Oracle® Solaris 11.1 での UUCP および PPP を使用したシリアルネットワークの管理

Part No: E36524–02
2012 年 10 月

ORACLE

121210@25097

目次

図目次

表目次

例目次

はじめに

『Oracle Solaris 11.1 での UUCP および PPP を使用したシリアルネットワークの管理』は Oracle Solaris システム管理情報の大部分を説明する複数巻から成るドキュメントセットの一部です。このドキュメントでは、Oracle Solaris 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。

注 – この Oracle 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使用するシステム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、Oracle Solaris OS: Hardware Compatibility Lists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。

対象読者

このドキュメントは、Oracle Solaris 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。このドキュメントを活用するには、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。

Oracle サポートへのアクセス

Oracle のお客様は、My Oracle Support を通じて電子的なサポートを利用することができます。詳細は、`http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` を参照してください。聴覚に障害をお持ちの場合は、`http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` を参照してください。

表記上の規則

次の表では、このドキュメントで使用される表記上の規則について説明します。

表 P–1 表記上の規則

字体	説明	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。 `machine_name% you have mail.`
`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。	`machine_name%` **`su`** `Password:`
aabbcc123	プレースホルダ: 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。
AaBbCc123	書名、新しい単語、および強調する単語を示します。	『ユーザーズガイド』の第 6 章を参照してください。 キャッシュは、ローカルに格納されるコピーです。 ファイルを保存しないでください。 注: いくつかの強調された項目は、オンラインでは太字で表示されます。

コマンド例のシェルプロンプト

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。

表 P–2 シェルプロンプト

シェル	プロンプト
Bash シェル、Korn シェル、および Bourne シェル	`$`
Bash シェル、Korn シェル、および Bourne シェルのスーパーユーザー	`#`
C シェル	`machine_name%`

表 P–2 シェルプロンプト (続き)

シェル	プロンプト
C シェルのスーパーユーザー	machine_name#

◆ ◆ ◆ 第 1 章

Solaris PPP 4.0 (概要)

このセクションでは、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 を紹介します。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できます。Solaris PPP 4.0 は基本インストールの一部として含まれています。

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、Oracle 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)

どのバージョンの Solaris PPP を使用すべきか

サポート対象の PPP 実装は、Solaris PPP 4.0 です。Solaris 9 リリース以降のリリースには、以前の非同期 Solaris PPP (`asppp`) ソフトウェアは含まれていません。詳細は、第 9 章「Asynchronous Solaris PPP から Solaris PPP 4.0 への移行 (タスク)」を参照してください。

Solaris PPP 4.0 を使用する理由

`asppp` を現在使用中の場合は、Solaris PPP 4.0 への移行を検討してください。この 2 つの Solaris PPP 技術の間には、次のような違いがあります。

- 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 ipadm 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 がインタフェースの作成や削除を動的に行います。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** では使用不可能な **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、163 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、Oracle Solaris システム管理、および前バージョンの PPP については、システム管理者のリソース 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

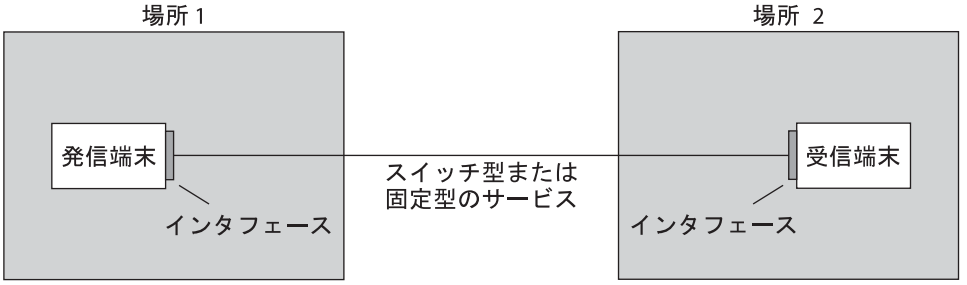
このセクションでは、PPP 構成について説明します。また、このガイドで使用する用語についても説明します。

Solaris PPP 4.0 はいくつか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 1–1 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。Oracle 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。

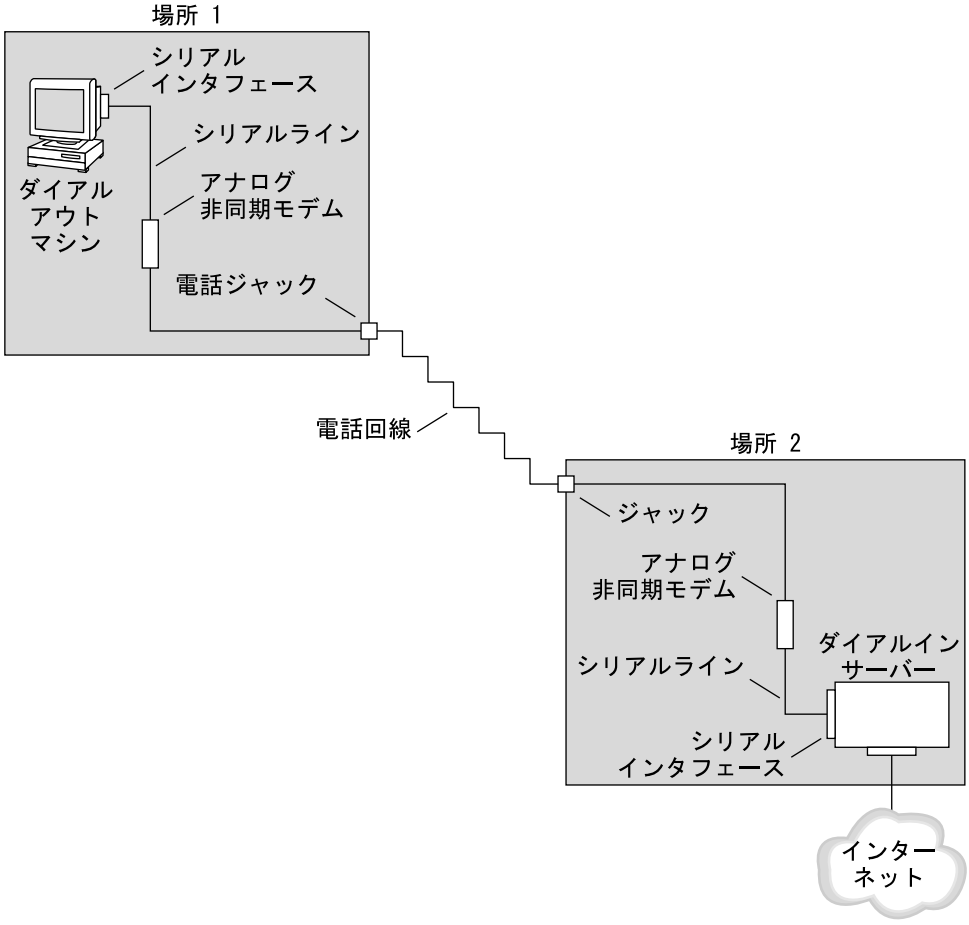
このガイドでは、ダイアルアップ接続を開始するローカルピアは、ダイアルアウトマシンと呼びます。着呼を受信するピアは、ダイアルインサーバーと呼びます。このマシンは実際にはダイアルアウトマシンがターゲットにするマシンに過ぎず、真の意味でのサーバーではない場合もあります。

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 1–2 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、Oracle 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、134 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

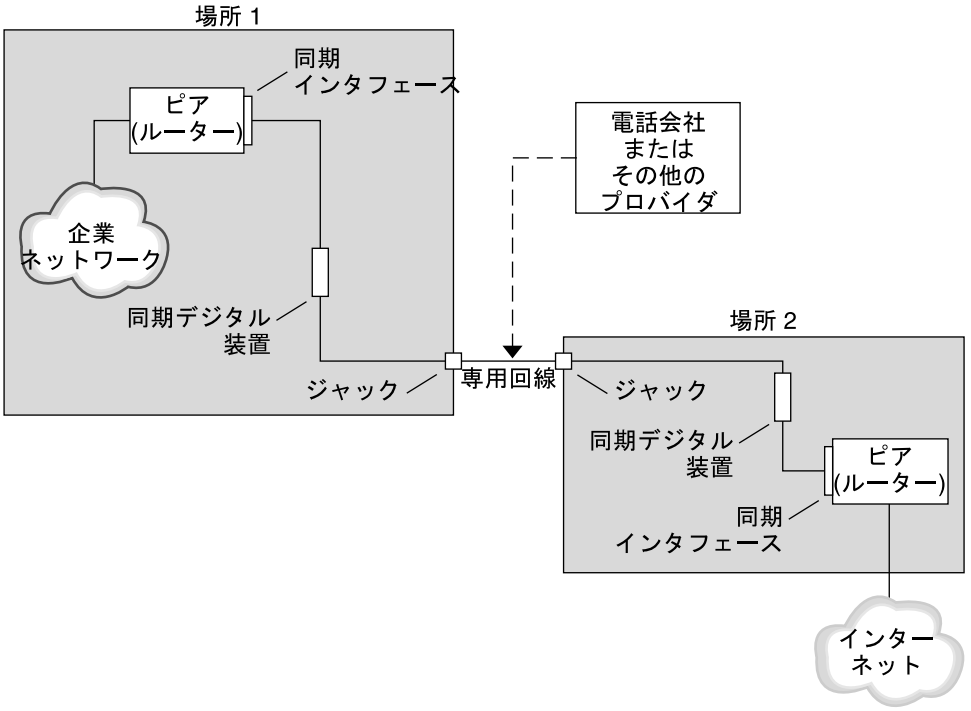
ダイアルアップと専用回線のリンクはともに、通信媒体で接続されている 2 つのピアから成っています。次の表は、2 つのリンクタイプの相違をまとめています。

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 1–3 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。Oracle 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 Oracle Solaris 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` コマンドをブート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、138 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、141 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を確認しますが、PPP 認証ではデータの機密性は提供されません。機密性には、IPsec、PGP、SSL、Kerberos、Secure Shell 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アな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。

このセクションでは、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

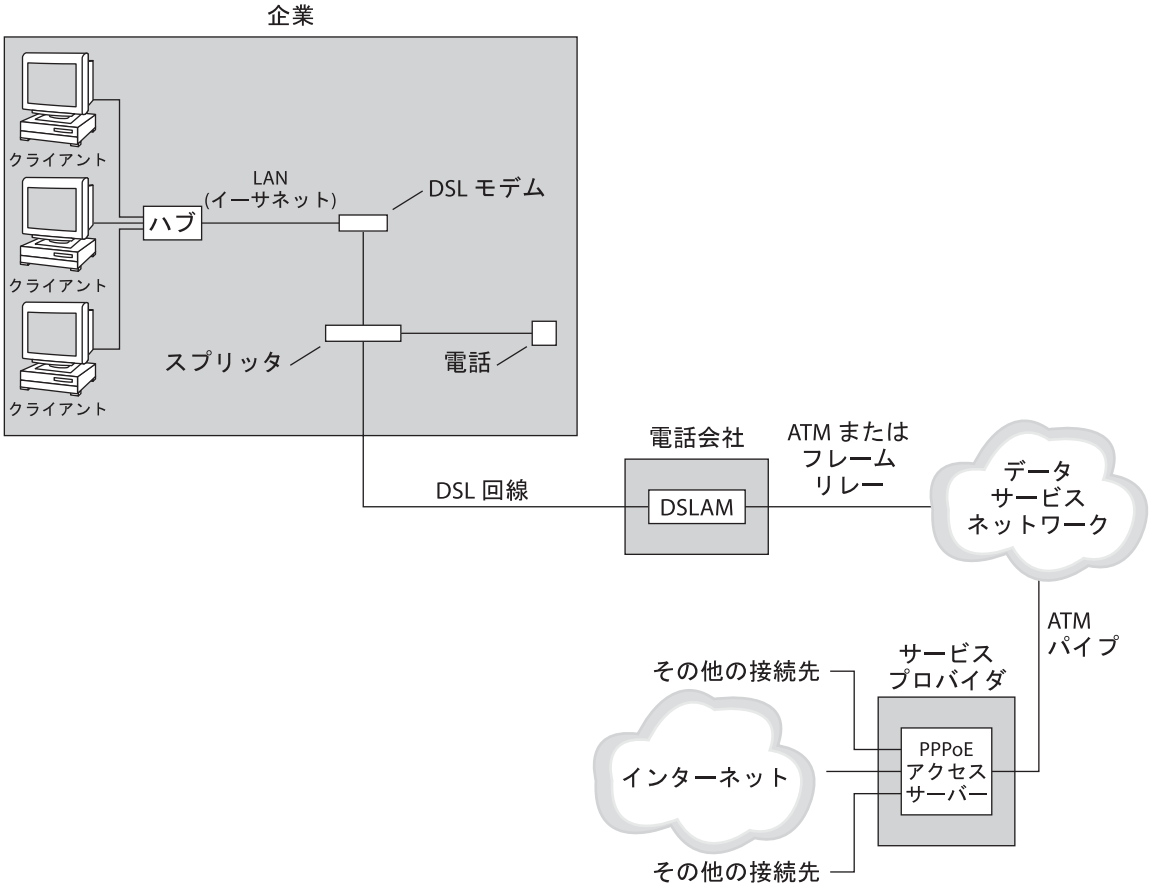
PPPoE は、RedBack Networks が生み出した独自のプロトコルです。PPPoE は、別バージョンの標準 PPP ではなく検出プロトコルです。PPPoE のシナリオでは、最初に PPP 通信を開始するマシンが、PPPoE を実行しているピアを検出する必要があります。PPPoE プロトコルは、Ethernet ブロードキャストパケットを使ってピアを検出します。

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、146 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 1–4　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、このガイドでは、「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 1–2 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 2 章

PPP リンクの計画 (タスク)

PPP リンクの設定には、タスク計画や PPP と無関係なタスクなど、さまざまな個別のタスクが含まれています。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 2 章「PPP リンクの計画 (タスク)」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 (タスクマップ)

PPP では、実際にリンクを設定する前にタスク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。次のタスクマップは、この章で説明する大規模なタスク計画を示しています。構成するリンクタイプによっては、一般的なタスクだけで十分な場合があります。また、リンク、認証、および PPPoE の各タスクが必要になる場合もあります。

表 2–1 PPP 計画 (タスクマップ)

タスク	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	38 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	41 ページの「専用回線リンクの計画」

表 2–1 PPP 計画 (タスクマップ) (続き)

タスク	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	44 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	49 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。このセクションでは、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 3 章「ダイアルアップ PPP リンクの設定 (タスク)」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、25 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 – このセクションの計画情報には、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、44 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、49 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 2–2 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ドキュメントを参照します。
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ドキュメントを参照します。
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 のドキュメントを参照します。

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 – このセクションの計画情報には、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、44 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、49 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 2–3 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ドキュメントを参照します。
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	63 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

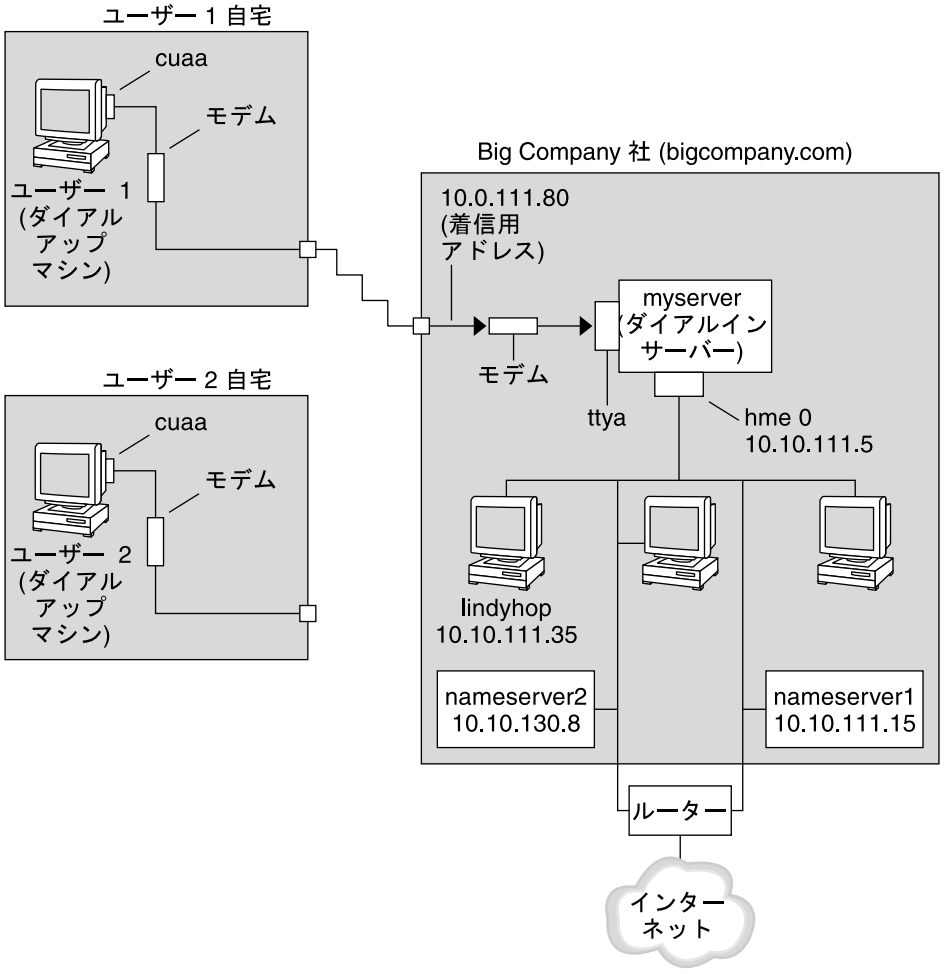
第 3 章「ダイアルアップ PPP リンクの設定 (タスク)」で説明するタスクでは、従業員が週に 2、3 日在宅勤務できるようにするための、ある小企業の要件を実行します。一部の従業員は、ホームマシンに Oracle Solaris OS が必要になります。また、社内イントラネット上にある作業マシンにリモートログインすることも必要になります。

タスクでは、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 3 章「ダイアルアップ PPP リンクの設定 (タスク)」で設定されているリンクを示します。

図 2–1 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 3–2 を参照してください。
- ダイアルインマシンを設定するには、表 3–3 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、25 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、115 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。

このセクションでは、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 2–2 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、29 ページの「専用回線 PPP の概要」を参照してください。専用回線の設定タスクについては、第 4 章「専用回線 PPP リンクの設定 (タスク)」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 2–4 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ドキュメントを参照します。
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ドキュメントを参照します。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ドキュメントを参照します。または、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ドキュメントを参照します。または、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ドキュメントを参照します。または、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ドキュメントを参照します。または、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

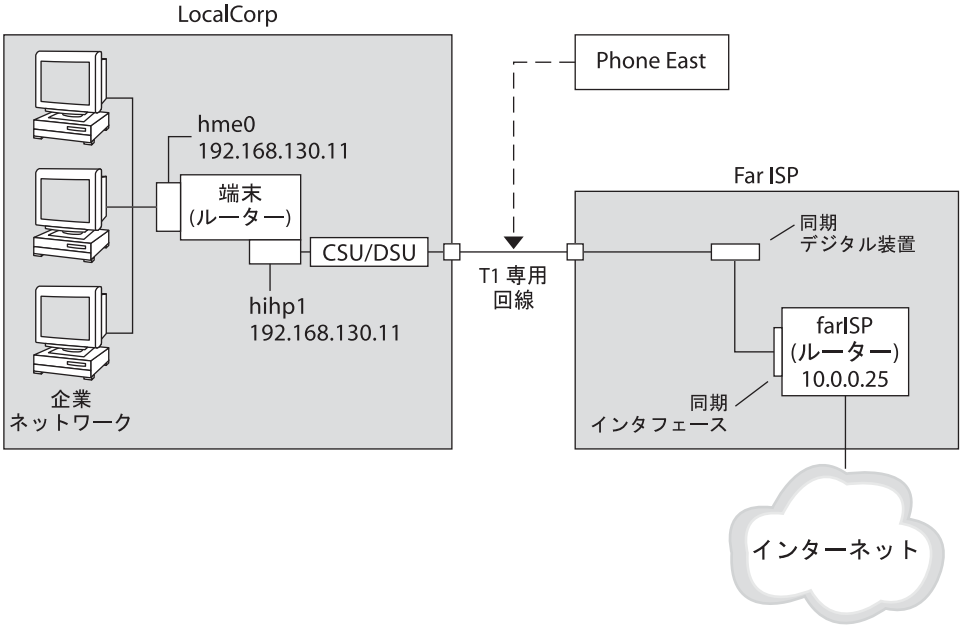
第 4 章「専用回線 PPP リンクの設定 (タスク)」のタスクは、従業員にインターネットアクセスを提供するために、ある中規模組織 (LocalCorp 社) で目的を達成する方法を示しています。現在、従業員のコンピュータは、会社の私設イントラネットに接続されています。

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リソース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。

タスクでは、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 2–2　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 4 章「専用回線 PPP リンクの設定 (タスク)」
- 29 ページの「専用回線 PPP の概要」

リンクへの認証計画

このセクションでは、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 5 章「PPP 認証の設定 (タスク)」は、自分のサイトで PPP 認証を実装するためのタスクを示しています。

PPP には、PAP と CHAP の 2 種類の認証があります。PAP の詳細は、138 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、141 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、33 ページの「PPP 認証を使用する理由」を参照してください。

このセクションでは、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 2–3 および図 2–4 に示されている認証事例の説明

認証の設定タスクについては、第 5 章「PPP 認証の設定 (タスク)」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 2–5 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タスク	第 3 章「ダイアルアップ PPP リンクの設定 (タスク)」
リンクのテストタスク	第 7 章「一般的な PPP 問題の解決 (タスク)」
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	33 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、138 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

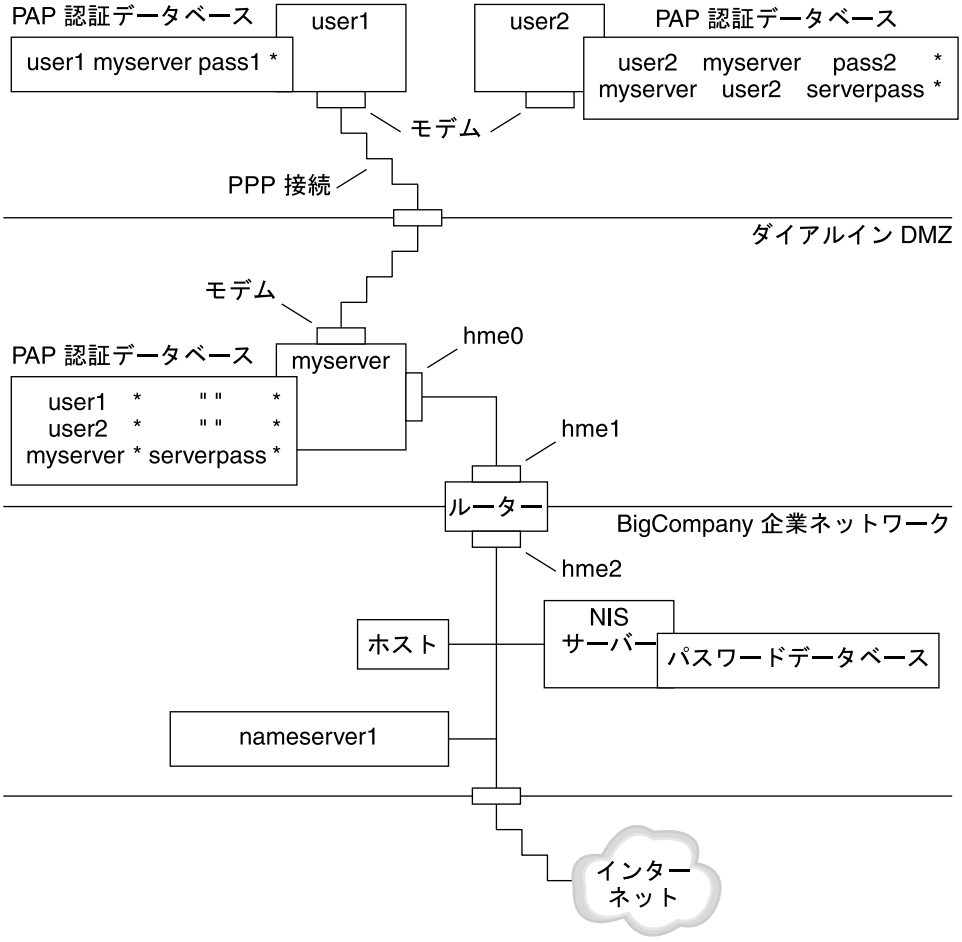
このセクションでは、第 5 章「PPP 認証の設定 (タスク)」の手順で使用されている認証事例について説明します。

PAP 認証による構成例

74 ページの「PAP 認証の構成」でのタスクは、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、39 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 2–3 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リソースが含まれています。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 2–3 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。

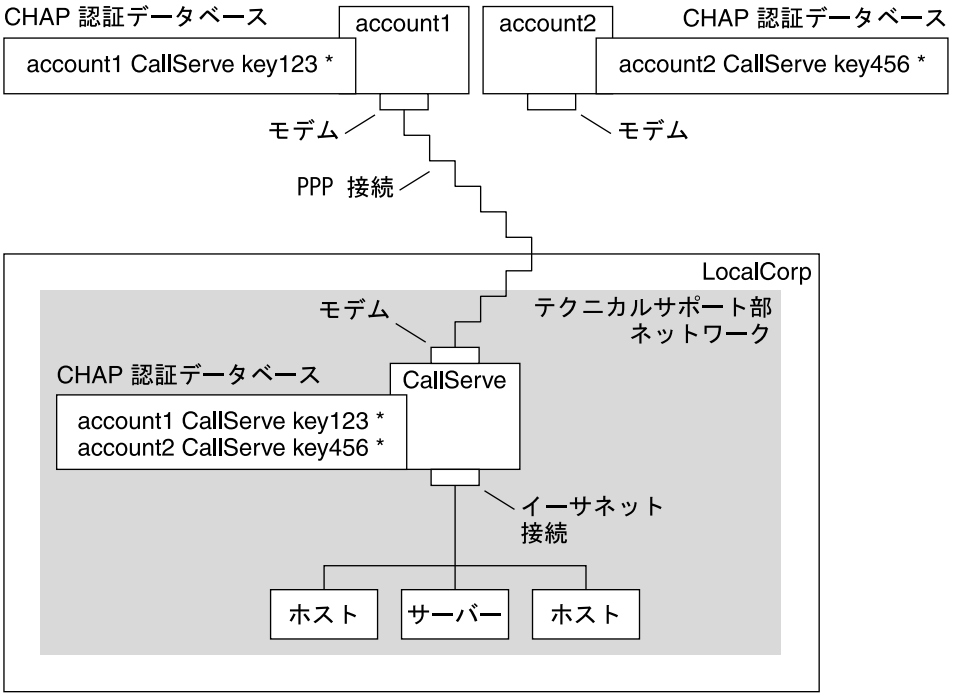
CHAP 認証による構成例

81 ページの「CHAP 認証の構成」でのタスクは、CHAP 認証の設定方法を示しています。手順では、42 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リモート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 2–4 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。

- 74 ページの「PAP 認証の構成」
- 81 ページの「CHAP 認証の構成」
- 138 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、33 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。

このセクションでは、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
- 51 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 トンネルの設定タスクについては、第 6 章「PPPoE トンネルの設定 (タスク)」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 2–6 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 `ipadm show-addr` 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。

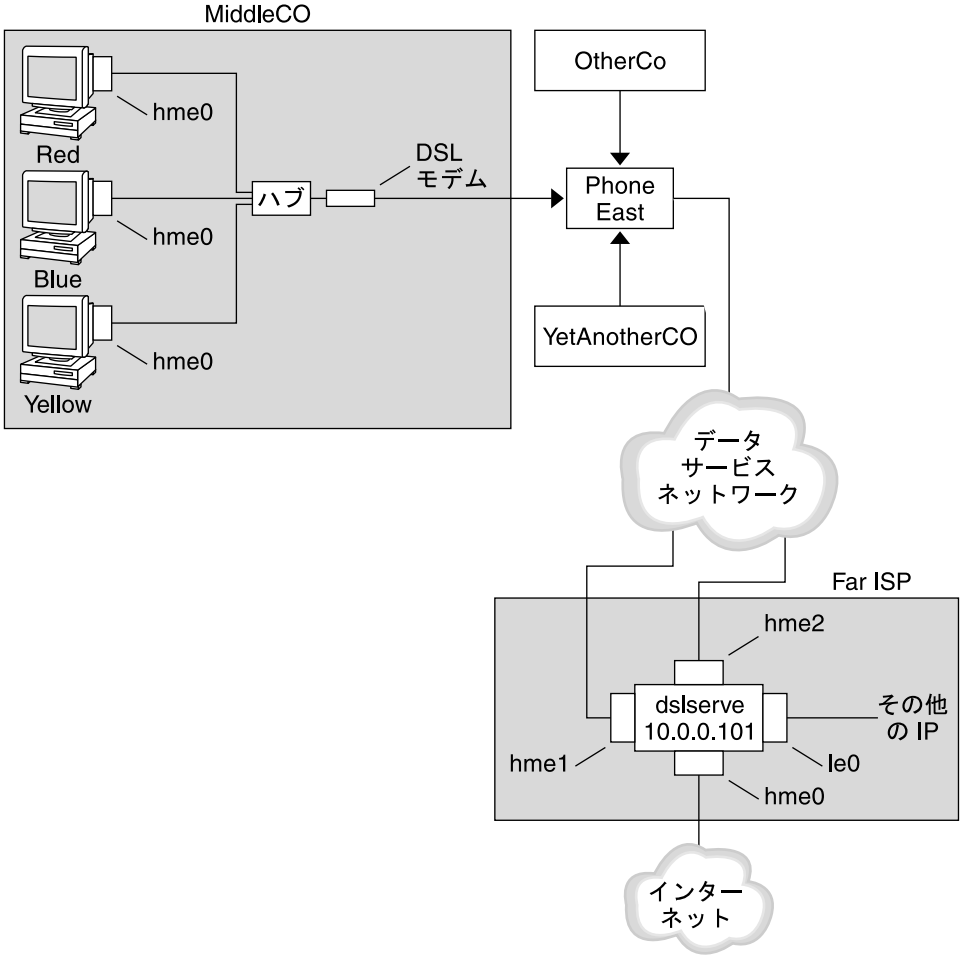
表 2–7 PPPoE アクセスサーバーの計画

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	`ipadm show-addr` 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

このセクションでは、第 6 章「PPPoE トンネルの設定 (タスク)」で説明するタスクの例として、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 2–5　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プライマリ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。

- 90 ページの「PPPoE クライアントの設定」
- 93 ページの「PPPoE アクセスサーバーの設定」
- 146 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 3 章

3 ダイアルアップ PPP リンクの設定 (タスク)

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タスク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タスク (タスクマップ)

ダイアルアップ PPP リンクの設定は、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 8–1 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タスクを示しています。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 3–1 ダイアルアップの PPP リンクの設定 (タスクマップ)

タスク	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	38 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	54 ページの「ダイアルアウトマシンの構成タスク (タスクマップ)」
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	61 ページの「ダイアルインサーバーの構成タスク (タスクマップ)」

表 3–1 ダイアルアップの PPP リンクの設定 (タスクマップ) (続き)

タスク	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	65 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

このセクションのタスクでは、ダイアルアウトマシンの構成方法について説明します。このタスクでは、図 2–1 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、会社でのタスクがあります。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タスク (タスクマップ)

表 3–2 ダイアルアウトマシンの設定 (タスクマップ)

タスク	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	38 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	56 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	57 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	58 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	59 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	65 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`	120 ページの「`/etc/ppp/options.tmpl` テンプレート」
`/etc/ppp/options.ttya.tmpl`	`/etc/ppp/options.`*ttyname*	122 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	130 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*	126 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タスクは、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 – モデムに適用するタスクは、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。

- ダイアルアウトマシンに Oracle Solaris リリースをインストールする
- 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、38 ページの「ダイアルアウトマシンを設定する前に」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、ほとんどのモデムは、Solaris PPP 4.0 用に正しく設定されて出荷されています。 次のリストは、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たものです。

   - **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ドキュメントを参照します。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、106 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で管理者になります。

   詳細は、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」を参照してください。

4. モデム方向を「発信専用」として指定します。

ダイアルアウトマシン上に通信を構成する

このセクションの手順では、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、56 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構成しておく必要があります。

次のタスクは、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、55 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で管理者になります。

詳細は、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」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、119 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、121 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 **62 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、127 ページの「ダイアルアップリンクのモデム速度の構成」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 Oracle 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、132 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、127 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 Oracle 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で管理者になります。

詳細は、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」を参照してください。

2 **DNS** およびネームサービススイッチサービスのリポジトリ情報を更新します。

```
# svccfg
svc:> select network/dns/client
svc:/network/dns/client> setprop config/domain = astring: "bigcompany.com"
svc:/network/dns/client> setprop config/nameserver = net_address: "10.10.111.15"
svc:/network/dns/client> addpropval config/nameserver "10.10.130.8"
svc:/network/dns/client> select network/dns/client:default
svc:/network/dns/client:default > refresh
svc:/network/dns/client:default > validate
svc:/network/dns/client:default > select system/name-service/switch
svc:/system/name-service/switch > setprop config/host = astring: "files dns"
 svc:/system/name-service/switch:default > select system/name-service/switch:default
svc:/system/name-service/switch:default > refresh
svc:/system/name-service/switch:default > validate
# svcadm enable network/dns/client
# svcadm refresh system/name-service/switch
```

3 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、56 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、115 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、61 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

このセクションのタスクでは、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。タスクでは、図 2–1 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タスク (タスクマップ)

表 3–3 ダイアルインサーバーの設定 (タスクマップ)

タスク	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	38 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	61 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	63 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	64 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。

- Oracle Solaris リリースをインストールする
- 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ドキュメントに従ってモデムのプログラムを作成します。
   ほかのヒントについては、56 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で管理者になります。
詳細は、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」を参照してください。

4 モデム方向を「着信専用」として指定します。

▼ 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関するヒントについては、127 ページの「ダイアルアップリンクのモデム速度の構成」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 **ttymon** を使用してシリアルポートをそのレートに固定します。

参照 関連情報の参照先は次のとおりです。

- 61 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)」
- 63 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。

このセクション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
- 61 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、144 ページの「呼び出し元の IP アドレス指定スキームの作成」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で管理者になります。

   詳細は、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」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。

   新しいユーザーを作成する手順については、『Oracle Solaris 11.1 のユーザーアカウントとユーザー環境の管理』の「CLI を使用したユーザーアカウントの設定と管理 (タスクマップ)」を参照してください。

3. 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。

- 63 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」
- 64 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

次のタスクは、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、115 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。

- 61 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
- 63 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で管理者になります。

詳細は、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」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、60 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、54 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、65 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。

次のタスクに進む前に、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。

- 54 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
- 60 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、59 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 (たとえば図 2–1 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 7 章「一般的な PPP 問題の解決 (タスク)」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、65 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 7 章「一般的な PPP 問題の解決 (タスク)」
- 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、115 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 4 章

専用回線 PPP リンクの設定 (タスク)

この章では、専用回線を使用した、ピア間での PPP リンクを構成する方法について説明します。主なセクションでは次の内容について説明します。

専用回線の設定 (タスクマップ)

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、または認証を構成する必要はありません。CSU/DSU の構成は複雑なので、これを構成する必要がある場合は、製造元のドキュメントを参照してください。

次の表のタスクマップでは、基本的な専用回線リンクの設定に必要なタスク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 4–1 専用回線リンクの設定 (タスクマップ)

タスク	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	42 ページの「専用回線のために収集する情報」
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	68 ページの「同期デバイスの構成方法」

表 4–1 専用回線リンクの設定 (タスクマップ)　(続き)

タスク	説明	参照先
3. 必要に応じて、インタフェースカードを構成する	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構成する	68 ページの「同期デバイスの構成方法」
4. リモートピアに関する情報に基づいて構成する	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	70 ページの「専用回線上のマシンの構成方法」
5. 専用回線への接続を開始する	ブート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構成する	70 ページの「専用回線上のマシンの構成方法」

専用回線上の同期デバイスの構成

このセクションのタスクでは、専用回線のトポロジに必要な機器を構成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、42 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)
- Oracle Solaris リリースがシステムにインストールされている
- システムに必要な同期インタフェースカード

▼ 同期デバイスの構成方法

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。
   製造元のドキュメントの手順に従います。

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ドキュメントの手順に従って、**CSU/DSU** を構成します。

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 を提供および構成する場合もあります。

4 必要に応じて、インタフェースのドキュメントの手順に従って、インタフェースカードを構成します。

インタフェースカードの構成時に、インタフェースの起動スクリプトを作成します。図 2–2 に示す専用回線構成では、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構成手順については、70 ページの「専用回線上のマシンの構成方法」を参照してください。

専用回線上のマシンの構成

このセクションのタスクでは、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、42 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構成する際の前提条件

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。

- 68 ページの「専用回線上の同期デバイスの構成」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構成する
- 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構成方法

1 ローカルマシン(ルーター)上で管理者になります。

詳細は、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」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 インタフェースを構成する
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

オプション	定義
/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、ブート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 /system/volatile/ppp-demand.pid ] &&
   /usr/bin/kill -s 0 ‘/bin/cat /system/volatile/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 /system/volatile/ppp-demand.pid] && /usr/bin/kill -s 0 ‘/bin/cat /system/volatile/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
/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 スクリプトがブート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、113 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、115 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 5 章

PPP 認証の設定 (タスク)

この章では、PPP 認証の設定タスクについて説明します。ここでは、次の内容を説明します。

- 74 ページの「PAP 認証の構成」
- 81 ページの「CHAP 認証の構成」

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構成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、専用回線に認証を構成することもできます。専用回線に認証を設定する場合は、この章のタスク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、33 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 138 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 (タスクマップ)

次のタスクマップに、PPP 認証に関連する作業を示します。

表 5–1 一般的な PPP 認証 (タスクマップ)

タスク	説明	参照先
PAP 認証を構成する	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	74 ページの「PAP 認証の設定 (タスクマップ)」
CHAP 認証を構成する	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	82 ページの「CHAP 認証の設定 (タスクマップ)」

PAP 認証の構成

このセクションでは、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、45 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 (タスクマップ)

次のタスクマップに、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 関連のタスクを示します。

表 5–2 PAP 認証についてのタスクマップ (ダイアルインサーバー)

タスク	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	44 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	75 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	75 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	77 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 5–3 PAP 認証についてのタスクマップ (ダイアルアウトマシン)

タスク	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	44 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	78 ページの「信頼できる呼び出し元に PAP 認証資格を構成する方法」
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	80 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 2–3 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、141 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 ユーザー名が UNIX ユーザー名と一致する必要はありません。標準の `/etc/ppp/pap-secrets` については、138 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によって指定される方法で、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で管理者になります。

詳細は、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」を参照してください。

4 **`/etc/ppp/pap-secrets`** ファイルを編集します。

このリリースは、`/etc/ppp` の `pap-secrets` ファイルを提供します。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が、オプション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 2–3 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` との接続が確立される場合に使用する 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。

- 77 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)」
- 78 ページの「信頼できる呼び出し元の PAP 認証の構成 (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)

このセクションのタスクでは、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、64 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で管理者になります。

詳細は、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」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 認証資格を構成する手順については、78 ページの「信頼できる呼び出し元の PAP 認証の構成 (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 認証の構成 (ダイアルアウトマシン)

このセクションでは、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、このセクションのタスクを指示することもできます。

信頼できる呼び出し元に PAP を構成するには、次の 2 つのタスクを実行します。

- 呼び出し元の PAP セキュリティー資格を構成します。
- 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構成します。

▼ 信頼できる呼び出し元に PAP 認証資格を構成する方法

ここでは、2 つ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 つは、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で管理者になります。

   詳細は、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」を参照してください。

   図 2–3 で紹介した PAP 構成のサンプルを使用して、user1 がダイアルアウトマシンを所有しているとします。

2. 呼び出し元の **pap-secrets** データベースを変更します。

   このリリースは、/etc/ppp/pap-secrets ファイルを提供します。このファイルには、役立つコメントが含まれていますが、オプションは含まれていません。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. ダイアルアウトマシン上で管理者になります。

   詳細は、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」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。

- 75 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)」
- 78 ページの「信頼できる呼び出し元に PAP 認証資格を構成する方法」

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)

次のタスクは、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。

この手順では、次のパラメータを使用して、図 2–3 で紹介した user2 が所有するダイアルアウトマシン上で、PAP 認証を構成します。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、57 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。この手順では、図 2–3 で示した user2 が所有するダイアルアウトマシンを構成します。

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 名はパスワードデータベースにあるそのユーザーの UNIX ユーザー名と同じである必要がある

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、59 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は、ピア myserver に関する PAP 要件を追加します。

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、65 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、138 ページの「パスワード認証プロトコル (PAP)」

CHAP 認証の構成

このセクションのタスクでは、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 2–4 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 (タスクマップ)

表 5–4 CHAP 認証についてのタスクマップ (ダイアルインサーバー)

タスク	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	83 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	83 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	84 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 5–5 CHAP 認証についてのタスクマップ (ダイアルアウトマシン)

タスク	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	83 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	86 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 認証の設定における最初のタスクは、`/etc/ppp/chap-secrets` ファイルの変更です。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、75 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)」で説明したような PPP `login` 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で管理者になります。

詳細は、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」を参照してください。

4 **/etc/ppp/chap-secrets** ファイルを変更します。

このリリースには、`/etc/ppp/chap-secrets` ファイルが含まれています。このファイルには、役立つコメントが含まれていますが、オプションは含まれていません。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。

- 83 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)」

- 84 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)」
- 85 ページの「信頼できる呼び出し元の CHAP 認証の構成 (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)

このセクションのタスクでは、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。

63 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 64 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 認証資格を構成する手順については、83 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 認証の構成 (ダイアルアウトマシン)

このセクションには、信頼できる呼び出し元のダイアルアウトマシンで、CHAP 認証を設定するタスクが含まれています。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 を構成する場合は、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` ファイル内の 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、このセクションで説明している CHAP 構成のタスクを指示します。

信頼できる呼び出し元に CHAP を構成するには、次の 2 つのタスクを実行します。

- 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構成します。

▼ 信頼できる呼び出し元に CHAP 認証資格を構成する方法

ここでは、2 つ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で管理者になります。

詳細は、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」を参照してください。

47 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。

このリリースには、`/etc/ppp/chap-secrets` ファイルが含まれています。このファイルには、役立つコメントが含まれていますが、オプションは含まれていません。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` が到達を試みているピアの名前です。`key123` は、`account1` と `CallServer` 間での接続に使用される CHAP シークレットです。

3 ダイアルアウトマシン上で管理者になります。

詳細は、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」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。

- 83 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)」
- 85 ページの「信頼できる呼び出し元に CHAP 認証資格を構成する方法」

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、141 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。次のタスクに従って、47 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構成します。

▼ 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の説明については、59 ページの「個々のピアとの接続を定義する方法」の同様の /etc/ppp/peers/myserver ファイル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、65 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 6 章

PPPoE トンネルの設定 (タスク)

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。

- 89 ページの「PPPoE トンネル設定の主なタスク (タスクマップ)」
- 90 ページの「PPPoE クライアントの設定」
- 93 ページの「PPPoE アクセスサーバーの設定」

ここでは、49 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、33 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タスク (タスクマップ)

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タスクを一覧表示します。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 6–1 PPPoE クライアントの設定 (タスクマップ)

タスク	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	91 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	91 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	57 ページの「シリアル回線を介した通信を定義する方法」

表 6–1 PPPoE クライアントの設定 (タスクマップ) (続き)

タスク	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	91 ページの「PPPoE アクセスサーバーピアを定義する方法」

表 6–2 PPPoE アクセスサーバーの設定 (タスクマップ)

タスク	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	93 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	63 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. (オプション) 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	95 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 Oracle Solaris 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 クライアント上で管理者になります。**

詳細は、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」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、147 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、148 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、124 ページの「`/etc/ppp/peers/`*peer-name* ファイル」を参照してください。

1 **PPPoE クライアント上で管理者になります。**

詳細は、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」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、51 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、156 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. **54 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。

121 ページの「 /etc/ppp/options.*ttyname* 構成ファイル」で説明する /etc/ppp/options.*ttyname* ファイル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、51 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。

- 90 ページの「PPPoE クライアントの設定」
- 146 ページの「DSL サポート用の PPPoE トンネルの作成」
- 第 7 章「一般的な PPP 問題の解決 (タスク)」
- 93 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 Ethernet 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構成する場合に行なってください。

1 アクセスサーバー上で管理者になります。

詳細は、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」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 51 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 ファイルには、図 2–5 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が一覧表示されています。

```
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、144 ページの「呼び出し元の IP アドレス指定スキームの作成」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、148 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 アクセスサーバー上で管理者になります。

詳細は、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」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。このタスクを実行する前に、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、51 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 93 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、74 ページの「PAP 認証の構成」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、146 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、101 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、90 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、78 ページの「信頼できる呼び出し元の PAP 認証の構成 (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、75 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 7 章

一般的な PPP 問題の解決 (タスク)

この章には、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が含まれています。次の項目について説明します。

- 98 ページの「PPP のトラブルシューティングのためのツール」
- 101 ページの「PPP および PPPoE 関連の問題の解決」
- 113 ページの「専用回線の問題の解決」
- 114 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、23 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 24 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 (タスクマップ)

次のタスク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 7–1 PPP のトラブルシューティング (タスクマップ)

タスク	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。	99 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。	100 ページの「PPP デバッグをオンに設定する方法」

表 7–1 PPP のトラブルシューティング (タスクマップ) (続き)

タスク	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。	101 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。	104 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。	105 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。	106 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。	107 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。	110 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。	113 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。	114 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。	111 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ます。次に、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。

このセクションでは、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。この章の残りのセクションでは、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管理者になります。

詳細は、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」を参照してください。

2 PPP に構成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 7–1 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 7–2 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

次のタスクは、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 管理者になります。

詳細は、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」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にする必要があります。

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、次のセクション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管理者になります。

詳細は、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」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ルーティングできない内部 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、リモート組織に問題を報告する必要があります。

8 ルーティングテーブルを調べます。

a. ローカルマシンとピアの両方でルーティングテーブルを確認します。

b. ルーティングテーブルで、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。

中間ルーターの構成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、103 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、101 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 7–2 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	ローカルホストのルーティング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	ローカルホストのルーティングテーブルに欠如している送信経路を追加する

表 7–2 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	ローカルホストのルーティング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	ローカルホストのルーティング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構成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 7–5 を参照してください。

1 ローカルマシン上で管理者になります。

詳細は、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」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、104 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、104 ページの「通信の問題を診断し解決する方法」の作業のログ出力に関連する症状を説明したものです。

表 7–3　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、あるいは構成に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 7–5 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 アドレスの構成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、モデム構成のエラー、不適切に構成された 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管理者になります。

詳細は、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」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、106 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、105 ページの「PPP 構成の問題を診断する方法」の作業のログ出力に関連する症状を説明したものです。

表 7–4 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 圧縮構成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。

モデムメーカーのドキュメントや 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. 100 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

次の手順を使用すると、chat からのデバッグ情報や一般的な問題の解決についてのヒントを取得できます。詳細は、108 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で管理者になります。
   詳細は、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」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、107 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 7–5 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構成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
`pppd debug` 出力は LCP がアクティブであ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

表 7–5 chat スクリプトの一般的な問題 (続き)

症状	問題	解決方法

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` の出力に `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
`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、回線に設定されていた元の速度を `/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、100 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 に構成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` はもともと構成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. 構成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. 100 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 7–6 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 メーカーのドキュメント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。
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構成によって生じた時間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 7–7 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 8 章

Solaris PPP 4.0 (リファレンス)

この章では、Solaris PPP 4.0 に関する詳細な概念情報を提供します。ここでは、次の内容を説明します。

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。このセクションでは、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 の構成は非常に柔軟です。PPP オプションを定義できる場所は次のとおりです。

- 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 8–1　PPP 構成ファイルとコマンドのサマリー

ファイルまたはコマンド	説明	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。	119 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。	121 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。	124 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。	124 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。	138 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。	142 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。	123 ページの「ダイアルインサーバーでの `$HOME/.ppprc` の構成」
`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。	116 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. (オプション) 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	ステータス	意味
domain	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	ステータス	意味
proxyarp	noproxyarp が指定されている場合、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。
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57 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、57 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、64 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、77 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、80 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、84 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* ファイルで構成できます。/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。それ以外の場合、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。

/etc/ppp/options.*ttyname* のダイアルインサーバーでの使用

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

/etc/ppp/options.*ttyname* のダイアルアウトマシンでの使用

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、57 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、64 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構成

このセクションでは、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 の構成

$HOME/.ppprc ファイルは、独自の PPP オプションを構成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 を構成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、62 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/.pprc の構成

$HOME/.pprc 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/.pprc も必要ありません。次を行う場合は、1 つ以上の .pprc 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 .pprc 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。

.pprc 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、.pprc の構成手順について 63 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構成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* ファイルで構成するのが最適です。

/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、noauth オプションをオーバーライドできません。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ションに使用される初期 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 ルート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ルートを使用します。
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、59 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、70 ページの「専用回線上のマシンの構成方法」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、80 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、86 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、90 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構成

モデムの構成で重要なのは、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ドキュメントを確認します。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、61 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。このセクションでは、モデムの構成および 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、起動時の構成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

このセクションでは、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。モデムの応答は即時であるべきです。
"" 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。
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

このリリースには、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。モデムの応答は即時であるべきです。
"" "AT&F1"	モデムを出荷時のデフォルトにリセットします。
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。

スクリプトの内容	意味
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。モデムの応答は即時であるべきです。
"" 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。

スクリプトの内容	意味
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、Oracle 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、58 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。モデムの応答は即時であるべきです。
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。

スクリプトの内容	意味
"" 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、ユーザーのステータスを知らせます。
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。
SAY "Logged in. Starting PPP on peer system.\n"	「Logged in. Starting PPP on peer system」という通知メッセージを表示して、ユーザーのステータスを知らせます。
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、63 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

次の chat スクリプトは、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ものです。

```
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。
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。

スクリプトの内容	意味
"" 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、187 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。chat ユーティリティーは、ユーザーの便宜を図るために提供されています。

▼ chat スクリプトを呼び出す方法 (タスク)

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 8–1 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、58 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

このセクションでは、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構成された 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

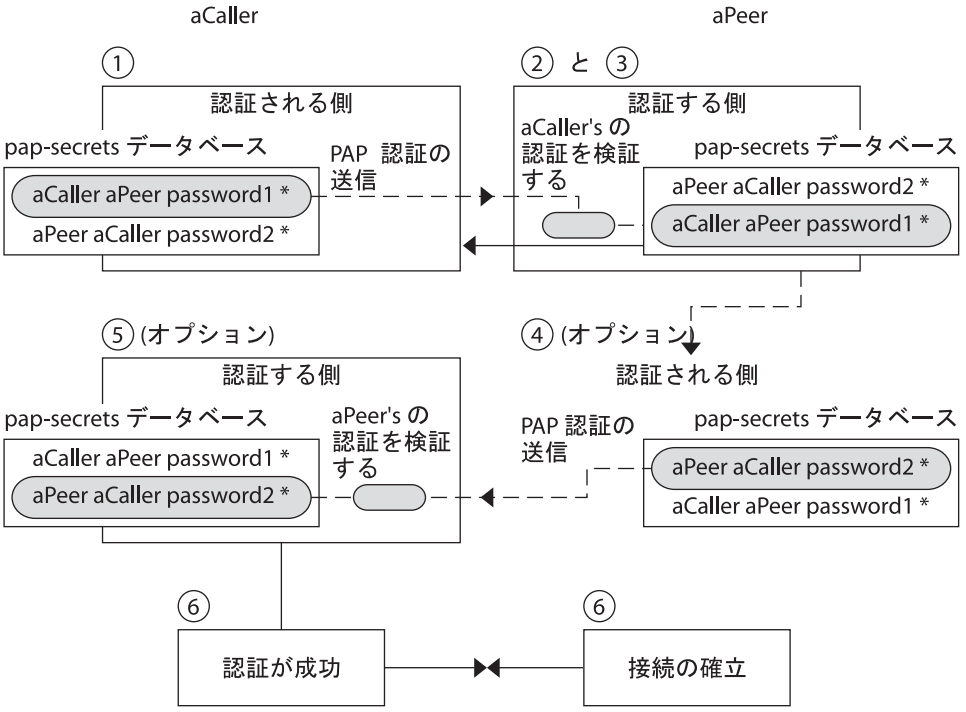
サーバー (認証する側) では、PAP パスワードは、次のどれかの方法で隠すことができます。

- `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 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 8–1　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 内で構成します。

/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。

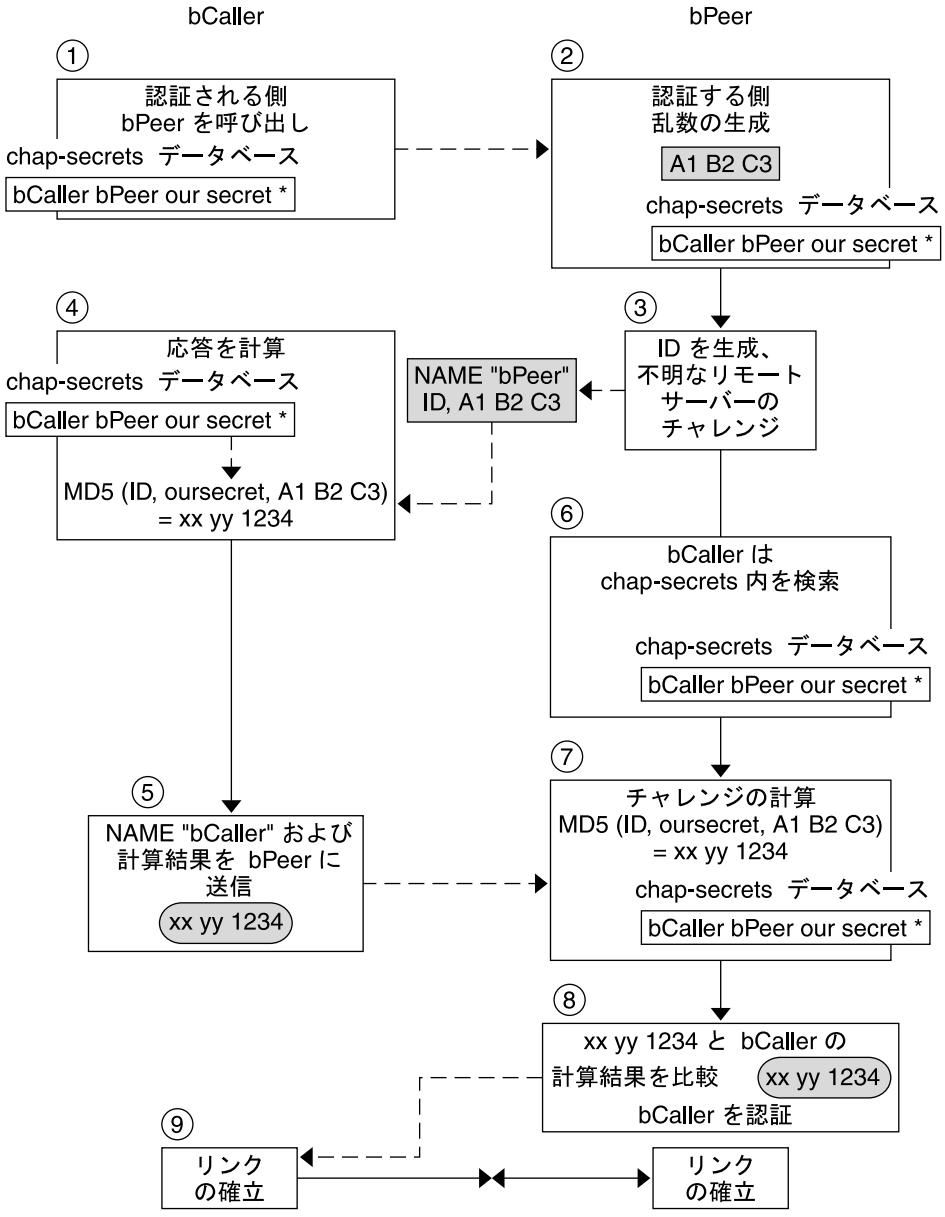
---

*　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 8–2 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ムの作成

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ムでは、/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、34 ページの「PPPoE の概要」を参照してください。

- PPPoE トンネルの設定タスクについては、第 6 章「PPPoE トンネルの設定 (タスク)」を参照してください。

このセクションでは、PPPoE コマンドおよびファイルについて詳しく説明します。サマリーを次の表に示します。

表 8–2 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	149 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	152 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	147 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 インタフェースを構成するためのコマンド	148 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	149 ページの「`/usr/lib/inet/pppoed` デーモン」

PPPoE のインタフェースを構成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、あらかじめ構成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 Oracle Solaris PPPoE クライアントおよび PPPoE アクセスサーバー上の Ethernet インタフェースを構成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 用に構成します。

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 はシステムのブート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 トンネルで使用される Ethernet インタフェースを ipadm コマンドと同様の方法で plumb します。ipadm とは異なり、2 つの Ethernet プロトコル番号が関係するため、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出力します。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、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。

- 148 ページの「/usr/sbin/sppptun コマンド」
- 149 ページの「/usr/lib/inet/pppoed デーモン」
- 149 ページの「/etc/ppp/pppoe ファイル」
- 152 ページの「/etc/ppp/pppoe.*device* ファイル」
- 155 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* ファイルで構成します。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 8–2　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記述します。/etc/ppp/pppoe. *device* には、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含まれています。/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、149 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

このセクションでは、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 8–3 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 8–4 アクセスサーバー用の /etc/ppp/options ファイル

```
     auth
     proxyarp
```

例 8–4 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オーバーライドします。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 オプションをオーバーライド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 8–5 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 8–6 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 8–7 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。

- 148 ページの「/usr/sbin/sppptun コマンド」
- 155 ページの「/usr/lib/inet/pppoec ユーティリティー」
- 155 ページの「pppoe.so 共有オブジェクト」
- 124 ページの「/etc/ppp/peers/*peer-name* ファイル」
- 119 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、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、91 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 8–8　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。

オプション	説明
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。
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 ルート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 9 章

9 Asynchronous Solaris PPP から Solaris PPP 4.0 への移行 (タスク)

Oracle Solaris OS の以前のバージョンでは、別の PPP 実装である Asynchronous 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。

- 159 ページの「asppp ファイルを変換する前に」
- 162 ページの「asppp2pppd 変換スクリプトの実行 (タスク)」

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、22 ページの「どのバージョンの Solaris PPP を使用すべきか」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

163 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。

```
#
ipadm create-if ipdptp0
ipadm create-addr -T static -a local=mojave,remote=gobi ipdptp0/ppaddr

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。

`ifpadm create-if ipdptp0`
  ipadm コマンドを実行し、ipdptp0 というインタフェースを作成する

`ipadm create-addr -T static -a local=mojave,remote=gobi ipdptp0/ppadd`
  ipadm 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 分間非アクティブな回線を終了する

`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

163 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

163 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

163 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 Oracle 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 Oraccle 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 (タスク)

`/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。

タスクの前提条件

次のタスクに進む前に、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 Oracle Solaris 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 10 章

10 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。

Oracle 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、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

Oracle 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実行します。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　リクエストされた転送 (uucp、uuto、uux) のステータスを表示する。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、ポーリングする時間を定義する
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

UUCP データベースは、171 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。171 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セクション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 11 章

UUCP の管理 (タスク)

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、Oracle Solaris OS が動作するマシンで UUCP を構成し保守するための、手順とトラブルシューティングについての情報が記載されています。

UUCP 管理 (タスクマップ)

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 11–1 UUCP 管理のタスクマップ

タスク	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	174 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	175 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	177 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	180 ページの「モデムまたは ACU の障害確認方法」

表 11–1 UUCP 管理のタスクマップ (続き)

タスク	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	180 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 管理者になります。

詳細は、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」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。Oracle 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 管理者になります。

詳細は、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」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.) のステータスを報告する。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出力する
- 結果のステータス情報を 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
- その日に収集したステータス情報のサマリーを、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、このセクション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. 管理者になります。

   詳細は、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」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 を実行する場合は、svc:/system/name-service/switch サービスの config/service が、nis の前に files を検査するようにしてください。config/service プロパティーが定義されていない場合は、config/default プロパティーを確認します。

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。このサービスのステータスは、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、『Oracle Solaris 11.1 でのサービスと障害の管理』の第 1 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。このセクションでは、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 のタスク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 ユーザー名の検証が行われます。

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、175 ページの「UUCP の起動方法」のセクションで述べたように、crontab ファイルが正しい場所に置かれているか確認する必要があります。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 管理者になります。

詳細は、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」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出力します。

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 管理者になります。

詳細は、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」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ステータス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 12 章

12 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、203 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 12–1 /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、184 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、185 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、186 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、186 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、187 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、187 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。183 ページの「UUCP /etc/uucp/Systems ファイル」を参照してください。例 12–1では、System-Name フィールドにリモートホスト 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間を指定します。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 12–1 では、Time フィールドに `Any` が示されており、これはホスト `Arabian` をいつでも呼び出せることを示します。

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、どのような時間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 12–2 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、197 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)。193 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 12–3 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 12–4 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 12–1 `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 12–1 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、192 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、193 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、193 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、193 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、194 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、次のセクションで説明しています。

/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、198 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。この命名スキームは、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 12–5 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、183 ページの「UUCP /etc/uucp/Systems ファイル」 を参照してください。

例 12–5 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。例:

```
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 Oracle 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構成を使用しませんが、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 12–6 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、197 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 12–7 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 (Example 12–9 で説明するように、例 12–9 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。192 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 12–8 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、183 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 12–9 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、183 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 12–2 /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、もっともよく使用される。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。

191 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 12–10 /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 12–11 /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 12–3 /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、183 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、アウトバウンドの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、インバウンドの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、アウトバウンドとインバウンド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、このセクション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 12–4　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出力でき、cu リストは uuname -C コマンドを使用して出力できます。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、183 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、211 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 のデフォルトをオーバーライドして、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。この検証処理では、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アではなくなります。

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検証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間が入ります。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オーバーライドできます。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、もっとも優先順位が高いのは A、もっと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 12–5　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

このセクションでは、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 12–6 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を対象とする作業 (ファイル転送やリモートコマンド実行など) 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
- 転送の完了について通知され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

このセクションには、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 12–7 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 12–7 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 12–8 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	ステータスは良好
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 12–8 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、220 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ステータス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 12–9 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 12–9 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。

索引

D

E

F

G

P

V

W

X

あ

い

え

お

す

せ

た

ち

て

と

ゆ

り

れ

ろ