

Oracle® Solaris 11.2 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

ORACLE®

Part No: E53886
2014 年 7 月

Copyright © 2002, 2014, Oracle and/or its affiliates. All rights reserved.

このソフトウェアおよび関連ドキュメントの使用と開示は、ライセンス契約の制約条件に従うものとし、知的財産に関する法律により保護されています。ライセンス契約で明示的に許諾されている場合もしくは法律によって認められている場合を除き、形式、手段に関係なく、いかなる部分も使用、複写、複製、翻訳、放送、修正、ライセンス供与、送信、配布、発表、実行、公開または表示することはできません。このソフトウェアのリバース・エンジニアリング、逆アセンブル、逆コンパイルは互換性のために法律によって規定されている場合を除き、禁止されています。

ここに記載された情報は予告なしに変更される場合があります。また、誤りが無いことの保証はいたしかねます。誤りを見つけた場合は、オラクル社までご連絡ください。

このソフトウェアまたは関連ドキュメントを、米国政府機関もしくは米国政府機関に代わってこのソフトウェアまたは関連ドキュメントをライセンスされた者に提供する場合は、次の通知が適用されます。

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

このソフトウェアもしくはハードウェアは様々な情報管理アプリケーションでの一般的な使用のために開発されたものです。このソフトウェアもしくはハードウェアは、危険が伴うアプリケーション（人的傷害を発生させる可能性があるアプリケーションを含む）への用途を目的として開発されていません。このソフトウェアもしくはハードウェアを危険が伴うアプリケーションで使用する場合、安全に使用するために、適切な安全装置、バックアップ、冗長性（redundancy）、その他の対策を講じることは使用者の責任となります。このソフトウェアもしくはハードウェアを危険が伴うアプリケーションで使用したことに起因して損害が発生しても、オラクル社およびその関連会社は一切の責任を負いかねます。

OracleおよびJavaはOracle Corporationおよびその関連企業の登録商標です。その他の名称は、それぞれの所有者の商標または登録商標です。

Intel, Intel Xeonは、Intel Corporationの商標または登録商標です。すべてのSPARCの商標はライセンスをもとに使用し、SPARC International, Inc.の商標または登録商標です。AMD, Opteron, AMDロゴ, AMD Opteronロゴは、Advanced Micro Devices, Inc.の商標または登録商標です。UNIXは、The Open Groupの登録商標です。

このソフトウェアまたはハードウェア、そしてドキュメントは、第三者のコンテンツ、製品、サービスへのアクセス、あるいはそれらに関する情報を提供することがあります。オラクル社およびその関連会社は、第三者のコンテンツ、製品、サービスに関して一切の責任を負わず、いかなる保証もいたしません。オラクル社およびその関連会社は、第三者のコンテンツ、製品、サービスへのアクセスまたは使用によって損失、費用、あるいは損害が発生しても一切の責任を負いかねます。

目次

このドキュメントの使用方法	11
1 Solaris Point-to-Point Protocol 4.0 について	13
Solaris PPP 4.0 の基本情報	13
Solaris PPP 4.0 の互換性	14
どのバージョンの Solaris PPP を使用すべきか	14
PPP の詳細情報	16
PPP 構成と用語	17
ダイヤルアップ PPP の概要	18
専用回線 PPP の概要	21
PPP 認証	23
認証する側と認証される側	24
PPP の認証プロトコル	25
PPP 認証を使用する理由	25
PPPoE による DSL ユーザーのサポート	25
PPPoE の概要	26
PPPoE の構成要素	26
PPPoE トンネルのセキュリティ	28
2 ポイントツーポイントプロトコルリンクの計画	29
全体的な PPP 計画 (タスマップ)	29
ダイヤルアップ PPP リンクの計画	30
ダイヤルアウトマシンを設定する前に	30
ダイヤルインサーバーを設定する前に	31
ダイヤルアップ PPP の構成例	31
ダイヤルアップ PPP の詳細情報	33
専用回線リンクの計画	33
専用回線リンクを設定する前に	33
専用回線リンクの構成例	34
専用回線の詳細情報	36

リンクへの認証計画	36
PPP 認証を設定する前に	37
PPP の認証構成例	37
認証の詳細情報	40
PPPoE トンネルを介した DSL サポートの計画	41
PPPoE トンネルを設定する前に	41
PPPoE トンネルの構成例	43
PPPoE の詳細情報	44
3 ダイアルアップポイントツーポイントプロトコルリンクの設定	45
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)	45
ダイアルアウトマシンの構成	46
ダイアルアウトマシンの構成タスク (タスクマップ)	46
ダイアルアップ PPP のテンプレートファイル	46
ダイアルアウトマシン上にデバイスを構成する	47
▼ 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)	47
ダイアルアウトマシン上に通信を構成する	48
▼ シリアル回線を介した通信を定義する方法	49
▼ ピアを呼び出すための命令群を作成する方法	50
▼ 個々のピアとの接続を定義する方法	51
ダイアルインサーバーの構成	53
ダイアルインサーバーの構成タスク (タスクマップ)	53
ダイアルインサーバーにデバイスを構成する	53
▼ 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)	54
▼ モデム速度を設定する方法	54
ダイアルインサーバーのユーザーを設定する	55
▼ ダイアルインサーバーのユーザーを構成する方法	55
ダイアルインサーバーを介した通信を構成する	56
▼ 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)	56
ダイアルインサーバーの呼び出し	57
▼ ダイアルインサーバーの呼び出し方法	58
4 専用回線ポイントツーポイントプロトコルリンクの設定	61
専用回線の設定 (タスクマップ)	61
専用回線上の同期デバイスの構成	62
同期デバイスを設定する際の前提条件	62
▼ 同期デバイスの構成方法	62
専用回線上のマシンの構成	63
専用回線上のローカルマシンを構成する際の前提条件	63

▼ 専用回線上のマシンの構成方法	64
5 ポイントツーポイントプロトコル認証の設定	67
PPP 認証の構成 (タスクマップ)	67
PAP 認証の構成	68
PAP 認証の設定 (タスクマップ)	68
ダイアルインサーバーに PAP 認証を構成する	69
▼ PAP 資格データベースの作成方法 (ダイアルインサーバー)	69
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)	71
▼ 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)	71
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)	72
▼ 信頼できる呼び出し元に PAP 認証資格を構成する方法	73
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)	74
▼ 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)	74
CHAP 認証の構成	76
CHAP 認証の設定 (タスクマップ)	76
ダイアルインサーバーに CHAP 認証を構成する	77
▼ CHAP 資格データベースの作成方法 (ダイアルインサーバー)	77
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)	78
▼ 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)	79
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)	79
▼ 信頼できる呼び出し元に CHAP 認証資格を構成する方法	80
CHAP を構成ファイルに追加する (ダイアルアウトマシン)	81
▼ 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)	81
6 PPP Over Ethernet トンネルの設定	83
PPPoE トンネル設定の主なタスク (タスクマップ)	83
PPPoE クライアントの設定	84
PPPoE クライアント設定の前提条件	84
▼ PPPoE クライアントのインタフェースを構成する方法	85
▼ PPPoE アクセスサーバーピアを定義する方法	85
PPPoE アクセスサーバーの設定	87
▼ PPPoE アクセスサーバーの設定方法	87
▼ 既存の /etc/ppp/pppoe ファイルを変更する方法	89
▼ 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	89

7 ポイントツーポイントプロトコルの一般的な問題のトラブルシューティング	93
PPP 問題の解決 (タスクマップ)	93
PPP のトラブルシューティングのためのツール	94
▼ pppd から診断情報を取得する方法	95
▼ PPP デバッグをオンに設定する方法	96
PPP および PPPoE 関連の問題の解決	97
▼ ネットワークの問題を診断する方法	98
PPP に影響を与える一般的なネットワークの問題	100
▼ 通信の問題を診断し解決する方法	101
PPP に影響を与える一般的な通信の問題	101
▼ PPP 構成の問題を診断する方法	102
一般的な PPP 構成の問題	102
▼ モデムの問題を診断する方法	103
▼ chat スクリプトのデバッグ情報を取得する方法	104
chat スクリプトの一般的な問題	105
▼ シリアル回線の速度の問題を診断して解決する方法	107
▼ PPPoE の診断情報を取得する方法	108
専用回線の問題の解決	110
認証の問題の診断と解決	111
8 Solaris PPP 4.0 (リファレンス)	113
ファイルおよびコマンド行での PPP オプションの使用	113
PPP オプションを定義する場所	113
PPP オプションの処理方法	114
PPP 構成ファイルにおける特権のしくみ	116
/etc/ppp/options 構成ファイル	118
/etc/ppp/options.ttyname 構成ファイル	119
ユーザー独自のオプションの構成	122
ダイアルインサーバーでの \$HOME/.ppprc の構成	122
ダイアルアウトマシンでの \$HOME/.ppprc の構成	122
ダイアルインサーバーと通信するための情報の指定	123
/etc/ppp/peers/peer-name ファイル	123
/etc/ppp/peers/myisp.tmpl テンプレートファイル	124
/etc/ppp/peers/peer-name ファイルの例 (参照先)	125
ダイアルアップリンクのモデム速度の構成	126
ダイアルアップリンクでの会話の定義	126
chat スクリプトの内容	126
chat スクリプトの例	127
chat スクリプトの呼び出し	134

▼ chat スクリプトを呼び出す方法 (タスク)	135
実行可能な chat ファイルの作成	136
▼ 実行可能な chat プログラムを作成する方法	136
接続時の呼び出し元の認証	136
パスワード認証プロトコル (PAP)	136
チャレンジハンドシェイク認証プロトコル (CHAP)	140
呼び出し元の IP アドレス指定スキームの作成	143
呼び出し元への IP アドレスの動的割り当て	143
呼び出し元への IP アドレスの静的割り当て	144
sppp ユニット番号による IP アドレスの割り当て	145
DSL サポート用の PPPoE トンネルの作成	145
PPPoE のインタフェースを構成するためのファイル	146
PPPoE アクセスサーバーのコマンドとファイル	148
PPPoE クライアントのコマンドとファイル	154
9 非同期 Solaris PPP から Solaris PPP 4.0 への移行 (タスク)	157
asppp ファイルを変換する前に	157
/etc/asppp.cf 構成ファイルの例	158
/etc/uucp/Systems ファイルの例	158
/etc/uucp/Devices ファイルの例	159
/etc/uucp/Dialers ファイルの例	160
asppp2pppd 変換スクリプトの実行 (タスク)	160
タスクの前提条件	160
▼ asppp から Solaris PPP 4.0 に変換する方法	161
▼ 変換結果を表示する方法	161
10 UNIX-to-UNIX Copy Program について	165
UUCP のハードウェア構成	165
UUCP ソフトウェア	166
UUCP デーモン	166
UUCP 管理プログラム	167
UUCP ユーザープログラム	168
UUCP データベースファイル	169
UUCP データベースファイルの構成	170
11 UNIX-to-UNIX Copy Program の管理	171
UUCP 管理 (タスクマップ)	171
UUCP のログインの追加	172

▼ UUCP ログインの追加方法	172
UUCP の起動	173
▼ UUCP の起動方法	173
uudemon.poll シェルスクリプト	174
uudemon.hour シェルスクリプト	174
uudemon.admin シェルスクリプト	174
uudemon.cleanup シェルスクリプト	175
TCP/IP を介した UUCP の実行	175
▼ TCP/IP 用 UUCP の起動方法	175
UUCP のセキュリティーと保守	176
UUCP のセキュリティーの設定	177
定期的な UUCP の保守	177
UUCP のトラブルシューティング	178
▼ モデムまたは ACU の障害確認方法	178
▼ 送信に関するデバッグ方法	179
UUCP /etc/uucp/Systems ファイルの検査	180
UUCP エラーメッセージの検査	180
基本情報の検査	181
12 UNIX - to-UNIX Copy Program ファイル	183
UUCP /etc/uucp/Systems ファイル	183
/etc/uucp/Systems ファイルの System-Name フィールド	184
/etc/uucp/Systems ファイルの Time フィールド	185
/etc/uucp/Systems ファイルの Type フィールド	186
/etc/uucp/Systems ファイルの Speed フィールド	186
/etc/uucp/Systems ファイルの Phone フィールド	187
/etc/uucp/Systems ファイルの Chat-Script フィールド	187
Chat スクリプトを使用したダイアルバックの有効化	189
/etc/uucp/Systems ファイルでのハードウェアフロー制御	190
/etc/uucp/Systems ファイルでのパリティの設定	190
UUCP /etc/uucp/Devices ファイル	191
/etc/uucp/Devices ファイルの Type フィールド	192
/etc/uucp/Devices ファイルの Line フィールド	193
/etc/uucp/Devices ファイルの Line2 フィールド	193
/etc/uucp/Devices ファイルの Class フィールド	194
/etc/uucp/Devices ファイルの Dialer-Token-Pairs フィールド	194
/etc/uucp/Devices ファイルの Dialer-Token-Pairs フィールドの構造	195
/etc/uucp/Devices ファイル内のプロトコル定義	197

UUCP /etc/uucp/Dialers ファイル	198
/etc/uucp/Dialers ファイルでのハードウェアフロー制御の有効化	201
/etc/uucp/Dialers ファイルでのパリティの設定	202
その他の基本的な UUCP 構成ファイル	202
UUCP /etc/uucp/Dialcodes ファイル	202
UUCP /etc/uucp/Sysfiles ファイル	204
UUCP /etc/uucp/Sysname ファイル	205
UUCP /etc/uucp/Permissions ファイル	205
UUCP 構造のエントリ	206
UUCP の考慮事項	206
UUCP REQUEST オプション	207
UUCP SENDFILES オプション	207
UUCP MYNAME オプション	208
UUCP READ オプションと WRITE オプション	208
UUCP NOREAD オプションと NOWRITE オプション	209
UUCP CALLBACK オプション	209
UUCP COMMANDS オプション	210
UUCP VALIDATE オプション	211
UUCP OTHER 用の MACHINE エントリ	213
UUCP の MACHINE エントリと LOGNAME エントリの結合	213
UUCP の転送	214
UUCP /etc/uucp/Poll ファイル	214
UUCP /etc/uucp/Config ファイル	215
UUCP /etc/uucp/Grades ファイル	215
UUCP User-job-grade フィールド	216
UUCP System-job-grade フィールド	216
UUCP Job-size フィールド	217
UUCP Permit-type フィールド	217
UUCP ID-list フィールド	218
その他の UUCP 構成ファイル	218
UUCP /etc/uucp/Devconfig ファイル	218
UUCP /etc/uucp/Limits ファイル	219
UUCP remote.unknown ファイル	219
UUCP の管理ファイル	220
UUCP のエラーメッセージ	222
UUCP の ASSERT エラーメッセージ	222
UUCP の STATUS エラーメッセージ	223
UUCP の数値エラーメッセージ	224

索引 227

このドキュメントの使用方法

- 概要 – PPP および UUCP サービスを有効にしてシリアルネットワークングを提供する方法について説明します。
- 対象読者 – システム管理者。
- 前提知識 – ネットワーク管理の基本的なスキルと一部の高度なスキル。

製品ドキュメントライブラリ

この製品の最新情報や既知の問題は、ドキュメントライブラリ (<http://www.oracle.com/pls/topic/lookup?ctx=E36784>) に含まれています。

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。

フィードバック

このドキュメントに関するフィードバックを <http://www.oracle.com/goto/docfeedback> からお聞かせください。

◆◆◆ 第 1 章

Solaris Point-to-Point Protocol 4.0 について

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。このバージョンの PPP では、PPT 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。

次の内容について説明します。

- [13 ページの「Solaris PPP 4.0 の基本情報」](#)
- [17 ページの「PPP 構成と用語」](#)
- [23 ページの「PPP 認証」](#)
- [25 ページの「PPPoE による DSL ユーザーのサポート」](#)

Solaris PPP 4.0 の基本情報

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である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。

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せん。

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 の高度に構成可能な機能を提供します。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への 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「非同期 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 を設定するには、asppp.cf 構成ファイル、3 つの UUCP ファイル、および ipadm コマンドを構成する必要があります。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 コマンドを発行する必要があります。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、このリリースで提供される変換スクリプトを使用できます。詳細は、[161 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 に関する情報が記載された資料は、印刷物または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 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イク認証プロトコル (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 の実装の技術的な詳細については、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。

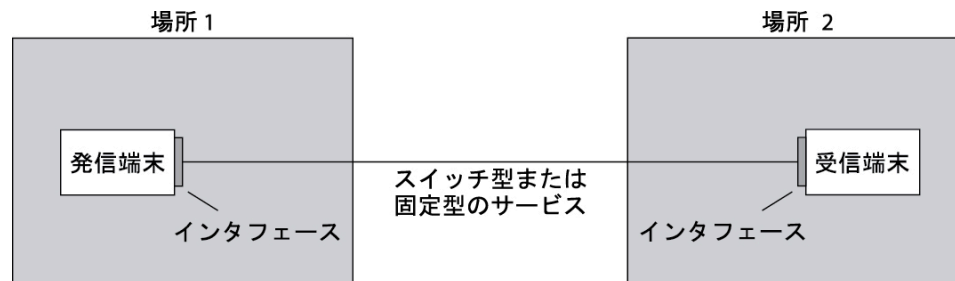
PPP 構成と用語

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 はさまざまな構成をサポートします。

- スイッチ型のアクセス構成 (ダイヤルアップ)
- 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。

- 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る 2 つのマシンです。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です。

ダイヤルアップ PPP の概要

もっともよく使用される PPP 構成は、*ダイヤルアップリンク*です。ダイヤ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ヤルアップして接続を確立し、PPP を実行します。ダイヤ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。

一般的なダイヤ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。別のダイヤ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。

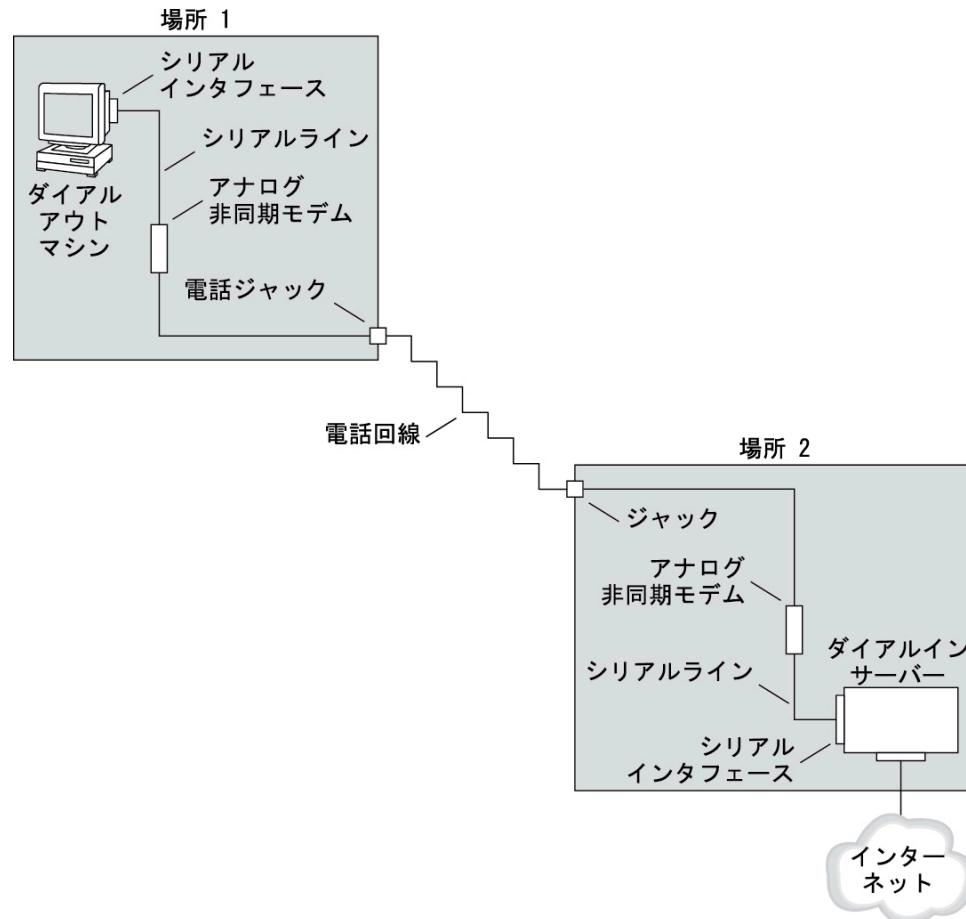
このガイドでは、ダイヤルアップ接続を開始するローカルピアは、*ダイヤルアウトマシン*と呼びます。着呼を受信するピアは、*ダイヤルインサーバー*と呼びます。このマシンは実際にはダイヤルアウトマシンがターゲットにするマシンに過ぎず、真の意味でのサーバーではない場合もあります。

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。ダイヤルインサーバーは、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。ダイヤルインサーバーという用語は、ダイヤルインマシンが複数のダイヤ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。それでもダイヤルインサーバーは、現実には、ダイヤルアウトマシンのターゲットピアにすぎません。

ダイヤルアップ 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ヤルアップ PPP リンク



リンクのダイヤルアウト側 (場所 1) の構成は、次の要素から成ります。

- ダイヤルアウトマシン。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。
- ダイヤルアウトマシン上のシリアルインターフェース。`/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ーフェースです。
- 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。

リンクのダイヤルイン側 (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
- ダイヤルインサーバー上のシリアルインタフェース (着呼には ttya または ttyb)
- ダイヤルインサーバー。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

ダイヤルアウトマシンで 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、[133 ページの「外部 ISDN TA 用 chat スクリプト」](#)を参照してください。

ダイヤルアップ通信中の動作

ダイヤルアウトとダイヤ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。ダイヤ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. ダイヤ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. ダイヤルアウトマシンは 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、ダイヤ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ヤルして、ダイヤルインサーバー側のモデムと電話接続を確立します。

ダイヤルアウトマシンが、モデムとダイヤ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。ダイヤルアウトマシンは、必要に応じて、ダイヤ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. ダイヤルインサーバーに接続されているモデムは、ダイヤ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、ダイヤ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。

ダイヤルアウトマシンでは、ダイヤルインサーバーを通し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。

ダイヤルアップリンクと専用回線リンクの比較

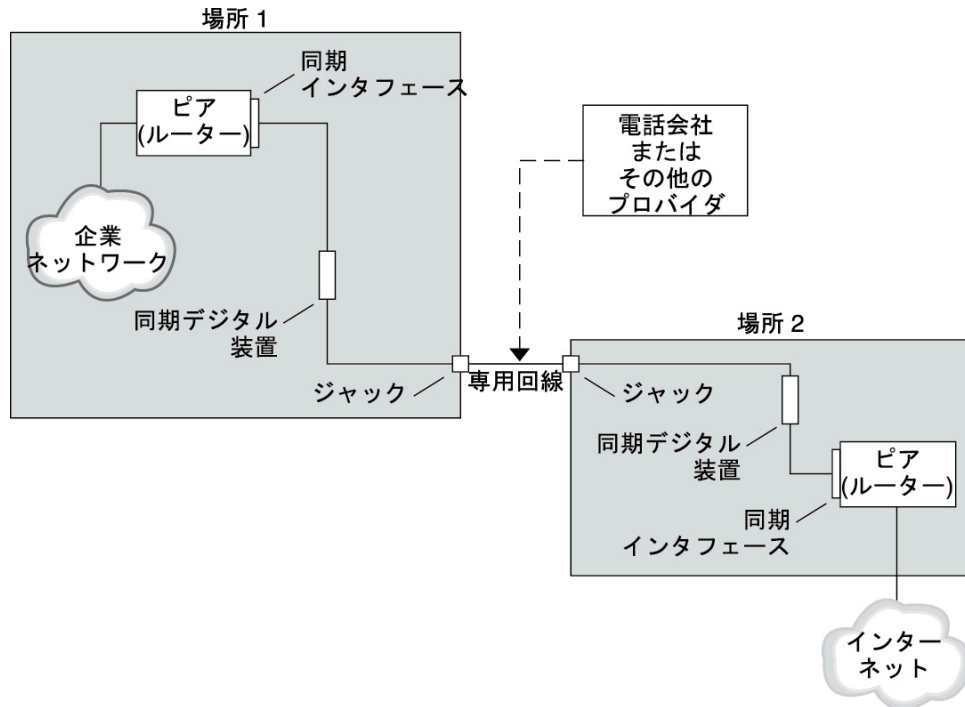
ダイヤルアップと専用回線のリンクはともに、通信媒体で接続されている 2 つのピアから成っています。次の表は、2 つのリンクタイプの相違をまとめています。

専用回線	ダイヤルアップ回線
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。
同期ユニットを必要とします。	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 同期デジタルユニット**、これは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**、これは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、ピアは互いにダイヤ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、ダイヤ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ため、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限り、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、ダイヤ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、ダイヤルアップリンクに PPP 認証を構成します。呼び出しが開始されると、ダイヤルアウトマシンが認証される側になります。ダイヤ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。

一部のダイヤルアウトマシンには、ダイヤ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、ダイヤルアウトマシンは認証する側になります。

注記 - 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イク認証プロトコル (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、[136 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、[140 ページの「チャレンジハンドシェイク認証プロトコル \(CHAP\)」](#)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。

注記 - 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。
- 会社のダイヤ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

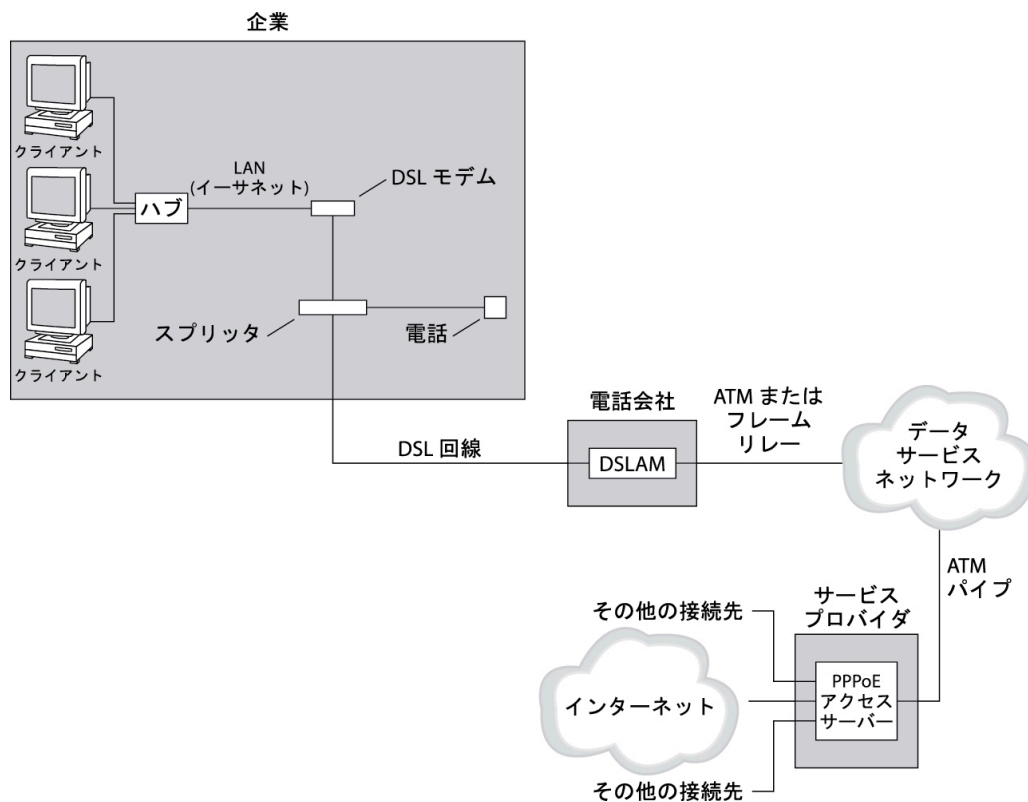
PPPoE は、RedBack Networks が開発した独自のプロトコルです。PPPoE は、別バージョンの標準 PPP ではなく検出プロトコルです。PPPoE のシナリオでは、最初に PPP 通信を開始するマシンが、PPPoE を実行しているピアを検出する必要があります。PPPoE プロトコルは、Ethernet ブロードキャストパケットを使ってピアを検出します。

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。Solaris PPP 4.0 は PPPoE を使用して、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、[145 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。

注記 - 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「基本的なアナログダイヤルアップ PPP リンク」](#)で紹介したダイヤ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

ポイントツーポイントプロトコルリンクの計画

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「ポイントツーポイントプロトコルリンクの計画」に続くタスクに関する章では、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。

- 30 ページの「ダイヤルアップ PPP リンクの計画」
- 33 ページの「専用回線リンクの計画」
- 36 ページの「リンクへの認証計画」
- 41 ページの「PPPoE トンネルを介した DSL サポートの計画」

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	手順の参照先
ダイヤルアップ PPP リンクを計画します	ダイヤルアウトマシンまたはダイヤルインサーバーの設定に必要な情報を収集します	30 ページの「ダイヤルアップ 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	33 ページの「専用回線リンクの計画」

タスク	説明	手順の参照先
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	36 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	41 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「ダイアルアップポイントツーポイントプロトコルリンクの設定」](#) で使用されるリンクの例の説明

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、[18 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。

注記 - 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、[36 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、[41 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	アクション
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。

注記 - 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、[36 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、[41 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	アクション
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	55 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

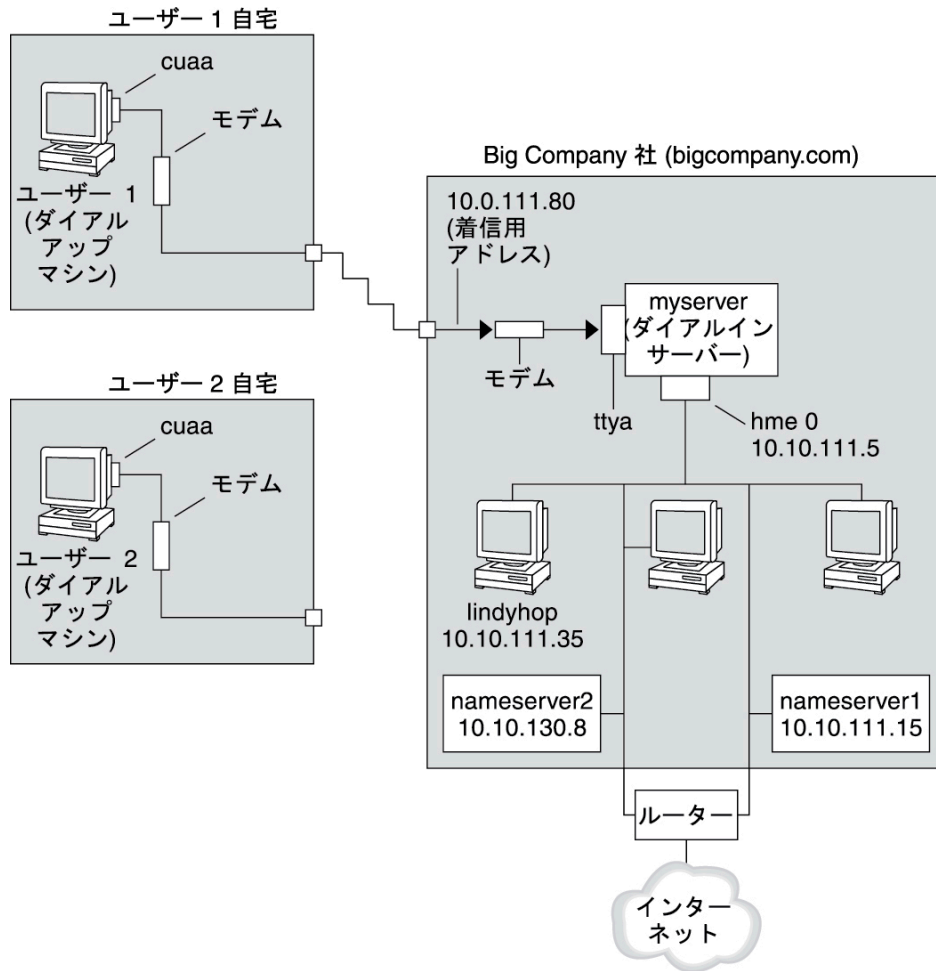
[第3章「ダイアルアップポイントツーポイントプロトコルリンクの設定」](#)で説明するタスクでは、小企業が従業員が週に数日自宅で作業できるようにするための要件を実施します。一部の従業員は、ホームマシンに Oracle Solaris OS が必要になります。また、社内イントラネット上にある作業マシンにリモートログインすることも必要になります。

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ポリシーには必要ありません。

次の図は、[第3章「ダイアルアップポイントツーポイントプロトコルリンクの設定」](#)で設定したリンクを示します。

図 2-1 ダイヤ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ムを介して Big Company 社のイントラネットにダイヤルアウトしています。もう 1 台のホストが Big Company 社にダイヤ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 社のダイヤ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。ダイヤルアウトマシンは、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。

ダイヤルアップ PPP の詳細情報

次を参照してください。

- ダイヤルアウトマシンを設定する手順については、[表3-2「ダイヤルアウトマシンの設定 \(タスクマップ\)」](#)を参照してください。
- ダイヤルインマシンを設定するには、[表3-3「ダイヤルインサーバーの設定 \(タスクマップ\)」](#)を参照してください。
- ダイヤルアップリンクの概要については、[18 ページの「ダイヤルアップ 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、[113 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「専用回線の構成例」](#)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、[21 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「専用回線ポイントツーポイントプロトコルリンクの設定」](#)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

リンク媒体のほかに、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	アクション
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するか、リモートピアのシステム管理者に問い合わせます。この情報は、2 つのピア間で IP アドレスがネゴシエートされない場合にだけ必要です。
(任意) リモートピアの名前	サービスプロバイダのドキュメントを参照するか、リモートピアのシステム管理者に問い合わせます。
(任意) リンクの種類	サービスプロバイダのドキュメントを参照するか、リモートピアのシステム管理者に問い合わせます。
(任意) リモートピアで使用する圧縮	サービスプロバイダのドキュメントを参照するか、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

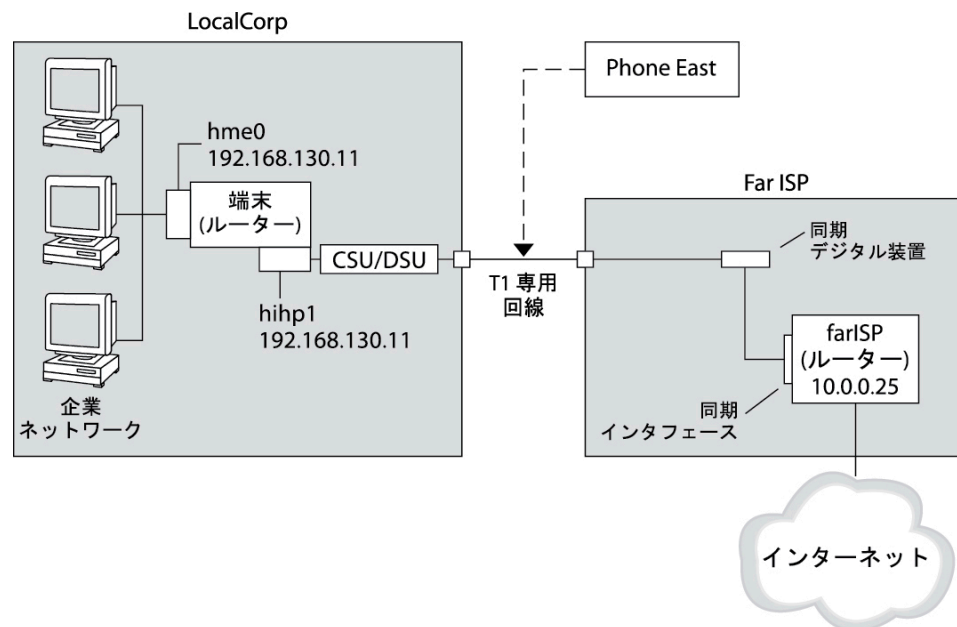
第4章「専用回線ポイントツーポイントプロトコルリンクの設定」のタスクは、従業員にインターネットアクセスを提供するという中堅規模の組織 (LocalCorp) の目標を実現する方法を示しています。現在、従業員のコンピュータは、会社の私設イントラネットに接続されています。

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 フア

イルの構成を終了したあとで、管理者が `/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「専用回線ポイントツーポイントプロトコルリンクの設定」](#)
- [21 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について説明します。[第5章「ポイントツーポイントプロトコル認証の設定」](#)では、自分のサイトで PPP 認証を実装するためのタスクを説明します。

PPP には、PAP と CHAP の 2 種類の認証があります。PAP の詳細は、[136 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、[140 ページの「チャレンジハンドシェイク認証プロトコル \(CHAP\)」](#)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、ダイヤルインマシンまたは呼び出し側のダイヤ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、[25 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「PAP 認証のシナリオ \(自宅で仕事する\) の例」](#) および [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」](#) に示されている認証事例の説明

認証の設定に関するタスクについては、[第5章「ポイントツーポイントプロトコル認証の設定」](#)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	手順の参照先
ダイヤルアップリンクの構成タスク	第3章「ダイヤルアップポイントツーポイントプロトコルリンクの設定」。
リンクのテストタスク	第7章「ポイントツーポイントプロトコルの一般的な問題のトラブルシューティング」。
サイトのセキュリティ要件	会社のセキュリティポリシー。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	25 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、136 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「ポイントツーポイントプロトコル認証の設定」の手順で使用されている認証シナリオの例について説明します。

- 37 ページの「PAP 認証による構成例」
- 39 ページの「CHAP 認証による構成例」

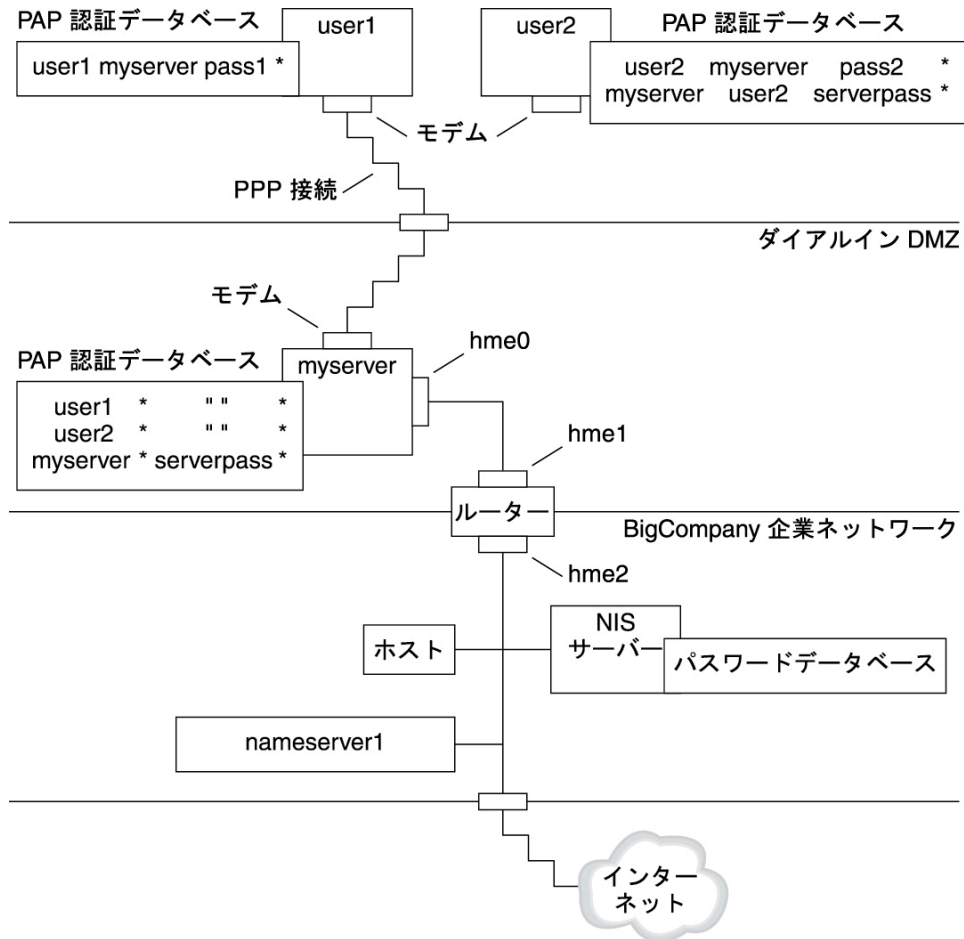
PAP 認証による構成例

68 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、31 ページの「ダイヤルアップ 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、ダイヤ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ヤルイン 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間に設置します。

図2-3「PAP 認証のシナリオ (自宅の仕事する) の例」に示すように、DMZ に存在するのは、ダイヤルインサーバーの `myserver` とルーターだけです。ダイヤ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、ダイヤルインサーバーは 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、ダイヤ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。

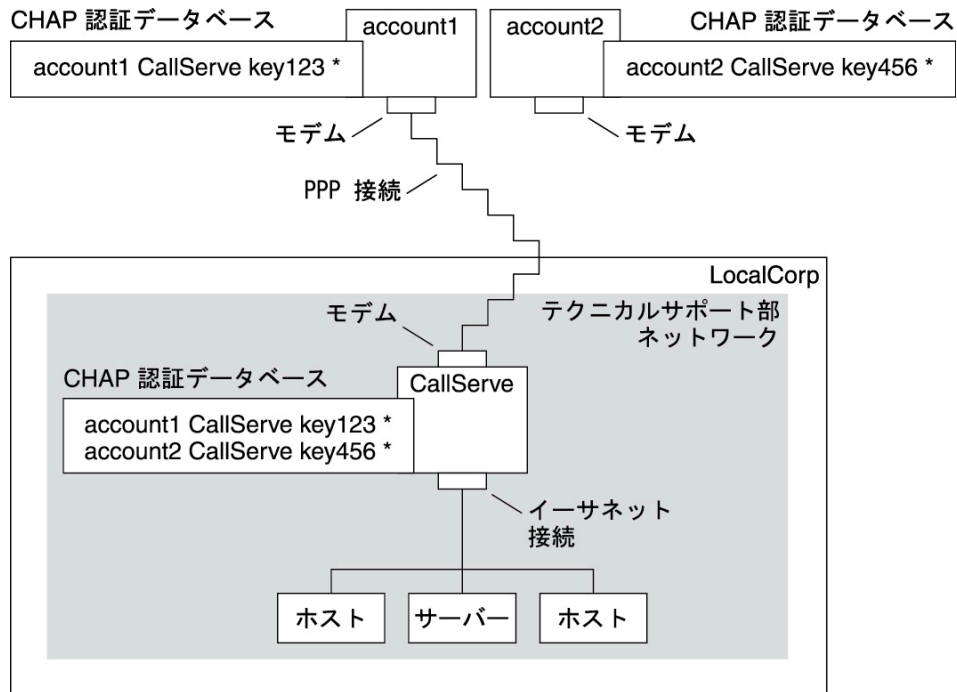
CHAP 認証による構成例

76 ページの「CHAP 認証の構成」でのタスクは、CHAP 認証を設定する方法を示しています。手順では、34 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、ダイヤルアップ 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、リンクのダイヤ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。ダイヤ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。

- [68 ページの「PAP 認証の構成」](#)
- [76 ページの「CHAP 認証の構成」](#)
- [136 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、[25 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。この章の内容は次のとおりです。

- PPPoE ホストとアクセスサーバーの計画情報
- [43 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「PPP Over Ethernet トンネルの設定」](#)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

多くのさまざまな 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	アクション
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 <code>ipadm show-addr</code> 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います。

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。

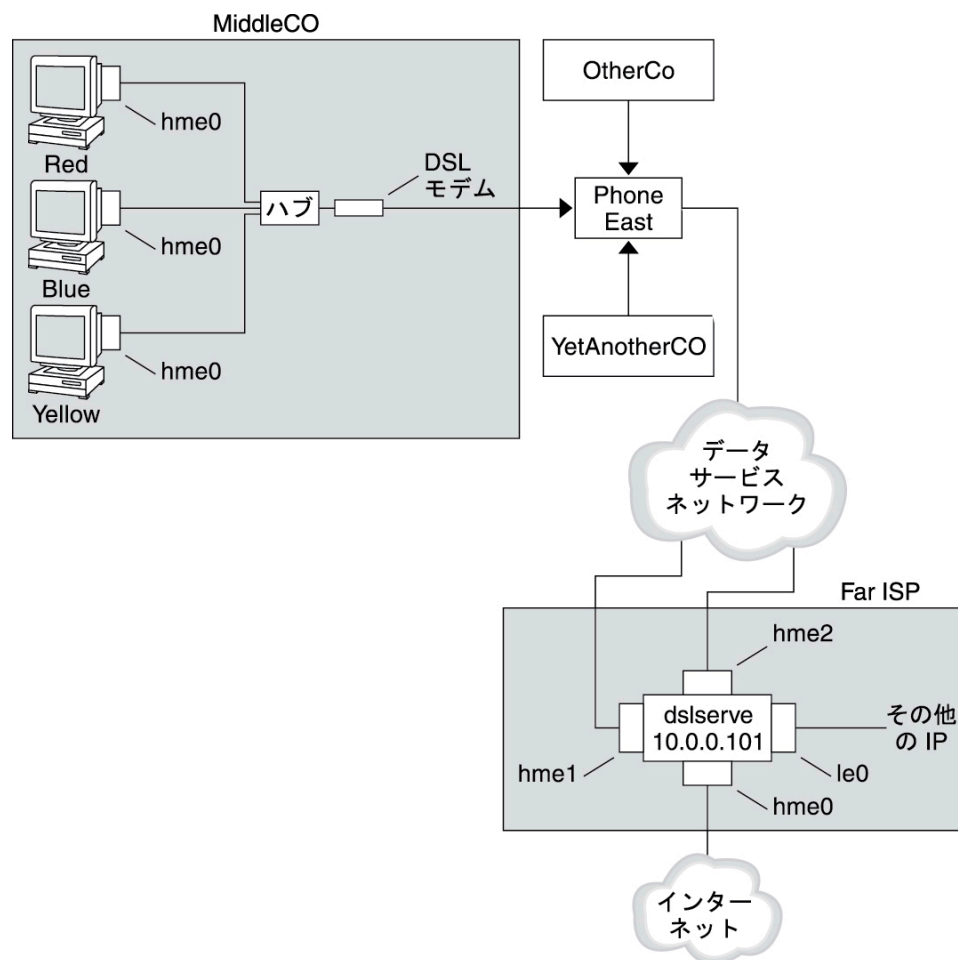
表 2-7 PPPoE アクセスサーバーの計画

情報	アクション
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	<code>ipadm show-addr</code> 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 認証に使います。

PPPoE トンネルの構成例

このセクションでは、第6章「PPP Over Ethernet トンネルの設定」のタスクの説明図として使用される 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。

- [84 ページの「PPPoE クライアントの設定」](#)
- [87 ページの「PPPoE アクセスサーバーの設定」](#)
- [145 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

ダイアルアップポイントツーポイントプロトコルリンクの設定

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。

- [46 ページの「ダイアルアウトマシンの構成」](#)
- [53 ページの「ダイアルインサーバーの構成」](#)
- [57 ページの「ダイアルインサーバーの呼び出し」](#)

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「PPP 構成ファイルとコマンドのサマリー」](#)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	手順の参照先
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	30 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	46 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	53 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)」
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	58 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「ダイアルアップリンクの例」](#)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	手順の参照先
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	30 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	47 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	49 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	50 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	51 ページの「個々のピアとの接続を定義する方法」
6. ピアを呼び出します	pppd コマンドを入力して、通信を開始する	58 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	手順の参照先
/etc/ppp/options.tpl	/etc/ppp/options	118 ページの「 /etc/ppp/options.tpl テンプレート 」
/etc/ppp/options.ttya.tpl	/etc/ppp/options.ttyname	121 ページの「 options.ttya.tpl テンプレートファイル 」
/etc/ppp/myisp-chat.tpl	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	128 ページの「 /etc/ppp/myisp-chat.tpl chat スクリプトテンプレート 」
/etc/ppp/peers/myisp.tpl	/etc/ppp/peers/peer-name	124 ページの「 /etc/ppp/peers/myisp.tpl 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.tpl) 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。

ダイヤルアウトマシン上にデバイスを構成する

ダイヤルアウト 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。

注記 - 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。

- ダイヤルアウトマシンに 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
- ダイヤルアウトマシンに使用するシリアルポートを決定する
- ダイヤ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、[30 ページの「ダイヤ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 (ダイヤルアウトマシン)

1. モデムを設定します。

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、[103 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ヤ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. ダイヤルアウトマシン上で管理者になります。
詳細は、『[Oracle Solaris 11.2 でのユーザーとプロセスのセキュリティー保護](#)』の「[割り当てられている管理権利の使用](#)」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。

ダイヤ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、ダイヤ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、[47 ページの「モデムとシリアルポートの構成方法 \(ダイヤルアウトマシン\)」](#)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、ダイヤルアウトマシンがダイヤ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、要求に合わせてカスタマイズできる、PPP 構成ファイルのテンプレートを提供します。これらのファイルについては、[46 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 11.2 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」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。

注記 - ダイアルアウトマシンに `/etc/ppp/options` ファイルがない場合は、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、[118 ページの「`/etc/ppp/options` 構成ファイル」](#)を参照してください。

3. **(オプション) 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、デバイス名が `/dev/cua/a` のポートの `/etc/ppp/options.ttyname` ファイルを示しています。

```
# cat /etc/ppp/options.cua.a
crtstcts
```

PPP オプション `crtstcts` 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、[119 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 [54 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

ダイヤルアウトマシンが PPP リンクを開始する前に、ピアになるダイヤ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、ダイヤルアウトマシンとピア間の実際の対話を記述します。

1. **ダイヤ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、[126 ページ](#)の「[ダイヤ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. **ダイヤ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. **ダイヤ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、ダイヤルインサーバー myserver を呼び出すために、次の chat スクリプト (/etc/ppp/mychat) を作成します。

```
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、[130 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、[126 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。

注記 - 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勧めします。

▼ 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 11.2 でのユーザーとプロセスのセキュリティ保護』の「割り当てられている管理権利の使用」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: "example.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、ダイヤルアウトマシンは最初に IP アドレス 0.0.0.0 を持つことを示します。myserver は、すべてのダイヤルアップセッションでダイヤルアウトマシンに IP アドレスを割り当てます。
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します。
noauth	ダイヤルアウトマシンとの接続をネゴシエートするときに、ピア myserver は認証資格を提供する必要がないことを示します。
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。

- 別のダイヤルアウトマシンを構成する手順については、[47 ページの「モデムとシリアルポートの構成方法 \(ダイヤルアウトマシン\)」](#)を参照
- 別のコンピュータにダイヤルアウトすることでモデムの接続性をテストするには、[cu\(1C\)](#) と [tip\(1\)](#) のマニュアルページを参照してください。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、[113 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
- ダイヤルインサーバーの構成手順については、[53 ページの「ダイヤ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、Figure 2-1 で紹介したダイアルインサーバー [図2-1「ダイアルアップリンクの例」](#)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	手順の参照先
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	30 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	54 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	55 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	56 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 (ダイヤルインサーバー)

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、[47 ページの「モデムとシリアルポートの構成方法 \(ダイヤルアウトマシン\)」](#)を参照してください。
2. モデムをダイヤルインサーバー上のシリアルポートに接続します。
3. ダイヤルインサーバー上で管理者になります。
詳細は、『Oracle Solaris 11.2 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」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、ダイヤ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、[126 ページの「ダイヤ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. ダイヤ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。

- [54 ページの「モデムとシリアルポートの構成方法 \(ダイヤルインサーバー\)」](#)
- [55 ページの「ダイヤ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
- [54 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、[143 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 11.2 でのユーザーとプロセスのセキュリティ保護](#)』の「[割り当てられている管理権利の使用](#)」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 11.2 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のタスクマップ](#)」を参照してください。
3. **各呼び出し元に対して、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記述されている \$HOME/.ppprc ファイルを作成します。**
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。

- [55 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」](#)

- [56 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ヤ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、[113 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。

- [54 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
- [55 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 11.2 でのユーザーとプロセスのセキュリティ保護](#)』の「[割り当てられている管理権利の使用](#)」を参照してください。

2. **次のエントリを指定して `/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。

注記 - ダイアルインサーバーに `/etc/ppp/options` ファイルがない場合は、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ます
```

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には、[cu\(1C\)](#) と [tip\(1\)](#) のマニュアルページを参照してください。これらのユーティリティを使用すると、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、[53 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、[46 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、[57 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、次のどちらかのタスクか両方のタスクを終了しておく必要があります。

- [46 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

- 53 ページの「ダイヤルインサーバーの構成」で説明しているとおりに、ダイヤルインサーバーを設定する

▼ ダイヤ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、ダイヤ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、ダイヤ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、ダイヤルアウトマシンとダイヤルインサーバー (`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す。これにより
`pppd`が、51 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` ファイル内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Figure 2-1 に示されている [図2-1「ダイヤルアップリンクの例」](#) ホストなど) にアクセスします。

```
ping lindyhop
```

リンクが正しく機能しない場合は、[第7章「ポイントツーポイントプロトコルの一般的な問題のトラブルシューティング」](#)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、ダイヤ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。

- ユーザーがダイヤルアウトマシン上で作業を開始する手順については、58 ページの「ダイヤルインサーバーの呼び出し方法」
- リンクの問題を修正するには、[第7章「ポイントツーポイントプロトコルの一般的な問題のトラブルシューティング」](#)を参照してください。

- 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、[113 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

専用回線ポイントツーポイントプロトコルリンクの設定

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。

- [62 ページの「専用回線上の同期デバイスの構成」](#)
- [63 ページの「専用回線上のマシンの構成」](#)

専用回線の設定 (タスクマップ)

専用回線リンクの設定は、ダイヤ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、ダイヤルサービス、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。

注記 - 専用回線の中には、対するピアのアドレスを「ダイヤル」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	手順の参照先
1. 構成前の情報を収集します	接続の設定に必要な情報を収集する	34 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	62 ページの「同期デバイスの構成方法」

タスク	説明	手順の参照先
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	62 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	64 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	起動プロセスの一部として、PPP が専用回線を介して開始されるようにマシンを構成する	64 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、[34 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。

注記 - 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があります。

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「専用回線の構成例」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=rxr txd=txd rxd=rxr 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、64 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、34 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。

- 62 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 11.2 でのユーザーとプロセスのセキュリティ保護』の「割り当てられている管理権利の使用」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'	リンクを開始する。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します。
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 デーモンに指示します。また、データキャリア検出 (DCD) 入力信号を無視することも pppd に指示する
/dev/hihp1	同期インタフェースのデバイス名を指定する
sync	リンクの同期エンコーディングを確立します。
noauth	ローカルシステムがピアに認証を要求する必要があるように設定します。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 デーモンに指示します。
persist	リンクが解除されたあとでもう一度リンクを開始するように、pppd デーモンに指示します。
noaccp, 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、ダイヤ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. ブートプロセスの一部として PPP リンクを作成する demand という初期設定スクリプトを作成します。

```
# 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	説明
<pre>if [-f /system/volatile/ppp-demand.pid] && /usr/bin/kill -s 0 `/bin/cat /system/volatile/ppp-demand.pid` /usr/bin/pppd call farISP</pre>	<p>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</p> <p>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 も読み取られます。</p>

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.d/S47pppd にある次の行は、\$PPPPDIR/demand というファイルが存在するかどうかを調べます。

```
if [ -f $PPPPDIR/demand ]; then
    . $PPPPDIR/demand
fi
```

\$PPPPDIR/demand が検出された場合は、それが実行されます。\$PPPPDIR/demand の一連の処理の実行中に、接続が確立されます。

注記 - 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、110 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、113 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

ポイントツーポイントプロトコル認証の設定

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。

- 68 ページの「PAP 認証の構成」
- 76 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、25 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 136 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 認証に関連するタスクを示します。

表 5-1 一般的な PPP 認証 (タスクマップ)

タスク	説明	手順の参照先
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	68 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	76 ページの「CHAP 認証の設定 (タスクマップ)」

PAP 認証の構成

このセクションのタスクは、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、[37 ページの「PPP の認証構成例」](#)の例を使用して、ダイヤルアップリンクで 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。

- ダイヤ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ヤルアウトマシン間で、ダイヤルアップリンクを設定しテストします。
- ダイヤ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、およびダイヤルインサーバーまたはダイヤ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、ダイヤルインサーバーおよびダイヤ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 (ダイヤルインサーバー)

タスク	説明	手順の参照先
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	36 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	69 ページの「PAP 資格データベースの作成方法 (ダイヤルインサーバー)」
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資格を <code>/etc/ppp/pap-secrets</code> に作成します。	69 ページの「PAP 資格データベースの作成方法 (ダイヤルインサーバー)」
4. PPP の構成ファイルを変更する	PAP 特有のオプションを <code>/etc/ppp/options</code> および <code>/etc/ppp/peers/peername</code> ファイルに追加します。	71 ページの「PPP 構成ファイルに PAP サポートを追加する方法 (ダイヤルインサーバー)」

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	手順の参照先
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	36 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資格を <code>/etc/ppp/pap-secrets</code> に作成する	73 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 <code>/etc/ppp/options</code> および <code>/etc/ppp/peers/peername</code> ファイルに追加します。	74 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「PAP 認証のシナリオ (自宅で仕事する) の例」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、139 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. **ダイヤ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。**

注記 - この確認は、この PAP 構成のサンプルにとって重要です。このサンプルでは、呼び出し元の認証に、PAP の login オプションを使用しています。PAP に login を実装しない場合は、呼び出し元の PAP ユーザー名と UNIX ユーザー名を一致させる必要はありません。標準の `/etc/ppp/pap-secrets` については、[137 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、呼び出し元がダイヤ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. **ダイヤルインサーバー上で管理者になります。**

詳細は、『[Oracle Solaris 11.2 でのユーザーとプロセスのセキュリティ保護](#)』の「[割り当てられている管理権利の使用](#)」を参照してください。

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

このリリースでは、`/etc/ppp` に `pap-secrets` ファイルがあります。このファイルには、PAP 認証の使用方法についてのコメントが含まれています。ただし、オプションについてのコメントは含まれていません。コメントの最後に、次のオプションを追加することができます。

```
user1    myserver    ""        *
user2    myserver    ""        *
myserver user2      serverpass *
```

`/etc/ppp/pap-secrets` の login オプションを使用するには、信頼できる呼び出し元の UNIX ユーザー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、ダイヤ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「PAP 認証のシナリオ \(自宅で仕事する\) の例」](#)では、信頼できる呼び出し元である `user2` は、リモートピアに認証を要求します。そのため、`myserver` の `/etc/ppp/pap-secrets` ファイルには、`user2` との接続を確立する場合に使用する PAP 資格が含まれています。

参照 関連情報の参照先は次のとおりです。

- 71 ページの「PPP 構成ファイルを PAP 用に変更する (ダイヤルインサーバー)」
- 72 ページの「信頼できる呼び出し元の PAP 認証の構成 (ダイヤルアウトマシン)」

PPP 構成ファイルを PAP 用に変更する (ダイヤルインサーバー)

このセクションでは、ダイヤルインサーバーで 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 (ダイヤルインサーバー)

ここでは、56 ページの「シリアル回線を介した通信を定義する方法 (ダイヤルインサーバー)」で紹介した PPP 構成ファイルを例として使用します。

1. **ダイヤルインサーバー上で管理者になります。**

詳細は、『Oracle Solaris 11.2 でのユーザーとプロセスのセキュリティ保護』の「割り当てられている管理権利の使用」を参照してください。

2. **認証オプションを /etc/ppp/options ファイルに追加します。**

たとえば、既存の /etc/ppp/options ファイルに、次の太字のオプションを追加すると、PAP 認証を実装することができます。

```
lock
auth
login
nodefaultroute
proxyarp
ms-dns 10.0.0.1
idle 120
```

auth リンクを確立する前に、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します。

<code>nodefaultroute</code>	ローカルシステム上の <code>pppd</code> セッションが <code>root</code> 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
<code>proxyarp</code>	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
<code>ms-dns 10.0.0.1</code>	<code>pppd</code> がクライアントにドメインネームサーバー (DNS) アドレス <code>10.0.0.1</code> を与えることができるようにする
<code>idle 120</code>	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。

```
* * "" *
```

注記 - 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 [ダイヤ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、[72 ページの「信頼できる呼び出し元の PAP 認証の構成 \(ダイヤルアウトマシン\)」](#)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 (ダイヤルアウトマシン)

このセクションでは、信頼できる呼び出し元のダイヤルアウトマシンで、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、このセクションの手順を指示することもできます。

信頼できる呼び出し元に PAP を構成するには、次の 2 つの手順を実行します。

- 呼び出し元の PAP セキュリティー資格を構成します。
- 呼び出し元のダイヤルアウトマシンが 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ヤルアウトマシンで PAP 資格を作成することを前提にしています。

1. ダイヤルアウトマシン上で管理者になります。

詳細は、『Oracle Solaris 11.2 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。

図2-3「PAP 認証のシナリオ (自宅仕事する) の例」で紹介した PAP 構成のサンプルを使用して、user1 がダイヤ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。

このリリースに含まれている /etc/ppp/pap-secrets ファイルには、役に立つ情報が含まれていますが、オプションについては触れていません。次のオプションをこの /etc/ppp/pap-secrets ファイルに追加できます。

```
user1 myserver pass1 *
```

user1 のパスワードである pass1 は、接続を通して、読み取り可能な ASCII 形式になることに注意してください。myserver は、呼び出し元 user1 がピアで使用する名前です。

3. ダイヤルアウトマシン上で管理者になります。

詳細は、『Oracle Solaris 11.2 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。

PAP 認証の例では、呼び出し元 user2 がこのダイヤ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ヤルインサーバー `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、ダイヤルインサーバーに 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。

注記 - 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。

- [69 ページの「PAP 資格データベースの作成方法 \(ダイヤルインサーバー\)」](#)
- [73 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 (ダイヤルアウトマシン)

次のタスクは、信頼できる呼び出し元のダイヤルアウトマシンで 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、Figure 2-3 で紹介した [図2-3「PAP 認証のシナリオ \(自宅で仕事する\) の例」](#) が所有するダイヤルアウトマシン上で、PAP 認証を構成します。`user2` は、ダイヤルイン `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 (ダイヤルアウトマシン)

ここでは、[49 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、Figure 2-3 で示した [図2-3「PAP 認証のシナリオ \(自宅で仕事する\) の例」](#) が所有するダイヤルアウトマシンを構成します。

1. **ダイヤ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 名として設定します。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、ダイヤ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

注記 - ほとんどのダイヤ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、このダイヤ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、How to Define the Connection With an Individual Peer で作成した既存の [51 ページの「個々のピアとの接続を定義する方法」](#) ファイルに、PAP サポートを追加する方法を示しています。

```
# cat /etc/ppp/peers/myserver
/dev/cua/a
57600
noipdefault
defaultroute
idle 120
user user2
remotename myserver
connect "chat -U 'mypassword' -f /etc/ppp/mychat"
```

太字で示した新しいオプションにより、ピア myserver に関する PAP 要件が追加されます。

user user2 user2 をローカルマシンのユーザー名として定義する

remotename myserver myserver をローカルマシンに認証資格を要求するピアとして定義する

参照 関連情報の参照先は次のとおりです。

- ダイアルインサーバーを呼び出して、PAP 認証の設定をテストする手順については、58 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、136 ページの「パスワード認証プロトコル (PAP)」

CHAP 認証の構成

このセクションでは、チャレンジハンドシェイク認証プロトコル (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「CHAP 認証シナリオ \(私設ネットワークを呼び出す\) の例」](#)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	手順の参照先
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	77 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資格を /etc/ppp/chap-secrets ファイルに追加します。	77 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 ファイルに追加します。	79 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マップ (ダイヤルアウトマシン)

タスク	説明	手順の参照先
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資格と、必要であれば、ダイヤ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資格を <code>/etc/ppp/chap-secrets</code> に作成する	77 ページの「CHAP 資格データベースの作成方法 (ダイヤルインサーバー)」
2. PPP の構成ファイルを変更する	CHAP 特有のオプションを <code>/etc/ppp/options</code> ファイルに追加する	81 ページの「PPP 構成ファイルに CHAP サポートを追加する方法 (ダイヤルアウトマシン)」

ダイヤルインサーバーに 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 セキュリティ資格が含まれています。このセキュリティ資格を使用して、接続時に呼び出し元を認証します。

注記 - UNIX の認証メカニズムまたは PAM の認証メカニズムを CHAP とともに使用することはできません。たとえば、How to Create a PAP Credentials Database (Dial-in Server) で説明したような PPP 69 ページの「PAP 資格データベースの作成方法 (ダイヤルインサーバー)」オプションを使用することはできません。認証時に、PAM または UNIX スタイルの認証が必要な場合は、代わりに PAP を選択してください。

次に、私設ネットワークにあるダイヤルインサーバーの 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 (ダイヤルインサーバー)

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。

注記 - 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ポリシーにより異なります。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. **ダイヤルインサーバー上で管理者になります。**

詳細は、『[Oracle Solaris 11.2 でのユーザーとプロセスのセキュリティ保護](#)』の「[割り当てられている管理権利の使用](#)」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することができます。

```
account1 CallServe key123 *
account2 CallServe key456 *
```

key123 は、信頼できる呼び出し元 account1 の CHAP シークレットです。

key456 は、信頼できる呼び出し元 account2 の CHAP シークレットです。

参照 関連情報の参照先は次のとおりです。

- [77 ページの「CHAP 資格データベースの作成方法 \(ダイヤルインサーバー\)」](#)
- [79 ページの「PPP 構成ファイルに CHAP サポートを追加する方法 \(ダイヤルインサーバー\)」](#)
- [79 ページの「信頼できる呼び出し元の CHAP 認証の構成 \(ダイヤルアウトマシン\)」](#)

PPP 構成ファイルを CHAP 用に変更する (ダイヤルインサーバー)

このセクションでは、ダイヤルインサーバーで 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 (ダイヤルインサーバー)

1. ダイヤ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 (この場合はダイヤルインサーバー) 上のユーザーの CHAP 名として定義します。

`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。
[55 ページの「ダイヤルインサーバーのユーザーを構成する方法」](#)および [56 ページの「シリアル回線を介した通信を定義する方法 \(ダイヤルインサーバー\)」](#)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、[77 ページの「CHAP 資格データベースの作成方法 \(ダイヤルインサーバー\)」](#)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 (ダイヤルアウトマシン)

このセクションには、信頼できる呼び出し元のダイヤルアウトマシンで、CHAP 認証を設定するタスクを説明します。企業のセキュリティポリシーによって、管理者と信頼できる呼び出し元のどちらかが CHAP 認証を設定するのが決まります。

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、ダイヤルインサーバーの `/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 構成の手順を指示します。

信頼できる呼び出し元に CHAP を構成するには、次の 2 つの手順を実行します。

- 呼び出し元の CHAP セキュリティー資格を作成します。
- 呼び出し元のダイヤルアウトマシンが 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、CHAP 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ヤルアウトマシンで CHAP 資格を作成することを前提にしています。

1. ダイヤルアウトマシン上で管理者になります。

詳細は、『Oracle Solaris 11.2 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。

39 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 がダイヤ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。

このリリースに含まれている /etc/ppp/chap-secrets ファイルには、役に立つ情報が含まれていますが、オプションについては触れていません。次のオプションを既存の /etc/ppp/chap-secrets ファイルに追加できます。

```
account1 CallServe key123 *
```

CallServe は、account1 がアクセスを試みているピアの名前です。key123 は、account1 と CallServer 間でのリンクに使用する CHAP シークレットです。

3. ダイヤルアウトマシン上で管理者になります。

詳細は、『Oracle Solaris 11.2 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」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。

- [77 ページの「CHAP 資格データベースの作成方法 \(ダイヤルインサーバー\)」](#)
- [80 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 (ダイヤルアウトマシン)

CHAP 認証の詳細を理解するには、[140 ページの「チャレンジハンドシェイク認証プロトコル \(CHAP\)」](#)を参照してください。次の手順に従って、Example of a Configuration Using CHAP Authenticationで紹介した呼び出し元 [39 ページの「CHAP 認証による構成例」](#)が所有するダイヤ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 (ダイヤルアウトマシン)

1. ダイヤ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` ファイルに次のオプションが含まれていることを確認します。

```
# 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、How to Define the Connection With an Individual Peerの同様の [51 ページの「個々のピアとの接続を定義する方法」](#) ファイルを参照してください。

参照 [ダイヤ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、58 ページの「ダイヤルインサーバーの呼び出し方法」を参照してください。](#)

PPP Over Ethernet トンネルの設定

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。

- [83 ページの「PPPoE トンネル設定の主なタスク \(タスクマップ\)」](#)
- [84 ページの「PPPoE クライアントの設定」](#)
- [87 ページの「PPPoE アクセスサーバーの設定」](#)

ここでは、[41 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、[25 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のリストを示します。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	手順の参照先
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	85 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	85 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	49 ページの「シリアル回線を介した通信を定義する方法」

タスク	説明	手順の参照先
4. トンネルを作成します	アクセスサーバーを呼び出す	85 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	手順の参照先
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	87 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	56 ページの「ダイヤ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	89 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、電話会社に取り付ける

▼ 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 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 11.2 でのユーザーとプロセスのセキュリティ保護』の「割り当てられている管理権利の使用」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、146 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. (オプション) 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、147 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、123 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 11.2 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」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、Example of a Configuration for a PPPoE Tunnel で紹介した Far ISP にあるアクセスサーバー 43 ページの「PPPoE トンネルの構成例」を定義しています。

```
# cat /etc/ppp/peers/dslserve
sppptun
plugin pppoe.so
connect "/usr/lib/inet/pppoc 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、155 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. `Configuring the Dial-out Machine` 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、46 ページの「ダイアルアウトマシンの構成」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が接続するシリアルポートの PPP オプションを定義します。
`/etc/ppp/options.ttyname` Configuration File で説明する `/etc/ppp/options.119` ページの「`/etc/ppp/options.ttyname` 構成ファイル」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、43 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/pppoc -i hme0
```

注記 - 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。

- 84 ページの「PPPoE クライアントの設定」
- 145 ページの「DSL サポート用の PPPoE トンネルの作成」
- 第7章「ポイントツーポイントプロトコルの一般的な問題のトラブルシューティング」を参照してください。
- 87 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 11.2 でのユーザーとプロセスのセキュリティ保護』の「割り当てられている管理権利の使用」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 43 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 `ds1serve` 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、Figure 2-5 で示したアクセスサーバー [図2-5「PPPoE トンネルの例」](#) 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、`ds1serve` 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. **ダイヤ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、143 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. **(オプション) 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/spptun query
hme1:pppoe
hme1:pppoed
hme2:pppoe
hme2:pppoed
```


前の例は、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、147 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 11.2 でのユーザーとプロセスのセキュリティー保護』の「[割り当てられている管理権利の使用](#)」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. pppoe デーモンに新しいサービスを認識させます。

```
# pkill -HUP pppoe
```

▼ 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。

注記 - 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、43 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. [87 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 というシンボリック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 シンボリック名が 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。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  yellowpasswd  *
```

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、[68 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、[145 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、[97 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、[84 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、[72 ページの「信頼できる呼び出し元の PAP 認証の構成 \(ダイヤルアウトマシン\)」](#)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、[69 ページの「ダイヤルインサーバーに PAP 認証を構成する」](#)を参照してください。

ポイントツーポイントプロトコルの一般的な問題のトラブルシューティング

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。この章の内容は次のとおりです。

- 94 ページの「PPP のトラブルシューティングのためのツール」
- 97 ページの「PPP および PPPoE 関連の問題の解決」
- 110 ページの「専用回線の問題の解決」
- 111 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、16 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」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	手順の参照先
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。	95 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。	96 ページの「 PPP デバッグをオンに設定する方法 」

タスク	定義	手順の参照先
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。	98 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。	101 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。	102 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。	103 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	ダイヤ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。	104 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	ダイヤ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。	107 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。	110 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。	111 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。	108 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、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モニターしたり、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 11.2 でのユーザーとプロセスのセキュリティ保護』の「割り当てられている管理権利の使用」を参照してください。
2. PPP 向けに構成されているシリアルデバイスを引数として指定した 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 <asynmap 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 <asynmap 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-asynmap 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 01>]]
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。

注記 - 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 11.2 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」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、[ステップ 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。

- [98 ページの「ネットワークの問題を診断する方法」](#)
- [100 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」](#)
- [101 ページの「通信の問題を診断し解決する方法」](#)
- [101 ページの「PPP に影響を与える一般的な通信の問題」](#)
- [102 ページの「PPP 構成の問題を診断する方法」](#)
- [102 ページの「一般的な PPP 構成の問題」](#)
- [103 ページの「モデムの問題を診断する方法」](#)
- [104 ページの「chat スクリプトのデバッグ情報を取得する方法」](#)
- [105 ページの「chat スクリプトの一般的な問題」](#)

- 107 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」
- 108 ページの「PPPoE の診断情報を取得する方法」

▼ 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 11.2 でのユーザーとプロセスのセキュリティ保護』の「割り当てられている管理権利の使用」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はない場合は、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. (オプション) 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 の詳細については、[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
```

注記 - 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、システムのリポ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、システムのリポ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、100 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

次の表に示したネットワークの問題を解決するには、`netstat -s` によって生成されたメッセージを使用できます。関連する作業情報として、98 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
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	<code>tracert</code> を使ってルーティング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。 <code>tracert</code> の詳細は、 tracert(1M) のマニュアルページを参照してください。
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「chat スクリプトの一般的な問題」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 11.2 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」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、101 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、101 ページの「通信の問題を診断し解決する方法」の作業で出力されるログに関連する症状を説明したものです。

表 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	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。 <ul style="list-style-type: none"> ■ 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「chat スクリプトの一般的な問題」を参照してください。

症状	問題	解決方法
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 11.2 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」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、102 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、102 ページの「PPP 構成の問題を診断する方法」の作業で出力されるログに関連する症状を説明したものです。

表 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. [96 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 packets 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 packets は受信されるが、大きい packets 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、[105 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. **ダイヤルアウトマシン上で管理者になります。**
詳細は、『[Oracle Solaris 11.2 でのユーザーとプロセスのセキュリティ保護](#)』の「[割り当てられている管理権利の使用](#)」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、ダイヤ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、104 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: <i>user-name</i> ssword: <i>password</i> 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: <i>pppuser</i> ssword: <i>\q\U</i> 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 が起動していることを示しているが、リンクがすぐに終了してしまう。	chat スクリプト内のパスワードが間違っている可能性がある	1. ローカルマシンの正しいパスワードを確認する 2. chat スクリプト内のパスワードを確認する。間違っている場合は修正する 3. 再度ピアを呼び出してみる 4. まだメッセージが表示される場合は、ISP に連絡して正しいログインシーケンスを問い合わせる
ピアからのテキストがチルダ (~) で始まる	chat スクリプトは、次のようにユーザー名とパスワードを指定している	1. chat スクリプトからログインとパスワードを削除する

症状	問題	解決方法
モデムが停止する。	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	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 LCP: timeout sending Config-Requests が含まれる	chat スクリプトに次の行が含まれており、ローカルマシンがピアからの CONNECT メッセージを待つように強制している CONNECT "	chat スクリプトがピアからの CONNECT を待つようにするときは、次の行を使用する CONNECT \c chat スクリプトを ~ \c で終了する
pppd debug 出力に Serial link is not 8-bit clean が含まれる	chat スクリプトに次の行が含まれており、ローカルマシンがピアからの CONNECT メッセージを待つように強制している CONNECT "	chat スクリプトがピアからの CONNECT を待つようにするときは、次の行を使用する CONNECT \c chat スクリプトを ~ \c で終了する
pppd debug の出力に Loopback detected が含まれる	chat スクリプトに次の行が含まれており、ローカルマシンがピアからの CONNECT メッセージを待つように強制している CONNECT "	chat スクリプトがピアからの CONNECT を待つようにするときは、次の行を使用する CONNECT \c chat スクリプトを ~ \c で終了する
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

ダイヤルインサーバーでは、速度の設定の矛盾が原因で問題が発生する可能性があります。次に示す手順は、接続の問題の原因がシリアル回線速度の衝突であることを特定するのに役立ちます。

速度の問題は、次のような原因で発生します。

- /bin/login のようなプログラムを介して PPP を起動し、回線の速度を指定した
- PPP を mgetty から起動し、誤ってビットレートを指定した

pppd は、はじめは回線に設定されていた速度を /bin/login または mgetty によって設定された速度に変更します。このことが回線の障害を発生させます。

1. **ダイヤ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、[96 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. [96 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/pppoc
-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>
asynmap 0x0 <magic 0x77d3e953><pcomp><acomp>
Sep 6 16:28:46 enyo pppd[100563]: [ID 702911 daemon.debug] rcvd
[LCP ConfReq id=0x2a <mru 1402>
asynmap 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/pppoe -v interface-name"
```

pppoe 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/pppoe -v hme0' started (pid 100564)
/usr/lib/inet/pppoe: PPPoE Event Open (1) in state Dead (0): action SendPADI (2)
/usr/lib/inet/pppoe: Sending PADI to ff:ff:ff:ff:ff:ff: 18 bytes
/usr/lib/inet/pppoe: PPPoE State change Dead (0) -> InitSent (1)
/usr/lib/inet/pppoe: Received Active Discovery Offer from 8:0:20:cd:c1:2/hme0:pppoe
/usr/lib/inet/pppoe: PPPoE Event rPADO+ (5) in state InitSent (1): action SendPADR+ (5)
/usr/lib/inet/pppoe: Sending PADR to 8:0:20:cd:c1:2: 22 bytes
/usr/lib/inet/pppoe: PPPoE State change InitSent (1) -> ReqSent (3)
/usr/lib/inet/pppoe: Received Active Discovery Session-confirmation from
      8:0:20:cd:c1:2/hme0:pppoe
/usr/lib/inet/pppoe: PPPoE Event rPADS (7) in state ReqSent (3): action Open (7)
/usr/lib/inet/pppoe: Connection open; session 0002 on hme0:pppoe
/usr/lib/inet/pppoe: PPPoE State change ReqSent (3) -> Convers (4)
/usr/lib/inet/pppoe: 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 = 0x00000002
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
接続が始まらない	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	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。

症状	問題	解決方法
	の側が ESF の B8ZS (Bit-8 Zero Substitute) 用に設定されています。	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刻時の問題がある可能性があります。	電話会社に連絡し、「ループ刻時」を使用していたことを確認します。 構造化されていない専用回線では、刻時を提供する必要がある場合があります。北米のユーザーはループクロックを使用するようにしてください。

認証の問題の診断と解決

次の表は、一般的な認証問題について説明したものです。

表 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。

- 113 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」
- 122 ページの「ユーザー独自のオプションの構成」
- 123 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」
- 126 ページの「ダイアルアップリンクのモデム速度の構成」
- 126 ページの「ダイアルアップリンクでの会話の定義」
- 136 ページの「接続時の呼び出し元の認証」
- 143 ページの「呼び出し元の IP アドレス指定スキームの作成」
- 145 ページの「DSL サポート用の PPPoE トンネルの作成」

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ています。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。	118 ページの「 /etc/ppp/options 構成ファイル 」
/etc/ppp/options.ttyname	シリアルポート <i>ttyname</i> 上のすべての通信の特性を記述するファイル。	119 ページの「 /etc/ppp/options.ttyname 構成ファイル 」
/etc/ppp/peers	通常、ダイヤ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。	123 ページの「 ダイヤ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 <i>peer-name</i> 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。	123 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資格を含むファイル。	137 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イク認証プロトコル (CHAP) の認証に必要なセキュリティ資格を含むファイル。	140 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。ダイヤ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。	122 ページの「 ダイヤ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。	114 ページの「 PPP オプションの処理方法 」

PPP ファイルの詳細については、[pppd\(1M\)](#) のマニュアルページを参照してください。[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 `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。ファイルが見つかったら、`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します。

`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には、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	説明
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。

オプション	ステータス	説明
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 <i>addresses</i>	特権がある	使用には特権が必要です。
name <i>hostname</i>	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。
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。
<i>ttyname</i>	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。

オプション	ステータス	説明
	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

注記 - 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 [49 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.tpl テンプレート

`/etc/ppp/options.tpl` 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

オプション	定義
<code>nodefaultroute</code>	デフォルトの送信経路を定義しないことを指定する
<code>noproxyarp</code>	<code>proxyarp</code> を許可しない

`/etc/ppp/options.tpl` 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.tpl` 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、[49 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、[56 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、[71 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、[74 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、[79 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` は特権ファイルであり、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で `/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。

注記 - 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。

注記 - 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.tpl` テンプレートファイル

`/etc/ppp/options.ttya.tpl` 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
asynmap 0xa0000
:192.168.1.1
```

オプション	定義
38400	ポート <code>ttya</code> でこのボーレートを使用する
<code>asynmap 0xa0000</code>	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 <code>asynmap</code> 値 <code>0xa0000</code> を割り当てる
<code>:192.168.1.1</code>	接続上で着信しているすべてのピアに IP アドレス <code>192.168.1.1</code> を割り当てる

サイトで `/etc/ppp/options.ttya.tpl` を使用するには、`/etc/ppp/options.tpl` 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、次を参照してください。

- ダイアルアウトマシン用は、[49 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、[56 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、[55 ページの「ダイアルインサーバーのユーザーを設定する」](#)を参照してください。

ダイアルアウトマシンでの \$HOME/.ppprc の構成

\$HOME/.ppprc ファイルは、ダイアルアウトマシン上で Solaris PPP 4.0 が正常に動作するのに必要ではありません。ダイアルアウトマシンでは、特別な場合を除いて \$HOME/.ppprc も必要ありません。次を行う場合は、1 つ以上の .ppprc ファイルを作成します。

- 通信のニーズが異なる複数のユーザーが同じマシンからリモートピアを呼び出すのを許可する場合。このような場合は、ダイアルアウトする必要がある各ユーザーのホームディレクトリに個別の .ppprc ファイルを作成します。
- Van Jacobson 圧縮を無効にするなど、接続に固有の問題を制御するオプションを指定する必要がある場合。接続に関する問題のトラブルシューティングについては、James Carlson 著『*PPP Design, Implementation, and Debugging*』および [pppd\(1M\)](#) のマニュアルページを参照してください。

.ppprc ファイルは、ダイアルインサーバーを構成するときにもっとも頻繁に使用されるため、55 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」の構成手順について How to Configure Users of the Dial-in Server 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

注記 - `/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、ダイヤルアウトマシンのログイン名として `user-name` をダイヤルインサーバーに指定します。

■ `remotename peer-name`

`peer-name` をダイヤ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ヤ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その他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.tpl` テンプレートファイル

`/etc/ppp/peers/myisp.tpl` 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.tpl を使用するには、/etc/ppp/peers/myisp.tpl 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、次を参照してください。

- ダイアルアウトマシン用は、51 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、64 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、74 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、81 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、84 ページの「[PPPoE クライアントの設定](#)」を参照してください。

ダイヤ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 ビット/秒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、最新のモデムで使用でき、ダイヤルアップリンクに十分な速度である 115200 ビット/秒に設定します。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 での制限は、非同期リンクの場合 460800 ビット/秒です。

ダイヤ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。

ダイヤルインサーバーでは、*Configuring Devices on the Dial-in Server* で説明したように、[53 ページの「ダイヤルインサーバーにデバイスを構成する」](#) 機能を使って速度を設定する必要があります。

ダイヤルアップリンクでの会話の定義

ダイヤ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。ダイヤ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

ダイヤ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。

注記 - chat スクリプトは、通常、ダイヤ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	説明
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、記録しません。 \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.tpl 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.tpl 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.tpl 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	説明
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"	<p>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。</p> <p>&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。</p>
SAY "Calling myisp\n"	ローカルマシン上に「Calling myisp (myisp を呼び出し中)」のメッセージを表示します。
TIMEOUT 60	タイムアウトを 60 秒にリセットし、接続ネゴシエーションにより多くの時間を割り当てます。
OK "ATDT1-123-555-1212"	電話番号 1-123-555-1212 を使ってリモートピアに発信します。
CONNECT \c	反対側のピアのモデムからの CONNECT メッセージを待ちます。

ISP を呼び出すためのモデムの chat スクリプト

ダイアルアウトマシンから U.S. Robotics Courier モデムを使用して ISP を呼び出すには、テンプレートとして次の chat スクリプトを使用します。

```

ABORT BUSY
ABORT 'NO CARRIER'

```

```
REPORT CONNECT
TIMEOUT 10
"" AT&F1M0&M5S2=255
SAY "Calling myisp\n"
TIMEOUT 60
OK "ATDT1-123-555-1212"
CONNECT \c
\r \d\c
SAY "Connected; running PPP\n"
```

次の表では、chat スクリプトの内容を説明します。

スクリプトの内容	説明
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。
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、[50 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	説明
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。
"" 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。

スクリプトの内容	説明
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、[55 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	説明
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。
"" 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。 <ul style="list-style-type: none"> ■ &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

スクリプトの内容	説明
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せん。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。

注記 - 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。

外部ファイル内の chat スクリプト

特定のピアに必要な 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、[50 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ません。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のも、違うものにするのもできます。

`/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。

<code>myclient</code>	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、ダイヤルインサーバーが PAP の <code>login</code> 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。
<code>ISP-server</code>	リモートマシンの名前。ダイヤルインサーバーである場合がしばしばあります。
<code>mypassword</code>	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、リンク上を *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

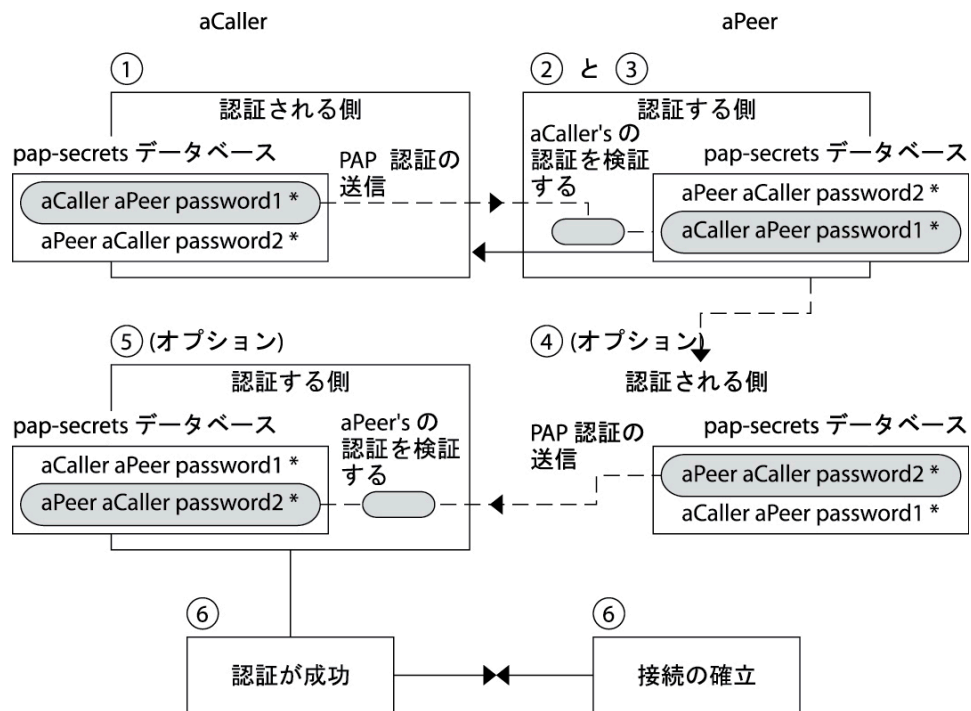
サーバー (認証する側) では、PAP パスワードは、次のどれかの方法で隠すことができます。

- pap-secrets ファイル内で papcrypt を指定し、encrypt によってハッシュ化されたパスワードを使用します。
- 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します。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。ピアが PAP の `login` オプションを使用する場合は、ピアはパスワードデータベースで呼び出し元のユーザー名とパスワードを検証します。
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。

注記 - 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` オプションを使用する `/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 | <code>joe</code> , <code>sally</code> , <code>sue</code> 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。 <code>name</code> 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 <code>passwd</code> 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イク認証プロトコル (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 (ハンドシェイク) を生成しピアに送信します。

CHAP セキュリティー資格証明には、CHAP ユーザー名と CHAP「シークレット」が含まれています。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。

注記 - 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。

<code>myclient</code>	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。
<code>myserver</code>	リモートマシンの名前。ダイヤ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。

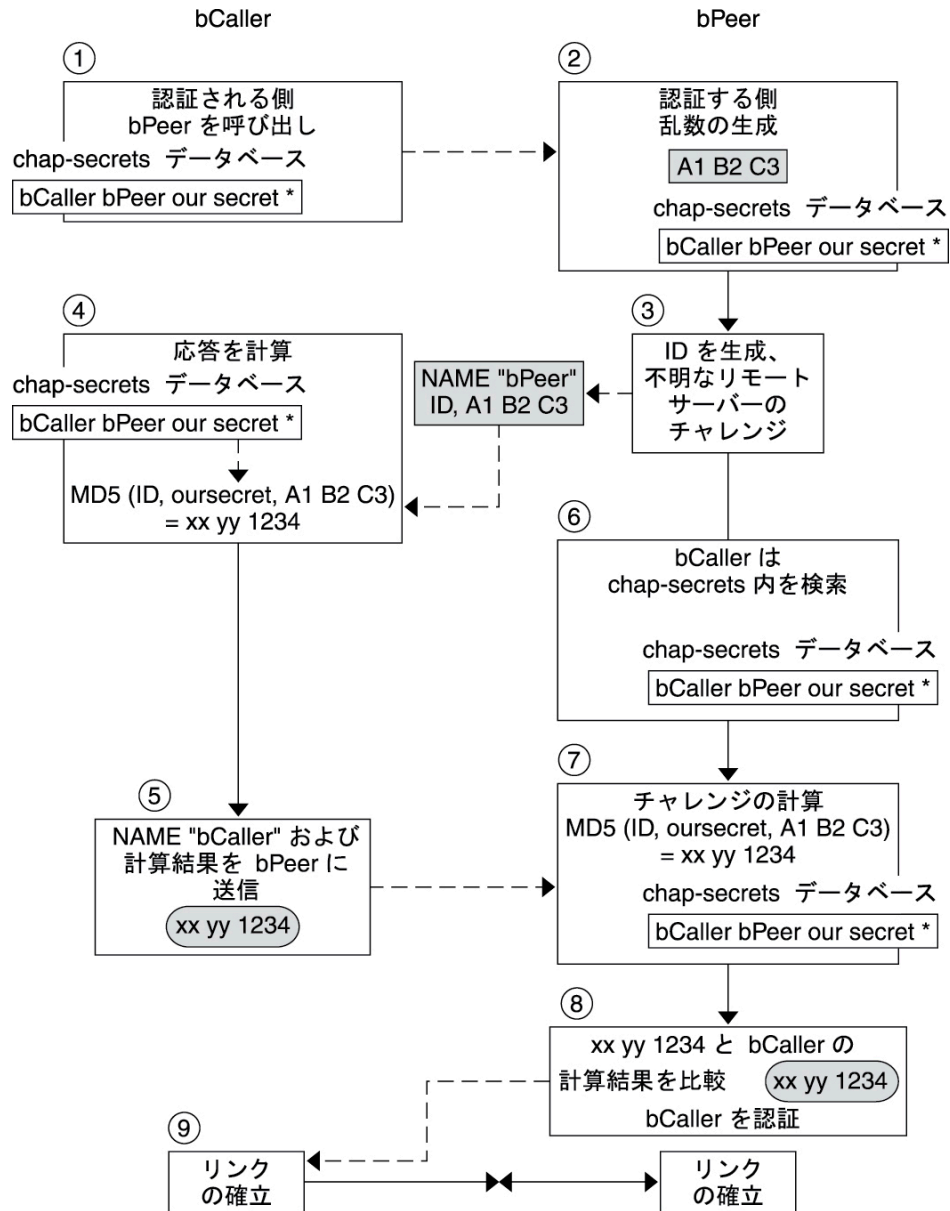
注記 - PAP パスワードと異なり、CHAP シークレットは送信されません。CHAP シークレットは、ローカルマシンが応答を処理するときに使用されます。

* 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 資格情報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

動的アドレス指定では、`/etc/ppp/options.ttyname` で定義されている IP アドレスが各呼び出し元に割り当てられます。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、呼び出し元は呼び出しのシリアルインタフェース用に `/etc/ppp/options.ttyname` ファイル内の IP アドレスを受け取ります。

たとえば、ダイアルインサーバーに、着呼に対してダイヤ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、ダイヤルアウトマシンがダイヤ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。次の点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、[26 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「PPP Over Ethernet トンネルの設定」](#)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。概要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	手順の参照先
/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	151 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	146 ページの「 /etc/ppp/pppoe.if ファイル 」
/usr/sbin/sppptun	PPPoE トンネルに関する Ethernet インタフェースを構成するためのコマンド	147 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。この目的には、`/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` ファイルには、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のリストが記述されています。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 pppoe hme0
hme0:pppoe
# /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:pppoe
hme1:pppoe
hme1:pppoe
hme2:pppoe
hme2:pppoe
```

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:pppoe
# 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、ダイヤルアップリンクでのダイヤルアウトマシンとダイヤ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。

- [147 ページの「/usr/sbin/sppptun コマンド」](#)
- [149 ページの「/usr/lib/inet/pppoe デーモン」](#)
- [149 ページの「/etc/ppp/pppoe ファイル」](#)
- [151 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 クライアントからのサービスのためのブロードキャストを受け入れます。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、想定 PPPoE クライアントからのブロードキャストにตอบสนอง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。

<i>global-options</i>	<code>/etc/ppp/pppoe</code> 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 <i>global-options</i> 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。 <code>/etc/ppp/pppoe</code> 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。 <code>devices</code> 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。 <i>device interface <interface></i>
-----------------------	---

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* は、提供されるサービスに適した任意の文字列です。

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 <code>pppd</code> 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 <code>pppd</code> 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 <code>hme1</code> に限定します。
pppd "debug name internet-server"	呼び出し元が <code>pppd</code> 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 <code>internet-server</code> 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、`/etc/ppp/pppoe` File で示した [149 ページの「`/etc/ppp/pppoe` ファイル」](#) 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、`pppoe` 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、サーバーは 3 つのサブネット green、orange、および purple に接続されています。pppoed は、サーバー上で root として実行します (デフォルト)。

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 インタフェースを通じて 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 オプションにより、サーバー名が、chap-secrets ファイルで使用される green-server に設定されます。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
nodefaulttroute
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リスト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 `pppoe` 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。

- [147 ページの「/usr/sbin/sppptun コマンド」](#)
- [154 ページの「/usr/lib/inet/pppoe ユーティリティー」](#)
- [155 ページの「pppoe.so 共有オブジェクト」](#)
- [123 ページの「/etc/ppp/peers/peer-name ファイル」](#)
- [118 ページの「/etc/ppp/options 構成ファイル」](#)

`/usr/lib/inet/pppoe` ユーティリティー

`/usr/lib/inet/pppoe` 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。`pppoe` は、`chat` ユーティリティーに似ています。`pppoe` 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/pppoe` 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 によってロードされ、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 をロードします。

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 によって検出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ロードするように指示します。
- `connect "/usr/lib/inet/pppoec device"` – 接続を開始します。次に、`connect` は PPPoE に `plumb` されているインタフェース `device` を介して pppoec ユーティリティを起動します。

`/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。ダイヤ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 リンクに必要な最小数に制限するようにしてください。

次の例は、[85 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
<code>sppptun</code>	<code>sppptun</code> をシリアルデバイスの名前として定義します。
<code>plugin pppoe.so</code>	<code>pppd</code> に <code>pppoe.so</code> 共有オブジェクトを読み込むように指示します。
<code>connect "/usr/lib/inet/pppoc hme0"</code>	<code>pppoc</code> 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 <code>hme0</code> を指定します。
<code>noccp</code>	接続上で CCP 圧縮をオフに設定します。 注記 - 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。
<code>noauth</code>	<code>pppd</code> 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。
<code>user Red</code>	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 <code>Red</code> の名前を設定します。
<code>password redsecret</code>	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 <code>redsecret</code> を定義します。
<code>noipdefault</code>	初期 IP アドレスとして 0.0.0.0 を割り当てます。
<code>defaultroute</code>	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 <code>pppd</code> に指示します。リンクがシステムのインターネットへのリンクである場合、これは PPPoE クライアントに該当し、 <code>/etc/ppp/peers/peer-name</code> 内に <code>defaultroute</code> を含める必要があります。

非同期 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 実装である非同期 Solaris PPP (asppp) が提供されていました。asppp を実行するピアを最新の PPP 4.0 に更新する場合は、変換スクリプトを実行する必要があります。この章では、PPP 変換に関する次のトピックについて説明します。

- 157 ページの「asppp ファイルを変換する前に」
- 160 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 の相違点の説明については、14 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` を使用すると、標準 asppp 構成を構成する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 の詳細については、<http://docs.oracle.com/cd/E19455-01/index.html> から入手可能な『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

161 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      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
ipdptp0

ipadm create-addr -T  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
static -a              ipdptp0 からリモートピア gobi へのリンクを確立する
local=mojave,remote=gobi
ipdptp0/ppaddr

inactivity_timeout    2 分間非アクティブな回線を終了する
120

interface             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
ipdptp0               する

peer_system_name      リモートピアの名前 Pgobi を指定する
Pgobi
```

/etc/uucp/Systems ファイルの例

161 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

161 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

161 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 Oracle 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、ダイヤ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、ダイヤルアウトマシン `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

UNIX-to-UNIX Copy Program について

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。この章の内容は次のとおりです。

- 165 ページの「UUCP のハードウェア構成」
- 166 ページの「UUCP ソフトウェア」
- 169 ページの「UUCP データベースファイル」

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 と呼ばれる基本ネットワークユーティリティ (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構成するすべてのファイルとユーティリティ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 (*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 (*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 ケーブルで接続することにより、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ヤルします。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実行されてい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を検査します。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` ファイルが見つかったら、`uuxqt` はそのファイル

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、uuxqt は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 uudemmon.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 uudemmon.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します。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 の使用が記録されます。詳細は、 uucp(1C) のマニュアルページを参照してください。
uucleanup	スプールディレクトリをクリーンアップします。uucleanup は通常、cron によって起動される uudemmon.cleanup シェルスクリプトから実行されます。詳細は、 uucleanup(1M) のマニュアルページを参照してください。
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ます。Uutry は uucico デーモンを呼び出して、このマシンと指定さ

れたりリモートコンピュータとの間の通信リンクを確立します。詳細は、[Uutry\(1M\)](#) のマニュアルページを参照してください。

uuccheck 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します。**uuccheck** はまた、`/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を検査します。詳細は、[uuccheck\(1M\)](#) のマニュアルページを参照してください。

UUCP ユーザープログラム

UUCP 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します。**cu** を使用すれば、初期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ます。詳細は、[cu\(1C\)](#) のマニュアルページを参照してください。

uucp あるマシンから別のマシンへファイルをコピーします。**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** デーモンを呼び出し、このデーモンは、リモートコンピュータへの接続を試みます。詳細は、[uucp\(1C\)](#) のマニュアルページを参照してください。

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します。**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してください。

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します。[uuto\(1C\)](#) のマニュアルページを参照してください。

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します。詳細は、[uux\(1C\)](#) のマニュアルページを参照してください。

uustat リクエストされた転送 (**uucp**, **uuto**, **uux**) のステータスを表示します。また、**uustat** はキューに入っている転送を制御する手段も提供します。詳細は、[uustat\(1C\)](#) のマニュアルページを参照してください。

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ます。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ます。
Devices	ネットワーク通信を構成するために使用されます。
Dialcodes	Systems ファイルエントリの電話番号フィールド内で使用できるダイヤルコード省略名が入っています。これは必須ではないが、UUCP のほかに <code>asppp</code> 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、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ます。Dialers は、UUCP のほかに <code>asppp</code> でも使用されます。
Grades	リモートコンピュータのキューにジョブを入れるときにユーザーが指定できるジョブグレードと、各ジョブグレードに関連付けられたアクセス権を定義します。
Limits	このマシンで同時に実行できる <code>uucico</code> 、 <code>uuxqt</code> 、および <code>uusched</code> の最大数を定義します。
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します。
Poll	このシステムがポーリングするマシンと、ポーリングする時点を定義します。
Sysfiles	<code>uucico</code> と <code>cu</code> が、Systems、Devices、および Dialers ファイルとして使用する、複数または異なるファイルを割り当てます。
Sysname	TCP/IP ホスト名のほかに、各マシンに固有の UUCP 名を定義できるようにします。
Systems	<code>uucico</code> デーモン、 <code>cu</code> 、および <code>asppp</code> が、リモートコンピュータへのリンクを確立するとき必要となる情報が入っています。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、[169 ページの「UUCP データベースファイル」](#)に示すファイルで構成されます。ただし、基本的な UUCP 構成に関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、フォルトファイルに含まれていないダイヤラを追加する予定がある場合を除き、通常はデフォルトのままで使用できます。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。[169 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。

注記 - 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

UNIX-to-UNIX Copy Program の管理

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、手順と障害の解明についての情報が記載されています。

- 171 ページの「UUCP 管理 (タスクマップ)」
- 172 ページの「UUCP のログインの追加」
- 173 ページの「UUCP の起動」
- 175 ページの「TCP/IP を介した UUCP の実行」
- 176 ページの「UUCP のセキュリティと保守」
- 178 ページの「UUCP のトラブルシューティング」

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	手順の参照先
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	172 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	173 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	TCP/IP 用 UUCP を有効にするため /etc/uucp/Systems ファイルを編集します。	175 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	178 ページの「モデムまたは ACU の障害確認方法」
	送信をデバッグするための診断手順を実行する	179 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`) 着信要求が正しく処理されるように、各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 11.2 でのユーザーとプロセスのセキュリティ保護』の「割り当てられている管理権利の使用」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、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する 4 つのシェルスクリプトが含まれています。4 つのスクリプトは次のとおりです。

- uudemmon.poll
- uudemmon.hour
- uudemmon.admin
- uudemmon.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行う必要があります。Oracle Solaris の全体インストールを選択した場合は、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/uudemmon.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、uudemmon.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 "@(#)uudemmon.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/uudemmon.admin
#20 3 * * * /usr/lib/uucp/uudemmon.cleanup
#0 * * * * /usr/lib/uucp/uudemmon.poll
#11,41 * * * * /usr/lib/uucp/uudemmon.hour
```

注記 - 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、uudemmon.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

uudemmon.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 11.2 でのユーザーとプロセスのセキュリティ保護』の「割り当てられている管理権利の使用」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` ファイルを有効にします。

```
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 ファイル内のいずれかのマシン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 管理ログインに送ります。

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収集し、それらのファイルをマージし、ほかの古いログ情報が入っている `/var/uucp/.Old` ディレクトリに入れます。
- 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します。
- 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、UUCP 管理ログイン (uucp) にメールで送信します。

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 11.2 でのユーザーとプロセスのセキュリティ保護』の「割り当てられている管理権利の使用」を参照してください。

2. `/etc/uucp/Systems` ファイルを編集し、対象エントリに次のフィールドが含まれ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ため、*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、UUCP 用のポートを設定します。**

```
uucp    540/tcp    uucpd        # uucp daemon
```

このエントリは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` プロパティが定義されていない場合は、`config/default` プロパティを確認します。

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 11.2 でのシステムサービスの管理](#)』の「SMF サービスインスタンスの管理」

5. **(オプション) 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、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。

定期的な UUCP の保守

UUCP の保守に必要な作業の量はさほど多くはありません。ただし、`How to Start UUCP` のセクションで述べたように、[173 ページの「UUCP の起動方法」](#) ファイルが正しい場所に置かれているかどうかを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、忘れずに `newaliases` コマンドを実行してください。

UUCP 公開ディレクトリ

ディレクトリ `/var/spool/uucppublic` は、すべてのシステムで UUCP がデフォルトでファイルのコピーできる場所です。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イック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、このディレクトリにコピーされ `uucp` が所有しているファイルをユーザーが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、セキュリティのために設定されているステイック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うのを防ぐために、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

次の手順では、UUCP に関する一般的な問題を解決する方法を説明します。

▼ モデムまたは ACU の障害確認方法

モデムや ACU が適正に動作していないかどうかを、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 11.2 でのユーザーとプロセスのセキュリティ保護](#)』の「[割り当てられている管理権利の使用](#)」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取得します。

```
# 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。

<code>machine</code>	コピー先のマシンの名前を指定する
<code>/dir/file</code>	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。

注記 - 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。比較的大きなテキストについて 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

特定のマシンに接続しようとする時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検証してください。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

基本的なネットワーク情報の検査に使用できるさまざまなコマンドがあります。

- `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、このコマンドは `Permissions` ファイルも検査して、設定されているアクセス権に関する情報を表示します。

◆◆◆ 第 12 章

UNIX - to-UNIX Copy Program ファイル

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。この章の内容は次のとおりです。

- 183 ページの「UUCP /etc/uucp/Systems ファイル」
- 191 ページの「UUCP /etc/uucp/Devices ファイル」
- 198 ページの「UUCP /etc/uucp/Dialers ファイル」
- 202 ページの「その他の基本的な UUCP 構成ファイル」
- 205 ページの「UUCP /etc/uucp/Permissions ファイル」
- 214 ページの「UUCP /etc/uucp/Poll ファイル」
- 215 ページの「UUCP /etc/uucp/Config ファイル」
- 215 ページの「UUCP /etc/uucp/Grades ファイル」
- 218 ページの「その他の UUCP 構成ファイル」
- 220 ページの「UUCP の管理ファイル」
- 222 ページの「UUCP のエラーメッセージ」

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する必要がある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 ファイルを使用すると、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。204 ページの「UUCP /etc/uucp/Sysfiles ファイル」の詳細は、UUCP /etc/uucp/Sysfiles File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 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
ssword:beledi

/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「[/etc/uucp/Systems のエントリ](#)」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「[/etc/uucp/Systems のエントリ](#)」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 分であるとする。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が入ります。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 つの英字と速度を含めることができます (たとえば C1200、D1200)。[194 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:0akgrass
```

このフィールドに情報を指定する必要がない場合は、フィールドのプレースホルダーとしてダッシュ (-) を使用してください。

/etc/uucp/Systems ファイルの Phone フィールド

このフィールドには、自動ダイヤラ (ポートセクタ) 用の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ヤルするという 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ヤルするという指示になります。

コンピュータがポートセクタに接続されている場合は、そのセ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。これら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ます。`-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` が送られてくると想定します。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 `" "` (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:xyzyy
```

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 `xyzyy` を送ります。

次の表に、便利なエスケープ文字をいくつか示します。

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
<code>\b</code>	バックスペース文字を送信または想定します。

エスケープ文字	意味
<code>\c</code>	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。
<code>\d</code>	後続の文字を送る前に 1 - 3 秒の遅延が生じます。
<code>\E</code>	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。
<code>\e</code>	エコーチェックをオフにします。
<code>\H</code>	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。
<code>\K</code>	BREAK 文字を送信します。
<code>\M</code>	CLOCAL フラグをオンにします。
<code>\m</code>	CLOCAL フラグをオフにします。
<code>\n</code>	改行文字を送信または想定します。
<code>\N</code>	NULL 文字 (ASCII NUL) を送信します。
<code>\p</code>	約 1/4 秒間または 1/2 秒間、一時停止します。
<code>\r</code>	キャリッジリターンを送信または想定します。
<code>\s</code>	スペース文字を送信または想定します。
<code>\t</code>	タブ文字を送信または想定します。
<code>EOT</code>	EOT とそれに続く 2 個の改行文字を送信します。
<code>BREAK</code>	BREAK 文字を送信します。
<code>\ddd</code>	8 進数 (<i>ddd</i>) 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

企業によっては、リモートコンピュータからの呼び出しを処理するダイヤ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ヤ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。ダイヤルインサーバーは、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。ダイヤ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、ダイヤ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 の直前と直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使用可能にしています。

```
unix Any ACU 2400 12015551212 "" \r ogin: Puucp ssword:Passuan "" \ STTY=crtscts
```

この擬似送信文字列は、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を検査し、パリティに誤りがあると回線を切断することがあるため、パリティ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文字列ペアを示します。

```
"" P_EVEN          パリティを偶数 (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せん。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、[194 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

このフィールドは、デバイスによって確立されるリンクの種類を記述します。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。

ポートセクタ

ポートセクタは、Type フィールド内で使用される変数であり、ポートセクタの名前に置き換えられます。ポートセクタは、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` ポートセクタと `develcon` ポートセ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、これによりリンクがこのマシンへの直接リンクであることが示されます。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「Devices ファイルと Systems ファイルの Type フィールドの比較」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 (これにも ACUEC というエントリが入っています) の Type フィールドと突き合わされます。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。特定のエントリに対応付けられているモデムが /dev/cua/a デバイス (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

このフィールドはプレースホルダーです。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 型のダイヤラですが、この種類は Oracle Solaris OS ではサポートされていません。801 型以外のダイヤ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を使用し、もう 1 つのネットワークは外線通信に使用するという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、モデムまたは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。

対応するダイヤラによっては、最後のダイヤ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 部分の有効エントリは、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ヤラタイプのうちの 1 つとなります。これらの特殊ダイヤ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、特殊なダイヤ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* の部分 (*hayes*) だけが示されている点に注意してください。これは、ダイヤラに渡す *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「ポートセクタに接続されたモデム用 UUCP Dialer フィールド」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、直接リンクエントリ `Direct` と `System-Name` の両方にあてはまります。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
```

上記に示すように、`token` 部分が空です。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。

ポートセクタに接続しているモデム – ポートセ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

プロトコル	説明
f	このプロトコルは X.25 接続を介した伝送に使用されます。f は、データストリームのフロー制御に関係しており、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ます。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、モデムを介した使用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、電話番号をダイヤ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、7 番目の位置から始まる奇数番号の各 Devices フィールドは、Dialers ファイル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」用としてダイヤ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、Oracle 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ある Dialers ファイルのエントリの例をいくつか示します。

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制します。
\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	エコーチェックを無効にします。
\E	低速デバイス用にエコーチェックを使用します。
\K	ブレイ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、その結果 = はすべて W (発信音待ち) に置き換えられ、- はすべて P (一時停止) に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
- > - > を待ちます。
- Q\c - キャリッジリターンを付けずに Q を送信する
- : - : を想定します。
- \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
- > - > を待ちます。
- 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
- : - : を待ちます。
- \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つてから処理を行う。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
- > - > を待ちます。
- 9\c - 改行を付けずに 9 を送信する
- OK - 文字列 OK を待ちます。

/etc/uucp/Dialers ファイルでの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。[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を検査し、パリティに誤りがあると回線を切断することがあるため、パリティのリセットが必要になります。expect-send の文字列ペアとして P_ZERO を使用すると、パリティが 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文字列ペアを示します。

```
"" P_EVEN          パリティを偶数 (デフォルト) に設定する
"" P_ODD           パリティを基数に設定する
"" P_ONE           パリティを 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 フィールドで使用できるダイヤ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ヤルシーケンスです。

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

Abbreviation	Dial-Sequence
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、uucico は Dialcodes ファイルで NY を検索し、それに該当するダイヤルシーケンス 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ヤルシーケンスです。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、ふたたび 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、ダイヤラとトークンのペアの中のトークンが \T であれば、uucico は 9=847-7867 というシーケンスをダイヤラに送ります。

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異なるファイルを割り当てます。cu の詳細については、[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。

- 異なる 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。
- 異なる Dialers ファイルにより、cu と uucp で異なるハンドシェイ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 (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を指定します。
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を指定します。

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 サービス用に異なる 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ど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には、一般にノード名と呼ばれる識別名が設定され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します。

注記 - 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します。

LOGNAME エントリには LOGNAME オプションが含まれます。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 エントリだけに含まれている必要があります。
- 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ます。

- 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ます。
- 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りません (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の設定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

リモートコンピュータは、ローカルコンピュータを呼び出して作業を完了したあとで、ローカルコンピュータによりリモートコンピュータ用にキューに入れられたその作業を取得しようとする場合があります。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 オプションを設定できます。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、UUCP VALIDATE Option で述べる [211 ページの「UUCP VALIDATE オプション」](#) 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指定できます。

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的ためのものです。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) 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ます。`NOWRITE` も `NOREAD` オプションと同様に機能します。`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。

注記 - 2つのサイトが互いに `CALLBACK` オプションを設定すると、通信が開始されません。

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、ユーザー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ユーザー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、`COMMAND` 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り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ます。
- 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ます。

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にとって危険性のあるものとなります。

UUCP VALIDATE オプション

マシンのセキュリティ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は常に、COMMANDS オプションとともに VALIDATE オプションを使用します。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、十分に検討してください。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 デーモンは、実行時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 リストを取得できます。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 エントリの間係を示しています。

```
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、リスト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 のいず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 のいずれか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いず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 \  
LOGNAME=uucpz VALIDATE=eagle:owl:hawk
```

```
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、中間マシンの `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 *cfile*) 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の定義が含まれています。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バインドできます。*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* に関連付けられているものなので、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 *User-job-grade* とシステムジョブグレードのバインディングを定義できます。*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、およびそれがバインドされているシステムジョブグレードとして、キーワード `default` を使用する必要があります。*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ますようにします。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。

<code>nnnn</code>	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
<code>nK</code>	K バイト数を表す 10 進数 (K はキロバイトの略号)
<code>nM</code>	M バイト数を表す 10 進数 (M はメガバイトの略号)
<code>Any</code>	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 バイトを表します
- `10K` は 10K バイトを表します
- `2M` は 2M バイトを表します

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
<code>User</code>	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
<code>Non-user</code>	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
<code>Group</code>	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
<code>Non-group</code>	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のキューに入れることが許可または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* は、Stream にプッシュする順序での STREAMS モジュールの名前に置き替えられます。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 usched 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、usched のいずれかであり、 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エントリです。

```
service=uucico max=5
service=uuxqt max=5
service=usched max=2
```

この例では、5 つの uucico、5 つの uuxqt、2 つの usched をマシンで実行できます。

UUCP remote.unknown ファイル

通信機能の使用に影響を与えるその他のファイルとして、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となります。

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となります。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際に、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動されます。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. <i>sys</i>	<i>sys</i> はファイルを使用しているコンピュータ名を表す
LCK. <i>dev</i>	<i>dev</i> はファイルを使用しているデバイス名を表す
LCK. <i>LOG</i>	<i>LOG</i> 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* オプションまたは *uuto -p* オプションが指定されている場合は、ダミー名 (*D.0*) が使用されます。
- 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せん。

- 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
- 戻りステータスの要求のためのオプション行です

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	説明またはアクション
CAN'T OPEN	open() または fopen() が失敗した
CAN'T WRITE	write()、fwrite()、fprintf()、または類似のコマンドが失敗しました。
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ました。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ありませんでした。
BAD UID	ユーザー ID がパスワードデータベース内にはない。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

エラーメッセージ	説明またはアクション
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記述されています (Class フィールドまたは Speed フィールド)。
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ました。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 メッセージ

エラーメッセージ	説明またはアクション
OK	ステータスは良好です。
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	名前どおりの機能です。
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。ダイヤラが不良であるか、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
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 ファイルでエラーメッセージを調べ、222 ページの「UUCP の ASSERT エラーメッセージ」セクションを参照してください。
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

エラーメッセージ	説明またはアクション
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してください。
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、初期ハンドシェイクに失敗した
CALLER SCRIPT FAILED	このエラーは通常 DIAL FAILED と同じです。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。

メッセージ番号	説明	意味
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、「フォークできません」「パイプを作成できません」などの条件に使用されるものです。たとえば、このエラーには passwd ファイルに存在しないユーザーに対する getuid の戻りが含まれます。
72	Critical OS File Missing	/etc/passwd や /var/admin/utmpx などのシステムファイルが存在しないか、開くことができないか、または構文エラーなどのエラーがあります。
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きました。
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。
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。このメッセージ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

数字・記号

- (ダッシュ)

- Line2 フィールドのプレースホルダー, 193, 193
- Speed フィールドのプレースホルダー, 187, 187
- ダイヤルコード省略名, 187, 187

- (ハイフン)

- Line2 フィールドのプレースホルダー, 193
- Speed フィールドのプレースホルダー, 187
- ダイヤルコード省略名, 187

.ppprc ファイル

- 作成, 55
- 定義, 114
- 特権, 116

.Status ディレクトリ, 181

/etc/asppp.cf 構成ファイル, 158

/etc/inet/services ファイル

- UUCP の検査, 176

/etc/mail/aliases ファイル

- UUCP と, 177

/etc/passwd ファイル

- UUCP ログインの許可, 172

/etc/ppp/chap-secrets ファイル

- アドレス指定
 - sppp ユニット番号による, 145
 - 静的, 144
- 構文, 140
- 作成
 - 信頼できる呼び出し元用に, 80
- 定義, 114
- 例, PPPoE アクセスサーバー用, 154

/etc/ppp/myisp-chat.tmpl テンプレート, 128

/etc/ppp/options.tmpl テンプレート, 118

/etc/ppp/options.ttya.tmpl テンプレート, 121

/etc/ppp/options.ttyname ファイル

- ダイヤルアウトマシン, 49
- ダイヤルアウトマシン用, 121

ダイヤルインサーバー, 56, 120

- 定義, 114, 119
- 動的アドレス指定, 143
- 特権, 116
- 例の一覧, 121

/etc/ppp/options ファイル

/etc/ppp/options.tmpl テンプレート, 118

CHAP 認証用の name オプション, 79

PAP 認証の変更, 75

PPPoE の例, 153

作成

- ダイヤルアウトマシン, 49
- ダイヤルインサーバー, 56

定義, 114, 118

特権, 116

例の一覧, 119

/etc/ppp/pap-secrets ファイル

アドレス指定

- sppp ユニット番号による, 145
- 静的, 144

構文, 137

作成

- PPPoE アクセスサーバー, 90
- ダイヤルインサーバー, 70

信頼できる呼び出し元用に作成, 73

定義, 114

例, PPPoE アクセスサーバー用, 153

/etc/ppp/peers/myisp.tmpl テンプレート, 124

/etc/ppp/peers/peer-name ファイル

便利なオプション, 124

/etc/ppp/peers/peer-name ファイル

作成

- 専用回線リンクの終端, 64

定義, 114, 123

特権, 116

変更

- PAP 認証用に, 75
- PPPoE クライアントの, 86
- 例、PPPoE クライアント, 155
- 例の一覧, 125
- /etc/ppp/peers ディレクトリ, 114
- /etc/ppp/pppoe.device ファイル
 - アクセスサーバー, 90
 - 構文, 151
 - 定義, 151
- /etc/ppp/pppoe.if ファイル
 - 作成
 - PPPoE クライアント, 85
 - アクセスサーバーの, 88
 - 定義, 146
 - 例, 147
- /etc/ppp/pppoe ファイル
 - 構文, 149
 - サービスのリスト, 88
 - 変更, 89
 - 例, 150, 152
- /etc/uucp/Config ファイル
 - 形式, 215
 - 説明, 169, 215
- /etc/uucp/Devconfig ファイル
 - 形式, 218
 - 説明, 169, 218
- /etc/uucp/Devices ファイル
 - Class フィールド, 194
 - Dialer-Token-Pairs フィールド, 194, 197
 - Line2 フィールド, 193
 - Line フィールド, 193
 - Systems ファイル、Speed フィールドと, 187
 - Systems ファイル、Type フィールド, 193
 - Type フィールド, 192
 - 形式, 192
 - 説明, 169, 169, 191
 - プロトコル定義, 197, 198
 - 例、asppp 構成の, 159
- /etc/uucp/Dialcodes ファイル, 169, 202
- /etc/uucp/Dialers ファイル
 - 説明, 169, 198
 - 例, 199
 - 例、asppp 構成の, 160
- /etc/uucp/Grades ファイル
 - ID-list フィールド, 217, 218
 - Job-size フィールド, 217
 - Permit-type フィールド, 217
 - System-job-grade フィールド, 216, 217
 - User-job-grade フィールド, 216, 216
 - キーワード, 216, 217
 - 説明, 169, 215
 - デフォルトグレード, 216
- /etc/uucp/Limits ファイル
 - 形式, 219
 - 説明, 169, 219
- /etc/uucp/Permissions ファイル
 - CALLBACK オプション, 209, 210
 - COMMANDS オプション, 210, 211, 214
 - LOGNAME
 - MACHINE との結合, 213
 - 説明, 206
 - リモートコンピュータ用のログイン ID, 206
 - MACHINE
 - LOGNAME との結合, 213
 - OTHER オプション, 213
 - 説明, 206
 - デフォルトのアクセス権または制約, 206
 - MYNAME オプション, 208
 - NOREAD オプション, 209
 - NOWRITE オプション, 209
 - OTHER オプション, 213
 - READ オプション, 208, 209
 - REQUEST オプション, 207
 - SENDFILES オプション, 207
 - uuchek コマンドと, 168
 - uuxqt デーモンと, 166
 - VALIDATE オプション, 211, 213
 - WRITE オプション, 208, 209
 - エントリの構造化, 206
 - 形式, 206
 - 考慮事項, 206, 207
 - セキュリティーの設定, 177
 - 説明, 169, 205
 - ダイヤルバックのアクセス権, 209, 210
 - 転送操作, 214
 - ノード名の変更, 208
 - ファイル転送のアクセス権, 207, 209
 - リモート実行のアクセス権, 210, 213
- /etc/uucp/Poll ファイル
 - 形式, 214
 - 説明, 169, 214

- /etc/uucp/Sysfiles ファイル
 - Systems リストの出力, 205
 - 形式, 204
 - 説明, 169, 204
 - 例, 204
 - /etc/uucp/Sysname ファイル, 169, 205
 - /etc/uucp/Systems ファイル
 - Chat Script フィールド, 187, 190
 - Devices ファイル、Class フィールド, 194
 - Devices ファイル、Type フィールド, 193
 - Phone フィールド, 187
 - Speed フィールド, 186, 187
 - System-Name フィールド, 184
 - TCP/IP 構成, 175, 176
 - Time フィールド
 - Never エントリ, 207
 - 説明, 185
 - Type フィールド, 186
 - エスケープ文字, 188
 - 形式, 184
 - 説明, 169, 183
 - ダイヤルコード省略名, 169
 - トラブルシューティング, 180, 180
 - パリティの設定, 190
 - ハードウェアのフロー制御, 190, 190
 - 複数または異なるファイル, 169, 184, 204
 - 例、asppp 構成の, 158
 - /usr/bin/cu コマンド
 - Systems リストの出力, 205
 - 説明, 168
 - 複数または異なる構成ファイル, 169, 204
 - モデムや ACU の検査, 179
 - /usr/bin/uucp コマンド
 - 説明, 168
 - 転送操作のアクセス権, 214
 - 伝送のデバッグ, 179
 - による uucico の実行, 166
 - ログイン ID のホームディレクトリ, 167
 - /usr/bin/uulog コマンド, 167, 181
 - /usr/bin/uupick コマンド, 168, 178
 - /usr/bin/uustat コマンド, 168, 178
 - /usr/bin/uuto コマンド
 - 公開ディレクトリファイルの削除, 178
 - 説明, 168
 - による uucico の実行, 166
 - /usr/bin/uux コマンド
 - 説明, 168
 - による uucico の実行, 166
 - /usr/lib/uucp/uuccheck コマンド, 168, 181
 - /usr/lib/uucp/uucleanup コマンド, 167
 - /usr/lib/uucp/Uutry コマンド, 167, 179, 180
 - /usr/sbin/inetd デーモン
 - によって呼び出される in.uucpd, 167
 - /usr/sbin/sppptun コマンド、定義, 147
 - /var/spool/uucppublic ディレクトリの保守, 178
 - /var/uucp/.Admin/errors ディレクトリ, 180
 - /var/uucp/.Status ディレクトリ, 181
 - = (等号)
 - ダイヤルコード省略名, 187
 - = (等号)、ダイヤルコード省略名内, 187
 - 8 進数エスケープ文字, 200
- あ**
- アクセスサーバー (PPP)
 - /etc/ppp/chap-secrets ファイル, 154
 - /etc/ppp/options ファイル, 153
 - /etc/ppp/pap-secrets ファイル, 153
 - PPPoE クライアントに対するインタフェースの限定使用, 89
 - 構成、PPPoE, 87, 89, 152
 - 構成のタスクマップ, 83
 - 構成のためのコマンドとファイル, 148, 149, 150
 - タスクマップの計画, 42
 - 定義, 26
 - アドレス割り当て
 - PPP, 143, 144, 145
 - 一時 (TM) UUCP データファイル, 220
 - インタフェース (PPP)
 - /usr/sbin/sppptun による PPPoE インタフェースの plumb, 147
 - HSI/P 構成スクリプト, 63
 - PPPoE アクセスサーバー用の構成, 88
 - PPPoE クライアントに対するインタフェースの限定使用, 89
 - PPPoE クライアント用の構成, 85, 85
 - 参照 /etc/ppp/pppoe.if ファイル
 - PPPoE のアクセスサーバーの構成, 146
 - PPP ダイアルアウトの非同期インタフェース, 19
 - PPP ダイアルインの非同期インタフェース, 20

- 専用回線の同期, 22
- インバウンド通信
 - UUCP chat スクリプトを使用した有効化, 189
 - コールバックのセキュリティ, 209, 210
- エコーチェック, 200
- エスケープ文字
 - Dialers ファイルの send 文字列, 200
 - Systems ファイルの chat スクリプト, 188
- オプション (PPP)
 - asynmap, 120
 - auth, 71
 - call , 58, 123
 - connect, 52, 134
 - crtstcts, 49
 - debug, 96
 - init, 65, 120
 - local, 65
 - login, 71, 139
 - name, 75
 - noauth, 52, 65
 - noccp, 55
 - noipdefault, 52
 - noservice, 153
 - passive, 65
 - persist, 65
 - pppd デーモンによる解析, 115
 - sync, 65
 - xonxoff, 57
 - オプション特権, 116
 - 使用上のガイドライン, 113
- か**
- 改行エスケープ文字 , 200
- 開始
 - chat スクリプトを使用したダイアルバックの有効化, 189
 - 有効化
 - エコーチェック, 200
- 管理コマンド (UUCP), 167, 168
- 管理ファイル (UUCP)
 - 一時データファイル (TM), 220
 - クリーンアップ, 175
 - 作業ファイル (C.), 220, 221
- 実行ファイル (x.), 166, 221
- ロックファイル (LCK), 220
- キーワード
 - Devices ファイル、Type フィールド, 192
 - Grades ファイル, 216, 217, 218
- 起動
 - UUCP シェルスクリプト, 173, 175
- キャリッジリターンエスケープ文字, 200
- キュー (UUCP)
 - uusched デーモン
 - 説明, 167
 - 同時実行の最大数, 169, 219, 219
 - 管理ファイル, 220, 221
 - クリーンアップコマンド, 167
 - ジョブグレード定義, 215, 218
 - スケジューリングデーモン, 167
 - スプールディレクトリ, 220
- 広域ネットワーク (WAN)
 - Usenet, 165, 183
- 公開ディレクトリの保守 (UUCP), 178
- 公開ディレクトリファイルのスティッキービット, 178
- 構成
 - UUCP
 - TCP/IP ネットワーク, 175, 176
 - シェルスクリプト, 173, 175
 - データベースファイル, 170
 - ログインの追加, 172, 172
 - UUCP データベースへの asppp リンク, 170
- 構成ファイル
 - UUCP, 215
- コールバック
 - chat スクリプトを使用したダイアルバックの有効化, 189
 - Permissions ファイルオプション, 210, 210
- コマンド
 - UUCP のトラブルシューティング, 181
 - 実行 (X.)UUCP ファイル, 166, 221
 - リモート実行、UUCP による, 207, 210, 213
- さ**
- サービスデータベース
 - UUCP ポート, 176
- 作業 (C.)UUCP ファイル
 - クリーンアップ, 175
 - 説明, 220, 221

- シェルスクリプト (UUCP), 173, 175
 - uudemon.admin, 174, 175
 - uudemon.cleanup, 175
 - uudemon.hour
 - uuxqt デーモンの実行, 166
 - 説明, 174
 - による uusched デーモンの実行, 167
 - uudemon.poll, 174, 215
 - 自動実行, 173
 - 手動実行, 173
 - 資格
 - CHAP 認証, 77
 - PAP 認証, 69
 - 実行 (X.)UUCP ファイル
 - uuxqt の実行, 166
 - クリーンアップ, 175
 - 説明, 221
 - 自動呼び出し装置 (ACU)
 - Devices ファイル、Type フィールド, 192
 - UUCP ハードウェア構成, 165
 - トラブルシューティング, 178
 - 受動モード, 207
 - シリアルポート
 - 構成
 - ダイアルアウトマシン, 47
 - ダイアルインサーバー, 54
 - ダイアルインサーバーでの構成, 120
 - 信頼できる呼び出し側, 24
 - 信頼できる呼び出し元
 - CHAP 認証の構成, 80
 - スクリプト
 - chat スクリプト (UUCP), 190
 - expect フィールド, 187, 188
 - エスケープ文字, 188
 - 基本的なスクリプト, 188
 - 形式, 187
 - ダイアルバックの有効化, 189
 - シェルスクリプト (UUCP), 173, 175
 - スケジューリングデーモン、UUCP 用, 167
 - スプール (UUCP)
 - uusched デーモン
 - 説明, 167
 - 同時実行の最大数, 169, 219, 219
 - 管理ファイル, 220, 221
 - クリーンアップコマンド, 167
 - ジョブグレード定義, 215, 218
 - ディレクトリ, 220
 - スペースエスケープ文字, 200
 - 静的アドレス指定
 - PPP, 144
 - セキュリティー
 - UUCP
 - COMMANDS オプション、Permissions ファイル, 210, 211
 - VALIDATE オプション、Permissions ファイル, 211, 213
 - 公開ディレクトリファイルのスティッキービット, 178
 - 設定, 177
 - 専用回線リンク
 - CSU/DSU, 22
 - demand スクリプト, 65
 - 一般的な問題の診断
 - 概要, 110
 - ネットワーク, 98
 - 計画, 33, 34, 36, 63
 - 構成, 34
 - 構成のタスクマップ, 61
 - 構成例, 34
 - 通信プロセス, 23
 - 定義, 21
 - 同期インタフェースの構成, 62
 - 媒体, 23
 - ハードウェア, 33
 - リンクの構成要素, 21
 - リンクの認証, 24
- た**
- ダイアルアウトマシン
 - /etc/ppp/options.ttyname でのシリアル回線の構成, 121
 - chat スクリプトの作成, 50
 - アドレス指定
 - 静的, 144
 - 動的, 143
 - 計画情報, 30
 - 構成
 - CHAP 認証, 79, 81
 - PAP 認証, 73
 - シリアル回線通信, 49
 - シリアルポート, 47

- ピアとの接続, 51
- モデム, 47
- 構成のタスクマップ, 46
- 定義, 18
- リモートピアの呼び出し, 58
- ダイヤルアップリンク
 - chat スクリプト
 - ISDN TA 用, 133
 - UNIX スタイルのログイン, 130
 - テンプレート, 128
 - 例, 127, 129, 134
 - chat スクリプトの作成, 126
 - 一般的な問題の診断
 - pppd による, 95
 - シリアル回線, 107
 - ネットワーク, 98
 - 計画, 30, 31, 31
 - 構成ファイルのテンプレート, 46
 - ダイヤルアッププロセス, 20
 - タスクマップ, 45
 - 定義, 18
 - ピアの呼び出しの開始, 58
 - リンクの構成要素, 18
 - リンクの認証, 24
 - 例, 31
- ダイヤルインサーバー
 - UUCP, 189
 - 計画情報, 31, 55
 - 構成
 - CHAP 認証, 77, 79
 - PAP 認証, 69, 70, 71
 - シリアル回線通信, 56, 120
 - シリアルポート, 54
 - モデム, 54
 - 構成のタスクマップ, 53
 - 定義, 18
 - 呼び出しの受信, 58
- ダイヤルコード省略名, 169, 187
- ダイヤルバック
 - CALLBACK オプション、Permissions ファイル, 209, 210
 - chat スクリプトを使用した有効化, 189
- 端末アダプタ (TA) 用 chat スクリプト, 133, 134
- 遅延エスケープ文字, 200
- チャレンジハンドシェイク認証プロトコル (CHAP)
 - /etc/ppp/chap-secrets の構文, 140
 - 構成のタスクマップ, 76
 - 定義, 140
 - 認証処理, 143
- チャンレンジハンドシェイク認証プロトコル (CHAP)
 - 構成例, 39
- 直接リンク、UUCP 構成, 165
- 停止
 - 無効化
 - エコーチェック, 200
- ディレクトリ (UUCP)
 - エラーメッセージ, 180
 - 管理, 167
 - 公開ディレクトリの保守, 178
- データ (D.)UUCP ファイル
 - クリーンアップ, 175
- デジタル加入者線アクセスマルチプレクサ (DSLAM)、PPPoE 用, 28
- デバイス伝送プロトコル, 197, 198
- デバッグ
 - UUCP 転送, 179, 180
- 電子メール
 - UUCP の保守, 177
- 転送操作 (UUCP), 214
- 転送速度、UUCP 通信リンクの, 186, 187
- テンプレートファイル (PPP)
 - /etc/ppp/myisp-chat.tmpl, 128
 - /etc/ppp/options.tmpl, 118
 - /etc/ppp/peers/myisp.tmpl, 124
 - options.ttya.tmpl, 121
 - テンプレートのリスト, 46
- 電話回線
 - UUCP 構成, 165
- 電話番号、Systems ファイル, 187
- 同期 PPP 参照 専用回線リンク
 - 同期デバイスの構成, 62
- 動的アドレス指定
 - PPP, 143
- トークン (ダイアラとトークンのペア), 194, 197
- トラブルシューティング
 - UUCP, 178, 224
 - ASSERT エラーメッセージ, 180, 222, 223
 - STATUS エラーメッセージ, 181, 223, 224
 - Systems ファイルの検査, 180
 - エラーメッセージの検査, 180, 224
 - 基本情報の検査, 181
 - 障害のあるモデムや ACU, 178

伝送のデバッグ, 179, 180
 トラブルシューティング用のコマンド, 181
 トンネル
 構成のタスクマップ, 83
 構成例, 43, 44
 定義 (PPP), 26

な
 名前と命名
 ノード名
 UUCP 別名, 169, 208
 UUCP リモートコンピュータ, 184, 205
 認証, 24
 参照 認証 (PPP)
 一般的な問題の解決, 111
 認証される側 (PPP), 24
 認証する側 (PPP), 24
 認証 (PPP)
 CHAP 資格データベースの構成, 77
 CHAP 資格の構成, 80
 CHAP の構成, 76
 参照 チャレンジハンドシェイク認証プロトコル (CHAP)
 ダイアルアウトマシン, 81
 ダイアルインサーバー, 77, 79
 CHAP の例, 39
 PAP の構成, 68
 参照 パスワード認証プロトコル (PAP)
 PAP の例, 37
 計画, 36, 39
 構成の前提条件, 37
 構成のタスクマップ, 67, 68, 76
 信頼できる呼び出し側, 24
 専用回線のサポート, 24
 デフォルトのポリシー, 24
 認証される側, 24
 認証する側, 24
 秘密ファイル
 PAP, 70
 PPP, 24
 プロセス図
 PAP の, 138
 ネットワークデータベースサービス、UUCP ポート, 176
 ノード名
 UUCP 別名, 169, 208

UUCP リモートコンピュータ, 184, 205

は

ハードウェア
 UUCP
 構成, 165
 ポートセレクタ, 192
 フロー制御
 Dialers ファイル, 201, 202
 Systems ファイル, 190, 190
 ハードウェアのフロー制御
 Dialers ファイル, 201, 202
 Systems ファイル, 190, 190
 パスワード
 UUCP、特権を持つ, 212, 212
 パスワード認証プロトコル (PAP)
 /etc/ppp/pap-secrets ファイル, 137
 PAP 資格データベースの作成, 69
 計画, 68
 構成
 信頼できる呼び出し側, 74
 信頼できる呼び出し元, 73, 75
 ダイアルインサーバー, 71
 構成例, 37
 タスクマップ, 68
 定義, 136
 パスワードのヒント, 137
 バックスラッシュエスケープ文字, 200
 Dialers ファイルの send 文字列, 200
 Systems ファイルの chat スクリプト, 188
 パリティー
 Dialers ファイル, 202
 Systems ファイル, 190
 ピア
 PPPoE クライアント, 26, 41
 アクセスサーバー, 26, 42
 専用回線のピア, 22
 ダイアルアウトマシン, 18
 ダイアルインサーバー, 18
 定義, 17
 認証される側, 24
 認証する側, 24
 非同期 PPP (asppp)
 Solaris PPP 4.0 との違い, 15

- Solaris PPP 4.0 への変換, 161
- UUCP データベースの構成, 170
 - 構成内のファイル, 157
 - ドキュメント, 14
- ファイル転送 (UUCP)
 - アクセス権, 207, 209
 - 作業ファイル C., 220, 221
 - デーモン, 166, 166
 - トラブルシューティング, 179, 180
- ブレイクエスケープ文字
 - Dialers ファイル, 200
- フレームリレー, 22, 61
- プロセス図
 - CHAP の, 142
- プロトコル定義、Devices ファイル, 197, 198
- ポイントツーポイントプロトコル 参照 PPP
- ポート
 - Devices ファイルのエントリ, 193
 - UUCP, 176

ま

- 無効化
 - エコーチェック, 200
- メッセージ
 - UUCP
 - ASSERT エラーメッセージ, 222, 223
 - STATUS エラーメッセージ, 223, 224
 - エラーメッセージの検査, 180
- モデム
 - モデムの問題の解決, 103
- モデム (PPP)
 - chat スクリプト
 - ISDN TA 用, 133
 - UNIX スタイルのログイン, 130
 - テンプレート, 128
 - 例, 50, 127, 129, 134
 - chat スクリプトの作成, 126
 - DSL, 28
 - 構成
 - ダイアルアウトマシン, 47
 - ダイアルインサーバー, 54
 - モデム速度の設定, 54
- モデム (UUCP)
 - UUCP データベース
 - DTP フィールド、Devices ファイル, 197

- UUCP データベースの DTP フィールド、Devices ファイル, 195, 196
- UUCP ハードウェア構成, 165
- 直接接続, 195
- 特性の設定, 190, 190, 201, 202
- トラブルシューティング, 178
- ポートセレクト接続, 196, 197

や

- 有効化
 - chat スクリプトを使用したダイアルバックの有効化, 189
 - エコーチェック, 200

ら

- リモートコンピュータのポーリング (UUCP), 169, 214
- リモート実行 (UUCP)
 - コマンド, 207, 210, 213
 - 作業ファイル C., 220, 221
 - デーモン, 166
- 例、PPP 構成 参照 PPP の構成例
- ローカルエリアネットワーク (LAN)
 - UUCP 構成, 166
- ロギング
 - UUCP ログファイルのクリーンアップ, 175
 - UUCP ログファイルの表示, 167
- ログイン (UUCP)
 - 追加, 172, 172
 - 特権, 212, 212
- ロック (LCK) UUCP ファイル, 220

A

- ACU キーワード、Type フィールド, 192
- aliases ファイル, 177
- ALL 変数、COMMANDS オプション, 211
- Any、Time フィールドのエントリ, 184
- Any キーワード
 - Grades ファイル (UUCP), 216, 218
 - Speed フィールド (UUCP), 186
- asppp2pppd 変換スクリプト

- Solaris PPP 4.0 に変換されたファイルの表示, 161
 - Solaris PPP 4.0 への変換, 161
 - 標準 asppp 構成, 157
 - asppp 参照 非同期 PPP (asppp)
 - ASSERT エラーメッセージ (UUCP), 180, 222, 223
 - asynmap オプション (PPP), 120
 - Australian National University (ANU) PPP
 - Solaris PPP 4.0 との互換性, 14
 - auth オプション (PPP), 71
- B**
- b エスケープ文字
 - Dialers ファイル, 200
- C**
- C.UUCP 作業ファイル
 - クリーンアップ, 175
 - 説明, 220, 221
 - c エスケープ文字
 - Dialers ファイル, 200
 - call オプション (PPP)
 - ダイヤルインサーバーの呼び出し, 58
 - CALLBACK オプション、Permissions ファイル, 209, 210
 - CHAP 資格データベース
 - 作成
 - 信頼できる呼び出し元に, 80
 - ダイヤルインサーバーの, 77
 - Chat Script フィールド
 - /etc/uucp/Systems ファイル, 187
 - chat スクリプト
 - chat スクリプトの設計, 126
 - 実行可能な chat プログラムの作成, 136
 - 呼び出す、PPP で, 135
 - 例 (PPP)
 - ISDN TA の, 133, 134
 - ISP を呼び出すためのスクリプト, 129
 - UNIX スタイルのログイン chat スクリプト, 130
 - UNIX 方式ログインの chat スクリプト, 50
 - 基本のモデム chat スクリプト, 127
 - Class フィールド
 - Devices ファイル, 194
 - COMMANDS オプション、Permissions ファイル, 210, 214
 - VALIDATE オプション, 213
 - connect オプション (PPP)
 - chat スクリプトを呼び出すには, 134
 - 例, 52
 - crontab ファイル
 - UUCP 用, 173
 - crtsects オプション (PPP), 49
 - CSU/DSU
 - 一般的な問題の解決, 110
 - 構成, 63
 - 定義, 22
 - cu コマンド
 - Systems リストの出力, 205
 - 説明, 168
 - 複数または異なる構成ファイル, 169, 204
 - モデムや ACU の検査, 179
- D**
- D. UUCP データファイル
 - クリーンアップ, 175
 - D エスケープ文字, 197
 - d エスケープ文字
 - Dialers ファイル, 200
 - d オプション
 - cu コマンド, 179
 - day エントリ、Time フィールド, 185
 - default キーワード、User-job-grade フィールド, 216
 - Devconfig ファイル
 - 形式, 218
 - 説明, 169, 218
 - Devices ファイル
 - Class フィールド, 194
 - Dialer-Token-Pairs フィールド, 194, 197
 - Line2 フィールド, 193
 - Line フィールド, 193
 - Systems ファイル、Speed フィールドと, 187
 - Systems ファイル、Type フィールド, 193
 - Type フィールド, 192
 - 形式, 192
 - 説明, 169, 169, 191
 - 複数または異なるファイル, 204

プロトコル定義, 197, 198
Dialcodes ファイル, 169, 202
Dialer-Token-Pairs フィールド
Devices ファイル
 同じポートセレクタ, 196
 構文, 194
 ダイアラタイプ, 195
 ポートセレクタ接続, 196
Dialers ファイル
 説明, 169, 198
 例, 199
direct キーワード、DTP フィールド, 194
Direct キーワード、Type フィールド, 192
DSL 参照 PPPoE
DSL モデム, 28

E

e エスケープ文字
 Dialers ファイル, 200
E エスケープ文字
 Dialers ファイル, 200
e プロトコル、Devices ファイル, 197
errors ディレクトリ (UUCP), 180
expect フィールド、Chat Script フィールド, 187, 188

F

f プロトコル、Devices ファイル, 198

G

g プロトコル、Devices ファイル, 197
Grades ファイル
 ID-list フィールド, 217, 218
 Job-size フィールド, 217
 Permit-type フィールド, 217
 System-job-grade フィールド, 216, 217
 User-job-grade フィールド, 216, 216
 キーワード, 216, 217
 説明, 169, 215
 デフォルトグレード, 216

Group キーワード、Permit-type フィールド, 217

I

ID-list フィールド、Grades ファイル, 217, 218
in.uucpd デーモン, 167
inetd デーモン
 によって呼び出される in.uucpd, 167
init コマンド
 PPP と, 65

J

Job-size フィールド、Grades ファイル, 217

K

k エスケープ文字
 Dialers ファイル, 200

L

-l オプション
 cu コマンド, 179
LCK UUCP ロックファイル, 220
Limits ファイル
 形式, 219
 説明, 169, 219
Line2 フィールド、Devices ファイル, 193
Line フィールド、Devices ファイル, 193
local オプション (PPP), 65
login オプション (PPP)
 /etc/ppp/pap-secrets, 139
 /etc/ppp/pap-secrets 内の, 75
 ダイヤルインサーバー用の /etc/ppp/options, 71
LOGNAME Permissions ファイル
 MACHINE との結合, 213
 SENDFILES オプション, 207
 VALIDATE オプション, 211, 213
 説明, 206
 リモートコンピュータ用のログイン ID, 206

M

MACHINE Permissions ファイル
 COMMANDS オプション, 210, 211
 LOGNAME との結合, 213
 OTHER オプション, 213
 説明, 206
 デフォルトのアクセス権または制約, 206
 MYNAME オプション、Permissions ファイル, 208

N

n エスケープ文字
 Dialers ファイル, 200
 N エスケープ文字
 Dialers ファイル, 200
 name オプション (PPP)
 /etc/ppp/pap-secrets 内の, 75
 CHAP 認証用, 79
 noservice, 153
 Never、Time フィールドのエントリ, 207
 newaliases コマンド
 UUCP と, 177
 newline エスケープ文字, 200, 200
 nnn エスケープ文字, 200
 noauth オプション (PPP), 52, 65
 noccp オプション (PPP), 55
 noipdefault オプション (PPP), 52
 Non-group キーワード、Permit-type フィールド, 217
 Non-user キーワード、Permit-type フィールド, 217
 NOREAD オプション、Permissions ファイル, 209, 209
 noservice オプション (PPP), 153
 NOWRITE オプション、Permissions ファイル, 209, 209
 Null エスケープ文字, 200

O

options.ttyname ファイル (PPP) 参照 /etc/ppp/
 options.ttyname
 options ファイル、PPP, 49
 Oracle Solaris
 UUCP バージョン, 183

OTHER オプション、Permissions ファイル, 213

P

p エスケープ文字
 Dialers ファイル, 200
 PAP 資格データベース
 作成
 信頼できる呼び出し元, 73
 ダイアルインサーバー, 70
 ダイアルインサーバーの作成, 69
 PAP 認証の構成, 69, 73, 74, 75
 passive オプション (PPP), 65
 passwd ファイル
 UUCP ログインの許可, 172
 Password Authentication Protocol (PAP)
 login オプションの使用, 139
 認証プロセス, 138
 penril エントリ、Dialers ファイル, 200
 Permissions ファイル
 CALLBACK オプション, 209, 210
 COMMANDS オプション, 210, 211, 214
 LOGNAME
 MACHINE との結合, 213
 説明, 206
 リモートコンピュータ用のログイン ID, 206
 MACHINE
 LOGNAME との結合, 213
 OTHER オプション, 213
 説明, 206
 デフォルトのアクセス権または制約, 206
 MYNAME オプション, 208
 NOREAD オプション, 209
 NOWRITE オプション, 209
 OTHER オプション, 213
 READ オプション, 208, 209
 REQUEST オプション, 207
 SENDFILES オプション, 207
 uucheck コマンドと, 168
 uuxqt デーモンと, 166
 VALIDATE オプション, 211, 213
 WRITE オプション, 208
 エントリの構造化, 206
 形式, 206
 考慮事項, 206, 207

- セキュリティの設定, 177
- 説明, 169, 205
- ダイヤルバックのアクセス権, 209, 210
- 転送操作, 214
- ノード名の変更, 208
- ファイル転送のアクセス権, 207, 209
- リモート実行のアクセス権, 210, 213
- Permit-type フィールド, Grades ファイル, 217
- persist オプション (PPP), 65
- Phone フィールド, Systems ファイル, 187
- Poll ファイル
 - 形式, 214
 - 説明, 169, 214
- Port Selector 変数, Devices ファイル, 192
- PPP
 - asppp との違い, 15
 - chat スクリプト例, 50
 - DSL のサポート, 25
 - ISDN のサポート, 20
 - pppd, 58
 - 参照 pppd コマンド
 - PPPoE, 26
 - PPP 計画のタスクマップ, 29
 - RFC の関連情報, 16
 - 一般的な問題, 94
 - 概要, 13
 - 構成ファイルのオプション 参照 オプション (PPP)
 - 構成ファイルのサマリー, 114
 - 互換性, 14
 - 情報、外部, 16
 - 専用回線リンク, 21
 - ダイヤルアップリンク, 18
 - 認証, 23, 25
 - 非同期 PPP からの変換, 161
 - ファイル特権, 116
 - 問題解決, 93
 - 参照 PPP のトラブルシューティング
 - リンクの構成要素, 17, 26
- PPP の構成タスク
 - PPPoE トンネル, 83
 - 構成の問題の診断, 102
 - 専用回線, 61
 - ダイヤルアップリンク, 45
 - 認証, 67
- PPP の構成例
 - CHAP 認証, 39
 - PAP 認証, 37
 - PPPoE トンネル, 43
 - 専用回線リンク, 34
 - ダイヤルアップリンク, 31
- PPP の初期設定スクリプト demand, 65
- PPP の診断
 - debug オプション, 96
 - PPPoE トンネルのログファイル, 108
 - オンに設定する
 - pppd で, 95
 - 専用回線リンク, 95
 - ダイヤルアップリンク, 95
- PPP のデバッグ
 - chat スクリプトのデバッグ, 104
 - PPPoE の問題の診断, 108
 - 通信の問題の解決, 101, 102
 - デバッグをオンに設定する, 96
 - ネットワークの問題の診断, 98
 - モデムの問題の解決, 103
- PPP のトラブルシューティング
 - 一般的な問題, 94
 - chat スクリプト, 105, 105, 106
 - PPP 構成, 102
 - 一般的な通信, 101
 - シリアル回線, 107
 - 専用回線リンク, 110
 - 認証, 111
 - ネットワーク, 100
 - シリアル回線の問題の診断, 107
 - 診断情報の取得, 95, 95, 96
 - タスクマップ, 93
- PPP の秘密ファイル 参照 /etc/ppp/pap-secrets
- ファイル
- PPP のリンクタイプ
 - 専用回線, 21
 - ダイヤルアップ, 18
 - ダイヤルアップと専用回線の比較, 21
 - 物理リンク媒体, 17
 - リンクの構成要素, 17
- PPP の chat プログラム 参照 chat スクリプト
- PPP の -debug オプション, 96
- PPP リンク上の ISDN, 20
- pppd コマンド
 - DSL 回線のテスト, 86
 - オプションの解析, 115

- 診断情報の取得, 95, 95, 108
 - 定義, 114
 - デバッグをオンに設定する, 96
 - 呼び出しの開始, 58
 - pppdebug ログファイル, 108
 - pppoe.so 共有オブジェクト, 151, 155
 - PPPoE
 - DSLAM, 28
 - snoop トレースの取得, 109
 - アクセスサーバーからのサービスの提供, 149, 151
 - アクセスサーバーの構成, 87, 89, 90
 - 一般的な問題の解決, 108, 108, 109
 - 概要, 26
 - 構成のタスクマップ, 83
 - コマンドとファイルの一覧, 146
 - トンネルの計画, 41, 41, 43, 44
 - PPPoE クライアント
 - /etc/ppp/peers/*peer-name* ファイルの使用 (PPPoE), 155
 - アクセスサーバーと, 155
 - アクセスサーバーの定義, 85
 - 機器, 41
 - 計画, 41, 84
 - 構成, 85
 - 構成のタスクマップ, 83
 - コマンド, 154
 - 定義, 26
 - ファイル, 154
 - pppoec ユーティリティ
 - 診断情報の取得, 108
 - 定義, 154
 - pppoed デーモン
 - 起動, 88
 - 定義, 149
- Q**
- q オプション
 - uustat コマンド, 178
- R**
- r エスケープ文字
 - Dialers ファイル, 200
 - r オプション
 - uucp コマンド, 179
 - Uucry コマンド, 179
 - READ オプション、Permissions ファイル, 208, 209
 - remote.unknown ファイル, 219
 - REQUEST オプション、Permissions ファイル, 207
 - Requests for Comments (RFC)
 - PPP, 16
 - retry サブフィールド、Time フィールド, 185, 185
 - RS-232 電話回線
 - UUCP 構成, 165
- S**
- s エスケープ文字
 - Dialers ファイル, 200
 - SENDFILES オプション、Permissions ファイル, 207
 - snoop トレース
 - PPPoE, 109
 - Solaris PPP 4.0 参照 PPP
 - Speed フィールド
 - Devices ファイル、Class フィールド, 194
 - Systems ファイル, 186, 187
 - sppp ユニット番号
 - PPP アドレス割り当て, 145
 - STATUS エラーメッセージ (UUCP), 181, 223, 224
 - STREAMS
 - デバイス構成, 218
 - STTY フロー制御, 190, 190, 201, 202
 - sync オプション (PPP), 65
 - Sys-Name 変数、Type フィールド, 192
 - Sysfiles ファイル
 - Systems リストの出力, 205
 - 形式, 204
 - 説明, 169, 204
 - 例, 204
 - Sysname ファイル, 169, 205
 - System-job-grade フィールド、Grades ファイル, 216, 217
 - Systems ファイル
 - Chat Script フィールド, 187, 190
 - Devices ファイル、Class フィールド, 194
 - Devices ファイル、Type フィールド, 193
 - Phone フィールド, 187
 - Speed フィールド, 186, 187

- System-Name フィールド, 184
 - TCP/IP 構成, 175, 176
 - Time フィールド
 - Never エントリ, 207
 - 説明, 185
 - Type フィールド, 186
 - エスケープ文字, 188
 - 形式, 184
 - 説明, 169, 183
 - ダイアルコード省略名, 169, 187
 - トラブルシューティング, 180, 180
 - パリティの設定, 190
 - ハードウェアのフロー制御, 190, 190
 - 複数または異なるファイル, 169, 184, 204
 - Systems ファイルの System-Name フィールド, 184
 - Systems ファイルの Time フィールド, 185
- T**
- T エスケープ文字
 - Devices ファイル, 197
 - Dialers ファイル, 197, 200
 - t プロトコル, Devices ファイル, 197
 - TCP/IP ネットワーク
 - UUCP, 175
 - を介した UUCP, 176
 - Time フィールド, Systems ファイル, 207
 - TM UUCP 一時データファイル, 220
 - Type フィールド
 - Devices ファイル, 192
 - Systems ファイル, 186
- U**
- uname - n コマンド, 205
 - Usenet, 165, 183
 - User-job-grade フィールド, Grades ファイル, 216, 216
 - User キーワード, Permit-type フィールド, 217
 - uuccheck コマンド, 168, 181
 - uucico デーモン
 - Dialcodes ファイル, 203
 - Systems ファイルと, 183
 - Systems リストの出力, 205
 - UUCP ログインの追加, 172, 172
 - uusched デーモンと, 167
 - Uutry コマンドと, 167
 - 説明, 166, 166
 - 同時実行の最大数, 169, 219, 219
 - 複数または異なる構成ファイル, 169, 184, 204
 - uucleanup コマンド, 167
 - UUCP
 - Oracle Solaris バージョン, 165, 183
 - remote execution
 - 作業ファイル C., 220, 221
 - STREAMS 構成, 218
 - 管理コマンド, 167, 168
 - 管理ファイル, 220, 221
 - 公開ディレクトリの保守, 178
 - 構成
 - TCP/IP を介した UUCP の実行, 176
 - TCP/IP を介した UUCP の実行, 175
 - UUCP ログインの追加, 172, 172
 - コールバックオプション, 209, 210
 - シェルスクリプト, 173, 175
 - 手動でパラメータをオーバーライドする, 215
 - 受動モード, 207
 - スプール
 - クリーンアップコマンド, 167
 - ジョブグレード定義, 215, 218
 - スケジューリングデーモン, 167
 - セキュリティ
 - COMMANDS オプション, Permissions ファイル, 210, 211
 - VALIDATE オプション, Permissions ファイル, 211, 213
 - 公開ディレクトリファイルのスティッキービット, 178
 - 設定, 177
 - 説明, 165, 183
 - ディレクトリ
 - エラーメッセージ, 180
 - 管理, 167
 - 公開ディレクトリの保守, 178
 - 定期的な保守, 178
 - 転送操作, 214
 - 転送速度, 186, 187, 194
 - データベースファイル, 169, 219
 - asppp 構成, 170
 - 基本構成ファイル, 170
 - 説明, 169, 170

- 複数または異なるファイル, 169, 184, 204
- デーモン
 - 概要, 166, 167
- 特権ログインとパスワード, 212, 212
- トラブルシューティング, 224
 - ACU の障害, 178
 - ASSERT エラーメッセージ, 180, 222, 223
 - STATUS エラーメッセージ, 181, 223, 224
 - Systems ファイルの検査, 180
 - エラーメッセージの検査, 180, 224
 - 基本情報の検査, 181
 - 伝送のデバッグ, 179, 180
 - トラブルシューティング用のコマンド, 181
 - モデムの障害, 178
- ノード名
 - 別名, 169, 208
 - リモートコンピュータ, 184, 205
- ハードウェア構成, 165
- ファイル転送
 - アクセス権, 207, 209
 - 作業ファイル C., 220, 221
 - デーモン, 166, 166
 - トラブルシューティング, 179, 180
- 保守, 177, 178
- メールの蓄積, 177
- ユーザーコマンド, 168, 168
- リモートコンピュータのポーリング, 169, 214
- リモート実行
 - コマンド, 207, 210, 213
 - デーモン, 166
- ログイン
 - 追加, 172, 172
 - 特権, 212, 212
- ログインシェル, 166
- ログファイル
 - クリーンアップ, 175
 - 表示, 167
 - ログファイルの表示, 167
- uucp コマンド
 - 説明, 168
 - 転送操作のアクセス権, 214
 - 伝送のデバッグ, 179
 - による uucico の実行, 166
 - ログイン ID のホームディレクトリ, 167
- UUUCP 通信リンクの転送速度, 194
- UUUCP 通信リンク用のデバイスタイプ, 186
- UUUCP の保守
 - 公開ディレクトリ, 178
 - シェルスクリプト, 173, 175
 - 定期的な保守, 177, 178
 - メール, 177
 - ログインの追加, 172, 172
- uucppublic ディレクトリの保守, 178
- uudemon.admin シェルスクリプト, 174, 175
- uudemon.cleanup シェルスクリプト, 175
- uudemon.crontab ファイル, 173
- uudemon.hour シェルスクリプト
 - uuxqt デーモンの実行, 166
 - 説明, 174
 - による uusched デーモンの実行, 167
- uudemon.poll シェルスクリプト, 174, 215
- uudirect キーワード、DTP フィールド, 194
- uulog コマンド, 167, 181
- uname コマンド, 181
- uupick コマンド
 - 公開ディレクトリファイルの削除, 178
 - 説明, 168
- uusched デーモン
 - uudemon.hour シェルスクリプトの呼び出し, 174
 - 説明, 167
 - 同時実行の最大数, 169, 219, 219
- uustat コマンド
 - uudemon.admin シェルスクリプト, 174, 175
 - 説明, 168
 - モデムや ACU の検査, 178
- uuto コマンド
 - 公開ディレクトリファイルの削除, 178
 - 説明, 168
 - による uucico の実行, 166
- Uutry コマンド, 167, 179, 180
- uux コマンド
 - 説明, 168
 - による uucico の実行, 166
- uuxqt デーモン
 - uudemon.hour シェルスクリプトの呼び出し, 174
 - 説明, 166
 - 同時実行の最大数, 169, 219, 219

V

-v オプション

 uucheck コマンド, 181

VALIDATE オプション、Permissions ファイル, 211,
213

 COMMANDS オプション, 210, 211

W

WRITE オプション、Permissions ファイル, 208

X

X.UUCP 実行ファイル

 uuxqt の実行, 166

 クリーンアップ, 175

 説明, 221

xonxoff オプション (PPP), 57