tmp/shells
```

このスクリプトでは、getent コマンドを使用して、/etc/nsswitch.conf 内に一覧表示されたパスワードファイルソースに組み込まれたシェルの名前を収集します。

4 /tmp/shells 内のシェルのリストを調べて編集します。

選択したエディタを使用し、組み込まないシェルを削除します。

5 ファイルを /etc/shells に移動します。

```
# mv /tmp/shells /etc/shells
```

メールサービスの障害対処とヒント (作業マップ)

次の表では、メールサービスのトラブルシューティング手順とヒントを説明します。

作業	説明	参照先
メール構成のテスト	sendmail 構成ファイルの変更をテストする手順	334 ページの「メール構成をテストする方法」
メール別名の確認	指定された受信者にメールを配信できるかどうかを確認する手順	335 ページの「メール別名を確認する方法」
ルールセットのテスト	sendmail ルールセットの入力と戻りを確認する手順	335 ページの「sendmail ルールセットをテストする方法」
ほかのシステムへの接続の確認	ほかのシステムへの接続を確認するためのヒント	336 ページの「ほかのシステムへの接続を調べる方法」

作業	説明	参照先
syslogd プログラムを使用したメッセージの記録	エラーメッセージ情報を収集するためのヒント	337 ページの「エラーメッセージの記録」
診断情報のその他の情報源の確認	ほかの情報源から診断情報を取得するためのヒント	338 ページの「メール診断情報のその他の情報源」

メールサービスのトラブルシューティング手順とヒント

この節では、メールサービスの問題解決に使用できる手順とヒントをいくつか示します。

▼ メール構成をテストする方法

構成ファイルに対して行なった変更をテストするには、次の手順に従います。

1. 変更した構成ファイルがあるシステムで sendmail を再起動します。

   ```
   # svcadm refresh network/smtp:sendmail
   ```

2. 各システムからテストメッセージを送信します。

   ```
   # /usr/lib/sendmail -v names </dev/null
   ```

   names　受信者の電子メールアドレスを指定します。

   このコマンドは、指定された受信者に NULL メッセージを送信し、画面にメッセージの動作を表示します。

3. メッセージを通常のユーザー名に送ることによって、メールを自分自身またはローカルシステム上のほかの人に送信します。

4. (省略可能) ネットワークに接続している場合は、別のシステムの個人宛に次の 3 方向でメールを送信します。

   - メインシステムからクライアントシステムへ
   - クライアントシステムからメインシステムへ
   - クライアントシステムから別のクライアントシステムへ

5. (省略可能) メールゲートウェイがある場合、メールホストから別のドメインにメールを送信して、中継メールプログラムおよびホストが適切に設定されていることを確認します。

6. (省略可能) 電話回線上で別のホストへの UUCP 接続を設定している場合は、そのホストのだれかにメールを送信します。メッセージを受信した時点で、メールを返信してもらうか、電話してもらいます。

7 **UUCP 接続を介してメールを送信するようにほかの人に頼みます。**

sendmail プログラムは、メッセージが配信されたかどうかは検出しません。これは、配信のためにプログラムがメッセージを UUCP に渡すためです。

8 異なるシステムからメッセージを postmaster 宛てに送信し、ポストマスターのメールボックスにそのメッセージが配信されたことを確認します。

メール別名を確認する方法

次の例は、別名を確認する方法を示します。

```
% mconnect
connecting to host localhost (127.0.0.1), port 25
connection open
220 your.domain.com ESMTP Sendmail 8.13.6+Sun/8.13.6; Tue, 12 Sep 2004 13:34:13 -0800 (PST)
expn sandy
250 2.1.5 <sandy@phoenix.example.com>
quit
221 2.0.0 your.domain.com closing connection
%
```

この例では、mconnect プログラムがローカルホスト上のメールサーバーとの接続を確立し、接続をテストできるようにします。プログラムは対話式で実行されるので、さまざまな診断コマンドを実行できます。詳細は、mconnect(1) のマニュアルページを参照してください。expn sandy のエントリに、展開されたアドレス sandy@phoenix.example.com が示されています。したがって、別名 sandy でメールを配信できることが確認されました。

ローカルおよびドメインの両方で別名を使用する場合は、ループやデータベースの不整合が生じないようにしてください。あるシステムから別のシステムにユーザーを移動するときは、別名のループが生じないよう特に注意してください。

▼ sendmail ルールセットをテストする方法

sendmail ルールセットの入力と戻りを確認するには、次の手順に従います。

1 アドレステストモードに変更します。

```
# /usr/lib/sendmail -bt
```

2 メールアドレスをテストします。

最後のプロンプト (>) で次の数値とアドレスを入力します。

```
> 3,0 mail-sraddress
```

mail-address　テストするメールアドレスを指定します。

3 セッションを終了します。
Control-D キーを押します。

例 13–6 アドレステストモードの出力

次にアドレステストモードの出力例を示します。

```
% /usr/lib/sendmail -bt
ADDRESS TEST MODE (ruleset 3 NOT automatically invoked)
Enter <ruleset> <address>
> 3,0 sandy@phoenix
canonify           input: sandy @ phoenix
Canonify2          input: sandy < @ phoenix >
Canonify2        returns: sandy < @ phoenix . example . com . >
canonify         returns: sandy < @ phoenix . example . com . >
parse              input: sandy < @ phoenix . example . com . >
Parse0             input: sandy < @ phoenix . example . com . >
Parse0           returns: sandy < @ phoenix . example . com . >
ParseLocal         input: sandy < @ phoenix . example . com . >
ParseLocal       returns: sandy < @ phoenix . example . com . >
Parse1             input: sandy < @ phoenix . example . com . >
MailerToTriple     input: < mailhost . phoenix . example . com >
     sandy < @ phoenix . example . com . >
MailerToTriple   returns: $# relay $@ mailhost . phoenix . example . com
     $: sandy < @ phoenix . example . com . >
Parse1           returns: $# relay $@ mailhost . phoenix . example . com
     $: sandy < @ phoenix . example . com . >
parse            returns: $# relay $@ mailhost . phoenix . example . com
     $: sandy < @ phoenix . example . com . >
```

ほかのシステムへの接続を調べる方法

mconnect プログラムは、指定したホスト上のメールサーバーへの接続を開き、接続をテストできるようにします。プログラムは対話式で実行されるので、さまざまな診断コマンドを実行できます。詳細は、mconnect(1) のマニュアルページを参照してください。次の例では、ユーザー名 sandy へのメールが配信可能かどうかを調べます。

```
% mconnect phoenix

connecting to host phoenix (172.31.255.255), port 25
connection open
220 phoenix.example.com ESMTP Sendmail 8.13.1+Sun/8.13.1; Sat, 4 Sep 2004 3:52:56 -0700
expn sandy
250 2.1.5 <sandy@phoenix.example.com>
quit
```

mconnect を使用して SMTP ポートに接続できない場合は、次の条件を確認してください。

- システム負荷が高すぎないか
- sendmail デーモンが動作しているか
- システムに適切な /etc/mail/sendmail.cf ファイルがあるか
- sendmail が使用するポート 25 がアクティブであるか

エラーメッセージの記録

メールサービスは、syslogd プログラムを使って大部分のエラーメッセージを記録します。デフォルトでは、syslogd プログラムはこれらのメッセージを /etc/hosts ファイルで指定されている loghost というシステムに送信します。loghost が NIS ドメイン全体のすべてのログを保持するように定義できます。loghost を指定しなければ、syslogd からのエラーメッセージはレポートされません。

/etc/syslog.conf ファイルは、syslogd プログラムがメッセージをどこに転送するかを制御します。/etc/syslog.conf ファイルを編集することにより、デフォルト構成を変更できます。変更内容を有効にするには、syslog デーモンを再起動する必要があります。メールに関する情報を収集するために、ファイルに次の選択を追加できます。

- mail.alert – ここで訂正する必要のある状態に関するメッセージ
- mail.crit – クリティカルメッセージ
- mail.warning – 警告メッセージ
- mail.notice – エラーではないが注意すべきメッセージ
- mail.info – 情報メッセージ
- mail.debug – デバッグメッセージ

/etc/syslog.conf ファイルの次のエントリは、クリティカルメッセージ、通知メッセージ、デバッグメッセージをすべて /var/log/syslog に送信します。

```
mail.crit;mail.info;mail.debug              /var/log/syslog
```

システムログの各行には、タイムスタンプ、そのログ行を生成したシステム名、およびメッセージが入っています。syslog ファイルは、大量の情報を記録できます。

ログは、連続したレベルとして並べられます。最下位レベルでは、異常なイベントだけが記録されます。最上位レベルでは、もっとも必須なイベントと注目する必要のないイベントが記録されます。通常、10 以下のログレベルが「有用」とみなされます。10 を超えるログレベルは通常、デバッグに使用されます。loghost および syslogd プログラムの詳細については、『Solaris のシステム管理 (上級編)』の「システムのメッセージ記録のカスタマイズ」を参照してください。

メール診断情報のその他の情報源

その他の診断情報については、次の情報源を確認してください。

- メッセージのヘッダーの Received 行を調べます。これらの行は、メッセージが中継されるときにとった経路を追跡できます。時間帯の違いを考慮するのを忘れないでください。
- MAILER-DAEMON からのメッセージを調べます。これらのメッセージは通常、配信上の問題をレポートします。
- ワークステーショングループの配信上の問題を記録するシステムログを確認します。sendmail プログラムは常に、その処理内容をシステムログに記録します。crontab ファイルを修正して、シェルスクリプトを夜間に実行できます。このスクリプトは、ログで SYSERR メッセージを検索し、検出したメッセージをポストマスターにメールで送信します。
- mailstats プログラムを使ってメールタイプをテストし、着信メッセージと発信メッセージの数を判定します。

エラーメッセージの解釈

この節では、sendmail 関連のエラーメッセージを解釈し対処する方法について説明します。http://www.sendmail.org/faq/ も参照してください。

次のエラーメッセージには、次の種類の情報が含まれます。

- 原因: メッセージ発生の原因となった可能性があるもの
- 説明: エラーメッセージが発生した時にユーザーが行なっていた操作
- 対処方法: 問題を解決するため、あるいは作業を続けるための操作

`451 timeout waiting for input during` *source*

原因: タイムアウトの可能性があるソース (SMTP 接続など) から読み取るとき、sendmail は、読み込みを開始する前にさまざまな Timeout オプションの値をタイマーに設定します。タイマーが期限切れになる前に読み取りが完了しなかった場合、このメッセージが表示され、読み取りが停止します。通常、この状況は RCPT 時に発生します。メールメッセージはキューに入れられて、あとで配信されます。

対処方法: このメッセージが頻繁に表示される場合は、/etc/mail/sendmail.cf ファイルの Timeout オプションの値を大きくします。タイマーがすでに大きな値に設定されている場合は、ネットワークの配線や接続などハードウェアの問題点を探します。

`550` *hostname*`... Host unknown`
原因: この `sendmail` のメッセージは、単価記号 (@) のあとのアドレス部分で指定されている受信先のホストマシンが、ドメインネームシステム (DNS) ルックアップ時に見つからなかったことを示します。

対処方法: `nslookup` コマンドを使用して、受信先ホストが、そのドメインまたはほかのドメインにあることを確認します。スペルが間違っている可能性があります。あるいは、受信者に連絡して正しいアドレスを確認します。

`550` *username*`... User unknown`
原因: この `sendmail` のメッセージは、単価記号 (@) の前のアドレス部分で指定されている受信者を受信先ホストマシンで検出できなかったことを示します。

対処方法: 電子メールアドレスを確認し、再度送信してみます。スペルが間違っている可能性があります。これで解決しない場合は、受信者に連絡して正しいアドレスを確認します。

`554` *hostname*`... Local configuration error`
原因: この `sendmail` メッセージは通常、ローカルホストがメールを自分宛に送信しようとしていることを示します。

対処方法: `/etc/mail/sendmail.cf` ファイル内の `$j` マクロの値が完全指定ドメイン名になっていることを確認します。

説明: 送信側のシステムが SMTP の `HELO` コマンドで受信側のシステムに自身のホスト名を示すと、受信側のシステムはそのホスト名を送信者の名前と比較します。これらの名前が同じ場合、受信側のシステムはこのエラーメッセージを発行し、接続を閉じます。`HELO` コマンドで提供される名前は、`$j` マクロの値です。

追加情報については、`http://www.sendmail.org/faq/section4#4.5` を参照してください。

`config error: mail loops back to myself.`
原因: このエラーメッセージが生成されるのは、MX レコードを設定し、ホスト *bar* をドメイン *foo* のメール交換局にした場合です。ただし、ホスト *bar* 自身がドメイン *foo* のメール交換局であることを認識するように設定されていません。

また、送信側システムと受信側システムの両方が同じドメインとして識別される場合にも、このメッセージを受け取ります。

対処方法: 手順については、`http://www.sendmail.org/faq/section4#4.5` を参照してください。

`host name configuration error`
: 説明: これは `sendmail` の古いメッセージで、「`I refuse to talk to myself`」というメッセージから置き換えられたもので現在は、「`Local configuration error`」メッセージに置き換えられています。

  対処方法: 次のエラーメッセージの対処方法で説明されている手順に従います。`554` *`hostname`*`... Local configuration error.`

`user unknown`
: 原因: メールをユーザー宛てに送信しようとすると、「`Username... user unknown`」のエラーが表示されます。ユーザーが同じシステム上にいます。

  対処方法: 入力した電子メールアドレスに誤字がないか確認します。あるいは、ユーザーが、`/etc/mail/aliases` またはユーザーの `.mailrc` ファイルに存在しない電子メールアドレスに別名を割り当てられている可能性があります。また、ユーザー名の大文字も確認してください。できれば、電子メールアドレスは大文字と小文字が区別されないようにします。

  追加情報については、`http://www.sendmail.org/faq/section4#4.17` を参照してください。

◆ ◆ ◆ 第 14 章

メールサービス (リファレンス)

sendmail プログラムは、メール転送エージェントです。前の章で説明したように、このプログラムは、構成ファイルを使用して、別名処理、転送、ネットワークゲートウェイへの自動ルーティング、柔軟な構成を提供します。Solaris OS では、ほとんどのサイトで使用できる標準構成ファイルが付属しています。第 12 章「メールサービス (概要)」では、メールサービスのコンポーネントと典型的なメールサービスの構成を紹介しています。第 13 章「メールサービス (手順)」では、電子メールシステムをセットアップして管理する方法について説明しています。この章では、次のトピックについて説明します。

上記の章で説明されていない内容については、次のマニュアルページを参照してください。

- sendmail(1M)
- mail.local(1M)
- mailstats(1)
- makemap(1M)
- editmap(1M)

Solaris 版の sendmail

ここでは、次の項目について sendmail の Solaris 版と一般的な Berkeley バージョンを比較します。

sendmail のコンパイルに使用できるフラグと使用できないフラグ

Solaris 10 以降のリリースで sendmail のコンパイルに使用されるフラグは、次のとおりです。構成にほかのフラグが必要な場合は、そのソースをダウンロードし、バイナリにコンパイルし直してください。このプロセスについては、http://www.sendmail.org を参照してください。

表 14–1　一般的な sendmail フラグ

フラグ	説明
SOLARIS=21000	Solaris 10 のサポート。
MILTER	メールフィルタ API のサポート。sendmail の version 8.13 では、このフラグはデフォルトで有効になっています。343 ページの「MILTER (sendmail のメールフィルタ API)」を参照してください。
NETINET6	IPv6 のサポート。このフラグは、conf.h から Makefile に移動されました。

表 14–2　マップとデータベースの種類

フラグ	説明
NDBM	ndbm データベースのサポート
NEWDB	Berkeley DB データベースのサポート
USERDB	ユーザーデータベースのサポート
NIS	nis データベースのサポート
NISPLUS	nisplus データベースのサポート

表 14–2 マップとデータベースの種類 (続き)

フラグ	説明
LDAPMAP	LDAP のマップのサポート
MAP_REGEX	正規表現のマップのサポート

表 14–3 Solaris のフラグ

フラグ	説明
SUN_EXTENSIONS	sun_compat.o に含まれる Sun の拡張をサポートします。
SUN_INIT_DOMAIN	下位互換性を確保するために、NIS ドメイン名をサポートしてローカルホスト名を完全指定します。詳細は、http://www.sendmail.org のベンダー固有の情報を参照してください。
SUN_SIMPLIFIED_LDAP	Sun 固有の簡略化された LDAP API をサポートします。詳細は、http://www.sendmail.org のベンダー固有の情報を参照してください。
VENDOR_DEFAULT=VENDOR_SUN	Sun をデフォルトのベンダーに選択します。

次の表に、Solaris 10 に添付されるバージョンの sendmail のコンパイルに使用されない一般的なフラグを示します。

表 14–4 sendmail の Solaris 版に使用されない一般的なフラグ

フラグ	説明
SASL	Simple Authentication and Security Layer (RFC 2554)
STARTTLS	Transaction Level Security (RFC 2487)

sendmail のコンパイルに使用するフラグのリストを参照するには、次のコマンドを使用します。

```
% /usr/lib/sendmail -bt -d0.10 < /dev/null
```

注 – 上記のコマンドでは、Sun 固有のフラグは表示されません。

MILTER (sendmail のメールフィルタ API)

MILTER (sendmail のメールフィルタ API) によって、サードパーティー製のプログラムが、メタ情報と本文にフィルタをかけるために処理されるときに、メール

メッセージにアクセスできるようになります。フィルタを作成する必要や、作成したフィルタを使用するように sendmailを構成する必要はありません。この API は、sendmail の version 8.13 ではデフォルトで有効になっています。

詳細は、次を参照してください。

- http://www.sendmail.org
- https://www.milter.org/

sendmail の代替コマンド

Solaris リリースには、sendmail.org による汎用リリースで提供されているコマンドの同義語がすべて組み込まれているわけではありません。次の表は、すべてのコマンドの別名を示したリストです。この表には、コマンドが Solaris リリースに組み込まれているかどうか、および sendmail を使って同じ動作を生成する方法も示しています。

表 14–5 代替 sendmail コマンド

代替名	Solaris への組み込み	sendmail を使用したオプション
hoststat	いいえ	sendmail -bh
mailq	はい	sendmail -bp
newaliases	はい	sendmail -bi
purgestat	いいえ	sendmail -bH
smtpd	いいえ	sendmail -bd

構成ファイルのバージョン

Solaris 10 以降のリリースに含まれている sendmail のバージョンには、sendmail.cf ファイルのバージョンを定義するための構成オプションが含まれます。現在のバージョンの sendmail でも以前のバージョンの構成ファイルを使用できます。バージョンレベルには 0 から 10 の値を設定できます。また、ベンダーの定義もできます。Berkeley または Sun をベンダーとして選択できます。ベンダーを定義しないでバージョンレベルだけを設定した場合は、Sun がデフォルトとして使用されます。次の表に有効なオプションを示します。

表 14–6 構成ファイルのバージョン値

フィールド	説明
V7/Sun	sendmail の version 8.8 で使用された設定。

表 14–6 構成ファイルのバージョン値 (続き)

フィールド	説明
V8/Sun	sendmail の version 8.9 で使用された設定。この設定は、Solaris 8 に含まれていました。
V9/Sun	sendmail の version 8.10 と 8.11 で使用された設定。
V10/Sun	sendmail の version 8.12 と 8.13 で使用される設定。version 8.12 は、Solaris 9 のデフォルトです。Solaris 10 以降のリリースでは、version 8.13 がデフォルトです。

注 – V1/Sun は使用しないでください。詳細は、http://www.sendmail.org/vendor/sun/differences.html#4 を参照してください。

作業手順については、第 13 章「メールサービス (手順)」の 305 ページの「sendmail 構成を変更する」を参照してください。

メールサービスのソフトウェアとハードウェアのコンポーネント

ここでは、メールシステムのソフトウェアとハードウェアの構成要素について説明します。

ソフトウェアコンポーネント

各メールサービスには、少なくとも次のいずれかのソフトウェアコンポーネントが含まれます。

ここでは、次のソフトウェアコンポーネントについても説明します。

メールユーザーエージェント

「メールユーザーエージェント」は、ユーザーとメール転送エージェント間のインタフェースとして機能するプログラムです。sendmail プログラムは、メール転送エージェントです。Solaris オペレーティングシステムは、次のメールユーザーエージェントを提供します。

- /usr/bin/mail
- /usr/bin/mailx
- /usr/dt/bin/dtmail

メール転送エージェント

「メール転送エージェント」は、メールメッセージのルーティングとメールアドレスの解釈を行います。このエージェントは、「メールトランスポートエージェント」とも呼ばれます。Solaris オペレーティングシステムの転送エージェントは sendmail です。転送エージェントは次の機能を実行します。

- メールユーザーエージェントからメッセージを受信する
- 宛先アドレスを認識する
- 適切な配信エージェントを選択してメールを配信する
- ほかのメール転送エージェントからのメールを受信する

ローカル配信エージェント

「ローカル配信エージェント」は、メールの配信プロトコルを実行するプログラムです。Solaris オペレーティングシステムには、次のローカル配信エージェントが提供されています。

- UUCP ローカル配信エージェント (uux を使ってメールを配信する)
- ローカル配信エージェント (標準の Solaris リリースでは mail.local)

389 ページの「sendmail の version 8.12 からの変更点」 では、次の関連項目について説明します。

- 400 ページの「sendmail の version 8.12 から追加された配信エージェントのフラグ」
- 401 ページの「sendmail の version 8.12 から追加された配信エージェントの設定」

メールプログラムと sendmail

「メールプログラム」は、sendmail 固有の用語です。「メールプログラム」は sendmail によって使用され、カスタマイズされたローカル配信エージェントまたはカスタマイズされたメール転送エージェントの特定のインスタンスを特定します。sendmail.cf ファイルに少なくとも 1 つのメールプログラムを指定する必要があります。作業手順については、第 13 章「メールサービス (手順)」の 305 ページの「sendmail 構成を変更する」を参照してください。ここでは、2 種類のメールプログラムについて説明します。

- 347 ページの「SMTP (Simple Mail Transfer Protocol) メールプログラム」
- 347 ページの「UUCP (UNIX-to-UNIX Copy Program) メールプログラム」

メールプログラムの詳細は、`http://www.sendmail.org/m4/readme.html` または `/etc/mail/cf/README` を参照してください。

SMTP (Simple Mail Transfer Protocol) メールプログラム

SMTP はインターネットで使用される標準のメールプロトコルです。このプロトコルが、メールプログラムを定義します。

- `smtp` は、ほかのサーバーへの標準 SMTP 転送機能を提供します。
- `esmtp` は、ほかのサーバーへの拡張 SMTP 転送機能を提供します。
- `smtp8` は、8 ビットデータを MIME に変更することなく、ほかのサーバーに SMTP 転送機能を提供します。
- `dsmtp` は、`F=%` メールプログラムフラグを使ってオンデマンド配信機能を提供します。400 ページの「`sendmail` の version 8.12 からの `MAILER()` の宣言についての変更点」と 400 ページの「`sendmail` の version 8.12 から追加された配信エージェントのフラグ」を参照してください。

UUCP (UNIX-to-UNIX Copy Program) メールプログラム

UUCP の使用は、できるだけ避けてください。説明については、`http://www.sendmail.org/m4/uucp_mailers.html` を参照するか、`/etc/mail/cf/README` で `USING UUCP MAILERS` という文字列を検索してください。

UUCP が、メールプログラムを定義します。

`uucp-old` `$=U` クラスの名前が `uucp-old` に送られます。`suucp` は、このメールプログラムの以前の名前です。`uucp-old` メールプログラムはヘッダーでは感嘆符を用いるアドレスを使用します。

`uucp-new` `$=Y` クラスの名前が `uucp-new` に送られます。受信側の UUCP メールプログラムが単一の転送で複数の受信者を管理できる場合は、このメールプログラムを使用します。`suucp` は、このメールプログラムの以前の名前です。`uucp-new` メールプログラムはヘッダーで感嘆符を用いるアドレスも使用します。

構成に `MAILER(smtp)` も指定されている場合は、さらに次の 2 つのメールプログラムが定義されます。

`uucp-dom` このメールプログラムは、ドメインスタイルアドレスを使用し、基本的に SMTP のリライトルールを適用します。

`uucp-uudom` `$=Z` クラスの名前が `uucp-uudom` に送られます。`uucp-uudom` と `uucp-dom` は、ドメインスタイルアドレスという同じヘッダーアドレス書式を使用します。

注 – smtp メールプログラムは UUCP メールプログラムを変更するので、.mc ファイルの MAILER(uucp) の前に必ず MAILER(smtp) を記述します。

メールアドレス

「メールアドレス」には、受信者の名前と、メールメッセージが配信されるシステムが含まれます。ネームサービスを使用しない小さなメールシステムを管理する場合、メールのアドレス指定は簡単です。つまり、ログイン名がユーザーを一意に識別します。メールボックスを含む複数のシステムで構成されるメールシステム、または 1 つ以上のドメインで構成されるメールシステムを管理する場合は複雑になります。UUCP またはその他のメールシステムによってネットワーク外部のサーバーに接続する場合は、さらに複雑になります。次の節で、メールアドレスの各部とその複雑さを説明しています。

- 348 ページの「ドメインとサブドメイン」
- 349 ページの「ネームサービスドメイン名とメールドメイン名」
- 349 ページの「メールアドレスの一般的な書式」
- 350 ページの「経路に依存しないメールアドレス」

ドメインとサブドメイン

電子メールのアドレス指定には、ドメインが使用されます。「ドメイン」は、ネットワークアドレスの命名のためのディレクトリ構造です。ドメインは 1 つ以上の「サブドメイン」を持つことができます。アドレスのドメインとサブドメインは、ファイルシステムの階層と比較できます。サブディレクトリが上位のディレクトリに含まれるように、メールアドレスの各サブドメインもその右のドメインに含まれると考えられます。

次の表に最上位のドメインを示します。

表 14–7 最上位のドメイン

ドメイン	説明
com	企業
edu	教育機関用
gov	米国の政府機関
mil	米国の軍事機関
net	ネットワーク組織
org	その他の非営利組織

ドメインには大文字と小文字の区別がありません。アドレスのドメイン部分には、大文字、小文字、またはその両方を混合したものを、問題なく使用できます。

ネームサービスドメイン名とメールドメイン名

ネームサービスドメイン名とメールドメイン名を操作するときは、次のことに注意します。

- `sendmail` プログラムは、デフォルトで NIS または NIS+ ドメイン名から最初の構成要素を取り除き、メールドメイン名とします。たとえば、NIS+ ドメイン名が `bldg5.example.com` の場合、メールドメイン名は `example.com` になります。
- メールドメインアドレスは大文字と小文字の区別をしませんが、NIS または NIS+ ドメイン名は異なります。メールと NIS または NIS+ ドメイン名を設定するときは、小文字を使用するのが最善です。
- DNS ドメイン名とメールドメイン名は同じでなければなりません。

詳細は、375 ページの「`sendmail` とネームサービスの相互作用」を参照してください。

メールアドレスの一般的な書式

一般に、メールアドレスは次のような書式になります。詳細は、350 ページの「経路に依存しないメールアドレス」を参照してください。

user@subdomain. subdomain2.subdomain1.top-level-domain

アドレスの `@` 記号より左の部分はローカルアドレスです。ローカルアドレスには、次の内容を含めることができます。

- 別のメールトランスポートを使用するルーティングに関する情報 (たとえば、`bob::vmsvax@gateway` または `smallberries%mill.uucp@gateway`)
- 別名 (たとえば、`iggy.ignatz`)

注 – 受信側のメールプログラムでアドレスのローカル部分を解釈する必要があります。メールプログラムの詳細は、346 ページの「メールプログラムと `sendmail`」を参照してください。

アドレスの `@` 記号より右の部分は、ローカルアドレスが位置するドメインレベルを示します。各サブドメインはドットで区切られます。アドレスのドメイン部分は、組織、物理的な場所、または地域を表すことができます。さらに、ドメイン情報の順序は階層的で、ローカルなサブドメインほど `@` 記号に近くなります。

経路に依存しないメールアドレス

メールアドレスは、経路に依存しないアドレス指定ができます。経路に依存しないアドレス指定では、電子メールメッセージの発信者は、受信者の名前と最終の宛先を指定する必要があります。インターネットなどの高速ネットワークでは、経路に依存しないアドレスを使用します。経路に依存しないアドレスは次のような書式になります。

user@*host*.*domain*

UUCP 接続の経路に依存しないアドレスは次のような書式になります。

host.*domain*!*user*

コンピュータのドメイン階層命名方式が普及したため、経路に依存しないアドレスがより一般的になってきました。実際、次に示すように、もっとも一般的な経路に依存しないアドレスはホスト名を省略し、電子メールメッセージの最終宛先の識別をドメインネームサービスに任せています。

user@*domain*

経路に依存しないアドレスは、まず @ 記号を検索して読み取られます。次に、ドメイン階層が右 (最上位) から左 (@ 記号の右側にあるもっとも固有な部分) へと読み取られます。

メールボックスファイル

「メールボックス」は、電子メールメッセージの最終的な宛先となるファイルです。メールボックス名には、ユーザー名または postmaster などの特定の機能の名前を指定できます。メールボックスは、ユーザーのローカルシステムかリモートのメールサーバーのいずれかの `/var/mail/`*username* ファイルにあります。ただし、いずれの場合でも、メールボックスはメールが配信されるシステム上にあります。

ユーザーエージェントがメールスプールからメールを取り出し、ローカルメールボックスに容易に格納できるように、メールは常にローカルファイルシステムに配信される必要があります。ユーザーのメールボックスの宛先として、NFS でマウントされたファイルシステムを使用しないでください。特にリモートサーバーから `/var/mail` ファイルシステムをマウントしているメールクライアントには、直接メールを送信しないでください。この場合ユーザー宛てのメールは、クライアントのホスト名ではなく、メールサーバーにアドレス指定する必要があります。NFS でマウントされたファイルシステムは、メールの配信と処理に問題を起こすことがあります。

`/etc/mail/aliases` ファイルと NIS や NIS+ といったネームサービスを利用すると、電子メールアドレスの別名を作成できます。したがって、ユーザーは、個々のユーザーのメールボックスの正確なローカル名を知る必要はありません。

次の表に、特殊な目的のメールボックスに対する共通の命名規則をいくつか示します。

表 14–8 メールボックス名の書式についての規則

表記形式	説明
username	多くの場合、ユーザー名はメールボックス名と同じです。
Firstname.*Lastname* *Firstname*_*Lastname* *Firstinitial*.*Lastname* *Firstinitial*_*Lastname*	ユーザー名は、ファーストネームとラストネームをドット (または下線) で区切ったフルネーム。または、ファーストネームをイニシャルにして、イニシャルとラストネームをドット (または下線) で区切ったもの。
`postmaster`	ユーザーは、`postmaster` のメールボックスに質問を送ったり、問題点を報告したりできます。通常は各サイトとドメインに `postmaster` メールボックスがあります。
`MAILER-DAEMON`	`sendmail` は、`MAILER-DAEMON` 宛てのメールを自動的にポストマスターに送ります。
aliasname`-request`	`-request` で終わる名前は、配布リストの管理アドレス。このアドレスは、配布リストを管理する人にメールをリダイレクトします。
`owner-`*aliasname*	`owner-` で始まる名前は、配布リストの管理アドレス。このアドレスは、メールエラーを処理する人にメールをリダイレクトします。
`owner-owner`	この別名は、エラーを戻す先の `owner-`*aliasname* の別名がない場合に使用されます。このアドレスは、メールエラーを処理する人にメールをリダイレクトします。このアドレスは、大量の別名を管理する任意のシステムで定義されます。
local`%`*domain*	パーセント記号 (`%`) は、メッセージがその宛先に着くと展開されるローカルアドレスを示します。ほとんどのメールシステムは、`%` 記号つきのメールボックス名を全メールアドレスとして翻訳します。`%` は `@` と置き換えられ、メールはそれに応じてリダイレクトされます。多くの人が `%` を使用しますが、これは正式な標準ではありません。この規則は、電子メールの世界では「パーセントハック」と呼ばれています。この機能は、メールに問題が起こった場合にデバッグに使用されることが多いです。

`sendmail` version 8 より、所有者の別名が存在する場合、グループの別名に送信されるメールの封筒の送信者は、所有者の別名から展開されるアドレスに変更されました。この変更によって、メールエラーは、送信者に返送されるのではなく、別名の所有者に送信されるようになりました。この変更によって、別名に送信されたメールは、別名の所有者から送信されたように見えます。次の別名の書式は、この変更に関連したいくつかの問題に対応します。

```
mygroup: :include:/pathname/mygroup.list
owner-mygroup: mygroup-request
mygroup-request: sandys, ignatz
```

この例では、`mygroup` の別名が、このグループの実際のメール別名です。`owner-mygroup` の別名は、エラーメッセージを受信します。`mygroup-request` の別名は、管理の要求に使用してください。この構造は、`mygroup` の別名に送信されたメールでは、封筒の送信者が `mygroup-request` に変更されることを意味します。

メール別名

「別名 (alias) 」とは、もう 1 つの別の名前を指します。電子メールでは、メールボックスの場所を割り当てたり、メールリストを定義したりするために別名を使用できます。作業マップについては、第 13 章「メールサービス (手順)」の 315 ページの「メール別名ファイルの管理 (作業マップ)」を参照してください。この章の 368 ページの「メール別名ファイル」も参照してください。

大きなサイトでは通常、メール別名は、メールボックスの場所を定義します。メール別名を提供することは、複数の部屋を占有する大きな会社の個人のアドレスに部屋番号を含めるようなものです。部屋番号を提供しない場合は、メールは中央アドレスに配信されます。部屋番号がなければ、ビルの内部のどこにメールを配信するかを特定するために余分な労力が必要になります。そして、誤りが発生する可能性も増加します。たとえば、同じ建物に Kevin Smith という名前の人が 2 人いる場合、一方だけがメールを受け取ることになる可能性があります。この問題を解決するには、それぞれの Kevin Smith のアドレスに部屋番号を追加する必要があります。

メールリストを作成するときは、なるべくドメインの場所に依存しないアドレスを使用してください。別名ファイルの移植性と柔軟性を高めるため、別名エントリをできるかぎり一般的でシステムに依存しない形式にしてください。たとえば、システム `mars` のドメイン `example.com` に `ignatz` というユーザー名がある場合、別名は `ignatz@mars` ではなく、`ignatz@example` としてください。ユーザー `ignatz` がシステム名を変更しても、`example` ドメインには存在し続ける場合、システム名の変更を反映するように別名ファイルを更新する必要はありません。

別名エントリを作成するときは、1 行ごとに 1 つの別名を入力します。ユーザーのシステム名を含むエントリは 1 つだけにしてください。たとえば、ユーザー `ignatz` には、次のエントリを作成できます。

```
ignatz: iggy.ignatz
iggyi: iggy.ignatz
iggy.ignatz: ignatz@mars
```

ローカル名やドメインに別名を作成できます。たとえば、システム `mars` にメールボックスがある、ドメイン `planets` 内のユーザー `fred` の別名エントリでは、NIS+ 別名テーブルに次のエントリを作成できます。

```
fred: fred@planets
```

ドメイン外のユーザーを含むメールリストを作成するときは、ユーザー名とドメイン名を持つ別名を作成してください。たとえば、`example.com` ドメインの `privet` システムに `smallberries` というユーザーが存在する場合は、`smallberries@example.com` という別名を作成します。送信者の電子メールアドレスは、メールがユーザードメイン外に発信されるときは、完全指定ドメイン名に自動的に変換されます。

次に、メール別名のファイルを作成して管理する方法を示します。

- NIS+ mail_aliases テーブル、NIS aliases マップ、または、ローカルの /etc/mail/aliases ファイルでグローバルに使用するメール別名を作成します。また、同じ別名ファイルを使用するメールリストを作成して管理することができます。
- メールサービスの構成によっては、NIS または NIS+ ネームサービスを使って別名を管理し、グローバルな aliases データベースを持てます。または、すべてのローカル /etc/mail/aliases ファイルを更新して、別名の同期を維持することもできます。
- また、ユーザー自身が別名を作成して使用できます。ユーザーは、別名をユーザーだけが使用できるようにローカル ~/.mailrc ファイルで作成することも、だれでも使用できるようにローカル /etc/mail/aliases ファイルで作成することもできます。通常、ユーザーは NIS や NIS+ 別名ファイルの作成および管理はできません。

ハードウェアコンポーネント

メールの構成に必要な 3 つの要素は、単一のシステムによって提供することも別々のシステムによって提供することもできます。

- 353 ページの「メールホスト」
- 354 ページの「メールサーバー」
- 355 ページの「メールクライアント」

ユーザーがドメイン外のネットワークと通信をするためには、4 番目の要素であるメールゲートウェイを追加する必要があります。詳細は、355 ページの「メールゲートウェイ」を参照してください。次の節では各ハードウェアコンポーネントについて説明しています。

メールホスト

「メールホスト」は、ネットワークのメインのメールマシンに指定するマシンです。メールホストはサイトにおいて、ほかのシステムでは配信できないメールを転送するためのマシンになります。hosts データベースにシステムをメールホストとして指定するには、ローカル /etc/hosts ファイルの IP アドレスの右に mailhost を追加します。または、ネームサービスのホストファイルに mailhost を同じように追加することもできます。作業手順については、301 ページの「メールホストを設定する方法」の 第 13 章「メールサービス (手順)」を参照してください。

メールホストの候補は、ネットワークからグローバルなインターネットネットワークへのルーターとして構成されたシステムです。詳細は、第 15 章「Solaris PPP 4.0 (概要)」、第 24 章「UUCP (概要)」、および『Solaris のシステム管理 (IP サービス)』の「IPv4 ルーターの構成」を参照してください。ローカルネットワークのどのシステムにもモデムがない場合は、システムの 1 つをメールホストに指定します。

サイトの中には、タイムシェアリング構成でネットワークに接続されていないスタンドアロンのマシンを使用するものがあります。具体的に言うと、スタンドアロンのマシンが、シリアルポートに接続された端末として機能する場合です。このような構成では、スタンドアロンのシステムをシングルシステムネットワークのメールホストに指定することで、電子メールを設定できます。288 ページの「ハードウェアコンポーネントの概要」の 第 12 章「メールサービス (概要)」に、典型的な電子メール構成を示す図があります。

メールサーバー

「メールボックス」は、特定のユーザーの電子メールを含む単一のファイルです。メールは、ローカルマシンまたはリモートサーバーのユーザーのメールボックスが存在するシステムに配信されます。「メールサーバー」は、`/var/mail` ディレクトリにユーザーのメールボックスを保持しているいずれかのシステムになります。作業手順については、297 ページの「メールサーバーを設定する方法」の 第 13 章「メールサービス (手順)」を参照してください。

メールサーバーはクライアントからすべてのメールをルーティングします。クライアントがメールを送信すると、メールサーバーは配信のためにそのメールをキューに入れます。メールがキューに入れられたら、ユーザーはこれらのメールメッセージを失わずに、クライアントをリブートしたり、電源を切ったりすることができます。受信者がクライアントからメールを受け取ると、メッセージの `From` 行のパスには、メールサーバー名が含まれます。受信者が応答すると、その応答はユーザーのメールボックスに送られます。メールサーバーとして適しているのは、ユーザーにホームディレクトリを提供するシステムか、定期的にバックアップされるシステムです。

メールサーバーがユーザーのローカルシステムでない場合、構成内で NFS ソフトウェアを使用するユーザーは、`root` アクセスがあれば、`/etc/vfstab` ファイルを使用することによって、`/var/mail` ディレクトリをマウントできます。それ以外の場合は、オートマウンタを使用できます。NFS サポートが利用できない場合、ユーザーはサーバーにログインしてメールを読み込めます。

ネットワーク上のユーザーが、オーディオファイル、DTP システムからのファイルなどほかの形式のファイルを送信する場合は、メールボックスのメールサーバーには、さらに多くの領域を割り当てる必要があります。

全メールボックス用に 1 台のメールサーバーを設定すると、バックアップ作業が簡単になります。メールが多くのシステムに分散しているとバックアップ作業が困難になる場合があります。1 台のサーバーに多くのメールボックスを保存する場合の短所は、サーバーに障害が発生した場合に多くのユーザーが影響を受けることです。ただし、十分なバックアップ機能を提供すれば、1 台のサーバーを採用する価値があります。

メールクライアント

「メールクライアント」は、メールサーバー上にメールボックスを持っている、メールサービスのユーザーです。メールクライアントにはさらに、/etc/mail/aliases ファイルで、メールボックスの位置を示すメール別名が設定されています。作業手順については、299 ページの「メールクライアントを設定する方法」の 第 13 章「メールサービス (手順)」を参照してください。

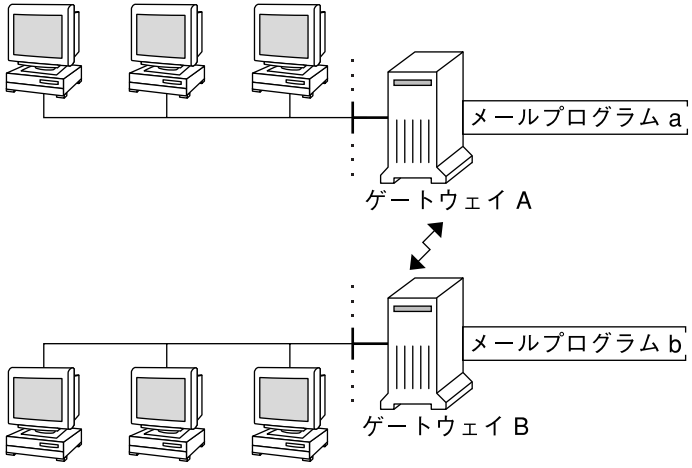
メールゲートウェイ

「メールゲートウェイ」は、異なる通信プロトコルを実行するネットワーク間の接続を処理したり、同じプロトコルを使用する異なるネットワーク間の通信を処理するマシンです。たとえば、メールゲートウェイでは、SNA (Systems Network Architecture) プロトコルセットを実行するネットワークに、TCP/IP ネットワークを接続する場合もあります。

設定のもっとも簡単なメールゲートウェイは、同じプロトコルかメールプログラムを使用する 2 つのネットワークを接続するものです。このシステムでは、sendmail がドメインで受信者を見つけられないアドレスのあるメールを処理します。メールゲートウェイがある場合、sendmail はメールゲートウェイを使用して、ドメイン外でメールの送受信を行います。

2 つのネットワーク間には、次の図に示すように内容の異なるメールプログラムを使ってメールゲートウェイを設定できます。この構成をサポートするには、メールゲートウェイシステムで sendmail.cf ファイルをカスタマイズする必要がありますが、これは困難で時間のかかる作業になる場合もあります。

図 14–1　異なる通信プロトコル間のゲートウェイ



インターネットに接続できるマシンがある場合は、そのマシンをメールゲートウェイとして構成できます。メールゲートウェイを構成するときは、まずサイトのセキュリティー要件を慎重に考慮する必要があります。社内ネットワークをほかのネットワークと接続するには、ファイアウォールゲートウェイを構築し、それをメールゲートウェイとして設定しなければならない場合があります。作業手順については、302 ページの「メールゲートウェイを設定する方法」の 第 13 章「メールサービス (手順)」を参照してください。

メールサービスのプログラムとファイル

メールサービスには、相互に対応する数多くのプログラムやデーモンが含まれています。ここでは、電子メールの管理に関連するファイル、プログラム、用語、および概念について説明します。

vacation ユーティリティーの拡張機能

Solaris 10 以降のリリースでは、vacation ユーティリティーが機能強化され、自動生成された応答をどの着信メッセージが受けるかをユーザーが指定できるようになりました。この拡張機能により、ユーザーは、知らない人と機密情報や連絡先を共有せずにすみます。スパマーや知らない人からのメッセージは、応答を受け取りません。

この拡張機能は、着信電子メールの送信者のアドレスを .vacation.filter ファイル内のドメインまたは電子メールアドレスのリストと付き合わせることによって機能します。このファイルは、ユーザーによって作成され、ユーザーのホームディレクトリにあります。ドメインまたは電子メールアドレスで一致するものがあると、応答が送られます。一致するものがなければ、応答は送られません。

.vacation.filter には、次のようなエントリが含まれます。

```
company.com
mydomain.com
onefriend@hisisp.com
anotherfriend@herisp.com
```

各行には、1 つのドメインまたは 1 つの電子メールアドレスが含まれます。1 つのエントリを 1 行に入力する必要があります。送信者の電子メールアドレスが電子メールアドレスエントリと一致するには、大文字と小文字の違いを除いて、完全に一致している必要があります。送信者のアドレスの文字が小文字であるか大文字であるかは無視されます。送信者の電子メールアドレスがドメインエントリと一致するには、一覧表示されているドメインに送信者のアドレスが含まれている必要があります。たとえば、`somebody@dept.company.com` と `someone@company.com` の両方が、`company.com` のドメインエントリと一致します。

詳細は、`vacation(1)` のマニュアルページを参照してください。

/usr/bin ディレクトリの内容

次の表にメールサービスに使用する `/usr/bin` ディレクトリの内容を示します。

名前	種類	説明
`aliasadm`	ファイル	NIS+ 別名マップを処理するプログラム。
`mail`	ファイル	ユーザーエージェント。
`mailcompat`	ファイル	メールを SunOS 4.1 メールボックスフォーマットに格納するフィルタ。
`mailq`	ファイル	メールキューの内容を一覧表示するプログラム。
`mailstats`	ファイル	`/etc/mail/statistics` ファイルに格納されたメール統計情報の読み込みに使用するプログラム (存在する場合のみ)。
`mailx`	ファイル	ユーザーエージェント。
`mconnect`	ファイル	アドレスの検証とデバッグのためメールプログラムに接続するプログラム。
`praliases`	ファイル	別名データベースを「ソースに展開」するコマンド。`praliases(1)` のマニュアルページにあるソース展開の情報を参照してください。
`rmail`	シンボリックリンク	`/usr/bin/mail` へのシンボリックリンク。メール送信だけに使用されるコマンド。
`vacation`	ファイル	メールへの自動応答を設定するコマンド。

/etc/mail ディレクトリの内容

次の表に、`/etc/mail` ディレクトリの内容を示します。

名前	種類	説明
Mail.rc	ファイル	mailx ユーザーエージェントのデフォルトの設定値。
aliases	ファイル	メール転送情報。
aliases.db	ファイル	newaliases の実行によって作成されるデフォルトのバイナリ形式のメール転送情報。
aliases.dir	ファイル	newaliases の実行によって作成されるバイナリ形式のメール転送情報。まだ使用できますが、Solaris 9 よりデフォルトでは使用できません。
aliases.pag	ファイル	newaliases の実行によって作成されるバイナリ形式のメール転送情報。まだ使用できますが、Solaris 9 よりデフォルトでは使用できません。
mailx.rc	ファイル	mailx ユーザーエージェントのデフォルトの設定値。
main.cf	シンボリックリンク	メインシステム用の構成ファイルのこの例から sendmail.cf へのシンボリックリンクが、下位互換性を確保するために提供されます。このファイルは、sendmail の version 8.13 では必要ありません。
relay-domains	ファイル	リレーを許容するすべてのドメインのリスト。デフォルトでは、ローカルドメインだけが使用できます。
sendmail.cf	ファイル	メールルーティング用の構成ファイル。
submit.cf	ファイル	メール配信プログラム (MSP) のための新しい構成ファイル。詳細は、390 ページの「sendmail の version 8.12 からの submit.cf 構成ファイル」を参照してください。
local-host-names	ファイル	メールホスト用の別名の数が多すぎるときに作成可能なオプションファイル。
helpfile	ファイル	SMTP HELP コマンドで使用するヘルプファイル。
sendmail.pid	ファイル	リスニングデーモンの PID を一覧表示し、現在は /var/run にあるファイル。
statistics	ファイル	sendmail 統計ファイル。このファイルが存在すると、sendmail は各メールプログラムのトラフィック量をログに記録します。このファイルは以前 sendmail.st と呼ばれていました。
subsidiary.cf	シンボリックリンク	サブシステム用の構成ファイルのこの例から sendmail.cf へのシンボリックリンクが、下位互換性を確保するために提供されます。このファイルは、sendmail の version 8.13 では必要ありません。

名前	種類	説明
trusted-users	ファイル	特定のメール操作を実行するための信頼を与えられたユーザーを一覧表示するファイル (各行 1 ユーザー)。デフォルトでは、root だけがこのファイルに入っています。信頼されていないユーザーが特定のメール操作を実行すると、X-Authentication-Warning: header being added to a message という警告が生成されます。

/etc/mail/cf ディレクトリの内容

/etc/mail ディレクトリには、sendmail.cf ファイルを構築するために必要なすべてのファイルを含む cf というサブディレクトリがあります。表 14–9 に cf ディレクトリの内容を示します。

Solaris 10 以降のリリースでは、読み取り専用の /usr ファイルシステムをサポートするために、/usr/lib/mail ディレクトリの内容が /etc/mail/cf ディレクトリに移動されました。ただし、例外があります。シェルスクリプト /usr/lib/mail/sh/check-hostname および /usr/lib/mail/sh/check-permissions は、/usr/sbin ディレクトリに置かれるようになりました。362 ページの「メールサービスに使用するその他のファイル」を参照してください。下位互換性を確保するために、シンボリックリンクが各ファイルの新しい位置を示します。

表 14–9 メールサービスに利用する /etc/mail/cf ディレクトリの内容

名前	種類	説明
README	ファイル	構成ファイルを説明します。
cf/main.cf	シンボリックリンク	Solaris 10 リリース以降、このファイル名は cf/sendmail.cf にリンクされます。このファイルはメインの構成ファイルとして使用されます。
cf/main.mc	シンボリックリンク	Solaris 10 リリース以降、このファイル名は cf/sendmail.mc にリンクされます。このファイルは、メインの構成ファイルを作成するためのファイルでした。
cf/Makefile	ファイル	新しい構成ファイルを作成する場合の規則を提供します。
cf/submit.cf	ファイル	メッセージを送信するためのメール配信プログラム (MSP) のための構成ファイルです。

表 14–9 メールサービスに利用する /etc/mail/cf ディレクトリの内容 (続き)

名前	種類	説明
cf/submit.mc	ファイル	submit.cf ファイルの構築に使用されるファイルです。このファイルは、メール配信プログラム (MSP) のための m4 マクロを定義します。
cf/sendmail.cf	ファイル	sendmail のためのメインの構成ファイルです。
cf/sendmail.mc	ファイル	sendmail.cf ファイルの生成に使用される m4 マクロが含まれています。
cf/subsidiary.cf	シンボリックリンク	Solaris 10 リリース以降、このファイル名は cf/sendmail.cf にリンクされます。別のホストから /var/mail を NFS マウントするホストのための構成ファイルとして使用されます。
cf/subsidiary.mc	シンボリックリンク	Solaris 10 リリース以降、このファイル名は cf/sendmail.mc にリンクされます。このファイルには、subsidiary.cf ファイルの生成に使用された m4 マクロが含まれています。
domain	ディレクトリ	サイトに依存するサブドメインの説明を提供します。
domain/generic.m4	ファイル	Berkeley Software Distribution からの汎用ドメインファイルです。
domain/solaris-antispam.m4	ファイル	sendmail 関数を以前の Solaris 版の sendmail のようにする変更を伴うドメインファイルです。ただし、リレーは完全に無効に設定されるので、ホスト名のない送信者アドレスは拒否され、解決されないドメインは拒否されます。
domain/solaris-generic.m4	ファイル	sendmail 関数を以前の Solaris 版の sendmail のようにする変更を伴うデフォルトのドメインファイルです。
feature	ディレクトリ	特定のホスト用の特別な機能の定義を含みます。機能の詳細な説明は README を参照してください。
m4	ディレクトリ	サイトに依存しないインクルードファイルを含みます。
mailer	ディレクトリ	local、smtp、uucp などのメールプログラムの定義を含みます。

表 14–9 メールサービスに利用する /etc/mail/cf ディレクトリの内容 (続き)

名前	種類	説明
main-v7sun.mc	ファイル	廃止: Solaris 10 リリース以降、このファイル名は cf/sendmail.mc に変更されました。
ostype	ディレクトリ	各種のオペレーティングシステム環境を説明します。
ostype/solaris2.m4	ファイル	デフォルトのローカルメールプログラムを mail.local に定義します。
ostype/solaris2.ml.m4	ファイル	デフォルトのローカルメールプログラムを mail.local に定義します。
ostype/solaris2.pre5.m4	ファイル	ローカルメールプログラムを mail に定義します。
ostype/solaris8.m4	ファイル	ローカルメールプログラムを LMTP モードで mail.local に定義し、IPv6 を有効にし、sendmail.pid ファイルのディレクトリとして /var/run を指定します。
subsidiary-v7sun.mc	ファイル	廃止: Solaris 10 リリース以降、このファイル名は cf/sendmail.mc に変更されました。

/usr/lib ディレクトリの内容

次の表にメールサービスに使用する /usr/lib ディレクトリの内容を示します。

表 14–10 /usr/lib ディレクトリの内容

名前	種類	説明
mail.local	ファイル	メールボックスにメールを配信するメールプログラム。
sendmail	ファイル	メール転送エージェントとしても知られるルーティングプログラム。
smrsh	ファイル	sendmail の |program 構文を使用して /var/adm/sm.bin ディレクトリにあるプログラムに対して sendmail を実行できるプログラムを制限するシェルプログラム (sendmail に限定されたシェル)。/var/adm/sm.bin に含める内容については、smrsh(1M) のマニュアルページを参照してください。有効にするには、この m4 コマンドと FEATURE('smrsh') を mc ファイルに含めます。

表 14–10 /usr/lib ディレクトリの内容 (続き)

名前	種類	説明
mail	シンボリックリンク	シンボリックリンクは /etc/mail/cf ディレクトリを示します。詳細は、359 ページの「/etc/mail/cf ディレクトリの内容」を参照してください。

メールサービスに使用するその他のファイル

メールサービスは、その他のいくつかのファイルおよびディレクトリを使用します。これらを表 14–11 に示します。

表 14–11 メールサービスに使用するその他のファイル

名前	種類	説明
/etc/default/sendmail	ファイル	sendmail の起動スクリプトの環境変数を一覧表示します。
/etc/shells	ファイル	有効なログインシェルを一覧表示します。
/etc/mail/cf/sh	ディレクトリ	m4 構築プロセスと移行補助に使用するシェルスクリプトを含みます。
/usr/sbin/check-permissions	ファイル	:include: 別名と .forward ファイルのアクセス権、および正確なアクセス権に必要なこれらの親ディレクトリのパスを確認します。
/usr/sbin/check-hostname	ファイル	sendmail が完全指定のホスト名を判別できることを確認します。
/usr/sbin/editmap	ファイル	sendmail のデータベースマップの単一のレコードに対してクエリーを実行して編集します。
/usr/sbin/in.comsat	ファイル	メール通知デーモン。
/usr/sbin/makemap	ファイル	入力されたマップのバイナリ形式を構築します。
/usr/sbin/newaliases	シンボリックリンク	/usr/lib/sendmail へのシンボリックリンク。別名データベースのバイナリ形式を作成するために使用します。以前は /usr/bin にありました。
/usr/sbin/syslogd	ファイル	sendmail が使用するエラーメッセージログをとるデーモン。

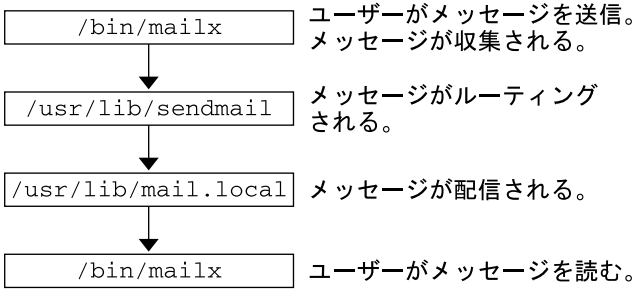
表 14–11　メールサービスに使用するその他のファイル　(続き)

名前	種類	説明
/usr/sbin/etrn	ファイル	クライアント側リモートメールキューを起動するための Perl スクリプト。
/usr/dt/bin/dtmail	ファイル	CDE メールユーザーエージェント。
/var/mail/*mailbox1*、 /var/mail/*mailbox2*	ファイル	配信されたメールのメールボックス。
/var/spool/clientmqueue	ディレクトリ	クライアントデーモンによって配信されるメールの記憶領域。
/var/spool/mqueue	ディレクトリ	マスターデーモンによって配信されるメールの記憶領域。
/var/run/sendmail.pid	ファイル	リスニングデーモンの PID を表示するファイル。

メールプログラム間の相互作用

メールサービスは次のプログラムで構成され、図 14–2 のように作用します。

図 14–2　メールプログラム間の相互作用



次に、メールプログラムの相互作用について説明します。

1. ユーザーは、mailx などのプログラムを使ってメッセージを送信します。詳細は、mailx(1) のマニュアルページを参照してください。
2. メッセージは、そのメッセージを生成したプログラムによって収集され、sendmail デーモンに渡されます。

3. sendmail デーモンがメッセージのアドレスを識別可能な各部に分割して解析します。sendmail デーモンは、/etc/mail/sendmail.cf という構成ファイルの情報を使ってネットワーク名の構文、別名、転送情報、およびネットワークトポロジを決定します。sendmail はこの情報を使用して、メッセージが受信者に到達する経路を決定します。
4. sendmail デーモンはメッセージを適切なシステムに渡します。
5. ローカルシステムの /usr/lib/mail.local プログラムは、メッセージの受信者の /var/mail/*username* ディレクトリのメールボックスにメールを配信します。
6. 受信者は、メールが届いたことが通知されるので、mail、mailx などのプログラムを使用してメールを受け取ります。

sendmail プログラム

次に、sendmail プログラムの機能の一部を示します。

- sendmail は、TCP/IP や UUCP などの異なる通信プロトコルを使用できます。
- sendmail は、SMTP サーバー、メッセージキュー、メーリングリストを実装します。
- sendmail は、次の命名規則に準拠したパターンマッチングシステムを使って名前の解釈を制御します。
  - ドメインベースの命名規則。ドメインの手法は、物理的なネーミングと論理的なネーミングの問題を分離します。詳細は、348 ページの「メールアドレス」を参照してください。
  - ほかのネットワークのホストからローカルに見えるネットワーク名を提供するなどの即席のテクニック。
  - 任意 (以前) の命名構文。
  - 異種の命名スキーム。

Solaris オペレーティングシステムでは、sendmail プログラムをメールルーターとして使用します。次に、機能の一部を示します。

- sendmail は、mail.local や procmail などのローカル配信エージェントとの間で、電子メールメッセージの受信や配信を行う役割を果たします。
- sendmail はメール転送エージェントであり、mailx や Mozilla Mail などのユーザーエージェントからメッセージを受け取り、そのメッセージをインターネット経由でその宛先までルーティングします。
- sendmail は、次の要領でユーザーが送信する電子メールメッセージを制御します。
  - 受信者のアドレスを確認します。
  - 適切な配信プログラムを選択します。
  - アドレスを配信エージェントが処理できるフォーマットに書き換えます。

- 必要に応じて、メールヘッダーをフォーマットし直します。
- 最後に転送されたメッセージをメール配信プログラムに渡します。

sendmail の詳細は、次のトピックを参照してください。

- 365 ページの「sendmail とその再ルーティングメカニズム」
- 366 ページの「sendmail プログラムの機能」
- 367 ページの「sendmail 構成ファイル」

sendmail とその再ルーティングメカニズム

sendmail プログラムでは、メールルーティングに必要な 3 つのメカニズムをサポートしています。適切なメカニズムは、変更の種類によって決まります。

- サーバーの変更
- ドメイン全体の変更
- 単独のユーザーの変更

さらに、選択する再ルーティングメカニズムによって必要な管理レベルが異なります。次のオプションを考慮してください。

1. 再ルーティングメカニズムの 1 つは「別名」です。

   別名を使用すれば、使用するファイルの種類に基づいて、サーバー全体またはネームサービス全体をベースにしてアドレス名をマップできます。

   次に、ネームサービスの別名の長所と短所を示します。

   - ネームサービス別名ファイルを使用すれば、メール再ルーティングの変更を単一のソースで管理できます。ただし、ネームサービスの別名指定では、再ルーティングの変更を伝達する際に遅延が起こります。
   - 通常、ネームサービスの管理は、特定のシステム管理者グループに制限されます。一般ユーザーは、このファイルを管理しません。

   次に、サーバー別名ファイルを使用する際の長所と短所を示します。

   - サーバー別名ファイルを使用すれば、指定されたサーバーの root になることができる任意のユーザーが再ルーティングを管理できます。
   - サーバー別名指定は、再ルーティングの変更を伝達する際の遅延はほとんどありません。
   - 変更はローカルサーバーだけに影響します。ほとんどのメールが単一のサーバーに送信される場合は、影響が少なくなります。ただし、この変更を多くのメールサーバーに伝達する必要がある場合は、ネームサービスの別名指定を使用します。
   - 一般ユーザーは、この変更を管理しません。

   詳細は、この章の 368 ページの「メール別名ファイル」を参照してください。作業マップについては、第 13 章「メールサービス (手順)」の 315 ページの「メール別名ファイルの管理 (作業マップ)」を参照してください。

2. 次のメカニズムは、「転送」です。

   このメカニズムでは、ユーザーがメールの再ルーティングを管理できます。ローカルユーザーは、受信メールを次の対象に再ルーティングできます。

   - 別のメールボックス
   - 別のメールプログラム
   - 別のメールホスト

   このメカニズムは、`.forward` ファイルによってサポートされます。.forward ファイルの詳細は、この章の 371 ページの「`.forward` ファイル」を参照してください。作業マップについては、330 ページの「`.forward` ファイルの管理 (作業マップ)」の 第 13 章「メールサービス (手順)」を参照してください。

3. 最後のメカニズムは、「取り込み」です。

   このメカニズムでは、`root` アクセス権を持たないユーザーも別名リストを保守できます。このメカニズムを提供するには、`root` ユーザーは、サーバー上の別名ファイル内に適切なエントリを作成する必要があります。このエントリが作成されると、ユーザーは必要に応じてメールをルーティングし直すことができるようになります。取り込みの詳細は、この章の 369 ページの「`/etc/mail/aliases` ファイル」を参照してください。作業マップについては、第 13 章「メールサービス (手順)」の 315 ページの「メール別名ファイルの管理 (作業マップ)」を参照してください。

注 – `/usr/bin/mailx` のようなメールを読み取るプログラムは、プログラム自身の別名を持つことができ、それらはメッセージが `sendmail` に達する前に展開されます。`sendmail` の別名は、ローカルファイル、NIS、NIS+ など、さまざまなネームサービスソースからのものでもかまいません。検索順序は `nsswitch.conf` ファイルによって決定されます。`nsswitch.conf(4)` のマニュアルページを参照してください。

`sendmail` プログラムの機能

`sendmail` プログラムには、次のような機能があります。

- `sendmail` は、信頼性の高いプログラムです。すべてのメッセージを正しく配信するように設計されています。どのようなメッセージも完全に失われることはありません。
- `sendmail` は、既存のソフトウェアを配信に随時使用します。たとえば、ユーザーは、メール生成プログラムおよびメール送信プログラムと対話します。メール送信が依頼されると、メール生成プログラムは `sendmail` を呼び出し、sendmail は適切なメールプログラムにメッセージを送信します。発信者の一部はネットワークサーバーであったり、またメールプログラムの一部はネットワーククライアントであるため、`sendmail` は、インターネットメールゲートウェイとしても使用できます。このプロセスの詳細は、363 ページの「メールプログラム間の相互作用」を参照してください。

- sendmail は、複数のネットワークなど、複雑な環境を処理するように構成できます。sendmail は、アドレスとその構文の内容を確認し、どのメールプログラムを使用するかを判断します。
- sendmail は、構成情報をコードにコンパイルする代わりに、構成ファイルを使ってメール構成を制御します。
- ユーザーは独自のメーリングリストを管理できます。さらに各ユーザーは、ドメイン全体で有効な別名ファイル (通常、NIS または NIS+ によって管理されるドメイン全体の別名の中にある) を修正することなく自分自身の転送メカニズムを指定できます。
- 各ユーザーは、受信メールを処理するカスタムメールプログラムを指定できます。カスタムメールプログラムは、次のようなメッセージを返すこともできます。「私は休暇中です」。詳細は、vacation(1) のマニュアルページを参照してください。
- sendmail は、1 つのホストでアドレスを処理し、ネットワークトラフィックを削減します。

sendmail 構成ファイル

「構成ファイル」は、sendmail がその機能を実行する方法を制御します。構成ファイルにより、配信エージェント、アドレスの変換の規則、およびメールヘッダーのフォーマットが選択されます。sendmail プログラムは、/etc/mail/sendmail.cf ファイルの情報を使用して、その機能を実行します。

Solaris オペレーティングシステムには、/etc/mail ディレクトリに次の 2 つのデフォルト構成ファイルがあります。

1. デーモンモードで sendmail を実行するために使用する sendmail.cf 構成ファイル。
2. デーモンモードの代わりにメール配信プログラムモードで sendmailを実行するために使用する submit.cf 構成ファイル。詳細は、390 ページの「sendmail の version 8.12 からの submit.cf 構成ファイル」を参照してください。

メールクライアント、メールサーバー、メールホスト、メールゲートウェイを設定するときは、次を考慮してください。

- メールクライアントまたはメールサーバーについては、デフォルト構成ファイルを設定または編集する必要はありません。
- メールホストやメールゲートウェイを設定するには、メール設定に必要な中継メールプログラムおよび中継ホストのパラメータを設定します。作業手順については、296 ページの「メールサービスの設定 (作業マップ)」の 305 ページの「sendmail 構成を変更する」または第 13 章「メールサービス (手順)」を参照してください。sendmail version 8.13 では、main.cf ファイルは必要ありません。

次に、サイトの要求に応じて変更が可能な構成パラメータをいくつか説明します。

- 次の情報を指定する時間値。
  - 読み取りのタイムアウト。
  - メッセージが送信者に返送されるまで、配信されずにキューに置かれる時間。402 ページの「sendmail の version 8.12 から追加されたキューの機能」を参照してください。作業マップについては、327 ページの「キューディレクトリの管理 (作業マップ)」を参照してください。
- メール配信の速度を指定する配信 (delivery) モード。
- ビジー期間中の効率を高めるためのロード制限。これらのパラメータは、sendmail が、長いメッセージ、多くの受信者へのメッセージ、および長時間ダウンしているサイトへのメッセージを配信しないようにします。
- ログ出力する問題の種類を指定するログレベル。

メール別名ファイル

別名を保守するには、次のファイル、マップ、またはテーブルを使用します。

- 368 ページの「.mailrc の別名」
- 369 ページの「/etc/mail/aliases ファイル」
- 370 ページの「NIS aliases マップ」
- 371 ページの「NIS+ mail_aliases テーブル」

別名を保守する方法は、だれが使用し、だれが変更するかによって決まります。別名のタイプにはそれぞれ固有の形式要件があります。

関連する作業については、315 ページの「メール別名ファイルの管理 (作業マップ)」の 第 13 章「メールサービス (手順)」を参照してください。

.mailrc の別名

.mailrc ファイルのリストに入っている別名には、ファイルを所有するユーザーしかアクセスできません。この制限により、ユーザーは自分で制御し、所有者だけが使用できる別名を作成できます。.mailrc ファイルの別名の形式は、次のとおりです。

alias *aliasname value value value* ...

aliasname は、ユーザーがメールの送信時に使用する名前であり、*value* は有効な電子メールアドレスです。

ユーザーが scott に個人的な別名を作成し、それがネームサービスの scott の電子メールアドレスと一致しない場合、エラーが発生します。そのユーザーが作成したメールにユーザーが返信しようとするときに、メールが間違ったユーザーに転送されることになります。これを回避するには、別の別名命名方式を使用する以外にありません。

/etc/mail/aliases ファイル

/etc/mail/aliases ファイルで作成したいずれの別名も、その別名の名前と、ファイルを含んでいるシステムのホスト名を知っているユーザーならだれでも使用できます。ローカルの /etc/mail/aliases ファイルの配布リストの形式は、次のとおりです。

aliasname: *value,value,value* ...

aliasname は、ユーザーがこの別名にメールを送信するときに使用する名前で、*value* は有効な電子メールアドレスになります。

使用するネットワークがネームサービスを実行していない場合は、各システムの /etc/mail/aliases ファイルにすべてのメールクライアントのエントリを入れておく必要があります。各システムのファイルを編集するか、1 つのシステムのファイルを編集してからそのファイルをほかのシステムに個々にコピーします。

/etc/mail/aliases ファイルの別名は、テキスト形式で保存されます。/etc/mail/aliases ファイルを編集するときには、newaliases プログラムを実行する必要があります。このプログラムは、データベースをコンパイルし直し、sendmail プログラムが別名をバイナリ形式で使用できるようにします。作業手順については、322 ページの「ローカルメール別名ファイルを設定する方法」の第 13 章「メールサービス (手順)」を参照してください。それ以外の場合、Solaris 管理コンソールの「メーリングリスト」機能を使ってローカルの /etc ファイルに保存されているメール別名を管理できます。

現在のホスト名やホスト名なしなどのローカル名のみに別名を作成できます。たとえば、システム saturn にメールボックスのあるユーザー ignatz の別名エントリには、/etc/mail/aliases ファイルの次のエントリが入ります。

```
ignatz: ignatz@saturn
```

各メールサーバーに管理アカウントを作成する必要があります。管理アカウントを作成するには、メールサーバーのメールボックスを root に割り当て、root のエントリを /etc/mail/aliases ファイルに追加します。たとえば、システム saturn がメールボックスサーバーの場合は、エントリ root: sysadmin@saturn を /etc/mail/aliases ファイルに追加します。

通常は、root ユーザーだけがこのファイルを編集できます。ただし、Solaris 管理コンソールを使用する場合は、sysadmin グループであるグループ 14 のすべてのユーザーが、ローカルファイルを変更できます。または、次のエントリを作成します。

aliasname: :include:*/path/aliasfile*

aliasname は、ユーザーがメールを送信するときに使用する名前であり、*/path/aliasfile* は別名リストを含むファイルへのフルパスになります。別名ファイルには、各行に 1 つの電子メールエントリを入れ、その他の表記は付けないでください。

user1@host1
user2@host2

`/etc/mail/aliases` に追加のメールファイルを定義して、ログやバックアップコピーの管理もできます。次のエントリでは、*aliasname* に送信されるすべてのメールを *filename* 内に格納します。

aliasname: `/home/backup/`*filename*

また、ほかのプロセスにメールを回送することもできます。次の例のように入力すると、メールメッセージのコピーが *filename* 内に格納され、コピーがプリントされます。

aliasname: `"|tee -a /home/backup/`*filename* `|lp"`

作業マップについては、第 13 章「メールサービス (手順)」の 315 ページの「メール別名ファイルの管理 (作業マップ)」を参照してください。

NIS aliases マップ

ローカルドメインのすべてのユーザーは、NIS `aliases` マップのエントリを使用できます。`sendmail` プログラムは、ローカルの `/etc/mail/aliases` ファイルの代わりに NIS `aliases` マップを使って送信アドレスを決定できるからです。詳細は、`nsswitch.conf(4)` のマニュアルページを参照してください。

NIS `aliases` マップの別名は、次のようになります。

aliasname: *value,value,value* ...

aliasname は、ユーザーがメールの送信時に使用する名前であり、*value* は有効な電子メールアドレスです。

NIS `aliases` マップには、すべてのメールクライアント用のエントリを含めてください。一般にこれらのエントリを変更できるのは、NIS マスターの root ユーザーだけです。この種の別名は頻繁に変更される場合には適していません。次の構文例のように、ほかの別名ファイルをポイントする場合には役立ちます。

aliasname: *aliasname@host*

aliasname はユーザーがメールを送信するときに使用する名前であり、*host* は `/etc/mail/alias` ファイルを含むサーバー用のホスト名です。

作業手順については、321 ページの「NIS `mail.aliases` マップを設定する方法」の第 13 章「メールサービス (手順)」を参照してください。

NIS+ mail_aliases テーブル

NIS+ mail_aliases テーブルには、名前が含まれており、それによってローカルドメインにおけるシステムや個人が登録されています。sendmail プログラムは、ローカルの /etc/mail/aliases ファイルの代わりに NIS+ mail_aliases テーブルを使用して、メールアドレスを決定できます。詳細は、aliasadm(1M) と nsswitch.conf(4) のマニュアルページを参照してください。

NIS+ mail_aliases テーブルの別名は次のようになります。

alias: *expansion* # ["*options*" # "*comments*"]

表 14–12 に、NIS+ mail_aliases テーブルの 4 つの列を示します。

表 14–12 NIS+ mail_aliases テーブルの列

列	説明
別名	別名の名前
expansion	sendmail /etc/mail/aliases ファイルに表示される別名の値または別名のリスト
options	今後の使用のために予約された列
comments	個々の別名のコメントのための列

NIS+ mail_aliases テーブルには、すべてのメールクライアントのエントリを含めてください。NIS+ aliases テーブルでは、aliasadm コマンドで、エントリの表示、作成、変更、および削除ができます。aliasadm コマンドを使用するには、aliases テーブルを所有する NIS+ グループのメンバーでなければなりません。作業手順については、315 ページの「メール別名ファイルの管理 (作業マップ)」の 第 13 章「メールサービス (手順)」を参照してください。Solaris 管理コンソールを使用して NIS+ メール別名を管理することもできます。

注 – 新規の NIS+ aliases テーブルを作成する場合は、エントリを作成する前にテーブルを初期設定する必要があります。テーブルが存在するときは、初期設定は不要です。

.forward ファイル

ホームディレクトリに .forward ファイルを作成すると、sendmail およびその他のプログラムは、メールのリダイレクトや送信にこのファイルを使用できます。次の節を参照してください。

- 372 ページの「回避すべき状況」

作業マップについては、330 ページの「.forward ファイルの管理 (作業マップ)」の第 13 章「メールサービス (手順)」を参照してください。

回避すべき状況

次に、容易に回避または修復できる状況を示します。

- メールが宛先のアドレスに配信されない場合は、ユーザーの .forward ファイルをチェックしてください。ユーザーは、ホームディレクトリ host1 に .forward ファイルを配置して、user@host2 にメールを転送するようにしたのかもしれません。host2 にメールが着信すると、sendmail は NIS または NIS+ 別名に user があるかどうかを確認し、メッセージを user@host1 に返送します。このルーティングによってループが発生し、バウンスメールの増加を引き起こしています。
- セキュリティーの問題を予防するために、root または bin アカウントに .forward ファイルを決して置かないでください。必要な場合は、代わりに aliases ファイルを使ってメールを転送してください。

.forward ファイルの制御

メール配信で .forward ファイルを有効に使用するために、アクセス権などの次の設定が正しく適用されていることを確認してください。

- .forward ファイルへの書き込みは、ファイルの所有者に制限されます。この制限によって、ほかのユーザーがセキュリティーに反することを防止します。
- ホームディレクトリの親パスは root だけが所有し、root だけが書き込めるようにする必要があります。たとえば、.forward ファイルが /export/home/terry にある場合、/export および /export/home は root が所有し、root だけが書き込めるようにします。
- また実際のホームディレクトリに書き込めるのは、そのユーザーだけであるべきです。
- .forward ファイルをシンボリックリンクにすることはできません。また、複数のハードリンクを持つこともできません。

.forward.*hostname* ファイル

.forward.*hostname* ファイルを作成すれば、特定のホストに送信されるメールをリダイレクトできます。たとえば、ユーザーの別名が sandy@phoenix.example.com から sandy@example.com に変更された場合は、sandy のホームディレクトリに .forward.phoenix ファイルを置きます。

```
% cat .forward.phoenix
sandy@example.com
"|/usr/bin/vacation sandy"
% cat .vacation.msg
From: sandy@example.com (via the vacation program)
Subject: my alias has changed

My alias has changed to sandy@example.com.
Please use this alias in the future.
The mail that I just received from you
has been forwarded to my new address.

Sandy
```

この例では、メールが正しい宛先に転送され、送信者には別名の変更が通知されます。vacation プログラムではメッセージファイルは 1 つしか使用できないため、この場合 1 回につき 1 つのメッセージしか転送できません。ただし、メッセージが特定のホストに限定されない場合、.forward ファイルで複数のホストに同じ休暇メッセージファイルを使用できます。

.forward+*detail* ファイル

転送メカニズムの拡張機能にはこの他に、.forward+*detail* ファイルがあります。*detail* 文字列には、演算子文字を除く任意の文字を使用できます。演算子文字は、.:%&!^[]+ です。この種のファイルを使用すれば、ほかのユーザーが電子メールアドレスを無断で使用しているかどうかを確認できます。たとえば、あるユーザーが、だれかに電子メールアドレス sandy+test1@example.com を使用するように指示した場合、ユーザーは、この別名に配信されるメールを、アドレスに送信されるメールの中から識別できます。デフォルトにより、sandy+test1@example.com の別名に送信されたメールはすべて、この別名と .forward+*detail* ファイルと突き合わせて検査されます。ここで一致しない場合は、そのメールは最終的に sandy@example.com に配信されますが、ユーザーは、これらのメールの To: メールヘッダー内の変更箇所を調べることができます。

/etc/default/sendmail ファイル

このファイルは、sendmail のための初期設定用オプションを保存し、ホストをアップグレードしたときにオプションが除去されないようにするために使用します。次の変数を使用することができます。

CLIENTOPTIONS="*string*"
: クライアントデーモンで使用する追加オプションを選択します。クライアントデーモンは、クライアントだけのキュー (/var/spool/clientmqueue) の内容を確認し、クライアントキューランナーとして動作します。構文の検査は行われないため、この変数を変更するときは間違えないように注意してください。

CLIENTQUEUEINTERVAL=#
: CLIENTQUEUEINTERVAL には、QUEUEINTERVAL オプションと同様に、メールキューの実行間隔を設定します。ただし、CLIENTQUEUEINTERVAL オプションは、マスターデーモンではなくクライアントデーモンの機能を制御します。一般に、マスターデーモンはすべてのメッセージを SMTP ポートに配信できます。ただし、メッセージ負荷が高すぎる場合、またはマスターデーモンが実行されていない場合、メッセージはクライアントだけのキューである /var/spool/clientmqueue に入ります。次に、クライアントだけのキューをチェックするクライアントデーモンがクライアントキューを処理します。

ETRN_HOSTS="*string*"
: SMTP クライアントとサーバーが、定期的なキューの実行を待たずに即座に対話を実行できるようにします。サーバーは、指定されたホストに送信されるキューを即座に配信できます。詳細は、etrn(1M) のマニュアルページを参照してください。

MODE=-bd
: sendmail を起動するためのモードを選択します。-bd オプションを使用するか、未定義のままにしておきます。

OPTIONS=*string*
: マスターデーモンで使用される追加オプションを選択します。構文の検査は行われないため、この変数を変更するときは間違えないように注意してください。

QUEUEINTERVAL=#
: マスターデーモンのメールキューの実行間隔を設定します。# は正の整数とし、そのあとに秒の場合は s、分の場合は m、時の場合は h、日の場合は d、週の場合は w を付けます。この構文は sendmail の起動前に確認されます。この間隔が負の場合、またはエントリの最後の文字が不適当な場合、この間隔は無視され、sendmail は 15 分のキュー間隔で起動します。

QUEUEOPTIONS=p
: キューを実行するたびに新しいキューランナーを作成する代わりに、各実行の間に休止する単一の永続的なキューランナーを使用できるようにします。このオプションに設定可能な値は p だけです。p 以外に設定すると、このオプションは無効になります。

メールアドレスとメールルーティング

配信時にメールメッセージがたどる経路は、クライアントシステムの設定とメールドメインのトポロジによって異なります。メールホストやメールドメインの各追加レベルでは、別名のもう 1 つの解釈を追加できますが、ルーティングプロセスは基本的にほとんどのホストで同じになります。

クライアントシステムは、メールをローカルに受信できるようにセットアップできます。メールをローカルで受信することは、ローカルモードでの sendmail の実行と

して知られています。すべてのメールサーバーと一部のクライアントでは、ローカルモードがデフォルトです。ローカルモードのメールサーバーまたはクライアントでは、メッセージは次の要領でルーティングされます。

注 – 次の例では、sendmail.cf ファイルに設定されたデフォルトの規則を使用することを前提にしています。

---

1. 可能な場合はメール別名を展開し、ローカルのルーティングプロセスを再起動します。

   ネームサービスでメール別名を確認し、見つかった場合に新しい値と置換することで、メールアドレスが展開されます。次にこの新しい別名が再度確認されます。

2. メールがローカルの場合、/usr/lib/mail.local に配信されます。

   メールはローカルのメールボックスに配信されます。

3. メールアドレスがこのメールドメインにホストを含んでいると、そのホストにメールを配信します。

4. アドレスがこのドメインにホストを含んでいない場合、メールホストにメールを転送します。

   メールホストはメールサーバーと同じルーティングプロセスを使用します。ただし、メールホストはホスト名に加えて、ドメイン名が宛先になっているメールも受信できます。

sendmail とネームサービスの相互作用

ここでは、 sendmail とネームサービスに適用されるドメイン名について説明します。さらに、ネームサービスを有効に利用するための規則、および sendmail とネームサービスの相互作用について説明します。詳細は、次のトピックを参照してください。

- 376 ページの「sendmail.cf とメールドメイン」
- 376 ページの「sendmail とネームサービス」
- 377 ページの「NIS と sendmail との相互作用」
- 378 ページの「sendmail と NIS および DNS との相互作用」
- 379 ページの「NIS+ と sendmail との相互作用」
- 380 ページの「sendmail と NIS+ および DNS との相互作用」

関連する作業については、304 ページの「sendmail で DNS を使用する方法」の 315 ページの「メール別名ファイルの管理 (作業マップ)」または第 13 章「メールサービス (手順)」を参照してください。

sendmail.cf とメールドメイン

標準の sendmail.cf ファイルは、メールドメインを使ってメールを直接配信するか、あるいはメールホストを経由して配信するかを決定します。ドメイン内メールは直接 SMTP 接続経由で配信され、ドメイン間メールはメールホストに送られます。

セキュリティーの高いネットワークでは、ほんの少数の選ばれたホストだけが、外部宛てのパケットを生成する権限を与えられています。ホストがメールドメインの外部のリモートホストの IP アドレスを持っている場合も、SMTP 接続の確立は保証されません。標準の sendmail.cf では次のことを仮定しています。

- 現在のホストは、パケットを直接メールドメイン外のホストに送信する権限がない。
- メールホストは、パケットを外部ホストに直接送信できる認可されたホストにメールを転送できる。実際には、メールホストが認可されたホストになることがある。

このように仮定すると、ドメイン間メールの配信または転送はメールホスト側の責任です。

sendmail とネームサービス

sendmail は各種の要件をネームサービスに課します。これらの要件の理解を深めるために、この節では、まずメールドメインからネームサービスドメインへの関係について説明します。その次に個々の要件について説明します。次を参照してください。

- 376 ページの「メールドメインとネームサービスドメイン」
- 377 ページの「ネームサービスの要件」
- NIS+(1)、nisaddent(1M)、および nsswitch.conf(4) のマニュアルページ

メールドメインとネームサービスドメイン

メールドメイン名はネームサービスドメイン名の接尾辞の 1 つでなければなりません。たとえば、ネームサービスのドメイン名が「A.B.C.D」ならば、メールドメイン名は次のうちのいずれかです。

- A.B.C.D
- B.C.D
- C.D
- D

メールドメイン名は、最初の確立時には、多くの場合、ネームサービスドメインと同じになります。ネームサービスドメインは、ネットワークが大きくなるにつれ

て、ネームサービスをより管理しやすくするために、より小さいドメインに分割することが可能です。他方、メールドメインは、多くの場合、一貫した別名を提供するために分割されないまま残ります。

ネームサービスの要件

ここでは、sendmail がネームサービスに必要とする要件について説明します。

ネームサービスにおけるホストテーブルまたはマップは、次の 3 種類の gethostbyname() による問い合わせをサポートするように設定しなければなりません。

- mailhost – いくつかのネームサービスの構成では、自動的にこの要件を満たします。
- 完全なホスト名 (たとえば、smith.admin.acme.com) – 多くのネームサービスの構成がこの要件を満たします。
- 短いホスト名 (たとえば、smith) – sendmail は、外部メールを転送するためにメールホストに接続する必要があります。メールアドレスが現在のメールドメイン内であるかどうかを判定するために、gethostbyname() が完全なホスト名で呼び出されます。エントリが見つかると、アドレスは内部にあるとみなされます。

  NIS、NIS+、および DNS は、短いホスト名を引数にする gethostbyname() をサポートします。したがって、この要件は自動的に満たされます。

ネームサービス内に有効な sendmail サービスを確立するために、ホストネームサービスに追加された次の 2 つの規則に従う必要があります。

- 完全なホスト名と短いホスト名の引数を持った gethostbyname() は、同一の結果を生成する必要があります。たとえば、両関数がメールドメイン admin.acme.com から呼び出された場合、gethostbyname (smith.admin.acme.com) と gethostbyname (smith) が同じ結果になるようにします。
- 共通のメールドメイン下のすべてのネームサービスドメインに対しては、短いホスト名による gethostbyname() で同じ結果を生じるようにします。たとえば、ebb.admin.acme.com ドメインおよび esg.admin.acme.com ドメインから smith.admin.acme.com メールドメインを呼び出した場合、どちらの場合も gethostbyname(smith) は同じ結果を返す必要があります。ネームサービスドメインはこの要件に各種ネームサービス用の特別な連携を与えているので、メールドメイン名は、通常ネームサービスドメインより短いです。

gethostbyname() 関数については、gethostbyname(3NSL) のマニュアルページを参照してください。

NIS と sendmail との相互作用

次に、sendmail と NIS との相互作用について説明し、ガイドラインを示します。

- メールドメイン名 – NIS をプライマリネームサービスとして設定している場合に、sendmail は、自動的に NIS ドメイン名の最初の構成要素を取り除いた結果をメールドメイン名として使用します。たとえば、ebs.admin.acme.com は、admin.acme.com となります。
- メールホスト名 – NIS のホストマップには、mailhost エントリが必要になります。
- 完全なホスト名 – 通常の NIS の設定では、完全なホスト名は認識されません。NIS に完全なホスト名を認識させようとするよりは、sendmail.cf ファイルを編集し %l を %y で置き換えて、sendmail 側からこの要件をなくしてください。この変更によって、sendmail のドメイン間のメール検出機能をオフにできます。ターゲットとするホストの IP アドレスを取得できれば、SMTP による直接配信が試みられます。NIS のホストマップに現在のメールドメインの外部のホストのエントリが含まれていないことを確認してください。もし、そのエントリがあれば、さらに sendmail.cf ファイルをカスタマイズする必要があります。
- ホストの完全名および短縮名のマッチング – 前述した手順を参考にして、完全なホスト名による gethostbyname() をオフにしてください。
- 1 つのメールドメイン内の複数の NIS ドメイン – 共通のメールドメインの NIS のホストマップ中のホストのエントリは同じである必要があります。たとえば、ebs.admin.acme.com ドメインのホストマップは、esg.admin.acme.com のホストマップと同じものにします。異なる場合には、ある NIS ドメインで有効なアドレスがほかの NIS ドメインでは無効になることがあります。

作業手順については、315 ページの「メール別名ファイルの管理 (作業マップ)」の第 13 章「メールサービス (手順)」を参照してください。

sendmail と NIS および DNS との相互作用

次に、sendmail と NIS および DNS との相互作用について説明し、ガイドラインを示します。

- メールドメイン名 – NIS をプライマリネームサービスとして設定している場合に、sendmail は、自動的に NIS ドメイン名の最初の構成要素を取り除いた結果をメールドメイン名として使用します。たとえば、ebs.admin.acme.com は、admin.acme.com となります。
- メールホスト名 – DNS の転送機能がオンになっていれば、NIS で解決できない照会は DNS に転送されるため、NIS ホストマップに mailhost エントリは必要ありません。
- 完全なホスト名 – NIS が完全なホスト名を認識できなくても、DNS が認識します。NIS と DNS の通常の設定手順を踏んでいる場合には、完全なホスト名の要件は満たされます。
- ホストの完全名および短縮名のマッチング – NIS のホストテーブルにおけるすべてのホストエントリに対して、DNS にも対応するホストエントリが必要です。

- 1 つのメールドメイン内の複数の NIS ドメイン – 共通のメールドメインの NIS のホストマップ中のホストのエントリは同じである必要があります。たとえば、ebs.admin.acme.com ドメインのホストマップは、esg.admin.acme.com のホストマップと同じものにします。異なる場合には、ある NIS ドメインで有効なアドレスがほかの NIS ドメインでは無効になることがあります。

作業手順については、304 ページの「sendmail で DNS を使用する方法」の 315 ページの「メール別名ファイルの管理 (作業マップ)」と 第 13 章「メールサービス (手順)」を参照してください。

NIS+ と sendmail との相互作用

次に、sendmail と NIS+ との相互作用について説明し、ガイドラインを示します。

- メールドメイン名 – プライマリネームサービスとして NIS+ を設定していれば、sendmail は NIS+ の sendmailvars テーブルからメールドメインを確認できます。この NIS+ テーブルには、キー列と値列が 1 つずつあります。メールドメインを設定するには、1 つのエントリをこのテーブルに追加する必要があります。このエントリは、キー列に文字列 maildomain が、値列には自分のメールドメイン名が設定されている必要があります。たとえば、admin.acme.com です。NIS+ では、sendmailvars テーブルにどのような文字列でも設定できますが、メールシステムが正常に機能するように接尾辞の規則が適用されます。nistbladm を使用して、maildomain エントリを sendmailvars テーブルに追加できます。次の例では、メールドメインが NIS+ ドメインの接尾辞になっています。

```
nistbladm -A key="maildomain" value=<mail domain> sendmailvars.org_dir.<NIS+ domain>
```

- メールホスト名 – NIS+ ホストテーブルには、mailhost エントリが必要です。
- 完全なホスト名 – NIS+ は完全なホスト名を「理解」します。通常の NIS+ の設定手順を行えば、この完全なホスト名の要件は満たされます。
- ホストの完全名および短縮名のマッチング – この要件を満たすには、ホストテーブルでエントリをコピーします。または、ユーザーネームサービスのドメイン中の全ホストのエントリを、メールドメインレベルのマスターホストテーブルに入力します。
- 1 つのメールドメイン内の複数の NIS ドメイン – この要件を満たすには、すべてのホストテーブルのエントリをコピーします。または、ユーザーネームサービスのドメイン中の全ホストのエントリを、メールドメインレベルのマスターホストテーブルに入力します。事実上、論理的または物理的に複数のホストテーブルを 1 つのホストテーブルにマージすることになります。したがって、メールドメインを共有する複数のネームサービスドメインで同じホスト名を使用することはできません。

作業手順については、315 ページの「メール別名ファイルの管理 (作業マップ)」の 第 13 章「メールサービス (手順)」を参照してください。

sendmail と NIS+ および DNS との相互作用

次に、sendmail と NIS+ および DNS との相互作用について説明し、ガイドラインを示します。

- メールドメイン名 – プライマリネームサービスとして NIS+ を設定していれば、sendmail は NIS+ の sendmailvars テーブルからメールドメインを確認できます。この NIS+ テーブルには、キー列と値列が 1 つずつあります。メールドメインを設定するには、1 つのエントリをこのテーブルに追加する必要があります。このエントリは、キー列に文字列 maildomain が、値列には自分のメールドメイン名が設定されている必要があります。たとえば、admin.acme.com です。NIS+ では、sendmailvars テーブルにどのような文字列でも設定できますが、メールシステムが正常に機能するように接尾辞の規則が適用されます。nistbladm を使用して、maildomain エントリを sendmailvars テーブルに追加できます。次の例では、メールドメインが NIS+ ドメインの接尾辞になっています。

  ```
  nistbladm -A key="maildomain" value=<mail domain> sendmailvars.org_dir.<NIS+ domain>
  ```

- メールホスト名 – ネットワークがホストデータベースのソースとして NIS+ と DNS の両方を使用しているときは、mailhost エントリを NIS+ あるいは DNS ホストテーブルのいずれかに置くことができます。NIS+ と DNS の両方をホストデータベースのソースとして /etc/nsswitch.conf ファイルに含めるようにしてください。
- 完全なホスト名 – NIS+ と DNS はどちらも、完全なホスト名を「理解」します。通常の NIS+ と DNS の設定手順を踏めば、この項目の要件は満たされます。
- ホストの完全名および短縮名のマッチング – NIS+ ホストテーブルの全ホストエントリに対して、対応するホストエントリが DNS に必要です。
- 1 つのメールドメイン内の複数の NIS ドメイン – この要件を満たすには、すべてのホストテーブルのエントリをコピーします。または、ユーザーネームサービスのドメイン中の全ホストのエントリを、メールドメインレベルのマスターホストテーブルに入力します。

作業手順については、315 ページの「メール別名ファイルの管理 (作業マップ)」の 304 ページの「sendmail で DNS を使用する方法」と 第 13 章「メールサービス (手順)」を参照してください。

sendmail の version 8.13 での変更点

sendmail のこの新しいバージョンには多くの新機能が用意されていますが、FallBackSmartHost オプションがもっとも重要な追加機能です。このオプションにより、main.cf ファイルおよび subsidiary.cf ファイルを使用する必要がなくなります。main.cf ファイルは、MX レコードをサポートする環境で使用されていました。subsidiary.cf ファイルは、完全に動作する DNS がない環境で使用されていました。そのような環境では、スマートホストが MX レコードの代わりに使用されてい

ました。FallBackSmartHost オプションは、統一された構成を提供します。このオプションは、すべての環境でもっとも優先順位の低い MX レコードのように動作します。このオプションは、有効である場合、メールが確実にクライアントに配信されるように、失敗した MX レコードのバックアップ (フェイルオーバー) として担う接続が保たれた (スマート) ホストを提供します。

version 8.13 の詳細については、次の各節を参照してください。

- 386 ページの「sendmail の version 8.13 で追加されたコマンド行オプション」
- 387 ページの「sendmail の version 8.13 で追加または改訂された構成ファイルオプション」
- 388 ページの「sendmail の version 8.13 で追加または改訂された FEATURE() の宣言」

さらに、Solaris 10 1/06 以降のリリースでは、TLS (Transport Layer Security) を使用して SMTP を実行できます。次に説明します。

sendmail の version 8.13 で TLS を使用して SMTP を実行するためのサポート

SMTP サーバーとそのクライアント間の通信は通常、どちらの側でも制御されたり信頼されたりしません。このようにセキュリティーが存在しないことにより、第三者は、サーバーとクライアントの間の通信を監視し、変更することさえ可能です。Solaris 10 1/06 以降のリリースでは、SMTP は sendmail の version 8.13 で Transport Layer Security (TLS) を使用して、この問題を解決できます。これにより SMTP サーバーおよびクライアントに対するサービスが拡張され、次の機能が実現されます。

- インターネットでの機密性の高い認証された通信
- 盗聴や攻撃からの保護

注 – TLS の実装は Secure Sockets Layer (SSL) プロトコルに基づいています。

STARTTLS は、TLS を使用して、セキュリティー保護された SMTP 接続を開始する SMTP キーワードです。このセキュリティー保護された接続は、2 台のサーバーの間、またはサーバーとクライアントの間で行われます。セキュリティー保護された接続は、次のように定義されます。

- 発信元電子メールアドレスと宛先電子メールアドレスが暗号化される。
- 電子メールメッセージの内容が暗号化される。

クライアントが STARTTLS コマンドを発行すると、サーバーは次のいずれかを使用して応答します。

- `220 Ready to start TLS`
- `501 Syntax error (no parameters allowed)`
- `454 TLS not available due to temporary reason`

220 応答では、クライアントが TLS ネゴシエーションを開始する必要があります。501 応答は、クライアントが STARTTLS コマンドを正しく発行しなかったことを示します。STARTTLS はパラメータなしで発行されます。454 応答では、クライアントがルールセットの値を適用して、接続を受け入れるか維持するかどうかを決定する必要があります。

インターネットの SMTP インフラストラクチャーを維持するため、公的に使用されるサーバーは TLS ネゴシエーションを要求してはならないことに注意してください。ただし、私的に使用されるサーバーは、クライアントが TLS ネゴシエーションを実行するよう要求しても構いません。このような場合、サーバーは次のような応答を返します。

```
530 Must issue a STARTTLS command first
```

530 応答は、 STARTTLS コマンドを発行して接続を確立するようクライアントに指示します。

認証とプライバシーのレベルが不十分である場合、サーバーまたはクライアントは接続を拒否できます。また、多くの SMTP 接続はセキュリティー保護されていないため、サーバーとクライアントはセキュリティー保護されていない接続を維持する場合があります。接続を維持するか拒否するかどうかは、サーバーとクライアントの構成により決まります。

TLS を使用して SMTP を実行するためのサポートは、デフォルトでは有効になっていません。TLS が有効になるのは、SMTP クライアントが STARTTLS コマンドを発行した場合です。SMTP クライアントがこのコマンドを発行する前に、sendmail が TLS を使用できるようにする証明書を設定する必要があります。309 ページの「TLS を使用するよう SMTP を構成する」を参照してください。この手順には、新しい構成ファイルのオプションの定義と、sendmail.cf ファイルの再構築が含まれることに注意してください。

TLS を使用して SMTP を実行するための構成ファイルのオプション

次の表で、TLS を使用して SMTP を実行するために使用される構成ファイルのオプションを説明します。これらのオプションを宣言する場合は、次の構文のどれかを使用します。

- O *OptionName=argument* # 構成ファイル用
- -O *OptionName=argument* # コマンド行用

- `define(‘`*m4Name*`',`*argument*`)` # m4 構成用

表 14–13 TLS を使用して SMTP を実行するための構成ファイルのオプション

オプション	説明
CACertFile	m4 名 : confCACERT 引数 : *filename* デフォルト値 : 未定義 1 つの CA 証明書を含むファイルを指定します。
CACertPath	m4 名 : confCACERT_PATH 引数 : *path* デフォルト値 : 未定義 複数の CA の証明書が含まれるディレクトリへのパスを指定します。
ClientCertFile	m4 名 : confCLIENT_CERT 引数 : *filename* デフォルト値 : 未定義 クライアントの証明書が含まれるファイルを指定します。sendmail がクライアントとして動作する場合にこの証明書が使用されることに注意してください。
ClientKeyFile	m4 名 : confCLIENT_KEY 引数 : *filename* デフォルト値 : 未定義 クライアントの証明書に属する秘密鍵が含まれるファイルを指定します。
CRLFile	m4 名 : confCRL 引数 : *filename* デフォルト値 : 未定義 X.509v3 認証に使用される、証明書の失効ステータスが含まれるファイルを指定します。
DHParameters	m4 名 : confDH_PARAMETERS 引数 : *filename* デフォルト値 : 未定義 Diffie-Hellman (DH) パラメータが含まれるファイルを指定します。

表 14–13 TLS を使用して SMTP を実行するための構成ファイルのオプション (続き)

オプション	説明
RandFile	m4 名 : confRAND_FILE 引数 : file:*filename* または egd:*UNIX socket* デフォルト値 : 未定義 file: 接頭辞を使用してランダムデータが含まれるファイルを指定するか、egd: 接頭辞を使用して UNIX ソケットを指定します。Solaris OS は乱数生成デバイスをサポートしているため、このオプションを指定する必要はありません。random(7D) のマニュアルページを参照してください。
ServerCertFile	m4 名 : confSERVER_CERT 引数 : *filename* デフォルト値 : 未定義 サーバーの証明書が含まれるファイルを指定します。sendmail がサーバーとして動作する場合にこの証明書が使用されます。
Timeout.starttls	m4 名 : confTO_STARTTLS 引数 : *amount of time* デフォルト値 : 1h STARTTLS コマンドに対する応答を SMTP クライアントが待機する時間を設定します。
TLSSrvOptions	m4 名 : confTLS_SRV_OPTIONS 引数 : V デフォルト値 : 未定義 サーバーがクライアントから証明書を要求するかどうかを決定します。このオプションが V に設定されている場合、クライアント検証は行われません。

sendmail で SMTP による TLS の使用をサポートできるようにするには、次のオプションを定義してください。

- CACertPath
- CACertFile
- ServerCertFile
- ClientKeyFile

そのほかのオプションは必須ではありません。

TLS を使用して SMTP を実行するためのマクロ

次の表で、STARTTLS コマンドにより使用されるマクロを説明します。

表 14–14　TLS を使用して SMTP を実行するためのマクロ

マクロ	説明
${cert_issuer}	証明書の発行元である認証局 (CA) の識別名 (DN) を保持します。
${cert_subject}	**cert subject** と呼ばれる証明書の DN を保持します。
${cn_issuer}	**cert issuer** である CA の共通名 (CN) を保持します。
${cn_subject}	**cert subject** と呼ばれる証明書の CN を保持します。
${tls_version}	接続に使用される TLS のバージョンを保持します。
${cipher}	接続に使用される (**cipher suite** と呼ばれる) 暗号アルゴリズムのセットを保持します。
${cipher_bits}	接続に使用される対称暗号化アルゴリズムのキーの長さをビット単位で保持します。
${verify}	提示された証明書の検証結果を保持します。取りうる値は次のとおり ■ OK – 検証成功。 ■ NO – 証明書は提示されません。 ■ NOT – 証明書は要求されません。 ■ FAIL – 証明書は提示されたが検証不可。 ■ NONE – STARTTLS は実行されません。 ■ TEMP – 一時エラーが発生。 ■ PROTOCOL – SMTP エラーが発生。 ■ SOFTWARE – STARTTLS ハンドシェークが失敗。
${server_name}	現在の出力 SMTP 接続のサーバー名を保持します。
${server_addr}	現在の出力 SMTP 接続のサーバーのアドレスを保持します。

TLS を使用して SMTP を実行するためのルールセット

次の表で、TLS を使用する SMTP 接続を、受け入れるか、継続するか、拒否するかを決定するルールセットを説明します。

表 14–15　TLS を使用して SMTP を実行するためのルールセット

ルールセット	説明
tls_server	クライアントとして動作する場合、sendmail はこのルールセットを使用して、サーバーが現在 TLS によってサポートされているかどうかを判別します。
tls_client	サーバーとして動作する場合、sendmail はこのルールセットを使用して、クライアントが現在 TLS によってサポートされているかどうかを判別します。
tls_rcpt	このルールセットは、受取人の MTA の検証を必要とします。この受取人の制限により、DNS スプーフィングなどの攻撃が不可能になります。

表 14–15 TLS を使用して SMTP を実行するためのルールセット (続き)

ルールセット	説明
TLS_connection	このルールセットは、アクセスマップの RHS により指定された要件を、現在の TLS 接続の実際のパラメータに照合して確認します。
try_tls	sendmail はこのルールセットを使用して、別の MTA への接続時に STARTTLS を使用できるかを判別します。MTA が適切に STARTTLS を実装できない場合、STARTTLS は使用されません。

詳細は、http://www.sendmail.org/m4/starttls.html を参照してください。

TLS を使用した SMTP の実行に関連するセキュリティーの検討事項

インターネットで動作するメールプログラムを定義する標準メールプロトコルとしては、SMTP はエンドツーエンドのメカニズムではありません。このプロトコルの制限により、SMTP を介した TLS のセキュリティーにはメールユーザーエージェントは含まれていません。メールユーザーエージェントは、ユーザーと (sendmail などの) メール転送エージェントの間のインタフェースとして動作します。

また、メールは複数のサーバーを経由してルーティングされる場合があります。SMTP のセキュリティーを完全にするには、SMTP 接続のチェーン全体に TLS のサポートが必要です。

最終的には、サーバーの各ペア、またはクライアントとサーバーのペアの間でネゴシエーションされる認証と機密性のレベルを考慮すべきです。詳細は、『Solaris のシステム管理 (セキュリティサービス)』の「認証サービス」を参照してください。

sendmail の version 8.13 で追加されたコマンド行オプション

次の表に、sendmail の version 8.13 で追加されたコマンド行オプションを示します。コマンド行のほかのオプションについては、sendmail(1M) のマニュアルページを参照してください。

表 14–16 sendmail の version 8.13 で使用可能になったコマンド行オプション

オプション	説明
-D *logfile*	この情報を標準出力に含めるのではなく、指定された *logfile* にデバッグ出力を送信します。
-q[!]Q*substr*	隔離 *reason* の部分文字列である *substr* を持つ隔離されたジョブの処理を指定します。-Q *reason* オプションの説明を参照。!が追加された場合、このオプションは、この *substr* を持たない隔離されたジョブを処理します。

表 14–16 sendmail の version 8.13 で使用可能になったコマンド行オプション (続き)

オプション	説明
-Q*reason*	この *reason* を持つ通常のキュー項目を隔離します。*reason* が指定されていない場合、隔離されるキュー項目が隔離されません。このオプションは、-q[!]Q*substr* オプションと連動します。*substr* は、*reason* の一部 (部分文字列) です。

sendmail の version 8.13 で追加または改訂された構成ファイルオプション

次の表に、追加または改訂された構成ファイルオプションを示します。これらのオプションを宣言する場合は、次の構文のどれかを使用します。

```
O OptionName=argument          # for the configuration file
-O OptionName=argument         # for the command line
define(‘m4Name’,argument)     # for m4 configuration
```

表 14–17 sendmail の version 8.13 で使用可能な構成ファイルオプション

オプション	説明
ConnectionRateWindowSize	m4 名 : confCONNECTION_RATE_WINDOW_SIZE 引数 : *number* デフォルト値 : 60 受信接続を維持する秒数を設定します。
FallBackSmartHost	m4 名 : confFALLBACK_SMARTHOST 引数 : *hostname* このオプションは、メールが確実にクライアントに配信されるように、失敗した MX レコードのバックアップ (フェイルオーバー) として担う接続が保たれたホストを提供します。
InputMailFilters	m4 名 : confINPUT_MAIL_FILTERS 引数 : *filename* sendmail デーモンの入力メールフィルタを示します。
PidFile	m4 名 : confPID_FILE 引数 : *filename* デフォルト値 : /var/run/sendmail.pid 今までのリリースのように、ファイルを開く前に、そのファイル名がマクロで展開されます。さらに、version 8.13 では、sendmail の終了時にファイルへのリンクが削除されます (unlinked)。

表 14–17 sendmail の version 8.13 で使用可能な構成ファイルオプション (続き)

オプション	説明
QueueSortOrder	m4 名 : confQUEUE_SORT_ORDER 追加された引数 : none version 8.13 では、ソート順序を指定しない場合に none を使用します。
RejectLogInterval	m4 名 : confREJECT_LOG_INTERVAL 引数 : *period-of-time* デフォルト値 : 3h (3 時間) 指定した *period-of-time* に対してデーモン接続が拒否された場合、その情報が記録されます。
SuperSafe	m4 名 : confSAFE_QUEUE 短縮名 : s 追加された引数 : postmilter デフォルト値 : true postmilter が設定されている場合、sendmail は、すべての milters がメッセージの受付の信号を送るまで、キューファイルとの同期を延期します。この引数を有効にするには、sendmail が SMTP サーバーとして実行される必要があります。それ以外の場合、postmilter は true 引数を使用しているように動作します。

sendmail の version 8.13 で追加または改訂された FEATURE() の宣言

次の表に、追加または改訂された FEATURE() の宣言を示します。この m4 マクロは次の構文を使用します。

FEATURE(‘*name*’, ‘*argument*’)

表 14–18 sendmail の version 8.13 で使用可能な FEATURE() の宣言

FEATURE() の名前	説明
conncontrol	access_db ルールセットと連動して、受信 SMTP 接続の数を確認します。詳細は、/etc/mail/cf/README を参照してください。
greet_pause	オープンプロキシと SMTP のスラミング保護を可能にする、greet_pause ルールセットを追加します。詳細は、/etc/mail/cf/README を参照してください。

表 14–18 sendmail の version 8.13 で使用可能な FEATURE() の宣言 (続き)

FEATURE() の名前	説明
local_lmtp	デフォルトの引数は引き続き mail.local であり、今回の Solaris のリリースでの LMTP を使用できるメールプログラムです。ただし、version 8.13 で、LMTP を使用できる別のメールプログラムを使用する場合は、パス名を 2 番目のパラメータとして指定し、2 番目のパラメータに渡される引数を 3 番目のパラメータとして指定します。次に例を示します。 FEATURE('local_lmtp', '/usr/local/bin/lmtp', 'lmtp')
mtamark	"TXT RRs による逆引き DNS でのメール転送エージェントのマーキング" (MTAMark) を試験的にサポートします。詳細は、/etc/mail/cf/README を参照してください。
ratecontrol	access_db ルールセットと連動して、ホストに対する接続速度を制御します。詳細は、/etc/mail/cf/README を参照してください。
use_client_ptr	この FEATURE() が有効になっている場合、ルールセット check_relay は $&{client_ptr} というこの引数で最初の引数を上書きします。

sendmail の version 8.12 からの変更点

この節では、次のトピックについて説明します。

sendmail の version 8.12 からの TCP ラッパーのサポート

TCP ラッパーは、特定のネットワークサービスを要求するホストのアドレスをアクセス制御リスト (ACL) と突き合わせて検査することによるアクセス制御の実装方法を提供します。要求は、状況に応じて、許可されたり拒否されたりします。このアクセス制御メカニズムを提供する以外に、TCP ラッパーは、ネットワークサービスに対するホストの要求を記録します。これは、有用な監視機能です。アクセス制御のもとに置かれるネットワークサービスの例として、`rlogind`、`telnetd`、`ftpd` などがあります。

version 8.12 より、`sendmail` で TCP ラッパーが使用できるようになりました。この検査によってほかのセキュリティー対策が省略されることはありません。`sendmail` で TCP ラッパーを有効にすることにより、検査が追加され、ネットワーク要求元の妥当性が検証されてから要求が許可されます。`hosts_access(4)` のマニュアルページを参照してください。

注 – `inetd(1M)` および `sshd(1M)` での TCP ラッパーは、Solaris 9 リリースからサポートされています。

ACL については、『Solaris のシステム管理 (セキュリティサービス)』の「アクセス制御リストによる UFS ファイルの保護」を参照してください。

sendmail の version 8.12 からの submit.cf 構成ファイル

version 8.12 より、`sendmail` に新しい構成ファイル `/etc/mail/submit.cf` が含まれるようになりました。この `submit.cf` ファイルを使用して、`sendmail` をデーモンモードではなく、メール配信プログラムモードで実行できます。デーモンモードとは異なり、メール配信プログラムモードでは `root` 権限は必要ありません。そのため、この新しいパラダイムを使用すると、セキュリティーが向上します。

`submit.cf` の機能については、次を参照してください。

- `sendmail` は、MSP (メール配信プログラム) モードでは `submit.cf` を使って実行します。submit.cf は、電子メールを送信したり、ユーザー以外の `mailx` のようなプログラムによって呼び出したりすることができます。`-Ac` オプションおよび `-Am` オプションについては、`sendmail(1M)` のマニュアルページを参照してください。
- `submit.cf` は、次の操作モードで使用します。
  - `-bm` デフォルトの操作モード
  - `-bs` 標準入力を使用して SMTP を実行する

  - -bt アドレスの解決に使用されるテストモード
- submit.cf を使用している場合には、sendmail は SMTP デーモンとして動作しません。
- submit.cf を使用している場合には、sendmail はクライアント専用のメールキューである /var/spool/clientmqueue を使用します。このキューにより、sendmail デーモンに配信されなかったメッセージが保持されます。クライアント専用キューにあるメッセージは、クライアントの「デーモン」によって配信されます。実際には、このデーモンが、クライアントキューを実行します。
- デフォルトでは、sendmail は submit.cf を使用して、定期的に MSP キュー (クライアント専用キュー) である /var/spool/clientmqueue を実行します。

```
/usr/lib/sendmail -Ac -q15m
```

次の事項に注意してください。

- Solaris 9 より、submit.cf は自動的にインストールされます。
- Solaris 9 以降をインストールする前に、submit.cf について計画および準備をする必要はありません。
- 構成ファイルを指定しないかぎり、sendmail は、必要に応じて、submit.cf を自動的に使用します。基本的に、sendmail は各タスクについて、submit.cf と sendmail.cf のどちらを使用するのが適切かを判断します。
- submit.cf を変更することはできません。

sendmail.cf と submit.cf の機能の相違点

構成ファイル sendmail.cf は、デーモンモードで使用します。このファイルを使用すると、sendmail は、メール転送エージェント (MTA) として動作します。sendmail は、root によって起動されます。

```
/usr/lib/sendmail -L sm-mta -bd -q1h
```

sendmail.cf 特有のほかの機能については、次を参照してください。

- デフォルトでは、sendmail.cf は、ポート 25 および 587 で SMTP 接続を受け入れます。
- デフォルトでは、sendmail.cf がメールキュー /var/spool/mqueue を実行します。

sendmail の version 8.12 からの機能の変更

submit.cf が追加されたため、次の機能が変更されました。

- sendmail の version 8.12 より、root だけがメールキューを実行できます。この変更の詳細は、mailq(1) のマニュアルページを参照してください。新しい作業手順については、327 ページの「キューディレクトリの管理 (作業マップ)」を参照してください。
- メール配信プログラムモードは、root 権限がなくても実行されるので、sendmail が特定のファイル (.forward ファイルなど) にアクセスできないことがあります。したがって、sendmail に -bv オプションを追加すると、ユーザーが誤解するような出力を発生させる可能性があります。回避策はありません。
- 8.12 より前のバージョンの sendmail では、sendmail をデーモンモードで実行しない場合は、受信メールの配信を防止することしかできませんでした。version 8.12 より、デフォルトの構成で、sendmail デーモンを実行しない場合、送信メールの配信もまた防止することができます。クライアントキューランナー (メール配信プログラム) を設定して、ローカル SMTP ポートのデーモンにメールを送信できるようにする必要があります。クライアントキューランナーが SMTP のセッションをローカルホストで開こうとした場合で、デーモンが SMTP ポートで待機していないときには、メールはキューにとどまります。デフォルトの構成では、デーモンが実行されます。そのため、デフォルト構成を使用する場合には、この問題は発生しません。ただし、デーモンを無効にした場合の解決方法については、314 ページの「sendmail.cf の代替構成を使ってメール配信を管理する方法」を参照してください。

sendmail の version 8.12 から追加されたまたは推奨されないコマンド行オプション

次の表では、sendmail の追加されたコマンド行オプションまたは推奨されないコマンド行オプションについて説明します。コマンド行のほかのオプションについては、sendmail(1M) のマニュアルページを参照してください。

表 14–19 sendmail の version 8.12 から追加されたまたは推奨されないコマンド行オプション

オプション	説明
-Ac	オペレーションモードが初期メール配信を示していない場合でも、構成ファイル submit.cf を使用します。submit.cf についての詳細は、390 ページの「sendmail の version 8.12 からの submit.cf 構成ファイル」を参照してください。
-Am	オペレーションモードが初期メール配信を示している場合でも、構成ファイル sendmail.cf を使用します。詳細は、390 ページの「sendmail の version 8.12 からの submit.cf 構成ファイル」を参照してください。
-bP	各キューのエントリ数を出力します。

表 14–19 sendmail の version 8.12 から追加されたまたは推奨されないコマンド行オプション (続き)

オプション	説明
-G	コマンド行から送信されたメッセージが、初期送信のためでなく、中継のためであることを示します。アドレスが絶対パスではない場合は、メッセージは拒否されます。正規化は実行されません。ftp://ftp.sendmail.org で sendmail とともに配布しているリリースノートで説明しているように、将来のリリースでは、不適切な形式のメッセージが拒否される可能性があります。
-L *tag*	指定された syslog メッセージに使用する識別子を タグ *(tag)* に設定します。
- q[!]I *substring*	受信者にこの部分文字列 *(substring)* を含むジョブだけを処理します。オプションに ! を追加すると、受信者にこの部分文字列 *(substring)* を含まないジョブだけを処理します。
- q[!]R *substring*	キュー ID にこの部分文字列 *(substring)* を含むジョブだけを処理します。オプションに ! を追加すると、キュー ID にこの部分文字列 *(substring)* を含まないジョブだけを処理します。
- q[!]S *substring*	送信者にこの部分文字列 *(substring)* を含むジョブだけを処理します。オプションに ! を追加すると、送信者にこの部分文字列 *(substring)* を含まないジョブだけを処理します。
-qf	キューにあるメッセージをシステムコール fork を使用しないで一度処理し、フォアグラウンドでプロセスを実行します。fork(2) のマニュアルページを参照してください。
-qG*name*	*name* で指定するキューグループにあるメッセージだけを処理します。
-qp*time*	各キュー用にフォークされた子プロセスを使用して、キューに保存されているメッセージを指定した間隔で処理します。次にキューが実行されるまでの間、その子プロセスはスリープしています。この新しいオプションは -q*time* に似ています。qtime は、定期的に子をフォークしてキューを処理します。
-U	ftp://ftp.sendmail.org で sendmail とともに配付しているリリースノートで説明しているように、このオプションは version 8.12 以降では使用できません。メールユーザーエージェントでは、引数 -G を使用することをお勧めします。

sendmail の version 8.12 から PidFile オプションおよび ProcessTitlePrefix オプションに追加された引数

次の表では、PidFile オプションおよび ProcessTitlePrefix オプションにおけるマクロ処理の追加引数について説明します。これらのオプションについては、sendmail(1M) のマニュアルページを参照してください。

表 14–20 PidFile オプションおよび ProcessTitlePrefix オプションの引数

マクロ	説明
${daemon_addr}	0.0.0.0 などのデーモンアドレスを提供します。
${daemon_family}	inet や inet6 などのデーモンファミリーを提供します。

表 14–20 PidFile オプションおよび ProcessTitlePrefix オプションの引数 (続き)

マクロ	説明
${daemon_info}	SMTP+queueing@00: 30:00 などのデーモン情報を提供します。
${daemon_name}	MSA などのデーモン名を提供します。
${daemon_port}	25 などのデーモンポートを提供します。
${queue_interval}	キューを実行する間隔を提供します (00:30:00 など)。

sendmail の version 8.12 から追加定義されたマクロ

次の表では、sendmail プログラムで使用するための追加マクロについて説明しています。マクロの値は、内部で割り当てられています。詳細は、sendmail(1M) のマニュアルページを参照してください。

表 14–21 sendmail に追加定義されたマクロ

マクロ	説明
${addr_type}	現在のアドレスを、エンベロープの送信側または受信者アドレスと認定します。
${client_resolve}	${client_name} の解釈処理コールの結果、つまり OK、FAIL、FORGED、または TEMP を保持します。
${deliveryMode}	DeliveryMode オプションの値ではなく、sendmail が使用している現在のデリバリモードを指定します。
${dsn_notify}、${dsn_envid}、${dsn_ret}	対応する DSN パラメータ値を保持します。
${if_addr}	インタフェースがループバックネット上にない場合に、受信接続用インタフェースのアドレスを提供します。このマクロは、特に仮想ホスティングに便利です。
${if_addr_out}、${if_name_out}、${if_family_out}	${if_addr} の再利用を避けます。次の値を、それぞれ保持します。 送信接続用インタフェースのアドレス 送信接続用インタフェースのホスト名 送信接続用インタフェースのファミリ

表 14–21 sendmail に追加定義されたマクロ (続き)

マクロ	説明
${if_name}	受信接続用のインタフェースのホスト名を提供します。これは、特に仮想ホスティングに便利です。
${load_avg}	実行キューにあるジョブの現在の平均数を確認して報告します。
${msg_size}	ESMTP ダイアログにあるメッセージサイズの値 (SIZE=parameter) を保持してから、メッセージを収集します。その後、sendmail によって計算されたメッセージサイズを保持したマクロを check_compat で使用します。check_compat については、表 14–25 を参照してください。
${nrcpts}	妥当性検査を行なった受信者の数を保持します。
${ntries}	配信を試みた回数を保持します。
${rcpt_mailer}、${rcpt_host}、${rcpt_addr}、${mail_mailer}、${mail_host}、および ${mail_addr}	引数 RCPT および MAIL を構文解析した結果を保持します。つまり、メール配信エージェント ($#*mailer*)、ホスト ($@*host*)、およびユーザー ($:*addr*) から解釈処理された RHS (Right-Hand Side) トリプレットを保持します。

sendmail の version 8.12 から追加されたマクロ

この節では、構成ファイル sendmail を構築するのに使用する追加マクロについて説明した表を示します。

表 14–22 構成ファイル sendmail を構築するのに使用する追加マクロ

マクロ	説明
LOCAL_MAILER_EOL	ローカルメールプログラムの行末を示すデフォルト文字列を置きかえます。
LOCAL_MAILER_FLAGS	デフォルトで Return-Path: ヘッダーを追加します。
MAIL_SETTINGS_DIR	メール設定ディレクトリのパスを格納します (末尾のスラッシュを含む)。
MODIFY_MAILER_FLAGS	*_MAILER_FLAGS を拡張します。このマクロは、フラグを設定、追加、または削除します。
RELAY_MAILER_FLAGS	中継メールプログラムの追加フラグを定義します。

sendmail の version 8.12 から追加された MAX マクロ

次のマクロを使用して、受け入れ可能なコマンドを最大数設定し、sendmail による配信の遅れを防止することができます。これらの MAX マクロは、コンパイル時に設定できます。次の表にある最大値は、現在のデフォルト値でもあります。

表 14–23 追加された MAX マクロ

マクロ	最大値	各マクロが検査するコマンド
MAXBADCOMMANDS	25	未知のコマンド
MAXNOOPCOMMANDS	20	NOOP、VERB、ONEX、XUSR
MAXHELOCOMMANDS	3	HELO、EHLO
MAXVRFYCOMMANDS	6	VRFY、EXPN
MAXETRNCOMMANDS	8	ETRN

注 – マクロによる確認を無効にするには、マクロの値を 0 に設定します。

sendmail の version 8.12 から追加または改訂された m4 構成マクロ

この節では、sendmail において追加または改訂された m4 構成マクロの表を示します。これらのマクロを宣言するには、次の構文を使用します。

symbolic-name(‘*value*’)

新しい sendmail.cf ファイルを構築する必要がある場合は、305 ページの「sendmail 構成を変更する」の 第 13 章「メールサービス (手順)」を参照してください。

表 14–24 sendmail において追加または改訂された m4 構成マクロ

m4 マクロ	説明
FEATURE()	詳細は、397 ページの「sendmail の version 8.12 からの FEATURE() の宣言についての変更点」を参照してください。
LOCAL_DOMAIN()	このマクロは、クラス w($=w) にエントリを追加します。
MASQUERADE_EXCEPTION ()	マスカレードできないホストやサブドメインを定義する新しいマクロ。
SMART_HOST()	このマクロは user@[*host*] のように、括弧で囲まれたアドレスに使用できます。

表 14–24 sendmail において追加または改訂された m4 構成マクロ (続き)

m4 マクロ	説明
VIRTUSER_DOMAIN() または VIRTUSER_DOMAIN_FILE()	これらのマクロを使用する場合は、$=R に $={VirtHost} を含めます。$=R は、中継が許可された一連のホスト名です。

sendmail の version 8.12 からの FEATURE() の宣言についての変更点

FEATURE() の宣言についての変更点については、次の表を参照してください。

FEATURE の新しい名前および改訂された名前を使用するには、次の構文を使用します。

FEATURE(‘*name*’, ‘*argument*’)

新しい sendmail.cf ファイルを構築する必要がある場合は、305 ページの「sendmail 構成を変更する」の 第 13 章「メールサービス (手順)」を参照してください。

表 14–25 追加または改訂された FEATURE() の宣言

FEATURE() の名前	説明
compat_check	引数 : 次の段落の例を参照してください。 この新しい FEATURE() によって、送信者アドレスと受信者アドレスからなるアクセスマップ内でキーを検索できます。この FEATURE() は、文字列 <@> で区切ります。たとえば、*sender*@ *sdomain*<@>*recipient* @*rdomain* のようにします。
delay_checks	引数 : friend にすると、スパムメールの friend テストを実行できます。また、hater にすると、スパムメールの hater テストを実行できます。 すべての検査作業を遅らせる新しい FEATURE()。FEATURE(‘delay_checks’) を使用すると、クライアントが接続する場合、またはクライアントが MAIL コマンドを発行する場合に、ルールセット check_mail および check_relay は呼び出されません。代わりに、これらのルールセットはルールセット check_rcpt によって呼び出されます。詳細については、/etc/mail/cf/README ファイルを参照してください。
dnsbl	引数 : この FEATURE() は、最大次の 2 つの引数を受け入れます。 ■ DNS サーバー名 ■ リジェクトメッセージ DNS 参照の戻り値を検査する回数を複数にできる新しい FEATURE()。この FEATURE() を使用して、参照が一時的に失敗した場合の動作を指定できます。

表 14–25 追加または改訂された FEATURE() の宣言 (続き)

FEATURE() の名前	説明
enhdnsbl	引数 : ドメイン名。 dnsbl の強化バージョンの新しい FEATURE()。この FEATURE を使用して、DNS 参照の戻り値を検査できます。詳細は、/etc/mail/cf/README を参照してください。
generics_entire_domain	引数 : なし。 genericstable を $=G のサブドメインに適用するのにも使用できる新しい FEATURE()。
ldap_routing	引数 : 詳細は、http://www.sendmail.org の「リリースノート」を参照してください。 LDAP アドレスルーティングを実装する新しい FEATURE()。
local_lmtp	引数 : LMTP (Local Mail Transfer Protocol) を使用できるメールプログラムのパス名。デフォルトは mail.local であり、今回の Solaris リリースでは LMTP を使用できます。 ローカルメールプログラムの DSN (delivery status notification) 診断コードのタイプを SMTP の正しい値に設定する FEATURE()。
local_no_masquerade	引数 : なし。 ローカルメールプログラムをマスカレードしないようにするために使用する新しい FEATURE()。
lookupdotdomain	引数 : なし。 アクセスマップの *.domain* を参照するのに使用する新しい FEATURE()。
nocanonify	引数 : canonify_hosts またはなし。 FEATURE() には次の機能が含まれます。 CANONIFY_DOMAIN または CANONIFY_DOMAIN_FILE で指定した、ドメインのリストを演算子 $[および $] に渡して正規化することができます。 canonify_hosts がそのパラメータとして指定されている場合には、<user@host> など、ホスト名だけを含むアドレスを正規化できます。 複数のコンポーネントを持つアドレスの末尾にドットを追加できます。
no_default_msa	引数 : なし。 sendmail のデフォルト設定を m4 構成ファイルでオフにする新しい FEATURE()。このファイルは、複数の異なるポート上で待機するために生成されたもので、RFC 2476 に実装されています。

表 14–25 追加または改訂された FEATURE() の宣言 (続き)

FEATURE() の名前	説明
nouucp	引数 : reject にすると、! トークンを使用できません。 nospecial にすると、! トークンを使用できます。 ! トークンをアドレスのローカルの部分に使用するかどうかを決定する FEATURE()。
nullclient	引数 : なし。 通常の構成ですべてのルールセットを提供する FEATURE()。スパムメール対策チェックを実行します。
preserve_local_plus_detail	引数 : なし。 sendmail がアドレスをローカル配信エージェントに渡す際に、アドレスの +detail の部分を保存できる新しい FEATURE()。
preserve_luser_host	引数 : なし。 LUSER_RELAY を使用している場合に、受信者のホスト名を保存できる新しい FEATURE()。
queuegroup	引数 : なし。 電子メールのアドレス全体または受信者のドメインに基づいたキューグループを選択できる新しい FEATURE()。
relay_mail_from	引数 : ドメインは、任意の引数です。 メールの送信側がアクセスマップに RELAY として指定されており、それをヘッダー行 From: でタグ付けされている場合に、リレーを許可する新しい FEATURE()。省略可能な引数 *domain* を指定すると、メール送信側のドメイン部も検査されます。
virtuser_entire_domain	引数 : なし。 $={VirtHost} を適用するのに使用する FEATURE()。$={VirtHost} は、VIRTUSER_DOMAIN または VIRTUSER_DOMAIN_FILE を使って生成できる virtusertable エントリを一致させるための新しいクラス。 また、FEATURE('virtuser_entire_domain') を使用して、クラス $={VirtHost} をサブドメイン全体に適用することもできます。

次の FEATURE () 宣言はサポートされなくなりました。

表 14–26 宣言がサポートされていない FEATURE()

FEATURE() の名前	代わりの FEATURE
rbl	削除されたこの FEATURE() の代わりに、FEATURE('dnsbl') および FEATURE('enhdnsbl') を使用してください。

表 14–26　宣言がサポートされていない FEATURE()　(続き)

FEATURE() の名前	代わりの FEATURE
remote_mode	/etc/mail/cf/subsidiary.mc では、FEATURE('remote_mode') の代わりに MASQUERADE_AS('$S') を使用できます。$S は、sendmail.cf における SMART_HOST の値。
sun_reverse_alias_files	FEATURE('genericstable')
sun_reverse_alias_nis	FEATURE('genericstable')
sun_reverse_alias_nisplus	FEATURE('genericstable')

sendmail の version 8.12 からの MAILER() の宣言についての変更点

MAILER() を宣言すると、配信エージェントのサポートを指定できます。配信エージェントを宣言するには、次の構文を使用します。

MAILER('*symbolic-name*')

次の変更に注意してください。

- この新しいバージョンの sendmail では、MAILER('smtp') を宣言すると、メールプログラム dsmtp が追加されます。dsmtp により、メールプログラムのフラグ F=% を使用して、オンデマンドに配信することができます。dsmtp メールプログラムを定義する際には、新しい DSMTP_MAILER_ARGS を使用します。DSMTP_MAILER_ARGS のデフォルトは IPC $h です。
- MAILER によって使用されるルールセットの番号は削除されました。MAILER('uucp') を除いて、MAILER の表示順を自由に設定できます。uucp-dom および uucp-uudom を使用する場合には、MAILER('smtp') のあとに MAILER('uucp') を配置する必要があります。

メールプログラムの詳細は、346 ページの「メールプログラムと sendmail」を参照してください。新しい sendmail.cf ファイルを構築する必要がある場合は、305 ページの「sendmail 構成を変更する」の 第 13 章「メールサービス (手順)」を参照してください。

sendmail の version 8.12 から追加された配信エージェントのフラグ

次の表では、配信エージェントの追加されたフラグについて説明しています。デフォルトでは、これらのフラグは設定されていません。これらの 1 文字のフラグはブール型です。このフラグを設定したりその設定を解除したりするには、次の例のように、フラグを構文ファイルの F= 文に含めるか除外します。

```
Mlocal,    P=/usr/lib/mail.local, F=lsDFMAw5:/|@qSXfmnz9, S=10/30, R=20/40,
Mprog,     P=/bin/sh, F=lsDFMoqeu9, S=10/30, R=20/40, D=$z:/,
Msmtp,     P=[IPC], F=mDFMuX, S=11/31, R=21, E=\r\n, L=990,
Mesmtp,    P=[IPC], F=mDFMuXa, S=11/31, R=21, E=\r\n, L=990,
Msmtp8,    P=[IPC], F=mDFMuX8, S=11/31, R=21, E=\r\n, L=990,
Mrelay,    P=[IPC], F=mDFMuXa8, S=11/31, R=61, E=\r\n, L=2040,
```

表 14–27 メールプログラムの追加されたフラグ

フラグ	説明
%	このフラグを使用するメールプログラムは、ETRN 要求や次のいずれかのキューオプションを使ってキューにあるメッセージを選択しないかぎり、最初の受信者宛にメールを配信したり、キューを実行したりしません。 -qI、-qR、または -qS。
1	このフラグは、\0 などのヌル文字を送信するメールプログラムの機能を無効にします。
2	このフラグは、ESMTP の使用を無効にし、代わりに SMTP を使用するように要求します。
6	このフラグを指定すると、メールプログラムでヘッダーを 7 ビットにすることができます。

sendmail の version 8.12 から追加された配信エージェントの設定

次の表では、配信エージェントを定義するコマンド M とともに使用できる新しい設定について説明しています。次の構文は、設定を新たに付加する方法、および構成ファイルの既存の設定に新しい引数を付加する方法を示しています。

Magent-name, *equate*, *equate*, ...

次の例には、新しい設定 W= が含まれています。この設定は、すべてのデータが送信されたあとでメールプログラムが戻るまでの最長待ち時間を指定します。

```
Msmtp, P=[IPC], F=mDFMuX, S=11/31, R=21, E=\r\n, L=990, W=2m
```

m4 の構成値の定義を変更するには、次の例のような構文を使用します。

```
define(‘SMTP_MAILER_MAXMSGS’, ‘1000’)
```

この例では、smtp メールプログラムで 1 回の接続で配信されるメッセージ数を 1000 に制限しています。

新しい sendmail.cf ファイルを構築する必要がある場合は、305 ページの「sendmail 構成を変更する」の 第 13 章「メールサービス (手順)」を参照してください。

注 – 通常、mailer ディレクトリで、この設定の定義を変更するのは、微調整が必要な場合だけです。

表 14–28 配信エージェントの追加された設定

設定	説明
/=	引数 : ディレクトリのパス。 メールプログラムのプログラムを実行する前に chroot() を適用するディレクトリを指定します。
m=	引数 : define() ルーチンを使って事前に定義した次の m4 の値。 smtp メールプログラムには SMTP_MAILER_MAXMSGS local メールプログラムには LOCAL_MAILER_MAXMSGS relay メールプログラムには RELAY_MAILER_MAXMSGS smtp、local、または relay の各メールプログラムで、1 回の接続で配信するメッセージの数を制限します。
W=	引数 : 増分時間。 すべてのデータの送信後、メールプログラムが戻るまでの最長待ち時間を指定します。

sendmail の version 8.12 から追加されたキューの機能

次に、キューの追加された機能について詳しく説明します。

- 本リリースでは、複数のキューディレクトリをサポートしています。複数のキューを使用するには、次の例のように、アスタリスク (*) で終わっている QueueDirectory オプション値を構成ファイルに追加します。

  ```
  O QueueDirectory=/var/spool/mqueue/q*
  ```

  このオプション値 /var/spool/mqueue/q* は、「q」で始まっているすべてのディレクトリ (またはディレクトリへのシンボリックリンク) をキューのディレクトリとして使用します。sendmail の実行中には、キューのディレクトリ構造を変更しないでください。キューを実行すると、デーモン以外のキューの実行時に冗長フラグ (-v) を使用しないかぎり、各キューについての実行プロセスが作成されます。この新しい項目が、無作為にキューに割り当てられます。
- この新しいキューのファイルの名前付けシステムで使用する名前は、60 年間一意であることが保証されます。このシステムでは、キュー ID が複雑なファイルシステムのロックを使用しないで割り当てられるため、キューにある項目を簡単にほかのキューに移動することができます。

- version 8.12 より、root だけがメールキューを実行できます。この変更の詳細は、mailq(1) のマニュアルページを参照してください。新しい作業手順については、327 ページの「キューディレクトリの管理 (作業マップ)」を参照してください。
- エンベロープの分割に対応するために、キューファイルの名前は 14 文字ではなく、15 文字にします。14 文字までの名前を持つファイルシステムは、サポートされません。

作業手順については、327 ページの「キューディレクトリの管理 (作業マップ)」を参照してください。

sendmail の version 8.12 からの LDAP の変更点

次に、LDAP (Lightweight Directory Access Protocol) を sendmail で使用する際の変更点について説明します。

- LDAPROUTE_EQUIVALENT() および LDAPROUTE_EQUIVALENT_FILE() を使用して、同じホスト名を指定することができます。これらのホスト名は、LDAP ルーティング参照のマスカレードドメイン名と置き換えられます。詳細は、/etc/mail/cf/README を参照してください。
- ftp://ftp.sendmail.org で sendmail とともに配布しているリリースノートで説明しているように、LDAPX マップの名前は LDAP に変更されました。LDAP には、次の構文を使用します。

  ```
  Kldap ldap options
  ```
- 本リリースでは、一度の LDAP 参照に複数の値を返すことができます。次の例のように、返す値を -v オプションを付加したコンマ区切りの文字列に配置します。

  ```
  Kldap ldap -v"mail,more-mail"
  ```
- LDAP マップの宣言で LDAP 属性が指定されていない場合は、一致した属性がすべて返されます。
- このバージョンの sendmail は、LDAP 別名ファイルに指定された引用符などで囲まれたキーや値の文字列内のコンマによって、1 つのエントリが複数のエントリに分割されるのを防止します。
- このバージョンの sendmail には、LDAP マップ用の新しいオプションがあります。この -V*separator* オプションを使用して、区切り文字を指定できます。そのため、検索を行うと、該当する *separator* によって区切られた属性と値の両方を返すことが可能です。
- %s トークンを使用した LDAP フィルタ指定の構文解析に加えて、新しいトークンである %0 を使用して、キーバッファーを符号化することもできます。%0 トークンは、LDAP の特殊文字に対して、文字どおりの意味を適用します。

次の例では、これらのトークンが「*」検索でどのように違うかを説明します。

表 14–29 トークンの比較

LDAP のマップ指定	同等の指定	結果
-k"uid=%s"	-k"uid=*"	ユーザー属性を持つ任意のレコードに一致します
-k"uid=%0"	-k"uid=\2A"	「 *」という名前を持つユーザーに一致します

次の表では、LDAP マップの追加されたフラグについて説明しています。

表 14–30 LDAP マップの追加されたフラグ

フラグ	説明
-1	一致したレコードが 1 つだけだった場合、そのレコードを返します。複数のレコードが一致して返される場合には、結果として、レコードが検出されなかったことと同じとなります。
-r never|always|search|find	LDAP 別名の参照を解除するオプションを設定します。
-Z size	一致したもののうち、返すレコード数を制限します。

sendmail の version 8.12 からの組み込まれたメールプログラムの変更

前のバージョンに組み込まれていたメールプログラム [TCP] は使用できません。代わりに、新しく組み込まれたメールプログラム P=[IPC] を使用してください。プロセス間通信 ([IPC]) 組み込みメールプログラムで、それをサポートするシステム上の UNIX ドメインソケットへの配信を行えるようになりました。このメールプログラムは、指定したソケットで待機している LMTP 配信エージェントとともに使用できます。次に、メールプログラムの例を示します。

```
Mexecmail, P=[IPC], F=lsDFMmnqSXzA5@/:|, E=\r\n,
S=10, R=20/40, T=DNS/RFC822/X-Unix, A=FILE /var/run/lmtpd
```

[IPC] メールプログラムの最初の引数の値が妥当であるか検査されるようになりました。次の表では、最初のメールプログラム引数に設定可能な値について説明しています。

表 14–31 最初のメールプログラム引数に設定可能な値

値	説明
A=FILE	UNIX ドメインソケットによる配信に使用します。

表 14–31 最初のメールプログラム引数に設定可能な値 (続き)

値	説明
A=TCP	TCP/IP 接続に使用します。
A=IPC	最初のメールプログラム引数としては使用できません。

sendmail の version 8.12 から追加されたルールセット

次の表では、追加されたルールセットとその動作について説明しています。

表 14–32 新しいルールセット

ルールセット	説明
check_eoh	ヘッダーから収集した情報を相関させ、欠けているヘッダーを検査します。このルールセットは、マクロストレージマップとともに使用し、すべてのヘッダーが収集されたあと、呼び出されます。
check_etrn	check_rcpt が RCPT を使用するように、ETRN コマンドを使用します。
check_expn	check_rcpt が RCPT を使用するように、EXPN コマンドを使用します。
check_vrfy	check_rcpt が RCPT を使用するように、VRFY コマンドを使用します。

次に、ルールセットの追加機能について説明します。

- 番号が付けられたルールセットには、名前も付けられました。ただし、これらのルールセットに、番号でアクセスすることもできます。
- H ヘッダー構成ファイルコマンドを使用して、デフォルトルールセットを指定し、ヘッダーを確認することができます。各ヘッダーに、独自のルールセットが割り当てられていない場合にだけ、このルールセットが呼び出されます。
- ルールセット内のコメント、つまり括弧内のテキストは、構成ファイルのバージョンが 9 かそれ以上である場合には削除されません。たとえば、次のルールは、入力 token (1) を照合します。ただし、入力 token は照合しません。

```
R$+ (1)         $@ 1
```

- TCP ラッパーまたは check_relay ルールセットが原因でコマンドを拒否する場合でも、sendmail は SMTP RSET コマンドを受け入れます。
- OperatorChars オプションを何度も設定すると、警告が送信されます。また、ルールセットを定義したあとで OperatorChars を設定しないでください。
- 無効なルールセットを宣言すると、行だけでなく、そのルールセットの名前も無視されます。そのルールセットの行は S0 に追加されません。

sendmail の version 8.12 からのファイルの変更点

次の変更に注意してください。

- Solaris 10 以降のリリースでは、読み取り専用の /usr ファイルシステムをサポートするために、/usr/lib/mail ディレクトリの内容が /etc/mail/cf ディレクトリに移動されました。詳細は、359 ページの「/etc/mail/cf ディレクトリの内容」を参照してください。ただし、シェルスクリプト /usr/lib/mail/sh/check-hostname および /usr/lib/mail/sh/check-permissions は、/usr/sbin ディレクトリに置かれるようになりました。362 ページの「メールサービスに使用するその他のファイル」を参照してください。下位互換性を確保するために、シンボリックリンクが各ファイルの新しい位置を示します。
- /usr/lib/mail/cf/main-v7sun.mc の新しい名前は /etc/mail/cf/cf/main.mc です。
- /usr/lib/mail/cf/subsidiary-v7sun.mc の新しい名前は /etc/mail/cf/cf/subsidiary.mc です。
- helpfile は /etc/mail/helpfile にあります。古い名前 (/etc/mail/sendmail.hf) には、新しい名前へのシンボリックリンクがあります。
- trusted-users ファイルは /etc/mail/trusted-users にあります。アップグレード中に、新しい名前は検出されず、古い名前である /etc/mail/sendmail.ct が検出されると、古い名前から新しい名前へのハードリンクが作成されます。それ以外の場合には、変更されません。デフォルトの内容は、root です。
- local-host-names ファイルは /etc/mail/local-host-names にあります。アップグレード中に、新しい名前は検出されず、古い名前である /etc/mail/sendmail.cw が検出されると、古い名前から新しい名前へのハードリンクが作成されます。それ以外の場合には、変更されません。デフォルトの内容は、ゼロ長です。

sendmail version 8.12 と構成内の IPv6 アドレス

sendmail の version 8.12 より、アドレスを正しく識別するために、構成に使用する IPv6 アドレスの前に IPv6: タグを付ける必要があります。IPv6 アドレスを識別しない場合は、タグを前に付けません。

パート V

シリアルネットワーキング (トピック)

シリアルネットワーキングについてのこのパートでは、PPP と UUCP の概要、作業、リファレンス情報について説明します。

◆ ◆ ◆ 第 15 章

Solaris PPP 4.0 (概要)

このパートでは、シリアルネットワーキングのトピックについて説明します。シリアルネットワーキングとは、RS-232 ポートや V.35 ポートのようなシリアルインタフェースを使用して、データ転送のために 2 つ以上のコンピュータを接続することをいいます。Ethernet などの LAN インタフェースとは異なり、これらのシリアルインタフェースは、距離の離れたシステムを接続するために使用します。PPP (ポイントツーポイントプロトコル) および UUCP (UNIX 間コピープログラム) は、シリアルネットワークを実装するために使用できる個別の技術です。シリアルインタフェースをネットワーク用に構成すると、複数のユーザーが、Ethernet などのほかのネットワークインタフェースとほぼ同様に使用できるようになります。

この章では、Solaris PPP 4.0 について紹介します。PPP のこのバージョンでは、PPP を使用することで、物理的に離れた場所にある 2 つのコンピュータがさまざまな媒体を介して互いに通信することができます。Solaris 9 リリースより、Solaris PPP 4.0 は基本インストールの一部に組み込まれています。

この章では、次の内容について説明します。

Solaris PPP 4.0 の基本

Solaris PPP 4.0 は、TCP/IP プロトコル群に含まれるデータリンクプロトコルとしてポイントツーポイントプロトコル (PPP) を実装しています。PPP は、2 つの端点にあるマシン間でデータを電話回線などの通信媒体を介して転送する方法について記述しています。

PPP は、1990 年代の初期から、通信リンクを介してデータグラムを送信するために幅広く使用されてきたインターネット標準です。PPP 標準は、Internet Engineering

Task Force (IETF) のポイントツーポイントワーキンググループによって RFC 1661 に定義されています。PPP は一般に、リモートコンピュータがインターネットサービスプロバイダ (ISP) を呼び出したり、着呼を受信するように構成されている企業サーバーを呼び出したりするときに使用されます。

Solaris PPP 4.0 は、広く普及している Australian National University (ANU) PPP–2.4 に基づいて PPP 標準を実装しています。PPP リンクは非同期と同期の両方をサポートしています。

Solaris PPP 4.0 の互換性

さまざまなバージョンの PPP 標準がインターネットコミュニティーで広く使用されています。ANU PPP-2.4 は、Linux、Tru64 UNIX、および次の BSD 系統の主要 OS で採用されています。

- FreeBSD
- OpenBSD
- NetBSD

Solaris PPP 4.0 は、Solaris オペレーティングシステムで実行されているマシンに ANU PPP-2.4 の高度な構成機能を提供します。Solaris PPP 4.0 が実行されているマシンでは、PPP 標準が実行されているマシンに PPP リンクを簡単に設定できます。

ANU ベースの PPP 以外で Solaris PPP 4.0 と正常に相互運用できるものは、次のとおりです。

- Solaris 2.4 から Solaris 8 までで稼働する Solaris PPP (別名 `asppp`)
- Solstice PPP 3.0.1
- Microsoft Windows 98 DUN
- Cisco IOS 12.0 (同期)

使用する Solaris PPP のバージョン

Solaris 9 より、Solaris PPP 4.0 がサポート対象の PPP となりました。Solaris 9 および Solaris 10 には、以前の非同期 Solaris PPP (`asppp`) ソフトウェアは組み込まれていません。詳細は、次を参照してください。

- 第 23 章「非同期 Solaris PPP から Solaris PPP 4.0 への移行 (手順)」
- Solaris 8 System Administrator Collection (http://docs.sun.com)

Solaris PPP 4.0 を使用する理由

現在 asppp を使用しているユーザーには、Solaris PPP 4.0 への移行をお薦めします。2 つの Solaris PPP 間の技術的な相違点は次のとおりです。

- 転送モード

  asppp は非同期通信だけに対応します。Solaris PPP 4.0 は非同期通信と同期通信の両方に対応します。

- 構成プロセス

  asppp の設定には、asppp.cf 構成ファイル、3 つの UUCP ファイル、および ifconfig コマンドが必要です。さらに、マシンにログインするユーザーのために、あらかじめインタフェースを構成しておく必要があります。

  Solaris PPP 4.0 の設定では、PPP 構成ファイルのオプションを定義するか、オプションを指定して pppd コマンドを発行します。また、構成ファイルとコマンド行の両方の方法を組み合わせて使用することもできます。Solaris PPP はインタフェースを動的に作成して削除します。各ユーザーのために PPP インタフェースを構成する必要はありません。

- **asppp** が提供しない **Solaris PPP 4.0** の機能
  - MS-CHAPv1 および MS-CHAPv2 認証
  - ADSL ブリッジをサポートする PPP over Ethernet (PPPoE)
  - PAM 認証
  - プラグインモジュール群
  - IPv6 アドレス指定
  - Deflate 圧縮または BSD 圧縮を使用するデータ圧縮
  - Microsoft のクライアント側のコールバックのサポート

Solaris PPP 4.0 のアップグレードパス

既存の asppp 構成を Solaris PPP 4.0 に変換する場合は、このリリースが提供する変換スクリプトを使用できます。詳細は、553 ページの「asppp から Solaris PPP 4.0 に変換する方法」を参照してください。

PPP の詳細情報

PPP に関する多くの情報は印刷物やオンラインで入手可能です。参考資料のいくつかを以降で示します。

PPP に関する専門技術者向けのリファレンスブック

ANU PPP など、幅広く使用されている PPP については、次の図書を参照してください。

- Carlson, James 著、『PPP Design, Implementation, and Debugging』第 2 版、Addison-Wesley、2000
- Sun, Andrew 著、『Using and Managing PPP』、O'Reilly & Associates、1999

PPP に関する Web サイト

PPP の一般的な情報については、次の Web サイトを参照してください。

- 技術情報、FAQ、Solaris システム管理、および前バージョンの PPP については、Sun Microsystems のシステム管理者の資源 (http://www.sun.com/bigadmin/home/index.html) を参照してください。
- さまざまな PPP のモデム構成とアドバイスについては、Stokely Consulting が提供する Web Project Management & Software Development の Web サイト (http://www.stokely.com/unix.serial.port.resources/ppp.slip.html) を参照してください。

PPP に関する RFC (Requests for Comments)

PPP に関する有用な Internet RFC は次のとおりです。

- RFC 1661 と RFC 1662。PPP の主な機能を解説しています
- RFC 1334。パスワード認証プロトコル (PAP) とチャレンジハンドシェーク認証プロトコル (CHAP) などの認証プロトコルを解説しています
- RFC 1332。PPP over Ethernet (PPPoE) を解説しています

PPP RFC のコピーを入手するには、IETF RFC の Web ページ (http://www.ietf.org/rfc.html) で RFC の番号を指定してください。

PPP に関するマニュアルページ

Solaris PPP 4.0 の実装については、次のマニュアルページを参照してください。

- `pppd(1M)`
- `chat(1M)`
- `pppstats(1M)`
- `pppoec(1M)`
- `pppoed(1M)`
- `sppptun(1M)`
- `snoop(1m)`

また、`pppdump(1M)` のマニュアルページも参照してください。PPP のマニュアルページについては、`man` コマンドを使用してください。

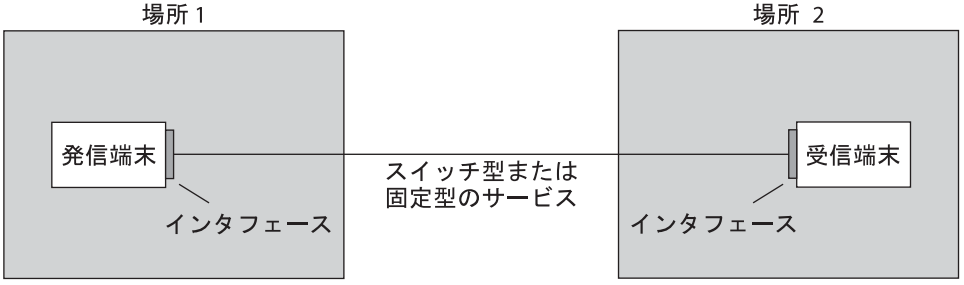
PPP 構成と用語

この節では、PPP 構成について説明します。また、このマニュアルで使用する用語についても説明します。

Solaris PPP 4.0 は次の構成をサポートします。

- スイッチ型のアクセス構成 (ダイアルアップ)
- 固定型の構成 (専用回線)

図 15–1 PPP リンクの構成要素



上図は、基本的な PPP リンクを示しています。リンクの構成要素は、次のようになります。

- 2 つのマシン。通常、ピアと呼ばれ、物理的に互いに離れた場所に配置されています。ピアは、サイトの要件によってパーソナルコンピュータ、エンジニアリングワークステーション、大規模サーバー、商用ルーターなどが考えられます。
- 各ピアに対するシリアルインタフェース。Solaris マシンのインタフェースは、構成する PPP が非同期か同期かによって、`cua`、`hihp` などが考えられます。
- シリアルケーブル、モデム接続などの物理リンク、またはネットワークプロバイダが提供する T1 回線や T3 回線などの専用回線。

ダイアルアップ PPP の概要

もっともよく使用される PPP 構成は、ダイアルアップリンクです。ダイアルアップリンクでは、ローカルピアがリモートピアをダイアルアップして接続を確立し、PPP を実行します。ダイアルアッププロセスでは、ローカルピアがリモートピアの電話番号を呼び出してリンクを開始します。

一般的なダイアルアップの使用例では、ユーザーの自宅にあるコンピュータが、着呼を受信するように構成されている ISP 側のピアを呼び出します。別のダイアルアップの使用例では、企業サイトでローカルマシンが PPP リンクを使用して、別の建物内にあるピアにデータを転送します。

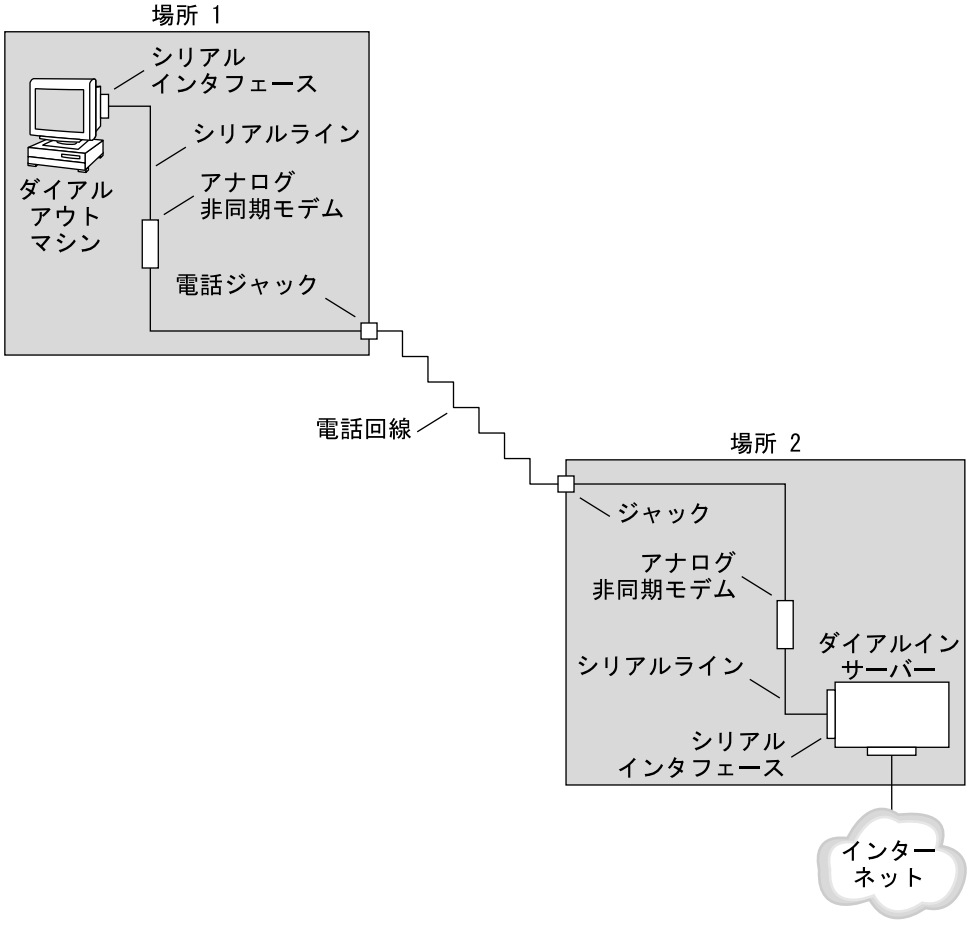
このマニュアルでは、ダイアルアップ接続を開始するローカルピアは、ダイアルアウトマシンと呼びます。着呼を受信するピアは、ダイアルインサーバーと呼びます。このマシンは実際にはダイアルアウトマシンがターゲットにするマシンに過ぎず、真の意味でのサーバーではない場合もあります。

PPP はクライアントサーバープロトコルではありません。PPP のドキュメントの中には、通話の確立に言及する場合に「クライアント」や「サーバー」という用語を使っているものもあります。ダイアルインサーバーは、ファイルサーバーやネームサーバーのような真の意味でのサーバーではありません。ダイアルインサーバーという用語は、ダイアルインマシンが複数のダイアルアウトマシンにネットワークでのアクセス可能性を「提供」していることから、PPP 用語として幅広く使用されています。それでもダイアルインサーバーは、現実には、ダイアルアウトマシンのターゲットピアにすぎません。

ダイアルアップ **PPP** リンクの構成要素

次の図を参照してください。

図 15–2　基本的なアナログダイアルアップ PPP リンク



リンクのダイアルアウト側 (場所 1) の構成は、次の要素から成ります。

- ダイアルアウトマシン。一般に、個々の家庭のパーソナルコンピュータやワークステーション。
- ダイアルアウトマシン上のシリアルインタフェース。 `/dev/cua/a` または `/dev/cua/b` は、Solaris ソフトウェアが実行されているマシン上で発呼に使用する標準のシリアルインタフェースです。
- 電話のジャックに接続される非同期モデムまたは ISDN 端末アダプタ (TA)。
- 電話会社の電話回線やサービス。

リンクのダイアルイン側 (場所 2) の構成は、次の要素から成ります。

- 電話ネットワークに接続される電話のジャックまたは類似のコネクタ
- 非同期モデムまたは ISDN TA
- ダイアルインサーバー上のシリアルインタフェース。`ttya` または `ttyb` は、ダイアルインサーバー上で着呼に使用するシリアルインタフェースです
- ダイアルインサーバー。企業のイントラネットなどのネットワークや ISP のインスタンス内からグローバルインターネットに接続されます

ダイアルアウトマシンで ISDN 端末アダプタを使用する

外付けの ISDN TA はモデムよりも高速ですが、両者の構成方法は基本的に同じです。両者の主な相違は chat スクリプト間の構成にあります。ISDN TA の場合、chat スクリプトの記述では、TA の製造元に固有のコマンドが必要になります。ISDN TA 用の chat スクリプトについては、524 ページの「外部 ISDN TA 用 chat スクリプト」を参照してください。

ダイアルアップ通信中の動作

ダイアルアウトとダイアルインの両方のピアにある PPP 構成ファイルには、リンクを設定するための命令群が含まれています。ダイアルアップリンクが開始されると、次のプロセスが発生します。

1. ダイアルアウトマシン上のユーザーまたはプロセスは、`pppd` コマンドを実行してリンクを開始します。
2. ダイアルアウトマシンは PPP 構成ファイルを読み取ります。次に、シリアル回線を介して、ダイアルインサーバーの電話番号などの命令群をモデムに送信します。
3. モデムは電話番号をダイアルして、ダイアルインサーバー側のモデムと電話接続を確立します。

   ダイアルアウトマシンが、モデムとダイアルインサーバーに送信する一連のテキスト文字列は、*chat* スクリプトと呼ばれるファイルに格納されています。ダイアルアウトマシンは、必要に応じて、ダイアルインサーバーにコマンドを送信し、サーバー側の PPP を呼び出します。
4. ダイアルインサーバーに接続されているモデムは、ダイアルアウトマシン側のモデムとリンクのネゴシエーションを開始します。
5. モデム同士のネゴシエーションが完了すると、ダイアルアウトマシン側のモデムは「CONNECT」を通知します。
6. 両方のピア側の PPP は確立フェーズに入ります。このフェーズでは、リンク制御プロトコル (LCP) が基本的なリンクパラメータと認証の使用をネゴシエートします。
7. ピアは、必要に応じて、互いを認証します。

8. PPP のネットワーク制御プロトコル (NCP) は、IPv4 や IPv6 などのネットワークプロトコルの使用をネゴシエートします。

ダイアルアウトマシンでは、ダイアルインサーバーを通って到達可能なホストに `telnet` または類似のコマンドを実行できます。

専用回線 PPP の概要

固定型の専用回線の PPP 構成には、リンクで接続された 2 つのピアが含まれます。リンクは、プロバイダからリースされたスイッチ型または非スイッチ型のデジタルサービスで構成されています。Solaris PPP 4.0 は、全二重でポイントツーポイントの専用回線媒体を介して動作します。通常、会社では、ネットワークプロバイダから専用リンクをレンタルして、ISP またはほかのリモートサイトに接続します。

ダイアルアップリンクと専用回線リンクの比較

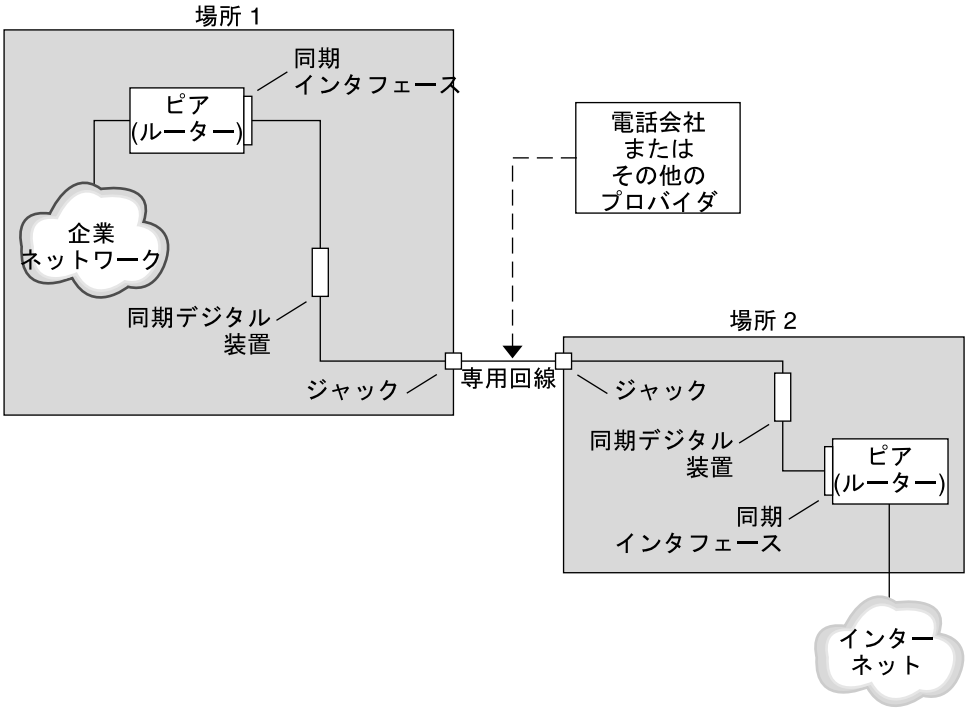
ダイアルアップと専用回線のリンクはともに、通信媒体で接続されている 2 つのピアから成っています。次の表は、2 つのリンクタイプの相違をまとめています。

専用回線	ダイアルアップ回線
システム管理者による電源切断または電源障害による電源切断がないかぎり常時接続されています。	ユーザーがリモートピアを呼び出そうとするとき開始されます。
同期通信と非同期通信を使用します。非同期通信では、多くの場合長距離モデムを使用します。	非同期通信を使用します。
プロバイダからレンタルします。	既存の電話回線を使用します。
同期装置を必要とします。	低コストのモデムを使用します。
ほとんどの SPARC システムで一般的に使用されている同期ポートを必要とします。ただし、同期ポートは、x86 システムおよび最新の SPARC システムでは通常使用されません。	通常のコンピュータに組み込まれている標準のシリアルインタフェースを使用します。

専用回線 PPP リンクの構成要素

次の図を参照してください。

図 15–3 専用回線の基本的な構成



専用回線リンクの構成要素は次のとおりです。

- **2** つのピア。リンクの両端に 1 つずつ存在します。各ピアは、ワークステーションかサーバーです。通常ピアは、ネットワークまたはインターネットともう一方の側のピアとの間のルーターとして機能します。
- 各ピア上の同期インタフェース。Solaris ソフトウェアが実行されている一部のマシンは、専用回線に接続するために、HSI/S などの同期インタフェースカードを購入する必要があります。UltraSPARC ワークステーションなどのマシンには同期インタフェースが内蔵されています。
- 各ピア上の **CSU/DSU** 同期デジタル装置。同期ポートを専用回線に接続します。

  現場の事情によって、CSU は DSU に組み込まれていたり、個人で所有していたり、プロバイダからリースしていたりします。DSU はマシンに標準の同期シリアルインタフェースを提供します。フレームリレーを使用する場合、フレームリレーアクセスデバイス (FRAD) が、シリアルインタフェースに適合するように調整します。
- 専用回線。スイッチ型または非スイッチ型のデジタルサービスを提供します。専用回線のデジタルサービスには、SONET/SDH、Frame Relay PVC、T1 などがあります。

専用回線通信中の動作

ほとんどのタイプの専用回線では、ピアは互いにダイアルすることはありません。会社では専用回線サービスを購入して、2 つの定められた場所の間を明示的に接続します。場合によって、専用回線の各端にある 2 つのピアは同じ会社でも物理的に離れた場所に存在することもあります。別の事例では、会社が、ISP に接続されている専用回線上にルーターを設定している場合があります。

専用回線の固定型のリンクは設定が簡単ですが、ダイアルアップリンクほどは普及していません。固定型のリンクは chat スクリプトを必要としません。専用回線の場合、両方のピアは互いを知っているので、認証を使用しないのが普通です。2 つのピアがリンクを介して PPP を開始すると、リンクはアクティブな状態を続けます。専用回線に障害が発生したり、どちらかのピアが明示的にリンクを終了したりしないかぎり、専用回線の固定型のリンクはアクティブな状態を続けます。

Solaris PPP 4.0 が実行されている専用回線上のピアは、ダイアルアップリンクを定義する構成ファイルとほぼ同じものを使用します。

専用回線を介した通信を開始する場合、次のプロセスが発生します。

1. 各ピアマシンは、`pppd` コマンドを起動プロセスや別の管理スクリプトの一部として実行します。
2. 両方のピアは自分の PPP 構成ファイルを読み取ります。
3. 両方のピアは通信パラメータをネゴシエートします。
4. IP リンクが確立されます。

PPP 認証

認証は、要求しているのがユーザー本人であることを確認するためのプロセスです。UNIX のログインの流れは、次のように簡単な認証形式です。

1. `login` コマンドを入力すると、ユーザーに名前とパスワードの入力を求めるプロンプトが表示されます。
2. 次に `login` は、ユーザーを認証するために、入力された名前とパスワードをパスワードデータベースから探そうとします。
3. データベース中にユーザー名とパスワードが存在する場合、ユーザーは認証されて、システムへのアクセスが許可されます。データベース中にユーザー名とパスワードが存在しない場合、ユーザーはシステムへのアクセスを拒否されます。

デフォルトでは、Solaris PPP 4.0 は、デフォルトの経路が指定されていないマシン上では認証を要求しません。したがって、デフォルトの経路が指定されていないローカルマシンはリモート呼び出しを認証しません。逆に、マシンにデフォルトの経路が定義されていれば、マシンは、常にリモート呼び出しを認証します。

必要な場合、自分のマシンに PPP リンクを設定しようとしている呼び出し側の識別情報を、PPP 認証プロトコルを使って確認できます。逆に、呼び出し側を認証するピアをローカルマシンが呼び出す必要がある場合は、PPP 認証情報をローカルマシンに構成しておく必要があります。

認証する側と認証される側

PPP リンク上の呼び出し側マシンは、リモートピアに対して識別情報を示す必要があるので、認証される側とみなされます。ピアは、認証する側とみなされます。認証する側は、呼び出し側の識別情報をセキュリティープロトコル用の適切な PPP ファイルから探し、その呼び出し側を認証したり認証を拒否したりします。

多くの場合、PPP 認証をダイアルアップリンクに構成します。呼び出しが開始されると、ダイアルアウトマシンが認証される側になります。ダイアルインサーバーは認証する側になります。サーバーはデータベースを秘密ファイルの形式で保持します。このファイルには、サーバーに PPP リンクを設定する許可が与えられているすべてのユーザーが記述されています。許可が与えられているユーザーは信頼できる呼び出し側とみなされます。

一部のダイアルアウトマシンには、ダイアルアウトマシンの呼び出しに対する応答でリモートピアに認証情報の提供を要求するものがあります。このような場合は、役割が逆転し、リモートピアは認証される側になり、ダイアルアウトマシンは認証する側になります。

注 – PPP 4.0 は専用回線でピアによる認証を禁止していませんが、通常は専用回線で認証を使用することはありません。専用回線規約では、回線の両端に存在する両者が互いをよく知っており、信頼していることが特徴となっています。しかし、PPP 認証は管理が簡単なので、専用回線にも認証を実装することをまじめに検討する必要があります。

PPP の認証プロトコル

PPP の認証プロトコルは、パスワード認証プロトコル (PAP) とチャレンジハンドシェーク認証プロトコル (CHAP) です。各プロトコルは、ローカルマシンにリンクする許可が与えられている各呼び出し側に対して、識別情報が格納された「秘密データベース」や「セキュリティー資格情報」を使用します。PAP については、528 ページの「パスワード認証プロトコル (PAP)」を参照してください。CHAP については、531 ページの「チャレンジハンドシェーク認証プロトコル (CHAP)」を参照してください。

PPP 認証を使用する理由

PPP リンクでの認証は任意です。また、認証では、ピアが信頼されていることは確認しますが、PPP 認証に機密保護を提供していません。機密保護では、IPsec、PGP、SSL、Kerberos、Solaris セキュアシェルなどの暗号化ソフトウェアを使用します。

注 – Solaris PPP 4.0 は、RFC 1968 に記述されている PPP Encryption Control Protocol (ECP) を実装していません。

次の場合に、PPP 認証の実装を検討してください。

- 会社が、公衆電話交換網を介してユーザーから着呼を受け取る。
- 会社のファイアウォールを介してネットワークにアクセスする場合やセキュリティーで保護されたトランザクションに関係する場合に、会社のセキュリティーポリシーでリモートユーザーに認証資格情報の提供を要求している。
- 標準の UNIX パスワードデータベース (`/etc/passwd`、NIS、LDAP、または PAM) と照合して呼び出し側を認証したいとする。この場合は PAP 認証を使用する。
- 会社のダイアルインサーバーがネットワークのインターネット接続も提供する。この場合は PAP 認証を使用する。
- シリアル回線が、リンクのどちらか端にあるネットワークやマシン上のパスワードデータベースよりもセキュリティーの保護が弱い。この場合は CHAP 認証を使用する。

PPPoE による DSL ユーザーのサポート

多くのネットワークプロバイダと自宅で仕事をしている個人は、デジタル加入者回線 (DSL) 技術を使用して、高速なネットワークアクセスを実現します。DSL ユーザーをサポートするために、Solaris PPP 4.0 は PPP over Ethernet (PPPoE) 機能を組み込んでいます。PPPoE 技術を使用することで、複数のホストが 1 つの Ethernet リンクを介して 1 つ以上の地点に PPP セッションを実行できます。

次の場合に、PPPoE を使用する必要があります。

- DSL ユーザー (自分自身も含む場合もある) をサポートする。DSL サービスプロバイダは、DSL 回線を介してサービスを受け取るために、ユーザーに PPPoE トンネルの構成を要求することがある。
- サイトが、顧客に PPPoE を提供する ISP である。

この節では、PPPoE に関連する用語と基本的な PPPoE 技術の概要について説明します。

PPPoE の概要

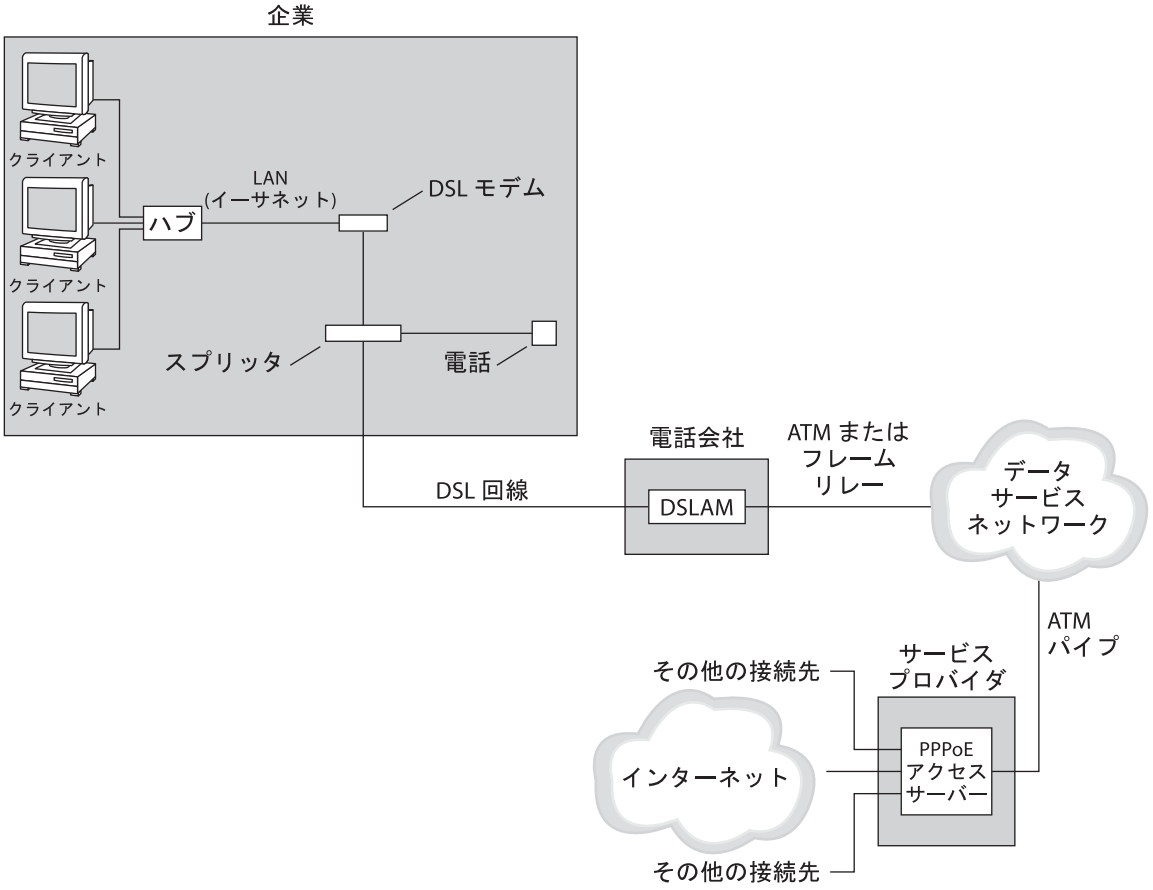
PPPoE は、RedBack Networks が生み出した独自のプロトコルです。PPPoE は、別バージョンの標準 PPP ではなく検出プロトコルです。PPPoE のシナリオでは、最初に PPP 通信を開始するマシンが、PPPoE を実行しているピアを検出する必要があります。PPPoE プロトコルは、Ethernet ブロードキャストパケットを使ってピアを検出します。

検出プロセスを終了したら、PPPoE は、開始したホスト (PPPoE クライアント) からピア (PPPoE アクセスサーバー) まで Ethernet ベースのトンネルを設定します。トンネリングとは、あるプロトコルを、別のプロトコルで実行する方法です。PPPoE を使用して、Solaris PPP 4.0 は PPP に Ethernet IEEE 802.2 を介したトンネルを作成します。PPP と Ethernet IEEE 802.2 はともにデータリンクプロトコルです。設定された PPP 接続は、PPPoE クライアントとアクセスサーバーの間で専用リンクのように動作します。PPPoE については、536 ページの「DSL サポート用の PPPoE トンネルの作成」を参照してください。

PPPoE の構成要素

次の図に示すように、PPPoE 構成には、消費者、電話会社、およびサービスプロバイダという 3 つの関係者が存在します。

図 15–4　PPPoE トンネル内の関係者



PPPoE の消費者

システム管理者として、消費者の PPPoE 構成を助けることがあります。PPPoE 消費者の一般的なタイプは、DSL 回線を介して PPPoE を実行する個人です。別の PPPoE 消費者は、上図に示すように、従業員が PPPoE トンネルを実行できるように DSL 回線を購入する会社です。

企業消費者が PPPoE を使用する主な理由は、高速の DSL 機器を介して多くのホストに PPP 通信を提供するためです。通常、単独の PPPoE クライアントは、個人で *DSL* モデムを持ちます。また、ハブに接続されているクライアントのグループは、Ethernet 回線によって同じハブに接続されている DSL モデムを共有することがあります。

注 – DSL 機器は技術的にはモデムではなくブリッジです。ただし、実際にはこれらのデバイスをモデムと呼んでいるので、このマニュアルでは、「DSL モデム」という用語を使用します。

PPPoE は、DSL モデムに接続されている Ethernet 回線上のトンネルを介して PPP を実行します。その回線はスプリッタに接続され、スプリッタは電話回線に接続しています。

電話会社の PPPoE

PPPoE のシナリオでは、電話会社は中間に位置します。電話会社は、電話回線を介して受信する信号を、デジタル加入者線アクセスマルチプレクサ *(DSLAM)* と呼ばれるデバイスを使って分割します。DSLAM は分割した信号を別の線、電話サービス用アナログ線、および PPPoE 用デジタル線に送り出します。デジタル線は ATM データネットワークを介してトンネルを DSLAM から ISP まで延長します。

サービスプロバイダの PPPoE

ISP は、ATM データネットワークから渡される PPPoE 転送をブリッジを介して受信します。ISP では、PPPoE が実行されているアクセスサーバーが PPP リンクのピアとして機能します。アクセスサーバーは、図 15–2 で紹介したダイアルインサーバーと機能的に類似していますが、アクセスサーバーがモデムを使用しない点が異なります。アクセスサーバーは、個々の PPPoE セッションをインターネットアクセスなどの通常の IP トラフィックに変換します。

ISP のシステム管理者は、アクセスサーバーの構成と維持を行います。

PPPoE トンネルのセキュリティー

PPPoE トンネルは最初からセキュリティー対策が行われていません。PAP または CHAP を使用することで、トンネルを介して実行している PPP リンクにユーザー認証を提供できます。

◆ ◆ ◆ 第 16 章

PPP リンクの計画 (手順)

PPP リンクの設定には、作業計画や PPP と無関係な作業など、さまざまな個別の作業が含まれています。この章では、もっとも一般的な PPP リンク、認証、および PPPoE を計画する方法について説明します。

第 16 章「PPP リンクの計画 (手順)」に続く各章では、特定リンクの設定方法について構成例を使って説明します。これらの構成例はこの章で紹介します。

ここでは、次の内容を説明します。

全体的な PPP 計画 (作業マップ)

PPP では、実際にリンクを設定する前に作業計画を立てる必要があります。さらに、PPPoE トンネルを使用する場合は、まず PPP リンクを設定し、それからトンネルを提供する必要があります。次の作業マップは、この章で説明する大規模な作業計画を示しています。構成するリンクタイプによっては、一般的な作業だけで十分な場合があります。また、リンク、認証、および PPPoE の各作業が必要になる場合もあります。

表 16–1 PPP 計画 (作業マップ)

作業	説明	参照先
ダイアルアップ PPP リンクを計画します	ダイアルアウトマシンまたはダイアルインサーバーの設定に必要な情報を収集します	426 ページの「ダイアルアップ PPP リンクの計画」
専用回線リンクを計画します	専用回線にクライアントを設定するための必要情報を収集します	429 ページの「専用回線リンクの計画」

表 16–1 PPP 計画 (作業マップ) (続き)

作業	説明	参照先
PPP リンクの認証を計画します	PPP リンクに PAP 認証または CHAP 認証を構成するための必要情報を収集します	432 ページの「リンクへの認証計画」
PPPoE トンネルを計画します	PPP リンクが実行できる PPPoE トンネルを設定するための必要情報を収集します	437 ページの「PPPoE トンネルを介した DSL サポートの計画」

ダイアルアップ PPP リンクの計画

ダイアルアップリンクはもっともよく使用される PPP リンクです。この節では、次の内容について説明します。

- ダイアルアップリンクの計画情報
- 第 17 章「ダイアルアップ PPP リンクの設定 (手順)」で使用されるリンク例の説明

通常は、マシンをダイアルアップ PPP リンク、ダイアルアウトマシン、またはダイアルインサーバーの一方の端に構成するだけです。ダイアルアップ PPP の概要については、413 ページの「ダイアルアップ PPP の概要」を参照してください。

ダイアルアウトマシンを設定する前に

ダイアルアウトマシンを構成する前に、次の表に示されている情報を収集します。

注 – この節の計画情報には、認証や PPPoE について収集する情報は含まれていません。認証計画については、432 ページの「リンクへの認証計画」を参照してください。PPPoE 計画については、437 ページの「PPPoE トンネルを介した DSL サポートの計画」を参照してください。

表 16–2 ダイアルアウトマシンの情報

情報	動作
最大モデム速度	モデムの製造元が提供するマニュアルを参照します。
モデム接続コマンド (AT コマンド)	モデムの製造元が提供するマニュアルを参照します。
リンクの一方の端で使用するダイアルインサーバーの名前	ダイアルインサーバーの識別が簡単な名前を作成します。
ダイアルインサーバーで必要なログインシーケンス	ダイアルインサーバーの管理者に問い合わせるか、ダイアルインサーバーが ISP 側に存在すれば、ISP のマニュアルを参照します。

ダイアルインサーバーを設定する前に

ダイアルインサーバーを構成する前に、次の表に示されている情報を収集します。

注 – この節の計画情報には、認証や PPPoE について収集する情報は含まれていません。認証計画については、432 ページの「リンクへの認証計画」を参照してください。PPPoE 計画については、437 ページの「PPPoE トンネルを介した DSL サポートの計画」を参照してください。

表 16–3 ダイアルインサーバーの情報

情報	動作
最大モデム速度	モデムの製造元が提供するマニュアルを参照します。
ダイアルインサーバーの呼び出しが許可されている人のユーザー名	451 ページの「ダイアルインサーバーのユーザーを構成する方法」で説明するようなホームディレクトリを設定する前に、予想されるユーザーの名前を入手します。
PPP 通信の専用 IP アドレス	会社での IP アドレスの委譲に責任を持つ担当者からアドレスを入手します。

ダイアルアップ PPP の構成例

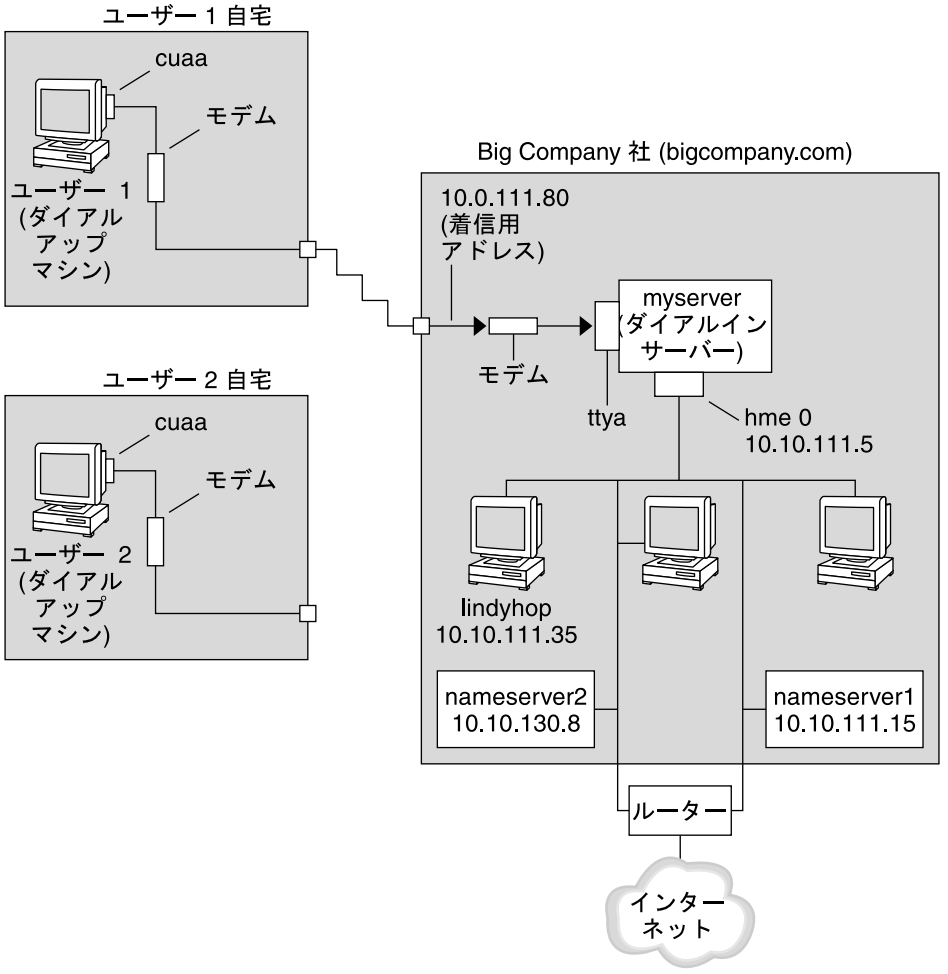
第 17 章「ダイアルアップ PPP リンクの設定 (手順)」で紹介する作業では、従業員に週に 2、3 日在宅勤務させるための小企業の要件を実行します。一部の従業員は、ホームマシンに Solaris OS が必要になります。また、社内イントラネット上にある作業マシンにリモートログインすることも必要になります。

作業では、次のような基本的なダイアルアップリンクを設定します。

- ダイアルアウトマシンが、社内イントラネットを呼び出す従業員の自宅に存在する。
- ダイアルインサーバーは、従業員からの着呼を受信するように構成された社内イントラネット上のマシンである。
- UNIX スタイルのログインを使用して、ダイアルアウトマシンを認証する。Solaris PPP 4.0 の強力な認証方法は、この会社のセキュリティーポリシーには必要ない。

次の図は、第 17 章「ダイアルアップ PPP リンクの設定 (手順)」で設定されているリンクを示します。

図 16–1 ダイアルアップリンクの例



この図では、リモートホストが電話回線上のモデルを介して Big Company 社のイントラネットにダイアルアウトしています。もう 1 台のホストが Big Company 社にダイアルアウトするように構成されていますが、現在アクティブではありません。Big Company 社のダイアルインサーバーに接続されているモデムが、リモートユーザーからの呼び出しに順に応答しています。PPP 接続はピア間で確立しています。ダイアルアウトマシンは、イントラネット上のホストマシンにリモートログインできます。

ダイアルアップ PPP の詳細情報

次を参照してください。

- ダイアルアウトマシンを設定する手順については、表 17–2 を参照してください。
- ダイアルインマシンを設定する手順については、表 17–3 を参照してください。
- ダイアルアップリンクの概要については、413 ページの「ダイアルアップ PPP の概要」を参照してください。
- PPP のファイルとコマンドの詳細については、505 ページの「ファイルおよびコマンド行での PPP オプションの使用」を参照してください。

専用回線リンクの計画

専用回線リンクの設定では、プロバイダからリースしているスイッチ型または非スイッチ型サービスの一方の端にピアを構成する必要があります。

この節では、次の内容について説明します。

- 専用回線リンクの計画情報
- 図 16–2 に示されているリンク例の説明

専用回線リンクの概要については、417 ページの「専用回線 PPP の概要」を参照してください。専用回線の設定作業については、第 18 章「専用回線 PPP リンクの設定 (手順)」を参照してください。

専用回線リンクを設定する前に

会社がネットワークプロバイダから専用回線リンクをレンタルしている場合は、リンクの自分側の端だけにシステムを構成します。リンクのもう一方の端にあるピアは、別の管理者が維持しています。この管理者は、会社から離れた場所にいるシステム管理者か、ISP 側のシステム管理者のどちらかです。

専用回線リンクに必要なハードウェア

リンク媒体の他に、リンクの端には次のハードウェアが必要です。

- システム用の同期インタフェース
- 同期装置 (CSU/DSU)
- 自分のシステム

一部のネットワークプロバイダでは、顧客宅内機器 (CPE) として、ルーター、同期インタフェース、および CSU/DSU が必要です。ただし、必要な機器は、プロバイダや国別の政府規制によって変わります。ネットワークプロバイダでは、必要な装置で専用回線と共に提供されないものは、それに関する情報を提供しています。

専用回線のために収集する情報

ローカルピアを構成する前に、次の表に示されている項目を調べておく必要があります。

表 16–4 専用回線リンクの計画

情報	動作
インタフェースのデバイス名	インタフェースカードのマニュアルを参照します。
同期インタフェースカードの構成手順	インタフェースカードのマニュアルを参照します。この情報は、HSI/S インタフェースを構成する場合に必要です。ほかのタイプのインタフェースカードでは、構成する必要がない場合があります。
(任意) リモートピアの IP アドレス	サービスプロバイダのマニュアルを参照するか、リモートピアのシステム管理者に問い合わせます。この情報は、2 つのピア間で IP アドレスがネゴシエートされない場合にだけ必要です。
(任意) リモートピアの名前	サービスプロバイダのマニュアルを参照するか、リモートピアのシステム管理者に問い合わせます。
(任意) リンクの速度	サービスプロバイダのマニュアルを参照するか、リモートピアのシステム管理者に問い合わせます。
(任意) リモートピアで使用する圧縮	サービスプロバイダのマニュアルを参照するか、リモートピアのシステム管理者に問い合わせます。

専用回線リンクの構成例

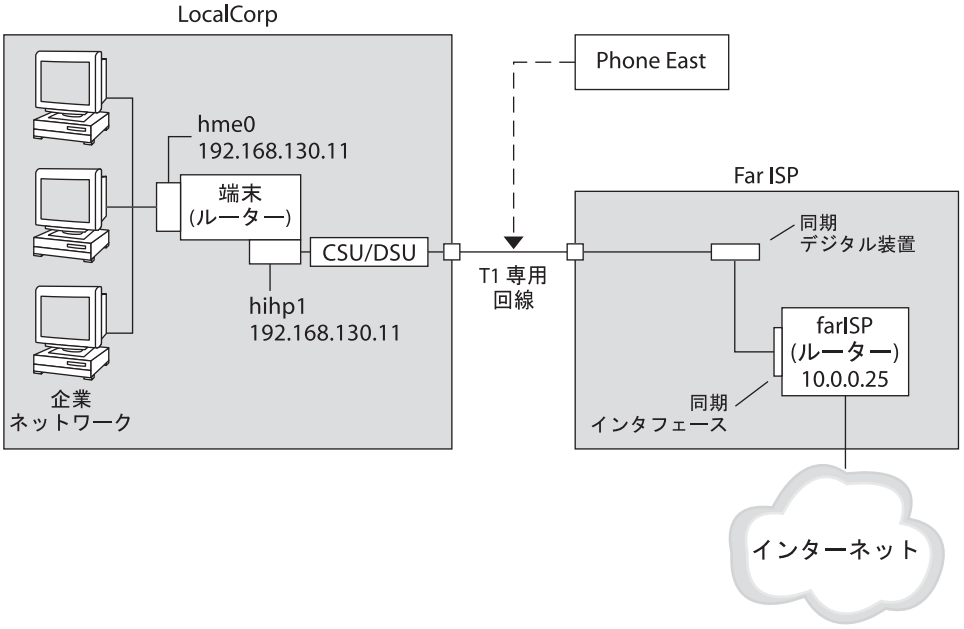
第 18 章「専用回線 PPP リンクの設定 (手順)」の作業は、中規模会社 (LocalCorp 社) で従業員がインターネットにアクセスできるように、専用回線リンクの構成を実装する方法を示しています。現在、従業員のコンピュータは、会社の私設イントラネットに接続されています。

LocalCorp 社では、高速なトランザクションとイントラネット上の多くの資源に迅速にアクセスすることが必要となっています。LocalCorp 社は、サービスプロバイダの Far ISP 社との間に専用回線を設定する契約を結びます。これにより、LocalCorp 社は電話会社の Phone East 社から T1 回線をリースします。Phone East 社は LocalCorp 社と Far ISP 社との間に専用回線を設置します。その後 Phone East 社は LocalCorp 社に構成済みの CSU/DSU を提供します。

作業では、次のような専用回線リンクを設定します。

- LocalCorp 社はシステムをゲートウェイルーターとして設定する。これにより、パケットは専用回線を介してインターネット上のホストに転送される。
- Far ISP 社でも顧客からの専用回線を接続するルーターとしてピアを設定する。

図 16–2 専用回線の構成例



この図では、LocalCorp 社側の PPP にルーターが設定されています。ルーターは、`hme0` インタフェースを介して社内イントラネットに接続されています。さらにマシンは、HSI/P インタフェース (hihp1) を介して CSU/DSU デジタル装置に接続されています。CSU/DSU は設置された専用回線に接続しています。LocalCorp 社の管理者が HSI/P インタフェースと PPP ファイルの構成を終了したあとで、`/etc/init.d/pppd` と入力すると、LocalCorp 社と Far ISP 社間でリンクが開始されます。

専用回線の詳細情報

次を参照してください。

- 第 18 章「専用回線 PPP リンクの設定 (手順)」
- 417 ページの「専用回線 PPP の概要」

リンクへの認証計画

この節では、PPP リンク上で認証を行うための計画情報を提供します。第 19 章「PPP 認証の設定 (手順)」は、自分のサイトで PPP 認証を実装するための作業を示しています。

PPP には、PAP と CHAP の 2 種類の認証があります。PAP の詳細は、528 ページの「パスワード認証プロトコル (PAP)」を参照してください。CHAP の詳細は、531 ページの「チャレンジハンドシェーク認証プロトコル (CHAP)」を参照してください。

認証をリンクに設定する前に、自分のサイトのセキュリティーポリシーに最適な認証プロトコルを選択する必要があります。認証プロトコルの選択が終了したら、ダイアルインマシンまたは呼び出し側のダイアルアウトマシンあるいは両方のマシンに秘密ファイルと PPP 構成ファイルを設定します。自分のサイトに最適な認証プロトコルを選択するには、421 ページの「PPP 認証を使用する理由」を参照してください。

この節では、次の内容について説明します。

- PAP 認証と CHAP 認証の計画情報
- 図 16–3 と図 16–4 に示されている認証事例の説明

認証の設定作業については、第 19 章「PPP 認証の設定 (手順)」を参照してください。

PPP 認証を設定する前に

自分のサイトで認証を設定することを、全体的な PPP 計画の必須部分として組み込む必要があります。認証を実装する前に、ハードウェアの組み立てや、ソフトウェアの構成、リンクの動作確認を行なってください。

表 16–5　認証構成の前提条件

情報	参照先
ダイアルアップリンクの構成作業	第 17 章「ダイアルアップ PPP リンクの設定 (手順)」
リンクのテスト作業	第 21 章「一般的な PPP 問題の解決 (手順)」
サイトのセキュリティー要件	会社のセキュリティーポリシー。ポリシーを設定していなければ、PPP 認証の設定を機にセキュリティーポリシーを設定します。
自分のサイトに PAP または CHAP を選択する場合のヒント	421 ページの「PPP 認証を使用する理由」。これらのプロトコルについては、528 ページの「接続時の呼び出し元の認証」を参照してください。

PPP の認証構成例

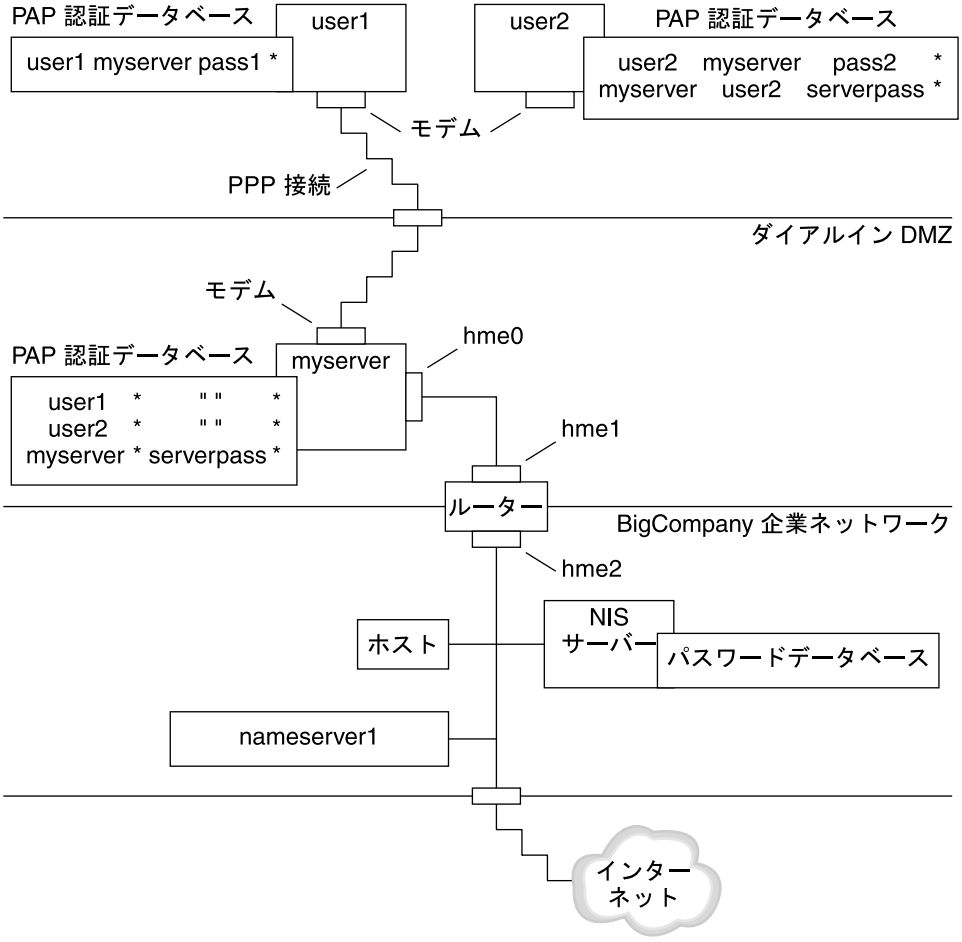
この節では、第 19 章「PPP 認証の設定 (手順)」の手順で使用されている認証事例について説明します。

PAP 認証による構成例

464 ページの「PAP 認証の設定」での作業は、PPP リンク上で PAP 認証を設定する方法を示しています。手順では、427 ページの「ダイアルアップ PPP の構成例」で紹介した架空の Big Company 社の PAP 事例を使用します。

Big Company 社では、自社のユーザーが自宅で仕事できるようにしたいと考えています。システム管理者は、ダイアルインサーバーに接続するシリアル回線にセキュリティー対策をしたいと考えています。NIS パスワードデータベースを使用する UNIX スタイルのログインは、これまで Big Company 社のネットワークで問題なく機能を果たしてきました。システム管理者は、PPP リンクを介してネットワークに進入してくる呼び出しに UNIX スタイルの認証機構を設定したいと考えています。その結果、システム管理者は PAP 認証を使用する次のシナリオを実装します。

図 16–3　PAP 認証のシナリオ (自宅で仕事する) の例



システム管理者は専用のダイアルイン DMZ を作成します。これは、ルーターによって会社のネットワークの後方部と分離されています。DMZ という用語は、軍事用語の「非武装地帯」に由来しています。DMZ はセキュリティー目的のために分離されたネットワークです。通常、DMZ には、Web サーバー、匿名 (anonymous) ftp サーバー、データベース、モデムサーバーなど、会社が一般に公開する資源が含まれています。ネットワーク設計者は通常、DMZ をファイアウォールと会社のインターネット接続の中間に設置します。

図 16–3 に示すように、DMZ に存在するのは、ダイアルインサーバーの `myserver` とルーターだけです。ダイアルインサーバーはリンクの設定時に、呼び出し側に PAP 資格 (ユーザー名とパスワードを含む) の提出を要求します。さらに、ダイアルインサーバーは PAP の `login` オプションも使用します。したがって、呼び出し側の PAP

のユーザー名とパスワードは、ダイアルインサーバーのパスワードデータベースにある UNIX のユーザー名とパスワードに正確に一致する必要があります。

PPP リンクが設定されたら、呼び出し側のパケットはルーターに転送されます。ルーターはパケットを会社のネットワーク上かインターネット上の宛先に転送します。

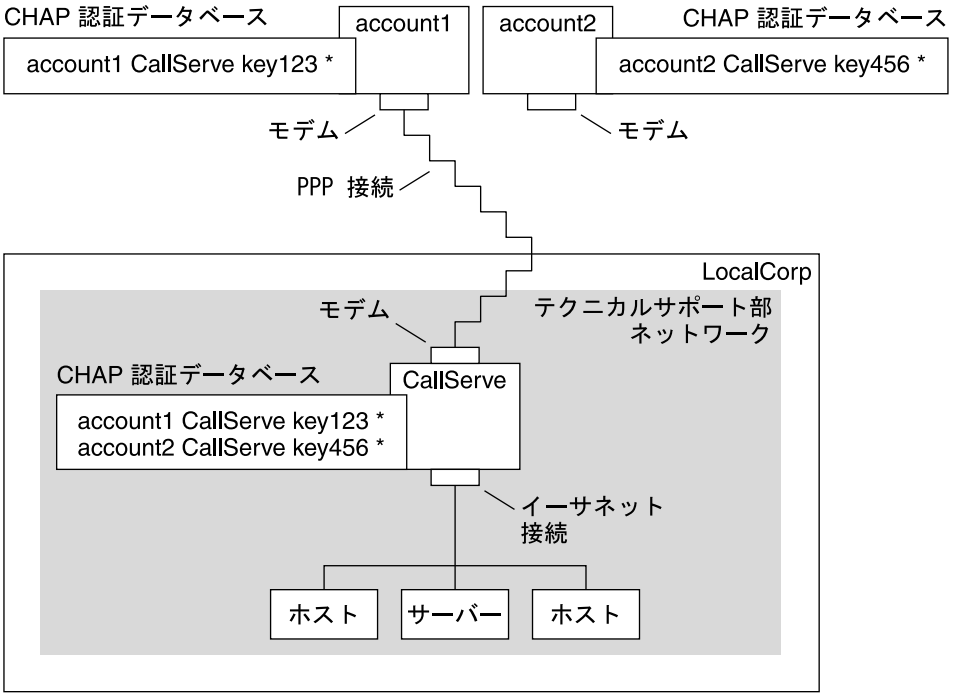
CHAP 認証による構成例

472 ページの「CHAP 認証の設定」での作業は、CHAP 認証の設定方法を示しています。手順では、430 ページの「専用回線リンクの構成例」で紹介した架空の LocalCorp 社の CHAP 事例を使用します。

LocalCorp 社は、ISP の専用回線を介してインターネットに接続できます。LocalCorp 社のテクニカルサポート部では、大量のネットワークトラフィックが発生するので、独立した私設ネットワークが必要になっています。部署のフィールドエンジニアは、問題解決のための情報を入手するために遠隔地からテクニカルサポートのネットワークに頻繁にアクセスする必要があります。私設ネットワークのデータベース内の機密情報を保護するには、リモートでの呼び出し側にログインの許可を与えるために、それらを認証する必要があります。

したがって、システム管理者は、ダイアルアップ PPP 構成に次の CHAP 認証シナリオを実装します。

図 16–4 CHAP 認証シナリオ (私設ネットワークを呼び出す) の例



テクニカルサポート部のネットワークから外部世界にリンクするのは、リンクのダイアルインサーバー側の端に接続しているシリアル回線だけです。システム管理者は、各フィールドサービスエンジニアが所持する PPP 用ラップトップコンピュータを CHAP シークレットなどを組み込んだ CHAP セキュリティーで構成します。ダイアルインサーバー上の CHAP シークレットデータベースには、テクニカルサポート内のネットワークに対する呼び出しが許されているすべてのマシンの CHAP 資格が含まれています。

認証の詳細情報

次を参照してください。

- 464 ページの「PAP 認証の設定」
- 472 ページの「CHAP 認証の設定」
- 528 ページの「接続時の呼び出し元の認証」 と pppd(1M) のマニュアルページ

PPPoE トンネルを介した DSL サポートの計画

一部の DSL プロバイダは、プロバイダの DSL 回線と高速のデジタルネットワーク上で PPP を実行するために、ユーザーのサイトに PPPoE トンネルを設定するように要求しています。PPPoE の概要については、421 ページの「PPPoE による DSL ユーザーのサポート」を参照してください。

PPPoE トンネルには、3 つの関係者が存在しています。 消費者、電話会社、および ISP です。PPPoE は、消費者 (会社の PPPoE クライアントや自宅の消費者など) 向けか ISP 側のサーバー上のどちらかに構成します。

この節では、クライアントとアクセスサーバーの両方で PPPoE を実行するための計画情報について説明します。次の項目について説明します。

- PPPoE ホストとアクセスサーバーの計画情報
- 439 ページの「PPPoE トンネルの構成例」で紹介されている PPPoE シナリオの説明

PPPoE トンネルの設定作業については、第 20 章「PPPoE トンネルの設定 (手順)」を参照してください。

PPPoE トンネルを設定する前に

構成前の作業は、トンネルをクライアント側に構成するかサーバー側に構成するかによって異なります。どちらの場合も、電話会社と契約を結ぶ必要があります。電話会社では、クライアントには DSL 回線を提供し、アクセスサーバーにはある形式のブリッジと ATM パイプを提供します。ほとんどの契約では、電話会社はユーザーのサイトに機器を設置します。

PPPoE クライアントを構成する前に

PPPoE クライアントの実装は、通常、次の機器から構成されます。

- 個人が使用するパーソナルコンピュータまたはシステム
- DSL モデム。通常は、電話会社かインターネットのアクセスプロバイダが設置する
- (任意) ハブ。複数のクライアントが関係するような会社の DSL 消費者向け
- (任意) スプリッタ。通常はプロバイダが設置する

多くの異なる DSL 構成が可能です。DSL 構成は、ユーザーや会社のニーズ、プロバイダが提供するサービスによって異なります。

表 16–6 PPPoE クライアントの計画

情報	動作
個人や自分自身のために自宅の PPPoE クライアントを設定する場合に、PPPoE の領域外の設定情報を入手します。	設定の手続きが必要なら、電話会社や ISP に問い合わせます。
会社のサイトに PPPoE クライアントを設定する場合に、PPPoE クライアントシステムが割り当てられているユーザーの名前を収集します。PPPoE リモートクライアントを構成する場合は、DSL 機器を自宅に設置するための情報をユーザーに提供する必要があります。	認可されたユーザーのリストを会社の管理者に問い合わせます。
PPPoE クライアント上で使用できるインタフェースを探します。	各マシン上で `ifconfig -a` コマンドを実行し、インタフェース名を探します。
(任意) PPPoE クライアントのパスワードを入手します。	ユーザーに、希望のパスワードを問い合わせます。または、ユーザーにパスワードを割り当てます。このパスワードは UNIX のログイン用ではなく、リンクの認証用に使用します。

PPPoE サーバーを構成する前に

PPPoE アクセスサーバーの計画は、データサービスネットワークへの接続を提供する電話会社と共同で行います。電話会社はユーザーのサイトに回線 (通常は ATM パイプ) を設置し、ユーザーのアクセスサーバーに、ある形式のブリッジを提供します。会社が提供するサービスにアクセスする Ethernet インタフェースを構成する必要があります。たとえば、インターネットにアクセスするためのインタフェースのほか、電話会社のブリッジが提供する Ethernet インタフェースも構成します。

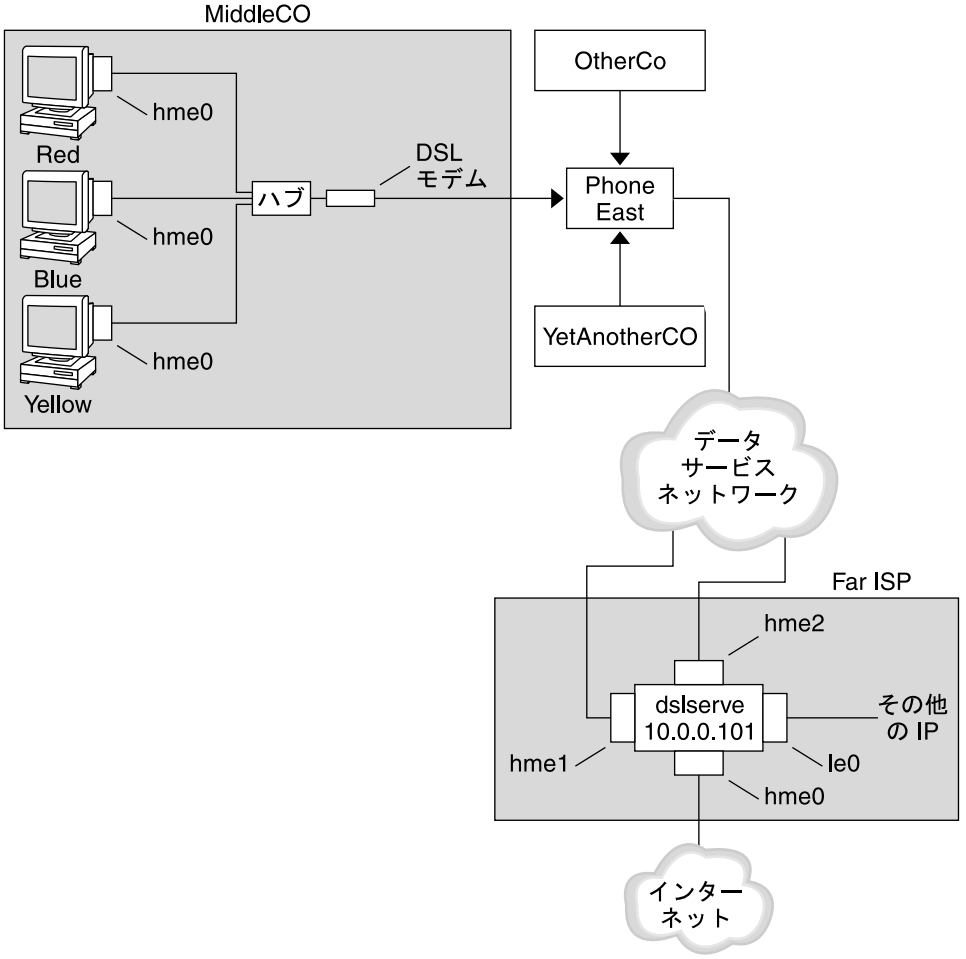
表 16–7 PPPoE アクセスサーバーの計画

情報	動作
データサービスネットワークの回線に使用するインタフェース	`ifconfig -a` コマンドを実行して、インタフェースを特定します。
PPPoE サーバーが提供するサービスの種類	管理者やネットワーク計画者に要件やヒントを問い合わせます。
(任意) 消費者に提供するサービスの種類	管理者やネットワーク計画者に要件やヒントを問い合わせます。
(任意) リモートクライアントのホスト名とパスワード	ネットワーク計画者や契約交渉の担当者に問い合わせます。ホスト名とパスワードは UNIX のログインではなく、PAP 認証や CHAP 認証に使用します。

PPPoE トンネルの構成例

この節では、第 20 章「PPPoE トンネルの設定 (手順)」で説明する作業の例として、PPPoE トンネルの例を示します。図では、トンネル内のすべてのパーティシパントを示していますが、ユーザーはクライアント側かサーバー側のどちらかの端を管理するだけです。

図 16–5　PPPoE トンネルの例



この例では、MiddleCo 社は従業員に高速なインターネットアクセスを提供することを望んでいます。MiddleCo 社は Phone East 社から DSL パッケージを購入し、Phone

East 社はサービスプロバイダの Far ISP 社と契約を結びます。Far ISP 社は、Phone East 社から DSL を購入する顧客にインターネットサービスや IP サービスを提供します。

PPPoE クライアントの構成例

MiddleCo 社は、サイトに DSL の 1 回線を提供する Phone East 社からパッケージを購入します。パッケージには、MiddleCo 社の PPPoE クライアント用に認証された ISP への専用接続が含まれています。システム管理者は予想される PPPoE クライアントをハブに配線します。Phone East 社の技術者はハブを DSL 機器に配線します。

PPPoE サーバーの構成例

FarISP 社では、Phone East 社との契約を履行するために、同社のシステム管理者がアクセスサーバー (`dslserve`) を構成します。このサーバーには、次の 4 つのインタフェースがあります。

- `eri0` – ローカルネットワークと接続する主要なネットワークインタフェース
- `hme0` – FarISP 社が顧客にインターネットサービスを提供するためのインタフェース
- `hme1` – 認証された PPPoE トンネル用に MiddleCo 社が使用するインタフェース
- `hme2` – PPPoE トンネル用に別の顧客が使用するインタフェース

PPPoE の詳細情報

次を参照してください。

- 480 ページの「PPPoE クライアントの設定」
- 483 ページの「PPPoE アクセスサーバーの設定」
- 536 ページの「DSL サポート用の PPPoE トンネルの作成」および `pppoed`(1M)、`pppoec`(1M)、`sppptun`(1M) のマニュアルページ

◆ ◆ ◆ 第 17 章

17 ダイアルアップ PPP リンクの設定 (手順)

この章では、もっとも一般的な PPP リンクであるダイアルアップリンクの構成作業について説明します。ここでは、次の内容を説明します。

ダイアルアップの PPP リンクを設定する主な作業 (作業マップ)

ダイアルアップ PPP の設定は、モデムの構成、ネットワークデータベースファイルの変更、および表 22–1 で説明している PPP 構成ファイルの変更によって行います。

次の表は、ダイアルアップ PPP リンクの両側を構成するための主な作業を示しています。通常は、リンクのどちらか一方 (ダイアルアウトマシンかダイアルインサーバー) だけを構成します。

表 17–1 ダイアルアップの PPP リンクの設定 (作業マップ)

作業	説明	参照先
1. 構成前の情報を収集する	リンクを設定する前に、ピアのホスト名、ターゲットの電話番号、モデムの速度など必要なデータを集める	426 ページの「ダイアルアップ PPP リンクの計画」
2. ダイアルアウトマシンを構成する	リンクを介して呼び出しを行うマシンに PPP を設定する	表 17–2
3. ダイアルインサーバーを構成する	着呼を受信するマシンに PPP を設定する	表 17–3

表 17–1 ダイアルアップの PPP リンクの設定 (作業マップ) (続き)

作業	説明	参照先
4. ダイアルイン サーバーを呼び出す	pppd コマンドを入力して、通信を開始する	454 ページの「ダイアルインサーバーの呼び出し方法」

ダイアルアウトマシンの構成

この節の作業では、ダイアルアウトマシンの構成方法について説明します。この作業では、図 16–1 で紹介した自宅からのダイアルイン事例を使用します。予想されるユーザーにマシンを渡す前に、会社での作業があります。経験豊富なユーザーであれば、自宅のマシンの設定を指導することもできます。ダイアルアウトマシンを設定する人は必ずそのマシンのスーパーユーザー権限を持つ必要があります。

ダイアルアウトマシンの構成作業 (作業マップ)

表 17–2 ダイアルアウトマシンの設定 (作業マップ)

作業	説明	参照先
1. 構成前の情報を収集する	リンクを設定する前に、ピアのホスト名、ターゲットの電話番号、モデムの速度など必要なデータを集める	426 ページの「ダイアルアップ PPP リンクの計画」
2. モデムとシリアルポートを構成する	モデムとシリアルポートを設定する	443 ページの「モデムとシリアルポートの構成方法 (ダイアルアウトマシン)」
3. シリアル回線通信を構成する	シリアル回線上の伝送特性を構成する	445 ページの「シリアル回線を介した通信を定義する方法」
4. ダイアルアウトマシンとピア間の対話を定義する	chat スクリプトを作成するときに使用する通信データを収集する	446 ページの「ピアを呼び出すための命令群を作成する方法」
5. 特定のピア情報を構成する	個々のダイアルインサーバーを呼び出すための PPP オプションを構成する	447 ページの「個々のピアとの接続を定義する方法」
6. ピアを呼び出す	pppd コマンドを入力して、通信を開始する	454 ページの「ダイアルインサーバーの呼び出し方法」

ダイアルアップ PPP のテンプレートファイル

Solaris PPP 4.0 はテンプレートファイルを提供します。各テンプレートファイルには、特定の PPP 構成ファイルのために一般的なオプションが含まれています。次の表は、ダイアルアップリンクの設定に使用できるテンプレートのサンプルと、それらと同等の Solaris PPP 4.0 ファイルを示します。

テンプレートファイル	PPP 構成ファイル	参照先
/etc/ppp/options.tmpl	/etc/ppp/options	510 ページの「/etc/ppp/options.tmpl テンプレート」
/etc/ppp/options.ttya.tmpl	/etc/ppp/options.*ttyname*	512 ページの「options.ttya.tmpl テンプレートファイル」
/etc/ppp/myisp-chat.tmpl	chat スクリプトを格納するための ユーザー指定の名前を持つファイル	520 ページ の「/etc/ppp/myisp-chat.tmpl chat スクリプトテンプレート」
/etc/ppp/peers/myisp.tmpl	/etc/ppp/peers/*peer-name*	516 ページ の「/etc/ppp/peers/myisp.tmpl テンプレートファイル」

テンプレートファイルを使用するように決めたら、そのテンプレートファイルの名前を同等の PPP 構成ファイルの名前に変更します。chat ファイルのテンプレート (/etc/ppp/myisp-chat.tmpl) だけは例外です。chat スクリプトには任意の名前を選択できます。

ダイアルアウトマシン上にデバイスを構成する

ダイアルアウト PPP マシンを設定するための最初の作業は、シリアル回線にデバイス (モデムとシリアルポート) を構成することです。

注 – モデムに適用する作業は、通常 ISDN TA にも適用します。

以降の手順を実行する前に、次の作業を終了しておく必要があります。

- Solaris 9 または Solaris 10 をダイアルアウトマシンにインストールする
- モデムの最適速度を決定する
- ダイアルアウトマシンに使用するシリアルポートを決定する
- ダイアルアウトマシンのルートパスワードを取得する

計画情報については、表 16–2 を参照してください。

▼ モデムとシリアルポートの構成方法 (ダイアルアウトマシン)

1 モデムの設定を行います。

さまざまなタイプのモデムを使用できますが、通常のモデムは Solaris PPP 4.0 用に正しく設定されて出荷されています。次は、Solaris PPP 4.0 を使用するモデムの基本的なパラメータ設定を示しています。

- **DCD** – キャリアの指示に従う
- **DTR** – モデムがハングアップするように Low に設定する (モデムをオンフックにする)
- **Flow Control** – 全二重ハードウェアのフロー制御用 RTS/CTS を設定する
- **Attention Sequences** – 使用不可

リンクの設定で問題が発生し、原因がモデムにあれば、まずモデムの製造元のマニュアルを参照します。また、多くの Web サイトが、役に立つモデムの設定情報を提供しています。最後に、496 ページの「モデムの問題を診断する方法」でモデム問題を解決するためのヒントを見つけることができます。

2 モデムケーブルをダイアルアウトマシンのシリアルポートと電話ジャックに接続します。

3 ダイアルアウトマシン上のスーパーユーザー、またはそれと同等の役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

4 **Setting Up Terminals and Modems With Serial Ports Tool (Overview)**『Solaris のシステム管理 (上級編)』の「シリアルポートツールによる端末とモデムの設定 (概要)」コマンドを実行します。このコマンドによって、**Solaris** 管理コンソールが開きます。

Solaris 管理コンソールを使用して、次を行います。

a. モデムを接続しているポートを選択します。

b. モデム方向を「発信専用」として指定します。

モデムを「発着信両用」としても設定できます。「発信専用」を選択すると、侵入者に対してセキュリティーが強力になります。

注 – `/usr/sadm/bin/smc` でボーレートやタイムアウトを設定できますが、`pppd` デーモンはこれらの設定を無視します。

---

5 「**OK**」をクリックして変更を有効にします。

ダイアルアウトマシン上に通信を構成する

この節の手順では、ダイアルアウトマシンのシリアル回線に通信を構成する方法を示します。これらの手順を使用する前に、443 ページの「モデムとシリアルポートの構成方法 (ダイアルアウトマシン)」で説明しているように、モデムとシリアルポートを設定しておく必要があります。

次の作業は、ダイアルアウトマシンがダイアルインサーバーとの通信を正常に開始できるようにする方法を示します。通信は、PPP 構成ファイルで定義されているオプションに基づいて開始されます。次のファイルを作成する必要があります。

- `/etc/ppp/options`
- `/etc/ppp/options.`*ttyname*
- chat スクリプト
- `/etc/ppp/peers/`*peer-name*

Solaris PPP 4.0 は、PPP 構成ファイルにテンプレートを提供します。これらのテンプレートは要求に合わせてカスタマイズできます。これらのファイルについては、442 ページの「ダイアルアップ PPP のテンプレートファイル」を参照してください。

▼ シリアル回線を介した通信を定義する方法

1. ダイアルアウトマシン上のスーパーユーザー、またはそれと同等の役割になります。

   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2. 次のオプションを指定して、`/etc/ppp/options` と呼ばれるファイルを作成します。

   `lock`

   `/etc/ppp/options` ファイルは、ローカルマシンが実行するすべての通信に適用されるグローバルパラメータの定義に使用されます。`lock` オプションによって、`/var/spool/locks/LK.xxx.yyy.zzz` 形式の UUCP スタイルのロックが可能です。

   注 – ダイアルアウトマシンが `/etc/ppp/options` ファイルを持たない場合は、スーパーユーザーだけが `pppd` コマンドを実行できます。ただし、`/etc/ppp/options` は空でもかまいません。

   `/etc/ppp/options` については、509 ページの「`/etc/ppp/options` 構成ファイル」を参照してください。

3. (省略可能) 特定のシリアルポートから通信を起動する方法を定義するために、`/etc/ppp/options.`*ttyname* と呼ばれるファイルを作成します。

   次の例は、デバイス名として `/dev/cua/a` を持つポートの `/etc/ppp/options.`*ttyname* ファイルを示しています。

   ```
   # cat /etc/ppp/options.cua.a
   crtscts
   ```

PPP オプション crtscts は、pppd デーモンに、シリアルポート a のハードウェアフロー制御をオンにするように指示します。

/etc/ppp/options.ttyname ファイルについては、511 ページの「 /etc/ppp/options.*ttyname* 構成ファイル」を参照してください。

4 モデム速度を**450** ページの「モデム速度を設定する方法」で説明しているとおりに設定します。

▼ ピアを呼び出すための命令群を作成する方法

ダイアルアウトマシンが PPP リンクを開始する前に、ピアになるダイアルインサーバーの情報を収集する必要があります。情報を収集したら、この情報を使用して chat スクリプトを作成します。chat スクリプトには、ダイアルアウトマシンとピア間の実際の対話を記述します。

1 ダイアルアウトマシンのモデムの実行速度を決定します。

詳細は、517 ページの「ダイアルアップリンクのモデム速度の設定」を参照してください。

2 ダイアルインサーバーのサイトから次の情報を入手します。

- サーバーの電話番号
- 必要な場合、使用している認証プロトコル
- chat スクリプトでピアが必要とするログインシーケンス

3 ダイアルインサーバーサイトのネームサーバーの名前と **IP** アドレスを入手します。

4 **chat** スクリプトに、特定ピアへの呼び出しを開始するための命令群を指定します。

たとえば、次の chat スクリプト (/etc/ppp/mychat) を作成して、ダイアルインサーバー (myserver) を呼び出します。

```
SAY "Calling the peer\n"
        TIMEOUT 10
        ABORT BUSY
        ABORT 'NO CARRIER'
        ABORT ERROR
        REPORT CONNECT
        "" AT&F1&M5S2=255
        TIMEOUT 60
        OK ATDT1-123-555-1234
        CONNECT \c
        SAY "Connected; logging in.\n"
        TIMEOUT 5
        ogin:--ogin: pppuser
        TIMEOUT 20
        ABORT 'ogin incorrect'
        ssword: \qmypassword
```

```
"% " \c
SAY "Logged in.  Starting PPP on peer system.\n"
ABORT 'not found'
"" "exec pppd"
~ \c
```

スクリプトには、ログインシーケンスを必要とする Solaris ダイアルインサーバーを呼び出すための命令群が含まれています。各命令については、522 ページの「UNIX 方式ログイン用に拡張された基本の chat スクリプト」を参照してください。chat スクリプトの作成については、518 ページの「ダイアルアップリンクでの会話の定義」を参照してください。

注 – chat スクリプトを直接呼び出さないでください。chat コマンドの引数に chat スクリプトのファイル名を指定して、スクリプトを呼び出します。

ピアが Solaris または類似のオペレーティングシステムを実行する場合は、ダイアルアウトマシンのテンプレートとして前述の chat スクリプトの利用をお薦めします。

▼ 個々のピアとの接続を定義する方法

1 ダイアルアウトマシン上のスーパーユーザー、またはそれと同等の役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 次の `/etc/resolv.conf` ファイルを作成して、**DNS** データベースを更新します。

```
domain bigcompany.com
nameserver 10.10.111.15
nameserver 10.10.130.8
```

`domain bigcompany.com`
ピアの DNS ドメインが `bigcompany.com` であることを示す

`nameserver 10.10.111.15` および **`nameserver 10.10.130.8`**
`bigcompany.com` 側にあるネームサーバーの IP アドレスの一覧を示す

3 ホスト情報として最初に **DNS** データベースが検索されるように、`/etc/nsswitch.conf` ファイルを編集します。

```
hosts:      dns [NOTFOUND=return] files
```

4 ピア用のファイルを作成します。

たとえば、次のファイルを作成して、ダイアルインサーバー (myserver) を定義します。

```
# cat /etc/ppp/peers/myserver
/dev/cua/a
57600
noipdefault
defaultroute
idle 120
noauth
connect "chat -U 'mypassword' -T 1-123-555-1213 -f /etc/ppp/mychat"
```

/dev/cua/a
: myserver を呼び出すためのシリアルインタフェースとして、デバイス (/dev/cua/a) を使用する必要があることを示す

57600
: リンクの速度を定義する

noipdefault
: ピア (myserver) のトランザクションでは、ダイアルアウトマシンは最初に 0.0.0.0 の IP アドレスを持つことを示す。myserver は、すべてのダイアルアップセッションのダイアルアウトマシンに IP アドレスを割り当てる

idle 120
: 120 秒のアイドル時間が経過するとリンクがタイムアウトになることを示す

noauth
: ダイアルアウトマシンとの接続をネゴシエートするとき、ピア (myserver) は認証資格を提供する必要がないことを示す

connect "chat -U 'mypassword' -T 1-123-555-1213 -f /etc/ppp/mychat"
: connect オプションとその引数を示す。引数には、ピアの電話番号、呼び出しの命令群を持つ chat スクリプト (/etc/ppp/mychat) などが指定されている

参照 関連情報の参照先は次のとおりです。

- 別のダイアルアウトマシンを構成する手順については、443 ページの「モデムとシリアルポートの構成方法 (ダイアルアウトマシン)」を参照
- 別のコンピュータにダイアルアウトすることでモデムの接続性をテストする手順については、cu(1C) と tip(1) のマニュアルページを参照。これらのユーティリティーを使用すると、モデムが正しく構成されているかをテストできる。また、別のマシンとの接続が確立できるかもテストできる
- 構成ファイルとオプションの詳細については、505 ページの「ファイルおよびコマンド行での PPP オプションの使用」を参照
- ダイアルインサーバーの構成手順については、449 ページの「ダイアルインサーバーにデバイスを構成する」を参照

ダイアルインサーバーの構成

この節の作業では、ダイアルインサーバーを構成します。ダイアルインサーバーは、ダイアルアウトマシンからの呼び出しを PPP リンクを介して受信するピアマシンです。作業では、図 16–1 で紹介したダイアルインサーバー (myserver) の構成方法を示します。

ダイアルインサーバーの構成作業 (作業マップ)

表 17–3 ダイアルインサーバーの設定 (作業マップ)

作業	説明	参照先
1. 構成前の情報を収集する	リンクを設定する前に、ピアのホスト名、ターゲットの電話番号、モデムの速度など必要なデータを集める	426 ページの「ダイアルアップ PPP リンクの計画」
2. モデムとシリアルポートを構成する	モデムとシリアルポートを設定する	450 ページの「モデムとシリアルポートの構成方法 (ダイアルインサーバー)」
3. ピア情報の呼び出しを構成する	ダイアルインサーバーへの呼び出しが許可されているすべてのダイアルアウトマシンにユーザー環境と PPP オプションを設定する	451 ページの「ダイアルインサーバーのユーザーを構成する方法」
4. シリアル回線通信を構成する	シリアル回線上の伝送特性を構成する	453 ページの「シリアル回線を介した通信を定義する方法 (ダイアルインサーバー)」

ダイアルインサーバーにデバイスを構成する

次の手順では、モデムとシリアルポートをダイアルインサーバーに構成する方法について説明します。

手順を実行する前に、ピアであるダイアルインサーバー上で次の作業を終了しておく必要があります。

- Solaris 9 または Solaris 10 のインストール
- モデムの最適速度を決定する
- 使用するシリアルポートの決定

▼ モデムとシリアルポートの構成方法 (ダイアルインサーバー)

1. モデムの製造元が発行するマニュアルに従ってモデムのプログラムを作成します。
   詳細は、443 ページの「モデムとシリアルポートの構成方法 (ダイアルアウトマシン)」を参照してください。
2. モデムをダイアルインサーバー上のシリアルポートに接続します。
3. ダイアルインサーバー上のスーパーユーザー、またはそれと同等の役割になります。
   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。
4. 『Solaris のシステム管理 (上級編)』の「シリアルポートツールによる端末とモデムの設定 (概要)」で説明しているように、**Solaris** 管理コンソールの `/usr/sadm/bin/smc` コマンドを使ってシリアルポートを構成します。
   Solaris 管理コンソールを使用して、次を行います。
   a. モデムを接続しているシリアルポートを選択します。
   b. モデム方向を「着信専用」として指定します。

      注 – Solaris PPP 4.0 は、モデムに対して双方向通信をサポートしています。

   c. 「**OK**」をクリックして変更を有効にします。

▼ モデム速度を設定する方法

次の手順では、ダイアルインサーバーのモデム速度を設定する方法について説明します。Sun Microsystems のコンピュータを使用する際のモデム速度については、517 ページの「ダイアルアップリンクのモデム速度の設定」を参照してください。

1. ダイアルインサーバーにログインします。
2. `tip` コマンドを使用して、モデムにアクセスします。
   `tip` によるモデム速度の設定については、`tip(1)` のマニュアルページを参照してください。
3. 固定 **DTE** レートでモデムを構成します。

4 『Solaris のシステム管理 (上級編)』の「シリアルポートツールによる端末とモデムの設定 (概要)」で説明しているように、ttymon または /usr/sadm/bin/smc を使ってシリアルポートをそのレートで固定します。

参照 関連情報の参照先は次のとおりです。

- 450 ページの「モデムとシリアルポートの構成方法 (ダイアルインサーバー)」
- 451 ページの「ダイアルインサーバーのユーザーを構成する方法」

ダイアルインサーバーのユーザーを設定する

ダイアルインサーバーの設定プロセスでは、既知の各リモート呼び出し側に関する情報を構成する必要があります。

この節の手順を開始する前に、次の作業を終了しておく必要があります。

- リモートダイアルアウトマシンからログインが許されているすべてのユーザーの UNIX ユーザー名を入手する
- 450 ページの「モデムとシリアルポートの構成方法 (ダイアルインサーバー)」で説明しているとおりに、モデムとシリアル回線を設定する
- IP アドレスを専用化して、リモートユーザーからの着呼に割り当てる。呼び出し側の数がダイアルインサーバー上のモデムとシリアルポートの数を超える可能性がある場合、着呼専用の IP アドレスの作成を検討する。専用 IP アドレスについては、534 ページの「呼び出し元の IP アドレス指定スキーマの作成」を参照

▼ ダイアルインサーバーのユーザーを構成する方法

1 ダイアルインサーバー上のスーパーユーザー、またはそれと同等の役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 各リモート PPP ユーザーに対して、ダイアルインサーバー上で新しいアカウントを作成します。

Solaris 管理コンソールを使用して、新しいユーザーを作成できます。/usr/sadm/bin/smc コマンドによって、Solaris 管理コンソールが開きます。Solaris 管理コンソールを使って新しいユーザーを作成するには、『Solaris のシステム管理 (基本編)』の「ユーザーアカウントの設定 (作業マップ)」を参照してください。

3 **Solaris 管理コンソールを使用して、新しいユーザーにパラメータを割り当てます。**

たとえば、次の表は、ダイアルアウトマシン (myhome) 上の user1 に対する pppuser と呼ばれるアカウントのパラメータを示しています。

パラメータ	値	定義
ユーザー名	pppuser	リモートユーザーのユーザーアカウント名。このアカウント名は、chat スクリプトのログインシーケンスで指定されているアカウント名と一致する必要がある。たとえば、pppuser は、446 ページの「ピアを呼び出すための命令群を作成する方法」 の chat スクリプトにあるアカウント名である
ログインシェル	/usr/bin/pppd	リモートユーザーのデフォルトのログインシェル。ログインシェル (/usr/bin/pppd) は最初から呼び出し側を専用 PPP 環境に制限する
「ホームディレクトリの作成」のパス	/export/home/pppuser	ホームディレクトリ (/export/home/pppuser) は、呼び出し側が正常にダイアルインサーバーにログインするとき設定される

4 **各呼び出し側に対して、$HOME/.ppprc ファイルを作成します。このファイルには、ユーザーの PPP セッションに固有のさまざまなオプションが格納されています。**

たとえば、pppuser に対して、次の .ppprc ファイルを作成します。

```
# cat /export/home/pppuser/.ppprc
noccp
```

noccp は、リンク上で圧縮制御をオフにします。

参照 関連情報の参照先は次のとおりです。

- 451 ページの「ダイアルインサーバーのユーザーを構成する方法」
- 453 ページの「シリアル回線を介した通信を定義する方法 (ダイアルインサーバー)」

ダイアルインサーバーを介した通信を構成する

次の作業は、ダイアルインサーバーが任意のダイアルアウトマシンと通信を開始できるようにする方法を示します。通信がどのように確立されるかは、次の PPP 構成ファイルで定義されているオプションに基づいて決まります。

- /etc/ppp/options
- /etc/ppp/options.*ttyname*

これらのファイルについては、505 ページの「ファイルおよびコマンド行での PPP オプションの使用」を参照してください。

先に進む前に、次の作業を終了しておく必要があります。

- 450 ページの「モデムとシリアルポートの構成方法 (ダイアルインサーバー)」で説明しているとおりに、ダイアルインサーバーにシリアルポートとモデムを構成する
- 451 ページの「ダイアルインサーバーのユーザーを構成する方法」で説明しているとおりに、ダイアルインサーバーの予想されるユーザー情報を構成する

▼ シリアル回線を介した通信を定義する方法 (ダイアルインサーバー)

1 ダイアルインサーバー上のスーパーユーザー、またはそれと同等の役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 次の引数を指定して、`/etc/ppp/options` ファイルを作成します。

`nodefaultroute`

`nodefaultroute` は、ローカルシステム上の `pppd` セッションが、`root` 権限がないとデフォルトの経路を確立できないことを示します。

注 – ダイアルインサーバーが `/etc/ppp/options` ファイルを持たない場合は、スーパーユーザーだけが `pppd` コマンドを実行できます。ただし、`/etc/ppp/options` ファイルは空でもかまいません。

---

3 `/etc/options.`*ttyname* ファイルを作成して、シリアルポート (*ttyname*) を介して受信される呼び出しの制御方法を定義します。

次の `/etc/options.ttya` ファイルでは、ダイアルインサーバーのシリアルポート (`/dev/ttya`) が着呼を制御する方法を定義しています。

```
:10.0.0.80
xonxoff
```

`:10.0.0.80` シリアルポート (`ttya`) を介して呼び出しているすべてのピアに IP アドレス (10.0.0.80) を割り当てる

`xonxoff` ソフトウェアのフロー制御を有効にすることで、シリアル回線はモデムからの通信を制御できる

参照　この章のすべての手順を実行すると、ダイアルアップリンクの構成が完成します。関連情報の参照先は次のとおりです。

- 別のコンピュータにダイアルアウトすることでモデムの接続性をテストする手順については、cu(1C) と tip(1) のマニュアルページを参照。これらのユーティリティーを使用すると、モデムが正しく構成されているかをテストできる。また、別のマシンとの接続が確立できるかもテストできる
- ダイアルインサーバーのオプションを追加して構成する手順については、449 ページの「ダイアルインサーバーの構成」
- ダイアルアウトマシンを追加して構成する手順については、442 ページの「ダイアルアウトマシンの構成」
- リモートマシンがダイアルインサーバーを呼び出す手順については、454 ページの「ダイアルインサーバーの呼び出し」

ダイアルインサーバーの呼び出し

ダイアルアウトマシンがダイアルインサーバーを呼び出すことで、ダイアルアップ PPP リンクを確立します。ローカルの PPP 構成ファイルに demand オプションを指定することで、ダイアルアウトマシンがサーバーを呼び出すように指示できます。リンクの確立でもっとも一般的な方法は、ユーザーがダイアルアウトマシン上で pppd コマンドを実行することです。

次の作業に進む前に、次のどちらかの作業か両方の作業を終了しておく必要があります。

- 442 ページの「ダイアルアウトマシンの構成」で説明しているとおりに、ダイアルアウトマシンを設定する
- 449 ページの「ダイアルインサーバーの構成」で説明しているとおりに、ダイアルインサーバーを設定する

▼ ダイアルインサーバーの呼び出し方法

1. root ではなく、通常のユーザーアカウントを使用して、ダイアルアウトマシンにログインします。

2. pppd コマンドを実行して、ダイアルインサーバーを呼び出します。

   たとえば、次のコマンドは、ダイアルアウトマシンとダイアルインサーバー (myserver) 間のリンクを開始します。

   ```
   % pppd 57600 call myserver
   ```

   pppd　　pppd デーモンを呼び出すことで呼び出しを開始する

57600	ホストとモデム間の回線速度を設定する
call myserver	pppd の call オプションを呼び出して、447 ページの「個々のピアとの接続を定義する方法」で作成された /etc/ppp/peers/myserver ファイルのオプション群を読み取る

3 サーバーのネットワーク上にあるホスト (図 16–1 に示されている lindyhop ホストなど) にアクセスします。

```
ping lindyhop
```

リンクが正しく動作しない場合、第 21 章「一般的な PPP 問題の解決 (手順)」を参照してください。

4 PPP セッションを終了します。

```
% pkill -x pppd
```

参照 この章のすべての手順を実行すると、ダイアルアップリンクの構成が完成します。関連情報の参照先は次のとおりです。

- ユーザーがダイアルアウトマシン上で作業を開始する手順については、454 ページの「ダイアルインサーバーの呼び出し方法」
- リンク上の問題を修正する手順については、第 21 章「一般的な PPP 問題の解決 (手順)」
- この章で使用するファイルとオプションについてさらに学習するときは、505 ページの「ファイルおよびコマンド行での PPP オプションの使用」

◆ ◆ ◆ 第 18 章

18 専用回線 PPP リンクの設定 (手順)

この章では、専用回線を使用した、ピア間での PPP リンクを設定する方法について説明します。主に次の内容について説明します。

専用回線の設定 (作業マップ)

専用回線リンクの設定は、ダイアルアップリンクのそれに比べて、比較的簡単です。ほとんどの場合、CSU/DSU、ダイアルサービス、または認証を設定する必要はありません。CSU/DSU の設定は複雑なので、これを設定する必要がある場合は、製造元のマニュアルを参照してください。

次の表の作業マップでは、基本的な専用回線リンクの設定に必要な作業について説明しています。

注 – 専用回線の中には、対するピアのアドレスを「ダイアル」するために、CSU/DSU を必要とするものもあります。たとえば、SVC (Switched Virtual Circuit) や Switched 56 サービスを使用するフレームリレーなどがあります。

表 18–1 専用回線リンクの設定 (作業マップ)

作業	説明	参照先
1. 構成前の情報を収集する	接続の設定に必要な情報を収集する	表 16–4
2. 専用回線への接続に使用するハードウェアを設定する	CSU/DSU および同期インタフェースカードを取り付ける	458 ページの「同期デバイスの設定方法」

表 18–1 専用回線リンクの設定 (作業マップ) (続き)

作業	説明	参照先
3. 必要に応じて、インタフェースカードを設定する	専用回線への接続を開始する際に使用するインタフェーススクリプトを設定する	458 ページの「同期デバイスの設定方法」
4. リモートピアに関する情報に基づいて設定する	ローカルマシンとリモートピア間の通信方法を定義する	460 ページの「専用回線上のマシンの設定方法」
5. 専用回線への接続を開始する	起動プロセスの一部として、PPP が専用回線を介して開始されるようにマシンを設定する	460 ページの「専用回線上のマシンの設定方法」

専用回線上の同期デバイスの設定

この節では、専用回線のトポロジに必要な機器を設定する方法について説明します。専用回線のトポロジについては、430 ページの「専用回線リンクの構成例」で紹介しています。専用回線への接続に必要な同期デバイスには、インタフェースとモデムが含まれています。

同期デバイスを設定する際の前提条件

次の手順に従う前に、下記の項目を確認する必要があります。

- プロバイダによって設置された専用回線が動作していること
- 同期装置 (CSU/DSU)
- システムに Solaris 9 または Solaris 10 がインストールされていること
- システムに必要な同期インタフェースカード

▼ 同期デバイスの設定方法

1. 必要に応じて、インタフェースカードをローカルマシンに取り付けます。
   製造元のマニュアルの手順に従います。

2. **CSU/DSU** とインタフェースをケーブルで接続します。
   必要に応じて、CSU/DSU と専用回線のジャックまたは同等のコネクタをケーブルで接続します。

3. 製造元またはネットワークプロバイダのマニュアルの手順に従って、**CSU/DSU** を設定します。

注 – 専用回線を貸し出しているプロバイダが、接続用の CSU/DSU を提供および設定する場合もあります。

4 必要に応じて、インタフェースのマニュアルの手順に従って、インタフェースカードを設定します。

インタフェースカードの設定時に、インタフェースの起動スクリプトを作成します。図 16–2 のような専用回線設定では、LocalCorp にあるルーターは、HSI/P インタフェースカードを使用します。

次のスクリプト `hsi-conf` によって、HSI/P インタフェースが開始されます。

```
#!/bin/ksh
/opt/SUNWconn/bin/hsip_init hihp1 speed=1536000 mode=fdx loopback=no \
nrzi=no txc=txc rxc=rxc txd=txd rxd=rxd signal=no 2>&1 > /dev/null
```

`hihp1`	使用されている同期ポートが HSI/P であることを示す
`speed=1536000`	CSU/DSU の速度を示すために設定する

参照 専用回線上のローカルマシンの設定手順については、460 ページの「専用回線上のマシンの設定方法」を参照してください。

専用回線上のマシンの設定

この節では、ルーターを専用回線の終端でローカルピアとして機能するように設定する方法について説明します。ここでは、430 ページの「専用回線リンクの構成例」で紹介した専用回線を例として使用します。

専用回線上のローカルマシンを設定する際の前提条件

以降の手順を実行する前に、次の作業を終了しておく必要があります。

- 458 ページの「専用回線上の同期デバイスの設定」の説明に従って、接続に使用する同期デバイスをセットアップおよび設定する
- 専用回線上のローカルマシンのスーパーユーザーパスワードを取得する
- ローカルマシンがネットワークのルーターとして動作し、専用回線プロバイダのサービスを使用するように設定する

▼ 専用回線上のマシンの設定方法

1 ローカルマシン (ルーター) 上のスーパーユーザー、またはそれと同等の役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 リモートピア用のエントリをルーターの /etc/hosts ファイルに追加します。

```
# cat /etc/hosts
#
# Internet host table
#
127.0.0.1       localhost
192.168.130.10  local2-peer        loghost
192.168.130.11  local1-net
10.0.0.25       farISP
```

/etc/hosts の例は、架空の LocalCorp のローカルルーター用のファイルです。サービスプロバイダのリモートピア farISP の IP アドレスおよびホスト名をメモしておきます。

3 プロバイダのピアに関する情報を保持する /etc/ppp/peers/*peer-name* ファイルを作成します。

この例の専用回線への接続用に、/etc/ppp/peers/farISP ファイルを作成します。

```
# cat /etc/ppp/peers/farISP
init '/etc/ppp/conf_hsi'
local
/dev/hihp1
sync
noauth
192.168.130.10:10.0.0.25
passive
persist
noccp
nopcomp
novj
noaccomp
```

次の表では、/etc/ppp/peers/farISP で使用されているオプションおよびパラメータについて説明しています。

オプション	定義
init '/etc/ppp/conf_hsi'	接続を開始する。次に、init はスクリプト /etc/ppp/conf_hsi のパラメータを使用して、HSI インタフェースを設定する

オプション	定義
local	データ端末レディー (DTR) 信号の状態を変更しないように、pppd デーモンに指示する。また、データキャリア検出 (DCD) 入力信号を無視することも pppd に指示する
/dev/hihp1	同期インタフェースのデバイス名を指定する
sync	接続の同期エンコーディングを確立する
noauth	ローカルシステムがピアに認証を要求する必要がないように設定する。ただし、ピアは認証を要求することができる
192.168.130.10:10.0.0.25	ローカルピアおよびリモートピアの IP アドレスをコロンで区切って定義する
passive	最大数の LCP Configure-Request を発行したら、ピアが起動するまで待機するように、ローカルマシンの pppd デーモンに指示する
persist	接続が解除されたあとでもう一度接続を開始するように、pppd デーモンに指示する
noccp、nopcomp、novj、noaccomp	CCP (Compression Control Protocol)、プロトコルフィールドの圧縮、Van Jacobson 圧縮、およびアドレスとコントロールフィールドの圧縮をそれぞれ無効にする。これらの圧縮形式を使用すると、ダイアルアップリンクでの伝送速度は速くなるが、専用回線での伝送速度は遅くなる可能性がある

4 demand という初期設定スクリプトを作成します。こうすると、起動プロセスの一部として **PPP** リンクが開始されます。

```
# cat /etc/ppp/demand
#!/bin/sh
if [ -f /var/run/ppp-demand.pid ] &&
   /usr/bin/kill -s 0 ‘/bin/cat /var/run/ppp-demand.pid‘
then
        :
else
        /usr/bin/pppd call farISP
fi
```

demand スクリプトには、専用回線リンクを確立するための pppd コマンドが含まれています。次の表では、$PPPDIR/demand の内容について説明しています。

コーディング例	意味
if [-f /var/run/ppp-demand.pid] && /usr/bin/kill -s 0 ‘/bin/cat /var/run/ppp-demand.pid‘	これらの行は、pppd が動作しているかどうかを確認する。pppd が動作している場合は、起動する必要はない

コーディング例	意味
`/usr/bin/pppd call farISP`	この行は、`pppd` を起動する。`pppd` は、`/etc/ppp/options` からオプションを読み取る。 `call farISP` オプションをコマンド行で指定すると、`/etc/ppp/peers/farISP` も読み取る

Solaris PPP 4.0 の起動スクリプト `/etc/rc2.d/S47pppd` によって、`demand` スクリプトが、Solaris の起動プロセスの一部として呼び出されます。`/etc/rc2.dS47pppd` にある次の行は、`$PPPDIR/demand` というファイルが存在するかどうかを調べます。

```
if [ -f $PPPDIR/demand ]; then
            . $PPPDIR/demand
    fi
```

`$PPPDIR/demand` が検出された場合は、それが実行されます。`$PPPDIR/demand` の一連の処理の実行中に、接続が確立されます。

注 – ローカルネットワークの外部にあるマシンにアクセスするためには、ユーザーに、`telnet`、`ftp`、`rsh`、または 同様のコマンドを実行させます。

参照 この章のすべての手順を実行すると、専用回線接続の構成が完了します。関連情報の参照先は次のとおりです。

- トラブルシューティングの情報については、503 ページの「専用回線の問題の解決」
- この章で使用するファイルとオプションについてさらに学習するときは、505 ページの「ファイルおよびコマンド行での PPP オプションの使用」

◆ ◆ ◆ 第 19 章

PPP 認証の設定 (手順)

この章では、PPP 認証の設定手順について説明します。ここでは、次の内容を説明します。

- 464 ページの「PAP 認証の設定」
- 472 ページの「CHAP 認証の設定」

ここでは、ダイアルアップリンクに認証を実装する方法について説明しています。これは、ダイアルアップリンクの方が、専用回線リンクよりも認証を設定することが多いためです。企業のセキュリティーポリシーにより認証が必要な場合には、専用回線に認証を設定することもできます。専用回線に認証を設定する場合は、この章の手順をガイドラインとして参照してください。

PPP 認証を使用する場合で、どのプロトコルを使用したらいいのかわからないときには、421 ページの「PPP 認証を使用する理由」を参照してください。PPP 認証の詳細は、pppd(1M) のマニュアルページおよび 528 ページの「接続時の呼び出し元の認証」を参照してください。

PPP 認証の構成 (作業マップ)

次の作業マップに、PPP 認証に関連する作業を示します。

表 19–1 一般的な PPP 認証 (作業マップ)

作業	説明	参照先
PAP 認証を設定する	ダイアルインサーバーおよびダイアルアウトマシン上で PAP 認証を可能にするための手順を使用する	464 ページの「PAP 認証の設定 (作業マップ)」
CHAP 認証を設定する	ダイアルインサーバーおよびダイアルアウトマシン上で CHAP 認証を可能にするための手順を使用する	472 ページの「CHAP 認証の設定 (作業マップ)」

PAP 認証の設定

この節では、パスワード認証プロトコル (PAP) を使用して、PPP リンクに認証を実装する方法について説明します。ここでは、433 ページの「PPP の認証構成例」の例を使用して、ダイアルアップリンクで PAP を動作させる方法について説明します。PAP 認証を実装する場合は、この手順を基準として使用してください。

以降の手順を実行する前に、次の作業を終了しておく必要があります。

- ダイアルインサーバーと信頼できる呼び出し元が所有するダイアルアウトマシン間で、ダイアルアップリンクを設定しテストします。
- ダイアルインサーバーでの認証に備えて、LDAP、NIS、またはローカルファイルなどでネットワークパスワードデータベースを管理しているマシンに対するスーパーユーザーとしてのアクセス権を取得することが理想的です。
- ローカルマシン、およびダイアルインサーバーまたはダイアルアウトマシンに対するスーパーユーザーとしての権限を取得します。

PAP 認証の設定 (作業マップ)

次の作業マップに、ダイアルインサーバーおよびダイアルアウトマシン上の信頼できる呼び出し元に対して実行する PAP 関連の作業を示します。

表 19–2　PAP 認証についての作業マップ (ダイアルインサーバー)

作業	説明	参照先
1. 構成前の情報を収集する	ユーザー名など、認証に必要なデータを収集する	432 ページの「リンクへの認証計画」
2. 必要に応じて、パスワードデータベースを更新する	候補となるすべての呼び出し元が、サーバーのパスワードデータベースに含まれていることを確認する	465 ページの「PAP 資格データベースの作成方法 (ダイアルインサーバー)」
3. PAP データベースを作成する	将来接続する可能性のあるすべての呼び出し元のセキュリティー資格を `/etc/ppp/pap-secrets` に作成する	465 ページの「PAP 資格データベースの作成方法 (ダイアルインサーバー)」
4. PPP の構成ファイルを変更する	PAP 特有のオプションを `/etc/ppp/options` および `/etc/ppp/peers/`*peer-name* ファイルに追加する	467 ページの「PPP 構成ファイルに PAP サポートを追加する方法 (ダイアルインサーバー)」

表 19–3　PAP 認証についての作業マップ (ダイアルアウトマシン)

作業	説明	参照先
1. 構成前の情報を収集する	ユーザー名など、認証に必要なデータを収集する	432 ページの「リンクへの認証計画」
2. 信頼できる呼び出し元のマシン用の PAP データベースを作成する	信頼できる呼び出し元のセキュリティー資格と、必要であれば、ダイアルアウトマシンを呼び出すほかのユーザーのセキュリティー資格を /etc/ppp/pap-secrets に作成する	469 ページの「信頼できる呼び出し元に PAP 認証資格を設定する方法」
3. PPP の構成ファイルを変更する	PAP 特有のオプションを /etc/ppp/options および /etc/ppp/peers/*peer-name* ファイルに追加する	470 ページの「PPP 構成ファイルに PAP サポートを追加する方法 (ダイアルアウトマシン)」

ダイアルインサーバーに PAP 認証を構成する

PAP 認証を設定するには、次の手順に従う必要があります。

- PAP 資格データベースを作成します。
- PAP をサポートするように PPP 構成ファイルを変更します。

▼ PAP 資格データベースの作成方法 (ダイアルインサーバー)

ここでは、/etc/ppp/pap-secrets ファイルを変更します。このファイルには、接続時に呼び出し元の認証に使用する PAP セキュリティー資格が含まれています。PPP リンクを行う両方のマシンに /etc/ppp/pap-secrets が必要です。

図 16–3 で紹介した PAP 構成のサンプルでは、PAP の login オプションが使用されています。このオプションを使用する場合は、ネットワークのパスワードデータベースも更新する必要がある可能性があります。login オプションの詳細については、531 ページの「/etc/ppp/pap-secrets での login オプションの使用」を参照してください。

1. 候補となる信頼できるすべての呼び出し元のリストを作成します。信頼できる呼び出し元とは、自分のリモートマシンからダイアルインサーバーを呼び出す権限を与えられているユーザーです。

2. ダイアルインサーバーのパスワードデータベースに、信頼できる呼び出し元全員の **UNIX** ユーザー名およびパスワードがあることを確認します。

注 – この確認は、この PAP 構成のサンプルにとって重要です。このサンプルでは、呼び出し元の認証に、PAP の login オプションを使用しています。PAP に login を実装しない場合は、呼び出し元の PAP ユーザー名と UNIX ユーザー名を一致させる必要はありません。標準の /etc/ppp/pap-secrets については、528 ページの「/etc/ppp/pap-secrets ファイル」を参照してください。

候補となる信頼できる呼び出し元に UNIX 名とパスワードがない場合は、次の手順に従います。

a. 呼び出し元に関する情報がない場合は、呼び出し元がダイアルインサーバーへのアクセス権を持っているかどうかをその呼び出し元の管理者に確認します。

b. 企業のセキュリティーポリシーが指定する方法に従って、これらの呼び出し元に **UNIX** ユーザー名およびパスワードを作成します。

3 ダイアルインサーバー上のスーパーユーザー、またはそれと同等の役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

4 /etc/ppp/pap-secrets ファイルを編集します。

Solaris PPP 4.0 では、/etc/ppp に pap-secrets ファイルがあります。このファイルには、PAP 認証の使用方法についてのコメントが含まれています。ただし、オプションについてのコメントは含まれていません。コメントの最後に、次のオプションを追加することができます。

```
user1      myserver        ""           *
user2      myserver        ""           *
myserver   user2           serverpass   *
```

/etc/ppp/pap-secrets の login オプションを使用するには、信頼できる呼び出し元の UNIX 名をすべて入力する必要があります。3 番目のフィールドのどこに二重引用符 (“”) が記述されても、呼び出し元のパスワードは、サーバーのパスワードデータベースで参照できます。

エントリ myserver * serverpass * には、ダイアルインサーバー用の PAP ユーザー名およびパスワードが含まれています。図 16–3 では、信頼できる呼び出し元である user2 は、リモートピアに認証を要求します。そのため、myserver の /etc/ppp/pap-secrets ファイルには、user2 との接続を確立する場合に使用する PAP 資格が含まれています。

参照 関連情報の参照先は次のとおりです。

- 467 ページの「PPP 構成ファイルを PAP 用に変更する (ダイアルインサーバー)」

- 468 ページの「信頼できる呼び出し元の PAP 認証の設定 (ダイアルアウトマシン)」

PPP 構成ファイルを PAP 用に変更する (ダイアルインサーバー)

この節では、ダイアルインサーバーで PAP 認証をサポートするように、既存の PPP 構成ファイルを更新する方法について説明します。

▼ PPP 構成ファイルに PAP サポートを追加する方法 (ダイアルインサーバー)

ここでは、453 ページの「シリアル回線を介した通信を定義する方法 (ダイアルインサーバー)」で紹介した PPP 構成ファイルを例として使用します。

1 ダイアルインサーバー上のスーパーユーザー、またはそれと同等の役割としてログインします。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 認証オプションを `/etc/ppp/options` ファイルに追加します。

たとえば、既存の `/etc/ppp/options` ファイルに、次の太字のオプションを追加すると、PAP 認証を実装することができます。

```
lock
auth
login
nodefaultroute
proxyarp
ms-dns 10.0.0.1
idle 120
```

`auth`	接続を確立する前に、サーバーが呼び出し元を認証する必要があることを示す
`login`	リモート呼び出し元が、標準的な UNIX ユーザー認証サービスを使用して認証されることを示す
`nodefaultroute`	ローカルシステム上の `pppd` セッションが `root` 権限がないとデフォルトの経路を確立できないことを示す
`proxyarp`	ピアの IP アドレスやシステムの Ethernet アドレスを指定するシステムのアドレス解決プロトコル (ARP) テーブルにエントリを追加する。このオプションを使用すると、ピアは、ほかのシステムのローカル Ethernet 上にあるように見える

`ms-dns 10.0.0.1`	pppd がクライアントにドメインネームサーバー (DNS) アドレス `10.0.0.1`を与えることができるようにする
`idle 120`	2 分後にアイドルユーザーの接続が切断されることを示す

3 `/etc/ppp/options.cua.a` ファイルに、`cua/a` ユーザーの次のアドレスを追加します。

```
:10.0.0.2
```

4 `/etc/ppp/options.cua.b` ファイルに、`cua/b` ユーザーの次のアドレスを追加します。

```
:10.0.0.3
```

5 `/etc/ppp/pap-secrets` ファイルに、次のエントリを追加します。

```
*     *         ""      *
```

注 – 前述したように、`login` オプションは、必要なユーザー認証を与えます。`/etc/ppp/pap-secrets` ファイルのこのエントリは、`login` オプションを使用して PAP を可能にする標準的な方法です。

参照 ダイアルインサーバーの信頼できる呼び出し元の PAP 認証資格を設定する手順については、468 ページの「信頼できる呼び出し元の PAP 認証の設定 (ダイアルアウトマシン)」を参照してください。

信頼できる呼び出し元の PAP 認証の設定 (ダイアルアウトマシン)

この節では、信頼できる呼び出し元のダイアルアウトマシンで、PAP 認証を設定する手順について説明します。システム管理者は、システムで PAP 認証を設定し、それらを将来接続する可能性のある呼び出し元に配布することができます。また、リモート呼び出し元にすでにマシンがある場合は、この節の手順を指示することもできます。

信頼できる呼び出し元に PAP を設定するには、次の 2 つの手順を実行します。

- 呼び出し元の PAP セキュリティー資格を設定します。
- 呼び出し元のダイアルアウトマシンが PAP 認証をサポートするように設定します。

▼ 信頼できる呼び出し元に PAP 認証資格を設定する方法

ここでは、2 人の信頼できる呼び出し元の PAP 資格を設定する方法について説明します。これらのうちの 1 人は、リモートピアに認証資格を要求します。この手順では、システム管理者が、信頼できる呼び出し元のダイアルアウトマシンで PAP 資格を作成することを前提にしています。

1. ダイアルアウトマシン上のスーパーユーザー、またはそれと同等の役割になります。

   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

   図 16–3 で紹介した PAP 構成のサンプルでは、`user1` がダイアルアウトマシンを所有しています。

2. 呼び出し元の **pap-secrets** データベースを変更します。

   Solaris PPP 4.0 には、`/etc/ppp/pap-secrets` ファイルがあります。このファイルには、便利な情報が含まれていますが、オプションについては触れていません。次のオプションをこの `/etc/ppp/pap-secrets` ファイルに追加できます。

   ```
   user1    myserver  pass1    *
   ```

   `user1` のパスワードである `pass1` は、接続を通して、読み取り可能な ASCII 形式になることに注意してください。`myserver` は、呼び出し元 `user1` がピアで使用する名前です。

3. ほかのダイアルアウトマシン上のスーパーユーザー、またはそれと同等の役割になります。

   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

   PAP 認証の例では、呼び出し元 `user2` がこのダイアルアウトマシンを所有しています。

4. 呼び出し元の **pap-secrets** データベースを変更します。

   次のオプションを既存の `/etc/ppp/pap-secrets` ファイルの終わりに追加できます。

   ```
   user2     myserver   pass2       *
   myserver  user2      serverpass  *
   ```

   この例では、`/etc/ppp/pap-secrets` に 2 つのエントリがあります。最初のエントリには、`user2` が認証のためにダイアルインサーバー `myserver` に渡す PAP セキュリティー資格が含まれています。

user2 は、接続のネゴシエーションの一部として、ダイアルインサーバーに PAP 資格を要求します。そのため、/etc/ppp/pap-secrets の 2 つ目の行に、myserver に要求される PAP 資格も含まれています。

注 – ほとんどの ISP は認証資格を提供しないため、ここで検討しているシナリオは、ISP との通信に関しては現実的ではありません。

参照 関連情報の参照先は次のとおりです。

- 465 ページの「PAP 資格データベースの作成方法 (ダイアルインサーバー)」
- 469 ページの「信頼できる呼び出し元に PAP 認証資格を設定する方法」

PPP 構成ファイルを PAP 用に変更する (ダイアルアウトマシン)

次の作業は、信頼できる呼び出し元のダイアルアウトマシンで PAP 認証をサポートするように、既存の PPP 構成ファイルを更新する方法について説明します。

ここでは、次のパラメータを使用して、図 16–3 で紹介した user2 が所有するダイアルアウトマシン上で、PAP 認証を設定します。user2 は、ダイアルイン myserver からの呼び出しを含む着信呼び出し元に、認証を要求します。

▼ PPP 構成ファイルに PAP サポートを追加する方法 (ダイアルアウトマシン)

ここでは、445 ページの「シリアル回線を介した通信を定義する方法」で紹介した PPP 構成ファイルを例として使用します。この手順に従って、図 16–3 で示した user2 が所有するダイアルアウトマシンを設定します。

1 ダイアルアウトマシンにスーパーユーザーとしてログインします。

2 /etc/ppp/options ファイルを変更します。

次の /etc/ppp/options ファイルには、太字で示した PAP サポート用のオプションが含まれています。

```
# cat /etc/ppp/options
lock
name user2
auth
require-pap
```

name user2　user2 をローカルマシン上のユーザーの PAP 名として設定する。login オプションを使用する場合は、PAP 名をパスワードデータベースにあるそのユーザーの UNIX ユーザー名と一致させる必要がある

auth　接続を確立する前に、ダイアルアウトマシンが呼び出し元を認証する必要があることを明示する

注 – ほとんどのダイアルアウトマシンはピアに対する認証要求を行いませんが、このダイアルアウトマシンはピアに認証を要求します。どちらも可能です。

require-pap　ピアに PAP 資格を要求する

3 リモートマシン myserver 用の */etc/ppp/peers/*peer-name ファイルを作成します。

次のサンプルは、447 ページの「個々のピアとの接続を定義する方法」で作成した既存の /etc/ppp/peers/myserver ファイルに、PAP サポートを追加する方法を示しています。

```
# cat /etc/ppp/peers/myserver
/dev/cua/a
57600
noipdefault
defaultroute
idle 120
user user2
remotename myserver
connect "chat -U 'mypassword' -f /etc/ppp/mychat"
```

太字で示した新しいオプションにより、ピア myserver に関する PAP 要件が追加されます。

user user2　user2 をローカルマシンのユーザー名として定義する

remotename myserver　myserver をローカルマシンに認証資格を要求するピアとして定義する

参照　関連情報の参照先は次のとおりです。

- ダイアルインサーバーを呼び出して、PAP 認証の設定をテストする手順については、454 ページの「ダイアルインサーバーの呼び出し方法」
- PAP 認証の詳細を理解するときは、528 ページの「パスワード認証プロトコル (PAP)」

CHAP 認証の設定

この節では、チャレンジハンドシェーク認証プロトコル (CHAP) を使用して、PPP リンクに認証を実装する方法について説明します。ここでは、図 16–4 の例を使用して、私設ネットワークへのダイアルアップで CHAP を動作させる方法について説明します。CHAP 認証を実装する場合は、この手順を基準として使用してください。

以降の手順を実行する前に、次の作業を終了しておく必要があります。

- ダイアルインサーバーと信頼できる呼び出し元が所有するダイアルアウトマシン間で、ダイアルアップリンクを設定しテストします。
- ローカルマシン (ダイアルインサーバーまたはダイアルアウトマシン) に対するスーパーユーザーとしてのアクセス権を取得します。

CHAP 認証の設定 (作業マップ)

表 19–4 CHAP 認証についての作業マップ (ダイアルインサーバー)

作業	説明	参照先
1. CHAP シークレットをすべての信頼できる呼び出し元に割り当てる	CHAP シークレットを作成する、または呼び出し元に作成させる	473 ページの「CHAP 資格データベースの作成方法 (ダイアルインサーバー)」
2. chap-secrets データベースを作成する	すべての信頼できる呼び出し元のセキュリティー資格を `/etc/ppp/chap-secrets` ファイルに追加する	473 ページの「CHAP 資格データベースの作成方法 (ダイアルインサーバー)」
3. PPP の構成ファイルを変更する	CHAP 特有のオプションを `/etc/ppp/options` および `/etc/ppp/peers/`*peer-name* ファイルに追加する	475 ページの「PPP 構成ファイルに CHAP サポートを追加する方法 (ダイアルインサーバー)」

表 19–5 CHAP 認証についての作業マップ (ダイアルアウトマシン)

作業	説明	参照先
1. 信頼できる呼び出し元のマシン用の CHAP データベースを作成する	信頼できる呼び出し元のセキュリティー資格と、必要であれば、ダイアルアウトマシンを呼び出すほかのユーザーのセキュリティー資格を `/etc/ppp/chap-secrets` に作成する	473 ページの「CHAP 資格データベースの作成方法 (ダイアルインサーバー)」
2. PPP の構成ファイルを変更する	CHAP 特有のオプションを `/etc/ppp/options` ファイルに追加する	477 ページの「PPP 構成ファイルに CHAP サポートを追加する方法 (ダイアルアウトマシン)」

ダイアルインサーバーに CHAP 認証を構成する

CHAP 認証を設定するには、最初に /etc/ppp/chap-secrets ファイルを変更します。このファイルには、CHAP シークレットを含む CHAP セキュリティー資格が含まれています。このセキュリティー資格を使用して、接続時に呼び出し元を認証します。

注 – UNIX の認証メカニズムまたは PAM の認証メカニズムを CHAP とともに使用することはできません。たとえば、How to Create a PAP Credentials Database (Dial-in Server) で説明したような PPP 465 ページの「PAP 資格データベースの作成方法 (ダイアルインサーバー)」 オプションを使用することはできません。認証時に、PAM または UNIX スタイルの認証が必要な場合は、代わりに PAP を選択してください。

次に、私設ネットワークにあるダイアルインサーバーの CHAP 認証を実装します。PPP リンクは、外部のネットワークに接続する場合にだけ使用します。ネットワークにアクセスできるのは、ネットワーク管理者からアクセス権を与えられている呼び出し元だけです。その中には、システム管理者が含まれることもあります。

▼ CHAP 資格データベースの作成方法 (ダイアルインサーバー)

1. 信頼できる呼び出し元のユーザー名をすべて含むリストを作成します。
   信頼できる呼び出し元とは、私設ネットワークを呼び出す権限を与えられているユーザーです。

2. 各ユーザーに CHAP シークレットを割り当てます。

注 – CHAP シークレットには、容易に予想しにくいものを選択してください。CHAP シークレットの内容については、予想しにくいものにするということ以外の制限はありません。

CHAP シークレットを割り当てる方法は、企業のセキュリティーポリシーにより違います。管理者がシークレットを作成するか、呼び出し元が自分のシークレットを作成する必要があります。自分が CHAP シークレットを割り当てる立場にない場合は、信頼できる呼び出し元によって、または信頼できる呼び出し元のために作成された CHAP シークレットを取得することを忘れないでください。

3. ダイアルインサーバー上のスーパーユーザー、またはそれと同等の役割になります。

   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

4. `/etc/ppp/chap-secrets` ファイルを変更します。

   Solaris PPP 4.0 には、`/etc/ppp/chap-secrets` ファイルがあります。このファイルには、便利な情報が含まれていますが、オプションについては触れていません。サーバー `CallServe` 用の次のオプションを既存の `/etc/ppp/chap-secrets` ファイルの最後に追加することができます。

   ```
   account1  CallServe   key123   *
   account2  CallServe   key456   *
   ```

   `key123` は、信頼できる呼び出し元 `account1` の CHAP シークレットです。

   `key456` は、信頼できる呼び出し元 `account2` の CHAP シークレットです。

参照 関連情報の参照先は次のとおりです。

- 473 ページの「CHAP 資格データベースの作成方法 (ダイアルインサーバー)」
- 475 ページの「PPP 構成ファイルに CHAP サポートを追加する方法 (ダイアルインサーバー)」
- 475 ページの「信頼できる呼び出し元の CHAP 認証の設定 (ダイアルアウトマシン)」

PPP 構成ファイルを CHAP 用に変更する (ダイアルインサーバー)

この節では、ダイアルインサーバーで CHAP 認証をサポートするように、既存の PPP 構成ファイルを更新する方法について説明します。

▼ PPP 構成ファイルに CHAP サポートを追加する方法 (ダイアルインサーバー)

1 ダイアルインサーバーにスーパーユーザーとしてログインします。

2 `/etc/ppp/options` ファイルを変更します。

太字で表示されているオプションを追加して、CHAP がサポートされるようにします。

```
# cat /etc/ppp/options
lock
nodefaultroute
name CallServe
auth
```

`name` *CallServe*	*CallServe* をローカルマシン上のユーザーの CHAP 名として定義する。この場合、ローカルマシンはダイアルインサーバーである
`auth`	ローカルマシンで呼び出し元を認証してから、接続を確立する

3 信頼できる呼び出し元をサポートするために必要なその他の PPP 構成ファイルを作成します。

451 ページの「ダイアルインサーバーのユーザーを構成する方法」および 453 ページの「シリアル回線を介した通信を定義する方法 (ダイアルインサーバー)」を参照してください。

参照 信頼できる呼び出し元の CHAP 認証資格を設定する手順については、473 ページの「CHAP 資格データベースの作成方法 (ダイアルインサーバー)」を参照してください。

信頼できる呼び出し元の CHAP 認証の設定 (ダイアルアウトマシン)

この節では、信頼できる呼び出し元のダイアルアウトマシンで、CHAP 認証を設定する手順について説明します。企業のセキュリティーポリシーによって、管理者と信頼できる呼び出し元のどちらが CHAP 認証を設定するのかが決まります。

リモート呼び出し元が CHAP を設定する場合は、呼び出し元のローカルの CHAP シークレットが、ダイアルインサーバーの `/etc/ppp/chap-secrets` ファイルに記述されている CHAP シークレットと一致していることを確認します。その後、呼び出し元に、この節で説明している CHAP 設定の手順を指示します。

信頼できる呼び出し元に CHAP を設定するには、次の 2 つの手順を実行します。

- 呼び出し元の CHAP セキュリティー資格を作成します。
- 呼び出し元のダイアルアウトマシンが CHAP 認証をサポートするように設定します。

▼ 信頼できる呼び出し元に CHAP 認証資格を設定する方法

ここでは、2 人の信頼できる呼び出し元に、PAP 資格を設定する方法について説明します。この手順では、システム管理者が、信頼できる呼び出し元のダイアルアウトマシンで CHAP 資格を作成することを前提にしています。

1 ダイアルアウトマシン上のスーパーユーザー、またはそれと同等の役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

435 ページの「CHAP 認証による構成例」の CHAP 構成のサンプルでは、信頼できる呼び出し元 `account1` がダイアルアウトマシンを所有しています。

2 **chap-secrets** データベースを呼び出し元 `account1` 用に変更します。

Solaris PPP 4.0 には、`/etc/ppp/chap-secrets` ファイルがあります。このファイルには、便利な情報が含まれていますが、オプションについては触れていません。次のオプションをこの既存の `/etc/ppp/chap-secrets` ファイルに追加できます。

```
account1   CallServe    key123    *
```

`CallServe` は、`account1` がアクセスを試みているピアの名前です。`key123` は、`account1` と `CallServer` 間での接続に使用する CHAP シークレットです。

3 ほかのダイアルアウトマシン上のスーパーユーザー、またはそれと同等の役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

呼び出し元 `account2` がこのマシンを所有しているとします。

4 `/etc/ppp/chap-secrets` データベースを呼び出し元 **account2** 用に変更します。

```
account2   CallServe    key456    *
```

`account2` に、シークレット `key456` が、ピア `CallServe` への接続に使用する CHAP 資格として設定されます。

参照　関連情報の参照先は次のとおりです。

- 473 ページの「CHAP 資格データベースの作成方法 (ダイアルインサーバー)」
- 476 ページの「信頼できる呼び出し元に CHAP 認証資格を設定する方法」

CHAP を構成ファイルに追加する (ダイアルアウトマシン)

CHAP 認証の詳細を理解するには、531 ページの「チャレンジハンドシェーク認証プロトコル (CHAP)」を参照してください。次の手順に従って、435 ページの「CHAP 認証による構成例」で紹介した呼び出し元 account1 が所有するダイアルアウトマシンを設定します。

▼ PPP 構成ファイルに CHAP サポートを追加する方法 (ダイアルアウトマシン)

1. ダイアルアウトマシンにスーパーユーザーとしてログインします。
2. /etc/ppp/options ファイルが次のオプションを持つことを確認します。

```
# cat /etc/ppp/options
lock
nodefaultroute
```

3. リモートマシン CallServe 用の /etc/ppp/peers/*peer-name* ファイルを作成します。

```
# cat /etc/ppp/peers/CallServe
/dev/cua/a
57600
noipdefault
defaultroute
idle 120
user account1
connect "chat -U 'mypassword' -f /etc/ppp/mychat"
```

オプション user account1 により、account1 が、CallServe に提供される CHAP ユーザー名として設定されます。前のファイルのほかのオプションについては、447 ページの「個々のピアとの接続を定義する方法」の /etc/ppp/peers/myserver ファイルにある同様のオプションの説明を参照してください。

参照　ダイアルインサーバーを呼び出して、CHAP 認証をテストする手順については、454 ページの「ダイアルインサーバーの呼び出し方法」を参照してください。

◆ ◆ ◆ 第 20 章

PPPoE トンネルの設定 (手順)

この章では、PPPoE トンネルの両端、つまり PPPoE クライアントと PPPoE アクセスサーバーを設定する方法について説明します。 ここでは、次の内容を説明します。

- 479 ページの「PPPoE トンネル設定の主な作業 (作業マップ)」
- 480 ページの「PPPoE クライアントの設定」
- 483 ページの「PPPoE アクセスサーバーの設定」

ここでは、437 ページの「PPPoE トンネルを介した DSL サポートの計画」で紹介したシナリオを例として使用します。PPPoE の概要については、421 ページの「PPPoE による DSL ユーザーのサポート」を参照してください。

PPPoE トンネル設定の主な作業 (作業マップ)

次の表に、PPPoE クライアントと PPPoE アクセスサーバーを構成するための主な作業を一覧表示します。サイトで PPPoE を実装するには、PPPoE トンネルの自分の側だけ、つまりクライアント側かアクセスサーバー側のどちらかを設定します。

表 20–1 PPPoE クライアントの設定 (作業マップ)

作業	説明	参照先
1. PPPoE のインタフェースを構成する	Ethernet インタフェースを PPPoE トンネルで使用するために定義する	481 ページの「PPPoE クライアントのインタフェースを構成する方法」
2. PPPoE アクセスサーバーに関する情報を構成する	PPPoE トンネルのサービスプロバイダ側にあるアクセスサーバーのパラメータを定義する	481 ページの「PPPoE アクセスサーバーピアを定義する方法」
3. PPP 構成ファイルを設定する	まだクライアントの PPP 構成ファイルを定義していない場合は、定義する	445 ページの「シリアル回線を介した通信を定義する方法」

表 20–1 PPPoE クライアントの設定 (作業マップ) (続き)

作業	説明	参照先
4. トンネルを作成する	アクセスサーバーを呼び出す	481 ページの「PPPoE アクセスサーバーピアを定義する方法」

表 20–2 PPPoE アクセスサーバーの設定 (作業マップ)

作業	説明	参照先
1. PPPoE のアクセスサーバーを構成する	PPPoE トンネルで使用する Ethernet インタフェースと、アクセスサーバーが提供するサービスを定義する	483 ページの「PPPoE アクセスサーバーの設定方法」
2. PPP 構成ファイルを設定する	まだクライアントの PPP 構成ファイルを定義していない場合は、定義する	452 ページの「ダイアルインサーバーを介した通信を構成する」
3. (省略可能) インタフェースの使用を限定する	PPPoE オプションと PAP 認証を使用して、特定の Ethernet インタフェースの使用を特定のクライアントに限定する	485 ページの「インタフェースの使用を特定のクライアントに限定する方法」

PPPoE クライアントの設定

DSL を介してクライアントシステムに PPP を提供するには、まずモデムまたはハブに接続されているインタフェースで PPPoE を構成する必要があります。次に、PPP 構成ファイルを変更して、PPPoE の反対側のアクセスサーバーを定義する必要があります。

PPPoE クライアント設定の前提条件

PPPoE クライアントを設定する前に、次を行なっておく必要があります。

- PPPoE トンネルを使用するため、クライアントマシンに Solaris 8 Update 6 以降のリリースをインストールする
- サービスプロバイダに連絡して PPPoE アクセスサーバーに関する情報を得る
- クライアントマシンが使用するデバイスを電話会社またはサービスプロバイダに取り付けてもらう。たとえば DSL モデムやスプリッタなどのデバイスがあるが、これらは自分で取り付けるのではなく、電話会社が取り付ける

▼ PPPoE クライアントのインタフェースを構成する方法

この作業は、PPPoE トンネルで使用するように Ethernet インタフェースを定義する場合に行なってください。

1 **PPPoE クライアント上でスーパーユーザーになるか、同等の役割になります。**

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 **DSL 接続のある Ethernet インタフェースの名前を `/etc/ppp/pppoe.if` ファイルに追加します。**

たとえば、DSL モデムに接続するネットワークインタフェースに `hme0` を使用する PPPoE クライアントの場合は、`/etc/ppp/pppoe.if` に次のエントリを追加します。

```
hme0
```

`/etc/ppp/pppoe.if` の詳細は、537 ページの「`/etc/ppp/pppoe.if` ファイル」を参照してください。

3 **PPPoE を使用するためのインタフェースを構成します。**

```
# /etc/init.d/pppd start
```

4 **(省略可能) インタフェースが PPPoE に plumb されたことを確認します。**

```
# /usr/sbin/sppptun query
hme0:pppoe
hme0:pppoed
```

`/usr/sbin/sppptun` コマンドを使ってインタフェースを手動で PPPoE に plumb することもできます。手順については、538 ページの「`/usr/sbin/sppptun` コマンド」を参照してください。

▼ PPPoE アクセスサーバーピアを定義する方法

`/etc/ppp/peers/`*peer-name* ファイルでアクセスサーバーを定義します。アクセスサーバーで使用されるオプションの多くは、ダイアルインサーバーをダイアルアップシナリオで定義するのにも使用できます。`/etc/ppp/peers.`*peer-name* の詳細は、514 ページの「`/etc/ppp/peers/`*peer-name* ファイル」を参照してください。

1 **PPPoE クライアント上でスーパーユーザーになるか、同等の役割になります。**

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 /etc/ppp/peers/*peer-name* ファイルでサービスプロバイダの **PPPoE** アクセスサーバーを定義します。

たとえば、次のファイル /etc/ppp/peers/dslserve は、439 ページの「PPPoE トンネルの構成例」で紹介した Far ISP にあるアクセスサーバー dslserve を定義しています。

```
# cat /etc/ppp/peers/dslserve
sppptun
plugin pppoe.so
connect "/usr/lib/inet/pppoec hme0"
noccp
noauth
user Red
password redsecret
noipdefault
defaultroute
```

このファイルのオプションの定義については、545 ページの「アクセスサーバーピアを定義するための /etc/ppp/peers/*peer-name* ファイル」を参照してください。

3 **PPPoE** クライアント上のほかの **PPP** 構成ファイルを変更します。

a. **442** ページの「ダイアルアウトマシンの構成」で説明したダイアルアウトマシンを構成する手順に従って、/etc/ppp/options を構成します。

b. /etc/ppp/options.sppptun ファイルを作成します。/etc/ppp/options.sppptun ファイルは、**PPPoE** に **plumb** されているインタフェースのシリアルポートの **PPP** オプションを定義します。

/etc/ppp/options. *ttyname* ファイル (511 ページの「 /etc/ppp/options. *ttyname* 構成ファイル」を参照) で使用できるオプションはすべて使用できます。sppptun は pppd 構成で指定されているデバイス名なので、ファイル名には /etc/ppp/options.sppptun を使用する必要があります。

4 すべてのユーザーがクライアント上で **PPP** を起動できることを確認します。

```
# touch /etc/ppp/options
```

5 **PPP** が **DSL** 回線上で動作できるかどうかをテストします。

```
% pppd debug updetach call dslserve
```

dslserve は、439 ページの「PPPoE トンネルの構成例」で示した ISP のアクセスサーバーに指定されている名前です。debug updetach オプションにより、デバッグ情報が端末のウィンドウに表示されます。

PPP が正しく動作した場合、端末の出力には、接続がアクティブになることが表示されます。PPP が動作しない場合は、次のコマンドを実行してサーバーが正しく動作しているかどうかを確認します。

```
# /usr/lib/inet/pppoec -i hme0
```

注 – 構成した PPPoE クライアントのユーザーは、次のコマンドを入力して DSL 回線上で PPP の実行を開始できます。

```
% pppd call ISP-server-name
```

続いてユーザーは、アプリケーションまたはサービスを実行できます。

参照 関連情報の参照先は次のとおりです。

- 480 ページの「PPPoE クライアントの設定」
- 536 ページの「DSL サポート用の PPPoE トンネルの作成」
- 第 21 章「一般的な PPP 問題の解決 (手順)」.
- 483 ページの「PPPoE アクセスサーバーの設定」

PPPoE アクセスサーバーの設定

サービスプロバイダ会社の場合、DSL 接続を介してサイトに到達するクライアントに対してインターネットサービスやその他のサービスを提供できます。作業としては、サーバー上のどのインタフェースを PPPoE トンネルに使用するかを決定するとともに、ユーザーに許可するサービスを決定します。

▼ PPPoE アクセスサーバーの設定方法

この作業は、PPPoE トンネルで使用する Ehernet インタフェースを定義し、アクセスサーバーが提供するサービスを設定する場合に行なってください。

1 アクセスサーバーのスーパーユーザーになるか、同等の役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 **PPPoE トンネル専用の Ethernet インタフェースの名前を `/etc/ppp/pppoe.if` ファイルに追加します。**

たとえば、次の `/etc/ppp/pppoe.if` ファイルを 439 ページの「PPPoE トンネルの構成例」で示したアクセスサーバー `dslserve` に使用します。

```
# cat /etc/ppp/pppoe.if
hme1
hme2
```

3 /etc/ppp/pppoe ファイルで、アクセスサーバーが提供する広域サービスを定義します。

次の /etc/ppp/pppoe ファイルは、図 16–5 で示したアクセスサーバー dslserve によって提供されるサービスを一覧表示しています。

```
device hme1,hme2
service internet
    pppd "proxyarp 192.168.1.1:"
service debugging
    pppd "debug proxyarp 192.168.1.1:"
```

このファイルの例では、dslserve の Ethernet インタフェース hme1 および hme2 でインターネットサービスが宣言されています。また、Ethernet インタフェース上の PPP リンクでデバッグがオンに設定されています。

4 ダイアルインサーバーと同じ方法で **PPP** 構成ファイルを設定します。

詳細は、534 ページの「呼び出し元の IP アドレス指定スキーマの作成」を参照してください。

5 pppoed デーモンを起動します。

```
# /etc/init.d/pppd start
```

pppd もまた、/etc/ppp/pppoe.if に一覧表示されるインタフェースを plumb します。

6 (省略可能) サーバー上のインタフェースが **PPPoE** に **plumb** されていることを確認します。

```
# /usr/sbin/sppptun query
hme1:pppoe
hme1:pppoed
hme2:pppoe
hme2:pppoed
```

この例は、インタフェース hme1 および hme2 が現在 PPPoE に plumb されていることを示しています。/usr/sbin/sppptun コマンドを使ってインタフェースを手動で PPPoE に plumb することもできます。手順については、538 ページの「/usr/sbin/sppptun コマンド」を参照してください。

▼ 既存の /etc/ppp/pppoe ファイルを変更する方法

1 アクセスサーバーのスーパーユーザーになるか、同等の役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 必要に応じて /etc/ppp/pppoe を変更します。

3 pppoed デーモンに新しいサービスを認識させます。

```
# pkill -HUP pppoed
```

▼ インタフェースの使用を特定のクライアントに限定する方法

次に、インタフェースを PPPoE クライアントのグループに限定する手順を説明します。この作業を実行する前に、インタフェースに割り当てているクライアントの実 Ethernet MAC アドレスを取得する必要があります。

注 – システムによっては、Ethernet インタフェース上で MAC アドレスを変更できます。この機能は便利ですが、セキュリティー対策としては考えないでください。

次の手順では、439 ページの「PPPoE トンネルの構成例」で示した例を使用して、dslserve のインタフェースの 1 つである hme1 を MiddleCo のクライアント用に予約する方法を示しています。

1 483 ページの「PPPoE アクセスサーバーの設定方法」に示されている手順に従ってアクセスサーバーのインタフェースを構成し、サービスについて定義します。

2 サーバーの /etc/ethers データベースにクライアントのエントリを作成します。

次は、Red、Blue、および Yellow というクライアントのエントリの例です。

```
8:0:20:1:40:30 redether
8:0:20:1:40:10 yellowether
8:0:20:1:40:25 blueether
```

この例では、クライアントの Red、Yellow、および Blue の Ethernet アドレスに redether、yellowether、および blueether という記号名を割り当てています。MAC アドレスへの記号名の割り当ては任意です。

3 特定のインタフェース上で提供されるサービスを限定するには、次の情報を /etc/ppp/pppoe.*device* ファイルで定義します。

このファイル名で、*device* は定義するデバイスの名前です。

```
# cat /etc/ppp/pppoe.hme1
service internet
     pppd "name dslserve-hme1"
     clients redether,yellowether,blueether
```

dslserve-hme1 はアクセスサーバーの名前で、pap-secrets ファイル内の同じエントリで使用されます。clients オプションは、インタフェース hme1 の使用を Ethernet 記号名が redether、yellowether、および blueether であるクライアントに限定します。

/etc/ethers でクライアントの MAC アドレスに記号名を定義していない場合は、clients オプションの引数として数値アドレスを使用できます。このとき、ワイルドカードも使用できます。

たとえば、clients 8:0:20:*:*:* のような数値アドレスを指定できます。ワイルドカードを使用することで、/etc/ethers 内の一致するアドレスすべてにアクセスが許可されます。

4 アクセスサーバーの /etc/ppp/pap-secrets ファイルを作成します。

```
Red         dslserve-hme1   redpasswd     *
Blue        dslserve-hme1   bluepasswd    *
Yellow      dslserve-hme1   yellowpassd   *
```

エントリは、dslserve の hme1 インタフェース上で PPP を実行することを許可されたクライアントの PAP 名およびパスワードです。

PAP 認証の詳細は、464 ページの「PAP 認証の設定」を参照してください。

参照 関連情報の参照先は次のとおりです。

- PPPoE の詳細については、536 ページの「DSL サポート用の PPPoE トンネルの作成」を参照してください。
- PPPoE と PPP のトラブルシューティングについては、491 ページの「PPP および PPPoE 関連の問題の解決」を参照してください。
- PPPoE クライアントの構成については、480 ページの「PPPoE クライアントの設定」を参照してください。
- クライアントの PAP 認証の構成については、468 ページの「信頼できる呼び出し元の PAP 認証の設定 (ダイアルアウトマシン)」を参照してください。
- サーバー上の PAP 認証の構成については、465 ページの「ダイアルインサーバーに PAP 認証を構成する」を参照してください。

◆ ◆ ◆ 第 21 章

一般的な PPP 問題の解決 (手順)

この章では、Solaris PPP 4.0 で発生する一般的な問題のトラブルシューティングに関する情報を提供します。次の項目について説明します。

- 488 ページの「PPP のトラブルシューティングのためのツール」
- 491 ページの「PPP および PPPoE 関連の問題の解決」
- 503 ページの「専用回線の問題の解決」
- 504 ページの「認証の問題の診断と解決」

James Carlson による『*PPP Design, Implementation, and Debugging*』やオーストラリア国立大学の Web サイトなどの情報源も、PPP のトラブルシューティングに詳細なアドバイスを提供しています。詳細は、412 ページの「PPP に関する専門技術者向けのリファレンスブック」および 412 ページの「PPP に関する Web サイト」を参照してください。

PPP 問題の解決 (作業マップ)

次の作業マップを使用すれば、一般的な PPP の問題のためのアドバイスや解決方法をすばやく探すことができます。

表 21–1 PPP のトラブルシューティング (作業マップ)

作業	定義	参照先
PPP リンクに関する診断情報を取得します	PPP 診断ツールを使ってトラブルシューティングの出力を取得します。	489 ページの「pppd から診断情報を取得する方法」
PPP リンクのデバッグ情報を取得します	pppd debug コマンドを使ってトラブルシューティングの出力を生成します。	490 ページの「PPP デバッグをオンに設定する方法」

表 21–1 PPP のトラブルシューティング (作業マップ) (続き)

作業	定義	参照先
ネットワークレイヤーでの一般的な問題をトラブルシューティングします	一連の確認作業を行いネットワーク関連の PPP 問題を特定し解決します。	491 ページの「ネットワークの問題を診断する方法」
一般的な通信の問題をトラブルシューティングします	PPP リンクに影響を与える通信の問題を特定し解決します。	494 ページの「通信の問題を診断し解決する方法」
構成の問題をトラブルシューティングします	PPP 構成ファイルで問題を特定し解決します。	495 ページの「PPP 構成の問題を診断する方法」
モデム関連の問題をトラブルシューティングします	モデムの問題を特定し解決します。	496 ページの「モデムの問題を診断する方法」
chat スクリプト関連の問題をトラブルシューティングします	ダイアルアウトマシン上の chat スクリプトの問題を特定し解決します。	497 ページの「chat スクリプトのデバッグ情報を取得する方法」
シリアル回線の速度の問題をトラブルシューティングします	ダイアルインサーバー上で回線速度の問題を特定し解決します。	500 ページの「シリアル回線の速度の問題を診断して解決する方法」
専用回線の一般的な問題をトラブルシューティングします	専用回線のパフォーマンスの問題を特定し解決します。	503 ページの「専用回線の問題の解決」
認証に関連する問題をトラブルシューティングします	認証データベースに関連する問題を特定し解決します。	504 ページの「認証の問題の診断と解決」
PPPoE の問題領域をトラブルシューティングします	PPP 診断ツールを使用して、PPPoE の問題を特定し解決するための出力を得ます。	501 ページの「PPPoE の診断情報を取得する方法」

PPP のトラブルシューティングのためのツール

PPP リンクは、一般に次の 3 つの主要な領域で障害が発生します。

- 接続の確立に失敗する
- 通常の使用の中で接続パフォーマンスが低下する
- 接続のどちらかの側でネットワークに原因と考えられる問題が発生する

PPP が動作しているかどうかを確認するためのもっとも簡単な方法は、リンクを介したコマンドを実行することです。`ping` や `traceroute` などのコマンドをピアのネットワーク上のホストに対して実行し、結果を調べます。ただし、確立されている接続のパフォーマンスを監視したり、問題のある接続をトラブルシューティングしたりするには、PPP および UNIX のデバッグツールを使用してください。

この節では、`pppd` および関連するログファイルから診断情報を取得する方法について説明します。この章の残りの節では、PPP トラブルシューティングツールを使って発見し解決できる PPP に関する一般的な問題を説明します。

▼ pppd から診断情報を取得する方法

次に、ローカルマシン上の接続の現在の動作を表示する手順を説明します。

1 ローカルマシン上でスーパーユーザーになるか、同等の役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 PPP に設定されているシリアルデバイスを引数として pppd を実行します。

```
# pppd cua/b debug updetach
```

次に、pppd をフォアグラウンドで実行したときに表示されるダイアルアップリンクおよび専用回線リンクの診断結果の例を示します。バックグラウンドで pppd debug を実行すると、作成される出力は /etc/ppp/connect-errors ファイルに送られます。

例 21–1 正常に動作しているダイアルアップ接続からの出力

```
# pppd /dev/cua/b debug updetach
have route to 0.0.0.0/0.0.0.0 via 172.21.0.4
serial speed set to 230400 bps
Using interface sppp0
Connect: sppp0 <--> /dev/cua/b
sent [LCP ConfReq id=0x7b <asyncmap 0x0> <magic 0x73e981c8> <pcomp> <accomp>]
rcvd [LCP Ident id=0x79 magic=0x0 "ppp-2.4.0b1 (Sun Microsystems, Inc., Oct  6
    2004 09:36:22)"]
Peer Identification: ppp-2.4.0b1 (Sun Microsystems, Inc., Oct  6 2004 09:36:22)
    rcvd [LCP ConfRej id=0x7b <asyncmap 0x0>]
sent [LCP Ident id=0x7c magic=0x0 "ppp-2.4.0b1 (Sun Microsystems, Inc., Sep 15
    2004 09:38:33)"
sent [LCP ConfReq id=0x7d <magic 0x73e981c8> <pcomp> <accomp>]
rcvd [LCP ConfAck id=0x7d <magic 0x73e981c8> <pcomp> <accomp>]
rcvd [LCP ConfAck id=0x78 <magic 0xdd4ad820> <pcomp> <accomp>]
sent [LCP ConfAck id=0x78 <magic 0xdd4ad820> <pcomp> <accomp>]
sent [LCP Ident id=0x7e magic=0x73e981c8 "ppp-2.4.0b1 (Sun Microsystems, Inc.,
    Sep 15 2004 09:38:33)"]
sent [IPCP ConfReq id=0x3d <addr 0.0.0.0> <compress VJ 0f 01>]
rcvd [LCP Ident id=0x7a magic=0xdd4ad820 "ppp-2.4.0b1 (Sun Microsystems, Inc.,
    Oct  6 2004 09:36:22)"]
Peer Identification: ppp-2.4.0b1 (Sun Microsystems, Inc., Oct  6 2004 09:36:22)
rcvd [IPCP ConfReq id=0x92 <addr 10.0.0.1> <compress VJ 0f 01>
sent [IPCP ConfAck id=0x92 <addr 10.0.0.1> <compress VJ 0f 01>
rcvd [IPCP ConfNak id=0x3d <addr 10.0.0.2>]]
sent [IPCP ConfReq id=0x3e <addr 10.0.0.2> <compress VJ 0f 01>]
rcvd [IPCP ConfAck id=0x3e <addr 10.0.0.2> <compress VJ 0f 01>]
local  IP address 10.0.0.2
remote IP address 10.0.0.1
```

例 21–2 正常に動作している専用回線リンクからの出力

```
# pppd /dev/se_hdlc1 default-asyncmap debug updetach
pppd 2.4.0b1 (Sun Microsystems, Inc., Oct 24 2004 07:13:18) started by root, uid 0
synchronous speed appears to be 0 bps
```

```
init option: '/etc/ppp/peers/syncinit.sh' started (pid 105122)
Serial port initialized.
synchronous speed appears to be 64000 bps
Using interface sppp0
Connect: sppp0 <--> /dev/se_hdlc1
sent [LCP ConfReq id=0xe9 <magic 0x474283c6><pcomp> <accomp>]
rcvd [LCP ConfAck id=0xe9 <magic 0x474283c6><pcomp> <accomp>]
rcvd [LCP ConfReq id=0x22 <magic 0x8e3a53ff><pcomp> <accomp>]
sent [LCP ConfReq id=0x22 <magic 0x8e3a53ff><pcomp> <accomp>]
sent [LCP Ident id=0xea magic=0x474283c6 "ppp-2.4.0b1 (Sun Microsystems, Inc., Oct
    22 2004 14:31:44)"]
sent [IPCP ConfReq id=0xf7 <addr 0.0.0.0> <compress VJ 0f o1>]]
sent [CCP ConfReq id=0x3f <deflate 15> <deflate(old#) 15> <bsd v1 15>]
rcvd [LCP Ident id=0x23 magic=0x8e3a53ff "ppp-2.4.0b1 (Sun Microsystems, Inc., Oct
    22 2004 14:31:44)"]
Peer Identification: ppp-2.4.0b1 (Sun Microsystems, Inc., Oct 22 2004 14:31:44)
rcvd [IPCP ConfReq id=0x25 <addr 10.0.0.1> <compress VJ 0f 01>]
sent [IPCP ConfAck id=0x25 <addr 10.0.0.1> <compress VJ 0f 01>]
rcvd [CCP ConfReq id=0x3 <deflate 15> <deflate(old#) 15 <bsd v1 15>]
sent [CCP ConfAck id=0x3 <deflate 15> <deflate(old#) 15 <bsd v1 15>]
rcvd [IPCP ConfNak id=0xf8 <addr 10.0.0.2>]
rcvd [IPCP ConfReq id=0xf7 <addr 10.0.0.2> <compress VJ 0f 01>]
rcvd [CCP ConfAck id=0x3f <deflate 15> <deflate(old#) 15 <bsd v1 15>]
Deflate (15) compression enabled
rcvd [IPCP ConfAck id=0xf8 <addr 10.0.0.2> <compress VJ 0f 01>]
local  IP address 10.0.0.2
remote IP address 10.0.0.1
```

▼ PPP デバッグをオンに設定する方法

次に、pppd コマンドを使ってデバッグ情報を取得する方法を示します。

注 – 手順 1 から手順 3 までは各ホストごとに 1 度実行するだけでかまいません。その後、手順 4 に進んでホストのデバッグをオンに設定できます。

---

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 pppd からの出力を保持するためのログファイルを作成します。

```
# touch /var/log/pppdebug
```

3 次の pppd 用の syslog 機能を /etc/syslog.conf に追加します。

```
daemon.debug;local2.debug       /var/log/pppdebug
```

4 syslogd を再起動します。

```
# pkill -HUP -x syslogd
```

5 pppd の次の構文を使用して、特定のピアに対する呼び出しのデバッグをオンに設定します。

```
# pppd debug call peer-name
```

peer-name は、`/etc/ppp/peers` ディレクトリにあるファイル名でなければなりません。

6 ログファイルの内容を表示します。

```
# tail -f /var/log/pppdebug
```

ログファイルの例については、手順 3 を参照してください。

PPP および PPPoE 関連の問題の解決

PPP 関連の問題と PPPoE 関連の問題を解決する方法については、次の節を参照してください。

▼ ネットワークの問題を診断する方法

PPP リンクがアクティブになったにもかかわらずリモートネットワーク上のほとんどのホストに到達できないという場合は、ネットワーク問題が見つかる可能性があります。ここでは、PPP リンクに影響を与えるネットワーク障害を特定し、解決する方法を示します。

1 ローカルマシン上でスーパーユーザーになるか、同等の役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 問題のある接続を切断します。

3 次のオプションを **PPP** 構成に追加して、構成ファイルのオプションのプロトコルを無効にします。

```
noccp novj nopcomp noaccomp default-asyncmap
```

このオプションは、もっとも単純で圧縮を行わない PPP を使用可能にします。コマンド行でこれらのオプションを引数として `pppd` を実行してみます。これまで接続できなかったホストに接続できれば、次のいずれかの位置にオプションを追加します。

- `/etc/ppp/peers/`*peer-name*、`call` オプションのあと
- `/etc/ppp/options`、オプションを広域的に適用する場合

4 リモートピアを呼び出します。次に、デバッグをオンに設定します。

```
% pppd debug call peer-name
```

5 `chat` の `-v` オプションを使用して、**chat** プログラムから冗長ログを取得します。

たとえば、PPP 構成ファイルで次の形式を使用します。

```
connect 'chat -v -f /etc/ppp/chatfile'
```

/etc/ppp/chatfile は、お使いの chat ファイルの名前を表します。

6 **Telnet** またはほかのアプリケーションを使ってリモートホストに接続し、問題を再度発生させてみます。

デバッグログを調べます。これでもリモートホストに接続できない場合は、PPP の問題はネットワークに関連している可能性があります。

7 リモートホストの **IP** アドレスが登録されているインターネットアドレスであることを確認します。

組織によっては、ローカルネットワーク内では通用するが、インターネットへは経路指定できない内部 IP アドレスを割り当てる場合があります。リモートホストが社内にある場合、インターネットに接続するためには、管理者は、NAT (名前 - アドレス変換) またはプロキシサーバーを設定する必要があります。リモートホストが社内にない場合は、遠隔組織に問題を報告する必要があります。

8 経路指定テーブルを調べます。

a. ローカルマシンとピアの両方で経路指定テーブルを確認します。

b. 経路指定テーブルで、ピアからリモートシステムへのパスにあるルーターをすべて確認します。また、リモートシステムからピアへの戻りのパスにあるルーターもすべて確認します。

中間ルーターの設定が間違っていないことを確認します。ピアへの戻りのパスに問題が見つかることがしばしばあります。

9 (省略可能) マシンがルーターである場合、オプションの機能を確認します。

```
# ndd -set /dev/ip ip_forwarding 1
```

ndd の詳細は、ndd(1M) のマニュアルページを参照してください。

Solaris 10 リリースでは、ndd(1M) ではなく routeadm(1M) を利用できます。

```
# routeadm -e ipv4-forwarding -u
```

注 – ndd コマンドに持続性はありません。このコマンドに設定された値は、システムのリブート時に消失します。routeadm コマンドは持続します。このコマンドに設定された値は、システムのリブート後も保持されます。

10 netstat -s および同様のツールから取得した統計を確認します。

netstat の詳細は、netstat(1M) のマニュアルページを参照してください。

a. ローカルマシン上で統計を実行します。

b. ピアを呼び出します。

c. netstat -s によって生成された新しい統計を調べます。詳細は、493 ページの「PPP に影響を与える一般的なネットワークの問題」を参照してください。

11 DNS 構成を確認します。

ネームサービス構成に問題があると、IP アドレスを解釈処理できないため、アプリケーションは障害を発生します。

PPP に影響を与える一般的なネットワークの問題

netstat -s によって生成されたメッセージを使用すると、次の表に示したネットワークの問題を解決できます。関連する作業情報として、491 ページの「ネットワークの問題を診断する方法」を参照してください。

表 21–2 PPP に影響を与える一般的なネットワークの問題

メッセージ	問題	解決方法
IP packets not forwardable	ローカルホストで送信経路が見つからない	ローカルホストの経路指定テーブルに欠如している送信経路を追加する
ICMP input destination unreachable	ローカルホストで送信経路が見つからない	ローカルホストの経路指定テーブルに欠如している送信経路を追加する

表 21–2　PPP に影響を与える一般的なネットワークの問題　(続き)

メッセージ	問題	解決方法
ICMP time exceeded	2 つのルーターが同じ着信アドレスを互いに送信し、パケットが互いに何度も往復し、TTL (存続時間) の値を超過した	traceroute を使ってルーティングループの源を見つけ、エラーになっているルーターの管理者に連絡する。traceroute の詳細は、traceroute(1M) のマニュアルページを参照
IP packets not forwardable	ローカルホストで送信経路が見つからない	ローカルホストの経路指定テーブルに欠如している送信経路を追加する
ICMP input destination unreachable	ローカルホストで送信経路が見つからない	ローカルホストの経路指定テーブルに欠如している送信経路を追加する

▼ 通信の問題を診断し解決する方法

通信の問題は、2 つのピアがリンクを正常に確立できない場合に発生します。これらは、chat スクリプトが不正に設定されているために起きるネゴシエーション問題であることもあります。ここでは、通信の問題を解決する方法を示します。誤りのある chat スクリプトによって発生するネゴシエーション問題を解決する方法については、表 21–5 を参照してください。

1. ローカルマシン上でスーパーユーザーになるか、同等の役割になります。

   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2. ピアを呼び出します。

3. リモートピアを呼び出します。次に、デバッグをオンに設定します。

   ```
   % pppd debug call peer-name
   ```

   通信の問題によっては、問題解決のためにピアからデバッグ情報を取得する必要がある場合があります。

4. 生成されたログをチェックし、通信の問題が報告されていないかを確認します。詳細は、494 ページの「PPP に影響を与える一般的な通信の問題」を参照してください。

PPP に影響を与える一般的な通信の問題

次の表は、494 ページの「通信の問題を診断し解決する方法」の作業で出力されるログに関連する症状を説明したものです。

表 21–3　PPP に影響を与える一般的な通信の問題

症状	問題	解決方法
`too many Configure-Requests` メッセージ	あるピアがほかのピアを認識できません。	次の問題を確認します。 ■ マシンまたはモデムの配線が間違っていないか。 ■ モデムの構成に不適切なビット設定がないか、あるいは間違ったフロー制御がないか。 ■ chat スクリプトが誤っていないか。この場合は、表 21–5 を参照してください。
`pppd debug` の出力は LCP が起動していることを示しているが、より上位のプロトコルが失敗したか、あるいは CRC エラーを示している	非同期制御文字マップ (ACCM) が正しく設定されていません。	`default-async` オプションを使用して ACCM を標準のデフォルトである FFFFFFFF に設定します。まずコマンド行で `pppd` のオプションとして `default-async` を使用します。問題が解決したら、`default-async` を `/etc/ppp/options` または call オプションのあとの `/etc/ppp/peers/`*peer-name* に追加します。
`pppd debug` の出力は IPCP が起動していることを示しているが、すぐに終了してしまう	IP アドレスの設定が間違っている可能性があります。	1. 間違った IP アドレスがないか確認するために、chat スクリプトを調べます。 2. chat スクリプトに誤りがない場合は、ピアのデバッグログを要求し、ピアのログで IP アドレスを確認します。
接続のパフォーマンスが非常に低い	フロー制御構成のエラー、モデム設定のエラー、不適切に設定された DTE レートなどにより、モデムが適切に構成されていない可能性があります。	モデム構成を確認し、適宜調整します。

▼ PPP 構成の問題を診断する方法

PPP の問題には、PPP 構成ファイルの問題が原因となっているものがあります。ここでは、一般的な構成問題を特定し、解決する方法を示します。

1 ローカルマシン上でスーパーユーザーになるか、同等の役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 リモートピアを呼び出します。次に、デバッグをオンに設定します。

```
% pppd debug call peer-name
```

3 生成されたログをチェックし、構成問題が報告されていないかを確認します。詳細は、**496 ページの「一般的な PPP 構成の問題」**を参照してください。

一般的な PPP 構成の問題

次の表は、495 ページの「PPP 構成の問題を診断する方法」の作業で出力されるログに関連する症状を説明したものです。

表 21–4 一般的な PPP 構成の問題

症状	問題	解決方法
pppd debug 出力に、「Could not determine remote IP address」というエラーメッセージが含まれる	/etc/ppp/peers/*peer-name* ファイルにそのピアの IP アドレスが存在しない。ピアが、接続ネゴシエーション時に IP アドレスを提供しない	次の形式を使用して、pppd コマンド行、あるいは /etc/ppp/peers/*peer-name* でピアの IP アドレスを指定する :10.0.0.10
pppd debug の出力が CCP データ圧縮が失敗したことを示す。出力には接続が解除されたことも表示する	ピアの PPP 圧縮設定が衝突している可能性がある	ピアの 1 つで /etc/ppp/options に noccp オプションを追加して CCP 圧縮を無効にする

▼ モデムの問題を診断する方法

モデムは、ダイアルアップリンクで問題の発生しやすい領域です。モデム構成でもっともよく発生する問題は、ピアからの応答がないことです。しかし、接続の問題の原因が本当にモデム構成の問題なのかどうかを判定することは難しい場合があります。

モデムの基本的なトラブルシューティングに関するヒントは、『Solaris のシステム管理 (上級編)』の「端末とモデムの問題を解決する方法」を参照してください。モデムメーカーのマニュアルや Web サイトは、特定の装置に関する問題の解決に役立ちます。次の手順は、問題のあるモデム構成が接続の問題の原因となっているかどうかを判定するのに役立ちます。

1. 490 ページの「PPP デバッグをオンに設定する方法」で説明した手順で、デバッグをオンに設定してピアを呼び出します。
2. 作成された /var/log/pppdebug ログを表示し、モデム構成に問題がないかを確認します。
3. ping を使用してさまざまなサイズのパケットを接続上に送信します。

   ping の詳細は、ping(1M) のマニュアルページを参照してください。

   小さいパケットは受信されるが、大きいパケットはドロップされる場合、モデムに問題があることを示します。

4 インタフェース sppp0 上のエラーを確認します。

```
% netstat -ni
Name  Mtu  Net/Dest   Address      Ipkts    Ierrs Opkts    Oerrs Collis Queue
lo0   8232 127.0.0.0  127.0.0.1    826808   0     826808   0     0      0
hme0  1500 172.21.0.0 172.21.3.228 13800032 0     1648464  0     0      0
sppp0 1500 10.0.0.2   10.0.0.1     210      0     128      0     0      0
```

インタフェースのエラーが時間がたつにつれて増えている場合は、モデム構成に問題がある可能性があります。

注意事項 作成された /var/log/pppdebug ログの表示で次の症状が認められる場合は、モデムの構成に問題がある可能性があります。ローカルマシンはピアを認識できますが、ピアはローカルマシンを認識できません。

- ピアから「recvd」メッセージが返されない。
- 出力にピアからの LCP メッセージが含まれるが、接続は失敗し、ローカルマシンから「too many LCP Configure Requests」のメッセージが送信される。
- 接続が SIGHUP 信号で終了する。

▼ chat スクリプトのデバッグ情報を取得する方法

次の操作は、chat のデバッグ情報を表示して一般的な問題の解決方法を知る手段として行なってください。詳細は、498 ページの「chat スクリプトの一般的な問題」を参照してください。

1 ダイアルアウトマシン上のスーパーユーザー、またはそれと同等の役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 /etc/ppp/peers/*peer-name* ファイルを編集してピアが呼び出されるようにします。

3 connect オプションで指定されている chat コマンドに引数として -v を追加します。

```
connect "/usr/bin/chat -v -f /etc/ppp/chat-script-name"
```

4 /etc/ppp/connect-errors ファイルの **chat** スクリプトのエラーを表示します。

次は、chat で発生する主なエラーです。

```
Oct 31 08:57:13 deino chat[107294]: [ID 702911 local2.info] expect (CONNECT)
Oct 31 08:57:58 deino chat[107294]: [ID 702911 local2.info] alarm
Oct 31 08:57:58 deino chat[107294]: [ID 702911 local2.info] Failed
```

この例は、(CONNECT) 文字列を待つ間にタイムアウトしたことを示します。chat が失敗すると、pppd から次のメッセージを受け取ります。

```
Connect script failed
```

chat スクリプトの一般的な問題

chat スクリプトは、ダイアルアップリンクにおいてもっとも問題が発生しやすい領域です。次の表に、chat スクリプトの一般的なエラーと、エラー解決のためのヒントを示します。操作方法については、497 ページの「chat スクリプトのデバッグ情報を取得する方法」を参照してください。

表 21–5 chat スクリプトの一般的な問題

症状	問題	解決方法
pppd debug の出力に Connect script failed が含まれる	chat スクリプトは、次のようにユーザー名とパスワードを指定している ogin: *user-name* ssword: *password* しかし、接続しようとしたピアはこの情報を要求していない	1. chat スクリプトからログインとパスワードを削除する 2. 再度ピアを呼び出してみる 3. まだメッセージが表示される場合は、ISP に連絡して正しいログインシーケンスを問い合わせる
/usr/bin/chat -v ログにメッセージ "expect (login:)" alarm read timed out が含まれる	chat スクリプトは、次のようにユーザー名とパスワードを指定している ogin: pppuser ssword: \q\U しかし、接続しようとしているピアはこの情報を要求していない	1. chat スクリプトからログインとパスワードを削除する 2. 再度ピアを呼び出してみる 3. まだメッセージが表示される場合は、ISP に連絡して正しいログインシーケンスを問い合わせる
pppd debug の出力にpossibly looped-back が含まれる	ローカルマシンまたはそのピアがコマンド行で停止していて PPP を実行していない。chat スクリプト内に間違って設定されたログイン名とパスワードがある	1. chat スクリプトからログインとパスワードを削除する 2. 再度ピアを呼び出してみる 3. まだメッセージが表示される場合は、ISP に連絡して正しいログインシーケンスを問い合わせる

表 21–5 chat スクリプトの一般的な問題 (続き)

症状	問題	解決方法
pppd debug 出力は LCP が起動していることを示しているが、接続がすぐに終了してしまう	chat スクリプト内のパスワードが間違っている可能性がある	1. ローカルマシンの正しいパスワードを確認する 2. chat スクリプト内のパスワードを確認する。間違っている場合は修正する 3. 再度ピアを呼び出してみる 4. まだメッセージが表示される場合は、ISP に連絡して正しいログインシーケンスを問い合わせる

ピアからのテキストがチルダ (~) で始まる	chat スクリプトは、次のようにユーザー名とパスワードを指定している
	`ogin: pppuser`
	`ssword: \q\U`
	しかし、接続しようとしているピアはこの情報を要求していない
		1. chat スクリプトからログインとパスワードを削除する
		2. 再度ピアを呼び出してみる
		3. まだメッセージが表示される場合は、ISP に連絡して正しいログインシーケンスを問い合わせる

モデムが停止する	chat スクリプトに次の行が含まれており、ローカルマシンがピアからの CONNECT メッセージを待つように強制している `CONNECT ”`	chat スクリプトがピアからの CONNECT を待つようにするときは、次の行を使用する `CONNECT \c` chat スクリプトを `~\c` で終了する
pppd debug の出力に `LCP: timeout sending Config-Requests` が含まれる	chat スクリプトに次の行が含まれており、ローカルマシンがピアからの CONNECT メッセージを待つように強制している `CONNECT ”`	chat スクリプトがピアからの CONNECT を待つようにするときは、次の行を使用する `CONNECT \c` chat スクリプトを `~\c` で終了する
pppd debug 出力に `Serial link is not 8-bit clean` が含まれる	chat スクリプトに次の行が含まれており、ローカルマシンがピアからの CONNECT メッセージを待つように強制している `CONNECT ”`	chat スクリプトがピアからの CONNECT を待つようにするときは、次の行を使用する `CONNECT \c` chat スクリプトを `~\c` で終了する
pppd debug の出力に `Loopback detected` が含まれる	chat スクリプトに次の行が含まれており、ローカルマシンがピアからの CONNECT メッセージを待つように強制している `CONNECT ”`	chat スクリプトがピアからの CONNECT を待つようにするときは、次の行を使用する `CONNECT \c` chat スクリプトを `~\c` で終了する

表 21–5 chat スクリプトの一般的な問題 (続き)

症状	問題	解決方法
pppd debug の出力に SIGHUP が含まれる	chat スクリプトに次の行が含まれており、ローカルマシンがピアからの CONNECT メッセージを待つように強制している `CONNECT ''`	chat スクリプトがピアからの CONNECT を待つようにするときは、次の行を使用する `CONNECT \c` chat スクリプトを ~\c で終了する

▼ シリアル回線の速度の問題を診断して解決する方法

ダイアルインサーバーは、速度の設定の衝突が原因で問題が発生する可能性があります。次に示す手順は、接続の問題の原因がシリアル回線速度の衝突であることを特定するのに役立ちます。

速度の問題は、次のような原因で発生します。

- /bin/login のようなプログラムを介して PPP を起動し、回線の速度を指定した
- PPP を mgetty から起動し、誤ってビットレートを指定した

pppd は、はじめは回線に設定されていた速度を /bin/login または mgetty によって設定された速度に変更します。このことが回線の障害を発生させます。

1 ダイアルインサーバーにログインします。デバッグをオンに設定してピアを呼び出します。

手順については、490 ページの「PPP デバッグをオンに設定する方法」を参照してください。

2 作成された /var/log/pppdebug ログを表示します。

出力に次のメッセージがないか確認します。

```
LCP too many configure requests
```

このメッセージは、PPP に設定されているシリアル回線の速度が衝突している可能性があることを示します。

3 **PPP** が /bin/login のようなプログラムを介して起動されているかどうかを調べ、設定されている回線速度を調べます。

このような状況では、pppd はもともと設定されていた回線速度を /bin/login で指定されている速度に変更します。

4 ユーザーが **PPP** を mgetty コマンドから起動し、誤ってビットレートを指定していないかどうか確認します。

この処理もまた、シリアル回線速度の衝突を引き起こします。

5 次のようにしてシリアル回線速度の衝突の問題を解決します。

a. モデムの **DTE** レートをロックします。

b. **autobaud** を使用しないようにします。

c. 設定後に回線速度を変更しないようにします。

▼ PPPoE の診断情報を取得する方法

PPP および標準の UNIX ユーティリティーを使用して PPPoE の問題を特定できます。接続上の問題の原因が PPPoE だと思われるとき、次の診断ツールを使ってトラブルシューティング情報を取得できます。

1 **PPPoE** トンネルを実行しているマシン、つまり **PPPoE** クライアントまたは **PPPoE** アクセスサーバーでスーパーユーザーになります。

2 490 ページの「**PPP** デバッグをオンに設定する方法」で説明した手順で、デバッグをオンに設定します。

3 ログファイル `/var/log/pppdebug` の内容を表示します。

次の例は、PPPoE トンネルとの接続で生成されたログファイルの一部です。

```
Sep  6 16:28:45 enyo pppd[100563]: [ID 702911 daemon.info] Plugin
  pppoe.so loaded.
Sep  6 16:28:45 enyo pppd[100563]: [ID 860527 daemon.notice] pppd
  2.4.0b1 (Sun Microsystems, Inc.,
Sep  5 2001 10:42:05) started by troot, uid 0
Sep  6 16:28:46 enyo pppd[100563]: [ID 702911 daemon.debug] connect option:
   '/usr/lib/inet/pppoec
-v hme0' started (pid 100564)
Sep  6 16:28:46 enyo pppd[100563]: [ID 702911 daemon.info] Serial connection established.
Sep  6 16:28:46 enyo pppd[100563]: [ID 702911 daemon.info] Using interface sppp0
Sep  6 16:28:46 enyo pppd[100563]: [ID 702911 daemon.notice] Connect: sppp0
   <--> /dev/sppptun
Sep  6 16:28:46 enyo pppd[100563]: [ID 702911 daemon.debug] /etc/ppp/pap-secrets
  is apparently empty
Sep  6 16:28:46 enyo pppd[100563]: [ID 702911 daemon.debug] /etc/ppp/chap-secrets
  is apparently empty
Sep  6 16:28:46 enyo pppd[100563]: [ID 702911 daemon.debug] sent
  [LCP ConfReq id=0xef <mru 1492>
asyncmap 0x0 <magic 0x77d3e953><pcomp><acomp>
Sep  6 16:28:46 enyo pppd[100563]: [ID 702911 daemon.debug] rcvd
  [LCP ConfReq id=0x2a <mru 1402>
asyncmap 0x0 <magic 0x9985f048><pcomp><acomp
```

デバッグの出力によって問題を特定できない場合は、次の手順に進みます。

4 **PPPoE** から診断メッセージを取得します。

```
# pppd connect "/usr/lib/inet/pppoec -v interface-name"
```

pppoec は、診断情報を stderr に送信します。pppd をフォアグラウンドで実行する場合、出力が画面に表示されます。pppd をバックグラウンドで実行する場合、出力は /etc/ppp/connect-errors に送られます。

次の例は、PPPoE トンネルがネゴシエートされたときに生成されるメッセージです。

```
Connect option: '/usr/lib/inet/pppoec -v hme0' started (pid 100564)
/usr/lib/inet/pppoec: PPPoE Event Open (1) in state Dead (0): action SendPADI (2)
/usr/lib/inet/pppoec: Sending PADI to ff:ff:ff:ff:ff:ff: 18 bytes
/usr/lib/inet/pppoec: PPPoE State change Dead (0) -> InitSent (1)
/usr/lib/inet/pppoec: Received Active Discovery Offer from 8:0:20:cd:c1:2/hme0:pppoed
/usr/lib/inet/pppoec: PPPoE Event rPADO+ (5) in state InitSent (1): action SendPADR+ (5)
/usr/lib/inet/pppoec: Sending PADR to 8:0:20:cd:c1:2: 22 bytes
/usr/lib/inet/pppoec: PPPoE State change InitSent (1) -> ReqSent (3)
/usr/lib/inet/pppoec: Received Active Discovery Session-confirmation from
   8:0:20:cd:c1:2/hme0:pppoed
/usr/lib/inet/pppoec: PPPoE Event rPADS (7) in state ReqSent (3): action Open (7)
/usr/lib/inet/pppoec: Connection open; session 0002 on hme0:pppoe
/usr/lib/inet/pppoec: PPPoE State change ReqSent (3) -> Convers (4)
/usr/lib/inet/pppoec: connected
```

診断メッセージによって問題を特定できない場合は、次の手順に進みます。

5 snoop を実行します。次にトレースをファイルに保存します。

snoop の詳細は、snoop(1m) のマニュアルページを参照してください。

```
# snoop -o pppoe-trace-file
```

6 snoop トレースファイルを表示します。

```
# snoop -i pppoe-trace-file -v pppoe

ETHER: ----- Ether Header -----
ETHER:
ETHER: Packet 1 arrived at 6:35:2.77
ETHER: Packet size = 32 bytes
ETHER: Destination = ff:ff:ff:ff:ff:ff, (broadcast)
ETHER: Source      = 8:0:20:78:f3:7c, Sun
ETHER: Ethertype = 8863 (PPPoE Discovery)
ETHER:
PPPoE: ----- PPP Over Ethernet -----
PPPoE:
PPPoE: Version = 1
PPPoE: Type = 1
PPPoE: Code = 9 (Active Discovery Initiation)
PPPoE: Session Id = 0
PPPoE: Length = 12 bytes
PPPoE:
PPPoE: ----- Service-Name -----
PPPoE: Tag Type = 257
PPPoE: Tag Length = 0 bytes
PPPoE:
PPPoE: ----- Host-Uniq -----
PPPoE: Tag Type = 259
PPPoE: Tag Length = 4 bytes
```

```
PPPoE: Data = Ox00000002
PPPoE:
.
.
.
ETHER: ----- Ether Header -----
ETHER:
ETHER: Packet 5 arrived at 6:35:2.87
ETHER: Packet size = 60 bytes
ETHER: Destination = 8:0:20:78:f3:7c, Sun)
ETHER: Source      = 0:2:fd:39:7f:7,
ETHER: Ethertype = 8864 (PPPoE Session)
ETHER:
PPPoE: ----- PPP Over Ethernet -----
PPPoE:
PPPoE: Version = 1
PPPoE: Type = 1
PPPoE: Code = 0 (PPPoE Session)
PPPoE: Session Id = 24383
PPPoE: Length = 20 bytes
PPPoE:
PPP: ----- Point-to-Point Protocol -----
PPP:
PPP-LCP: ----- Link Control Protocol -----
PPP-LCP:
PPP-LCP: Code = 1 (Configure Request)
PPP-LCP: Identifier = 80
PPP-LCP: Length = 18
```

専用回線の問題の解決

専用回線でもっとも一般的な問題は、パフォーマンスの低下です。ほとんどの場合、問題を解決するためには、電話会社に相談する必要があります。

表 21–6 一般的な専用回線の問題

症状	問題	解決方法
接続が開始しない	CSU BPV (CSU 極性違反) が原因の可能性があります。接続の一方の側が AMI 回線用に設定されており、もう一方の側が ESF の B8ZS (Bit-8 Zero Substitute) 用に設定されています。	米国またはカナダのユーザーは、この問題を CSU/DSU のメニューから直接解決できます。詳細は、CSU/DSU メーカーのマニュアルを参照してください。 その他の地域のユーザーは、プロバイダが CSU BPV の解決策を用意している可能性があります。

表 21–6 一般的な専用回線の問題 *(続き)*

症状	問題	解決方法
接続のパフォーマンスが非常に低い	接続上でトラフィックが持続しているときに、pppd debug の出力が CRC エラーを示します。回線に、電話会社とネットワークの間の誤った設定によって生じた刻時の問題がある可能性があります。	電話会社に連絡し、「ループ刻時」を使用していたことを確認します。 構造化されていない専用回線では、刻時を提供する必要がある場合があります。北米のユーザーはループクロックを使用するようにしてください。

認証の問題の診断と解決

次の表は、一般的な認証問題について説明したものです。

表 21–7 一般的な認証の問題

症状	問題	解決方法
pppd debug の出力が「Peer is not authorized to use remote address *address*」というメッセージを示す	PAP 認証を使用しており、リモートピアの IP アドレスが /etc/ppp/pap-secrets ファイルに存在しない	/etc/ppp/pap-secrets ファイルで、ピアのエントリのあとにアスタリスク (*) を追加する
pppd debug の出力は LCP が起動していることを示しているが、その直後に終了してしまう	特定のセキュリティープロトコルのデータベースでパスワードが間違っている可能性がある	/etc/ppp/pap-secrets または /etc/ppp/chap-secrets ファイルでピアのパスワードを確認する

◆ ◆ ◆ 第 22 章

Solaris PPP 4.0 (リファレンス)

この章では、Solaris PPP 4.0 について詳細で概念的な情報を提供します。トピックは次のとおりです。

ファイルおよびコマンド行での PPP オプションの使用

Solaris PPP 4.0 には、PPP 構成の定義に使用するオプションが多数含まれます。これらのオプションは、PPP 構成ファイルまたはコマンド行で使用するほか、ファイルでの使用とコマンド行での使用を組み合わせることもできます。この節では、PPP オプションの構成ファイルでの使用と PPP コマンドの引数としての使用について詳細に説明します。

PPP オプションを定義する場所

Solaris PPP 4.0 は柔軟に構成できます。PPP オプションを次の場所で定義できます。

- PPP 構成ファイル
- コマンド行で実行される PPP コマンド
- 前記 2 つの場所の組み合わせ

次の表に、PPP 構成ファイルとコマンドを一覧表示します。

表 22–1 PPP 構成ファイルとコマンドの概要

ファイルまたはコマンド	定義	参照先
`/etc/ppp/options`	たとえば、マシンがピアにピア自身の認証を要求するかどうかなど、システム上のすべての PPP リンクにデフォルトで適用される特性を含むファイル。このファイルがない場合、スーパーユーザー以外のユーザーは PPP の使用を禁止されます。	509 ページの「`/etc/ppp/options` 構成ファイル」
`/etc/ppp/options.`*ttyname*	シリアルポート *ttyname* 上のすべての通信の特性を記述するファイル。	511 ページの「`/etc/ppp/options.`*ttyname* 構成ファイル」
`/etc/ppp/peers`	通常、ダイアルアウトマシンが接続するピアに関する情報を含むディレクトリ。このディレクトリ内のファイルは、`pppd` コマンドの `call` オプションで使用されます。	514 ページの「ダイアルインサーバーと通信するための情報の指定」
`/etc/ppp/peers/`*peer-name*	リモートピア *peer-name* の特性を含むファイル。通常、リモートピアの電話番号やピアとの接続をネゴシエートするための chat スクリプトなどの特性が含まれます。	514 ページの「`/etc/ppp/peers/`*peer-name* ファイル」
`/etc/ppp/pap-secrets`	パスワード認証プロトコル (PAP) の認証に必要なセキュリティー資格を含むファイル。	528 ページの「`/etc/ppp/pap-secrets` ファイル」
`/etc/ppp/chap-secrets`	チャレンジハンドシェーク認証プロトコル (CHAP) の認証に必要なセキュリティー資格を含むファイル。	532 ページの「`/etc/ppp/chap-secrets` ファイル」
`~/.ppprc`	PPP ユーザーのホームディレクトリ内のファイル。ダイアルインサーバーでもっともよく使用されます。このファイルには、各ユーザーの構成に関する特定の情報が含まれます。	513 ページの「ダイアルインサーバーでの `$HOME/.ppprc` の設定」
`pppd` *options*	PPP リンクの開始および PPP リンクの特性の説明のためのコマンドとオプション。	506 ページの「PPP オプションの処理方法」

PPP ファイルの詳細は、pppd(1M) のマニュアルページを参照してください。`pppd(1M)` には、`pppd` で使用できるすべてのオプションに関する詳細な説明もあります。すべての PPP 構成ファイルのサンプルテンプレートは、`/etc/ppp` にあります。

PPP オプションの処理方法

1. `pppd` デーモンが次を構文解析する。

Solaris PPP 4.0 のすべての操作は、ユーザーが pppd コマンドを実行すると起動する pppd デーモンによって処理されます。ユーザーがリモートピアを呼び出すと、次が発生します。

- /etc/ppp/options
- $HOME/.ppprc
- /etc/ppp/options または $HOME/.ppprc の中で file または call オプションによって開かれたファイル

2. pppd がコマンド行を走査して使用中のデバイスを判定する。デーモンはまだ遭遇したオプションを解釈しない。
3. pppd は次の条件に基づいて使用するシリアルデバイスを検出しようとする。
   - シリアルデバイスがコマンド行またはそれ以前に処理した構成ファイルで指定されている場合、pppd はそのデバイス名を使用します。
   - シリアルデバイスが指定されていない場合、pppd はコマンド行で notty、pty、または socket オプションを検索します。これらのオプションが指定されている場合、pppd はデバイス名が存在しないとみなします。
   - 上記以外の場合で、標準入力が tty に接続されていることを pppd が検出した場合は、tty の名前を使用します。
   - それでも pppd がシリアルデバイスを見つけられない場合は、接続を終了し、エラーを発生させます。
4. pppd は次に /etc/ppp/options.*ttyname* ファイルが存在するかどうかをチェックする。ファイルが見つかると、pppd はそのファイルを構文解析する。
5. pppd はコマンド行のオプションを処理する。
6. pppd はリンク制御プロトコル (LCP) のネゴシエーションを行い、接続を確立する。
7. (省略可能) 認証が必要な場合、pppd は、/etc/ppp/pap-secrets または /etc/ppp/chap-secrets を読み取り、反対側のピアを認証する。

pppd デーモンがコマンド行またはほかの構成ファイルで call *peer-name* オプションを検出すると、/etc/ppp/peers/*peer-name* ファイルが読み取られます。

PPP 構成ファイルにおける特権のしくみ

Solaris PPP 4.0 構成には特権の概念が含まれます。特権は、特に、同じオプションが複数の場所で呼び出された時に、構成オプションの優先度を判定します。特権ソースから呼び出されたオプションは、非特権ソースから呼び出された同じオプションよりも優先されます。

ユーザー特権

唯一の特権ユーザーは、UID の値が 0 のスーパーユーザー (root) です。その他のすべてのユーザーは特権を与えられません。

ファイル特権

次に、所有者にかかわらず特権を与えられる構成ファイルを示します。

- /etc/ppp/options
- /etc/ppp/options.*ttyname*
- /etc/ppp/peers/*peer-name*

$HOME/.ppprc は、ユーザーが所有するファイルです。$HOME/.ppprc およびコマンド行から読み取られたオプションは、pppd を起動しているユーザーが root である場合にだけ特権が与えられます。

file オプションの引数は特権が与えられます。

オプション特権の意味

オプションの中には、呼び出したユーザーまたはソースが特権を与えられていないと動作しないものがあります。コマンド行で呼び出されたオプションは、pppd コマンドを実行中のユーザーの特権を割り当てられます。これらのオプションは、pppd を起動しているユーザーが root でなければ、特権が与えられません。

オプション	状態	意味
ドメイン	特権がある	使用には特権が必要です。
linkname	特権がある	使用には特権が必要です。
noauth	特権がある	使用には特権が必要です。
nopam	特権がある	使用には特権が必要です。
pam	特権がある	使用には特権が必要です。
plugin	特権がある	使用には特権が必要です。
privgroup	特権がある	使用には特権が必要です。
allow-ip *addresses*	特権がある	使用には特権が必要です。
name *hostname*	特権がある	使用には特権が必要です。
plink	特権がある	使用には特権が必要です。
noplink	特権がある	使用には特権が必要です。
plumbed	特権がある	使用には特権が必要です。

オプション	状態	意味
proxyarp	noproxyarp が指定されている場合、特権がある	特権のない使用はこのオプションを優先指定できません。
defaultroute	nodefaultroute が特権ファイルで、または特権ユーザーによって設定されている場合、特権がある	非特権ユーザーはこのオプションを優先指定できません。
disconnect	特権ファイルで、または特権ユーザーによって設定されている場合、特権がある	非特権ユーザーはこのオプションを優先指定できません。
bsdcomp	特権ファイルで、または特権ユーザーによって設定されている場合、特権がある	非特権ユーザーは特権ユーザーが指定したサイズより大きいコードサイズを指定できません。
deflate	特権ファイルで、または特権ユーザーによって設定されている場合、特権がある	非特権ユーザーは特権ユーザーが指定したサイズより大きいコードサイズを指定できません。
connect	特権ファイルで、または特権ユーザーによって設定されている場合、特権がある	非特権ユーザーはこのオプションを優先指定できません。
init	特権ファイルで、または特権ユーザーによって設定されている場合、特権がある	非特権ユーザーはこのオプションを優先指定できません。
pty	特権ファイルで、または特権ユーザーによって設定されている場合、特権がある	非特権ユーザーはこのオプションを優先指定できません。
welcome	特権ファイルで、または特権ユーザーによって設定されている場合、特権がある	非特権ユーザーはこのオプションを優先指定できません。
ttyname	特権ファイルで設定されている場合、特権がある	pppd をだれが起動したかに関係なく、スーパーユーザー特権で開かれます。
	非特権ファイルで設定されている場合、特権がない	pppd を起動したユーザーの特権で開かれます。

/etc/ppp/options 構成ファイル

ローカルマシン上のすべての PPP 通信にグローバルオプションを定義するには、/etc/ppp/options ファイルを使用します。/etc/ppp/options は特権ファイルです。pppd によって強制される規則ではありませんが、/etc/ppp/options は root が所有する必要があります。/etc/ppp/options で定義するオプションは、ほかのすべてのファイルおよびコマンド行内で定義される同じオプションより優先されます。

/etc/ppp/options で使用する可能性がある代表的なオプションを次に示します。

- **lock** – UUCP 形式のファイルロックを有効にします
- **noauth** – マシンが呼び出し元を認証しないことを示します

注 – Solaris PPP 4.0 ソフトウェアには、デフォルトの /etc/ppp/options ファイルは含まれていません。pppd の動作に、/etc/ppp/options ファイルは必要ありません。マシンに /etc/ppp/options ファイルがない場合、そのマシンで pppd を実行できるのは root だけです。

How to Define Communications Over the Serial Lineの説明に従って、テキストエディタを使用して 445 ページの「シリアル回線を介した通信を定義する方法」 を作成する必要があります。マシンがグローバルオプションを必要としない場合は、空の /etc/ppp/options ファイルを作成できます。これで、root および一般ユーザーの両方がローカルマシン上で pppd を実行できます。

/etc/ppp/options.tmpl テンプレート

/etc/ppp/options.tmpl には、/etc/ppp/options ファイルに関する有用なコメントのほかに、グローバルな /etc/ppp/options ファイルに共通の次の 3 つのオプションが含まれます。

```
lock
nodefaultroute
noproxyarp
```

オプション	定義
lock	UUCP 形式のファイルロックを有効にする
nodefaultroute	デフォルトの送信経路を定義しないことを指定する
noproxyarp	proxyarp を許可しない

/etc/ppp/options.tmpl をグローバルオプションファイルとして使用するには、/etc/ppp/options.tmpl の名前を /etc/ppp/options に変更します。次に、サイトの必要に応じてファイルの内容を変更します。

/etc/ppp/options ファイルの例 (参照先)

/etc/ppp/options ファイルの例は、次の節を参照してください。

- ダイアルアウトマシン用は、445 ページの「シリアル回線を介した通信を定義する方法」を参照してください。
- ダイアルインサーバー用は、453 ページの「シリアル回線を介した通信を定義する方法 (ダイアルインサーバー)」を参照してください。
- ダイアルインサーバーでの PAP サポート用は、467 ページの「PPP 構成ファイルに PAP サポートを追加する方法 (ダイアルインサーバー)」を参照してください。
- ダイアルアウトマシンでの PAP サポート用は、470 ページの「PPP 構成ファイルに PAP サポートを追加する方法 (ダイアルアウトマシン)」を参照してください。
- ダイアルインサーバーでの CHAP サポート用は、475 ページの「PPP 構成ファイルに CHAP サポートを追加する方法 (ダイアルインサーバー)」を参照してください。

/etc/ppp/options.*ttyname* 構成ファイル

シリアル回線上の通信の特性を /etc/ppp/options.*ttyname* ファイルで設定できます。/etc/ppp/options.*ttyname* は特権ファイルです。既存の /etc/ppp/options および $HOME/.ppprc ファイルを構文解析したあとで pppd によって読み取られます。$HOME/.ppprc が存在しない場合、pppd は /etc/ppp/options を構文解析したあと /etc/ppp/options.*ttyname* を読み取ります。

ttyname は、ダイアルアップリンク、専用回線リンクの両方で使用されます。*ttyname* は、モデムまたは ISDN TA が接続されている可能性があるマシン上の特定のシリアルポート (cua/a、cua/b など) を表します。

/etc/ppp/options.*ttyname* ファイルに名前を付けるときは、デバイス名にあるスラッシュ (/) をドット (.) に置き換えます。たとえば、デバイス cua/b 用の options ファイルの名前は /etc/ppp/options.cua.b になります。

注 – Solaris PPP 4.0 が正常に動作するうえで、/etc/ppp/options.*ttyname* ファイルは必要ありません。サーバーが PPP 用のシリアル回線を 1 つだけ持ち、オプションはほとんど必要ない場合、必要なオプションを別の構成ファイルまたはコマンド行で指定することができます。

ダイアルインサーバーでの /etc/ppp/options.*ttyname* の使用

ダイアルアップリンクでは、ダイアルインサーバー上のモデムが接続されているすべてのシリアルポートごとに、/etc/ppp/options. *ttyname* ファイルを個別に作成することもできます。通常のオプションは次のとおりです。

- ダイアルインサーバーが必要とする IP アドレス

  シリアルポート *ttyname* に着信する呼び出し元に特定の IP アドレスを使用させる必要がある場合は、このオプションを設定します。使用するアドレス空間により、予想される呼び出し元の数に比べて、PPP で使用可能な IP アドレスの数に制限がある場合があります。その場合は、ダイアルインサーバー上の PPP で使用されるシリアルインタフェースごとに IP アドレスを割り当てることを考えます。この割り当ては、PPP に動的なアドレス指定を実装します。

- asyncmap *map-value*

  asyncmap オプションは、特定のモデムまたは ISDN TA がシリアル回線上で受け取らない制御文字を割り当てます。xonxoff オプションを使用すると、pppd は自動的に 0xa0000 の asyncmap を設定します。

  map-value は、16 進数で入力し、問題のある制御文字を指定します。

- init "chat -U -f /etc/ppp/mychat"

  init オプションは、chat -U コマンド内の情報を使用して、シリアル回線上で通信を開始するようにモデムに指示します。モデムは、/etc/ppp/mychat ファイル内の chat 文字列を使用します。

- pppd(1m) のマニュアルページに一覧表示されているセキュリティーパラメータ

ダイアルアウトマシンでの /etc/ppp/options.*ttyname* の使用

ダイアルアウトシステムでは、モデムが接続されているシリアルポート用に /etc/ppp/options.*ttyname* ファイルを作成することも、あるいは /etc/ppp/options.*ttyname* を使用しないでおくこともできます。

注 – Solaris PPP 4.0 が正常に動作するうえで、/etc/ppp/options.*ttyname* ファイルは必要ありません。ダイアルアウトマシンが PPP 用のシリアル回線を 1 つだけ持ち、オプションはほとんど必要ない場合、必要なオプションを別の構成ファイルまたはコマンド行で指定することができます。

---

options.ttya.tmpl テンプレートファイル

/etc/ppp/options.ttya.tmpl ファイルには、/etc/ppp/options.*tty-name* ファイルに関して有用なコメントが含まれています。また、テンプレートには /etc/ppp/options.*tty-name* ファイルに共通の次の 3 つのオプションが含まれます。

```
38400
asyncmap 0xa0000
:192.168.1.1
```

オプション	定義
38400	ポート ttya でこのボーレートを使用する
asyncmap 0xa0000	ローカルマシンが接続に失敗したピアと通信できるように asyncmap 値 0xa0000 を割り当てる
:192.168.1.1	接続上で着信しているすべてのピアに IP アドレス 192.168.1.1 を割り当てる

サイトで /etc/ppp/options.ttya.tmpl を使用するには、/etc/ppp/options.tmpl の名前を /etc/ppp/options.*ttya-name* に変更します。*ttya-name* をモデムが接続しているシリアルポートの名前に置き換えます。次に、サイトの必要に応じてファイルの内容を変更します。

/etc/ppp/options.*ttyname* ファイルの例 (参照先)

/etc/ppp/options.*ttyname* ファイルの例は、次の節を参照してください。

- ダイアルアウトマシン用は、445 ページの「シリアル回線を介した通信を定義する方法」を参照してください。
- ダイアルインサーバー用は、453 ページの「シリアル回線を介した通信を定義する方法 (ダイアルインサーバー)」を参照してください。

ユーザー独自のオプションの設定

この節では、ダイアルインサーバー上でユーザーを設定する方法について詳細に説明します。

ダイアルインサーバーでの $HOME/.ppprc の設定

$HOME/.ppprc ファイルは、独自の PPP オプションを設定するユーザーを対象としています。管理者が、ユーザーのために $HOME/.ppprc を設定することもできます。

$HOME/.ppprc 内のオプションは、ファイルを呼び出しているユーザーに特権がある場合だけ、特権を与えられます。

呼び出し元が pppd コマンドを使って呼び出しを開始した場合、pppd デーモンは、.ppprc ファイルを 2 番目に確認します。

ダイアルインサーバーで $HOME/.ppprc を設定する手順については、451 ページの「ダイアルインサーバーのユーザーを設定する」を参照してください。

ダイアルアウトマシンでの $HOME/.ppprc の設定

$HOME/.ppprc ファイルは、ダイアルアウトマシン上で Solaris PPP 4.0 が正常に動作するのに必要ではありません。ダイアルアウトマシンでは、特別な場合を除いて $HOME/.ppprc も必要ありません。次を行う場合は、1 つ以上の .ppprc ファイルを作成します。

- 通信のニーズが異なる複数のユーザーが同じマシンからリモートピアを呼び出すのを許可する場合。このような場合は、ダイアルアウトする必要がある各ユーザーのホームディレクトリに個別の .ppprc ファイルを作成します。
- Van Jacobson 圧縮を無効にするなど、接続に固有の問題を制御するオプションを指定する必要がある場合。接続に関する問題のトラブルシューティングについては、James Carlson による『PPP Design, Implementation, and Debugging』および pppd(1M) のマニュアルページを参照してください。

.ppprc ファイルは、ダイアルインサーバーを構成するときにもっとも頻繁に使用されるため、.ppprc の構成手順について 451 ページの「ダイアルインサーバーのユーザーを構成する方法」 を参照してください。

ダイアルインサーバーと通信するための情報の指定

ダイアルインサーバーと通信するには、サーバーに関する情報を収集し、いくつかのファイルを編集する必要があります。特に大切なのは、ダイアルアウトマシンが呼び出す必要があるすべてのダイアルインサーバーについて通信要件を設定する必要があることです。ダイアルインサーバーに関する ISP 電話番号などのオプションは、/etc/ppp/options.*ttyname* ファイルで指定できます。ただし、ピア情報は、/etc/ppp/peers/*peer-name* ファイルで設定するのが最適です。

/etc/ppp/peers/*peer-name* ファイル

注 – /etc/ppp/peers/*peer-name* ファイルは、ダイアルアウトマシン上で Solaris PPP 4.0 が正常に動作するのに必要ではありません。

特定のピアと通信するための情報を指定するには、/etc/ppp/peers/*peer-name* ファイルを使用します。/etc/ppp/peers/*peer-name* を使用すると、一般ユーザーは、自分で設定することを許可されていない、あらかじめ選択された特権オプションを呼び出すことができます。

たとえば、非特権ユーザーの場合、noauth オプションが /etc/ppp/peers/*peer-name* ファイルで指定されていると、このオプションが優先されます。ユーザーが、認証資格を提供されていない peerB への接続を設定したいとします。ユーザーは

スーパーユーザーとして、noauth オプションを含む /etc/ppp/peers/peerB ファイルを作成できます。noauth は、ローカルマシンが peerB からの呼び出しを認証しないことを示します。

pppd デーモンは、次のオプションを検出すると、/etc/ppp/peers/*peer-name* を読み取ります。

`call` *peer-name*

ダイアルアウトマシンが通信する必要があるターゲットピアごとに /etc/ppp/peers/*peer-name* ファイルを作成できます。これは、スーパーユーザーの権限がなくても特定のダイアルアウト接続を呼び出すことを一般ユーザーに許可できる点で特に便利です。

/etc/ppp/peers/*peer-name* で指定する代表的なオプションを次に示します。

- `user` *user-name*

  PAP または CHAP 認証を行う場合に、ダイアルアウトマシンのログイン名として *user-name* をダイアルインサーバーに指定します。

- `remotename` *peer-name*

  peer-name をダイアルインマシンの名前として使用します。remotename は、/etc/ppp/pap-secrets または /etc/ppp/chap-secrets ファイルを走査するときに、PAP または CHAP 認証と連携して使用されます。

- `connect "chat` *chat_script* `..."`

  chat スクリプト内の命令を使ってダイアルインサーバーへの通信を開きます。

- `noauth`

  通信開始時に、ピア *peer-name* の認証を行いません。

- `noipdefault`

  ピアとのネゴシエートに使用する初期 IP アドレスを 0.0.0.0 に設定します。ほとんどの ISP への接続を設定するときに noipdefault を使用すると、ピア間で容易に IPCP ネゴシエーションを行うことができます。

- `defaultroute`

  接続上で IP が確立されたときに、デフォルトの IPv4 経路指定をインストールします。

特定のターゲットピアに適用する可能性がある上記以外のオプションについては、pppd(1M) のマニュアルページを参照してください。

/etc/ppp/peers/myisp.tmpl テンプレートファイル

/etc/ppp/peers/myisp.tmpl ファイルには、/etc/ppp/peers/*peer-name* ファイルに関して有用なコメントが含まれています。また、テンプレートには、/etc/ppp/peers/*peer-name* ファイルで使用する可能性がある次の一般的なオプションが含まれます。

```
connect "/usr/bin/chat -f /etc/ppp/myisp-chat"
user myname
remotename myisp
noauth
noipdefault
defaultroute
updetach
noccp
```

オプション	定義
connect "/usr/bin/chat -f /etc/ppp/myisp-chat"	chat スクリプト /etc/ppp/myisp-chat を使ってピアを呼び出します。
user myname	このアカウント名をローカルマシンに使用します。myname は、ピアの /etc/ppp/pap-secrets ファイル内でのこのマシンの名前です。
remotename myisp	myisp をローカルマシンの /etc/ppp/pap-secrets ファイル内のピア名として認識します。
noauth	認証資格を提供するためのピアの呼び出しを要求しません。
noipdefault	ローカルマシンにデフォルトの IP アドレスを使用しません。
defaultroute	ローカルマシンに割り当てられているデフォルトの経路指定を使用します。
updetach	標準出力ではなく、PPP ログファイル内にエラーを記録します。
noccp	CCP 圧縮を使用しません。

サイトで /etc/ppp/peers/myisp.tmpl を使用するには、/etc/ppp/peers/myisp.tmpl の名前を /etc/ppp/peers/*peer-name* に変更します。*peer-name* は、呼び出されるピアの名前に置き換えます。次に、サイトの必要に応じてファイルの内容を変更します。

/etc/ppp/peers/*peer-name* ファイルの例 (参照先)

/etc/ppp/peers/*peer-name* ファイルの例は、次の節を参照してください。

- ダイアルアウトマシン用は、447 ページの「個々のピアとの接続を定義する方法」を参照してください。
- 専用回線上のローカルマシン用は、460 ページの「専用回線上のマシンの設定方法」を参照してください。
- ダイアルアウトマシンでの PAP 認証のサポート用は、470 ページの「PPP 構成ファイルに PAP サポートを追加する方法 (ダイアルアウトマシン)」を参照してください。
- ダイアルアウトマシンでの CHAP 認証のサポート用は、477 ページの「PPP 構成ファイルに CHAP サポートを追加する方法 (ダイアルアウトマシン)」を参照してください。
- クライアントシステムでの PPPoE のサポート用は、480 ページの「PPPoE クライアントの設定」を参照してください。

ダイアルアップリンクのモデム速度の設定

モデムの設定で重要なのは、モデムが動作する速度の指定です。Sun Microsystems のコンピュータで使用するモデムには、次のガイドラインを適用してください。

- 旧 SPARC システム – システムに添付されているハードウェアマニュアルを確認します。SPARCstation マシンの多くは、38400 bps を超えないモデム速度を要求します。
- UltraSPARC マシン – モデム速度を 115200 bps に設定します。これは、最新のモデムで使用でき、ダイアルアップリンクに十分な速度です。デュアルチャネル ISDN TA を圧縮して使用する場合は、モデム速度を上げる必要があります。UltraSPARC での最大値は非同期接続で 460800 bps です。

ダイアルアウトマシンでは、/etc/ppp/peers/*peer-name* などの PPP 構成ファイルでモデム速度を設定するか、あるいは pppd のオプションとして速度を指定します。

ダイアルインサーバーでは、449 ページの「ダイアルインサーバーにデバイスを構成する」で説明したように、ttymon 機能または Solaris 管理コンソールを使って速度を設定する必要があります。

ダイアルアップリンクでの会話の定義

ダイアルアウトマシンとそのリモートピアは、さまざまな命令をネゴシエーションしたり交換したりして PPP リンク上で通信します。ダイアルアウトマシンを構成するときは、ローカルおよびリモートモデムから要求される命令の内容を判定する必要があります。次に、その命令を含む chat スクリプトと呼ばれるファイルを作成します。この節では、モデムの設定および chat スクリプトの作成について説明します。

chat スクリプトの内容

ダイアルアウトマシンが接続する必要があるリモートピアは、通常、それぞれピア自身の chat スクリプトを必要とします。

注 – chat スクリプトは、通常、ダイアルアップリンクだけで使用されます。専用回線リンクは、起動時の設定が必要な非同期インタフェースを使用しないかぎり、chat スクリプトを使用しません。

---

chat スクリプトの内容は、モデムまたは ISDN TA の要件、およびリモートピアの要件によって決まります。スクリプトの内容は、一連の送信予期文字列として表示されます。ダイアルアウトマシンとリモートピアは、この文字列を通信の開始処理時に交換します。

予期文字列には、会話を開始するためにダイアルアウトホストマシンがリモートピアから受け取ると予想される文字が含まれます。送信文字列には、ダイアルアウトマシンが、予期文字列を受け取ったあとでリモートピアに送信する文字が含まれます。

chat スクリプト内の情報には、通常、次が含まれます。

- モデムコマンド。しばしば *AT* コマンドと呼ばれる。モデムが電話を通じてデータを伝送することを可能にする
- ターゲットピアの電話番号

  この電話番号は、ISP または企業サイトのダイアルインサーバー、あるいは個別のマシンが要求する番号の場合がある。
- タイムアウト値 (必要な場合)
- リモートピアからの予想されるログインシーケンス
- ダイアルアウトマシンが送信するログインシーケンス

chat スクリプトの例

この節では、独自の chat スクリプトを作成する際の参考になる chat スクリプトの例を紹介します。モデムメーカーのガイドや ISP およびほかのターゲットホストからの情報には、モデムおよびターゲットピアの chat の要件が含まれています。また、数多くの PPP Web サイトで chat スクリプトのサンプルが提供されています。

基本のモデム chat スクリプト

次は、独自の chat スクリプトを作成するためのテンプレートとして使用できる基本の chat スクリプトです。

```
ABORT   BUSY
ABORT   'NO CARRIER'
REPORT  CONNECT
TIMEOUT 10
"" AT&F1M0&M5S2=255
SAY     "Calling myserver\n"
TIMEOUT 60
OK      "ATDT1-123-555-1212"
ogin: pppuser
ssword: \q\U
% pppd
```

次の表では、chat スクリプトの内容を説明します。

スクリプトの内容	意味
`ABORT BUSY`	モデムが反対側のピアからこのメッセージを受け取った場合、伝送を中止します。
`ABORT 'NO CARRIER'`	ダイアル時にモデムが `ABORT 'NO CARRIER'` を報告した場合、伝送を中止します。このメッセージは、通常、ダイアルまたはモデムのネゴシエーションが失敗したときに発生します。
`REPORT CONNECT`	`CONNECT` 文字列をモデムから収集し、その文字列を出力します。
`TIMEOUT 10`	初期タイムアウトを 10 秒に設定します。モデムは即時に応答する必要があります。
`"" AT&F1M0&M5S2=255`	`M0` – 接続中、スピーカーをオフに設定します。 `&M5` – モデムにエラー制御を要求させます。 `S2=255` – TIES "+++" ブレークシーケンスを無効にします。
`SAY "Calling myserver\n"`	ローカルマシン上に「`Calling myserver` (myserver を呼び出し中)」のメッセージを表示します。
`TIMEOUT 60`	タイムアウトを 60 秒にリセットし、接続ネゴシエーションにより多くの時間を割り当てます。

スクリプトの内容	意味
OK "ATDT1-123-555-1212"	電話番号 1-123-555-1212 を使ってリモートピアに発信します。
ogin: pppuser	UNIX 方式のログインを使ってピアにログインします。ユーザー名 pppuser を指定します。
ssword: \q\U	\q – -v オプションを使ってデバッグする場合、ログをとりません。
	\U – -U のあとに続く文字列の内容をこの位置に挿入します。文字列はコマンド行に指定されるもので、通常はパスワードが含まれます。
% pppd	% シェルプロンプトを待ち、pppd コマンドを実行します。

/etc/ppp/myisp-chat.tmpl chat スクリプトテンプレート

Solaris PPP 4.0 には、ユーザーが自分のサイトで使用するために変更できる /etc/ppp/myisp-chat.tmpl という chat スクリプトテンプレートが用意されています。/etc/ppp/myisp-chat.tmpl は、基本のモデム chat スクリプトと似ていますが、ログインシーケンスが含まれていません。

```
ABORT   BUSY
ABORT   'NO CARRIER'
REPORT  CONNECT
TIMEOUT 10
""      "AT&F1"
OK      "AT&C1&D2"
SAY     "Calling myisp\n"
TIMEOUT 60
OK      "ATDT1-123-555-1212"
CONNECT \c
```

スクリプトの内容	意味
ABORT BUSY	モデムが反対側のピアからこのメッセージを受け取った場合、伝送を中止します。
ABORT 'NO CARRIER	ダイアル時にモデムが ABORT 'NO CARRIER' を報告した場合、伝送を中止します。このメッセージは、通常、ダイアルまたはモデムのネゴシエーションが失敗したときに発生します。
REPORT CONNECT	CONNECT 文字列をモデムから収集し、その文字列を出力します。
TIMEOUT 10	初期タイムアウトを 10 秒に設定します。モデムは即時に応答する必要があります。
"" "AT&F1"	モデムを出荷時のデフォルトにリセットします。

スクリプトの内容	意味
OK "AT&C1&D2"	モデムをリセットします。その結果、&C1 では、モデムからの DCD がキャリアを追跡します。リモート側がなんらかの理由で電話を切った場合、DCD はドロップします。 &D2 では、DTR の High-Low 遷移により、モデムが「オンフック」状態になるか、またはハングアップします。
SAY "Calling myisp\n"	ローカルマシン上に「Calling myisp (myisp を呼び出し中)」のメッセージを表示します。
TIMEOUT 60	タイムアウトを 60 秒にリセットし、接続ネゴシエーションにより多くの時間を割り当てます。
OK "ATDT1-123-555-1212"	電話番号 1-123-555-1212 を使ってリモートピアに発信します。
CONNECT \c	反対側のピアのモデムからの CONNECT メッセージを待ちます。

ISP を呼び出すためのモデムの chat スクリプト

ダイアルアウトマシンから U.S. Robotics Courier モデムを使用して ISP を呼び出すには、テンプレートとして次の chat スクリプトを使用します。

```
ABORT   BUSY
ABORT   'NO CARRIER'
REPORT  CONNECT
TIMEOUT 10
"" AT&F1M0&M5S2=255
SAY     "Calling myisp\n"
TIMEOUT 60
OK      "ATDT1-123-555-1212"
CONNECT \c
\r \d\c
SAY "Connected; running PPP\n"
```

次の表では、chat スクリプトの内容を説明します。

スクリプトの内容	意味
ABORT BUSY	モデムが反対側のピアからこのメッセージを受け取った場合、伝送を中止します。
ABORT 'NO CARRIER'	モデムが反対側のピアからこのメッセージを受け取った場合、伝送を中止します。
REPORT CONNECT	CONNECT 文字列をモデムから収集し、その文字列を出力します。
TIMEOUT 10	初期タイムアウトを 10 秒に設定します。モデムは即時に応答する必要があります。

スクリプトの内容	意味
"" AT&F1M0M0M0M0&M5S2=255	M0 – 接続中、スピーカーをオフに設定します。 &M5 – モデムにエラー制御を要求させます。 S2=255 – TIES “+++” ブレークシーケンスを無効にします。
SAY "Calling myisp\n"	ローカルマシン上に「Calling myisp (myisp を呼び出し中)」のメッセージを表示します。
TIMEOUT 60	タイムアウトを 60 秒にリセットし、接続ネゴシエーションにより多くの時間を割り当てます。
OK "ATDT1-123-555-1212"	電話番号 1-123-555-1212 を使ってリモートピアに発信します。
CONNECT \c	反対側のピアのモデムからの CONNECT メッセージを待ちます。
\r \d\c	CONNECT メッセージの最後まで待ちます。
SAY “Connected; running PPP\n”	ローカルマシン上に「Connected; running PPP (接続完了。PPP を実行中)」という通知メッセージを表示します。

UNIX 方式ログイン用に拡張された基本の chat スクリプト

次の chat スクリプトは、Solaris のリモートピアまたはほかの UNIX タイプのピアを呼び出すために基本のスクリプトを拡張したものです。この chat スクリプトは、446 ページの「ピアを呼び出すための命令群を作成する方法」で使用されています。

```
SAY "Calling the peer\n"
TIMEOUT 10
ABORT BUSY
ABORT 'NO CARRIER'
ABORT ERROR
REPORT CONNECT
"" AT&F1&M5S2=255
TIMEOUT 60
OK ATDT1-123-555-1234
CONNECT \c
SAY "Connected; logging in.\n"
TIMEOUT 5
ogin:--ogin: pppuser
TIMEOUT 20
ABORT 'ogin incorrect'
ssword: \qmypassword
"% " \c
SAY "Logged in.  Starting PPP on peer system.\n"
ABORT 'not found'
"" "exec pppd"
~ \c
```

次の表では、chat スクリプトのパラメータを説明します。

スクリプトの内容	意味
TIMEOUT 10	初期タイムアウトを 10 秒に設定します。モデムは即時に応答する必要があります。
ABORT BUSY	モデムが反対側のピアからこのメッセージを受け取った場合、伝送を中止します。
ABORT 'NO CARRIER'	モデムが反対側のピアからこのメッセージを受け取った場合、伝送を中止します。
ABORT ERROR	モデムが反対側のピアからこのメッセージを受け取った場合、伝送を中止します。
REPORT CONNECT	CONNECT 文字列をモデムから収集し、その文字列を出力します。
"" AT&F1&M5S2=255	&M5 – モデムにエラー制御を要求させます。 S2=255 – TIES "+++" ブレークシーケンスを無効にします。
TIMEOUT 60	タイムアウトを 60 秒にリセットし、接続ネゴシエーションにより多くの時間を割り当てます。
OK ATDT1-123-555-1234	電話番号 1-123-555-1212 を使ってリモートピアに発信します。
CONNECT \c	反対側のピアのモデムからの CONNECT メッセージを待ちます。
SAY "Connected; logging in.\n"	「Connected; logging in (接続完了。ログイン中)」という通知メッセージを表示して、ユーザーの状態を知らせます。
TIMEOUT 5	タイムアウトを変更し、ログインプロンプトを迅速に表示できるようにします。
ogin:--ogin: pppuser	ログインプロンプトを待ちます。ログインプロンプトを受け取らなかった場合は、RETURN を送信して待機します。次にユーザー名 pppuser をピアに送信します。この後に続くシーケンスは、ほとんどの ISP から PAP ログインと呼ばれています。ただし、PAP 認証とはまったく無関係です。
TIMEOUT 20	タイムアウトを 20 秒に変更し、パスワードの検証により多くの時間をかけられるようにします。
ssword: \qmysecrethere	ピアからのパスワードプロンプトを待ちます。プロンプトを受け取ると、パスワード \qmysecrethere を送信します。\q は、パスワードがシステムログファイルに書き込まれるのを防ぎます。
"% " \c	ピアからのシェルプロンプトを待ちます。chat スクリプトは C シェルを使用します。ユーザーが異なるシェルを使ってログインすることを希望する場合は、この値を変更します。

スクリプトの内容	意味
SAY "Logged in. Starting PPP on peer system.\n"	「Logged in. Starting PPP on peer system (ログイン完了。ピアシステム上で PPP を開始中)」という通知メッセージを表示してユーザーの状態を通知します。
ABORT 'not found'	シェルがエラーに遭遇した場合、伝送を中止します。
"" "exec pppd"	ピア上で pppd を起動します。
~ \c	PPP がピア上で開始するのを待ちます。

ISP は、CONNECT \c の直後に PPP を開始することをしばしば「PAP ログイン」といいます。しかし、実際には、PAP ログインは PAP 認証とは無関係です。

ogin:--ogin: pppuser 句は、ダイアルインサーバーからのログインプロンプトに対してユーザー名 pppuser を送信するようにモデムに指示します。pppuser は、ダイアルインサーバー上のリモートユーザー user1 用に作成された専用の PPP ユーザーアカウント名です。ダイアルインサーバー上に PPP ユーザーアカウントを作成する方法については、451 ページの「ダイアルインサーバーのユーザーを構成する方法」を参照してください。

外部 ISDN TA 用 chat スクリプト

次は、ダイアルアウトマシンから ZyXEL omni.net. ISDN TA を使って呼び出すための chat スクリプトです。

```
SAY "Calling the peer\n"
TIMEOUT 10
ABORT BUSY
ABORT 'NO CARRIER'
ABORT ERROR
REPORT CONNECT
"" AT&FB40S83.7=1&K44&J3X7S61.3=1S0=0S2=255
OK ATDI18882638234
CONNECT \c
\r \d\c
SAY "Connected; running PPP\n"
```

次の表では、chat スクリプトのパラメータを説明します。

スクリプトの内容	意味
SAY "Calling the peer"	ダイアルアウトマシンの画面上にこのメッセージを表示します。
TIMEOUT 10	初期タイムアウトを 10 秒に設定します。
ABORT BUSY	モデムが反対側のピアからこのメッセージを受け取った場合、伝送を中止します。

スクリプトの内容	意味
ABORT 'NO CARRIER'	モデムが反対側のピアからこのメッセージを受け取った場合、伝送を中止します。
ABORT ERROR	モデムが反対側のピアからこのメッセージを受け取った場合、伝送を中止します。
REPORT CONNECT	CONNECT 文字列をモデムから収集し、その文字列を出力します。
"" AT&FB40S83.7=1&K44&J3X7S61.3=1S0=0S2=255	この行内の文字は、次を意味します。 ■ &F – 出荷時のデフォルトを使用します ■ B40 – 非同期 PPP 変換を実行します ■ S83.7=1 – スピーチベアラーにデータを使用します ■ &K44 – CCP 圧縮を有効にします ■ &J3 – MP を有効にします ■ X7 – DCE 側のレートを報告します ■ S61.3=1 – パケット断片化を使用します ■ S0=0 – 自動応答を行いません ■ S2=255 – TIES エスケープを無効にします
OK ATDI18882638234	ISDN 呼び出しを行います。マルチリンクでは、2 番目の呼び出しは、同じ電話番号に対して行われます。これは、通常、ほとんどの ISP の条件です。リモートピアが 2 番目の電話番号に異なる番号を要求する場合は、「+*nnnn*」を付け加えます。*nnnn* は 2 番目の電話番号を表します。
CONNECT \c	反対側のピアのモデムからの CONNECT メッセージを待ちます。
\r \d\c	CONNECT メッセージの最後まで待ちます。
SAY "Connected; running PPP\n"	ダイアルアウトマシンの画面上にこのメッセージを表示します。

chat スクリプトのオプションの説明およびその他の詳細な情報については、chat(1M) のマニュアルページを参照してください。expect-send 文字列の説明については、577 ページの「/etc/uucp/Systems ファイルの Chat-Script フィールド」を参照してください。

その他の chat スクリプト例の参照先

数多くの Web サイトで、chat スクリプトのサンプルとスクリプト作成のヒントが提供されています。たとえば、http://ppp.samba.org/ppp/index.html を参照してください。

chat スクリプトの呼び出し

chat スクリプトを呼び出すには、connect オプションを使用します。PPP 構成ファイルまたはコマンド行で connect "chat ..." を使用できます。

chat スクリプトは実行可能ではありませんが、connect によって呼び出されるプログラムは実行可能でなければなりません。connect によって呼び出されるプログラムとして chat ユーティリティーを使用することがあります。この場合、-f オプションを使用して chat スクリプトを外部ファイルに保存すると、chat スクリプトファイルは実行可能にはなりません。

chat(1m) で説明されている chat プログラムは、実際の chat スクリプトを実行します。pppd デーモンは、pppd が connect "chat ..." オプションを検出すると必ず、chat プログラムを起動します。

注 – Perl や Tcl などの外部プログラムを使って機能を拡張した chat スクリプトを作成することもできます。Solaris PPP 4.0 で chat ユーティリティーが提供されているのは、ユーザーの便宜を図るためです。

▼ chat スクリプトを呼び出す方法 (手順)

1. **ASCII** ファイル形式で **chat** スクリプトを作成します。
2. 次の構文を使用して、任意の **PPP** 構成ファイル内で **chat** スクリプトを呼び出します。

   ```
   connect 'chat  -f /etc/ppp/chatfile'
   ```

   -f フラグは、ファイル名があとに続くことを示します。*/etc/ppp/chatfile* は、chat ファイルの名前を表します。
3. 外部 **chat** ファイルの読み取り権を pppd コマンドを実行するユーザーに与えます。

注意 – connect 'chat ...' オプションが特権ソースから呼び出された場合でも、chat プログラムは、常にユーザーの権限と連携して実行します。したがって、-f オプションを使って読み取る個別の chat ファイルを呼び出すユーザーは、そのファイルの読み取り権を備えている必要があります。chat スクリプトにパスワードやその他の機密情報が含まれる場合、この特権はセキュリティーの問題にかかわる可能性があります。

例 22–1 インライン chat スクリプト

次に示すように、chat スクリプトの全会話を 1 つの行に入れることができます。

```
connect 'chat "" "AT&F1" OK ATDT5551212 CONNECT "\c"'
```

chat スクリプトは、chat キーワードのあとに続きます。スクリプトは "\c"' で終了します。この形式は、pppd の引数として、PPP 構成ファイルまたはコマンド行で使用できます。

参考 外部ファイル内の chat スクリプト

特定のピアで必要な chat スクリプトが長くて複雑な場合は、スクリプトを別ファイルとして作成することを考えます。外部 chat ファイルは、簡単に維持、作成できます。ハッシュ記号 (#) のあとに続けて chat ファイルについてのコメントを追加できます。

外部ファイルに含まれる chat スクリプトの使用については、446 ページの「ピアを呼び出すための命令群を作成する方法」の手順を参照してください。

実行可能な chat ファイルの作成

実行可能なスクリプトの chat ファイルを作成して、ダイアルアップリンクが開始されたときに自動的に実行されるようにできます。これにより、接続開始時に、従来の chat スクリプトに含まれるコマンドのほかに、パリティー設定のための stty のような追加コマンドを実行できます。

この実行可能な chat スクリプトは、7 ビット長/ 偶数パリティーを要求する旧スタイルの UNIX システムにログインし、PPP 実行時に 8 ビット長/ パリティーなしに移行します。

```
#!/bin/sh
chat "" "AT&F1" OK "ATDT555-1212" CONNECT "\c"
stty evenp
chat ogin: pppuser ssword: "\q\U" % "exec pppd"
stty -evenp
```

▼ 実行可能な chat プログラムを作成する方法

1. テキストエディタを使用して、前述の例のような実行可能な chat プログラムを作成します。
2. chat プログラムを実行可能にします。

```
# chmod +x /etc/ppp/chatprogram
```

3 **chat** プログラムを呼び出します。

```
connect /etc/ppp/chatprogram
```

chat プログラムの場所は、/etc/ppp ファイルシステム内である必要はありません。chat プログラムは任意の場所に保存できます。

接続時の呼び出し元の認証

この節では、PPP 認証プロトコルの動作と認証プロトコルに関連するデータベースについて説明します。

パスワード認証プロトコル (PAP)

PAP 認証は、UNIX の login プログラムと動作が多少似ていますが、PAP はユーザーにシェルアクセスを許可しない点が異なります。PAP は、PPP 構成ファイルと /etc/ppp/pap-secrets ファイルの形式の PAP データベースを使って認証を設定します。また、PAP セキュリティー資格の定義にも /etc/ppp/pap-secrets を使用します。この資格には、ピア名 (PAP の用語では「ユーザー名」)とパスワードが含まれます。また、ローカルマシンへの接続を許可されている呼び出し元に関する情報も含まれます。PAP のユーザー名とパスワードは、パスワードデータベース内の UNIX ユーザー名およびパスワードと同じものにすることも、違うものにすることもできます。

/etc/ppp/pap-secrets ファイル

PAP データベースは、/etc/ppp/pap-secrets ファイルに実装されています。認証が成功するためには、PPP リンクの両側にある各マシンで、/etc/ppp/pap-secrets ファイル内に適切に設定された PAP 資格が必要です。呼び出し元 (認証される側) は、/etc/ppp/pap-secrets ファイルまたは旧バージョンの +ua ファイルの user 列および password 列で資格を提供します。サーバー (認証する側) は、UNIX の passwd データベースまたは PAM 機能により /etc/ppp/pap-secrets 内の情報と対照してこの資格の妥当性を検証します。

/etc/ppp/pap-secrets ファイルの構文は、次のとおりです。

```
myclient ISP-server mypassword *
```

パラメータの意味は次のとおりです。

myclient	呼び出し元の PAP ユーザー名。この名前は、呼び出し元の UNIX ユーザー名と同じ場合があります。特に、ダイアルインサーバーが PAP の login オプションを使用する場合は、多くの場合同じになります。
ISP-server	リモートマシンの名前。ダイアルインサーバーである場合がしばしばあります。

`mypassword` 呼び出し元の PAP パスワード。

`*` 呼び出し元に関連付けられている IP アドレス。任意の IP アドレスを表すには、アスタリスク (*) を使用します。

PAP パスワードの作成

PAP パスワードは、接続上をクリアテキストで (読み取り可能な ASCII 形式で) 送信されます。呼び出し元 (認証される側) では、PAP パスワードを次のどこかにクリアテキストで格納する必要があります。

- `/etc/ppp/pap-secrets` 内
- 別の外部ファイル内
- `pap-secrets` @ 機能による名前付きパイプ内
- `pppd` のオプションとして、コマンド行上または PPP 構成ファイル内のどちらか
- `+ua` ファイルを介して

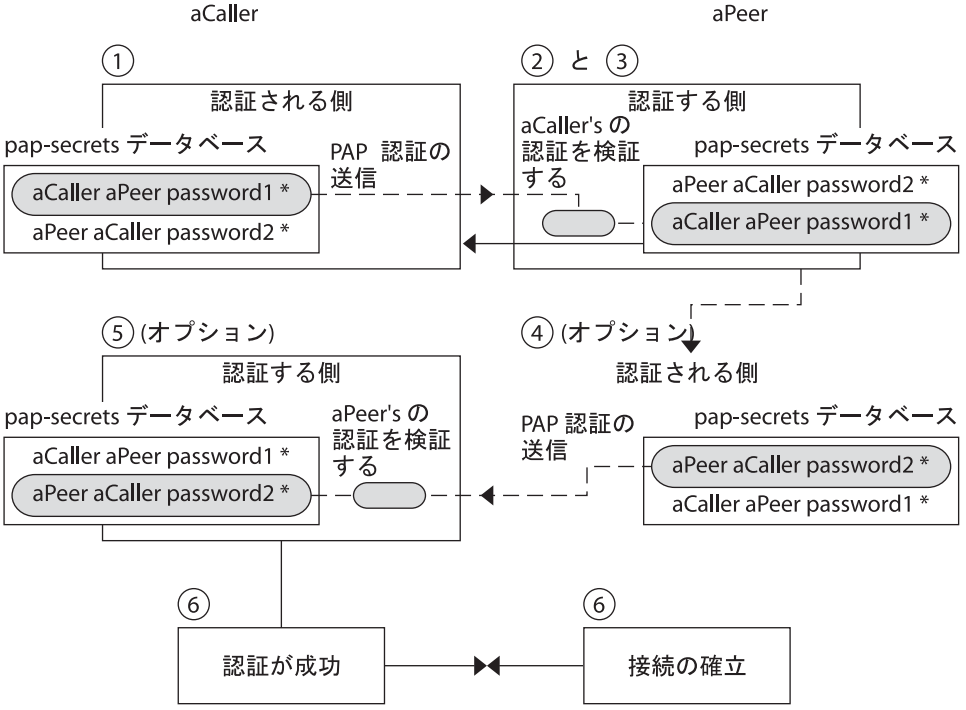
サーバー (認証する側) では、PAP パスワードは、次のどれかの方法で隠すことができます。

- `pap-secrets` ファイル内で `papcrypt` を指定し、`crypt(3C)` によってハッシュ化されたパスワードを使用する。
- `pppd` に `login` オプションを指定し、パスワード列に二重引用符 ("") を入れることにより `pap-secrets` ファイルからパスワードを除外する。この場合、認証は UNIX の passwd データベースまたは `pam(3PAM)` メカニズムを利用して行われます。

PAP 認証時の動作

PAP 認証は、次の順序で発生します。

図 22–1　PAP 認証処理



1. 呼び出し元 (認証される側) がリモートピア (認証する側) を呼び出し、接続ネゴシエーションの一環として PAP ユーザー名とパスワードを伝えます。
2. ピアは、`/etc/ppp/pap-secrets` ファイルで呼び出し元の識別情報を検証します。PAP の `login` オプションを使用する場合は、呼び出し元のユーザー名とパスワードの検証にパスワードデータベースが使用されます。
3. 認証が成功すると、ピアは呼び出し元との接続ネゴシエーションを継続します。認証に失敗すると、接続は切られます。
4. (オプション) 呼び出し元がリモートピアからの応答を認証する場合は、リモートピアが自身の PAP 資格を呼び出し元に送信する必要があります。したがって、リモートピアは認証される側になり、呼び出し側は認証する側になります。
5. (オプション) 最初の呼び出し元が自身の `/etc/ppp/pap-secrets` を読み取り、リモートピアの識別情報を検証します。

注 – 最初の呼び出し元がリモートピアに認証資格を要求する場合は、手順 1 と手順 4 が並行して行われます。

ピアが認証されると、ネゴシエーションが継続されます。認証されない場合は、接続が切られます。

6. 呼び出し元とピアのネゴシエーションは、接続の確立に成功するまで継続されます。

/etc/ppp/pap-secrets での login オプションの使用

PAP 資格を認証するための login オプションを PPP 構成ファイルに追加できます。たとえば /etc/ppp/options で login を指定した場合、pppd は呼び出し元の PAP 資格が Solaris のパスワードデータベース内に存在するかどうかを検証します。次に、login オプションを追加した /etc/ppp/pap-secrets ファイルの形式を示します。

```
joe    *  ""  *
sally  *  ""  *
sue    *  ""  *
```

パラメータの意味は次のとおりです。

呼び出し元　joe、sally、sue は、承認された呼び出し元の名前です。

サーバー　アスタリスク (*) は、任意のサーバー名が有効であることを示します。name オプションは PPP 構成ファイルでは必須ではありません。

パスワード　二重引用符は、任意のパスワードが有効であることを示します。

この列にパスワードがある場合、ピアからのパスワードは、PAP パスワードと UNIX passwd データベースの両方に一致しなければなりません。

IP アドレス　アスタリスク (*) は、任意の IP アドレスが許可されることを示します。

チャレンジハンドシェーク認証プロトコル (CHAP)

CHAP 認証は、「チャレンジ」と「応答」という概念を使用します。つまり、ピア (認証する側) は識別情報を証明するために呼び出し元 (認証される側) にチャレンジします。チャレンジには、乱数、および認証する側によって生成された一意の ID が含まれます。呼び出し元は、ID、乱数、および呼び出し元の CHAP セキュリティー資格を使って適切な応答 (ハンドシェーク) を生成しピアに送信します。

CHAP セキュリティー資格には、CHAP ユーザー名と CHAP「シークレット」が含まれます。CHAP シークレットは、PPP リンクネゴシエーションを行う前に、あらかじめ呼び出し元とピアの両方が知っている任意の文字列です。CHAP セキュリティー資格は、CHAP データベース /etc/ppp/chap-secrets 内で設定します。

/etc/ppp/chap-secrets ファイル

CHAP データベースは、/etc/ppp/chap-secrets ファイルに実装されています。認証が成功するためには、PPP リンクの両側にある各マシンで、/etc/ppp/chap-secrets ファイル内に互いのマシンの CHAP 資格が必要です。

注 – PAP と異なり、共有シークレットは、両方のピアでクリアテキストでなければなりません。CHAP では、crypt、PAM、または PPP ログインオプションは使用できません。

/etc/ppp/chap-secrets ファイルの構文は、次のとおりです。

```
myclient myserver secret5748 *
```

パラメータの意味は次のとおりです。

myclient　呼び出し元の CHAP ユーザー名。呼び出し元の UNIX ユーザー名と同じ名前にすることも、違う名前にすることもできます。

myserver　リモートマシンの名前。ダイアルインサーバーである場合がしばしばあります。

secret5748　呼び出し元の CHAP シークレット。

注 – PAP パスワードと異なり、CHAP シークレットは送信されません。CHAP シークレットは、ローカルマシンが応答を処理するときに使用されます。

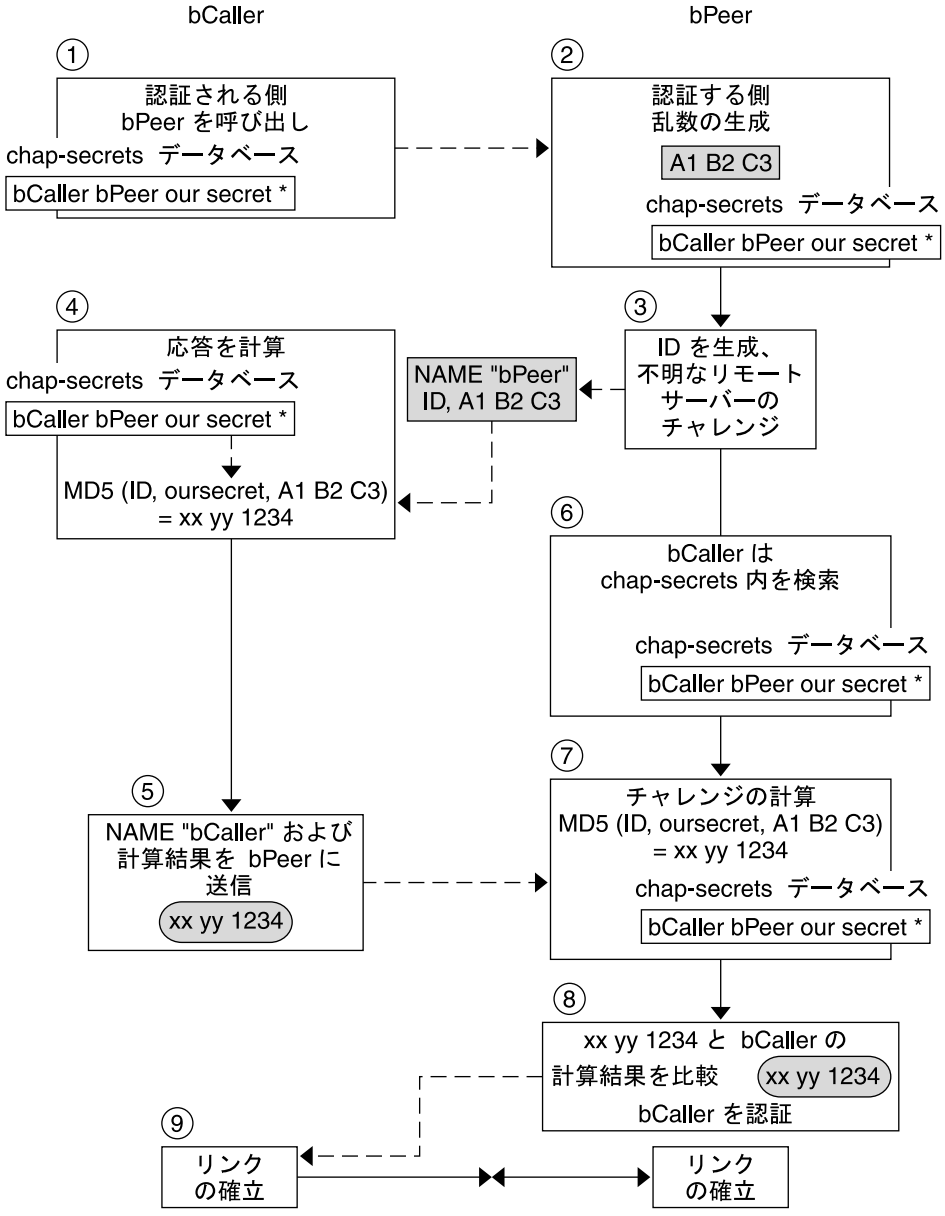
---

*　呼び出し元に関連付けられている IP アドレス。任意の IP アドレスを表すには、アスタリスク (*) を使用します。

CHAP 認証時の動作

CHAP 認証は、次の順序で発生します。

図 22–2　CHAP 認証手順



1. 通信を開始しようとする 2 つのピアが、PPP リンクのネゴシエーション時に認証に使用するシークレットについて合意します。

2. 両方のマシンの管理者が、シークレット、CHAP ユーザー名、その他の CHAP 資格をそれぞれのマシンの /etc/ppp/chap-secrets データベースに追加します。
3. 呼び出し元 (認証される側) がリモートピア (認証する側) を呼び出します。
4. 認証する側が乱数と ID を生成し、それらを認証される側にチャレンジとして送信します。
5. 認証される側は、/etc/ppp/chap-secrets データベース内でピアの名前とシークレットを調べます。
6. 認証される側は、シークレットとピアの乱数チャレンジに MD5 計算アルゴリズムを適用することにより、応答を計算します。次に、認証される側は、認証する側に結果を応答として送信します。
7. 認証する側は、/etc/ppp/chap-secrets データベース内で認証される側の名前とシークレットを調べます。
8. 認証する側は、チャレンジとして生成された数値と /etc/ppp/chap-secrets 内の認証される側のシークレットに MD5 を適用することにより、自身の数値を計算します。
9. 認証する側は、呼び出し元からの応答と結果を比較します。2 つの数字が同じ場合、ピアは、呼び出し元の認証に成功し、接続ネゴシエーションが続けられます。認証されない場合は、接続が切られます。

呼び出し元の IP アドレス指定スキーマの作成

リモートユーザーごとに一意の IP アドレスを割り当てる代わりに、すべての着呼のために 1 つ以上の IP アドレスを作成することを考えます。専用 IP アドレスは、予想される呼び出し元の数が、ダイアルインサーバー上のシリアルポートとモデムの数を上回る場合、特に重要です。サイトの必要性に応じて、さまざまなシナリオを実現できます。さらに、シナリオは、相互に排他的ではありません。

呼び出し元への IP アドレスの動的割り当て

動的アドレス指定は、/etc/ppp/options.*ttyname* で定義されている IP アドレスを各呼び出し元に割り当てます。動的アドレス指定は、シリアルポート単位で発生します。シリアル回線に呼が着信すると、呼び出しを処理するシリアルインタフェース用に /etc/ppp/options.*ttyname* ファイルで定義されている IP アドレスが呼び出し元に与えられます。

たとえば、ダイアルインサーバーに、着呼に対してダイアルアップサービスを提供するシリアルインタフェースが 4 つあると仮定します。

- シリアルポート `term/a` 用に、次のエントリがある `/etc/ppp/options.term.a` ファイルを作成します。

  ```
  :10.1.1.1
  ```
- シリアルポート `term/b` 用に、次のエントリがある `/etc/ppp/options.term.b` ファイルを作成します。

  ```
  :10.1.1.2
  ```
- シリアルポート `term/c` 用に、次のエントリがある `/etc/ppp/options.term.c` ファイルを作成します。

  ```
  :10.1.1.3
  ```
- シリアルポート `term/d` 用に、次のエントリがある `/etc/ppp/options.term.d` ファイルを作成します。

  ```
  :10.1.1.4
  ```

この以前のアドレス指定スキーマでは、`/dev/term/c` のシリアルインタフェースに着信する呼び出しは、呼び出しを行なっている間中、IP アドレス 10.1.1.3 が与えられます。最初の呼び出し元が回線を切ったあと、次にシリアルインタフェース `/dev/term/c` に着信する呼も、IP アドレス 10.1.1.3 を与えられます。

動的アドレス指定には、次のような利点があります。

- PPP ネットワークの使用状況をシリアルポートまで追跡できる
- PPP 使用で割り当てる IP アドレスの数を最小限にできる
- IP フィルタリングをより簡単に管理できる

呼び出し元への IP アドレスの静的割り当て

サイトが PPP 認証を実装する場合は、個々の呼び出し元に特定の「静的」 IP アドレスを割り当てることができます。この場合、ダイアルアウトマシンがダイアルインサーバーを呼び出すたびに、呼び出し元は同じ IP アドレスを受け取ります。

静的アドレスは、pap-secrets または chap-secrets のどちらかのデータベースで実装します。次に、静的 IP アドレスを定義した `/etc/ppp/pap-secrets` ファイルの例を示します。

```
joe    myserver  joepasswd   10.10.111.240
sally  myserver  sallypasswd 10.10.111.241
sue    myserver  suepasswd   10.10.111.242
```

呼び出し元　　`joe`、`sally`、`sue` は、承認された呼び出し元の名前です。

サーバー　　`myserver` は、サーバーの名前を示します。

パスワード　joepasswd、sallypasswd、suepasswd は、各呼び出し元のパスワードを示します。

IP アドレス　10.10.111.240、10.10.111.241、10.10.111.242 は、各呼び出し元に割り当てられた IP アドレスです。

次に、静的 IP アドレスを定義した /etc/ppp/chap-secrets ファイルの例を示します。

```
account1 myserver secret5748  10.10.111.244
account2 myserver secret91011 10.10.111.245
```

呼び出し元　account1 と account2 は、呼び出し元の名前を示します。

サーバー　myserver は、各呼び出し元のサーバーの名前を示します。

パスワード　secret5748 と secret91011 は、各呼び出し元の CHAP シークレットを示します。

IP アドレス　10.10.111.244 と 10.10.111.245 は、各呼び出し元の IP アドレスです。

sppp ユニット番号による IP アドレスの割り当て

PAP 認証または CHAP 認証を使用している場合は、sppp ユニット番号を使って IP アドレスを呼び出し元に割り当てることができます。次にこの方法の例を示します。

```
myclient ISP-server mypassword 10.10.111.240/28+
```

正符号 (+) は、ユニット番号が IP アドレスに追加されていることを示します。次の事項に注意してください。

- アドレス 10.10.111.240 から 10.10.111.255 までがリモートユーザーに割り当てられます。
- sppp0 は IP アドレス 10.10.111.240 を取得します。
- sppp1 は IP アドレス 10.10.111.241 を取得し、以下同様に続きます。

DSL サポート用の PPPoE トンネルの作成

PPPoE を使用することにより、1 台以上の DSL モデムを使用している複数のクライアントに PPP 超高速デジタルサービスを提供できます。PPPoE は、3 つの関係者、つまり企業、電話会社、サービスプロバイダを通して Ethernet トンネルを作成することにより、このサービスを実現します。

- PPPoE の動作の概要と説明については、422 ページの「PPPoE の概要」を参照してください。

- PPPoE トンネルの設定作業については、第 20 章「PPPoE トンネルの設定 (手順)」を参照してください。

この節では、PPPoE コマンドおよびファイルについて詳しく説明します。概要を次の表に示します。

表 22–2 PPPoE のコマンドと構成ファイル

ファイルまたはコマンド	説明	参照先
/etc/ppp/pppoe	PPPoE がシステムに設定したすべてのトンネルに対してデフォルトで適用される特性を含むファイル	539 ページの「/etc/ppp/pppoe ファイル」
/etc/ppp/pppoe.*device*	PPPoE がトンネルに使用する特定のインタフェースの特性を含むファイル	542 ページの「/etc/ppp/pppoe.*device* ファイル」
/etc/ppp/pppoe.if	PPPoE が設定したトンネルが動作する Ethernet インタフェースを一覧表示したファイル	537 ページの「/etc/ppp/pppoe.if ファイル」
/usr/sbin/sppptun	PPPoE トンネルに関係する Ethernet インタフェースを設定するためのコマンド	538 ページの「/usr/sbin/sppptun コマンド」
/usr/lib/inet/pppoed	PPPoE を使ってトンネルを設定するためのコマンドとオプション	539 ページの「/usr/lib/inet/pppoed デーモン」

PPPoE のインタフェースを設定するためのファイル

PPPoE トンネルの両端で使用されるインタフェースは、トンネルが PPP 通信をサポートする前に、あらかじめ設定しておく必要があります。設定には、/usr/sbin/sppptun および /etc/ppp/pppoe.if ファイルを使用します。これらのツールを使用して、すべての Solaris PPPoE クライアントおよび PPPoE アクセスサーバー上の Ethernet インタフェースを設定する必要があります。

/etc/ppp/pppoe.if ファイル

/etc/ppp/pppoe.if ファイルは、ホスト上の PPPoE トンネルで使用されるすべての Ethernet インタフェースの名前を一覧表示します。このファイルはシステムのブート時に処理され、ファイルに一覧表示されているインタフェースは PPPoE トンネルで使用するために plumb されます。

/etc/ppp/pppoe.if は明示的に作成する必要があります。各行ごとにインタフェース名を 1 つずつ入力して PPPoE 用に設定します。

次に、PPPoE トンネルに 3 つのインタフェースを提供するサーバーの /etc/ppp/pppoe.if ファイルの例を示します。

```
# cat /etc/ppp/pppoe.if
hme1
hme2
hme3
```

PPPoE クライアントは通常、/etc/ppp/pppoe.if に一覧表示されているインタフェースを 1 つだけ使用します。

/usr/sbin/sppptun コマンド

/usr/sbin/sppptun コマンドを使用すると、PPPoE トンネルで使用する Ethernet インタフェースを手動で plumb したり unplumb したりできます。これに対して、/etc/ppp/pppoe.if はシステムの起動時だけ読み取られます。これらのインタフェースは、/etc/ppp/pppoe.if に一覧表示されているインタフェースと一致する必要があります。

sppptun は、PPPoE トンネルで使用する Ethernet インタフェースを ifconfig コマンドと同様の方法で plumb します。ifconfig とは異なり、2 つの Ethernet プロトコル番号が必要なため、PPPoE をサポートするにはインタフェースを 2 回 plumb する必要があります。

sppptun の基本的な構文を次に示します。

```
# /usr/sbin/sppptun plumb pppoed device-name
      device-name:pppoed
# /usr/sbin/sppptun plumb pppoe device-name
      device-name:pppoe
```

この構文で、*device-name* は PPPoE に plumb されるデバイス名です。

上の 1 つめの sppptun コマンドを実行したときは、発見プロトコル pppoed がインタフェースに plumb されます。2 つめの sppptun を実行したときは、セッションプロトコル pppoe が plumb されます。sppptun は、plumb されたインタフェースの名前を表示します。必要な場合は、この名前を使ってインタフェースを unplumb します。

詳細は、sppptun(1M) のマニュアルページを参照してください。

インタフェースを管理する sppptun コマンドの例

次の例は、/usr/sbin/sppptun を使用して PPPoE のインタフェースを手動で plumb します。

```
# /usr/sbin/sppptun plumb pppoed hme0
hme0:pppoed
# /dev/sppptun plumb pppoe hme0
 hme0:pppoe
```

次の例は、PPPoE に plumb されたアクセスサーバー上のインタフェースを表示します。

```
# /usr/sbin/sppptun query
hme0:pppoe
hme0:pppoed
hme1:pppoe
hme1:pppoed
hme2:pppoe
hme2:pppoed
```

次の例は、インタフェースを unplumb する方法を示しています。

```
# sppptun unplumb hme0:pppoed
# sppptun unplumb hme0:pppoe
```

PPPoE アクセスサーバーのコマンドとファイル

DSL のサービスまたはサポートを顧客に提供するサービスプロバイダは、Solaris PPPoE を実行するアクセスサーバーを使用できます。PPPoE アクセスサーバーとクライアントは、従来のクライアントとサーバーの関係で機能します。この関係は、ダイアルアップリンクでのダイアルアウトマシンとダイアルインサーバーの関係に似ています。つまり、ある PPPoE システムが通信を開始し、別の PPPoE システムが応答します。これに対して、PPP プロトコルにはクライアントとサーバーの関係という概念はなく、両方のマシンが同等のピアとみなされます。

PPPoE アクセスサーバーを設定するコマンドおよびファイルには、次が含まれます。

- 538 ページの「/usr/sbin/sppptun コマンド」
- 539 ページの「/usr/lib/inet/pppoed デーモン」
- 539 ページの「/etc/ppp/pppoe ファイル」
- 542 ページの「/etc/ppp/pppoe.*device* ファイル」
- 545 ページの「pppoe.so 共有オブジェクト」

/usr/lib/inet/pppoed デーモン

pppoed デーモンは、将来の PPPoE クライアントからサービス提供用ブロードキャストを受け取ります。さらに、pppoed は PPPoE トンネルのサーバー側とネゴシエーションし、PPP デーモン pppd をそのトンネル上で実行します。

pppoed サービスは、/etc/ppp/pppoe および /etc/ppp/pppoe.*device* ファイルで設定します。システムのブート時に /etc/ppp/pppoe が存在する場合は、pppoed が自動的に実行します。コマンド行で /usr/lib/inet/pppoed と入力することにより、pppoed デーモンを明示的に実行することもできます。

/etc/ppp/pppoe ファイル

/etc/ppp/pppoe ファイルは、アクセスサーバーが提供するサービスと、PPP が PPPoE トンネル上でどのように実行するかを定義するオプションを説明します。インタフェースごとに個別にサービスを定義することも、広域的にアクセスサーバー上の

すべてのインタフェースに対してサービスを定義することもできます。アクセスサーバーは、将来の PPPoE クライアントからのブロードキャストに応答して、/etc/ppp/pppoe ファイル内の情報を送信します。

次に、/etc/ppp/pppoe の基本的な構文を示します。

```
global-options
service  service-name
      service-specific-options
      device  interface-name
```

パラメータの意味は次のとおりです。

global-options
: /etc/ppp/pppoe ファイルのデフォルトのオプションを設定します。このオプションには、pppoed または pppd で使用可能なオプションはすべて使用できます。オプションの完全なリストについては、pppoed(1M) および pppd(1M) のマニュアルページを参照してください。

  たとえば、この *global-options* には、PPPoE トンネルで使用できる Ethernet インタフェースを一覧表示する必要があります。/etc/ppp/pppoe でデバイスを定義しないと、インタフェースでサービスを提供できません。

  devices をグローバルオプションとして定義するには、次の形式を使用します。

  device interface <,interface>

  interface は、サービスが将来の PPPoE クライアントを待つインタフェースを指定します。複数のインタフェースがサービスに関連付けられている場合は、名前をコンマで区切って指定します。

service service-name
: *service-name* というサービスの定義を開始します。*service-name* には、提供されるサービスに適した任意の文字列を指定できます。

service-specific-options
: このサービスに固有の PPPoE および PPP のオプションを表示します。

device *interface-name*
: 上記で一覧表示したサービスを利用できるインタフェースを指定します。

/etc/ppp/pppoe のその他のオプションについては、pppoed(1M) および pppd(1M) のマニュアルページを参照してください。

次に、典型的な /etc/ppp/pppoe ファイルの例を示します。

例 22–2　基本的な /etc/ppp/pppoe ファイル

```
device hme1,hme2,hme3
service internet
   pppd "name internet-server"
service intranet
   pppd "192.168.1.1:"
service debug
   device hme1
   pppd "debug name internet-server"
```

このファイルでは、次の値が適用されています。

hme1,hme2,hme3	PPPoE トンネルに使用されるアクセスサーバー上の 3 つのインタフェース。
service internet	想定クライアントに対して internet というサービスを通知します。また、サービスを提供するプロバイダは internet の定義方法についても決定します。たとえば、プロバイダは、internet とは、インターネットへのアクセスだけでなく、さまざまな IP サービスを意味するものと解釈する場合があります。
pppd	呼び出し元が pppd を呼び出したときに使用されるコマンド行オプションを設定します。"name internet-server" オプションは、ローカルマシン (アクセスサーバー) の名前を internet-server と付けます。
service intranet	intranet という別のサービスを想定クライアントに通知します。
pppd "192.168.1.1:"	呼び出し元が pppd を呼び出したときに使用されるコマンド行オプションを設定します。呼び出し元が pppd を呼び出すと、ローカルマシン (アクセスサーバー) の IP アドレスとして 192.168.1.1 が設定されます。
service debug	PPPoE 用に定義されているインタフェースに 3 番目のサービス、デバッグを通知します。
device hme1	PPPoE トンネルに対するデバッグを hme1 に限定します。
pppd "debug name internet-server"	呼び出し元が pppd を起動したときに使用されるコマンド行オプション、この場合は PPP デバッグをローカルマシン internet-server に設定します。

/etc/ppp/pppoe.*device* ファイル

/etc/ppp/pppoe.*device* ファイルは、PPPoE アクセスサーバーの 1 つのインタフェース上で提供されるサービスを説明します。PPP が PPPoE トンネル上でどのように実行するかを定義するオプションも説明します。/etc/ppp/pppoe. *device* はオプションのファイルで、グローバルの /etc/ppp/pppoe とまったく同様に動作します。ただし、/etc/ppp/pppoe.*device* がインタフェース用に定義されている場合、そのインタフェースでは、このファイルのパラメータが、/etc/ppp/pppoe で定義されているグローバルパラメータより優先されます。

次に、/etc/ppp/pppoe.*device* の基本的な構文を示します。

```
service service-name
        service-specific-options
service another-service-name
        service-specific-options
```

上記の構文と /etc/ppp/pppoe の構文の違いは、539 ページの「/etc/ppp/pppoe ファイル」で示した device オプションを使用できない点だけです。

pppoe.so プラグイン

pppoe.so は PPPoE 共有オブジェクトファイルで、PPPoE のアクセスサーバーおよびクライアントによって呼び出されます。このファイルは、MTU および MRU を 1492 に制限し、ドライバからのパケットにフィルタをかけ、pppoed とともに PPPoE トンネルをネゴシエートします。アクセスサーバー側では、pppoe.so は pppd デーモンによって自動的に呼び出されます。

アクセスサーバー構成のための PPPoE および PPP ファイルの使用

この節では、あるアクセスサーバーを構成するために使用するすべてのファイルのサンプルを紹介します。このアクセスサーバーはマルチホームで、3 つのサブネット green、orange、および purple が接続されています。pppoed は、サーバー上で root として実行します。これはデフォルトの動作です。

PPPoE クライアントは、hme0 および hme1 インタフェースを通じて orange および purple ネットワークにアクセスできます。クライアントは、標準の UNIX ログインを使ってサーバーにログインします。サーバーは、クライアントを PAP を使って認証します。

green ネットワークは、クライアントに通知されません。クライアントが green にアクセスできるためには、直接「green-net」を指定し、CHAP 認証資格を提供しなければなりません。さらに、クライアント joe および mary だけが、静的 IP アドレスを使用して green ネットワークにアクセスできます。

例 22–3　アクセスサーバー用の /etc/ppp/pppoe ファイル

```
service orange-net
     device hme0,hme1
     pppd "require-pap login name orange-server orange-server:"
service purple-net
     device hme0,hme1
     pppd "require-pap login name purple-server purple-server:"
service green-net
     device hme1
     pppd "require-chap name green-server green-server:"
     nowildcard
```

このサンプルは、アクセスサーバーから使用できるサービスを説明します。1 番目の service セクションは、orange ネットワークのサービスを説明します。

```
service orange-net
     device hme0,hme1
     pppd "require-pap login name orange-server orange-server:"
```

クライアントは、hme0 および hme1 インタフェース上で orange ネットワークにアクセスできます。pppd コマンドに指定されているオプションにより、サーバーは、想定クライアントからの PAP 資格を要求します。また、pppd オプションはサーバーの名前を orange-server に設定します。この名前は pap-secrets ファイルで使用されます。

purple ネットワーク用の service セクションは、ネットワーク名とサーバー名が異なる以外は、orange ネットワーク用の service セクションと同じです。

次の service セクションは、green ネットワークのサービスを説明します。

```
service green-net
     device hme1
     pppd "require-chap name green-server green-server:"
     nowildcard
```

このセクションは、クライアントのアクセスをインタフェース hme1 に限定しています。pppd コマンドに指定されているオプションにより、サーバーは、想定クライアントからの CHAP 資格を要求します。また、pppd オプションはサーバー名を green-server に設定しています。この名前は chap-secrets ファイルで使用されます。nowildcard オプションは、green ネットワークの存在をクライアントに通知しないことを指定します。

このアクセスサーバーのシナリオでは、次のような /etc/ppp/options ファイルを設定する場合があります。

例 22–4　アクセスサーバー用の /etc/ppp/options ファイル

```
     auth
     proxyarp
```

例 22–4 アクセスサーバー用の /etc/ppp/options ファイル (続き)

```
nodefaultroute
name no-service    # don't authenticate otherwise
```

name no-service オプションは、通常、PAP または CHAP 認証時に検索されるサーバー名を無効にします。サーバーのデフォルト名は、/usr/bin/hostname コマンドを使って得られます。前述の例の name オプションは、サーバー名を no-service に変更します。no-service は、pap または chap-secrets ファイルで見つかる可能性がほとんどない名前です。この処理により、任意のユーザーが pppd を実行したり、/etc/ppp/options で設定されている auth および name オプションを上書きするのを防ぐことができます。pppd は、no-service のサーバー名ではクライアントのシークレットを見つけることができないため、失敗します。

このアクセスサーバーのシナリオでは、次の /etc/hosts ファイルを使用します。

例 22–5 アクセスサーバー用の /etc/hosts ファイル

```
172.16.0.1    orange-server
172.17.0.1    purple-server
172.18.0.1    green-server
172.18.0.2    joes-pc
172.18.0.3    marys-pc
```

次に、orange および purple ネットワークにアクセスしようとするクライアントの PAP 認証に使用する /etc/ppp/pap-secrets ファイルを示します。

例 22–6 アクセスサーバー用の /etc/ppp/pap-secrets ファイル

```
* orange-server "" 172.16.0.2/16+
* purple-server "" 172.17.0.2/16+
```

次に、CHAP 認証に使用される /etc/ppp/chap-secrets ファイルを示します。joe および mary というクライアントだけがファイルに一覧表示されていることに注意してください。

例 22–7 アクセスサーバー用の /etc/ppp/chap-secrets ファイル

```
 joe green-server "joe's secret" joes-pc
mary green-server "mary's secret" marys-pc
```

PPPoE クライアントのコマンドとファイル

DSL モデム上で PPP を実行するには、マシンが PPPoE クライアントになる必要があります。PPPoE を実行するためにインタフェースを plumb し、次に pppoec

ユーティリティーを使ってアクセスサーバーの存在を「発見」する必要があります。その後、クライアントは DSL モデム上に PPPoE トンネルを作成し PPP を実行できます。

PPPoE クライアントは、従来のクライアント - サーバーモデルでアクセスサーバーに接続します。PPPoE トンネルはダイアルアップリンクではありませんが、ほぼ同じような方法で構成され、操作されます。

PPPoE クライアントを設定するコマンドおよびファイルには、次が含まれます。

- 538 ページの「/usr/sbin/sppptun コマンド」
- 545 ページの「/usr/lib/inet/pppoec ユーティリティー」
- 545 ページの「pppoe.so 共有オブジェクト」
- 514 ページの「/etc/ppp/peers/*peer-name* ファイル」
- 509 ページの「/etc/ppp/options 構成ファイル」

/usr/lib/inet/pppoec ユーティリティー

/usr/lib/inet/pppoec ユーティリティーは、PPPoE トンネルのクライアント側をネゴシエーションします。pppoec は、Solaris PPP 4.0 の chat ユーティリティーに似ています。pppoec は直接起動しません。直接起動するのではなく、pppd の connect オプションの引数として /usr/lib/inet/pppoec を起動します。

pppoe.so 共有オブジェクト

pppoe.so は PPPoE 共有オブジェクトで、PPPoE によって読み込まれ、PPPoE 機能をアクセスサーバーとクライアントに提供します。共有オブジェクト pppoe.so は、MTU および MRU を 1492 に制限し、ドライバからのパケットにフィルタをかけ、実行時 PPPoE メッセージを処理します。

クライアント側では、ユーザーが plugin pppoe.so オプションを指定すると、pppd が pppoe.so を読み込みます。

アクセスサーバーピアを定義するための /etc/ppp/peers/*peer-name* ファイル

アクセスサーバーが pppoec によって発見されるように定義する場合は、pppoec および pppd デーモンの両方に適用されるオプションを使用します。アクセスサーバーの /etc/ppp/peers/*peer-name* ファイルは次のパラメータを必要とします。

- sppptun – PPPoE トンネルが使用するシリアルデバイスの名前。
- plugin pppoe.so – pppd に pppoe.so 共有オブジェクトを読み込むように指示します。
- connect "/usr/lib/inet/pppoec *device*" – 接続を開始します。次に、PPPoE に plumb されているインタフェース *device* 上で pppoec ユーティリティーを起動します。

/etc/ppp/peers/ *peer-name* ファイル内の残りのパラメータは、サーバー上の PPP リンクに適用されます。ダイアルアウトマシン上の /etc/ppp/peers/*peer-name* と同じオプションを使用します。オプションの数を PPP リンクで必要な最小数に制限するようにしてください。

次の例は、481 ページの「PPPoE アクセスサーバーピアを定義する方法」で紹介されています。

例 22–8 リモートアクセスサーバーを定義するための /etc/ppp/peers/*peer-name*

```
# cat /etc/ppp/peers/dslserve
sppptun
plugin pppoe.so
connect "/usr/lib/inet/pppoec hme0"
noccp
noauth
user Red
password redsecret
noipdefault
defaultroute
```

このファイルは、アクセスサーバー dslserve に PPPoE トンネルと PPP リンクを設定するときに使用するパラメータを定義します。オプションには、次が含まれます。

オプション	説明
sppptun	sppptun をシリアルデバイスの名前として定義します。
plugin pppoe.so	pppd に pppoe.so 共有オブジェクトを読み込むように指示します。
connect "/usr/lib/inet/pppoec hme0"	pppoec を実行し、PPPoE トンネルおよび PPP リンク用のインタフェースとして hme0 を指定します。
noccp	接続上で CCP 圧縮をオフに設定します。 注 – 多くの ISP は独自の圧縮アルゴリズムだけを使用します。公開された CCP アルゴリズムをオフにすると、ネゴシエーションの時間を節約し、偶発的な相互運用性の問題を避けることができます。
noauth	pppd 認証資格をアクセスサーバーに要求するのを 停止します。ほとんどの ISP は認証資格を顧客に提供しません。
user Red	アクセスサーバーによる PAP 認証に必要なクライアントのユーザー名として Red の名前を設定します。
password redsecret	PAP 認証のためにアクセスサーバーに提供されるパスワードとして redsecret を定義します。

オプション	説明
`noipdefault`	初期 IP アドレスとして 0.0.0.0 を割り当てます。
`defaultroute`	IPCP ネゴシエーション後にデフォルトの IPv4 経路指定をインストールするよう `pppd` に指示します。接続がシステムのインターネットへの接続である場合、`/etc/ppp/peers/`*peer-name* 内に `defaultroute` を含める必要があります。PPPoE クライアントの場合これにあてはまります。

◆ ◆ ◆ 第 23 章

非同期 Solaris PPP から Solaris PPP 4.0 への移行 (手順)

Solaris OS の以前のバージョンでは、別の PPP 実装である非同期 Solaris PPP (asppp) が提供されていました。asppp を実行するピアを最新の PPP 4.0 に更新する場合は、変換スクリプトを実行する必要があります。この章では、PPP 変換に関する次のトピックについて説明します。

- 549 ページの「asppp ファイルを変換する前に」
- 552 ページの「asppp2pppd 変換スクリプトの実行 (作業)」

この章では、サンプルの asppp 構成を使用して、PPP 変換を実施する方法について説明します。Solaris PPP 4.0 と asppp の相違点については、410 ページの「使用する Solaris PPP のバージョン」を参照してください。

asppp ファイルを変換する前に

変換スクリプト /usr/sbin/asppp2pppd を使用して、標準 asppp 構成を構成する次のファイルを変換できます。

- /etc/asppp.cf – 非同期 PPP 構成ファイル
- /etc/uucp/Systems – リモートピアの特性を記述する UUCP ファイル
- /etc/uucp/Devices – ローカルマシン上のモデムを記述する UUCP ファイル
- /etc/uucp/Dialers – /etc/uucp/Devices ファイルに記述されているモデムが使用するログインシーケンスが含まれる UUCP ファイル

asppp については、http://docs.sun.com に掲載されている「Solaris 8 System Administrator Collection - Japanese」の『Solaris 8 のシステム管理 (第 3 巻)』を参照してください。

/etc/asppp.cf 構成ファイルの例

553 ページの「asppp から Solaris PPP 4.0 に変換する方法」に示す手順は、次の /etc/asppp.cf ファイルを使用します。

```
#
ifconfig ipdptp0 plumb mojave gobi up

path
 inactivity_timeout 120     # Approx. 2 minutes
    interface ipdptp0
    peer_system_name Pgobi    # The name we log in with (also in
                                  # /etc/uucp/Systems
```

このファイルには次のパラメータが含まれています。

`ifconfig ipdptp0 plumb mojave gobi up`	ifconfig コマンドを実行し、ローカルマシン mojave の PPP インタフェース ipdptp0 からリモートピア gobi へのリンクを確立する
`inactivity_timeout 120`	2 分間アクティブでない回線を終了する
`interface ipdptp0`	ダイアルアウトマシン上のインタフェース ipdptp0 を非同期 PPP に構成する
`peer_system_name Pgobi`	リモートピアの名前 Pgobi を指定する

/etc/uucp/Systems ファイルの例

553 ページの「asppp から Solaris PPP 4.0 に変換する方法」 に示す手順は、次の /etc/uucp/Systems ファイルを使用します。

```
#ident  "@(#)Systems    1.5  92/07/14 SMI"  /* from SVR4 bnu:Systems 2.4 */
#

# .
# .
Pgobi Any ACU 38400 15551212 in:--in: mojave word: sand
```

このファイルには次のパラメータが含まれています。

Pgobi	Pgobi をリモートピアのホスト名として使用します。
Any ACU	ダイアルアウトマシン mojave 上のモデムに、任意の時点で Pgobi 上のモデムとリンクを確立するように指示します。Any ACU は「/etc/uucp/Devices ファイル内で ACU を探す」ことを意味します。

38400	リンクの最大速度として 38400 を設定します。
15551212	Pgobi の電話番号を指定します。
in:–in: mojave word: sand	Pgobi が必要とするログインスクリプトを定義して、ダイアルアウトマシン mojave を認証します。

/etc/uucp/Devices ファイルの例

553 ページの「asppp から Solaris PPP 4.0 に変換する方法」に示す手順は、次の /etc/uucp/Devices ファイルを使用します。

```
#ident  "@(#)Devices  1.6  92/07/14 SMI"  /* from SVR4 bnu:Devices 2.7 */

.
.
#

TCP,et - - Any TCP -
.
.
#
ACU cua/b - Any hayes
# 0-7 are on a Magma 8 port card
Direct cua/0 - Any direct
Direct cua/1 - Any direct
Direct cua/2 - Any direct
Direct cua/3 - Any direct
Direct cua/4 - Any direct
Direct cua/5 - Any direct
Direct cua/6 - Any direct
Direct cua/7 - Any direct
# a is the console port (aka "tip" line)
Direct cua/a - Any direct
# b is the aux port on the motherboard
Direct cua/b - Any direct
# c and d are high speed sync/async ports
Direct cua/c - Any direct
Direct cua/d - Any direct
```

このファイルは、シリアルポート cua/b に接続されている Hayes モデムをサポートします。

/etc/uucp/Dialers ファイルの例

553 ページの「asppp から Solaris PPP 4.0 に変換する方法」に示す手順は、次の /etc/uucp/Dialers ファイルを使用します。

```
#
#       <Much information about modems supported by Solaris UUCP>
```

```
penril    =W-P    "" \d > Q\c : \d- > s\p9\c )-W\p\r\ds\p9\c-) y\c : \E\TP > 9\c OK
ventel    =&-%    "" \r\p\r\c $ k\c ONLINE!
vadic    =K-K    "" \005\p *-\005\p-*\005\p-* D\p BER? \E\T\e \r\c LINE
develcon ""    "" \pr\ps\c est:\007 \E\D\e \n\007
micom    ""    "" \s\c NAME? \D\r\c GO
direct
#
#
#
#   Hayes Smartmodem -- modem should be set with the configuration
#   switches as follows:
#
#       S1 - UP        S2 - UP        S3 - DOWN    S4 - UP
#       S5 - UP        S6 - DOWN    S7 - ?        S8 - DOWN
#
hayes    =,-,    "" \dA\pTE1V1X1Q0S2=255S12=255\r\c OK\r \EATDT\T\r\c CONNECT
```

<much more information about modems supported by Solaris UUCP>

このファイルには、あらゆるタイプのモデムの chat スクリプトが含まれます。`/etc/uucp/Dialers` ファイルでサポートされている Hayes モデムの chat スクリプトも含まれます。

asppp2pppd 変換スクリプトの実行 (作業)

`/usr/sbin/asppp2pppd` スクリプトは、`/etc/asppp.cf` に含まれる PPP 情報と PPP 関連の UUCP ファイルを、Solaris PPP 4.0 ファイル内の適切な場所にコピーします。

作業の前提条件

次の作業に進む前に、次のことを完了しておく必要があります。

- `asppp` と UUCP 構成ファイルがあるマシン上に Solaris 9 または 10 リリースをインストールする
- PPP ファイルがあるマシン、たとえば `mojave` 上でスーパーユーザーになる

▼ asppp から **Solaris PPP 4.0** に変換する方法

1 変換スクリプトを実行します。

```
# /usr/sbin/asppp2pppd
```

変換処理が開始し、画面に次のようなメッセージが表示されます。

```
This script provides only a suggested translation for your existing aspppd
configuration.  You will need to evaluate for yourself whether the translation
is appropriate for your operating environment.
Continue [Yn]?
```

2 「**Y**」と入力して、処理を継続します。

画面に次のようなメッセージが表示されます。

```
Chat cannot do echo checking; requests for this removed.
Adding 'noauth' to /etc/ppp/options

Preparing to write out translated configuration:
    1 chat file:
        1.  /etc/ppp/chat.Pgobi.hayes
    2 option files:
        2.  /etc/ppp/peers/Pgobi
        3.  /etc/ppp/options
    1 script file:
        4.  /etc/ppp/demand
```

新しい Solaris PPP 4.0 ファイルが生成されました。

▼ 変換結果を表示する方法

変換処理の最後に、`/usr/sbin/asppp2pppd` 変換スクリプトによって作成された Solaris PPP 4.0 ファイルを表示できます。次に示すオプションリストが表示されます。

```
Enter option number:
        1 - view contents of file on standard output
        2 - view contents of file using /usr/bin/less
        3 - edit contents of file using /usr/bin/vi
        4 - delete/undelete file from list
        5 - rename file in list
        6 - show file list again
        7 - escape to shell (or "!")
        8 - abort without saving anything
        9 - save all files and exit (default)
Option:
```

1 **1** を入力して、画面上にファイルの内容を表示します。

表示するファイルの番号の入力を求めるプロンプトが表示されます。

```
File number (1 .. 4):
```

この番号は、前述の手順 2 で示したように、変換処理中に表示された変換ファイルを示します。

2 1 を入力して、**chat** ファイル /etc/ppp/chat.Pgobi.hayes を表示します。

```
File number (1 .. 4):  1
"" \d\dA\p\pTE1V1X1Q0S2=255S12=255\r\c
OK\r ATDT\T\r\c
CONNECT \c
in:--in: mojave
word: sand
```

chat スクリプトには、サンプルの /etc/uucp/Dialers ファイルの hayes 行に記述されているモデムの “chat” 情報が含まれています。また、/etc/ppp/chat.Pgobi.hayes にはサンプルの /etc/uucp/Systems ファイルに記述されている Pgobi のログインシーケンスが含まれています。したがって、現時点では、chat スクリプトは /etc/ppp/chat.Pgobi.hayes ファイルにあります。

3 2 を入力して、ピアファイル /etc/ppp/peers/Pgobi を表示します。

```
File number (1 .. 4):  2
/dev/cua/b
38400
demand
idle 120
connect "/usr/bin/chat -f /etc/ppp/chat.Pgobi.hayes -T ’15551212’"
user NeverAuthenticate
mojave:gobi
```

/etc/uucp/Devices ファイル内のシリアルポート情報 (/dev/cua/b) と、/etc/asppp.cf ファイル内のリンク速度、アイドル時間、認証情報、ピア名が表示されています。“demand” は “demand” スクリプトを意味します。このスクリプトは、ダイアルアウトマシンがピア Pgobi に接続を試みるときに呼び出されます。

4 3 を入力して、ダイアルアウトマシン mojave 用に作成された /etc/ppp/options ファイルを表示します。

```
File number (1 .. 4):  3
#lock
noauth
```

/etc/ppp/options ファイル内の情報は /etc/asppp.cf ファイルから得られたものです。

5 4 を入力して、demand スクリプトの内容を表示します。

```
File number (1 .. 4):  4
/usr/bin/pppd file /etc/ppp/peers/Pgobi
```

このスクリプトが実行されると、pppd コマンドが実行されます。このコマンドは、/etc/ppp/peers/Pgobi を読み込んで、mojave と Pgobi の間のリンクを確立します。

6 9 を入力して、作成したファイルを保存し、変換スクリプトを終了します。

◆ ◆ ◆ 第 24 章

24 UUCP (概要)

この章では、 UNIX 間コピープログラム (UUCP) と、このプログラムのデーモンについて説明します。次の項目について説明します。

UUCP を使用すると、コンピュータシステム間で相互にファイルの転送とメールの交換を行えます。また、UUCP を使用して Usenet のような大規模なネットワークにコンピュータを接続することもできます。

Solaris OS では、HoneyDanBer UUCP とも呼ばれる基本ネットワークユーティリティー (BNU) バージョンの UUCP が提供されています。UUCP という用語はシステムを形成するすべてのファイルとユーティリティーを意味するものであり、uucp プログラムはそのシステムの一部にすぎません。UUCP のユーティリティーには、コンピュータ間でファイルをコピーするためのユーティリティー (uucp と uuto) から、リモートログインやリモートコマンド実行のためのユーティリティー (cu と uux) まで、さまざまなものがあります。

UUCP のハードウェア構成

UUCP は、次のハードウェア構成で利用できます。

直接リンク
: 2 つのマシンのシリアルポート間を RS-232 ケーブルで結ぶことにより、ほかのコンピュータとの間の直接リンクを作成できます。2 つのコンピュータが常時互いに通信を行い、両者の間の距離が 15m 以内の場合は、直接リンクを使用すると便利です。この制限距離は、短距離モデムを使用することによりある程度延長できます。

電話回線
: 高速モデムなどの自動呼び出し装置 (ACU) を使用すれば、通常の電話回線を介してほかのコンピュータと通信できます。モデム

は、UUCP が要求する電話番号をダイアルします。受信側のモデムは、着信に応答できなければなりません。

ネットワーク　UUCP は、TCP/IP またはその他のプロトコルファミリが機能するネットワークを介しても通信できます。コンピュータがネットワーク上でホストとして確立されていれば、そのネットワークに接続されているほかのどのホストとも通信できます。

この章では、UUCP ハードウェアをすでに設置、構成してあるものとして説明を進めます。モデムを設定する必要がある場合は、『Solaris のシステム管理 (基本編)』と、モデムに付属のマニュアルを参照してください。

UUCP ソフトウェア

Solaris インストールプログラムを実行するときに全体ディストリビューションを選択していれば、UUCP ソフトウェアは自動的に組み込まれています。あるいは、`pkgadd` を使用して UUCP を単独で追加することもできます。UUCP のプログラムは、デーモン、管理プログラム、およびユーザープログラムの 3 種類に分類されます。

UUCP デーモン

UUCP システムには、`uucico`、`uuxqt`、`uusched`、および `in.uucpd` の 4 つのデーモンがあります。これらのデーモンは、UUCP のファイル転送とコマンド実行を処理します。これらのデーモンは、必要に応じて、シェルから手動で実行することもできます。

`uucico`　リンクに使用するデバイスを選択し、リモートコンピュータへのリンクを確立し、必要なログインシーケンスとアクセス権の検査を行います。また、データファイルを転送し、ファイルを実行し、結果をログに記録し、転送の完了をメールによりユーザーに通知します。`uucico` は、UUCP ログインアカウント用の「ログインシェル」として働きます。ローカル `uucico` デーモンはリモートマシンを呼び出して、セッションの間、リモート `uucico` デーモンと直接通信します。

必要なファイルがすべて作成されたら、`uucp`、`uuto`、および `uux` プログラムが `uucico` デーモンを実行してリモートコンピュータに接続します。`uusched` と `Uutry` は、どちらも `uucico` を実行します。詳細は、`uucico`(1M) のマニュアルページを参照してください。

`uuxqt`　リモート実行要求を処理します。このデーモンは、スプールディレクトリを検索して、リモートコンピュータから送られた実行ファイル (名前は常に `X.`*file*) を見つけます。`X.`*file* が見つかると、`uuxqt` はそのファイル

を開いて、実行に必要なデータファイルのリストを取得します。次に、必要なデータファイルが使用可能でアクセスできるかどうかを確認します。ファイルが使用可能であれば、uuxqt は Permissions ファイルを調べて、要求されたコマンドを実行する権限があるかどうかを確認します。uuxqt デーモンは、cron により起動される uudemon.hour シェルスクリプトから実行されます。詳細は、uuxqt(1M) のマニュアルページを参照してください。

uusched　スプールディレクトリ内でキューに入っている作業をスケジュールします。uusched デーモンは、cron により起動される uudemon.hour シェルスクリプトによって、ブート時に最初に実行されます。詳細は、uusched(1M) のマニュアルページを参照してください。uusched は uucico デーモンを起動する前に、リモートコンピュータを呼び出す順序をランダム化します。

in.uucpd　ネットワークを介した UUCP 接続をサポートします。リモートホスト上の inetd は、UUCP 接続が確立されるたびに in.uucpd を呼び出します。次に、uucpd がログイン名を要求します。呼び出し側ホストの uucico は、これに対してログイン名を応答しなければなりません。次に in.uucpd はパスワードを要求します (不要な場合を除く)。詳細は、in.uucpd(1M) のマニュアルページを参照してください。

UUCP 管理プログラム

ほとんどの UUCP 管理プログラムは /usr/lib/uucp に置かれています。基本データベースファイルの多くは、/etc/uucp に置かれています。ただし、uulog だけは例外で、これは /usr/bin に置かれています。uucp ログイン ID のホームディレクトリは /usr/lib/uucp です。su または login を使用して管理プログラムを実行するときには、uucp ユーザー ID を使用します。このユーザー ID は、プログラムとスプールデータファイルを所有しています。

uulog　指定したコンピュータのログファイルの内容を表示する。ログファイルは、このマシンが通信する各リモートコンピュータごとに作成される。ログファイルには、uucp、uuto、uux の使用が記録される。詳細は、uucp(1C) のマニュアルページを参照

uucleanup　スプールディレクトリをクリーンアップする。これは通常、cron によって起動される uudemon.cleanup シェルスクリプトから実行される。詳細は、uucleanup(1M) のマニュアルページを参照

Uutry　呼び出し処理機能をテストし、簡単なデバッグを行うことができる。uucico デーモンを呼び出して、このマシンと指定されたリモートコンピュータとの間の通信リンクを確立する。詳細は、Uutry(1M) のマニュアルページを参照

uucheck　UUCP のディレクトリ、プログラム、およびサポートファイルの有無を検査する。また、/etc/uucp/Permissions ファイルの所定の部分に、明らかな構文エラーがないかどうかも検査する。詳細は、uucheck(1M) のマニュアルページを参照

UUCP ユーザープログラム

UUCP のユーザープログラムは /usr/bin にあります。これらのプログラムを使用するのに、特別な権限は必要ありません。

cu　このマシンをリモートコンピュータに接続して、ユーザーが両方のマシンに同時にログインできるようにする。cu を使用すれば、接続したリンクを切断することなく、どちらのマシンでもファイルを転送したり、コマンドを実行したりできる。詳細は、cu(1C) のマニュアルページを参照

uucp　あるマシンから別のマシンへファイルをコピーする。uucp は作業ファイルとデータファイルを作成し、転送するジョブをキューに入れ、uucico デーモンを呼び出す。このデーモンは、リモートコンピュータへの接続を試みる。詳細は、uucp(1C) のマニュアルページを参照

uuto　ローカルマシンから、リモートマシン上の公開スプールディレクトリ /var/spool/uucppublic/receive にファイルをコピーする。uucp はリモートマシン上のアクセス可能な任意のディレクトリにファイルをコピーするのに対して、uuto は所定のスプールディレクトリにファイルを格納し、リモートユーザーに uupick を使用してそのファイルを取り出すように指示する。詳細は、uuto(1C) のマニュアルページを参照

uupick　uuto を使用してコンピュータにファイルが転送されてきたときに、/var/spool/uucppublic/receive からファイルを取得する。詳細は、uuto(1C) のマニュアルページを参照

uux　リモートマシン上でコマンドを実行するために必要な作業ファイル、データファイル、および実行ファイルを作成する。詳細は、uux(1C) のマニュアルページを参照

uustat　要求された転送 (uucp、uuto、uux) の状態を表示する。また、キューに入っている転送を制御する手段も提供する。詳細は、uustat(1C) のマニュアルページを参照

UUCP データベースファイル

UUCP 設定の主要部分の 1 つは、UUCP データベースを形成するファイルを構成することです。これらのファイルは /etc/uucp ディレクトリにあります。マシン上で UUCP または asppp を設定するには、これらのファイルを編集する必要があります。使用できるファイルを次に示します。

Config	変数パラメータのリストが入っている。これらのパラメータは、ネットワークを構成するために手動で設定できる
Devconfig	ネットワーク通信を構成するために使用される
Devices	ネットワーク通信を構成するために使用される
Dialcodes	Systems ファイルのエントリの電話番号フィールド内で使用できるダイアルコード省略名が入っている。これは必須ではないが、UUCP のほかに asppp でも使用できる
Dialers	リモートコンピュータとの接続を確立するときに、モデムとのネゴシエーションを行うために必要な文字列が入っている。これは、UUCP のほかに asppp でも使用される
Grades	ジョブの処理順序と、ジョブの各処理順序に関連付けられたアクセス権を定義する。これらは、リモートコンピュータのキューにジョブを入れる際に、ユーザーが指定できる
Limits	このマシンで同時に実行できる uucico、uuxqt、および uusched の最大数を定義する
Permissions	このマシンにファイルを転送したり、コマンドを実行しようとしているリモートホストに与えられるアクセスのレベルを定義する
Poll	このシステムがポーリングするマシンと、ポーリングする時刻を定義する
Sysfiles	uucico と cu が、Systems、Devices、および Dialers ファイルとして、別のファイルや複数のファイルを使用するときに、その割り当てを行う
Sysname	TCP/IP ホスト名の他に、各マシンに固有の UUCP 名を定義できる
Systems	uucico デーモン、cu、および asppp が、リモートコンピュータへのリンクを確立するために必要とする情報が入っている。この情報には次のものが含まれる。 ■ リモートホストの名前 ■ リモートホストに対応する接続デバイス名 ■ そのホストに接続できる日時 ■ 電話番号 ■ ログイン ID

- パスワード

サポートデータベースの一部とみなすことのできるファイルが他にもいくつかありますが、それらは、リンクの確立とファイルの転送には直接関係しません。

UUCP データベースファイルの構成設定

UUCP データベースは、561 ページの「UUCP データベースファイル」に示したファイルから構成されます。ただし、基本的な UUCP 構成に関係する重要なファイルは次に示すものだけです。

- `/etc/uucp/Systems`
- `/etc/uucp/Devices`
- `/etc/uucp/Dialers`

`asppp` は UUCP データベースの一部を使用するので、`asppp` を構成する予定がある場合は、少なくともこれらのデータベースファイルだけは理解しておく必要があります。これらのデータベースを構成してしまえば、その後の UUCP の管理はきわめて簡単です。通常、`Systems` ファイルを最初に編集し、次に `Devices` ファイルを編集します。`/etc/uucp/Dialers` ファイルは、普通はデフォルトのままで使用できますが、デフォルトファイルに含まれていないダイアラを追加する予定がある場合は編集が必要になります。基本的な UUCP 構成と `asppp` 構成には、さらに次のファイルを加えることもできます。

- `/etc/uucp/Sysfiles`
- `/etc/uucp/Dialcodes`
- `/etc/uucp/Sysname`

これらのファイルは互いに関係しながら機能するので、何らかの変更を加える場合は、全部のファイルの内容を理解しておくことが必要です。あるファイルのエントリに変更を加えた場合に、別のファイル内の関連エントリに対しても変更が必要になることがあります。561 ページの「UUCP データベースファイル」に挙げたその他のファイルは、上記のファイルほど緊密な相互関係を持っていません。

注 – `asppp` が使用するファイルはこの節で説明するものだけです。ほかの UUCP データベースファイルは使用しません。

◆ ◆ ◆ 第 25 章

25 UUCP の管理 (手順)

この章では、使用するマシンに合わせてデータベースファイルを変更したあと、UUCP 処理を起動する方法について説明します。この章には、Solaris OS が動作するマシンで UUCP を構成し保守するための、手順と障害の解明についての情報が記載されています。

UUCP 管理 (作業マップ)

次の表に、この章で説明する手順の参照先と、各手順についての簡単な説明を示します。

表 25–1 UUCP 管理の作業マップ

作業	説明	参照先
リモートマシンにユーザーシステムへのアクセスを許可する	`/etc/passwd` ファイルを編集し、ユーザーのシステムへのアクセスを許可するマシンを識別するようエントリを追加する	564 ページの「UUCP ログインの追加方法」
UUCP を起動する	UUCP の起動用に提供されているシェルスクリプトを使用する	565 ページの「UUCP の起動方法」
UUCP を TCP/IP ネットワーク上で有効にする	`/etc/inetd.conf` ファイルと `/etc/uucp/Systems` ファイルを編集し、TCP/IP 用の UUCP を起動する	567 ページの「TCP/IP 用 UUCP の起動方法」
UUCP に起こりがちな問題を解決する	モデムまたは ACU の異常を確認するための診断手順を実行する	570 ページの「モデムまたは ACU の障害確認方法」

表 25–1 UUCP 管理の作業マップ (続き)

作業	説明	参照先
	送信をデバッグするための診断手順を実行する	570 ページの「送信に関するデバッグ方法」

UUCP のログインの追加

リモートマシンからの UUCP (uucico) 着信要求が正しく取り扱われるように、各リモートマシンはローカルシステム上にログインを持っていなければなりません。

▼ UUCP ログインの追加方法

ユーザーのシステムへのアクセスをリモートマシンに許可するには、次の手順を行なって /etc/passwd ファイルにエントリを追加する必要があります。

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 /etc/passwd ファイルを編集し、システムにアクセスを許可するマシンを識別するためのエントリを追加します。

通常、UUCP 接続でのシステムへのアクセスを許可するリモートマシンについて、次のようなエントリを /etc/passwd ファイルに入力します。

```
Ugobi:*:5:5:gobi:/var/spool/uucppublic:/usr/lib/uucp/uucico
```

リモートマシンのログイン名は慣例的に、そのマシン名の前に大文字の U を付けたものです。8 文字を超える名前は使用できないので、一部を短縮した名前や省略名を使用しなければならない場合もあります。

例に示したエントリは、Ugobi からのログイン要求に /usr/lib/uucp/uucico が応答することを示しています。ホームディレクトリは /var/spool/uucppublic です。パスワードは /etc/shadow ファイルから取得されます。パスワードとログイン名は、リモートマシンの UUCP 管理者と協議して決める必要があります。リモート側の管理者は、ログイン名と暗号化されていないパスワードを含む正しいエントリを、リモートマシンの Systems ファイルに追加する必要があります。

3 ほかのシステムの UUCP 管理者と、ローカルマシン名を調整します。

同様に、ローカルマシン名とパスワードについて、UUCP を介して通信する相手方のすべてのマシンの UUCP 管理者と協議する必要があります。

UUCP の起動

UUCP には、次に示す 4 つのシェルスクリプトが付属しています。これらのスクリプトは、リモートマシンをポーリングし、転送を再スケジュールし、古いログファイルと成功しなかった転送を整理します。4 つのスクリプトは次のとおりです。

- uudemon.poll
- uudemon.hour
- uudemon.admin
- uudemon.cleanup

UUCP を円滑に運用するには、これらのスクリプトを定期的に実行する必要があります。Solaris の全体インストールを行なった場合は、これらのスクリプトを実行するための crontab ファイルが、インストールプロセスの一環として自動的に /usr/lib/uucp/uudemon.crontab の中に作成されます。全体インストールでない場合は、UUCP パッケージをインストールするときにこのファイルが作成されます。

UUCP シェルスクリプトは手動でも実行できます。次に示すのは、 uudemon.crontab のプロトタイプです。このファイルは、マシンの運用の都合に合わせて適宜変更できます。

```
#
#ident  "@(#)uudemon.crontab    1.5     97/12/09 SMI"
#
# This crontab is provided as a sample. For systems
# running UUCP edit the time schedule to suit, uncomment
# the following lines, and use crontab(1) to activate the
# new schedule.
#
#48 8,12,16 * * * /usr/lib/uucp/uudemon.admin
#20 3 * * * /usr/lib/uucp/uudemon.cleanup
#0 * * * * /usr/lib/uucp/uudemon.poll
#11,41 * * * * /usr/lib/uucp/uudemon.hour
```

注 – デフォルトでは、UUCP の操作は無効にされています。UUCP を有効にするには、タイムスケジュールを編集し、uudemon.crontab ファイルの適切な行のコメントを解除してください。

▼ UUCP の起動方法

uudemon.crontab ファイルは、次の手順に従って起動します。

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 /usr/lib/uucp/uudemon.crontab ファイルを編集し、必要に応じてエントリを変更します。

3 次のコマンドを入力して、uudemon.crontab ファイルを起動します。

```
crontab < /usr/lib/uucp/uudemon.crontab
```

uudemon.poll シェルスクリプト

デフォルトの uudemon.poll シェルスクリプトは 1 時間に 1 回 /etc/uucp/Poll ファイルを読み取ります。Poll ファイル内のマシンのどれかに対するポーリングがスケジュールされると、作業ファイル (C.*sysnxxxx*) が /var/spool/uucp/*nodename* ディレクトリに入れられます。*nodename* は、そのマシンの UUCP ノード名です。

このシェルスクリプトは、1 時間に 1 回ずつ uudemon.hour の前に実行されるようにスケジュールされているので、uudemon.hour が呼び出されたときには、作業ファイルが存在しています。

uudemon.hour シェルスクリプト

デフォルトの uudemon.hour シェルスクリプトは次のことを行います。

- uusched プログラムを呼び出し、スプールディレクトリを検索して未処理の作業ファイル (C.) を見つける。そして、それらの作業ファイルをリモートマシンに転送するためにスケジュールする
- uuxqt デーモンを呼び出し、スプールディレクトリを検索して、ローカルコンピュータに転送済みで、転送時に処理されなかった実行ファイル (X.) を見つける

デフォルトでは、uudemon.hour は 1 時間に 2 回実行されます。リモートマシンに対する呼び出しが頻繁に失敗すると予測される場合は、このスクリプトの実行頻度を増やすこともできます。

uudemon.admin シェルスクリプト

デフォルトの uudemon.admin シェルスクリプトは次のことを行います。

- p オプションと q オプション付きで uustat コマンドを実行する。q は、キューに入っている作業ファイル (C.)、データファイル (D.)、および実行ファイル (X.) の状態を報告する。p は、ロックファイル (/var/spool/locks) 中に列挙されているネットワークプロセス用のプロセス情報を表示する
- 結果の状態情報を mail により uucp 管理ログインに送る

uudemon.cleanup シェルスクリプト

デフォルトの uudemon.cleanup シェルスクリプトは次のことを行います。

- /var/uucp/.Log ディレクトリから個々のマシンに関するログファイルを取り出し、それらをマージし、ほかの古いログ情報とともに /var/uucp/.Old ディレクトリに入れる
- 7 日以上経過している作業ファイル (C.)、7 日以上経過しているデータファイル (D.)、および 2 日以上経過している実行ファイル (X.) を、スプールファイルから削除する
- 配送できなかったメールを送信元に戻す
- その日に収集した状態情報の要約を、メールにより UUCP 管理ログイン (uucp) に送る

TCP/IP を介した UUCP の実行

TCP/IP ネットワーク上で UUCP を実行するには、この節で説明するようにいくつかの変更が必要になります。

▼ TCP/IP 用 UUCP の起動方法

1. スーパーユーザーになるか、同等の役割を引き受けます。

   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2. /etc/uucp/Systems ファイルを編集し、対象エントリが次のフィールドを持っていることを確認します。

   System-Name Time TCP Port networkname Standard-Login-Chat

   典型的なエントリは次のようになります。

   ```
   rochester Any TCP - ur-seneca login: Umachine password: xxx
   ```

   networkname フィールドには、TCP/IP ホスト名を明示的に指定できます。この機能は一部のサイトにとっては重要です。上の例に示したサイトの UUCP ノード名は rochester であり、これは TCP/IP ホスト名 ur-seneca と異なります。さらに、rochester という TCP/IP ホスト名を持ち、UUCP を実行するまったく別のマシンが存在することもあり得ます。

   Systems ファイル内の Port フィールドにはエントリ - を指定するようにしてください。これは、エントリを uucp と指定するのと同じです。ほとんどの場合、*networkname* はシステム名と同じで、Port フィールドは - となります。これ

は、services データベースから標準 uucp ポートを使用することを意味します。in.uucpd デーモンは、認証のためにリモートマシンがログインとパスワードを送ることを想定しているので、getty や login と同様に、ログインとパスワードを要求します。

3 /etc/inet/services ファイルを編集し、次のように **UUCP** 用のポートを設定します。

```
uucp    540/tcp    uucpd        # uucp daemon
```

このエントリを変更する必要はありません。ただし、マシンがネームサービスとして NIS または NIS+ を実行する場合は、/etc/services の /etc/nsswitch.conf エントリを変更して、まず files、次に nis または nisplus が検査されるようにする必要があります。

4 **UUCP** が有効になっているか確認します。

```
# svcs network/uucp
```

UUCP サービスは、サービス管理機能によって管理されます。このサービスの状態は、svcs コマンドを使用して確認できます。サービス管理機能の概要については、『Solaris のシステム管理 (基本編)』の第 18 章「サービスの管理 (概要)」を参照してください。

5 (省略可能) 必要に応じ、次のように入力して **UUCP** を有効にします。

```
# inetadm -e network/uucp
```

UUCP のセキュリティーと保守

UUCP の設定が終われば、その後の保守は簡単です。この節では、セキュリティー、保守、およびトラブルシューティングに関連する UUCP の作業について説明します。

UUCP のセキュリティーの設定

デフォルトの /etc/uucp/Permissions ファイルは、UUCP リンクに関する最大限のセキュリティーを提供します。デフォルトの Permissions ファイルには、エントリは入っていません。

定義する各リモートマシンについて、次に示す追加パラメータを設定できます。

- ローカルマシンからファイルを受け取る方法
- 読み取り権と書き込み権が与えられるディレクトリ
- リモート実行に使用できるコマンド

典型的な Permissions のエントリは次のようになります。

```
MACHINE=datsun LOGNAME=Udatsun VALIDATE=datsun
COMMANDS=rmail REQUEST=yes SENDFILES=yes
```

このエントリでは、システム内の任意の場所ではなく、通常の UUCP ディレクトリとの間でのファイルの送信と受信が可能となります。また、ログイン時に UUCP ユーザー名の認証が行われます。

日常の UUCP の保守

UUCP の保守に必要な作業の量はさほど多くはありません。ただし、How to Start UUCP で述べたように、565 ページの「UUCP の起動方法」 ファイルが正しい場所に置かれているか確認するとともに、メールファイルと公開ディレクトリが次第に大きくなることに注意する必要があります。

UUCP に関連する電子メール

UUCP のプログラムとスクリプトが生成する電子メールメッセージは、すべてユーザー ID uucp に送信されます。管理者がユーザー uucp として頻繁にログインしていないと、メールが蓄積され、ディスク空間を浪費していることに気付かない場合があります。この問題を解決するには、/etc/mail/aliases の中に別名を 1 つ作り、root か自分自身、そしてほかの UUCP 保守責任者に、電子メールを転送します。aliases ファイルを変更したあとで、newaliases コマンドを実行するのを忘れないようにしてください。

UUCP 公開ディレクトリ

ディレクトリ /var/spool/uucppublic は、UUCP がデフォルトでファイルをコピーできる場所として、すべてのシステムに対して提供されているディレクトリです。すべてのユーザーが、/var/spool/uucppublic への移動と、このディレクトリ内のファイルの読み書きを行う権限を持っています。しかし、このディレクトリのスティッキービットが設定されているため、このディレクトリのモードは 01777 です。したがって、ユーザーには、このディレクトリにコピーされ uucp に所有されているファイルを削除することはできません。 このディレクトリからファイルを削除できるのは、root または uucp としてログインした UUCP 管理者だけです。このディレクトリ内に無秩序にファイルが蓄積するのを防ぐために、定期的にファイルを削除する必要があります。

このような保守作業がユーザーにとって不都合な場合は、セキュリティーのために設定されているスティッキービットを削除するのではなく、uuto と uupick を使用するよう各ユーザーに推奨してください。uuto と uupick の使い方については、uuto(1C) のマニュアルページを参照してください。このディレクトリのモードの制限の度合を強めて、特定のユーザーグループに使用を限定することもできます。ユーザーによってディスク空間が使い切ってしまわれないように、そのディスクへの UUCP アクセスを拒否することもできます。

UUCP のトラブルシューティング

ここでは、UUCP に関する一般的な問題を解決するための手順について説明します。

▼ モデムまたは ACU の障害確認方法

モデムや ACU で、適正に動作していないものがないかどうかを、いくつかの方法で検査できます。

1. スーパーユーザーになるか、同等の役割を引き受けます。

   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2. 次のコマンドを実行し、接続障害の回数と理由を表示します。

   ```
   # uustat -q
   ```

3. 特定の回線を介した呼び出しを行い、その試行に関するデバッグ情報を表示します。

   この回線は、`/etc/uucp/Devices` ファイル内で `direct` として定義されていなければなりません。回線が自動ダイアラに接続されている場合は、コマンド行の終わりに電話番号を追加する必要があります。または、デバイスを `direct` として設定する必要があります。次のように入力します。

   ```
   # cu -d -lline
   ```

   line は `/dev/cua/a` です。

▼ 送信に関するデバッグ方法

特定のマシンに接続できない場合は、`Uutry` と `uucp` を使用して、そのマシンに対する通信を検査できます。

1. スーパーユーザーになるか、同等の役割を引き受けます。

   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2. 接続を試行します。

   ```
   # /usr/lib/uucp/Uutry -r machine
   ```

machine には、接続できないマシンのホスト名を指定します。このコマンドは次のことを行います。

- デバッグ機能を指定して転送デーモン (uucico) を起動する。root としてログインしていれば、さらに多くのデバッグ情報が得られる
- デバッグ出力を /tmp/*machine* に送る
- 次のように入力すると、デバッグ出力を端末に表示する

  ```
  # tail -f
  ```

  出力を終了するには Control-C キーを押します。この出力を保存する場合は、/tmp/*machine* から出力内容をコピーします。

3 Uutry を使用しても問題の原因がわからない場合は、ジョブをキューに入れてみます。

 # uucp -r *file machine*\!*/dir/file*

file 転送するファイルの名前を指定する

machine コピー先のマシンの名前を指定する

/dir/file 相手のマシンのどこにファイルを転送するかを指定する

4 次のコマンドを入力します。

```
# Uutry
```

それでも問題が解決できないときは、ご購入先へお問い合わせください。デバッグ出力は問題の診断に役立つため、保存しておいてください。

注 – Uutry で -x *n* オプションを使用して、デバッグのレベルを増減することもできます。*n* はデバッグレベルを指定します。Uutry のデフォルトのデバッグレベルは 5 です。

デバッグレベル 3 では、接続がいつどのように確立されたかについての基本的な情報は提供されますが、転送について提供される情報は多くはありません。一方、デバッグレベル 9 では、転送処理に関するすべての情報が網羅されます。デバッグは転送の両端で行われるという点に注意してください。比較的大きいテキストについて 5 より高いレベルのデバッグを行いたい場合は、相手サイトの管理者に連絡して、いつレベルを変更するか決定してください。

UUCP /etc/uucp/Systems ファイルの検査

特定のマシンと接続しようとすると障害が発生する場合は、Systems ファイルの中の情報が最新のものであることを確認してください。マシンに関する次の情報が、最新でない可能性があります。

- 電話番号
- ログイン ID
- パスワード

UUCP エラーメッセージの検査

UUCP のエラーメッセージには、 ASSERT と STATUS の 2 つの種類があります。

- プロセスが異常終了した場合は、ASSERT エラーメッセージが /var/uucp/.Admin/errors に記録されます。この種類のメッセージには、ファイル名、sccsid、回線番号、およびテキストが含まれています。この種類のメッセージが送られるのは、通常、システムに問題がある場合です。
- STATUS エラーメッセージは /var/uucp/.Status ディレクトリに格納されます。このディレクトリ内には、ローカルコンピュータが通信しようとした各リモートマシンについて、それぞれファイルが作られます。これらのファイルには、試行した通信と、その通信が成功したかどうかについての状態情報が入っています。

基本情報の検査

次のコマンドを使用して、基本的なネットワーク情報を検査できます。

- uuname コマンドは、ローカルマシンが接続できるマシンのリストを表示する場合に使用します。
- uulog コマンドは、特定のホストのためのログディレクトリの内容を表示するために使用します。
- uucheck -v コマンドは、uucp が必要とするファイルとディレクトリが存在しているかどうかを検査するために使用します。また、Permissions ファイルも検査して、設定してあるアクセス権に関する情報を表示します。

◆ ◆ ◆ 第 26 章

UUCP (リファレンス)

この章では、UUCP を使用する場合のリファレンス情報について説明します。次の項目について説明します。

UUCP /etc/uucp/Systems ファイル

/etc/uucp/Systems ファイルには、uucico デーモンがリモートコンピュータとの通信リンクを確立するために必要な情報が入っています。/etc/uucp/Systems は、UUCP を構成するときに編集しなければならない最初のファイルです。

Systems ファイルの中の各エントリは、このホストが通信するリモートコンピュータを表します。1 つのホストについて複数のエントリがある場合もあります。付加的なエントリは、順番に試される代替通信パスを表します。さらに、UUCP のデフォルト状態では、/etc/uucp/Systems ファイルに含まれていないコンピュータがこのホストにログインできないようになっています。

Sysfiles ファイルを使用して、Systems ファイルとして使用されるファイルをいくつか定義できます。Sysfiles の詳細は、593 ページの「UUCP /etc/uucp/Sysfiles ファイル」を参照してください。

Systems ファイルのエントリの形式は次のとおりです。

```
System-Name    Time    Type    Speed    Phone    Chat Script
```

次に、Systems ファイルのエントリ例を示します。

例 26–1 /etc/uucp/Systems のエントリ

```
Arabian     Any  ACUEC 38400 111222  ogin: Puucp ssword:beledi
```

Arabian	System-Name フィールドのエントリ。詳細は、574 ページの「/etc/uucp/Systems ファイルの System-Name フィールド」を参照
Any	Time フィールドのエントリ。詳細は、575 ページの「/etc/uucp/Systems ファイルの Time フィールド」を参照
ACUEC	Type フィールドのエントリ。詳細は、576 ページの「/etc/uucp/Systems ファイルの Type フィールド」を参照
38400	Speed フィールドのエントリ。詳細は、576 ページの「/etc/uucp/Systems ファイルの Speed フィールド」を参照
111222	Phone フィールドのエントリ。詳細は、577 ページの「/etc/uucp/Systems ファイルの Phone フィールド」を参照
ogin: Puucp ssword:beledi	Chat Script フィールドのエントリ。詳細は、577 ページの「/etc/uucp/Systems ファイルの Chat-Script フィールド」を参照

/etc/uucp/Systems ファイルの System-Name フィールド

このフィールドには、リモートコンピュータのノード名が入ります。TCP/IP ネットワークでは、この名前は、マシンのホスト名でも、/etc/uucp/Sysname ファイルによって UUCP 通信用として特別に作成した名前でもかまいません。573 ページの「UUCP /etc/uucp/Systems ファイル」を参照してください。例 26–1 では、System-Name フィールドにはリモートホスト Arabian に関するエントリが含まれています。

/etc/uucp/Systems ファイルの Time フィールド

このフィールドには、リモートコンピュータを呼び出すことのできる曜日と時刻を指定します。Time フィールドの形式は次のとおりです。

```
daytime[;retry]
```

Time フィールドの *day* 部

day の部分には、次のエントリのいくつかを含むリストを指定できます。

`Su Mo Tu We Th Fr Sa`	個々の曜日
`Wk`	任意の平日
`Any`	任意の日
`Never`	このホストはこのリモートコンピュータの呼び出しをいっさい行わない。呼び出しはリモートコンピュータ側から行う必要がある。それを受けて、このホストは「受動モード」で稼動する

Time フィールドの *time* 部

例 26–1 では、Time フィールドに `Any` が示されています。これは、ホスト `Arabian` をいつでも呼び出せるということです。

time の部分には、24 時間表記で表した時間の範囲を指定します。たとえば、午前 8 時 00 分から午後 12 時 30 分までなら `0800-1230` とします。*time* の部分を指定しなかった場合は、どのような時刻にでも呼び出しができるものとみなされます。

0000 の前後にまたがる時間範囲も指定できます。たとえば、`0800-0600` は、午前 6 時から午前 8 時までの間を除くすべての時間帯で呼び出し可能であることを示します。

Time フィールドの *retry* 部

retry サブフィールドには、試行が失敗してから次の再試行までの間に最小限必要な時間 (分単位) を指定できます。デフォルトの待ち時間は 60 分です。サブフィールド区切り文字はセミコロン (;) です。たとえば、`Any;9` は、呼び出しはいつでもできるが、失敗したときは次の再試行までに少なくとも 9 分は待たなければならないことを意味します。

retry エントリを指定しなかった場合は、待ち時間倍加アルゴリズムが使用されます。これは、UUCP がデフォルトの待ち時間から始めて、失敗した試行の回数が増えるほど待ち時間を長くしていくことを意味します。たとえば、最初の再試行待ち

時間が 5 分であるとします。応答がない場合は、次の再試行は 10 分後となります。次の再試行は 20 分後というようになり、最大再試行時間の 23 時間に達するまで増加します。*retry* を指定した場合は、常にその値が再試行待ち時間となります。指定がなければ待ち時間倍加アルゴリズムが使用されます。

/etc/uucp/Systems ファイルの Type フィールド

このフィールドには、リモートコンピュータとの通信リンクを確立するために使用するデバイスタイプを指定します。このフィールドで使用するキーワードは、Devices ファイル中のエントリの最初のフィールドと突き合わされます。

例 26–2 Type フィールドのキーワード

```
Arabian     Any     ACUEC, g     38400     1112222     ogin: Puucp ssword:beledi
```

Type フィールドでは、さらに、システムとの接続に使用するプロトコルを定義できます。上記の例では、デバイスタイプ ACUEC に g プロトコルを組み合わせる方法を示しています。プロトコルの詳細は、587 ページの「/etc/uucp/Devices ファイル内のプロトコル定義」を参照してください。

/etc/uucp/Systems ファイルの Speed フィールド

このフィールド (Class フィールドとも呼ばれます) は、通信リンクの確立に使用するデバイスの転送速度を指定します。UUCP speed フィールドには、ダイアラのクラスを区別するために、1 個の英字と速度を含めることができます (たとえば C1200、D1200)。583 ページの「/etc/uucp/Devices ファイルの Class フィールド」を参照してください。

デバイスにはどのような速度でも使用できるものがあり、その場合はキーワード Any を使用できます。このフィールドは、Devices ファイルの対応するエントリの Class フィールドに一致していなければなりません。

例 26–3 Speed フィールドのエントリ

```
eagle     Any     ACU, g     D1200     NY3251     ogin: nuucp ssword:Oakgrass
```

このフィールドに情報を入れる必要がない場合は、フィールドの数を合わせるためにダッシュ (-) を指定してください。

/etc/uucp/Systems ファイルの Phone フィールド

このフィールドには、自動ダイアラ (ポートセレクタ) に与えるリモートコンピュータの電話番号 (トークン) を指定できます。電話番号は、オプションの英字による省略名と数字部分で構成されます。省略名を使用する場合は、Dialcodes ファイル内に列挙されているものの 1 つでなければなりません。

例 26–4 Phone フィールドのエントリ

```
nubian     Any     ACU       2400    NY555-1212     ogin: Puucp ssword:Passuan
eagle      Any     ACU, g    D1200   NY=3251        ogin: nuucp ssword:Oakgrass
```

Phone フィールドでは、等号 (=) は二次発信音を待ってから残りの数字をダイアルするという ACU への指示となります。文字列の中にダッシュ (-) があれば、4 秒間待ってから次の数字をダイアルするという指示になります。

コンピュータがポートセレクタに接続されている場合は、そのセレクタに接続しているほかのコンピュータにアクセスできます。この種のリモートマシン用の Systems ファイルエントリの Phone フィールドには、電話番号を入れません。代わりに、このフィールドにはスイッチに渡すトークンを指定します。このようにすれば、このホストがどのリモートマシンとの通信を望んでいるかを、ポートセレクタが判断できます。この場合は、システム名だけを指定するのが普通です。対応する Devices ファイルエントリでは、エントリの末尾に \D を指定して、このフィールドが Dialcode ファイルを使用して解釈されないようにしなければなりません。

/etc/uucp/Systems ファイルの Chat-Script フィールド

このフィールド (Login フィールドとも呼ばれる) には、「chat スクリプト」と呼ばれる文字列が入ります。chat スクリプトには、ローカルマシンとリモートマシンが対話の最初の時点で互いに受け渡ししなければならない文字が含まれています。chat スクリプトの形式は次のとおりです。

expect send [expect send] ....

expect は、対話を開始するために、ローカルホストがリモートホストから受信することを想定している文字列です。*send* は、ローカルホストが、リモートホストからの *expect* 文字列を受信したあとで送信する文字列です。chat スクリプトには、複数の expect-send シーケンスを含めることもできます。

基本的な chat スクリプトには次の情報が含まれます。

- ローカルホストがリモートマシンから受信することを想定しているログインプロンプト
- ログインするためにローカルホストがリモートマシンに送るログイン名
- ローカルホストがリモートマシンから受信することを想定しているパスワードプロンプト
- ローカルホストがリモートマシンに送るパスワード

expect フィールドは、次の形式のサブフィールドを持つことができます。

expect[-send-expect]...

-send は、その前の *expect* が正常に読み取れなかった場合に送られるものであり、*-send* のあとの *-expect* は、その次に送られてくると想定されている文字列です。

たとえば、`login--login` という文字列を指定した場合、ローカルホストの UUCP は `login` が送られてくることを想定します。リモートマシンから `login` を受信すると、UUCP は次のフィールドに進みます。`login` を受信しなかった場合は、UUCP はキャリッジリターンを送信し、再度 `login` が送られてくるのを待ちます。ローカルコンピュータが、初期状態でどのような文字も想定していない場合は、*expect* フィールドで文字列 "" (NULL 文字列) を指定します。*send* 文字列が *\c* で終わっている場合を除き、`send` フィールドの送信のあとには必ずキャリッジリターンが伴うという点に注意してください。

次に示すのは、*expect-send* 文字列を使用する `Systems` ファイルエントリの例です。

```
sonora Any ACUEC 9600 2223333 "" \r \r ogin:-BREAK-ogin: Puucpx ssword:xyzzy
```

この例は、ローカルホストの UUCP に、2 個のキャリッジリターンを送ってから `ogin:` (`Login:` という場合もあるため) を待つように指示しています。`ogin:` を受信しなかった場合は、 `BREAK` を送ります。`ogin:` を受信した場合は、ログイン名 `Puucpx` を送ります。`ssword:` (`Password:` を表す) を受け取ったら、パスワード `xyzzy` を送ります。

次の表に、便利なエスケープ文字をいくつか紹介します。

表 26–1 `Systems` ファイルの chat スクリプトで使用されるエスケープ文字

エスケープ文字	意味
`\b`	バックスペース文字を送信または想定します。
`\c`	文字列の末尾で使用すると、普通なら送信されるキャリッジリターンが抑止されます。その他の場合は無視されます。

表 26–1 Systems ファイルの chat スクリプトで使用されるエスケープ文字 *(続き)*

エスケープ文字	意味
\d	後続の文字を送る前に 1 3 秒の遅延が生じます。
\E	エコーチェックを開始します。これ以降は、1 文字送信するたびに、UUCP はその文字が受信されるまで待ち、その後、チェックを続行します。
\e	エコーチェックをオフにします。
\H	ハングアップを 1 回無視します。このオプションはコールバックモデム用に使用します。
\K	BREAK 文字を送信します。
\M	CLOCAL フラグをオンにします。
\m	CLOCAL フラグをオフにします。
\n	改行文字を送信または想定します。
\N	NULL 文字 (ASCII NUL) を送信します。
\p	約 1/4 秒間または 1/2 秒間、一時停止します。
\r	キャリッジリターンを送信または想定します。
\s	スペース文字を送信または想定します。
\t	タブ文字を送信または想定します。
EOT	EOT とそれに続く 2 個の改行文字を送信します。
BREAK	BREAK 文字を送信します。
ddd	8 進数 (*ddd*) で表される文字を送信または想定します。

Chat スクリプトを使用したダイアルバックの有効化

組織によっては、リモートコンピュータからの呼び出しを処理するダイアルインサーバーを設定する場合があります。たとえば、コールバックモデムを持つダイアルインサーバーを配備し、社員が自宅のコンピュータから呼び出せるようにすることができます。ダイアルインサーバーは、リモートマシンを識別すると、そのリモートマシンとのリンクを切断し、逆にそのリモートマシンを呼び出して、通信リンクが再確立されます。

Systems ファイルの chat スクリプトで、コールバックが必要な箇所で \H オプションを使用することにより、コールバックの操作を簡素化することができます。ダイアルインサーバーのハングアップが予想される箇所で、expect 文字列の一部として \H を使用します。

たとえば、ダイアルインサーバーを呼び出す chat スクリプトに、次のような文字列が含まれているとします。

```
INITIATED\Hogin:
```

ローカルホストの UUCP ダイアル機能は、ダイアルインサーバーから INITIATED という文字列を受け取ることを想定しています。文字列 INITIATED を受け取ると、ダイアル機能は、ダイアルインサーバーがハングアップするまで、その後受信するすべての文字をフラッシュします。またダイアル機能は、expect 文字列のその次の部分、つまり ogin: という文字列がダイアルインサーバーから送られてくるのを待ちます。ogin: を受け取ると、ダイアル機能は chat スクリプトを先へ進めます。

上記のサンプルでは \H の前後に文字列が指定されていますが、これらはなくてもかまいません。

/etc/uucp/Systems ファイルでのハードウェアフロー制御

擬似送信文字列 STTY= *value* を用いることによっても、モデムの特性を設定できます。たとえば、STTY=crtscts を使用すると、ハードウェアフロー制御が可能になります。STTY はすべての stty モードを受け入れます。詳細は、stty(1) と termio(7I) のマニュアルページを参照してください。

次の例は、Systems ファイルのエントリ内でハードウェアフロー制御を指定しています。

```
unix Any ACU 2400 12015551212 "" \r ogin: Puucp ssword:Passuan "" \ STTY=crtscts
```

擬似送信文字列は、Dialers ファイルのエントリの中でも使用できます。

/etc/uucp/Systems ファイルでのパリティーの設定

場合によっては、呼び出そうとしているシステムがポートのパリティーを検査し、パリティーに誤りがあると回線を切断することがあります。そのため、パリティーのリセットが必要になります。expect-send (予期 - 送信) の文字列ペアとして "" P_ZERO を使用すると、上位ビット (パリティービット) が 0 に設定されます。この expect-send ペアの例を次に示します。

```
unix Any ACU 2400 12015551212 "" P_ZERO "" \r ogin: Puucp ssword:Passuan
```

次に、expect-send 文字列ペア "" P_ZERO のあとに続けることができるパリティー文字列ペアを示します。

"" P_EVEN　パリティーを偶数 (デフォルト) に設定する

"" P_ODD　パリティーを基数に設定する

"" P_ONE　パリティービットを 1 に設定する

これらのパリティー設定は、chat スクリプトのどこにでも挿入できます。この設定は、chat スクリプト内の "" P_ZERO (expect-send 文字列ペア) よりあとにあるすべての情報に適用されます。パリティー文字列ペアは、Dialers ファイルのエントリの中でも使用できます。次の例には、パリティー文字列ペア "" P_ONE が含まれています。

```
unix Any ACU 2400 12015551212 "" P_ZERO "" P_ONE "" \r ogin: Puucp ssword:Passuan
```

UUCP /etc/uucp/Devices ファイル

/etc/uucp/Devices ファイルには、リモートコンピュータへのリンクを確立するために使用できるすべてのデバイスに関する情報が入っています。この種のデバイスには、ACU (高速モデムを含む)、直接リンク、ネットワーク接続などがあります。

/etc/uucp/Devices ファイルのエントリは、次の構文を使用します。

```
Type   Line   Line2   Class   Dialer-Token-Pairs
```

次に示す Devices ファイルエントリは、ポート A に接続され、38,400 bps で動作する U.S. Robotics V.32bis モデムを表しています。

```
ACUEC   cua/a   -   38400   usrv32bis-ec
```

ACUEC　Type フィールド内のエントリ。詳細は、582 ページの「/etc/uucp/Devices ファイルの Type フィールド」を参照

cua/a　Line フィールド内のエントリ。詳細は、583 ページの「/etc/uucp/Devices ファイルの Line フィールド」を参照

-　Line2 フィールド内のエントリ。詳細は、583 ページの「/etc/uucp/Devices ファイルの Line2 フィールド」を参照

38400　Class フィールド内のエントリ。詳細は、583 ページの「/etc/uucp/Devices ファイルの Class フィールド」を参照

usrv32bis-ec　Dialer-Token-Pairs フィールド内のエントリ。詳細は、584 ページの「/etc/uucp/Devices ファイルの Dialer-Token-Pairs フィールド」を参照

各フィールドについては、次の節で説明しています。

/etc/uucp/Devices ファイルの Type フィールド

このフィールドで、デバイスによって確立されるリンクの種類を説明します。このフィールドには次のセクションに示すキーワードのいずれかを入れることができます。

キーワード Direct

キーワード Direct は、主として cu 接続用のエントリ内で使用されます。このキーワードは、このリンクがほかのコンピュータまたはポートセレクタへの直接リンクであることを示します。cu の -l オプションで参照する各回線について、それぞれ独立したエントリを作成する必要があります。

キーワード ACU

キーワード ACU は、(cu、UUCP、asppp、または Solaris PPP 4.0 を介した) リモートコンピュータへのリンクを、モデムを介して確立することを示します。このモデムは、直接ローカルコンピュータに接続しているものでも、ポートセレクタを介して間接的に接続しているものでもかまいません。

ポートセレクタ

ポートセレクタは、ポートセレクタの名前で置き換えるものとして、Type フィールド内で使用される変数です。ポートセレクタは、ネットワークに接続されたデバイスで、呼び出し側モデムの名前を要求し、アクセスを許可します。/etc/uucp/Dialers ファイルに入っている呼び出しスクリプトは、micom ポートセレクタと develcon ポートセレクタについてのものだけです。ユーザーは、Dialers ファイルに独自のポートセレクタエントリを追加できます。詳細は、588 ページの「UUCP /etc/uucp/Dialers ファイル」を参照してください。

System-Name 変数

Type フィールド内のこの変数は、特定のマシンの名前で置き換えられます。これは、リンクがこのマシンへの直接リンクであることを示します。この命名スキーマは、この Devices エントリ内の行と、コンピュータ *System-Name* についての /etc/uucp/Systems ファイルエントリを対応付けるために使用されます。

Devices ファイルおよび Systems ファイルの Type フィールド

例 26–5 は、/etc/uucp/Devices のフィールドと /etc/uucp/Systems のフィールドの比較を示しています。フィールドの書体を変えて示したように、Devices ファイルの Type フィールドで使用されているキーワードは、Systems ファイルエントリの 3 番目のフィールドと突き合わされます。Devices ファイルの Type フィールドには ACUEC というエントリが入っており、これは自動呼び出し装置、つまりこの例では V.32bis

モデムを示しています。この値は、Systems ファイルの Type フィールドと突き合わされます。このフィールドにも ACUEC というエントリが入っています。詳細は、573 ページの「UUCP /etc/uucp/Systems ファイル」 を参照してください。

例 26–5 Devices ファイルと Systems ファイルの Type フィールドの比較

次に、Devices ファイルのエントリ例を示します。

```
ACUEC cua/a - 38400 usrv32bis-ec
```

次に、Systems ファイルのエントリ例を示します。

```
Arabian Any ACUEC 38400 111222 ogin: Puucp ssword:beledi
```

/etc/uucp/Devices ファイルの Line フィールド

このフィールドには、Devices エントリに対応付けられる回線 (ポート) のデバイス名が入ります。たとえば、特定のエントリに対応付けられているモデムが /dev/cua/a (シリアルポート A) に接続されている場合、このフィールドに入力する名前は cua/a です。Line フィールドでオプションのモデム制御フラグ M を使用すると、キャリアを待たないでデバイスをオープンすることを指定できます。次に例を示します。

```
cua/a,M
```

/etc/uucp/Devices ファイルの Line2 フィールド

このフィールドは、フィールドの数を合わせるために存在しているだけです。ここには常にハイフン (-) を指定します。Line2 フィールドを使用するのは 801 型のダイアラですが、この種類は Solaris OS ではサポートされていません。801 型以外のダイアラは通常はこの設定を使用しませんが、このフィールドにダッシュだけは入れておく必要があります。

/etc/uucp/Devices ファイルの Class フィールド

Type フィールドでキーワード ACU または Direct を使用した場合は、Class フィールドにはデバイスの速度が入ります。ただし、このフィールドには、ダイアラのクラス (Centrex や Dimension PBX など) を区別するために、1 個の英字と速度値を含めることができます (C1200、D1200 など)。

大規模な事業所では複数種の電話ネットワークを使用することが多いため、このような指定が必要になります。たとえば、1 つのネットワークは事業所内の内線通信専

用に使用し、もう 1 つのネットワークは外線通信に使用するといった方式が考えられます。このような場合は、内線回線と外線回線とを区別する必要があります。

Devices ファイルの Class フィールドで使用するキーワードは、Systems ファイルの Speed フィールドと突き合わされます。

例 26–6 Devices ファイルの Class フィールド

```
ACU   cua/a   -   D2400  hayes
```

どのような速度でも使用できるデバイスでは、Class フィールドにキーワード Any を使用します。Any を使用した場合は、回線は、Systems ファイルの Speed フィールドで要求された任意の速度に適合します。このフィールドが Any で、Systems ファイルの Speed フィールドも Any である場合は、速度はデフォルトの 2400 bps となります。

/etc/uucp/Devices ファイルの Dialer-Token-Pairs フィールド

Dialer-Token-Pairs (DTP) フィールドには、ダイアラの名前とそれに渡すトークンが入ります。DTP フィールドの構文は次のとおりです。

dialer token [dialer token]

dialer の部分は、モデムかポートモニターの名前あるいは直接リンクデバイスの場合は direct または uudirect です。ダイアラとトークンのペアはいくつでも指定できます。*dialer* の部分がない場合は、Systems ファイル内の関連エントリから取得されます。*token* 部は、dialer 部の直後に指定できます。

対応するダイアラによっては、最後のダイアラとトークンのペアはない場合もあります。ほとんどの場合は、最後のペアには *dialer* 部だけが含まれます。*token* 部は、対応する Systems ファイルエントリの Phone フィールドから取得されます。

dialer 部の有効エントリは、Dialers ファイル内で定義されているものか、いくつかの特殊ダイアラタイプのうちの 1 つとなります。これらの特殊ダイアラタイプはコンパイル時にソフトウェア中に組み込まれているので、Dialers ファイル内に該当エントリがなくても使用できます。次に、特殊なダイアラタイプを示します。

TCP	TCP/IP ネットワーク
TLI	トランスポートレベルインタフェースネットワーク (STREAMS を使用しないもの)
TLIS	トランスポートレベルインタフェースネットワーク (STREAMS を使用するもの)

詳細は、587 ページの「/etc/uucp/Devices ファイル内のプロトコル定義」を参照してください。

/etc/uucp/Devices ファイルの Dialer-Token-Pairs フィールドの構造

DTP フィールドの構造は、エントリに対応するデバイスに応じて 4 通りに設定できます。

次に 1 つ目の方法を示します。

直接接続モデム – コンピュータのポートにモデムが直接接続されている場合は、対応する Devices ファイルエントリの DTP フィールドに入るペアは 1 つだけです。このペアは、通常はモデムの名前です。この名前は、Devices ファイルの特定のエントリと、Dialers ファイル内のエントリとを対応付けるために使用されます。したがって、Dialer フィールドは、Dialers ファイルエントリの最初のフィールドに一致している必要があります。

例 26–7 直接接続モデム用 Dialers フィールド

```
Dialers   hayes =,-,  ""          \\dA\pTE1V1X1Q0S2=255S12=255\r\c
                                  \EATDT\T\r\c CONNECT
```

Devices ファイルエントリの DTP フィールドには、dialer 部 (hayes) だけが示されている点に注意してください。これは、ダイアラに渡す *token* (この例では電話番号) が、Systems ファイルエントリの Phone フィールドから取得されることを意味します (例 26–9 で説明するように、\T が暗黙で指定されます)。

次に、DTP フィールドの構造化に利用できる 2 つ目と 3 つ目の方法を示します。

- 直接リンク – 特定のコンピュータへの直接リンクの場合は、対応するエントリの DTP フィールドには、キーワード direct が入ります。これは、Direct、*System-Name* の両方の直接リンクエントリにもあてはまります。582 ページの「/etc/uucp/Devices ファイルの Type フィールド」を参照してください。
- 同じポートセレクタ上のコンピュータ – 通信するコンピュータが、ローカルコンピュータと同じポートセレクタスイッチ上にある場合は、ローカルコンピュータはまずそのスイッチにアクセスする必要があります。そのスイッチが、相手のコンピュータとの接続を確立します。この種のエントリでは、ペアは 1 つだけです。*dialer* 部が Dialers ファイルのエントリと突き合わされます。

例 26–8 同一ポートセレクタ上のコンピュータ用 UUCP Dialer フィールド

```
Dialers    develcon ,""   ""            \pr\ps\c est:\007 \E\D\e \007
```

token 部が空である点に注意してください。このように指定されている場合は、この部分が Systems ファイルから取得されることを示しています。このコンピュータ用の Systems ファイルエントリには、Phone フィールドにトークンが含まれています。このフィールドは、通常、コンピュータの電話番号用として確保されています。詳細は、573 ページの「UUCP /etc/uucp/Systems ファイル」を参照してください。この種類の DTP にはエスケープ文字 (\D) が含まれています。これは、Phone フィールドの内容が、Dialcodes ファイル内の有効エントリとして解釈されないことを保証します。

次に、DTP フィールドの構造化に利用できる 4 つ目の方法を示します。

ポートセレクタに接続しているモデム – ポートセレクタに高速モデムが接続されている場合は、ローカルコンピュータはまずポートセレクタスイッチにアクセスする必要があります。そして、そのスイッチがモデムとの接続を確立します。この種類のエントリには、ダイアラとトークンのペアが 2 つ必要です。各ペアの *dialer* 部 (エントリの 5 番目と 7 番目のフィールド) が、Dialers ファイル内のエントリと突き合わされます。

例 **26–9** ポートセレクタに接続されたモデム用 UUCP Dialer フィールド

```
develcon ""     ""    \pr\ps\c  est:\007    \E\D\e      \007
ventel   =&-%   t""   \r\p\r\c  $           <K\T%\r>\c  ONLINE!
```

最初のペアでは、develcon がダイアラで、vent が Develcon スイッチに渡されるトークンです。トークンは、コンピュータに接続するデバイス (たとえば Ventel モデム) をダイアラに指示しています。各スイッチごとに設定が異なることがあるので、このトークンは各ポートセレクタに固有のものにします。Ventel モデムが接続されたあと、第 2 のペアがアクセスされます。このペアでは、Ventel がダイアラで、トークンは Systems ファイルから取得されます。

DTP フィールドで使用できるエスケープ文字が 2 つあります。

- \T – Phone (*token*) フィールドを、/etc/uucp/Dialcodes ファイルを使用して解釈することを指定します。通常、モデム (Hayes、US Robotics など) に対応する各呼び出しスクリプトについて、/etc/uucp/Dialers ファイルにこのエスケープ文字を組み込みます。したがって、呼び出しスクリプトがアクセスされるまでは、解釈は行われません。
- \D – Phone (*token*) フィールドを、/etc/uucp/Dialcodes ファイルを使用して解釈しないことを指定します。Devices エントリの末尾にエスケープ文字が何も指定されていないときは、デフォルトで \D があるものと想定します。\D は、/etc/uucp/Dialers ファイルの中でも、ネットワークスイッチ develcon と micom に関連したエントリで使用されます。

/etc/uucp/Devices ファイル内のプロトコル定義

/etc/uucp/Devices では、各デバイスに使用するプロトコルを定義できます。通常は、デフォルトを使用するか、または呼び出そうとしている特定のシステムに対してプロトコルを定義できるので、この指定は不要です。詳細は、573 ページの「UUCP /etc/uucp/Systems ファイル」を参照してください。プロトコルを指定する場合は、次の形式を使用する必要があります。

Type,*Protocol [parameters]*

たとえば、TCP/IP プロトコルを指定するには、TCP,te と入力します。

次の表に、Devices ファイルで使用できるプロトコルを示します。

表 26–2 /etc/uucp/Devices で使用されるプロトコル

プロトコル	説明
t	このプロトコルは、TCP/IP や、その他の信頼性のある接続を介した伝送に、最もよく使用される。t はエラーのない伝送を前提としている
g	UUCP のネイティブプロトコル。g は低速で信頼性があり、ノイズの多い電話回線を介した伝送に適している
e	このプロトコルは、(TCP/IP のようなバイトストリーム指向ではなく) メッセージ指向でエラーのないチャネルを介した伝送を前提としている
f	このプロトコルは X.25 接続を介した伝送に使用される。f は、データストリームのフロー制御に関係している。特に X.25/PAD リンクなどのように、完全に (またはほとんど) エラーがないことが保証されるリンクでの使用を意図している。検査合計はファイル全体についてのみ実施される。伝送が失敗した場合は、受信側は再伝送を要求できる

次に、デバイスエントリ用のプロトコル指定の例を示します。

```
TCP,te - - Any TCP -
```

この例は、デバイス TCP について t プロトコルの使用を試みるように指示しています。相手側がそれを拒否した場合は、e プロトコルが使用されます。

e と t のどちらも、モデムを介した通信には適していません。モデムがエラーのない伝送を保証するものであったとしても、モデムと CPU との間でデータが失われる可能性があります。

UUCP /etc/uucp/Dialers ファイル

/etc/uucp/Dialers ファイルには、よく使用される多くのモデムに関するダイアリング指示が入っています。標準外のモデムの使用や、UUCP 環境のカスタマイズを予定している場合以外は、通常このファイルのエントリの変更や追加は必要ありません。しかし、このファイルの内容と、Systems ファイルや Devices ファイルとの関係は理解しておく必要があります。

このファイルの中のテキストは、回線をデータ転送に使用できるようにするために、最初に行わなければならない対話を指定します。chat スクリプトと呼ばれるこの対話は、通常は送受信される一連の ASCII 文字列で、電話番号をダイアルするためによく使用されます。

581 ページの「UUCP /etc/uucp/Devices ファイル」の例に示したように、Devices ファイルエントリの 5 番目のフィールドは Dialers ファイルへのインデックスか、または特殊ダイアラタイプ (TCP、TLI、TLIS など) です。uucico デーモンは、Devices ファイルの 5 番目のフィールドを、Dialers ファイルの各エントリの最初のフィールドと突き合わせます。さらに、Devices の 7 番目の位置から始まる奇数番号の各フィールドは、Dialers ファイルへのインデックスとして使用されます。これらが一致すると、その Dialers のエントリがダイアラ対話を行うために解釈されます。

Dialers ファイルの各エントリの構文は次のとおりです。

```
dialer   substitutions   expect-send
```

次に、US Robotics V.32bis モデム用のエントリの例を示します。

例 26–10 /etc/uucp/Dialers ファイルのエントリ

```
usrv32bis-e    =,-,  ""    dA\pT&FE1V1X1Q0S2=255S12=255&A1&H1&M5&B2&W\r\c OK\r
                           \EATDT\T\r\c CONNECT\s14400/ARQ STTY=crtscts
```

usrv32bis-e
: Dialer フィールドのエントリです。Dialer フィールドは、Devices ファイルの中の 5 番目以降の奇数番号のフィールドと突き合わされます。

=,-, ""
: Substitutions フィールドのエントリです。Substitutions フィールドは変換文字列です。各文字ペアの最初の文字が 2 番目の文字に変換されます。このマッピングは通常、 = と - を、「発信音待ち」と「一時停止」用としてダイアラが必要とする文字に変換するために使用されます。

dA\pT&FE1V1X1Q0S2=255S12=255&A1&H1&M5&B2&W\r\c OK\r
: Expect-Send フィールドのエントリです。Expect-Send フィールドは文字列です。

\EATDT\T\r\c CONNECT\s14400/ARQ STTY=crtscts
: Expect-Send フィールドのエントリの続きです。

次に、Dialers ファイルのエントリの例をいくつか示します。これは、Solaris インストールプログラムの一環として UUCP をインストールするときに提供されるファイルです。

例 26–11 /etc/uucp/Dialers の抜粋

```
penril    =W-P "" \d > Q\c : \d- > s\p9\c )-W\p\r\ds\p9\c-) y\c : \E\TP > 9\c OK

ventel    =&-%    "" \r\p\r\c $ <K\T%%\r>\c ONLINE!

vadic    =K-K    "" \005\p *-\005\p-*\005\p-* D\p BER? \E\T\e \r\c LINE

develcon    ""    "" \pr\ps\c est:\007

\E\D\e \n\007 micom    ""    "" \s\c NAME? \D\r\c GO

hayes    =,-,    "" \dA\pTE1V1X1Q0S2=255S12=255\r\c OK\r \EATDT\T\r\c CONNECT

#   Telebit TrailBlazer
tb1200    =W-,    "" \dA\pA\pA\pTE1V1X1Q0S2=255S12=255S50=2\r\c OK\r
\EATDT\T\r\c CONNECT\s1200
tb2400    =W-,    "" \dA\pA\pA\pTE1V1X1Q0S2=255S12=255S50=3\r\c OK\r
\EATDT\T\r\c CONNECT\s2400
tbfast    =W-,    "" \dA\pA\pA\pTE1V1X1Q0S2=255S12=255S50=255\r\c OK\r
\EATDT\T\r\c CONNECT\sFAST

# USrobotics, Codes, and DSI modems

dsi-ec  =,-,    "" \dA\pTE1V1X5Q0S2=255S12=255*E1*F3*M1*S1\r\c OK\r \EATDT\T\r\c
CONNECT\sEC STTY=crtscts,crtsxoff

dsi-nec =,-,    "" \dA\pTE1V1X5Q0S2=255S12=255*E0*F3*M1*S1\r\c OK\r \EATDT\T\r\c CONNECT
STTY=crtscts,crtsxoff

usrv32bis-ec =,-,  "" \dA\pT&FE1V1X1Q0S2=255S12=255&A1&H1&M5&B2&W\r\c OK\r \EATDT\T\r\c
CONNECT\s14400/ARQ STTY=crtscts,crtsxoff

usrv32-nec =,-, "" \dA\pT&FE1V1X1Q0S2=255S12=255&A0&H1&M0&B0&W\r\c OK\r \EATDT\T\r\c
CONNECT STTY=crtscts,crtsxoff

codex-fast =,-, "" \dA\pT&C1&D2*MF0*AA1&R1&S1*DE15*FL3S2=255S7=40S10=40*TT5&W\r\c OK\r
\EATDT\T\r\c CONNECT\s38400 STTY=crtscts,crtsxoff

tb9600-ec =W-,  "" \dA\pA\pA\pTE1V1X1Q0S2=255S12=255S50=6\r\c OK\r
\EATDT\T\r\cCONNECT\s9600 STTY=crtscts,crtsxoff

tb9600-nec =W-, "" \dA\pA\pA\pTE1V1X1Q0S2=255S12=255S50=6S180=0\r\c OK\r \EATDT\T\r\c
CONNECT\s9600 STTY=crtscts,crtsxoff
```

次の表に、Dialers ファイルの send 文字列でよく使用されるエスケープ文字を示します。

表 26–3 /etc/uucp/Dialers で使用するエスケープ文字

文字	説明
\b	バックスペース文字を送信または想定します。
\c	改行、キャリッジリターンを抑止します。
\d	約 2 秒の遅延が生じます。
\D	Dialcodes 変換なしの電話番号またはトークン
\e	エコーチェックを使用しません。
\E	低速デバイス用にエコーチェックを使用します。
\K	ブレーク文字を挿入します。
\n	改行文字を送信します。
\nnn	8 進数値を送信します。使用できるその他のエスケープ文字については、573 ページの「UUCP /etc/uucp/Systems ファイル」を参照してください。
\N	NULL 文字 (ASCII NUL) を送信または想定します。
\p	約 12 から 14 秒の一時停止が生じます。
\r	リターン。
\s	スペース文字を送信または想定します。
\T	Dialcodes 変換を伴う電話番号またはトークン。

次に示すのは、Dialers ファイルの penril エントリです。

```
penril =W-P "" \d > Q\c : \d- > s\p9\c )-W\p\r\ds\p9\c-) y\c : \E\TP > 9\c OK
```

最初に、電話番号引数の置換メカニズムが確立されます。その結果、= はすべて W (発信音待ち) で置き換えられ、- はすべて P (一時停止) で置き換えられるようになります。

上記の行の残りの部分に指定されているハンドシェークの働きは、次のとおりです。

- "" – 何も待たない (つまり次へ進む)
- \d – 2 秒間の遅延のあとキャリッジリターンを送信する
- > – > を待つ
- Q\c – キャリッジリターンを付けずに Q を送信する
- : – : を待つ
- \d- – 2 秒間の遅延のあと - とキャリッジリターンを送信する

- > – > を待つ
- s\p9\c – s を送信し、一時停止し、9 を送信するが、キャリッジリターンは送信しない
-)-W\p\r\ds\p9\c-) –) を待つ。) が受信されない場合は、- 文字の間の文字列を処理する。つまり、W を送信し、一時停止し、キャリッジリターンを送信し、遅延し、s を送信し、一時停止し、9 を送信し、キャリッジリターンを送信しないで) を待つ
- y\c – キャリッジリターンを付けずに y を送信する
- : – : を待つ
- \E\TP – \E はエコーチェックを有効にする。これ以降は、1 文字送信するたびに、UUCP はその文字が受信されるまで待ってから処理を行う。次に電話番号を送信する。\T は、引数として渡された電話番号をとることを意味する。\T は Dialcodes 変換と、このエントリのフィールド 2 で指定されたモデム機能変換を適用する。次に、\T は P とキャリッジリターンを送信する
- > – > を待つ
- 9\c – 改行を付けずに 9 を送信する
- OK – 文字列 OK を待つ

/etc/uucp/Dialers ファイルによるハードウェアフロー制御の有効化

擬似送信文字列 STTY=*value* を用いることによっても、モデムの特性を設定できます。たとえば、STTY=crtscts を使用すると、出力ハードウェアフロー制御が可能になります。 STTY=crtsxoff を使用すると、入力ハードウェアフロー制御が可能になります。STTY=crtscts,crtsxoff を使用すると、入出力の両方のハードウェアフロー制御が可能になります。

STTY はすべての stty モードを受け入れます。詳細は、stty(1) と termio(7I) のマニュアルページを参照してください。

次の例は、Dialers ファイルエントリ内でハードウェアフロー制御を使用可能にしています。

```
dsi =,–, "" \dA\pTE1V1X5Q0S2=255S12=255*E1*F3*M1*S1\r\c OK\r \EATDT\T\r\c
CONNECT\sEC STTY=crtscts
```

この擬似送信文字列は、Systems ファイルのエントリの中でも使用できます。

/etc/uucp/Dialers ファイルでのパリティーの設定

場合によっては、呼び出そうとしているシステムがポートのパリティーを検査し、パリティーに誤りがあると回線を切断することがあります。そのため、パリティーのリセットが必要になります。expect-send の対を成す文字列として P_ZERO を使用すると、パリティーが 0 に設定されます。

```
foo =,-, "" P_ZERO "" \dA\pTE1V1X1Q0S2=255S12=255\r\c OK\r\EATDT\T\r\c CONNECT
```

次に、expect-send 文字列ペアのあとに続けることができるパリティー文字列ペアを示します。

"" P_EVEN　パリティーを偶数 (デフォルト) に設定する

"" P_ODD　パリティーを基数に設定する

"" P_ONE　パリティーを 1 に設定する

この擬似送信文字列は、Systems ファイルのエントリの中でも使用できます。

その他の基本的な UUCP 構成ファイル

基本的な UUCP 構成を行うときに、Systems、 Devices、および Dialers の各ファイルに加えて、この節で紹介するファイルを使用できます。

UUCP /etc/uucp/Dialcodes ファイル

/etc/uucp/Dialcodes ファイルにより、/etc/uucp/Systems ファイルの Phone フィールドで使用するダイアルコードの省略名を定義できます。Dialcodes ファイルは、同じサイトにある複数のシステムが使用する基本的な電話番号について、付加的な情報を指定するために使用できます。

各エントリの構文は次のとおりです。

```
Abbreviation   Dial-Sequence
```

Abbreviation　このフィールドは、Systems ファイルの Phone フィールドで使用される省略名です。

Dial-Sequence　このフィールドは、個々の Systems ファイルエントリがアクセスされるときにダイアラに渡されるダイアルシーケンスです。

この 2 つのファイル内のフィールドを比較してみます。次に、Dialcodes ファイルのエントリを示します。

Abbreviation　Dial-Sequence

次に、Systems ファイルのエントリを示します。

System-Name　Time　Type　Speed　**Phone**　Chat Script

次の表に、Dialcodes ファイルのフィールドのコンテンツ例を示します。

表 26–4　Dialcodes ファイルのエントリ

略語	ダイアルシーケンス
NY	1=212
jt	9+847

最初の行の NY は、Systems ファイルの Phone フィールドで使用される省略名です。Systems ファイルのエントリは、たとえば次のようになります。

NY5551212

uucico は、Systems ファイルから NY を読み取ると、Dialcodes ファイルから NY を探し、それに該当するダイアルシーケンス 1=212 を取得します。1=212 は、New York City への電話呼び出しに必要なダイアルシーケンスです。このシーケンスは、1 という番号と、一時停止して次の発信音を待つことを示す等号 (=) と、市外局番 212 で構成されています。uucico はこの情報をダイアラに送り、再び Systems ファイルに戻って残りの電話番号 5551212 を処理します。

jt 9=847- というエントリは、Systems ファイル内の jt7867 などのような Phone フィールドを取り扱います。uucico は、jt7867 を含むエントリを Systems ファイルから読み取り、ダイアラとトークンのペアの中のトークンが \T であれば、9=847-7867 というシーケンスをダイアラに送ります。

UUCP /etc/uucp/Sysfiles ファイル

/etc/uucp/Sysfiles ファイルでは、uucp と cu が Systems、Devices、Dialers ファイルとして使用する別のファイルを割り当てます。cu の詳細は、cu(1C) のマニュアルページを参照してください。Sysfiles は次の目的に使用できます。

- 別の Systems ファイルにより、uucp のサービスとは異なるアドレスに対してログインサービスを要求できます。
- 別の Dialers ファイルにより、cu と uucp で異なるハンドシェークを割り当てることができます。
- 複数の Systems、Dialers、Devices ファイル。特に Systems ファイルはサイズが大きくなるので、いくつかの小さいファイルに分割しておくと便利です。

Sysfiles ファイルの構文は次のとおりです。

```
service=w systems=x:x dialers=y:y devices=z:z
```

w uucico、cu、またはその両方をコロンで区切って指定します。

x Systems ファイルとして使用される 1 つまたは複数のファイルをコロンで区切って指定します。これらは指定された順序で読み込まれます。

y Dialers ファイルとして使用される 1 つまたは複数のファイルです。

z Devices ファイルとして使用される 1 つまたは複数のファイルです。

フルパスで指定しないかぎり、各ファイル名は /etc/uucp ディレクトリからの相対パスとみなされます。

次に示すのは、標準の /etc/uucp/Systems に加えて使用するローカル Systems ファイル (Local_Systems) を定義する /etc/uucp/Sysfiles の例です。

```
service=uucico:cu systems=Systems :Local_Systems
```

/etc/uucp/Sysfiles の中にこのエントリがある場合、uucico と cu はどちらも、まず標準 /etc/uucp/Systems ファイルを調べます。呼び出そうとしているシステムのエントリがそのファイル内にないか、またはそのファイル内の該当エントリの処理に失敗した場合は、両コマンドは /etc/uucp/Local_Systems を調べます。

上記のエントリの場合は、cu と uucico は、Dialers ファイルと Devices ファイルを共有します。

uucico サービス用と cu サービス用に別の Systems ファイルを定義した場合は、マシンは 2 つの異なる Systems のリストを持つことになります。uucico リストは uuname コマンドを使用して表示でき、cu リストは uuname -C コマンドを使用して表示できます。このファイルのもう 1 つの例として、代替ファイルの方を先に調べ、デフォルトファイルは必要なときだけ調べる場合を次に示します。

```
service=uucico systems=Systems.cico:Systems
  dialers=Dialers.cico:Dialers \
devices=Devices.cico:Devices
  service=cu systems=Systems.cu:Systems \
dialers=Dialers.cu:Dialers \
  devices=Devices.cu:Devices
```

UUCP /etc/uucp/Sysname ファイル

UUCP を使用するすべてのマシンは、一般にノード名と呼ばれる識別名を持っている必要があります。このノード名は、リモートマシンの /etc/uucp/Systems ファイルに、chat スクリプトやその他の識別情報とともに格納されています。通常は、UUCP は、uname -n コマンドから返されるものと同じノード名を使用し、TCP/IP でもこの名前を使用します。

/etc/uucp/Sysname ファイルを作成することによって、TCP/IP ホスト名とは別の UUCP ノード名を指定できます。このファイルには、ローカルシステムの UUCP ノード名が入った 1 行のエントリが含まれています。

UUCP /etc/uucp/Permissions ファイル

/etc/uucp/Permissions ファイルは、ログイン、ファイルアクセス、およびコマンド実行に関するリモートコンピュータのアクセス権を指定します。リモートコンピュータがファイルを要求する権限と、ローカルマシンでキューに入れられたファイルを受け取る権限を制限するオプションがあります。また、リモートマシンがローカルコンピュータ上で実行できるコマンドを指定するオプションもあります。

UUCP 構造のエントリ

各エントリは 1 行の論理行で、行末にバックスラッシュ (\) がある場合は次の行と継続していることを示します。エントリは、スペースで区切られたオプションから構成されます。各オプションは、次の形式の名前と値のペアです。

name=value

values はコロンで区切ってリストとすることもできます。オプション指定の中では、スペースは使用できないので注意してください。

コメント行はポンド記号 (#) で始まり、その行の改行文字までの全部分を占めます。空行は無視されます (複数行エントリの中の空行も同じです)。

Permissions ファイルのエントリの種類を次に示します。

- LOGNAME – リモートマシンがローカルマシンにログインする (呼び出す) ときに有効になるアクセス権を指定する

  注 – リモートマシンがローカルマシンを呼び出すとき、固有のログインと検証可能なパスワードを使用しないかぎり、そのリモートマシンの識別情報は正確なものとはなりません。

- MACHINE – ローカルマシンがリモートコンピュータにログインする (呼び出す) ときに有効になるアクセス権を指定する

LOGNAME には LOGNAME オプションが含まれ、MACHINE エントリには MACHINE オプションが含まれます。1 つのエントリに両方のオプションを含めることもできます。

UUCP の考慮事項

Permissions ファイルを使用して、リモートコンピュータに付与されているアクセスのレベルを制限するときは、次のことを考慮に入れる必要があります。

- リモートコンピュータが、UUCP 通信を目的としてログインするために使用するすべてのログイン ID は、1 つの LOGNAME エントリだけに指定されていなければならない
- 呼び出されたサイトの名前が MACHINE エントリにない場合、そのサイトには次に示すデフォルトのアクセス権または制約が適用される
  - ローカルの送信要求と受信要求は実行される
  - リモートコンピュータは、ローカルコンピュータの /var/spool/uucppublic ディレクトリにファイルを送信できる
  - リモートコンピュータがローカルコンピュータで実行するために送信するコマンドは、デフォルトのコマンドのどれかでなければならない (通常は rmail)

UUCP REQUEST オプション

リモートコンピュータがローカルコンピュータを呼び出し、ファイルの受信を要求したときに、その要求を承認することも拒否することもできます。REQUEST オプションは、リモートコンピュータがローカルコンピュータからのファイル転送を要求できるかどうかを指定します。REQUEST=yes は、リモートコンピュータがローカルコンピュータからのファイル転送を要求できることを指定します。REQUEST=no は、リモートコンピュータがローカルコンピュータからのファイルの受信を要求できないことを指定します。REQUEST=no は、REQUEST オプションを指定しなかった場合に使用されるデフォルト値です。REQUEST オプションは、LOGNAME エントリ (リモートコンピュータがローカルコンピュータを呼び出す場合) と、MACHINE エントリ (ローカルコンピュータがリモートコンピュータを呼び出す場合) のどちらにも使用できます。

UUCP SENDFILES オプション

ローカルコンピュータを呼び出す作業を完了したあとで、リモートコンピュータはローカルコンピュータのキュー中のリモートコンピュータ用の作業を受け取ろうとすることがあります。 SENDFILES オプションは、ローカルコンピュータが、リモートコンピュータ用にキューに入れた作業を送信できるかどうかを指定します。

文字列 SENDFILES=yes は、リモートコンピュータが LOGNAME オプションに指定されている名前の 1 つを使用してログインしていれば、ローカルコンピュータがキューに入れた作業を送信できることを指定します。/etc/uucp/Systems の Time フィールドに Never を入力してある場合は、この文字列の使用は必須です。その場合、ローカル

マシンは受動モードに設定され、相手のリモートコンピュータへの呼び出しを開始することはできなくなります。詳細は、573 ページの「UUCP /etc/uucp/Systems ファイル」 を参照してください。

文字列 SENDFILES=call は、ローカルコンピュータがリモートコンピュータを呼び出したときにかぎり、ローカルコンピュータのキュー中のファイルを送信することを指定します。call の値は SENDFILES オプションのデフォルト値です。MACHINE エントリはリモートコンピュータへの呼び出しを送る場合に適用されるものなので、このオプションが意味を持つのは LOGNAME エントリの中で使用した場合だけです。MACHINE エントリでこのオプションを使用しても無視されます。

UUCP MYNAME オプション

このオプションを使用すると、hostname コマンドから戻される TCP/IP ホスト名以外に、固有の UUCP ノード名をローカルシステムに与えることができます。たとえば、偶然にほかのシステムと同じ名前をローカルホストに付けてしまった場合などに、Permissions ファイルの MYNAME オプションを指定できます。自分の所属組織が widget という名前で認識されるようにするとします。すべてのモデムが gadget というホスト名を持つマシンに接続されている場合は、gadget の Permissions ファイルに次のようなエントリを含めることができます。

```
service=uucico systems=Systems.cico:Systems
  dialers=Dialers.cico:Dialers \
  devices=Devices.cico:Devices
service=cu systems=Systems.cu:Systems \
  dialers=Dialers.cu:Dialers \
  devices=Devices.cu:Devices
```

これで、システム world は、あたかも widget にログインしているかのようにマシン gadget にログインできます。ローカルマシンから world マシンを呼び出したときにも、world が widget という別名で認識するようにする場合は、次のようなエントリを作成します。

```
MACHINE=world MYNAME=widget
```

MYNAME オプションによってローカルマシンが自分自身を呼ぶこともできるので、このオプションはテスト目的にも利用できます。しかし、このオプションはマシンの実際の識別情報を隠す目的にも使用できてしまうので、601 ページの「UUCP VALIDATE オプション」で述べる VALIDATE オプションを使用するようにしてください。

UUCP READ オプションと WRITE オプション

これらのオプションは、uucico がファイルシステムのどの部分を読み書きできるかを指定します。READ オプションと WRITE オプションは、MACHINE エントリと LOGNAME エントリのどちらにも使用できます。

次の文字列に示すように、READ オプションと WRITE オプションのどちらも、デフォルトは uucppublic ディレクトリです。

```
READ=/var/spool/uucppublic WRITE=/var/spool/uucppublic
```

文字列 READ=/ と WRITE=/ は、Other 権を持つローカルユーザーがアクセスできるすべてのファイルにアクセスできる権限を指定します。

これらのエントリの値は、コロンで区切ったパス名のリストです。READ オプションはリモート側からのファイル要求のためのものであり、WRITE オプションはリモート側からのファイル送出のためのものです。値の 1 つは、入力ファイルまたは出力ファイルのフルパス名の接頭辞でなければなりません。公開ディレクトリのほかに /usr/news にもファイルを送出する権限を付与するには、WRITE オプションに次の値を指定します。

```
WRITE=/var/spool/uucppublic:/usr/news
```

パス名はデフォルトのリストに追加されるものではないので、READ オプションと WRITE オプションを使用するときはすべてのパス名を指定する必要があります。たとえば、WRITE オプションでパス名として /usr/news のみを指定した場合、公開ディレクトリにファイルを送出する権限は失われます。

リモートシステムがどのディレクトリに読み書きのアクセスができるかは、注意して決定しなければなりません。たとえば、/etc ディレクトリには多数の重要なシステムファイルが入っています。したがって、このディレクトリにファイルを送出する権限はリモートユーザーには付与しない方が賢明です。

UUCP NOREAD オプションと NOWRITE オプション

NOREAD オプションと NOWRITE オプションは、READ と WRITE オプションまたはデフォルトに対する例外を指定します。次のエントリは、/etc ディレクトリ (およびこの下の各サブディレクトリ) の中のファイルを除くすべてのファイルの読み取りを許可しています。このパス名は接頭辞であることを忘れないでください。

```
READ=/ NOREAD=/etc WRITE=/var/spool/uucppublic
```

このエントリは、デフォルトの /var/spool/uucppublic ディレクトリへの書き込みだけを許可しています。NOWRITE も NOREAD オプションと同じ形で働きます。NOREAD オプションと NOWRITE オプションは、LOGNAME エントリと MACHINE エントリのどちらにも使用できます。

UUCP CALLBACK オプション

LOGNAME エントリの中で CALLBACK オプションを使用すると、呼び出し側システムがコールバックするまで、トランザクションを一切行わないことを指定できます。CALLBACK を設定する理由を次に示します。

- セキュリティー – マシンをコールバックすることで、それが正しいマシンであることを確認できます。
- 課金 – データの伝送を長時間行うときに、その長時間の呼び出しの料金を課すマシンを選択できます。

文字列 CALLBACK=yes は、ファイル転送を行う前に、ローカルコンピュータがリモートコンピュータをコールバックしなければならないということを指定します。

CALLBACK オプションのデフォルトは CALLBACK=no です。CALLBACK を yes に設定する場合は、呼び出し側に対応する MACHINE エントリの中で、以後の通信に影響を与えるアクセス権を指定する必要があります。これらのアクセス権は、LOGNAME の中や、リモートマシンがローカルホストに対して設定している LOGNAME エントリの中では指定しないでください。

注 – 2 つのサイトが互いに CALLBACK オプションを設定すると、通信が開始されないので注意してください。

UUCP COMMANDS オプション

注意 – COMMANDS オプションは、システムのセキュリティーを低下させる恐れがあります。このオプションは十分に注意して使用してください。

COMMANDS オプションは、リモートコンピュータがローカルコンピュータ上で実行できるコマンドを指定するために、MACHINE エントリの中で使用できます。uux プログラムは、リモート実行要求を生成し、それらの要求をリモートコンピュータに転送するためにキューに入れます。ファイルとコマンドはターゲットコンピュータに送

られて、リモート実行されます。MACHINE エントリは、ローカルシステムが呼び出しを行う場合にかぎり適用されるという規則がありますが、このオプションは例外です。

COMMANDS は LOGNAME エントリの中では使えないという点に注意してください。MACHINE エントリの中の COMMANDS は、ローカルシステムがリモートシステムを呼び出すのか、リモートシステムがローカルシステムを呼び出すのかに関係なく、コマンド権限を定義します。

リモートコンピュータがローカルコンピュータ上で実行できるデフォルトのコマンドは、文字列 COMMANDS=rmail となります。MACHINE エントリの中でコマンド文字列を使用した場合は、デフォルトのコマンドよりも優先されます。たとえば、次のエントリは、COMMANDS のデフォルトを無効にして、owl、raven、hawk、dove という名前の各コンピュータが、rmail、rnews、lp の各コマンドをローカルコンピュータで実行できるようにします。

```
MACHINE=owl:raven:hawk:dove COMMANDS=rmail:rnews:lp
```

上記で指定した名前に加えて、コマンドのフルパス名も指定できます。たとえば、次のエントリは、rmail コマンドがデフォルトの検索パスを使用することを指定しています。

```
COMMANDS=rmail:/usr/local/rnews:/usr/local/lp
```

UUCP のデフォルトの検索パスは、/bin と /usr/bin です。リモートコンピュータが、実行するコマンドとして rnews または /usr/local/rnews を指定した場合は、デフォルトのパスに関係なく /usr/local/rnews が実行されます。同様に、実行される lp コマンドは /usr/local/lp です。

リストに ALL という値を含めると、エントリに指定されたリモートコンピュータから、すべてのコマンドが実行できます。この値を使用した場合は、リモートコンピュータにローカルマシンへのフルアクセスを与えることになります。

注意 – これは、通常のユーザーが持っているよりもはるかに多くのアクセス権を与えることになります。この値を使用するのは、両方のマシンが同じサイトにあり、緊密に接続されていて、ユーザーが信頼できる場合に限定するようにしてください。

ALL が追加された文字列を次に示します。

```
COMMANDS=/usr/local/rnews:ALL:/usr/local/lp
```

この文字列は、次の 2 点を示しています。

- ALL の値は文字列の中のどこでも使用できる
- 要求されたコマンドに rnews や lp コマンドのフルパス名が指定されていない場合は、デフォルトではなく、 rnews や lp それぞれに指定されているパス名が使用される

COMMANDS オプションで cat や uucp などのように、潜在的な危険性のあるコマンドを指定するときは、VALIDATE オプションを使用するようにしてください。UUCP リモート実行デーモン (uuxqt) により実行する場合、ファイルを読み書きするコマンドは、どれもローカルセキュリティーにとって危険性のあるものとなります。

UUCP VALIDATE オプション

VALIDATE オプションは、マシンのセキュリティーにとって危険性があると考えられるコマンドを指定するときに、COMMANDS オプションといっしょに使用します。VALIDATE は、コマンドアクセスを開放する方法としては ALL より安全ですが、COMMANDS オプションのセキュリティーのレベルを補強するだけのものです。

VALIDATE は、呼び出し側マシンのホスト名と、そのマシンが使用しているログイン名とを相互にチェックするものであり、呼び出し側の識別情報について、ある程度の検証機能を備えています。この例では、widget または gadget 以外のマシンが Uwidget としてログインしようとすると、接続は拒否されます。

```
LOGNAME=Uwidget VALIDATE=widget:gadget
```

VALIDATE オプションを使用する場合、権限が与えられたコンピュータは UUCP トランザクション用に固有のログインとパスワードを持っていなければなりません。この認証処理では、このエントリに対応するログインとパスワードを保護することが重要な条件の 1 つです。部外者がこの情報を入手してしまうと、VALIDATE オプションはセキュリティーに関する役割をまったく果たさなくなります。

UUCP トランザクションについて、特権を持つログインとパスワードをどのリモートコンピュータに付与するかについては、十分に検討する必要があります。ファイルアクセスとリモート実行の権限をリモートコンピュータに与えるということは、そのリモートコンピュータのすべてのユーザーに対して、ローカルコンピュータに対する通常のログインとパスワードを与えるのと同じことです。したがって、リモートコンピュータに信頼のおけないユーザーがいると判断した場合は、そのコンピュータには特権的なログインとパスワードは付与しないようにしてください。

次のような LOGNAME エントリは、eagle、owl、または hawk としてのいずれかのリモートコンピュータがローカルコンピュータにログインする場合に、そのコンピュータがログイン uucpfriend を使用している必要があることを指定します。

```
LOGNAME=uucpfriend VALIDATE=eagle:owl:hawk
```

部外者が uucpfriend を入手したとすれば、簡単に偽装することができます。

それでは、MACHINE エントリの中でだけ使用される COMMANDS オプションに対して、このエントリはどのような効果を持つのでしょうか。このエントリは、MACHINE エントリ (および COMMANDS オプション) を、特権ログインに対応する LOGNAME エントリにリンクします。このリンクが必要なのは、リモートコンピュータがログインしている時点では、実行デーモンはまだ動作していないためです。実際に、このリンクはどのコンピュータが実行要求を送ったのかを認識しない非同期プロセスです。ここで問題になるのが、実行ファイルがどこから送られてきたのかを、ローカルコンピュータがどのようにして知るかという点です。

各リモートコンピュータは、ローカルマシン上にそれぞれ専用スプールディレクトリを持っています。これらのスプールディレクトリの書き込み権限は、UUCP プログラムだけに与えられています。リモートコンピュータからの実行ファイルは、ローカルコンピュータに転送されたあとに、このスプールディレクトリに入れられます。uuxqt デーモンが動作するときには、スプールディレクトリ名を使用して、Permissions ファイルから MACHINE エントリを見つけ、COMMANDS リストを取得します。Permissions ファイル内に該当するコンピュータ名が見つからない場合は、デフォルトのリストが使用されます。

次の例は、MACHINE エントリと LOGNAME エントリの関係を示しています。

```
MACHINE=eagle:owl:hawk REQUEST=yes \
COMMANDS=rmail:/usr/local/rnews \
READ=/ WRITE=/
LOGNAME=uucpz VALIDATE=eagle:owl:hawk \
REQUEST=yes SENDFILES=yes \
READ=/ WRITE=/
```

COMMANDS オプションの値は、リモートユーザーが、rmail と /usr/local/rnews を実行できることを示しています。

最初のエントリでは、一覧表示されているコンピュータのどれかを呼び出す場合に、実際には eagle、owl、hawk のどれかを呼び出すということを理解しておく必要があります。したがって、eagle、owl、および hawk のスプールディレクトリに置かれるファイルはすべて、それらのコンピュータのどれかによって置かれます。あるリモートコンピュータがログインし、この 3 つのコンピュータのどれかであることを主張した場合、その実行ファイルもこの特権スプールディレクトリに入れられます。したがって、ローカルコンピュータでは、そのコンピュータが特権ログイン uucpz を持っていることを確認する必要があります。

UUCP OTHER 用の MACHINE エントリ

特定の MACHINE エントリに記述されていないリモートマシンについて、異なるオプション値を指定したい場合があります。これが必要になるのは、多数のコンピュータがローカルホストを呼び出し、コマンドセットがそのたびに異なるような場合です。次の例に示すように、このようなエントリでは、コンピュータ名として OTHER という名前を使用します。

```
MACHINE=OTHER \
COMMANDS=rmail:rnews:/usr/local/Photo:/usr/local/xp
```

ほかの MACHINE エントリに記述されていないコンピュータについても、MACHINE エントリに使用できるすべてのオプションを設定できます。

UUCP の MACHINE エントリと LOGNAME エントリの結合

共通オプションが同じである場合、MACHINE エントリと LOGNAME エントリを結合して、単一のエントリにすることができます。たとえば、次の 2 セットのエントリは、同じ REQUEST、READ、WRITE オプションを共有しています。

```
MACHINE=eagle:owl:hawk REQUEST=yes \
READ=/ WRITE=/
```

および

```
LOGNAME=uupz REQUEST=yes SENDFILES=yes \
READ=/ WRITE=/
```

この 2 つのエントリを結合したものを次に示します。

```
MACHINE=eagle:owl:hawk REQUEST=yes \
logname=uucpz SENDFILES-yes \
READ=/ WRITE=/
```

MACHINE エントリと LOGNAME エントリを結合することによって、Permissions ファイルは、効率的で管理しやすくなります。

UUCP の転送

一連のマシンを介してファイルを送信するときは、リレー (中継) マシンの COMMANDS オプションの中に uucp コマンドが含まれていなければなりません。次のコマンドを入力した場合、マシン willow がマシン oak に対して uucp プログラムの実行を許可する場合にかぎり、この転送操作は正常に機能します。

```
% uucp sample.txt oak\!willow\!pine\!/usr/spool/uucppublic
```

oak もローカルマシンに uucp のプログラムの実行を許可している必要があります。最終宛先マシンである pine は、転送動作を行わないため、uucp コマンドを許可する必要はありません。通常、マシンはこのように設定されていません。

UUCP /etc/uucp/Poll ファイル

/etc/uucp/Poll ファイルには、リモートコンピュータをポーリングするための情報が入っています。Poll ファイル内の各エントリには、呼び出すリモートコンピュータの名前と、それに続くタブ文字またはスペース、最後にそのコンピュータを呼び出す時刻が入ります。Poll ファイル内のエントリの形式は次のとおりです。

sys-name hour ...

たとえば、エントリを **eagle 0 4 8 12 16 20** と指定すると、コンピュータ eagle が 4 時間ごとにポーリングされます。

uudemon.poll スクリプトは Poll ファイルを処理しますが、実際にポーリングを行うわけではありません。単にスプールディレクトリ内にポーリング作業ファイル (名前は常に C.*file*) を設定するだけです。uudemon.poll スクリプトはスケジューラを起動し、スケジューラは、スプールディレクトリ内のすべての作業ファイルを調べます。

UUCP /etc/uucp/Config ファイル

/etc/uucp/Config ファイルを使用すると、いくつかのパラメータを手動で上書きできます。Config ファイルの各エントリの形式は次のとおりです。

parameter=value

構成可能な全パラメータ名のリストについては、システムに付属している Config ファイルを参照してください。

次の Config エントリは、デフォルトのプロトコル順序を Gge に設定し、G プロトコルのデフォルト値を、ウィンドウ数 7、パケットサイズ 512 バイトに変更します。

```
Protocol=G(7,512)ge
```

UUCP /etc/uucp/Grades ファイル

/etc/uucp/Grades ファイルには、リモートコンピュータへのジョブをキューに入れるときに指定できるジョブグレードが入っています。また、個々のジョブグレードに関するアクセス権も含まれています。このファイルのエントリは、ユーザーがジョブをキューに入れるときに使用する、管理者が定義したジョブグレードの定義を表しています。

Grades ファイルのエントリの形式は次のとおりです。

User-job-grade System-job-grade Job-size Permit-type ID-list

各エントリには、スペースで区切ったいくつかのフィールドがあります。エントリの最後のフィールドは、同じくスペースで区切ったいくつかのサブフィールドから構成されます。1 つのエントリが複数の物理行にわたる場合は、バックスラッシュを使用して、エントリを次の行に継続させることができます。コメント行はポンド記号 (#) で始まり、その行の全体を占めます。空の行は常に無視されます。

UUCP User-job-grade フィールド

このフィールドには、管理者が 64 文字以内で定義したユーザージョブのグレード名が入ります。

UUCP System-job-grade フィールド

このフィールドには、*User-job-grade* が対応付けされる 1 文字のジョブグレードが入ります。有効な文字は A Z、a z で、最も優先順位が高いのは A、最も優先順位が低いのは z です。

ユーザージョブグレードとシステムジョブグレードの関係

ユーザージョブグレードは複数のシステムジョブグレードに割り当てることができます。Grades ファイルは、ユーザージョブグレードのエントリを見つけるために先頭から検索されるという点に注意してください。したがって、最大ジョブサイズの制限値に応じて、複数のシステムジョブグレードのエントリが列挙されます。

ユーザージョブグレードの最大数には制限はありませんが、システムジョブグレードの許容最大数は 52 です。その理由は、1 つの *System-job-grade* には複数の *User-job-grade* を対応付けできるが、個々の *User-job-grade* はファイル内でそれぞれ単独の行でなければならないという点にあります。次に例を示します。

```
mail N Any User Any netnews N Any User Any
```

Grades ファイル内でこのような構成をした場合、2 つの *User-job-grade* が同じ *System-job-grade* を共有します。ジョブグレードに関するアクセス権は、*System-job-grade* ではなく *User-job-grade* に割り当てられるものなので、2 つの *User-job-grade* は同じ *System-job-grade* を共有しながら、それぞれ異なるアクセス権のセットを持つことができます。

デフォルトグレード

デフォルトのユーザージョブグレードとして、システムジョブグレードを割り当てることができます。そのためには、Grades ファイルの *User-job-grade* フィールドのユーザージョブグレードとしてキーワード default を使用し、そのデフォルトに割り当てるシステムジョブグレードを指定します。Restriction フィールドと ID フィールドは Any と定義して、どのようなユーザー、どのようなサイズのジョブでも、このグレードでキューに入れることができるようにします。次に例を示します。

```
default a Any User Any
```

デフォルトのユーザージョブグレードを定義しなかった場合は、組み込まれているデフォルトグレードである Z が使用されます。Restriction フィールドのデフォルトは Any なので、デフォルトグレードのエントリが複数存在していても検査されません。

UUCP Job-size フィールド

このフィールドは、キューに入れることのできる最大ジョブサイズを指定します。*Job-size* はバイト数で表され、次のリストに示すオプションを使用できます。

nnnn	このジョブグレードの最大ジョブサイズを指定する整数
n K	K バイト数を表す 10 進数 (K はキロバイトの略号)
n M	M バイト数を表す 10 進数 (M はメガバイトの略号)
Any	最大ジョブサイズが指定されないことを指定するキーワード

次に例をいくつか示します。

- 5000 は 5000 バイトを表す
- 10K は 10K バイトを表す
- 2M は 2M バイトを表す

UUCP Permit-type フィールド

このフィールドには、ID リストをどのように解釈するかを指示するキーワードを指定します。次の表に、キーワードとそれぞれの意味を示します。

表 26–5　Permit-type フィールド

キーワード	ID リストの内容
User	このジョブグレードの使用を許可されているユーザーのログイン名
Non-user	このジョブグレードの使用を許可されていないユーザーのログイン名
Group	このジョブグレードの使用を許可されているメンバーのグループ名
Non-group	このジョブグレードの使用を許可されていないメンバーのグループ名

UUCP ID-list フィールド

このフィールドには、このジョブグレードへキューを入れることが許可、禁止されるログイン名またはグループ名のリストが入ります。名前のリストはそれぞれスペースで区切り、改行文字で終了します。このジョブグレードへキューを入れることをだれにでも許可する場合は、キーワード Any を使用します。

その他の UUCP 構成ファイル

この節では、UUCP の機能に影響を与えるファイルのうち、比較的変更頻度の低い 3 つのファイルについて説明します。

UUCP /etc/uucp/Devconfig ファイル

/etc/uucp/Devconfig ファイルを使用すると、サービス別に、つまり uucp 用や cu 用などに分けて、デバイスを構成できます。Devconfig のエントリは、個々のデバイスで使用される STREAMS モジュールを定義します。これらの書式は次のとおりです。

service=*x* device=*y* push=*z*[:*z*...]

x は、cu か uucico、またはその両方のサービスをコロンで区切ったものです。*y* はネットワークの名前で、これは Devices ファイルのエントリに一致していなければなりません。*z* には、STREAMS モジュールの名前を、Stream にプッシュする順序で指定します。cu サービスと uucp サービスについて、それぞれ異なるモジュールとデバイスを定義できます。

次のエントリは STARLAN ネットワーク用のもので、このファイル内でもっともよく使用されるものです。

```
service=cu       device=STARLAN     push=ntty:tirdwr
service=uucico   device=STARLAN     push=ntty:tirdwr
```

この例では、まず ntty、次に tirdwr がプッシュされます。

UUCP /etc/uucp/Limits ファイル

/etc/uucp/Limits ファイルは、uucp ネットワーク処理で同時に実行できる uucico、uuxqt、および uusched の最大数を制御します。ほとんどの場合は、デフォルトの値が最適であり、変更の必要はありません。変更する場合は、任意のテキストエディタを使用してください。

Limits ファイルの形式は次のとおりです。

service=*x* max=*y*:

x は uucico、uuxqt、uusched のどれかで、*y* はそのサービスについての制限値です。フィールドは、小文字を使用して任意の順序で入力できます。

次に示すのは、Limits ファイルの中で一般的に使用される内容です。

```
service=uucico max=5
service=uuxqt max=5
service=uusched max=2
```

この例は、5 つの uucico、5 つの uuxqt、2 つの uusched をマシンで実行できることを示しています。

UUCP remote.unknown ファイル

通信機能の使用に影響を与えるファイルとして、もう 1 つ remote.unknown ファイルがあります。このファイルは、どの Systems ファイルにも含まれていないマシンが通信を開始したときに実行されるバイナリプログラムです。このプログラムはその通信をログに記録し、接続を切断します。

注意 – remote.unknown ファイルのアクセス権を変更して、このファイルが実行できないようにすると、ローカルシステムはどのシステムからの接続も受け入れることになります。

このプログラムが実行されるのは、どの Systems ファイルにも含まれていないマシンが対話を開始した場合です。このプログラムは、その対話を記録し、接続を失敗させます。このファイルのアクセス権を変更して実行できないようにしてしまうと (chmod 000 remote.unknown)、ローカルシステムはすべての通信要求を受け入れることになります。妥当な理由がないかぎり、この変更は行わないようにしてください。

UUCP の管理ファイル

次に、UUCP 管理ファイルについて説明します。これらのファイルは、デバイスのロック、一時データの保管、リモート転送や実行に関する情報の保存などのために、スプールディレクトリ内に作成されます。

- 一時データファイル (TM) – これらのデータファイルは、ほかのコンピュータからファイルを受け取るときに、UUCP プロセスによりスプールディレクトリ /var/spool/uucp/*x* の下に作成されます。ディレクトリ *x* は、ファイルを送信しているリモートコンピュータと同じ名前です。一時データファイル名の形式は次のとおりです。

  TM.*pid.ddd*

  pid はプロセス ID、*ddd* は 0 から始まる 3 桁のシーケンス番号です。

  ファイルの全体が受信されると、TM. *pid.ddd* ファイルは、伝送を発生させた C. *sysnxxxx* ファイル (次で説明) の中で指定されているパス名に移されます。処理が異常終了した場合は、TM.*pid.ddd* ファイルが *x* ディレクトリ内に残ることがあります。このファイルは、uucleanup を使用することにより自動的に削除されます。

- ロックファイル (LCK) – ロックファイルは、使用中のデバイスごとに、/var/spool/locks ディレクトリ内に作成されます。ロックファイルは、対話の重複、複数の試行による同じ呼び出しデバイスの使用が発生するのを防ぎます。次の表に、UUCP ロックファイルの種類を示します。

表 26–6 UUCP ロックファイル

ファイル名	説明
LCK. *sys*	*sys* はファイルを使用しているコンピュータ名を表す
LCK. *dev*	*dev* はファイルを使用しているデバイス名を表す
LCK.*LOG*	*LOG* はロックされている UUCP ログファイルを表す

通信リンクが予定外のときに切断された場合 (コンピュータがクラッシュしたときなど)、これらのファイルがスプールディレクトリ内に残ることがあります。親プロセスが有効でなくなったあとは、ロックファイルは無視 (削除) されます。ロックファイルには、ロックを引き起こしたプロセスのプロセス ID が入っています。

- 作業ファイル (C.) – 作業ファイルは、リモートコンピュータに送る作業 (ファイル転送またはリモートコマンド実行) がキューに入れられたときに、スプールディレクトリ内に作成されます。作業ファイル名の形式は次のとおりです。

  C.*sysnxxxx*

sys はリモートコンピュータ名、*n* は作業のグレード (優先順位) を表す ASCII 文字、*xxxx* は、UUCP が割り当てる 4 桁のジョブシーケンス番号です。作業ファイルには次の情報が含まれています。

- 送信または要求するファイルのフルパス名
- 宛先、ユーザー名、またはファイル名を表すフルパス名
- ユーザーのログイン名
- オプションのリスト
- スプールディレクトリ内の関連データファイルの名前。`uucp -C` オプションか `uuto -p` オプションが指定されている場合は、ダミー名 (`D.0`) が使用される
- ソースファイルのモードビット
- 転送完了の通知を受け取るリモートユーザーのログイン名

- データファイル (`D.`) – コマンド行でスプールディレクトリへのソースファイルのコピーを指定すると、データファイルが作成されます。作業ファイル名の形式は次のとおりです。

  `D.`*systmxxxxyyy* – *systm* はリモートコンピュータ名の最初の 5 文字で、*xxxx* は `uucp` が割り当てる 4 桁のジョブシーケンス番号です。4 桁のジョブシーケンス番号のあとにサブシーケンス番号を続けることができます。*yyy* は、1 つの作業 (`C.`) ファイルについて複数の `D.` ファイルが作成された場合に使用されます。

- `X.` *(*実行ファイル*)* – 実行ファイルは、リモートコマンドの実行の前にスプールディレクトリ内に作成されます。実行ファイル名の形式は次のとおりです。

  `X.`*sysnxxxx*

  sys はリモートコンピュータ名で、*n* は作業のグレード (優先順位) を表す文字です。*xxxx* は、UUCP が割り当てる 4 桁のシーケンス番号です。実行ファイルには次の情報が入ります。

  - 要求元のログイン名とコンピュータ名
  - 実行に必要なファイル名
  - コマンド文字列への標準入力として使用する入力
  - コマンド実行の標準出力を受け取るコンピュータとファイルの名前
  - コマンド文字列
  - 終了ステータスの要求のためのオプション行

UUCP のエラーメッセージ

この節には、UUCP に関連したエラーメッセージを示します。

UUCP の ASSERT エラーメッセージ

次の表に、ASSERT エラーメッセージを示します。

表 26–7 ASSERT エラーメッセージ

エラーメッセージ	説明または処置
CAN'T OPEN	open() または fopen() が失敗した
CAN'T WRITE	write()、fwrite()、fprint()、または類似のコマンドが失敗した
CAN'T READ	read()、fgets()、または類似のコマンドが失敗した
CAN'T CREATE	creat() 呼び出しが失敗した
CAN'T ALLOCATE	動的割り当てが失敗した
CAN'T LOCK	LCK (ロック) ファイルを作成しようとしたが失敗した。場合によってはこのエラーは致命的である
CAN'T STAT	stat() 呼び出しが失敗した
CAN'T CHMOD	chmod() 呼び出しが失敗した
CAN'T LINK	link() 呼び出しが失敗した
CAN'T CHDIR	chdir() 呼び出しが失敗した
CAN'T UNLINK	unlink() 呼び出しが失敗した
WRONG ROLE	内部ロジックの問題
CAN'T MOVE TO CORRUPTDIR	不良な C. ファイルまたは X. ファイルを、/var/spool/uucp/.Corrupt ディレクトリに移動しようとしたが失敗した。このディレクトリが存在しないか、モードまたは所有者が正しくない
CAN'T CLOSE	close() または fclose() 呼び出しが失敗した
FILE EXISTS	C. ファイルまたは D. ファイルを作成しようとしたが、そのファイルがすでに存在している。このエラーは、シーケンスファイルのアクセスに問題がある場合に生じる。これは通常、ソフトエラーを示す
NO uucp SERVICE NUMBER	TCP/IP 呼び出しを試みたが、/etc/services 内に UUCP に関するエントリがない
BAD UID	ユーザー ID がパスワードデータベース内にない。ネームサービス構成の検査が必要
BAD LOGIN_UID	前記と同じ
BAD LINE	Devices ファイル内に不良な行がある。引数が足りない行が 1 つ以上ある
SYSLST OVERFLOW	gename.c の内部テーブルがオーバーフローした。1 つのジョブが 30 を超えるシステムに接続しようとした
TOO MANY SAVED C FILES	前記と同じ
RETURN FROM fixline ioctl	失敗するはずのない ioctl(2) が失敗した。システムドライバに問題がある
BAD SPEED	Devices ファイルまたは Systems ファイルの中に不適正な回線速度がある (Class フィールドまたは Speed フィールド)

表 26–7 ASSERT エラーメッセージ (続き)

エラーメッセージ	説明または処置
BAD OPTION	Permissions ファイルの中に不適正な行またはオプションがある。ただちに修正が必要
PKCGET READ	リモートマシンがハングアップした可能性がある。処置は不要
PKXSTART	リモートマシンが回復不可能な状態で異常終了した。通常このエラーは無視できる
TOO MANY LOCKS	内部的な問題がある。システムの購入先への問い合わせが必要
XMV ERROR	ファイル、またはディレクトリのどこかに問題が発生している。このプロセスが実行される前に、宛先のモードがチェックされるべきであるが実行されていないなど、スプールディレクトリに問題がある可能性がある
CAN'T FORK	fork と exec を実行しようとしたが失敗した。現行ジョブは失われず、あとで再試行される (uuxqt)。処置は不要

UUCP の STATUS エラーメッセージ

次の表に一般的な STATUS エラーメッセージを示します。

表 26–8 UUCP の STATUS エラーメッセージ

エラーメッセージ	説明または処置
OK	状態は良好
NO DEVICES AVAILABLE	現在、この呼び出し用に使用可能なデバイスがない。該当のシステムについて Devices ファイル内に有効なデバイスがあるかどうかを確認してください。そのシステムの呼び出しに使用するデバイスが Systems ファイル内にあるかどうかを検査してください
WRONG TIME TO CALL	Systems ファイルに指定されている日時以外の時点で、システムに対する呼び出しが行われた
TALKING	会話中
LOGIN FAILED	特定のマシンのログインが失敗した。ログインまたはパスワードが正しくないか、番号が正しくないか、低速のマシンであるか、Dialer-Token-Pairs スクリプトによる処理が失敗した
CONVERSATION FAILED	起動に成功したあとで対話が失敗した。一方の側がダウンしたか、プログラムが異常終了したか、回線 (リンク) が切断されたことが考えられる
DIAL FAILED	リモートマシンがまったく応答しない。ダイアラが不良であるか、電話番号が正しくない可能性がある
BAD LOGIN/MACHINE COMBINATION	あるマシンが、Permissions ファイルの条件を満たしていないログインとマシン名を使用して、ローカルマシンを呼び出そうとした。偽装の疑いがある

表 26–8 UUCP の STATUS エラーメッセージ (続き)

エラーメッセージ	説明または処置
DEVICE LOCKED	使用しようとしている呼び出しデバイスは、現在ロックされ、ほかのプロセスに使用されている
ASSERT ERROR	ASSERT エラーが発生した。/var/uucp/.Admin/errors ファイルにエラーメッセージが入っているかどうかを検査し、610 ページの「UUCP の ASSERT エラーメッセージ」を参照
SYSTEM NOT IN Systems FILE	システムが Systems ファイルの中に記述されていない
CAN'T ACCESS DEVICE	アクセスしようとしたデバイスが存在しないか、またはモードが正しくない。Systems ファイルと Devices ファイルの中の該当のエントリを検査する
DEVICE FAILED	デバイスがオープンできない
WRONG MACHINE NAME	呼び出されたマシンは、予期したのとは異なる名前である
CALLBACK REQUIRED	呼び出されたマシンは、そのマシンがローカルマシンをコールバックする必要があることを示している
REMOTE HAS A LCK FILE FOR ME	リモートマシンは、ローカルマシンに関連する LCK ファイルを持っている。そのリモートマシンがローカルマシンを呼び出そうとしている可能性がある。リモートマシンの UUCP のバージョンが古い場合は、プロセスがローカルマシンに接続しようとして失敗し、LCK ファイルがそのまま残されたことが考えられる。リモートマシンの UUCP のバージョンが新しく、ローカルマシンと通信していない場合は、LCK を持っているプロセスはハングアップする
REMOTE DOES NOT KNOW ME	リモートマシンの Systems ファイルの中に、ローカルマシンのノード名がない
REMOTE REJECT AFTER LOGIN	ローカルマシンがログインのために使用したログインが、リモートマシンが予期している内容に一致していない
REMOTE REJECT, UNKNOWN MESSAGE	理由は不明だが、リモートマシンがローカルマシンとの通信を拒否した。リモートマシンが標準バージョンの UUCP を使用していない可能性がある
STARTUP FAILED	ログインは成功したが、初期ハンドシェークに失敗した
CALLER SCRIPT FAILED	通常、これは DIAL FAILED と同じ。しかしこのエラーが頻発する場合は、Dialers ファイル内の呼び出し側スクリプトに原因があることが考えられる。Uutry を使用して検査する

UUCP の数値エラーメッセージ

次の表に、/usr/include/sysexits.h ファイルにより生成されるエラー状態メッセージの終了コード番号を示します。これらのすべてが現在 uucp で使用されているわけではありません。

表 26–9 番号による UUCP のエラーメッセージ

メッセージ番号	説明	意味
64	Base Value for Error Messages	エラーメッセージはこの番号から始まります。
64	Command–Line Usage Error	コマンドの使い方に誤りがあります。。たとえば、引数の数が正しくない、誤ったフラグ、誤った構文などです。
65	Data Format Error	入力データになんらかの誤りがあります。このデータ形式はユーザーデータだけに使用されるもので、システムファイルには使用されません。
66	Cannot Open Input	入力ファイル (システムファイルでない) が存在しないか、または読み取れません。これには、メールプログラムに対する「No message」のようなエラーも含まれます。
67	Address Unknown	指定されたユーザーが存在しません。このエラーは、メールアドレスやリモートログインに使用されます。
68	Host Name Unknown	ホストが存在しません。このエラーは、メールアドレスやネットワーク要求に使用されます。
69	Service Unavailable	サービスが使用できません。このエラーは、サポートプログラムまたはファイルが存在しない場合に起こることがあります。このメッセージは、何かが正常に働かず、現時点ではその原因が特定できないことを示す場合もあります。
70	Internal Software Error	内部ソフトウェアエラーが検出されました。このエラーは、できるだけオペレーティングシステム関係以外のエラーに限定されるべきです。
71	System Error	オペレーティングシステムエラーが検出されました。このエラーは、「フォーク不可」や「パイプ作成不可」などの場合に使用されることが想定されています。たとえば、`getuid` が `passwd` ファイル内に存在しないユーザーを戻した場合などが含まれます。
72	Critical OS File Missing	`/etc/passwd` や `/var/admin/utmpx` などのシステムファイルのどれかが存在しないか、開くことができません。あるいは、構文エラーなどがあります。
73	Can't Create Output File	ユーザーが指定した出力ファイルが作成できません。
74	Input/Output Error	あるファイルについて入出力を行なっているときにエラーが起こりました。
75	Temporary Failure. User is invited to retry	実際のエラーではない一時的な障害。たとえば `sendmail` では、これは、メールプログラムが接続を確立できなかったため、あとで要求を再試行する必要があることなどを意味します。
76	Remote Error in Protocol	プロトコルの交換中に、リモートシステムが「使用不可」を示す何かを戻しました。

表 26–9 番号による UUCP のエラーメッセージ (続き)

メッセージ番号	説明	意味
77	Permission Denied	この操作を行うための適正なアクセス権がユーザーにありません。これはファイルシステムの問題を示すものではなく (その場合は NOINPUT や CANTCREAT などが使用される)、より高いレベルのアクセス権が必要であることを意味します。たとえば、kre は、メールを送ることのできる学生を制限するためにこのメッセージを使用します。
78	Configuration Error	システムの構成にエラーがあります。
79	Entry Not Found	エントリが見つかりません。
79	Maximum Listed Value	エラーメッセージの最大番号。

パート VI

リモートシステムの利用 (トピック)

このパートでは、Solaris 環境で FTP サーバーを管理し、リモートシステムにアクセスする方法について説明します。

◆ ◆ ◆ 第 27 章

27 リモートシステムの利用 (概要)

この章では、リモートファイルの利用について説明します。

FTP サーバーとは

FTP サーバーは wu-ftpd に基づいています。ワシントン大学 (セントルイス) で開発された wu-ftpd は、インターネット上での大量データの配布に幅広く使用され、大規模な FTP サイトではよく使われる規格です。ライセンス条項については、`/var/sadm/pkg/SUNWftpu/install/copyright` から利用できるドキュメントを参照してください。

リモートシステムとは

この章では、リモートシステムとは、物理ネットワークによってローカルシステムに接続され、TCP/IP 通信用に構成されたワークステーションまたはサーバーであると想定します。

Solaris リリースのシステム上では、TCP/IP は起動時に自動的に構成されます。詳細については、『Solaris のシステム管理 (IP サービス)』を参照してください。

Solaris 10 リリース FTP サービスの変更点

Solaris 10 リリースでは、FTP サーバーの機能強化、`ftpcount` 、`ftpwho`、`ftp` コマンドの変更など、いくつかの点で FTP サービスの変更が行われています。

FTP サーバーの機能強化により、スケーラビリティーと転送のロギングが向上しました。これらのオプションの説明は、650 ページの「多忙なサイトにおける構成についてのヒント」と、`ftpaccess(4)` のマニュアルページに挙げられています。次に、具体的な変更点を示します。

- `sendfile()` 関数により、バイナリダウンロードを行う
- `ftpaccess` ファイルでサポートされる新機能
  - `flush-wait` により、ダウンロードまたはディレクトリ表示の最後の動作を制御する
  - `ipcos` により、制御接続またはデータ接続の「IP CoS (サービスクラス)」を設定する
  - `passive ports` を設定し、待機する TCP ポートをカーネルに選択させることができる
  - `quota-info` により、割り当て情報を取得できる
  - `recvbuf` により、バイナリ転送に使用される受信 (アップロード) バッファーサイズを設定する
  - `rhostlookup` により、リモートホスト名検索の許可または拒否を設定する
  - `sendbuf` により、バイナリ転送に使用される送信 (ダウンロード) バッファーサイズを設定する
  - `xferlog` により、転送ログエントリの書式をカスタマイズする
- `-4` オプションを指定すると、スタンドアロンモードで稼働している FTP サーバーは IPv4 ソケットを使用した接続だけを待機する

また、このリリースでは `ftpcount` と `ftpwho` に `-v` オプションが追加されました。このオプションは、仮想ホスト `ftpaccess` ファイルに定義されている FTP サーバークラスのユーザー数とプロセス情報を表示するものです。詳細は、`ftpcount(1)` と `ftpwho(1)` のマニュアルページを参照してください。

このリリースでは、FTP クライアントとサーバーは Kerberos をサポートするようになりました。詳細は、`ftp(4)` のマニュアルページと、『Solaris のシステム管理 (セキュリティサービス)』の「Kerberos ユーザーコマンド」を参照してください。

`ftp` コマンドは変更されました。デフォルトでは、Solaris FTP サーバーに接続された Solaris FTP クライアントは、クライアントに対して `ls` コマンドが発行されるとディレクトリとプレーンファイルの一覧を表示します。FTP サーバーが Solaris OS で稼働していない場合は、ディレクトリは表示されない可能性があります。Solaris 以外の FTP サーバーに接続している場合に Solaris のデフォルト動作が起きるようにす

るには、Solaris クライアントごとに /etc/default/ftp ファイルを適宜編集します。個々のユーザーについてこの変更を行うには、FTP_LS_SENDS_NLST 環境変数を yes に設定します。詳細は、ftp(4) のマニュアルページを参照してください。

ftpd デーモンは、サービス管理機能によって管理されます。このサービスに関する有効化、無効化、再起動などの管理アクションは svcadm コマンドを使用して実行できます。すべてのデーモンのサービスの状態は、svcs コマンドを使用して照会することができます。サービス管理機能の概要については、『Solaris のシステム管理 (基本編)』の第 18 章「サービスの管理 (概要)」を参照してください。

Solaris 9 の FTP サーバーの新機能

Solaris 9 リリースでは FTP サーバーに大幅な変更が加えられたため、Solaris 10 リリースでも引き続きこの節を残してあります。Solaris 9 の FTP サーバーは、Solaris 8 の FTP ソフトウェアとの互換性を保ちながら、新しい機能を提供してパフォーマンスの向上を図っています。

表 27–1　Solaris 9 の FTP サーバーの新機能

機能	説明	参照先
タイプと場所による ユーザーの分類	タイプとアドレスに基づいて、ユーザーのクラスを定義できる	625 ページの「FTP サーバークラスの定義方法」
クラスごとの制限	ftpaccess ファイルに設定されている制限に基づいて、同時にログインできる特定クラスのユーザー数を制御する	626 ページの「ユーザーログインの制限を設定する方法」
システム全体および ディレクトリ関連の メッセージ	特定のイベントに対して指定されるメッセージを表示する	636 ページの「ユーザーに送信する メッセージの作成方法」
ディレクトリごとの アップロード権	ファイルおよびディレクトリの作成やアクセス権など、FTP サーバーへのアップロードを制御できる	639 ページの「FTP サーバーへのアップロードの制御方法」
ファイル名フィルタ	アップロードしたファイルの名前に使用できる文字とその順序を指定できる	639 ページの「FTP サーバーへのアップロードの制御方法」
仮想ホストのサポート	単一のマシンで複数のドメインをサポートするように FTP サーバーを構成できる	644 ページの「完全仮想ホスティングを有効にする方法」
コマンドのログ	実ユーザー、ゲストユーザー、匿名ユーザーの各 FTP ユーザーが実行したコマンドのログを記録できる	650 ページの「FTP ユーザーにより実行されたコマンドの検査」
転送処理のログ	実ユーザー、ゲストユーザー、匿名ユーザーの各 FTP ユーザーが実行した転送処理のログを記録できる	ftpaccess(4)、xferlog(4)、in.ftpd(1M) のマニュアルページ

表 27–1 Solaris 9 の FTP サーバーの新機能 (続き)

機能	説明	参照先
必要時の圧縮とアーカイブ	必要に応じて、ftpconversions ファイルに指定されている変換方法で圧縮およびアーカイブができる	ftpconversions(4)、ftpaccess(4)のマニュアルページ

次に、Solaris 8 よりも後のリリースではサポートされない Solaris 8 の機能を示します。

- Solaris 8 よりも後のリリースでは、Solaris 8 の /etc/default/ftpd はサポートされていません。Solaris 8 から Solaris 9 へのアップグレード中に、BANNER および UMASK の各エントリは、wu-ftpd 用の対応するエントリに変換されます。ただし、いくつかの BANNER 行は、ftpaccess のメッセージ機能に合わせて手作業で変換する必要があります。詳細は、ftpaccess(4) のマニュアルページを参照してください。
- Solaris 8 の FTP サーバーで提供されていたサブログイン機能は、Solaris 9 の FTP サーバーではサポートされていません。

◆ ◆ ◆ 第 28 章

FTP サーバーの管理 (手順)

この章では、次の表に示す FTP サーバーを設定し、管理するための作業について説明します。

FTP サーバーの管理 (作業マップ)

表 28–1 作業マップ : FTP サーバーの管理

作業	説明	参照先
FTP サーバーへのアクセスの構成	`/etc/ftpd` ディレクトリに置かれた `ftpaccess`、`ftpusers`、`ftphosts` の各ファイルを使用して、FTP サーバーへのアクセスを確立または制限する	626 ページの「ユーザーログインの制限を設定する方法」 627 ページの「無効なログインの試行回数を制御する方法」 628 ページの「特定のユーザーの FTP サーバーへのアクセスを拒否する方法」 629 ページの「デフォルト FTP サーバーへのアクセスを制限する方法」 625 ページの「FTP サーバークラスの定義方法」

表 28–1 作業マップ: FTP サーバーの管理 (続き)

作業	説明	参照先
FTP サーバーのログインの設定	実ユーザー、ゲストユーザー、匿名ユーザーのログインアカウントを設定する	631 ページの「実 FTP ユーザーの設定方法」 632 ページの「ゲスト FTP ユーザーの設定方法」 633 ページの「匿名 FTP ユーザーの設定方法」 634 ページの「/etc/shells ファイルの作成方法」
メッセージファイルをカスタマイズする	/etc/ftpd/ftpaccess ファイルを編集して、特定のイベントに関連して FTP サーバーが FTP クライアントにメッセージを返すように構成する	635 ページの「メッセージファイルのカスタマイズ方法」 636 ページの「ユーザーに送信するメッセージの作成方法」 636 ページの「README オプションの構成方法」
FTP サーバー上のファイルへのアクセスを構成する	/etc/ftpd/ftpaccess ファイルを使用して、特定のコマンドの実行、FTP サーバーからのファイルのダウンロード、FTP サーバーへのファイルのアップロードを許可するユーザーのクラスを指定する	248 ページの「ダイアルアップネットワークに対する DA 検出の構成方法」 639 ページの「FTP サーバー上のアップロードとダウンロードの制御」
限定された仮想ホスティングまたは完全な仮想ホスティングを有効化する	/etc/ftpd/ftpaccess ファイルを使用して、FTP サーバーが同一マシン上の複数ドメインをサポートするように構成する	643 ページの「限定仮想ホスティングを有効にする方法」 644 ページの「完全仮想ホスティングを有効にする方法」
FTP サーバーを起動する	FTP サーバーが nowait モード、スタンドアロンモードまたはフォアグラウンドモードで起動するようにサービスプロパティーを変更する	646 ページの「SMF を使用して FTP サーバーを起動する方法」 647 ページの「FTP サーバーをバックグラウンドで起動する方法」 647 ページの「FTP サーバーをフォアグラウンドで起動する方法」
FTP サーバーを停止する	/etc/ftpd/ftpaccess ファイルを使用し、ftpshut を実行して FTP サーバーを停止する	648 ページの「FTP サーバーの停止」
一般的な FTP サーバーの問題をトラブルシューティングする	syslogd を検査し、greeting text と log commands を使用して FTP サーバー上の問題をデバッグする	649 ページの「syslogd 内の FTP サーバーのメッセージを検査する方法」 650 ページの「greeting text を使用して ftpaccess を検査する方法」 650 ページの「FTP ユーザーにより実行されたコマンドの検査」

FTP サーバーへのアクセスの制御

/etc/ftpd ディレクトリに置かれた構成ファイルを使用して FTP サーバーへのアクセスを制御します。次に構成ファイルを示します。

- ftpusers には、FTP サーバーへのアクセスを拒否するユーザーが列挙されています。
- ftphosts は、複数のホストから FTP サーバー上の複数のアカウントへのログインを許可または拒否するために使用します。
- ftpaccess は、メインの FTP 構成ファイルです。-a オプション付きで呼び出された場合、FTP サーバーは /etc/ftpd/ftpaccess ファイルだけを読み取ります。ftpaccess を使用する場合、すべてのユーザーは FTP サーバーへのアクセスを許可されたクラスのメンバーである必要があります。特定のクラスにのみ適用される ftpaccess 指令を複数指定することができます。

  詳細は、ftpusers(4)、ftphosts(4)、ftpaccess(4) のマニュアルページを参照してください。

注 – FTP サーバーのすべての構成ファイルで、# 記号で始まる行はコメントとして扱われます。

▼ FTP サーバークラスの定義方法

ftpaccess を使用する場合、FTP サーバーにログインするには、ユーザーはクラスのメンバーである必要があります。class 指令を ftpaccess に追加するには、特定のホストからアクセスを許可されているユーザーの *class* 名と *typelist* を指定します。

1. スーパーユーザーになるか、同等の役割を引き受けます。

   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2. ftpaccess ファイルに匿名ユーザー、ゲストユーザー、実ユーザーのエントリを追加します。

   class *class typelist addrglob*[*addrglob...*]

   class　FTP ユーザーの定義に使用するキーワード

   class　class キーワードを使用して定義する名前。各ログインは、定義されているクラスのリストと比較される。ログインしたユーザーは、一致した最初のクラスのメンバーとみなされる

   typelist　3 種類のユーザー (anonymous、guest、real) に一致するキーワードからなる、コンマで区切られたリスト

addrglob 展開されたドメイン名または展開された数値アドレス。*addrglob* は、スラッシュ ('/') で始まるファイル名にすることもできる。 `address:netmask` または `address/cidr` という形のアドレス展開を追加できる

次に、展開されたアドレスの例を示す

- 数値の IPv4 アドレス: **`10.1.2.3`**
- 展開されたドメイン名 **`*.provider.com`**
- 展開された数値の IPv4 アドレス **`10.1.2.*`**
- 数値の IPv4 アドレス : ネットマスク **`10.1.2.0:255.255.255.0`**
- 数値の IPv4 アドレス/CIDR **`10.1.2.0/24`**
- 数値の IPv6 アドレス : **`2000::56:789:21ff:fe8f:ba98`**
- 数値の IPv6 アドレス/CIDR: **`2000::56:789:21ff:fe8f:ba98/120`**

例 **28–1** FTP サーバークラスの定義

```
class  local  real,guest,anonymous *.provider.com
class  remote real,guest,anonymous *
```

この例では、`local` クラスを、`*.provider.com` からログインする `real`、`guest`、または `anonymous` のいずれかの種類のユーザーとして定義します。最後の行では、`remote` を、`*.provider.com` 以外からログインするユーザーとして定義します。

▼ ユーザーログインの制限を設定する方法

`ftpaccess` ファイルに設定された指令により、特定のクラスのユーザーが同時にログインできる数を制限できます。各ログインの制限には、クラス名、UUCP スタイルの曜日リスト、制限を超過した場合に表示するメッセージファイルが含まれます。

ユーザーログインの制限を設定するには、次の手順を実行します。

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 次のエントリを `ftpaccess` ファイルに追加します。

`limit` *class n times* [*message-file*]

`limit` 定義されたクラスの特定時刻の同時ログイン数を、指定したユーザー数に制限するキーワード

class `class` キーワードを使用して定義する名前。各ログインは、定義されているクラスのリストと比較される。ログインしたユーザーは、一致した最初のクラスのメンバーとみなされる

n ユーザー数

times クラスが接続可能な曜日と 1 日の時間帯。任意の曜日を指定する場合は Any を指定する

message-file ユーザーがアクセスを拒否された場合に表示されるメッセージファイル

例 28–2 ユーザーログインの制限の設定

```
limit   anon     50  Wk0800-1800        /etc/ftpd/ftpmsg.deny
limit   anon    100  Any                /etc/ftpd/ftpmsg.deny
limit   guest   100  Any                /etc/ftpd/ftpmsg.deny
```

前述の例の最初の行では、毎週勤務時間中のクラス anon のユーザーの同時ログイン数が 50 に制限されています。2 行目では、勤務時間外の anon のユーザーの同時ログイン数を 100 に制限しています。最後の行では、常時 guest ユーザーの同時ログイン数が 100 に制限されています。日時パラメータの指定方法の詳細は、ftpaccess(4) のマニュアルページを参照してください。

前述の例では、そのほかに、指定したログイン制限数に達した場合に /etc/ftpd/ftpmsg.deny ファイルの内容が返されることを示しています。この場合、ftpmsg.deny は存在するものと仮定しています。/usr/sbin/ftpcount コマンドを使用して、特定の時刻にログインしている各クラスのユーザーの数とログイン制限を表示する方法については、ftpcount(1) のマニュアルページを参照してください。

ユーザーは、その時刻の指定ログイン制限数に達していなければ、FTP サーバーへのログインを許可されます。匿名ユーザーは、ユーザー ftp としてログインします。実ユーザーは、自分自身としてログインします。ゲストユーザーは、アクセス特権を制限する chroot 環境を持つ実ユーザーとしてログインします。

/usr/sbin/ftpwho コマンドを使用して、FTP サーバーにログインするユーザーの識別情報を検査する方法については、ftpwho(1) のマニュアルページを参照してください。

▼ 無効なログインの試行回数を制御する方法

必要な情報を誤入力するなどの理由で FTP サーバーへのログインが失敗すると、通常はログインが繰り返されます。ユーザーは、特定回数連続してログインを試行できます。その回数を超えるとメッセージが syslog ファイルに記録されます。その時点でユーザーとの接続は切断されます。ログイン失敗時の試行回数を制限するには、次の手順を実行します。

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 次のエントリを ftpaccess ファイルに追加します。

```
loginfails n
```

loginfails　FTP 接続が切断されるまでのログインに失敗できる回数を割り当てるキーワード

n　ログインに失敗できる回数

例 28–3 無効なログイン試行回数の制御

```
loginfails 10
```

この例では、ユーザーがログイン試行に 10 回失敗すると FTP サーバーから接続を切断されることを示します。

▼ 特定のユーザーの FTP サーバーへのアクセスを拒否する方法

/etc/ftpd/ftpusers には、FTP サーバーへのログインを拒否するユーザーの名前が列挙されています。ログインが試行されると、FTP サーバーは /etc/ftpd/ftpusers ファイルを検査して、そのユーザーからのアクセスを拒否するかどうかを判定します。ユーザーの名前がそのファイルにない場合は、次に /etc/ftpusers ファイルを検査します。

/etc/ftpusers の中にユーザー名に一致するものがあった場合、使用を差し控えるべきファイルで一致するユーザー名が見つかったことを示す syslogd メッセージが書き込まれます。また、このメッセージでは、/etc/ftpusers の代わりに /etc/ftpd/ftpusers を使用することを推薦します。

注 – /etc/ftpusers ファイルのサポートは、本リリースでは推奨されていません。FTP サーバーをインストールするときに /etc/ftpusers ファイルがすでに存在している場合は、/etc/ftpd/ftpusers に移動されます。

詳細は、syslogd(1M)、in.ftpd(1M)、ftpusers(4) のマニュアルページを参照してください。

1. スーパーユーザーになるか、同等の役割を引き受けます。

   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2. FTP サーバーへのログインを拒否するユーザーのエントリを /etc/ftpd/ftpusers に追加します。

例 28–4 FTP サーバーへのアクセスを拒否する

```
root
daemon
bin
sys
adm
lp
uccp
nuucp
listen
nobody
noaccess
nobody4
```

この例では、ftpusers ファイルの通常のエントリが列挙されています。ユーザー名は /etc/passwd ファイルのエントリに一致します。通常このリストには、root、その他の管理に使用するユーザー、システムアプリケーションを示すユーザーが含まれます。

root エントリは、セキュリティー手段の 1 つとして ftpusers に追加されています。デフォルトのセキュリティーポリシーでは、root のリモートログインを拒否します。また、/etc/default/loginfile ファイルの CONSOLE エントリとして設定されているデフォルト値も同じポリシーに従っています。login(1) のマニュアルページを参照してください。

▼ デフォルト FTP サーバーへのアクセスを制限する方法

これまでに説明した制御方法以外に、ftpaccess ファイルに明示的に文を追加して FTP サーバーへのアクセスを制限することができます。

1. スーパーユーザーになるか、同等の役割を引き受けます。

   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 次のエントリを ftpaccess ファイルに追加します。

a. デフォルトでは、すべてのユーザーはデフォルト (非仮想) FTP サーバーへのアクセスを許可されています。特定のユーザー (anonymous 以外) のアクセスを拒否するには、次のエントリを追加します。

```
defaultserver deny username [username...]
```

`defaultserver`	アクセスの拒否または許可を設定する非仮想サーバーの識別に使用するキーワード
username	`defaultserver` へのアクセスを制限するユーザーのログイン名。

b. deny 行に列挙されていないユーザーのアクセスを許可するには、次の行を追加します。

```
defaultserver allow username [username...]
```

c. 匿名ユーザーのアクセスを拒否するには、次のエントリを追加します。

```
defaultserver private
```

例 28–5 デフォルト FTP サーバーへのアクセスの制限

```
defaultserver deny *
defaultserver allow username
```

この例では、FTP サーバーは、anon ユーザーと `allow` 行に列挙されているユーザー以外のユーザーのアクセスをすべて拒否するように設定されています。

また、`ftphosts` ファイルを使用して、複数のホストからの特定のログインアカウントのアクセスを拒否することができます。詳細は、`ftphosts(4)` のマニュアルページを参照してください。

FTP サーバーのログインの設定

FTP サーバーにアクセスするには、まずログインする必要があります。FTP サーバーは、「実ユーザー」、「ゲストユーザー」、「匿名ユーザー」の 3 種類のユーザーログインアカウントをサポートします。

- 実ユーザーには、FTP サーバーが動作するシステム上の端末セッションの確立を許可するアカウントがあります。ディレクトリとファイルのアクセス権の制約は受けますが、実ユーザーはディスク構造全体を参照できます。
- ゲストユーザーも、FTP サーバーにログインするためのアカウントを必要とします。各ゲストアカウントは、ユーザー名とパスワードを使用して設定されます。端末セッションを確立できないように、ゲストには有効なログインシェルは

割り当てません。ログイン時に、FTP サーバーは chroot(2) 操作を実行して、ゲストユーザーが参照できるサーバーのディスク構造を制限します。

注 – FTP サーバーへのアクセスを許可できるように、実ユーザーとゲストユーザーのログインシェルを /etc/shells ファイルに列挙する必要があります。

- 匿名ユーザーは、ftp または anonymous をユーザー名として使って FTP サーバーにログインします。規約では、匿名ユーザーはパスワードの代わりに電子メールアドレスを入力します。

  ログイン時に、FTP サーバーは chroot(2) 操作を実行して、匿名ユーザーが参照できるサーバーのディスク構造を制限します。ゲストユーザーにはそれぞれ独立した領域を作成できますが、匿名ユーザーは全員が単一のファイル領域を共有します。

  実ユーザーとゲストユーザーは、個別のアカウントとパスワードを使用してログインしますが、本人以外はその存在を知りません。匿名ユーザーは、だれでも知っているアカウントでログインします。可能性として、このアカウントはだれでも使用できます。大規模なファイル配布システムのほとんどは匿名アカウントを使用して作成します。

▼ 実 FTP ユーザーの設定方法

実ユーザーの FTP サーバーへのアクセスを有効にするには、次の手順を実行します。

1. ユーザーに、端末セッションの確立に使用可能なユーザー名とパスワードで設定されたアカウントがあることを確認します。

   詳細は、『Solaris のシステム管理 (基本編)』の第 4 章「ユーザーアカウントとグループの管理 (概要)」を参照してください。

2. 実ユーザーが ftpaccess ファイルのクラスのメンバーであることを確認します。

   ftpaccess ファイルに定義されたユーザークラスについては、625 ページの「FTP サーバークラスの定義方法」を参照してください。

3. ユーザーのログインシェルが /etc/shells ファイルに列挙されていることを確認します。

▼ ゲスト FTP ユーザーの設定方法

ftpconfig スクリプトを使用して、すべての必要なシステムファイルをホームディレクトリにコピーします。ゲストユーザーとゲストのホームディレクトリがすでに存在する場合は、ftpconfig スクリプトは現在のシステムファイルでホームディレクトリ下の領域を更新します。

詳細は、ftpconfig(1M) のマニュアルページを参照してください。

注 – 匿名ユーザー用のユーザー名は anonymous か ftp ですが、FTP ゲスト用のユーザー名は固定されていません。実ユーザー名として使用できる名前であれば、どのような名前でも選択できます。

ゲストユーザーの FTP サーバーへのアクセスを有効にするには、次の手順を実行します。

1 useradd スクリプトを使用して、ログインシェルが /bin/true、ホームディレクトリが */root-dir/./home-dir* であるようなゲストユーザーアカウントを作成します。

詳細は、useradd(1M) のマニュアルページと『Solaris のシステム管理 (基本編)』の第 4 章「ユーザーアカウントとグループの管理 (概要)」を参照してください。

注 – この手順では、**/home/guests/./guest1** は guest1 というユーザーのホームディレクトリの名前として使用されます。

```
# /usr/sbin/useradd -m -c "Guest FTP" -d \
  /home/guests/./guest1 -s /bin/true guest1
```

2 ゲストアカウントにパスワードを割り当てます。

3 guestuser エントリを ftpaccess ファイルに追加します。

```
guestuser guest1
```

注 – さらに、ftpaccess ファイルで guestgroup 機能を使用して、複数のゲストユーザーを指定することができます。ftpaccess ファイルで guest-root 機能を使用すると、ゲストユーザーのホームディレクトリパスで /./ を指定する必要がなくなります。

---

4 ftpaccess ファイルで、ゲストユーザーが class のメンバーであることを確認します。詳細は、**625 ページの「FTP サーバークラスの定義方法」**を参照してください。

5 ftpconfig スクリプトを使用して、chroot 領域の必要なファイルを作成します。

```
/usr/sbin/ftpconfig -d /home/guests
```

6 /bin/true が /etc/shells ファイルに列挙されていることを確認します。634 ページの「/etc/shells ファイルの作成方法」を参照してください。

例 28–6 ゲスト FTP サーバーの設定

この例では、FTP 領域は /home/guests ディレクトリに設定されます。

```
# /usr/sbin/ftpconfig -d /home/guests
Updating directory /home/guests
```

▼ 匿名 FTP ユーザーの設定方法

ftpconfig スクリプトは、anonymous アカウントを作成し、ホームディレクトリに必要なファイルを作成します。

詳細は、ftpconfig(1M) のマニュアルページを参照してください。

匿名ユーザーの FTP サーバーへのアクセスを有効にするには、次の手順を実行します。

1 ftpconfig スクリプトを使用して、匿名ユーザーアカウントを作成します。

```
/usr/sbin/ftpconfig anonymous-ftp-directory
```

2 ftpaccess ファイルで、匿名ユーザーが class に割り当てられていることを確認します。
詳細は、625 ページの「FTP サーバークラスの定義方法」を参照してください。

例 28–7 匿名 FTP ユーザーの設定

この例では、FTP 領域は /home/ftp ディレクトリに設定されます。

```
# /usr/sbin/ftpconfig /home/ftp
Creating user ftp
Updating directory /home/ftp
```

▼ /etc/shells ファイルの作成方法

1. スーパーユーザーになるか、同等の役割を引き受けます。
   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。
2. /etc/shells ファイルを作成します。
3. /etc/shells を編集します。各行のシェルにフルパスを追加します。

例 28–8 /etc/shells ファイルの作成

次に、FTP ゲストユーザー用の /bin/true が含まれる /etc/shells ファイルの例を示します。

```
/sbin/sh
/bin/csh
/bin/jsh
/bin/ksh
/bin/remsh
/bin/rksh
/bin/rsh
/bin/sh
/usr/bin/csh
/usr/bin/ksh
/usr/bin/bash
/usr/bin/tcsh
/usr/bin/zsh
/bin/true
```

メッセージファイルのカスタマイズ

FTP サーバーを構成して、特定のイベントに関連するメッセージを FTP クライアントに返すことができます。ユーザーが FTP サーバーにログインするときに表示される開始メッセージを設定することができます。また、ユーザーが別のディレクトリに移動する場合にメッセージを表示することもできます。

メッセージファイルには、プレーンテキストだけでなく、1 つまたは複数のマジッククッキーを設定できます。マジッククッキーは、% (パーセント記号) と、そのあとに続く 1 文字から構成されます。クッキーをメッセージテキストに組み込む場合、メッセージファイルが呼び出された時点でクッキーに関連付けられた情報が画面に表示されます。

たとえば、メッセージテキストにクッキー %L が含まれているとします。

```
Welcome to %L!
```

メッセージが表示されるとき、マジッククッキー %L は、ftpaccess ファイルの hostname 文で定義されたサーバー名で置き換えられます。サポートされているメッセージクッキーの完全なリストは、ftpaccess(4) のマニュアルページを参照してください。

注 – ホスト名が ftpaccess ファイルに定義されていない場合、ローカルマシンのデフォルトホスト名が使用されます。

▼ メッセージファイルのカスタマイズ方法

1. スーパーユーザーになるか、同等の役割を引き受けます。

   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2. メッセージファイルを編集して、適宜マジッククッキーを追加します。

   使用できるクッキーのリストは、ftpaccess(4) のマニュアルページを参照してください。

例 28–9 メッセージファイルのカスタマイズ

次に、マジッククッキーを使用するメッセージファイルの例を示します。

```
Welcome to %L -- local time is %T.

You are number %N out of a maximum of %M.
All transfers are logged.

If your FTP client crashes or hangs shortly after login
please try
using a dash (-) as the first character of your password.
This will
turn off the informational messages that may be confusing
your FTP
client.

Please send any comments to %E.
```

▼ ユーザーに送信するメッセージの作成方法

ユーザーがログインしたあと、システム関連またはアプリケーション関連のメッセージが画面に表示されます。`ftpaccess` ファイルには、関連付けられた `message` 文をトリガするイベントが列挙されています。

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 次のエントリを `ftpaccess` ファイルに追加します。

`message` *message-file* [*when* [*class* ...]]

`message`	ユーザーがログインしたとき、または作業ディレクトリを変更するコマンドを実行したときに表示されるメッセージファイルの指定に使用するキーワード
message-file	表示するメッセージファイルの名前
when	`login` または `cwd=`*dir* と設定されるパラメータ。例を参照すること
class	`class` を指定すると、メッセージの表示を特定のクラスのメンバーに限定できる

例 28–10 ユーザーに送信するメッセージの作成

```
message    /etc/ftpd/Welcome    login   anon guest
message    .message    cwd=*
```

この例では、クラス `anon` または `guest` のユーザーがログインするときに `/etc/ftpd/Welcome` ファイルが表示されることを示します。2 行目では、すべてのユーザーに対して現在の作業ディレクトリにある `.message` ファイルが表示されることを示します。

メッセージファイルは、ゲストユーザーおよび匿名ユーザーの `chroot` ディレクトリからの相対位置に作成します。

▼ README オプションの構成方法

ディレクトリに最初に移動したとき、`README` ファイルを表示することができます。README オプションを構成するには、次のエントリを `ftpaccess` ファイルに追加します。

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 次のエントリを ftpaccess ファイルに追加します。

```
readme message-file [when [class...]]
```

readme	ユーザーがログインするか、作業ディレクトリを変更するときに確認するメッセージファイルの指定に使用されるキーワード。メッセージファイルが存在する場合、ユーザーはその存在を示す通知と、ファイルの更新日付を受け取る
message-file	確認するメッセージファイルの名前
when	login または cwd=*dir* と設定されるパラメータ。例を参照すること
class	class を指定すると、メッセージの表示を特定のクラスのメンバーに限定できる

注 – greeting キーワードと banner キーワードを使用して、ユーザーにメッセージを送信することもできます。詳細は、ftpaccess(4) のマニュアルページを参照してください。

例 **28–11** README オプションの構成

```
readme  README*     login
readme  README*     cwd=*
```

この例では、ログイン時、またはディレクトリ変更時に、README* に一致するファイルをすべて表示することを示します。この例で使用されている設定に基づいたログイン例を次に示します。

```
% ftp earth
Connected to earth.
220 earth FTP server ready.
Name (earth:rimmer): ftp
331 Guest login ok, send your complete e-mail address as password.
Password:
230-
230-Welcome to earth -- local time is Thu Jul 15 16:13:24
1999.
230-
230-You are number 1 out of a maximum of 10.
230-All transfers are logged.
230-
230-If your FTP client crashes or hangs shortly after login
please try
230-using a dash (-) as the first character of your
```

```
password.  This will
230-turn off the informational messages that may be
confusing your FTP
230-client.
230-
230-Please send any comments to ftpadmin@earth.
230-
230 Guest login ok, access restrictions apply.
ftp> cd pub
250-Please read the file README
250-  it was last modified on Thu Jul 15 16:12:25 1999 - 0
days ago
250 CWD command successful.
ftp> get README /tmp/README
200 PORT command successful.
150 Opening ASCII mode data connection for README (0
bytes).
226 ASCII Transfer complete.
ftp> quit
221 Goodbye.
```

FTP サーバー上のファイルへのアクセスの制御

ここで説明する FTP サーバーのアクセス制御は、Solaris リリースで使用できる標準のファイルとディレクトリのアクセス制御を補足するものです。Solaris の標準コマンドを使用して、ファイルへのアクセス、ファイルの変更、またはファイルのアップロードが可能なユーザーを制限します。chmod(1)、chown(1)、chgrp(1) のマニュアルページを参照してください。

▼ ファイルアクセスコマンドの制御方法

ftpaccess ファイル内のアクセス権機能を使用してどの種類のユーザーにどのコマンドの実行を許可するかを指定するには、次の手順を実行します。

1. スーパーユーザーになるか、同等の役割を引き受けます。

   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2. 次のエントリを ftpaccess ファイルに追加します。

   command yes|no *typelist*

   command chmod、delete、overwrite、rename、または umask のいずれか

   yes|no　ユーザーにコマンドの発行を許可または拒否する

typelist anonymous、guest、および real のキーワードを任意に組み合わせてコンマで区切って並べたリスト

例 28–12 ファイルアクセスコマンドの制御方法

次に、FTP サーバー上のファイルアクセス機能に対して設定されているアクセス権の例を示します。

```
chmod no anonymous, guest
delete    no anonymous
overwrite no anonymous
rename    no anonymous
umask     no guest, anonymous
```

この例では、次のことが示されています。

- 匿名ユーザーは、ファイルの削除、上書き、名前変更を行うことができない
- ゲストユーザーと匿名ユーザーは両方とも、アクセスモードの変更、umask のリセットができない

FTP サーバー上のアップロードとダウンロードの制御

FTP サーバー上のディレクトリのアクセス権を設定することにより、FTP サーバーへのアップロードと FTP サーバーからのダウンロードを制御できます。デフォルトでは、匿名ユーザーはアップロードを実行できません。匿名ユーザーにアップロードを許可する場合は、充分な注意が必要です。

▼ FTP サーバーへのアップロードの制御方法

ftpaccess ファイルに指令を追加して、アップロードのアクセス権と、アップロード異常終了時に表示するエラーメッセージを指定します。

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 次のエントリを ftpaccess ファイルに追加します。

ユーザーにファイルのアップロードを許可するには、次のエントリを追加します。

```
upload [absolute|relative] [class=<classname>]... [-] root-dir \
dirglob yes|no owner group mode [dirs|nodirs] [<d_mode>]

path-filter typelist mesg allowed-charset {disallowed regexp...}
```

`upload`	ホームディレクトリ (`chroot()` の引数) として *root-dir* を持つユーザーに適用するキーワード。*root-dir* に “*” を指定して、任意のホームディレクトリを一致させることができます。
`absolute|relative`	*root-dir* ディレクトリパスを、絶対パスとして解釈するか、または現在の `chroot` ディレクトリからの相対パスとして解釈するかを指定するパラメータ。
`class`	任意個数の `class=<classname>` 制限の指定に使用するキーワード。制限が指定された場合、upload 節が有効になるのは、現在のユーザーが指定されたクラスのメンバーである場合に限定されます。
root-dir	ユーザーのルートディレクトリと匿名ユーザーのホームディレクトリ。
dirglob	ディレクトリ名に一致するパターン。アスタリスクを任意の場所に使用することも、単独で使用して任意のディレクトリを表すこともできます。
`yes|no`	FTP サーバーへのアップロードを許可または拒否する変数。
owner	`dirnames` にアップロードされたファイルの所有者。
group	`dirnames` にアップロードされたファイルに関連付けられているグループ。
mode	アップロードされたファイルのアクセス権の指定に使用するパラメータ。デフォルトモード `0440` の場合、匿名アカウントのユーザーはアップロードされたファイルを読み取れません。
`dirs|nodirs`	`dirnames` に列挙されたディレクトリに、ユーザーがサブディレクトリを作成することを許可または拒否するキーワード。
`d_mode`	新しく作成したディレクトリのアクセス権を決定するオプションモード。
`path-filter`	アップロードされたファイルの名前を制御するキーワード。

typelist	anonymous、guest、および real のキーワードを任意に組み合わせてコンマで区切って並べたリスト。
mesg	regexp 条件に一致しない場合に表示されるメッセージファイル。
allowed-charset {disallowed regexp...}	ファイル名で使用できる、または使用できない英数字。

例 28–13 FTP サーバーへのアップロードの制御

```
upload /export/home/ftp /incoming yes ftpadm ftpadmin 0440 nodirs
path-filter anonymous /etc/ftpd/filename.msg ^[-A-Za-z0-9._]*$ ^[.-]
```

この例では、次のことが示されています。

- /export/home/ftp への chroot を使用する FTP ユーザーアカウントは、/incoming ディレクトリにアップロードすることができる。アップロードされたファイルの所有者は、ユーザー ftpadm、グループ ftpadmin である。モードは nodirs キーワード付きで 0440 に設定され、匿名ユーザーによるサブディレクトリの作成を拒否する
- 匿名ユーザーの場合、ファイル名は A-Z、a-z、0-9、.(ドット記号)、-(ダッシュ記号)、_(下線) の任意の並びである。ファイル名を .(ドット記号) または -(ダッシュ記号) で始めることはできない。ファイル名がこの条件を満足しない場合、/etc/ftpd/filename.msg メッセージファイルが FTP 管理者により作成済みであれば、そのファイルが表示される。このメッセージのあとに、FTP サーバーのエラーメッセージが表示される

匿名ユーザーによるアップロードが許可されているディレクトリの所有者とアクセス権は、厳密に制御する必要があります。FTP 管理者は FTP サーバーにアップロードされるすべてのファイルの所有者である必要があります。匿名ユーザーにファイルのアップロードを許可する場合、FTP 管理者を作成する必要があります。ディレクトリの所有者はユーザー ftpadm、グループ ftpadm、アクセス権は 3773 である必要があります。

FTP サーバーにアップロードされるファイルのアクセスモードは 0440 である必要があります。モードを 0440 にすると、匿名アカウントのユーザーはアップロードされたファイルを読み取れません。この制限により、サーバーが第三者によってファイル配布の場所として使用されるのを防ぎます。

アップロードされたファイルを配布可能にするために、FTP 管理者はそれらのファイルを公開ディレクトリに移動することができます。

▼ FTP サーバーに対するダウンロードの制御方法

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 次のエントリを ftpaccess ファイルに追加して、ユーザーがファイルを読み取れないようにします。

```
noretrieve [absolute|relative] [class=classname]... [-] filename ...
```

`noretrieve`	特定のファイル (複数可) の読み取りの拒否に使用するキーワード
`absolute|relative`	*root-dir* ディレクトリパスを、絶対パスとして解釈するか、または現在の `chroot` ディレクトリからの相対パスとして解釈するかを指定するパラメータ。
`class`	`noretrieve` 制限を適用するユーザーの `class=<classname>` の指定に使用するキーワード
filename	ユーザーによる読み取りを拒否するファイルの名前

例 **28–14** FTP サーバーへのダウンロードの制御

```
noretrieve /etc/passwd
```

この例では、すべてのユーザーが `/etc/passwd` の読み取りを拒否されます。

仮想ホスティング

仮想ホスティングにより、FTP サーバーは同一マシン上の複数ドメインをサポートできます。各仮想ホストには、独立した論理インタフェースと IP アドレスが必要です。

FTP サーバーは、「限定」と「完全」の 2 種類の 仮想ホスティングをサポートします。限定仮想ホスティングでは、すべての仮想ホストが同じ構成ファイルを使用します。完全仮想ホスティングでは、各仮想ホストは個別の構成ファイルを使用できます。

注 – デフォルトでは、実ユーザーとゲストユーザーは、仮想ホストへのログインを拒否されます。次に示す ftpaccess 指令を設定すると、デフォルトを上書きできます。

```
To allow access to specific users:
virtual address allow username
To deny access to anonymous users:
virtual address private username
```

詳細は、ftpaccess(4) のマニュアルページを参照してください。

▼ 限定仮想ホスティングを有効にする方法

限定仮想ホスティングでは、仮想 FTP サーバーの部分的なサポートを提供します。限定仮想ホスティングのサポートを有効にするには、仮想ルートディレクトリを指定します。必要であれば、次に示す仮想ホストのパラメータを ftpaccess ファイルに設定することもできます。

- banner
- logfile
- email
- hostname

ftpaccess ファイル内のすべての指令は、すべての仮想サーバーによりグローバルに共有されます。

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 次のエントリを ftpaccess ファイルに追加します。

```
virtual address root|banner|logfile path
virtual address hostname|email string
```

virtual	仮想サーバー機能を有効にするために使用するキーワード
address	仮想サーバーの IP アドレス
root	仮想サーバーのルートディレクトリ
banner	仮想サーバーへの接続が確立したときに表示されるバナーファイル
logfile	仮想サーバーに対するファイル転送の記録
path	仮想サーバー上のディレクトリとファイルの場所の指定に使用する変数

email	メッセージファイルと HELP コマンドで使用される電子メールアドレス
hostname	グリーティングメッセージやステータスコマンドで表示されるホスト名
string	email パラメータまたは hostname パラメータの指定に使用する変数

注 – hostname を仮想サーバーの *address* として使用することは可能ですが、それよりも IPv4 アドレスの使用を強く推奨します。hostname に一致するホストを見つけられるようにするには、FTP 接続を受信するときに DNS が使用可能になっている必要があります。IPv6 ホストの場合は、IPv6 アドレスよりもホスト名を使用します。

例 28–15 ftpaccess ファイルによる限定仮想ホスティングの有効化

```
virtual 10.1.2.3 root    /var/ftp/virtual/ftp-serv
virtual 10.1.2.3 banner  /var/ftp/virtual/ftp-serv/banner.msg
virtual 10.1.2.3 logfile /var/log/ftp/virtual/ftp-serv/xferlog
```

この例では、仮想 FTP サーバー上の root ディレクトリ、banner、logfile の場所を設定します。

例 28–16 コマンド行での限定仮想ホスティングの有効化

ftpaddhost(1M) スクリプトを -l オプション付きで使用して、限定仮想ホストを構成できます。

次の例では、ftpaddhost を -l -b -x オプションとともに実行して、テストバナーと、仮想ルート /var/ftp/virtual/10.1.2.3 の下にあるログファイル /var/ftp/virtual/10.1.2.3/xferlog を使用する限定仮想ホスティングを構成します。

```
# ftpaddhost -l -b -x /var/ftp/virtual/10.1.2.3/xferlog \
/var/ftp/virtual/10.1.2.3
```

▼ 完全仮想ホスティングを有効にする方法

完全仮想ホスティングでは、各仮想ドメインは個別の構成ファイルを使用できます。FTP サーバー上の仮想ホスティングの完全サポートを有効にするには、特定のドメインについて次に示す FTP 構成ファイルを作成または変更します。

- ftpaccess
- ftpusers
- ftpgroups
- ftphosts
- ftpconversions

詳細は、ftpaccess(4)、ftpusers(4)、ftpgroups(4)、ftphosts(4)、ftpconversions(4) のマニュアルページを参照してください。

注 – 構成ファイルの個別のバージョンが見つからない場合は、/etc/ftpd ディレクトリに置かれた構成ファイルのマスターバージョンを使用します。

---

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 次のエントリを /etc/ftpd/ftpservers ファイルに追加します。

address */config-file-dir*

address 仮想サーバーの IP アドレス

config-file-dir 仮想ホスト用にカスタマイズされた構成ファイルが置かれるディレクトリ

注 – hostname を仮想サーバーの *address* として使用することは可能ですが、それよりも IPv4 アドレスの使用を強く推奨します。hostname に一致するホストを見つけられるようにするには、FTP 接続を受信するときに DNS が使用可能になっている必要があります。IPv6 ホストの場合は、IPv6 アドレスよりもホスト名を使用します。

---

3 仮想ホスト用にカスタマイズされた **FTP** サーバー構成ファイルを作成するには、/etc/ftpd ディレクトリにある構成ファイルのマスターバージョンを /config-file-dir ディレクトリにコピーします。

詳細は、ftpservers(4) のマニュアルページを参照してください。

例 28–17 ftpservers ファイルによる完全仮想ホスティングの有効化

```
#
# FTP Server virtual hosting configuration file
#

10.1.2.3 /net/inet/virtual/somedomain/
10.1.2.4 /net/inet/virtual/anotherdomain/
```

この例では、仮想サーバー上の 2 つの異なるドメインの IP アドレスを指定します。

例 28–18 コマンド行での完全仮想ホスティングの有効化

ftpaddhost(1M) スクリプトを -c オプション付きで使用して、完全仮想ホストを構成できます。

次の例では、ftpaddhost を -c -b -x オプションとともに実行して、テストバナーと、仮想ルート /var/ftp/virtual/10.1.2.3 の下にあるログファイル /var/ftp/virtual/10.1.2.3/xferlog を使用する完全仮想ホスティングを構成します。

```
# ftpaddhost -c -b -x /var/ftp/virtual/10.1.2.3/xferlog \
/var/ftp/virtual/10.1.2.3
```

FTP サーバーの自動起動

FTP サーバーを起動するには、次の 3 つの方法があります。

- nowait サーバー。inetd から起動される
- スタンドアロンサーバー。バックグラウンドで実行される
- スタンドアロンサーバー。inittab ファイルからフォアグラウンドで実行される

スタンドアロンサーバーの応答時間は常に可能なかぎり最短であり、FTP サービス専用の大規模サーバー向けです。スタンドアロンサーバーは一切再起動する必要がないので、専用サーバーに適した短い接続待ち時間を実現します。スタンドアロンサーバーは、混雑していない時間帯も含めて、常に動作しており、永久に接続要求を待ちます。

▼ SMF を使用して FTP サーバーを起動する方法

デフォルトでは、SMF サービスは nowait モードで FTP サーバーを起動するように設定されています。サイトで処理する接続が多数になる場合、FTP サーバーをスタンドアロンモードで実行することもできます。その他のコマンド行オプションについては、in.ftpd(1M) のマニュアルページを参照してください。

1. スーパーユーザーになるか、同等の役割を引き受けます。

   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2. FTP サーバーの wait プロパティーを確認します。

   wait=FALSE の行は、サーバーが nowait モードで起動することを示します。

```
# inetadm -l network/ftp
SCOPE    NAME=VALUE
         name="ftp"
         endpoint_type="stream"
         proto="tcp6"
         isrpc=FALSE
         wait=FALSE
```

```
        exec="/usr/sbin/in.ftpd -a"
        user="root"
default  bind_addr=""
default  bind_fail_max=-1
default  bind_fail_interval=-1
default  max_con_rate=-1
default  max_copies=-1
default  con_rate_offline=-1
default  failrate_cnt=40
default  failrate_interval=60
default  inherit_env=TRUE
default  tcp_trace=FALSE
default  tcp_wrappers=FALSE
```

3 FTP サーバーを起動します。

```
# svcadm enable network/ftp
```

▼ FTP サーバーをバックグラウンドで起動する方法

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 FTP サーバーを無効にします。

```
# svcadm disable network/ftp
```

3 スタンドアロン FTP サーバーを起動します。

```
# /usr/sbin/in.ftpd -a -S
```

この行を FTP サーバーの起動スクリプトに追加します。システムの起動スクリプト作成についての詳細は、『Solaris のシステム管理 (基本編)』の「実行制御スクリプトの使用」を参照してください。

▼ FTP サーバーをフォアグラウンドで起動する方法

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 FTP サーバーを無効にします。

```
# svcadm disable network/ftp
```

3 サービスを開始するためのエントリを inittab ファイルに追加します。

次に、/etc/inittab 内の新しいエントリ例を示します。

```
ftpd:3:respawn:/usr/sbin/in.ftpd -a -s
```

4 init を使用し、/etc/inittab をもう一度確認します。

このコマンドにより、FTP サービスが起動されます。

```
# init q
```

FTP サーバーの停止

ftpshut(1M) コマンドは、特定の時刻に FTP サーバーを停止します。

ftpshut を実行する場合、コマンド行オプションでシステム停止時刻を指定するファイルを作成します。この時刻になると、それ以上の新しい接続は受け付けられなくなり、既存の接続は切断されます。この停止時刻の情報に基づいて、サーバーが停止することがユーザーに通知されます。ftpshut により作成されるファイルの場所は、ftpaccess ファイルの shutdown 指令によって指定します。

▼ FTP サーバーの停止方法

ftpshut を実行し、ftpaccess ファイルに shutdown 指令を追加するには、次の手順を実行します。

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 次のエントリを ftpaccess ファイルに追加します。

```
shutdown path
```

shutdown　FTP サーバーの停止時刻が予定されているかどうかを定期的に検査するファイルへの *path* の指定に使用するキーワード

path　ftpshut コマンドが作成したファイルの場所

3 ftpshut コマンドを実行します。

```
ftpshut [ -V ] [ -l min] [ -d min] time [warning-message...]
```

ftpshut　FTP サーバーが停止することをユーザーに通知する手順を提供するコマンド

`-V`	著作権情報とバージョン情報を表示したあと、接続を切断するように指定するオプション
`-l`	FTP サーバーへの新しい接続を拒否する時間の調整に使用されるフラグ
`-d`	FTP サーバーへの既存の接続を切断する時間の調整に使用されるフラグ
`time`	停止時刻として `now` を指定すると即時停止する。未来における停止時刻を指定するには、「*+number*」または「*HHMM*」のどちらかの形式で指定する
`[warning-message...]`	停止通知メッセージ

4 `ftprestart` コマンドを使用して、**FTP** サーバーを停止後に再起動します。

詳細は、`ftpshut(1M)`、`ftpaccess(4)`、`ftprestart(1M)` のマニュアルページを参照してください。

FTP サーバーのデバッグ

ここでは、FTP サーバーに関する問題をデバッグする方法についていくつか説明します。

▼ syslogd 内の FTP サーバーのメッセージを検査する方法

FTP サーバーは、`/etc/syslog.conf` ファイルでデーモンメッセージの出力先として指定された場所に、デバッグに役立つメッセージを書き込みます。FTP サーバーに問題が発生した場合、まずこのファイルで関連するメッセージを検査します。

FTP サーバーメッセージは、機能デーモンにより制御されます。FTP サーバーから `/var/adm/message` にメッセージを送信し、`syslogd` にその構成ファイルを再読み取りさせるには、次の手順を実行します。

1 次のようなエントリを `/etc/syslog.conf` ファイルに追加します。

```
daemon.info /var/adm/message
```

2 `syslogd` にシグナルを送信して、その構成を再読み取りさせます。

```
# svcadm refresh system/system-log
```

この操作により、FTP サーバーから有益な情報を含むメッセージが `/var/adm/messages` に書き込まれます。

▼ greeting text を使用して ftpaccess を検査する方法

greeting text 機能を使用して、適切な内容の ftpaccess ファイルが使用されていることを検査するには、次の手順を実行します。

1. 次の指令を **ftpaccess** ファイルに追加します。

   ```
   greeting text message
   ```

2. **FTP** サーバーに接続します。

3. メッセージが表示されない場合、次の手順を実行します。

   a. ftpaccess ファイルが正しい場所に置かれていることを確認します。strings(1) コマンドを使用して、**FTP** サーバーバイナリからファイルの場所を取得します。

      ```
      # strings /usr/sbin/in.ftpd | grep "^/.*ftpaccess"
      ```

   b. 仮想ホスティングが構成されているかどうか ftpservers ファイルを検査します。

      詳細は、ftpaccess(4)、ftpservers(4)、strings(1)、syslog.conf(4)、pgrep(1) のマニュアルページを参照してください。

▼ FTP ユーザーにより実行されたコマンドの検査

FTP ユーザーがどのコマンドを実行したかを確認するには、ftpaccess の log commands ロギングを使用します。

1. 次の指令を ftpaccess ファイルに追加し、*typelist* で指定されたユーザーによるコマンドを個別に記録します。

   ```
   log commands typelist
   ```

2. /etc/syslog.conf で指定した場所に書き込まれたメッセージを検査します。

多忙なサイトにおける構成についてのヒント

次に、多忙な FTP サイトのパフォーマンスを向上させる上でのヒントをいくつか挙げます。

1. 常に多数の同時接続をサポートしているサイトでは、FTP サーバーをスタンドアロンモードで稼働させることをお薦めします。646 ページの「FTP サーバーの自動起動」を参照してください。

2. vmstat などのシステムユーティリティーを使用し、FTP サーバーのホストに当たるシステムを監視してください。システムで低リソース状態が続く場合は、許可する同時接続数を制限してください (626 ページの「ユーザーログインの制限を設定する方法」を参照)。システム監視の詳細は、『Solaris のシステム管理 (上級編)』の第 13 章「システムパフォーマンスの監視 (手順)」を参照してください。
3. 接続制限を行う場合は、ftpaccess ファイル内の limit-time 機能と timeout idle 機能を使用し、ユーザーが接続を独占しないように設定してください。接続制限を行わない場合は、in.ftpd に -Q オプションを指定してください。
4. /var/adm/wtmpx 内の ftp ログイン / ログアウト記録が不要な場合は、in.ftpd に -W オプションを指定してください。
5. FTP サーバーのホストに当たるシステムの負荷を減らすには、ftpaccess ファイル内の recvbuf 機能と sendbuf 機能を使用して転送バッファーサイズを増やしてください。バッファーサイズを大きくした場合は、必要に応じて、ftpaccess ファイル内の timeout data 機能を使用し、データ処理のタイムアウト値を大きくしてください。
6. FTP サーバーは、さまざまなデータベース (hosts、passwd、group、group、services など) から読み取りを行います。検索速度が遅いと、FTP サーバーのログインで大きな遅延が発生する可能性があります。nsswitch.conf で files ソースが初めに来るように設定すると、検索時間を最短に抑えることができます。詳細は、nsswitch.conf(4) のマニュアルページを参照してください。
7. デフォルトでは、FTP サーバーはリモートホストの名前を検索します。この検索に時間がかかり、ログインに相当な遅延が発生する可能性があります。この検索は、ftpaccess ファイル内の rhostlookup 機能を使用して抑止できます。しかし、リモートホストの名前検索が行われないと、ftpaccess ファイル内のほかの機能が使用される場合や、ftphosts ファイル内のエントリの照合が行われる場合には IP アドレスの突き合わせしか行われないことに注意してください。リモートホストの IP アドレスはメッセージ内でも使用されるほか、%R マジッククッキーの代用としても使用されます。詳細は、ftpaccess(4) のマニュアルページに挙げられた rhostlookup 機能の説明を参照してください。
8. FTP サーバーへのログイン時には、割り当て情報の取得も大きな遅延を引き起こす可能性があります。このため、割り当てマジッククッキーを使用する場合は、ftpaccess ファイル内で quota-info 機能だけを使用するようにしてください。割り当てマジッククッキーの一覧は、ftpaccess(4) のマニュアルページを参照してください。

◆ ◆ ◆ 第 29 章

29 リモートシステムへのアクセス (手順)

本章では、リモートシステムにログインし、リモートシステムのファイルを操作するために必要なすべての作業について説明します。この章で説明する手順は次のとおりです。

リモートシステムへのアクセス (作業マップ)

この章では、次の表に示すリモートシステムにログインし、ファイルをコピーするための作業について説明します。

表 29–1 作業マップ: リモートシステムへのアクセス

作業	説明	参照先
リモートシステムにログインする (rlogin)	■ .rhosts ファイルを削除する ■ rlogin コマンドを使用してリモートシステムにアクセスする	659 ページの「.rhosts ファイルを検索して削除する方法」 659 ページの「リモートシステムが動作中かどうかを調べる方法」 660 ページの「リモートシステムにログインしているユーザーを検索する方法」 661 ページの「リモートシステムにログインする方法 (rlogin)」 662 ページの「リモートシステムからログアウトする方法 (exit)」

表 29–1 作業マップ: リモートシステムへのアクセス (続き)

作業	説明	参照先
リモートシステムにログインする (ftp)	■ ftp 接続のオープンとクローズを行う ■ リモートファイルから、およびリモートシステムに、ファイルをコピーする	663 ページの「ftp によりリモートシステムへ接続する方法」 664 ページの「リモートシステムとの ftp 接続を終了する方法」 665 ページの「リモートシステムからファイルをコピーする方法 (ftp)」 667 ページの「ファイルをリモートシステムにコピーする方法 (ftp)」
rcp を使用してリモートファイルをコピーする	rcp コマンドを使用して、リモートシステムから、およびリモートシステムに、ファイルをコピーする	671 ページの「ローカルシステムとリモートシステム間でファイルをコピーする方法 (rcp)」

リモートシステムへのログイン (rlogin)

rlogin コマンドを使用すると、リモートシステムにログインできます。ログインした後は、リモートファイルシステム内で移動し、その内容を (リモートシステムによる承認にしたがって) 操作したり、ファイルをコピーしたり、リモートコマンドを実行したりできます。

ログイン先のシステムがリモートドメインに属している場合は、必ずシステム名にドメイン名を追加してください。次の例では、SOLAR はリモートドメイン名です。

```
rlogin pluto.SOLAR
```

また、Control-d と入力すると、リモートログイン処理をいつでも中断できます。

リモートログイン (rlogin) の認証

rlogin 処理の認証 (ログインするユーザーの確認処理) は、リモートシステムまたはネットワーク環境で実行されます。

この 2 つの認証形式の主な違いは、要求される対話操作と、認証の確立方法にあります。リモートシステムがユーザーを認証しようとする場合に、/etc/hosts.equiv または .rhosts ファイルを設定していなければ、パスワードの入力を促すプロンプトが表示されます。ネットワークがユーザーを認証しようとする場合は、ユーザーはすでにネットワークに認識されているので、パスワードプロンプトは表示されません。

リモートシステムがユーザーを認証しようとする場合は、特に次のいずれかに該当する場合は、リモートシステム上のローカルファイル内の情報を使用した認証が行われます。

- ユーザーが属するシステム名とユーザー名がリモートシステム上の /etc/hosts.equiv ファイルに列挙されている
- システム名とユーザー名が、リモートユーザーのホームディレクトリの下にある .rhosts ファイルに入っている場合

ネットワークによる認証は、次のどちらかの場合に利用されます。

- ローカルネットワーク情報サービスとオートマウンタを使用して設定された「信頼できるネットワーク環境」がある場合
- リモートシステムの /etc/nsswitch.conf ファイルが指定するネットワーク情報サービスがユーザーに関する情報を持っている場合

注 – 通常は、ネットワークによる認証がシステムによる認証より優先されます。

/etc/hosts.equiv ファイル

/etc/hosts.equiv ファイルには、リモートシステムの「信頼されるホスト」が 1 行に 1 つずつ入っています。ユーザーがこのファイルに含まれるホストから (rlogin を使用して) リモートログインしようとする場合、リモートシステムがそのユーザーのパスワードエントリにアクセスできれば、ユーザーはパスワードを入力しなくてもログインできます。

典型的な hosts.equiv ファイルの構造は次のとおりです。

```
host1
host2 user_a
+@group1
-@group2
```

上記の host1 のエントリのように、hosts.equiv にホストに対して 1 つのエントリが記述されていれば、そのホストは信頼されているため、そのマシン上のユーザーも信頼できることを意味します。

この例の第 2 のエントリのようにユーザー名も含まれていると、その指定されたユーザーがアクセスしようとする場合にのみ、そのホストが信頼されます。

グループ名の先頭にプラス記号 (+) が付いている場合は、そのネットグループ内のすべてのマシンが信頼されていることを意味します。

グループ名の先頭にマイナス記号 (–) が付いている場合は、そのネットグループ内には信頼できるマシンがないことを意味します。

/etc/hosts.equiv ファイルを使用する場合のセキュリティーの問題

/etc/hosts.equiv ファイルにはセキュリティー上の問題があります。/etc/hosts.equiv ファイルをシステム上で管理する場合は、ネットワーク内で信頼されるホストのみを含めるようにしてください。別のネットワークに所属するホストまたは公共領域にあるマシンを追加しないでください。たとえば、端末室に置かれているホストは追加しないでください。

信頼できないホストを使用すると、重大なセキュリティー上の問題が発生する可能性があります。/etc/hosts.equiv ファイルを正しく構成されたファイルと置き換えるか、ファイルを削除してください。

/etc/hosts.equiv ファイルに + のみの 1 行しか入っていない場合は、認識されているすべてのホストが信頼されることを示します。

.rhosts ファイル

.rhosts ファイルは、/etc/hosts.equiv ファイルに対応するユーザー用のファイルです。このファイルには、通常、ホストとユーザーの組み合わせのリストが入っています。このファイルにホストとユーザーの組み合わせが含まれている場合、そのユーザーには、パスワードを入力しなくても、そのホストからリモートログインする許可が与えられます。

.rhosts ファイルはユーザーのホームディレクトリの一番上のレベルに置かれていなければなりません。サブディレクトリに置かれている .rhosts ファイルは参照されません。

ユーザーは、各自のホームディレクトリ内で .rhosts ファイルを作成できます。.rhosts ファイルを使用することによって、/etc/hosts.equiv ファイルを使用しなくても、異なるシステムのユーザー自身のアカウント間で信頼できるアクセスを行うことができます。

.rhosts ファイルを使用する場合のセキュリティーの問題

.rhosts ファイルにはセキュリティー上、重大な問題があります。/etc/hosts.equiv ファイルはシステム管理者の制御下にあり、効率よく管理できますが、だれでも .rhosts ファイルを作成して、システム管理者が知らないうちに自分が選んだユーザーにアクセス権を与えることができます。

すべてのユーザーのホームディレクトリが 1 台のサーバー上にあって、特定のユーザーだけがそのサーバーに対してスーパーユーザーのアクセス権を持っている場合、ユーザーが .rhosts ファイルを使用できないようにするためには、スーパーユーザーとして、空の .rhosts ファイルを各ユーザーのホームディレクトリに作成します。次に、このファイルのアクセス権を 000 に変更します。こうしておけば、スーパーユーザーでも、そのファイルを変更することが難しくなります。これにより、ユーザーが .rhosts を無責任に使用することによって生じるセ

キュリティー問題を防ぐことができます。ただし、ユーザーが自分のホームディレクトリへの実効パスを変更できる場合、この方法は何の解決にもなりません。

.rhosts ファイルを確実に管理する唯一の方法は、それを完全に使用できないようにすることです。詳細は、659 ページの「.rhosts ファイルを検索して削除する方法」を参照してください。システム管理者は、システムを頻繁にチェックして、このポリシーに対する違反を調べることができます。このポリシーに対する例外は、root アカウントです。ネットワークのバックアップや他のリモートサービスを実行するには、.rhosts ファイルが必要な場合があります。

リモートログインのリンク

システムが正しく構成されていれば、リモートログインをリンクできます。たとえば、earth 上のユーザーが jupiter にログインし、そこから pluto にログインします。

このユーザーは jupiter からログアウトして pluto に直接ログインすることもできますが、このリンク方法の方が便利です。

パスワードを入力せずにリモートログインをリンクするには、/etc/hosts.equiv または .rhosts ファイルを正しく設定しておく必要があります。

直接リモートログインと間接リモートログイン

rlogin コマンドにより、リモートシステムに直接的または間接的にログインできます。

直接リモートログインは、デフォルトユーザー名、すなわち現在ローカルシステムにログインしている個人のユーザー名を使用します。これは、最も一般的なリモートログイン形式です。

間接リモートログインは、リモートログイン処理中に別のユーザー名を入力することによって行います。これは、一時的に借りているワークステーションから行うタイプのリモートログインです。たとえば、ユーザーが同僚のオフィスにいるときに自分のホームディレクトリに置かれているファイルを確認する必要がある場合、同僚のシステムからリモートで自分のシステムにログインすることができます。この場合、自分のユーザー名を入力して間接リモートログインを実行することになります。

次の表は、直接ログインや間接ログインと認証方式の依存関係を示しています。

表 29–2 ログイン方式と認証方式 (rlogin) の依存関係

ログイン方式	ユーザー名の提供	認証	パスワード
直接	システム	ネットワーク	なし
		システム	必要
間接	ユーザー	ネットワーク	なし
		システム	必要

リモートログイン後の処理

リモートシステムにログインするときに、rlogin コマンドはホームディレクトリを見つけようとします。ホームディレクトリが見つからなければ、リモートシステムのルートディレクトリ (/) が割り当てられます。次に例を示します。

```
Unable to find home directory, logging in with /
```

ただし、rlogin コマンドがホームディレクトリを見つけると、.cshrc ファイルと .login ファイルを生成します。したがって、リモートログイン後は、プロンプトが標準ログインプロンプトになり、現在のディレクトリはローカルにログインするときと同じになります。

たとえば、通常のプロンプトにシステム名と作業用ディレクトリが表示される場合と、ログイン時の作業用ディレクトリがホームディレクトリの場合、ログインプロンプトは次のようになります。

```
earth(/home/smith):
```

リモートシステムにログインすると、同じようなプロンプトが表示され、rlogin コマンドをどのディレクトリから入力したかに関係なく、作業用ディレクトリがホームディレクトリになります。

```
earth(/home/smith): rlogin pluto
.
.
.
pluto(/home/smith):
```

唯一の違いは、プロンプトの先頭にローカルシステムではなくリモートシステムの名前が表示されることです。リモートファイルシステムは、ホームディレクトリと並んで存在します。

/home ディレクトリに移動して ls を実行すると、次のように表示されます。

```
earth(home/smith): cd ..
earth(/home): ls
smith  jones
```

▼ .rhosts ファイルを検索して削除する方法

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2 find(1) コマンドを使用し、.rhosts ファイルを検索して削除します。

```
# find home-directories -name .rhosts -print -exec rm {} \;
```

home-directories ユーザーのホームディレクトリがあるディレクトリへのパス。複数のパスを指定すると、複数のホームディレクトリを一度に検索できる

`-name .rhosts` ここでは .rhosts を指定する

`-print` 現在のパス名を出力する

`-exec rm {} \;` 指定したファイル名に一致するファイルすべてに、rm コマンドを適用するように find コマンドに伝える

find コマンドは、指定したディレクトリから始めて .rhosts というファイルを検索します。ファイルが見つかると、find はファイルのパスを画面上に表示し、ファイルを削除します。

例 29–1 .rhosts ファイルを検索して削除する

次の例では、/export/home ディレクトリ内で、すべてのユーザーのホームディレクトリ内の .rhosts ファイルを検索し削除します。

```
# find /export/home -name .rhosts -print | xargs -i -t rm {} \;
```

リモートシステムが動作中かどうかを調べる方法

ping コマンドを使用して、リモートシステムが動作中かどうかを調べます。

```
$ ping system-name | ip-address
```

system-name リモートシステム名

ip-address リモートシステムの IP アドレス

ping コマンドは、次の 3 つのメッセージのどれかを返します。

状態メッセージ	意味
system-name is alive	このシステムにはネットワーク経由でアクセスできる
ping: unktawn host *system-name*	未知のシステム名
ping: no answer from *system-name*	システムは認識されるが、現在は動作していない

ping を実行した対象のシステムが別のドメイン内にある場合は、出力メッセージにルーティング情報も含まれることがありますが、これは無視してかまいません。

ping コマンドのタイムアウトは 20 秒です。つまり、20 秒以内に応答がなければ、第 3 のメッセージを返します。*time-out* 値を秒単位で入力すると、ping の待ち時間を増減させることができます。

```
$ ping system-name | ip-address time-out
```

詳細は、ping(1M) のマニュアルページを参照してください。

リモートシステムにログインしているユーザーを検索する方法

rusers(1) コマンドを使用して、リモートシステムにログインしているユーザーを検索します。

```
$ rusers [-l] remote-system-name
```

rusers　(オプションなし) システム名と、root など現在ログインしているユーザー名を表示する

-l　各ユーザーの詳細な情報を表示する。ユーザーのログインウィンドウ、ログイン日時、ログインしている時間、ユーザーのログイン元のリモートシステム名など

例 29–2　リモートシステムにログインしているユーザーを検索する

次の例は、rusers の短い形式の出力を示しています。

```
$ rusers pluto
pluto    smith  jones
```

次の例では、rusers の長い形式の出力は、2 人のユーザーがリモートシステム starbug にログインしていることを示します。第 1 のユーザーは 9 月 10 日にシステムコンソールからログインし、ログイン時間は 137 時間 15 分でした。第 2 のユーザーはリモートシステム mars から 9 月 14 日にログインしました。

例 29–2 リモートシステムにログインしているユーザーを検索する (続き)

```
$rusers -l starbug
root         starbug:console           Sep 10 16:13  137:15
rimmer       starbug:pts/0             Sep 14 14:37         (mars)
```

リモートシステムにログインする方法 (rlogin)

rlogin(1) コマンドを使用してリモートシステムにログインします。

```
$ rlogin [-l user-name] system-name
```

rlogin — (オプションなし) 現在のユーザー名を使用して、リモートシステムに直接ログインする

-l *user-name* — ユーザー名を入力して、リモートシステムに間接的にログインする

ネットワークがユーザーを認証しようとする場合には、パスワードを求めるプロンプトは表示されません。リモートシステムがユーザーを認証しようとする場合は、パスワードの入力を求めるプロンプトが表示されます。

処理が成功すると、rlogin コマンドは、そのシステムへの前回のリモートログイン、リモートシステム上で動作中のオペレーティングシステムのバージョン、ホームディレクトリに未処理のメールがあるかどうかに関して、簡潔な情報を表示します。

例 29–3 リモートシステムにログインする (rlogin)

次の例は、pluto へ直接リモートログインした出力結果を示しています。このユーザーはネットワークから認証されています。

```
$ rlogin starbug
Last login: Mon Jul 12 09:28:39 from venus
Sun Microsystems Inc.   SunOS 5.8       February 2000
starbug:
```

次の例は、pluto へ間接リモートログインした出力結果を示しています。この場合、ユーザーはリモートシステムから認証されています。

```
$ rlogin -l smith pluto
password: user-password
Last login: Mon Jul 12 11:51:58 from venus
Sun Microsystems Inc.   SunOS 5.8       February 2000
starbug:
```

リモートシステムからログアウトする方法 (exit)

exit(1) コマンドを使用して、リモートシステムからログアウトします。

```
$ exit
```

例 29–4 リモートシステムからログアウトする (exit)

次の例は、ユーザー smith がシステム pluto からログアウトする様子を示しています。

```
$ exit
pluto% logout
Connection closed.
earth%
```

リモートシステムへのログイン (ftp)

ftp コマンドは、インターネットのファイルトランスポートプロトコルへのユーザーインタフェースを提供します。このユーザーインタフェースはコマンドインタプリタと呼ばれ、リモートシステムにログインし、そのファイルシステムについて様々な処理を実行できるようにします。基本操作については次の表を参照してください。

rlogin および rcp と比較した場合に ftp が優れている最大のポイントは、ftp はリモートシステムでの UNIX の実行を要求しないことです。ただし、リモートシステムを TCP/IP 通信ができるように構成する必要はあります。逆に、rlogin の優れている点は、ftp よりも豊富なファイル操作コマンドを使用できることです。

リモートログインの認証 (ftp)

ftp によるリモートログインの認証は、次のいずれかの方法により確立できます。

- パスワードエントリをリモートシステムの /etc/passwd ファイルか、同等のネットワーク情報サービスマップまたはテーブルに追加する
- リモートシステム上で匿名 ftp アカウントを確立する

重要な ftp コマンド

表 29–3　重要な ftp コマンド

コマンド	説明
ftp	ftp コマンドインタプリタにアクセスする
ftp*remote-system*	リモートシステムへの ftp 接続を確立する。詳細は、663 ページの「ftp によりリモートシステムへ接続する方法」を参照
open	コマンドインタプリタからリモートシステムにログインする
close	リモートシステムからログアウトしてコマンドインタプリタに戻る
bye	ftp コマンドインタプリタを終了する
help	すべての ftp コマンドを表示するか、コマンド名が指定されている場合は、コマンドの機能に関する簡単な説明を表示する
reset	リモートの ftp サーバーとコマンド応答シーケンスの同期をとり直す
ls	リモートの作業用ディレクトリの内容を表示する
pwd	リモートの作業用ディレクトリ名を表示する
cd	リモートの作業用ディレクトリを変更する
lcd	ローカルの作業用ディレクトリを変更する
mkdir	リモートシステム上でディレクトリを作成する
rmdir	リモートシステム上でディレクトリを削除する
get、mget	リモートの作業用ディレクトリからローカルの作業用ディレクトリに 1 つ以上のファイルをコピーする
put、mput	ローカルの作業用ディレクトリからリモートの作業用ディレクトリに 1 つ以上のファイルをコピーする
delete、mdelete	リモートの作業用ディレクトリから 1 つ以上のファイルを削除する

詳細は、ftp(1) のマニュアルページを参照してください。

▼ ftp によりリモートシステムへ接続する方法

1　ftp 認証を持っていることを確認します。

662 ページの「リモートログインの認証 (ftp)」で説明しているように、ftp 認証を持っている必要があります。

2 ftp コマンドを使用してリモートシステムへ接続します。

```
$ ftp remote-system
```

接続に成功すると、確認メッセージとプロンプトが表示されます。

3 ユーザー名を入力します。

```
Name (remote-system:user-name): user-name
```

4 プロンプトが表示されたら、パスワードを入力します。

```
331 Password required for user-name:
Password: password
```

アクセス中のシステムで匿名 ftp アカウントが設定されている場合は、パスワードとして電子メールアドレスの入力を求めるプロンプトが表示されます。ftp インタフェースがパスワードを受け付けると、確認メッセージと (ftp>) プロンプトを表示します。

これで、help など、ftp インタフェースから提供されるどのコマンドでも使用できます。主なコマンドについては、表 29–3 を参照してください。

例 29–5 ftp によりリモートシステムへ接続する

次の ftp セッションは、リモートシステム pluto 上でユーザー smith によって確立されました。

```
$ ftp pluto
Connected to pluto.
220 pluto FTP server ready.
Name (pluto:smith): smith
331 Password required for smith:
Password: password
230 User smith logged in.
ftp>
```

リモートシステムとの ftp 接続を終了する方法

bye コマンドを使用して、リモートシステムとの ftp 接続を終了します。

```
ftp> bye
221-You have  transferred 0 bytes in 0 files.
221-Total traffic for this sessions was 172 bytes in 0 transfers.
221-Thanks you for using the FTP service on spdev.
221 Goodbye.
```

接続を終了するメッセージに続いて、通常のシェルプロンプトが表示されます。

▼ リモートシステムからファイルをコピーする方法 (ftp)

1. リモートシステムからファイルをコピーする、ローカルシステム上のディレクトリに変更します。

   ```
   $ cd target-directory
   ```

2. ftp により接続します。

   663 ページの「ftp によりリモートシステムへ接続する方法」を参照してください。

3. コピー元ディレクトリに変更します。

   ```
   ftp> cd source-directory
   ```

   システムがオートマウンタを使用している場合、リモートシステムのユーザーのホームディレクトリは、/home の下にユーザーのホームディレクトリと並行して表示されます。

4. コピー元ファイルの読み取り権があることを確認します。

   ```
   ftp> ls -l
   ```

5. 転送タイプを binary に設定します。

   ```
   ftp> binary
   ```

6. ファイルを 1 つコピーするには、get コマンドを使用します。

   ```
   ftp> get filename
   ```

7. 一度に複数のファイルをコピーするには、mget コマンドを使用します。

   ```
   ftp> mget filename [filename ...]
   ```

   個々のファイル名を続けて入力するか、ワイルドカード文字を使用できます。mget コマンドでは、個々のファイルがコピーされ、そのたびに確認を求めるプロンプトが表示されます。

8. ftp による接続を終了します。

   ```
   ftp> bye
   ```

例 29–6 リモートシステムからファイルをコピーする (ftp)

次の例では、ユーザー kryten は、システム pluto と ftp 接続し、get コマンドを使用して /tmp ディレクトリから自分のホームディレクトリにファイルを 1 つコピーします。

```
$ cd $HOME
ftp pluto
```

```
Connected to pluto.
220 pluto FTP server (SunOS 5.8) ready.
Name (pluto:kryten): kryten
331 Password required for kryten.
Password: xxx
230 User kryten logged in.
ftp> cd /tmp
250 CWD command successful.
ftp> ls
200 PORT command successful.
150 ASCII data connection for /bin/ls (129.152.221.238,34344)
(0 bytes).
dtdbcache_:0
filea
files
ps_data
speckeysd.lock
226 ASCII Transfer complete.
53 bytes received in 0.022 seconds (2.39 Kbytes/s)
ftp> get filea
200 PORT command successful.
150 ASCII data connection for filea (129.152.221.238,34331)
(0 bytes).
221 Goodbye.
```

次の例では、同じユーザー kryten が mget コマンドを使用して、/tmp ディレクトリから自分のホームディレクトリに複数のファイルをコピーします。kryten は、個々のファイルについてコピーするか、しないかの選択ができることに注意してください。

```
$ ftp> cd /tmp
250 CWD command successful.
ftp> ls files
200 PORT command successful.
150 ASCII data connection for /bin/ls (129.152.221.238,34345)
(0 bytes).
fileb
filec
filed
remote: files
21 bytes received in 0.015 seconds (1.36 Kbytes/s)
ftp> cd files
250 CWD command successful.
ftp> mget file*
mget fileb? y
200 PORT command successful.
150 ASCII data connection for fileb (129.152.221.238,34347)
(0 bytes).
226 ASCII Transfer complete.
mget filec? y
200 PORT command successful.
150 ASCII data connection for filec (129.152.221.238,34348)
(0 bytes).
226 ASCII Transfer complete.
mget filed? y
200 PORT command successful.
150 ASCII data connection for filed (129.152.221.238,34351)
```

```
(0 bytes).
226 ASCII Transfer complete.200 PORT command successful.
ftp> bye
221 Goodbye.
```

▼ ファイルをリモートシステムにコピーする方法 (ftp)

1. ローカルシステム上のコピー元ディレクトリに変更します。

   ftp コマンドを入力して接続するディレクトリは、ローカルの作業用ディレクトリ、つまりこの操作のコピー元ディレクトリになります。

2. ftp により接続します。

   663 ページの「ftp によりリモートシステムへ接続する方法」を参照してください。

3. コピー先ディレクトリに変更します。

   ```
   ftp> cd target-directory
   ```

   ローカルシステムでオートマウンタを使用中であれば、/home の下に自分のホームディレクトリと並行してリモートシステムのユーザーのホームディレクトリが表示されるので注意してください。

4. コピー先ディレクトリへの書き込み権があることを確認します。

   ```
   ftp> ls -l target-directory
   ```

5. 転送タイプを **binary** に設定します。

   ```
   ftp> binary
   ```

6. ファイルを 1 つコピーするには、put コマンドを使用します。

   ```
   ftp> put filename
   ```

7. 一度に複数のファイルをコピーするには、mput コマンドを使用します。

   ```
   ftp> mput filename [filename ...]
   ```

   個々のファイル名を続けて入力するか、ワイルドカード文字を使用できます。mput コマンドでは、個々のファイルがコピーされ、そのたびに確認を求めるプロンプトが表示されます。

8. ftp による接続を終了するには、bye と入力します。

   ```
   ftp> bye
   ```

例 29–7 ファイルをリモートシステムにコピーする (ftp)

次の例では、ユーザー kryten はシステム pluto へ ftp により接続し、put コマンドを使用して自分のシステムからシステム pluto の /tmp ディレクトリにファイルをコピーします。

```
$ cd /tmp
ftp pluto
Connected to pluto.
220 pluto FTP server (SunOS 5.8) ready.
Name (pluto:kryten): kryten
331 Password required for kryten.
Password: xxx
230 User kryten logged in.
ftp> cd /tmp
250 CWD command successful.
ftp> put filef
200 PORT command successful.
150 ASCII data connection for filef (129.152.221.238,34356).
226 Transfer complete.
ftp> ls
200 PORT command successful.
150 ASCII data connection for /bin/ls (129.152.221.238,34357) (0 bytes).
dtdbcache_:0
filea
filef
files
ps_data
speckeysd.lock
226 ASCII Transfer complete.
60 bytes received in 0.058 seconds (1.01 Kbytes/s)
ftp> bye
221 Goodbye.
```

次の例では、同じユーザー kryten は mput コマンドを使用して自分のホームディレクトリから pluto の /tmp ディレクトリに複数のファイルをコピーします。kryten は、個々のファイルについてコピーするか、しないかの選択ができることに注意してください。

```
$ cd $HOME/testdir
$ ls
test1   test2   test3
$ ftp pluto
Connected to pluto.
220 pluto FTP server (SunOS 5.8) ready.
Name (pluto:kryten): kryten
331 Password required for kryten.
Password: xxx
230 User kryten logged in.
ftp> cd /tmp
250 CWD command successful.
ftp> mput test*
mput test1? y
200 PORT command successful.
150 ASCII data connection for test1 (129.152.221.238,34365).
226 Transfer complete.
```

```
mput test2? y
200 PORT command successful.
150 ASCII data connection for test2 (129.152.221.238,34366).
226 Transfer complete.
mput test3? y
200 PORT command successful.
150 ASCII data connection for filef (129.152.221.238,34356).
226 Transfer complete.
ftp> bye
221 Goodbye.
```

rcp によるリモートコピー

rcp コマンドは、ローカルシステムとリモートシステム間、または 2 台のリモートシステム間でファイルやディレクトリをコピーします。このコマンドは、リモートシステムから (rlogin コマンドでログイン後に)、またはローカルシステムから (リモートシステムにログインせずに) 使用できます。

rcp を使用すると、次のリモートコピー操作を実行できます。

- 自分のシステムからリモートシステムにファイルやディレクトリをコピーする
- リモートシステムからローカルシステムにファイルやディレクトリをコピーする
- ローカルシステムを経由したリモートシステム間でファイルやディレクトリをコピーする

オートマウンタを実行中の場合は、これらのリモート操作を cp コマンドで実行できます。ただし、cp の範囲は、オートマウンタにより作成された仮想ファイルシステムと、ユーザーのホームディレクトリから相対的に指定できる操作に制限されます。rcp はそのような制限を受けずに同じ操作を実行するので、ここでは rcp を使用する場合に限定して説明します。

コピー操作のセキュリティー上の注意事項

システム間でファイルやディレクトリをコピーするには、ログインしてファイルをコピーする許可を持っていなければなりません。

注意 – cp コマンドと rcp コマンドではともに、警告が表示されずにファイルが上書きされることがあります。コマンドを実行する前に、ファイル名が正しいかどうかを確認してください。

コピー元とコピー先の指定

C シェル内で rcp コマンドを使用すると、絶対パス名または相対パス名を使用して、コピー元 (コピーするファイルやディレクトリ) とコピー先 (ファイルやディレクトリをコピーする場所) を指定できます。

	絶対パス名	相対パス名
ローカルシステムから	mars:/home/jones/myfile.txt	~jones/myfile.txt
リモートログイン後	/home/jones/myfile.txt	~jones/myfile.txt

絶対パス名は、特定のシステムにマウントされているファイルやディレクトリを表します。上記の例で、第 1 の絶対パス名は mars システム上のファイル (myfile.txt) を表します。相対パス名は、ファイルやディレクトリがある位置を、ユーザーのホームディレクトリからの相対パスで表します。上記の例で、相対パス名は絶対パスと同じ myfile.txt を表しますが、jones のホームディレクトリを示すために「~」(チルド記号) を使用しています。

```
~ = mars:/home/jones
```

上記の 2 行目の例は、リモートログイン後の絶対パス名と相対パス名を示しています。相対パス名では明確な違いは見られません。しかし、リモートログイン操作により、jones のホームディレクトリがローカルシステム上にマウントされた (ローカルユーザーのホームディレクトリと並列に存在する) ので、絶対パス名ではシステム名 mars を指定する必要はありません。リモートログイン操作によって別のユーザーのホームディレクトリがどのようにマウントされるかについては、658 ページの「リモートログイン後の処理」を参照してください。

次の表に、C シェルが認識する絶対パス名と相対パス名の例を示します。このサンプルでは、次の用語を使用します。

- 作業用ディレクトリ – rcp コマンドの入力元のディレクトリ。リモート、ローカルのどちらの場合もあり
- 現在のユーザー – rcp コマンドを入力するユーザーの名前

表 29–4 ディレクトリ名とファイル名に使用できる構文

ログイン先	構文	説明
ローカルシステム	.	ローカルの作業用ディレクトリ
	path/filename	ローカルの作業用ディレクトリ内の *path* と *filename*
	~	現在のユーザーのホームディレクトリ
	~/path/filename	現在のユーザーのホームディレクトリの下の *path* と *filename*
	~user	*user* のホームディレクトリ

表 29–4 ディレクトリ名とファイル名に使用できる構文 (続き)

ログイン先	構文	説明
	~user/path/filename	*user* のホームディレクトリの下の *path* と *filename*
	remote-system:path/filename	リモートの作業用ディレクトリ内の *path* と *filename*
リモートシステム	.	リモートの作業用ディレクトリ
	filename	リモートの作業用ディレクトリ内の *filename*
	path/filename	リモートの作業用ディレクトリ内の *path* と *filename*
	~	現在のユーザーのホームディレクトリ
	~/path/filename	現在のユーザーのホームディレクトリ内の *path* と *filename*
	~user	*user* のホームディレクトリ
	~/user/path/filename	*user* のホームディレクトリの下の *path* と *filename*
	local-system:path/filename	ローカルの作業用ディレクトリ内の *path* と *filename*

▼ ローカルシステムとリモートシステム間でファイルをコピーする方法 (rcp)

1 コピーする許可を持っているかどうかを確認します。

少なくとも、コピー元システム上で読み取り権を持ち、コピー先システム上で書き込み権を持っているべきです。

2 コピー元とコピー先の場所を決定します。

コピー元またはコピー先のパスがわからない場合は、まず `rlogin` コマンドを使用してリモートシステムにログインします (661 ページの「リモートシステムにログインする方法 (rlogin)」を参照)。次に、そのパスが見つかるまでリモートシステム上を移動します。その後は、ログアウトしなくても次の手順を実行できます。

3 ファイルまたはディレクトリをコピーします。

```
$ rcp [-r] source-file|directory  target-file|directory
```

`rcp` (オプションなし) コピー元からコピー先にファイルを 1 つコピーする

`-r` コピー元からコピー先にディレクトリをコピーする

この構文は、リモートシステムとローカルシステムのどちらにログインするかに関係なく適用されます。表 29–4 で説明したとおり、ファイルやディレクトリのパス名のみをこのあとで示す例のように変更します。

「~」と「.」を使用すると、ローカルのファイル名やディレクトリ名のパス部分を指定できます。ただし、「~」はリモートシステムではなく現在のユーザーに適用されることと、「.」はログイン先のシステムに適用されることに注意してください。この 2 つの記号については、表 29–4 を参照してください。

例 29–8 rcp を使用してリモートファイルをローカルシステムにコピーする

次の例では、rcp はファイル letter.doc をリモートシステム pluto の /home/jones ディレクトリから、ローカルシステム earth 上の作業用ディレクトリ (/home/smith) にコピーします。

```
earth(/home/smith): rcp pluto:/home/jones/letter.doc .
```

この例では、リモートログインをしないで rcp 操作を実行しています。コマンド行の最後にある「.」記号は、リモートシステムではなく、ローカルシステムを表します。

コピー先ディレクトリもローカルユーザーのホームディレクトリなので、「~」記号で指定することもできます。

例 29–9 rlogin と rcp を使用してリモートファイルをローカルシステムにコピーする

次の例では、rlogin コマンドの実行後に rcp 操作が実行され、リモートシステムからローカルシステムにファイルをコピーしています。操作の流れは前述の例と同じですが、リモートログインによりパスが変更になります。

```
earth(/home/smith): rlogin pluto
.
.
.
pluto(/home/jones): rcp letter.doc ~
```

コマンド行の最後に「.」記号を使用するのは、この例では不適切です。リモートログインが行われているので、「.」記号はリモートシステムを指し、実際には rcp に重複したファイルを作成させることになります。ただし、「~」は、リモートシステムにログインするときにも現在のユーザーのホームディレクトリを指します。

例 29–10 rcp を使用してローカルファイルをリモートシステムにコピーする

次の例で、rcp はファイル notice.doc をローカルシステム earth のホームディレクトリ (/home/smith) からリモートシステム pluto の /home/jones ディレクトリにコピーします。

```
earth(/home/smith): rcp notice.doc pluto:/home/jones
```

リモートファイル名が指定されていないので、ファイル notice.doc は /home/jones ディレクトリに同じ名前でコピーされます。

次の例では、前の例と同じように rcp 操作が行われますが、rcp はローカルシステム上の別の作業用ディレクトリ (/tmp) で入力されます。現在のユーザーのホームディレクトリを指すために「~」記号が使われているので注意してください。

```
earth(/tmp): rcp ~/notice.doc pluto:/home/jones
```

例 29–11 rlogin と rcp を使用してローカルファイルをリモートシステムにコピーする

次の例では、rlogin コマンドの実行後に rcp 操作が実行され、ローカルファイルをリモートディレクトリにコピーしています。操作の流れは前に示した例と同じですが、リモートログインによりパスが変更になります。

```
earth(/home/smith): rlogin pluto
.
.
.
pluto(/home/jones): rcp ~/notice.doc .
```

現在のユーザーのホームディレクトリはローカルシステム上にありますが、「~」記号によりそのディレクトリが表されます。ユーザーはリモートシステムにログインしているので、「.」記号はリモートシステム上の作業用ディレクトリを表します。次の構文を使用しても同じ操作を実行します。

```
pluto(/home/jones): rcp earth:/home/smith/notice.doc /home/jones
```

パート VII

ネットワークサービスの監視(トピック)

このパートでは、ネットワークサービスの監視の手順について説明します。

◆ ◆ ◆ 第 30 章

ネットワークパフォーマンスの監視 (手順)

この章ではネットワークのパフォーマンスを監視する方法について説明します。この章で説明する手順は次のとおりです。

ネットワークパフォーマンスの監視

表 30–1 に、ネットワークのパフォーマンスを監視するために使用できるコマンドを示します。

表 30–1 ネットワーク監視コマンド

コマンド	説明
`ping`	ネットワーク上でホストの応答を調べる
`spray`	送信したパケットサイズの信頼性を検査する。パケットが遅延されていないか、落とされていないか判定できる
`snoop`	ネットワークからパケットを捕捉し、各クライアントから各サーバーへの呼び出しを追跡する
`netstat`	TCP/IP トラフィックに使用されるインタフェースや IP ルーティングテーブルなどに関するネットワーク状態と、`UDP`、`TCP`、`ICMP`、および `IGMP` についてのプロトコル別の統計情報を表示する
`nfsstat`	NFS の問題を解析するのに使用できる、サーバーおよびクライアントの統計情報の要約を表示する

ネットワーク上でホストの応答を検査する方法

ping コマンドを使用して、ネットワーク上のホストの応答を検査します。

```
$ ping hostname
```

物理的な問題が発生していると思われる場合は、ping コマンドを使用して、ネットワーク上にある複数のホストの応答時間を調べることができます。あるホストからの応答が期待していたものと異なる場合は、そのホストについて調査します。物理的な問題が発生する理由を次に示します。

- ケーブルまたはコネクタの緩み
- 接地不良
- 終端処理の欠落
- 信号の反射

このコマンドの詳細は、ping(1M) のマニュアルページを参照してください。

例 30–1 ネットワーク上のホストの応答を検査する

もっとも簡単な ping コマンドの使い方は、ネットワーク上のホストへ 1 つのパケットを送信することです。ping コマンドが正しい応答を受信すると、「*host* is alive」というメッセージが表示されます。

```
$ ping elvis
elvis is alive
```

-s オプションを指定すると、ping は 1 秒ごとにデータグラムをホストへ送り、次に各応答と、往復に要した時間を表示します。次に例を示します。

```
$ ping -s pluto
64 bytes from pluto (123.456.78.90): icmp_seq=0. time=3.82 ms
64 bytes from pluto (123.456.78.90): icmp_seq=5. time=0.947 ms
64 bytes from pluto (123.456.78.90): icmp_seq=6. time=0.855 ms
^C
----pluto PING Statistics----
3 packets transmitted, 3 packets received, 0% packet loss

round-trip (ms) min/avg/max/sttdev = 0.855/1.87/3.82/1.7
```

ネットワーク上でホストへパケットを送信する方法

spray コマンドを使用すると、送信したパケットサイズの信頼性を検査できます。

```
$ spray [ -c count -d interval -l packet-size] hostname
```

-i *count* 送信するパケット数

`-d` *interval*	パケットの送信ごとに一時停止するマイクロ秒数。遅延を使用しないと、バッファーを使い果たす可能性がある
`-l` *packet-size*	パケットサイズ
hostname	パケットを送信するシステム

このコマンドの詳細は、spray(1M) のマニュアルページを参照してください。

例 30–2 ネットワーク上のホストへパケットを送信する

次の例では、各パケットサイズが 2048 バイト (`-l 2048`) のパケット 100 個 (`-c 100`) を、ホストへ送信します。パケットは、各バースト間に 20 マイクロ秒の遅延時間 (`-d 20`) を入れて送信されます。

```
$ spray -c 100 -d 20 -l 2048 pluto
sending 100 packets of length 2048 to pluto ...
no packets dropped by pluto
279 packets/sec, 573043 bytes/sec
```

ネットワークからパケットを捕捉する方法

ネットワークからパケットを捕捉し、各クライアントから各サーバーへの呼び出しを追跡するには、`snoop` コマンドを使用します。このコマンドは、ネットワークのパフォーマンスの問題をすばやく解析するための、正確なタイムスタンプを提供します。詳細は、snoop(1m) のマニュアルページを参照してください。

```
# snoop
```

パケットがドロップするのは、バッファーの領域不足か、CPU の過負荷が原因となっている場合があります。

ネットワークの状態を調べる方法

`netstat` コマンドを使用すると、ネットワークインタフェース、ルーティングテーブル、各種プロトコルの状態に関する統計情報など、ネットワーク状態に関する情報を表示できます。

```
$ netstat [-i] [-r] [-s]
```

`-i`	TCP/IP インタフェースの状態を表示する
`-r`	IP ルーティングテーブルを表示する
`-s`	`UDP`、`TCP`、`ICMP`、および `IGMP` プロトコルについての統計情報を表示する

詳細は、netstat(1M) のマニュアルページを参照してください。

例 – ネットワークの状態を調べる

次の表示例は、netstat -i コマンドの出力を示したものです。このコマンドは、TCP/IP トラフィックに使用されるインタフェースの情報を表示します。

```
$ netstat -i
Name  Mtu  Net/Dest    Address      Ipkts  Ierrs Opkts  Oerrs Collis Queue
lo0   8232 software    localhost     1280   0     1280     0       0    0
eri0   1500 loopback    venus      1628480   0   347070    16   39354    0
```

上記の表示例は、マシンが各インタフェース上で送受信したパケット数を示しています。有効なネットワークトラフィックが存在するマシンでは、Ipkts と Opkts が継続的に増加しています。

ネットワーク衝突率は、衝突カウント (Collis) を出力パケットの数 (Opkts) で割ることにより算出できます。上記の例では、衝突率は 11% です。ネットワーク全体の衝突率が 5 から 10 % を超える場合には、問題が発生している可能性があります。

入力パケットのエラー率 (Ierrs/Ipkts) は、入力エラー数を合計入力パケット数で割ることにより算出できます。出力パケットのエラー率 (Oerrs/Opkts) は、出力エラー数を合計出力パケット数で割ることにより算出できます。入力エラー率が高い場合 (0.25% を超えている場合)、ホストがパケットを落としている可能性があります。

次に、netstat -s コマンドの出力例を示します。このコマンドは、UDP、TCP、ICMP、および IGMP についてプロトコル別の統計情報を表示します。

```
UDP
    udpInDatagrams      =196543    udpInErrors         =     0
    udpOutDatagrams     =187820

TCP
    tcpRtoAlgorithm     =     4    tcpRtoMin           =   200
    tcpRtoMax           = 60000    tcpMaxConn          =    -1
    tcpActiveOpens      = 26952    tcpPassiveOpens     =   420
    tcpAttemptFails     =  1133    tcpEstabResets      =     9
    tcpCurrEstab        =    31    tcpOutSegs          =3957636
    tcpOutDataSegs      =2731494   tcpOutDataBytes     =1865269594
    tcpRetransSegs      = 36186    tcpRetransBytes     =3762520
    tcpOutAck           =1225849   tcpOutAckDelayed    =165044
    tcpOutUrg           =     7    tcpOutWinUpdate     =   315
    tcpOutWinProbe      =     0    tcpOutControl       = 56588
    tcpOutRsts          =   803    tcpOutFastRetrans   =   741
    tcpInSegs           =4587678
    tcpInAckSegs        =2087448   tcpInAckBytes       =1865292802
    tcpInDupAck         =109461    tcpInAckUnsent      =     0
    tcpInInorderSegs    =3877639   tcpInInorderBytes   =-598404107
    tcpInUnorderSegs    = 14756    tcpInUnorderBytes   =17985602
    tcpInDupSegs        =    34    tcpInDupBytes       = 32759
    tcpInPartDupSegs    =   212    tcpInPartDupBytes   =134800
    tcpInPastWinSegs    =     0    tcpInPastWinBytes   =     0
    tcpInWinProbe       =   456    tcpInWinUpdate      =     0
```

```
    tcpInClosed         =    99    tcpRttNoUpdate      =  6862
    tcpRttUpdate        =435097    tcpTimRetrans       = 15065
    tcpTimRetransDrop   =    67    tcpTimKeepalive     =   763
    tcpTimKeepaliveProbe=     1    tcpTimKeepaliveDrop =     0

IP
    ipForwarding        =     2    ipDefaultTTL        =   255
    ipInReceives        =11757234  ipInHdrErrors       =     0
    ipInAddrErrors      =     0    ipInCksumErrs       =     0
    ipForwDatagrams     =     0    ipForwProhibits     =     0
    ipInUnknownProtos   =     0    ipInDiscards        =     0
    ipInDelivers        =4784901   ipOutRequests       =4195180
    ipOutDiscards       =     0    ipOutNoRoutes       =     0
    ipReasmTimeout      =    60    ipReasmReqds        =  8723
    ipReasmOKs          =  7565    ipReasmFails        =  1158
    ipReasmDuplicates   =     7    ipReasmPartDups     =     0
    ipFragOKs           = 19938    ipFragFails         =     0
    ipFragCreates       =116953    ipRoutingDiscards   =     0
    tcpInErrs           =     0    udpNoPorts          =6426577
    udpInCksumErrs      =     0    udpInOverflows      =   473
    rawipInOverflows    =     0

ICMP
    icmpInMsgs          =490338    icmpInErrors        =     0
    icmpInCksumErrs     =     0    icmpInUnknowns      =     0
    icmpInDestUnreachs  =   618    icmpInTimeExcds     =   314
    icmpInParmProbs     =     0    icmpInSrcQuenchs    =     0
    icmpInRedirects     =   313    icmpInBadRedirects  =     5
    icmpInEchos         =   477    icmpInEchoReps      =    20
    icmpInTimestamps    =     0    icmpInTimestampReps =     0
    icmpInAddrMasks     =     0    icmpInAddrMaskReps  =     0
    icmpInFragNeeded    =     0    icmpOutMsgs         =   827
    icmpOutDrops        =   103    icmpOutErrors       =     0
    icmpOutDestUnreachs =    94    icmpOutTimeExcds    =   256
    icmpOutParmProbs    =     0    icmpOutSrcQuenchs   =     0
    icmpOutRedirects    =     0    icmpOutEchos        =     0
    icmpOutEchoReps     =   477    icmpOutTimestamps   =     0
    icmpOutTimestampReps=     0    icmpOutAddrMasks    =     0
    icmpOutAddrMaskReps =     0    icmpOutFragNeeded   =     0
    icmpInOverflows     =     0

IGMP:
        0 messages received
        0 messages received with too few bytes
        0 messages received with bad checksum
        0 membership queries received
        0 membership queries received with invalid field(s)
        0 membership reports received
        0 membership reports received with invalid field(s)
        0 membership reports received for groups to which we belong
        0 membership reports sent
```

次に、netstat -r コマンドの出力例を示します。このコマンドは、IP ルーティングテーブルを表示します。

```
Routing Table:
  Destination        Gateway           Flags  Ref   Use    Interface
```

```
------------------ ------------------- ----- ----- ------ ---------
localhost          localhost           UH        0   2817  lo0
earth-bb           pluto               U         3  14293  eri0
224.0.0.0          pluto               U         3      0  eri0
default            mars-gate           UG        0  14142
```

表 30–2 は、netstat -r コマンドが出力するレポート中のフィールドを説明しています。

表 **30–2** netstat -r コマンドの出力

フィールド名		説明
Flags	U	ルートが正常に動作している
	G	ルートはゲートウェイを経由する
	H	ルートはホスト宛である
	D	ルートはリダイレクトを使用して動的に作成された
Ref		同じリンク層を共有している現在のルート数を示す
Use		送信されたパケット数を示す
Interface		ルートに使用されるネットワークインタフェースを表示する

NFS サーバーとクライアントの統計情報を表示する方法

NFS 分散型ファイルサービスは、ローカルコマンドをリモートホストへの要求に変換する、遠隔手続き呼び出し (RPC) 機能を使用します。遠隔手続き呼び出しは同期型の呼び出しです。サーバーが呼び出しを完了してその結果を返すまで、クライアントアプリケーションはブロックまたは中断されます。NFS のパフォーマンスに影響を与える主要な要素の 1 つに再伝送率があります。

ファイルサーバーがクライアントの要求に応答できない場合、そのクライアントは、指定された回数だけ要求を再伝送して終了します。再伝送のたびにシステムにオーバーヘッドがかかり、ネットワークトラフィックが増加します。過度の再伝送はネットワークのパフォーマンスを低下させます。再伝送率が高い場合、次を調べてください。

- サーバーが過負荷になっており、要求の処理に時間がかかりすぎていないか
- Ethernet インタフェースがパケットを落としていないか
- ネットワークの輻輳によりパケットの伝送が低下していないか

表 30–3 に、クライアントとサーバーの統計情報を表示するための nfsstat コマンドのオプションとその説明を示します。

表 30–3 クライアントとサーバーの統計情報を表示するためのコマンド

コマンド	表示される情報
`nfsstat -c`	クライアントの統計情報
`nfsstat -s`	サーバーの統計情報
`netstat -m`	ファイルシステムごとのネットワーク統計情報

クライアントの統計情報を表示するには `nfsstat -c` を使用し、サーバーの統計情報を表示するには `nfsstat -s` を使用します。また、ファイルシステムごとのネットワークの統計情報を表示するには、`nfsstat -m` を使用します。詳細は、`nfsstat(1M)` のマニュアルページを参照してください。

例 – NFS サーバーとクライアントの統計情報を表示する

次の例は、クライアント `pluto` の RPC と NFS データを表示します。

```
$ nfsstat -c

Client rpc:
Connection oriented:
calls    badcalls  badxids  timeouts newcreds  badverfs   timers
1595799  1511      59       297      0         0          0
cantconn nomem     interrupts
1198      0         7
Connectionless:
calls    badcalls  retrans  badxids  timeouts  newcreds   badverfs
80785    3135      25029    193      9543      0          0
timers   nomem     cantsend
17399    0         0

Client nfs:
calls    badcalls  clgets   cltoomany
1640097  3112      1640097  0
Version 2: (46366 calls)
null     getattr   setattr  root     lookup     readlink  read
0 0%     6589 14%  2202 4%  0 0%     11506 24%  0 0%      7654 16%
wrcache  write     create   remove   rename     link      symlink
0 0%     13297 28% 1081 2%  0 0%     0 0%       0 0%      0 0%
mkdir    rmdir     readdir  statfs
24 0%    0 0%      906 1%   3107 6%
Version 3: (1585571 calls)
null    getattr    setattr  lookup     access     readlink  read
0 0%    508406 32% 10209 0% 263441 16% 400845 25% 3065 0%  117959 7%
write    create     mkdir    symlink    mknod    remove   rmdir
69201 4% 7615 0%    42 0%    16 0%      0 0%     7875 0%  51 0%
rename   link       readdir  readdir+   fsstat   fsinfo   pathconf
929 0%   597 0%     3986 0%  185145 11% 942 0%   300 0%   583 0%
commit
4364 0%

Client nfs_acl:
```

```
Version 2: (3105 calls)
null        getacl      setacl      getattr     access
0 0%        0 0%        0 0%        3105 100%   0 0%
Version 3: (5055 calls)
null        getacl      setacl
0 0%        5055 100%   0 0%
```

表 30–4 に、nfsstat -c コマンドの出力とその説明を示します。

表 30–4 nfsstat -c コマンドの出力とその説明

フィールド	説明
calls	送信された合計呼び出し数
badcalls	RPC によって拒否された合計呼び出し数
retrans	再伝送の合計数。このクライアントの場合、再伝送回数は 1% 未満 (6888 回の呼び出しのうち、10 回程度のタイムアウト)。再伝送は一時的な異常により発生する可能性がある。1% 以上の再伝送率の場合は、問題が発生している可能性がある
badxid	1 つの NFS 要求に対して重複する承認を受信した回数
timeout	タイムアウトした呼び出しの回数
wait	利用可能なクライアントハンドルがないために呼び出しが待機した回数
newcred	認証情報を書き換えなければならなかった回数
timers	タイムアウト値が、呼び出しに対して指定されたタイムアウト値以上であった回数
readlink	シンボリックリンクに対して読み取りが行われた回数。この値が大きい (10 % を超える) 場合は、シンボリックリンクが多すぎる可能性がある

次に、nfsstat -m コマンドの出力例を示します。

```
pluto$ nfsstat -m
/usr/man from pluto:/export/svr4/man
Flags: vers=2,proto=udp,auth=unix,hard,intr,dynamic,
        rsize=8192, wsize=8192,retrans=5
 Lookups: srtt=13 (32ms), dev=10 (50ms), cur=6 (120ms)
 All:     srtt=13 (32ms), dev=10 (50ms), cur=6 (120ms)
```

表 30–5 に、ミリ秒単位で表示される nfsstat -m コマンドの出力を示します。

表 30–5 nfsstat -m コマンドの出力

フィールド	説明
srtt	平準化された平均往復時間
dev	平均偏差

表 30–5 nfsstat -m コマンドの出力 (続き)

フィールド	説明
cur	現在の「予測」応答時間

ネットワークのハードウェアコンポーネントに問題の原因があると思われる場合は、ケーブルおよびコネクタを綿密にチェックしてください。

用語集

asppp
Solaris 2.4 から Solaris 8 リリースまでの Solaris オペレーティングシステムに含まれる PPP のバージョンの 1 つ。asppp は非同期 PPP 通信のみサポートします。

CHAP
チャレンジハンドシェーク認証プロトコルは、PPP リンク上の発呼者識別情報の検証に使用できる認証プロトコルです。CHAP 認証では、「チャレンジ」と「応答」の概念を使用します。呼び出しを受信したマシンが呼び出し側にチャレンジを送信してその識別情報を確認します。

パスワード認証プロトコル (PAP)も参照してください。

CHAP シークレット
識別目的で使用される ASCII またはバイナリ文字列。PPP リンク上の両ピアにより認識されます。CHAP シークレットはシステムの /etc/ppp/chap-secrets ファイル内に平文のまま保存されますが、PPP リンク上には、たとえ暗号化された形であっても、決して送信されることはありません。CHAP プロトコルは、呼び出し元が使用する CHAP シークレットのハッシュと、受け取り側の /etc/ppp/chap-secrets ファイルに設定されている呼び出し元の CHAP シークレットエントリのハッシュが一致することを検証します。

chat スクリプト
モデムとリモートピアの間の通信リンクを確立する方法を、モデムに指示する手順。PPP プロトコルと UUCP プロトコルは、ともにダイアルアップリンク確立とダイアルバック呼び出しに chat スクリプトを使用します。

CSU/DSU
CSU デバイスと DSU デバイスを組み合わせた同期通信装置。専用回線 PPP リンク上で使用します。CSU/DSU はピアからの信号を専用回線に変換します。CSU/DSU の多くはリンクを確立するための chat スクリプトを必要としません。多くの場合、CSU/DSU は専用回線プロバイダにより構成されます。

チャネルサービス装置 (CSU) と 加入者線終端装置 (DSU)も参照してください。

ISDN 端末アダプタ (TA)
信号変換装置。ISDN ネットワーク上でダイアルアップ PPP リンクにモデムと同等のインタフェースを提供します。ISDN TA を構成する場合、標準モデムを構成する場合と同じ Solaris PPP 4.0 構成ファイルを使用します。

Microsoft CHAP (MS-CHAP)
独自の PPP 用 Microsoft 認証プロトコル。Solaris PPP 4.0 では、クライアントモードとサーバーモードの両方において、このプロトコルの version 1 と 2 をサポートします。

PPPoE (PPP over Ethernet)
RedBack Networks 独自のプロトコル。このプロトコルを使用して、ホストがイーサネットリンク上で PPP セッションを実行できます。PPPoE は通常デジタル加入者回線 (DSL) サービスで使用されます。

SLP デーモン **(slpd)**	SLP の Solaris 実装で DA または SA サーバーとして動作するデーモンプロセス。ホスト上でのサービス処理は、通知を個々に保持するのではなく、`slpd` を使用してサービス通知を登録します。SLP デーモンが SA サーバーとして構成される場合、各プロセスには、`slpd` と通信する SA クライアントライブラリが含まれます。SLP デーモンはすべての登録と登録解除を DA に転送します。デーモンは有効期限が切れたサービス通知を時間切れとし、アクティブまたはパッシブな DA 検出を実行して、利用可能な DA のテーブルを保守します。これらの仕組みを通して、DA の情報が UA クライアントに提供されます。UA クライアントは DA 情報についてのみホスト上で `slpd` を使用します。オプションで `slpd` を DA として構成できます。
アカウンティングの拡張	Solaris オペレーティングシステムで、タスクまたはプロセスに基づいた資源消費量を柔軟に記録できる方法。
圧縮制御プロトコル **(CCP)**	PPP のサブプロトコル。リンク上でのデータ圧縮の使用についてネゴシエーションを行います。ヘッダー圧縮とは異なり、CCP はリンク上に送信されたパケット内のすべてのデータを圧縮します。
インターネットプロトコル制御プロトコル **(IPCP)**	PPP のサブプロトコル。リンク上のピアの IP アドレスについてネゴシエーションを行います。また、リンクのヘッダー圧縮をネゴシエーションし、ネットワーク層プロトコルを使用可能にします。
インターネットプロトコルバージョン **6** 制御プロトコル **(IPV6CP)**	インターネットプロトコル制御プロトコル (IPCP) を参照してください。
加入者線終端装置 **(DSU)**	専用回線 PPP リンク上で使用する同期通信装置。DSU は通信回線上で使用されるデータフレーミング形式間の変換を行い、標準データ通信インタフェースを提供します。 チャネルサービス装置 (CSU) と CSU/DSU も参照してください。
コールバック制御プロトコル **(CBCP)**	Microsoft 独自の PPP 拡張機能。コールバックセッションのネゴシエーションに使用します。Solaris PPP 4.0 ではこのプロトコルのクライアント側 (最初の呼び出し側) のみサポートします。
サービス **URL**	サービスのネットワークロケーションを通知するために使用される URL。URL は、サービスの種類、ホスト名、サービスホストのネットワークアドレスから構成されます。URL には、ポート番号や、サービスを使用するために必要なその他の情報が使用される場合もあります。
サービスエージェント **(SA)**	ネットワークサービスのサービス通知を保守する SLP エージェント。DA が使用できない場合は、SA が UA からのサービス要求のマルチキャストに答えます。DA が使用できる場合は、SA はそのスコープをサポートする DA にサービスを登録、あるいはオプションで登録解除します。
サービス通知	サービスを定義する SA により配布される情報。サービス通知は、サービスを説明する、URL および、属性と値の対のリストの集合です。すべてのサービス通知には有効期限があります。期限が切れると、サービス通知は再登録されない限り無効になります。

信頼できる呼び出し元
: PPP において、ダイアルインサーバーがアクセスを許可するリモートピア。リモートピアのセキュリティー資格をダイアルインサーバーの PAP または CHAP シークレットデータベースに追加することによりアクセスを許可します。

スコープ
: 管理上、位相上、またはその他の関係により整理された UA と SA のグループ化。スコープを使用して、企業全体のサービスへのアクセスを提供する方法を変更できます。

専用回線 **PPP** リンク
: ホストと、プロバイダからリースした同期ネットワーク媒体に接続された CSU/DSU からなる PPP 接続。専用回線媒体の一般的な例として OC3、T1 があります。管理は簡単ですが、専用回線リンクはダイアルアップ PPP リンクよりも費用がかかることから、広くは使われていません。

ダイアルアウトマシン
: ダイアルアップリンクを確立するための呼び出しを開始するピア。構成後は、ダイアルアウトマシンは任意の台数のダイアルインサーバーを呼び出すことができます。一般に、ダイアルアップリンクを確立するには、ダイアルアウトマシンが認証資格を提供する必要があります。

ダイアルアップ **PPP** リンク
: 電話回線または ISDN が提供する媒体など、通信媒体の一方の端にピアとモデムが使用されている PPP 接続。「ダイアルアップ」という用語は、ローカルモデムがリモートピアの電話番号を使用してダイアルアップする場合のリンクネゴシエーションにおけるシーケンスを指します。ダイアルアップリンクは最も広く使用され、最小コストの PPP 構成です。

ダイアルインサーバー
: ダイアルアウトマシンから呼び出しを受け、ダイアルアップ PPP リンクの受け取り側をネゴシエーションし、確立するピア。「ダイアルインサーバー」という用語が一般に使用されていますが、クライアントサーバーという形では動作しません。形としては、ピアがダイアルアップリンクの設定要求に応答するだけです。構成後は、ダイアルインサーバーは任意の台数のダイアルアウトマシンからの呼び出しを受信できます。

チャネルサービス装置 **(CSU)**
: 専用通信回線へのローカルインタフェースを提供し、その回線を終端する同期通信装置。米国内では、CSU は T1 回線を終端し、DS1 インタフェースまたは DSX インタフェースを提供します。国際的には、電話会社プロバイダが CSU を所有するのが一般的です。

  CSU/DSU と 加入者線終端装置 (DSU)も参照してください。

ディレクトリエージェント **(DA)**
: オプションの SLP エージェント。サービスエージェント (SA) が送信するサービス通知をキャッシュに保存し、維持します。DA が配置された場合、DA がユーザーエージェント (UA) のサービス要求を解決します。DA はディレクトリ通知に対して、SA および UA からの能動的な要請に応答します。その応答により、SA と UA は関連付けられた DA とスコープを検出します。DA は定期的に非要請通知を送りますが、この通知を通して SA および UA は共有のスコープ内で DA を検出します。

同期 **PPP**
: 同期デジタル回線上の PPP の形式。生のビットを連続ストリームとして転送します。専用回線 PPP リンクは同期 PPP を使用します。

認証	プログラムなどのエンティティーまたはリモートユーザーがネットワークを通して提供する識別情報の検証作業。一部の認証プロトコルでは、潜在的ユーザーから認証資格のデータベースを構築できます。その他の認証プロトコルでは、認証を目的として認証局が生成する信頼の証明書チェーンを使用します。これらの資格を使用して、通信やサイトのサービスの利用を要求するユーザーを認証することができます。
パスワード認証プロトコル **(PAP)**	PPP リンク上の発呼者識別情報の検証に使用できる認証プロトコル。PAP は平文パスワードを使用し、このパスワードはリンク上に送信されるので、パスワードを端点のマシンの中の 1 つに保存できます。たとえば、呼び出しを受信するマシン上の UNIX `password` データベース内のログインとパスワードエントリを使用して、呼び出し元の識別情報を検証することができます。 CHAP も参照してください。
ピア	PPP では、PPP 通信リンクの一端にある 1 台のコンピュータのこと。PPP 通信リンクは、通信媒体により接続された 2 台のピアから構成されます。ワークステーション、パソコン、ルーター、メインフレームなど、多様な機器をピアとして構成できます。
非同期 **PPP**	非同期シリアル回線上の PPP の形式。同時に 1 文字ずつデータ転送します。最も一般的な PPP の形式であるダイアルアップリンクでは、非同期 PPP 通信が使用されています。
ブロードキャスト	サブネット上の全マシンにパケットを転送するデータリンク層の手順。一般にブロードキャストパケットがサブネットを超えてルーティングされることはありません。
ポイントツーポイントプロトコル **(PPP)**	ポイントツーポイント媒体上でデータグラムを転送する標準方法を提供するデータリンク層プロトコル。PPP 構成はピアと呼ばれる 2 台の端点コンピュータ、およびピアが通信に使用する電話回線またはその他の双方向リンクから構成されます。2 台のピア間のハードウェアおよびソフトウェア接続が *PPP* リンクであると考えられます。 PPP は、PAP、CHAP、LCP、CCP などの複数のサブプロトコルから構成されます。利用できる PPP 実装は多数存在します。Solaris 9 オペレーティングシステムでは Solaris PPP 4.0 が実装されています。
マルチキャスト	ネットワーク層手順。IP ネットワーク上の複数マシンにデータグラムパケットを送信するのに使用されます。ブロードキャストルーティングの場合と同じく、パケットはすべてのマシンによって処理されるわけではありません。マルチキャストでは、ルーターを特殊なルーティングプロトコルで構成する必要があります。
ユーザーエージェント **(UA)**	ユーザーアプリケーションの代わりに動作する SLP エージェント。ユーザーエージェントは、対応する適用範囲、ディレクトリエージェント、サービス通知の識別情報を問い合わせます。
予期-送信 **(expect-send)**	PPP chat スクリプトや UUCP chat スクリプトで使用されるスクリプト記述形式。chat スクリプトは、リモートピアからの受け取りを期待する *(expect)* テキストまたは手順で始まります。次の行には、リモートピアから期待どおりの文字列を受信した後にローカルホストが送信する *(send)* 応答が記述されます。その後に続く行では、通信確立に必要な手順が正常にネゴシエーションされるまで、ローカルホストとリモートピア間の予期-送信 (expect-send) 手順が繰り返されます。
リンク	PPP では、2 つのピア間でネゴシエーションされ、確立される通信接続のこと。Solaris PPP 4.0 では、ダイアルアップと専用回線の 2 種類のリンクをサポートします。

リンク制御プロトコル **(LCP)**	PPP のサブプロトコル。ピア間リンクパラメータの初期セットのネゴシエーションに使用されます。LCP の機能に接続完全性テストが含まれるため、リンク関連の問題の多くは LCP 異常として検出されます。
レガシーサービス	SLP 対応していないネットワーク化サービス。プロキシ登録を作成して、レガシーサービスを SLP に登録することができます。そうすると、SLP ベースのクライアントはレガシーサービスを検出できます (第 10 章「レガシーサービスの組み込み」を参照)。

索引

B

C

D

E

F

G

H

M

N

O

P

Q

R

S

T

U

V

W

X

あ

し

す

せ

ち

て

と

な

に

み