Solaris のシステム管理 (第 2 巻)

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303
U.S.A. 650–960–1300

Part Number 806-2718-10
2000 年 3 月

原典: *System Administration Guide, Volume 2*

Part No: 805-7229-10

Revision A

Adobe PostScript

目次

はじめに

『*Solaris*™ のシステム管理 *(*第 *2* 巻*)*』は、Solaris システム管理に関する重要な情報を提供する、3 巻構成のマニュアルの第 2 巻です。SPARC™ プラットフォームおよび IA プラットフォームにおけるシステム管理について説明します。

このマニュアルでは、システム管理者である読者が SunOS™ 5.8 オペレーティングシステムをすでにインストールしていて、ネットワークソフトウェアの設定を終了していることを想定しています。SunOS 5.8 オペレーティングシステムは Solaris 8 製品の一部で、Solaris 共通デスクトップ環境 (CDE) などの多くの機能を含みます。また、SunOS 5.8 は、AT&T System V リリース 4 オペレーティングシステムに準拠しています。

システム管理者にとって重要な Solaris 8 リリースの新機能については、各章のはじめにある新機能に関する節を参照してください。

注 - Solaris オペレーティング環境は、SPARC と IA という 2 種類のハードウェア、つまりプラットフォームで動作します。Solaris オペレーティング環境は、64 ビットのアドレス空間でも 32 ビットのアドレス空間でも動作します。このマニュアルの情報は、章、節、項、注、リスト、表、例、コード例などで特に明記する場合を除き、両方のプラットフォームとアドレス空間に適用されます。

対象読者

このマニュアルは、Solaris 7 リリースを実行するシステムの管理者を対象にしています。このマニュアルを読むには、UNIX のシステム管理について 1 ～ 2 年の経験

が必要です。UNIX システム管理のトレーニングコースに参加することも、知識の習得に役立ちます。

『Solaris のシステム管理』全 3 巻の構成

3 冊のシステム管理マニュアルで説明している内容を次に示します。

『Solaris のシステム管理 (第 1 巻)』

- 「ユーザーアカウントとグループの管理」
- 「サーバーとクライアントサポートの管理」
- 「システムのシャットダウンとブート」
- 「取り外し可能な媒体の管理」
- 「ソフトウェアの管理」
- 「デバイスの管理」
- 「ディスクの管理」
- 「ファイルシステムの管理」
- 「データのバックアップと復元」

『Solaris のシステム管理 (第 2 巻)』

- 「印刷サービスの管理」
- 「リモートシステムの利用」
- 「端末とモデムの管理」
- 「システムセキュリティの管理」
- 「システム資源の管理」
- 「システム性能の管理」
- 「Solaris ソフトウェアで発生する問題の解決」

『Solaris のシステム管理 (第 3 巻)』

- 「ネットワークサービストピック」
- 「IP アドレス管理トピック」
- 「モデム関連ネットワークサービスのトピック」
- 「遠隔ファイルシステムへのアクセスについてのトピック」
- 「メールサービスについてのトピック」
- 「ネットワークサービスの監視についてのトピック」

Sun のマニュアルの注文方法

専門書を扱うインターネットの書店 Fatbrain.com から、米国 Sun Microsystems™, Inc. (以降、Sun™ とします) のマニュアルをご注文いただけます。

マニュアルのリストと注文方法については、`http://www1.fatbrain.com/documentation/sun` の Sun Documentation Center をご覧ください。

Sun のオンラインマニュアル

`http://docs.sun.com` では、Sun が提供しているオンラインマニュアルを参照することができます。マニュアルのタイトルや特定の主題などをキーワードとして、検索を行うこともできます。

表記上の規則

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

表 P–1 表記上の規則

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

ただし AnswerBook2™ では、ユーザーが入力する文字と画面上のコンピュータ出力は区別して表示されません。

コード例は次のように表示されます。

- C シェルプロンプト

```
system% command y|n [filename]
```

- Bourne シェルおよび Korn シェルのプロンプト

```
system$ command y|n [filename]
```

- スーパーユーザーのプロンプト

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

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

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

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

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

一般規則

- このマニュアルでは、英語環境での画面イメージを使っています。このため、実際に日本語環境で表示される画面イメージとこのマニュアルで使っている画面イメージが異なる場合があります。本文中で画面イメージを説明する場合には、日本語のメニュー、ボタン名などの項目名と英語の項目名が、適宜併記されています。
- このマニュアルでは、「IA」という用語は、Intel 32 ビットのプロセッサアーキテクチャを意味します。これには、Pentium、Pentium Pro、Pentium II、Pentium II Xeon、Celeron、Pentium III、Pentium III Xeon の各プロセッサ、および AMD、Cyrix が提供する互換マイクロプロセッサチップが含まれます。

印刷サービスの管理

ここでは、Solaris 環境で印刷サービスを管理する手順について説明します。次の章が含まれます。

第 2 章	ネットワーク上で印刷サービスを管理する場合の概要について説明します。プリンタサーバー、印刷クライアント、および LP 印刷サービスを取り上げます。
第 3 章	ネットワーク上で印刷サービスの計画を作成する場合の概要について説明します。ネットワーク上のシステム資源の割り当てやプリンタの定義が含まれます。
第 4 章	システム上でプリンタを設定し、ネットワーク上の他のシステムが利用できるようにする詳しい手順を説明します。
第 5 章	プリンタの削除、印刷ポリシーの設定、印刷要求の管理など、プリンタを管理する詳しい手順を説明します。
第 6 章	文字セット、印刷フィルタ、フォーム、およびフォントを設定し管理する詳しい手順を説明します。
第 7 章	プリンタポート特性の調整や、サポートされていないプリンタの `terminfo` エントリの追加など、LP 印刷サービスをカスタマイズする詳しい手順を説明します。
第 8 章	LP 印刷サービスの参照情報を説明します。

第 2 章

印刷管理の概要

この章では、Solaris 環境のプリンタの管理について基本的なことがらを説明します。この章の内容は次のとおりです。

印刷管理作業の詳しい手順については、次の章を参照してください。

- 第 4 章
- 第 5 章
- 第 6 章
- 第 7 章

印刷での新規機能

この節では、Solaris 8 リリースの新しい印刷機能について説明します。

Solaris プリンタマネージャ

Solaris プリンタマネージャは Java ベースのグラフィカルユーザーインタフェースで、ローカルおよびリモートのプリンタ構成を管理できます。このツールを使用で

きるネームサービス環境は、NIS、NIS+、フェデレーテッド・ネーミング・サービス (xfn) を使用する NIS+、および files (/etc ファイルを使用する環境) です。このツールを使用するには、スーパーユーザーとしてログインする必要があります。

プリンタ構成情報の管理には、Admintool: プリンタ (Printer) よりも Solaris プリンタマネージャの使用をお勧めします。Solaris プリンタマネージャをネームサービスとともに使用すれば、プリンタ情報を一元的に管理できるためです。ネームサービスを使用してプリンタ構成情報を格納するとよいのは、ネットワーク上のすべてのシステムからプリンタ情報にアクセスできるようになり、印刷管理が容易になるためです。

このリリースでは、下層にある xfn アプリケーション層を持たない NIS+ ネームサービス環境で、プリンタ構成情報を Solaris プリンタマネージャによって管理できます。そのため、プリンタ構成情報により速くアクセスできます。NIS+ (xfn) プリンタ情報を NIS+ プリンタ情報に変換する方法については、92ページの「NIS+ (+xfn) のプリンタ構成情報を NIS+ 形式に変換する方法」を参照してください。

Solaris プリンタマネージャは、プリンタサーバー、印刷クライアント、ネームサービスデータベースにあるプリンタ情報を認識します。印刷クライアントで Solaris 2.6 リリースまたは互換性のあるバージョンが動作している限り、新しい Solaris プリンタマネージャを使用するために変換作業は必要ありません。

Solaris プリンタマネージャのパッケージは `SUNWppm` です。

印刷ネーミングの拡張

この Solaris リリースは、ネームサービススイッチファイル `/etc/nsswitch.conf` に指定された `printers` データベースをサポートします。`printers` データベースは、ネットワーク上の印刷クライアントにプリンタ構成情報を一元的に提供します。

ネームサービススイッチファイルに `printers` データベースとそれに対応する情報源を指定すると、印刷クライアントからプリンタ構成情報に自動的にアクセスできるようになるため、この情報を自分のシステムに追加する必要はありません。

次の表に、`/etc/nsswitch.conf` ファイルに指定するデフォルトの `printers` エントリを files、NIS、NIS+ の環境ごとに示します。`nisplus` キーワードは `printers.org_dir` テーブルを表します。`xfn` キーワードは FNS プリンタコンテキストを表します。

ネームサービス	デフォルトの printers エントリ
files	printers: user files
nis	printers: user files nis
nis+	printers: user nisplus files xfn

たとえば、ネームサービスが NIS の場合、印刷クライアントのプリンタ構成情報は次のソースから次に記載する順に検索されます。

- user - ユーザーの $HOME/.printers ファイルを表します。
- files - /etc/printers.conf ファイルを表します。
- nis - printers.conf.byname テーブルを表します。

詳細は、nsswitch.conf(4) のマニュアルページと『*Solaris* ネーミングの管理』を参照してください。

バナーページ印刷の有効化と無効化

この Solaris リリースでバナーページ印刷をシステム全体で有効または無効にするには、lpadmin コマンドを使用します。

このコマンドでは、バナーページを常に印刷するか、いっさい印刷しないか、lpadmin の新しいオプションを使用してバナーページ印刷をオプションにするかを指定します (always、never、optional)。バナーページ印刷をオプションにすると、デフォルトでバナーが印刷されますが、lp -o nobanner コマンドを指定してバナーページ印刷を無効にすることができます。

詳細は、106ページの「バナーページをオプションにする方法」と lpadmin(1M) のマニュアルページを参照してください。

Solaris オペレーティング環境での印刷

Solaris 印刷ソフトウェアは、ネットワーク上のプリンタへのクライアントアクセスを設定および管理するための環境を提供します。

Solaris 印刷ソフトウェアは、次のコンポーネントから構成されます。

- Solaris プリンタマネージャ。ローカルシステム上やネームサービス内の印刷構成を管理するグラフィカルユーザーインタフェースです。
- AdminTool。ローカルシステム上で印刷を管理するためのグラフィカルユーザーインタフェースです。
- LP 印刷サービスコマンド。プリンタを設定および管理するためのコマンド行インタフェースです。上記の機能に加え、他の印刷管理ツールにない機能も提供します。

印刷の設定には Solaris プリンタマネージャを使用しますが、Solaris 環境で印刷を完全に制御するためには、LP コマンドの一部を使用する必要があります。詳細は、第 5 章を参照してください。

Solaris 印刷ソフトウェアには次の制約があります。

- 以前の Solaris リリースで `s5` (System V 印刷プロトコル) として定義されたプリンタサーバーはサポートされません。
- 印刷クライアントでの印刷のフィルタリングはできません。

プリンタの管理方法の選択

プリンタ情報をネームサービスに追加すると、ネットワークのすべてのシステムからプリンタにアクセスできるようになります。さらに、プリンタに関するすべての情報が一元化されるため、プリンタ管理が一般に簡単になります。

ネームサービス	プリンタ情報を一元化する方法
ネームサービスを使用する	プリンタを NIS、NIS+、NIS+ (xfn) データベースのどれかに追加すると、プリンタはネットワーク上のすべてのシステムからアクセスできるようになる
ネームサービスを使用しない	プリンタを追加しても、プリンタ情報はプリンタサーバーの構成ファイルにしか追加されない。したがって、印刷クライアントがそのプリンタを自動的に認識することはできない。 プリンタを必要とする印刷クライアントにはプリンタ情報を追加する必要がある

表 2–1 に、印刷関連の主な作業と、印刷作業に利用できるツールを示します。

表 2–1 Solaris 印刷コンポーネントの機能

コンポーネント	対応するリリース	グラフィカルユーザーインタフェース	ネットワークプリンタの設定	印刷クライアントとプリンタサーバーの管理	**NIS**、**NIS+**、**NIS+ (xfn)** の使用
Solaris プリンタマネージャ	Solaris 8 および Solaris Easy Access Severs 3.0	あり	できる	できる	する
Admintool	Solaris 8 および互換バージョン	あり	できない	できる	しない
LP コマンド	Solaris 8 および互換バージョン	なし	できる	できる	する

表 2–1 を使用して、各ネットワーク環境に最適な印刷ツールを決めてから、プリンタの設定情報について第 4 章を参照してください。

印刷構成作業のほとんどは Solaris プリンタマネージャで完了設定できます。ただし、インタフェーススクリプトの作成や独自フィルタの追加など、特別な必要がある場合は、LP 印刷サービスコマンドを使用します。LP コマンドは、Solaris プリンタマネージャや Admintool の直接の元となるコマンドです。LP コマンドで印刷管理作業をする方法については、第 5 章で説明しています。

LP 印刷サービス

「LP 印刷サービス」は、ユーザーが作業を続けながらファイルを印刷できるようにするソフトウェアユーティリティのセットです。当初、印刷サービスは LP スプーラと呼ばれていました (LP はラインプリンタの意味ですが、現在ではレーザプリンタなどのさまざまな種類のプリンタも含まれます。スプール (Spool) は、system peripheral operation off-line の頭文字です)。

印刷サービスは、LP 印刷サービスソフトウェアとスプーラ (Solaris プリンタマネージャも含む)、管理者が提供する印刷フィルタ、ハードウェア (プリンタ、システム、およびネットワーク接続) から構成されます。

LP 印刷サービスの参照情報については、第 8 章を参照してください。

LP 印刷サービスに関するその他の情報について、以降の項で説明します。

ネットワークプリンタの管理

「ネットワークプリンタ」はネットワークに直接接続されているハードウェアデバイスであり、ネットワークを介して出力デバイスにデータを直接送信します。プリンタやネットワークに接続されたハードウェアには、固有のシステム名と IP アドレスがあります。

一般に、ネットワークプリンタには、プリンタベンダーから提供されているソフトウェアが必要です。プリンタベンダー提供のソフトウェアがあれば、必ずそのソフトウェアを使用してください。ネットワークプリンタのベンダーがソフトウェアサポートを提供していない場合には、Sun が提供するソフトウェアを利用できます。このソフトウェアはネットワークプリンタの汎用サポートを提供するもので、必ずしもプリンタが持つすべての機能を利用できるわけではありません。

ネットワークプリンタの設定手順については、第 4 章を参照してください。

プリンタの管理

プリンタサーバーと印刷クライアントの設定が完了したら、以下に示すような様々な管理作業を実行します。これらの作業は頻繁に行わなければならないこともあります。

- プリンタとリモートプリンタへのアクセスを削除する
- プリンタの状態をチェックする
- 印刷スケジューラを再起動する

プリンタ管理作業を実行する手順については、第 5 章を参照してください。

プリンタの定義の設定

ネットワーク上のプリンタの定義を設定することによって、より効果的な印刷環境をユーザーに提供できます。たとえば、サイトにあるすべてのプリンタに説明を付ければ、ユーザーはプリンタがどこにあるのかを見つけやすくなります。あるいは、プリンタのクラスを定義することにより、印刷要求を迅速に処理できます。

プリンタ定義の設定については、第 3 章を参照してください。

文字セット、フィルタ、フォーム、およびフォントの管理

作業環境とネットワーク上のプリンタの型式に応じて、LP 印刷サービスのプリンタ固有の機能を設定し、管理する必要があります。たとえば、異なる印字ホイール、フィルタ、フォームをプリンタごとに割り当てることができます。文字セット、印刷フィルタ、フォーム、およびフォントを設定し管理する方法の手順については、第 6 章を参照してください。

LP 印刷サービスのカスタマイズ

LP 印刷サービスは、ほとんどのプリンタと印刷ニーズに対応できる十分な柔軟性を持つように設計されていますが、あらゆる状況に対処できるわけではありません。LP 印刷サービスの標準機能では対処できない印刷要求が発生する場合もあります。また、LP 印刷サービスによるプリンタの処理方法に当てはまらないプリンタを使用することもあります。

LP 印刷サービスは、次のような方法でカスタマイズできます。

- 第 4 章
- 第 5 章
- 第 6 章
- 第 7 章

LP 印刷サービスのカスタマイズの概要と手順については、第7 章を参照してください。

Solaris 印刷クライアントサーバーの処理手順

この節では、Solaris で印刷がどのように進むかについて概要を説明します。

印刷クライアントの処理手順

図 2–1 に、ユーザーが要求を発行してから印刷されるまでの、印刷要求の流れを示します。

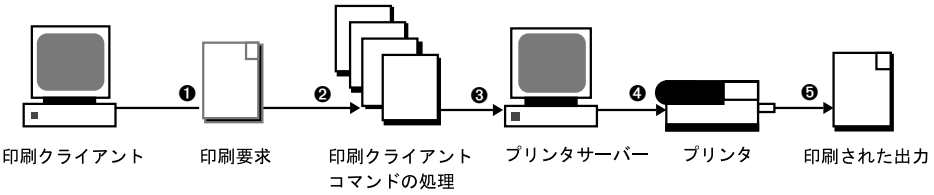


図 2–1　印刷クライアントの処理手順の概要

1. ユーザーは印刷クライアントから印刷要求を出します。
2. 印刷コマンドは印刷構成資源の階層をチェックして、印刷要求をどこに送信するか決定します。
3. 印刷コマンドは、印刷要求を適切なプリンタサーバーに直接送信します。プリンタサーバーは、BSD 印刷プロトコルを受け付ける任意のサーバーです。SVR4 (LP) プリンタサーバーや BSD プリンタサーバー (SunOS 4.1 の BSD プリンタサーバーなど) がプリンタサーバーとなります。
4. プリンタサーバーは印刷要求を適切なプリンタに送信します。
5. 印刷要求が印刷されます。

印刷クライアントの使用

この節では「印刷クライアント」、つまり印刷要求をプリンタサーバーに送信できるシステムと、印刷クライアントが印刷要求を発行するための印刷コマンドを中心に説明します。

図 2–2 に、印刷手順の中で、ユーザーが印刷クライアントから印刷要求を発行する処理を強調して示します。

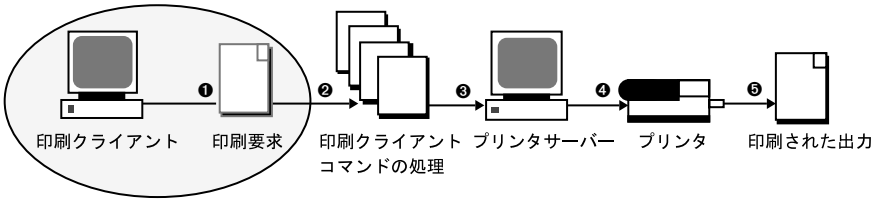


図 2–2　ユーザーが印刷クライアントから印刷要求を発行する

印刷クライアントとは

システムに Solaris 印刷ソフトウェアをインストールして、リモートプリンタにアクセスできるようにすると、そのシステムは印刷クライアントになります。

Solaris 印刷ソフトウェアは、次の資源からプリンタやプリンタ構成情報を見つけます。

- コマンド行インタフェースの `lp -d` コマンド (単独名形式または POSIX 形式)
- ユーザーの `LPDEST` 変数または `PRINTER` 変数
- `/etc/nsswitch.conf` ファイルに `printers` データベースとして設定されたソースの `_default` 変数
- ユーザーの `$HOME/.printers` ファイル
- NIS ネームサービスのローカル `/etc/printers.conf` ファイル
- NIS+ ネームサービスの `printers.org_dir` テーブル
- NIS+ (xfn) ネームサービスの FNS 印刷コンテキスト

印刷クライアントは、その要求をプリンタサーバーの待ち行列に送信します。つまり、クライアントは、ローカルの待ち行列を持ちません。クライアントが印刷要求を一時スプール領域に書き込むのは、プリンタサーバーが利用できない場合か、エラーが発生した場合だけです。サーバーまでの経路がこのように簡素化されているために、印刷クライアントは少ない資源で処理を実行でき、印刷障害の発生する可能性が減り、性能が向上します。

プリンタ構成資源

この節では、印刷ソフトウェアが、プリンタ名とプリンタ構成情報を見つけるのに使用する資源について説明します。

印刷ソフトウェアは、ネットワーク上のすべてのプリンタのプリンタ構成情報を格納するネットワーク (共有) 資源である、ネームサービスを使用できます。ネームサービス (NIS、または NIS+ (xfn)) は、プリンタ構成情報の管理を簡単にします。プリンタをネームサービスに追加すると、ネットワーク上のすべての印刷クライアントは、そのプリンタにアクセスできます。

図 2–3 に、印刷手順の中で、印刷ソフトウェアによりプリンタ構成資源の階層を調べ、どこに印刷要求を送信するか決定する処理を強調して示します。

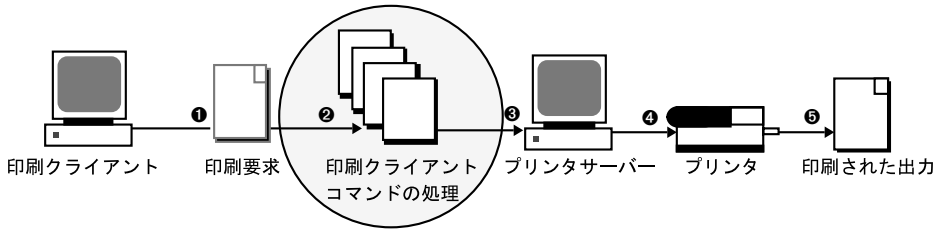
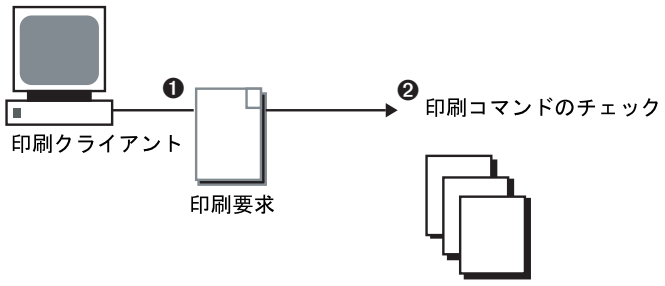


図 2–3　印刷クライアントが資源を調べてプリンタを見つける

印刷ソフトウェアがプリンタを見つける手順

図 2–4 に示すように、印刷ソフトウェアには、プリンタとプリンタ構成情報を見つけるためにより多くのオプションを指定できます。



A. 単独名、POSIX、またはコンテキストベースのプリンタ名またはプリンタクラス

B. ユーザーが PRINTER または LPDEST 環境変数に設定したデフォルトプリンタ

C. /etc/nsswitch.conf の printers データベースに設定された _default 変数

D. ユーザーの $HOME/.printers ファイル

E. ローカルの /etc/printers.conf ファイル

F. NIS の printers.conf.byname マップ

G. NIS+ の printers.org_dir テーブル

H. NIS+ (xfn) の FNS 印刷コンテキスト

図 2–4　印刷クライアントソフトウェアがプリンタを見つける手順

1. ユーザーは lp コマンドまたは lpr コマンドを使用して、印刷クライアントから印刷要求を出します。ユーザーは、次の 3 つの形式のいずれかを使用して、出力先のプリンタ名またはプリンタクラスを指定できます。

   - 単独名形式。次の例に示すように、印刷コマンドとオプションの後にプリンタ名またはプリンタクラスが続きます。

     ```
     % lp -d neptune filename
     ```

   - POSIX 形式。次の例に示すように、印刷コマンドとオプションの後に *server:printer* が続きます。

     ```
     % lpr -P galaxy:neptune filename
     ```

   - コンテキストベース形式。次の例に示すように、『*Federated Naming Service Programming Guide*』で規定されている指定形式です。

     ```
     % lpr -d thisdept/service/printer/printer-name filename
     ```

2. 印刷コマンドは、次の手順でプリンタとプリンタ構成情報を見つけます。

- ユーザーが宛先のプリンタ名またはプリンタクラスを 3 つの有効な形式のいずれかで指定しているかどうかを調べます。
- ユーザーがプリンタ名またはプリンタクラスを有効な形式で指定していない場合、ユーザーの PRINTER 環境変数または LPDEST 環境変数にデフォルトプリンタ名が指定されていないか調べます。
- どちらの環境変数にもデフォルトプリンタが指定されていない場合は、/etc/nsswitch.conf ファイルに printers データベースとして設定されたソースを調べます。

プリンタサーバーの使用

概要のこの節では、プリンタサーバーに焦点を当てて説明します。プリンタサーバーにはローカルプリンタが接続されており、プリンタサーバーは、ネットワーク上の他のシステムがそのプリンタを利用できるようにします。図 2–5 に、印刷手順の中で、プリンタサーバーが印刷要求をプリンタに送信する処理を強調して示します。

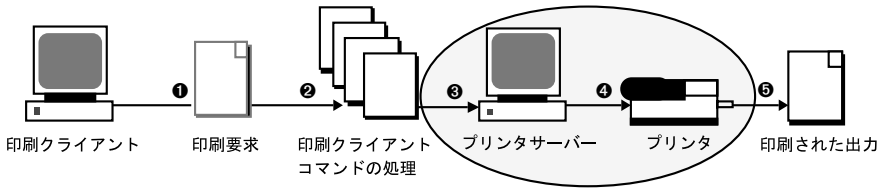


図 2–5　プリンタサーバーが印刷要求をプリンタに送信する

BSD 印刷プロトコル

印刷コマンドは、BSD 印刷プロトコルを使用します。このプロトコルの利点の 1 つは、さまざまなプリンタサーバーと通信できることです。

- SunOS 4.1 BSD (LPD) プリンタサーバー
- SunOS 5.8 プリンタサーバーおよび互換性のある SVR4 (LP) プリンタサーバー
- BSD 印刷プロトコルをサポートする、その他のプリンタサーバーやプリンタ

BSD 印刷プロトコルは業界標準のプロトコルです。このプロトコルは広く使用されており、さまざまなメーカーの異なるタイプのシステム間で互換性を提供します。Sun は、将来の相互運用性を提供するために、BSD 印刷プロトコルをサポートします。

次に進む手順

Solaris プリンタマネージャを使用して新しいプリンタを設定する詳しい手順を知りたい場合は、第 4 章に進みます。計画を立てるための情報が必要な場合は、第 3 章を参照してください。

第 3 章

ネットワーク上でのプリンタの計画方法の概要

ネットワーク上のプリンタを設定する目的は、ユーザーが 1 つまたは複数のプリンタにアクセスできるようにすることです。この章では、最も効率よくネットワーク間でプリンタを分散する方法とプリンタ設定を計画する方法について説明します。

印刷管理作業の手順については、次の章を参照してください。

- 第 4 章
- 第 5 章
- 第 6 章
- 第 7 章

ネットワーク上でのプリンタの分散

管理者として、各プリンタを 1 台のシステム専用にするのが効率がよいか、多数のシステムが利用できるようにするのがよいかを判断しなければなりません。ネットワーク環境では、複数のプリンタサーバー上にプリンタを分散するのが通常は最もよい方法です。複数のプリンタサーバーを設定する利点は、あるプリンタサーバーに問題が発生しても、別のプリンタサーバーに印刷要求を振り替えられることです。

集中化した印刷構成を採用した場合も、使い勝手をよくしたり、応答時間を短縮したりするために、プリンタをユーザーのシステムに接続できます。ユーザーのシステムに接続されたプリンタも、ネットワーク上の他のシステムから利用できます。

図 3–1 は、集中化印刷構成を採用した場合もプリンタをユーザーのシステムに接続できる例を示しています。

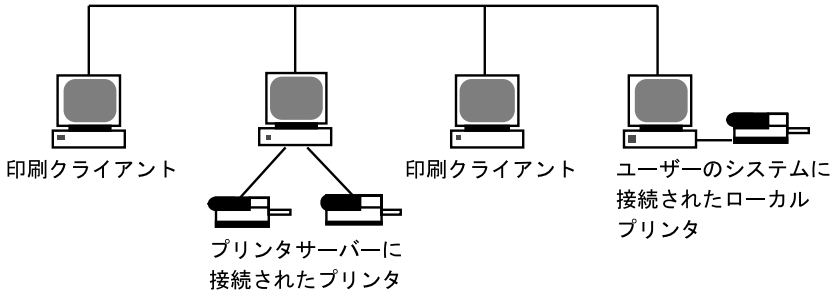


図 3–1　ネットワーク上でプリンタを分散する方法

プリンタサーバーと印刷クライアントを割り当てる

どのシステムにローカルプリンタを接続し、どのシステムでは他のシステム上のプリンタを使用するかを決定する必要があります。ローカルプリンタを接続し、そのプリンタをネットワーク経由で他のシステムでも利用できるようにしているシステムを「プリンタサーバー」と呼びます。プリンタサーバーに印刷要求を送るシステムを「印刷クライアント」と呼びます。

LP 印刷サービスは、Solaris 環境で印刷サービスを管理するソフトウェアです。プリンタをシステムに物理的に接続するだけではなく、LP 印刷サービスに対してプリンタの特性を定義し、システムをプリンタサーバーにしなければなりません。プリンタサーバーを設定し終わったら、他のシステムを印刷クライアントとして設定できます。

プリンタサーバーと印刷クライアントは、それぞれ異なるバージョンの SunOS オペレーティングシステム上で実行できます。SunOS 5.8 またはその互換バージョンのオペレーティングシステムを稼動しているシステムから、SunOS 4.1 オペレー

ティングシステムを稼動している既存のプリンタサーバーに印刷要求を送ることができ、またその逆も可能です。

注 - SunOS 5.8 は Solaris 8 オペレーティング環境に含まれます。

図 3–2 は、SunOS 5.8 および互換バージョンと SunOS 4.1 のオペレーティングシステムが動作しているシステムからなるネットワーク上の印刷構成の例を示しています。

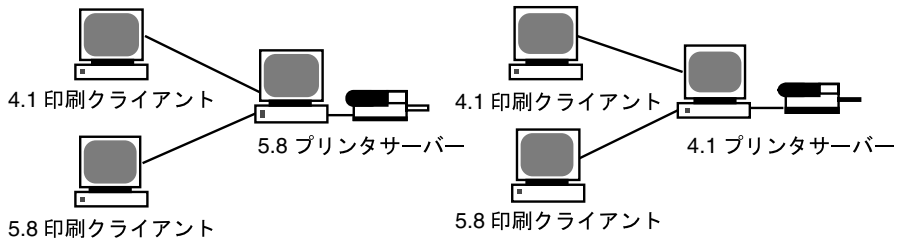


図 3–2　SunOS 5.8 システムと SunOS 4.1 システムからなる印刷構成の例

プリンタサーバーの必要事項と推奨事項

プリンタは、スタンドアロンシステムまたはネットワーク上の任意のシステムに接続できます。ネットワークに接続されていて、プリンタを備えているシステムは、印刷負荷を管理できる十分な資源があれば、どれでもプリンタサーバーとして使用できます。

スプーリング空間

「スプーリング空間」とは、印刷待ち行列内で要求を格納し、処理するためのディスク空間です。スプーリング空間は、どのシステムをプリンタサーバーとして指定するかを決定する場合の唯一重要な要素です。ユーザーがファイルの印刷要求を出すと、それらのファイルは印刷が完了するまで `/var/spool/lp` ディレクトリに格納されます。`/var` ディレクトリのサイズは、ディスクのサイズとディスクのパーティション分割方法によって異なります。スプーリング空間は、プリンタサーバーのハードディスク上の `/var` ディレクトリに割り当てられるか、ファイルサーバーからマウントされてネットワーク上でアクセスされます。

注 - /var が別のスライスとして作成されていない場合、/var ディレクトリはルート (/) ファイルシステムの空間を使用します。これは、不十分な空間になる可能性があります。

ディスク空間

システムをプリンタサーバーの候補として評価するときは、利用できるディスク空間を考慮してください。大きなスプールディレクトリでは、600M バイトのディスク空間を消費することもあります。プリンタサーバーとして指定できるシステム上で、ディスク空間のサイズと分割設定を調べてください。

また、印刷のニーズと印刷クライアントシステムの利用パターンを慎重に調べてください。小さなグループのユーザーが短い電子メールメッセージ、つまり複雑なフォームを必要としない単純な ASCII ファイルだけを印刷する場合は、20～25M バイトのディスク空間を /var に割り当てたプリンタサーバーで十分でしょう。ただし、多数の印刷クライアントユーザーが長い文書、ビットマップ、またはラスタイメージファイルを印刷する場合は、スプーリング空間が頻繁にフルになります。ユーザーがジョブを印刷待ち行列に入れることができないと、作業の流れは中断してしまいます。より多くのスプーリング空間を要求すると、システム管理者はスプーリング用のディスク空間を追加するか、異なるシステムをグループのプリンタサーバーとして指定しなくてはならない場合があります。

プリンタサーバーが使用する/var ディレクトリが小さなパーティション上にあり、大きなディスク空間がディスク上の他の場所で利用可能な場合は、そのディスク空間をプリンタサーバーの /var ディレクトリにマウントすることにより、スプーリング空間として使用できます。ファイルシステムのマウントと vfstab ファイルの編集方法については、『*Solaris* のシステム管理 *(*第 *1* 巻*)*』の「ファイルシステムのマウントとマウント解除 (手順)」を参照してください。

メモリー

Solaris 環境で実行するには、少なくとも 64M バイトのメモリーが必要です。プリンタサーバー用にメモリーを追加する必要はありません。ただし、メモリーが多ければ、印刷要求のフィルタリング処理が高速になります。

スワップ空間

プリンタサーバー上のスワップ空間は、LP 印刷サービスを処理するのに十分な量を割り当ててください。スワップ空間を増やす方法については、『*Solaris* のシステム管理 *(*第 *1* 巻*)*』の「追加スワップ空間の構成 (手順)」を参照してください。

ハードディスク

最適の性能を発揮させるには、プリンタサーバーにハードディスクとローカル /var ディレクトリが必要です。プリンタサーバー用のスプーリング空間をローカルのハードディスク上にマウントしてください。プリンタサーバーに専用のハードディスクとローカルの /var ディレクトリがあれば、印刷処理ははるかに高速になり、印刷要求の処理に必要な時間をより正確に予測できます。

プリンタ設定の計画

この節では、Solaris 環境で印刷するための計画の立て方の概要を説明します。

- プリンタ名、プリンタの説明、プリンタポートなどのプリンタの定義の設定
- プリンタタイプとファイル内容の形式の選択
- 障害通知とデフォルトプリンタの宛先の設定
- バナーページを設定するかどうか、あるいはプリンタへのユーザーのアクセスを制限するかどうかの決定
- プリンタクラスと障害回復の設定

プリンタの定義の設定

ネットワーク上でのプリンタの定義は、より効率的な印刷環境をユーザーに提供するための継続的な作業です。この作業によって、たとえばユーザーがプリンタの位置を見つけやすいようにすべてのプリンタのパラメタを設定したり、プリンタのクラスを定義して印刷要求への応答時間を短縮したりできます。

lpadmin コマンドを使用すると、すべての印刷定義を設定できます。一方、Solaris プリンタマネージャを使用すると、プリンタのインストールまたは変更時に印刷定

義の一部だけを設定できます。表 3–1 は、印刷定義と、その定義を Solaris プリンタマネージャで割り当てることができるかどうかを示しています。

表 3–1 Solaris プリンタマネージャで設定される印刷定義

印刷定義	**Solaris** プリンタマネージャで設定できるか
プリンタ名	設定できる
プリンタの説明	設定できる
プリンタポート	設定できる
プリンタタイプ	設定できる
ファイル内容形式	設定できる。ただし `lpadmin` コマンドより機能は少ない
障害通知	設定できる。ただし `lpadmin` コマンドより機能は少ない
デフォルトプリンタ	設定できる
バナーページの印刷	設定できる。ただし `lpadmin` コマンドより機能は少ない
プリンタへのユーザーアクセスの制限	設定できる。ただし `lpadmin` コマンドより機能は少ない
プリンタクラス	設定できない
障害回復	設定できない

プリンタ名

システムにプリンタを追加するときは、その「プリンタ名」を指定します。プリンタ名は、次の規則に従ってください。

- 管理ドメイン内のすべてのプリンタ間で一意であること
- 最大 14 文字までの英数字 (ハイフンと下線も含む) であること

- 覚えやすく、プリンタのタイプ、場所、プリンタサーバー名などを識別できること

サイトに合った命名規則を設定してください。たとえば、ネットワーク上で異なるタイプのプリンタを使用する場合は、プリンタ名の一部にプリンタタイプを含めると、ユーザーは適切なプリンタを選択しやすくなります。たとえば、PostScript プリンタは文字 `PS` で識別できます。ただし、サイトのプリンタがすべて PostScript プリンタである場合は、`PS` というイニシャルをプリンタ名の一部として含める必要はありません。

プリンタの説明

`lpadmin -D` コマンドまたは Solaris プリンタマネージャを使用すると、プリンタに説明を割り当てることができます。プリンタに割り当てる説明には、ユーザーがプリンタを識別できるような情報を含める必要があります。プリンタが設置されている部屋番号、プリンタのタイプやメーカー、印刷に問題がある場合に連絡する担当者名などを含めることができます。

次のコマンドでプリンタの説明を調べることができます。

```
$ lpstat -D -p printer-name
```

プリンタポート

プリンタのインストール時、またはその設定を後から変更するときに、`lpadmin -p` *printer-name* `-v` *device-name* コマンドまたは Admintool を使用して、プリンタの接続先となるデバイス、つまり「プリンタポート」を指定できます。

ほとんどのシステムが、2 つのシリアルポートと 1 つのパラレルポートを持っています。システムにポートを追加しないかぎり、3 台以上のシリアルプリンタおよび 2 台以上のパラレルプリンタを直接接続することはできません。

Solaris プリンタマネージャを使用すると、`/dev/term/a` または `/dev/term/b` を選択するか、「その他 (Other)」を選択してプリンタサーバーで認識されるポート名を指定できます。これらのオプションには、`lpadmin` コマンドと同じ柔軟性があります。

LP 印刷サービスは、標準プリンタインタフェースプログラムからの設定を使用してプリンタポートを初期設定します。プリンタインタフェースプログラムの詳細は、141ページの「印刷フィルタの管理」を参照してください。デフォルト設定で機能しないパラレルプリンタやシリアルプリンタがある場合は、165ページの「プリンタポート特性の調整」のポート設定をカスタマイズする方法を参照してください。

注 - IA 搭載システムで複数のポートを使用している場合、デフォルトでは最初のポートだけが有効です。2 番目以降のポートはデフォルトでは無効です。複数のポートを使用するためには、追加の `asy` (シリアル) ポートや `lp` (パラレル) ポートごとに、デバイスドライバのポート構成ファイルを手作業で編集しなければなりません。IA ポート構成ファイルのパスは、次のとおりです。

```
/platform/i86pc/kernel/drv/asy.conf
```

```
/platform/i86pc/kernel/drv/lp.conf
```

IA 搭載システム上のシリアルポートとパラレルポートを構成する方法については、『*Solaris 8* デバイスの構成 *(Intel* 版*)*』を参照してください。

プリンタタイプ

プリンタタイプとは、プリンタの種類を表す一般名です。プリンタタイプは、プリンタの様々な制御シーケンスが入っている `terminfo` データベースエントリを識別します。通常、プリンタタイプはメーカーのモデル名からとります。たとえば、DECwriter のプリンタタイプ名は `decwriter` です。ただし、共通プリンタタイプ `PS` はこの規則に従いません。PS は Apple LaserWriter I や Apple LaserWriterII プリンタなど、多くの PostScript™ プリンタモデルのプリンタタイプとして使用されます。

`lpadmin -T` コマンドまたは Solaris プリンタマネージャを使用すると、プリンタタイプを指定できます。

Solaris プリンタマネージャを使用すると、メニューからプリンタタイプを選択するか、「その他 (Other)」を選択して `terminfo` データベース内でプリンタタイプを指定できます。この方法には、`lpadmin` コマンドと同じ機能があります。

`terminfo` データベース内のプリンタ名

各プリンタタイプに関する情報は、`terminfo` データベース (`/usr/share/lib/terminfo`) に格納されています。この情報には、プリンタの機能と初期設定制御データが含まれます。インストールするプリンタは、`terminfo` データベース内のエントリに対応していなければなりません。

```
$ pwd
/usr/share/lib/terminfo
$ ls
1  3  5  7  9  B  H  P  a  c  e  g  i  k  m  o  q  s  u  w  y
2  4  6  8  A  G  M  S  b  d  f  h  j  l  n  p  r  t  v  x  z
$
```

各サブディレクトリには、端末またはプリンタに関してコンパイル済みのデータベースエントリが入っています。各エントリは、プリンタまたは端末のタイプの頭文字別に編成されています。たとえば、Epson プリンタがある場合は、/usr/share/lib/terminfo/e 内を探すと、Epson プリンタの特定のモデルが見つかります。

```
$ cd /usr/share/lib/terminfo/e
$ ls
emots           ep2500+high   ep48            ergo4000      exidy2500
env230          ep2500+low    epson2500       esprit
envision230     ep40          epson2500-80    ethernet
ep2500+basic    ep4000        epson2500-hi    ex3000
ep2500+color    ep4080        epson2500-hi80  exidy
$
```

上記のように、Epson プリンタのエントリがあります。

NEC プリンタがある場合は、/usr/share/lib/terminfo/n ディレクトリ内を探すと、使用中の NEC プリンタモデルが見つかります。

```
$ cd /usr/share/lib/terminfo/n
$ ls
ncr7900         ncr7900iv      netronics      network        nuc
ncr7900-na      ncr7901        netty          netx           nucterm
ncr7900i        nec            netty-Tabs     newhp
ncr7900i-na     net            netty-vi       newhpkeyboard
$
```

上記のように、このディレクトリには、NEC のエントリが含まれています。

プリンタタイプの選択

ローカル PostScript プリンタの場合は、プリンタタイプとして PostScript (PS) または Reverse PostScript (PSR) を使用します。使用するプリンタが PostScript をサポートしていれば、プリンタタイプが terminfo データベースに含まれていても、PS または PSR を選択してください。

PostScript プリンタでページの印刷面を上にして印刷すると、文書は逆方向に印刷されます。1 ページ目はスタックの 1 番下になり、最終ページは 1 番上になります。プリンタのタイプを PSR として指定すると、LP 印刷サービスはプリンタに送る前にページの順序を逆転させます。つまり、最終ページが最初に印刷され、各ページは正順にスタックされます。ただし、LP 印刷サービスがページ順を確実に変更できるのは、『PostScript Language Reference Manual』(Adobe Systems Incorporated 制作、Addison-Wesley 社、1990 年刊) の付録 C の Adobe Document Structuring 規格に準拠する PostScript ファイルの場合だけです。

プリンタで複数の種類のプリンタをエミュレートできる場合は、lpadmin -T コマンドを使用して複数のタイプを割り当てることができます。複数のプリンタタイプを指定すると、LP 印刷サービスは各印刷要求に適したタイプを使用します。

該当する terminfo ディレクトリ内でプリンタタイプが見つからないことがあります。プリンタのタイプは、そのプリンタのメーカー名に対応しているとは限りません。たとえば、PostScript プリンタのタイプによっては、メーカーや製品名に固有のエントリの代わりに、PS または PSR エントリ (/usr/share/lib/terminfo/P ディレクトリに入っています) を使用できます。

例外的なタイプのプリンタを使用する場合は、さまざまなエントリを試してみなければ、プリンタのモデルに使用できる特定の terminfo エントリを判断できないことがあります。できれば、プリンタに使用できるエントリを terminfo データベース内で見つけてください。その方が、新しくエントリを作成するよりもはるかに簡単です。独自のエントリを作成しなければならない場合は、168ページの「サポートされていないプリンタの terminfo エントリを追加する」を参照してください。役立つヒントが掲載されています。

ファイル内容形式の選択

印刷フィルタはファイルの内容を、目的のプリンタが受け付けることができる形式に変換します。「ファイル内容形式」は、フィルタを通さずに直接印刷できるファイル内容の形式を LP 印刷サービスに通知します。フィルタなしに印刷するには、必要なフォントをプリンタ上でも利用できなければなりません (他のファイルタイプにはフィルタを設定して使用しなければなりません)。

lpadmin -I コマンドまたは Solaris プリンタマネージャを使用すると、プリンタのファイル内容形式を指定できます。Solaris プリンタマネージャを使用すると、メニューからファイル内容形式を選択できます。一部のファイル内容形式はメニューにありません。lpadmin コマンドを使用して、Solaris プリンタマネージャメニューにないファイル内容形式を指定してください。

多くのプリンタでは、次の 2 種類のファイルを直接印刷できます。

- プリンタタイプと同じタイプ (PostScript プリンタには `PS` など)
- `simple` タイプ (ASCII ファイル)

ユーザーがファイルの印刷要求を出すときは、そのファイルの内容形式を指定します (`lp -T` *content-type*)。要求を出すときにファイルの内容形式を指定しないと、LP サーバーは要求の最初のファイルを見て内容形式を判定します。ファイルが ^D%! または %! で始まっている場合、印刷要求は PostScript データと見なされます。それ以外の場合、ファイルは `simple` (ASCII) テキストと見なされます。LP 印刷サービスはファイル内容形式を使用して、ファイル内容をプリンタで処理できる形式に変換するためのフィルタを決めます。

Solaris プリンタマネージャではファイル内容形式のリストが表示されるので、ローカルプリンタをインストールまたは変更するときに、そこから形式を選択できます。選択結果は LP 印刷サービスが使用する名前に変換されます。表 3–2 は、Solaris プリンタマネージャで選択できるファイル内容形式を示しています。

表 3–2 Solaris プリンタマネージャによるファイル内容形式の選択

ファイル内容形式	**LP** 印刷サービス名	説明
PostScript	`postscript`	PostScript ファイルはフィルタを通す必要がない
ASCII	`simple`	ASCII ファイルはフィルタを通す必要がない
PostScript と ASCII	`simple,postscript`	PostScript ファイルも ASCII ファイルもフィルタを通す必要がない
None	`""`	プリンタのタイプに一致するもの以外は、すべてのファイルがフィルタを通す必要がある
Any	`any`	フィルタは使用されない。プリンタがファイル内容形式を直接処理できなければ、そのファイルは印刷されない

プリンタの機能に最も適合するファイル内容形式を選択してください。PostScript は、Solaris プリンタマネージャのデフォルトの選択で、通常はほとんどこのまま使用できます (PostScript ファイルには、フィルタ処理が不要なことを示します)。

通常使用するプリンタ

この節では、Solaris ソフトウェアで最も一般的に使用されるプリンタのプリンタタイプとファイル内容形式について説明します。掲載されていませんが、ここで説明するプリンタの多くは、`simple` 内容形式のファイルも直接印刷できます。

PostScript プリンタがある場合は、プリンタタイプ PS または PSR と内容形式 `postscript` を使用してください。PSR はページの順序を逆転させ、各ページを逆順で印刷してバナーページを最後に印刷します。

表 3–3 は、PostScript 以外の他のプリンタと、各プリンタの構成に使用するプリンタタイプを示しています。これらのプリンタでは、ファイル内容形式は `simple` です。

注 - Sun では表 3–3 のプリンタをサポートしていませんが、フィルタ処理を行うか、プリンタがファイル内容形式を直接印刷できれば、サポートしていないプリンタを使用できます。以下の製品に不明な点がある場合は、製造元に問い合わせてください。

表 3–3 Sun がフィルタを提供していない PostScript 以外のプリンタ

プリンタ	プリンタタイプ
Daisy	`daisy`
Datagraphix	`datagraphix`
DEC LA100	`la100`
DEC LN03	`ln03`
DECwriter	`decwriter`
Diablo	`diablo`
	`diablo-m8`
Epson 2500 系列	`epson2500`
	`epson2500-80`

表 3–3　Sun がフィルタを提供していない PostScript 以外のプリンタ　続く

プリンタ	プリンタタイプ
	epson2500-hi
	epson2500-hi80
Hewlett-Packard HPCL printer	hplaser
IBM Proprinter	ibmproprinter

terminfo データベースにないプリンタを設定したい場合は、171ページの「サポートされていないプリンタの terminfo エントリを追加する方法」を参照してください。

第 4 章

プリンタの設定手順

この章では、プリンタを設定してネットワーク上のシステムから Solaris プリンタマネージャを使用してアクセスできるようにする手順を説明します。Solaris プリンタマネージャは、Solaris Easy Access Server (SEAS) 3.0 リリースで使用可能になっていました。この章で説明する手順は次のとおりです。

- 70ページの「Solaris プリンタマネージャを起動する方法」
- 74ページの「新しく接続するプリンタを Solaris プリンタマネージャを使用して追加する方法」
- 77ページの「Solaris プリンタマネージャにプリンタアクセスを追加する方法」
- 78ページの「`.printers` ファイルを設定する方法」
- 84ページの「プリンタベンダー提供のツールを使用してネットワークプリンタを追加する方法」
- 85ページの「LP コマンドを使用してネットワークプリンタを追加する方法」

プリンタの概要については、第 2 章を参照してください。

印刷の設定

68ページの「印刷の設定の作業マップ」は、プリンタサーバーの設定 (プリンタの追加) と印刷クライアントの設定 (プリンタへのアクセスの追加) に必要な作業の概要を示しています。ローカルプリンタとは、プリンタサーバーに物理的にケーブル接続されたプリンタのことです。ネットワークプリンタとは、ネットワークに物理的に接続されたプリンタのことです。プリンタへのアクセスを追加する (つまり、リ

モートアクセスを追加する) とは、印刷クライアント (サーバー以外のすべてのマシン) がプリンタにアクセスできるようにする手順のことです。

印刷の設定の作業マップ

表 4–1 作業マップ: 印刷の設定

作業	説明	手順の説明
1. 新しく接続したプリンタの追加	Solaris プリンタマネージャを使用する プリンタをシステムに物理的に接続したあとで、プリンタを印刷に使用できるようにする	74ページの「新しく接続するプリンタを Solaris プリンタマネージャを使用して追加する方法」
2. プリンタへのアクセスの追加	Solaris プリンタマネージャを使用する プリンタへのアクセスの追加	77ページの「Solaris プリンタマネージャにプリンタアクセスを追加する方法」
3. `.printers` ファイルの設定	**(**省略可能**)** `$HOME/.printers` ファイルを使用して、ユーザーが独自のカスタムプリンタ別名を設定できるようにする	78ページの「`.printers` ファイルを設定する方法」
4. ネットワークプリンタの追加	プリンタベンダー提供のツールを使用する プリンタをネットワークに物理的に接続した後、ベンダー提供のソフトウェアを使用してネットワークプリンタを構成する LP コマンドを使用する プリンタをネットワークに物理的に接続した後、Solaris ソフトウェアコマンドを使用してネットワークプリンタを構成する	84ページの「プリンタベンダー提供のツールを使用してネットワークプリンタを追加する方法」 85ページの「LP コマンドを使用してネットワークプリンタを追加する方法」
5. バナーページのオフ	**(**省略可能**)** 印刷されないように、バナーページをオフにできる	107ページの「バナーページをオフにする方法」
6. 障害警告の設定	**(**省略可能**)** プリンタ用に Solaris プリンタマネージャよりも限定的な障害警告を設定できる	110ページの「プリンタの障害警告を設定する方法」

表 4–1 作業マップ: 印刷の設定 続く

作業	説明	手順の説明
7. 障害回復機能の設定	(省略可能) Solaris プリンタマネージャでは、障害が起きた後でプリンタを回復させる方法を設定できない	113ページの「プリンタの障害回復を設定する方法」
8. プリンタへのアクセス制限	(省略可能) Solaris プリンタマネージャで、許可リストを設定できる。ただし、プリンタへアクセスできるユーザーを制限したい場合には、拒否リストを設定する	115ページの「プリンタへのユーザーアクセスを制限する方法」

Solaris プリンタマネージャによるプリンタの設定

次の表はプリンタ属性を示しています。 Solaris プリンタマネージャを使用してプリンタを設定するときに、必要となる情報を判断するのに役立ててください。

プリンタ属性	説明	例	デフォルトの設定	必須か省略可能か
プリンタ名	プリンタの名前	`laser1`	なし	接続されたプリンタやネットワークプリンタをインストールし、プリンタへのアクセスを追加するのに必須
プリンタサーバー	プリンタサーバーの名前	`venus`	ローカルシステム	接続されたプリンタやネットワークプリンタをインストールし、プリンタへのアクセスを追加するのに必須
名称	ユーザー定義の文字列	`laser printer near breakroom`	なし	省略可能
プリンタポート	デバイスプリンタが接続されているポート	`/dev/term/a`	`/dev/term/a`	接続されたプリンタをインストールするのに必須
プリンタタイプ	プリンタのタイプ	`unknown`	`PostScript`	接続されたプリンタやネットワークプリンタをインストールするのに必須

プリンタ属性	説明	例	デフォルトの設定	必須か省略可能か
ファイル内容	印刷する内容	any	PostScript	接続されたプリンタやネットワークプリンタをインストールするのに必須
宛先	ネットワークプリンタの宛先名	例については、82 ページの「宛先 (またはネットワークプリンタアクセス) 名の選択」を参照	なし	ネットワークプリンタをインストールするのに必須
プロトコル	プリンタとの通信に使用するプロトコル	TCP	BSD	ネットワークプリンタをインストールするのに必須
失敗の通知	ユーザーにエラーを知らせる方法	Mail to superuser	Write to superuser	省略可能
デフォルトプリンタ	デフォルトプリンタを識別する	なし	なし	省略可能
バナーを常に印刷	バナーが印刷される	なし	バナーが印刷される	省略可能
ユーザーアクセスリスト	印刷できるユーザーのリスト	rimmer,lister	すべてのユーザーが印刷できる	省略可能

Solaris プリンタマネージャの起動

Solaris プリンタマネージャを使用してプリンタを設定するには、CDE ワークスペースメニューから「プリンタ管理 (Printer Administration)」を選択して Solaris プリンタマネージャを起動するか、コマンド行から Solaris プリンタマネージャを起動します。詳細については、次の節を参照してください。

▼ Solaris プリンタマネージャを起動する方法

1. **次の前提条件を満たしていることを確認します。Solaris プリンタマネージャを使用するには、次の条件を満たす必要があります。**

- ビットマップディスプレイモニター。Solaris プリンタマネージャは、Sun ワークステーションの標準ディスプレイモニターなど、ビットマップ画面のコンソールを使用するシステムでだけ使用できます。
- CDE 環境のような X Window System を実行しているか、xhost 環境が動作するシステムでリモート表示機能を使用している。
- 接続されたプリンタやネットワークプリンタをインストールする場合は、プリンタサーバーにスーパーユーザーとしてログインしている。プリンタへのアクセスを追加する場合は、印刷クライアントにスーパーユーザーとしてログインしている。
- NIS、NIS+、または NIS+ (xfn) データベースを管理するのに必要なアクセス特権を持っている。
  - ネームサービスが NIS の場合は、NIS マスターの root パスワードが必要です。
  - ネームサービスが NIS+ の場合は、次の手順を実行しなければならない場合があります。

    1. NIS+ マスターにスーパーユーザーとしてログインします。
    2. 次に示すようにプリンタテーブルを所有するグループを確認します。

```
# niscat -o printers.org_dir.domain_name.com
            .
            .
            .
        Group   : "admin.domain_name.com"
```

    3. 必要なら、printers.org_dir.<*domain*> ファイルの更新を許可された NIS+ admin グループに、Solaris プリンタマネージャを実行するシステムを追加します。

```
# nisgrpadm -a admin.domain_name.com host_name
```

    4. Solaris プリンタマネージャを実行するシステムにスーパーユーザーとしてログインします。NIS+ 構成によっては、/usr/bin/keylogin コマンドも実行しなければならない場合があります。詳細は、keylogin(1) のマニュアルページを参照してください。

  - ネームサービスが NIS+ (xfn) の場合は、次の手順を実行しなければならない場合があります。

1. NIS+ マスターにスーパーユーザーとしてログインします。
2. フェデレーテッド・ネーミングテーブルを所有するグループを確認します。

```
# niscat -o fns.ctx_dir.domain_name.com
            .
            .
            .
         Group   : "admin.domain_name.com"
```

3. 必要なら、fns.ctx_dir.<*domain*> ファイルの更新を許可された NIS+ admin グループに、Solaris プリンタマネージャを実行するシステムを追加します。

```
# nisgrpadm -a admin.domain_name.com host_name
```

4. Solaris プリンタマネージャを実行するシステムにスーパーユーザーとしてログインします。NIS+ 構成によっては、/usr/bin/keylogin コマンドも実行しなければならない場合があります。詳細は、keylogin(1) のマニュアルページを参照してください。

- SUNWppm パッケージをインストールします。

```
# pkginfo | grep SUNWppm
system      SUNWppm        Solaris Print Manager
```

2. **CDE** のワークスペースメニューの「ツール **(Tools)**」オプションから「プリンタ管理 **(Printer Administration)**」を選択して、**Solaris** プリンタマネージャを起動します。あるいは、**CDE** フロントパネルから「アプリケーション **(Applications)**」メニューを選択し、**Application Manager** の「**System_Admin**」ウィンドウで「プリンタ管理 **(Printer Administration)**」アイコンをクリックします。次のコマンドを使用しても **Solaris** プリンタマネージャを起動できます。

```
# /usr/sadm/admin/bin/printmgr &
```

Solaris プリンタマネージャのメインウィンドウ上に「ネームサービスを選択 (Select Naming Service)」ウィンドウが重なって表示されます。

リモートシステムから Solaris プリンタマネージャを使用する場合は、DISPLAY 環境変数を設定してから Solaris プリンタマネージャを起動します。

```
# DISPLAY=hostname:display_number
# export DISPLAY
# /usr/sadm/admin/bin/printmgr &
```

注 - CDE メニューやコマンド行から Solaris プリンタマネージャを起動できない場合は、次の確認をしてください。

1. ローカルシステムやリモートシステムの Xserver プロセスに接続する権限がスーパーユーザー (root) にない可能性があります。その場合は、次のコマンドを入力します。

```
$ xhost +hostname
$ su
(Enter root's password)
# /usr/sadm/admin/bin/printmgr &
```

Solaris プリンタマネージャを再起動する前に、ローカルシステムまたはリモートシステムの名前で *hostname* を置き換えます。

2. ローカルシステムまたはリモートシステムに SUNWppm パッケージがインストールされていることを確認します。

```
$ pkginfo | grep SUNWppm
```

---

3. ネットワークで使用されているネームサービスを「ネームサービスを選択 **(Select Naming Service)**」ウィンドウから選択します。選択肢には **NIS+ (xfn)**、**NIS+**、**NIS**、**files** があります。

4. ドメイン名が正しいことを確認します。

   ネームサービスが正常に読み込まれると、Solaris プリンタマネージャのメインウィンドウが表示されます。

プリンタサーバーの設定

接続されたプリンタとネットワークプリンタ、またはそのどちらかをシステムに追加すると、そのプリンタにローカルシステムからアクセスできるようになります。プリンタをインストールするシステムが「プリンタサーバー」になります。

次の各項では、新しい Solaris プリンタマネージャを使用して、接続されているプリンタやネットワークプリンタをプリンタサーバーに追加する方法を説明します。Solaris プリンタマネージャを使用する手順に続く例は、LP コマンドを使用してプリンタを追加する方法です。

▼ 新しく接続するプリンタを Solaris プリンタマネージャを使用して追加する方法

1. プリンタサーバーにするシステムを選択します。

2. プリンタをプリンタサーバーに接続し、プリンタの電源を入れます。

   ハードウェアのスイッチとケーブルの要件については、プリンタのインストールマニュアルを参照してください。

3. プリンタを接続したプリンタサーバー上で **Solaris** プリンタマネージャを起動します。

   詳細は 70ページの「Solaris プリンタマネージャを起動する方法」の手順を参照してください。

4. 「プリンタ **(Printer)**」メニューから「新規に接続したプリンタ **(New Attached Printer)**」を選択します。

   「新規に接続したプリンタ (New Attached Printer)」ウィンドウが表示されます。

5. ウィンドウに情報を入力します。

   フィールドに情報を入力する必要がある場合は、「ヘルプ (Help)」ボタンをクリックします。

6. 「了解 **(OK)**」をクリックします。

7. プリンタがインストールされていることを確認します。確認するには、**Solaris** プリンタマネージャのメインウィンドウに新しいプリンタエントリがあるか調べ

ます。インストールされていたら、次のコマンドを使用してプリンタに要求を印刷できるか確認します。

```
$ lp -d printer-name filename
```

8. **Solaris** プリンタマネージャを終了します。

   「プリンタマネージャメニュー (Print Manager Menu)」から「終了 (Exit)」を選択します。

例 — LP コマンドを使用して新規に接続したプリンタを追加する

この例では、ローカルの PostScript プリンタをプリンタサーバーで印刷できるようにする方法を示しています。この例のコマンドは、プリンタが接続されているプリンタサーバーで実行しなければなりません。この例では次の情報を使用しています。実際に指定する情報はこれとは異なります。

- プリンタ名: `luna`
- ポートデバイス: `/dev/term/b`
- プリンタタイプ: `PS`
- ファイル内容形式: `postscript`

```
# chown lp /dev/term/b
# chmod 600 /dev/term/b 1
# lpadmin -p luna -v /dev/term/b 2
# lpadmin -p luna -T PS 3
# lpadmin -p luna -I postscript 4
# cd /etc/lp/fd
# for filter in *.fd;do
    > name=`basename $filter .fd`
    > lpfilter -f $name -F $filter
    > done 5
# accept luna
 destination ``luna'' now accepting requests
# enable luna 6
printer ``luna'' now enabled
# lpadmin -p luna -D "Room 1954 ps" 7
# lpstat -p luna 8
 printer luna is idle. enabled since Jul 12 11:17 1999. available.
```

1. `lp` に所有権とポートデバイスへの単独アクセスを設定する。
2. プリンタ名とプリンタが使用するポートデバイスを定義する。

3. プリンタのプリンタタイプを設定する。
4. プリンタが直接印刷できるファイル内容形式を指定する。
5. 印刷フィルタをプリンタサーバーに追加する。
6. プリンタが印刷要求を受け入れて、印刷できるようにする。
7. プリンタの説明を追加する。
8. プリンタが用意できていることを確認する。

次に進む手順

次の表を見て、次に進む手順を決めてください。

必要な処理	参照
印刷クライアントに新しくインストールしたプリンタへのアクセスを追加する (プリンタ情報をネームサービスデータベースに追加していない場合)	77ページの「Solaris プリンタマネージャにプリンタアクセスを追加する方法」を参照
`.printers` ファイルを設定する	78ページの「`.printers` ファイルを設定する方法」を参照

印刷クライアントの設定

印刷クライアントは、プリンタ用のサーバーではないが、プリンタにアクセスできるシステムのことです。印刷クライアントは、プリンタサーバーのサービスを使用して、印刷ジョブのスプール、スケジュール、およびフィルタリングを実行します。1 つのシステムがあるプリンタのプリンタサーバーになり、同時に他のプリンタの印刷クライアントになることも可能です。

プリンタへのアクセスは、ドメイン全体で有効にすることも、マシン単位で有効にすることもできます。これは、プリンタ情報をネームサービスデータベースに追加するかどうかによって異なります。

次の項では、新しい Solaris プリンタマネージャを使用して、印刷クライアントでプリンタへのアクセスを追加する方法を説明します。手順に続く例は、LP コマンドを使用してプリンタアクセスを追加する方法です。

▼ Solaris プリンタマネージャにプリンタアクセスを追加する方法

1. リモートプリンタへのアクセスを追加したいシステム上で **Solaris** プリンタマネージャを起動します。

   詳細は、70ページの「Solaris プリンタマネージャを起動する方法」の手順を参照してください。

2. 「プリンタ **(Printer)**」メニューから「追加 **(Add)**」、「プリンタへのアクセス **(Access to Printer)**」の順に選択します。

   「プリンタへのアクセス (Add Access to Printer)」ウィンドウが表示されます。

3. ウィンドウに情報を入力します。

   フィールドに情報を入力する必要がある場合は、「ヘルプ (Help)」ボタンをクリックして、このウィンドウのフィールドの定義を表示します。

4. 「了解 **(OK)**」をクリックします。

   プリンタが Solaris プリンタマネージャのメインウィンドウに表示されます。

5. プリンタへのアクセスが追加されていることを確認します。確認するには、**Solaris** プリンタマネージャのメインウィンドウに新しいプリンタエントリがあるか調べます。追加されていたら、次のコマンドを使用してプリンタに要求を印刷できることを確認します。

   ```
   $ lp -d printer-name filename
   ```

6. **Solaris** プリンタマネージャを終了します。

   「プリンタマネージャメニュー (Print Manager Menu)」から「終了 (Exit)」を選択します。

例 — LP コマンドでプリンタアクセスを追加する

リモートプリンタで印刷する場合、リモートプリンタにアクセスを追加しなければなりません。次の例は、プリンタサーバー `saturn` のプリンタ `luna` にアクセスを構成する方法を示しています。システム `saturn` は、プリンタ `luna` の印刷クライアントになります。

```
# lpadmin -p luna -s saturn 1
# lpadmin -p luna -D "Room 1954 ps" 2
# lpadmin -d luna 3
# lpstat -p luna 4
  printer luna is idle. enabled since Jul 12 11:17 1999. available.
```

1. プリンタとプリンタサーバーを指定する。
2. プリンタの説明を追加する。
3. プリンタをシステムのデフォルトのプリンタ宛先として設定する。
4. プリンタが用意できていることを確認する。

.printers ファイルの設定

プリンタ情報をカスタマイズする必要がなければ、ユーザーのホームディレクトリに .printers ファイルを設定する必要はありません。しかし、.printers ファイルは、ユーザーが独自のカスタムプリンタ別名を設定できる場所です。別名 _default を使用すれば、デフォルトのプリンタを設定できます。また、特殊別名 _all を設定すれば、印刷要求を取り消したりプリンタの情報をチェックしたりするときの対象となるプリンタのリストを定義できます。

LP 印刷サービスが .printers ファイルを使用するかどうかは、ネームサービススイッチ (/etc/nsswitch.conf) によって制御されます。デフォルト構成では、印刷サービスは、ユーザーのホームディレクトリでプリンタ構成情報を探してから他のネームサービスを調べます。つまり、ユーザーのプリンタ構成ファイルを設定することによって、ネームサービスの共有情報ではなく、好みのプリンタ情報を使用できます。

.printers ファイルの詳細は、printers(4) のマニュアルページを参照してください。ネームサービススイッチの詳細は、nsswitch.conf(4) のマニュアルページを参照してください。

▼ .printers ファイルを設定する方法

1. スーパーユーザーとしてシステムにログインします。
2. 任意のエディタで、.printers ファイルをユーザーのホームディレクトリに作成します。

3. **(省略可能)** `_default` 別名を設定して、特定のプリンタをデフォルトプリンタに指定します。次の例に示すようなエントリを使用します。

```
_default printer_name
```

4. **(省略可能)** `_all` 別名を設定して、印刷要求を取り消したりプリンタの状態をチェックしたりするときの対象となるプリンタを定義します。次の例に示すようなエントリを使用します。

```
_all printer1,printer2,printer3
```

5. そのファイルを `.printers` として保存します。

ネットワークプリンタの追加

「ネットワークプリンタ」とは、ネットワークに直接接続されているハードウェアデバイスです。これは、ネットワークプリンタがプリンタサーバーにケーブルで実際に接続されていなくても、プリンタサーバーからアクセスできることを意味します。ネットワークプリンタは専用のシステム名と IP アドレスを持っています。ネットワークプリンタがプリンタサーバーに接続されていない場合でも、専用のプリンタサーバーを設定しておく必要があります。プリンタサーバーは、ネットワークプリンタの待ち行列化機能と印刷管理機能を提供します。

ネットワークプリンタは、ベンダー提供の印刷プログラムを必要とする特別なプロトコルを 1 つ以上使用することがあります。ベンダーから提供される印刷プログラムの設定手順は、それぞれ異なることがあります。プリンタにベンダー提供サポートが付いていない場合、ほとんどのデバイスについて Solaris のネットワークプリンタサポートを使用できます。ただし、可能な限りプリンタベンダー提供のソフトウェアを利用することを強く推奨します。

ベンダーは、SVR4 プリンタインタフェーススクリプトを提供して標準プリンタインタフェーススクリプトを置き換えている場合があります。その場合、SVR4 インタフェーススクリプトはベンダー提供の印刷プログラムを呼び出して、ジョブをプリンタに送ります。このスクリプトが提供されない場合は、標準インタフェーススクリプトを変更してベンダー提供の印刷プログラムを呼び出す必要があります。こ

の作業は、標準インタフェーススクリプトのプリンタごとのコピーをベンダー提供の印刷プログラムを呼び出すように編集することで実行できます。

ネットワークプリンタ構成で使用する用語を説明します。

- プリンタサーバー: プリンタ用にジョブをスプールおよびスケジュールするマシン。このマシンにプリンタが構成されます。
- プリンタホストデバイス: プリンタホストデバイスは、ネットワークに対応していないプリンタにネットワークプリンタサポートを提供する、ベンダー提供のソフトウェアおよびハードウェアです。プリンタホストデバイスとそれに接続された 1 つまたは複数のプリンタの組み合わせを「ネットワークプリンタ」と呼びます。
- プリンタノード: 物理的なプリンタまたはプリンタホストデバイス。ネットワークサポートが物理的なプリンタにあるときは、物理的なプリンタです。ネットワークインタフェースを提供するために外部ボックスを使用しているときは、プリンタホストデバイスです。プリンタノード名は、IP アドレスが与えられているマシン名です。この名前はシステム管理者が選択するもので、デフォルトやベンダーの要件はありません。ノードと同様に、プリンタノード名も一意でなければなりません。
- プリンタ名: プリンタコマンドを使用するときにコマンド行に入力する名前。この名前は、システム管理者がプリンタ構成時に選択します。1 つの物理的なプリンタは、複数のプリンタ名または待ち行列名を持つことができます。それぞれ、そのプリンタへのアクセスを提供します。
- 宛先またはネットワークプリンタアクセス名: プリンタサブシステムがプリンタにアクセスするときに使用するプリンタノードポートの内部名。プリンタノード名か、プリンタベンダーポート指定付きのプリンタノード名です。プリンタベンダーポート指定は、プリンタベンダーのマニュアルで明示的に定義されています。これはプリンタに固有です。プリンタがプリンタホストデバイスでありプリンタでもある場合、ポート指定は、プリンタホストデバイスのマニュアルに説明されています。書式は次のいずれかです。

  printer_node_name

  または

  printer_node_name:*port_designation*

- プロトコル: プリンタとケーブル経由で通信するために使用するプロトコル。プリンタのマニュアルには、選択するプロトコルについての説明があります。ネットワークプリンタサポートは、BSD プリンタプロトコルと raw TCP の両方を提供します。実装によって、両方を使用するように設定できます。

- タイムアウト (再試行間隔): タイムアウトは、プリンタへの接続の試行間に待機する秒数を表すシード (seed) 数です。このシード数は、接続の試行間に待機する最小の秒数であり、接続が失敗するごとに増えます。プリンタへの接続が繰り返して失敗すると、ユーザーの介入を要求するメッセージがユーザーに戻されます。接続が成功するか、ジョブの所有者がジョブを取り消すまで、再接続の試行は続けられます。

ネットワークプリンタ用のプリンタベンダー提供のソフトウェア

多くの場合、ネットワークプリンタには、プリンタベンダー提供のソフトウェアサポートが提供されます。プリンタにプリンタベンダー提供のソフトウェアがある場合は、プリンタベンダー提供のソフトウェアを利用することを強く推奨します。プリンタベンダー提供のソフトウェアは、そのプリンタの特性をサポートするように設計されていて、プリンタの能力をフルに活用します。プリンタのマニュアルをよく読んで、プリンタを LP 印刷システムにインストールおよび構成してください。

Sun のネットワークプリンタのサポート

ネットワークプリンタベンダーがソフトウェアサポートを提供していない場合、Sun が提供するソフトウェアを利用できます。このソフトウェアは、ネットワークプリンタの汎用サポートを提供するもののため、必ずしもプリンタで利用できるすべての機能を使用できません。

ネットワークプリンタを追加するための一般的な説明については、第 4 章を参照してください。次は、Sun 提供のソフトウェアを使用したプリンタの管理を説明します。

ネットワークプリンタサポートの呼び出し

ネットワークプリンタ用のソフトウェアサポートは、インタフェーススクリプト経由で呼び出されます。ネットワークインタフェーススクリプト `netstandard` でネットワークプリンタを構成すると、ネットワークプリンタサポートモジュールが呼び出されます。次に、ネットワークサポートでプリンタを構成するコマンドを示します。

```
lpadmin -p printer_name -m netstandard
```

プロトコルの選択

印刷サブシステムは、BSD 印刷プロトコルと raw TCP を使用してプリンタと通信します。プリンタのマニュアルには、使用するプロトコルについての情報が提供されています。一般に、プリンタに使用するのは TCP プロトコルです。

プロトコルを選択するコマンドは次のいずれかです。

```
lpadmin -p printer_name -o protocol=bsd
```

または

```
lpadmin -p printer_name -o protocol=tcp
```

選択したプロトコルが BSD 印刷プロトコルの場合、さらにコントロールファイルをプリンタに送信する順番を選択できます。一部のプリンタは、コントロールファイルの後にデータファイルという順番を仮定しますが、その逆を仮定するプリンタもあります。この情報については、プリンタベンダーのマニュアルを参照してください。デフォルトでは、コントロールファイルを先に送信します。

順番を選択するコマンドは次のいずれかです。

```
lpadmin -p printer_name -o bsdctrl=first
```

または

```
lpadmin -p printer_name -o bsdctrl=last
```

プリンタノード名の選択

システム管理者はプリンタノード名を選択します。ネットワーク上のノードと同様に、この名前は一意でなければなりません。プリンタノード名は、プリンタの IP アドレスと関連付けられます。

宛先 (またはネットワークプリンタアクセス) 名の選択

印刷サブシステムはプリンタのアクセス情報を必要とします。これは、プリンタへのネットワーク接続を行うときにサブシステムが使用する名前です。この名前は、システム管理者が `lpadmin` コマンドで印刷サブシステムに提供します。これは、プリンタ構成データベースの一部になります。プリンタアクセス名はプリンタノード名であり、ポート名で修飾される場合もあります。ポート指定はプリンタベンダー間で異なります。ポート指定については、プリンタのマニュアルを参照してください。次に、プリンタアクセス名の書式を示します。

printer_node-name [: *port_designation*]

例 1 — ポート指定 (番号) 付き宛先名 (またはネットワークプリンタアクセス名)

TCP の共通ポート指定は 9100 です。プリンタノード名が pn1 で、プリンタベンダーがそのポートを 9100 と定義していた場合、プリンタアクセス名は `pn1:9100` になります。この場合にプリンタを構成するには、次のコマンドを使用します。

```
lpadmin -p printer_name -o dest=pn1:9100
```

例 2 — ポート指定 (名前) 付き宛先名 (またはネットワークプリンタアクセス名)

BSD プロトコルを使用するとき、ポート指定は番号でなく、プリンタベンダーが定義した名前です (例: `xxx_parallel_1`)。プリンタノード名が cardboard の場合、プリンタアクセス名は `cardboard:xxx_parallel_1` になります。この場合プリンタを構成するには、次のコマンドを使用します。

```
lpadmin -p printer_name -o dest=cardboard:xxx_parallel_1
```

例 3 — ポート指定なしの宛先名 (またはネットワークプリンタアクセス名)

ポート指定がなく、プリンタノード名が newspaper の場合、プリンタアクセス名はプリンタノード名 `newspaper` になります。この場合にプリンタを構成するには、次のコマンドを使用します。

```
lpadmin -p printer_name -o dest=newspaper
```

タイムアウト値の設定

タイムアウトオプションは、プリンタに接続しようとする試行間で待機する時間 (秒数) を個別に選択するためのものです。ウォームアップ時間が長いプリンタの場合は、タイムアウト値を大きくします。デフォルトは 10 秒です。

タイムアウト値は、印刷プロセスが成功するか失敗するかには影響を与えません。これは、ソフトウェアが初期タイムアウトカウントとして使用するシード値です。失敗が続くと、このカウントは増えます。プリンタへの接続の試行が連続して失敗すると、メッセージがスプーラに送信されます。これによって、ユーザーの介入が必要であることをユーザーに警告します。プリンタの電源が入っていなかったり、用紙がなくなっていたりするときにも、このメッセージが生成される可能性があります。たとえば、プリンタがウォームアップしているときに、このようなメッセー

ジが頻繁に生成されるようであれば、タイムアウト値を増やすことで間違ったメッセージを減らすことができます。

システム管理者は最適なタイムアウト値を探してください。次に、タイムアウト値を設定するコマンドを示します。

```
lpadmin -p printer_name -o timeout=n
```

ネットワークプリンタアクセスの管理

各ネットワークプリンタは、そのプリンタへのアクセスを提供するサーバーを 1 つだけ持っています。これによって、サーバーはそのプリンタへのアクセスを管理して、ジョブの一貫性を保つことができます。

ネットワークプリンタのデフォルトデバイスは /dev/null です。プリンタに待ち行列が 1 つしかない場合はこれで十分です。複数の待ち行列が必要であれば、そのデバイスをファイルに設定します。これによって、印刷システムはプリンタへのアクセスを待ち行列間で制限できます。次のコマンドは、デバイスファイルを作成して、ネットワークプリンタデバイスとして構成しています。

```
touch /path/filename
chmod 600 /path/filename
lpadmin -p printer_name -v /path/filename
```

次の例では、devtreedown というデバイスファイルを作成しています。

```
# touch /var/tmp/devtreedown
# chmod 600 /var/tmp/devtreedown
# lpadmin -p treedown -v /var/tmp/devtreedown
```

▼ プリンタベンダー提供のツールを使用してネットワークプリンタを追加する方法

1. プリンタをネットワークに接続し、電源を入れます。

   ハードウェアのスイッチとケーブルの要件については、プリンタのインストールマニュアルを参照してください。IP アドレスを取得して、プリンタノード名を選択します。これは、ネットワークにノードを追加することと同じです。

2. プリンタのマニュアルに従って、**SVR4 LP** 印刷スプーラのある **SunOS 5.8** システムにネットワークプリンタを追加します。

   プリンタのマニュアルを使用して、ネットワークプリンタを構成してください。手順は、ベンダーやプリンタに固有です。

3. 新しいプリンタへのアクセスをクライアントに追加します。

   これでプリンタは追加されました。プリンタへのアクセスをクライアントに作成します。詳細は、76ページの「印刷クライアントの設定」を参照してください。

4. オプションの作業を完了します。

   ネットワークプリンタを設定するときは、オプションの作業がいくつかあります。残りの作業の参照先については、68ページの「印刷の設定の作業マップ」を参照してください。

▼ LP コマンドを使用してネットワークプリンタを追加する方法

注 - ここでは、ネットワークプリンタサポートソフトウェアを使用して、ネットワークプリンタを設定するのに必要な手順を説明しています。このソフトウェアを使用するのは、プリンタにベンダー提供のソフトウェアが付いていない場合だけです。

---

1. プリンタをネットワークに接続して、プリンタの電源を入れます。

   ハードウェアのスイッチとケーブル接続の要件については、プリンタのインストールマニュアルを参照してください。IP アドレスを取得して、プリンタノード名を選択します。これは、ネットワークにノードを追加することと同じです。

2. ネットワークプリンタを構成するのに必要な情報を収集します。
   - プリンタ名
   - プリンタサーバー
   - ネットワークプリンタアクセス名
   - プロトコル
   - タイムアウト

   詳細は、79ページの「ネットワークプリンタの追加」に説明されている用語を参照してください。

3. lpadmin(1M) コマンドを使用して、プリンタ名、デバイス、プリンタタイプ、および内容形式を定義します。

   a. プリンタ名とプリンタが使用するポートデバイスを定義します。

```
# lpadmin -p printer-name -v /dev/null
```

   使用するデバイスは /dev/null です。

   b. プリンタが使用するインタフェースを指定します。

```
# lpadmin -p printer-name -m /netstandard
```

   ネットワークプリンタサポートソフトウェアで提供されるインタフェーススクリプトは /usr/lib/lp/model/netstandard です。

   c. プリンタ宛先、プロトコル、およびタイムアウト値を設定します。

```
# lpadmin -p printer-name -o dest=access-name:port -o protocol=protocol -o timeout=value
```

-p *printer-name*	ネットワークプリンタ名を指定する
-o dest=*access-name:port*	ネットワークプリンタアクセス名と、プリンタのマニュアルに定義されていれば指定されたプリンタベンダーポートに、プリンタ宛先を設定する。詳細は、79ページの「ネットワークプリンタの追加」を参照
-o protocol:*protocol*	プリンタとケーブル経由で通信するために使用するプロトコルを設定する。BSD と raw TCP の両方をサポートしている
-o timeout:*value*	プリンタへの接続の試行間で待機する秒数を表す再試行タイムアウト値を設定する。詳細は、79ページの「ネットワークプリンタの追加」を参照

   d. プリンタのファイル内容形式とプリンタタイプを指定します。

```
# lpadmin -p printer-name -I content-type -T printer-type
```

4. lpfilter(1M) コマンドを使用して、プリンタサーバーにフィルタを追加します。

```
# cd /etc/lp/fd
# for filter in *.fd;do
    > name=`basename $filter .fd`
    > lpfilter -f $name -F $filter
    > done
```

5. プリンタがプリンタ要求を受け入れて、その要求を印刷できるようにします。

```
# accept printer-name
# enable printer-name
```

6. lpstat(1M) コマンドを使用して、プリンタが正しく構成されていることを確認します。

```
# lpstat -p printer-name
```

7. 新しいプリンタへのアクセスをクライアントに追加します。

   これでプリンタは追加されました。プリンタへのアクセスをクライアントに作成します。詳細は、76ページの「印刷クライアントの設定」を参照してください。

8. オプションの作業を完了します。

   プリンタを設定するときは、オプションの作業がいくつかあります。残りの作業の参照先については、68ページの「印刷の設定の作業マップ」を参照してください。

この例のコマンドは、プリンタサーバーで実行しなければなりません。この例では次の情報を使用していますが、これらの情報は状況によって異なります。

- プリンタ名: luna1
- サーバー: saturn

- ネットワークプリンタアクセス名: nimquat:9100
- プロトコル: tcp
- タイムアウト: 5
- インタフェース: /usr/lib/lp/model/netstandard
- プリンタタイプ: PS
- 内容形式: postscript
- デバイス: /dev/null

```
# lpadmin -p luna1 -v /dev/null [1]
# lpadmin -p luna1 -m netstandard [2]
# lpadmin -p luna1 -o dest=nimquat:9100 -o protocol=tcp -o timeout=5 [3]
# lpadmin -p luna1 -I postscript -T PS [4]
# cd /etc/lp/fd
# for filter in *.fd;do
   > name=`basename $filter .fd`
   > lpfilter -f $name -F $filter
   > done [5]
# accept luna1
 destination "luna1" now accepting requests
# enable luna1 [6]
printer "luna1" now enabled
# lpadmin -p luna1 -D "Room 1954 ps" [7]
# lpstat -p luna1 [8]
  printer luna1 is idle. enabled since Jul 12 11:17 1999.  available.
```

1. プリンタ名を定義する。デバイスを /dev/null に設定する。
2. ネットワークプリンタ用のインタフェーススクリプトを定義する。
3. 宛先、プロトコル、およびタイムアウトを設定する。
4. プリンタが直接印刷できるファイル内容形式とプリンタタイプを指定する。
5. 印刷フィルタをプリンタサーバーに追加する。
6. プリンタが印刷要求を受け入れて、印刷できるようにする。
7. プリンタの説明を追加する。
8. プリンタが用意できていることを確認する。

プリンタ構成情報を変換する

この節では、サイトにある SunOS 5.5.1 またはその互換バージョンを稼動しているシステムのプリンタ構成情報を変換して、その情報を印刷クライアントにコピーすることにより、既存のプリンタにアクセスできるようにする方法を説明します。

注 - 既存のプリンタの台数が少ない場合は、Solaris プリンタマネージャか Admintool を使用してプリンタへのアクセスを追加する方が、プリンタ構成情報を変換して印刷クライアントに配布するよりも簡単です。プリンタへアクセスを追加する方法については、表 4–1 を参照してください。

プリンタ構成情報を変換する作業マップ

表 4–2 に、プリンタ構成情報を変換する作業の概要を示します。

表 4–2 作業マップ: プリンタ構成情報を変化する

作業	説明	手順の説明
既存のプリンタ構成情報を変換する	*unOS 5.5.1* または互換バージョンのシステムのプリンタ構成情報を変換する SunOS 5.5.1 リリースを使用しているサイトでは、/etc/lp/printers ディレクトリ内のプリンタ構成情報を、/etc/printers.conf 構成ファイルに変換する。通常、この作業は 1 回行うだけでよい	91ページの「SunOS 5.5.1 リリースのシステムの印刷情報を変換する方法」を参照
	SunOS 4.1 を稼動するシステムのプリンタ構成情報を変換する SunOS 4.1 ソフトウェアを使用しているサイトでは、4.1 システムの /etc/printcap ファイル内のプリンタ構成情報を、/etc/printers.conf 構成ファイルに変換する。通常、この作業は 1 回行うだけでよい	91ページの「SunOS 4.1 リリースのシステムの印刷情報を変換する方法」を参照
NIS+ (+xfn) のプリンタ構成情報を NIS+ 形式に変換する	下層の xfn アプリケーション層を持たない NIS+ ネームサービスでプリンタ構成情報を管理するとアクセス性能が向上する	92ページの「NIS+ (+xfn) のプリンタ構成情報を NIS+ 形式に変換する方法」を参照

既存のプリンタ構成情報の変換

既存のプリンタ構成情報は、Solaris 8 またはその互換バージョンをインストールあるいはアップグレードするときに自動的に変換されます。この節では、SunOS 5.5.1 リリースあるいは SunOS 4.1 のシステムのプリンタ構成情報を、/etc/printers.conf プリンタ構成ファイルに変換する方法を説明します。変換作業を自動化するために、2 つの新しい印刷管理コマンドのいずれかを使用します。

- conv_lp(1M) コマンドは、SunOS 5.8 システムの /etc/lp/printers ディレクトリ内の情報を、そのシステムの /etc/printers.conf ファイルのエントリに変換します。手順については、91ページの「SunOS 5.5.1 リリースのシステムの印刷情報を変換する方法」を参照してください。
- conv_lpd(1M) コマンドは、SunOS 4.1 システムの /etc/printcap 構成ファイル内の情報を、/etc/printers.conf ファイルのエントリに変換します。手順

については、91ページの「SunOS 4.1 リリースのシステムの印刷情報を変換する方法」を参照してください。

ネームサービスを使用していない場合、サイトにある既存のプリンタを含む、/etc/printers.conf のマスターファイルを作成します。次に、このマスターファイルをすべての印刷クライアントにコピーするか、(ネームサービスを使用している場合は)、NIS または NIS+ にロードします。新しい印刷クライアントは、最初から、サイトにある既存のプリンタにアクセスできるようにしておくことを推奨します。

注意 - NIS または NIS+ ネームサービスを使用してプリンタ情報を構成している場合、印刷クライアント上の /etc/printers.conf ファイルは使用しないでください。印刷クライアントは、最初に /etc/printers.conf ファイルを使用してプリンタを検出します。そのときに、/etc/printers.conf ファイル内の情報と、NIS マップまたは NIS+ マップ内のプリンタ情報に矛盾がある場合は、予期せぬ結果が生じることがあります。この問題を回避するために、印刷クライアントが NIS ネームサービスまたは NIS+ を使用してプリンタ情報を構成するときは、印刷クライアント上の /etc/printers.conf ファイルを削除してください。

▼ SunOS 5.5.1 リリースのシステムの印刷情報を変換する方法

1. **SunOS 5.8** を実行しているシステムに、スーパーユーザーとしてログインします。

2. そのシステムの /etc/lp/printers ディレクトリ内のプリンタ構成情報を /etc/printers.conf ファイルに変換します。

```
# /usr/lib/print/conv_lp
```

▼ SunOS 4.1 リリースのシステムの印刷情報を変換する方法

1. **SunOS 4.1** システムの /etc/printcap ファイルを、**SunOS 5.8** で実行しているシステムにコピーします。

2. /etc/printcap ファイルをコピーした、**SunOS 5.8** を実行しているシステムに、スーパーユーザーとしてログインします。

3. /etc/printcap ファイル内のプリンタ構成情報を /etc/printers.conf ファイルに変換します。

```
# /usr/lib/print/conv_lpd
```

NIS+ (+xfn) のプリンタ構成情報を NIS+ 形式に変換する方法

次の変換スクリプトは、Solaris 8 リリースが動作するシステムでだけ実行できます。

1. **NIS+** マスターにスーパーユーザーとしてログインします。

2. たとえば /tmp/convert という名前で次の変換スクリプトをシステムにコピーします。

```
#!/bin/sh
#
# Copyright (C) 1999 by Sun Microsystems, Inc.
# All Rights Reserved
#
PRINTER=""

for LINE in `lpget -n xfn list | tr "\t " "^A^B"` ; do
 LINE=`echo ${LINE} | tr "^A^B" "\t " | sed -e 's/^  \t//g'`
 case "${LINE}" in
  *:)
   PRINTER=`echo ${LINE} | sed -e 's/://g'`
   ;;
  *=*)
   lpset -n nisplus -a "${LINE}" ${PRINTER}
   ;;
 esac
done
```

注 - カット＆ペーストを使用してこのスクリプトのテキストファイルを作成する場合は、2 つある ^A^B (caratAcaratB) シーケンスを Control A Control B に変更してください。

---

3. スクリプトを実行可能ファイルにします。

```
# chmod 755 /tmp/convert
```

4. 変換スクリプトを実行します。

```
# /tmp/convert
```

第 5 章

プリンタの管理手順

この章では、プリンタを管理する手順について説明します。この章で説明する手順は次のとおりです。

- 127ページの「印刷要求を別のプリンタに移動する方法」
- 129ページの「印刷要求の優先順位を変更する方法」

印刷と LP 印刷サービスの概要については、第 2 章を参照してください。

プリンタと印刷スケジューラの管理

この節では、プリンタと印刷スケジューラを管理するために日常的に行う作業について説明します。

プリンタとプリンタアクセスの削除

プリンタの交換が必要な場合や、プリンタを別の場所に移動したい場合は、プリンタサーバーから物理的に削除する前に、LP 印刷サービスからプリンタ情報を削除しなければなりません。また、プリンタ上の現在の印刷要求がすべて印刷されるか、別のプリンタに移動して印刷されるかを確認する必要があります。

プリンタ情報をプリンタサーバーから削除するだけでなく、印刷クライアントまたはネットワークネームサービスからも削除する必要があります。プリンタサーバーからローカルプリンタを削除する場合は、印刷クライアントまたはネットワークネームサービスからリモートプリンタエントリを削除する必要があります。プリンタを別のプリンタサーバーに移動する場合は、印刷クライアントまたはネットワークネームサービスから古いリモート印刷エントリを削除し、リモートプリンタへのアクセスを新しい位置に追加する必要があります。

ローカルとリモートのプリンタの削除方法については、96ページの「プリンタとリモートプリンタへのアクセスを削除する方法」を参照してください。Solaris プリンタマネージャを使用して、ローカルプリンタまたはリモートプリンタを削除できます。ただし、Solaris プリンタマネージャでは、待ち行列に入っている印刷要求を別のプリンタに移動できません。

▼ プリンタとリモートプリンタへのアクセスを削除する方法

1. 削除したいプリンタへアクセスできる印刷クライアントに、スーパーユーザーまたは `lp` としてログインします。

2. 印刷クライアントからプリンタに関する情報を削除します。

```
print-client# lpadmin -x printer-name
```

-x	指定したプリンタを削除する
printer-name	削除したいプリンタ名

指定したプリンタに関する情報が、印刷クライアントの /etc/lp/printers ディレクトリから削除されます。

3. 印刷クライアントが同じプリンタサーバー上の別のプリンタを使用しない場合は、そのプリンタサーバーに関する情報を印刷クライアントから削除します。

```
print-client# lpsystem -r print-server
```

-r	指定したプリンタサーバーを削除する
print-server	削除したいプリンタサーバー名

プリンタサーバーが、印刷クライアントの /etc/lp/Systems ファイルから削除されます。

4. プリンタへアクセスできる各印刷クライアント上で、97ページの手順 2 から 97 ページの手順 3 までを繰り返します。

5. プリンタサーバーにスーパーユーザーまたは lp としてログインします。

6. プリンタ上で印刷要求を停止します。

```
print-server# reject printer-name
```

reject *printer-name*	指定したプリンタの印刷要求を拒否する

この手順を実行すると、プリンタの削除処理中は、そのプリンタの待ち行列に新しい要求が入らなくなります。詳細は、120ページの「プリンタへの印刷要求を受け付けるまたは拒否する方法」を参照してください。

7. プリンタを停止します。

```
print-server# disable printer-name
```

この手順を実行すると、印刷要求の印刷は停止されます。印刷を停止する方法については、122ページの「プリンタを使用可能または使用不可にする方法」を参照してください。

8. 待ち行列に残っている印刷要求がある場合は、別のプリンタに移動します。

印刷要求を別のプリンタに移動する方法については、127ページの「印刷要求を別のプリンタに移動する方法」を参照してください。

9. プリンタをプリンタサーバーから削除します。

```
print-server# lpadmin -x printer-name
```

プリンタの構成情報が、プリンタサーバーの /etc/lp/printers ディレクトリから削除されます。

10. 削除したばかりのプリンタを使用していた印刷クライアントが、まだプリンタサーバー上で別のプリンタを使用中でなければ、その印刷クライアントに関する情報を削除します。

```
print-server# lpsystem -r print-client1 [,print-client2...]
```

-r	指定したプリンタサーバーを削除する
print-client	プリンタサーバーから削除したい印刷クライアント名。このコマンドで複数の印刷クライアントを指定できる。印刷クライアント名を区切るには空白またはコンマを使用する。空白を使用する場合は、印刷クライアントのリストを引用符で囲む

指定した印刷クライアントが、プリンタサーバーの /etc/lp/Systems ファイルから削除されます。

11. プリンタ情報が削除されていることを確認します。

a. 印刷クライアント上でプリンタ情報が削除されていることを確認します。

```
print-client$ lpstat -p printer-name -l
```

上記のコマンドの出力で、プリンタが存在しないことを示すエラーが表示されます。

b. プリンタサーバー上でプリンタ情報が削除されていることを確認します。

```
print-server$ lpstat -p printer-name -l
```

上記のコマンドの出力で、プリンタが存在しないことを示すエラーが表示されます。

例 — プリンタとリモートプリンタへのアクセスを削除する

次の例では、コマンドは印刷クライアント terra とプリンタサーバー jupiter からプリンタ luna を削除し、印刷クライアント terra をプリンタサーバーから削除します。

```
terra# lpadmin -x luna
Removed ``luna''.
terra# lpstat -p luna -l
jupiter# lpadmin -x luna
jupiter# lpsystem -r terra
Removed ``terra''.
jupiter# lpstat -p luna -l
```

プリンタの状態のチェック

多くの日常的なプリンタ管理作業には、LP 印刷サービスや特定のプリンタの状態に関する情報が必要です。たとえば、どのプリンタが使用できるかを判別し、そのプリンタの特性を検査しなければならない場合があります。lpstat コマンドを使用すると、LP 印刷サービスや特定のプリンタに関する状態情報を調べることができます。

▼ プリンタの状態をチェックする方法

1. ネットワーク上の任意のシステムにログインします。

2. lpstat コマンドを使用してプリンタの状態をチェックします。

   ここには、最も一般的に使用するオプションのみを掲載してあります。他のオプションについては、lpstat(1) のマニュアルページを参照してください。

```
$ lpstat [-d] [-p printer-name [-D] [-l]] [-t]
```

−d	システムのデフォルトプリンタが表示される
−p *printer-name*	プリンタが使用可能かアイドル状態か、いつ使用可能または使用不可になったか、および印刷要求を受け付けているかどうかが表示される。 このコマンドで複数の印刷クライアントを指定できる。印刷クライアント名を区切るには空白またはコンマを使用する。空白を使用する場合は、印刷クライアントのリストを引用符で囲む。*printer-name* を指定しなければ、すべてのプリンタの状態が表示される
−D	指定した *printer-name* の記述が表示される
−l	指定した *printer-name* の特性が表示される
−t	すべてのプリンタの状態、使用可能かどうか、印刷要求を受け付けているかどうかなど、LP 印刷サービスに関する状態情報が表示される

例 — プリンタの状態をチェックする

次の例では、コマンドはシステムのデフォルトプリンタ名を表示します。

```
$ lpstat -d
system default destination: luna
```

次の例では、コマンドはプリンタ luna の状態を表示します。

```
$ lpstat -p luna
printer luna is idle. enabled since Jul 12 11:17 1999. available.
```

次の例では、コマンドはプリンタ asteroid と luna の記述を表示します。

```
$ lpstat -p "asteroid luna" -D
printer asteroid faulted. enabled since Jul 12 11:35 1999. available.
unable to print: paper misfeed jam

Description: Printer by break room
printer luna is idle. enabled since Jul 12 11:36 1999. available.
Description: Printer by server room.
```

次の例では、コマンドはプリンタ luna の特性を表示します。

```
$ lpstat -p luna -l
printer luna is idle. enabled since Mon Jul 12 15:02:32 ...
        Form mounted:
        Content types: postscript
        Printer types: PS
        Description:
        Connection: direct
        Interface: /usr/lib/lp/model/standard
        After fault: continue
        Users allowed:
                (all)
        Forms allowed:
                (none)
        Banner not required
        Character sets:

        Default pitch:
        Default page size: 80 wide 66 long
        Default port settings:
```

印刷スケジューラの再起動

印刷スケジューラ lpsched は、プリンタサーバー上の印刷要求を処理します。ただし、印刷スケジューラがシステム上で動作を停止したために、印刷要求の受け付けや印刷が停止されることがあります。

印刷スケジューラを再起動するには、/usr/lib/lp/lpsched コマンドを使用できます。印刷スケジューラが動作を停止するときに印刷要求が印刷中だった場合は、印刷スケジューラを再起動すると、その印刷要求全体が印刷されます。

▼ 印刷スケジューラを停止する方法

1. プリンタサーバーにスーパーユーザーまたは lp としてログインします。
2. 印刷スケジューラが動作しているかどうかをチェックします。

```
# lpstat -r
```

印刷スケジューラが動作していない場合は、「scheduler is not running」というメッセージが表示されます。

3. 印刷スケジューラが動作している場合は停止します。

```
# /usr/lib/lp/lpshut
```

▼ 印刷スケジューラを再起動する方法

1. プリンタサーバーにスーパーユーザーまたは lp としてログインします。
2. 印刷スケジューラが動作しているかどうかをチェックします。

```
# lpstat -r
```

印刷スケジューラが動作していない場合は、「scheduler is not running」というメッセージが表示されます。

3. 印刷スケジューラが動作していない場合は起動します。

```
# /usr/lib/lp/lpsched
```

その他のプリンタ定義の設定とリセット

この節では、プリンタ定義の設定またはリセットの手順を説明します。次のプリンタ定義の一部は、Solaris プリンタマネージャを使用して設定できます。次の手順では、迅速にプリンタ定義を設定またはリセットするために、lp コマンドを使用しています。

▼ プリンタ記述を追加する方法

1. プリンタサーバーにスーパーユーザーまたは lp としてログインします。

2. lpadmin(1M) コマンドを使用してプリンタ記述を追加します。

```
# lpadmin -p printer-name -D "comment"
```

p *printer-name*	記述を追加するプリンタ名
-D *"comment"*	設置場所や管理担当者など、プリンタの特性を指定する。シェルが解釈する文字 (*、?、\、!、^ など) は、一重引用符で囲む

プリンタ記述はプリンタサーバーの /etc/lp/printers/*printer-name*/comment ファイルに追加されます。

3. Description 情報をチェックします。

```
$ lpstat -p printer-name -l
```

例 — プリンタ記述を追加する

次の例では、コマンドはプリンタ luna のプリンタ記述を追加します。

```
# lpadmin -p luna -D "Nathans office"
```

デフォルトプリンタの指定

印刷コマンドを使用するときにプリンタ名を入力しなくてもすむように、ユーザーのデフォルトプリンタを指定できます。あるプリンタをデフォルトとして指定する

前に、そのプリンタをシステム上の印刷サービスに認識させなければなりません。次のいずれかを設定すれば、ユーザーのデフォルトプリンタを設定できます。

- LPDEST 環境変数
- PRINTER 環境変数
- ユーザーの .PRINTERS ファイルの _default 変数
- システムのデフォルトプリンタ (lpadmin -d コマンドまたは Admintool を使用)

アプリケーションがプリンタを指定する場合は、システムのデフォルトプリンタを設定したかどうかに関係なく、その出力先が印刷サービスに使用されます。アプリケーションにプリンタの出力先がない場合や、印刷コマンドの使用時にプリンタ名が指定されていない場合は、印刷コマンドはデフォルトプリンタを特定の順序で検索します。表 5–1 は、システムのデフォルトプリンタの検索順序を示しています。

表 5–1 デフォルトプリンタの検索順序

検索順序	/usr/bin/lp コマンドを使用	**SunOS/BSD** 互換コマンド **(**lpr、lpq、または lprm**)** を使用
1	LPDEST 変数	PRINTER 変数
2	PRINTER 変数	LPDEST 変数
3	システムのデフォルトプリンタ	システムのデフォルトプリンタ

▼ システムのデフォルトプリンタを設定する方法

1. デフォルトプリンタを設定したいシステムにスーパーユーザーまたは lp としてログインします。
2. lpadmin コマンドを使用してシステムのデフォルトプリンタを設定します。

```
# lpadmin -d [printer-name]
```

-d *printer-name* システムのデフォルトプリンタとして割り当てるプリンタ名。*printer-name* を指定しなければ、システムはデフォルトプリンタなしで設定される

デフォルトプリンタ名がシステムの /etc/lp/default ファイルに入力されます。

3. lpstat コマンドを使用して、システムのデフォルトプリンタをチェックします。

```
$ lpstat -d
```

例 — システムのデフォルトプリンタを設定する

次の例では、プリンタ luna をシステムのデフォルトプリンタとして設定します。これは、LPDEST または PRINTER 環境変数が設定されていない場合、luna がシステムのデフォルトプリンタとして使用されることを意味します。

```
# lpadmin -d luna
# lpstat -d
system default destination: luna
```

バナーページの印刷

バナーページには、印刷要求を出したユーザー、印刷要求 ID、要求の印刷時期が出力されます。また、バナーページには、ユーザーがプリントアウトを識別しやすいように変更可能なタイトルを付けることもできます。

バナーページは、印刷ジョブの所有者を簡単に識別できるようにします。これは、多数のユーザーが同じプリンタにジョブを依頼するときに特に便利です。ただし、バナーページを印刷すると用紙の消費量が増えますが、1 台のプリンタを使用するユーザーが少ない場合は必要ないことがあります。また場合によっては、バナーページを印刷しない方がよいこともあります。たとえば、プリンタに支払い小切手などの特殊な用紙やフォームが装着されている場合は、バナーページを印刷すると問題が起きることがあります。

デフォルトでは、印刷サービスはバナーページを強制的に印刷します。ただしユーザーは、印刷要求を出すときにバナーページの印刷をオフにするかどうかを選択できます。この選択肢は lpadmin コマンドまたは Admintool を通じて設定できます。ユーザーが選択できるようにする場合、ユーザーがバナーページの印刷をオフに切り替えるには、–o nobanner オプションを使用する必要があります。

また、プリンタのバナーページをオフにして印刷できないようにすることもできます。これは、バナーページが不要な状況では重要です。バナーページの印刷は、lpadmin コマンドを使用することによってオフにできます。

表 5–2 バナーページの印刷

コマンド	バナーページ印刷は	変更
`lpadmin -p` *printer* `–o banner` または `lpadmin -p` *printer* `–o banner=always`	常に行われる	一般ユーザーが `lp -o nobanner` コマンドを使用すると、要求は印刷されるが `nobanner` 引数は無視される root または lp の場合は、`nobanner` 引数が使用される
`lpadmin -p` *printer* `–o nobanner` `lpadmin -p` *printer* `–o banner=optional`	デフォルトでオン。ただし、`lp -o nobanner` コマンドを使えば要求単位で無効にできる	該当せず
`lpadmin -p` *printer* `–o banner=never`	無効	できない

詳細は、107ページの「バナーページをオフにする方法」を参照してください。

▼ バナーページをオプションにする方法

1. プリンタサーバーにスーパーユーザーまたは `lp` としてログインします。
2. `lpadmin` コマンドを使用してバナーページをオプションにします。

```
# lpadmin -p printer-name -o nobanner=optional
```

`-p` *printer-name*	バナーページをオプションにするプリンタ名
`–o nobanner=optional`	ユーザーが印刷要求を出すときにバナーページなしを指定できるようにする

すべての印刷要求でバナーページを強制印刷したい場合は、
`-o banner=always` オプションを指定します。
バナーページの設定がプリンタサーバーの
`/etc/lp/printers/`*printer-name*`/configuration` ファイルに入力されます。

3. 次のコマンドの出力には、「`Banner not required`」という行が入っています。

```
$ lpstat -p printer-name -l
```

例 — バナーページをオプションにする

次の例では、コマンドはユーザーがプリンタ `luna` 上でバナーページなしを要求できるようにします。

```
# lpadmin -p luna -o nobanner=optional
```

▼ バナーページをオフにする方法

1. プリンタサーバーにスーパーユーザーまたは `lp` としてログインします。

2. `lpadmin` コマンドを使用してバナー印刷をオフにします。

```
lpadmin -p printer-name -o nobanner=never
```

–p *printer-name*	バナーページ印刷をオプションにするプリンタ名
–o nobanner=never	どのような状況でもバナーページ印刷を無効にする

バナーページの設定は、プリンタサーバーの
`/etc/lp/printers/`*printer-name*`/configuration` ファイルに指定します。

3. 次のコマンドの出力に `Banner not printed` という行が含まれていることを確認します。

```
$ lpstat -p printer-name -l
```

4. プリンタに印刷要求を送ってバナーページが印刷されないことを確認します。

例 — バナーページ印刷をオフにする

次の例では、プリンタ luna に対するバナーページ印刷をオフにします。

```
# lpadmin -p luna -o nobanner=never
```

プリンタクラスの設定

印刷サービスを使用すると、複数のローカルプリンタを 1 つのクラスにグループ化できます。この作業は、lpadmin -c コマンドを使用しなければ実行できません。

プリンタクラスを設定すると、ユーザーは印刷要求の出力先として (個々のプリンタではなく) そのクラスを指定できます。そのクラスで空いている最初のプリンタが印刷に使用されます。その結果、プリンタはできる限りビジーに保たれるので、応答時間が短縮されます。

印刷サービスに認識されるデフォルトのプリンタクラスはなく、定義したプリンタクラスのみが存在することになります。プリンタクラスを定義するには、次の 3 つの方法があります。

- プリンタタイプ別 (PostScript など)
- 場所別 (5 階など)
- 作業グループや部署別 (経理など)

また、1 つのクラスには特定の順序で使用される複数のプリンタを含めることができます。LP 印刷サービスでは、常に各プリンタがクラスに追加された順番に従って利用できるプリンタをチェックします。したがって、最初に高速プリンタにアクセスしたい場合は、高速プリンタを低速プリンタよりも先にクラスに追加します。その結果、高速プリンタで最大限の印刷要求が処理されることになります。低速プリンタは、高速プリンタが使用されているときのバックアッププリンタとして確保されます。

注 - 印刷要求の負荷は、ローカルプリンタのクラス内のプリンタ間でのみ調整されます。

クラス名も、プリンタ名と同様に固有の名前でなければなりません。クラス名は 14 文字以内の英数字で、下線を使用できます。

プリンタクラスは定義しなくてもかまいません。プリンタクラスを使用するとネットワーク上のユーザーに利点があると判断した場合にのみ、クラスを追加してください。

▼ プリンタのクラスを定義する方法

1. プリンタサーバーにスーパーユーザーまたは lp としてログインします。
2. lpadmin コマンドを使用して、プリンタのクラスを定義します。

```
# lpadmin -p printer-name -c printer-class
```

-p *printer-name*　　プリンタのクラスに追加するプリンタ名

-c *printer-class*　　プリンタのクラス名

指定したプリンタが、プリンタサーバーの /etc/lp/classes/*printer-class* ファイル内でそのクラスのリストの最後に追加されます。プリンタクラスが存在しない場合は、作成されます。

3. lpstat コマンドを使用して、プリンタクラスのプリンタを表示します。

```
$ lpstat -c printer-class
```

例 — プリンタのクラスを定義する

次の例では、コマンドはプリンタ luna をクラス roughdrafts に追加します。

```
# lpadmin -p luna -c roughdrafts
```

障害の通知の設定

事前に選択しておくと、印刷サービスはプリンタ障害を検出したときに通知できます。次のいずれかの方法を選択すると、lpadmin -A コマンドまたは Solaris プリンタマネージャを使用してプリンタ障害通知を受け取ることができます。

- root がログインしている端末にメッセージを書き込む
- root に電子メールを送る

■ 通知しない

ただし、lpadmin -A コマンドを使用すると、他にも選択したプログラムで指定されるメッセージをオプションとして受信できます。また、すでに知っているエラーに関する通知をオフにすることもできます。

障害通知を配信するプログラムを指定しなければ、障害警告の内容は事前に定義済みのメッセージです。このメッセージは、プリンタが印刷を停止しており、解決が必要であることを示します。

表 5–3 は、lpadmin -A コマンドでプリンタに設定できる警告値を示しています。これらの警告値は、印字ホイール、フォントカートリッジ、フォームについても設定できます。

表 5–3 印刷障害の警告値

–A alert の値	説明
'mail [*user-name*]'	警告メッセージをプリンタサーバー上の root か lp、またはユーザー名として指定した *user-name* に電子メールで送信する
'write [*user-name*]'	警告メッセージをプリンタサーバー上の root か lp のコンソールウィンドウ、またはユーザー名として指定した *user-name* のコンソールウィンドウに送信する。指定したユーザーが警告メッセージを受け取るには、プリンタサーバーにログインしていなければならない
'コマンド'	警告ごとに *command* ファイルを実行する。環境変数とカレントディレクトリは、ファイルの実行の前後で保存復元される
quiet	障害が解決されるまで警告を停止する。この値は、ユーザー (root または指定したユーザー) が繰り返し警告を受け取るときに使用する
none	警告を送信しない。プリンタの障害警告を指定しない場合は、これがデフォルト値である

▼ プリンタの障害警告を設定する方法

1. プリンタサーバーにスーパーユーザーまたは lp としてログインします。
2. lpadmin コマンドを使用してプリンタの障害警告を設定します。

```
# lpadmin -p printer-name -A alert [-W minutes]
```

-p *printer-name*	プリンタ障害の警告を指定するプリンタ名
-A *alert*	プリンタ障害が起きたときに出される警告の種類を指定する。*alert* に有効な値については、表 5–3 を参照。有効な値は mail、write、quiet などである
-W *minutes*	障害警告が出される間隔 (分単位) を指定する。このオプションを指定しなければ、警告は一度だけ送信される

障害警告の設定は、プリンタサーバーの /etc/lp/printers/*printer-name*/alert.sh ファイルに入力されます。

3. 次のコマンドの出力から、「On fault」見出しに続く情報をチェックします。

```
$ lpstat -p printer-name -l
```

例 — プリンタの障害警告を設定する

次の例で、コマンドは障害警告をユーザー joe に電子メールで送信し、その後は 5 分ごとに送信するようにプリンタ mars を設定します。

```
# lpadmin -p mars -A 'mail joe' -W 5
```

次の例で、コマンドは障害警告をコンソールウィンドウに送信し、その後は 10 分ごとに送信するようにプリンタ venus を設定します。

```
# lpadmin -p venus -A write -W 10
```

次の例で、コマンドはプリンタ mercury の障害警告を停止します。

```
# lpadmin -p mercury -A none
```

次の例で、コマンドはプリンタ venus の障害が解決されるまで、障害警告を停止します。

```
# lpadmin -p venus -A quiet
```

プリンタの障害回復の設定

障害通知を送信しないことを選択した場合には、問題を解決するために印刷障害を検出することができます。LP 印刷サービスは、障害のあるプリンタを継続して使用しません。プリンタ障害の警告に加えて、印刷要求が必要とするときに、印字ホイール、フォントカートリッジ、およびフォームを取り付けるようにシステム管理者に知らせる警告も設定できます。

`lpadmin -F` コマンドを使用すると、プリンタ専用の障害回復オプションを定義できます。これは、Solaris プリンタマネージャではできません。

プリンタ障害は、用紙切れやトナーカートリッジの交換が必要であるなど、きわめて単純な場合があります。より重大な問題としては、完全なプリンタ障害や電源障害などがあります。プリンタ障害を解決すると、障害が発生したときに有効だった印刷要求は、次のいずれかの方法で印刷を開始します。

- 印刷を最初から開始する
- 印刷を停止したページの先頭から印刷を再開する
- プリンタを使用可能にした後に、印刷を停止したページの先頭から印刷を再開する

印刷を停止したページの先頭から印刷を再開するには、印刷フィルタが必要です。印刷フィルタは、プリンタに使用される制御シーケンスを記録してページ境界を追跡します。この処理は、印刷サービスに使用されるデフォルトフィルタでは実行できません。指定した印刷フィルタで回復処理を実行できなければ、印刷サービスから通知されます。フィルタの作成方法については、189ページの「新しい印刷フィルタを作成する方法」を参照してください。

プリンタ障害を解決した直後に印刷を再開したい場合は、`enable` コマンドを使用してプリンタを使用可能にします。

表 5–4 は、`lpadmin -F` コマンドでプリンタに設定できる障害回復値を示しています。

表 5–4　プリンタ障害回復の値

−F *recover-options* の値	説明
beginning	障害回復後に、ファイルの先頭から印刷を再開する
continue	障害回復後に、印刷が停止されたページの先頭から印刷を開始する。この回復オプションには印刷フィルタが必要
wait	障害回復後に、プリンタを使用可能にするまで印刷が停止される。(enable コマンドで) プリンタを使用可能にすると、印刷は停止されたページの先頭から始まる。この回復オプションには印刷フィルタが必要

▼ プリンタの障害回復を設定する方法

1. プリンタサーバーにスーパーユーザーまたは lp としてログインします。
2. lpadmin(1M) コマンドを使用してプリンタ障害から回復する方法を設定します。

```
# lpadmin -p printer-name -F recovery-options
```

-p *printer-name*　障害からの回復方法を指定するプリンタ名

-F *recovery-options*　beginning、continue、wait の 3 つの有効な回復オプション。*recovery-options* に有効な値については、表 5–4 を参照

障害回復の設定がプリンタサーバーの /etc/lp/printers/*printer-name*/configuration ファイルに入力されます。

3. 次のコマンドの出力から、「After fault」見出しに続く情報をチェックします。

```
$ lpstat -p printer-name -l
```

例 — プリンタの障害回復を設定する

次の例では、コマンドは、印刷が停止されたページの先頭から再開するようにプリンタ luna を設定します。

```
# lpadmin -p luna -F continue
```

プリンタへのユーザーアクセスを制限する

利用できるプリンタの一部またはすべてにアクセスできるユーザーを制限する必要がある場合があります。たとえば、一部のユーザーが高品質プリンタ上で印刷できないようにして経費を抑えることができます。プリンタへのユーザーアクセスを制限するには、プリンタサーバー上で lpadmin -u コマンドを使用して「許可」リストと「拒否」リストを作成できます (Solaris プリンタマネージャを使用すると、許可リストのみを作成できます)。どちらのリストも作成しなければ、プリンタはそこにアクセスできる全ユーザーが利用できます。

許可リストには、指定したプリンタへのアクセスを許可されるユーザー名が入っています。拒否リストには、指定したプリンタへのアクセスを拒否されるユーザー名が入っています。

許可リストと拒否リストには、次の規則が適用されます。

リストの状態	アクセスの制限
許可リストも拒否リストも作成しない、または両方のリストが空	全ユーザーがそのプリンタを使用できる
許可リストで all を指定する	そのプリンタには全ユーザーがアクセスできる
拒否リストで all を指定する	サーバー上の root と lp 以外の全ユーザーのアクセスが拒否される
許可リストにエントリを作成する	リストに指定されているユーザーだけがプリンタにアクセスできる。拒否リストは無視される
拒否リストを作成し、許可リストは作成しないか許可リストを空にする	拒否リストで指定されたユーザーはプリンタにアクセスできない

実際にプリンタへのアクセスを制御しているのはプリンタサーバーなので、許可リストと拒否リストを作成できるのはプリンタサーバー上でだけです。許可リストと拒否リストを作成した場合、プリンタサーバーは、プリンタへのユーザーアクセスを排他的に制御します。

表 5–5 は、プリンタへのユーザーアクセスを制限するために許可リストまたは拒否リストに追加できる値を示しています。

表 5–5 許可リストと拒否リストの値

user-list の値	説明
user	任意のシステム上の特定ユーザー
`all`	すべてのシステム上の全ユーザー
`none`	すべてのシステム上の全ユーザーが該当しない
system`!`*user*	特定システム上の特定ユーザー
`!`*user*	ローカルシステム上の特定ユーザー
`all!`*user*	任意のシステム上の特定ユーザー
`all!all`	すべてのシステム上の全ユーザー
system`!all`	特定システム上の全ユーザー
`!all`	ローカルシステム上の全ユーザー

▼ プリンタへのユーザーアクセスを制限する方法

1. プリンタサーバーにスーパーユーザーまたは `lp` としてログインします。
2. `lpadmin` コマンドを使用して、プリンタへのユーザーアクセスを許可または拒否します。

```
# lpadmin -p printer-name -u allow:user-list [deny:user-list]
```

-p *printer-name*	許可または拒否ユーザーアクセスリストを適用するプリンタ名
-u allow:*user-list*	許可ユーザーアクセスリストに追加するユーザー名。このコマンドで複数のユーザーを指定できる。空白またはコンマを使用して名前を区切る。空白を使用する場合は、名前のリストを引用符で囲む。*user-list* に有効な値については、表 5–5 を参照
-u deny:*user-list*	拒否ユーザーアクセスリストに追加するユーザー名。このコマンドで複数のユーザーを指定できる。空白またはコンマを使用して名前を区切る。空白を使用する場合は、名前のリストを引用符で囲む。*user-list* に有効な値については、表 5–5 を参照

指定したユーザーが、プリンタサーバーの次のファイル内でプリンタの許可または拒否ユーザーアクセスリストに追加されます。

```
/etc/lp/printers/printer-name/users.allow
```

```
/etc/lp/printers/printer-name/users.deny
```

注 - 許可ユーザーアクセスリストの *user-list* に none を指定した場合、プリンタサーバー用に次のファイルは作成されません。

/etc/lp/printers/*printer-name*/alert.sh

/etc/lp/printers/*printer-name*/alert.var

/etc/lp/printers/*printer-name*/users.allow

/etc/lp/printers/*printer-name*/users.deny

---

3. 次のコマンドの出力から、「Users allowed」または「Users denied」見出しに続く情報をチェックします。

```
$ lpstat -p printer-name -l
```

例 — プリンタへのユーザーアクセスを制限する

次の例で、コマンドはユーザー nathan と george によるプリンタ luna へのアクセスのみを許可します。

```
# lpadmin -p luna -u allow:nathan,george
```

次の例で、コマンドはユーザー nathan と george によるプリンタ asteroid へのアクセスを拒否します。

```
# lpadmin -p asteroid -u deny:"nathan george"
```

印刷要求の管理

ユーザーが印刷クライアントから印刷要求を出すと、その印刷要求はプリンタサーバー上の待ち行列に追加されてからプリンタに送信されます。印刷要求が待ち行列に入っている間は、クライアントシステムからの要求で取り消したり、その状態情報を取得したりできます。LP コマンドで印刷要求の移動、停止、再開または優先度の変更などを実行するには、プリンタサーバーにログインする必要があります。これらの動作によって、印刷サービスを絶えずスムーズに機能させることができます。

表 5–6 は、lp -H コマンドを使用して印刷要求の優先順位を変更するための値を示しています。

表 5–6 印刷要求の優先順位を変更する値

-H *change-priority* の値	説明
–hold	ユーザーが取り消すか、LP 印刷サービスに要求の印刷再開を指示するまで、印刷要求を停止する
–resume	停止されていた印刷要求を待ち行列に戻す。印刷要求は、優先順位と待ち行列内での位置に従って印刷される。すでに印刷中の印刷ジョブを停止すると、resume は印刷要求が次に印刷される要求になるように待ち行列の先頭に配置する
–immediate	印刷要求を待ち行列の先頭に配置する。要求がすでに印刷中であれば、次の要求をすぐに印刷できるように、印刷中の要求を停止する

表 5–6　印刷要求の優先順位を変更する値　続く

▼ 印刷要求の状態をチェックする方法

1. ネットワーク上の任意のシステムにログインします。

2. lpstat コマンドを使用して、プリンタと印刷要求の状態をチェックします。

   ここには、最も一般的に使用するオプションだけを掲載してあります。他の有効なオプションについては、lpstat(1) のマニュアルページを参照してください。

```
$ lpstat -o [list] | -u [user-list]
```

–o *list*	特定のプリンタの印刷要求の状態が表示される。*list* には、1 つ以上のプリンタ名、プリンタクラス名、または印刷要求 ID を指定できる。 *list* には、複数のプリンタ名、クラス名、ID を指定できる。値を区切るには空白またはコンマを使用する。空白を使用する場合は、値のリストを引用符で囲む。 *list* を指定しなければ、すべてのプリンタへの印刷要求の状態が表示される
–u *user-list*	特定のユーザーの印刷要求の状態が表示される。*user-list* には 1 人以上のユーザー名を指定できる。 このコマンドで複数のユーザーを指定できる。ユーザー名を区切るには空白またはコンマを使用する。空白を使用する場合は、ユーザー名のリストを引用符で囲む。 *user-list* を指定しなければ、すべてのユーザーの印刷要求の状態が表示される

   lpstat コマンドを使用して印刷要求の状態をチェックすると、印刷要求ごとに 1 行ずつ表示されます。各行には、左から右に要求 ID、ユーザー、出力バイト数、要求日時、「being filtered」のような要求に関する情報が表示されます。

例 — 印刷要求の状態をチェックする

次の例で、コマンドはユーザー fred の印刷要求がプリンタ luna への待ち行列に 1 つ入っていることを示しています。

```
$ lpstat
luna-1    fred     1261    Jul 12 17:34
```

次の例は、ユーザー paul には待ち行列に入っている印刷要求がないことを示しています。

```
$ lpstat -u paul
```

次の例は、プリンタ moon に 2 つの印刷要求があることを示しています。

```
$ lpstat -o moon
moon-78    root     1024   Jul 14 09:07
moon-79    root     1024   Jul 14 09:08
```

印刷の処理または停止

enable(1) コマンドと disable(1) コマンドを使用すると、プリンタが待ち行列に入っている要求の印刷を処理または停止するかを制御できます。プリンタを使用不可にすると、プリンタは待ち行列内の要求の印刷を停止します。ただし、要求はそのまま待ち行列に追加されます (要求が待ち行列に追加されないように、プリンタを設定して印刷要求を拒否させなければなりません。印刷要求を拒否する方法については、121ページの「印刷要求の受け付けまたは拒否」を参照してください)。

Solaris プリンタマネージャを使用してプリンタを追加すると、プリンタは有効になり印刷要求を受け付けます。Solaris プリンタマネージャは、それ以上のプリンタ管理は提供しません。

プリンタが使用不可になっている場合は、使用可能にしなければなりません。この状態は、プリンタ障害が起きると発生することがあります。プリンタを使用可能にすると、印刷サービスがそれ以後に印刷待ち行列の要求を拒否しても、待ち行列が空になるまで、印刷待ち行列からの要求が印刷されます。

図 5–1 は、プリンタが使用不可になったときに印刷要求の処理が中断される様子を示しています。

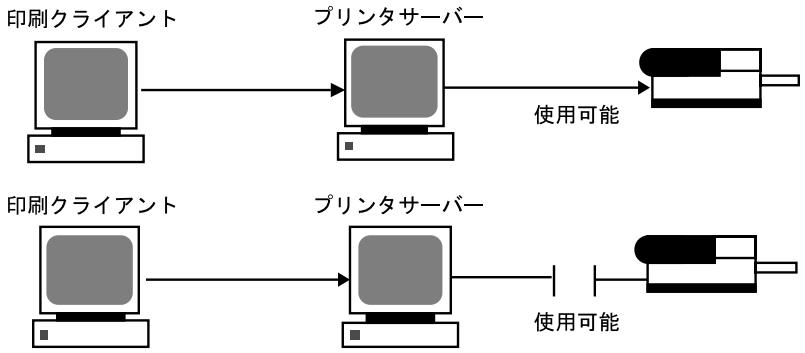


図 5–1　プリンタが使用可能または使用不可になる場合

▼ プリンタへの印刷要求を受け付けるまたは拒否する方法

1. プリンタサーバーにスーパーユーザーまたは `lp` としてログインします。

2. `reject(1M)` コマンドを使用して、プリンタで印刷要求を停止します。

```
# reject [-r "reason"] printer-name
```

–r "reason"	プリンタが印刷要求を拒否している理由をユーザーに示す。理由は格納され、ユーザーがプリンタの状態をチェックする (`lpstat -p`) と表示される
printer-name	印刷要求を停止するプリンタ名

待ち行列に入れられた要求は、プリンタが使用可能になっている限りそのまま印刷されます。印刷を停止するようにプリンタを使用不可にする手順については、122ページの「プリンタを使用可能または使用不可にする方法」を参照してください。

3. `accept(1M)` コマンドを使用して、プリンタで印刷要求を受け付けるようにします。

```
# accept printer-name
```

4. lpstat コマンドを使用してプリンタの状態をチェックし、印刷要求を受け付けているか停止しているかを調べます。

```
$ lpstat -p printer-name
```

例 — プリンタへの印刷要求を受け付けるまたは拒否する

次の例で、コマンドはプリンタ luna の印刷要求を停止します。

```
# reject -r "luna is down for repairs" luna
destination "luna" will no longer accept requests
```

次の例で、コマンドはプリンタ luna が印刷要求を受け付けるように設定します。

```
# accept luna
destination "luna" now accepting requests
```

印刷要求の受け付けまたは拒否

accept コマンドと reject コマンドを使用すると、印刷要求が格納される印刷待ち行列のオンとオフを切り替えることができます。

reject コマンドを使用すると、指定したプリンタの印刷待ち行列がオフになり、新しい印刷要求はプリンタサーバーの待ち行列に入れなくなります。その待ち行列に入っているすべての印刷要求は、そのまま印刷されます。すでに待ち行列に入っている要求の印刷を停止したい場合は、そのプリンタを使用不可にしなければなりません。表 5–7 では accept、reject、enable、および disable コマンドの機能を比較します。

表 5–7 accept/reject コマンドと enable/disable コマンドの機能

コマンド	機能
accept	印刷待ち行列に送信された印刷要求を受け付ける
enable	印刷待ち行列にある要求を印刷する

表 5–7　accept/reject コマンドと enable/disable コマンドの機能　続く

コマンド	機能
reject	印刷待ち行列に送信された印刷要求を拒否する
disable	現在印刷待ち行列にある印刷要求を停止する

プリンタを使用不可にする方法については、119ページの「印刷の処理または停止」を参照してください。

印刷要求が拒否されると、印刷サービスはその要求を出したユーザーにメッセージを送り、指定されたプリンタには印刷要求が受け付けられていないことを通知します。

また、要求を受け付けない理由をコマンド行から指定できます。その理由は、ユーザーがプリンタの待ち行列をチェックしようとするとユーザーのシステムに表示されます。図 5–2 は、印刷待ち行列が拒否されたときに印刷要求が中断される様子を示しています。

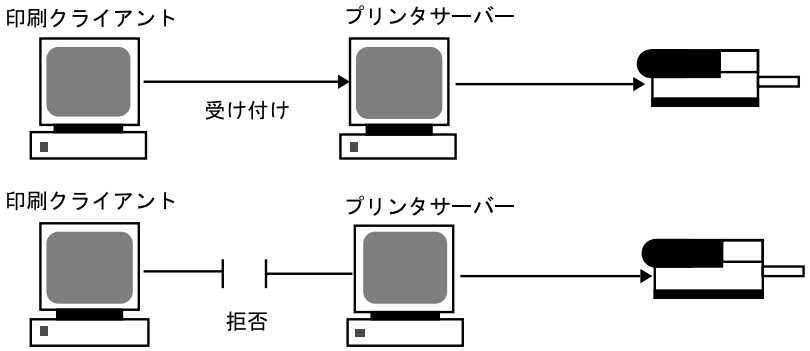


図 5–2　印刷待ち行列が要求を受け付けるか拒否する場合

▼ プリンタを使用可能または使用不可にする方法

1. プリンタサーバーにスーパーユーザーまたは lp としてログインします。
2. disable コマンドを使用して、プリンタ上の印刷要求の印刷を停止します。

```
# disable [-c | -W] [-r "reason"] printer-name
```

`disable`	オプションを指定しなければ、現在のジョブを取り消してプリンタを使用不可にする。現在のジョブは保存され、プリンタが使用可能になると再び印刷される
`–c`	現在のジョブを取り消してから、プリンタを使用不可にする。現在のジョブが後から印刷されることはない
`–W`	現在のジョブが終了するまで待ってからプリンタを使用不可にする
`–r "`*reason*`"`	プリンタが使用不可になった理由をユーザーに示す。理由は格納され、ユーザーがプリンタの状態をチェックする (`lpstat -p`) と表示される
printer-name	印刷要求の印刷を停止するプリンタ名

注 - プリンタのクラスを使用可能または使用不可にすることはできません。個々のプリンタのみを使用可能または使用不可にすることができます。

---

3. `enable` を使用して、プリンタ上で印刷要求の印刷を開始します。

```
# enable printer-name
```

4. プリンタが使用可能になっていることを確認します。

```
$ lpstat -p printer-name
```

例 — プリンタを使用可能または使用不可にする

次の例では、プリンタ `luna` 上の現在のジョブを停止し、後から印刷できるように保存して、プリンタが印刷要求の印刷を停止した理由を表示します。

```
# disable -r "changing the form" luna
```

次の例では、プリンタ `luna` 上で印刷要求の印刷を開始します。

```
# enable luna
printer "luna" enabled
```

印刷要求の取り消し

cancel(1) を使用すると、印刷待ち行列から印刷要求を取り消したり、印刷中のジョブを取り消したりできます。cancel コマンドには、次の 3 つの使用方法があります。

- 要求識別番号 (要求 ID) を使用して要求を取り消す。
- すべてまたは指定したプリンタ上で特定のユーザーからの要求を取り消す。
- 現在印刷中のジョブを取り消す。

cancel を使用すると、要求が取り消され、待ち行列内の次の要求が印刷されることを示すメッセージが表示されます。次の場合にのみ、印刷要求を取り消すことができます。

- 要求を出したユーザーが、要求を出したシステムにログインしている状態で取り消しを実行した場合
- プリンタサーバーの /etc/printers.conf ファイルで user-equivalence オプションが構成されていて、要求を出したユーザーが任意のクライアントシステムから取り消しを実行した場合
- プリンタサーバーにスーパーユーザーまたは lp としてログインしている状態で取り消しを実行した場合

特定の要求を取り消すには、その要求 ID を知る必要があります。要求 ID は、luna-185 のように、プリンタ名、ハイフン、印刷要求番号からなっています。印刷要求を依頼すると、その要求 ID が表示されます。印刷要求 ID を忘れた場合は、-o printer オプションを指定して lpstat コマンドを使用すると ID を調べることができます。

▼ 印刷要求を取り消す方法

1. **他のユーザーの印刷要求を取り消す場合は、スーパーユーザーまたは lp になります。**

2. **lpstat コマンドを使用して、取り消す印刷要求の要求 ID を判別します。**

   詳細は、118ページの「印刷要求の状態をチェックする方法」を参照してください。

3. cancel コマンドを使用して印刷要求を取り消します。

```
$ cancel request-id | printer-name
```

request-id 取り消す印刷要求の要求 ID。このコマンドで複数の要求 ID を指定できる。要求 ID を区切るには空白またはコンマを使用する。空白を使用する場合は、要求 ID のリストを引用符で囲む

printer-name 現在印刷中の印刷要求を取り消したいプリンタを指定する。

このコマンドで複数のプリンタ名を指定できる。プリンタ名を区切るには空白またはコンマを使用する。空白を使用する場合は、プリンタ名のリストを引用符で囲む

4. 印刷要求が取り消されていることを確認します。

```
$ lpstat -o printer-name
```

例 — 印刷要求を取り消す

次の例では、luna-3 と luna-4 の印刷要求を取り消します。

```
$ cancel luna-3 luna-4
request "luna-3" cancelled
request "luna-4" cancelled
```

次の例では、現在プリンタ luna 上で印刷中の印刷要求を取り消します。

```
# cancel luna
request "luna-9" cancelled
```

▼ 特定のユーザーからの印刷要求を取り消す方法

1. **(省略可能)** 他のユーザーの印刷要求を取り消す場合は、スーパーユーザーまたは lp になります。

2. cancel コマンドを使用して、特定のユーザーからの印刷要求を取り消します。

```
$ cancel -u user-list [printer-name]
```

-u *user-list*
: 特定のユーザーの印刷要求を取り消す。

  user-list では複数のユーザー名を指定できる。ユーザー名を区切るには空白またはコンマを使用する。空白を使用する場合は、ユーザー名のリストを引用符で囲む

printer-name
: 指定したユーザーの印刷要求を取り消したいプリンタを指定する

  printer-name では複数のプリンタ名を指定できる。プリンタ名を区切るには空白またはコンマを使用する。空白を使用する場合は、プリンタ名のリストを引用符で囲む

  printer-name を指定しないと、ユーザーの印刷要求はすべてのプリンタで取り消される

例 — 特定のユーザーからの印刷要求を取り消す

次の例で、コマンドはプリンタ `luna` 上でユーザー `george` から依頼されたすべての印刷要求を取り消します。

```
# cancel -u george luna
request "luna-23" cancelled
```

次の例で、コマンドはユーザー `george` から依頼されたすべての印刷要求をすべてのプリンタ上で取り消します。

```
# cancel -u george
request "asteroid-3" cancelled
request "luna-8" cancelled
```

印刷要求の移動

プリンタの使用方法を変更する計画がある場合や、プリンタの使用を中止する場合は、それ以後の印刷要求を拒否するように LP 印刷サービスを設定し、現在待ち行列に入っている要求があればプリンタに移動するか取り消す必要があります。`lpmove(1M)` コマンドを使用すると、個々の印刷要求またはすべての印刷要求を別のローカルプリンタに移動できます。

要求 ID は印刷要求を移動しても変更されないので、ユーザーは引き続き各自の要求を調べることができます。新しく指定したプリンタでは満たせない要件 (ファイル内容形式やフォームなど) を持つ印刷要求は移動できません。この種の印刷要求は取り消さなければなりません。

▼ 印刷要求を別のプリンタに移動する方法

あるプリンタから別のプリンタにすべての要求を移動する場合は、要求 ID がわからなくてもかまいません。ただし移動する前に、影響を受ける印刷要求の数を調べておくとよいでしょう。

1. プリンタサーバーにスーパーユーザーまたは lp としてログインします。
2. **(省略可能) 元のプリンタ上で印刷要求の要求 ID をチェックします。**

```
# lpstat -o printer-name1
```

3. **(省略可能) 宛先プリンタが印刷要求を受け付けているかどうかをチェックします。**

```
# lpstat -p printer-name2
```

-p *printer-name2*　印刷要求の移動先となるプリンタ名

4. 元のプリンタから宛先プリンタにすべての印刷要求を移動します。

```
# lpmove printer-name1 printer-name2
```

printer-name1　すべての印刷要求の移動元となるプリンタ名

printer-name2　すべての印刷要求の移動先となるプリンタ名

一部の要求を宛先プリンタ上で印刷できない場合は、元のプリンタの待ち行列内に残ります。要求 ID を使用すると、lpmove コマンドで特定の印刷要求を別のプリンタに移動することもできます。

5. 元のプリンタ上で印刷要求の受け付けを開始します。

すべての印刷要求を別のプリンタに移動すると、lpmove コマンドはそのプリンタへの印刷要求の受け付けを自動的に停止します。そのプリンタへの新しい印刷要求の受け付けを開始したい場合は、この手順が必要です。

```
# accept printer-name1
```

6. 次のコマンドを使用して、元のプリンタの待ち行列に残っている印刷要求をチェックします。

```
$ lpq -P printer-name1
```

次のコマンドを使用して、すべての指定した印刷要求が宛先のプリンタ待ち行列に移動していることを確認します。

```
$ lpq -P printer-name2
```

例 — 印刷要求を別のプリンタに移動する

次の例では、lpmove コマンドにより印刷要求をプリンタ luna からプリンタ terra へ移動し、accept コマンドによりプリンタ luna に対し印刷要求の受け付けを再開するよう通知します。

```
# lpmove luna terra
# accept luna
```

印刷要求の優先順位の変更

印刷要求を出し終わったら、その優先順位をプリンタサーバーの待ち行列内で次のように変更できます。

- 印刷が終了していない場合は、その印刷要求を停止できます。要求を保留状態にして停止すると、現在印刷されている場合は、再開するまで印刷されません。他の印刷要求は、停止中の要求よりも先に印刷されます。
- 任意の印刷要求を待ち行列の先頭に移動できます。その場合、先頭の印刷要求は次に印刷されます。ジョブの印刷を即座に開始したい場合は、現在印刷中のジョブを停止して中断できます。
- 引き続き印刷したいジョブの優先順位を変更して、待ち行列内で優先順位が低い要求の前と、優先順位が同じか高い要求の後の間に移動できます。

▼ 印刷要求の優先順位を変更する方法

1. 印刷要求を停止中のプリンタサーバーにスーパーユーザーまたは lp としてログインします。

2. lpstat コマンドを使用して、優先順位を変更したい印刷要求の要求 **ID** を判別します。

   詳細は、118ページの「印刷要求の状態をチェックする方法」を参照してください。

3. lp コマンドを使用して、印刷要求の優先順位を変更します。

```
# lp -i request-id -H change-priority
```

-i *request-id*	変更したい印刷要求の要求 ID。 このコマンドで複数の要求 ID を指定できる。空白またはコンマを使用する。空白を使用する場合は、要求 ID のリストを引用符で囲む
-H *change-priority*	印刷要求の優先順位を変更する方法として、hold、resume、immediate のいずれかを指定する。 *change-priority* に有効な値については、表 5–6 を参照

また、lp の –q コマンドを使用すると、指定した印刷要求の優先順位を変更できます。優先順位は、最上位の 0 から最下位の 39 までの間で変更できます。

例 — 印刷要求の優先順位を変更する

次の例では、コマンドは要求 ID が asteroid-79 の印刷要求を優先順位 1 に変更します。

```
# lp -i asteroid-79 -q 1
```

第 6 章

文字セット、フィルタ、フォーム、フォントの管理手順

この章では、文字セット、印刷フィルタ、フォーム、およびフォントについての基本事項と、設定して管理する手順について説明します

この章で説明する手順は次のとおりです。

- 157ページの「フォームへのプリンタアクセスを制限する方法」
- 161ページの「ダウンロードされた PostScript フォントをインストールする方法」
- 162ページの「ホスト常駐 PostScript フォントをインストールする方法」

印刷については、第 2 章を参照してください。

文字セットの管理

プリンタでテキストを各種フォント書体で印刷する方法は、それぞれ異なります。たとえば、PostScript プリンタは、テキストをグラフィックスとして処理します。これらのプリンタは、複数のフォントを使用してテキストを生成し、ページ上の任意の位置、サイズ、または方向にテキストを配置できます。その他の形式のプリンタは、印字ホイール、フォントカートリッジ、プログラムされた選択可能な文字セットのいずれかを使用するため、フォントの種類と大きさには制限があります。通常、1 つのプリンタ形式には 1 つの印刷方法が適用されます。

必要に応じてプリンタにフォントを装着する必要があるという点で、LP 印刷サービスでは印字ホイールとフォントカートリッジを同様に扱うことができます。ホイールまたはカートリッジを物理的に装着する必要がある文字セットを、「プリンタに装着する文字セット」といいます。物理的に装着する必要がなく、プリンタにあらかじめプログラムされていて、印刷要求によって選択可能な文字セットを、「プリンタに組み込みの文字セット」といいます。

PostScript 以外のプリンタを設定する場合は、ユーザーが利用可能な、印字ホイールまたは選択可能な文字セットを LP 印刷サービスに指定する必要があります。ユーザーが印刷要求を出すときに、`lp -S` コマンドを使用すると、ジョブの印刷に使用する印字ホイールまたは選択可能な文字セットを指定できます。ユーザーは、すでに定義してある名前でフォントを参照するだけなので、実際に使用される文字セットの種類を知る必要はありません。たとえば、印刷ホイールを `gothic` と定義したとします。この `gothic` 印字ホイールを要求するには、`lp -S gothic` と入力します。

選択可能な文字セット

プリンタによってサポートされる選択可能文字セットは、そのプリンタの `terminfo` エントリに表示されています。たとえば、`ln03` プリンタのエントリは、`/usr/share/lib/terminfo/l/ln03` です。`tput` コマンドを使用し

て、terminfo データベースの任意のプリンタタイプの選択可能文字セットの名前を選択できます。tput コマンドの構文は次のとおりです。

```
tput -T printer-type csn
```

cs*n* オプションは文字セット番号 (character set number) の省略形です。番号は、プリンタが初期化された後に常に設定されるデフォルトの文字セット番号である 0 で始まります。その他の文字セット名を表示するには、–0 の代わりに –1、–2、–3 などを使用してコマンドを繰り返してください。選択可能文字セットごとに、terminfo 名 (たとえば usascii、english、finnish など) が返されます。

通常、terminfo 文字セット名は、プリンタのマニュアルで使用されている文字セット名となるべく一致させてください。同じ文字セット名を使用しないメーカーもあるため、terminfo 名はプリンタタイプごとに異なる場合があります。

LP 印刷サービスを使用して選択可能文字セットを登録する必要はありません。ただし、より意味のある名前または別名を与えることができます。

注 - プリンタで使用できる選択可能文字セットを指定しない場合、LP 印刷サービスは、プリンタが任意の文字セット名 (cs0、cs1、cs2 など) またはプリンタが認識する terminfo 名を受け付けることができると仮定します。

---

lpstat -p -l コマンドを使用して、プリンタサーバーに接続されているプリンタごとに、定義されている選択可能文字セット名を表示できます。

注 - PostScript のフォントは、terminfo データベースのエントリではなく PostScript フィルタによって制御されるため、lpstat -p -l コマンドを使用しても PostScript プリンタ用の文字セットは表示されません。PostScript フォントの管理方法については、158ページの「フォントの管理」を参照してください。

プリンタに装着する文字セット

別の文字セットを使用するもう一つの方法は、物理的にプリンタに装着できる取り外し可能な印字ホイールまたはフォントカートリッジを使用することです。

プリンタに装着する文字セットを管理するには、LP 印刷サービスに、使用したい印字ホイール名と、プリンタが異なる印字ホイールを必要とするときの警告方法を指定します。次に、ユーザーが lp -S コマンドを使用して特定の文字セットを要求すると、スケジューラは印字ホイールを装着するよう警告を送信し、印刷要求が印刷

待ち行列に入れられます。正しい印字ホイールを装着して、印字ホイールを装着したことを LP 印刷サービスに指示すると、ジョブが印刷されます。詳細は、136ページの「印字ホイールまたはフォントカートリッジを取り外すまたは装着する方法」を参照してください。

1 台のプリンタに対して複数の印字ホイールやカートリッジを指定しなければ、LP 印刷サービスは、プリンタが 1 つの固定印字ホイールまたはカートリッジしか持っておらず、ユーザーはプリンタを使用する際に特殊な印字ホイールやカートリッジを指定できないと見なします。

選択可能文字セットとは違って、印字ホイールまたはカートリッジ用に選択する名前は、`terminfo` データベースのエントリとは関係がありません。印字ホイール名またはカートリッジ名は、ユーザーが LP 印刷サービスと通信を行うためにだけ使用されます。

ただし、印字ホイールまたはカートリッジ用に選択する名前は、ユーザーがわかりやすいものにしてください。その名前がフォントの書体を表すようにしてください。さらに、その名前は、同じ種類の印字ホイールやカートリッジ、または選択可能文字セットを持つプリンタの場合には、同じ名前にします。それによって、ユーザーは、どのプリンタ、印字ホイール、カートリッジ、選択可能文字セットを使用するかに関係なく、フォントの書体 (文字セット) を指定できます。

システム管理者とプリンタユーザーは、印字ホイールまたはカートリッジに同じ名前を使用してください。そうしないと、ユーザーが指定する文字セットと管理者が装着するものが異なる可能性があります。

印字ホイールの確認

印字ホイールを確認する手順は、フォームを確認する手順と似ています。一部のプリンタは (通常、文字ベースの印字を行うプリンタ)、特定のフォントや文字セットを提供する印字ホイールや印字カートリッジのような、取り外し可能な印字ヘッドを持っています。ユーザーは名前の付いた文字セットを要求できます。その文字セットがない場合、LP 印刷サービスは要求元または管理者に通知します。印刷ジョブは、印字ホイールが変更されるまで、印刷待ち行列に格納されます。

印字ホイールまたはカートリッジの装着の警告

LP 印刷サービスから出す警告を指定するのと同じ方法で、印字ホイールまたはカートリッジを装着する際に出す警告を指定します。警告の概要については、109ページの「障害の通知の設定」を参照してください。

▼ 印字ホイールとフォントカートリッジを定義する方法

1. プリンタサーバーにスーパーユーザーまたは `lp` としてログインします。

2. プリンタで使用できる印字ホイールまたはフォントカートリッジを定義します。

```
print-server# lpadmin -p printer-name -S hard-charset1[,hard-charset2...]
```

`-p printer-name`	プリンタで使用できる印字ホイールまたはフォントカートリッジを定義する
`-s hard-charset`	印字ホイールまたはフォントカートリッジを定義するプリンタ名。 印字ホイールまたはフォントカートリッジのプリンタに装着する文字セット名。このコマンドで複数のプリンタに装着する文字セット名を指定できる。文字セット名を区切るには空白またはコンマを使用する。空白を使用する場合は、文字セット名のリストを引用符で囲む。 ユーザーにとって意味のある名前を定義して、ユーザーに通知する

印字ホイールまたはフォントカートリッジの定義が、プリンタサーバーの `/etc/lp/printers/`*printer-name*`/configuration` ファイルに追加されます。

3. プリンタサーバーの印刷クライアントであるシステムにスーパーユーザーまたは `lp` としてログインします。

4. 印刷クライアントに対して同じ印字ホイールまたはフォントカートリッジを定義します。

```
print-client# lpadmin -p printer-name -S hard-charset1[,hard-charset2...]
```

このコマンドの変数は、135ページの手順 2 と同じです。

印字ホイールまたはフォントカートリッジの定義が、印刷クライアントの /etc/lp/printers/*printer-name*/configuration ファイルに追加されます。

5. 印字ホイールまたはフォントカートリッジを使用する必要がある印刷クライアントごとに、135ページの手順 3 と 135ページの手順 4 を繰り返します。

6. プリンタサーバーと印刷クライアント上で、次のコマンド出力の「Character sets」見出しの後にある情報を確認します。

```
$ lpstat -p printer-name -l
```

例 — 印字ホイールを定義する

次の例は、印刷クライアント asteroid のプリンタ luna 上で印字ホイール pica を定義するコマンドを示しています。

```
asteroid# lpadmin -p luna -S pica
```

▼ 印字ホイールまたはフォントカートリッジを取り外すまたは装着する方法

1. プリンタサーバーにスーパーユーザーまたは lp としてログインします。

2. lpadmin コマンドを使用して、プリンタ内の印字ホイールまたはフォントカートリッジを取り外します。

```
# lpadmin -p printer-name -M -S none
```

-p *printer-name*	印字ホイールまたはフォントカートリッジを取り外すプリンタ名
-M -S none	現在の印字ホイールまたはフォントカートリッジを取り外すように指定する

現在の印字ホイールまたはフォントカートリッジが、プリンタサーバーの /etc/lp/printers/*printer-name*/configuration ファイルから削除されます。

3. 印字ホイールまたはフォントカートリッジをプリンタから削除します。

4. プリンタに新しい印字ホイールまたはフォントカートリッジを入れます。

5. lpadmin コマンドを使用して、新しい印字ホイールまたはフォントカートリッジを装着します。

```
# lpadmin -p printer-name -M -S hard-charset
```

-p *printer-name*	印字ホイールまたはフォントカートリッジを装着するプリンタ名
-M -S *hard-charset*	装着したい印字ホイールまたはフォントカートリッジのプリンタに装着する文字セット名

印字ホイールまたはフォントカートリッジが、プリンタサーバーの /etc/lp/printers/*printer-name*/configuration ファイルに追加されます。装着された印字ホイールまたはフォントカートリッジは、取り外されるか、新しいものが装着されるまで使用可能です。

6. 次のコマンドの出力の中で、「Print wheels」または「Character set」の見だしの下にある情報をチェックします。印刷ホイール名または文字セット名と注意「mounted」が表示されます。

```
$ lpstat -p printer-name -l
```

例 — 印字ホイールを取り外すまたは装着する

次の例は、プリンタ luna から現在の印字ホイールを取り外し、印字ホイール pica を装着するコマンドを示します。

```
# lpadmin -p luna -M -S none
# lpadmin -p luna -M -S pica
```

▼ 印字ホイールまたはフォントカートリッジの装着を促す警告を設定する方法

1. プリンタサーバーにスーパーユーザーまたは lp としてログインします。

2. lpadmin(1M) コマンドを使用して、印字ホイールまたはフォントカートリッジの装着を促す警告を設定します。

```
# lpadmin -S hard-charset -A alert [-Q requests] [-W minutes]
```

-S *hard-charset*	警告を設定したい印字ホイールまたはフォントカートリッジのプリンタに装着する文字セット名
-A *alert*	印字ホイールまたはフォントカートリッジが要求されたときに出される警告の種類を指定する。*alert* に有効な値については、表 5–3 を参照。有効な値は mail、write、quiet など。 mail または write を指定すると、あらかじめ定義された警告メッセージが表示される。この警告メッセージは、指定した印字ホイールまたはフォントカートリッジの装着を促すもので、それを使用するように設定されている 1 つ以上のプリンタ名が含まれる
-Q *requests*	警告が出される前に、印字ホイールまたはフォントカートリッジが、待ち行列に入っていなければならない印刷要求の数を指定する。このオプションを指定しなければ、待ち行列に印刷要求が 1 つ入っただけで警告が出される
-W *minutes*	警告が出される間隔 (分単位) を指定する。このオプションを指定しなければ、警告は一度だけ送られる

警告は、プリンタサーバーの /etc/lp/pwheels/*charset-name*/alert.sh ファイルに追加されます。

3. 次のコマンドの出力をチェックして、印字ホイールまたはフォントカートリッジの装着を促す警告が追加されているかどうかを確認します。

```
# lpadmin -S hard-charset -A list
```

あるいは、警告を出すために印刷要求に低い番号を設定した場合、最小限の要求を満たすために十分な印刷要求を出し、印字ホイールまたはフォントカートリッジの装着を促す警告を受け取ることを確認します。

例 — 印字ホイールまたはフォントカートリッジの装着を促す警告を設定する

次の例は、印刷待ち行列に elite 印字ホイールに対する 10 の印刷要求があるときに、elite に関して 5 分間隔で電子メールで警告が送られるように設定するコマンドを示しています。

```
# lpadmin -S elite -A mail -Q 10 -W 5
```

次の例は、印刷待ち行列に finnish フォントカートリッジに対する 5 つの印刷要求があるときに、finnish に関して 1 分間隔で電子メールで警告が送られるように設定するコマンドを示しています。

```
# lpadmin -S finnish -A mail -Q 5 -W 1
```

次の例は、印刷待ち行列に elite 印字ホイールに対する 5 つの印刷要があるときに、elite に関して 10 分間隔でコンソールウィンドウに警告が送られるように設定するコマンドを示しています。

```
# lpadmin -S elite -A write -Q 5 -W 10
```

次の例は、elite 印字ホイールに警告が送られないように設定するコマンドを示します。

```
# lpadmin -S elite -A none
```

▼ 選択可能文字セットの別名を設定する方法

注 - 選択可能文字セットの terminfo(4) 名が正しい場合は、この手順を実行する必要はありません。terminfo データベースの使用方法については、168ページの「サポートされていないプリンタの terminfo エントリを追加する」を参照してください。

---

1. プリンタサーバーにスーパーユーザーまたは lp としてログインします。

2. tput(1) コマンドを使用して、指定したプリンタタイプの選択可能文字セット名を表示します。

   ```
   # tput -T printer-type csn
   ```

-T *printer-type*	terminfo データベースに入っているプリンタタイプ。terminfo データベースのエントリについては、60ページの「プリンタタイプ」を参照
n	指定したプリンタタイプの選択可能文字セットを表す番号 (0、1、2、3、4、5 など)。プロンプト記号に続いて選択可能文字セット名が表示される。たとえば、cs1 と指定すると、english# と表示される

3. 選択可能文字セットの別名を設定します。

```
# lpadmin -p printer-name -S select-charset1=alias1[,select-charset2=alias2...]
```

-p *printer-name*	選択可能文字セットの別名を設定するプリンタ名
-S *select-charset*	別名を設定する選択可能文字セット名。この名前は、139ページの手順 2 で検索できる
alias	指定した選択可能文字セットの別名。選択可能文字セット名の他に、この別名を使用できる。 このコマンドで複数の別名を設定できる。別名を区切るには空白またはコンマを使用する。空白を使用する場合は、別名のリストを引用符で囲む

別名は、プリンタサーバーの /etc/lp/printers/*printer-name*/configuration ファイルに追加されます。

4. プリンタサーバーの印刷クライアントであるシステムにスーパーユーザーまたは lp としてログインします。

5. 選択可能文字セットの別名を設定します。

```
# lpadmin -p printer-name -S select-charset1=alias1[,select-charset2=alias2...]
```

このコマンドの変数は、140ページの手順 3 と同じです。

別名は印刷クライアントの /etc/lp/printers/*printer-name*/configuration ファイルに追加されます。

6. 別名を使用する必要が考えられる印刷クライアントごとに、140ページの手順 4 と 140ページの手順 5 を繰り返します。

7. プリンタサーバーと印刷クライアント上で、次のコマンドの出力の中に選択可能文字セットの別名のリストがあることを確認します。

```
$ lpstat -p printer-name -l
```

   または、選択可能文字セットに別名を使用する印刷要求を出して、出力をチェックします。

例 — 選択可能文字セットの別名を設定する

次の例は、選択可能文字セット名を表示し、ln03 プリンタタイプのプリンタ luna 上の usascii 選択可能文字セットの別名として text を指定するコマンドを示しています。

```
# tput -T ln03 cs0
usascii# tput -T ln03 cs1
english# tput -T ln03 csn2
finnish# tput -T ln03 csn3
japanese# tput -T ln03 cs4
norwegian#
# lpadmin -p luna -S usascii=text
```

印刷フィルタの管理

「印刷フィルタ」とは、ファイルの内容形式を出力先プリンタが受け付けられる内容形式に変換するプログラムのことです。印刷サービスはフィルタを使用して、次の機能を提供します。

- ファイルを特定タイプのプリンタで正しく印刷できるように、1つのデータ形式から別のデータ形式に変換する。
- 両面印刷、横方向印刷、ドラフト印刷、または高品質印刷などの特別な印刷モードを処理する。

- プリンタの障害を検出して LP 印刷サービスに通知する。その結果、印刷サービスはユーザーとシステム管理者に警告を出すことができる。

すべての印刷フィルタが上記のすべての機能を実行できるわけではありません。各機能はプリンタに固有なので、別々に実装できます。

LP 印刷サービスは、表 6–1 に示す PostScript フィルタを提供します。これらのフィルタプログラムは、`/usr/lib/lp/postscript` ディレクトリに入っています。通常、PostScript 印刷を行う場合は、プリンタサーバーの設定時にフィルタプログラムをインストールする以外に何も行う必要はありません。Solaris プリンタマネージャが提供されるフィルタを自動的に使用可能にします。ただし、他のプリンタを管理する場合は、それらのプリンタの印刷フィルタを管理する必要がある場合があります。

印刷フィルタの作成

新しい印刷フィルタを作成するには、印刷フィルタプログラムを書き、印刷フィルタの定義を作成しなければなりません。フィルタには、入力形式、出力形式、フィルタ内でコマンド行引数を処理するための言語を提供する複雑なオプションが含まれます。説明と手順については、178ページの「新しい印刷フィルタの作成」を参照してください。

印刷フィルタの追加、変更、削除、および復元

印刷フィルタは、プリンタサーバーだけで追加、変更、または削除できます。

`lpfilter(1M)` コマンドを使用して、利用できるフィルタのリストを管理します。フィルタに関するシステム情報は、`/etc/lp/filter.table` ファイルに格納されます。`lpfilter` コマンドは、テーブルに書き出すフィルタに関する情報を、フィルタ記述子ファイルから取得します。提供されているフィルタ記述子ファイル (PostScript のみ) は、`/etc/lp/fd` ディレクトリに入っています。実際のフィルタプログラムは、`/usr/lib/lp` の下にあります。

LP 印刷サービスでは、定義できる印刷フィルタの数に制限はありません。使用しないフィルタを削除して LP 印刷サービスによる処理を減らすことができます。(その場合は、LP はすべてのフィルタを検査して特定の印刷要求に使用するフィルタを 1 つ見つけます。) 確信が持てない場合は、フィルタを削除しないでください。

フィルタを追加、変更、または削除すると、LP 印刷サービスによって提供されている元のフィルタの一部を上書きしたり、削除したりしてしまう可能性があります。必要に応じて元のフィルタを復元し、追加したフィルタを削除できます。

SunOS ソフトウェアには、PostScript フィルタのデフォルトセットが組み込まれています。デフォルトセットは、Solaris プリンタマネージャによってプリンタサーバーに自動的に追加されます。SunOS 4.1 で使用されていた TranScript フィルタは、SunOS 5.8 にも相当するものがある場合とない場合があります。表 6–1 は、デフォルトの PostScript フィルタと、該当する TranScript フィルタが存在する場合はそのフィルタ名を示しています。

表 6–1 デフォルトの PostScript フィルタ

フィルタ	動作	相当する **TranScript**
`download`	ダウンロードフォント	
`dpost`	`ditroff` から PostScript へ	`psdit`
`postdaisy`	`daisy` から PostScript へ	
`postdmd`	`dmd` から PostScript へ	
`postio`	PostScript プリンタへのシリアルインタフェース	`pscomm`
`postior`	プリンタとの通信	
`postmd`	マトリックス型グレースケールから PostScript へ	
`postplot`	`plot` から PostScript へ	`psplot`
`postprint`	`simple` から PostScript へ	`enscript`
`postreverse`	ページの反転または選択	`psrev`
`posttek`	TEK4014 から PostScript へ	`ps4014`

SunOS ソフトウェアには、次のフィルタは組み込まれていません。

- TEX
- oscat (NeWSprint™ opost)
- Enscript

Enscript の代わりに postreverse、postprint、postio、dpost の各フィルタが組み込まれています。

Solaris プリンタマネージャは、プリンタサーバーにデフォルトの PostScript フィルタを追加します。これらのフィルタでは処理できない印刷ニーズがある場合は、カスタム印刷フィルタの作成方法については、189ページの「新しい印刷フィルタを作成する方法」を参照してください。

▼ 印刷フィルタを追加する方法

1. プリンタサーバーにスーパーユーザーまたは lp としてログインします。
2. lpfilter コマンドを使用し、印刷フィルタの定義に基づく印刷フィルタを追加します。

```
# lpfilter -f filter-name -F filter-def
```

-f *filter-name*	印刷フィルタ用に選択する名前
-F *filter-def*	印刷フィルタの定義名

印刷フィルタは、プリンタサーバーの /etc/lp/filter.table ファイルに追加されます。

3. 次のコマンドの出力の中の印刷フィルタについての情報をチェックして、印刷フィルタが追加されているか確認します。

```
# lpfilter -f filter-name -l
```

例 — 印刷フィルタを追加する

次の例は、daisytroff.fd 印刷フィルタ定義を持つ daisytroff 印刷フィルタを追加するコマンドを示しています。

```
# lpfilter -f daisytroff -F /etc/lp/fd/daisytroff.fd
```

▼ 印刷フィルタを削除する方法

1. プリンタサーバーにスーパーユーザーまたは lp としてログインします。
2. lpfilter コマンドを使用して印刷フィルタを削除します。

```
# lpfilter -f filter-name -x
```

-f *filter-name*　削除する印刷フィルタ名

-x　指定したフィルタを削除する

印刷フィルタが、プリンタサーバーの /etc/lp/filter.table ファイルから削除されます。

3. 次のコマンドを使用して、フィルタが削除されていることを確認します。指定した名前のフィルタがないというエラーメッセージが表示されます。

```
# lpfilter -f filter-name -l
```

例 — 印刷フィルタを削除する

次の例は、daisytroff 印刷フィルタを削除するコマンドを示しています。

```
# lpfilter -f daisytroff -x
```

▼ 印刷フィルタに関する情報を表示する方法

1. プリンタサーバーにスーパーユーザーまたは lp としてログインします。
2. lpfilter コマンドを使用して、印刷フィルタに関する情報を要求します。

```
# lpfilter -f filter-name -l
```

`-f` *filter-name*	情報を表示したい印刷フィルタ。利用できるすべての印刷フィルタに関する情報を表示するには、*filter-name* に `all` を指定する
`-l`	指定したフィルタに関する情報を表示する

指定した 1 つ以上の印刷フィルタに関する情報が表示されます。

例 — 印刷フィルタに関する情報を表示する

次の例は、`postdaisy` 印刷フィルタに関する情報を要求するコマンドと、それに応答して表示される情報を示しています。

```
# lpfilter -f postdaisy -l
Input types: daisy
Output types: postscript
Printer types: any
Printers: any
Filter type: slow
Command: /usr/lib/lp/postscript/postdaisy
Options: PAGES * = -o*
Options: COPIES * = -c*
Options: MODES group = -n2
Options: MODES group\=\([2-9]\) = -n\1
Options: MODES portrait = -pp
Options: MODES landscape = -pl
Options: MODES x\=\(\-*[\.0-9]*\) = -x\1
Options: MODES y\=\(\-*[\.0-9]*\) = -y\1
Options: MODES magnify\=\([\.0-9]*\) = -m\1
```

次の例は、`daisytroff` フィルタに関する情報をファイルにリダイレクトするコマンドを示しています (そのフィルタのフィルタ定義が作成されます)。これは、うっかりフィルタ定義を削除してしまった場合に便利です。

```
# lpfilter -f daisytroff -l > daisytroff.fd
```

次の例は、システムに追加されたすべての印刷フィルタを表示するコマンドと、それに応答して表示される情報を示しています。

```
# lpfilter -f all -l | grep Filter
(Filter "download")
Filter type: fast
(Filter "postio")
Filter type: fast
(Filter "postior")
```

(続く)

```
Filter type: fast
(Filter "postreverse")
Filter type: slow
```

フォームの管理

「フォーム」は、あらかじめ決められている形式に従って情報が印刷されている用紙です。普通紙と違って、通常、フォームにはテキストまたはグラフィックスが前もって印刷されています。フォームの一般的な例としては、企業のレターヘッド、送り状、小切手、領収書、ラベルなどがあります。

「フォーム」という用語には 2 つの意味があります。一つは物理的な媒体 (用紙) という意味で、もう一つは LP 印刷サービスの形式を定義するソフトウェアという意味です。

LP 印刷サービスを使用すると、フォームの使用方法を制御できます。この節では、フォームの追加、変更、削除、取り付けを行う方法と、フォームへのアクセスを制御する方法について説明します。

フォームの追加、変更、または削除

フォームを追加するときには、LP 印刷サービスに指示を与えて、そのフォームを利用可能なフォームのリストに加えます。また、フォームの記述と定義に必要な情報を与えなければなりません。フォームを追加するときに、その定義を入力できますが、はじめに定義を作成しておいて、ファイルに保存しておくことをお勧めします。ファイルを編集すれば、フォーム定義を変更できます。

注 - LP 印刷サービスでは、フォーム定義は提供されません。

フォームを変更するには、異なる定義を持つフォームを追加し直さなければなりません。

LP 印刷サービスでは、定義できるフォームの数に制限はありません。ただし、不要なフォームは削除してください。不要なフォームがあると、印刷サービスに余計な負担をかける可能性があります。

フォームの取り付け

フォームを印刷するには、プリンタに紙を装着し、コマンドを使用してフォームを「取り付け」、これによって、プリンタに送られる印刷要求がこのフォーム定義を使用して印刷されることを LP 印刷サービスに通知します。複数のフォームを使用する場合など、1 台のプリンタで異なる種類の印刷を行う場合には、次の作業を実行します。

- 紙を装着してフォームを取り付ける前にプリンタを使用不可にします。
- フォームの準備ができたらプリンタを再び使用可能にします。そうしないと、LP 印刷サービスはプリンタでそのフォームを必要としないファイルを印刷し続けます。

フォームを取り付けるときには、正しく揃っているかどうかを確認してください。揃え方がフォームに対して定義されている場合は、揃え方が正しくなるようにプリンタを調整し終わるまで、フォームを取り付けた後でパターン印刷を繰り返すように要求できます。

プリンタに取り付けられているフォームの使用を変更または中止したい場合は、フォームを取り外して LP 印刷サービスに通知しなければなりません。

フォームの確認

LP 印刷サービスにより、各プリンタにどのフォームが装着されているかを確認できます。また、フォームに印刷するときに必要な記述がなければ、LP 印刷サービスが通知します。フォームの記述を作成したり、各プリンタにフォームを装着したり取り外したりするのはシステム管理者の責任です。この作業はプリンタの設定時か、LP 印刷サービスからの警告時に行います。

ユーザーは印刷ジョブを印刷したいフォームを指定します。管理者は特定のフォームを装着して、フォームが使用できる状態にあり、どのプリンタに装着されているかを LP 印刷サービスに伝えます。ユーザーは特定のフォームを指定することによって印刷要求を出すことができます。LP 印刷サービスが要求を受け取ると、フォームの装着要求を警告メッセージとして管理者に送信します。

フォームの取り付けに関する警告の定義

LP 印刷サービスから他の警告を要求するのと同じ方法で、フォームの取り付けに関する警告を要求します。警告の概要については、109ページの「障害の通知の設定」を参照してください。

フォームのチェック

LP 印刷サービスに対してフォームを定義し終わったら、チェックしたい情報に応じて 2 つのコマンドのどちらかでフォームの定義をチェックできます。

- `lpforms(1M)` コマンドを使用してフォームの属性を表示します。また、コマンドの出力をファイルにリダイレクトして将来の参照に備えて保存できます。
- `lpstat` コマンドを使用してフォームの現在の状態を表示します。内容を保護するため、位置揃えパターンは表示されません。

既存のフォーム名がわからない場合は、`/etc/lp/forms` ディレクトリの内容の一覧を表示して調べることができます。

フォームへのアクセスの制限

どのプリンタやユーザーが、ネットワーク上で利用可能な一部またはすべてのフォームを使用できるかを制御できます。たとえば、経理部に属するユーザーだけが小切手のフォームを印刷できるようにしたい場合があります。また、特定のプリンタだけで利用できる小切手のフォームが必要な場合もあります。

フォームへのユーザーアクセスを制限するには、156ページの「フォームへのユーザーアクセスを制限する方法」、フォームへのプリンタアクセスを制限するには、157ページの「フォームへのプリンタアクセスを制限する方法」を参照してください。

▼ フォームを追加する方法

1. **プリンタサーバーにスーパーユーザーまたは `lp` としてログインします。**

2. **`lpforms` コマンドを使用し、フォーム定義に基づくフォームを追加します。**

```
# lpforms -f form-name -F /etc/lp/forms/form
```

-f *form-name*	フォーム用に選択した名前
-F /etc/lp/forms/*form*	フォーム定義名

フォームは、プリンタサーバーの /etc/lp/forms/*form-name*/describe ファイルに追加されます。

3. 次のコマンドの出力に、フォームについての情報があるかをチェックして、フォームが追加されているか確認します。

```
# lpforms -f form-name -l
```

例 — フォームを追加する

次の例は、medical.fmd フォーム定義を使用する medical フォームを追加するコマンドを示します。

```
# lpforms -f medical -F /etc/lp/forms/medical.fmd
```

注 - フォームを使用する前に、そのフォームへのアクセスを 1 つ以上のプリンタに与えておかなければなりません。157ページの「フォームへのプリンタアクセスを制限する方法」を参照してください。

▼ フォームを削除する方法

1. プリンタサーバーにスーパーユーザーまたは lp としてログインします。
2. lpforms コマンドを使用してフォームを削除します。

```
# lpforms -f form-name -x
```

-f *form-name*	削除するフォーム名
-x	指定したフォームを削除する

フォームが /etc/lp/forms/*form-name* ファイルから削除されます。

3. 次のコマンドを使用して、フォームが削除されたか確認します。指定したフォーム名がないことを示すエラーメッセージが表示されます。

```
# lpforms -f form-name -l
```

例 — フォームを削除する

次の例は、medical フォームを削除するコマンドを示しています。

```
# lpforms -f medical -x
```

▼ フォームを取り外し、装着する方法

1. プリンタサーバーにスーパーユーザーまたは lp としてログインします。

2. reject コマンドを使用して、現在のフォームを取り外そうとしているプリンタ上で印刷要求を停止します。

```
# reject printer-name
```

printer-name　　フォームを取り外すプリンタ名

新しい印刷要求 (フォームを必要としない場合もある) は、そのプリンタの待ち行列に入れられなくなります。

3. lpadmin コマンドを使用して、現在のフォームを取り外します。

```
# lpadmin -p printer-name -M -f none
```

このコマンドの変数 *printer-name* は、151ページの手順 2 と同じです。

現在のフォームは、プリンタサーバーの /etc/lp/printers/*printer-name*/configuration ファイルから削除されます。

4. プリンタからフォーム用紙を取り外します。

5. 次の印刷要求のためにフォーム用紙を装着します。

6. lpadmin コマンドを使用してフォームを装着します。

```
# lpadmin -p printer-name -M -f form-name [-a -o filebreak]
```

-p *printer-name*	フォームを装着するプリンタ名
-M -f *form-name*	装着するフォーム名
–a -o filebreak	省略可能。フォームに位置揃えパターンが定義されている場合は、そのコピーを印刷できるようにする

指定したフォームは、プリンタサーバーの /etc/lp/printers/*printer-name*/configuration ファイルに追加されます。

7. プリンタ上で印刷要求の受け付けを開始します。

```
# accept printer-name
```

これで、プリンタは新しく装着したフォームで印刷する準備ができました。

8. 次のコマンド出力の「Form mounted」見だしの下にあるフォーム名をチェックし、フォームが装着されていることを確認します。

```
$ lpstat -p printer-name -l
```

あるいは、新しいフォームを必要とする印刷要求を出して、プリンタの出力をチェックします。

例 — フォームを取り外し、装着する

次の例は、現在装着されているフォームをプリンタ luna から取り外すプロセスを示しています。

```
# reject luna
destination "luna" will no longer accept requests
# lpadmin -p luna -M f none
# accept luna
destination "luna" now accepting requests
```

次の例は、medical フォームをプリンタ luna 上に装着するプロセスを示しています。

```
# reject luna
destination "luna" will no longer accept requests
# lpadmin -p luna -M f medical -a -o filebreak
# accept luna
destination "luna" now accepting requests
```

▼ フォームの装着に関する警告を設定する方法

1. プリンタサーバーにスーパーユーザーまたは lp としてログインします。
2. lpadmin コマンドを使用して、フォームの装着に関する要求警告を設定します。

```
# lpforms -f form-name -A alert [-Q requests] [-W minutes]
```

-f *form-name*	要求警告を設定したいフォーム名
-A *alert*	フォームが要求されるときに出す警告の種類を指定する。*alert* に有効な値については、表 5–3 を参照。有効な値は mail、write、または quiet。 mail または write を選択すると、あらかじめ定義された警告メッセージが表示される。この警告メッセージは、指定されたフォームの装着を促すもので、そのフォームを使用するように設定されている 1 つ以上の複数のプリンタ名が含まれる
-Q *requests*	警告が出される前に、フォームが必要な印刷要求がいくつ待ち行列に入っていなければならないかを指定する。このオプションを指定しなければ、印刷要求が待ち行列に 1 つ入っただけで警告が出される
-W *minutes*	警告が出される間隔 (分単位) を指定する。このオプションを指定しなければ、警告は一度だけ送られる

要求警告は、プリンタサーバーの /etc/lp/forms/*form-name*/alert.sh ファイルに追加されます。

3. 次のコマンドの出力をチェックして、そのフォームに関する警告が追加されていることを確認します。

```
# lpforms -f form-name -A list
```

あるいは、警告を出すために印刷要求の低い番号を設定した場合、最小限の要求を満たすために十分な印刷要求を出し、フォームの装着を促す警告を受け取ることを確認します。

例 — フォームの装着に関する警告を設定する

次の例は、印刷待ち行列に letterhead フォームに関する 10 の印刷要求があるときに、letterhead に関して 5 分ごとに電子メールで警告が送られるように設定するコマンドを示します。

```
# lpforms -f letterhead -A mail -Q 10 -W 5
```

次の例は、印刷待ち行列に letterhead フォームに関する 5 の印刷要求があるときに、letterhead に関して 10 分ごとにコンソールウィンドウに警告が送られるように設定するコマンドを示します。

```
# lpforms -f letterhead -A write -Q 5 -W 10
```

次の例は、invoice フォームに関して要求警告が送られないように設定するコマンドを示します。

```
# lpforms -f invoice -A none
```

▼ フォームに関する情報を表示する方法

1. プリンタサーバーにスーパーユーザーまたは lp としてログインします。
2. lpforms コマンドを使用して、フォームに関する情報を要求します。

```
# lpforms -f form-name -l
```

-f *form-name*	情報を表示したいフォーム名。利用できるすべてのフォームに関する情報を表示するには、*form-name* に all を指定する
-l	指定したフォームを表示する

指定した 1 つ以上のフォームに関する情報が表示されます。

例 — フォームに関する情報を表示する

次の例は、medical フォームに関する情報を表示するコマンドを示しています。

```
# lpforms -f medical -l
Page length: 62
Page width: 72
Number of pages: 2
Line pitch: 6
Character pitch: 12
Character set choice: pica
Ribbon color: black
Comment:
Medical claim form
```

次の例は、medical フォームに関する情報をファイルにリダイレクトするコマンドを示しています (このコマンドは、そのフォームのフォーム定義を作成します)。これは、うっかりフォーム定義を削除してしまった場合に便利です。

```
# lpforms -f medical -l > medical.fmd
```

▼ フォームの現在の状態を表示する方法

1. プリンタサーバーにログインします。
2. lpstat(1) コマンドを使用して、フォームの現在の状態に関する情報を要求します。

```
$ lpstat -f form-name
```

–f *form-name*	現在の状態を表示したいフォーム名。すべてのフォームの現在の状態を表示したい場合は、*form-name* に all を指定する

指定した 1 つ以上のフォームの現在の状態に関する情報が表示されます。

例 — フォームの現在の状態を表示する

次の例は、medical フォームの状態を表示しています。

```
$ lpstat -f medical
form medical is available to you
```

▼ フォームへのユーザーアクセスを制限する方法

1. プリンタサーバーにスーパーユーザーまたは lp としてログインします。

2. lpforms コマンドを使用して、フォームへのユーザーアクセスを許可または拒否します。

```
# lpforms -f form-name -u allow:user-list | deny:user-list
```

-f *form-name*	ユーザーアクセスの許可または拒否リストを作成するためのフォーム名
-u allow:*user-list*	ユーザーアクセスの許可リストに追加するユーザー名。複数のユーザーログイン ID を指定する場合は、空白またはコンマで区切る。空白で区切る場合は、ID のリストを引用符で囲む。*user-list* の有効な値については、表 5–5 を参照
-deny:*user-list*	ユーザーアクセス拒否リストに追加するユーザー名。複数のユーザーログイン名を指定する場合は、空白またはコンマで区切る。空白で区切る場合は、ID のリストを引用符で囲む。*user-list* の有効な値については、表 5–5 を参照

プリンタサーバーの次のどちらかのファイルの指定されたフォーム用の許可または拒否のユーザーアクセスリストに、指定した 1 人以上のユーザーが追加されます。

/etc/lp/forms/*form-name*/allow または

/etc/lp/forms/*form-name*/deny

3. lpforms コマンドを使用して、ユーザーアクセスの許可リストと拒否リストを確認します。

```
# lpforms -f form-name -l
```

例 — フォームへのユーザーアクセスを制限する

次の例は、ユーザー nathan と marcia にのみ check フォームへのアクセスを許可するコマンドを示しています。

```
# lpforms -f check -u allow:nathan,marcia
```

次の例は、ユーザー jones と smithによる dental フォームへのアクセスを拒否するコマンドを示しています。

```
# lpforms -f dental -u deny:"jones,smith"
```

▼ フォームへのプリンタアクセスを制限する方法

1. プリンタサーバーにスーパーユーザーまたは lp としてログインします。
2. lpadmin コマンドを使用して、プリンタ上でのフォームの使用を許可または拒否します。

```
# lpadmin -p printer-name -f allow:form-list | deny:form-list
```

-p *printer-name*	フォームの許可または拒否リストを作成するプリンタ名
-f allow:*form-list* | deny:*form-list*	許可または拒否リストに追加されるフォーム名。複数のフォーム名は空白またはコンマで区切る。空白で区切る場合は、フォーム名のリストを引用符で囲む

指定した 1 つ以上のフォームは、プリンタサーバーの次のどちらかのファイルの許可または拒否フォームリストに追加されます。

/etc/lp/printers/*printer-name*/form.allow

/etc/lp/printers/*printer-name*/form.deny

3. 次のコマンドを使用して、許可リストと拒否リストを確認します。

```
# lpstat -p printer-name -l
```

例 — フォームへのプリンタアクセスを制限する

次の例では、プリンタ luna に medical、dental、check の各フォームへのアクセスのみを許可します。

```
# lpadmin -p luna -f allow:medical,dental,check
```

次の例では、プリンタ luna による medical、dental、check の各フォームへのアクセスを拒否します。

```
# lpadmin -p luna -f deny:"medical dental payroll"
```

フォントの管理

レーザプリンタがある場合は、PostScript 用のフォントをインストールして管理する必要がある可能性があります。PostScript フォントをインストールするシステムと、その管理方法を決定する必要がある可能性があります。多くのプリンタの場合、プリンタのインストール作業の一部としてフォントを設定します。

PostScript フォントは、プリンタかプリンタと通信を行うシステムのどちらかに、アウトライン形式で格納されます。文書の印刷時に、PostScript インタプリタは、アウトライン記述から適切な大きさの各文字を必要に応じて生成します。文書に必要なフォントが使用するプリンタに格納されていない場合は、文書が印刷される前にそのフォントをプリンタに転送しなければなりません。この転送処理を「フォントのダウンロード」といいます。

フォントは、次のいくつかの方法で格納または使用されます。

- 「プリンタ常駐フォント」は、プリンタに常時格納されています。これらのフォントは、メーカーによってプリンタの読み取り専用メモリー (ROM) にインストールされている場合があります。プリンタがディスクを備えている場合は、そのディスクにフォントをインストールしなければならないことがあります。ほとんどの PostScript プリンタには、35 の標準フォントが付いています。
- 「常時ダウンロードフォント」は、PostScript の exitserver プログラムを使用してプリンタに転送されます。常時ダウンロードフォントは、プリンタの電源を切るまでプリンタのメモリーに残っています。ダウンロードフォントに割り当てられたメモリーによって、PostScript 印刷要求では利用可能なサーバーのメモリーが減少します。exitserver プログラムを使用するには、プリンタシステム

のパスワードが必要で、プリンタ管理者が使用します。プリンタで出力する大部分の印刷要求に特定のフォントを使用するときは、そのフォントを常時ダウンロードするようにしてください。

- あまり頻繁に使用されないか、特殊な目的で使用されるフォントは、ユーザーのシステムに格納できます。ユーザーは、印刷要求を出すときにこれらのフォントを指定できます。そのフォントは印刷要求に追加されてプリンタに転送されます。印刷要求が処理されると、フォントに割り当てられたメモリー空間は、他の印刷要求が使用できるように解放されます。
- 「ホスト常駐フォント」は、多数のユーザーによって共有されるシステムに格納されます。フォントを格納するシステムは、プリンタサーバーでも印刷クライアントでもかまいません。各ユーザーは印刷する文書のフォントを要求指定できます。この方法は、多数のフォントを利用できるときや、これらのフォントがすべての印刷要求で使用されるとは限らないときに便利です。そのフォントがプリンタサーバーに接続されたプリンタでのみ使用される場合は、プリンタサーバーに格納してください。そのフォントが1つのシステム上で、ネットワーク上の複数のプリンタに要求を依頼する可能性があるユーザーによって使用される場合は、フォントはそのユーザーのシステムに格納してください。

  LP 印刷サービスには、ホスト常駐フォントを管理するための特殊なダウンロード用のフィルタがあります。また、`troff` プログラムで使用するために、多くの PostScript プリンタに搭載された 35 の標準 PostScript フォント用の `troff(1)` のフォント幅テーブルも提供しています。

プリンタ常駐フォントの管理

ほとんどの PostScript プリンタは、プリンタ内蔵の ROM にフォントが搭載されています。プリンタによっては、追加フォントを格納するためのディスクが用意されています。プリンタをインストールするときに、そのプリンタ用のフォントリストにプリンタ常駐フォントを追加してください。プリンタ常駐フォントがわかっていれば、フォントをネットワーク経由で必要以上に転送することがなくなります。各プリンタには搭載されているフォントの独自のリストがあります。これは、次のファイルに入っています。

```
/etc/lp/printers/printer-name/residentfonts
```

プリンタをプリンタサーバーに接続するときには、プリンタサーバー上にあってプリンタにダウンロードできるフォントが、`residentfonts` ファイル内のリストに含まれているかどうかを確認します。

プリンタ常駐フォントのリストが入っているファイルは、vi などのテキストエディタを使用して編集しなければなりません。

ホスト常駐フォントのダウンロード

PostScript の文書がプリンタにダウンロードされていないフォント指定を含んでいるときは、「ダウンロードフィルタ」がこの印刷要求を管理します。ダウンロードフィルタは PostScript の文書作成規則を使用して、ダウンロードするフォントを決定します。

LP 印刷フィルタには、高速フィルタと低速フィルタがあります。高速フィルタは、印刷するファイルをすばやく準備し、フィルタが処理している間にプリンタにアクセスしなければなりません。低速フィルタはファイルの変換に時間がかかり、フィルタが処理している間にプリンタにアクセスする必要はありません。低速フィルタの例には、ASCII ファイルから PostScript ファイルへのフィルタがあります。

ダウンロードフィルタは高速フィルタです。フォントがプリンタサーバー上にある場合は、フォントを自動的にダウンロードします。また、ダウンロードフィルタを使用して、プリンタサーバーにフォントを転送することもできます。そのためには、`lp -y` コマンドを指定して、ダウンロードフィルタを低速フィルタとして呼び出すための新しいフィルタテーブルのエントリを作成します。あるいは、入力タイプを変更して、このフィルタの選択を強制することもできます。

ダウンロードフィルタは、次の 5 つの作業を実行します。

1. PostScript の文書を検索して、要求されているフォントを判別します。これらの要求は、ヘッダコメントの PostScript 構造化コメント `%%DocumentFonts:` *font1 font2 …* で指定されます。
2. プリンタ常駐フォントのリストを検索して、要求されたフォントをダウンロードしなければならないかどうかを判別します。
3. フォントがプリンタ上になければ、ダウンロードフィルタは (マップテーブルから適切なファイル名を読み取って) ホスト常駐フォントのディレクトリを検索し、要求されたフォントが利用可能かどうかを判別します。
4. そのフォントが利用可能であれば、フィルタはそのフォントのファイルを取り出し、印刷するファイルに追加します。
5. フォント定義ファイルとソースファイル (印刷するファイル) を PostScript プリンタに送ります。

ホスト常駐フォントのインストールと管理

フォントによっては、ホストシステムに格納されており、特定の印刷要求に応じてプリンタに転送されるものがあります。管理者は、システム上のすべてのユーザーが PostScript フォントを使用できるように管理する必要があります。そのためには、これらのフォントのインストール方法とインストール場所を知っておかなければなりません。フォントは名前で要求され、ファイルに格納されているので、LP 印刷サービスはフォント名とフォントを定義しているファイル名を対応付けるマップファイルを持っています。ホスト常駐フォントをインストールするときには、マップファイルとフォントリストの両方を更新しなければなりません。

PostScript プリンタで利用できるフォントは、ユーザーが作成した `/usr/share/lib/hostfontdir/`*typeface*`/`*font* ディレクトリに格納されます。この場合、*typeface* は `palatino` や `helvetica` などの名前に置き換えられ、*font* は `bold` や `italic` などの名前に置き換えられます。

▼ ダウンロードされた PostScript フォントをインストールする方法

1. プリンタサーバーまたは印刷クライアント上でスーパーユーザーまたは `lp` としてログインします。

2. `/etc/lp/printers/`***printer-name*** ディレクトリに変更します。

```
# cd /etc/lp/printers/printer-name
```

printer-name	ダウンロードされた PostScript フォントをインストールするプリンタ名

3. `residentfonts` ファイルが存在しない場合は作成します。

```
# touch residentfonts
```

常駐させるダウンロードフォントを初めて追加する場合は、このファイルが存在しないことがあります。

4. `residentfonts` ファイルを編集して、すべてのプリンタ常駐フォントとダウンロードフォントを追加します。

▼ ホスト常駐 PostScript フォントをインストールする方法

1. プリンタサーバーまたは印刷クライアント上でスーパーユーザーまたは `lp` としてログインします。

2. `hostfontdir` ディレクトリが存在しない場合は作成します。

```
# cd /usr/share/lib
# mkdir hostfontdir
# chmod 775 hostfontdir
```

3. 新しい書体のディレクトリが存在しない場合は作成します。

```
# mkdir typeface
```

4. フォントファイルを適切なディレクトリにコピーします。

```
# cp filename /usr/share/lib/hostfontdir/typeface/font
```

5. マップテーブルに組み込むフォント名とファイル名を追加します。

   a. `/usr/share/lib/hostfontdir` ディレクトリに変更します。

   b. **vi** などのテキストエディタを使用して `map` ファイルを編集します。
   テーブルに追加したいフォントごとに 1 行ずつエントリを追加します。エントリには、フォント名、スペース 1 個、フォントが常駐するファイル名の順に入力します。たとえば、次のようになります。

```
Palatino-Bold /usr/share/lib/hostfontdir/palatino/bold
```

   c. ファイルを保存します。
   適切なシステム上のマップテーブルにサンプルエントリを入れておけば、ユーザーは各自の印刷ジョブに (たとえば、Palatino Bold などの) フォントを適用できます。このフォントを含む印刷要求を依頼すると、LP 印刷サー

ビスはそのファイルに /usr/share/lib/hostfontdir/palatino/bold のコピーを追加してから、プリンタに送信します。

6. troff を使用している場合は、このフォント用の新しいフォント幅テーブルを標準 troff フォントディレクトリ内で作成します。

第 7 章

LP 印刷サービスのカスタマイズの手順

この章では、LP 印刷サービスのカスタマイズについての概要と手順について説明します。

この章で説明する手順は次のとおりです。

- 167ページの「プリンタポート特性を調整する方法」
- 171ページの「サポートされていないプリンタの terminfo エントリを追加する方法」
- 176ページの「独自のプリンタインタフェースプログラムを設定する方法」
- 189ページの「新しい印刷フィルタを作成する方法」
- 194ページの「新しいフォーム定義を作成する方法」

プリンタの概要については、第 2 章を参照してください。

プリンタポート特性の調整

LP 印刷サービスによって設定されるプリンタポート特性には、プリンタの通信設定と互換性がなければなりません。LP 印刷サービスから提供されたデフォルトのプリンタポート設定値がプリンタで機能しない場合は、プリンタのマニュアルを参照し、そのプリンタが LP 印刷サービスに対してどのような設定値が必要かを調べてください。プリンタ通信設定を設定および表示するには、stty コマンドを使用します。

表 7–1 は、LP 印刷サービスに使用されるデフォルトの stty 設定値を示しています。

表 7–1 LP 印刷サービスに使用される stty のデフォルト設定値

オプション	意味
–9600	ボーレートを 9600 に設定する
–cs8	8 ビットバイトを設定する
–cstopb	1 バイトあたり 1 ストップビットを送信する
–parity	パリティを生成しない
–ixon	XON/XOFF (START/STOP または DC1/DC3 ともいう) を使用可能にする
–opost	この表の以下のすべての設定値を使用して「処理後出力」を実行する
–olcuc	小文字を大文字に割り当てない
–onlcr	改行をキャリッジリターン/改行に変更する
–ocrnl	キャリッジリターンを改行に変更しない
–onocr	カラム 0 の位置でもキャリッジリターンを出力する
–n10	改行の後に遅延を入れない
–cr0	キャリッジリターンの後に遅延を入れない
–tab0	タブの後に遅延を入れない
–bs0	バックスペースの後に遅延を入れない
–vt0	垂直タブの後に遅延を入れない
–ff0	用紙送りの後に遅延を入れない

▼ プリンタポート特性を調整する方法

1. プリンタサーバーにスーパーユーザーまたは lp としてログインします。

2. lpadmin コマンドを使用してプリンタポート特性を調整します。

```
# lpadmin -p printer-name -o "stty=options"
```

-p *printer-name*	ポート特性を調整するプリンタ名
-o "stty=*options*"	*options* で指定したポート特性 (stty オプション) を設定する。このコマンドで 1 つ以上の stty オプションを変更できる。オプションを区切るには空白を使用し、オプションを一重引用符で囲む。 オプションについては、stty(1) のマニュアルページを参照。LP 印刷サービスで使用されるデフォルトの stty 設定については、表 7–1 を参照

3. 次のコマンドを使用して、プリンタポート特性が変更されたかどうかを確認します。

```
# stty -a
```

例 — プリンタポート特性を調整する

次の例で、コマンドはプリンタ luna のポート特性を設定します。parenb オプションはパリティチェック/生成を使用可能にし、parodd は奇数パリティの生成を設定し、cs7 は文字サイズを 7 ビットに設定します。

```
# lpadmin -p luna -o "stty='parenb parodd cs7'"
```

次の例で、コマンドはプリンタ venus の端末ボーレートを 19200 に設定します。

```
# lpadmin -p venus -o "stty=19200"
```

サポートされていないプリンタの terminfo エントリを追加する

LP 印刷サービスは、インタフェースプログラムと terminfo データベースを使用して各プリンタを初期化し、選択されたページサイズ、文字ピッチ、行ピッチ、および文字セットを設定します。

各プリンタは、terminfo データベース内で短縮名を使用して識別されます。terminfo データベースで使用される名前は、TERM シェル変数を設定するのに使用する名前と同じです。また、この名前はプリンタを設定するときに指定するプリンタタイプでもあります。たとえば、各種の PostScript プリンタのエントリは、/usr/share/lib/terminfo/P に入っています。SunOS システムに組み込まれているデフォルトエントリは、PS (PostScript 用) と PSR (Reverse PostScript 用) です。

プリンタの terminfo エントリが見つからなくても、ページサイズ、ピッチ、および文字セットを自動選択しないで、LP 印刷サービスでプリンタを使用できます。ただし、印刷要求ごとに正しいモードでプリンタが設定されないという問題が起きることがあります。

使用中のプリンタタイプの terminfo エントリがないが、プリンタを正しいモードに設定しておきたい場合は、プリンタで使用するインタフェースプログラムをカスタマイズするか、terminfo データベースにエントリを追加できます。terminfo データベース内の端末エントリやプリンタエントリには、多数の項目が入っており、定義されています。しかし、LP 印刷サービスはそのうち 50 未満の項目しか使用しません。表 7–2 は、プリンタに必要な terminfo 項目を示しています。

表 7–2 プリンタに必要な terminfo 項目

項目	意味
ブール値	
cpix	文字ピッチを変更すると解像度が変わる
daisy	プリンタで文字セットを変更するには演算子が必要
lpix	行ピッチを変更すると解像度が変わる

表 7–2 プリンタに必要な terminfo 項目 続く

項目		意味
数値		
	bufsx	印刷前にバッファされるバイト数
	cols	1 行の列数
	cps	1 秒あたりの文字の平均印刷速度
	it	最初はスペース *n* 個分ごとのタブ
	lines	1 ページの行数
	orc	1 文字あたりの水平解像度
	orhi	1 インチあたりの水平解像度
	orl	1 行あたりの垂直解像度
	orvi	1 インチあたりの垂直解像度
文字列		
	chr	水平解像度を変更する
	cpi	1 インチあたりの文字数を変更する
	cr	キャリッジリターン
	csnm	文字セット名のリスト
	cudl	1 行下げる
	cud	キャリッジを *n* 行下に移動する
	cuf	キャリッジを *n* 列右に移動する
	cvr	垂直解像度を変更する

表 7–2 プリンタに必要な terminfo 項目 続く

項目		意味
	ff	紙送りする
	hpa	水平絶対位置
	ht	次の 8 スペースタブストップまでのタブ
	if	初期化ファイル名
	iprog	初期化プログラムのパス名
	is1	プリンタ初期化文字列
	is2	プリンタ初期化文字列
	is3	プリンタ初期化文字列
文字列		
	lpi	1 インチあたりの行数を変更する
	mgc	マージン (上、下、横) をすべて消去する
	rep	文字を *n* 回繰り返す
	rwidm	倍幅の印刷を使用不可にする
	scs	文字セットを選択する
	scsd	文字セットの定義を開始する
	slines	ページの長さを 1 ページあたり *n* 行に設定する
	smgl	現在の列の左マージンを設定する
	smglp	左マージンを設定する
	smgr	現在の列の右マージンを設定する

表 7–2　プリンタに必要な terminfo 項目　続く

項目	意味
smgrp	右マージンを設定する
smglr	左右のマージンを設定する
msgt	現在の行の上マージンを設定する
smgtp	上マージンを設定する
smgb	現在の行の下マージンを設定する
smgbp	下マージンを設定する
smgtb	上下のマージンを設定する
swidm	倍幅の印刷を使用可能にする
vpa	垂直絶対位置

▼ サポートされていないプリンタの terminfo エントリを追加する方法

注 - プリンタの terminfo エントリを作成する前に、まず、そのプリンタをサポートする既存の terminfo エントリがないことを確認してください。そのためには、類似するプリンタがあれば、そのエントリを使用してプリンタを設定してください。

1. プリンタサーバーにスーパーユーザーまたは lp としてログインします。
2. プリンタの terminfo エントリ名を決定します。

   /usr/share/lib/terminfo ディレクトリ内のディレクトリには、有効な terminfo エントリがすべて入っています。それを参考にしてプリンタ名を選択してください。
3. プリンタの terminfo エントリファイルを作成します。

LP 印刷サービスに新しいプリンタを追加するために terminfo エントリ内で定義しなければならない項目については、表 7–2 を参照してください。terminfo データベースの構造については、terminfo(4) のマニュアルページを参照してください。

新しい terminfo エントリを作成しやすいように、infocmp コマンドを使用して既存の terminfo エントリをファイルに保存します。これは、作成したいエントリに似た terminfo エントリがある場合に便利な方法です。たとえば、次のコマンドで ps エントリを ps_cust ファイルに保存すると、新しい terminfo エントリになります。

```
infocmp ps > ps_cust
```

4. terminfo エントリをコンパイルして terminfo データベースに入れます。

```
# tic terminfo_entry
```

terminfo_entry　作成した terminfo エントリファイル

5. /usr/share/lib/terminfo ディレクトリ内で新しい terminfo エントリファイルをチェックします。

プリンタインタフェースプログラムのカスタマイズ

標準プリンタインタフェースプログラムでサポートされないプリンタを使用する場合は、独自のプリンタインタフェースプログラムを提供できます。標準プログラムをコピーし、指定したプリンタを使用するように LP 印刷サービスに指示できます。ただし、まず標準プログラムについて理解する必要があります。次の節では、標準プログラムについて説明します。

プリンタインタフェースプログラムの機能は次のとおりです。

- 必要に応じてプリンタポートを初期化する。標準プリンタインタフェースプログラムは、stty コマンドを使用してプリンタポートを初期化する。

- プリンタハードウェアを初期化する。標準プリンタインタフェースプログラムは、terminfo データベースと TERM シェル変数から制御シーケンスを取得する。
- 必要に応じてバナーページを印刷する。
- 印刷要求で指定された部数を印刷する。

注意 - リリース 3.2 より前の UNIX System V のプリンタインタフェースプログラムを使用している場合でも、そのプログラムは、SunOS 5.8 または互換 LP 印刷サービスで使用できます。ただし、一部の –o オプションは SunOS 5.8 または互換 LP 印刷サービスでは標準化されていて、すべてのプリンタインタフェースプログラムに渡されます。これらのオプションは、古いインタフェースで使用される同じ名前のオプションの妨げとなることがあります。

プリンタポートを開く処理は、プリンタインタフェースプログラムではなく LP 印刷サービスが受け持ちます。プリンタポートは標準出力としてプリンタインタフェースプログラムに与えられ、プリンタはプリンタインタフェースプログラムの「制御端末」として識別されるので、ポートが「ハング」するとプリンタインタフェースプログラムに SIGHUP 信号が送信されます。

標準プリンタインタフェースプログラム

LP 印刷サービスは、標準 (モデル) プリンタインタフェースプログラム /usr/lib/lp/model/standard を使用して、表 7–3 の印刷デフォルトを設定します。

表 7–3 デフォルトのプリンタポート特性

特性	デフォルト設定
デフォルトフィルタ	None
文字ピッチ	None
行ピッチ	None
ページ幅	None
ページ長	None

表 7–3 デフォルトのプリンタポート特性 続く

特性	デフォルト設定
文字セット	None
stty オプション	9600 cs8 -cstopb -parenb -parodd ixon -ixany opost -olcuc onlcr -ocrnl -onocr -onlret -ofill nl0 cr0 tab0 bs0 vt0 ff0
終了コード	0

stty モードのカスタマイズ

ボーレートや出力オプションなどの端末特性を変更する必要がある場合は、標準プリンタインタフェースプログラム内で、次のコメントから始まるセクションを探します。

```
## Initialize the printer port
```

終了コード

印刷し終わると、インタフェースプログラムは印刷ジョブの状態を示すコードを返して終了します。終了コードは、プリンタインタフェースプログラムの最後のエントリです。

表 7–4 は、終了コードとそのコードが LP 印刷サービスでどのように解釈されるかを示しています。

表 7–4 プリンタインタフェースプログラムの終了コード

コード	**LP** 印刷サービスにとっての意味
0	印刷要求は正常に完了した。プリンタ障害が発生した場合は、クリアされた
1 ～ 127	要求の印刷中に問題が発生した (たとえば、印字できない文字が多すぎる、要求がプリンタの容量を超えているなど)。LP 印刷サービスは、その要求を依頼したユーザーに、印刷中にエラーが発生したことを通知する。このエラーはその後の印刷要求には影響しない。プリンタ障害が発生するとクリアされる

表 7–4 プリンタインタフェースプログラムの終了コード 続く

コード	LP 印刷サービスにとっての意味
128	このコードは、LP 印刷サービスが内部で使用するために予約されている。インタフェースプログラムは、このコードを返して終了してはいけない
129	要求の印刷中にプリンタ障害が発生した。この障害は、その後の印刷要求に影響を及ぼす。プリンタの障害回復が LP 印刷サービスに管理者によって問題が解決されるまで待つように指示すると、LP 印刷サービスはプリンタを使用不可にする。障害回復後に印刷を続けようとすると、LP 印刷サービスはプリンタを使用不可にしないが、数分後にそのまま印刷しようとする
129 より大きい場合	これらのコードは、LP 印刷サービスが内部で使用するために予約されている。インタフェースプログラムは、この範囲内のコードを返して終了してはいけない

プログラムがコード 129 を返して終了すると、root はプリンタ障害を警告されます。また、LP 印刷サービスは、障害がクリアされた後に要求を最初から印刷し直さなければなりません。要求全体を印刷し直したくない場合は、インタフェースプログラムに障害メッセージを LP 印刷サービスへ送信させることもできますが、障害がクリアされるまで待つことになります。障害がクリアされると、インタフェースプログラムはファイルの印刷を再開できます。印刷が終了すると、プリンタインタフェースプログラムは障害が発生しなかった場合と同様に終了コード 0 を返すことができます。このアプローチには、障害が自動的にクリアされた場合に、それをインタフェースプログラムが検出できるので、管理者がプリンタを再び使用可能にする必要がないという利点もあります。

障害メッセージ

`lp.tell` プログラムを使用すると、LP 印刷サービスに障害メッセージを送信できます。このプログラムは、標準プリンタインタフェースコード内の `LPTELL` シェル変数によって参照されます。プログラムは標準入力を取り込んで LP 印刷サービスに送信し、LP 印刷サービスは管理者にプリンタ障害を警告するメッセージを出します。標準入力が空であれば、`lp.tell` は警告を開始しません。`lp.tell` プログラムの例として、次のコメントの直後の標準プリンタインタフェースコードを確認してください。

```
# Set up the $LPTELL program to capture fault messages here
```

特殊な終了コード 129 または lp.tell プログラムを使用すると、プリンタインタフェースプログラムはプリンタ自体を使用不可にする必要がありません。インタフェースプログラムは、プリンタを直接使用不可にできますが、その場合は障害警告メカニズムが無効になります。LP 印刷サービスがプリンタ障害を検出した場合にのみ警告が送信され、特殊終了コードと lp.tell プログラムはその主要検出ツールです。

LP 印刷サービスがいずれかの時点でファイルの印刷を中断しなければならない場合は、TERM 信号 (トラップ番号 15 と、kill(1) および signal(3B) のマニュアルページを参照) を使用してインタフェースプログラムを強制終了します。プリンタインタフェースプログラムが他の信号を受信しなくなると、LP 印刷サービスはその後の印刷要求は影響されないものとみなし、そのプリンタを使用し続けます。LP 印刷サービスは、要求を依頼したユーザーに、その要求が正常に終了しなかったことを通知します。

インタフェースが最初に呼び出されると、信号 HUP、INT、QUIT、PIPE (トラップ番号 1、2、3、13) は無視されます。標準インタフェースは、信号が適切な時期にトラップされるように、この動作を変更します。標準インタフェースはこれらの信号の受信をプリンタの問題を示す警告として解釈し、信号を受信すると障害警告を発します。

カスタマイズされたプリンタインタフェースプログラムの使用方法

カスタマイズされたプリンタインタフェースプログラムを作成し、プリンタサーバー上で標準プリンタインタフェースプログラムの代わりに使用できます。そのためには、lpadmin コマンドを使用して、プログラムを特定のプリンタの LP 印刷サービスに登録します。

▼ 独自のプリンタインタフェースプログラムを設定する方法

1. プリンタサーバーにスーパーユーザーまたは lp としてログインします。

2. 独自のプリンタインタフェースプログラムがあるかどうかに基づいて次の手順を選択します。

独自のプリンタインタフェースプログラムの有無	次の手順
独自のプリンタインタフェースプログラムがないので作成する必要がある	177ページの手順 3
すでに独自のプリンタインタフェースプログラムがある	177ページの手順 5

3. 標準プリンタインタフェースプログラムをコピーします。

```
# cp /var/spool/lp/model/standard custom-interface
```

4. 標準プリンタインタフェースプログラムのコピーを必要に応じて変更します。

   詳しくは、173ページの「標準プリンタインタフェースプログラム」のプログラムの説明を参照して、何を変更する必要があるかを判断してください。

5. 特定のプリンタ独自のプリンタインタフェースプログラムを設定します。

```
# lpadmin -p printer-name -i custom-interface
```

-p *printer-name*	独自のプリンタインタフェースプログラムを使用するプリンタ名
-i *custom-interface*	独自のプリンタインタフェースプログラム名

   独自のプリンタインタフェースプログラムが LP 印刷サービスに登録され、ユーザーが印刷要求を出すと、そのプリンタに使用されます。

6. 独自のプリンタインタフェースプログラムが `/etc/lp/printers/`***printer-name***`/configuration` ファイルに追加されたかどうかを確認します。

例 — 独自のプリンタインタフェースプログラムを設定する

次の例では、プリンタ `luna` の独自のプリンタインタフェースプログラム `custom` を設定します。

```
# lpadmin -p luna -i custom
```

次の例では、システム venus がプリンタ asteroid 上で使用中の独自のプリンタインタフェースプログラムを設定します。

```
# lpadmin -p asteroid -e venus
```

新しい印刷フィルタの作成

フィルタは、LP 印刷サービスがプリンタで解釈できないタイプのファイルを印刷する必要があるたびに使用されます。新しい印刷フィルタを作成するのは簡単ではありません。通常は広範囲の経験が必要です。新しい印刷フィルタを定義するには、次の 2 つの手順があります。

- 印刷フィルタプログラムを作成する
- 印刷フィルタ定義を作成する

印刷フィルタは、必要に応じて簡単なものでも複雑なものでもかまいません。フィルタには、入力形式、出力形式、そのフィルタ内でコマンド行引数を処理する言語を提供する複雑なオプションが入っています。

PostScript 以外のプリンタを使用する場合は、必要に応じて印刷フィルタを作成して追加する必要があります。まず、印刷フィルタの機能と、フィルタプログラムが満たさなければならない要件を理解しておく必要があります。

印刷フィルタプログラムの作成

LP 印刷サービスには、/usr/lib/lp/postscript ディレクトリにフィルタプログラムが組み込まれています。これらのフィルタは、宛先プリンタが PostScript 形式のデータを必要とするほとんどの PostScript 印刷の状況に対応します。印刷フィルタプログラムは、2 進の実行可能プログラムでなければなりません。

フィルタのタイプ

印刷フィルタには、高速フィルタと低速フィルタの 2 種類があります。

高速フィルタは、ファイルの印刷準備にあまり処理時間がかかりません。また、実行するときにはプリンタにアクセスしなければなりません。印刷障害を検出する印刷フィルタは、高速フィルタでなければなりません。PRINTER キーワードをフィルタオプションとして使用するフィルタは、高速フィルタとしてインストールしなければなりません。

低速フィルタは、ファイルの印刷準備に大量の処理時間がかかります。また、低速フィルタは実行するときにプリンタにアクセスする必要がありません。低速フィルタはバックグラウンドで実行されるので、プリンタと連結する必要がなく、低速フィルタを必要としない他のファイルを印刷できます。

ファイルの変換

LP 印刷サービスは、印刷フィルタを使用して、ある内容形式から別の内容形式にファイルを変換します。プリンタごとに、受け付けられるファイル内容形式を指定できます。ユーザーは印刷要求を出すときにファイル内容形式を指定し、LP 印刷サービスはその内容形式のファイルを印刷できるプリンタを探します。多くのアプリケーションは各種プリンタに合わせてファイルを生成できるので、通常はこれで十分です。ただし、アプリケーションによっては、利用できるプリンタ上で印刷できないファイルを生成するものがあります。

LP 印刷サービスは、プリンタで直接受け付けられない形式のファイルの印刷要求を受信するたびに、印刷要求の内容形式を利用できる (または指定された) プリンタの内容形式と一致させようとします。一致すると、そのファイルはフィルタをかけなくてもプリンタに直接送信できます。一致する形式が見つからない場合や、内容形式でフィルタを使用するように指定されている場合は、LP 印刷サービスはファイルの内容形式を利用できるフィルタの入力内容形式と一致させ、フィルタの出力形式をプリンタの内容形式と一致させようとします。適切なフィルタが見つかると、印刷要求はフィルタを通して渡されます。

特殊印刷モードの処理

印刷フィルタは、特殊モードを処理し特殊ページの印刷を要求します。特殊印刷モードは、カスタマイズされたフィルタが必要な印刷要求の特性を印刷するために必要です。フィルタは、次の特性を処理します。

- プリンタタイプ
- 文字ピッチ
- 行ピッチ

- ページ長
- ページ幅
- 印刷ページ数
- 文字セット
- フォーム名
- 部数

LP 印刷サービスは、これらの特性にデフォルト設定を提供します。ただし、印刷フィルタの方が一部の特性を効率よく処理できます。たとえば、プリンタによっては、LP 印刷サービスより効率よく複数のコピーを処理できるものがあり、その場合は複数コピーのページ制御用フィルタを提供できます。

プリンタ障害の検出

各プリンタは、独自の方法でプリンタ障害を検出し、障害信号を LP 印刷サービスに送信します。LP 印刷サービスは、印刷中にハングアップ (キャリア喪失) と過剰な遅延の有無をチェックするだけです。

プリンタによっては、優れた障害処理能力を持ち、障害の原因を記述するメッセージを送信できるものがあります。また、キャリア信号の喪失やデータフローのシャットオフを示すものとは異なる信号を使用して障害を示すプリンタもあります。これらの付加的なプリンタ障害情報を解釈するには、フィルタが必要です。

また、フィルタは印刷要求を保留し、プリンタ障害がクリアされるまで待ってから印刷を再開します。この機能を使用すると、中断された印刷要求全体を印刷し直す必要がありません。プリンタに使用される制御シーケンスを認識するフィルタだけが、ファイルの改ページ位置を判別できます。したがって、その種のフィルタ以外は、障害がクリアされた後で印刷を再開する必要があるファイル内の位置を検出できません。

印刷フィルタがメッセージを生成すると、そのメッセージは LP 印刷サービスによって処理され、警告が使用可能になっている場合は、システム管理者に警告が送信されます。109ページの「障害の通知の設定」を参照してください。

印刷フィルタプログラムの要件

印刷フィルタは単純なものでも複雑なものでもかまいませんが、次の要件を満たさなければなりません。

- フィルタは、標準入力からファイルの内容を取得して、変換後のファイルを標準出力に送信しなければなりません。
- 外部ファイルを参照するプログラムはフィルタとして使用できません。一般にユーザーは troff、nroff などのワープロプログラムをフィルタとして使用しがちです。LP 印刷サービスは、フィルタプログラムから「組み込みファイル」と呼ばれる他のファイルへの参照を認識しません。troff と nroff はファイルを組み込めるので、フィルタとして使用すると失敗することがあります。プログラムが処理を完了するために他のファイルを必要とする場合は、フィルタとして使用しないでください。
- フィルタは、一般にユーザーがアクセスできないファイルに依存しないでください。ユーザーが直接実行するときにフィルタが失敗すると、LP印刷サービスで実行しても失敗します。
- 低速フィルタは、ファイル内のエラーに関するメッセージを標準エラーに送信しますが、高速フィルタは送信しません。低速フィルタからのエラーメッセージは収集され、印刷要求を出したユーザーに送信されます。
- 低速フィルタが信号を受信したために終了すると、印刷要求が停止され、その要求を出したユーザーに通知されます。同様に、低速フィルタが0 以外の終了コードを返して終了すると、印刷要求が停止され、ユーザーに通知されます。高速フィルタからの終了コードは、異なる方法で処理されます。

フィルタにプリンタ障害を検出させたい場合は、次の要件も満たしてください。

- フィルタは終了する前にできるだけ障害がクリアされるまで待つ必要があります。また、障害がクリアされたら、印刷を停止したページの先頭から印刷を再開してください。継続機能を使用したくない場合は、LP印刷サービスは管理者に警告する前にフィルタを停止します。
- フィルタは、障害が認識されたらすぐにプリンタ障害メッセージを標準エラーに送信する必要があります。終了する必要はなく、障害がクリアされるまで待つことができます。
- フィルタは、ファイル内のエラーに関するメッセージを標準エラーに送信してはいけません。これらのメッセージは、ユーザーが読み込める標準出力に含める必要があります。
- フィルタは、ファイルの印刷が終了すると (ファイル内のエラーによって正常に印刷できなかった場合も) 終了コード 0 を返して終了する必要があります。
- フィルタは、プリンタ障害によって印刷要求を終了できなかった場合にのみ、0 以外の終了コードを返して終了する必要があります。

- フィルタをフィルタテーブルに追加する場合は、高速フィルタとして追加しなければなりません。

印刷フィルタ定義の作成

印刷フィルタ定義は、フィルタ、実行する印刷フィルタプログラム、実行する変換の種類などを LP 印刷サービスに指示します。/etc/lp/fd ディレクトリには、一連のフィルタ記述子ファイルが組み込まれています。これらのファイルは、フィルタの特性 (高速または低速フィルタなど) を記述し、フィルタプログラム (/usr/lib/lp/postscript/postdaisy など) を指します。

新しい印刷フィルタを定義するときには、フィルタプログラムを作成するだけでなく、印刷フィルタ定義を作成しなければなりません。印刷フィルタ定義には、LP 印刷サービスが使用する次の情報が入っています。

- 実行するフィルタプログラム名
- 受け付ける入力タイプ
- 生成する出力タイプ
- ジョブを送信できるプリンタタイプ
- ジョブを送信できる特定のプリンタ名
- フィルタタイプ (高速または低速)
- オプション

lpfilter コマンドへの直接入力として特性を入力できます。また、フィルタの特性を指定するファイルを作成し、ファイル名を lpfilter コマンドへの入力として使用することもできます。この種のファイルは「フィルタ記述子ファイル」と呼ばれ、/etc/lp/fd ディレクトリに配置してください。これらのファイルはフィルタそのものではなく、フィルタを指すものです。

情報をファイルに格納するかコマンド行に直接入力するかに関係なく、次の形式を使用してください。

```
Command: command-pathname [options]
Input types: input-type-list
Output types: output-type-list
Printer types: printer-type-list
Printers: printer-list
Filter type: fast または slow
```

(続く)

```
Options: template-list
```

注 - Options 以外のフィルタ特性に複数の定義 (つまり複数行) を与えると、2 番目の定義のみが印刷サービスに使用されます。

情報はどんな順序で配置してもかまいません。また、すべての情報が必要とは限りません。値を指定しなければ、表 7–5 の値がデフォルトで割り当てられます。デフォルト値はあまり便利ではないので、明示的な値を指定するようにしてください。

表 7–5 lpfilter 引数のデフォルト値

項目	デフォルト
入力タイプ	任意
出力タイプ	任意
プリンタタイプ	任意
プリンタ	任意
フィルタタイプ	低速

コマンド

フィルタプログラムのフルパスを使用してください。プログラムが必ず必要とする固定オプションがある場合は、それらをこのコマンドに含めます。

入力タイプ

印刷フィルタが処理できるファイル内容形式のリストです。LP 印刷サービスには入力タイプ数の制限はありませんが、ほとんどのフィルタは 1 つのタイプしか受け付

けません。いくつかのファイルタイプは、フィルタで処理できる程度に類似しています。14 文字以内の英数字とダッシュを使用した名前を指定できます。入力タイプ名の一部として下線 (_) は使用できません。

LP 印刷サービスは、一貫した命名規則に合致するように、これらの名前を使用してフィルタをファイルタイプと一致させます。たとえば、複数のフィルタが同じ入力タイプを受け付ける場合は、各フィルタ用に指定するときに、その入力タイプに同じ名前を使用します。ユーザーには、ファイルの印刷を依頼するときにファイルタイプの識別方法がわかるように、これらの名前を通知してください。

出力タイプ

出力タイプは、フィルタが出力として生成できるファイルタイプのリストです。入力タイプごとに、フィルタは 1 つずつ出力タイプを生成します。ただし、出力タイプはジョブごとに異なることがあります。出力タイプ名は、14 文字以内の英数字とダッシュ (–) です。

出力タイプ名は、利用できる (ローカルまたはリモート) プリンタのタイプと一致するか、他のフィルタで処理される入力タイプと一致しなければなりません。LP 印刷サービスは、ファイルを変換するために異なるフィルタを複数回かける必要があることがわかると、フィルタをシェルパイプラインにグループ化します。このように複雑な処理はほとんど必要ありませんが、LP 印刷サービスではこの処理を実行できます。ユーザーが印刷したいあらゆるファイルを入力タイプにするフィルタの集合を探して、各ファイルをプリンタで処理できるファイルタイプに直接変換してください。

プリンタタイプ

プリンタタイプは、印刷フィルタがファイルを変換できるプリンタタイプのリストです。ほとんどのプリンタとフィルタの場合、これは出力タイプのリストと同じなので、フィルタ定義のこの部分はブランクにしておいてかまいません。しかし、別のタイプを使用することもできます。たとえば、初期化には 1 つのプリンタタイプを使用するプリンタでも、複数のファイル内容形式を認識できるものがあります。実際には、このプリンタには、各種ファイルタイプを、処理できるファイルタイプに変換する内部フィルタが付いています。したがって、1 つのフィルタは、プリンタで処理できるファイルタイプに合った複数の出力タイプのいずれかを生成できます。印刷フィルタには、そのプリンタタイプで機能することを示すマークを付ける必要があります。

もう 1 つの例として、同じファイルタイプを受け付けるものとして表示される 2 つのモデルのプリンタを使用できます。ただし、内部に若干違いがあるので、一方のプリンタが生成する結果と異なります。それらのプリンタには、A と B など、異なるプリンタタイプであることを示すラベルを付けます。この場合、B はズレがある方のプリンタです。タイプ B のプリンタで生成されるズレを考慮してファイルを調整するフィルタを作成します。このフィルタはこの種のプリンタタイプにのみ必要なので、タイプ B プリンタでのみ機能するものとして指定します。

プリンタ

一般に、印刷フィルタはその出力を受け付けるどのプリンタでも機能するので、通常はフィルタ定義のこの部分をスキップできます。

ただし、プリンタによっては、フィルタが生成する出力に適したものと適さないものがあります。たとえば、1 台のプリンタを高速出力専用にして、フィルタを通す必要がないファイルのみをそのプリンタに送信できます。同じタイプの他のプリンタは、印刷前に広範囲にフィルタを通す必要があるファイルに使用できます。

フィルタタイプ

LP 印刷サービスは、178ページの「フィルタのタイプ」で説明しているように、高速フィルタと低速フィルタを認識します。

印刷モードで (`lp -y` コマンドを使用して) 呼び出される低速フィルタは、印刷要求が出されたシステム上で実行しなければなりません。LP 印刷サービスはモード値をプリンタサーバーに渡せません。ただし、ファイル内容形式 (`lp` コマンドの `-T` オプションの後で指定) をプリンタサーバー上の内容形式と一致させることはできます。したがって、プリンタサーバー上で特殊モードを有効にしたい場合は、LP 印刷サービスが入力タイプと出力タイプを一致することができる内容形式を指定しなければなりません。

オプション

オプションは、各種の情報をフィルタコマンドのコマンド行引数にどのように変換するかを指定します。この情報には、ユーザーからの (印刷要求を伴う) 指定、プリンタ定義、要求の処理に使用されるフィルタによって実装される仕様などを含めることができます。

テンプレートによる印刷フィルタオプションの定義

印刷フィルタオプションの定義には 13 個の情報源があり、それぞれが「キーワード」で表されています。各オプションは「テンプレート」内で定義されます。テンプレートとは、フィルタのいずれかの特性値に基づいてフィルタコマンドに渡されるオプションを定義する、フィルタ定義内のステートメントです。

フィルタ定義で指定するオプションには、13 個のキーワードを使用しなくても、すべて使用しても、そのサブセットを使用してもかまいません。また、完全なフィルタ定義に複数の定義が必要な場合は、1 つのキーワードを複数回指定することもできます。印刷フィルタ定義における `Options` を定義するための 13 個のキーワードについては、表 7–6 を参照してください。

表 7–6 印刷フィルタオプションのキーワード

特性	キーワード	考えられるパターン	例
内容形式 (入力)	`INPUT`	内容形式	`troff`
内容形式 (出力)	`OUTPUT`	内容形式	`postscript`、`impress`
プリンタタイプ	`TERM`	プリンタタイプ	`att495`
プリンタ名	`PRINTER`	プリンタ名	`lp1`
文字ピッチ	`CPI`	四捨五入された 10 進数	`10`
行ピッチ	`LPI`	四捨五入された 10 進数	`6`
ページ長	`LENGTH`	四捨五入された 10 進数	`66`
ページ幅	`WIDTH`	四捨五入された 10 進数	`80`
印刷ページ数	`PAGES`	ページリスト	`1-5,13-20`
文字セット	`CHARSET`	文字セット	`finnish`

表 7–6 印刷フィルタオプションのキーワード 続く

特性	キーワード	考えられるパターン	例
フォーム名	FORM	フォーム名	invoice2
部数	COPIES	整数	3
特殊モード	MODES	モード	landscape

印刷フィルタ定義には、複数のテンプレートを含めることができます。複数のテンプレートを指定する場合は、1 行にコンマで区切って入力するか、先頭に Options: を付けて複数行に入力します。

テンプレートの形式は次のとおりです。

keyword*pattern* = *replacement*

keyword は、フィルタの特定の特性に関して登録されたオプションのタイプを識別します。

pattern は、キーワードの特定のオプションです。

replacement は、キーワードが指定した値のときに発生する動作です。

特定のフィルタのオプションを定義する方法を示す例として、印刷サービススケジューラを使って次の条件でフィルタに印刷要求を割り当てたいものと仮定します。

- フィルタで生成される OUTPUT のタイプが impress の場合は、フィルタに -I オプションを渡す。
- フィルタで生成される OUTPUT のタイプが postscript の場合は、フィルタに -P オプションを渡す。

上記の条件を指定するには、lpfilter コマンドのオプションとして次のテンプレートを与えます。

```
Options: OUTPUT impress=–I, OUTPUT postscript=–P
```

Options 行が長くなりすぎる場合は、次のように各テンプレートを別々の行に入れます。

```
Options: OUTPUT impress=-I
Options: OUTPUT postscript=-P
```

どちらのテンプレートでも、*keyword* は OUTPUT として定義されています。第 1 のテンプレートでは、パターンは impress で、「replacement」の値は -I です。第 2 のテンプレートでは、「pattern」の値は postscript で、「replacement」の値は -P です。

各種テンプレート (つまり、各キーワードの *pattern* 引数と *replacement* 引数) に与える値を見つけるには、次のことに注意する必要があります。

- INPUT テンプレートの値は、フィルタによる変換が必要なファイル内容形式からとっています。
- OUTPUT テンプレートの値は、フィルタが生成すべき出力タイプからとっています。
- TERM テンプレートの値はプリンタタイプです。
- PRINTER テンプレートの値は、最終出力を印刷するプリンタ名です。
- CPI、LPI、LENGTH、WIDTH の各テンプレートの値は、ユーザーの印刷要求、使用するフォーム、またはプリンタのデフォルト値からとっています。
- PAGES テンプレートの値は、印刷すべきページのリストです。通常、これはコンマで区切ったページ範囲のリストです。各ページ範囲は、ダッシュで区切った 1 対の数値、または単一の数字からなります (たとえば、1–5,6,8,10 は 1 ページから 5 ページまでと、6 ページ、8 ページ、10 ページを示します)。ただし、印刷要求に –P オプションで値を与えると、印刷要求は変更されずに渡されます。
- CHARSET テンプレートの値は、使用する文字セット名です。
- FORM テンプレートの値は、lp -fコマンド (印刷要求を出すのに使用するコマンド) で要求したフォーム名です。
- COPIES テンプレートの値は、ファイルの印刷部数です。フィルタがこのテンプレートを使用する場合、この「1 部」にはフィルタによって生成される複数のコピーが含まれるので、LP 印刷サービスはフィルタがかけられたファイルの印刷部数から 1 を引きます。
- MODES テンプレートの値は、lp –y コマンドからとっています。ユーザーは複数の –y オプションを指定できるので、MODES テンプレートは複数の値をとることがあります。値はユーザーが与えた順に左から右に適用されます。

テンプレートの *replacement* 部は、テンプレートの値をフィルタプログラムに与える方法を示します。通常はリテラルオプションですが、値の位置を示すプレースホルダのアスタリスク (*) が付いていることがあります。*pattern* と *replacement* には、ユーザー入力オプションからフィルタオプションへの複雑な変換を行うために、ed(1) の正規表現の構文を使用することもできます。ed(1) の正規表現の構文

はすべて使用できます。たとえば、\(... \) や \n 構成を使用すると、*pattern* の各部を抽出して *replacement* にコピーできます。また、& を使用すると、*pattern* 全体を *replacement* にコピーできます。

注 - *pattern* または *replacement* にコンマまたは等号 (=) を含める場合は、その前にバックスラッシュ (\) を付けてください。これらの文字の前にバックスラッシュが付いていると、*pattern* または *replacement* が使用されるときに削除されます。

▼ 新しい印刷フィルタを作成する方法

1. プリンタサーバーにスーパーユーザーまたは lp としてログインします。

2. 印刷フィルタプログラムを作成します。

   印刷フィルタプログラムの概要については、178ページの「印刷フィルタプログラムの作成」を参照してください。印刷フィルタ定義はテキストファイルに保存する必要があります。使用しやすいように、通常、フィルタ定義は /usr/lib/lp/postscript ディレクトリに入っています。作成したプログラムは、選択したディレクトリ内の /usr/lib/lp の下に入れる必要があります。

3. 印刷フィルタ定義を作成します。

   印刷フィルタ定義の概要については、182ページの「印刷フィルタ定義の作成」を参照してください。印刷フィルタ定義はテキストファイルに保存する必要があります。使用しやすいように、通常、フィルタ定義は /etc/lp/fd ディレクトリに入っており、接尾辞 .fd で識別されます。

4. 印刷フィルタをプリンタサーバーに追加します。

   詳細は、144ページの「印刷フィルタを追加する方法」を参照してください。

例 — 新しい印刷フィルタを作成する

次の例は、N37 または Nlp を simple に変換する印刷フィルタ定義を示します。

```
Input types: N37, Nlp, simple
Output types: simple
Command: /usr/bin/col
Options: MODES expand = -x
Options: INPUT simple = -p -f
```

次の例で、印刷フィルタプログラム名は col です。新しい印刷フィルタをプリンタサーバーに追加すると、ユーザーの印刷要求は次のように処理されます。

- ユーザーが次のコマンドを入力した場合

```
$ lp –y expand report.doc
```

印刷プログラムは次の引数を使用して実行され、ファイルが変換されます。

```
/usr/bin/col –x –p –f
```

- ユーザーが次のコマンドを入力した場合

```
$ lp –T N37 –y expand report.doc
```

印刷プログラムは次の引数を使用して実行され、ファイルが変換されます。

```
/usr/bin/col –x
```

次の例は、troff から PostScript に変換する印刷フィルタ定義を示します。

```
Input types: troff

Output types: postscript

Printer types: PS

Filter type: slow

Command: /usr/lib/lp/postscript/dpost

Options: LENGTH * = –l*

Options: MODES port = –pp, MODES land = -pl
Options: MODES group \=\([1-9]\) = –n\l
```

次の例で、フィルタプログラム名は dpost です。このプログラムは入力タイプ troff をとり、postscript 出力を生成し、タイプ PS (PostScript) のプリンタに機能します。ユーザーは、用紙方向を縦モードにするか横モードにするかを尋ねるプロンプトが表示されたときに、それぞれの省略形 port または land を指定するだけですみます。これらのオプションは LP 印刷サービスに固有ではないので、ユーザーは lp –y コマンドを使用して指定しなければなりません。

新しい印刷フィルタをプリンタサーバーに追加すると、印刷要求は次のように処理されます。

■ ユーザーが次のコマンドを入力し、横方向、ページ長60行で、troff ファイルタイプを PostScript プリンタ (タイプ PS) で印刷するように要求した場合

```
$ lp -T troff -o length=60 -y land -d luna ch1.doc
```

印刷フィルタプログラム dpost は、次の引数を使用して実行され、ファイルが変換されます。

```
/usr/lib/lp/postscript/dpost –l60 –pl luna ch1.doc
```

■ ユーザーが次のコマンドを入力した場合

```
$ lp -T troff -y group=4 -d luna ch1.doc
```

次の引数が指定された印刷フィルタプログラム dpost コマンドは、ファイルを変換します。

```
/usr/lib/lp/postscript/dpost -n4
```

新しいプリンタフォームの作成

新しいフォームを提供したい場合は、lpforms コマンドへの入力として 9 個の必須特性 (ページ長とページ幅など) に関する情報を入力し、その特性を定義しなければなりません。LP 印刷サービスは、この情報を次の 2 つの目的に使用します。

■ フォーム上に正しく印刷されるようにプリンタを初期化する。

■ フォームの処理方法に関する留意事項をシステム管理者に送信する。

フォーム名は、14 文字以内の英数字と下線であれば、任意に選択して使用できます。情報は次の形式でなければなりません。

```
Page length: scaled number
Page width: scaled number
Number of pages: integer
Line pitch: scaled number
Character pitch: scaled number
Character set choice: character-set-name [,mandatory]
Ribbon color: ribbon-color
Comment:
informal notes about the form
```

(続く)

```
Alignment pattern: [content-type] alignment pattern
```

省略可能な句、[,mandatory] は、ユーザーがフォームの文字セット選択を無効にできないことを意味します。*content-type* は位置揃えパターンを使用して指定できますが省略可能です。この属性を指定すると、印刷サービスは必要に応じて使用し、ファイルにフィルタをかけて印刷する方法を決定します。

2 つの例外がありますが、情報は任意の順序で指定できます。例外は、Alignment pattern (位置揃えパターン) (必ず最後に配置しなければなりません) と *comment* (コメント) (必ず Comment: プロンプトの行に続かなければなりません) です。コメントにキー句 (Page length、Page width など) で始まる行が含まれている場合は、キー句が行頭にならないように、その行を > 文字で始めます。先頭の > 文字は、コメントから除去されて表示されません。

すべての情報を与えなければならないわけではありません。表 7–7 の項目の値を指定しないときは、デフォルト値が割り当てられます。lpforms コマンドを実行する前に、新しいフォームに関して次の情報を収集してください。

表 7–7 フォームのデフォルト値

項目	デフォルト	説明
ページ長	66 行	フォームの長さ、または複数ページのフォームの場合は各ページの長さ。この情報は、行数でもインチ単位やセンチメートル単位でもかまわない
ページ幅	80 列	文字数、インチ数、またはセンチメートル数によるフォームの幅
ページ数	1	複数ページのフォームのページ数。LP 印刷サービスは、この数値と印刷フィルタ (利用できる場合) を使用して、位置揃えパターンを 1 つのフォームの長さに制限する。「位置揃えパターン」の説明を参照
行ピッチ	1 インチあたり 6 行	フォーム上の行間隔。これは「リーディング」とも呼ばれる。2 行間の間隔、つまりベースラインからベースラインまでの間隔を 1 インチまたは 1 センチあたりの行数で表す

表 7–7 フォームのデフォルト値 続く

項目	デフォルト	説明
文字ピッチ	1 インチあたり 10 文字	フォームに表示される文字間隔。文字の間隔を 1 インチまたは 1 センチあたりの文字数で表す
文字セット選択肢	任意	このフォームに使用しなければならない文字セット、印字ホイール、またはフォントカートリッジ。ユーザーは、このフォームを使用するときに独自の印刷要求に別の文字セットを選択できる。また、単一の文字セットのみを使用するように指示できる
リボンの色	任意	フォームを常に特定のカラーリボンで印刷しなければならない場合、LP 印刷サービスはどの色を使用すべきかを示す装着警告メッセージを表示できる
コメント	(デフォルトなし)	ユーザーがフォームを理解する上で参考になる任意の情報。たとえば、フォーム名、そのバージョン、用途、または使用上の制限を示すことができる
位置揃えパターン	(デフォルトなし)	LP 印刷サービスが 1 枚のブランクフォームを埋めるために使用するサンプルファイル。フォームを装着するときに、このパターンを印刷して正しく位置揃えすることができる。また、印刷サービスに印刷方法が認識されるように、このパターンの内容形式を定義することもできる

注 - LP 印刷サービスは、位置揃えパターン内の重要な情報にマスクをかけようとしません。小切手を位置揃えするときなど、サンプルフォームに重要な情報を印刷したくない場合は、該当するデータにマスクをかける必要があります。LP 印刷サービスは、スーパーユーザーまたは `lp` としてログインしたユーザー以外は読み取れないように、位置揃えパターンを安全な場所に格納します。

フォーム情報を収集し終わったら、フォームを `lpforms` コマンドに入力します。`lpforms` コマンドに入力する前にこの情報を編集できるように、まず、この情報を別のファイルに記録してください。そうすれば、プロンプトの後で個々の情報を入力しなくても、そのファイルを入力として使用できます。

▼ 新しいフォーム定義を作成する方法

1. プリンタサーバー上でスーパーユーザーまたは **lp** としてログインします。

2. フォーム定義ファイルを作成します。

   印刷フォームの作成方法の概要については、191ページの「新しいプリンタフォームの作成」を参照してください。プリンタ定義はテキストファイルに保存してください。

3. `lpadmin` コマンドを使用して、フォームを LP 印刷サービスに追加します。

```
# lpadmin -p printer-name -M -f form-name
```

4. フォームをプリンタサーバーに追加します。

   手順については、149ページの「フォームを追加する方法」を参照してください。

第 8 章

LP 印刷サービスの参照情報

この章では、LP 印刷サービスの内容説明を提供します。

印刷管理作業の手順については、次の章を参照してください。

- 第 4 章
- 第 5 章
- 第 6 章
- 第 7 章

LP 印刷サービス

「LP 印刷サービス」とは、ユーザーが作業を続けながらファイルを印刷できるようにするソフトウェアユーティリティの集合です。当初、印刷サービスは LP スプーラと呼ばれていました (LP はラインプリンタを意味しますが、現在ではレーザプリンタなど、他の多数のプリンタも含まれています。スプール (spool) は、system peripheral operation off-line の頭文字です)。

印刷サービスは、LP 印刷サービスソフトウェア、システム管理者が提供する印刷フィルタ、ハードウェア (プリンタ、システム、およびネットワーク接続) からなっています。

LP 印刷サービスの構造

この節では、LP 印刷サービスのディレクトリ構造、ファイル、ログ、およびコマンドについて説明します。

LP 印刷サービスのディレクトリ

LP 印刷サービスのファイルは、表 8–1 のように 7 つのディレクトリに分散されています。

表 8–1 LP 印刷サービスのディレクトリ

ディレクトリ	内容
/usr/bin	LP 印刷サービスのユーザーコマンド
/etc/lp	LP 構成ファイルの階層
/usr/share/lib	terminfo データベースディレクトリ
/usr/sbin	LP 印刷サービスの管理コマンド
/usr/lib/lp	LP デーモン。バイナリファイルと PostScript フィルタのディレクトリ。model ディレクトリ (標準プリンタインタフェースプログラムが入っている)

表 8–1　LP 印刷サービスのディレクトリ　続く

ディレクトリ	内容
/var/lp/logs	LP 動作のログ: lpsched.*n* – lpsched からのメッセージ。requests.*n* – 完了した印刷要求に関する情報
/var/spool/lp	ファイルが印刷待ち行列に入れられるスプーリングディレクトリ
/var/spool/print	LP 印刷サービスのクライアント側要求格納域

LP 印刷サービスの構成ファイル

スケジューラは、表 8–2 のように、/etc/lp ディレクトリに入っている LP 構成ファイルに構成情報を格納します。

注意 - 表 8–2 の構成ファイルはプライベートインタフェースで、将来のリリースでは変更される可能性があります。現在の場所にあるファイルに依存する、または現在使用している形式のデータに依存するソフトウェアを構築しないようにしてください。

表 8–2　/etc/lp ディレクトリの内容

ファイル	タイプ	説明
classes	ディレクトリ	lpadmin -c コマンドで与えたクラスを識別するファイル
fd	ディレクトリ	既存のフィルタの記述
filter.table	ファイル	印刷フィルタ照合テーブル
forms	ディレクトリ	各フォームのファイルを格納する場所。最初は、このディレクトリは空になっている
interfaces	ディレクトリ	プリンタインタフェースプログラムファイル

表 8–2 /etc/lp ディレクトリの内容 続く

ファイル	タイプ	説明
logs	/var/lp/logs へのリンク	印刷動作のログファイル
model	/usr/lib/lp/model へのリンク	標準プリンタインタフェースプログラム
printers	ディレクトリ	各ローカルプリンタのディレクトリ。各ディレクトリには、個々のプリンタの構成情報と警告ファイルが入っている
pwheels	ディレクトリ	印字ホイールまたはカートリッジファイル

これらの構成ファイルは、SunOS 4.1 リリースの /etc/printcap ファイルと同様の機能を提供します。

注 - これらのファイルの内容を確認できますが、直接編集しないでください。代わりに、lpadmin(1M) コマンドを使用して構成を変更します。変更内容は /etc/lp ディレクトリ内の構成ファイルに入力されます。lpsched デーモンは、構成ファイルを管理して構成します。

/etc/lp/printers ディレクトリには、システムに認識されるローカルプリンタごとに 1 つずつサブディレクトリが入っています。次の例は、プリンタ sparc1 と luna の /etc/lp/printers サブディレクトリを示します。

```
$ ls -l /etc/lp/printers
drwxrwxr-x 2 lp lp 512 Jan 23 23:53 luna
drwxrwxr-x 2 lp lp 512 Jan 11 17:50 sparc1
```

プリンタ固有の各ディレクトリ内では、次のファイルを使用してプリンタを記述できます。

ファイル名	説明
alert.sh	警告に応答して実行するシェル
alert.vars	警告変数
configuration	構成ファイル
users.deny	プリンタアクセスが拒否されるユーザーのリスト
comment	プリンタ記述

通常、プリンタ luna の構成ファイル /etc/lp/printers/luna/configuration は、次のようになっています。

```
Banner: on: Always
Content types: PS
Device: /dev/term/b
Interface: /usr/lib/lp/model/standard
Printer type: PS
Modules: default
```

terminfo データベース

/usr/share/lib ディレクトリには、terminfo データベースのディレクトリが入っており、そのディレクトリには多数のタイプの端末とプリンタに関する定義が入っています。LP 印刷サービスは、terminfo データベース内の情報を使用してプリンタを初期設定し、選択されたページサイズ、文字ピッチ、行ピッチ、および文字セットを設定し、一連のコードをプリンタに送ります。

各プリンタは、terminfo データベース内で短縮名を使用して識別されます。terminfo データベースの構造については、60ページの「プリンタタイプ」を参照してください。必要であれば、terminfo データベースにエントリを追加できますが、これは煩雑で時間のかかる作業です。詳細は、168ページの「サポートされていないプリンタの terminfo エントリを追加する」 を参照してください。

デーモンと LP 内部ファイル

/usr/lib/lp ディレクトリには、表 8–3 に示すような LP 印刷サービスに使用されるデーモンとファイルが入っています。

表 8–3　/usr/lib/lp ディレクトリの内容

ファイル	タイプ	説明
bin	ディレクトリ	印刷警告、低速フィルタ、待ち行列管理プログラムを生成するファイルが入っている
lpsched	デーモン	LP 印刷要求のスケジューリングを管理する
model	ディレクトリ	標準プリンタインタフェースプログラムが入っている
postscript	ディレクトリ	LP 印刷サービスで提供されているすべての PostScript フィルタプログラムが入っている。これらのフィルタには、フィルタの特性とその格納場所を LP 印刷サービスに指示する /etc/lp/fd ディレクトリ内の記述子ファイルが含まれる

LP 印刷サービスのログファイル

LP 印刷サービスは、次の 2 組のログファイルを管理します。

ログファイル名	説明
syslogd(1M)	/etc/syslog.conf の lpr.debug を設定して LP 印刷サービスロギングを有効にする
/var/spool/lp	印刷待ち行列に入っている現在の待ち行列のリスト
/var/lp/logs/requests	進行中の印刷要求の履歴

印刷待ち行列ログ

各システムのスケジューラは、ディレクトリ /var/spool/lp/tmp/*system* と /var/spool/lp/requests/*system* 内で印刷要求のログを保管します。各印刷要求には、(ディレクトリごとに 1 つずつ) 情報が入った 2 つのファイルがあります。/var/spool/lp/requests/*system* ディレクトリ内の情報には、スーパーユーザーまたは lp しかアクセスできません。/var/spool/lp/tmp/*system* 内の情報には、その要求を出したユーザー、スーパーユーザー、または lp しかアクセスできません。

次の例では、/var/spool/lp/tmp/terra ディレクトリの内容を示します。

```
$ ls /var/spool/lp/tmp/terra
20-0 21-0
terra$ cat 21-0
C 1
D slw2
F /etc/default/login
P 20
t simple
U tamiro
s 0x1000
```

これらのファイルは、印刷要求が待ち行列に入っている限り、そのディレクトリ内に残っています。要求が完了すると、ファイル内の情報は組み合わされ、ファイル /var/lp/logs/requests に追加されます。このファイルについては、次の節で説明します。

現在待ち行列に入っている印刷要求の状態を追跡したい場合は、/var/spool/lp/logs ログ内の情報を使用します。

履歴ログ

LP 印刷サービスは、lpsched と requests という 2 つのログファイルに印刷サービスの履歴を記録します。これらのログファイルは、/var/lp/logs ディレクトリに入っています。これらのログ内の情報を使用し、印刷の問題を診断して解決できます。次の例は、/var/lp/logs ディレクトリの内容を示します。

```
# cd /var/lp/logs
# ls
lpsched.1    requests    requests.2
lpsched     lpsched.2    requests.1
#
```

接尾辞 .1 と .2 が付いているファイルは、前日のログのコピーです。毎日、lp cron ジョブは lpsched ログファイルと requests ログファイルを消去して、2 日分のコピーを保管します。requests ログを消去する cron ジョブを変更するためのヒントについては、545ページの「crontab ファイルの作成と編集」を参照してください。

問題の解決に最も重要なのは、ローカル印刷要求に関する情報が入っている lpsched ログです。

requests ログには、完了して印刷待ち行列から消去された印刷要求に関する情報が入っています。印刷要求が終了すると、/var/spool/lp ログファイル内の情報が組み合わされ、/var/lp/logs/requests ログに追加されます。

requests ログの構造は単純なので、共通の UNIX シェルコマンドを使用してデータを抽出できます。要求は、出力された順番に要求 ID を示す行で区切って表示されます。区切り行の下の各行には、その行に入っている情報の種類を識別する 1 文字が付いています。各文字は、空白文字 1 つでデータから区切られます。

次の例は、requests ログの内容を示しています。

```
# pwd
/var/lp/logs
# tail requests.2
= slw2-20, uid 200, gid 200, size 5123, Tue Jun 17 10:16:10 MDT
1998
z slw2
C 1
D slw2
F /etc/motd
P 20
t simple
U irving
s 0x0100
#
```

表 8–4 に、文字コードとそれに対応する LP requests ログ内の行を示します。

表 8–4 LP requests ログ内の文字コード

文字	行の内容
=	区切り行。ユーザーの要求 ID、ユーザー ID (UID)、グループ ID (GID)、元の (フィルタを通す前の) ファイルサイズ、および要求が待ち行列に入れられた時刻が入っている
C	印刷部数
D	出力先のプリンタかクラス、または any
F	印刷されたファイル名。この行は印刷されたファイルごとに区切られ、各ファイルは表示された順序で印刷される
f	使用されたフォーム名

表 8–4 LP requests ログ内の文字コード 続く

文字	行の内容
H	特殊処理 resume、hold、または immediate
N	印刷要求が正常に完了したときに使用された警告のタイプ。ユーザーが電子メールで通知を受けたときは M タイプ、端末へのメッセージで通知を受けた場合は W タイプ
O	プリンタ依存の –o オプション (nobanner など)
P	印刷要求の優先順位
p	印刷されたページのリスト
r	ユーザーがファイルの「raw」処理を要求したとき (lp -r コマンド) に含まれる 1 文字の行
S	使用された文字セット、印字ホイール、またはカートリッジ
s	16 進数形式の各ビットの組み合わせで表される要求の結果。印刷サービスの内部では複数のビットが使用される。各ビットとその意味については、表 8–5 を参照
T	バナーページに印刷されるタイトル
t	ファイル内で見つかった内容形式
U	印刷要求を出したユーザー名
x	印刷要求に使用された低速フィルタ
Y	要求の印刷に使用された印刷フィルタの特殊モードのリスト
z	要求に使用されたプリンタ。要求がプリンタまたはプリンタクラスの待ち行列に入れられた場合や、要求が別の出力先に移動された場合は、このプリンタは出力先 (D 行) とは異なる

表 8–5 に、LP requests ログ内の結果コードとその説明を示します。

表 8–5 LP requests ログ内の結果コード

結果コード	説明
0x0001	要求は保留され再開を待機中
0x0002	低速フィルタを実行中
0x0004	低速フィルタを正常に完了
0x0008	要求はプリンタ上にある
0x0010	印刷を正常に完了
0x0020	要求は保留されユーザーによる変更を待機中
0x0040	要求は取り消し済み
0x0080	要求は次に印刷される
0x0100	フィルタ処理または印刷要求に失敗
0x0200	要求はリモートプリンタに転送中 (現在は使用されない)
0x0400	ユーザーに通知
0x0800	通知が動作中
0x1000	リモートシステムが要求を受け付け済み (現在は使用されない)
0x2000	管理者が要求を保留した
0x4000	プリンタのフィルタを変更しなければならなかった
0x8000	要求は一時的に停止された

スプーリングディレクトリ

印刷待ち行列に入れられたファイルは、印刷されるまで /var/spool/lp ディレクトリに格納されますが、それがわずか数秒の場合があります。表 8–6 は、/var/spool/lp ディレクトリの内容を示します。

表 8–6 /var/spool/lp ディレクトリの内容

ファイル	タイプ	説明
SCHEDLOCK	ファイル	スケジューラのロックファイル。スケジューラが停止し、再起動されない場合は、このファイルをチェックする
admins	ディレクトリ	/etc/lp へのリンク
bin	ディレクトリ	/usr/lib/lp/bin へのリンク
logs	リンク	完了した印刷要求のログが記録される ../lp/logs へのリンク
model	リンク	/usr/lib/lp/model へのリンク
requests	ディレクトリ	印刷要求が印刷されるまで記録される構成済みプリンタごとのサブディレクトリが入ったディレクトリ。ユーザーはこのログにアクセスできない
system	ディレクトリ	システムの印刷状態ファイル
temp	リンク	スプールされた要求が入っている /var/spool/lp/tmp/*hostname* へのリンク
tmp	ディレクトリ	印刷要求が印刷されるまでログが記録される構成済みの各プリンタのディレクトリ。既存の印刷要求を変更した場合も、このログに記録される

LP 印刷サービスのコマンド

表 8–7 に、頻繁に使用する LP 印刷サービスのコマンドを示します。1M コマンドを使用するには、スーパーユーザーまたは lp にならなければなりません。

表 8–7　LP 印刷サービスコマンド早見表

コマンド	機能
enable(1)	プリンタを使用可能にする
cancel(1)	印刷要求を取り消す
lp(1)	1 つ以上のファイルをプリンタに送る
lpstat(1)	LP 印刷サービスの状態を出力する
disable(1)	1 台以上のプリンタを無効にする
accept(1M)	印刷要求を特定の出力先の待ち行列に入れられるようにする
reject(1M)	印刷要求が特定の出力先の待ち行列に入れられないようにする
lpadmin(1M)	プリンタの構成を設定または変更する
lpfilter(1M)	フィルタの定義を設定または変更する
lpforms(1M)	あらかじめ印刷されたフォームを設定または変更する
lpadmin(1M)	フォームを取り付ける
lpmove(1M)	ある出力先から別の出力先に出力要求を移動する
lpsched(1M)	LP 印刷サービススケジューラを起動する
lpshut(1M)	LP 印刷サービススケジューラを停止する
lpusers(1M)	デフォルトの優先順位と、LP 印刷サービスのユーザーが要求できる優先順位の制限を設定または変更する

LP 印刷サービスの機能

LP 印刷サービスは、次の機能を実行します。

- ファイルを管理してローカル印刷要求をスケジュールする。

- ネットワーク要求を受け取り、スケジュールする。
- ファイルが正しく印刷されるように必要に応じてフィルタを通す。
- プリンタとインタフェースするプログラムを起動する。
- ジョブの状態を追跡する。
- プリンタに取り付けられたフォームを追跡する。
- 現在装着されている印字ホイールを追跡する。
- 新しいフォームや別の印字ホイールを取り付け、装着するよう警告を発する。
- 印刷問題に関する警告を発する。

ディレクトリ構造とコマンドについては、196ページの「LP 印刷サービスの構造」を参照してください。

LP によるファイルの管理とローカル印刷要求のスケジューリングの方法

LP 印刷サービスには、`lpsched` というスケジューラデーモンが組み込まれています。スケジューラデーモンは、プリンタの設定と構成に関する情報を使用して LP システムファイルを更新します。

また、`lpsched` デーモンは、ユーザーが要求をアプリケーションから出すかコマンド行から出すかに関係なく、図 8–1 のように、プリンタサーバー上のすべてのローカル印刷要求をスケジュールします。さらに、スケジューラはプリンタとフィルタの状態を追跡します。プリンタが要求を印刷し終わると、プリンタサーバー上の待ち行列に残っているものがあれば、スケジューラは次の要求をスケジュールします。

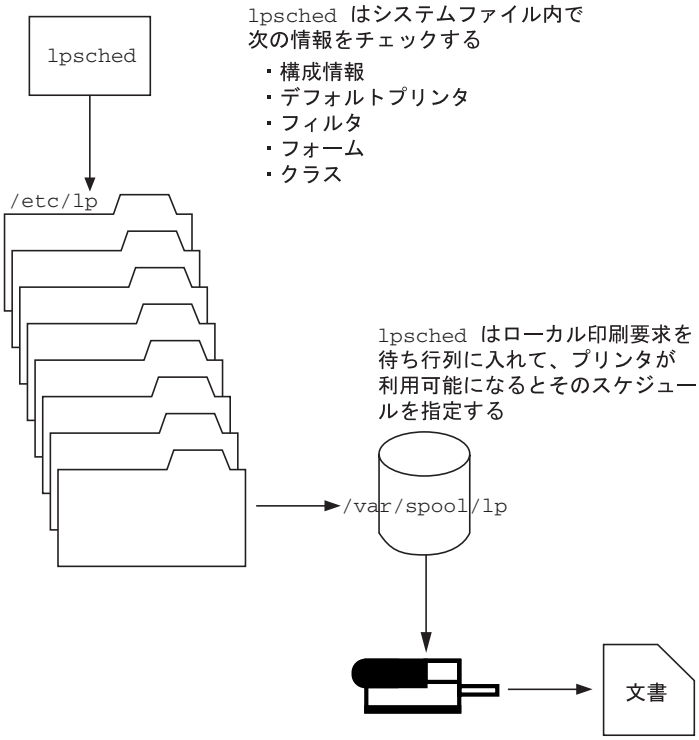


図 8–1　lpsched デーモンによるローカル印刷要求のスケジューリング方法

各プリンタサーバーは、LP スケジューラを 1 つだけ実行していなければなりません。スケジューラは、システムのブート時に (または実行レベル 2 を入力すると)、制御スクリプト /etc/rc2.d/S80lp によって起動されます。システムを再起動しなくても、/usr/lib/lp/lpshut コマンドを使用してスケジューラを停止し、lpsched コマンドを使用して再起動できます。各システムのスケジューラは、lp コマンドによってシステムに出された要求を管理します。

ネットワーク印刷要求のスケジューリング

各印刷クライアントは、ネットワーク経由で直接、プリンタサーバーと通信します。通信は、要求コマンド (lp、lpstat、cancel、lpr、lpq、または lprm) とプリンタサーバー上の印刷サービス間で行われます。これによって、クライアント専用システムにおける印刷システムのオーバーヘッドが軽減され、その結果、拡張性、性能、およびデータの正確性が向上します。

プリンタサーバーは、現在、インターネットサービスデーモン (inetd) で印刷要求を待機します。ネットワークから印刷サービスへ要求があると、inetd は「プロト

コルアダプタ」と呼ばれるプログラム (in.lpd) を起動します。プロトコルアダプタは、印刷要求を翻訳して、印刷スプーラに送信し、その結果を要求依頼元に戻します。プロトコルアダプタは要求の発生時に起動して、ネットワーク要求の処理が完了すると終了します。これによって、印刷のためのアイドル状態のシステムのオーバーヘッドが解消されます。また、Solaris の以前の印刷機能にあった、ネットワークに接続された印刷サポート用の余分なシステム構成が不要になります。

印刷ファイルにフィルタを適用する

印刷フィルタは、待ち行列内のファイルの内容をある形式から別の形式に変換するプリンタサーバー上のプログラムです。

印刷フィルタは、必要に応じて単純なものでも複雑なものでもかまいません。SunOS リリースの場合は、出力先プリンタが PostScript 形式へのデータ変換を必要とするほとんどの PostScript 印刷状況に対応する印刷フィルタが、/usr/lib/lp/postscript ディレクトリに組み込まれています。PostScript 以外のプリンタ用のフィルタが必要な場合は、そのフィルタを作成し、目的のシステムに追加しなければなりません。

一連の「印刷フィルタ記述子ファイル」が /etc/lp/fd ディレクトリに用意されています。これらの記述子ファイルは、フィルタの特性 (高速フィルタや低速フィルタなど) を記述し、フィルタプログラム (/usr/lib/lp/postscript/postdaisy など) を指します。

プリンタインタフェースプログラムの機能

LP 印刷サービスは、オペレーティングシステムの他の部分と情報をやり取りします。また、標準プリンタインタフェースプログラムを使用して、次の作業を実行します。

- 必要に応じてプリンタポートを初期化する。標準プリンタインタフェースプログラムは、stty コマンドを使用してプリンタポートを初期化する。
- プリンタを初期化する。標準プリンタインタフェースプログラムは、terminfo データベースと TERM シェル変数を使用して、適切な制御シーケンスを見つける。
- 必要に応じてバナーページを印刷する。
- 印刷要求で指定された部数だけ印刷する。

LP 印刷サービスは、別のプログラムが指定されなければ、標準インタフェースプログラム (/usr/lib/lp/model ディレクトリに入っています) を使用します。独自のインタフェースプログラムを作成することもできますが、独自のプログラムがプリンタへの接続を終了させないことや正しいプリンタの初期設定を妨げないことを確認しなければなりません。

lpsched デーモンによる印刷ジョブ状態の確認

プリンタサーバー上と印刷クライアント上の lpsched デーモンは、処理する印刷要求ごとに 1 つずつログを保管し、印刷処理中に発生するエラーを記録します。このログは /var/lp/logs/lpsched ファイルに保管されます。毎晩、lp cron ジョブは /var/lp/logs/lpsched を新しい lpsched.*n* ファイル名に変更し、新しいログファイルを開始します。エラーが発生したり、印刷待ち行列からジョブが消えたりした場合は、ログファイルを使用して lpsched で実行された印刷ジョブへの処理を判別できます。

ログファイルの消去

/var/lp/logs ディレクトリ内の lpsched および requests ログファイルは、情報が追加されるにつれて大きくなります。LP 印刷サービスは、デフォルトの cron ジョブを使用してログファイルを消去します。lp cron ジョブは /var/spool/cron/crontabs/lp ファイルに入っています。このジョブはログファイルの内容を定期的に移動します。ログの内容は *log*.1 に移動され、*log*.1 の内容は *log*.2 に移動されるというようになります。*log*.2 が上書きされると、その内容は失われます (つまり、*log*.1 の前の内容に置き換えられます)。

```
# pwd
/var/lp/logs
# tail requests
s 0x1010
= slw2-20, uid 200, gid 200, size 5123, Mon Jun 16 12:27:33 MDT 1997
z slw2
C 1
D slw2
F /etc/motd
P 20
t simple
U irving
s 0x1010
```

(続く)

```
#
```

▼ プリンタ要求のログの交換間隔を変更する方法

Solaris 2.6 リリースより、プリンタサーバー上の requests ログファイルは、毎日ではなく、毎週交換されることになりました。プリンタサーバーの使用頻度が高い場合は、交換間隔を毎日に再設定できます。

1. プリンタサーバー上でスーパーユーザーまたは lp になります。

2. EDITOR 環境変数を設定します。

```
# EDITOR=vi
# export EDITOR
```

3. lp の crontab ファイルを編集します。

```
# crontab -e lp
```

4. requests ログファイルの交換間隔を指定するファイルの先頭行を、毎日曜日 (0) から、毎日を示すアスタリスク (*) に変更します。

```
13 3 * * * cd /var/lp/logs; if [ -f requests ]; then if
[ -f requests.1 ]; then /bin/mv requests.1 requests.2; fi; /usr/bin/cp
requests requests.1; >requests; fi
```

5. ファイルを保存して、終了します。

ローカル印刷の処理スケジュール

図 8–2 に、ユーザーがローカルプリンタ上に PostScript ファイルを印刷する要求を出したときに実行される処理を示します。ローカルプリンタとは、ユーザーのシステムに接続されたプリンタです。すべての処理はローカルシステムによって実行されます。ただし、印刷要求は、クライアントとサーバーが異なるシステムにある場合と同じ経路をたどります。要求は常に同じ経路をたどり、クライアントからサーバーに流れます。

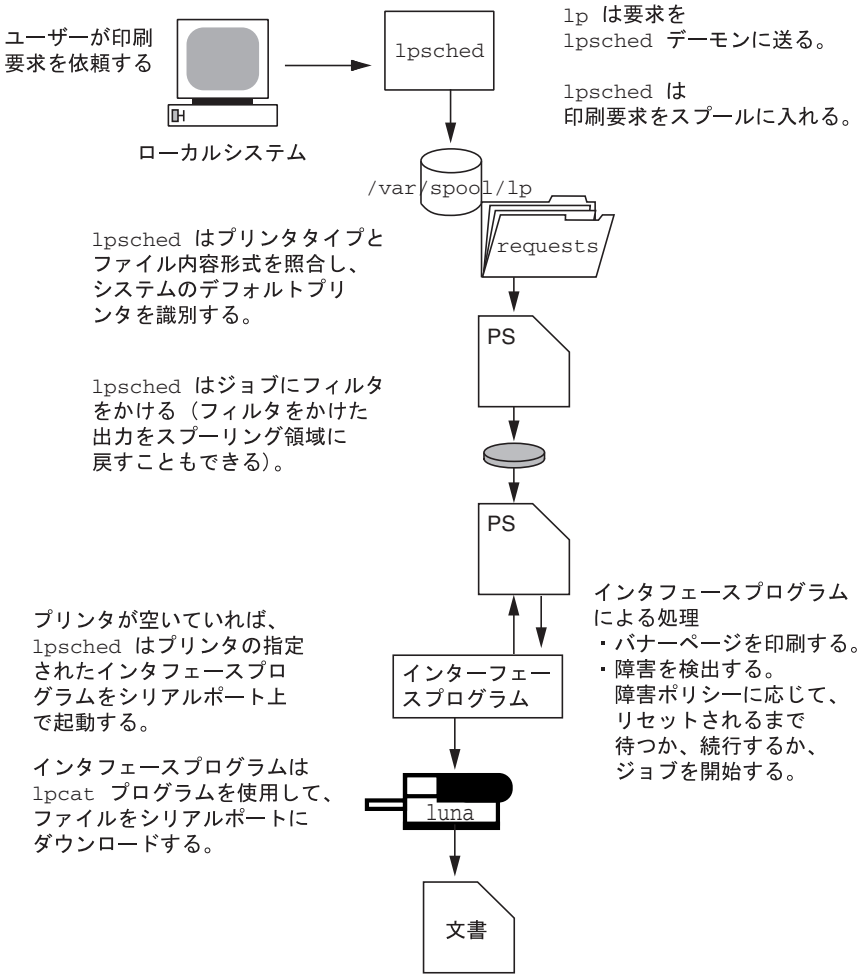


図 8–2　ローカルでの印刷処理

リモート印刷の処理スケジュール

図 8–3 に、SunOS 5.8 印刷クライアント上のユーザーが SunOS 4.1 プリンタサーバーに印刷要求を出したときに実行される処理を示します。このコマンドは接続を開いて、プリンタサーバーとの通信を直接処理します。

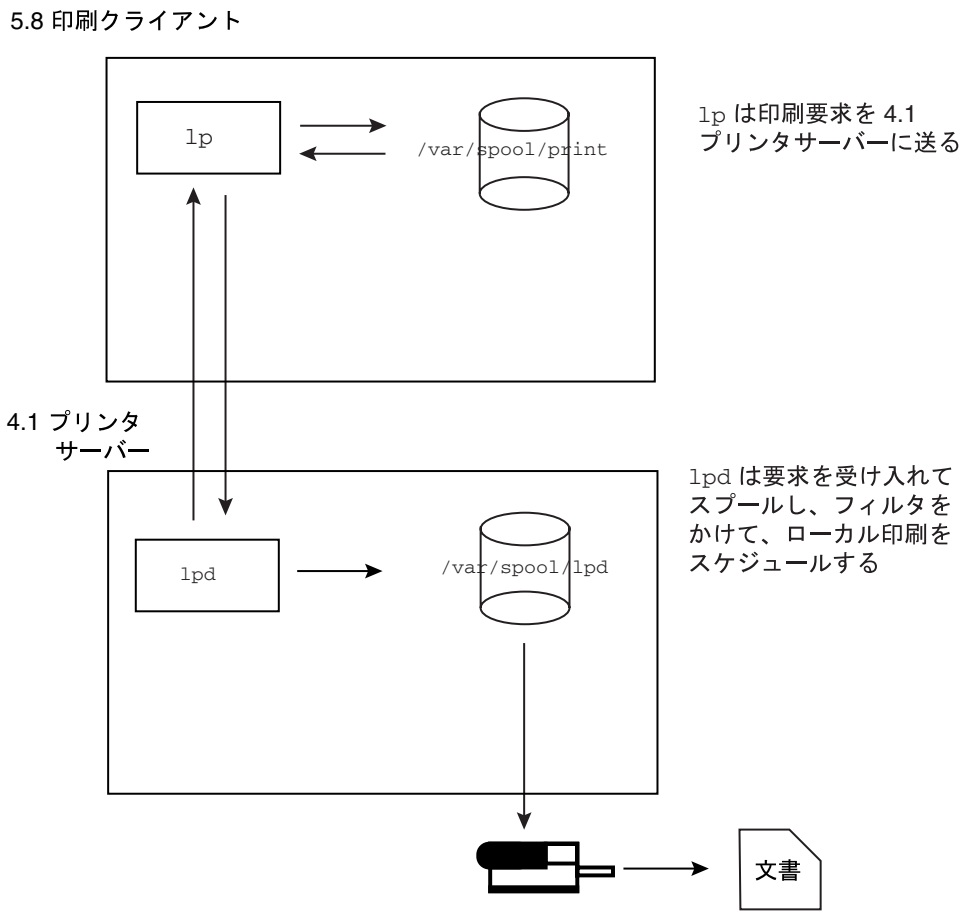


図 8–3　SunOS 5.8 印刷クライアントと SunOS 4.1 プリンタサーバー間のネットワーク印刷

図 8–4 に、SunOS 5.8 プリンタサーバーに印刷要求を出す SunOS 4.1 印刷クライアントを示します。`lpd` デーモンは、印刷要求のローカル部分とプリンタサーバーへの接続を処理します。プリンタサーバー上のネットワーク待機プロセス `inetd` は、ネットワーク印刷要求を待って、プロトコルアダプタを起動して要求を処理しま

す。プロトコルアダプタは lpsched デーモンと通信し、このデーモンがプリンタサーバー上で要求を処理します。

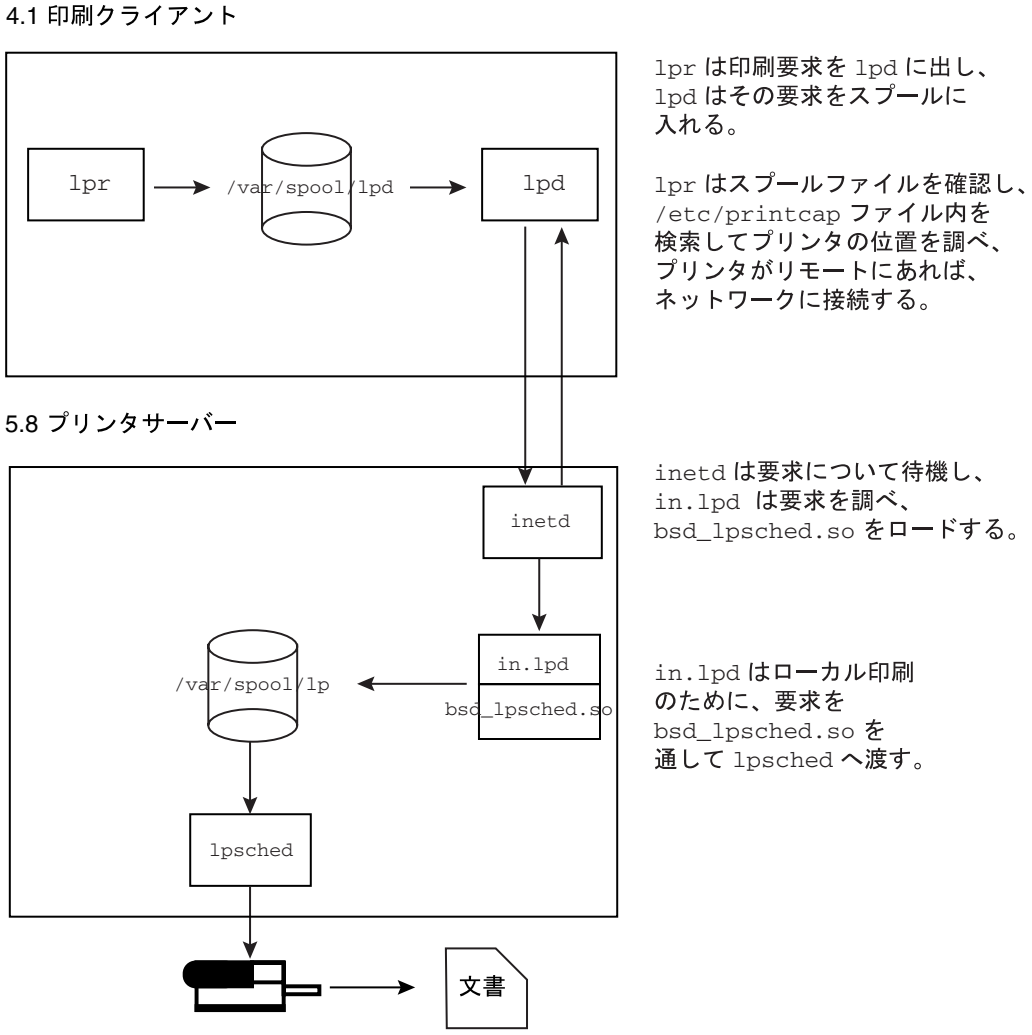


図 8–4　SunOS 4.1 印刷クライアントと SunOS 5.8 プリンタサーバー間のネットワーク印刷

図 8–5 に、SunOS 5.8 印刷クライアントのユーザーが SunOS 5.8 プリンタサーバーに印刷要求を出したときに実行される処理を示します。印刷クライアント上の印刷コマンドは、プリンタサーバーと直接通信することにより、各印刷要求のローカル部分を処理します。

プリンタサーバー上の inetd プロセスは、ネットワーク上の印刷要求を監視し、プロトコルアダプタを起動して、プリンタサーバー上の lpsched デーモンと通信します。このデーモンが印刷要求を処理します。

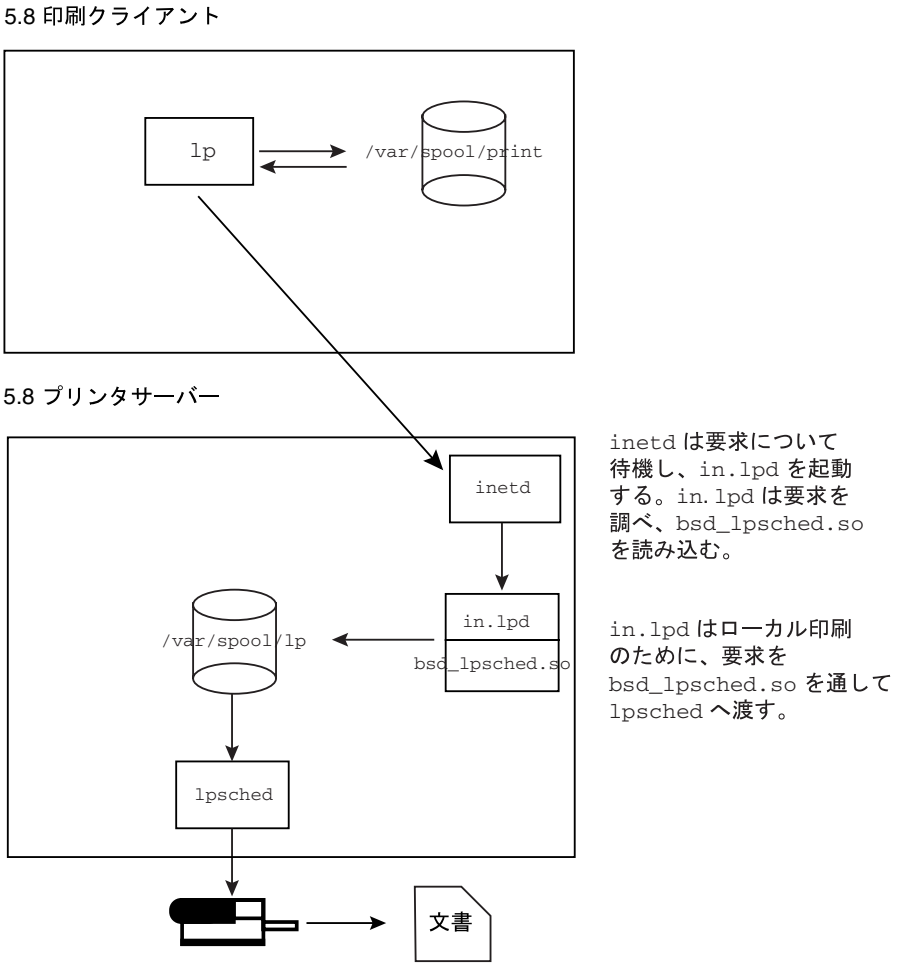


図 8–5　SunOS 5.8 印刷クライアントと SunOS 5.8 プリンタサーバー間のネットワーク印刷

第 9 章

リモートシステムの利用

ここでは、Solaris 環境でリモートシステムを利用する方法について説明します。次の章が含まれます。

第 10 章	`rlogin`、`ftp`、`rcp`、およびリモートによる承認と認証を使用してリモートシステムを利用する手順を説明します。

第 10 章

リモートシステムの利用

この章では、リモートシステムにログインしてそのファイルを利用するために必要なすべての作業について説明します。この章で説明する手順は次のとおりです。

リモートシステムとは

この章では、リモートシステムとは、図 10–1 のように物理ネットワークによってローカルシステムに接続され、TCP/IP 通信用に構成されたワークステーションまたはサーバーであると想定します。

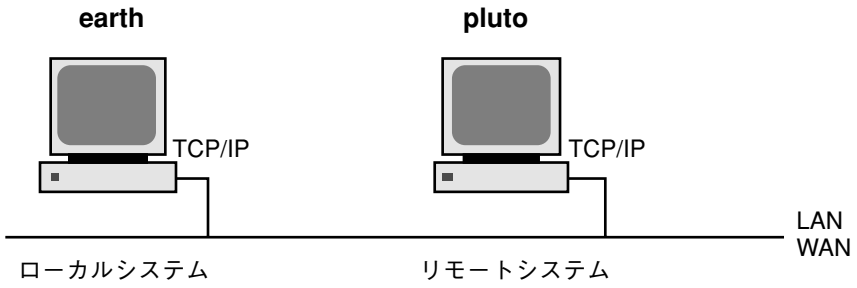


図 10–1　リモートシステム

Solaris リリースのシステム上では、TCP/IP は起動時に自動的に構成されます。詳細は、『*Solaris* のシステム管理 *(*第 *3* 巻*)*』を参照してください。

リモートシステムへのログイン（rlogin）

rlogin コマンドを使用すると、リモートシステムにログインできます。ログインした後は、リモートファイルシステム内で移動し、その内容を (リモートシステムによる承認にしたがって) 操作したり、ファイルをコピーしたり、リモートコマンドを実行したりできます。

ログイン先のシステムがリモートドメインに所属している場合は、システム名にドメイン名を追加してください。次の例では、SOLAR はリモートドメイン名です。

```
rlogin pluto.SOLAR
```

また、Control-d と入力すると、リモートログイン処理をいつでも中断できます。

リモートログイン (rlogin) の認証

rlogin 処理の認証 (ログインするユーザーの確認処理) は、リモートシステムまたはネットワーク環境で実行されます。

この 2 つの認証形式の主な違いは、要求される対話操作と、認証の確立方法にあります。リモートシステムがユーザーを認証しようとする場合に、/etc/hosts.equiv または .rhosts ファイルを設定していなければ、パスワードの入力を促すプロンプトが表示されます。ネットワークがユーザーを認証しようとする場合は、ユー

ザーはすでにネットワークに認識されているので、パスワードプロンプトは表示されません。図 10–2 は、リモートログインの認証処理を簡単に表したものです。

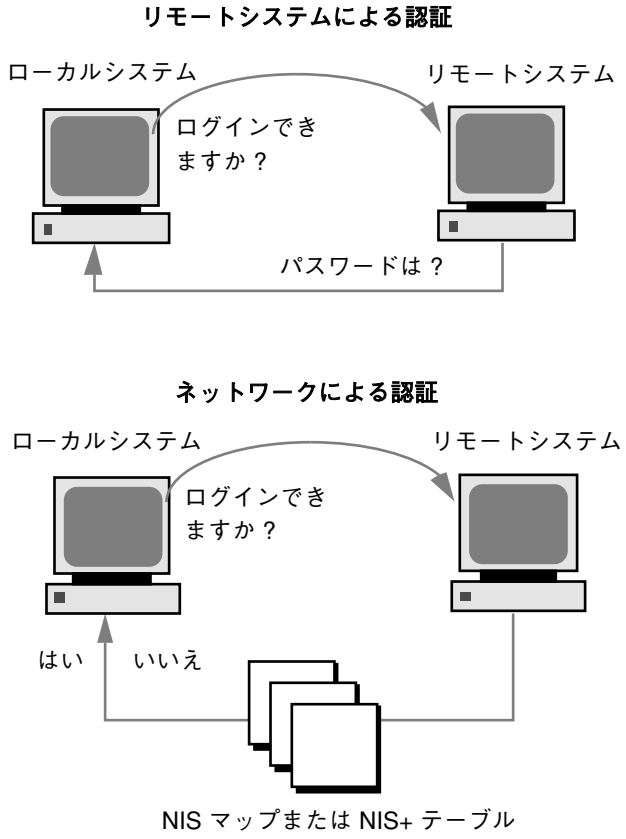


図 10–2　リモートログイン (rlogin) の認証

リモートシステムは、ユーザーを認証するときに、特に次のどちらかの場合にはローカルファイル内の情報を利用します。

- システム名とユーザー名がリモートシステムの /etc/hosts.equiv ファイルに入っている場合
- システム名とユーザー名が、リモートユーザーのホームディレクトリの下にある .rhosts ファイルに入っている場合

ネットワークによる認証は、次のどちらかの場合に利用されます。

- ローカルネットワーク情報サービスとオートマウンタを使用して設定された「信頼できるネットワーク環境」がある場合
- リモートシステムの /etc/nsswitch.conf ファイルが指定するネットワーク情報サービスがユーザーに関する情報を持っている場合

注 - 通常は、ネットワークによる認証がシステムによる認証より優先されます。

/etc/hosts.equiv ファイル

/etc/hosts.equiv ファイルには、リモートシステムの「信頼される (trusted) ホスト」が 1 行に 1 つずつ入っています。ユーザーがこのファイルに含まれるホストから (rlogin を使用して) リモートログインしようとする場合、リモートシステムがそのユーザーのパスワードエントリにアクセスできれば、ユーザーはパスワードを入力しなくてもログインできます。

典型的な hosts.equiv ファイルの構造は次のとおりです。

```
host1
host2 user_a
+@group1
-@group2
```

上記の host1 のように、ホスト名だけのエントリであれば、そのホストが信頼されているため、そのマシン上のユーザーも信頼できることを意味します。

この例の第 2 のエントリのようにユーザー名も含まれていると、その指定されたユーザーがアクセスしようとする場合にのみ、そのホストが信頼されます。

グループ名の先頭にプラス記号 (+) が付いている場合は、そのネットグループ内のすべてのマシンが信頼されていることを意味します。

グループ名の先頭にマイナス記号 (–) が付いている場合は、そのネットグループ内には信頼できるマシンがないことを意味します。

/etc/hosts.equiv ファイルを使用する場合のセキュリティの問題

/etc/hosts.equiv ファイルにはセキュリティ上の問題があります。/etc/hosts.equiv ファイルをシステム上で管理する場合は、ネットワーク内で信頼されるホストのみを含めるようにしてください。このファイルには、別の

ネットワークに所属するホストや、公共で利用されるマシンを含めないでください(たとえば、端末室に設置されているホストは含めないでください)。

このような信頼できないホストを含んでいると、セキュリティの面で重要な問題を引き起こす可能性があります。/etc/hosts.equiv ファイルを正しく構成されたファイルと置き換えるか、ファイルを削除してください。

/etc/hosts.equiv ファイルに + のみの 1 行しか入っていない場合は、認識されているすべてのホストが信頼されることを示します。

.rhosts ファイル

.rhosts ファイルは、/etc/hosts.equiv ファイルに対応するユーザー用のファイルです。このファイルには、通常、ホストとユーザーの組み合わせのリストが入っています。このファイルにホストとユーザーの組み合わせが含まれている場合、そのユーザーには、パスワードを入力しなくても、そのホストからリモートログインする許可が与えられます。

.rhosts ファイルはユーザーのホームディレクトリの一番上のレベルに置かれていなければなりません。サブディレクトリに置かれている .rhosts ファイルは参照されません。

ユーザーは、各自のホームディレクトリ内で .rhosts ファイルを作成できます。.rhosts ファイルを使用することによって、/etc/hosts.equiv ファイルを使用しなくても、異なるシステムのアカウント間で信頼できるアクセスを行うことができます。

.rhosts ファイルを使用する場合のセキュリティの問題

.rhosts ファイルにはセキュリティ上、重大な問題があります。/etc/hosts.equiv ファイルはシステム管理者の制御下にあり、効率よく管理できますが、誰でも .rhosts ファイルを作成して、システム管理者が知らないうちに自分が選んだユーザーにアクセス権を与えることができます。

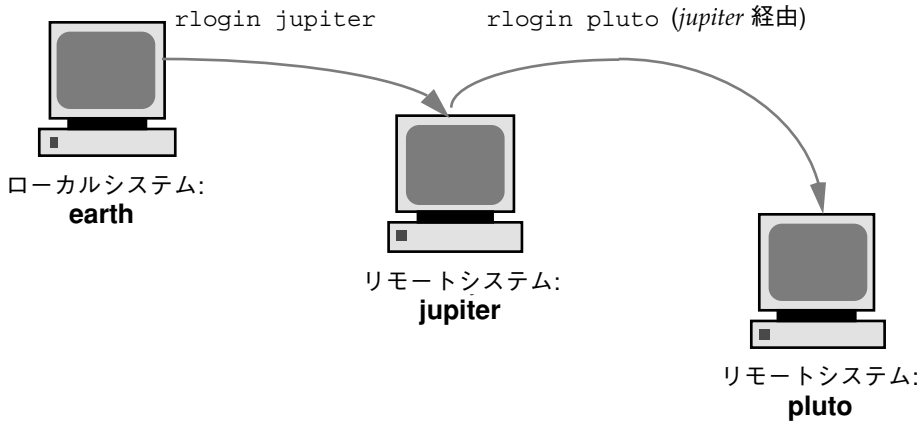
すべてのユーザーのホームディレクトリが 1 台のサーバー上にあって、特定のユーザーだけがそのサーバーに対してスーパーユーザーのアクセス権を持っている場合、ユーザーが .rhosts ファイルを使用できないようにするためには、スーパーユーザーとして、空の .rhosts ファイルを各ユーザーのホームディレクトリに作成します。次に、このファイルのアクセス権を 000 に変更します。こうしておけば、スーパーユーザーでも、そのファイルを変更することが難しくなります。これにより、ユーザーが .rhosts を無責任に使用することによって生じるセキュリ

ティ問題を防ぐことができます。ただし、ユーザーが自分のホームディレクトリへの実効パスを変更できる場合、この方法は何の解決にもなりません。

.rhosts ファイルを確実に管理する唯一の方法は、それを完全に使用できないようにすることです。詳細は、227ページの「.rhosts ファイルを検索して削除する方法」を参照してください。システム管理者は、システムを頻繁にチェックして、このポリシーに対する違反を調べることができます。このポリシーに対する例外は、root アカウントです。ネットワークのバックアップや他のリモートサービスを実行するには、.rhosts ファイルが必要な場合があります。

リモートログインのリンク

システムが正しく構成されていれば、リモートログインをリンクできます。次の例では、earth 上のユーザーが jupiter にログインし、そこから pluto にログインします。



もちろん、このユーザーは jupiter からログアウトして pluto に直接ログインすることもできますが、このリンク方法の方が便利です。

パスワードを入力せずにリモートログインをリンクするには、/etc/hosts.equiv または .rhosts を正しく設定しておかなければなりません。

直接リモートログインと間接リモートログイン

rlogin コマンドを使用すると、図 10–3 のように、リモートシステムに直接または間接的にログインできます。

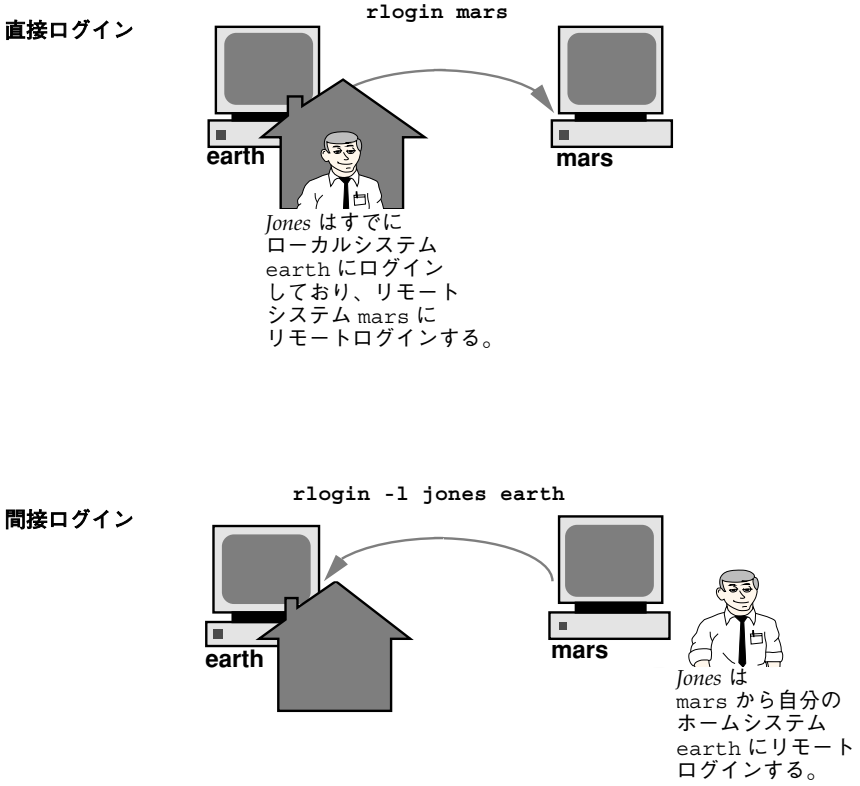


図 10–3　直接ログインと間接ログイン

直接リモートログインは、デフォルトのユーザー名、つまり、その時点でローカルシステムにログインしているユーザーのユーザー名を使用して行われます。これは、最も一般的なリモートログイン形式です。

間接リモートログインは、リモートログイン処理中に別のユーザー名を入力することによって行います。これは、一時的に借りているワークステーションから行うタイプのリモートログインです。たとえば、同僚のオフィスにいるときに自分のホームディレクトリ内でファイルを調べたい場合は、同僚のシステムから自分のシステムにリモートログインできますが、自分のユーザー名を入力して間接リモートログインを実行することになります。

表 10–1 は、直接ログインや間接ログインと認証方式の依存関係を示しています。

表 10–1 ログイン方式と認証方式 (rlogin) の依存関係

ログイン方式	ユーザー名の提供	認証	パスワード
直接	システム	ネットワーク	なし
		システム	必要
間接	ユーザー	ネットワーク	なし
		システム	必要

リモートログイン後の処理

リモートシステムにログインするときに、rlogin コマンドはホームディレクトリを見つけようとします。ホームディレクトリが見つからなければ、リモートシステムのルートディレクトリ (/) が割り当てられます。たとえば、次のようになります。

```
No directory! Logging in with home=/
```

ただし、 rlogin コマンドがホームディレクトリを見つけると、.cshrc ファイルと .login ファイルを生成します。したがって、リモートログイン後は、プロンプトが標準ログインプロンプトになり、現在のディレクトリはローカルにログインするときと同じになります。

たとえば、通常のプロンプトにシステム名と作業用ディレクトリが表示される場合と、ログイン時の作業用ディレクトリがホームディレクトリの場合、ログインプロンプトは次のようになります。

```
earth(/home/smith):
```

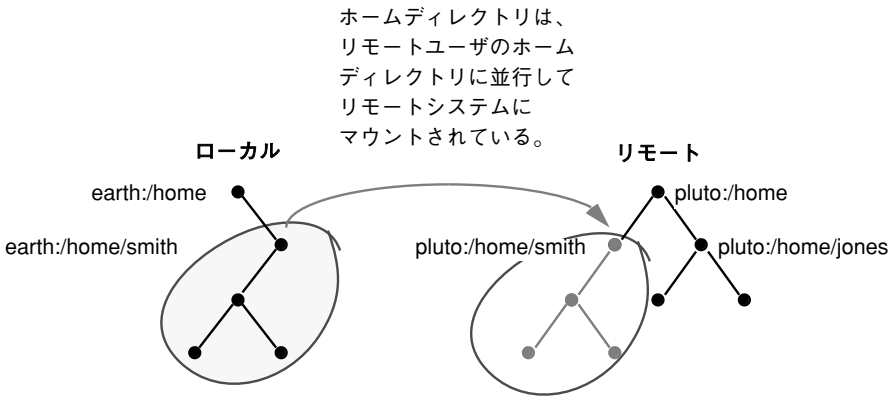
リモートシステムにログインすると、同じようなプロンプトが表示され、rlogin コマンドをどのディレクトリから入力したかに関係なく、作業用ディレクトリがホームディレクトリになります。

```
earth(/home/smith): rlogin pluto
.
```

(続く)

```
.
.
pluto(/home/smith):
```

唯一の違いは、プロンプトの先頭にローカルシステムではなくリモートシステムの名前が表示されることです。その場合、リモートファイルシステムは、次の図のようにホームディレクトリと並んで存在します。



cd を使って /home に移動してから ls を実行すると、次のように表示されます。

```
earth(home/smith): cd ..
earth(/home): ls
smith  jones
```

▼ .rhosts ファイルを検索して削除する方法

1. スーパーユーザーになります。

2. find(1) コマンドを使用し、.rhosts ファイルを検索して削除します。

```
# find home-directories -name .rhosts -print -exec rm{}
```

home-directories	ユーザーのホームディレクトリがあるディレクトリへのパス。複数のパスを指定すると、複数のホームディレクトリを一度に検索できる
-name .rhosts	ここでは .rhosts を指定する
-print	現在のパス名を出力する
-exec rm {} \;	指定したファイル名に一致するファイルすべてに、rm コマンドを適用するように find コマンドに伝える

find コマンドは、指定したディレクトリから始めて .rhosts というファイルを検索します。ファイルが見つかると、そのパスを画面に出力して削除します。

例 —.rhosts ファイルを検索して削除する

次の例では、/export/home ディレクトリ内で、すべてのユーザーのホームディレクトリ内の .rhosts ファイルを検索し削除します。

```
# find /export/home -name .rhosts -print | xargs -i -t rm{}
```

▼ リモートシステムが動作中かどうかを調べる方法

ping コマンドを使用して、リモートシステムが動作中かどうかを調べます。

```
$ ping system-name | ip-address
```

system-name	リモートシステム名
ip-address	リモートシステムの IP アドレス

ping コマンドは、次の 3 つのメッセージのどれかを返します。

状態メッセージ	意味
system-name is alive	このシステムにはネットワーク経由でアクセスできる
ping:unknown host *system-name*	未知のシステム名
ping:no answer from *system-name*	システムは認識されるが、現在は動作していない

ping を実行した対象のシステムが別のドメイン内にある場合は、出力メッセージにルーティング情報も含まれることがありますが、これは無視してかまいません。

ping コマンドのタイムアウトは 20 秒です。つまり、20 秒以内に応答がなければ、第 3 のメッセージを返します。 *time-out* 値を秒単位で入力すると、ping の待ち時間を増減させることができます。

```
$ ping system-name | ip-address  time-out
```

詳細は、ping(1M) のマニュアルページを参照してください。

▼ リモートシステムにログインしているユーザーを検索する方法

rusers(1) コマンドを使用して、リモートシステムにログインしているユーザーを検索します。

```
$ rusers [-l] remote-system-name
```

rusers　(オプションなし) システム名と、root など現在ログインしているユーザー名を表示する

-l　ユーザーのログインウィンドウ、ログイン日時、ログインしている時間、ユーザーのログイン元のリモートシステム名など、各ユーザーの詳細な情報を表示する

例 — リモートシステムにログインしているユーザーを検索する

次の例は、rusers の短い形式の出力を示しています。

```
$ rusers pluto
pluto    smith  jones
```

次の例では、rusers の長い形式の出力は、2 人のユーザーがリモートシステム starbug にログインしていることを示します。第 1 のユーザーは 9 月 10 日にシステムコンソールからログインし、ログイン時間は 137 時間 15 分でした。第 2 のユーザーはリモートシステム mars から 9 月 14 日にログインしました。

```
$ rusers -l starbug
root         starbug:console           Sep 10 16:13  137:15
rimmer       starbug:pts/0             Sep 14 14:37         (mars)
```

▼ リモートシステムにログインする方法 (rlogin)

rlogin(1) コマンドを使用してリモートシステムにログインします。

```
$ rlogin [-l user-name] system-name
```

rlogin	(オプションなし) リモートシステムに直接、つまり現在のユーザー名を使用してログインする
-l *user-name*	リモートシステムに間接的に、つまり入力するユーザー名を使用してログインする

ネットワークがユーザーを認証しようとする場合には、パスワードを求めるプロンプトは表示されません。リモートシステムがユーザーを認証しようとする場合は、パスワードの入力を求めるプロンプトが表示されます。

操作が成功すると、rlogin コマンドは、そのシステムへの前回のリモートログイン、リモートシステム上で動作中のオペレーティングシステムのバージョン、ホームディレクトリに未処理のメールがあるかどうかに関して、簡潔な情報を表示します。

例 — リモートシステムにログインする (rlogin)

次の例は、pluto へ直接リモートログインした出力結果を示しています。このユーザーはネットワークから認証されています。

```
$ rlogin starbug
Last login: Mon Jul 12 09:28:39 from venus
Sun Microsystems Inc.   SunOS 5.8       February 2000
starbug:
```

次の例は、pluto へ間接リモートログインした出力結果を示しています。この場合、ユーザーはリモートシステムから認証されています。

```
$ rlogin -l smith pluto
password: user-password
Last login: Mon Jul 12 11:51:58 from venus
Sun Microsystems Inc.   SunOS 5.8       February 2000
starbug:
```

▼ リモートシステムからログアウトする方法 (exit)

exit(1) コマンドを使用して、リモートシステムからログアウトします。

```
$ exit
```

例 — リモートシステムからログアウトする (exit)

次の例は、ユーザー smith がシステム pluto からログアウトする様子を示しています。

```
$ exit
pluto% logout
Connection closed.
earth%
```

リモートシステムへのログイン (ftp)

ftp コマンドは、インターネットのファイル転送プロトコルへのユーザーインタフェースを提供します。このユーザーインタフェースはコマンドインタプリタと呼ばれ、リモートシステムにログインし、そのファイルシステムについて様々な処理を実行できるようにします。基本操作については、表 10–2 を参照してください。

rlogin や rcp と比較して ftp の主な利点は、ftp の場合はリモートシステムで UNIX が実行されていなくてもかまわないことです (ただし、リモートシステムは TCP/IP 通信を利用できることが必要です)。これに対して、rlogin の方が ftp よりも多数のファイル操作コマンドを使用できます。

リモートログインの認証 (ftp)

ftp リモートログインの認証は、次のどちらかの方法で行われます。

- パスワードエントリをリモートシステムの /etc/passwd ファイル、またはそれに相当するネットワーク情報サービスマップかテーブルに追加する。
- リモートシステム上で anonymous ftp アカウントを確立する。

重要な ftp コマンド

表 10–2 重要な ftp コマンド

コマンド	機能
ftp	ftp コマンドインタプリタにアクセスする
ftp *remote-system*	リモートシステムへの ftp 接続を確立する。詳細は、233ページの「ftp によりリモートシステムへ接続する方法」を参照
open	コマンドインタプリタからリモートシステムにログインする
close	リモートシステムからログアウトしてコマンドインタプリタに戻る
bye	ftp コマンドインタプリタを終了する
help	すべての ftp コマンドを表示するか、コマンド名が指定されている場合は、コマンドの機能に関する簡単な説明を表示する
reset	リモートの ftp サーバーとコマンド応答シーケンスの同期をとり直す
ls	リモートの作業用ディレクトリの内容を表示する
pwd	リモートの作業用ディレクトリ名を表示する
cd	リモートの作業用ディレクトリを変更する

表 10–2　重要な ftp コマンド　続く

コマンド	機能
lcd	ローカルの作業用ディレクトリを変更する
mkdir	リモートシステム上でディレクトリを作成する
rmdir	リモートシステム上でディレクトリを削除する
get、mget	リモートの作業用ディレクトリからローカルの作業用ディレクトリに 1 つ以上のファイルをコピーする
put、mput	ローカルの作業用ディレクトリからリモートの作業用ディレクトリに 1 つ以上のファイルをコピーする
delete、mdelete	リモートの作業用ディレクトリから 1 つ以上のファイルを削除する

詳細は、ftp(1) のマニュアルページを参照してください。

▼ ftp によりリモートシステムへ接続する方法

1. ftp 認証を持っていることを確認します。

   232ページの「リモートログインの認証 (ftp)」で説明しているように、ftp 認証を持っている必要があります。

2. ftp コマンドを使用してリモートシステムへ接続します。

```
$ ftp remote-system
```

   接続に成功すると、確認メッセージとプロンプトが表示されます。

3. ユーザー名を入力します。

```
Name (remote-system:user-name): user-name
```

4. プロンプトが表示されたら、パスワードを入力します。

```
331 Password required for user-name:
Password: password
```

アクセス中のシステムに anonymous ftp アカウントが確立されている場合は、パスワードの入力を求めるプロンプトが表示されません。ftp インタフェースがパスワードを受け入れると、確認メッセージと (ftp>) プロンプトを表示します。

これで、help など、ftp インタフェースから提供されるどのコマンドでも使用できます。主なコマンドについては、表 10–2 を参照してください。

例 — ftp によりリモートシステムへ接続する

次の ftp セッションは、リモートシステム pluto 上でユーザー smith によって確立されました。

```
$ ftp pluto
Connected to pluto.
220 pluto FTP server (SunOS 5.8) ready.
Name (pluto:smith): smith
331 Password required for smith:
Password: password
230 User smith logged in.
ftp>
```

▼ リモートシステムとの ftp 接続を終了する方法

bye コマンドを使用して、リモートシステムとの ftp 接続を終了します。

```
ftp> bye
221 Goodbye.
earth%
```

接続を終了するメッセージに続いて、通常のシェルプロンプトが表示されます。

▼ リモートシステムからファイルをコピーする方法 (ftp)

1. リモートシステムからファイルをコピーしたい、ローカルシステム上のディレクトリに変更します。

```
$ cd target-directory
```

2. ftp により接続します。

233ページの「ftp によりリモートシステムへ接続する方法」を参照してください。

3. コピー元ディレクトリに変更します。

```
ftp> cd source-directory
```

システムがオートマウンタを使用している場合、リモートシステムのユーザーのホームディレクトリは、/home の下にユーザーのホームディレクトリと並行して表示されます。

4. コピー元ファイルの読み取り権があることを確認します。

```
ftp> ls -l
```

5. ファイルを 1 つコピーするには、get コマンドを使用します。

```
ftp> get filename
```

6. 一度に複数のファイルをコピーするには、mget コマンドを使用します。

```
ftp> mget filename [filename ...]
```

個々のファイル名を続けて入力するか、ワイルドカード文字を使用できます。mget コマンドでは、個々のファイルがコピーされ、そのたびに確認を求めるプロンプトが表示されます。

7. ftp による接続を終了します。

```
ftp> bye
```

例 — リモートシステムからファイルをコピーする (ftp)

次の例では、ユーザー kryten は、システム pluto と ftp 接続し、get コマンドを使用して /tmp ディレクトリから自分のホームディレクトリにファイルを 1 つコピーします。

```
$ cd $HOME
ftp pluto
Connected to pluto.
220 pluto FTP server (SunOS 5.8) ready.
Name (pluto:kryten): kryten
331 Password required for kryten.
Password: xxx
230 User kryten logged in.
ftp> cd /tmp
250 CWD command successful.
ftp> ls
200 PORT command successful.
150 ASCII data connection for /bin/ls (129.152.221.238,34344)
(0 bytes).
dtdbcache_:0
filea
files
ps_data
speckeysd.lock
226 ASCII Transfer complete.
53 bytes received in 0.022 seconds (2.39 Kbytes/s)
ftp> get filea
200 PORT command successful.
150 ASCII data connection for filea (129.152.221.238,34331)
(0 bytes).
226 ASCII Transfer complete.
ftp> bye
221 Goodbye.
```

次の例では、同じユーザー kryten が mget コマンドを使用して、/tmp ディレクトリから自分のホームディレクトリに複数のファイルをコピーします。kryten は、個々のファイルについてコピーするか、しないかの選択ができることに注意してください。

```
$ ftp> cd /tmp
250 CWD command successful.
ftp> ls files
200 PORT command successful.
150 ASCII data connection for /bin/ls (129.152.221.238,34345)
(0 bytes).
fileb
```

```
filec
filed
226 ASCII Transfer complete.
remote: files
21 bytes received in 0.015 seconds (1.36 Kbytes/s)
ftp> cd files
250 CWD command successful.
ftp> mget file*
mget fileb? y
200 PORT command successful.
150 ASCII data connection for fileb (129.152.221.238,34347)
(0 bytes).
226 ASCII Transfer complete.
mget filec? y
200 PORT command successful.
150 ASCII data connection for filec (129.152.221.238,34348)
(0 bytes).
226 ASCII Transfer complete.
mget filed? y
200 PORT command successful.
150 ASCII data connection for filed (129.152.221.238,34351)
(0 bytes).
226 ASCII Transfer complete.200 PORT command successful.
ftp> bye
221 Goodbye.
```

▼ ファイルをリモートシステムにコピーする方法 (ftp)

1. ローカルシステム上のコピー元ディレクトリに変更します。

   ftp コマンドを入力して接続するディレクトリは、ローカルの作業用ディレクトリ、つまりこの操作のコピー元ディレクトリになります。

2. ftp により接続します。

   233ページの「ftp によりリモートシステムへ接続する方法」を参照してください。

3. コピー先ディレクトリに変更します。

   ```
   ftp> cd target-directory
   ```

ローカルシステムでオートマウンタを使用中であれば、/home の下に自分のホームディレクトリと並行してリモートシステムのユーザーのホームディレクトリが表示されるので注意してください。

4. コピー先ディレクトリへの書き込み権があることを確認します。

```
ftp> ls -l target-directory
```

5. ファイルを 1 つコピーするには、put コマンドを使用します。

```
ftp> put filename
```

6. 一度に複数のファイルをコピーするには、mput コマンドを使用します。

```
ftp> mput filename [filename ...]
```

個々のファイル名を続けて入力するか、ワイルドカード文字を使用できます。mput コマンドでは、個々のファイルがコピーされ、そのたびに確認を求めるプロンプトが表示されます。

7. ftp による接続を終了するには、bye と入力します。

```
ftp> bye
```

例 — ファイルをリモートシステムにコピーする (ftp)

次の例では、ユーザー kryten はシステム pluto へ ftp により接続し、put コマンドを使用して自分のシステムからシステム pluto の /tmp ディレクトリにファイルをコピーします。

```
$ cd /tmp
ftp pluto
Connected to pluto.
220 pluto FTP server (SunOS 5.8) ready.
Name (pluto:kryten): kryten
331 Password required for kryten.
Password: xxx
230 User kryten logged in.
ftp> cd /tmp
250 CWD command successful.
```

(続く)

```
ftp> put filef
200 PORT command successful.
150 ASCII data connection for filef (129.152.221.238,34356).
226 Transfer complete.
ftp> ls
200 PORT command successful.
150 ASCII data connection for /bin/ls (129.152.221.238,34357) (0 bytes).
dtdbcache_:0
filea
filef
files
ps_data
speckeysd.lock
226 ASCII Transfer complete.
60 bytes received in 0.058 seconds (1.01 Kbytes/s)
ftp> bye
221 Goodbye.
```

次の例では、同じユーザー kryten は mput コマンドを使用して自分のホームディレクトリから pluto の /tmp ディレクトリに複数のファイルをコピーします。kryten は、個々のファイルについてコピーするか、しないかを選択できることに注意してください。

```
$ cd $HOME/testdir
$ ls
test1   test2   test3
$ ftp pluto
Connected to pluto.
220 pluto FTP server (SunOS 5.8) ready.
Name (pluto:kryten): kryten
331 Password required for kryten.
Password: xxx
230 User kryten logged in.
ftp> cd /tmp
250 CWD command successful.
ftp> mput test*
mput test1? y
200 PORT command successful.
150 ASCII data connection for test1 (129.152.221.238,34365).
226 Transfer complete.
mput test2? y
200 PORT command successful.
150 ASCII data connection for test2 (129.152.221.238,34366).
226 Transfer complete.
mput test3? y
200 PORT command successful.
150 ASCII data connection for filef (129.152.221.238,34356).
226 Transfer complete.
```

(続く)

```
ftp> bye
221 Goodbye.
```

rcp によるリモートコピー

rcp コマンドは、ローカルシステムとリモートシステム間、または 2 台のリモートシステム間でファイルやディレクトリをコピーします。このコマンドは、リモートシステムから (rlogin コマンドでログイン後に)、またはローカルシステムから (リモートシステムにログインせずに) 使用できます。

rcp を使用すると、次のリモートコピー操作を実行できます。

- 自分のシステムからリモートシステムにファイルやディレクトリをコピーする。
- リモートシステムからローカルシステムにファイルやディレクトリをコピーする。
- ローカルシステムを経由したリモートシステム間でファイルやディレクトリをコピーする。

オートマウンタを実行中の場合は、これらのリモート操作を cp コマンドで実行できます。ただし、cp の対象範囲は、オートマウンタで作成される仮想ファイルシステムと、ユーザーのホームディレクトリから、またはそこへの操作に限定されています。また、rcp は同じ操作を実行しますがこれらの制約がないので、この節では、rcp を使うこれらの作業についてのみ説明します。

コピー操作のセキュリティ上の注意事項

システム間でファイルやディレクトリをコピーするには、ログインしてファイルをコピーする許可を持っていなければなりません。

注意 - cp コマンドと rcp コマンドではともに、警告が表示されずにファイルが上書きされることがあります。コマンドを実行する前に、ファイル名が正しいかどうかを確認してください。

コピー元とコピー先の指定

C シェル内で rcp コマンドを使用すると、絶対パス名または相対パス名を使用して、コピー元 (コピーしたいファイルやディレクトリ) とコピー先 (ファイルやディレクトリをコピーする位置) を指定できます。

	絶対パス名	相対パス名
ローカルシステムから	mars:/home/jones/myfile.txt	~jones/myfile.txt
リモートログイン後	/home/jones/myfile.txt	~jones/myfile.txt

絶対パス名は、特定のシステムにマウントされているファイルやディレクトリを表します。上記の例で、第 1 の絶対パスは mars システム上のファイル (myfile.txt) を表します。相対パス名は、ファイルやディレクトリがある位置を、ユーザーのホームディレクトリからの相対パスで表します。上記の第 1 の例で、相対パス名は絶対パスと同じ myfile.txt を表しますが、jones のホームディレクトリを示すために “~” (チルド記号) を使用しています。チルドは次の値を表します。

```
~  =  mars:/home/jones
```

上記の 2 行目の例は、リモートログイン後の絶対パス名と相対パス名を示しています。相対パス名との間に違いはありませんが、リモートログイン操作によってホームディレクトリ jones がローカルシステムに (ローカルユーザーのホームディレクトリと並行して) マウントされたので、絶対パス名にはシステム名 mars が不要になります。リモートログイン操作によって別のユーザーのホームディレクトリがどのようにマウントされるかについては、226ページの「リモートログイン後の処理」を参照してください。

表 10–3 に、C シェルが認識する絶対パス名と相対パス名の例を示します。この表では次の用語を使用します。

- 作業用ディレクトリ — rcp コマンドを入力するディレクトリ。リモートまたはローカル
- 現在のユーザー — rcp コマンドを入力するユーザー名

表 10–3 ディレクトリ名とファイル名に使用できる構文

ログイン先	構文	説明
ローカルシステム	.	ローカルの作業用ディレクトリ
	path/*filename*	ローカルの作業用ディレクトリ内の path と *filename*
	~	現在のユーザーのホームディレクトリ
	~/path/*filename*	現在のユーザーのホームディレクトリの下の path と *filename*
	~*user*	*user* のホームディレクトリ
	~*user*/*path*/*filename*	*user* のホームディレクトリの下の path と *filename*
	remote-system : *path*/*filename*	リモートの作業ディレクトリ内の path と *filename*
リモートシステム	.	リモートの作業用ディレクトリ
	filename	リモートの作業用ディレクトリ内の *filename*
	path/*filename*	リモートの作業用ディレクトリ内の path と *filename*
	~	現在のユーザーのホームディレクトリ
	~/*path*/*filename*	現在のユーザーのホームディレクトリ内の path と *filename*
	~*user*	*user* のホームディレクトリ
	~*user*/*path*/*filename*	*user* のホームディレクトリの下の path と *filename*
	local-system : *path/filename*	ローカルの作業用ディレクトリ内の path と *filename*

表 10–3　ディレクトリ名とファイル名に使用できる構文　続く

▼ ローカルシステムとリモートシステム間でファイルをコピーする方法 (rcp)

1. コピーする許可を持っているかどうかを確認します。

   少なくとも、コピー元システム上で読み取り権を持ち、コピー先システム上で書き込み権を持っていなければなりません。

2. コピー元とコピー先の位置を決定します。

   コピー元またはコピー先のパスがわからない場合は、まず `rlogin` コマンドを使用してリモートシステムにログインし、位置が見つかるまでリモートシステム上を移動できます。手順については、230ページの「リモートシステムにログインする方法 (rlogin)」を参照してください。その後は、ログアウトしなくても次の手順を実行できます。

3. ファイルまたはディレクトリをコピーします。

```
$ rcp [-r] source-file | directory    target-file | directory
```

`rcp`　(オプションなし) コピー元からコピー先にファイルを 1 つコピーする。

`-r`　コピー元からコピー先にディレクトリをコピーする。

この構文は、リモートシステムとローカルシステムのどちらにログインするかに関係なく適用されます。表 10–3 で説明したとおり、ファイルやディレクトリのパス名のみをこの後で示す例のように変更します。

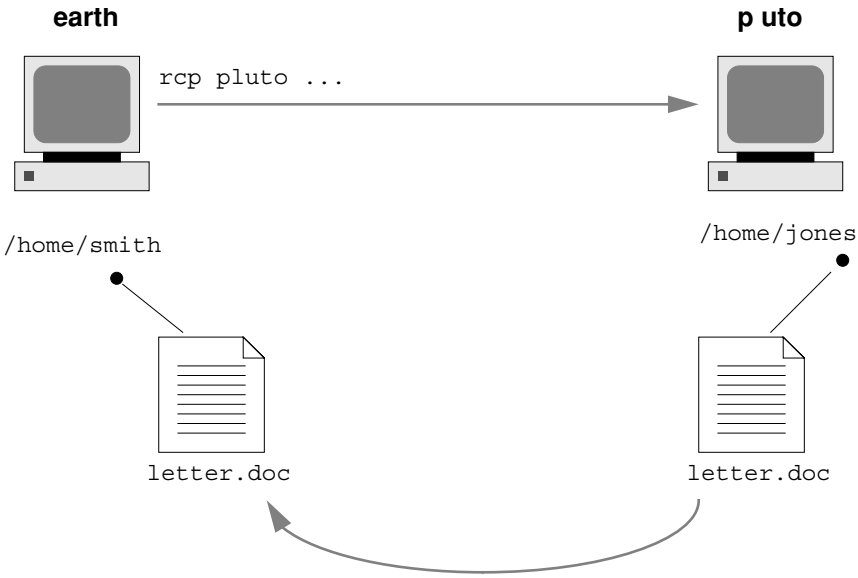
「~」と「.」を使用して、ローカルのファイル名やディレクトリ名のパス部分を指定できます。ただし、「~」はリモートシステムではなく現在のユーザーに適用されることと、「.」はログイン先のシステムに適用されることに注意してください。この 2 つの記号については、表 10–3 を参照してください。

例 — ローカルシステムとリモートシステム間でファイルをコピーする (rcp)

次にいくつかの例を示します。最初の 2 つの場合はコピー元がリモートで、あとの 2 つの場合はローカルです。

次の例では、rcp はファイル letter.doc をリモートシステム pluto の /home/jones ディレクトリから、ローカルシステム earth 上の作業用ディレクトリ (/home/smith) にコピーします。

```
earth(/home/smith): rcp pluto:/home/jones/letter.doc .
```



rcp 操作はリモートログインせずに実行されるので、「.」はリモートシステムではなくローカルシステムに適用されます。

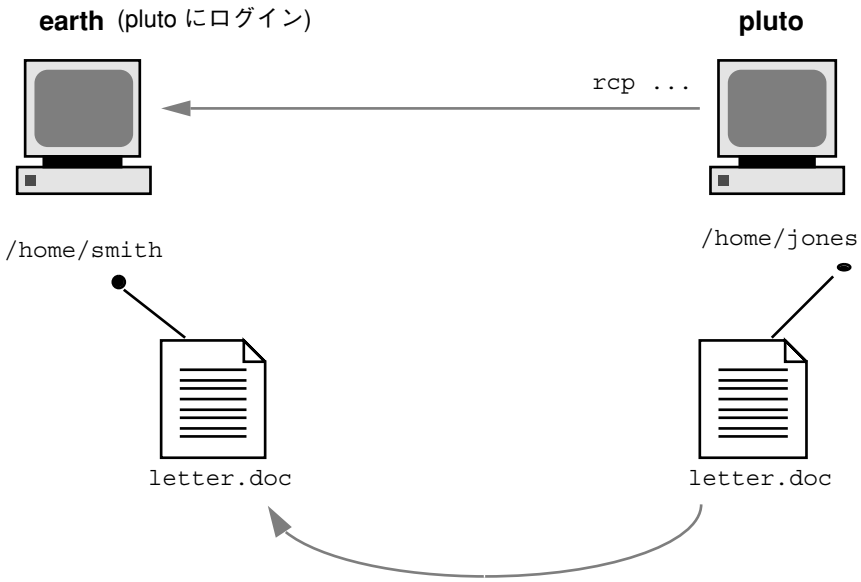
作業用ディレクトリはローカルユーザーのホームディレクトリになるので、「~」も使用して指定できます。

```
earth(home/smith): rcp pluto:/home/jones/letter.doc ~
```

次の例では、rcp を使用して同じ操作を実行しますが、リモートシステムにログインします。操作の流れは同じですが、リモートログインしているためパスを変更します。

```
earth(/home/smith): rlogin pluto
.
```

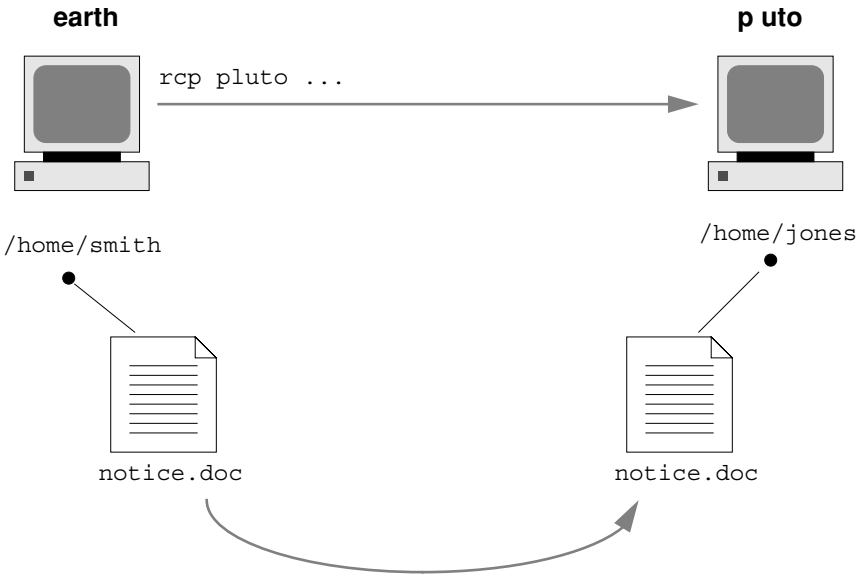
```
.
.
pluto(/home/jones): rcp letter.doc ~
```



この例では、リモートログインを行うので「.」を使用できません。「.」はリモートシステムにのみ適用され、実際には rcp にファイルのコピーを作成するように指示します。ただし、「~」は、リモートシステムにログインするときにも現在のユーザーのホームディレクトリを指します。

次の例で、rcp はファイル notice.doc をローカルシステム earth のホームディレクトリ (/home/smith) からリモートシステム pluto の /home/jones ディレクトリにコピーします。

```
earth(/home/smith): rcp notice.doc pluto:/home/jones
```



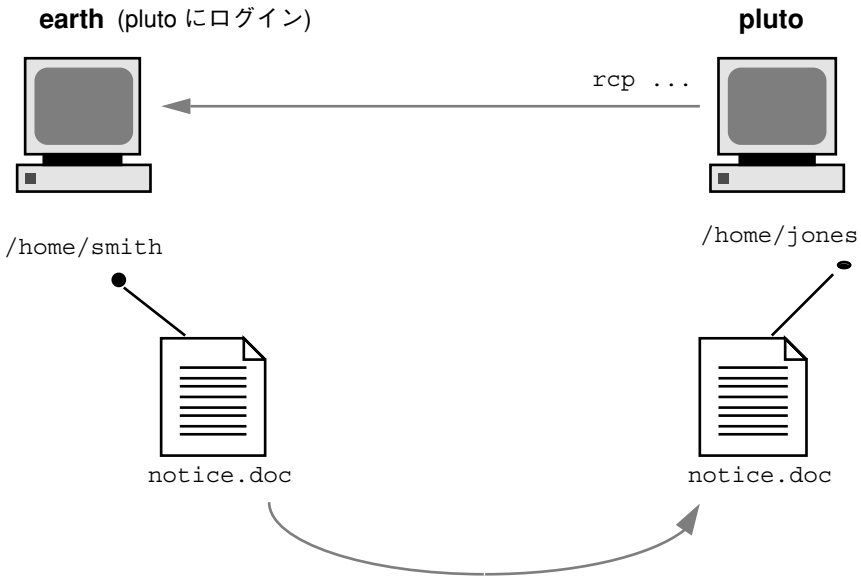
リモートファイル名が指定されていないので、ファイル notice.doc は /home/jones ディレクトリに同じ名前でコピーされます。

次の例でも同じ操作を繰り返しますが、rcp はローカルシステム上の別の作業用ディレクトリ (/tmp) から入力されます。現在のユーザーのホームディレクトリを指すために「~」記号が使われているので注意してください。

```
earth(/tmp): rcp ~/notice.doc pluto:/home/jones
```

次の例では、リモートシステムにログインしている間に rcp を使用して前記の例と同じ操作を実行します。操作の流れは同じですが、パスはリモートログインを考慮して変更されます。

```
earth(/home/smith): rlogin pluto
.
.
.
pluto(/home/jones): rcp ~/notice.doc .
```



この例では、現在のユーザーのホームディレクトリがローカルシステム上にあっても、「~」を使用して指定できます。ユーザーはリモートシステムにログインしているので、「.」はリモートシステム上の作業用ディレクトリを指します。次の構文を使用しても同じ操作を実行します。

```
pluto(/home/jones): rcp earth:/home/smith/notice.doc /home/jones
```

第 11 章

端末とモデムの管理

ここでは端末とモデムの管理手順について説明します。次の章が含まれます。

第 12 章	端末とモデムの概要について説明します。
第 13 章	端末とモデムを設定する手順について説明します。
第 14 章	SAF コマンドを使用して端末とモデムを設定する手順について説明します。

第 12 章

端末とモデム管理の概要

この章では、端末やモデムを管理する場合の概要を説明します。この章の内容は次のとおりです。

Admintool で端末とモデムを設定する手順については、第 13 章を参照してください。

SAF で端末とモデムを設定する手順については、第 14 章を参照してください。

端末、モデム、ポート、およびサービス

端末とモデムは、システム資源とネットワーク資源へのローカルおよびリモートのアクセスを提供します。端末とモデムの設定は、システム管理者の重要な作業です。この節では、Solaris 環境におけるモデムと端末の管理についての概要を説明します。

端末

システムのビットマップグラフィックスディスプレイは、シリアルポートに接続され、テキストしか表示できない英数字端末とは異なります。グラフィックスディスプレイは、特別な手順に従って管理する必要はありません。

モデム

モデムには次の 3 つの基本構成があります。

- 発信専用
- 着信専用
- 発着信両用

家庭用コンピュータに接続されるモデムの中には、「発信専用」サービス向けに設定されていることがあります。その場合、ユーザーは家から他のコンピュータにアクセスできますが、外からは誰もユーザーのコンピュータにアクセスできません。

「着信専用」サービスは発信専用のちょうど逆です。つまり、リモートサイトからはシステムにアクセスできますが、そのシステムから外側には呼び出しができません。

「発着信両用」アクセスは、その名前が示すとおり、着信専用、発信専用の両機能を持っています。

ポート

「ポート」とは、装置がオペレーティングシステムと通信するためのチャネルのことです。具体的には、端末やモデムのケーブルを差し込む「コンセント」と考えると一番わかりやすいでしょう。

ただし、ポートは厳密には物理的なコンセントではなく、その実体はハードウェア (ピンとコネクタ) とソフトウェア (デバイスドライバ) からなっています。多くの場合、1 つの物理的コンセントが複数のポートを備えており、複数の装置を接続できます。

一般的なポートとして、シリアル、パラレル、Small Computer Systems Interface (SCSI)、Ethernet などがあります。

「シリアルポート」は、標準の通信プロトコルを使用し、1 本の信号線で 1 バイト単位の情報を 1 ビットずつ送信します。

RS-232-C または RS-423 規格に準拠して設計されている装置 (大部分のモデム、英数字端末、プロッタ、一部のプリンタ) は、同様に設計されたコンピュータのシリアルポートにはどれにでも、標準ケーブルを使用して接続できます。

1 台のコンピュータに多数のシリアルポート装置を接続する場合は、システムに「アダプタボード」を追加する必要があることがあります。アダプタボードは、ドライバソフトウェアを使用することにより、より多くの装置を接続できるための追加のシリアルポートを提供します。

サービス

モデムや端末を使用すると、シリアルポートのソフトウェアを介してコンピュータ資源にアクセスできます。シリアルポートソフトウェアは、ポートに接続する装置向けに特定の「サービス」を提供するように設定しなければなりません。たとえば、モデムに対してはシリアルポートは発着信両用サービスを提供するように構成できます。

ポートモニター

特定のサービスへのアクセスは、主に「ポートモニター」を通じて行います。ポートモニターとは、ログイン要求や、プリンタまたはファイルのアクセス要求を常に監視しているプログラムのことです。

ポートモニターは要求を検出すると、オペレーティングシステムとサービスを要求する装置間の通信を確立するのに必要なすべてのパラメータを設定します。次に、必要なサービスを提供する他のプロセスに制御を移します。

表 12–1 に、Solaris 環境で提供されている 2 つのタイプのポートモニターとその説明を示します。

表 12–1 ポートモニターのタイプ

ポートモニター	説明
`listen(1M)`	Solaris 2.6 より前のシステムからのリモート印刷要求の処理など、ネットワークサービスへのアクセスを制御する。デフォルトの Solaris オペレーティング環境では、このタイプのポートモニターは使用されていない
`ttymon(1M)`	モデムや英数字端末が必要とするログインサービスへのアクセスを提供する。Admintool は、それらの装置からのログイン要求を処理するように、`ttymon` ポートモニターを自動的に設定する。

`getty(1M)` という従来のポートモニターに慣れているユーザーも、新しい `ttymon` を使用してください。新しい `ttymon` はさらに強力なツールとなっており、1 つの `ttymon` で複数の `getty` に相当する処理が行えます。それ以外の点では、どちらのプログラムも同じ機能を提供します。

端末とモデムを管理するツール

表 12–2 に、端末とモデムの管理に使用する推奨ツールを示します。表 12–3 には、サービスアクセス機能 (SAF) と Admintool : シリアルポート (Serial Ports) の機能の比較を示します。

表 12–2 端末とモデムの管理に使用する推奨ツール

必要な作業	推奨ツール	参照箇所
管理作業全般	サービスアクセス機能 (SAF) のコマンド	256ページの「サービスアクセス機能 (SAF)」
クイック設定	Admintool のグラフィカルユーザーインタフェース (ローカルシステムのみ)	第 13 章

表 12–3 Admintool とサービスアクセス機能の機能比較

手順	推奨ツール	説明
ポートが使用不可であることをユーザーに通知する	サービスアクセス機能 ttyadm -i	ttyadmin -i はポートが動作していない (使用不可にされている) ことを示す応答メッセージを引数で指定する。ポートが使用不可のときにユーザーがログインしようとすると、端末やモデムにこのメッセージが送られる。この機能は、Admintool を使用してポートを使用不可にした場合は提供されない
ホストからログアウトするときにモデムの接続を継続する	サービスアクセス機能 ttyadm -h	ttyadm -h は、デフォルトや特定の指定値に設定したりリセットしたりする前に、システムがモデムをハングアップしないよう指定するttyadm -h を使用しなければ、ホストからログアウトするとき、ホストがモデムをハングアップさせる
ユーザーが文字を入力した後にシステムにプロンプトを表示させる	サービスアクセス機能 ttyadm -r	ttyadm -r は、ユーザーが 1 文字入力するか、Return キーを指定回数押すと、ログインプロンプトが表示されるように指定する。-r を指定しなくても、Return キーを 1 回以上押すと、結局プロンプトは表示される。-r オプションを使用することにより、Solaris ホストが誤解してログインを試みることがないように、端末サーバーに歓迎メッセージを発行させないようにできる。-r オプションを使用しなければ、ホストと端末サーバーがループに入ってしまい、互いにプロンプトを発行し合う恐れがある

Admintool

Admintool : シリアルポート (Serial Ports) は、pmadm コマンドを呼び出すことにより、シリアルポートソフトウェアを設定して端末やモデムを管理します。また、次の機能も提供します。

- 共通の端末およびモデム構成用テンプレート
- 複数ポートの設定、変更、または削除
- 各ポートの状態の簡易表示

サービスアクセス機能 (SAF)

SAF は、端末、モデム、その他のネットワーク装置の管理用のツールです。SAF では特に次の設定を行います。

- `ttymon` および `listen` ポートモニター (`sacadm` コマンドを使用)
- `ttymon` ポートモニターサービス (`pmadm`、`ttyadm` コマンドを使用)
- `listen` ポートモニターサービス (`pmadm`、`nlsadmin` コマンドを使用)
- `tty` 装置に関する問題の解決
- ネットワークからの印刷サービス要求に関する問題の解決
- サービスアクセスコントローラに関する問題の解決 (`sacadm` コマンドを使用)

SAF は、`tty` 装置やローカルエリアネットワーク (LAN) を通して行われるシステム資源やネットワーク資源へのアクセスを制御するオープンシステムソリューションです。SAF はプログラムではなく、バックグランドプロセスと管理用コマンドの階層構造になっています。

第 13 章

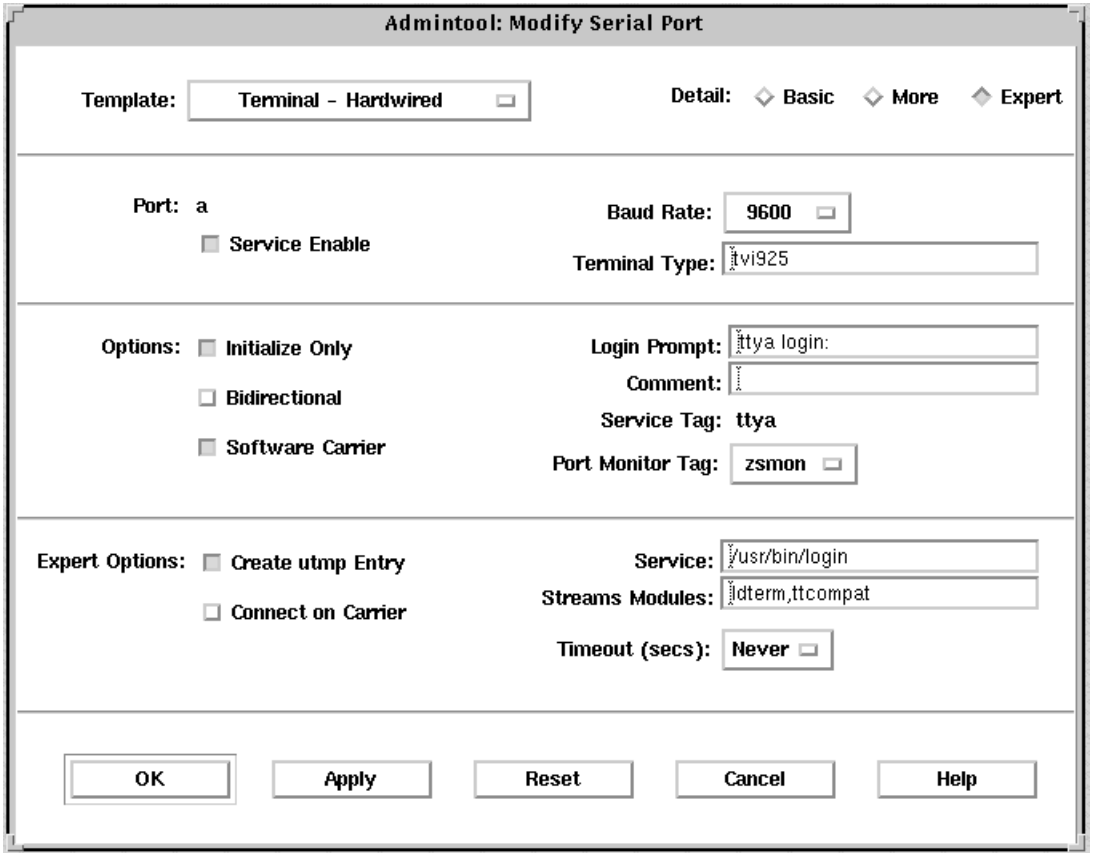
端末とモデムの設定

この章では、Admintool を使用して端末とモデムを設定する手順を示します。この章で説明する手順は次のとおりです。

端末とモデムの概要については、第 12 章を参照してください。

端末とモデムの設定

シリアルポート情報を設定するときは、Admintool を起動して、「ブラウズ (Browse)」メニューから「シリアルポート (Serial Ports)」を選択します。「Admintool : シリアルポート (Serial Ports)」ウィンドウからシリアルポートを選択して、次に「編集 (Edit)」メニューから「変更 (Modify)」を選択すると、「Admintool : シリアルポートの設定 (Modify Serial Port)」ウィンドウが表示されます。また、このウィンドウには、「基本 (Basic)」、「中級 (More)」、「上級 (Expert)」という 3 つの認定レベルで、ポートに関する情報が表示されます。



注 - 「Admintool : シリアルポートの設定 (Modify Serial Ports)」ウィンドウは「基本 (Basic)」モードで表示されます。「中級 (More)」や「上級 (Expert)」レベルを表示するには、「認定レベル (Detail)」メニューから「中級 (More)」または「上級 (Expert)」オプションを選択してください。

表 13–1 で「Admintool : シリアルポートの設定 (Modify Serial Ports)」ウィンドウの各項目を説明します。

表 13–1 「Admintool : シリアルポートの設定 (Modify Serial Ports)」ウィンドウの項目

認定レベル	項目	説明
基本 (Basic)	ポート (Port)	シリアルポートのメインウィンドウから選択した 1 つ以上のポートのリストを表示する
	サービスの利用 (Service Enable)	ポートに対するサービスをオン (有効) にする
	ボーレート (Baud Rate)	端末と通信する回線速度を指定する。回線速度は `/etc/ttydefs` ファイル内のエントリで指定される
	端末タイプ (Terminal Type)	ansi や vt100 などのように端末タイプの省略形を指定する。省略名については `/etc/termcap` を参照。この値は `$TERM` 環境変数に設定される
中級 (More)	オプション: 初期化操作のみ (Option: Initialize Only)	ポートソフトウェアが初期化されるが、構成されないように指定する
	オプション: 発着信両用 (Option : Bidirectional)	ポート回線が発着信両用に使用されるように指定する
	オプション: ソフトウェアキャリア (Option: Software Carrier)	ソフトウェアのキャリア検出機能が使用されるように指定する。このオプションをチェック (選択) しないと、「ハードウェア」のキャリア検出機能が使用される
	ログインプロンプト (Login Prompt)	接続が確立後に、ユーザー向けにプロンプトを表示する
	備考欄 (Comment)	サービス用のコメントフィールドを表示する
	サービスタグ (Service Tag)	このポートに対応するサービスタグを表示する。このタグは通常 `/dev/term` ディレクトリ内のエントリである
	ポートモニタータグ (Port Monitor Tag)	このポートに使用されるポートモニター名を指定する。 注: 通常、モニターはデフォルトで正しく設定される
上級 (Expert)	`utmpx` エントリの作成 (Create utmp Entry)	ログイン時にアカウンティングファイルに `utmpx` エントリが作成されるように指定する。 注: ログインサービスを使用する場合は、この項目を選択しなければならない。「サービス」の項目を参照

表 13–1 「Admintool : シリアルポートの設定 (Modify Serial Ports)」ウィンドウの項目 続く

認定レベル	項目	説明
	キャリア検出時に接続 (Connect on Carrier)	接続指示を受け取るとすぐにポート対応するサービスが起動されるように指定する
	サービス (Service)	接続時に実行されるプログラムを表示する
	ストリームモジュール (Streams Modules)	サービスが起動される前にプッシュされる STREAMS モジュールを表示する
	タイムアウト (秒)	ポートのオープンプロセスが正常に終了しているが入力データが受信されない場合に、ポートを閉じるまでの秒数を指定する

端末の設定

表 13–2 にシリアルポートを使用して端末を設定する際のメニュー項目 (およびそれらのデフォルト値) を示します。

表 13–2 端末 — ハード接続のデフォルト値

認定レベル	項目	デフォルト値
基本 (Basic)	ポート (Port)	-
	サービスの利用 (Service Enable)	有効
	ボーレート (Baud Rate)	9600
	端末タイプ (Terminal Type)	-
中級 (More)	オプション: 初期化操作のみ (Option: Initialize Only)	なし
	オプション: 発着信両用 (Option: Bidirectional)	なし

表 13–2 端末 — ハード接続のデフォルト値 続く

認定レベル	項目	デフォルト値
	オプション: ソフトウェアキャリア (Option : Software Carrier)	あり
	ログインプロンプト (Login Prompt)	`login:`
	備考欄 (Comment)	端末 — ハード接続
	サービスタグ (Service Tag)	-
	ポートモニタータグ (Port Monitor Tag)	`zsmon`
上級 (Expert)	`utmpx` エントリの作成 (Create utmp Entry)	あり
	キャリア検出時に接続 (Connect on Carrier)	なし
	サービス (Service)	`/usr/bin/login`
	ストリームモジュール (Streams Modules)	`ldterm`, `ttcompat`
	タイムアウト (秒) (Timeout) (secs)	なし

モデムの設定

表 13–3 に、Solstice シリアルポートでモデムを設定する際に使用できる 3 つのモデム用テンプレートを示します。

表 13–3 モデム用テンプレート

モデム構成	説明
モデム - 着信専用	モデムに着信はできるが、発信はできない。
モデム - 発信専用	モデムから発信はできるが、着信はできない。
モデム - 発着信両用	モデムへ着信も、モデムから発信もできる。

表 13–4 に各テンプレートのデフォルト値を示します。

表 13–4 モデム用テンプレートのデフォルト値

認定レベル	項目	モデム - 着信専用	モデム - 発信専用	モデム - 発着信両用
基本 (Basic)	ポート (Port)	-	-	-
	サービスの利用 (Service Enable)	有効	有効	有効
	ボーレート (Baud Rate)	9600	9600	9600
	端末タイプ (Terminal Type)	-	-	-
中級 (More)	オプション: 初期化操作のみ	あり	なし	なし
	オプション: 発着信両用	なし	なし	あり
	オプション: ソフトウェアキャリア (Option: Software Carrier)	なし	なし	なし
	ログインプロンプト (Login Prompt)	login:	login:	login:
	備考欄 (Comment)	モデム - 着信専用	モデム - 発信専用	モデム - 発着信両用
	サービスタグ (Service Tag)	-	-	-
	ポートモニタータグ	zsmon	zsmon	zsmon
上級 (Expert)	utmpx エントリの作成 (Create utmp Entry)	あり	あり	あり
	キャリア検出時に接続 (Connect on Carrier)	なし	なし	なし
	サービス (Service)	/usr/bin/login	/usr/bin/login	/usr/sbin/login

表 13–4 モデム用テンプレートのデフォルト値 続く

認定レベル	項目	モデム - 着信専用	モデム - 発信専用	モデム - 発着信両用
	ストリームモジュール (Streams Modules)	`ldterm, ttcompat`	`ldterm, ttcompat`	`ldterm, ttcompat`
	タイムアウト (秒) (Timeout) (secs)	なし	なし	なし

表 13–5 では、「初期化操作のみ」テンプレートの各デフォルト値を示します。

表 13–5 「初期化操作のみ (Initialize Only)」のデフォルト値

認定レベル	項目	説明
基本 (Basic)	ポート (Port)	-
	サービスの利用 (Service Enable)	有効
	ボーレート (Baud Rate)	9600
	端末タイプ (Terminal Type)	-
中級 (More)	オプション: 初期化操作のみ (Option: Initialize Only)	あり
	オプション: 発着信両用 (Option: Bidirectional)	なし
	オプション: ソフトウェアキャリア (Option: Software Carrier)	なし
	ログインプロンプト (Login Prompt)	`login:`
	備考欄 (Comment)	初期化操作のみ - 接続なし
	サービスタグ (Service Tag)	-
	ポートモニタータグ (Port Monitor Tag)	`zsmon`

表 13–5 「初期化操作のみ (Initialize Only)」のデフォルト値 続く

認定レベル	項目	説明
上級 (Expert)	utmpx エントリの作成 (Create utmp Entry)	あり
	キャリア検出時に接続 (Connect on Carrier)	なし
	サービス (Service)	/usr/bin/login
	ストリームモジュール (Streams Modules)	ldterm,ttcompat
	タイムアウト (秒) (Timeout) (secs)	なし

▼ Admintool を起動する方法

1. **Admintool** を使用するには次の条件が必要です。
   - ビットマップディスプレイモニターがある。Admintool ソフトウェアは、Sun のワークステーションの標準ディスプレイモニターなど、ビットマップ画面のコンソールを使用するシステムでだけ使用できます。
   - CDE 環境などの X Window System を実行している。
   - sysadmin グループ (グループ 14) のメンバーになっている。

   コンソールとして ASCII 端末を使用するシステムで管理作業を行いたい場合は、Solaris のコマンドを使用してください。

2. **Admintool** を起動します。

```
$ admintool &
```

「Admintool : ユーザー (Users)」のメインウィンドウが表示されます。

▼ 端末を設定する方法

1. **Admintool** を起動します。

   詳細は、264ページの「Admintool を起動する方法」を参照してください。

2. 「ブラウズ **(Browse)**」メニューから「シリアルポート **(Serial Ports)**」を選択します。

   「Admintool : シリアルポート (Serial Ports)」ウィンドウが表示されます。

3. 端末に使用するポートを **1** つまたは複数選択します。

4. 「編集 **(Edit)**」メニューから「変更 **(Modify)**」を選択します。

   「Admintool : シリアルポートの設定 (Modify Serial Ports)」ウィンドウが「基本 (Basic)」モードで表示されます。より詳細なレベルにするには、「中級 (More)」または「上級 (Expert)」モードを選択してください。

5. 「テンプレート **(Template)**」メニューから「端末 **-** ハード接続 **(Terminal-Hardwired)**」を選択します。

   「端末 – ハード接続 (Terminal–Hardwired)」メニューの項目については、表 13–2 の説明を参照してください。

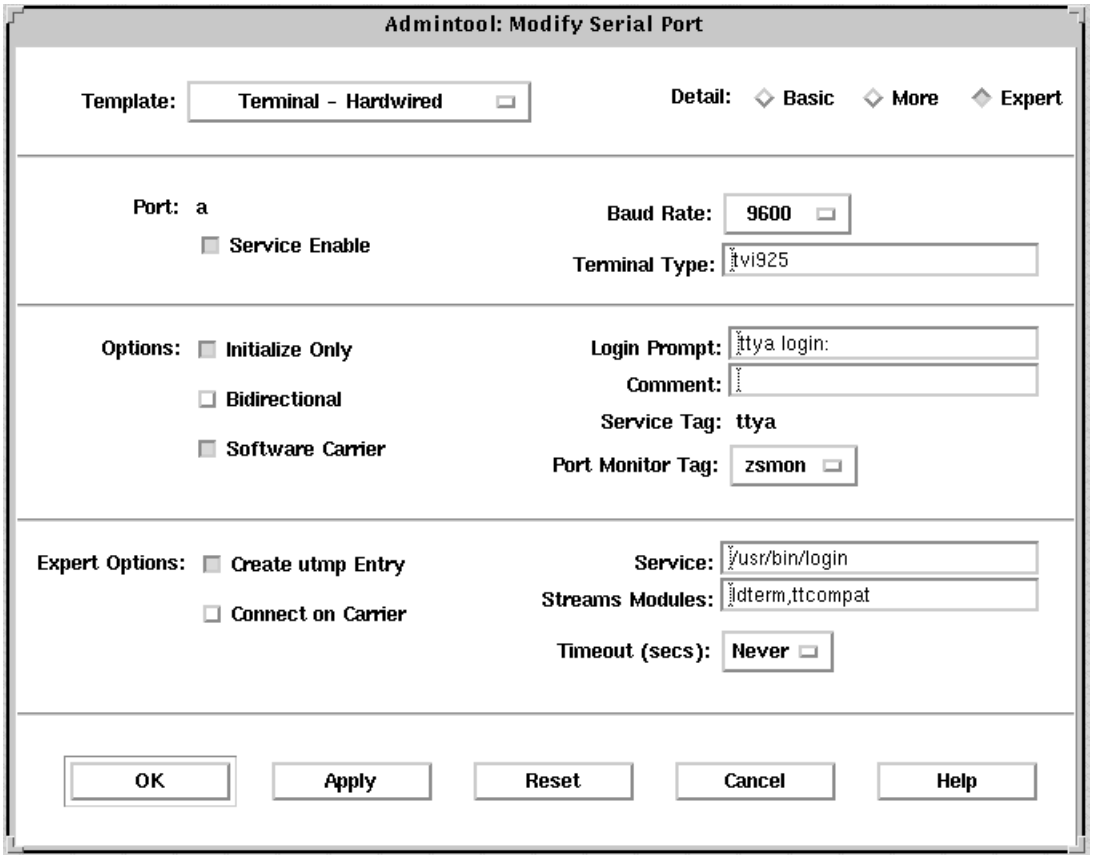
6. 必要な場合は、テンプレートエントリの値を変更します。

7. 「了解 **(OK)**」をクリックしてポートを設定します。

8. 次のように pmadm コマンドを使用して、端末サービスが設定されていることを確認します。

```
$ pmadm -l -s ttya
```

端末を設定する「**Admintool：シリアルポートの設定 (Modify Serial Ports)**」ウィンドウの入力例



▼ モデムを設定する方法

1. **Admintool** を起動します。
   詳細は、264ページの「Admintool を起動する方法」を参照してください。
2. 「ブラウズ **(Browse)**」メニューから「シリアルポート **(Serial Ports)**」を選択します。
   「Admintool：シリアルポート (Serial Ports)」ウィンドウが表示されます。
3. モデムに使用するポートを **1** つまたは複数選択します。
4. 「編集 **(Edit)**」メニューから「変更 **(Modify)**」を選択します。

「Admintool : シリアルポートの設定 (Modify Serial Ports)」ウィンドウが「基本 (Basic)」モードで表示されます。より詳細なレベルにするには、「中級 (More)」または「上級 (Expert)」モードを選択してください。

5. 「テンプレート **(Template)**」メニューから、目的のモデムサービスに合う、または最もよく一致するモデム設定テンプレートを選択します。

   各テンプレートについては、表 13–3 の説明を参照してください。

   各テンプレートのデフォルト値については、表 13–4 を参照してください。
   UUCP サービスを使用して Solaris システムのモデムを着信専用にする場合は、268ページの「モデムを UUCP 用に設定する方法」を参照してください。

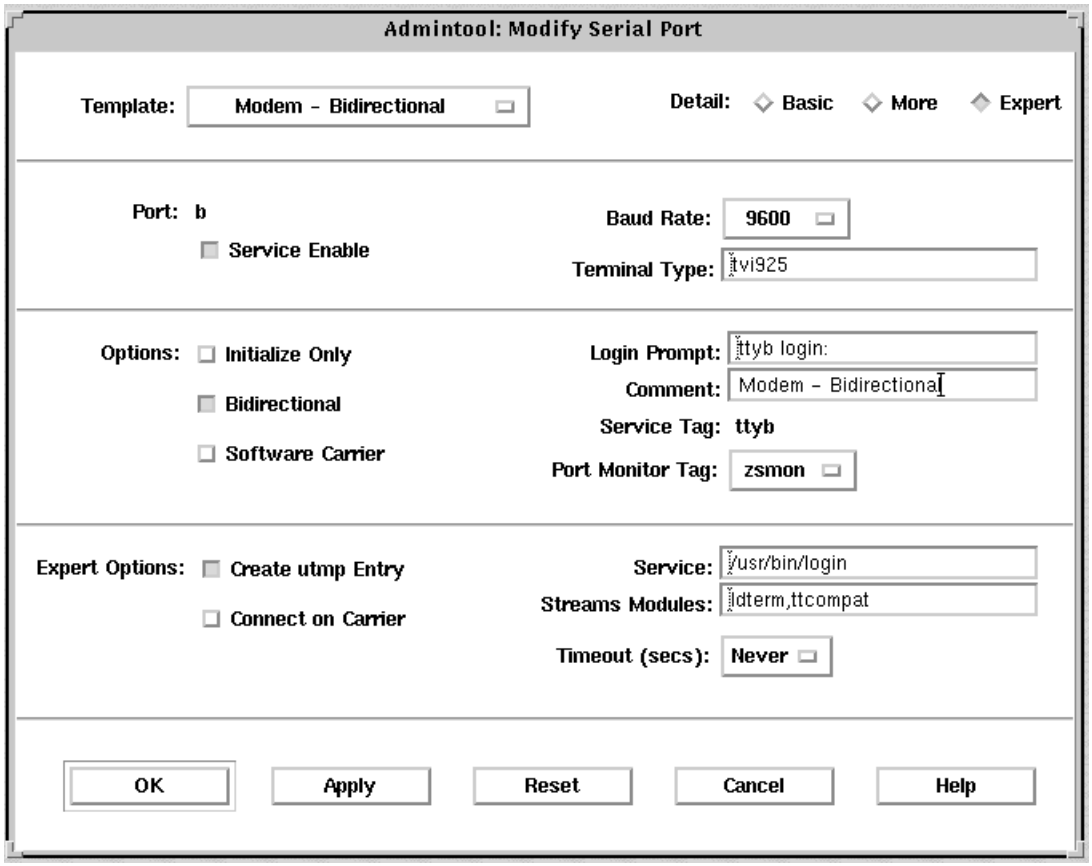
6. 必要な場合は、テンプレートエントリの値を変更します。

7. 「了解 **(OK)**」をクリックしてポートを設定します。

8. 次のように pmadm コマンドを使用して、**UUCP** 用のモデムサービスが構成されていることを確認します。

```
$ pmadm -l -s ttyb
```

モデムを設定する「**Admintool：シリアルポートの設定 (Modify Serial Ports)**」ウィンドウの入力例



▼ モデムを UUCP 用に設定する方法

UUCP は、7 ビット、偶数パリティを使用してサービスに情報を送ります。Solaris のモデム設定では、国際化対応の目的から、8 ビット、パリティなしが使用されます。次の手順でモデムサービスを UUCP 用に設定してください。

1. **Admintool** を起動します。

   詳細は、264ページの「Admintool を起動する方法」を参照してください。

2. 「ブラウズ **(Browse)**」メニューから「シリアルポート **(Serial Ports)**」を選択します。

   「Admintool：シリアルポート (Serial Ports)」ウィンドウが表示されます。

3. モデムに使用するポートを **1** つまたは複数選択します。

4. 「編集 **(Edit)**」メニューから「変更 **(Modify)**」を選択します。

   「Admintool : シリアルポートの設定 (Modify Serial Ports)」ウィンドウが「基本 (Basic)」モードで表示されます。より詳細なレベルにするには、「中級 (More)」または「上級 (Expert)」モードを選択してください。

5. 「ボーレート **(Baud Rate)**」メニューから「その他 **... (Other)**」を選択します。

   `/etc/ttydefs` ファイルにあるボーレートリストを示すウィンドウが表示されます。

6. **7** ビット、偶数パリティのサービスを提供するボーレートを入力します。「了解 **(OK)**」をクリックします。

7. 必要な場合は、他のテンプレートエントリの値を変更します。

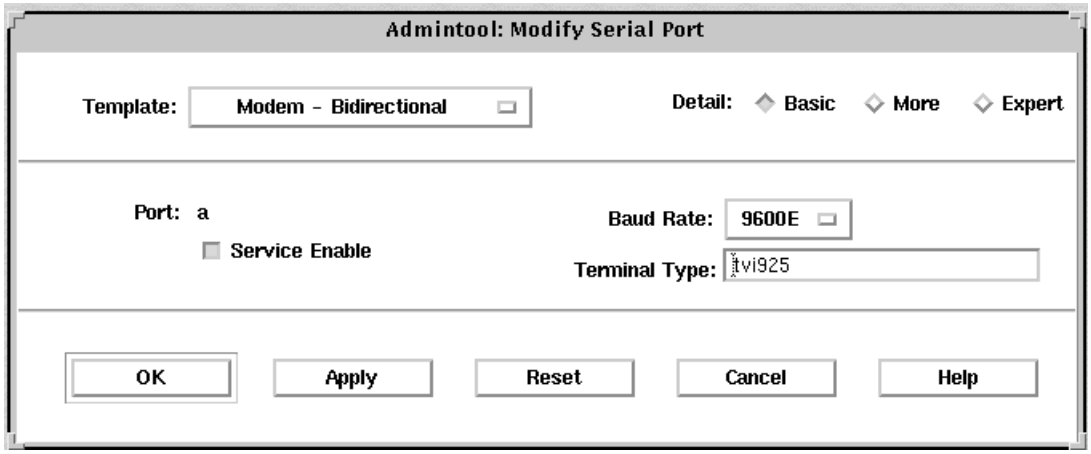
8. 「了解 **(OK)**」をクリックしてポートを設定します。

9. 次のように `pmadm` コマンドを使用して、**UUCP** 用のモデムサービスが構成されていることを確認します。

```
$ pmadm -l -s ttya
```

例 — モデムを UUCP 用に設定する

次の例では、9600E ボーレートが選択されています。これで、9600 ボーレート、7 ビット、偶数パリティのサービスが提供されます。



▼ ポートを初期化する方法

1. **Admintool** を起動します。

   詳細は、264ページの「Admintool を起動する方法」を参照してください。

2. 「ブラウズ **(Browse)**」メニューから「シリアルポート **(Serial Ports)**」を選択します。

   「Admintool : シリアルポートの設定 (Modify Serial Ports)」ウィンドウが表示されます。

3. 初期化したいポートを **1** つまたは複数選択します。

4. 「編集 **(Edit)**」メニューから「変更 **(Modify)**」を選択します。

   「Admintool : シリアルポートの設定 (Modify Serial Ports)」ウィンドウが「基本 (Basic)」モードで表示されます。より詳細なレベルにするには、「中級 (More)」または「上級 (Expert)」モードを選択してください。

5. 「テンプレート **(Template)**」メニューから「初期化操作のみ **-** 接続なし **(Initialize Only - No Connection)**」を選択します。

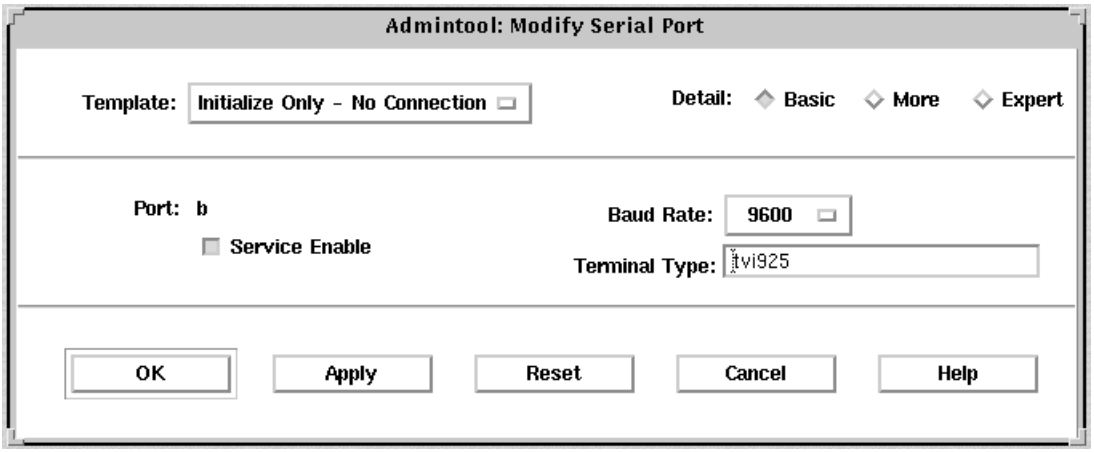
   「初期化操作のみ - 接続なし (Initialize Only - No Connection)」テンプレートについては、表 13–5 の説明を参照してください。

6. 「了解 **(OK)**」をクリックしてポートを初期化します。

7. 次のように pmadm コマンドを使用して、ポートが初期化されていることを確認します。

```
$ pmadm -l -s ttyb
```

ポートを初期化する変更ウィンドウの入力例



▼ ポートを使用不可にする方法

1. **Admintool** を起動します。

   詳細は、264ページの「Admintool を起動する方法」を参照してください。

2. 「ブラウズ **(Browse)**」メニューから「シリアルポート **(Serial Ports)**」を選択します。

   「Admintool : シリアルポート (Serial Ports)」ウィンドウが表示されます。

3. 使用不可にしたいポートを **1** つまたは複数選択します。

4. 「編集 **(Edit)**」メニューから「変更 **(Modify)**」を選択します。

5. 「サービスの利用 **(Service Enable)**」ボタンをクリックして、「**Admintool :** シリアルポートの設定 **(Modify Serial Ports)**」ウィンドウ上のポートサービスを使用不可にします。

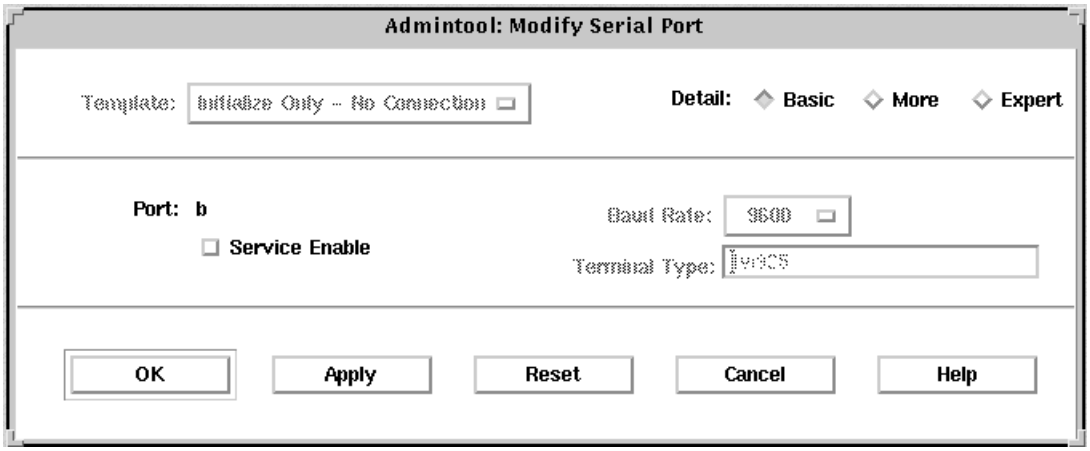
   このボタンは、ポートサービスを使用可能と使用不可とに切り替えます。

6. 「了解 **(OK)**」をクリックしてポートを使用不可にします。

7. 次のように pmadm コマンドを使用して、ポートサービスが使用不可にされていることを確認します。

```
$ pmadm -l -s ttya
```

ポートを使用不可にする変更ウィンドウの入力例



▼ ポートサービスを削除する方法

1. **Admintool** を起動します。

   詳細は、264ページの「Admintool を起動する方法」を参照してください。

2. 削除したいサービスを提供するポートを **1** つまたは複数選択します。

3. 「編集 **(Edit)**」メニューから「削除 **(Delete)**」を選択します。

   指定したポートのサービスを本当に削除したいかどうかの確認を求められます。削除操作は取り消すことも、そのまま実行することもできます。

4. 次のように pmadm コマンドを使用して、ポートサービスが削除されていることを確認します。

```
$ pmadm -l -s ttya
```

端末とモデムの問題を解決する方法

端末またはモデムを追加し、適切なサービスを設定したにもかかわらず、シリアルポート回線を通してログインできない場合は、次のような順序で問題を解決してください。

- ユーザーを確認します。

  端末やモデムが正しく動作しないという報告は、多くの場合、ログインや着信ができなかったユーザーから寄せられます。したがって、まず、デスクトップに問題がないかどうかを確認することから始めてください。

  ログインできない主な原因は、次のとおりです。

  - ログイン ID またはパスワードが正しくない
  - 端末が X-ON フロー制御キー (Control-q) の入力を待っている
  - シリアルケーブルの接続が緩んでいるか外れている
  - 端末の設定が正しくない
  - 端末の電源が切られたか、端末に電源が入っていない

- 端末の設定を確認します。

  次に、端末またはモデムの設定を調べます。端末またはモデムとの通信の正しい tty 名を調べ、それぞれの設定が tty 名の設定と一致することを確認します。

- 端末サーバーの設定を確認します。

  端末に問題のないことがわかったら、端末またはモデムのサーバーに問題がないかどうかを調べます。pmadm コマンドを使用して、ポートモニターが端末またはモデムをサービスするように設定されていて、関連する tty 名が正しいことを確認します。

```
$ pmadm -l -t ttymon
```

  /etc/ttydefs を調べ、ラベル定義を端末設定と照合してチェックします。sacadm を使用してポートモニターの状態を調べます。pmadm を使用して、端末が使用するポートのサービスを調べます。

- シリアル接続を確認します。

  サービスアクセスコントローラが TTY ポートモニターを起動し、pmadm が端末のポートに対するサービスが有効になっていると報告し、さらに端末の設定が

ポートモニターの設定と一致する場合は、シリアル接続を調べて問題の原因を探します。シリアル接続は、シリアルポート、ケーブル、端末から構成されています。これらの構成部分のうち 2 つを、信頼性が確認されている他のものに取り替えて、1 箇所ずつテストしてください。

次の構成部分をすべてテストします。

- シリアルポート
- モデム
- ケーブル
- コネクタ

- シリアルポートをコンソールとして使用している場合は、Admintool からシリアルポートの設定を変更しないでください。コンソール設定を正しく変更するには、/etc/inittab ファイルの次の行を変更してください。

```
co:234:respawn:/usr/lib/saf/ttymon -g -h -p "`uname -n` console
login: " -T terminal_type -d /dev/console -l console -m
ldterm,ttcompat
```

- IA 搭載のシステムにモデムを接続する場合は、『*Solaris 8* ハードウェア互換リスト *(Intel* 版*)*』を見て、モデムがサポートされているか確認します。

第 14 章

サービスアクセス機能による端末とモデムの設定手順

この章では、Solaris 環境のサービスアクセス機能 (SAF) について、システム管理者またはネットワーク管理者が知っておく必要があることがらを詳細に説明します。

SAF コマンドの使用例については、次の表で必要な手順の説明を探してください。

端末とモデムの概要については、第 12 章を参照してください。

サービスアクセス機能 (SAF) の概要

SAF は端末、モデム、および他のネットワーク装置を管理するためのツールです。SAF プログラムの最上位には、サービスアクセスコントローラ (SAC) があります。SAC は、管理者が sacadm コマンドにより管理するポートモニターを制御します。各ポートモニターは 1 つ以上のポートを管理できます。

管理者は pmadm コマンドを使用して、ポートに対応するサービスを管理します。SAC が提供するサービスはネットワークによって異なりますが、SAC と管理プログラム sacadm と pmadm はネットワークには依存しません。

表 14–1 に SAF の制御階層を示します。sacadm コマンドを使用すると、ttymon および listen ポートモニターを制御する SAC を管理できます。

また、ttymon と listen のサービスは pmadm により制御されます。ttymon の 1 つのインスタンスは複数のポートにサービスを提供し、listen の 1 つのインスタンスはネットワークインタフェース上で複数のサービスを提供できます。

表 14–1 SAF の制御階層

機能	プログラム	説明
全体の管理	sacadm	ポートモニターの追加および削除用コマンド
サービスアクセスコントローラ	sac	SAF のマスタープログラム
ポートモニター	ttymon	シリアルポートのログイン要求を監視する
	listen	ネットワークのサービス要求を監視する
ポートモニターサービスの管理	pmadm	ポートモニターのサービス制御用コマンド
サービス	ログイン、リモートプロシージャコール、その他	SAF がアクセスを可能にするサービス
コンソールの管理	コンソールログイン	コンソールは、/etc/inittab ファイル中のエントリ経由で、ttymon の express モードを使用して、自動的に設定される。pmadm か sacadm を使用して、コンソールを直接管理しないでください。詳細は、281ページの「ttymon とコンソールポート」を参照

全体の管理: sacadm コマンド

sacadm コマンドは SAF 階層の最上位のコマンドです。sacadm コマンドは主に、ttymon および listen などのポートモニターを追加または削除するのに使用します。このコマンドにはそれ以外に、ポートモニターの現在の状態の表示、ポートモニターの構成スクリプトの管理などの機能があります。

サービスアクセスコントローラ: SAC プログラム

サービスアクセスコントローラ (SAC) プログラムはすべてのポートモニターを管理します。システムはマルチユーザーモードになると自動的に SAC を起動します。

SAC は、起動されるとまず、各システムの構成スクリプトを探して解釈し、それによって SAC の環境をカスタマイズします。ここで行われる SAC の環境に対する変更は、SAC のすべての「子プロセス」に継承されます。継承された環境は継承した子プロセスで変更できます。

SAC プログラムは、システムごとの構成スクリプトの解釈が終わると、SAC の管理ファイルを読み取り、指定されたポートモニターを起動します。各ポートモニターについて、SAC はそれ自身のコピーを実行します (技術的には、SAC が子プロセスをフォークします)。次に、各子プロセスは、それぞれのポートモニターごとの構成スクリプトがあればそれを解釈します。

各ポートモニターの構成スクリプトに指定されている環境を変更すると、それぞれのポートモニターが影響を受け、さらにそれがポートモニターのすべての子プロセスに継承されます。最後に、子プロセスは SAC 管理ファイル内のコマンドを使用して親であるポートモニタープログラムを実行します。

SAC の初期化プロセス

次に、SAC を最初に起動したときに行われる一連の処理を要約します。

1. init が実行レベル 2 で SAC プログラムを生成します。
2. SAC プログラムがシステムごとの構成スクリプト /etc/saf/_safconfig を読み取ります。
3. SAC プログラムが SAC 管理ファイル /etc/saf/_sactab を読み取ります。
4. SAC プログラムが起動する各ポートモニターの子プロセスをフォークします。

5. 各ポートモニターがポートモニターごとの構成スクリプト /etc/saf/pmtag/_config を読み取ります。

ポートモニターサービス管理: pmadm コマンド

pmadm コマンドでポートモニターのサービスを管理できます。pmadm コマンドは特にサービスを追加または削除したり、サービスを有効または無効にしたりする場合に使用します。このコマンドでは、さらに、各サービスの構成スクリプトをインストールしたり置き換えたり、サービスに関する情報を出力したりすることもできます。

サービスの各インスタンスは、ポートモニター別、ポート別に一意に識別できなければなりません。pmadm コマンドを使用してサービスを管理する場合、*pmtag* 引数で特定のポートモニターを、また *svctag* 引数で特定のポートをそれぞれ指定します。

ポートモニターのタイプごとに、SAF はポートモニター固有の構成データのフォーマットを定義するための特別なコマンドを必要とします。この構成データは pmadm コマンドで使用します。ttymon および listen ポートモニター用の特別なコマンドは、それぞれ ttyadm と nlsadmin です。

ポートモニターの動作: ttymon

直結モデムまたは英数字端末を通してログインしようとするたびに、ttymon は次のように動作を開始します。

図 14–1 に示すように、init プログラムがブート時に最初に起動されるプロセスです。init は、その管理ファイル (/etc/inittab) を参照して、必要に応じて他のプロセスを起動します。それらのプロセスの 1 つに SAC があります。

SAC が起動されると、今度は SAC がその管理ファイル (/etc/saf/_sactab) に指定されているポートモニターを自動的に起動します。図 14–1 には ttymon モニターが 1 つしか示されていません。

ttymon は起動されると、シリアルポート回線を監視してサービス要求がないかどうかを調べます。

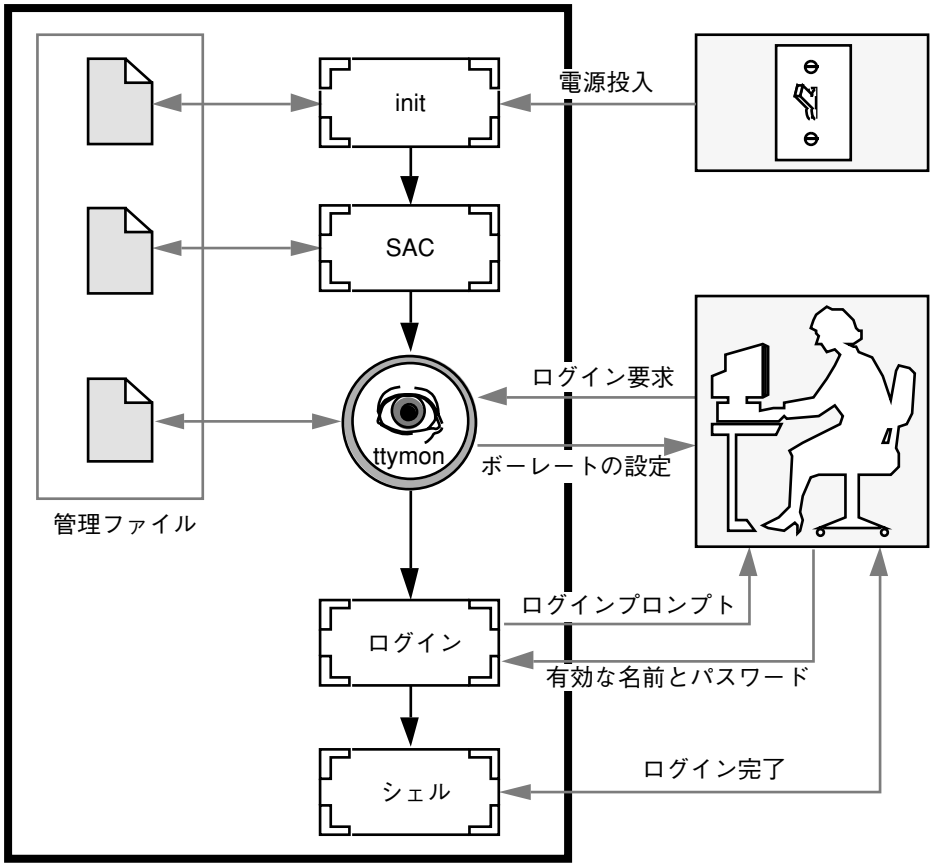


図 14–1　ttymon によるログイン要求の処理

ユーザーが英数字端末やモデムを通してログインしようとすると、シリアルポートドライバはその操作をオペレーティングシステムに伝えます。ttymon ポートモニターはシリアルポートの操作を監視し、通信リンクを確立しようとします。つまり、装置との通信に必要なデータ転送速度、回線制御手順、ハンドシェークプロトコルを決定します。

モデムや端末との通信用の正しいパラメータの設定が終わると、ttymon はそれらのパラメータをログインプログラムに渡し、制御を移します。

ポートの初期化プロセス

SAC が ttymon のインスタンスを起動すると、ttymon はそのポートの監視を開始します。ttymon は、各ポートについて指定されている場合、まず回線制御手順を

初期化し、次に回線速度と端末の設定を初期化します。初期化に使用される値は、/etc/ttydefs の該当するエントリから得られます。

ttymon ポートモニターは、次に、プロンプトを表示してユーザーからの入力を待ちます。ユーザーが Break キーを押して回線速度が不適当であるという指示を与えると、ttymon は次の速度を設定して、再びプロンプトを表示します。

ポートに対して「自動ボーレート」が有効に設定されている場合は、ttymon はそのポートのボーレートを自動的に決定しようとします。ここで Return キーを押してください。そうすると、ttymon はボーレートを認識し、プロンプトを表示できます。

有効な入力を受け取ると、ttymon はポートのサービスごとの構成ファイルを解釈し、必要な場合は /etc/utmp エントリを作成し、サービス環境を設定し、ポートに対応するサービスを起動します。

サービスが終了すると、ttymon は /etc/utmpx エントリがあれば削除し、ポートを初期状態に戻します。

発着信両用サービス

ポートが発着信両用サービス用に構成してある場合は、ttymon は次の処理を行います。

- ユーザーをサービスに接続可能にする。
- uucico、cu、または ct が、(空いていれば) ポートを発信専用モードで使用できるようにする。
- 文字を読み取ってからプロンプトを表示する。
- 接続要求があると (connect-on-carrier フラグが設定してある場合)、プロンプトメッセージを送らないでポートの対応サービスを起動する。

ポートモニター: TTY モニターとネットワークリスナー

SAF は、将来のモニターや他社製のポートモニターの管理に対応するために総合的な管理方法を提供していますが、Solaris 環境では ttymon と listen の 2 つだけが実装されています。

TTY ポートモニター: ttymon

ttymon ポートモニターは STREAMS をベースにしています。このモニターは、ポートを監視し、端末のモード、ボーレート、回線制御手順を設定し、ログインプロセスを起動します。(ttymon は、以前のバージョンの SunOS 4.1 ソフトウェアのもとで getty が提供していたのと同じサービスを Solaris ユーザーに提供します。)

ttymon ポートモニターは SAC プログラムで実行されます。ttymon は sacadm コマンドを使用して構成します。ttymon の各インスタンスはそれぞれに複数のポートを監視できます。それらのポートはポートモニターの管理ファイル内に指定します。また、この管理ファイルは pmadm および ttyadm コマンドを使用して構成します。

ttymon とコンソールポート

コンソールサービスを管理するのはサービスアクセスコントローラではなく、また、明示的に ttymon 管理ファイルを実行して管理するのでもありません。/etc/inittab ファイル内のエントリを使用して、ttymon を express モードで使用するコンソールポートを管理します。express モードとは、ttymon の特別なモードのことで、ログインサービスが必要なコマンドによって直接呼び出されます。

/etc/inittab ファイル内のデフォルトのコンソールエントリは、次のようになります。

```
co:234:respawn:/usr/lib/saf/ttymon -g -h -p "`uname -n` console login: "
 -T terminal_type -d /dev/console -l console -m ldterm,ttcompat
```

`co:234:respawn:`	co は、エントリをコンソールとして指定する。234 は、動作 respawn の実行レベルを指定する。つまり失敗した場合、あるいは実行レベル 2、3、および 4 でない場合、このコンソールエントリが再起動されるべきであることを示す
`/usr/lib/saf/ttymon -g -h`	–g オプションを使用するため、正しいボーレートと正しい端末設定をポート上で設定でき、SAC による事前構成なしに、ログインサービスに接続できる。–h オプションは、デフォルトまたは指定した速度に設定する前に、回線速度をゼロに設定することにより、回線をハングアップさせる
`–p "`‘`uname -n`‘ `console login:`	コンソールポート用のプロンプト文字列を指定する
–t*terminal_type*	コンソールの端末タイプを指定する
`–d /dev/console -l console -m ldterm,ttcompat`	–d オプションは、コンソールデバイスを指定する。–l オプションは、/etc/ttydefs ファイル内の tty 名を指定する。–m オプションは、プッシュする STREAMS モジュールを指定する

ttymon 固有の管理コマンド: ttyadm

ttymon の管理ファイルは、sacadm および pmadm の他に ttyadm コマンドによっても更新できます。ttyadm コマンドは、ttymon 固有の情報の書式を定義し、それらの情報を標準出力に書き込み、書式付きの ttymon 固有のデータを sacadm および pmadm コマンドに提示する手段を提供します。

したがって、ttyadm は ttymon を直接管理するのではなく、一般的な管理用コマンドである sacadm および pmadm を補足するものです。詳細は、ttyadm(1M) のマニュアルページを参照してください。

ネットワークリスナーサービス: listen

listen ポートモニターは SAC プログラムのもとで実行されます。このモニターは、ネットワークを監視してサービス要求がないかを調べ、入ってきたら要求を受け付け、それらのサービス要求に応答してサーバーを呼び出します。

listen ポートモニターは sacadm コマンドを使用して構成します。listen の各インスタンスはそれぞれに複数のサービスを提供できます。それらのサービスは listen ポートモニターの管理ファイルに指定します。この管理ファイルは pmadm および nlsadmin コマンドを使用して構成します。

ネットワークリスナープロセスは、トランスポート層インタフェース (TLI) 仕様に準拠する任意の接続型トランスポートプロバイダで使用できます。Solaris 環境では、listen ポートモニターは、inetd が提供しない追加ネットワークサービスを提供します。

listen 固有の管理コマンド: nlsadmin

listen ポートモニターの管理ファイルは、sacadm および pmadm の他に、nlsadmin コマンドでも更新できます。nlsadmin コマンドは、listen 固有の情報の書式を定義して標準出力に書き込み、書式付きの listen 固有のデータを sacadm および pmadm コマンドに提示する手段を提供します。

したがって、nlsadmin は listen を直接管理するのではなく、一般的な管理用コマンドである sacadm および pmadm を補足するものです。

各ネットワークには、ネットワークリスナープロセスのインスタンスが少なくとも 1 つは存在します。各ネットワークはそれぞれ個別に構成されます。nlsadmin コマンドは listen ポートモニターの動作状態を制御します。

nlsadmin コマンドは、与えられたネットワークに対して listen ポートモニターを設定し、そのポートモニターの固有の属性を構成し、そのモニターを起動したり、強制終了させたりすることができます。さらに、マシン上にある listen ポートモニターについて報告することもできます。

詳細は、nlsadmin(1M) のマニュアルページを参照してください。

ttymon ポートモニターの管理

sacadm コマンドを使用して ttymon ポートモニターを追加、表示、削除、終了、起動、あるいは有効または無効にすることができます。

注 - 次の手順を行うには、スーパーユーザーでなければなりません。

▼ ttymon ポートモニターを追加する方法

ttymon ポートモニターを追加するには、次のように入力します。

```
# sacadm -a -p mbmon -t ttymon -c /usr/lib/saf/ttymon -v `ttyadm
-V` -y "TTY Ports a & b"
```

-a　　ポートモニターフラグを追加する。

-p　　mbmon をポートモニタータグとして指定する。

-t　　ポートモニタータイプを ttymon として指定する。

-c　　ポートモニターを起動するのに使用するコマンド文字列を定義する。

-v　　ポートモニターのバージョン番号を指定する。

-y　　ポートモニターのインスタンスを説明するコメントを指定する。

▼ ttymon ポートモニターの状態を表示する方法

ttymon ポートモニターの状態を表示するには、次のように入力します。

```
# sacadm -l -p mbmon
```

-l　　ポートモニターの状態フラグを表示する。

-p　　mbmon をポートモニターのタグとして指定する。

例 - ttymon ポートモニターの状態を表示する

```
# sacadm -l -p mbmon
PMTAG   PMTYPE   FLGS RCNT STATUS     COMMAND
mbmon   ttymon   -    0    STARTING   /usr/lib/saf/ttymon #TTY Ports a & b
```

PMTAG mbmon	ポートモニター名 mbmon を指定する。
PMTYPE ttymon	ポートモニターのタイプ ttymon を指定する。
FLGS -	次の 2 つのフラグが設定されているかどうかを示す。 d は、新しいポートモニターを有効にしない。 x は、新しいポートモニターを起動しない。この例では、どちらのフラグも設定されていない。
RCNT 0	戻りカウント値を示す。0 の戻りカウントは、ポートモニターが失敗した場合でも再起動しないことを示す。
STATUS STARTING	ポートモニターの現在の状態を示す。
COMMAND /usr/lib/saf ...	ポートモニターを起動するコマンドを指定する。
#TTY Ports a & b	ポートモニターを説明するコメントを指定する。

▼ ttymon ポートモニターを停止する方法

ttymon ポートモニターを終了させるには、次のように入力します。

```
# sacadm -k -p mbmon
```

-k　ポートモニターの状態フラグを終了する。

-p　mbmon をポートモニターのタグとして指定する。

▼ ttymon ポートモニターを起動する方法

終了した ttymon ポートモニターを起動するには、次のように入力します。

```
# sacadm -s -p mbmon
```

-s　　ポートモニターの状態フラグを起動する。

-p　　mbmon をポートモニターのタグとして指定する。

▼ ttymon ポートモニターを無効にする方法

ポートモニターを無効にすると、以前から存在しているサービスをそのまま有効にするため、新しいサービスが起動できなくなります。

ttymon ポートモニターを無効にするには、次のように入力します。

```
# sacadm -d -p mbmon
```

-d　　ポートモニターの状態フラグを無効にする。

-p　　mbmon をポートモニターのタグとして指定する。

▼ ttymon ポートモニターを有効にする方法

ttymon ポートモニターを有効にすると、そのモニターが新しい要求にサービスを提供できるようになります。

ttymon ポートモニターを有効にするには、次のように入力します。

```
# sacadm -e -p mbmon
```

-e　　ポートモニターの状態フラグを有効にする。

-p　　mbmon をポートモニターのタグとして指定する。

▼ ttymon ポートモニターを削除する方法

ttymon ポートモニターを削除するには、次のように入力します。

```
# sacadm -r -p mbmon
```

-r　　ポートモニターの状態フラグを削除する。

-p　　mbmon をポートモニターのタグとして指定する。

注 - ポートモニターを削除すると、それに関連するすべての構成ファイルが削除されます。ポートモニター構成ファイルは sacadm では更新や変更ができません。ポートモニターを再構成するには、古いポートモニターを削除してから新しいポートモニターを追加してください。

---

ttymon サービスの管理

pmadm コマンドを使用してサービスを追加したり、ポートモニターに関連付けられている 1 つ以上のポートのサービスを表示したり、サービスを有効または無効にしたりできます。

注 - 次の手順を行うにはスーパーユーザーでなければなりません。

▼ サービスを追加する方法

標準の端末サービスを mbmon ポートモニターに追加するには、次のように入力します。

```
# pmadm -a -p mbmon -s a -i root -v `ttyadm -V` -m "`ttyadm -i 'Terminal disabled'
 -l contty -m ldterm,ttcompat -S y -d /dev/term/a -s /usr/bin/login`"
```

注 - 上記の入力例では、contty の後が次の行にまたがっていますが、実際には Return キーを押さずに (改行なしに) 入力します。

`-a` ポートモニターの状態フラグを追加 (add) する。

`-p` `mbmon` をポートモニタータグとして指定する。

`-s` `a` をポートモニターサービスタグとして指定する。

`-i` 識別情報を、実行中にポートモニターサービスタグに割り当てられるように指定する。

`-v` ポートモニターのバージョン番号を指定する。

`-m` `ttymon` により書式化された `ttyadm` 固有の構成データを指定する。

上記の `pmadm` コマンドには `ttyadm` コマンドが組み込まれています。その組み込みコマンドの中の指定項目の意味は次のとおりです。

`-b` ポートフラグを発着信両用に指定する。

`-i` 無効応答メッセージを指定する。

`-l` `/etc/ttydefs` にあるどの TTY 名を使用するか指定する。

`-m` サービスを起動する前にプッシュする STREAMS モジュールを指定する。

`-d` TTY ポートに使用する装置へのフルパス名を指定する。

`-s` 接続要求を受信したとき起動するサービスへのフルパス名を指定する。引数が必要な場合、コマンドと引数を二重引用符 (") で囲む。

▼ TTY ポートサービスの状態を表示する方法

`pmadm` コマンドを次に示すように使用して、特定のポートモニターに設定されている 1 つまたはすべての TTY ポートを表示します。

1 つのサービスを表示する場合

ポートモニターの 1 つのサービスを表示するには、次のように入力します。

```
# pmadm -l -p mbmon -s a
```

-l　　サービス情報を表示するフラグ

-p　　mbmon をポートモニタータグとして指定する

-s　　a をポートモニターサービスタグとして指定する

すべてのポートモニターのすべてのサービスを表示する場合

すべてのポートモニターのすべてのサービスを表示するには、次のように入力します。

```
# pmadm -l
```

-l　　サービス情報を表示するフラグ

特定のポートモニターのすべてのサービスを表示する場合

特定のポートモニターのすべてのサービスを表示するには、次のように入力します。

```
# pmadm -l -p mbmon
```

-l　　サービス情報を表示するフラグ

-p　　mbmon をポートモニタータグとして指定する

例 - 特定のポートモニターのすべてのサービスを表示する

```
# pmadm -l -p mbmon
PMTAG   PMTYPE   SVCTAG FLAGS ID     <PMSPECIFIC>
mbmon   ttymon   a      -     root   /dev/term/a - - /usr/bin/login - contty
ldterm,ttcompat login:  Terminal disabled - y  #
```

mbmon	pmadm -p コマンドを使用して設定された、ポートモニター名 mbmon を指定する
ttymon	ポートモニターのタイプ ttymon を指定する
a	pmadm -s コマンドを使用して設定された、サービスタグ値を示す
-	次のフラグが pmadm -f コマンドを使用して設定されているかどうかを指定する。 x — サービスを有効にしないことを意味する u — サービス用の utmpx エントリを作成することを意味する。この例では、どちらのフラグも設定されていない
root	起動時にサービスに割り当てられた ID を指定する。この値は、pmadm -i コマンドを使用して設定される
<PMSPECIFIC> *Information*	
/dev/term/a	ttyadm -d コマンドを使用して設定された、TTY ポートパス名を示す
-	次のフラグが ttadm -c -b -h -I -r コマンドを使用して設定されているかどうかを示す。 c — ポート用のキャリアフラグに接続を設定する b — ポートが双方向性である (着信トラフィックと発信トラフィックの両方を許可する) と設定する h — 着呼が受信された直後の自動ハングアップを抑制する I — ポートを初期化する r — login: メッセージを出力する前に、ポートから文字を受信するまで、ttymon を待機させる
-	ttyadm -r オプションを使用して設定された値を示す。このオプションは、ポートからデータを受信後に、ttymon がプロンプトを表示するときを決定する。カウントが 0 の場合、ttymon は、任意の文字を受信するまで待機する。カウントが 0 より大きい場合、ttymon は、カウント新規行を受信するまで待機する。この例では、値は設定されていない
/usr/bin/login	接続を受信したときに呼び出されるサービスのフルパス名を指定する。この値は、ttyadm -s コマンドを使用して設定される

-	ttyadm -t コマンドの (タイムアウト) 値を指定する。このオプションは、ポートを開くことが成功した場合に、ttymon がポートを閉じること、および入力データがタイムアウト秒内に受信されていないことを指定する。この例では、タイムアウト値は設定されていない
contty	/etc/ttydefs ファイル中の TTY 名を指定する。この値は、ttyadm -l コマンドを使用して設定される
ldterm,ttcompat	プッシュする STREAMS モジュールを指定する。これらのモジュールは、ttyadmin -m コマンドを使用して設定される
login: Terminal disabled	ポートが無効であるときに表示される、アクティブでないメッセージを指定する。このメッセージは、ttyadm -i コマンドを使用して設定される
tvi925	ttyadm -T コマンドを使用して設定されている場合、端末タイプを指定する。この例では、端末タイプは、tvi925
y	ttyadm -S コマンドを使用して設定されたソフトウェアキャリア値を指定する。n は、ソフトウェアキャリアをオフにする。y は、ソフトウェアキャリアをオンにする。この例では、ソフトウェアキャリアはオン
#	pmadm -y コマンドで指定した任意のコメントを指定する。この例では、コメントは存在しない

▼ ポートモニターサービスを有効にする方法

無効になっているポートモニターサービスを有効にするには、次のように入力します。

```
# pmadm -e -p mbmon -s a
```

-e　　フラグを有効にする。

-p　　mbmon をポートモニタータグとして指定する。

-s　　a をポートモニターサービスタグとして指定する。

▼ ポートモニターサービスを無効にする方法

ポートモニターサービスを無効にするには、次のように入力します。

```
# pmadm -d -p mbmon -s a
```

-d　　フラグを無効にする。

-p　　mbmon をポートモニタータグとして指定する。

-s　　a をポートモニターサービスタグとして指定する。

サービスアクセス機能管理のための参照情報

SAF の関連ファイル

SAF は構成ファイルを使用しますが、このファイルは sacadm および pmadm コマンドを使用して変更できます。構成ファイルは手作業で編集する必要はありません。

ファイル名	説明
/etc/saf/_sysconfig	システムごとの構成スクリプト
/etc/saf/_sactab	SAC の管理ファイル。SAC が制御するポートモニターの構成データを内容とする
/etc/saf/*pmtag*	ポートモニター *pmtag* のホームディレクトリ

ファイル名	説明
/etc/saf/*pmtag*/_config	存在する場合、ポートモニター *pmtag* のポートモニターごとの構成スクリプト
/etc/saf/*pmtag*/*_pmtab*	ポートモニター *pmtag* の管理ファイル。*pmtag* が提供するサービスのポートモニター固有の構成データを内容とする
/etc/saf/*pmtag*/*svctag*	サービス *svctag* のサービスごとの構成スクリプト
/var/saf/log	SAC のログファイル
/var/saf/*pmtag*	*pmtag* によって作成されるファイルのディレクトリ。たとえば、ログファイルのディレクトリなど

/etc/saf/_sactab ファイル

/etc/saf/_sactab は、次のようになります。

```
# VERSION=1
zsmon:ttymon::0:/usr/lib/saf/ttymon     #
```

# VERSION=1	サービスアクセス機能のバージョン番号を指定する
zsmon	ポートモニター名
tcp	
ttymon	ポートモニターのタイプ
::	次の 2 つのフラグが設定されているかどうかを示す。 d — ポートモニターを有効にしない x — ポートモニターを起動しない。この例では、どちらのフラグも設定されていない

0 999	戻りコード値を示す。0 の戻りカウントは、ポートモニターが失敗した場合でも再起動しないことを示す
/usr/lib/saf/ttymon /usr/lib/saf/listen	ポートモニターのパス名を示す

/etc/saf/*pmtab*/*_pmtab* ファイル

/etc/saf/*pmtab*/*_pmtab* ファイル (/etc/saf/zsmon/_pmtab など) は、次のようになります。

```
# VERSION=1
ttya:u:root:reserved:reserved:reserved:/dev/term/a:I::/usr/bin/login::9600:ldterm,
ttcompat:ttya login\: ::tvi925:y:#
```

# VERSION=1	サービスアクセス機能のバージョン番号を示す
ttya	サービスタグを示す
x,u	次のフラグが設定されているかどうかを指定する。 x — サービスを有効にしないことを意味する u — サービス用の utmpx エントリを作成することを意味する
root	サービスタグに割り当てられた ID を示す
reserved	このフィールドは予約されている
reserved	このフィールドは予約されている
reserved	このフィールドは予約されている
/dev/term/a	TTY ポートパス名を示す
/usr/bin/login	接続を受信したときに呼び出されるサービスのフルパス名を指定する

`:c,b,h,I,r:`	次のフラグが設定されているかどうかを示す。 `c` — ポート用のキャリアフラグに接続を設定する `b` — ポートが双方向性である (着信トラフィックと発信トラフィックの両方を許可する) と設定する `h` — 着呼が受信された直後の自動ハングアップを抑制する `I` — ポートを初期化する `r` — `login:` メッセージを出力する前に、ポートから文字を受信するまで、`ttymon` を待機させる
`9600`	`/etc/ttydefs` ファイルに定義されている TTY 名を指定する
`ldterm,ttcompat`	プッシュする STREAMS モジュールを指定する
`ttya login\:`	表示するプロンプトを指定する
`:y/n:`	
message	任意のアクティブでない (無効な) 応答メッセージを指定する
`tvi925`	端末タイプを指定する
`y`	ソフトウェアキャリアが設定されているかどうかを示す (`y/n`)

サービスの状態

`sacadm` コマンドはサービスの状態を制御します。次にそれらの状態を示します。

状態	意味
有効	デフォルト状態 – ポートモニターを追加したとき、サービスが有効になる
無効	デフォルト状態 – ポートモニターを削除したとき、サービスは停止する

特定のサービスの状態を知るには、次のように入力します。

```
# pmadm -l -p portmon_name -s svctag
```

ポートモニターの状態

sacadm コマンドは、ttymon および listen ポートモニターの状態を制御します。次に示す状態があります。

状態	意味
起動	デフォルト状態 – ポートモニターは追加されると自動的に起動される
有効	デフォルト状態 – ポートは追加されると自動的にサービス要求を受け付け可能になる
停止	デフォルト状態 – ポートモニターは削除されると自動的に停止する
無効	デフォルト状態 – ポートモニターは削除されると自動的に提供中であったサービスを続行し、新しいサービスの追加を拒否する
起動中	中間状態 – ポートモニターの起動が進行中
停止中	中間状態 – ポートモニターは手作業で終了過程に入っているが、まだシャットダウン手続きは完了していない。停止状態になるまでの途中の状態
非動作中	アクティブではない状態 – ポートモニターが強制終了された状態。前の動作状態のときに監視していたすべてのポートがアクセス不可になる。外部のユーザーからはポートが無効なのか、非動作状態なのか区別できない
障害	アクティブではない状態 – ポートモニターを起動して動作状態を維持できない

特定のポートモニターの状態を知るには、次のように入力します。

```
# sacadm -l -p portmon_name
```

ポートの状態

ポートは、ポートを制御するポートモニターの状態によって、有効または無効にできます。

状態	注記
シリアル (ttymon) ポートの状態	
有効	ttymon ポートモニターはポートにプロンプトメッセージを送り、ログインサービスを提供する
無効	ttymon が強制終了されているか、無効の場合のすべてのポートのデフォルト状態。このように指定した場合、接続要求を受け取ると、ttymon は「disabled」メッセージを送信する

第 15 章

システムセキュリティの管理

ここでは、Solaris 環境においてシステムセキュリティを管理する方法について説明します。次の章が含まれます。

第 16 章	ファイル、システム、およびネットワークのセキュリティの概要について説明します。
第 17 章	ファイル情報を表示し、ファイルの所有権とアクセス権を変更し、特殊なアクセス権を設定する手順を説明します。
第 18 章	ログイン状態をチェックし、ダイヤルアップパスワードを設定し、root へのアクセスを制限し、root アクセスと `su` コマンドの試行を監視する手順を説明します。
第 19 章	役割によるアクセス制御の概要と、使用する手順を説明します。
第 20 章	Kerberos ログイン認証と Pluggable Authentication Module (PAM) を設定する手順を説明します。
第 21 章	Sun Enterprise Authentication Mechanism (SEAM) セキュリティ製品の概要を説明します。
第 22 章	使用するネットワークで SEAM を構成する手順を説明します。

第 23 章	SEAM セキュリティ製品に関する参照情報を提供します。
第 24 章	自動セキュリティ拡張ツール (ASET) についての概要と、ASET を対話形式で、または定期的に (cron ジョブを使用して) 実行する手順を説明します。クライアントの ASET レポートをサーバー上で収集する方法についても説明します。

第 16 章

システムセキュリティの管理の概要

システム情報のセキュリティを保つことは、重要なシステム管理作業です。この章では、ファイルレベル、システムレベル、およびネットワークレベルでシステムセキュリティを管理する方法について説明します。

この章の内容は以下のとおりです。

Solaris システムセキュリティでの新機能

この節では、新しいセキュリティ機能について説明します。

システムファイルとシステムディレクトリに対する新しいデフォルトの所有権とアクセス権

この Solaris リリースでは、多くのシステムファイルとシステムディレクトリのデフォルト所有権が以前のリリースから変更になり、アクセス権が以前のリリースより厳格になりました。デフォルトの所有権とアクセス権は次のように変更されました。

- ファイルとディレクトリのデフォルト所有権は bin から root に変わった
- デフォルトのアクセス権が従来 775 であったファイルとディレクトリは、デフォルトのアクセス権が 755 になる
- デフォルトのアクセス権が従来 664 であったファイルとディレクトリは、デフォルトのアクセス権が 644 になる
- システムのデフォルト umask は 022

Solaris 8 が動作するシステムに追加するパッケージを作成する場合は、次の点に注意してください。

- すべてのファイルとディレクトリのデフォルト所有者は root でなければならない
- ディレクトリと実行可能ファイルのデフォルトアクセス権は 555 か 755 でなければならない
- 通常ファイルのデフォルトアクセス権は 644 か 444 でなければならない
- `setuid` や `setgid` 所有権をもつファイルに所有者が書き込むためには、所有者が root でなければならない

これらの変更がこのリリースのすべてのファイルやディレクトリに適用されるわけではありません。たとえば、OpenWindows や CDE のファイルやディレクトリには、これらの変更は適用されません。

役割によるアクセス制御

役割によるアクセス制御 (RBAC) は、スーパーユーザー特権をパッケージ化してユーザーアカウントに割り当てる柔軟な方法を提供します。この方法を利用すれば、特定の問題を解決しなければならないユーザーにすべてのスーパーユーザー特権を与える必要がなくなります。

詳細は、第 19 章を参照してください。

Sun Enterprise Authentication Mechanism (SEAM) または Kerberos V5 クライアントサポート

この機能は、Pluggable Authentication Module (PAM) を拡張する Kerberos V5 クライアント側インフラストラクチャと、NFS サービスなど RPC に基づくアプリケーションのセキュリティを保護するユーティリティプログラムを提供します。Kerberos は、ユーザーやサーバーレベルでの強力な認証、統合、またはプライバシサポートを提供します。選択可能な SEAS 3.0 の一部である Sun Enterprise Authentication Mechanism (SEAM) または Kerberos V5 ソフトウェア (たとえば、MIT ディストリビューション) と Kerberos クライアントをともに使用すれば、ネットワークへの完全なシングルサインオンを実現できます。

詳細は、第 21 章を参照してください。

システムセキュリティ作業の参照先

システムセキュリティの設定手順については、次の項目を参照してください。

- 第 17 章
- 第 18 章
- 第 19 章
- 第 20 章
- 第 24 章

コンピュータシステムへのアクセスを制御する

ファイルレベルでは、SunOS 5.8 オペレーティングシステムにいくつかの標準セキュリティ機能が組み込まれているため、ファイル、ディレクトリ、およびデバイスの保護に使用できます。システムレベルとネットワークレベルでは、セキュリティの内容はほぼ同じです。サイトでは、1 台のサーバーに接続された多数のシステムを 1 つの大規模で多面的なシステムと見なすことができます。システム管理者は、この大規模なシステム、つまりネットワークシステムのセキュリティ管理に責任があります。ネットワークの外側からの侵入を防ぐことだけでなく、ネットワーク内部のシステムのデータの完全性を確保することも重要です。

防御の第 1 歩は、システムへのアクセスを制御することです。次の方法でシステムへのアクセスを制御または監視できます。

- サイトの物理的なセキュリティの管理
- ログイン制御の管理
- ファイル内のデータへのアクセス制限
- ネットワーク制御の管理
- システムの使用状況の監視
- 正しいパス変数の設定
- ファイルの保護
- ファイアウォールのインストール
- セキュリティ問題の報告

サイトの物理的なセキュリティの管理

システムへのアクセスを制御するには、コンピュータ環境の物理的なセキュリティを管理しなければなりません。たとえば、システムにログイン後そのままそこから離れてしまうと、そのシステムを使用できるユーザーであれば誰でもオペレーティングシステムとネットワークにアクセスできます。コンピュータの周囲に注意して、許可されていないアクセスから物理的に保護する必要があります。

ログインとアクセス制御の管理

システムやネットワークへの許可されていないログインも制限する必要がありますが、この作業はパスワードとログイン制御を使用して実行できます。システム上のすべてのアカウントには、パスワードを設定しなければなりません。アカウントにパスワードを設定しないと、ユーザー名を推測できるユーザーであれば誰でもネットワーク全体にアクセスできることになります。

Solaris ソフトウェアでは、特定のシステムデバイスの制御をユーザーのログインアカウントに制限しています。`/etc/logindevperm` を編集しない限り、スーパーユーザーまたはコンソールユーザーとして実行中のプロセス以外は、システムのマウス、キーボード、およびフレームバッファーにアクセスできません。詳細は、`logindevperm(4)` のマニュアルページを参照してください。

ファイル内のデータへのアクセス制限

ログイン制限を設定したら、システム上のデータへのアクセスを制御できます。一部のユーザーには特定のファイルの読み取りを許可し、他のユーザーには特定のファイルを変更または削除するアクセス権を与えることができます。誰にも見せたくないデータがある場合もあります。ファイルのアクセス権の設定方法については、第 17 章を参照してください。

ネットワーク制御の管理

通常、コンピュータは「ネットワーク」と呼ばれるシステム構成の一部です。ネットワーク上では、接続されているシステムは、そのネットワークに接続されている他のシステムと情報を交換し、相手のデータや他の資源にアクセスできます。ネットワーク化することによって、コンピュータの処理能力と性能が高まります。しかし、コンピュータのセキュリティが危険にさらされる可能性もあります。

たとえば、ネットワーク内では、個々のシステムは情報を共有できるように開放されています。また、多数の人々がネットワークにアクセスするので、特にパスワードの誤用などのユーザーエラーを通じて、不要なアクセスが発生する可能性も大きくなります。

システム使用状況の監視

システム管理者は、次のようにシステムのあらゆる側面に注意してシステムの活動を監視する必要があります。

- 通常の負荷はどの程度か
- 誰がシステムへのアクセス権を持っているか
- 各ユーザーはいつシステムにアクセスするか

この種の情報を把握していれば、ツールを使用してシステムの使用状況を監査し、各ユーザーの活動を監視できます。セキュリティ違反が疑われる場合は、監視作業が特に役立ちます。

正しいパスの設定

パス変数を正しく設定することが重要です。正しく設定しないと、他人が持ち込んだプログラムを偶然に実行して、データやシステムを破壊する可能性があります。

この種のプログラムはセキュリティ上の危険を招くので、「トロイの木馬」と呼ばれます。たとえば、公共のディレクトリの中に別の su プログラムを入れておくと、システム管理者が気づかずに実行してしまう可能性があります。この種のスクリプトは通常の su コマンドとまったく同じに見えます。実行後はスクリプトそのものが削除されるので、実際に「トロイ」の木馬を実行してしまったのかを調べるのは困難です。

パス変数は、ログイン時に起動ファイル .login、.profile、.cshrc により自動的に設定されます。カレントディレクトリ (.) への検索パスを最後に指定すれば、この種のトロイの木馬を実行するのを防ぐことができます。root のパス変数には、カレントディレクトリを指定しないでください。ASET ユーティリティは起動ファイルを検査して、パス変数が正しく設定されているかと、ドット (.) エントリが入っていないかを確認します。

ファイルの保護

SunOS 5.8 オペレーティングシステムはマルチユーザーシステムなので、ファイルシステムの保護は、システムの最も基本的で重要な問題です。ファイルの保護には、従来の UNIX のファイル保護と、より確実なアクセス制御リスト (ACL) の両方が使用できます。

また、多くの実行可能プログラムは、スーパーユーザー (root) として実行されなければ適切に動作しません。これらの実行可能プログラムは、ユーザー ID を 0 に設定して (setuid=0) 実行します。これらのプログラムを実行するユーザーは、root ID を使用するため、プログラムがセキュリティを念頭において作成されていない場合には、セキュリティ上の問題が発生する可能性があります。

setuid を root に設定した状態で把握されている実行可能プログラムを除き、setuid プログラムを使用不可にするか、少なくとも使用を最小限度に制限しておく必要があります。

ファイアウォールのインストール

ネットワークを保護するには、ファイアウォール、つまりセキュリティ保護ゲートウェイシステムを使用する方法もあります。ファイアウォールは 2 つのネットワークを分離する専用システムで、各ネットワークは相手に対し信頼されない (untrusted) ネットワークとしてアクセスします。内部ネットワークと、内部ネットワークユーザーに通信させたいインターネットなどの外部ネットワークとの間に、このような設定を必ず行うようにしてください。

ファイアウォールは、一部の内部ネットワーク間でも有効です。たとえば、ファイアウォール、つまりセキュリティ保護ゲートウェイコンピュータは、ゲートウェイコンピュータがパケットの発信元または宛先アドレスでない限り、2 つのネットワーク間でパケットを送信しません。また、ファイアウォールは、特定のプロトコルについてのみパケットを転送するように設定する必要があります。たとえば、パケットでメールを転送できるが、`telnet` や `rlogin` は転送できないようにできます。ASET ユーティリティは、高度なセキュリティを適用して実行すると、インターネットプロトコル (IP) パケットの転送機能を無効にします。

セキュリティ問題の報告

セキュリティ違反が発生したと思われる場合は、Computer Emergency Response Team/Coordination Center (CERT/CC) に連絡できます。これは、カーネギーメロン大学の Software Engineering Institute に Defense Advanced Research Projects Agency (DARPA) の後援で設立されたプロジェクトです。CERT/CC はセキュリティ問題の解決を支援できます。また、特定のニーズに合った他の Computer Emergency Response Team を紹介することもできます。CERT/CC に連絡するには、24 時間のホットラインに電話する方法と、電子メールを `cert@cert.sei.cmu.edu` に送る方法があります。

ファイルのセキュリティ

SunOS オペレーティングシステムはマルチユーザーシステムです。これは、システムにログインしたユーザーであれば、アクセス権を持っている限り誰でも他のユーザーのファイルを読み取って使用できることを意味します。表 16–1 では、ファイルシステム管理コマンドについて説明します。ファイルのセキュリティについては、第 17 章を参照してください。

ファイル管理コマンド

表 16–1 は、ファイルやディレクトリの監視およびセキュリティの設定に使用できるファイル管理コマンドを示します。

表 16–1 ファイル管理コマンド

コマンド	説明
`ls(1)`	ディレクトリ内のファイルとファイル情報を表示する
`chown(1)`	ファイルの所有権を変更する
`chgrp(1)`	ファイルのグループ所有権を変更する
`chmod(1)`	ファイルのアクセス権を変更する。記号モード (英字と記号) または絶対モード (8 進数) を使用して、ファイルのアクセス権を変更できる

ファイルの暗号化

重要なファイルをアクセスできないディレクトリに格納し (700 モード)、そのファイルを他のユーザーが読み取れないようにすると (600 モード)、ほとんどの場合はセキュリティが保たれます。しかし、他人がユーザーのパスワードや root パスワードを推測して発見すれば、そのファイルを読み書きできます。また、重要なファイルは、システムファイルのバックアップをテープにとるたびに、バックアップテープ上に保存されます。

アクセス制御リスト (ACL)

SunOS オペレーティングシステムの従来の UNIX ファイル保護機能では不十分な場合は、ACL によりファイルアクセス権の制御が強化されます。従来の UNIX ファイル保護機能は、所有者、グループ、その他という 3 つのユーザークラスに読み取り権、書き込み権、実行権を提供します。ACL を使用すると、所有者、所有者のグループ、その他、特定のユーザーおよびグループのファイルアクセス権を定義でき、またこれらのカテゴリごとにデフォルトのアクセス権を定義できるため、ファイルのセキュリティが強化されます。ACL を設定する個々の手順については、338ページの「アクセス制御リスト (ACL)」を参照してください。

表 16–2 に、ファイルやディレクトリに対して ACL を管理するコマンドを示します。

表 16–2 ACL コマンド

コマンド	説明
setfacl(1)	ACL エントリの設定、追加、変更、および削除を行う
getfacl(1)	ACL エントリを表示する

システムのセキュリティ

この節では、侵入者がシステムにログインするのを防ぐ方法、パスワードファイルを管理する方法、重要なシステムファイルとプログラムに対する許可されていないスーパーユーザーアクセスを防ぐ方法など、システムを許可されていないアクセスから保護する方法について説明します。

システム上で 2 つのセキュリティバリアを設定できます。第 1 のセキュリティバリアはログインプログラムです。このバリアをクリアしてシステムにアクセスするには、ローカルシステムまたはネームサービス (NIS または NIS+) で認識されるユーザー名と対応するパスワードを入力しなければなりません。

第 2 のセキュリティバリアは、システムファイルとプログラムをスーパーユーザーしか変更または削除できないように設定することです。root になろうとするユーザーは、スーパーユーザーのユーザー名とその正しいパスワードを入力しなければなりません。

ログインアクセスの制限

ユーザーがシステムにログインすると、ログインプログラムは /etc/nsswitch.conf ファイル内の情報に従って、該当するデータベースを照会します。このファイル内のエントリには、files (/etc 内のファイルを示す)、nis (NIS データベースを示す)、nisplus (NIS+ データベースを示す) を含めることができます。このファイルについては、『*Solaris* ネーミングの管理』または nsswitch.conf(4) のマニュアルページを参照してください。

ログインプログラムは、入力されたユーザー名とパスワードを確認します。ユーザー名がパスワードファイルに入っていない場合や、パスワードがユーザー名と一致していない場合は、システムへのアクセスが拒否されます。ユーザーがパスワードファイルから名前を入力し、パスワードがその名前の正しいパスワードであるときは、そのユーザーにシステムへのアクセス権が与えられます。

特別なログイン

システムにアクセスするには、従来のユーザーログインを使用する方法と、root ログインを使用する方法の 2 つが一般的です。また、多数の特別な「システム」ログインを使用すると、ユーザーは root アカウントを使用しなくても管理コマンドを実行できます。管理者は、これらのログインアカウントにパスワードを割り当てます。

表 16–3 に、システムのログインアカウントとその用途を示します。システムログインは特殊な機能を実行し、それぞれに固有のグループ識別子番号 (GID) が付いています。これらの各ログインには固有のパスワードを設定し、必要のある人だけに知らせるようにしてください。

表 16–3 システムログイン

ログインアカウント	**GID**	用途
`root`	`0`	ほぼ無制限で、他のすべてのログイン、保護、アクセス権より優先する。root アカウントはシステム全体へのアクセス権を持つ。root ログインのパスワードはきわめて厳密に保護する必要がある。ほとんどの Solaris コマンドを所有する
`daemon`	`1`	バックグラウンド処理を制御する
`bin`	`2`	ほとんどのコマンドを所有する
`sys`	`3`	多数のシステムファイルを所有する
`adm`	`4`	特定の管理ファイルを所有する
`lp`	`71`	プリンタ用のオブジェクトとスプールデータファイルを所有する

表 16–3　システムログイン　続く

ログインアカウント	GID	用途
uucp	5	UNIX 間のコピープログラム、UUCP 用のオブジェクトとスプールデータファイルを所有する
nuucp	9	システムにログインしてファイル転送を開始するためにリモートシステムで使用される

また、パスワードが必要な eeprom のセキュリティも設定する必要があります。詳細は、eeprom(1M) のマニュアルページを参照してください。

パスワード情報の管理

ユーザーはシステムにログインするときに、ユーザー名とパスワードの両方を入力しなければなりません。ログイン名は公開されますが、パスワードは秘密にしてユーザー以外には知られないようにします。また、ユーザーが各自のパスワードを慎重に選択し、頻繁に変更するようにしなければなりません。

パスワードは、最初にユーザーアカウントを設定するときに作成されます。ユーザーアカウントの機密性を保つために、パスワードの有効期間を設定し、ユーザーに各自のパスワードを定期的に変更させたり、パスワードをロックしてユーザーアカウントを使用できないようにすることもできます。パスワードの設定と管理については、『*Solaris* のシステム管理 *(*第 *1* 巻*)*』の「ユーザーアカウントとグループの管理 (概要)」と passwd(1) のマニュアルページを参照してください。

NIS+ パスワードファイル

ネットワークで NIS+ を使用している場合、パスワード情報は NIS+ データベースに格納されます。NIS+ データベース内の情報は、アクセス権を許可されたユーザーを制限することによって保護できます。AdminSuite™ 2.3 のユーザーアカウントマネージャまたは passwd コマンドを使用すると、ユーザーの NIS+ パスワードを変更できます。

NIS パスワードファイル

ネットワークで NIS を使用している場合、パスワードは NIS パスワードマップに格納されます。NIS では、パスワードの有効期間を指定できません。AdminSuite 2.3 のユーザーマネージャまたは `passwd` コマンドを使用すると、ユーザーの NIS パスワードを変更できます。

/etc ディレクトリ内のファイル

ネットワークで `/etc` 内のファイルを使用している場合、パスワード情報はシステムの `/etc/passwd` ファイルと `/etc/shadow` ファイルに格納されます。ユーザー名と他の情報は別の「シャドウ」ファイル `/etc/shadow` に格納されます。これは、ユーザーが暗号化されたパスワードにアクセスするのを防ぐセキュリティ上の手段です。`/etc/passwd` ファイルは、マシンにログインするユーザーであれば誰でも使用できますが、`/etc/shadow` ファイルを読み取ることができるのはスーパーユーザーだけです。AdminSuite 2.3 のユーザーマネージャ、Admintool、または `passwd` コマンドを使用すると、ローカルシステム上でユーザーのパスワードを変更できます。

制限付きシェルの使用

標準シェルを使用すると、ユーザーはファイルを開く、コマンドを実行するなどの操作を行うことができます。制限付きセルを使用すると、ユーザーによるディレクトリの変更やコマンドの実行を制限できます。制限付きシェル (`rsh`) は、ディレクトリ `/usr/lib` に入っています (これはリモートシェル `/usr/sbin/rsh` ではないので注意してください)。制限付きシェルには、通常のシェルに比べて次のような違いがあります。

- ユーザーはホームディレクトリに制限されます (`cd` を使用してディレクトリを変更できません)。
- ユーザーはシステム管理者が設定した `PATH` 内でしかコマンドを使用できません (`PATH` 変数を変更できません)。
- ユーザーはホームディレクトリとそのサブディレクトリ内のファイルにしかアクセスできません (完全パス名でコマンドやファイルを指定できません)。
- ユーザーは `>` または `>>` を使用して出力をリダイレクトできません。

制限付きシェルを使用すると、システム管理者はユーザーによるシステムファイルの操作を制限できます。このシェルは、主として特定の作業を実行しなければなら

ないユーザーを設定するためのものです。ただし、rsh は完全にセキュリティ保護されてはおらず、あくまでも経験の少ないユーザーが問題を起こさないようにするために使用します。

制限付きシェルについては、rsh(1) のマニュアルページを参照してください。

スーパーユーザー (root) ログインの追跡

システムには、スーパーユーザーモードに対して root パスワードが必要です。デフォルトの構成では、ユーザーはリモートのシステムに root としてログインできません。リモートログインするとき、ユーザーは自分のユーザー名でログインしてから、su コマンドを使用してスーパーユーザーにならなければなりません。これによって、管理者は、システム上でスーパーユーザー特権を使用している人を追跡できます。

スーパーユーザーまたは他のユーザーに切り替えようとするユーザーの監視

スーパーユーザーになりたい場合などは、su コマンドを使用して別のユーザーに変更する必要があります。セキュリティ上の理由から、su コマンドを使用中のユーザー、特にスーパーユーザーのアクセス権を取得しようとしているユーザーを監視する必要があります。

詳細は、360ページの「su コマンドを使用中のユーザーを監視する方法」を参照してください。

ネットワークのセキュリティ

ネットワーク上でのアクセスが容易になるほど、ネットワークシステムにとっては利点が増えます。ただし、データや資源に自由にアクセスして共有できる状況では、セキュリティ上の問題が生じます。一般にネットワークのセキュリティは、リモートシステムからの操作を制限またはブロックすることを指しています。図 16–1 に、リモート操作に適用できるセキュリティ制限を示します。

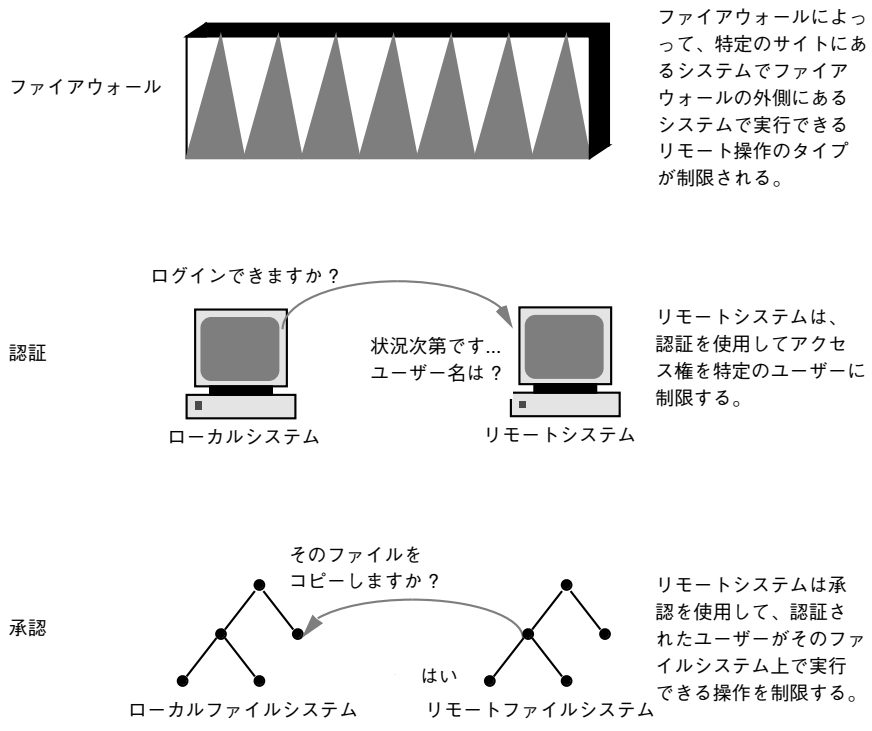


図 16–1　リモート操作のセキュリティ制限

ファイアウォールシステム

ファイアウォールシステムを設定すると、ネットワーク内のリソースを外部のアクセスから保護できます。「ファイアウォールシステム」は、内部ネットワークと外部ネットワークの間の防壁として機能するセキュリティ保護ホストです。

ファイアウォールには 2 つの機能があります。ネットワーク間でデータを渡すゲートウェイとして機能する一方で、データが勝手にネットワークを出入りしないようにブロックする防壁として機能します。ファイアウォールは、内部ネットワーク上のユーザーに対して、ファイアウォールシステムにログインしてリモートネットワーク上のホストにアクセスするように要求します。また、外部ネットワーク上のユーザーは、内部ネットワーク上のホストにアクセスする前に、ファイアウォールシステムにログインしなければなりません。

さらに、内部ネットワークから送信されるすべての電子メールは、ファイアウォールシステムに送信されてから、外部ネットワーク上のホストに転送されます。

ファイアウォールシステムは、すべての着信電子メールを受信して、内部ネットワーク上のホストに配信します。

注意 - ファイアウォールは、アクセス権のないユーザーが内部ネットワーク上のホストにアクセスする行為を防止します。ファイアウォールに適用される厳密で確実なセキュリティを管理しなければなりませんが、ネットワーク上の他のホストのセキュリティはもっと緩やかでもかまいません。ただし、ファイアウォールシステムを突破できる侵入者は、内部ネットワーク上の他のすべてのホストへのアクセスを取得できる可能性があります。

ファイアウォールシステムに「信頼される (trusted) ホスト」が含まれるべきではありません (「信頼されるホスト」とは、ユーザーがパスワードを入力しなくてもログインできるホストです)。ファイアウォールシステムは、ファイルシステムを共有してはならず、他のサーバーからファイルシステムをマウントしてはなりません。

自動セキュリティ拡張ツール (ASET) を使用すると、システムをファイアウォールにして高度なセキュリティを確保できます。詳細は、第 24 章を参照してください。

パケットスマッシング

ほとんどのローカルエリアネットワークでは、データはパケットと呼ばれるブロック単位でコンピュータ間で転送されます。アクセス権のないユーザーは、「パケットスマッシング」という方法により、データを損傷または破壊する可能性があります。パケットスマッシングでは、パケットは宛先に到達する前に捕捉され、その内容になんらかのデータが挿入されてから、元のコースに送り返されます。ローカルエリアネットワーク上では、パケットはサーバーを含むすべてのシステムに同時に到達するので、パケットスマッシングは不可能です。ただし、ゲートウェイ上ではパケットスマッシングが可能なので、ネットワーク上のすべてのゲートウェイを保護しなければなりません。

最も危険なのは、データの完全性に影響するような攻撃です。この種の攻撃を受けると、パケットの内容が変更されたり、ユーザーが偽装されたりします。会話を記録したり、後からユーザーを偽装せずに再生したりするなどの盗聴だけの場合、データの完全性は損なわれません。ただし、この種の攻撃はプライバシーに影響を及ぼします。ネットワーク上でやりとりされるデータを暗号化すると、重要な情報のプライバシーを保護できます。

認証と承認

「認証」とは、リモートシステムにアクセスできるユーザーを特定の人に限定する方法で、システムレベルまたはネットワークレベルで設定できます。ユーザーがリモートシステムにアクセスした後は、「承認」という方法でそのユーザーがリモートシステム上で実行できる操作が制限されます。表 16–4 に、ネットワーク上のシステムを許可されていない使い方から保護できる認証と承認の種類を示します。

表 16–4 認証と承認の種類

種類	説明	参照先
NIS+	NIS+ ネームサービスは、認証と承認をネットワークレベルで提供できる	『*Solaris* ネーミングの管理』
リモートログインプログラム	リモートログインプログラム (`rlogin`、`rcp`、`ftp`) を使用すると、ユーザーはネットワーク経由でリモートシステムにログインし、その資源を使用できる。「信頼される (trusted) ホスト」の場合、認証は自動的に処理されるが、それ以外の場合は自分自身を認証するように求められる	第 10 章
Secure RPC	Secure RPC を使用すると、リモートシステム上で要求を出したユーザーの認証が行われ、ネットワーク環境のセキュリティが高まる。Secure RPC には、UNIX、DES、または Kerberos 認証システムを使用できる	『*Solaris* のシステム管理 (第 *3* 巻)』
	Secure RPC を使用すると、NFS 環境に Secure NFS というセキュリティを追加できる	380ページの「NFS サービスと Secure RPC」
DES 暗号化	データ暗号化規格 (DES) 暗号化機能は 56 ビットのキーを使用して、秘密鍵を暗号化する	380ページの「DES 暗号化」
Diffie-Hellman 認証	この認証方法は、送信側のシステムの共通鍵を使用して現在の時刻を暗号化する機能を利用する。受信側のシステムは、現在の時刻で復号化およびチェックできる	381ページの「Diffie-Hellman 認証」

表 16–4　認証と承認の種類　続く

種類	説明	参照先
Kerberos Version 4	Kerberos は DES 暗号化を使用して、システムのログイン時にユーザーを認証する	第 20 章
AdminSuite 2.3	AdminSuite 2.3 には、認証機構と承認機構が組み込まれており、システムをリモート管理できる	『*Solstice AdminSuite 2.3* 管理者ガイド』

ファイルの共有

ネットワークファイルサーバーは、どのファイルを共有できるかを制御できます。また、共有ファイルにアクセスできるクライアント、それらのクライアントに許されるアクセスのタイプも制御できます。一般に、ファイルサーバーは、すべてのクライアントまたは特定のクライアントに、読み書きまたは読み取り専用権を与えることができます。アクセス制御は、`share` コマンドで資源を利用可能にするときに指定します。

サーバーでは、`/etc/dfs/dfstab` ファイルを使用して、ネットワーク上のクライアントに利用させることができるファイルシステムを表示できます。ファイルの共有の詳細は、『*Solaris* のシステム管理 *(*第 *3* 巻*)*』を参照してください。

スーパーユーザー (root) アクセスの制限

一般的にスーパーユーザーは、ネットワーク上で共有されるファイルシステムにはスーパーユーザーとしてアクセスできません。サーバーが特別にスーパーユーザー特権を与えなければ、クライアントにスーパーユーザーとしてログインしたユーザーは、そのクライアントにリモートでマウントされたファイルへのスーパーユーザーアクセスを取得できません。NFS システムは、要求側のユーザー ID をユーザー名 `nobody` のユーザー ID に変更してスーパーユーザーアクセスを提供します。一般に、`nobody` のユーザー ID は `60001` です。ユーザー `nobody` のアクセス権は、特定のファイルに関して公共 (つまり、資格を持たないユーザー) に与えられるものと同じです。たとえば、ファイルの実行しか公共に許可していなければ、ユーザー `nobody` はそのファイルを実行することしかできません。

NFS サーバーは、share コマンドの root=*hostname* オプションを使用して、共有ファイルシステムのスーパーユーザー特権をホスト単位で与えることができます。

特権付きポートの使用

Secure RPC を実行したくない場合は、代わりに Solaris の「特権付きポート」機構を使用できます。特権付きポートは、1024 未満のポート番号を持つスーパーユーザーによって設定されます。クライアントシステムは、クライアントの資格を認証した後で、特権付きポート経由でサーバーへの接続を設定します。その後、サーバーは接続のポート番号を検査してクライアントの資格を確認します。

ただし、Solaris 以外のクライアントは、特権付きポート経由で通信できないことがあります。その場合は、次のようなエラーメッセージが表示されます。

```
''Weak Authentication
NFS request from unprivileged port''
```

自動セキュリティ拡張ツール (ASET)

ASET セキュリティパッケージには、システムのセキュリティを制御して監視できるように、自動管理ツールが組み込まれています。ASET を実行するセキュリティレベルとして、低、中、または高レベルを指定できます。上のレベルほど、ASET のファイル制御機能が増え、ファイルアクセスが減少し、システムセキュリティが厳しくなります。

詳細は、第 24 章を参照してください。

第 17 章

ファイルのセキュリティの適用手順

この章では、ファイルにセキュリティを適用する手順について説明します。この章で説明する手順は次のとおりです。

ファイルのセキュリティに関する機能

この節では、ファイルのセキュリティを構成する機能について説明します。

ユーザークラス

各ファイルには、セキュリティのレベルを指定する 3 つのユーザークラスがあります。

- ファイルやディレクトリの所有者 — 通常は、ファイルを作成したユーザーです。ファイルの所有者は、ファイルの読み取り権、書き込み権(変更する権利)、または実行権 (コマンドの場合) を与えるユーザーを決定できます。
- グループのメンバー
- ファイルやグループの所有者以外のすべてのユーザー

ファイルのアクセス権を割り当てたり変更したりできるのは、スーパーユーザーかそのファイルの所有者だけです。

ファイルのアクセス権

表 17–1 に、各ユーザークラスに与えることができるファイルのアクセス権を示します。

表 17–1 ファイルのアクセス権

記号	アクセス権	指定されたユーザーが実行できる操作
r	読み取り	ファイルを開いて内容を読み取る
w	書き込み	ファイルに書き込んだり (その内容を変更したり)、追加したり、削除したりできる
x	実行	ファイルを実行できる (プログラムまたはシェルスクリプトの場合)、あるいは exec(1) システムコールの 1 つを使用してファイルを実行できる
-	拒否	ファイルを読み取ったり、書き込んだり、実行したりできない

これらのファイルアクセス権は、通常のファイルと同様にデバイス、ソケット、名前付きパイプ (FIFO) などの特殊ファイルにも適用できます。

シンボリックリンクには、そのリンクが指すファイルのアクセス権が適用されます。

ディレクトリのアクセス権

表 17–2 に、各ユーザークラスに与えることができるディレクトリのアクセス権を示します。

表 17–2 ディレクトリのアクセス権

記号	アクセス権	指定されたユーザーが実行できる操作
`r`	読み取り	ディレクトリ内のファイルを表示できる
`w`	書き込み	ディレクトリ内のファイルやリンクを追加または削除できる
`x`	実行	ディレクトリ内のファイルを開いたり実行したりできる。また、ディレクトリを作成し、その下にサブディレクトリを作成できる

ディレクトリへアクセスできないようにすると、そのディレクトリ (およびすべてのサブディレクトリ) 内のファイルを保護できます。ただし、スーパーユーザーはシステム上のすべてのファイルとディレクトリにアクセスできます。

特殊なファイルアクセス権 (`setuid`、`setgid`、スティッキビット)

実行可能ファイルと公共ディレクトリには、3 種類の特殊なアクセス権を設定できます。これらのアクセス権を設定すると、その実行可能ファイルを実行するユーザーは、そのファイルの所有者 (またはグループ) のユーザー ID を持つことができます。

特殊なアクセス権はセキュリティ上の問題を引き起こすため、特殊なアクセス権を設定するときは十分な注意が必要です。たとえば、ユーザーはユーザー ID が root に設定されているプログラムを実行することにより、スーパーユーザーのアクセス権を取得できます。また、すべてのユーザーは、所有するファイルに対して特殊なアクセス権を設定できるので、これもセキュリティ上の問題の原因となります。

setuid や setgid アクセス権を使用して、不正にスーパーユーザー権限が取得されていないかどうか絶えずシステムを監視しなければなりません。これらのアクセス権を使用しているすべてのプログラムを、ファイルシステム内で検索し、そのリストを出力する方法については、335ページの「setuid アクセス権が設定されているファイルを検索する方法」を参照してください。この種のプログラムの所有権を root や bin 以外のユーザーに与えているものが出力中にあれば、そのプログラムはセキュリティに違反している可能性があります。

setuid アクセス権

set-user 識別 (setuid) アクセス権を実行可能ファイルに設定すると、このファイルを実行するプロセスには、その実行可能ファイルを実行しているユーザーではなく、ファイルの所有者 (通常は root) に基づいてアクセス権が与えられます。このため、通常は所有者しか利用できないファイルやディレクトリにユーザーがアクセスできます。たとえば次に示すように、passwd コマンドは root の setuid アクセス権が設定されているので、ユーザーは root の権限でパスワードを変更できます。

```
-r-sr-sr-x   3 root     sys        104580 Sep 16 12:02 /usr/bin/passwd
```

これは、プロセスの実行が終了した後でも、高度な知識のあるユーザーは setuid プロセスによって与えられたアクセス権を維持する手段を見つけることができるため、セキュリティ上危険であることを示しています。

注 - プログラムから予約済み UID (0 – 99) で setuid アクセス権を使用しても、実効 UID は正しく設定されない場合があります。シェルスクリプトを代わりに使用するか、setuid アクセス権では予約済み UID を使用しないようにしてください。

---

setgid アクセス権

set-group 識別 (setgid) アクセス権は setuid に似ていますが、プロセスの実効グループ ID (GID) はファイルのグループ所有者に変更され、ユーザーにはそのグループに与えられたアクセス権に基づくアクセス権が与えられます。/usr/bin/mail プログラムには setgid アクセス権が設定されています。

```
-r-x--s--x   1 root     mail        63628 Sep 16 12:01 /usr/bin/mail
```

setgid アクセス権がディレクトリに適用されると、このディレクトリ内で作成されたファイルは、生成するプロセスが所属するグループではなく、ディレクトリが所属するグループに含まれることになります。ディレクトリ内で書き込み権と実行

権を持つユーザーは、そこでファイルを作成できます。ただし、そのファイルは、そのユーザーのグループではなく、ディレクトリを所有するグループに所属することになります。

管理者は、setuid アクセス権や setgid アクセス権の認証されていない使用によってスーパーユーザー特権が獲得されないようにシステムを監視しなければなりません。ファイルシステムを検索して、このようなアクセス権を使用しているすべてのプログラムのリストを出力する方法については、335ページの「setuid アクセス権が設定されているファイルを検索する方法」を参照してください。このようなプログラムの所有権が、root や bin ではなく、一般ユーザーになっているものが疑わしいと考えられます。これらのアクセス権を設定できるのは、スーパーユーザーだけです。

スティッキビット

「スティッキビット」は、ディレクトリ内のファイルを保護するアクセス権ビットです。ディレクトリにスティッキビットが設定されている場合、そのファイルを削除できるのはその所有者、ディレクトリの所有者、または root だけです。これにより、ユーザーは /tmp などの公共ディレクトリから他のユーザーのファイルを削除できなくなります。

```
drwxrwxrwt 7  root  sys   400 Sep  3 13:37 tmp
```

TMPFS ファイルシステム上で公共ディレクトリを設定するときには、スティッキビットを手作業で設定してください。

デフォルトの umask

ファイルやディレクトリを作成するときには、デフォルトのアクセス権が設定されます。これらのデフォルトのアクセス権は、システムファイル /etc/profile、.cshrc、または .login ファイル内の umask(1) の値によって決定されます。デフォルトでは、システムはテキストファイルのアクセス権を 666 に設定してユーザー、グループ、その他に読み取り権と書き込み権を与え、ディレクトリまたは実行可能ファイルに対しては 777 に設定します。

umask によって割り当てられる値は、デフォルトから差し引かれます。これには、chmod がアクセス権を与えるのと同じ方法で拒否する効果があります。たとえば、コマンド chmod 022 はグループとその他に書き込み権を与えますが、umask 022 はグループとその他の書き込み権を拒否します。

表 17–3 に、典型的な umask の設定とその設定が実行可能ファイルに与える影響を示します。

表 17–3 各種セキュリティレベルの umask 設定

セキュリティレベル	umask	使用できないユーザー
緩やか (744)	022	グループとその他による w
中程度 (740)	027	グループによる w、その他による rwx
中程度 (741)	026	グループによる w、その他による rw
厳しい (700)	077	グループとその他による rwx

ファイル情報の表示

この節では、ファイルの情報を表示する方法について説明します。

▼ ファイル情報を表示する方法

ls コマンドを使用して、ディレクトリ内のすべてのファイルに関する情報を表示します。

```
$ ls -la
```

-l	長形式で表示する
-a	ドット (.) で始まるファイルを含め、すべてのファイルを表示する

表示画面の各行には、ファイルに関して次の情報が表示されます。

- ファイル形式

  ファイルには 7 つの形式があります。表 17–4 にファイル形式を示します。

表 17–4 ファイル形式

記号	形式
-	テキストまたはプログラム
d	ディレクトリ
b	ブロック型特殊ファイル
c	キャラクタ型特殊ファイル
p	名前付きパイプ (FIFO)
l	シンボリックリンク
s	ソケット

- アクセス権 (表 17–1 と表 17–2 を参照)
- ハードリンク数
- ファイルの所有者
- ファイルのグループ
- ファイルのバイト数
- ファイルの作成日または前回の変更日
- ファイル名

例 — ファイル情報を表示する

次の例では、/sbin ディレクトリ内のファイルが部分的に表示されています。

```
$ cd /sbin
$ ls -la
total 13456
drwxr-xr-x   2 root     sys          512 Sep  1 14:11 .
drwxr-xr-x  29 root     root        1024 Sep  1 15:40 ..
-r-xr-xr-x   1 root     bin       218188 Aug 18 15:17 autopush
lrwxrwxrwx   1 root     root          21 Sep  1 14:11 bpgetfile -> ...
-r-xr-xr-x   1 root     bin       505556 Aug 20 13:24 dhcpagent
```

(続く)

```
-r-xr-xr-x   1 root     bin       456064 Aug 20 13:25 dhcpinfo
-r-xr-xr-x   1 root     bin       272360 Aug 18 15:19 fdisk
-r-xr-xr-x   1 root     bin       824728 Aug 20 13:29 hostconfig
-r-xr-xr-x   1 root     bin       603528 Aug 20 13:21 ifconfig
-r-xr-xr-x   1 root     sys       556008 Aug 20 13:21 init
-r-xr-xr-x   2 root     root      274020 Aug 18 15:28 jsh
-r-xr-xr-x   1 root     bin       238736 Aug 21 19:46 mount
-r-xr-xr-x   1 root     sys         7696 Aug 18 15:20 mountall
   .
   .
   .
```

ファイルの所有権の変更

この節では、ファイルの所有権を変更する方法について説明します。

▼ ファイルの所有者を変更する方法

1. スーパーユーザーになります。

   デフォルトでは、所有者は chown コマンドを使用して、ファイルやディレクトリの所有者を変更できません。ただし、システム管理者が次の行をシステムの /etc/system ファイルに追加して、システムをリブートすれば、所有者は chown コマンドを使用できるようになります。

   ```
   set rstchown = 0
   ```

   詳細は、chown(1) のマニュアルページを参照してください。また、NFS マウントされているファイルシステム上で所有者を変更するときは、他にも制約があるので注意してください。

2. chown コマンドを使用してファイルの所有者を変更します。

   ```
   # chown newowner filename
   ```

newowner	ファイルまたはディレクトリの新しい所有者のユーザー名または UID を指定する
filename	ファイルまたはディレクトリを指定する

3. ファイルの所有者が変更されていることを確認します。

```
$ ls -l filename
```

例 — ファイルの所有者を変更する

次の例は、myfile の所有権をユーザー rimmer に設定します。

```
$ chown rimmer myfile
# ls -l myfile
-rw-r--r--   1 rimmer   scifi   112640 May 24 10:49 myfile
```

▼ ファイルのグループ所有権を変更する方法

1. スーパーユーザーになります。

   デフォルトでは、所有者は chgrp コマンドを使用しても、ファイルのグループをその所有者が属するグループ以外には変更できません。たとえば、ファイルの所有者が staff と sysadm グループだけに属する場合、所有者は、ファイルのグループを staff か sysadm グループ以外には変更できません。

   ただし、システム管理者が次の行をシステムの /etc/system ファイルに追加して、システムをリブートすれば、所有者は、ファイルのグループを、所有者が属していないグループにも変更できるようになります。

```
set rstchown = 0
```

   詳細は、chgrp(1) のマニュアルページを参照してください。また、NFS マウントされているファイルシステムでグループを変更するときは、他にも制約があるので注意してください。

2. chgrp コマンドを使用して、ファイルのグループ所有者を変更します。

```
$ chgrp group filename
```

group	ファイルまたはディレクトリの新しいグループ名を指定する
filename	ファイルまたはディレクトリを指定する

グループ設定の詳細については、『*Solaris* のシステム管理 *(*第 *1* 巻*)*』の「ユーザーアカウントとグループの設定と管理 (手順)」を参照してください。

3. ファイルのグループ所有権が変更されていることを確認します。

```
$ ls -l filename
```

例 — ファイルのグループ所有権を変更する

次の例は、myfile のグループ所有権をグループ scifi に設定します。

```
$ chgrp scifi myfile
$ ls -l myfile
-rwxrw-- 1 rimmer scifi 12985 Nov 12 16:28 myfile
```

ファイルのアクセス権の変更

chmod コマンドを使用すると、ファイルのアクセス権を変更できます。ファイルまたはディレクトリの所有者、あるいはスーパーユーザーだけがそのアクセス権を変更できます。

chmod コマンドを使用して、次のどちらかのモードでアクセス権を設定できます。

- 絶対モード - ファイルのアクセス権を表す数値を使用します。これは、アクセス権を設定するときに最も一般的に使用される方法です。絶対モードを使用してアクセス権を変更するときは、3 つ 1 組のアクセス権を 8 進数で表します。
- 記号モード - 英字と記号の組み合わせを使用して、アクセス権を追加または削除します。

表 17–5 に、絶対モードでファイルのアクセス権を設定するための 8 進数値を示します。これらの数字を 3 つ組み合せて、所有者、グループ、その他のファイルアクセ

ス権をこの順に設定します。たとえば、値 644 は、所有者に対して読み取り権と書き込み権を設定し、グループとその他に対しては読み取り権だけを設定します。

表 17–5 絶対モードによるファイルのアクセス権の設定

8 進数値	ファイルのアクセス権	設定されるアクセス権
0	---	なし
1	--x	実行権のみ
2	-w-	書き込み権のみ
3	-wx	書き込み権と実行権
4	r--	読み取り権のみ
5	r-x	読み取り権と実行権
6	rw-	読み取り権と書き込み権
7	rwx	読み取り権、書き込み権、実行権

ファイルには、絶対モードまたは記号モードで特殊アクセス権を設定できます。絶対モードでは、3 つ 1 組のアクセス権の左端に新しい 8 進数値を追加して、特殊アクセス権を設定します。表 17–6 に、ファイルに特殊アクセス権を設定する 8 進数値を示します。

表 17–6 絶対モードによる特殊アクセス権の設定

8 進数値	特殊アクセス権の設定
1	スティッキビット
2	setguid
4	setuid

表 17–7 に、記号モードでファイルのアクセス権を設定するための記号を示します。記号では、アクセス権を設定または変更できる対象ユーザー、実行される操作、あるいは割り当てるまたは変更するアクセス権を指定できます。

表 17–7 記号モードによるファイルのアクセス権の設定

記号	機能	説明
u	対象ユーザー	ユーザー (所有者)
g	対象ユーザー	グループ
o	対象ユーザー	その他
a	対象ユーザー	すべて
=	操作	割り当て
+	操作	追加
-	操作	削除
r	アクセス権	読み取り
w	アクセス権	書き込み
x	アクセス権	実行
l	アクセス権	強制ロック、setgid ビットはオン、グループ実行ビットはオフ
s	アクセス権	setuid または setgid ビットはオン
S	アクセス権	suid ビットはオン、ユーザー実行ビットはオフ
t	アクセス権	スティッキビットはオン、その他の実行ビットはオン
T	アクセス権	スティッキビットはオン、その他の実行ビットはオフ

機能列の「対象ユーザー」、「操作」、および「アクセス権」を順番に連結した文字列によって、ファイルまたはディレクトリのアクセス権を変更する記号を指定します。

対象ユーザー	アクセス権を変更する対象となるユーザーを指定する
操作	実行する操作を指定する
アクセス権	変更するアクセス権を指定する

▼ アクセス権を絶対モードで変更する方法

1. ファイルまたはディレクトリの所有者でない場合は、スーパーユーザーになります。

   現在の所有者またはスーパーユーザーだけが、chmod コマンドを使用してファイルまたはディレクトリのアクセス権を変更できます。

2. chmod コマンドを使用してアクセス権を絶対モードで変更します。

```
$ chmod nnn filename
```

nnn	ファイル所有者、ファイルグループ、その他のアクセス権をこの順序で表す 8 進数値を指定する。有効な 8 進数値については、表 17–5 を参照
filename	ファイルまたはディレクトリを指定する

注 - chmod を使用して ACL エントリを持つファイルのファイルグループアクセス権を変更する場合、ファイルグループアクセス権と ACL マスクの両方が新しいアクセス権に変更されます。新しい ACL マスクアクセス権は、そのファイル上に ACL エントリを持つ追加ユーザーおよびグループの実効アクセス権を変更する場合があるので注意が必要です。getfacl(1) コマンドを使用して、すべての ACL エントリに適切なアクセス権が設定されていることを確認してください。

---

3. ファイルのアクセス権が変更されていることを確認します。

```
$ ls -l filename
```

例 — アクセス権を絶対モードで変更する

次の例は、公共ディレクトリのアクセス権を 744 (読み取り／書き込み／実行権、読み取り権のみ、読み取り権のみ) から 755 (読み取り／書き込み／実行権、読み取り／実行権、読み取り／実行権) に変更します。

```
$ ls -ld public_dir
drwxr--r--  1 ignatz   staff    6023 Aug  5 12:06 public_dir
$ chmod 755 public_dir
$ ls -ld public_dir
drwxr-xr-x  1 ignatz   staff    6023 Aug  5 12:06 public_dir
```

次の例は、実行可能シェルスクリプトのアクセス権を読み取り／書き込み権から読み取り／書き込み／実行権へ変更します。

```
$ ls -l my_script
-rw------- 1 ignatz   staff    6023 Aug  5 12:06 my_script
$ chmod 700 my_script
$ ls -l my_script
-rwx------ 1 ignatz   staff    6023 Aug  5 12:06 my_script
```

▼ 特殊アクセス権を絶対モードで変更する方法

1. ファイルまたはディレクトリの所有者でない場合は、スーパーユーザーになります。

   現在の所有者またはスーパーユーザーだけが、chmod コマンドを使用してファイルまたはディレクトリの所有者を変更できます。

2. chmod コマンドを使用して特殊アクセス権を絶対モードで変更します。

```
$ chmod nnnn filename
```

nnnn	ファイルまたはディレクトリのアクセス権を変更する 8 進数値。左端の 8 進数値で、ファイルの特殊アクセス権を設定する。特殊アクセス権に有効な 8 進数値のリストについては、表 17–6 を参照
filename	ファイルまたはディレクトリを指定する

注 - chmod を使用して ACL エントリを持つファイルのファイルグループアクセス権を変更する場合、ファイルグループアクセス権と ACL マスクの両方が新しいアクセス権に変更されます。新しい ACL マスクアクセス権は、そのファイル上に ACL エントリを持つ追加ユーザーおよびグループの実効アクセス権を変更する場合があるので注意が必要です。getfacl(1) コマンドを使用して、すべての ACL エントリに適切なアクセス権が設定されていることを確認してください。

---

3. ファイルのアクセス権が変更されていることを確認します。

```
$ ls -l filename
```

例 — 特殊アクセス権を絶対モードで設定する

次の例は、dbprog ファイルに setuid アクセス権を設定します。

```
$ chmod 4555 dbprog
$ ls -l dbprog
-r-sr-xr-x   1 db     staff        12095 May  6 09:29 dbprog
```

次の例は、dbprog2 ファイルに setgid アクセス権を設定します。

```
$ chmod 2551 dbprog2
$ ls -l dbprog2
-r-xr-s--x   1 db     staff       24576 May  6 09:30 dbprog2
```

次の例は、pubdir ディレクトリにスティッキビットアクセス権を設定します。

```
$ chmod 1777 pubdir
```

▼ アクセス権を記号モードで変更する方法

1. ファイルまたはディレクトリの所有者でない場合は、スーパーユーザーになります。

   現在の所有者またはスーパーユーザーだけが、chmod コマンドを使用してファイルまたはディレクトリの所有者を変更できます。

2. chmod コマンドを使用してアクセス権を記号モードで変更します。

```
$ chmod who operator permission filename
```

who operator permission	*who* では、アクセス権を変更するユーザーを指定し、*operator* では実行する操作を指定し、*permission* では変更後のアクセス権を指定する。有効な記号については、表 17–7 を参照
filename	ファイルまたはディレクトリを指定する

3. ファイルのアクセス権が変更されていることを確認します。

```
$ ls -l filename
```

例 — アクセス権を記号モードで変更する

次の例は、その他のユーザーの読み取り権を削除します。

```
$ chmod o-r filea
```

次の例は、ユーザー、グループ、その他のユーザーの読み取り権と実行権を追加します。

```
$ chmod a+rx fileb
```

次の例は、グループに読み取り権、書き込み権、および実行権を割り当てます。

```
$ chmod g=rwx filec
```

特殊なファイルアクセス権の検索

setuid や setgid アクセス権を使用して、不正にスーパーユーザー権限が取得されていないかどうか絶えずシステムを監視しなければなりません。この種のプログラムの所有権を root や bin 以外のユーザーに与えているものが出力中にあれば、そのプログラムはセキュリティに違反している可能性があります。

▼ setuid アクセス権が設定されているファイルを検索する方法

1. スーパーユーザーになります。

2. find コマンドを使用して setuid アクセス権が設定されているファイルを検索します。

```
# find directory -user root -perm -4000 -exec ls -ldb {} \; >/tmp/filename
```

find *directory*	指定したディレクトリから始めて、マウントされているすべてのパスをチェックする。ディレクトリとしてルート (/)、sys、bin、または mail を指定できる
-user root	root が所有するファイルのみを表示する
-perm -4000	アクセス権が 4000 に設定されているファイルのみを表示する
-exec ls -ldb	find コマンドの出力を ls -ldb 形式で表示する
>/tmp/*filename*	結果がこのファイルに書き込まれる

3. 結果を /tmp/***filename*** に出力する。

setuid については、322ページの「setuid アクセス権」を参照してください。

例 — setuid アクセス権が設定されているファイルを検索する

```
# find / -user root -perm -4000 -exec ls -ldb {} \; > /tmp/ckprm
# cat /tmp/ckprm
```

```
-r-sr-xr-x 1 root bin 38836 Aug 10 16:16 /usr/bin/at
-r-sr-xr-x 1 root bin 19812 Aug 10 16:16 /usr/bin/crontab
---s--x--x 1 root sys 46040 Aug 10 15:18 /usr/bin/ct
-r-sr-xr-x 1 root sys 12092 Aug 11 01:29 /usr/lib/mv_dir
-r-sr-sr-x 1 root bin 33208 Aug 10 15:55 /usr/lib/lpadmin
-r-sr-sr-x 1 root bin 38696 Aug 10 15:55 /usr/lib/lpsched
---s--x--- 1 root rar 45376 Aug 18 15:11 /usr/rar/bin/sh
-r-sr-xr-x 1 root bin 12524 Aug 11 01:27 /usr/bin/df
-rwsr-xr-x 1 root sys 21780 Aug 11 01:27 /usr/bin/newgrp
-r-sr-sr-x 1 root sys 23000 Aug 11 01:27 /usr/bin/passwd
-r-sr-xr-x 1 root sys 23824 Aug 11 01:27 /usr/bin/su
 #
```

アクセス権のないユーザー (rar) が /usr/bin/sh の個人用コピーを作成し、setuid としてのアクセス権を root に設定しています。これは、rar は /usr/rar/bin/sh を実行して特権付きユーザーになれることを意味します。この出力を参考のために保存したい場合は、ファイルを /tmp ディレクトリの外へ移動してください。

実行可能スタックとセキュリティ

セキュリティのバグの多くは、デフォルトの実行可能スタックのアクセス権が読み取り可能、書き込み可能、および実行可能に設定されたときに発生します。実行権が設定されたスタックは SPARC ABI と Intel ABI によって許可されていますが、ほとんどのプログラムは、実行可能スタックを使用しなくても正常に機能します。

Solaris 2.6 リリースより、noexec_user_stack 変数が利用できるようになりました。この変数によって、システム管理者は、スタックを実行可能としてマッピングするかどうかを指定できます。デフォルトではこの変数はゼロで、ABI 準拠の動作を提供します。この変数がゼロ以外に設定された場合、システムはシステム中のすべてのプロセスのスタックに読み取り可能と書き込み可能のマークをつけますが、実行可能のマークは付けません。

この変数が設定されている場合、プログラムがスタック上でコードを実行しようとすると SIGSEGV シグナルが送信されます。通常、このシグナルが送信されると、プログラムはコアダンプして終了します。このようなプログラムは、違反しているプログラム名、プロセス ID、およびプログラムを実行した実ユーザー ID を含む警告メッセージも生成します。たとえば、次のとおりです。

```
a.out[347] attempt to execute code on stack by uid 555
```

メッセージは、syslog kern 機能が notice レベルに設定されているときに、syslogd(1M) デーモンによってログに記録されます。このログへの記録は、デフォルトで syslog.conf(4) ファイルに設定されていて、メッセージがコンソールと /var/adm/messages ファイルの両方に送信されることを意味します。

このメッセージは、潜在的なセキュリティの問題を調べるときに役立ちます。また、この変数を設定することによって、正しく動作しなくなった、実行可能スタックに依存する有効なプログラムを確認するのにも役立ちます。メッセージを記録しない場合、管理者は、/etc/system ファイルで noexec_user_stack_log 変数をゼロに設定して無効にします。この場合でも実行プログラムは、SIGSEGV シグナルによってコアダンプします。

プログラムのスタックが実行可能であると明示的にマークを付ける場合は、mprotect(2) を使用します。

ハードウェアの制限のため、実行可能スタックの問題を捕捉して報告する機能は、sun4m、sun4d、および sun4u プラットフォームでしか利用できません。

▼ プログラムが実行可能スタックを使用できないようにする方法

1. スーパーユーザーになります。

2. /etc/system ファイルを編集して、次の行を追加します。

```
set noexec_user_stack=1
```

3. システムをリブートします。

```
# init 6
```

▼ 実行可能スタックのメッセージ記録を無効にする方法

1. スーパーユーザーになります。

2. /etc/system ファイルを編集して、次の行を追加します。

```
set noexec_user_stack_log=0
```

3. システムをリブートします。

```
# init 6
```

アクセス制御リスト (ACL)

従来の UNIX ファイル保護機能は、所有者、グループ、その他という 3 つのユーザークラスに読み取り権、書き込み権、実行権を提供します。ACL を使用すると、所有者、所有者のグループ、その他、特定のユーザーおよびグループのファイルアクセス権を定義でき、またこれらのカテゴリごとにデフォルトのアクセス権を定義できるため、ファイルのセキュリティが強化されます。

たとえば、グループ内のすべてのユーザーがファイルを読み取れるようにしたい場合は、単にそのファイルにグループの読み取り権を設定します。その場合に、そのグループ内の 1 人のユーザーだけに書き込み権を与えたいとします。標準の UNIX ではファイルセキュリティをこのように設定することはできませんが、ACL では可能です。

ACL エントリはファイルの ACL を定義する手段であり、`setfacl(1)` コマンドにより設定します。ACL エントリは、次のようにコロンで区切ったフィールドからなっています。

```
entry_type:[uid|gid]:perms
```

entry_type	ファイルのアクセス権を設定する ACL エントリのタイプ。たとえば、*entry_type* は `user` (ファイルの所有者) または `mask` (ACL マスク) に設定できる
uid	ユーザー名または識別番号
gid	グループ名または識別番号
perms	*entry_type* に設定するアクセス権を表す。*perms* は、記号文字 `rwx` または番号 (`chmod` コマンドに使用するのと同じアクセス権番号) で指定できる

次の例に、ユーザー `nathan` の読み取り権および書き取り権を設定する ACL エントリを示します。

```
user:nathan:rw-
```

注意 - ACL などの UFS ファイルシステム属性は UFS ファイルシステムだけでサポートされます。つまり、`/tmp` ディレクトリ (通常は、TMPFS ファイルシステムとしてマウントされている) で ACL エントリを持つファイルを復元またはコピーすると、その ACL エントリは失われます。UFS ファイルの一時的な格納には、`/var/tmp` ディレクトリを使用してください。

ファイルの **ACL** エントリ

表 17–8 に、有効な ACL エントリを示します。最初の 3 つの ACL エントリは、基本的な UNIX のファイル保護機能を提供します。

表 17–8 ファイルの ACL エントリ

ACL エントリ	説明
`u[ser]::`*perms*	ファイル所有者のアクセス権
`g[roup]::`*perms*	ファイルグループのアクセス権
`o[ther]:`*perms*	所有者やファイルグループのメンバー以外のユーザーのアクセス権

表 17–8 ファイルの ACL エントリ 続く

ACL エントリ	説明
`m[ask]:`*perms*	ACL マスク。マスクエントリは、ユーザー (所有者以外) とグループに許される最大アクセス権を示す。マスクは、すべてのユーザーとグループのアクセス権を手早く変更する手段である。 たとえば、`mask:r--` マスクエントリは、ユーザーとグループが書き込みまたは実行権を持っていても、読み取り権しか使用できないことを示す
`u[ser]:`*uid:perms*	特定のユーザーのアクセス権。*uid* には、ユーザー名か UID の数値を指定できる
`g[roup]:`*gid:perms*	特定のグループのアクセス権。*gid* には、グループ名か GID の数値を指定できる

ディレクトリの **ACL** エントリ

表 17–8 に示した ACL エントリの他に、ディレクトリにはデフォルトの ACL エントリも設定できます。デフォルトの ACL エントリを持つディレクトリ内で作成されたファイルまたはディレクトリは、デフォルトの ACL エントリと同じ ACL エントリを持つことになります。表 17–9 に、ディレクトリのデフォルト ACL エントリを示します。

ディレクトリ上で特定のユーザーとグループのデフォルトの ACL エントリを初めて設定するときは、ファイル所有者、ファイルグループ、その他、および ACL マスクにデフォルトの ACL エントリも設定しなければなりません (表 17–9 の最初の 4 つのデフォルト ACL エントリでは、この設定は必須です)。

表 17–9 ディレクトリのデフォルト ACL エントリ

デフォルトの **ACL** エントリ	説明
`d[efault]:u[ser]::`*perms*	ファイル所有者のデフォルトアクセス権
`d[efault]:g[roup]::`*perms*	ファイルグループのデフォルトアクセス権
`d[efault]:o[ther]:`*perms*	ファイル所有者やファイルグループのメンバー以外のユーザーのデフォルトアクセス権

表 17–9　ディレクトリのデフォルト ACL エントリ　続く

デフォルトの **ACL** エントリ	説明
`d[efault]:m[ask]:`*perms*	デフォルトの ACL マスク
`d[efault]:u[ser]:`*uid*`:`*perms*	特定のユーザーのデフォルトアクセス権。*uid* には、ユーザー名か UID の数値を指定できる
`d[efault]:g[roup]:`*gid*`:`*perms*	特定のグループのデフォルトアクセス権。*gid* には、グループ名か GID の数値を指定できる

▼ ファイルの ACL を設定する方法

1. `setfacl` コマンドを使用してファイルの **ACL** エントリを設定します。

```
$ setfacl -s user::perms,group::perms,other:perms,mask:perms,acl_entry_list  filename  ...
```

`-s`	ファイルに対して ACL を設定する。すでに ACL が設定されている場合、新しい ACL に置き換える。このオプションには、少なくともファイル所有者、ファイルグループ、およびその他のエントリを指定する必要がある
`user::`*perms*	ファイル所有者のアクセス権を指定する
`group::`*perms*	ファイルグループのアクセス権を指定する
`other:`*perms*	ファイル所有者またはファイルグループのメンバー以外のユーザーのアクセス権を指定する
`mask:`*perms*	ACL マスクのアクセス権。マスクは、ユーザー (所有者以外) とグループに許される最大アクセス権を示す
acl_entry_list	ファイルまたはディレクトリ上で特定のユーザーとグループに関して設定する 1 つ以上の ACL エントリのリスト。ディレクトリ上でデフォルトの ACL エントリを設定することもできる。有効な ACL エントリについては、表 17–8 と表 17–9 を参照
filename	ACL を設定する 1 つまたは複数のファイルまたはディレクトリを指定する

2. ファイルに **ACL** が設定されたかどうかを確認する方法については、343ページの「ファイルに ACL が設定されているかどうかをチェックする方法」を参照してください。ファイルにどの **ACL** エントリが設定されているかを確認するには、getfacl コマンドを使用します。

```
$ getfacl filename
```

注意 - すでにファイル上に ACL が存在する場合、–s オプションを指定すると、ACL 全体が新しい ACL に置き換えられます。

例 — ファイルの ACL を設定する

次の例は、ch1.doc ファイルで、ファイルの所有者に読み取り／書き込み権、ファイルグループに読み取り権のみ、その他のユーザーにアクセス権「なし」を設定します。また、ユーザー george には、このファイルの読み取り権／書き込み権が与えられ、ACL マスクに読み取り権／書き込み権が設定されます。これは、ユーザーやグループは実行権を持たないことを意味します。

```
$ setfacl -s user::rw-,group::r--,other:---,mask:rw-,user:george:rw- ch1.doc
$ ls -l
total 124
-rw-r-----+  1 nathan  sysadmin   34816 Nov 11 14:16 ch1.doc
-rw-r--r--   1 nathan  sysadmin   20167 Nov 11 14:16 ch2.doc
-rw-r--r--   1 nathan  sysadmin    8192 Nov 11 14:16 notes
$ getfacl ch1.doc
# file: ch1.doc
# owner: nathan
# group: sysadmin
user::rw-
user:george:rw-    #effective:rw-
group::r--         #effective:r--
mask:rw-
other:---
```

次の例は、ch2.doc ファイルで、ファイル所有者に読み取り権／書き込み権／実行権、ファイルグループに読み取り権のみ、その他のユーザーにアクセス権「なし」を設定し、ACL マスクに読み取り権を設定します。さらに、ユーザー george には

読み取り権／書き込み権が与えられます。ただし、ACL マスクの設定により、george の実効アクセス権は読み取りだけです。

```
$ setfacl -s u::7,g::4,o:0,m:4,u:george:7 ch2.doc
$ getfacl ch2.doc
# file: ch2.doc
# owner: nathan
# group: sysadmin
user::rwx
user:george:rwx         #effective:r--
group::r--              #effective:r--
mask:r--
other:---
```

▼ ACL をコピーする方法

getfacl の出力先を変更することにより、ファイルの ACL を他のファイルへコピーします。

```
$ getfacl filename1 | setfacl -f - filename2
```

filename1	ACL のコピー元ファイルを指定する
filename2	ACL のコピー先ファイルを指定する

例 — ACL をコピーする

次の例は、ch2.doc の ACL を ch3.doc へコピーします。

```
$ getfacl ch2.doc | setfacl -f - ch3.doc
```

▼ ファイルに ACL が設定されているかどうかをチェックする方法

ls コマンドを使用して、ファイルに ACL が設定されているかどうかをチェックします。

```
$ ls -l filename
```

filename　　チェックするファイルまたはディレクトリを指定する

モードフィールドの右側の (+) は、ファイルに ACL が設定されていることを示します。

注 - さらにユーザーやグループの ACL エントリをファイルに追加しないかぎり、ファイルの ACL は「弱い」とみなされ、「+」は表示されません。

例 — ファイルに ACL が設定されているかどうかをチェックする

次の例は、モードフィールドの右側に + が付いているため、`ch1.doc` に ACL が設定されています。

```
$ ls -l ch1.doc
-rwxr-----+  1 nathan   sysadmin      167 Nov 11 11:13 ch1.doc
```

▼ ファイルの ACL エントリを変更する方法

1. `setfacl` コマンドを使用してファイルの **ACL** エントリを変更します。

```
$ setfacl -m acl_entry_list filename1 [filename2 ...]
```

`-m`	既存の ACL エントリを変更する
acl_entry_list	ファイルまたはディレクトリで変更する 1 つ以上の ACL エントリのリスト。ディレクトリのデフォルト ACL エントリを変更することもできる。有効な ACL エントリについては、表 17–8 と表 17–9 を参照を指定する
filename ...	1 つまたは複数のファイルまたはディレクトリを指定する

2. ファイルの **ACL** エントリが追加または変更されたことを確認するには、`getfacl` コマンドを使用します。

```
$ getfacl filename
```

例 — ファイルの ACL エントリを変更する

次の例は、ch3.doc ファイルのユーザー george のアクセス権を読み取り権／書き込み権に変更します。

```
$ setfacl -m user:george:6 ch3.doc
$ getfacl ch3.doc
# file: ch3.doc
# owner: nathan
# group: staff
user::rw-
user::george:rw-    #effective:r--
group::r-               #effective:r--
mask:r--
other:r-
```

次の例は、book ディレクトリに関して、グループ staff のデフォルトのアクセス権を読み取りに変更し、デフォルトの ACL マスクを読み取り権／書き込み権に変更します。

```
$ setfacl -m default:group:staff:4,default:mask:6 book
```

▼ ファイルから ACL エントリを削除する方法

1. setfacl コマンドを使用してファイルから **ACL** エントリを削除します。

```
$ setfacl -d acl_entry_list filename1 ...
```

-d	指定した ACL エントリを削除する
acl_entry_list	ファイルまたはディレクトリから (アクセス権を指定せずに) 削除する ACL エントリのリスト。特定のユーザーとグループの ACL エントリとデフォルトの ACL エントリ以外は削除できない。有効な ACL エントリについては、表 17–8 と表 17–9 を参照
filename ...	1 つまたは複数のファイルまたはディレクトリを指定する

setfacl -s コマンドを使用すると、ファイルからすべての ACL エントリを削除して、新たに指定した ACL エントリに置き換えることができます。

2. ファイルから **ACL** エントリが削除されたことを確認するには、getfacl コマンドを使用します。

```
$ getfacl filename
```

例 — ファイルから ACL エントリを削除する

次の例は、ユーザー george を ch4.doc ファイルから削除します。

```
$ setfacl -d user:george ch4.doc
```

▼ ファイルの ACL エントリを表示する方法

getfacl コマンドを使用してファイルの ACL エントリを表示します。

```
$ getfacl [-a | -d] filename1 ...
```

–a	指定したファイルまたはディレクトリのファイル名、ファイル所有者、ファイルグループ、ACL エントリを表示する
–d	指定したディレクトリのファイル名、ファイル所有者、ファイルグループ、デフォルトの ACL エントリを表示する
filename ...	1 つまたは複数のファイルまたはディレクトリを指定する

コマンド行で複数のファイル名を指定すると、各 ACL エントリはブランク行で区切られます。

例 — ファイルの ACL エントリを表示する

次の例は、ch1.doc ファイルのすべての ACL エントリを示します。ユーザーエントリとグループエントリの隣の #effective: は、ACL マスクによって変更された後のアクセス権の設定を示します。

```
$ getfacl ch1.doc
# file: ch1.doc
# owner: nathan
# group: sysadmin
user::rw-
user:george:r--         #effective:r--
group::rw-              #effective:rw-
```

(続く)

```
mask:rw-
other:---
```

次の例は、book ディレクトリのデフォルトの ACL エントリを示します。

```
$ getfacl -d book
# file: book
# owner: nathan
# group: sysadmin
user::rwx
user:george:r-x         #effective:r-x
group::rwx              #effective:rwx
mask:rwx
other:---
default:user::rw-
default:user:george:r--
default:group::rw-
default:mask:rw-
default:other:---
```

第 18 章

システムのセキュリティの手順

この章では、システムにセキュリティを適用する手順について説明します。この章で説明する手順は次のとおりです。

- 350ページの「ユーザーのログイン状態を表示する方法」
- 351ページの「パスワードを持たないユーザーを表示する方法」
- 352ページの「ユーザーのログインを一時的に無効にする方法」
- 353ページの「失敗したログイン操作を保存する方法」
- 356ページの「ダイヤルアップパスワードを作成する方法」
- 358ページの「ダイヤルアップログインを一時的に無効にする方法」
- 359ページの「スーパーユーザー (root) ログインをコンソールに限定する方法」
- 360ページの「su コマンドを使用中のユーザーを監視する方法」
- 361ページの「コンソールへのスーパーユーザー (root) アクセス操作を表示する方法」
- 361ページの「システムのアボートシーケンスを無効または有効にする方法」

システムにセキュリティを適用する概要については、309ページの「システムのセキュリティ」を参照してください。

セキュリティ情報の表示

この節では、ユーザーのログイン情報を表示する方法について説明します。

▼ ユーザーのログイン状態を表示する方法

1. スーパーユーザーになります。

2. `logins` コマンドを使用してユーザーのログイン状態を表示します。

```
# logins -x -l username
```

`-x`	ログイン状態情報の拡張セットを表示する
`-l` *username*	指定するユーザーのログイン状態を表示する。*username* はユーザーのログイン名。複数のログイン名は、コンマで区切って指定する

`logins(1M)` コマンドは、ローカルの `/etc/passwd` ファイルと NIS または NIS+ パスワードデータベースを使用して、ユーザーのログイン状態を表示します。

例 — ユーザーのログイン状態を表示する

次の例には、ユーザー `rimmer` のログイン状態が表示されています。

```
# logins -x -l rimmer
rimmer       500     staff           10   Arnold J. Rimmer
                     /export/home/rimmer
                     /bin/sh
                     PS 010170 10 7 -1
```

`rimmer`	ユーザーのログイン名を示す
`500`	UID (ユーザー ID) を示す
`staff`	ユーザーの一次グループを示す
`10`	GID (グループ ID) を示す
`Arnold J. Rimmer`	コメントを示す

/export/home/rimmer	ユーザーのホームディレクトリを示す
/bin/sh	ログインシェルを示す
PS 010170 10 7 -1	次のパスワードの有効日数情報を示す • パスワードの最終変更日 • 次に変更するまでに必要な日数 • 変更しないで使用できる日数 • 警告期間

▼ パスワードを持たないユーザーを表示する方法

ユーザー全員が有効なパスワードを持っているかどうかを確認する必要があります。

1. スーパーユーザーになります。
2. logins コマンドを使用して、パスワードを持っていないユーザーを表示します。

```
# logins -p
```

-p	パスワードを持っていないユーザーのリストを表示する

logins コマンドは、ローカルの /etc/passwd ファイルと NIS または NIS+ パスワードデータベースを使用して、ユーザーのログイン状態を表示します。

例 — パスワードを持たないユーザーを表示する

次の例には、パスワードを持っていないユーザー pmorph が表示されています。

```
# logins -p
pmorph          501     other           1       Polly Morph
#
```

ユーザーのログインを一時的に無効にする

ユーザーのログインを一時的に無効にするには、次のようにします。

- /etc/nologin ファイルを作成します。
- システムを実行レベル 0 (シングルユーザーモード) にします。システムをシングルユーザーモードに移行する方法については、『*Solaris* のシステム管理 *(*第 *1* 巻*)*』の「システムのシャットダウン (手順)」を参照してください。

/etc/nologin ファイルの作成

このファイルを作成する目的は、システムシャットダウンや定期保守のためにシステムが一定の時間利用できなくなるときに、ユーザーのログインを禁止して、ユーザーに通知することです。

このファイルが存在するシステムにユーザーがログインしようとすると、nologin(4) ファイルの内容が表示されて、ユーザーのログインは中断されます。スーパーユーザーのログインは影響を受けません。

▼ ユーザーのログインを一時的に無効にする方法

1. スーパーユーザーになります。
2. エディタを使用して、/etc/nologin ファイルを作成します。

```
# vi /etc/nologin
```

3. システムの利用に関するメッセージを入力します。
4. ファイルを閉じて、保存します。

例 — ユーザーのログインを無効にする

この例は、システムが利用できないことをユーザーに通知する方法を示しています。

```
# vi /etc/nologin
(ここでシステムメッセージを追加する。)

# cat /etc/nologin
```

(続く)

```
***No logins permitted.***
***The system will be unavailable until 12 noon.***
```

失敗したログイン操作の保存

root 専用の読み取り権と書き込み権を使用して /var/adm/loginlog ファイルを作成すると、失敗したログイン操作を保存できます。loginlog ファイルを作成した後は、操作に 5 回以上失敗すると、失敗したログイン操作がすべてこのファイルに自動的に書き込まれます。詳細は、353ページの「失敗したログイン操作を保存する方法」を参照してください。

loginlog ファイルには、失敗した操作ごとに 1 つずつエントリが入っています。各エントリには、ユーザーのログイン名、tty デバイス、操作の失敗回数が入っています。4 回以下の失敗であれば、ログに記録されません。

loginlog ファイルは急激に大きくなることがあります。このファイル内の情報を使用し、ファイルが大きくなりすぎないようにするには、ファイルの内容をときどきチェックして消去しなければなりません。このファイルが多数の作業を示す場合は、コンピュータシステムに誰かが侵入しようとした可能性があります。このファイルの詳細は、loginlog(4) のマニュアルページを参照してください。

▼ 失敗したログイン操作を保存する方法

1. スーパーユーザーになります。
2. /var/adm ディレクトリ内で loginlog ファイルを作成します。

```
# touch /var/adm/loginlog
```

3. loginlog ファイル上で **root** 用の読み取り権と書き込み権を設定します。

```
# chmod 600 /var/adm/loginlog
```

4. loginlog ファイル上でグループのメンバーシップを sys に変更します。

```
# chgrp sys /var/adm/loginlog
```

5. ログが機能していることを確認するには、loginlog ファイルを作成した後で、間違ったパスワードを使用してシステムに **5** 回ログインします。次に、/var/adm/loginlog ファイルを表示します。

```
# more /var/adm/loginlog
rimmer:/dev/pts/4:Mon Jul 12 13:52:15 1999
rimmer:/dev/pts/4:Mon Jul 12 13:52:23 1999
rimmer:/dev/pts/4:Mon Jul 12 13:52:31 1999
rimmer:/dev/pts/4:Mon Jul 12 13:52:39 1999
#
```

ダイヤルアップパスワードによるパスワード保護

モデムやダイヤルアップポートを通じてシステムにアクセスするユーザーに「ダイヤルアップパスワード」を要求して、パスワード機構にセキュリティ層を追加できます。ダイヤルアップパスワードは、ユーザーがシステムへのアクセス権を取得する前に入力しなければならないパスワードです。

スーパーユーザー以外はダイヤルアップパスワードを作成または変更できません。システムの完全性を確保するために、月に一度はパスワードを変更する必要があります。この機構の最も有効な使用方法は、ゲートウェイシステムへのアクセス権を取得するためのダイヤルアップパスワードを要求することです。

ダイヤルアップパスワードの作成には、/etc/dialups と /etc/d_passwd という 2 つのファイルが必要です。/etc/dialups には、ダイヤルアップパスワードが必要なポートのリストが入っています。/etc/d_passwd には、ダイヤルアップパスワードとして暗号化されたパスワードを要求するシェルプログラムのリストが入っています。

dialups(4) ファイルには、次のような端末装置のリストが含まれます。

```
/dev/term/a
/dev/term/b
```

d_passwd(4) ファイルには 2 つのフィールドがあります。1 つのフィールドはパスワードを要求するログインシェルで、もう 1 つのフィールドは暗号化されたパスワードです。/etc/dialups ファイルと /etc/d_passwd ファイルは次のように使用されます。

ユーザーが /etc/dialups にリストされたポート上でログインしようとすると、ログインプログラムは /etc/passwd に格納されたユーザーのログインエントリを検索し、ログインシェルを /etc/d_passwd 内のエントリと比較します。これらのエントリによって、ユーザーがダイヤルアップパスワードを入力する必要があるかどうかが決まります。

```
/usr/lib/uucp/uucico:encrypted_password:
/usr/bin/csh:encrypted_password:
/usr/bin/ksh:encrypted_password:
/usr/bin/sh:encrypted_password:
```

図 18–1 に、基本的なダイヤルアップパスワードシーケンスを示します。

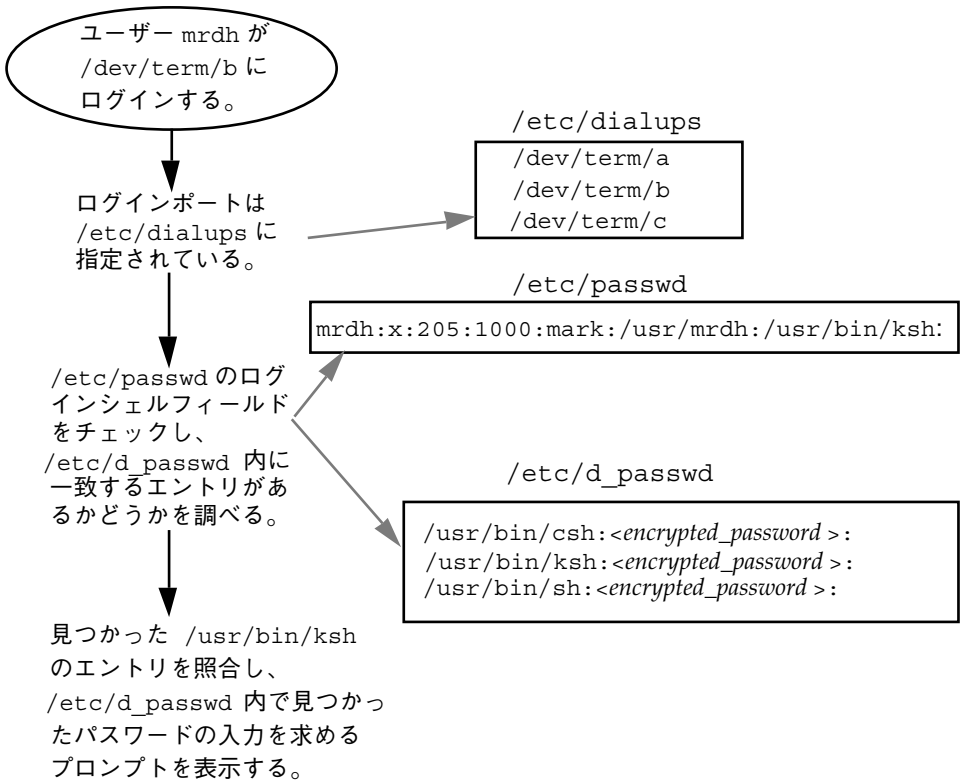


図 18–1　基本的なダイヤルアップパスワードシーケンス

/etc/d_passwd ファイル

ほとんどのユーザーはログインするときにシェルを実行しているので、すべてのシェルプログラムのエントリが /etc/d_passwd 内に必要です。この種のプログラムは、uucico、sh、ksh、csh などです。一部のユーザーがログインシェルから何か実行する場合は、そのログインシェルもファイルに含めてください。

ユーザーのログインプログラム (/etc/passwd 内で指定) が /etc/d_passwd 内で見つからない場合や、/etc/passwd 内のログインシェルフィールドが空 (NULL) の場合は、/usr/bin/sh のパスワードエントリが使用されます。

- /etc/passwd 内のユーザーのログインシェルが /etc/d_passwd 内のエントリと一致する場合、そのユーザーはダイヤルアップパスワードを入力しなければなりません。
- /etc/passwd 内のユーザーのログインシェルが /etc/d_passwd 内で見つからない場合、そのユーザーはデフォルトのパスワードを入力しなければなりません。デフォルトのパスワードは /usr/bin/sh のエントリです。
- /etc/passwd 内のログインシェルフィールドが空の場合、そのユーザーはデフォルトのパスワード (/usr/bin/sh のエントリ) を入力しなければなりません。
- /etc/d_passwd に /usr/bin/sh のエントリがない場合、/etc/passwd 内のログインシェルフィールドが空のユーザー、または /etc/d_passwd 内のエントリと一致しないユーザーには、ダイヤルアップパスワードの入力を促すパスワードは表示されません。
- /etc/d_passwd にエントリ /usr/bin/sh:*: しか入っていない場合、ダイヤルアップログインは使用できません。

▼ ダイヤルアップパスワードを作成する方法

注意 - 最初にダイヤルアップパスワードを設定するときには、少なくとも 1 つの端末にログインしている状態で、別の端末上でパスワードをテストしてください。余分のパスワードをインストールし、ログアウトして新しいパスワードをテストする間にミスすると、元どおりログインできなくなることがあります。まだ別の端末にログインしていれば、元に戻ってミスを訂正できます。

1. スーパーユーザーになります。
2. ダイヤルアップパスワード保護が必要なすべてのポートなど、端末装置のリストが入った /etc/dialups ファイルを作成します。

/etc/dialups ファイルは次のようになります。

```
/dev/term/a

/dev/term/b

/dev/term/c
```

3. ダイヤルアップパスワードを要求するログインプログラムと暗号化されたダイヤルアップパスワードが入った /etc/d_passwd ファイルを作成します。

   uucico、sh、ksh、csh など、ユーザーがログイン時に実行できるシェルプログラムを含めます。/etc/d_passwd ファイルは次のようになります。

```
/usr/lib/uucp/uucico:encrypted_password:

/usr/bin/csh:encrypted_password:

/usr/bin/ksh:encrypted_password:

/usr/bin/sh:encrypted_password:
```

4. **2** つのファイルの所有権を **root** に設定します。

```
# chown root /etc/dialups /etc/d_passwd
```

5. **2** つのファイルのグループの所有権を **root** に設定します。

```
# chgrp root /etc/dialups /etc/d_passwd
```

6. **2** つのファイルの **root** の読み取り権と書き込み権を設定します。

```
# chmod 600 /etc/dialups /etc/d_passwd
```

7. 暗号化パスワードを作成します。

   a. ダミーユーザーを作成します。

```
# useradd user-name
```

b. ダミーユーザーのパスワードを作成します。

```
# passwd user-name
```

c. 暗号化パスワードを取り出します。

```
# grep user-name /etc/shadow > user-name.temp
```

d. ***user-name*.temp** ファイルを編集します。

暗号化パスワード (第 2 のフィールド) を除くすべてのフィールドを削除します。

たとえば、次の行では、暗号化パスワードは U9gp9SyA/JlSk です。

```
temp:U9gp9SyA/JlSk:7967:::::7988:
```

e. ダミーユーザーを削除します。

```
# userdel user-name
```

8. ***user-name*.temp** ファイルから /etc/d_passwd ファイルに暗号化パスワードをコピーします。

ログインシェルごとに別のパスワードを作成するか、共通のパスワードを使用できます。

▼ ダイヤルアップログインを一時的に無効にする方法

1. スーパーユーザーになります。

2. 次のエントリ **1** 行を /etc/d_passwd ファイルに挿入します。

```
/usr/bin/sh:*:
```

コンソールのスーパーユーザー (root) アクセスの制限

スーパーユーザーアカウントは、基本的な機能を実行するためにオペレーティングシステムに使用され、オペレーティングシステム全体を広範囲に制御します。また、スーパーユーザーアカウントは重要なシステムプログラムにアクセスして実行できます。このため、スーパーユーザーによって実行されるプログラムの場合、セキュリティ上の制約はほとんどありません。

/etc/default/login ファイルを通じて特定の装置へのスーパーユーザーアクセスを制限すると、システム上のスーパーユーザーアカウントを保護できます。たとえば、スーパーユーザーアクセスをコンソールに限定しておけば、コンソールからしかスーパーユーザーとしてシステムにログインできなくなります。誰かがシステムにリモートログインして管理作業を実行する場合は、まず自分のユーザーログインを使用してログインしてから、su(1M) コマンドを使用してスーパーユーザーにならなければなりません。詳細は、次の節を参照してください。

注 - システムをインストールするときには、コンソールへのスーパーユーザーログインはデフォルトで制限されます。

▼ スーパーユーザー (root) ログインをコンソールに限定する方法

1. スーパーユーザーになります。
2. /etc/default/login ファイルを編集します。
3. 次の行のコメントを解除します。

```
CONSOLE=/dev/console
```

   このシステムにリモートログインするユーザーは、まず自分のユーザーログインを使用してログインしてから、su コマンドを使用してスーパーユーザーにならなければなりません。

4. このシステムにスーパーユーザーとしてリモートログインして、操作が失敗することを確認してください。

su コマンドを使用するユーザーの監視

su コマンドの試行に対する監視は /etc/default/su ファイルを通じて開始できます。このファイルを通じて、/var/adm/sulog ファイルを使用可能にし、su コマンドを使用して別のユーザーに変更されるたびに監視できます。詳細は、360ページの「su コマンドを使用中のユーザーを監視する方法」を参照してください。

sulog ファイルには、ユーザーをスーパーユーザーに切り替えるコマンドではなく、su コマンドのすべての使用状況がリストされます。各エントリは、コマンドが入力された日時、su コマンドの成否 (+ または -)、コマンドが実行されたポート、およびユーザー名と切り替え後の識別名を示します。

/etc/default/su ファイルを通じて、リモートシステムから su コマンドを使用してスーパーユーザーアクセス権を取得しようとする操作が発生するたびにコンソールに表示されるようにシステムを設定することもできます。これは、現在作業中のシステム上で誰かがスーパーユーザーアクセス権を取得しようとした場合に、それを即座に検出する優れた方法です。詳細は、次の節を参照してください。

▼ su コマンドを使用中のユーザーを監視する方法

1. スーパーユーザーになります。
2. /etc/default/su ファイルを編集します。
3. 次の行のコメントを解除します。

```
SULOG=/var/adm/sulog
```

4. /etc/default/su ファイルを変更し終わったら、su コマンドを何度か使用して /var/adm/sulog ファイルを表示します。su コマンドを使用した時刻ごとにエントリが表示されます。

```
# more /var/adm/sulog
SU 12/20 16:26 + pts/0 nathan-root
SU 12/21 10:59 + pts/0 nathan-root
SU 01/12 11:11 + pts/0 root-joebob
SU 01/12 14:56 + pts/0 pmorph-root
SU 01/12 14:57 + pts/0 pmorph-root
```

▼ コンソールへのスーパーユーザー (root) アクセス操作を表示する方法

1. スーパーユーザーになります。
2. /etc/default/su ファイルを編集します。
3. 次の行のコメントを解除します。

```
CONSOLE=/dev/console
```

su コマンドを使用してスーパーユーザーになり、システムコンソールにメッセージが出力されるかどうかを確認してください。

システムのアボートシーケンスの変更

システムのアボートシーケンスを無効または有効にするには、次の手順を使用します。デフォルトのシステム動作では、システムのアボートシーケンスは有効になっています。

サーバーシステムの中には、キースイッチがあり、それが安全な位置に設定されていると、ソフトウェアキーボードのアボート設定変更を無効にするものがあります。そのため、次の手順で変更を行っても変更が有効にならない場合があります。

▼ システムのアボートシーケンスを無効または有効にする方法

1. スーパーユーザーになります。
2. 次のどちらかの手順で、システムのアボートシーケンスを無効または有効にします。
   a. /etc/default/kbd ファイルの次の行からポンド記号 **(#)** を削除して、システムのアボートシーケンスを無効にします。

```
#KEYBOARD_ABORT=disable
```

   b. /etc/default/kbd ファイルの次の行にポンド記号 **(#)** を追加して、システムのアボートシーケンスを有効にします。

```
KEYBOARD_ABORT=disable
```

3. キーボードのデフォルトを更新します。

```
# kbd -i
```

第 19 章

役割によるアクセス制御

この章では、Solaris 8 リリースの新しいセキュリティ機能である、役割によるアクセス制御について説明します。

役割によるアクセス制御の概要

役割によるアクセス制御 (RBAC) は、スーパーユーザーに依存する従来のシステムの「すべてを許可するか、すべてを許可しない」というセキュリティ方式に代わるものです。これまでの方式では、スーパーユーザーの権限が強すぎるだけでなく、他のユーザーの権限が弱すぎるためユーザーが自分の問題を解決できないという問題がありました。RBAC を使用すると、スーパーユーザー権限をパッケージ化してユーザーアカウントに割り当てることができます。

RBAC を使用すると、適切な権限をパッケージ化してユーザーに割り当てることによって、ユーザーが自分の問題を解決できるようになります。スーパーユーザーの権限をいくつかのパッケージに分割し、それぞれを管理責任を分担する人たちに割り当てることによって、スーパーユーザーの権限は縮小します。

このように RBAC を使用すると、権限を分離したり、他のユーザーへの特権操作の委譲を管理したり、アクセス制御度合いを変更したりできます。

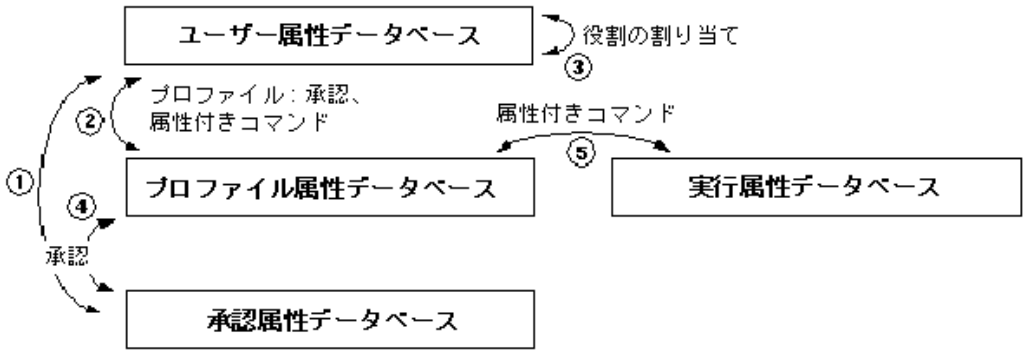
RBAC には次の機能があります。

- 承認 - 制限された機能へのアクセス権を付与する権利
- 実行プロファイル (または単にプロファイル) - 承認やコマンドを特別な属性を使ってグループ化するための統合機構。たとえば、ユーザー ID やグループ ID
- 役割 (Role) - 一連の管理作業を行うことを目的にした特殊なユーザーアカウント

RBAC では、4 つのデータベースを使用して特権操作のアクセス権をユーザーに与えます。

- `user_attr` (拡張ユーザー属性データベース) - ユーザーおよび役割を承認や実行プロファイルと関連付けます。
- `auth_attr` (承認属性データベース) - 承認とその属性を定義し、関連するヘルプファイルを識別します。
- `prof_attr` (実行プロファイル属性データベース) - プロファイルを定義し、プロファイルに割り当てられている承認を列挙し、関連するヘルプファイルを識別します。
- `exec_attr` (プロファイル実行属性データベース) - プロファイルに割り当てられている特権操作を定義します。

次の図は RBAC の動作を示したものです。データベースは四角枠で示され、矢印はデータベース間の関係を意味します。この関係に割り当てられるエンティティは矢印の横に示されます。



`user_attr` データベースでは、承認 (1) またはプロファイル (2) をユーザーに割り当てることができます。これは特権操作をユーザーに直接割り当てる方法です。こ

の方法とは別にユーザーを役割 (3) に割り当てて、その役割に関連するすべての特権操作へのアクセス権をユーザーに与えることもできます。プロファイルは prof_attr データベースに定義され、auth_attr に定義されている承認 (4) と、そのプロファイル用に exec_attr に定義されている属性付きコマンド (5) を含むことができます。

プロファイルに割り当てられたコマンドは、「プロファイルシェル」と呼ぶ特別なシェルで実行されます。プロファイルシェルには pfsh、pfcsh、pfksh があり、それぞれ Bourne シェル (sh)、C シェル (csh)、Korn シェル (ksh) に対応しています。

拡張ユーザー属性データベース (user_attr)

/etc/user_attr データベースは passwd データベースと shadow データベースを補足します。このデータベースには、承認プロファイルや実行プロファイルなど、拡張ユーザー属性が含まれます。このデータベースではユーザーに役割を割り当てることもできます。

role は、一連の管理作業を行うための特殊なユーザーアカウントです。ほとんどの点で通常のユーザーアカウントと同じですが、このアカウントには su コマンドからしかアクセスできません。たとえば CDE のログインウィンドウから、通常のログインを行う場合にこのアカウントにアクセスすることはできません。role アカウントでは、通常のアカウントでは使用できない特殊な属性 (一般には root ユーザー ID) を使用してコマンドにアクセスできます。

user_attr データベースの各フィールドは次のようにコロンで区切ります。

```
user:qualifier:res1:res2:attr
```

フィールドの意味は次のとおりです。

ファイル名	説明
user	passwd(4) データベースに指定されているユーザー名
qualifier	将来、使用される
res1	将来、使用される

ファイル名	説明
res2	将来、使用される
attr	セミコロン (;) で区切られた、キーと値のペアからなるリスト (省略可能)。これは、ユーザーがコマンドを実行したときに使用されるセキュリティ属性を表す。有効なキーは auths、profiles、roles、type ■ auths には、auth_attr.4 データベースに定義されている名前から選択した承認名をコンマで区切って指定する。承認名には、ワイルドカードとしてアスタリスク (*) を使用できる。たとえば、solaris.device.* はすべての Solaris デバイスの承認を意味する ■ profiles には、prof_attr(4) から選択したプロファイル名をコンマで区切り特定の順序で指定する。ユーザーがどのコマンドをどのコマンド属性で実行できるかはプロファイルで決まる。user_attr の各ユーザーには少なくとも All プロファイルが指定されていなければならない。このプロファイルでは、ユーザーはすべてのコマンドを属性なしで実行できる。プロファイルの順序は重要である。この順序は UNIX の検索パスと同じように働く。実行するコマンドにどの属性 (属性を使用する場合) が適用されるかは、そのコマンドが含まれている、リストの最初のプロファイルによって決まる ■ roles には、ユーザーに割り当てる役割名をコンマで区切って指定する。役割も同じ user_attr データベースに定義されることに注意する。役割の場合は、type 値に role が設定される。役割を他の役割に割り当てることはできない ■ type には、アカウントが通常ユーザーの場合は normal、役割の場合は role を設定する。役割は、通常ユーザーがログインしたあとそのユーザーに与えられる

一般的な値を使用した user_attr データベースの例を次に示します。

ユーザー属性データベース

```
root:::type=normal;auths=solaris.*,solaris.grant;profiles=All
sysadmin::::type=role;profiles=..., Device Management,Filesystem
    Management,All
johndoe::::type=normal;auths=solaris.system.date;roles=sysadmin;
    profiles=All
```

一般的な役割の割り当てを次の user_attr データベースを使用して説明します。この例では、sysadmin の役割がユーザー johndoe に割り当てられます。johndoe

は、sysadmin の役割を与えられると、Device Management、Filesystem Management などのプロファイルや All プロファイルにアクセスできます。

承認

承認とは、制限された機能へのアクセス権を付与する、ユーザーの権利です。承認は、何が承認されていて、誰が承認を作成したかを示す固有の文字列です。

制限された機能をユーザーが実行できるかどうかは、一定の特権プログラムが承認を検査して判定します。たとえば、あるユーザーが別のユーザーの crontab ファイルを編集するには、solaris.jobs.admin 承認が必要です。

承認はすべて auth_attr データベースに格納されます。承認は、ユーザー (または役割) に直接割り当てることもできます。その場合は、承認を user_attr データベースに指定します。実行プロファイルに承認を割り当て、実行プロファイルをユーザーに割り当てることもできます。

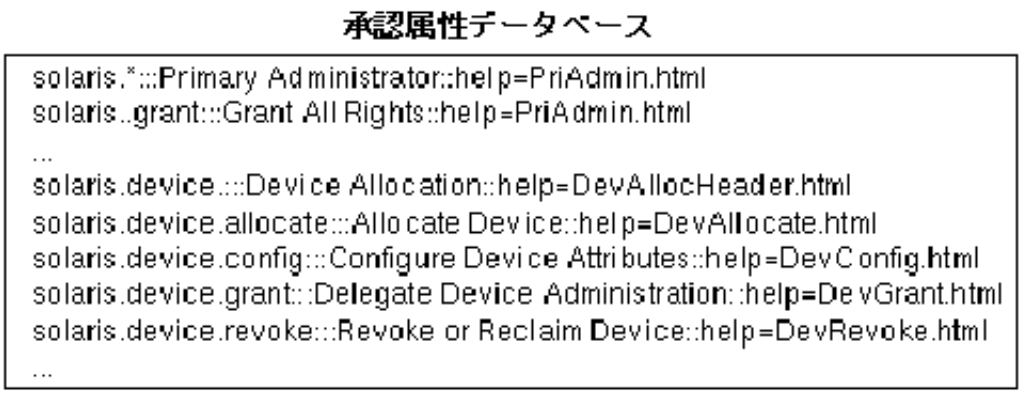
auth_attr データベースのフィールドは次のようにコロンで区切ります。

```
authname:res1:res2:short_desc:long_desc:attr
```

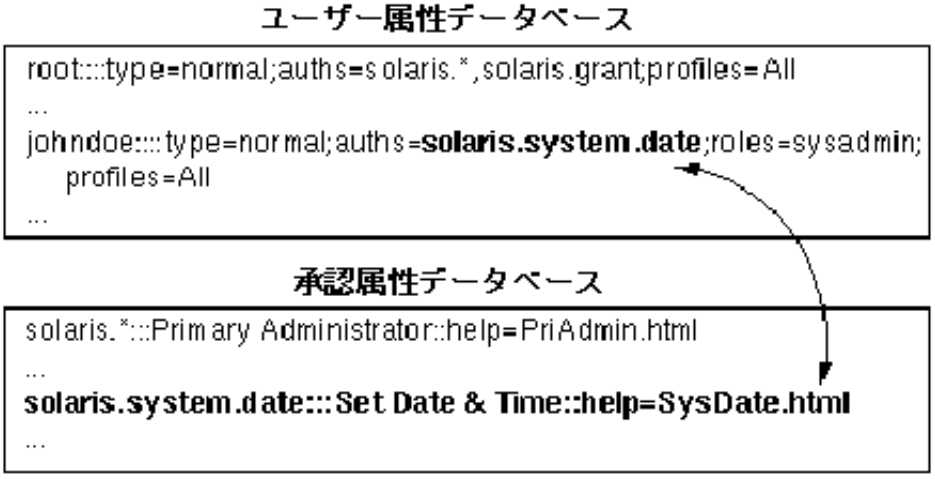
フィールドの意味は次のとおりです。

フィールド名	説明
authname	承認を識別する固有の文字列。形式は *prefix*.[*suffix*] 。Solaris オペレーティング環境では、承認の接頭辞として Solaris を使用する。他のすべての承認には、承認を作成する組織のインターネットドメインを逆にしたもので始まる接頭辞を使用する (たとえば、com.xyzcompany)。接尾辞は、一般には機能分野と機能操作など、承認されるものを示す 接尾辞がない (つまり、authname が接頭辞と機能分野からなり、ピリオドで終わっている) 場合には、authname は、承認としてよりも、アプリケーションによって GUI の中でヘッダーとして使用される。たとえば、authname solaris.printmgr はヘッダーの例である authname が grant という単語で終わっている場合には、authname は grant 承認として使用され、ユーザーは関連する承認 (同じ接頭辞と機能分野をもつ承認) を他のユーザーに委譲できる。たとえば、authname solaris.printmgr.grant は grant 承認の例である。ユーザーは、solaris.printmgr.admin、solaris.printmgr.nobanner などの承認を他のユーザーに委譲できる
res1	将来、使用される
res2	将来、使用される
short_desc	GUI のスクロールリストの中など、ユーザーインタフェースに表示するのに適している承認の簡略名
long_desc	詳しい記述。このフィールドには、承認の目的、承認が使用されるアプリケーション、この使用に関心があるユーザーのタイプなどを記述する。詳しい記述は、アプリケーションのヘルプテキストに表示できる
attr	承認の属性を記述するキーと値のペアをセミコロン (;) で区切ったリスト (省略可能)。ゼロまたは 1 つ以上のキーを指定できる キーワードは、HTML 形式のヘルプファイルを識別するのに役立つ。ヘルプファイルは、/usr/lib/help/auths/locale/C ディレクトリの index.html ファイルからアクセスできる

一般的な値を使用した auth_attr データベースの例を次に示します。



auth_attr データベースと user_attr データベースの関係を次の例で示します。auth_attr データベースに定義されている solaris.system.date 承認が user_attr データベースのユーザー johndoe に割り当てられます。



実行プロファイル

実行プロファイルは、承認やコマンドを特別な属性を使用してグループ化し、それをユーザーや役割に割り当てるための統合機構です。特別な属性には、実 UID、実 GID、実効 UID、実効 GID が含まれます。最も一般的な属性では、実効 UID に root を設定します。実効プロファイルの定義は prof_attr データベースに格納されます。

prof_attr データベースのフィールドは次のようにコロンで区切ります。

```
profname:res1:res2:desc:attr
```

フィールドの意味は次のとおりです。

フィールド名	説明
profname	プロファイル名。プロファイル名では大文字と小文字が区別される
res1	将来、使用される
res2	将来、使用される
desc	詳しい記述。このフィールドでは、どのようなユーザーがこの使用に関心があるかなど、プロファイルの目的を説明する。詳しい記述は、アプリケーションのヘルプテキストとして適しているものでなければならない
attr	実行時にそのオブジェクトに適用するセキュリティ属性を記述するキーと値のペアをセミコロン (;) で区切ったリスト (省略可能)。ゼロまたは 1 つ以上のキーを指定できる。有効なキーは help と auths キーワード help には HTML 形式のヘルプファイルを指定する。ヘルプファイルは、/usr/lib/help/auths/locale/C ディレクトリの index.html ファイルからアクセスできる auths には、auth_attr.4 データベースに定義されている名前から選択した承認名をコンマで区切って指定する。承認名には、ワイルドカードとしてアスタリスク (*) を使用できる

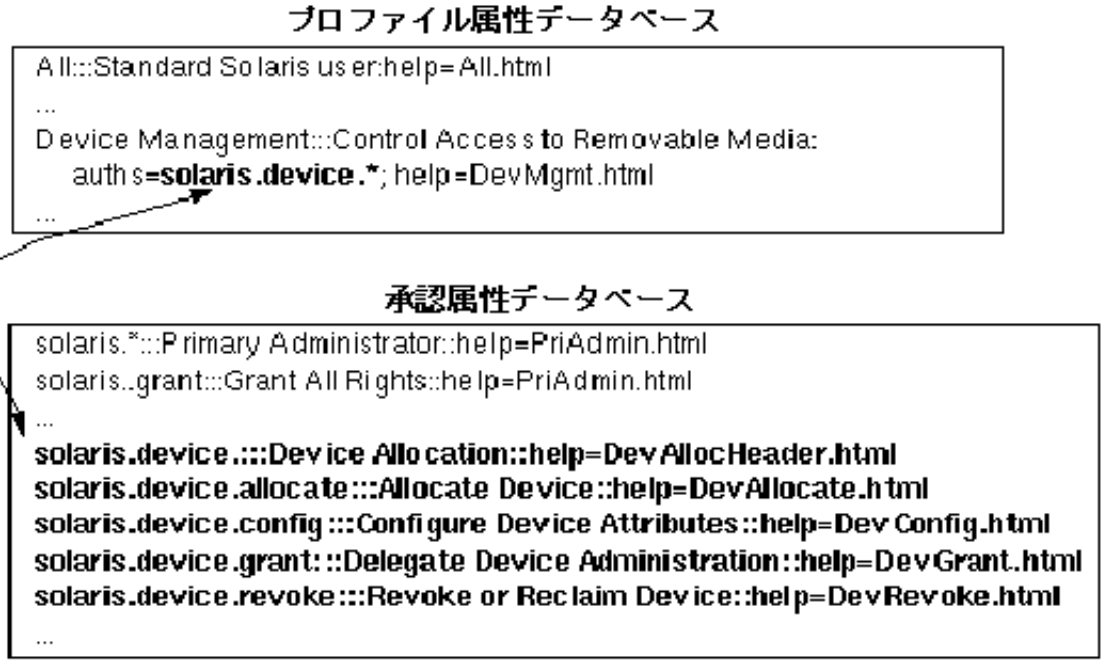
一般的な値を使用した prof_attr データベースの例を次に示します。

プロファイル属性データベース

```
All:::Standard Solaris user:help=All.html
...
Printer Management:::Manage print jobs: help=Printmgt.html
Device Management:::Control Access to Removable Media:
    auths=solaris.device.*; help=DevMgmt.html
...
```

prof_attr データベースと user_attr データベースの関係を次の例で示します。prof_attr データベースに定義されている Device Management プロファイルが user_attr データベースの役割 sysadmin に割り当てられます。

prof_attr データベースと auth_attr データベースの関係を次の例で示します。prof_attr データベースには、solaris.device 文字列で始まるすべての承認が Device Management プロファイルに割り当てられています。これらの承認は auth_attr データベースに定義されています。



実行属性

プロファイルに関連付けられた実行属性は、そのプロファイルが割り当てられているユーザーや役割によって実行できる (特別なセキュリティ属性をもつ) コマンドです。特別なセキュリティ属性とは、コマンドを実行するときにプロセスに追加できる UID、EUID、GID、EGID などの属性のことです。

実行属性の定義は `exec_attr` データベースに格納されます。

`exec_attr` データベースのフィールドは次のようにコロンで区切って指定します。

```
name:policy:type:res1:res2:id:attr
```

フィールドの意味は次のとおりです。

フィールド名	意味
`name`	プロファイル名。プロファイル名では大文字と小文字が区別される
`policy`	このエントリに関連付けるセキュリティポリシー。現在は、`suser` (スーパーユーザーポリシーモデル) が唯一の有効なポリシーである

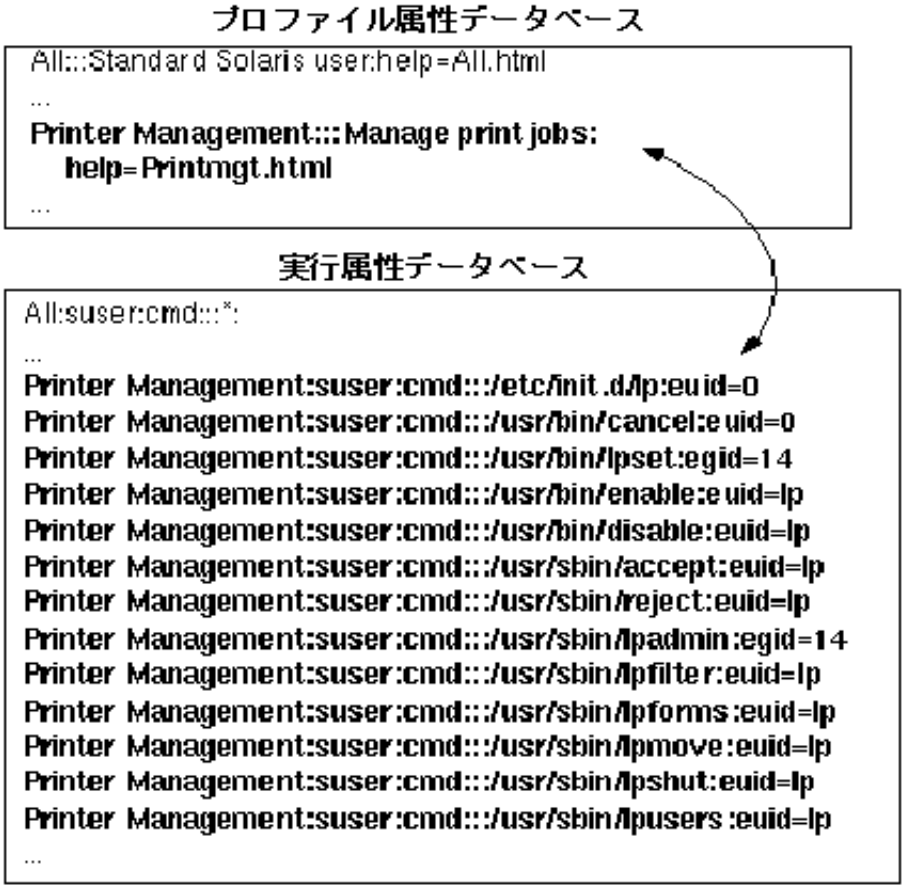
フィールド名	意味
type	属性が指定されるエンティティのタイプ。現在は、cmd (コマンド) が唯一の有効なタイプである
res1	将来、使用される
res2	将来、使用される
id	このエンティティを表す文字列。ワイルドカードとしてアスタリスクを使用できる。コマンドには、完全パスかワイルドカードをもつパスを指定する。引数を指定する場合は、引数をもつスクリプトを作成し、そのスクリプトを id に指定する
attr	実行時にそのエンティティに適用するセキュリティ属性を記述するキーと値のペアをセミコロン (;) で区切ったリスト (省略可能)。ゼロまたは 1 つ以上のキーを指定できる。キーワードのリストは、適用するポリシーによって異なる。有効なキーは euid、uid、egid、および gid である euid と uid には、単一のユーザー名か数値ユーザー ID を指定する。euid を使用すると、コマンドは指定された実効 UID で動作する。これは、実行可能ファイルの setuid ビットを有効にするのと同じことである。uid を使用すると、コマンドは指定された実 UID と実効 UID で動作する egid と gid には、単一のグループ名か数値グループ ID を指定する。egid を使用すると、コマンドは指定された実効 GID で動作する。これは、実行可能ファイルの setgid ビットを有効にするのと同じことである。gid を使用すると、コマンドは指定された実 GID と実効 GID で動作する

一般的な値を使用した exec_attr データベースの例を次に示します。

実行属性データベース

```
All:suser:cmd:::*:
...
Printer Management:suser:cmd:::/usr/lib/lp/lpsched:euid=0
Printer Management:suser:cmd:::/usr/lib/lp/lpshut:euid=0
Printer Management:suser:cmd:::/usr/lib/lp/lpmove:euid=0
Printer Management:suser:cmd:::/bin/lp:euid=0
Printer Management:suser:cmd:::/bin/lpadmin:euid=0
Printer Management:suser:cmd:::/usr/sbin/lpadmin:euid=0
Printer Management:suser:cmd:::/usr/bin/enable:euid=0
Printer Management:suser:cmd:::/usr/bin/disable:euid=0
Printer Management:suser:cmd:::/usr/sbin/accept:euid=0
Printer Management:suser:cmd:::/usr/sbin/reject:euid=0
Printer Management:suser:cmd:::/usr/sbin/lpsystem:euid=0
...
```

`exec_attr` と `prof_attr` データベースの関係を次の例で示します。Printer Management プロファイルは `prof_attr` データベースに定義されています。このプロファイルには 13 の実行属性があり、適切なセキュリティ属性が `exec_attr` データベースで割り当てられています。



▼ 役割によるアクセス制御を設定する方法

役割を設定するには su コマンドを使用します。役割にログインすることはできません。

```
% su <自分の役割>
Password: <自分の役割のパスワード>
#
```

プロファイル内のコマンドを使用するには、コマンドをシェルに入力します。

```
# lpadmin -p myprinter options
```

lpadmin コマンドは、その役割のプロファイルで lpadmin コマンドに割り当てられているプロセス属性 (特別な UID や GID) を使用して実行されます。

役割によるアクセス制御を管理するツール

データベースを直接編集する他に、役割によるアクセス制御を使用して管理するためのツールには次のものがあります。

コマンド	説明
auths(1)	ユーザーに対する承認を表示する
makedbm(1M)	dbm ファイルを作成する
nscd(1M)	ネームサービスキャッシュデーモン。user_attr、prof_attr、および exec_attr データベースをキャッシュするときに使用する
pam_role_auth(5)	PAM 用の役割アカウント管理モジュール。役割を行う承認があるかを調べる
pfexec(1)	プロファイルシェル。exec_attr データベースに指定されている属性を使用してコマンドを実行するときにプロファイルシェルで使用される
policy.conf(4)	セキュリティポリシーの構成ファイル。付与されている承認をリストする
profiles(1)	指定されたユーザーのプロファイルを表示する
roles(1)	ユーザーに付与されている役割を表示する
roleadd(1M)	役割のアカウントをシステムに追加する
roledel(1M)	役割のアカウントをシステムから削除する
rolemod(1M)	システムにある役割のアカウント情報を変更する
useradd(1M)	ユーザーアカウントをシステムに追加する。ユーザーのアカウントに役割を割り当てるには、-P オプションを使用する

コマンド	説明
userdel(1M)	ユーザーのログインをシステムから削除する
usermod(1M)	システムにあるユーザーのアカウント情報を変更する

第 20 章

認証サービスの使用手順

この章の最初の節では、Secure RPC で使用できる Diffie-Hellman 認証機構について説明します。 2 番目の節では、Pluggable Authentication Module (PAM) フレームワークについて説明します。PAM は、認証サービスを「プラグイン」する方法を提供して、複数の認証サービスを使用できるようにします。

この章で説明する手順は次のとおりです。

Secure RPC の概要

Secure RPC は、ホストと、要求を依頼したユーザーの両方を認証する認証方法です。Secure RPC は、Diffie-Hellman 認証を使用します。この認証機構は DES 暗号化を使用します。Secure RPC を使用するアプリケーションには、NFS と NIS+ ネームサービスがあります。

NFS サービスと Secure RPC

NFS ソフトウェアを使用すると、複数のホストがネットワーク上でファイルを共有できます。NFS システムでは、サーバーは、複数のクライアントから利用できるデータと資源を格納します。クライアントは、サーバーがクライアントと共有するファイルシステムにアクセスできます。クライアントマシンにログインしたユーザーは、ファイルシステムをサーバーからマウントすることによって、そのファイルシステムにアクセスできます。このとき、クライアントマシン上のユーザーには、ファイルはクライアントのローカルファイルシステム上にあるように見えます。NFS 環境の最も一般的な使用形態として、システムを各オフィスにインストールして、すべてのユーザーファイルを 1 箇所で集中管理する方法が挙げられます。`mount –nosuid` オプションなどのいくつかの NFS 機能を使用すると、権限のないユーザーがデバイスやファイルシステムにアクセスすることを禁止できます。

NFS 環境では Secure RPC を使用して、要求を出したユーザーをネットワーク上で認証します。これを Secure NFS と呼びます。認証機構 `AUTH_DH` は、Diffie-Hellman 認証で DES 暗号化を使用し、許可されたアクセスを確認します。`AUTH_DH` 機構は、`AUTH_DES` とも呼びます。

『*Solaris* のシステム管理 *(*第 *3* 巻*)*』では、Secure NFS を設定および管理する方法を説明しています。NIS+ テーブルの設定と `cred` テーブルへの名前の入力は、『*Solaris* ネーミングの管理』で説明しています。RPC 認証に含まれる手順の概要については、381ページの「Diffie-Hellman 認証の実装」を参照してください。

DES 暗号化

Data Encryption Standard (DES) 暗号化機能は 56 ビットの鍵を使用して、データ鍵を暗号化します。資格を持つ 2 人のユーザー (プリンシパル) が同じ DES 鍵を知っている場合、彼らはその鍵を使用してテキストを暗号化または復号化することによって、プライベートに通信できます。DES は比較的高速な暗号化機構です。DES チップは暗号化をより高速にします。しかし、チップがなくても、ソフトウェアが実行します。

DES 鍵だけを使用する危険性とは、同じ鍵を使用して暗号化した暗号テキストメッセージを十分に収集すれば、鍵を発見し、メッセージを復号化できることです。この理由のため、Secure NFS などのセキュリティシステムは鍵を頻繁に変更します。

Kerberos 認証

Kerberos は、マサチューセッツ工科大学 (MIT) で開発された認証システムです。Kerberos は DES 暗号を使用します。Kerberos Version 4 は Secure RPC ではサポートされませんが、RPCSEC_GSS を使用する Kerberos Version 5 のクライアント側実装は、Solaris 8 リリースに含まれます。詳細は、第 21 章を参照してください。

Diffie-Hellman 認証

Diffie-Hellman のユーザー認証方法は簡単には破られません。クライアントとサーバーは、それぞれ独自の非公開鍵 (秘密鍵とも呼ぶ) を持っていて、共通鍵が利用できるように公開鍵と組み合わせて使用します。クライアントとサーバーはお互いにこの共通鍵を使用し、両者で合意された暗号化復号化機能 (DES など) を使用して通信します。この方法は、以前の Solaris リリースの DES 認証と同じです。

認証は、送信側のシステムの共通鍵を使用して現在の時刻を暗号化する機能を利用します。受信側のシステムは、その現在の時刻を復号し、自分の時刻と照合します。クライアントとサーバーで時刻が同期していることを確認してください。

公開鍵と非公開鍵は、NIS または NIS+ のデータベースに格納されます。NIS では、`publickey` マップに鍵を格納します。NIS+ では、`cred` テーブルに鍵を格納します。これらのファイルには、すべてのユーザーの公開鍵と非公開鍵が入っています。

システム管理者は、NIS または NIS+ のテーブルを設定して、ユーザーごとに公開鍵と非公開鍵を生成する義務があります。公開鍵は、ユーザーのパスワードで暗号化されて格納されます。これによって、その公開鍵はそのユーザーだけが知っていることになります。

Diffie-Hellman 認証の実装

この節では、DH 認証 (`AUTH_DH`) を使用するクライアントサーバーセッションにおける一連のトランザクションを説明します。

公開鍵と秘密鍵の生成

トランザクションの前に、管理者は `newkey` コマンドか `nisaddcred` コマンドを使用して、公開鍵と秘密鍵を生成します (各ユーザーは一意の公開鍵と秘密鍵を持ちます)。公開鍵は公開データベースに格納されます。秘密鍵は、暗号化された形式で、

同じデータベースに格納されます。鍵のペアを変更するには、chkey コマンドを使用します。

keylogin コマンドの実行

通常、ログインパスワードは Secure RPC パスワードと同じです。この場合、keylogin は必要ありません。パスワードが異なる場合、ユーザーはログインしてから明示的に keylogin を実行しなければなりません。

keylogin プログラムは、Secure RPC パスワードを求めるプロンプトをユーザーに出して、そのパスワードを使用して秘密鍵を復号します。keylogin プログラムは、復号した秘密鍵をキーサーバーというプログラムに渡します (キーサーバーは、すべてのコンピュータ上にローカルインスタンスを持つ RPC サービスです)。キーサーバーは、復号された秘密鍵を保存して、ユーザーがサーバーで Secure RPC トランザクションを発行するのを待ちます。

パスワードが同じ場合は、ログインプロセスが秘密鍵をキーサーバーに渡します。パスワードが異なる必要があり、ユーザーが常に keylogin を実行しなければならない場合は、keylogin プログラムをユーザーの環境の構成ファイル (~/.login、~/.cshrc、~/.profile など) に入れておいて、ユーザーがログインするたびに自動的に実行されるようにします。

対話鍵の生成

ユーザーがサーバーとのトランザクションを開始すると、次の動作が行われます。

1. キーサーバーはランダムに対話鍵を生成します。
2. カーネルはこの対話鍵を使用して、クライアントのタイムスタンプを暗号化します (他の動作も行います)。
3. キーサーバーは公開鍵データベースでサーバーの公開鍵を検索します (publickey(4) のマニュアルページを参照してください)。
4. キーサーバーはクライアントの秘密鍵とサーバーの公開鍵を使用して、共通鍵を作成します。
5. キーサーバーは共通鍵を使用して対話鍵を暗号化します。

サーバーとの最初の接触

次に、暗号化したタイムスタンプと暗号化した対話鍵を含む伝送データがサーバーに送信されます。伝送データには資格とベリファイアが含まれます。資格は、次の 3 つの構成要素を持ちます。

- クライアントのネット名
- 共通鍵で暗号化された対話鍵
- 対話鍵で暗号化された「ウィンドウ」

この場合の「ウィンドウ」とは、クライアントが主張する、サーバーの時刻とクライアントのタイムスタンプとの許容されるべき差のことです。サーバーの時刻とクライアントのタイムスタンプとの間の差がウィンドウより大きい場合、サーバーはクライアントの要求を拒否します。クライアントは RPC セッションを開始する前にサーバーと同期を取るため、通常の状態では、このような事態は発生しません。

クライアントのベリファイアは、次の構成要素を持ちます。

- 暗号化されたタイムスタンプ
- 指定したウィンドウの暗号化されたベリファイアから 1 を引いた値

ウィンドウベリファイアが必要な理由は次の場合です。誰かが別のユーザーになりすまそうとして、資格とベリファイアの暗号化されたフィールドに書き込む代わりに、ランダムなビットだけを埋め込むプログラムを書いたと仮定します。サーバーはこの対話鍵をなんらかのランダム鍵に復号化し、それを使用してウィンドウとタイムスタンプを復号化しようと試みます。その結果、乱数が生成されるだけです。しかし、数千回の試行を重ねるうちには、このランダムなウィンドウタイムスタンプのペアが認証システムを通過することが十分ありえます。ウィンドウベリファイアは、正しい資格の解読をより困難にします。

対話鍵の復号化

サーバーがクライアントから伝送データを受信すると、次の動作が行われます。

1. サーバーのローカルなキーサーバーが、公開鍵データベースでクライアントの公開鍵を検索します。
2. キーサーバーは、クライアントの公開鍵とサーバーの秘密鍵を使用して、共通鍵を計算します。この共通鍵はクライアントが計算したものと同じです。共通鍵を計算するためには、どちらか一方の秘密鍵を知っている必要があるため、これを行えるのはサーバーとクライアントだけです。
3. カーネルは共通鍵を使用して、対話鍵を復号します。
4. カーネルはキーサーバーを呼び出して、復号された対話鍵によりクライアントのタイムスタンプを復号します。

サーバーへの情報の格納

サーバーは、クライアントのタイムスタンプを復号した後、次の 4 種類の情報を資格テーブルに格納します。

- クライアントのコンピュータ名
- 対話鍵
- ウィンドウ
- クライアントのタイムスタンプ

サーバーは、最初の 3 つの情報を将来の使用のために格納します。サーバーはタイムスタンプを格納して、同じタイムスタンプが再度使用できないようにします。サーバーは、最後に参照したタイムスタンプよりも時間的に後のタイムスタンプだけを受け付けるため、同じタイムスタンプのトランザクションはすべて拒否されることが保証されます。

注 - この手順において暗黙的に仮定されているのは呼び出し側の名前であり、何らかの方法でこの名前を認証しなければなりません。キーサーバーは、この目的には DES 認証を使用できません。DES 認証を使用すれば、デッドロックが発生するからです。キーサーバーは、 UID ごとに秘密鍵を格納し、ローカルの root プロセスへの要求だけを許可することによってこの問題を解決します。

クライアントに返されるベリファイア

サーバーは、ベリファイアをクライアントに返します。ベリファイアは、次の構成要素を持ちます。

- サーバーが自分の資格キャッシュに記録するインデックス ID
- 対話鍵によって暗号化された、クライアントのタイムスタンプから 1 を引いたもの

タイムスタンプから 1 を引く理由は、これを無効化して、クライアントのベリファイアとして再利用できないようにするためです。

クライアントによるサーバーの認証

クライアントがベリファイアを受信し、そのサーバーを認証します。クライアントは、このベリファイアを送信できるのはサーバーだけであることを知っています。

その理由は、クライアントが送信したタイムスタンプの内容を知っているのはサーバーだけだからです。

追加のトランザクション

一番目以降のすべてのトランザクションごとに、クライアントは 2 番目のトランザクションでインデックス ID をサーバーに返し、もう 1 つの暗号化されたタイムスタンプを送信します。サーバーは、クライアントのタイムスタンプから 1 を引いたものを対話鍵で暗号化して、返送します。

Diffie-Hellman 認証の管理

システム管理者は、ネットワークを安全にするためのポリシーをネットワーク上に実装できます。必要なセキュリティのレベルはサイトによって異なります。この節では、ネットワークセキュリティに関連するいくつかの作業手順を説明します。

▼ キーサーバーを再起動する方法

1. スーパーユーザーになります。

2. keyserv デーモン (キーサーバー) が動作していることを確認します。

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

3. キーサーバーが動作していない場合は、キーサーバーを起動します。

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

▼ Diffie-Hellman 認証で NIS+ 資格を設定する方法

NIS+ セキュリティの詳細は、『*Solaris* ネーミングの管理』を参照してください。

NIS+ クライアント上で *root* 用の新しい鍵を設定するには

1. スーパーユーザーになります。

2. /etc/nsswitch.conf ファイルを編集して、次の行を追加します。

```
publickey: nisplus
```

3. **NIS+** クライアントを起動します。

```
# nisinit -cH hostname
```

   hostname は、そのテーブルにクライアントマシン用のエントリを持つ、信頼されている NIS+ サーバー名です。

4. 次のコマンドを入力して、クライアントを cred テーブルに追加します。

```
# nisaddcred local
# nisaddcred des
```

5. keylogin コマンドを使用して、設定を確認します。
   パスワードを求めるプロンプトが出たら、この手順は成功です。

例 — NIS+ クライアント上で root 用の新しい鍵を設定する

次の例は、ホスト pluto を使用して、earth を NIS+ クライアントとして設定しています。警告は無視できます。keylogin コマンドが受け付けられて、earth が Secure NIS+ クライアントとして正しく設定されていることを確認しています。

```
# nisinit -cH pluto
NIS Server/Client setup utility.
This machine is in the North.Abc.COM. directory.
Setting up NIS+ client ...
All done.
# nisaddcred local
# nisaddcred des
DES principal name : unix.earth@North.Abc.COM
Adding new key for unix.earth@North.Abc.Com (earth.North.Abc.COM.)
```

(続く)

```
Network password: xxx <Press Return>
Warning, password differs from login password.
Retype password: xxx <Press Return>

# keylogin
Password:
#
```

NIS+ ユーザー用の新しい鍵を設定するには

1. 次のコマンドを入力して、ユーザーを **root** マスターサーバー上の `cred` テーブルに追加します。

   ```
   # nisaddcred -p unix.UID@domainname -P username.domainname. des
   ```

   この場合、「*username-domainname*」はドット (.) で終了しなければなりません。

2. クライアントとしてログインし、`keylogin` コマンドを入力して、設定を確認します。

例 — NIS+ ユーザー用の新しい鍵を設定する

次の例は、DES セキュリティ認証をユーザー `george` に与えています。

```
# nisaddcred -p unix.1234@North.Abc.com -P george.North.Abc.COM. des
DES principal name : unix.1234@North.Abc.COM
Adding new key for unix.1234@North.Abc.COM (george.North.Abc.COM.)

Password:
Retype password:

# rlogin rootmaster -l george
# keylogin
Password:
#
```

▼ Diffie-Hellman 認証で NIS 資格を設定する方法

クライアント上でスーパーユーザー用の新しい鍵を作成するには

1. クライアント上でスーパーユーザーになります。
2. /etc/nsswitch.conf ファイルを編集して、次の行を追加します。

```
publickey: nis
```

3. newkey コマンドを使用して、新しい鍵ペアを作成します。

```
# newkey -h hostname
```

hostname は、クライアント名です。

例 — Diffie-Hellman セキュリティを使用するように NIS+ クライアントを設定する

次の例は、earth を Secure NIS クライアントとして設定しています。

```
# newkey -h earth
Adding new key for unix.earth@North.Abc.COM
New Password:
Retype password:
Please wait for the database to get updated...
Your new key has been successfully stored away.
#
```

ユーザー用の新しい鍵を作成するには

1. サーバーにスーパーユーザーとしてログインします。

   ユーザー用の新しい鍵を作成できるのは、NIS+ サーバーにログインしたシステム管理者だけです。

2. ユーザー用の新しい鍵を作成します。

```
# newkey -u username
```

username はユーザー名です。システムはパスワードを求めるプロンプトを出します。システム管理者は汎用パスワードも入力できます。非公開鍵は汎用パスワードで暗号化されて格納されます。

```
# newkey -u george
Adding new key for unix.12345@Abc.North.Acme.COM
New Password:
Retype password:
Please wait for the database to get updated...
Your new key has been successfully stored away.
#
```

3. ログインして chkey -p コマンドを入力するように、ユーザーに伝えます。

   これによって、そのユーザーは自分だけが知っているパスワードを使用して、自分の非公開鍵を再び暗号化できます。

```
earth% chkey -p
Updating nis publickey database.
Reencrypting key for unix.12345@Abc.North.Acme.COM
Please enter the Secure-RPC password for george:
Please enter the login password for george:
Sending key change request to pluto...
#
```

注 - chkey コマンドを使用すると、新しい鍵ペアをユーザーに作成できます。

▼ Diffie-Hellman 認証でファイルを共有およびマウントする方法

前提条件

Diffie-Hellman の publickey 認証がネットワークで有効にされていなければなりません。385ページの「Diffie-Hellman 認証で NIS+ 資格を設定する方法」と 388 ページの「Diffie-Hellman 認証で NIS 資格を設定する方法」を参照してください。

Diffie-Hellman 認証でファイルシステムを共有するには

1. スーパーユーザーになります。
2. **Diffie-Hellman** 認証でファイルシステムを共有します。

```
# share -F nfs -o sec=dh /filesystem
```

Diffie-Hellman 認証でファイルシステムをマウントするには

1. スーパーユーザーになります。
2. **Diffie-Hellman** 認証でファイルシステムをマウントします。

```
# mount -F nfs -o sec=dh server:resource    mountpoint
```

`-o sec=dh` オプションは、`AUTH_DH` 認証でファイルシステムをマウントします。

PAM について

Pluggable Authentication Module (PAM) フレームワークを使用すると、`login`、`ftp`、`telnet` などのシステムに入るためサービスを変更しなくても、新しい認証技術を「プラグイン」できるようになります。また、PAM を使用すれば、 UNIX ログインを DCE や Kerberos のような他のセキュリティ機構と統合できます。また、アカウント、セッション、およびパスワードの管理機構もプラグインできます。

PAM を使用する利点

PAM フレームワークを使用すると、システム管理者は任意のシステムに入るため、サービス (`ftp`、`login`、`telnet`、`rsh` など) とユーザー認証用を組み合わせることができます。次に PAM の利点をいくつか挙げます。

- 柔軟な構成ポリシー

- アプリケーションごとの認証ポリシー
- デフォルトの認証機構を選択する機能
- 高度なセキュリティシステムにおける複数のパスワード

- エンドユーザーにも使いやすい

  - 機構が異なってもパスワードが同じであれば、パスワードを再入力する必要がない
  - 各認証方法に関連するパスワードが異なっている場合でも、パスワードマッピング機能により、複数の認証方法で 1 つのパスワードを使用する機能
  - ユーザーが複数のコマンドを入力しなくても、複数の認証方法のパスワードを求めるプロンプトを出す機能

- オプションパラメタをユーザー認証サービスに渡す機能

PAM の概要

PAM は、実行時に取り外しが可能なモジュールを使用して、システムに入るためのサービスに認証を提供します。これらのモジュールは、その機能に基づき、4 つの異なるタイプに分かれます。認証、アカウント管理、セッション管理、およびパスワード管理です。スタッキング機能によって、複数のサービス経由でユーザーを認証できます。また、パスワードマッピング機能によって、ユーザーは複数のパスワードを覚えておく必要がありません。

PAM モジュールのタイプ

モジュールタイプはモジュールのインターフェースを定義するため、PAM モジュールのタイプを理解することは重要です。実行時 PAM モジュールには、次の 4 つのタイプがあります。

- 「認証モジュール」は、ユーザーの認証を提供して、資格を設定、更新、または削除できます。認証モジュールは、ユーザーの識別に役立つ管理ツールを提供します。
- 「アカウントモジュール」は、パスワードの有効期限、アカウントの有効期限、およびアクセス時間制限をチェックします。アカウントモジュールは、認証

モジュールでユーザーを識別した後に、そのユーザーにアクセス権を与えるべきかどうかを決定します。

- 「セッションモジュール」は、認証セッションの開閉を管理します。セッションモジュールは、動作を記録したり、セッション終了後のクリーンアップを実行したりできます。
- 「パスワードモジュール」によって、実際のパスワードを変更できます。

スタッキング機能

PAM フレームワークは、「スタッキング機能」を使用して、複数のサービスでユーザーを認証する方法を提供します。構成によって、認証方法ごとにパスワードを求めるプロンプトをユーザーに出すことも可能です。認証サービスが使用される順序は、PAM 構成ファイルで決定されます。

パスワードマッピング機能

スタッキング機能を使用する方法では、ユーザーが複数のパスワードを覚えておかなければなりません。「パスワードマッピング機能」を使用すれば、主要パスワードから他のパスワードを復号できるので、ユーザーは複数のパスワードを覚えたり入力したりする必要はありません。各認証機構間でパスワードの同期を取るためのオプションもあります。スタック内で使用される最も安全性の低いパスワードによって各機構のセキュリティが制限されてしまうので、この方法はセキュリティの危険性を増大してしまうことに注意してください。

PAM の機能

PAM ソフトウェアは、ライブラリ、いくつかのモジュール、および構成ファイルからなります。いくつかのシステムに入るためのコマンドまたはデーモンの新しいバージョンは、PAM インタフェースを利用できます。

図 20–1 は、アプリケーション、PAM ライブラリ、`pam.conf` ファイル、および PAM モジュール間の関係を示しています。

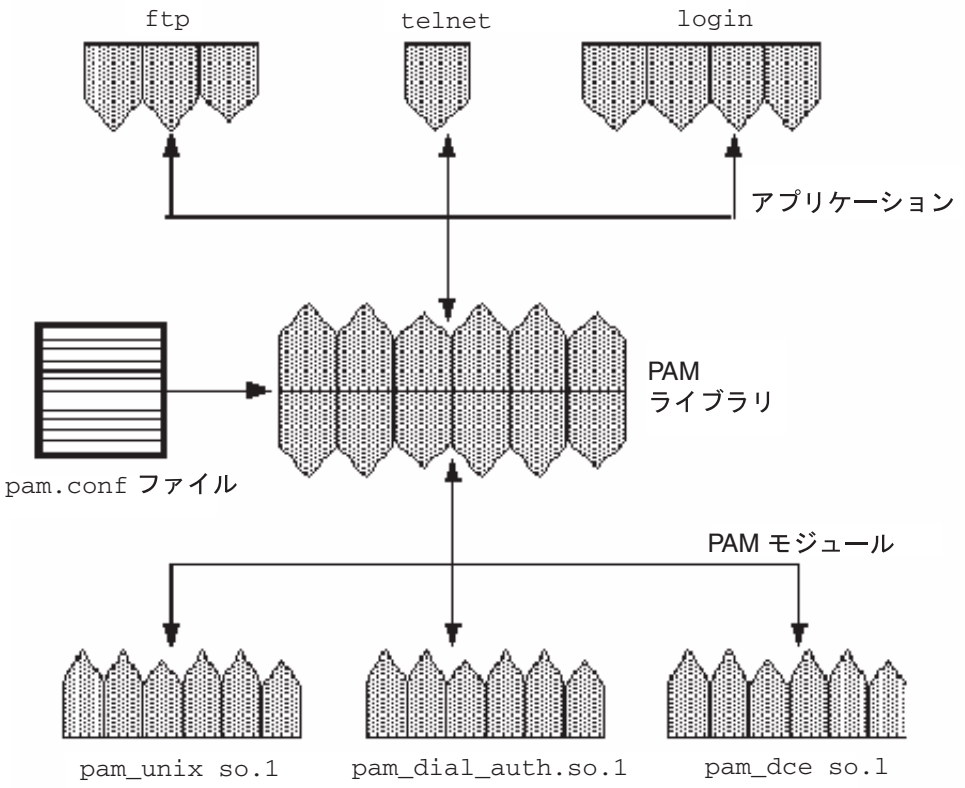


図 20–1　PAM の動作

アプリケーション (ftp、telnet、および login) は、PAM ライブラリを使用して、適切なモジュールにアクセスします。pam.conf ファイルは、使用するモジュールを定義して、各アプリケーションがモジュールを使用する順番を定義します。モジュールからの応答は、ライブラリ経由でアプリケーションに戻されます。

次の節では、この関係について説明します。

PAM ライブラリ

PAM ライブラリ /usr/lib/libpam は、適切なモジュールをロードして、スタッキング手順を管理するためのフレームワークを提供します。また、すべてのモジュールを取り外すことができる汎用構造も提供します。

PAM モジュール

各 PAM モジュールは、特定の機構を実装します。PAM 認証を設定するときは、モジュールとモジュールタイプの両方を指定する必要があります。モジュールタイプは、モジュールが実行する処理を定義します。複数のモジュールタイプ (auth、account、session、または password) を各モジュールに関連付けることができます。

次のリストで各 PAM モジュールについて説明します。

- pam_unix モジュール /usr/lib/security/pam_unix.so.1 は、認証、アカウント管理、セッション管理、およびパスワード管理に使用できます。このモジュールでは、4 種類すべてのモジュールタイプの定義が使用できます 。このモジュールは、UNIX パスワードを認証に使用します。Solaris 環境では、パスワードを取得するための適切なネームサービスの選択は、/etc/nsswitch.conf ファイルで制御されます。詳細は、pam_unix(5) のマニュアルページを参照してください。
- dial_auth モジュール /usr/lib/security/pam_dial_auth.so.1 は、認証だけに使用できます。このモジュールは、/etc/dialups ファイルと /etc/d_passwd ファイルに格納されたデータを認証するのに使用します。このモジュールは主に login で使用されます。詳細については、pam_dial_auth(5) のマニュアルページを参照してください。
- rhosts_auth モジュール /usr/lib/security/pam_rhosts_auth.so.1 も、認証だけに使用できます。このモジュールは、~/.rhosts ファイルと /etc/hosts.equiv ファイルに格納されたデータを ruserok 経由で使用します。このモジュールは、主に rlogin コマンドと rsh コマンドで使用されます。詳細については、pam_rhosts_auth(5) のマニュアルページを参照してください。

  セキュリティ上の理由から、これらのモジュールファイルの所有者は root でなければならず、また、その書き込み権を group と other に与えてはなりません。ファイルの所有者が root でない場合、PAM はモジュールをロードしません。
- krb5 モジュール /usr/lib/security/pam_krb5_auth.so.1 は、認証、アカウント管理、セッション管理、およびパスワード管理のサポートを提供します。認証には Kerberos 資格が使用されます。

セキュリティ上の理由から、これらのモジュールファイルの所有者は root でなければならず、また、その書き込み権を group と other に与えてはなりません。ファイルの所有者が root でない場合、PAM はモジュールをロードしません。

PAM 構成ファイル

PAM 構成ファイル /etc/pam.conf は、使用する認証サービスとそれらを使用する順序を決定します。このファイルを編集すれば、システムに入るためのアプリケーションごとに認証機構を選択できます。

構成ファイルの構文

PAM 構成ファイルは、次の構文のエントリからなります。

```
service_name  module_type  control_flag  module_path  module_options
```

service_name	サービス名 (たとえば、ftp、login、telnet など)
module_type	サービスのモジュールタイプ
control_flag	モジュールの継続または失敗の意味を決定する
module_path	サービス機能を実装するライブラリオブジェクトのパス
module_options	サービスモジュールに渡される特定のオプション

pam.conf ファイルにコメントを追加するには、その行を # (ポンド記号) で始めます。フィールドを区切るには、空白を使用します。

注 - 次の 3 つの状態のいずれかが存在する場合、PAM 構成ファイル内のエントリは無視されます。(1) 行のフィールド数が 4 つより少ない。(2) *module_type* または *control_flag* に無効な値が指定されている。(3) 指定したモジュールが見つからない。

有効なサービス名

表 20–1 は、有効なサービス名、そのサービスで使用できるモジュールタイプ、およびサービス名に関連するデーモンまたはコマンドを示しています。

サービスごとに適切でないモジュールタイプもいくつかあります。たとえば、password モジュールタイプは、passwd コマンドだけに指定できます。このコマン

ドは認証には関連しないので、このコマンドに関連する auth モジュールタイプはありません。

表 20–1 /etc/pam.conf での有効なサービス名

サービス名	デーモンまたはコマンド	モジュールタイプ
dtlogin	/usr/dt/bin/dtlogin	auth、account、session
ftp	/usr/sbin/in.ftpd	auth、account、session
init	/usr/sbin/init	session
login	/usr/bin/login	auth、account、session
passwd	/usr/bin/passwd	password
rexd	/usr/sbin/rpc.rexd	auth
rlogin	/usr/sbin/in.rlogind	auth、account、session
rsh	/usr/sbin/in.rshd	auth、account、session
sac	/usr/lib/saf/sac	session
su	/usr/bin/su	auth、account、session
telnet	/usr/sbin/in.telnetd	auth、account、session
ttymon	/usr/lib/saf/ttymon	session
uucp	/usr/sbin/in.uucpd	auth、account、session

制御フラグ

認証プロセス中にモジュールの処理を継続するか失敗するかを決定するには、エントリごとに 4 つの制御フラグの 1 つを選択しなければなりません。制御フラグは、各

モジュールで正常終了と異常終了がどのように処理されるかを示します。これらのフラグはすべてのモジュールに適用されますが、次の説明では、これらのフラグは認証モジュールで使用されていると仮定します。制御フラグは、次のとおりです。

- required - 最終的に成功を返すためには、このモジュールが正常終了を返さなければなりません。

  すべてのモジュールに required フラグを付けた場合、ユーザーの認証が成功するには、すべてのモジュールの認証が成功しなければなりません。

  一部のモジュールが失敗した場合、最初に失敗したモジュールのエラー値が報告されます。

  required フラグを付けたモジュールが失敗しても、スタック中のすべてのモジュールは継続して処理されますが、異常終了が返されます。

  どのモジュールにも required フラグを付けなかった場合、ユーザーの認証が成功するには、そのサービスの少なくとも 1 つのエントリが成功しなければなりません。

- requisite - 後続の認証が行われるには、このモジュールが正常終了を返さなければなりません。

  requisite フラグの付いたモジュールが失敗した場合、すぐにエラーがアプリケーションに返され、それ以上認証は行われません。スタック中で、このモジュールより前に required というラベルの付いたモジュールが失敗していなければ、このモジュールからのエラーが返されます。このモジュールより前に、required というラベルの付いたモジュールが失敗している場合は、required モジュールからのエラーメッセージが返されます。

- optional - このモジュールが失敗した場合、このスタック内の他のモジュールが正常終了を戻せば、最終的に成功が返される可能性があります。

  optional フラグは、スタック内の 1 つのモジュールが成功すればユーザーの認証が成功するときに使用します。このフラグは、この認証機構が成功することが重要でない場合だけに使用します。

  ユーザーが作業をするためには、特定の機構に関連する許可を取得する必要がある場合、そのモジュールに optional フラグを付けてはなりません。

- sufficient - このモジュールが成功すると、スタック内の残りのモジュールは required フラグが付いていてもスキップされます。

  sufficient フラグは、1 つの認証が成功すれば、ユーザーにアクセス権を与えてもかまわないことを示します。

これらのフラグの詳細は、デフォルトの /etc/pam.conf ファイルについて説明している 400ページの「PAM の構成」を参照してください。

汎用 pam.conf ファイル

次の例は、汎用 pam.conf ファイルを示しています。

```
# PAM configuration
# Authentication management
#
login auth required /usr/lib/security/pam_unix.so.1
login auth required /usr/lib/security/pam_dial_auth.so.1
rlogin auth sufficient /usr/lib/security/pam_rhost_auth.so.1
rlogin auth required /usr/lib/security/pam_unix.so.1
dtlogin auth required /usr/lib/security/pam_unix.so.1
telnet auth required /usr/lib/security/pam_unix.so.1
su auth required /usr/lib/security/pam_unix.so.1
ftp auth required /usr/lib/security/pam_unix.so.1
uucp auth required /usr/lib/security/pam_unix.so.1
rsh auth required /usr/lib/security/pam_rhost_auth.so.1
OTHER auth required /usr/lib/security/pam_unix.so.1
#
# Account management
#
login account required /usr/lib/security/pam_unix.so.1
rlogin account required /usr/lib/security/pam_unix.so.1
dtlogin account required /usr/lib/security/pam_unix.so.1
telnet account required /usr/lib/security/pam_unix.so.1
ftp account required /usr/lib/security/pam_unix.so.1
OTHER account required /usr/lib/security/pam_unix.so.1
#
# Session management
#
login session required /usr/lib/security/pam_unix.so.1
rlogin session required /usr/lib/security/pam_unix.so.1
dtlogin session required /usr/lib/security/pam_unix.so.1
telnet session required /usr/lib/security/pam_unix.so.1
uucp session required /usr/lib/security/pam_unix.so.1
OTHER session required /usr/lib/security/pam_unix.so.1
#
# Password management
#
passwd password required /usr/lib/security/pam_unix.so.1
OTHER password required /usr/lib/security/pam_unix.so.1
```

この汎用 pam.conf ファイルは、次の内容を指定しています。

1. login を実行するとき、pam_unix モジュールと pam_dial_auth モジュールの両方による認証が成功しなければなりません。
2. rlogin を実行するとき、pam_unix モジュールによる認証が失敗した場合は、pam_rhost_auth モジュールによる認証が成功しなければなりません。

3. sufficient 制御フラグは、rlogin の場合は、pam_rhost_auth モジュールによる認証が成功すれば十分であり、次のエントリが無視されることを示しています。
4. 認証を必要とする他のほとんどのコマンドの場合、pam_unix モジュールによる認証が成功しなければなりません。
5. rsh の場合、pam_rhost_auth モジュールによる認証が成功しなければなりません。

OTHER サービス名を使用すれば、認証を必要とするがこのファイルには含まれていない他のコマンドに対するデフォルトとして設定できます。OTHER オプションを使用すると、同じモジュールを使用する多数のコマンドを 1 つのエントリだけでカバーできるので、ファイルの管理が簡単になります。また、OTHER サービス名を「すべてを捕捉する」と言う意味で使用すると、1 つのモジュールですべてのアクセスをカバーできます。通常、OTHER エントリは、各モジュールタイプのセクションの最後に指定します。

ファイル内の残りのエントリは、アカウント、セッション、およびパスワードの管理を制御します。

デフォルトサービス名 OTHER を使用すると、汎用 PAM 構成ファイルは、次のように簡単になります。

```
# PAM configuration
#
# Authentication management
#
login auth required /usr/lib/security/pam_unix.so.1
login auth required /usr/lib/scurty/pam_dial_auth.so.1
rlogin auth sufficient /usr/lib/security/pam_unix.so.1
rlogin auth required /usr/lib/security/pam_rhost_auth.so.1
rsh auth required /usr/lib/security/pam_rhost_auth.so.1
OTHER auth required /usr/lib/security/pam_unix.so.1
#
# Account management
#
OTHER account required /usr/lib/security/pam_unix.so.1
#
# Session management
#
OTHER session required /usr/lib/security/pam_unix.so.1
#
# Password management
#
OTHER password required /usr/lib/security/pam_unix.so.1
```

通常、*module_path* のエントリには「ルートからのパス名」を指定します。*module_path* に入力したファイル名がスラッシュ (/) で始まらない場合、そのファイル名の前にパス /usr/lib/security/ が付きます。モジュールが他のディレクトリにある場合は、フルパスを使用しなければなりません。

module_options の値については、そのモジュールのマニュアルページ (たとえば、pam_unix(5)) を参照してください。

pam_unix モジュールでサポートされている use_first_pass オプションと try_first_pass オプションを使用すると、ユーザーは認証用の同じパスワードを再入力しなくても再利用できます。

login が pam_local と pam_unix の両方による認証を指定した場合、ユーザーは、モジュールごとにパスワードを入力するようにプロンプトが表示されます。パスワードが同じ場合、use_first_pass モジュールオプションを使用すれば、パスワードの入力を求めるプロンプトは 1 度だけ表示されます。そのパスワードを両方のモジュールで使用して、ユーザーを認証します。パスワードが異なる場合、認証は失敗します。通常、このオプションは、次に示すように、optional 制御フラグといっしょに使用して、依然としてユーザーのログインが可能なようにします。

```
# Authentication management
#
login auth required /usr/lib/security/pam_unix.so.1
login auth optional /usr/lib/security/pam_local.so.1 use_first_pass
```

try_first_pass モジュールオプションを代わりに使用すると、パスワードが一致しなかった場合またはエラーが発生した場合、ローカルモジュールは、2 番目のパスワードを求めるプロンプトを表示します。必要なすべてのツールにアクセスするために、ユーザーが両方の認証方法を必要とする場合、このオプションを使用すると 1 つのタイプの認証だけでアクセスできるので、ユーザーが混乱する場合があります。

PAM の構成

この節では、PAM のフレームワークを完全に機能させるために必要な作業について説明します。特に、PAM 構成ファイルに関連するセキュリティのいくつかの問題について注意する必要があります。

PAM の計画

どのように PAM を使用すればユーザーのサイトに最適であるかを決定するために、次の問題から始めます。

- 何が必要か、特にどのモジュールを選択するかを決定します。
- 特別な注意が必要なサービスを確認します。適宜、OTHER を使用します。
- モジュールを実行する順番を決定します。
- そのモジュールに対する制御フラグを選択します。
- モジュールに必要な任意のオプションを選択します。

ここで、構成ファイルを変更する前に考慮すべき問題を示します。

- すべてのアプリケーションを指定しなくてもいいように、モジュールタイプごとに OTHER エントリを使用します。
- sufficient 制御フラグと optional 制御フラグのセキュリティの意味を考慮します。
- モジュールに関連するマニュアルページを参照して、各モジュールがどのように機能するか、どのオプションが使用できるか、およびスタック中のモジュール間の相互作用を理解します。

注意 - PAM 構成ファイルの構成を間違えたり壊したりすると、スーパーユーザーでもログインできなくなる可能性があります。sulogin は PAM を使用しないので、スーパーユーザーは、マシンをシングルユーザーモードでブートして問題を解決しなければなりません。

/etc/pam.conf ファイルの変更後、スーパーユーザーとしてログインしている間にできるだけ調査します。変更によって影響を受けるコマンドは、すべてテストします。たとえば、新しいモジュールを telnet サービスに追加した場合、telnet コマンドを使用して、行なった変更が期待どおりに動作しているかどうかを確認します。

▼ PAM モジュールを追加する方法

1. **スーパーユーザーになります。**

2. **使用される制御フラグやオプションを決定します。**

   モジュールについては、394ページの「PAM モジュール」を参照してください。

3. 新しいモジュールを /usr/lib/security にコピーします。

4. モジュールファイルの所有者が **root** で、そのアクセス権が 555 になるように、アクセス権を設定します。

5. **PAM** 構成ファイル /etc/pam.conf を編集して、このモジュールを適切なサービスに追加します。

確認

構成ファイルが間違って構成されていた場合などのために、システムをリブートする前にテストすることは非常に重要です。システムをリブートする前に、rlogin、su、および telnet を実行します。サービスが、システムがブートするときだけに生成されるデーモンの場合は、システムをリブートしなければ、モジュールが正しく追加されていることを確認できません。

▼ PAM を使用して、リモートシステムからの承認されていないアクセスを防ぐ方法

PAM 構成ファイルから「rlogin auth rhosts_auth.so.1」エントリを削除します。これによって、rlogin セッション中、~/.rhosts ファイルは読み込まれなくなります。したがって、リモートシステムからローカルシステムへの認証されていないアクセスを防ぐことができます。~/.rhosts ファイルまたは /etc/hosts.equiv ファイルの存在またはその内容にかかわらず、すべての rlogin アクセスにはパスワードが必要になります。

注 - ~/.rhosts ファイルへの承認されていない他のアクセスを防ぐには、rsh サービスも無効にする必要があります。サービスを無効にする最良の方法は、/etc/inetd.conf からサービスエントリを削除することです。PAM 構成ファイルを変更しても、サービスを無効にはできません。

▼ PAM のエラー報告を有効にする方法

1. /etc/syslog.conf を編集して、次の **PAM** のエラー報告に関するエントリを追加します。
   - auth.alert — 即座に修正しなければならない状態についてのメッセージ
   - auth.crit — 致命的なメッセージ
   - auth.err — エラーメッセージ

- auth.info — 情報通知用メッセージ
- auth.debug — デバッグ用メッセージ

2. syslog デーモンを再起動するか、SIGHUP シグナルをこのデーモンに送信して、**PAM** のエラー報告を有効にします。

例 — PAM のエラー報告を有効にする

次の例では、警戒メッセージはすべてコンソールに表示されます。致命的なメッセージは root に電子メールで送信されます。情報メッセージとデバッグ用メッセージは、/var/log/pamlog ファイルに追加されます。

```
auth.alert /dev/console
auth.crit 'root'
auth.info;auth.debug /var/log/pamlog
```

ログ内の各行は、タイムスタンプ、メッセージを生成したシステム名とメッセージ自身からなります。pamlog ファイルには、大量の情報が記録される可能性があります。

第 21 章

SEAM の概要

この章では、Solaris 8 の SEAM 製品の概要を説明します。SEAM 1.0 には、Kerberos V5 ネットワーク認証プロトコルが実装されています。これは Sun Easy Access Server (SEAS) 3.0 リリースに含まれています。Solaris 8 リリースには、SEAM 製品のすべては含まれていません。このリリースには、クライアント側の製品だけが含まれています。この章では、SEAM 製品のクライアント側の製品とサーバー側の製品を両方とも説明しているため、製品全体の相互関係がわかります。この章の内容は次のとおりです。

SEAM とは

Sun Eneterprise Authentication Mechanism (SEAM) は、強力なユーザー認証およびデータの完全性とプライバシを提供することにより、ネットワークトランザクションのセキュリティを保護するクライアントサーバーアーキテクチャです。認証により、ネットワークトランザクションの送信者と受信者の識別情報が正しいことが保証されます。さらに SEAM を使用して、送受信するデータの完全性が検査され (「完全性」)、伝送時にデータが暗号化されます (「プライバシ」)。SEAM を使用して、他のマシンにログインしてコマンドを実行したり、データを交換したりファイ

ルを安全に転送したりできます。SEAM は認証サービスも提供するため、管理者はサービスやマシンへのアクセスを制限でき、ユーザーは自分のアカウントに他人がアクセスするのを制限できます。

SEAM は「シングルサインオン」システムです。つまり、SEAM からセッションについて一度だけ認証を受ければ、そのセッションでは、それ以後のすべてのトランザクションが自動的に認証されます。いったん認証されたら、パスワードを再び入力する必要はありません。つまり、これらのサービスを使用するたびに、ネットワークを介してパスワードを送り、傍受される危険を冒す必要はありません。

SEAM は、マサチューセッツ工科大学 (MIT) で開発された Kerberos V5 ネットワーク認証プロトコルに基づいています。そのため、Kerberos V5 を使用したことがあれば、SEAM にはすぐ慣れるはずです。Kerberos V5 はネットワークセキュリティの業界標準 (RFC 1510 を参照) で、SEAM では他の製品との相互運用性が増します。つまり、SEAM は Kerberos V5 を使用するシステムと協調して動作するため、異機種構成のネットワークであってもトランザクションのセキュリティが保護されます。さらに SEAM では、ドメイン間でも単一のドメイン内でも認証やセキュリティの機能を使用できます。

注 - SEAM は Kerberos V5 に基づいて動作し、Kerberos V5 と相互運用が可能なように設計されているため、このマニュアルでは「Kerberos」と「SEAM」を同じ意味で使用することがあります (「Kerberos レルム」や「SEAM ベースユーティリティ」など)。(Kerberos と Kerberos V5 も同じ意味で使用されています。) 必要な場合はこれらを区別します。

SEAM には、Solaris アプリケーションを実行するための柔軟性が備わっています。NFS サービスなどのネットワークサービスを、SEAM ベースでも非 SEAM ベースでも使用できるように SEAM を設定できます。そのため、SEAM がインストールされていないシステムで動作する現在の Solaris アプリケーションも正しく動作します。もちろん、SEAM ベースのネットワーク要求だけを許可するように SEAM を設定することもできます。

さらに、他のセキュリティ機構が開発された場合には、アプリケーションで使用するセキュリティ機構を SEAM に限定しておく必要はありません。SEAM は、Generic Security Service API の下のモジュール構造の下で使用するように設計されているため、GSS-API を使用するアプリケーションは、必要に応じたセキュリティ機構を使用できます。

SEAM 技術

この節では、この後の SEAM の説明で使用される用語とその定義について説明します。SEAM の説明を理解するには、これらの用語を理解しておく必要があります。

Kerberos 固有の用語

KDC 管理の節を理解するには、次の用語を理解してください。

「Key Distribution Center」または「KDC」は、資格の発行を担当する SEAM の一部分です。資格は、KDC データベースに格納されている情報に基づいて作成されます。各レルムには少なくとも 2 つの KDC が必要です (マスターと少なくとも 1 つのスレーブ)。資格はすべての KDC が生成できますが、KDC データベースを変更できるのはマスターだけです。

KDC のマスター鍵を暗号化したコピーは「stash ファイル」に入っています。サーバーがリブートされると、この鍵を使用して KDC が自動的に認証されてから `kadmind` と `krb5kdc` が起動されます。このファイルにはマスター鍵が入っているため、このファイルやそのバックアップは安全な場所に保管する必要があります。暗号が破られると、この鍵を使用して KDC データベースのアクセスや変更が可能になります。

認証固有の用語

認証プロセスを理解するには、次の用語の理解が必要です。プログラマやシステム管理者はこれらの用語に精通していなければなりません。

「クライアント」とは、ユーザーのワークステーションで動作しているソフトウェアです。クライアント上で動作する SEAM ソフトウェアは、このプロセスの間に多くの要求を行います。したがって、このソフトウェアの動作とユーザーの操作を区別することが大切です。

サーバーとサービスはしばしば同じ意味で使われます。正確にいえば、「サーバー」は SEAM ソフトウェアが動作している物理システムであり、「サービス」はサーバー上でサポートされている特定の機能 (`ftp` や `nfs` など) です。サーバーがサービスの一部として使用されることがよくありますが、これはこれらの用語の意味をあいまいにします。したがって、サーバーは物理システムを、サービスはソフトウェアをそれぞれ指すことにします。

SEAM 製品には 3 種類の鍵があります。1 つは「非公開鍵」です。この鍵は各ユーザー (プリンシパル) に与えられ、プリンシパルのユーザーと KDC だけに知られています。ユーザープリンシパルに対しては、鍵はユーザーのパスワードに基づいています。サーバーとサービスに対する鍵は「サービス鍵」と呼ばれます。この鍵は非公開鍵と同じ目的で使われますが、これはサーバーとサービスによって使用されます。3 つめの鍵は「セッション鍵」です。この鍵は、認証サービスまたはチケット許可サービスによって生成されます。セッション鍵は、クライアントとサービス間のトランザクションのセキュリティを保護するために生成されます。

「チケット」は、ユーザーの識別情報をサーバーやサービスに安全に渡すために使用される情報パケットです。チケットは、単一のクライアントと特定のサーバーの特定のサービスだけに有効です。チケットには、サービスのプリンシパル名、ユーザーのプリンシパル名、ユーザーのホストの IP アドレス、タイムスタンプ、チケットの有効期限を定義する値などが入っています。チケットは、クライアントとサービスによって使用されるランダムセッション鍵を使用して作成されます。チケットは、作成されてから有効期限が過ぎるまで再使用できます。

「資格」は、対応するセッション鍵とチケットをもつ情報パケットです。一般に資格は、資格を暗号化するものに応じて、非公開鍵かサービス鍵を使用して暗号化されます。

「認証」はさらに別のタイプの情報です。これをチケットとともに使用すれば、ユーザープリンシパルを認証できます。認証には、ユーザーのプリンシパル名、ユーザーのホストの IP アドレス、タイムスタンプが含まれています。チケットとは異なり、認証は一度しか使用できません。認証を使用するのは、通常、サービスへのアクセスが要求されたときです。認証は、そのクライアントとそのサーバーのセッション鍵を使用して暗号化されます。

SEAM の構成要素

SEAS 3.0 の SEAM 1.0 の完全リリースには、次のものを含む多数の構成要素が含まれています。

- Key Distribution Center (KDC)
- データベース管理プログラム
- チケットの取得、表示、または破棄を行うユーザープログラム
- Kerberos 対応アプリケーション — `telnet`

- 管理ユーティリティ
- Pluggable Authentication Module (PAM) の拡張

SEAM 1.0 リリースに含まれる全構成要素は、『*Sun Enterprise Authentication Mechanism* ガイド』の「SEAM の概要」に記載されています。

Solaris 8 リリースに含まれているのは SEAM のクライアント側部分だけで、これらの構成要素の多くは含まれていません。そのため、Solaris 8 リリースが動作するシステムであれば、SEAM を別にインストールしなくても SEAM クライアントとして動作します。この機能を使用するには、SEAS 3.0、MIT 配布、Windows2000 のどれかを使用して KDC をインストールする必要があります。クライアント側の構成要素は、チケットを配布する構成済み KDC がないと使用できません。このリリースには、次の構成要素が含まれています。

- チケットの取得、表示、破棄を行うユーザープログラム (`kinit`、`klist`、`kdestroy`) と SEAM パスワードを変更するユーザープログラム (`kpasswd`)
- 鍵テーブル管理ユーティリティ (`ktutil`)
- Pluggable Authentication Module (PAM) の拡張。アプリケーションはいろいろな認証機構を使用できるようになる。PAM を使用すると、ログインとログアウトをユーザーが意識する必要がなくなる
- GSS_API プラグイン。Kerberos プロトコルおよび暗号サポートを提供する
- NFS クライアントおよびサーバーサポート

SEAM の動作

この節の説明は SEAM 認証システムの一般的な概要です。詳細は、446ページの「認証システムの動作」を参照してください。

ユーザーの観点からいえば、SEAM は、SEAM セッションが起動された後はほとんど目につきません。SEAM セッションの初期化には通常、ログインと Kerberos パスワードの入力しか必要ありません。

SEAM システムは「チケット」の概念を中心に動作します。チケットは、ユーザーや NFS などのサービスの識別情報となる一連の電子情報です。運転免許証が運転する人と免許の種類を表すのと同じように、チケットもユーザーとユーザーのネットワークアクセス権を表します。SEAM ベースのトランザクションを行うと (たとえば、新しいプリンシパルに対し `kinit` を行う場合)、チケットの要求が Key

Distribution Center (KDC) に透過的に送信されます。KDC はデータベースにアクセスしてこの識別情報を認証して、その他のマシンへのアクセスを許可するチケットをユーザーに戻します。「透過的」とは、チケットを明示的に要求する必要がないという意味です。この要求は kinit コマンドの中で行われます。特定のサービスのチケットを取得できるのは認証されたクライアントだけで、別のクライアントが識別情報を仮定して kinit を行うことはできません。

チケットには一定の属性が与えられています。たとえば、チケットは「転送可能」(新しい認証プロセスを経なくても別のマシンで使用できる) や「遅延」(指定の日付にならないと有効でない) の属性をもつことができます。どのユーザーがどのタイプのチケットを取得できるかなど、チケットをどのように使用するかは、SEAM のインストールや管理の際に決める「方針」によって設定されます。

注 - 「資格」と「チケット」という用語がよく出てきます。広い意味の Kerberos では、これらの用語は同じ意味で使われることがあります。しかし、技術的には資格はチケットとそのセッションに対するセッション鍵からなります。この違いについては、447ページの「SEAM によるサービスへのアクセス」 の「SEAM によるサービスのアクセス」で詳しく説明します。

プリンシパル

SEAM 内のクライアントはその「プリンシパル」で識別されます。プリンシパルとは、KDC がチケットを割り当てることができる固有の識別情報です。プリンシパルは、joe などのユーザーや nfs などのサービスです。

プリンシパル名は慣習で「一次」、「インスタンス」、「レルム」という 3 つの部分からなります。joe/admin@ENG.ACME.COM は一般的な SEAM プリンシパルの例です。各文字列は次の意味を持ちます。

- joe が一次です。これには、この例のようなユーザー名や nfs などのサービスを指定します。一次には host を指定することもできます。host は、これが、さまざまなネットワークサービスを提供するために設定されたサービスプリンシパルであることを意味します。
- admin はインスタンスです。インスタンスは、ユーザープリンシパルでは省略可能ですが、サービスプリンシパルでは必須です。たとえば、ユーザー joe がときどきシステム管理者として活動する場合は、joe/admin として通常のユーザー識別情報と区別することができます。同じように、joe が 2 つのホストにアカウントを持っている場合は、インスタンスは異なる 2 つのプリンシパル名を持つこと

ができます (たとえば、joe/denver.acme.com と joe/boston.acme.com)。SEAM では、joe と joe/admin は全く別のプリンシパルとして扱われます。

サービスプリンシパルでは、インスタンスは完全修飾されたホスト名です。bigmachine.eng.acme.com はこのようなインスタンスの例です。したがって、一次とインスタンスは、たとえば、nfs/bigmachine.eng.acme.com または host/bigmachine.eng.acme.com と表します。

- ENG.ACME.COM は SEAM レルムです。レルムについては、次の項で説明します。

次に示すのはすべて、有効なプリンシパル名です。

- joe
- joe/admin
- joe/admin@ENG.ACME.COM
- nfs/host.eng.acme.com@ENG.ACME.COM
- host/eng.acme.com@ENG.ACME.COM

レルム

レルムとはドメインのようなもので、同じ「master KDC」の下にあるシステムをグループとして定義する論理ネットワークです (下記を参照)。図 21–1 はレルムの相互関係を表したものです。階層的なレルムでは、1 つのレルムが他のレルムのスーパーセットになります。非階層的なレルムでは、2 つのレルム間のマッピングを定義する必要があります。SEAM では、レルム間で共通の認証が可能です。その場合、各レルムの KDC に、他のレルムのプリンシパルエントリが必要になるだけです。

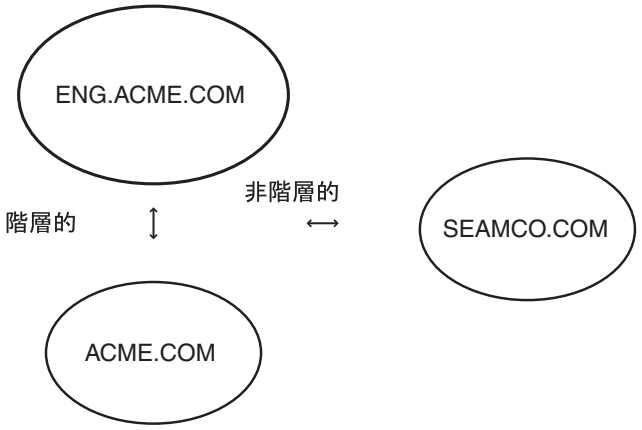


図 21–1　レルム

レルムとサーバー

それぞれのレルムには、プリンシパルデータベースのマスターコピーを保守するサーバーが含まれていなければなりません。これを「マスター KDC サーバー」と呼びます。さらにレルムには、プリンシパルデータベースの重複データベースを保持する「スレーブ KDC サーバー」が少なくとも 1 つ必要です。マスターおよびスレーブ KDC サーバーは共に認証を確立するために使用するチケットを作成します。

レルムにはさらに SEAM サーバーを 2 種類持つことができます。1 つはSEAM ネットワーク「アプリケーションサーバー」で、Kerberos 対応のアプリケーション (`ftp`、`telnet`、`rsh` など) へのアクセスを提供するサーバーです。もう 1 つは、Kerberos 認証を使用する「NFS サーバー」です。NFS サーバーは NFS サービスを提供します。

図 21–2 は、レルムの構成例を示したものです。

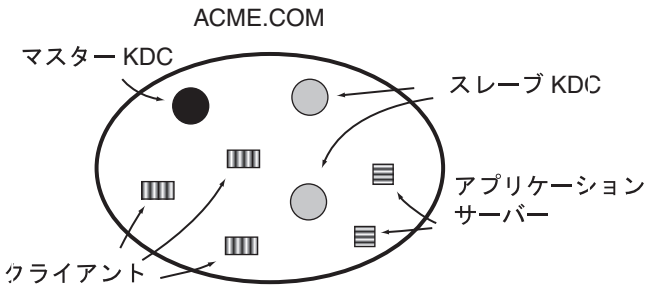


図 21–2　一般的なレルム

セキュリティサービス

SEAM は、ユーザーの認証を行う他に、次の 2 つのセキュリティサービスを提供します。

- 「完全性」。認証があるネットワーク上のクライアントが本人であるかどうかを確認するのと同様に、完全性は、クライアントの送信データが有効なもので、伝送の間に改ざんされていないことを確認します。この確認は、データの暗号チェックサムによって行われます。完全性にはユーザー認証も含まれます。
- 「プライバシ」。プライバシでは、セキュリティの保護がさらに強化されます。プライバシは、伝送データの完全性を検査するだけでなく、伝送前にデータを暗号化して盗聴を防ぎます。プライバシは認証も行います。

注 - プライバシサポートは Solaris Encryption Kit CD に含まれています。

開発者は、RPCSEC_GSS プログラミングインタフェースを使用することにより、セキュリティサービスを選択可能な RPC ベースのアプリケーションを設計できます。

第 22 章

SEAM の構成

この章では、ネットワークアプリケーションサーバー、NFS サーバー、および SEAM クライアントの構成手順を説明します。手順の多くは root アクセスを必要とするため、この作業はシステム管理者や上級ユーザーが行ってください。

SEAM 管理作業マップ

表 22–1 は、Solaris 8 の SEAM に必要な管理作業を示したものです。これらの手順を行うには、KDC が admin サーバーにインストールされていなければなりません。

表 22–1 SEAM 管理作業マップ

作業	説明	使用する手順
SEAM クライアントの構成	SEAM クライアントを手動で構成する手順	416ページの「SEAM クライアントの構成」
NTP のインストール (省略可能)	SEAM が正しく動作するためには、レルムにあるすべてのシステムのクロックが同期していなければならない	425ページの「KDC と SEAM クライアントのクロックの同期化」
SEAM NFS サーバーの構成	Kerberos 認証を必要とするファイルシステムを共有するようにサーバーを設定する手順	419ページの「SEAM NFS サーバーの構成作業マップ」

SEAM クライアントの構成

SEAM クライアントは、SEAM サービスを使用する同じネットワーク上のすべてのホスト (KDC サーバーを除く) です。この節では、SEAM クライアントのインストール手順と、root 認証を使用して NFS ファイルシステムをマウントするための特定の方法を説明します。

SEAM クライアントを構成する手順は 2 つあります。419ページの「SEAM クライアントの構成を完成する方法」は、システムのインストール中に部分的に設定された SEAM クライアントを構成する情報です。416ページの「SEAM クライアントを構成する方法」は、Solaris 8 リリースのインストールで SEAM の構成を全く行わなかった場合の SEAM クライアントの構成手順です。

▼ SEAM クライアントを構成する方法

次の構成パラメータが使用されます。

レルム名 = `ACME.COM`

DNS ドメイン名 = `acme.com`

マスター KDC = `kdc1.acme.com`

スレーブ KDC = `kdc2.acme.com`

クライアント = `client.acme.com`

admin プリンシパル = kws/admin

ユーザープリンシパル = mre

1. **SEAM** クライアントを構成するための前提条件

   admin サーバーの KDC がすでに構成され、動作していなければなりません。さらに、DNS がインストールされ、/etc/resolv.conf ファイルが正しく構成されている必要があります。

2. クライアント上でスーパーユーザーになります。

3. **PAM** 構成ファイル **(**pam.conf**)** を編集します。

   最後の 8 行からコメント記号を削除して Kerberos PAM モジュールを有効にします。

```
client1 # tail -11 /etc/pam.conf
#
# Support for Kerberos V5 authentication (uncomment to use Kerberos)
#
rlogin auth optional   /usr/lib/security/$ISA/pam_krb5.so.1 try_first_pass
login  auth optional   /usr/lib/security/$ISA/pam_krb5.so.1 try_first_pass
dtlogin        auth optional   /usr/lib/security/$ISA/pam_krb5.so.1 try_first_pass
other  auth optional   /usr/lib/security/$ISA/pam_krb5.so.1 try_first_pass
dtlogin        account optional /usr/lib/security/$ISA/pam_krb5.so.1
other  account optional /usr/lib/security/$ISA/pam_krb5.so.1
other  session optional /usr/lib/security/$ISA/pam_krb5.so.1
other  password optional /usr/lib/security/$ISA/pam_krb5.so.1 try_first_pass
```

4. **NFS** セキュリティサービス構成ファイル **(**nfssec.conf**)** を編集します。

   Kerberos サービスを記述する行からコメント記号を削除します。

```
client1 # cat /etc/nfssec.conf
  .
  .
#
# Uncomment the following lines to use Kerberos V5 with NFS
#
krb5            390003  kerberos_v5     default -       # RPCSEC_GSS
krb5i           390004  kerberos_v5     default integrity       # RPCSEC_GSS
default          1       -       -       -       # default is AUTH_SYS
```

5. **Kerberos** 構成ファイル (`krb5.conf`) を編集します。

   デフォルトのファイルを変更する場合は、レルム名とサーバー名を変更する必要があります。

```
client1 # cat /etc/krb5/krb5.conf
[libdefaults]
        default_realm = ACME.COM

[realms]
                ACME.COM = {
                kdc = kdc1.acme.com
                kdc = kdc2.acme.com
                admin_server = kdc1.acme.com
        }

[domain_realm]
        .acme.com = ACME.COM
```

6. (省略可能) **NTP** または別のクロック同期化機構を使用して、マスター **KDC** のクロックと同期させます。

   NTP については、425ページの「KDC と SEAM クライアントのクロックの同期化」を参照してください。

7. 新しいプリンシパルを追加します。

   KDC とともに提供される管理ツールを使用して、クライアントに対して新しいプリンシパルを追加します。

   a. **NFS** サービスプリンシパルを次の名前で作成します。

      `nfs/client1.acme.com`

   b. **root** プリンシパルを次の名前で作成します。

      `root/client1.acme.com`

   c. `host` プリンシパルを次の名前で作成します。

      `host/client1.acme.com`

   d. `root` プリンシパルを **keytab** ファイルに追加します。

      `root/client1.acme.com` プリンシパルが keytab ファイルに追加されていることを確認します。

8. クライアントから **Kerberos** チケットの期限切れをユーザーに警告する場合は、`/etc/krb5/warn.conf` ファイルのエントリを構成します。

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

▼ SEAM クライアントの構成を完成する方法

クライアントをインストールするときに部分的にインストールされた SEAM クライアントを構成する場合は、416ページの「SEAM クライアントを構成する方法」の手順に従ってください。ただし、インストールはすでに開始されているため、`pam.conf`、`nfssec.con`、`krb5.conf` を編集しないでその内容を確認してください。

SEAM NFS サーバーの構成作業マップ

NFS サービスは UNIX UID を使用してユーザーを識別しますが、プリンシパルを直接使用することはできません。そのため、プリンシパルを UID に対応づけるために、ユーザープリンシパルを UNIX UID に対応させる資格テーブルを作成する必要があります。次の手順は、SEAM NFS サーバーの構成、資格テーブルの作成、NFS マウントするファイルシステムの Kerberos セキュリティモードの開始に必要な作業を中心とします。表 22–2 はこの節で行う作業の説明です。

表 22–2 SEAM NFS サーバーの構成作業マップ

作業	説明	使用する手順
SEAM NFSサーバーの構成	Kerberos 認証を必要とするファイルシステムを共有するようにサーバーを設定する手順	420ページの「SEAM NFS サーバーを構成する方法」
資格テーブルのバックエンド機構の変更	`gsscred` によって使用されるバックエンド機構を定義する手順	421ページの「`gsscred` テーブルのバックエンド機構を変更する方法」
資格テーブルの作成	資格テーブルを作成する手順	421ページの「資格テーブルを作成する方法」

表 22–2　SEAM NFS サーバーの構成作業マップ　続く

作業	説明	使用する手順
ユーザープリンシパルを UNIX UID に対応させる資格テーブルを変更する方法	資格テーブルの情報を更新する手順	423ページの「資格テーブルに 1 つのエントリを追加する方法」
Kerberos 認証を使用した、ファイルシステムの共有	セキュリティモードを使用してファイルシステムを共有し、Kerberos 認証を必須にする手順	423ページの「複数の Kerberos セキュリティモード使用して安全な NFS 環境を設定する方法」

▼ SEAM NFS サーバーを構成する方法

この手順を行う場合には、マスター KDC がすでに構成されていなければなりません。この手順のプロセスを完全にテストするには、いくつかのクライアントが必要です。この手順では、次の構成パラメタを使用します。

レルム名 = `ACME.COM`

DNS ドメイン名 = `acme.com`

NFS サーバー = `denver.acme.com`

admin プリンシパル = `kws/admin`

1. **SEAM NFS サーバーを構成するための前提条件**

   SEAM クライアントソフトウェアがインストールされていなければなりません。

2. **(省略可能) NTP クライアントまたは別のクロック同期化機構をインストールします。**

   NTP については、425ページの「KDC と SEAM クライアントのクロックの同期化」を参照してください。

3. **新しいプリンシパルを追加します。**

   KDC とともに提供される管理ツールを使用して、NFS サーバーの新しいプリンシパルを追加します。

   a. **サーバーの NFS サービスプリンシパルを次の名前で作成します。**

      `nfs/denver.acme.com`

b. **(省略可能) NFS サーバーの root プリンシパルを次の名前で作成します。**

   `root/denver.acme.com`

c. **サーバーの NFS サービスプリンシパルをサーバーの keytab に追加します。**

   `nfs/denver.acme.com` プリンシパルが keytab ファイルに追加されていることを確認します。

4. `gsscred` テーブルを作成します。

   詳細は、421ページの「資格テーブルを作成する方法」を参照してください。

5. **Kerberos セキュリティモードを使用して NFS ファイルシステムを共有します。**

   詳細は、423ページの「複数の Kerberos セキュリティモード使用して安全な NFS 環境を設定する方法」を参照してください。

6. **各クライアントでユーザープリンシパルと root プリンシパルを認証します。**

▼ gsscred テーブルのバックエンド機構を変更する方法

1. **NFS サーバーでスーパーユーザーになります。**

2. `/etc/gss/gsscred.conf` を編集してこの機構を変更します。

   バックエンド機構には、`files`、`xfn_files`、`xfn_nis`、`xfn_nisplus`、`xfn` のどれか 1 つを使用できます。個々の機構の利点については、450ページの「`gsscred` テーブルの使用」で説明します。

▼ 資格テーブルを作成する方法

`gsscred` 資格テーブルは、SEAM プリンシパルを UID に対応づけるために NFS サーバーによって使用されます。NFS クライアントが Kerberos 認証を使用して NFS サーバーのファイルシステムをマウントするには、このテーブルを作成するか使用可能にしなければなりません。

1. **適切なサーバーでスーパーユーザーになります。**

このコマンドをどのサーバーからどの ID を使用して実行するかは、この gsscred テーブルをサポートするバックエンド機構として何が選択されているかによって異なります。xfn_nisplus 以外の機構では、root になる必要があります。

使用するバックエンド機構	実行する場所
files	NFS サーバーで実行します。
xfn	デフォルトの xfn ファイル設定にしたがってホストを選択します。
xfn_files	NFS サーバーで実行します。
xfn_nis	NIS マスターで実行します。
xfn_nisplus	NIS+ データを変更する権限がある限りどこでも実行できます。

2. **(省略可能)** /var/fn が存在せず、かつ xfn オプションの **1** つを使用したい場合は、最初の **XFN** データベースを作成します。

```
# fnselect files
# fncreate -t org -o org//
```

3. gsscred を使用して資格テーブルを作成します。

   次のコマンドは、/etc/nsswitch.conf に passwd エントリとともにリストされているすべてのソースから情報を収集します。資格テーブルにローカルのパスワードエントリを入れたくない場合は、files エントリを一時的に削除する必要があることがあります。詳細は、gsscred(1M) のマニュアルページを参照してください。

```
# gsscred -m kerberos_v5 -a
```

▼ 資格テーブルに 1 つのエントリを追加する方法

この手順を行うには、gsscred テーブルが NFS サーバーにインストールされていなければなりません。

1. **NFS サーバーでスーパーユーザーになります。**

2. gsscred を使用してエントリをテーブルに追加します。

```
# gsscred -m [mech] -n [name] -u [uid] -a
```

mech	使用するセキュリティ機構
name	KDC に定義されている、ユーザーのプリンシパル名
uid	パスワードデータベースに定義されている、ユーザーの UID
–a	UID をプリンシパル名マッピングに追加します

例 — 資格テーブルの単一エントリを変更する

次の例では、sandy という名前のユーザーエントリを追加し、UID 3736 に対応させます。UID をコマンド行に指定しないと、パスワードファイルのものが使用されます。

```
# gsscred -m kerberos_v5 -n sandy -u 3736 -a
```

▼ 複数の Kerberos セキュリティモード使用して安全な NFS 環境を設定する方法

1. **NFS サーバーでスーパーユーザーになります。**

2. /etc/dfs/dfstab ファイルを編集して、必要なセキュリティモードを sec= オプションに指定して適切なエントリに追加します。

```
# share -F nfs -o sec=mode filesystem
```

mode	共有するときに使用するセキュリティモード。複数のセキュリティモードを追加すると、デフォルトとしてリストの最初のモードが autofs によって使用される
filesystem	共有するファイルシステムへのパス

指定されたファイルシステムのファイルにアクセスするすべてのクライアントは、Kerberos 認証が必要です。ファイルのアクセスを完了するには、NFS クライアント上のユーザープリンシパルと root プリンシパルが両方とも認証されなければなりません。

3. **NFS** サービスがサーバーで動作しているか確認します。

   これが最初の share コマンドまたは最初の一連の share コマンドなら、NFS デーモンが動作していない可能性があります。次のコマンドを実行すると、デーモンが終了し、再起動します。

```
# /etc/init.d/nfs.server stop
# /etc/init.d/nfs.server start
```

4. **(**省略可能**) autofs** を使用する場合は、auto_master データを編集してデフォルト以外のセキュリティモードを選択します。

   ファイルシステムのアクセスに autofs を使用しない場合やセキュリティモードとしてデフォルトを使用する場合は、この手順を行う必要はありません。

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

5. **(**省略可能**)** 手動で mount コマンドを実行し、デフォルト以外のモードを使用してファイルシステムにアクセスします。

   この代わりに、mount コマンドにセキュリティモードを指定できますが、手順のオートマウンタは利用できません

```
# mount -F nfs -o sec=krb5p /export/home
```

例 — 1 つの Kerberos セキュリティモードを使用してファイルシステムを共有する

次の例を実行すると、ファイルにアクセスする前に Kerberos 認証が必要になります。

```
# share -F nfs -o sec=krb5 /export/home
```

例 — 複数の Kerberos セキュリティモードを使用してファイルシステムを共有する

次の例では、3 つの Kerberos セキュリティモードをすべて選択します。マウント要求でセキュリティモードが指定されていないと、NFS V3 のすべてのクライアントに対しリストの最初のモードが使用されます (この場合は krb5)。さらに、440ページの「share コマンドの変更」を参照してください。

```
# share -F nfs -o sec=krb5:krb5i:krb5p /export/home
```

KDC と SEAM クライアントのクロックの同期化

Kerberos 認証システムを利用するホストはすべて、それぞれの内部クロックが、指定された時間の範囲 (「クロックスキュー」) 内で相互に同期していなければなりません。クロックスキューは、もう 1 つの Kerberos セキュリティチェックとして使用されます。参加するホストの同期差がクロックスキューを超えていると、クライアント要求は拒否されます。

アプリケーションサーバーが再実行要求を認識し拒否する目的で、すべての Kerberos プロトコルメッセージをどのくらいの間追跡管理しなければならないかも、クロックスキューで決まります。そのため、クロックスキュー値が長いほど、アプリケーションサーバーはそれだけ多くの情報を収集しなければなりません。

最大クロックスキューのデフォルト値は 300 秒 (5 分) です。この値は、krb5.conf ファイルの libdefaults セクションで変更できます。

注 - セキュリティ上の理由から、クロックスキュー値は 300 秒より大きくしないでください。

KDC と SEAM クライアントの間でクロックの同期を維持することは重要であるため同期の維持に Network Time Protocole (NTP) を使用することをお奨めします。University of Delaware が作成した NTP パブリックドメインソフトウェアが Solaris 2.6 以降の Solaris ソフトウェアに含まれています。

注 - クロックを同期化するもう 1 つの方法では、`cron` ジョブで `rdate` コマンドを使用します。この方が NTP を使用するよりも簡単ですが、ここでは NTP を中心に説明します。ネットワークを使用してクロックを同期化する場合は、クロック同期化プロトコル自体も安全でなければなりません。

NTP を使用すると、正確な時間とネットワーククロック同期をネットワーク環境で管理できます。NTP は基本的にはクライアントサーバー実装の状態をとります。1 つのシステムをマスタークロック (NTP サーバー) として選択し、他のすべてのシステムをマスタークロックと同期するクライアントとして設定します (NTP クライアント)。同期化は `xntpd` デーモンによって行われます。このデーモンは、UNIX システムの時刻をインターネット標準時刻サーバーに合わせて設定および保守します。NTP のクライアントサーバー実装の例を図 22–1 で示します。

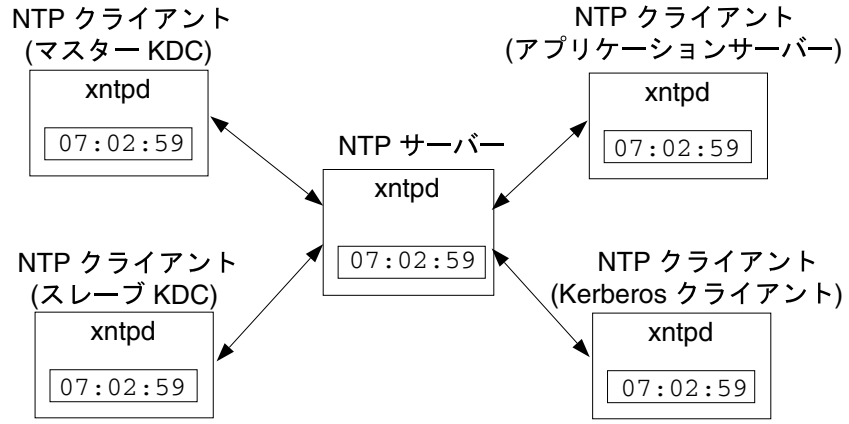


図 22–1　NTP を使用したクロック同期

KDC と SEAM クライアントでクロックの同期を維持するためには、次の手順が必要です。

1. ネットワークに NTP サーバーを設定します。(NTP サーバーは、マスター KDC 以外であればどのシステムでもかまいません。) 500ページの「NTP サーバーを設定する方法」を参照してください。
2. ネットワークの KDC と SEAM クライアントを構成するときに、それらを NTP サーバーの NTP クライアントとして設定します。500ページの「NTP クライアントを設定する方法」を参照してください。

SEAM クライアントのエラーメッセージ

SEAM のすべてのエラーメッセージが、『*Sun Enterprise Authentication Mechanism* ガイド』の「SEAM のエラーメッセージと問題の解決」に記載されています。

第 23 章

SEAM リファレンス

この章では、SEAM が動作するシステムでチケットの取得、表示、および破棄を行う方法と Kerberos パスワードの選択や変更を行う方法を説明します。この章には、多数の SEAM 製品ファイルがリストされています。さらに、この章では、Kerberos 認証システムがどのように動作するかを詳しく説明します。

チケットの管理

この節では、チケットの取得、表示、および破棄を行う方法を説明します。チケットの概要については、409ページの「SEAM の動作」を参照してください。

チケットを意識する必要があるか

PAM の構成方法によっては、ログインしたときにチケットが自動的に取得されます。これがデフォルトの動作ですが、使用する SEAM 構成によっては、チケットの自動転送機能が含まれていないことがあります。

Kerberos 化されたほとんどのコマンドは、終了するときにチケットを自動的に破棄します。しかし、念のために、コマンドが終了したときに Kerberos チケットを明示的に破棄したい場合は、kdestroy を使用します。kdestroy の詳細は、432ページの「チケットを破棄する方法」を参照してください。

チケットの有効期限については、444ページの「チケットの有効期間」を参照してください。

▼ チケットを作成する方法

通常は、ログインするとチケットが自動的に作成されるため、チケットを取得するために特別な作業をする必要はありません。ただし、次の場合には、チケットを作成しなければならないことがあります。

- チケットの有効期限が切れている。
- デフォルトのプリンシパルの他に別のプリンシパルを使用する必要がある (たとえば、rlogin -l を使って他人としてマシンにログインする)。

チケットを作成するには、kinit コマンドを使用します。

```
% /usr/bin/kinit
```

kinit からはパスワードのプロンプトが表示されます。kinit コマンドの詳しい構文については、kinit(1) のマニュアルページを参照してください。

例 — チケットを作成する

この例では、ユーザー jennifer が自分のシステムにチケットを作成します。

```
% kinit
Password for jennifer@ENG.ACME.COM:  <パスワードの入力>
```

次の例では、ユーザー david が –l オプションを使用して 3 時間有効なチケットを作成します。

```
% kinit -l 3h david@ACME.ORG
Password for david@ACME.ORG:  <パスワードの入力>
```

この例では、ユーザー david が –f オプションを使用して自分用の転送可能チケットを作成します。この転送可能チケットを使用すると、ユーザーは、たとえば、別のシステムにログインし、telnet を使用してさらに別のシステムにログインできます。

```
% kinit -f david@ACME.ORG
Password for david@ACME.ORG:     <パスワードの入力>
```

転送可能チケットをどのように使用するかについては、441ページの「チケットの種類」を参照してください。

▼ チケットを表示する方法

すべてのチケットが同じ属性をもっているわけではありません。たとえば、あるチケットは「転送可能 (forwardable)」、別のチケットは「遅延 (postdated)」、さらに別のチケットはその両方である可能性があります。現在のチケットが何で、どのような属性をもつかを知るには、klist コマンドで –f オプションを使用します。

```
% /usr/bin/klist -f
```

次の記号はチケットに関連付けられる属性です。klist によって表示されます。

F	転送可能 (Forwardable)
f	転送済み (Forwarded)
P	プロクシ可能 (Proxiable)
p	プロクシ (Proxy)
D	遅延可能 (Postdateable)
d	遅延 (Postdated)
R	更新可能 (Renewable)

I	初期 (Initial)
i	無効 (Invalid)

チケットがもつことのできる属性については、441ページの「チケットの種類」を参照してください。

例 — チケットを表示する

次の例は、ユーザー `jennifer` の「初期 (*initial*)」チケットが「転送可能 (*forwardable*)」(`F`) と「遅延 (*postdated*)」(`d`) のプロパティを持っていて、まだ検証 (`i`) されていないことを示します。

```
% /usr/bin/klist -f
Ticket cache: /tmp/krb5cc_74287
Default principal: jenniferm@ENG.ACME.COM

Valid starting                 Expires                 Service principal
09 Mar 99 15:09:51  09 Mar 99 21:09:51  nfs/ACME.SUN.COM@ACME.SUN.COM
        renew until 10 Mar 99 15:12:51, Flags: Fdi
```

次の例は、ユーザー `david` が別のホストから自分のホストに「転送された (*forwarded*)」(`f`) チケットを 2 つ持っていることを示します。これらのチケットは「再転送可能 (*forwardable*)」(`F`) です。

```
% klist -f
Ticket cache: /tmp/krb5cc_74287
Default principal: david@ACME.SUN.COM

Valid starting                 Expires                 Service principal
07 Mar 99 06:09:51  09 Mar 99 23:33:51  host/ACME.COM@ACME.COM
        renew until 10 Mar 99 17:09:51, Flags: fF

Valid starting                 Expires                 Service principal
08 Mar 99 08:09:51  09 Mar 99 12:54:51  nfs/ACME.COM@ACME.COM
        renew until 10 Mar 99 15:22:51, Flags: fF
```

▼ チケットを破棄する方法

チケットは通常、チケットを作成したコマンドが終了すると自動的に破棄されます。しかし、念のために、コマンドが終了したときに Kerberos チケットを明示的に破棄したい場合があります。チケットは盗まれることがあります。チケットを盗ん

だ人は、暗号解除する必要はありますが、その有効期限が切れるまでチケットを使用できます。

チケットを破棄するには、kdestroy コマンドを使用します。

```
% /usr/bin/kdestroy
```

kdestroy はそのユーザーの「すべての」チケットを破棄します。特定のチケットを選択して破棄することはできません。

システムを離れるときに侵入者が権限を使用する危険がある場合は、kdestroy を使用してチケットを破棄するか、スクリーンセーバーを使って画面をロックする必要があります。

注 - チケットを確実に破棄する 1 つの方法は、ホームディレクトリの .logout ファイルに kdestroy コマンドを追加することです。

PAM モジュールが適切に構成されていれば、チケットはログアウト時に自動的に破棄されるため、.login ファイルに kdestroy 呼び出しを追加する必要はありません。ただし、PAM モジュールが適切に構成されていない場合や、構成されているかどうかわからない場合は、システムを終了するときにチケットを確実に破棄するために、kdestroy を .login ファイルに追加することもできます。

パスワード管理

SEAM をインストールすると、2 つのパスワードをもつことになります。通常の Solaris パスワードと Kerberos パスワードです。これらのパスワードは同じでも、異なっていてもかまいません。

login など、Kerberos 化されていないコマンドは、PAM を使用して Kerberos と UNIX の両方で認証するように設定できます。2 つのパスワードが異なっている場合は、ログインで適切な認証を得るために両方のパスワードを入力する必要があります。しかし、2 つのパスワードが同じ場合は、UNIX 用に入力した最初のパスワードが Kerberos で使用されます。

ただし、UNIX と Kerberos に同じパスワードを使用すると、セキュリティを損うおそれがあります。つまり、他人が Kerberos パスワードを入手した場合、UNIX パスワードも安全ではありません。ただし、UNIX と Kerberos に同じパスワードを使用したとしも、Kerberos 環境ではパスワードがネットワークを超えて送信されるこ

とはないため、Kerberos のないサイトよりは安全です。通常、どの方法を選ぶかは、サイトごとに決められる方針に従います。

Kerberos では、Kerberos パスワードはユーザーの識別を行う唯一の情報です。Kerberos パスワードを他人に知られると、Kerberos セキュリティは無意味になります。その人が本人になり代わって「本人の」電子メールを送信したり、本人のファイルの読み取り、編集、削除などをしたり、本人として他のホストにログインしても、違いはだれにもわかりません。したがって、適切なパスワードを選択し、その秘密を保持することは極めて重要です。パスワードは、システム管理者を含め誰にも教えてはいけません。さらに、パスワードは頻繁に変更してください。他人に知られた可能性のある場合は特に変更が必要です。

パスワード選択のヒント

パスワードには、制御キーや Return キーなどを除き、入力できるキーであればどの文字でもほとんど使用できます。良いパスワードとは、覚え易く、しかも他人が簡単に推定できないものです。悪い例は次のようなものです。

- 辞書に出てくる言葉
- よく見られるありふれた名前
- 有名な人やキャラクタの名前
- 形式に関係なく、自分の名前やユーザー名 (逆方向、2 度繰り返すなどを含む)
- 配偶者、子、ペットの名前
- 自分の誕生日や親戚の誕生日
- 社会保険番号、運転免許書番号、パスポート番号、またはこれに類した身分証明書番号
- このマニュアルや他のマニュアルに出てくるサンプルパスワード

良いパスワードとは少なくとも 8 文字からなり、大文字、小文字、番号、句読記号などが混在しているものです。次に例を挙げます。

- 「I2LMHinSF」などの短縮形。(「I too left my heart in San Francisco」と覚える)
- 「WumpaBun」、「WangDangdoodle!」など、発音しやすい意味のない語句
- 「6o′cluck」、「RrriotGrrrlsRrrule!」など、わざとスペルを間違えた語句

注意 - これらの例は使用しないでください。マニュアルの例に使用されているパスワードは侵入者が最初に試みるパスワードです。

パスワードの変更

Kerberos パスワードは 2 つの方法で変更できます。

- 通常の UNIX passwd コマンド。SEAM がインストールされていると、Solaris passwd コマンドでも新しい Kerberos パスワードを求めるプロンプトが自動的に表示されます。

  kpasswd の代わりに passwd を使用する利点は、UNIX と Kerbeors 両方のパスワードを同時に設定できることです。しかし、passwd で両方のパスワードを同時に変更することは一般的に必要ありません。UNIX パスワードだけを変更して Kerberos パスワードは変更しなかったり、その逆はよくあります。

注 - passwd の動きは、PAM モジュールがどのように構成されているかによって異なります。構成によっては、両方のパスワードを変更しなければならないことがあります。あるサイトでは UNIX パスワードの変更が必須であり、別のサイトでは Kerberos パスワードの変更が必須であるということがあります。

- kpasswd パスワードコマンド。kpasswd は passwd とよく似ていますが、違う点の 1 つは、kpasswd では Kerberos パスワードだけを変更するということです。UNIX パスワードを変更する場合は、passwd を使用する必要があります。

  もう 1 つの違いは、kpasswd では、有効な UNIX ユーザーではない Kerberos プリンシパルのパスワードを変更できる点です。たとえば、david/admin は Kerberos プリンシパルですが、実際の UNIX ユーザーではありません。したがって、この場合は、passwd の代わりに kpasswd を使用する必要があります。

注意 - kpasswd を使用するには、SEAS 3.0 リリースに含まれている SEAM 1.0 管理システムを使用する必要があります。さらに、プライバシサポートを読み込んで、このパスワードの変更要求を防止する必要があります。

パスワードを変更しても、変更がシステム全体に伝達されるまでには (とりわけ大きなネットワークでは)、ある程度の時間が必要です。システムの設定方法によりますが、この時間は数分から 1 時間以上になることがあります。パスワードを変更したあとすぐに新しい Kerberos チケットを取得する場合は、新しいパスワードをまず試してください。新しいパスワードが有効でない場合は、前のパスワードを使用して再度試してください。

Kerberos V5 では、システム管理者が有効なパスワードの基準をユーザーごとに設定できます。このような基準は、ユーザーごとに「ポリシー」セット (またはデフォルトポリシー) で定義します。たとえば、jenpol と呼ぶ jennifer のポリシーで

は、パスワードは少なくとも 8 文字からなり、少なくとも 2 種類の文字が混在すると定義されているとします。その場合、パスワードとして sloth を入力すると、kpasswd によって拒否されます。

```
% kpasswd
kpasswd: Changing password for jennifer@ENG.ACME.COM.
Old password:    <jennifer が現在のパスワードを入力する>
kpasswd: jennifer@ENG.ACME.COM's password is controlled by
the policy jenpol
which requires a minimum of 8 characters from at least 2 classes
(the five classes are lowercase, uppercase, numbers, punctuation,
and all other characters).
New password: <jennifer が「sloth」と入力する>
New password (again):  <jennifer が再び「sloth」と入力する>
kpasswd: New password is too short.
Please choose a password which is at least 4 characters long.
```

次に、jennifer はパスワードとして slothrop49 と入力します。slothrop49 は長さが 8 文字以上で、2 種類の文字 (数字と小文字) が混在しているため基準に合っています。

```
% kpasswd
kpasswd: Changing password for jennifer@ENG.ACME.COM.
Old password:  <jennifer が現在のパスワードを入力する>
kpasswd: jennifer@ENG.ACME.COM's password is controlled by
the policy jenpol
which requires a minimum of 8 characters from at least 2 classes
(the five classes are lowercase, uppercase, numbers, punctuation,
and all other characters).
New password:  <jennifer が「slothrop49」と入力する>
New password (again):  <jennifer が再び「slothrop49」と入力する>
Kerberos password changed.
```

例 — パスワードを変更する

次の例では、david が passwd を使用して UNIX と Kerberos のパスワードを両方とも変更します。

```
% passwd
 passwd:  Changing password for david
 Enter login (NIS+) password:          <現在の UNIX パスワードを入力する>
 New password:                         <新しい UNIX パスワードを入力する>
 Re-enter password:                    <新しい UNIX パスワードを確認する>
 Old KRB5 password:                    <現在の Kerberos パスワードを入力する>
 New KRB5 password:                    <新しい Kerberos パスワードを入力する>
```

```
Re-enter new KRB5 password:          <新しい Kerberos パスワードを確認する>
```

上の例では、passwd によって UNIX と Kerberos のパスワードが要求されます。しかし、PAM モジュールで *try_first_pass* が設定されていると、Kerberos パスワードは自動的に UNIX パスワードと同じ内容に設定されます (これがデフォルトの設定です)。この場合、david が Kerberos パスワードを他のものに設定するには、次の例のように kpasswd を使用する必要があります。

次の例では、david が kpasswd を使用して Kerberos パスワードだけを変更します。

```
% kpasswd
kpasswd: Changing password for david@ENG.ACME.COM.
Old password:            <現在の Kerberos パスワードを入力する>
New password:            <新しい Kerberos パスワードを入力する>
New password (again):    <新しい Kerberos パスワードを確認する>
Kerberos password changed.
```

次の例では、david が Kerberos のプリンシパル david/admin (有効な UNIX ユーザーではない) を変更します。この場合、kpasswd を使用します。

```
% kpasswd david/admin
kpasswd:  Changing password for david/admin.
Old password:            <現在の Kerberos パスワードを入力する>
New password:           <新しい Kerberos パスワードを入力する>
New password (again):     <新しい Kerberos パスワードを確認する>
Kerberos password changed.
```

SEAM ファイル

この節では、SEAM 製品に含まれているファイルについて説明します。

表 23–1 SEAM ファイル

ファイル名	説明
/etc/gss/gsscred.conf	gsscred テーブルのデフォルトファイル形式
/etc/gss/mech	RPCSEC_GSS のメカニズム
/etc/gss/qop	RPCSEC_GSS の Quality of Protection (保護品質) パラメータ
/etc/nfssec.conf	NFS 認証セキュリティモードを定義する
/etc/krb5/krb5.conf	Kerberos レルム構成ファイル
/etc/krb5/krb5.keytab	ネットワークアプリケーションサーバーの keytab
/etc/krb5/warn.conf	Kerberos 警告構成ファイル
/etc/pam.conf	PAM 構成ファイル
/tmp/krb5cc_*uid*	デフォルト資格キャッシュ (*uid* はユーザーの 10 進数 UID)
/tmp/ovsec_adm.*xxxxxx*	パスワード変更操作の間だけ有効な一時資格キャッシュ (*xxxxxx* はランダムな文字列)

PAM 構成ファイル

SEAM とともに提供されるデフォルトの PAM 構成ファイルでは、Kerberos 機能を使用するためのエントリがコメント化されています。この新しいファイルには、認証サービス、アカウント管理、セッション管理、パスワード管理の各モジュールを表すエントリが含まれています。

認証モジュールの場合は、新しいエントリとして rlogin、login、dtlogin が含まれています。これらのエントリの例を次に示します。これらのサービスはすべて PAM ライブブラリ /usr/lib/security/pam_krb5.so.1 を使用して Kerberos 認証を行います。

最初の 3 つのエントリでは try_first_pass オプションが使用されています。この場合、ユーザーの最初のパスワードを使用して認証が行われます。最初のパ

スワードを使用するとは、複数のメカニズムが表示されていても、ユーザーは別のパスワードを要求されないという意味です。指定されていない、認証を必要とするすべてのエントリのデフォルトとして other エントリが 1 つ含まれています。

```
# cat /etc/pam.conf
 .
 .
rlogin auth optional /usr/lib/security/pam_krb5.so.1 try_first_pass
login auth optional /usr/lib/security/pam_krb5.so.1 try_first_pass
dtlogin auth optional /usr/lib/security/pam_krb5.so.1 try_first_pass
krlogin auth required /usr/lib/security/pam_krb5.so.1 acceptor
ktelnet auth required /usr/lib/security/pam_krb5.so.1 acceptor
krsh auth required /usr/lib/security/pam_krb5.so.1 acceptor
other auth optional /usr/lib/security/pam_krb5.so.1 try_first_pass
```

アカウント管理では、Kerberos ライブラリを使用する新しいエントリが dtlogin に次のように含まれています。other エントリはデフォルトルールを提供するために 1 つ含まれています。現状では、other エントリによって何の動作も行われません。

```
dtlogin account optional /usr/lib/security/pam_krb5.so.1
other account optional /usr/lib/security/pam_krb5.so.1
```

次に /etc/pam.conf ファイルの最後の 2 つのエントリを示します。セッション管理の other エントリではユーザー資格を破棄します。パスワード管理の新しい other エントリでは Kerberos ライブラリを選択します。

```
other session optional /usr/lib/security/pam_krb5.so.1
other password optional /usr/lib/security/pam_krb5.so.1 try_first_pass
```

SEAM コマンド

この節では、SEAM 製品に含まれているコマンドの一部を示します。

表 23–2 SEAM コマンド

ファイル名	説明
/usr/bin/kdestroy	Kerberos チケットを破棄する
/usr/bin/kinit	Kerberos チケットを許可するチケットを取得およびキャッシュする
/usr/bin/klist	現在の Kerberos チケットをリストする
/usr/bin/kpasswd	Kerberos パスワードを変更する
/usr/bin/ktutil	keytab 保守ユーティリティ
/usr/sbin/gsscred	NFS サービスのための GSS-API トークンを生成および検証する

share コマンドの変更

新しい SEAM コマンドの他に、Solaris 8 リリースには、share コマンドとともに使用する新しいセキュリティモードが含まれています。このモードは /etc/nfssec.conf ファイルに定義されます。share コマンドでは、次のセキュリティモードを使用できます。

krb5 Kerberos 認証を選択する。

krb5i 完全性機能付き Kerberos 認証を選択する。

krb5p 完全性機能とプライバシ機能付き Kerberos 認証を選択する。

share コマンドに複数のモードを指定すると、クライアントがセキュリティモードを指定していない場合は、指定した最初のモードがデフォルトで選択されます。クライアントがセキュリティモードを指定している場合は、指定したモードが使用されます。

Kerberos モードを使用したマウント要求が失敗すると、マウントはセキュリティモード none で完了します。この状態は、NFS クライアントの root プリンシパルが認証されない場合によく発生します。マウント要求が完了したとしても、ファイルが Kerberos で認証されない限り、ユーザーがこのファイルにアクセスすることはで

きません。ファイルシステムが Kerberos セキュリティモードを使用してマウントされていなくても、クライアントとサーバー間のトランザクションには Kerberos 認証が必要です。

SEAM デーモン

SEAM 製品で使用するデーモンを表 23–3 で示します。

表 23–3 SEAM デーモン

ファイル名	説明
`/usr/lib/krb5/ktkt_warnd`	Kerberos 警告デーモン
`/usr/lib/gss/gssd`	GSSAPI デーモン

チケットリファレンス

この節では、チケットについて補足して説明します。

チケットの種類

チケットには、チケットがどのように使用されるかを決めるプロパティがあります。これらのプロパティは、チケットの作成時にチケットに割り当てられます。ただし、チケットのプロパティはあとから変更できます。たとえば、チケットは「転送可能 (*forwardable*)」から「転送済み (*forwarded*)」に変更できます。チケットのプロパティを表示するには `klist` コマンドを使用します (431ページの「チケットを表示する方法」を参照)。

チケットは、次の 1 つまたはそれ以上のプロパティで表されます。

転送可能 / 転送済み
: 「転送可能 (forwardable)」チケットはホストからホストに転送されます。これによって、クライアントは再び認証を受ける必要がありませ

ん。たとえば、ユーザー david が jennifer のマシンで転送可能チケットを取得した場合、david は自身のマシンにログインするときに新しいチケットを取得する必要はありません (再び認証を受ける必要もありません)。転送可能チケットと「プロクシ可能」チケットの違いを比較してください。

初期

「初期 (initial)」チケットは、チケット許可チケットを使わずに直接発行されるチケットです。パスワードを変更するアプリケーションなど、サービスの中には、「初期」チケットを要求するものがあります。これは、そのクライアントが自らの秘密鍵を知っているクライアントであることを確認するためです。つまり、「初期」チケットは、クライアントが認証を受けたばかりであることを示します。これに対し、チケット許可チケットに依存する場合は、そのチケットがしばらくの間使用されていたことを表しています。

無効

「無効 (invalid)」チケットとは、まだ使用可能になっていない遅延チケットです (次の項を参照)。無効チケットは、有効になるまでアプリケーションサーバーから拒否されます。これを有効にするには、開始時期が過ぎたあと、TGS 要求で VALIDATE フラグをオンにしてクライアントがこのチケットを KDC に提示する必要があります。

遅延可能 **(postdatable) /** 遅延

「遅延 (postdated)」チケットとは、作成されても指定された時期まで有効にならないチケットです。たとえばこのようなチケットは、夜遅く実行されるバッチジョブに使用するのに便利です。チケットが盗まれてもバッチジョブが実行されるまで使用できないためです。「遅延」チケットは「無効」チケットとして発行されます。開始時期がきて、クライアントが KDC から検証を受けるまで「無効」のままです (上の「無効」を参照)。「遅延」チケットは通常、チケット許可

チケットの有効期限まで有効です。ただし、チケットが「更新可能」な場合、チケットの有効期限は通常、チケット許可チケットの全有効期限と同じに設定されます (「更新可能」を参照)。

プロクシ可能 / プロクシ

場合によっては、プリンシパルがサービスにサービス自身のために操作を行わせたい場合があります。たとえば、プリンシパルがサービスに対し第 3 のホストで印刷ジョブを実行するように要求する場合です。サービスは、その操作の間に限り、クライアントの識別情報を代わりに使用します。その場合、サービスは、クライアントの「プロクシ (proxy)」として動作するといいます。チケットを作成するときには、プロクシのプリンシパル名を指定する必要があります。

「プロクシ可能 (proxiable)」チケットは「転送可能」チケットに似ていますが、「プロクシ可能」チケットが 1 つのサービスに対してのみ有効であるのに対し、「転送可能」チケットはサービスに対しクライアントの識別情報の完全な使用を許可します。したがって、「転送可能」チケットは一種のスーパープロクシと考えられます。

更新可能

チケットに非常に長い有効期限を与えるとセキュリティを損うおそれがあるため、チケットを「更新可能 (renewable)」にすることができます。「更新可能」チケットには 2 つの有効期限があります。1 つはチケットの現在のインスタンスの有効期限、もう 1 つは任意のチケットの最長有効期限です。クライアントがチケットの使用を継続したいときは、最初の有効期限が切れる前にチケットの有効期限を更新します。たとえば、すべてのチケットの最長有効期限が 10 時間のときに、あるチケットが 1 時間だけ有効だとします。このチケットを保持するクライアントが 1 時間を超えて使用したい場合は、その時間内にチケットの有効期限を更新する必要があります。チケットが最長有効期限 (10 時間)

に達すると、チケットの有効期限が自動的に切れ、それ以上更新できなくなります。

チケットを表示してその属性を見る方法については、431ページの「チケットを表示する方法」を参照してください。

チケットの有効期間

プリンシパルがチケットを取得すると、チケット許可チケットであっても、チケットの有効期限は次の値のうち最も小さいものに設定されます。

- `kinit` 使用してチケットを取得する場合、`kinit` の `–l` オプションに指定した有効期限値
- `kdc.conf` ファイルに指定された最長有効期限値 (`max_life`)
- チケットを提供するサービスプリンシパルに対し Kerberos データベースに指定されている最長有効期限値 (`kinit` の場合、サービスプリンシパルは `krbtgt/`*realm*)
- チケットを要求するユーザープリンシパルに対し Kerberos データベースに指定されている最長有効期限値

図 23–1 で、TGT の有効期限がどのようにして決まるか、4 つの有効期限の値がどこで指定されるかを示します。この図は TGT の有効期限がどのようにして決まるかを示していますが、基本的には、どのプリンシパルがチケットを取得する場合でも同じことです。違いは、`kinit` で有効期限を与えないことと、`krbtgt/`*realm* プリンシパルの代わりに、チケットを提供するサービスプリンシパルが最長有効期限を提供することです。

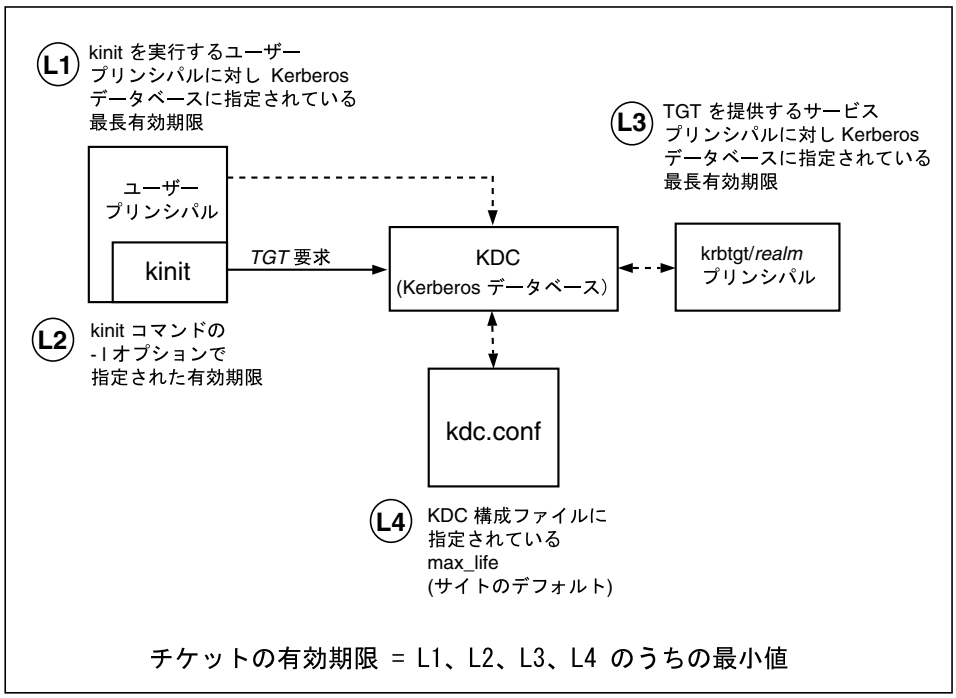


図 23–1　TGT の有効期限の決め方

「更新可能」チケットの有効期限も次の 4 つの値の最小値で決まります。ただし、この場合は更新可能有効期限の値が使用されます。

- `kinit` を使用してチケットを取得または継続する場合、`kinit` の `–r` オプションに指定した更新可能有効期限値
- `kdc.conf` ファイルに指定された最長更新可能有効期限値 (`max_renewable_life`)
- チケットを提供するサービスプリンシパルに対し Kerberos データベースに指定されている最長有効期限更新可能値 (`kinit` の場合、サービスプリンシパルは `krbtgt/`*realm*)
- チケットを要求するユーザープリンシパルに対し Kerberos データベースに指定されている最長有効期限更新可能値

プリンシパル名

チケットはプリンシパル名で識別され、プリンシパル名はユーザーやサービスを識別します。表 23–4 にプリンシパル名の例を示します。

表 23–4　プリンシパル名の例

プリンシパル名	説明
root/boston.acme.com@ACME.COM	NFS クライアントの root アカウントに関連付けられたプリンシパル。root プリンシパルと呼ばれ、認証された NFS マウントを行う場合に必要
host/boston.acme.com@ACME.COM	Kerberos 化されたアプリケーション (klist など) やサービス (NFS サービスなど) で使用するプリンシパル
username@ACME.COM	ユーザー用のプリンシパル
username/admin@ACME.COM	KDC データベースを管理するために使用できる admin プリンシパル
nfs/boston.acme.com@ACME.COM	nfs サービスによって使用されるプリンシパル。これは host プリンシパルの代わりに使用できる

認証システムの動作

アプリケーションを使用してリモートシステムにログインするには、識別情報を証明するチケットとそれに対応するセッション鍵を指定する必要があります。セッション鍵には、ユーザーやアクセスするサービスに特有の情報が含まれています。ユーザーすべてのチケットとセッション鍵は、ユーザーが最初にログインするときに KDC によって作成されます。チケットとそれに対応するセッション鍵が 1 つの資格となります。複数のネットワークサービスを使用する場合には、ユーザーは多数の資格を収集できます。ユーザーは特定のサーバーで動作するサービスごとに 1 つの資格を必要とします。たとえば、boston というサーバーで動作する ftp サービスにアクセスするには 1 つの資格が必要であり、別のサーバーで動作する ftp サービスにアクセスするにはそれ独自の別の資格が必要です。

資格の作成や格納は透過的に行われます。資格は KDC によって作成され、要求者に送信されます。資格は、受信されると資格キャッシュに格納されます。

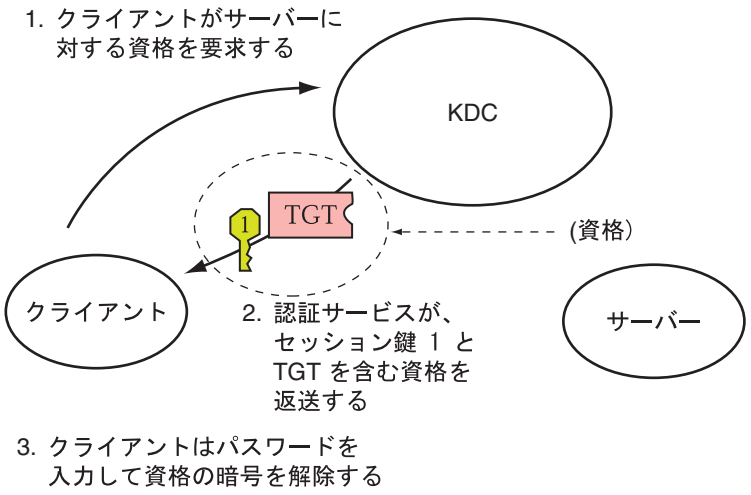
SEAM によるサービスへのアクセス

特定のサーバーの特定のサービスにアクセスする場合、ユーザーは 2 つの資格を取得する必要があります。最初は TGT として知られるチケット許可サービスに対する資格です。チケット許可サービスは、この資格の暗号を解除すると、ユーザーからアクセスを要求されているサーバーの資格をさらに作成します。ユーザーは、この 2 つめの資格を使用してサーバー上のサービスへのアクセスを要求します。サーバーがこの資格の暗号を解除すると、ユーザーはアクセスを許可されます。この処理を次の項とそれに続く図で説明します。

チケット許可サービスに対する資格の取得

1. 認証処理を開始するために、クライアントが特定のユーザープリンシパルの要求を認証サーバーに送信します。この要求の送信では暗号は使用されません。要求には機密情報が含まれていないため、暗号を使う必要はありません。
2. 認証サービスは要求を受信すると、ユーザーのプリンシパル名を KDC データベースから探します。合致するプリンシパルが見つかると、認証サービスはそのプリンシパルの非公開鍵を取得します。次に認証サービスは、クライアントとチケット許可サービスが使用するセッション鍵 (セッション鍵 1) とチケット許可サービス用のチケット (チケット 1) を生成します。このチケットはチケット許可チケット (TGT) ともいいます。セッション鍵とチケットはユーザーの非公開鍵を使って暗号化され、クライアントに返送されます。
3. クライアントは、ユーザープリンシパルの非公開鍵を使用して、この情報からセッション鍵 1 とチケット 1 の暗号を解除します。非公開鍵を知っているのはユーザーと KDC データベースだけである必要があるため、パケットの情報は安全に保たれなければなりません。クライアントはこの情報を資格キャッシュに格納します。

通常、この処理では、ユーザーはパスワードの入力を要求されます。非公開鍵の作成で KDC データベースから取り出されたパスワードが入力したパスワードと同じであると、認証サービスから送信された情報は正しく暗号解除されます。これでクライアントは、チケット許可サービスに対して使用する資格を取得したことになります。次にクライアントはサーバーに対する資格を要求します。



TGT = チケット許可チケット (ticket-granting ticket)	KDC = 鍵発行センター (Key Distribution Center)

図 23–2　チケット許可サービスに対する資格の取得

サーバーに対する資格の取得

1. 特定のサーバーにアクセスするには、クライアントがその前にサーバーに対する資格を認証サービスから取得していなければなりません (447ページの「チケット許可サービスに対する資格の取得」を参照)。次にクライアントは、チケット許可サービスに要求を送信します。この要求には、サービスプリンシパル名、チケット 1 と、セッション鍵 1 で暗号化された認証者が含まれています。チケット 1 は、チケット許可サービスのサービス鍵を使用して認証サービスによって暗号化されたものです。
2. チケット許可サービスはチケット許可サービスのサービス鍵を知っているため、チケット 1 の暗号を解除できます。チケット 1 の情報にはセッション鍵 1 が含まれているため、チケット許可サービスは認証者の暗号を解除できます。この時点で、ユーザープリンシパルはチケット許可サービスによって認証されます。
3. 認証が正常に終ると、チケット許可サービスは、ユーザープリンシパルとサーバーに対するセッション鍵 (セッション鍵 2) とサーバーに対するチケット

(チケット 2) を生成します。セッション鍵 2 とチケット 2 はセッション鍵 1 を使って暗号化されます。セッション鍵 1 を知っているのはクライアントとチケット許可サービスだけですので、この情報は安全であり、ネットワークを介して安全に送信されます。

4. クライアントはこの情報パケットを受信すると、前に資格キャッシュに格納したセッション鍵 1 を使用して情報の暗号を解除します。クライアントは、サーバーに対して使用する資格を取得したことになります。次にクライアントは、そのサーバーの特定のサービスにアクセスする要求を行います。

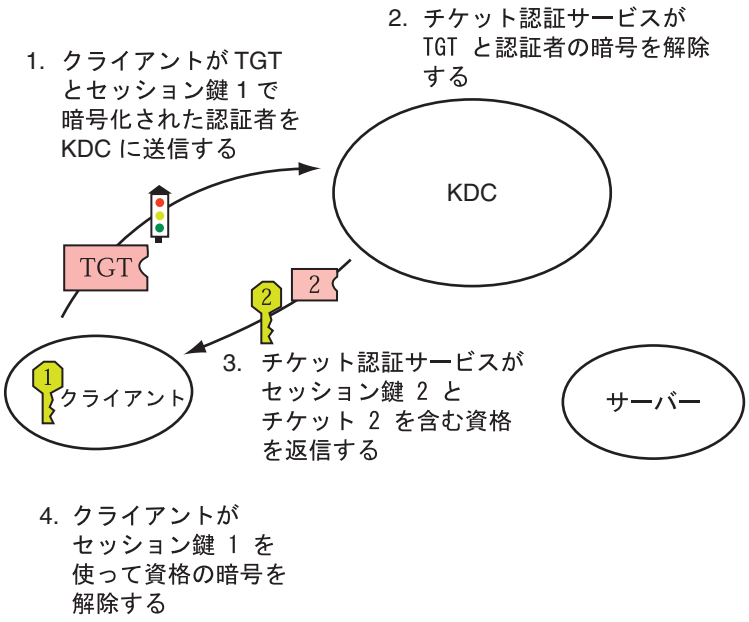


図 23–3　サーバーに対する資格の取得

特定のサービスへのアクセス権の取得

1. 特定のサービスにアクセスするには、クライアントがその前に認証サーバーからチケット許可サービスの資格を、チケット許可サービスからサーバーの資格をそれぞれ取得していなければなりません (447ページの「チケット許可サービスに対する資格の取得」と 448ページの「サーバーに対する資格の取得」を参照)。クライアントは、チケット 2 と別の認証者を含む要求をサーバーに送信します。認証者はセッション鍵 2 を使用して暗号化されます。

2. チケット 2 は、サービスのサービス鍵を使用してチケット許可サービスによって暗号化されています。サービス鍵はサービスプリンシパルが知っているため、サービスはチケット 2 の暗号を解除し、セッション鍵 2 を取得できます。次に、セッション鍵 2 を使用して認証者の暗号が解除されます。認証者の暗号が正しく解除されると、サーバーへのアクセスがクライアントに許可されます。

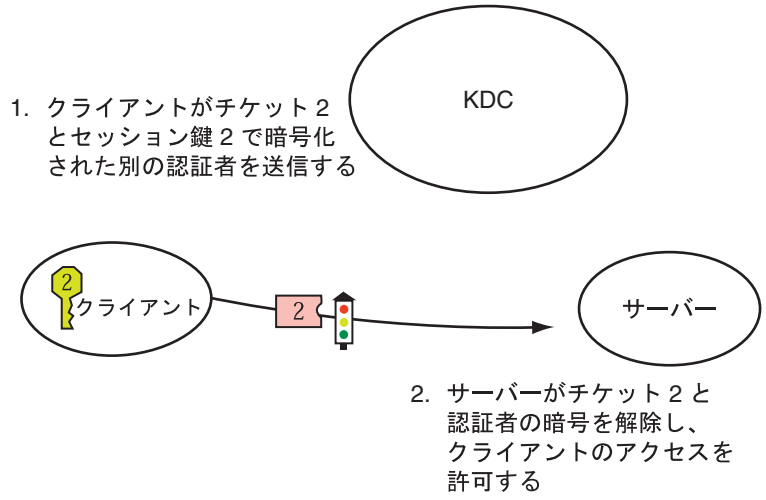


図 23–4　特定のサーバーへのアクセス権を取得

gsscred テーブルの使用

gsscred テーブルは、NFS サーバーが SEAM ユーザーを識別するときに使用します。NFS サービスは UNIX ID を使用してユーザーを識別しますが、この ID はユーザープリンシパルや資格の一部ではありません。gsscred テーブルは、パスワードファイルから得られる UNIX ID とプリンシパル名を対応付けるテーブルです。このテーブルは、KDC データベースにデータを入力したあとに作成および開始する必要があります。

クライアントの要求が到着すると、NFS サービスはプリンシパル名を UNIX ID に対応づけようとします。そして、この変換に失敗すると、gsscred テーブルを使用します。kerberos_v5 メカニズムでは、root/hostname プリンシパルは自動的に UID 0 に対応付けられるため、gsscred テーブルは使用されません。したがっ

て、gsscred テーブルを使用して root を特別な UID に対応づけることはできません。

gsscred テーブル用メカニズムの選択

gsscred テーブルに対しどのメカニズムを選択するかは、次の要素で決まります。

- 検索時間の短縮に関心があるか
- データアクセスのセキュリティ向上に関心があるか
- ファイルを短時間で作成する必要があるか

次に、選択可能なバックエンドメカニズムとその利点を説明します。

files gsscred テーブルはファイルシステムに格納されます。テーブルが作成された後はネットワークを介した伝送は行われないため、共有されないローカルファイルシステムが最も安全なバックエンドです。このタイプのファイルが最も短時間で作成されます。

xfn_files gsscred テーブルは /var/fn ファイルシステムに格納されます。このファイルシステムは共有されていても、されていなくてもかまいません。xfn ファイルはどれも作成に長い時間がかかります。

xfn_nis gsscred テーブルは NIS 名前空間に格納されます。このファイルシステムの検索は安全ではありません。xfn ファイルはどれも作成に長い時間がかかります。

xfn_nisplus gsscred テーブルは NIS+ 名前空間に格納されます。このファイルシステムの検索は安全ではありません。xfn ファイルはどれも作成に長時間かかります。

xfn gsscred テーブルは xfn のデフォルトシステムに格納されます。xfn ファイルはどれも作成に長時間かかります。

files バックエンドメカニズムでは、最初の検索が遅いことがあります。他のメカニズムでは、最初の検索はネームサービスを使用してこれより速く行われます。どのメカニズムでも、データがキャッシュされたあとの検索時間はほぼ同じです。

第 24 章

自動セキュリティ拡張ツールの使用手順

この章では、自動セキュリティ拡張ツール (ASET) を使用して、システムファイルおよびディレクトリへのアクセスを監視または制限する方法について説明します。

この章で説明する手順は次のとおりです。

自動セキュリティ拡張ツール (ASET)

SunOS 5.8 システムソフトウェアには、自動セキュリティ拡張ツール (ASET) が組み込まれています。ASET を使用すると、他の場合には手作業で実行する作業が自動的に実行され、システムのセキュリティを監視して制御できます。

ASET セキュリティパッケージには、システムのセキュリティを制御して監視できるように、自動管理ツールが組み込まれています。ASET を実行するとセキュリティレベルとして、低、中、または高レベルを指定できます。上のレベルほど、ASET のファイル制御機能が増え、ファイルアクセスが減少し、システムセキュリティが厳しくなります。

ASET に関連して 7 つのタスクがあり、それぞれがシステムファイルに対して特定のチェックと調整を行います。ASET のタスクはファイルのアクセス権を厳格にし、重要なシステムファイルの内容にセキュリティ上の弱点がないかどうかをチェック

し、重要な領域を監視します。ASET は、ゲートウェイシステムとして機能するシステムにファイアウォールシステムの基本要件を適用し、ネットワークを保護できます (詳細は、458ページの「ファイアウォールの設定」を参照してください)。

ASET は、構成用のマスターファイルを使用します。マスターファイルやレポートなどの ASET ファイルは、ディレクトリ `/usr/aset` にあります。これらのファイルは、サイトの特定の要件に合わせて変更できます。

各タスクは、検出されたセキュリティ上の弱点と、システムファイルに対して行なった変更を示すレポートを生成します。上位のセキュリティレベルで実行すると、ASET はシステムセキュリティ上の弱点をすべて変更しようとします。潜在的なセキュリティ問題を解決できなければ、ASET は問題の存在を報告します。

`/usr/aset` コマンドを対話的に実行すると、ASET セッションを開始できます。また、`crontab` ファイルにエントリを追加すると、ASET を定期的に実行するように設定できます。

ASET のタスクはディスクをかなり使用するため、通常の活動の妨げになることがあります。システム性能に及ぼす影響を最小限度に抑えるために、24 時間ごとまたは 48 時間ごとに深夜など、システムの稼働レベルが最も低いときに ASET を実行するようにスケジュールしてください。

ASET のセキュリティレベル

ASET は、低、中、高の 3 つのセキュリティレベルのいずれかで動作するように設定できます。上のレベルほど、ASET のファイル制御機能が増え、ファイルアクセスが減少し、システムのセキュリティが厳しくなります。これらの機能には、ユーザーによるファイルアクセスを制限せずにシステムセキュリティを監視する最低レベルから、システムが完全にセキュリティ保護される最高レベルまで、アクセス権が段階的に厳格になります。

次に、この 3 つのセキュリティレベルについて説明します。

セキュリティレベル	このレベル
低セキュリティ	ファイルシステムの属性が標準リリース値に設定されることが保証されます。ASET は複数のチェックを実行し、セキュリティ上の潜在的な弱点を報告します。このレベルでは、ASET は動作せず、システムサービスは影響を受けません。
中セキュリティ	ほとんどの環境で十分にセキュリティが制御されます。ASET はシステムファイルとパラメタの設定の一部を変更し、システムアクセスを制限し、セキュリティ上の攻撃によるリスクを減少させます。ASET は、セキュリティ上の弱点と、アクセスを制限するために行なった変更を報告します。このレベルでは、ASET はシステムサービスに影響しません。
高セキュリティ	システムに高度なセキュリティが適用されます。ASET は多数のシステムファイルとパラメタの設定を調整して、アクセス権を最小限度に抑えます。ほとんどのシステムアプリケーションとコマンドは引き続き正常に機能しますが、このレベルではシステム動作よりもセキュリティ上の検討事項が優先されます。

注 - セキュリティレベルを下げるか、システムを ASET 実行前の設定に意図的に戻さなければ、ASET によってファイルのアクセス権が緩められることはありません。

ASET のタスク

この節では、ASET のタスクについて説明します。レポートを解釈して活用するには、各 ASET のタスク、つまり、その目的、実行される処理、および影響を受けるシステム構成要素を理解しておく必要があります。

ASET のレポートファイルには、各 ASET タスクで検出された問題をできるだけ詳細に記述するメッセージが入っています。これらのメッセージを調べると、問題を診断して解決できます。ただし、ASET を活用するには、システム管理とシステム構成要素を全般的に理解していることが前提となります。管理者になったばかりの方は、他の SunOS 5.8 システム管理マニュアルと関連するマニュアルページを参照して、ASET の管理の概要を把握してください。

`taskstat` ユーティリティは、完了したタスクとまだ実行中のタスクを識別します。完了したタスクごとにレポートファイルが生成されます。`taskstat` ユーティリティの詳細は、`taskstat(1M)` のマニュアルページを参照してください。

システムファイルのアクセス権の確認

このタスクでは、システムファイルのアクセス権が指定したセキュリティレベルに設定されます。このタスクは、システムをインストールするときに実行されます。以前に設定したレベルを後から変更したい場合は、このタスクをもう一度実行してください。低セキュリティレベルでは、アクセス権は開放型の情報共有環境に適した値に設定されています。中セキュリティレベルでは、アクセス権はほとんどの環境に十分なセキュリティが適用される程度に厳格です。高セキュリティレベルでは、アクセスが厳しく制限されます。

このタスクによってシステムファイルのアクセス権やパラメタの設定に加えられた変更は、`tune.rpt` ファイル内でレポートされます。アクセス権を設定するときに ASET が参照するファイルの例については、472ページの「調整ファイル」を参照してください。

システムファイルのチェック

このタスクでは、システムファイルが検査され、マスターファイル内にリストされたファイルの記述と比較されます。マスターファイルは、ASET がこのタスクを実行すると初めて作成されます。マスターファイルには、指定したセキュリティレベル `checklist` によって適用されるシステムファイル設定が入っています。

ファイルがチェックされるディレクトリのリストは、セキュリティレベルごとに定義されます。デフォルトのリストを使用するか、レベルごとに異なるディレクトリを指定して変更できます。

ファイルごとに次の基準がチェックされます。

- 所有者とグループ
- アクセス権ビット
- サイズとチェックサム
- リンク数
- 最終変更時刻

矛盾が見つかると、`cklist.rpt` ファイル内でレポートされます。このファイルには、システムファイルのサイズ、アクセス権、チェックサムの値、およびマスターファイルと比較した結果が入っています。

ユーザーとグループのチェック

このタスクでは、passwd ファイルと group ファイル内で定義されているユーザーアカウントとグループの整合性と完全性がチェックされます。ローカルパスワードファイルと、NIS または NIS+ パスワードファイルがチェックされます。NIS+ パスワードファイルの問題はレポートされますが、解決されません。このタスクでは、次の違反がチェックされます。

- 重複する名前または ID
- 不正な形式によるエントリ
- パスワードが付いていないアカウント
- 無効なログインディレクトリ
- アカウント nobody
- 空のグループパスワード
- NIS (または NIS+) サーバー上の /etc/passwd ファイル内のプラス記号 (+)

矛盾は usrgrp.rpt ファイル内でレポートされます。

システム構成ファイルのチェック

このタスクの実行中に、ASET は各種システムテーブルをチェックしますが、そのほとんどは /etc ディレクトリに入っています。次のファイルがチェックされます。

- /etc/default/login
- /etc/hosts.equiv
- /etc/inetd.conf
- /etc/aliases
- /var/adm/utmpx
- /.rhosts
- /etc/vfstab
- /etc/dfs/dfstab
- /etc/ftpusers

ASET は、これらのファイルに関して各種のチェックと変更を実行し、すべての問題を sysconf.rpt ファイル内でレポートします。

環境のチェック

このタスクでは、root 用とその他ユーザー用の PATH 環境変数と UMASK 環境変数が /.profile、/.login、/.cshrc ファイル内でどのように設定されているかがチェックされます。

環境のセキュリティ状況をチェックした結果は、env.rpt ファイル内でレポートされます。

eeprom のチェック

このタスクでは、eeprom セキュリティパラメタの値がチェックされ、適切なセキュリティレベルに設定されているかどうかが確認されます。eeprom セキュリティパラメタは、none、command、または full に設定できます。

ASET はこの設定を変更しませんが、推奨値を eeprom.rpt ファイル内でレポートします。

ファイアウォールの設定

このタスクでは、システムをネットワークリレーとして安全に使用できることが保証されます。314ページの「ファイアウォールシステム」で説明したように、ファイアウォール専用システムが設定され、内部ネットワークが外部の公共ネットワークから保護されます。ファイアウォールシステムは、相互に信頼されない (untrusted) システムとしてアクセスし合う 2 つのネットワークを分離します。ハードウェアの設定作業によって、インターネットプロトコル (IP) パケットを転送できなくなり、ルーティング情報は外部ネットワークから隠されます。

ファイアウォールのタスクはすべてのセキュリティレベルで実行されますが、ファイアウォールとしての本来の機能は最上位レベルでのみ動作します。ASET を高セキュリティレベルで実行したいが、システムにはファイアウォール保護が不要であることがわかった場合は、asetenv ファイルを編集してファイアウォールタスクを除去できます。

行われた変更はすべて firewall.rpt ファイル内にレポートされます。

ASET 実行ログ

ASET を対話形式またはバックグラウンドで実行すると、実行ログが生成されます。デフォルトでは、ASET はログファイルを標準出力に生成します。実行ログ

は、ASET が指定された時刻に実行されたことを確認するもので、実行エラーメッセージも入っています。aset -n コマンドを使用すると、ログを指定したユーザーに電子メールで配信できます。ASET オプションのリストについては、aset(1M) のマニュアルページを参照してください。

実行ログファイルの例

```
ASET running at security level low

Machine=example; Current time = 0325_08:00


aset: Using /usr/aset as working directory

Executing task list...
        firewall
        env
        sysconfig
        usrgrp
        tune
        cklist
        eeprom
All tasks executed. Some background tasks may still be running.

Run /usr/aset/util/taskstat to check their status:
     $/usr/aset/util/taskstat     aset_dir
Where aset_dir is ASET's operating directory, currently=/usr/aset

When the tasks complete, the reports can be found in:
     /usr/aset/reports/latest/*.rpt
You can view them by:
more /usr/aset/reports/latest/*.rpt
```

第 1 のログは、ASET が実行されたシステムと時刻を示します。その後に、開始された各タスクがリストされています。

455ページの「ASET のタスク」で説明しているように、ASET はこれらのタスクごとにバックグラウンドプロセスを呼び出します。タスクは開始されると実行ログにリストされますが、これはタスクの完了を示しているわけではありません。バックグラウンドタスクの状態をチェックするには、taskstat ユーティリティを使用します。

ASET レポート

ASET タスクから生成されたすべてのレポートファイルは、ディレクトリ /usr/aset/reports の下のサブディレクトリに入っています。この節では、/usr/aset/reports ディレクトリの構造と、レポートファイルを管理するためのガイドラインについて説明します。

ASET はレポートファイルを指定されたサブディレクトリに格納し、レポートの生成日時を反映させます。このため、ASET を実行するたびに変化するシステムの状態を示すレコードを順番に追跡できます。これらのレポートを監視し、比較して、システムセキュリティの状況を判断できます。

図 24–1 に reports ディレクトリ構造の例を示します。

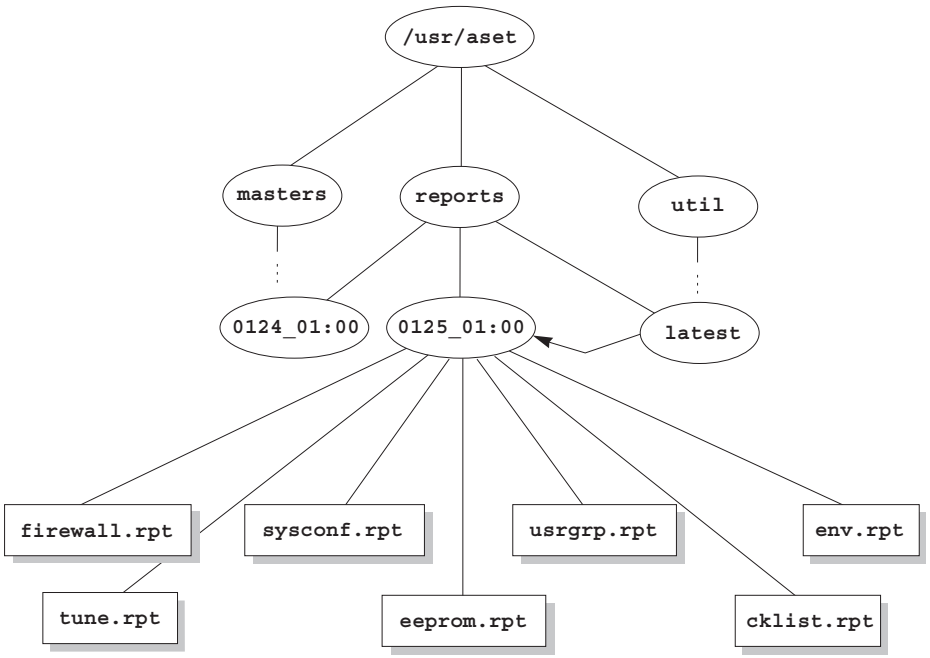


図 24–1　ASET reports ディレクトリ構造

この例は、2 つのレポートサブディレクトリを示しています。

- 0124_01:00
- 0125_01:00

サブディレクトリ名は、レポートの生成日時を示します。各レポートサブディレクトリ名の形式は次のとおりです。

monthdate_hour:minute

この場合、「month」、「date」、「hour」、「minute」は、いずれも 2 桁の数値です。たとえば、0125_01:00 は 1 月 25 日の午前 1 時を表します。

2 つのレポートディレクトリには、それぞれ ASET を 1 度実行した結果、生成されたレポートの集合が入っています。

latest ディレクトリは、常に最新レポートが入っているサブディレクトリを指すシンボリックリンクです。したがって、ディレクトリ /usr/aset/reports/latest に移動すれば、ASET で生成された最新レポートを調べることができます。このディレクトリには、前回の ASET で実行された各タスクのレポートファイルが入っています。

ASET レポートファイルの形式

各レポートファイルは、それを生成したタスクから取った名前が付けられます。表 24–1 にタスクとそのレポートのリストを示します。

表 24–1 ASET のタスクと生成されるレポート

タスク	レポート
システムファイルのアクセス権の調整 (tune)	tune.rpt
システムファイルチェックリスト (cklist)	cklist.rpt
ユーザー/グループのチェック (usrgrp)	usrgrp.rpt
システム構成ファイルのチェック (sysconf)	sysconf.rpt
環境チェック (env)	env.rpt
eeprom チェック (eeprom)	eeprom.rpt
ファイアウォールの設定 (firewall)	firewall.rpt

各レポートファイル内で、メッセージの前後はバナー行で囲まれています。ASET の構成要素を誤って削除したり損傷したりした場合など、タスクが途中で終了することがあります。ほとんどの場合、レポートファイルの末尾の方に、途中で終了した原因を示すメッセージが入っています。

次にサンプルレポートファイル usrgrp.rpt を示します。

```
*** Begin User and Group Checking ***

Checking /etc/passwd ...
Warning! Password file, line 10, no passwd
:sync::1:1::/:/bin/sync
..end user check; starting group check ...
Checking /etc/group...
*** End User And group Checking ***
```

ASET レポートファイルの検査

最初に ASET を実行するか構成し直したら、レポートファイルを詳しく検査する必要があります (構成し直す作業には、サブディレクトリ masters 内の asetenv ファイルやマスターファイルの変更や、ASET が動作するセキュリティレベルの変更が含まれます)。レポートには、構成し直したために発生したエラーが記録されます。レポートを詳しく検査すると、問題が発生した時点で対処して解決できます。

ASET レポートファイルの比較

構成上の変更やシステム更新がない期間中にレポートファイルを監視すると、レポートの内容が安定状態になり、予期しない情報が入っていてもわずかであることがわかります。diff ユーティリティを使用して、レポートを比較できます。

ASET マスターファイル

ASET のマスターファイル tune.high、tune.low、tune.med、および uid_aliases は、ディレクトリ /usr/aset/masters に入っています。ASET は、マスターファイルを使用してセキュリティレベルを定義します。

調整ファイル

tune.low、tune.med、tune.high マスターファイルでは、利用できる ASET セキュリティレベルが定義されます。各ファイルでは、各レベルのシステムファイルの属性が指定され、比較と参照に使用されます。

uid_aliases ファイル

uid_aliases ファイルには、同じ ID を共有する複数のユーザーアカウントのリストが入っています。この種のアカウントがあると責任の所在があいまいになるので、通常は ASET が警告を出します。uid_aliases ファイル内で例外をリストすると、この規則に例外を設けることができます。重複するユーザー ID を持つ passwd ファイル内のエントリを uid_aliases ファイル内で指定しておくと、これらのエントリは ASET でレポートされません。

複数のユーザーアカウント (パスワードエントリ) に同じユーザー ID を共有させないでください。他の方法で目的を達成することを検討する必要があります。たとえば、複数のユーザーにアクセス権一式を共有させたい場合は、グループアカウントを作成できます。ユーザー ID の共有は最後の手段であり、どうしても必要で、他の方法では目的を達成できない場合にだけに使用します。

環境変数 UID_ALIASES を使用すると、別の別名ファイルを指定できます。デフォルトは /usr/aset/masters/uid_aliases です。

チェックリストファイル

システムファイルのチェックリストに使用されるマスターファイルは、初めて ASET を実行するときか、セキュリティレベルの変更後に ASET を実行するときに生成されます。

このタスクでチェックされるファイルは、次の環境変数で定義されます。

- CKLISTPATH_LOW
- CKLISTPATH_MED
- CKLISTPATH_HIGH

ASET 環境ファイル (asetenv)

環境ファイル asetenv には、ASET タスクに影響する変数のリストが入っています。各変数を変更すると ASET の動作を変更できます。

ASET の構成

この節では、ASET を構成する方法とその処理の基礎となる環境について説明します。

ASET の管理と構成は最小限度ですみ、ほとんどの場合はデフォルト値で実行できます。ただし、ASET の処理や動作に影響する一部のパラメタを調整して、利点を最大限に発揮させることができます。デフォルト値を変更する前に、ASET の機能と、システムの構成要素に及ぼす影響を理解しておく必要があります。

ASET は、次の 4 つの構成ファイルに依存してタスクの動作を制御します。

- /usr/aset/asetenv
- /usr/aset/masters/tune.low
- /usr/aset/masters/tune.med
- /usr/aset/masters/tune.high

環境ファイルの変更 (asetenv)

/usr/aset/asetenv ファイルは、次の 2 つの主要セクションに分かれています。

- ユーザーが構成できるパラメタセクション
- 内部環境変数セクション

ユーザーが構成できるパラメタセクションは変更できます。しかし、内部環境変数セクションの設定は内部使用専用で変更できません。

ユーザーが構成できるパラメタセクション内のエントリを編集して、次の作業を実行できます。

- 実行するタスクを選択する
- チェックリストタスク用のディレクトリを指定する
- ASET の実行スケジュールを指定する
- 別名ファイルを指定する
- チェック対象を NIS+ テーブルまで拡張する

実行するタスクの選択：TASKS

ASET が実行する各タスクでは、システムセキュリティの特定の領域が監視されます。ほとんどのシステム環境では、すべてのタスクでバランスがとれたセキュリティ範囲を提供する必要があります。ただし、1 つ以上のタスクを除外してもかまいません。

たとえば、ファイアウォールタスクはすべてのセキュリティレベルで実行されますが、本来の機能は最上位レベルでのみ動作します。ASET を高セキュリティレベルで実行したいが、ファイアウォール保護は不要な場合があります。

asetenv ファイル内で環境変数の TASKS リストを編集すると、ファイアウォール機能を使用しないで高レベルで実行するように ASET を設定できます。デフォルトでは、TASKS リストにはすべての ASET タスクが入っています (次の例を参照してください)。タスクを削除するには、そのタスクの設定をファイルから削除します。この場合は、リストから firewall 環境変数を削除することになります。次回に ASET を実行すると、除外したタスクは実行されません。

```
TASKS=''env sysconfig usrgrp tune cklist eeprom firewall''
```

チェックリストタスク用のディレクトリの指定： CKLISTPATH

システムファイルチェックでは、選択したシステムディレクトリ内のファイルの属性がチェックされます。次のチェックリストパス環境変数を使用して、どのディレクトリをチェックするかを定義できます。

- CKLISTPATH_LOW
- CKLISTPATH_MED
- CKLISTPATH_HIGH

CKLISTPATH_LOW 変数は、低セキュリティレベルでチェックされるディレクトリを定義します。CKLISTPATH_MED と CKLISTPATH_HIGH 環境変数は、中程度と高度のセキュリティレベルに同じように機能します。

低セキュリティレベルの変数で定義したディレクトリリストは、1 つ上位レベルで定義するディレクトリリストのサブセットにする必要があります。たとえば、CKLISTPATH_LOW に定義したすべてのディレクトリを CKLISTPATH_MED に含め、CKLISTPATH_MED に指定したすべてのディレクトリを CKLISTPATH_HIGH に含めます。

これらのディレクトリに対して実行されるチェックは再帰的ではありません。ASET は変数内に明示的にリストされたディレクトリのみをチェックします。そのサブディレクトリはチェックされません。

これらの変数の定義を編集して、ASET にチェックさせたいディレクトリを追加または削除できます。これらのチェックリストは、一般に毎日変化しないシステムファイ

ルにのみ有効なので注意してください。たとえば、ユーザーのホームディレクトリは動的な変化が大きすぎるので、チェックリストの候補にはならないのが普通です。

ASET の実行スケジュールの指定： PERIODIC_SCHEDULE

ASET を起動するときには、対話形式で起動する方法と、–p オプションを使用して ASET タスクをスケジュール指定した時刻と期間に実行する方法があります。ASET は、システム需要が少ないときに定期的に実行できます。たとえば、ASET は PERIODIC_SCHEDULE を照会して、ASET タスクの実行頻度と実行時刻を判断します。ASET を定期的に実行するように設定する方法については、476ページの「ASET を定期的に実行する方法」を参照してください。

PERIODIC_SCHEDULE の形式は、crontab エントリの形式と同じです。詳細は、crontab(1) のマニュアルページを参照してください。

別名ファイルの指定： UID_ALIASES

UID_ALIASES 変数は、共有ユーザー ID がリストされる別名ファイルを指定します。 デフォルトは /usr/aset/masters/uid_aliases です。

チェック範囲を NIS+ テーブルまで拡張する： YPCHECK

YPCHECK 環境変数は、ASET でシステム構成ファイルテーブルもチェックするかどうかを指定します。YPCHECK はブール型変数なので、true または false しか指定できません。デフォルト値は false で、NIS+ テーブルのチェックは無効になっています。

この変数の機能を理解するために、passwd ファイルに与える影響を考えてみてください。この変数を false に設定すると、ASET はローカルの passwd ファイルをチェックします。true に設定すると、NIS+ の passwd ファイル内でシステムのドメインもチェックされます。

注 - ASET ではローカルテーブルが自動的に修復されますが、NIS+ テーブル内の潜在的な問題はレポートされるだけで変更されません。

調整ファイルの変更

ASET は、3 つのマスター調整ファイル、tune.low、tune.med、tune.high を使用して、重要なシステムファイルへのアクセス制限を緩めたり厳しくしたりします。この 3 つのマスターファイルは /usr/aset/masters ディレクトリに入っており、環境に合わせて調整できます。詳細は、472ページの「調整ファイル」を参照してください。

tune.low ファイルは、アクセス権をデフォルトのシステム設定に適した値に設定します。tune.med ファイルは、これらのアクセス権をさらに制限し、tune.low に含まれていないエントリを追加します。tune.high ファイルは、アクセス権をさらに厳しく制限します。

注 - 調整ファイル内の設定を変更するには、ファイルのエントリを追加または削除します。アクセス権を現在の設定よりも制限が緩やかになるような値に設定できません。システムセキュリティを下位レベルに下げない限り、ASET はアクセス権の制限を緩和しません。

ASET で変更されたシステムファイルの復元

ASET を初めて実行すると、元のシステムファイルが保存され保管されます。aset.restore ユーティリティは、これらのファイルを復元します。また、ASET を定期的に実行するようにスケジュール指定している場合は、そのスケジュールを解除します。aset.restore ユーティリティは、ASET の動作ディレクトリ /usr/aset に入っています。

システムファイルに対して行われた変更は、aset.restore を実行すると失われます。

次の場合に aset.restore を使用してください。

- ASET の変更を削除して元のシステムを復元したい場合。ASET を永久に無効にしたい場合は、以前に root の crontab に aset コマンドが追加されていれば、cron スケジュールから削除できます。cron を使用して自動実行を削除する方法については、477ページの「ASET の定期的な実行を中止する方法」を参照してください。
- ASET を短期間実行した後に、元のシステム状態を復元する場合
- 一部の主要なシステム機能が正常に動作せず、ASET が原因だと思われる場合

NFS システムを使用するネットワーク操作

通常、ネットワークの一部となっているシステム上でも、ASET はスタンドアロンモードで使用されます。スタンドアロンシステムのシステム管理者は、システムのセキュリティとシステムを保護する ASET の実行と管理を担当することになります。

また、ASET は NFS 分散環境でも使用できます。ネットワーク管理者は、すべてのクライアントの各種管理タスクのインストール、実行、管理を担当します。複数のクライアントシステム間で ASET を管理しやすくするために、構成変更を行なってすべてのクライアントに一括して適用すれば、各システムにログインしてプロセスを繰り返す必要がなくなります。

ネットワークシステム上で ASET の設定方法を決めるときには、ユーザーに各自のシステム上でセキュリティをどのように制御させるかと、セキュリティ制御に関する責任をどの程度集中させるかを検討する必要があります。

各セキュリティレベルの一括構成の提供

複数のネットワーク構成を設定したい場合があります。たとえば、低セキュリティレベルに指定したクライアント用に 1 つ、中レベルのクライアント用に 1 つ、さらに高レベルのクライアント用に 1 つというように設定できます。

セキュリティレベルごとに別の ASET ネットワーク構成を作成したい場合は、サーバー上でレベルごとに 1 つずつ合計 3 つの ASET 構成を作成できます。各構成を該当するセキュリティレベルのクライアントにエクスポートすることになります。3 つの構成すべてに共通の ASET 構成要素は、リンクを使用して共有できます。

ASET レポートの収集

スーパーユーザー特権を持つか持たないかに関係なく、クライアントにアクセスされるサーバー上に ASET 構成要素を集中できるだけでなく、サーバー上でディレクトリを設定して、各種クライアント上で実行中のタスクによって生成されるすべてのレポートを収集できます。収集機構を設定する方法については、477ページの「サーバー上で ASET レポートを収集する方法」を参照してください。

サーバー上でレポートを収集するように設定すると、すべてのクライアントに関するレポートを 1 箇所で検討できます。この方法は、クライアントがスーパーユーザー特権を持っているかどうかに関係なく使用できます。また、ユーザーに各自の ASET レポートを監視させたい場合は、ローカルシステム上にレポートディレクトリを残しておいてもかまいません。

ASET 環境変数

表 24–2 に ASET 環境変数と各変数で指定する値を示します。

表 24–2 ASET 環境変数とその意味

環境変数	指定内容
ASETDIR (以下を参照)	ASET の作業ディレクトリ
ASETSECLEVEL (以下を参照)	セキュリティレベル
PERIOD_SCHEDULE	定期的なスケジュール
TASKS	実行するタスク
UID_ALIASES	別名ファイル
YPCHECK	チェックを NIS と NIS+ まで拡張する
CKLISTPATH_LOW	低セキュリティ用のディレクトリリスト
CKLISTPATH_MED	中セキュリティ用のディレクトリリスト
CKLISTPATH_HIGH	高セキュリティ用のディレクトリリスト

次に示す環境変数は、ファイル /usr/aset/asetenv に入っています。ASETDIR 変数と ASETSECLEVEL 変数はオプションで、aset コマンドを使用してシェルからでなければ設定できません。他の環境変数は、ファイルを編集して設定できます。次に、各変数について説明します。

ASETDIR 変数

ASETDIR は、ASET の作業ディレクトリを指定します。

C シェルから次のように入力します。

```
% setenv ASETDIR pathname
```

Bourne シェルまたは Korn シェルからは、次のように入力します。

```
$ ASETDIR=pathname
$ export ASETDIR
```

pathname を ASET 作業ディレクトリの完全パス名に設定してください。

ASETSECLEVEL 変数

ASETSECLEVEL は、ASET タスクが実行されるセキュリティレベルを指定します。

C シェルから次のように入力します。

```
% setenv ASETSECLEVEL level
```

Bourne シェルまたは Korn シェルから、次のように入力します。

```
$ ASETSECLEVEL=level
export ASETSECLEVEL
```

上記のコマンドで、*level* を次のいずれかに設定できます。

low	低セキュリティレベル
med	中セキュリティレベル
high	高セキュリティレベル

PERIODIC_SCHEDULE 変数

PERIODIC_SCHEDULE の値の形式は、crontab ファイルと同じです。変数の値は二重引用符で囲んだ 5 つのフィールドからなる文字列として指定します。各フィールドは空白文字 1 つで区切ってください。

```
"minutes  hours  day-of-month  month  day-of-week"
```

表 24–3 Periodic_Schedule 変数の値

変数	値
minutes hours	開始時刻を分 (0-59) と時間 (0-23) で指定します。
day-of-month	ASET を実行する日付を 1 から 31 までの値で指定します。
month	ASET を実行する月を 1 から 12 までの値で指定します。
day-of-week	ASET を実行する曜日を 0 から 6 までの値で指定します。この方式では、日曜日の値は 0 です。

次の規則が適用されます。

- どのフィールドでも、値のリストをコンマで区切って指定できます。
- 値を数値または範囲として指定できます。範囲とは、1 対の数値をハイフンで結合したものです。範囲は、範囲に含まれるすべての時刻に ASET タスクを実行することを示します。
- どのフィールドでも、値としてアスタリスク (*) を指定できます。アスタリスクは、そのフィールドに有効なすべての値を指定します。

PERIODIC_SCHEDULE 変数のデフォルトエントリでは、ASET が毎日午前 12:00 に実行されます。

```
PERIODIC_SCHEDULE=''0 0 * * *''
```

TASKS 変数

TASKS 変数は、ASET で実行されるタスクをリストします。デフォルトでは、7 つのタスクがすべてリストされます。

```
TASKS=''env sysconfig usrgrp tune cklist eeprom firewall''
```

UID_ALIASES 変数

UID_ALIASES 変数は、別名ファイルを指定します。別名ファイルがあると、ASET は使用可能な複数の別名のリストをこのファイル内で照会します。形式は UID_ALIASES=*pathname* です。*pathname* は、別名ファイルの完全パス名です。

デフォルトは次のとおりです。

```
UID_ALIASES=${ASETDIR}/masters/uid_aliases
```

YPCHECK 変数

YPCHECK 変数は、システムテーブルをチェックするタスクを拡張して NIS または NIS+ テーブルを含めます。これはブール変数なので、true または false に設定できます。

デフォルトは false で、ローカルシステムテーブルがチェックされます。

```
YPCHECK=false
```

CKLISTPATH_level 変数

3 つのチェックリストパス変数は、チェックリストタスクでチェックされるディレクトリをリストします。次の変数定義はデフォルトで設定されていて、各種レベルの変数の関係を示しています。

```
CKLISTPATH_LOW=${ASETDIR}/tasks:${ASETDIR}/util:${ASETDIR}/masters:
/etc
CKLISTPATH_MED=${CKLISTPATH_LOW}:/usr/bin:/usr/ucb
CKLISTPATH_HIGH=${CKLISTPATH_MED}:/usr/lib:/sbin:/usr/sbin:/usr/ucblib
```

チェックリストパス環境変数の値は、シェルパス変数の値と同様で、ディレクトリ名がコロン (:) で区切られたリストです。等号 (=) を使用すると、変数名にその値を設定できます。

ASET ファイルの例

この節では、調整ファイルや別名ファイルなど、ASET ファイルの例を示します。

調整ファイル

ASET は 3 つの調整ファイルを管理します。3 つの調整ファイル内のエントリについては、表 24–4 で説明しています。

表 24–4 調整ファイルのエントリ形式

エントリ	説明
pathname	ファイルのフルパス名
mode	アクセス権の設定を表す 5 桁の数値
owner	ファイルの所有者
group	ファイルのグループ
type	ファイルの形式

次の規則が適用されます。

- パス名には、アスタリスク (*) や疑問符 (?) など、通常のシェルワイルドカード文字を使用して、複数のエントリを指定できます。`sh(1)` のマニュアルページを参照してください。
- *mode* は、最も制限が緩やかな値を表します。現在の設定が指定した値よりもすでに厳密な制限を表している場合、ASET はアクセス権の設定を緩和しません。たとえば、指定した値が 00777 の場合、00777 は常に現在の設定よりも緩やかな制限を表すので、アクセス権は変更されません。

  セキュリティレベルを下げるか、ASET を削除するのでない限り、ASET ではこの方法でモード設定が処理されます。セキュリティレベルを前回の実行時よりも下げるときや、システムファイルを ASET を最初に実行する前の状態に復元したいときには、ASET は操作の内容を認識して保護レベルを下げます。
- *owner* と *group* には、数値 ID ではなく名前を使用しなければなりません。
- *owner*、*group*、*type* の代わりに疑問符 (?) を使用すると、ASET によってこれらのパラメタの既存の値が変更されるのを防止できます。
- *type* には、symlink (シンボリックリンク)、directory、または file (他のすべて) を指定できます。
- セキュリティレベルが高くなるほど、調整ファイルは下位レベルよりも緩やかなファイルアクセス権にリセットされます。また、上位レベルになるほど、リストに多数のファイルが追加されます。
- 1 つのファイルで複数の調整ファイルエントリを照合できます。たとえば、`etc/passwd` は `etc/pass*` エントリと `/etc/*` エントリに一致します。

- 2 つのエントリのアクセス権が異なる場合は、ファイルアクセス権は最も厳しいアクセス権を表す値に設定されます。次の例では、/etc/passwd のアクセス権は 00755 に設定されますが、これは 00755 は 00770 よりも厳密な制限であることを表します。

```
/etc/pass*    00755   ? ?  file
/etc/*        00770   ? ?  file
```

- 2 つのエントリの *owner* 指定または *group* 指定が異なる場合は、最後のエントリが優先されます。次の例は、tune.low ファイルの最初の数行を示します。

```
/ 02755 root root directory
/bin 00777 root bin symlink
/sbin 02775 root sys directory
/usr/sbin 02775 root bin directory
/etc 02755 root sys directory
/etc/chroot 00777 bin bin symlink
```

別名ファイル

別名ファイルには、同じユーザー ID を共有する別名のリストが入っています。

各エントリの書式は次のとおりです。

uid=alias1=alias2=alias3=...

uid	共有ユーザー ID
aliasn	ユーザー ID を共有するユーザーアカウント

たとえば、次のエントリでは、sysadm と root に共有されるユーザー ID 0 を示しています。

```
0=root=sysadm
```

ASET の実行

この節では、ASET を対話的にまたは定期的に実行する方法について説明します。

▼ ASET を対話的に実行する方法

1. スーパーユーザーになります。

2. aset コマンドを使用して **ASET** を対話的に実行します。

```
# /usr/aset/aset -l level -d pathname
```

level	セキュリティレベルを指定する。有効な値は low、medium、または high。デフォルト設定は low。セキュリティレベルについては、454ページの「ASET のセキュリティレベル」を参照
pathname	ASET の作業ディレクトリを指定する。デフォルトは /usr/aset

3. 画面に表示される **ASET** 実行ログを見て、**ASET** が動作していることを確認します。

   実行ログメッセージは、動作しているタスクを示します。

例 — ASET を対話的に実行する

次の例では、デフォルトの作業ディレクトリを使用して低セキュリティレベルで ASET を実行します。

```
# /usr/aset/aset -l low
======= ASET Execution Log =======

ASET running at security level low

Machine = jupiter; Current time = 0111_09:26

aset: Using /usr/aset as working directory

Executing task list ...
 firewall
 env
 sysconf
 usrgrp
 tune
 cklist
 eeprom

All tasks executed. Some background tasks may still be running.
```

(続く)

```
Run /usr/aset/util/taskstat to check their status:
 /usr/aset/util/taskstat [aset_dir]

where aset_dir is ASET's operating
 directory,currently=/usr/aset.

When the tasks complete, the reports can be found in:
 /usr/aset/reports/latest/*.rpt

You can view them by:
 more /usr/aset/reports/latest/*.rpt
```

▼ ASET を定期的に実行する方法

1. スーパーユーザーになります。

2. 必要であれば、**ASET** を定期的に実行したい時刻を設定します。

   システム需要が少ないときに ASET を実行してください。/usr/aset/asetenv ファイル内の PERIODIC_SCHEDULE 環境変数を使用して、ASET を定期的に実行する時刻を設定します。デフォルトでは、この時刻は 24 時間ごとに真夜中に設定されています。

   別の時刻を設定したい場合は、/usr/aset/asetenv ファイル内で PERIODIC_SCHEDULE 変数を編集します。PERIODIC_SCHEDULE 変数の設定の詳細は、470ページの「PERIODIC_SCHEDULE 変数」を参照してください。

3. aset コマンドを使ってエントリを crontab ファイルに追加します。

```
# /usr/aset/aset -p
```

   -p　　/usr/aset/asetenv ファイル内の PERIODIC_SCHEDULE 環境変数で決めた時刻に ASET の実行を開始する行を crontab ファイルに挿入する

4. 次のコマンドを実行すると crontab エントリが表示され、**ASET** の実行スケジュールを確認できます。

```
# crontab -l root
```

▼ ASET の定期的な実行を中止する方法

1. スーパーユーザーになります。
2. crontab ファイルを編集します。

```
# crontab -e root
```

3. **ASET** エントリを削除します。
4. 変更結果を保存して終了します。
5. crontab エントリを表示して、**ASET** エントリが削除されていることを確認します。

```
# crontab -l root
```

▼ サーバー上で ASET レポートを収集する方法

1. スーパーユーザーになります。
2. サーバー上でディレクトリを作成します。
   a. /usr/aset ディレクトリに移動します。

```
mars# cd /usr/aset
```

   b. ***rptdir*** ディレクトリを作成します。

```
mars# mkdir rptdir
```

   c. ***rptdir*** ディレクトリに移動して、***client_rpt*** ディレクトリを作成します。

```
mars# cd rptdir
mars# mkdir client_rpt
```

d. このコマンドによって、クライアント用のサブディレクトリ **(*client_rpt*)** が作成されます。レポートを収集したいクライアントごとに、この手順を繰り返します。

次の例では、ディレクトリ `all_reports` とサブディレクトリ `pluto_rpt` と `neptune_rpt` が作成されます。

```
mars# cd /usr/aset
mars# mkdir all_reports
mars# cd all_reports
mars# mkdir pluto_rpt
mars# mkdir neptune_rpt
```

3. ***client_rpt*** ディレクトリを `/etc/dfs/dfstab` ファイルに追加します。

このディレクトリには、読み取りまたは書き込みオプションがあります。

たとえば、`dfstab` 内の次のエントリは、読み取り／書き込み権によって共有されます。

```
mars# cd /usr/aset
mars# mkdir all_reports
mars# cd all_reports
mars# mkdir pluto_rpt
mars# mkdir neptune_rpt
```

4. `dfstab` ファイル内のリソースをクライアントが利用できるようにします。

```
# shareall
```

5. 各クライアント上でクライアントのサブディレクトリを、マウントポイント `/usr/aset/masters/reports` にサーバーからマウントします。

```
# mount server:/usr/aset/client_rpt /usr/aset/masters/reports
```

6. /etc/vfstab ファイルを編集して、ブート時にディレクトリを自動的にマウントします。

   neptune 上の /etc/vfstab 内の次のサンプルエントリには、mars からマウントされるディレクトリ /usr/aset/all_reports/neptune_rpt と、neptune 上のマウントポイント /usr/aset/reports がリストされています。ブート時には、vfstab 内にリストされたディレクトリが自動的にマウントされます。

```
mars:/usr/aset/all_reports/neptune.rpt /usr/aset/reports nfs - yes
hard
```

ASET の問題を解決する方法

この章では、ASET によって生成されるエラーメッセージについて説明します。

ASET のエラーメッセージ

```
ASET failed: no mail program found.
```

意味

ASET は実行ログをユーザーに送るように指示されましたが、メールプログラムが見つからない。

対処方法

メールプログラムをインストールしてください。

```
Usage: aset [-n user[@host]] in /bin/mail or /usr/ucb/mail.
Cannot decide current and previous security levels.
```

意味

ASET は、今回と前回の呼び出しのセキュリティレベルを判別できない。

対処法

現在のセキュリティレベルがコマンド行オプションまたは ASETSECLEVEL 環境変数によって設定されているかどうかを確認してください。また、ASETDIR/archives/asetseclevel.arch の最終行に、以前のセキュリティレベルが正しく反映されているかどうかを確認してください。これらの値が設定されていないか、間違っている場合は、正しく指定し直してください。

```
ASET working directory undefined.
To specify, set ASETDIR environment variable or use command line
option -d.
ASET startup unsuccessful.
```

意味

ASET の作業 (操作) ディレクトリが定義されていないか、正しく定義されていない。

対処法

ASETDIR 環境変数または -d コマンド行オプションを使用して正しく指定し直し、ASET を再起動してください。

```
ASET working directory $ASETDIR missing.
ASET startup unsuccessful.
```

意味

ASET の作業 (操作) ディレクトリが定義されていないか、正しく定義されていない。ASETDIR 変数または -d コマンド行オプションによって、存在しないディレクトリが参照されている可能性がある。

対処法

正しいディレクトリ、つまり ASET ディレクトリ階層が入っているディレクトリが正しく参照されているかどうかを確認してください。

```
Cannot expand $ASETDIR to full pathname.
```

意味

ASETDIR 変数または -d コマンド行オプションで指定されたディレクトリ名を完全パス名に展開できない。

対処法

ディレクトリ名を正しく指定したかどうかと、ユーザーがアクセス権を持っている既存のディレクトリを参照しているかどうかを確認してください。

```
aset: invalid/undefined security level.
To specify, set ASETSECLEVEL environment variable or use command
line option -l, with argument= low/med/high.
```

意味

セキュリティレベルが定義されていないか、正しく定義されていない。low、med、または high の値以外は定義できない。

対処法

ASETSECLEVEL 変数または -l コマンド行オプションを使用して、3 つの値のいずれかを指定してください。

```
ASET environment file asetenv not found in $ASETDIR.
ASET startup unsuccessful.
```

意味

ASET は asetenv ファイルを作業用ディレクトリ内で見つけることができない。

対処法

ASET の作業ディレクトリ内に asetenv ファイルが入っているかどうかを確認してください。このファイルについては、asetenv(4) のマニュアルページを参照してください。

```
filename doesn't exist or is not readable.
```

意味

filename で指定されたファイルが存在しないか、読み取れない。このエラーは、特にチェックしたいユーザーのリストが入ったファイルを指定するときに -u オプションを使用すると発生することがある。

対処法

-u オプションの引数が存在し、読み取れるかどうかを確認してください。

```
ASET task list TASKLIST undefined.
```

意味

asetenv ファイル内で定義されているはずの ASET タスクリストが定義されていない。asetenv ファイルが無効である可能性がある。

対処法

asetenv ファイルを検査してください。タスクリストが User Configurable セクションで定義されているかどうかを確認します。また、ファイルの他の部分をチェックして、ファイルが変更されていないことを確認します。正常な asetenv ファイルの内容については、asetenv(4) のマニュアルページを参照してください。

```
ASET task list $TASKLIST missing.
ASET startup unsuccessful.
```

意味

asetenv ファイル内で定義されているはずの ASET タスクリストが定義されていない。asetenv ファイルが無効である可能性がある。

対処法

asetenv ファイルを検査してください。タスクリストが User Configurable セクションで定義されているかどうかを確認します。また、ファイルの他の部分をチェックして、ファイルが変更されていないことを確認します。正常な asetenv ファイルの内容については、asetenv(4) のマニュアルページを参照してください。

```
Schedule undefined for periodic invocation.
No tasks executed or scheduled. Check asetenv file.
```

意味

–p オプションを使用して ASET のスケジュール指定が要求されたが、変数 PERIODIC_SCHEDULE が asetenv ファイル内で定義されていない。

対処法

asetenv ファイルの User Configurable セクションをチェックして、変数が定義されていて、正しい書式になっているかどうかを確認してください。

```
Warning! Duplicate ASET execution scheduled.
Check crontab file.
```

意味

ASET のスケジュールが複数回指定されている。つまり、スケジュールがまだ有効な間に別のスケジュールを指定するように要求されている。複数のスケジュールはエラーであるとは限らず、複数のスケジュールが必要な場合は crontab(1) のスケジュール書式を使用するので、通常はこの指定が不要であることを示す警告にすぎない。

対処法

このコマンドを使って、正しいスケジュールが有効になっているかどうかを検査してください。ASET に関して不要な crontab エントリがないかどうかを確認してください。

第 25 章

システム資源の管理

ここでは、Solaris 環境におけるシステム資源の管理について説明します。次の章が含まれます。

第 26 章	ディスク割り当てシステム、アカウンティングプログラム、cron および at コマンドなどを使用して、システム資源を管理するための Solaris のコマンドとユーティリティについて説明します。
第 27 章	ワークステーション情報メニューなどの、一般的なシステム情報を調べたり、変更したりする手順を説明します。
第 28 章	使用されていないファイルや大きなディレクトリを見つけて、ディスク空間を最適化するための手順を説明します。
第 29 章	ディスク割り当てを設定し、管理する手順を説明します。
第 30 章	crontab および at を使用して、ルーチンまたは 1 度限りのシステムイベントのスケジュールを設定する手順を説明します。
第 31 章	アカウンティングを設定し、管理する手順を説明します。
第 32 章	システムアカウンティングソフトウェアについての参照情報を示します。

第 26 章

システム資源の管理

この章では、Solaris オペレーティング環境やその他の UNIX ソフトウェア製品が提供するシステム資源管理用のさまざまな機能について概要を説明します。これらの機能には、一般的なシステム情報の表示、ディスク空間の監視、ディスク割り当ての設定、アカウンティングプログラムの使い方、決められたコマンドを自動的に実行する `crontab` コマンドと `at` コマンドのスケジューリングなどが含まれます。

この章では、次の内容について概要を説明します。

システム資源に関する作業の参照先

システム資源を管理する手順については、次の内容を参照してください。

- 第 27 章
- 第 28 章
- 第 29 章
- 第 30 章
- 第 31 章

システム資源管理の新機能

今回の Solaris リリースでは、仮想端末が動的に割り当てられます。そのため、システムの仮想端末の数を増やす目的で、/etc/system ファイルの pt_cnt 変数を設定する必要はありません。

システム情報の表示と変更

第 27 章では、システが実行している Solaris リリース、システムのメモリー容量、使用可能なディスク容量など、一般的なシステム情報を確認する方法を説明します。

この章では、システムの日付や時刻を設定する方法、システム資源を増やす方法についても説明します。

ディスクの割り当て

ディスクの割り当て機能を使用することにより、システム管理者は、各ユーザーが使用できるディスク容量と i ノード数 (おおよそのファイル数に該当) を制限して UFS ファイルシステムを制御できます。これは特にユーザーのホームディレクトリがあるファイルシステム上で効果があります。(これにより公開ファイルシステムと /tmp ファイルシステムにはディスクが十分割り当てられないことがあります。)

割り当てを設定する一般的な手順は次のとおりです。

1. いくつかのコマンドを使用してファイルシステムにディスク割り当てを決め、システムがリブートし、そのファイルシステムがマウントされるたびに割り当てを確認することができます。/etc/vfstab ファイルにエントリを追加し、また、そのファイルシステムの一番上のディレクトリに quotas ファイルを作成する必要があります。
2. まず 1 人のユーザー用に割り当てを設定し、それを他のユーザー用にコピーします。
3. 割り当てが有効になる前に、他のコマンドが現在のディスクの使用状態をチェックし、競合していないかどうかを確認します。

4. 最後に、コマンドは 1 つ以上のファイルシステムでの割り当てを有効にします。

以上の手順により、あるファイルシステムがマウントされるたびに、そのファイルシステムのディスク割り当てが有効になるように設定できます。詳細は、第 29 章を参照してください。

一度設定しても、割り当てを変更して、ユーザーが使用できるディスク容量と i ノード数を調整できます。また、システムに変更が必要な場合は、それに合わせて割り当てを追加または削除できます。割り当ての変更、割り当てを超えてもかまわない時間の長さの設定、各割り当てを無効または削除する方法などについては、535 ページの「割り当ての変更と削除」を参照してください。

ディスク割り当てを監視できます。割り当てコマンドを使用することによりシステム管理者は、ファイルシステムでの割り当てを表示したり、割り当てを超えて使用しているユーザーを検索したりできます。これらのコマンドの使用方法については、532ページの「割り当てのチェック」を参照してください。

定型作業の自動実行

多くの定型的なシステムイベントは、自動的に実行されるように設定できます。それらの作業のなかには、定期的に実行する必要があるものがあります。その他の作業は、1 回しか実行する必要がありません。大部分は、夜間や週末などの就業時間外に実行できます。

この節では、crontab と at という 2 つのコマンドについて説明します。これらのコマンドでは、ピーク時間帯を避けて、または固定スケジュールに従って繰り返し実行して、定型的なコマンドが自動的に実行されるようにスケジュールすることができます。crontab は繰り返し実行されるコマンドをスケジュールし、at は 1 回実行されるコマンドをスケジュールします。

反復ジョブのスケジューリング (crontab)

定型的なシステム管理用コマンドは、crontab コマンドを使用して、毎日、毎週、または毎月それぞれ 1 回ずつ実行するようにスケジュールできます。

毎日 1 回の crontab によるシステム管理作業には次のようなものがあります。

- 作成後、数日以上経過した不要なファイルを一時ディレクトリから削除する
- アカウンティング要約コマンドを実行する

- df および ps コマンドを使用してシステムのスナップショットを取る
- 日常のセキュリティ監視を実行する
- システムのバックアップを実行する

毎週 1 回の crontab システム管理作業には次のようなものがあります。

- man -k で処理する catman データベースを構築し直す
- fsck -n を実行して存在するディスク問題のリストを表示する

毎月 1 回の crontab システム管理作業には次のようなものがあります。

- 当月使用されなかったファイルのリストを表示する
- 月次アカウンティングレポートを生成する

上記に加えて、連絡事項の通知の転送やバックアップファイルの削除や、さらに他の定型的システム作業を実行するように crontab コマンドをスケジュールすることもできます。

crontab ジョブのスケジューリングの詳細については、第 30 章を参照してください。

1 つのジョブのスケジューリング (at)

特定の 1 つのジョブを後で実行するように at コマンドを使用してスケジュールできます。

crontab と同様、at でも定型的コマンドの自動実行をスケジュールできます。しかし、crontab ファイルとは異なって、at ファイルはそれぞれのコマンドを 1 回実行して、その後はディレクトリから削除されてしまいます。したがって、at はそれぞれ 1 つのコマンドまたはスクリプトを実行して、後で調べられるようにそれらの出力を別々のファイルに送るのが最も効果的です。

at ジョブの実行を依頼するには、単にコマンド構文に従って at オプションで実行時刻を指定してください。at ジョブの実行依頼の詳細は、554ページの「at コマンドの説明」を参照してください。

at コマンドは、入力されたコマンドまたはスクリプトを、現在の環境変数のコピーと一緒に /var/spool/cron/atjobs ディレクトリに格納します。作成された at ジョブには、ファイル名として、at 待ち行列内での位置を指定する長い数値と .a 拡張子からなる、たとえば 793962000.a のような文字列が与えられます。

cron デーモンは、通常 15 分間隔で定期的に atrun プログラムを実行します。atrun は次に、それぞれのスケジュールされた時刻に各 at ジョブを実行しま

す。at ジョブが実行し終わると、それぞれのファイルが atjobs ディレクトリから削除されます。

at ジョブのスケジューリングの詳細については、第 27 章を参照してください。

システムアカウンティング

SunOS 5.8 のシステムアカウンティングソフトウェアは、ユーザー接続時間、プロセスに使用された CPU 時間、およびディスク使用率についてのデータを収集および記録できるプログラムセットです。一度このデータを収集すると、レポートを生成して、システム使用率に対して料金を請求できます。

アカウンティングプログラムは、次のような目的に使用できます。

- システム使用率の監視
- 問題発生時の対処
- 性能上の問題の追跡と解決
- システムセキュリティの管理

システムアカウンティングプログラムは、設定が済むと、ほとんどの場合自動的に実行されます。

アカウンティングの構成要素

アカウンティングユーティリティは、データから要約ファイルとレポートを生成する C 言語プログラムとシェルスクリプトを提供します。これらのプログラムは、ディレクトリ /usr/adm/acct と /usr/lib/acct にあります。

日次アカウンティングによって、次の 4 種類のアカウンティングを簡単に実行できます。

- 接続
- プロセス
- ディスク
- 料金計算

アカウンティングの動作

自動アカウンティングを設定するには、cron で自動的に起動できるように、それらのスクリプトを crontab ファイルに入れます。

次に、アカウンティングが機能する概要を次に示します。

1. システムを起動してからシャットダウンするまでの間に、システムの利用に関する (ユーザーログイン、実行されたプロセス、データの格納などの) raw データがアカウンティングファイルに収集されます。
2. 定期的に (通常 1 日に 1 回)、/usr/lib/acct/runacct プログラムが各種のアカウンティングファイルを処理して、累積要約ファイルと日次アカウンティングレポートを生成します。この日次レポートは /usr/lib/acct/prdaily プログラムによって出力されます。
3. runacct によって生成される累積要約ファイルは、monacct プログラムを実行して月に 1 回処理され出力できます。monacct によって生成される要約レポートは、月次またはその他の会計期間ベースのユーザーに対する効率的な課金手段になります。

アカウンティングソフトウェアを設定する手順については、第 31 章を参照してください。アカウンティングの機能の参照情報については、第 32 章を参照してください。

第 27 章

システム情報の確認と変更

この章では、最も一般的なシステム情報を確認および変更するために必要な手順を示します。

コマンドを使用したシステム情報の表示

表 27–1 に、一般的なシステム情報を表示するためのコマンドを示します。

表 27–1 システム情報を表示するためのコマンド

コマンド	表示できるシステム情報
psrinfo(1M)	プロセッサタイプ
isainfo(1)	サポートされるアプリケーション、および動作しているシステムのネイティブアプリケーションによってサポートされるビット数。ビット数は、トークンとしてスクリプトに渡すことができる
showrev(1M)	ホスト名、ホスト ID 番号、リリース、カーネルアーキテクチャ、アプリケーションアーキテクチャ、ハードウェアプロバイダ、ドメイン、およびカーネルのバージョン
uname(1)	オペレーティングシステム名、リリース、バージョン、ノード名、ハードウェア名、プロセッサタイプ
hostid(1)	ホスト ID 番号
prtconf(1)	インストールされているメモリー量
date(1)	日付と時刻

▼ システムが 64 ビット Solaris オペレーティング環境を実行できるか調べる方法

現在、64 ビット Solaris オペレーティング環境をサポートするプラットフォームは UltraSPARC システムだけです。システムが UltraSPARC システムか調べるには、次のコマンドを使用します。

```
$ uname -m
sun4u
```

uname -m コマンドの出力が sun4u なら、そのマシンは UltraSPARC システムです。

Solaris 8 リリースを実行している場合は、psrinfo コマンドを次のように使用します。

```
# psrinfo -v
Status of processor 0 as of: 07/12/99 09:41:47
  Processor has been on-line since 07/08/99 13:51:11.
  The sparcv9 processor operates at 333 MHz,
        and has a sparcv9 floating point processor.
```

プロセッサタイプが sparcv9 であれば、そのプラットフォームで 64 ビット Solaris オペレーティング環境が実行できます。以前のバージョンの psrinfo コマンドでは、すべてのプラットフォームがプロセッサタイプ sparc として報告されるため、この検査は機能しません。

▼ 64 ビット Solaris 機能が有効になっているか調べる方法

システムで 64 ビット Solaris 機能が有効になっているか調べるには isainfo コマンドが使用できます。有効になっていれば、システムは 64 ビットカーネルでブートされています。

例— 64 ビット Solaris 機能が有効になっているか調べる

32ビットカーネルが動作する UltraSPARC システムは、次のように表示されます。

```
$ isainfo -v
32-bit sparc applications
```

この出力は、システムが 32 ビットアプリケーションだけをサポートすることを示します。

64 ビットカーネルが動作する UltraSPARC システムは、次のように表示されます。

```
$ isainfo -v
64-bit sparcv9 applications
32-bit sparc applications
```

この出力は、システムが 32 ビットと 64 ビットのアプリケーションを両方サポートすることを示しています。

動作しているシステムのネイティブアプリケーションによってサポートされるビット数を表示するには、isainfo -b コマンドを使用します。

32 ビット Solaris オペレーティング環境が動作する SPARC、IA、UltraSPARC システムは、次のように表示されます。

```
$ isainfo -b
32
```

64 ビット Solaris オペレーティング環境が動作する 64 ビット UltraSPARC システムは、次のように表示されます。

```
$ isainfo -b
64
```

コマンドは 64 だけを返します。64 ビット UltraSPARC システムでは 32 ビットと 64 ビットのアプリケーションが両方動作しますが、64 ビットシステムで実行するには 64 ビットアプリケーションが最適です。

uname -p コマンドは sparc または i386 を返します。これは、既存の 32 ビットアプリケーションが問題なく動作することを示します。

▼ システムとソフトウェアのリリース情報を表示する方法

特定のシステムとソフトウェアのリリース情報を表示するには、showrev コマンドを使用します。

```
$ showrev [-a]
```

-a　　利用できるすべてのシステムおよびリリース情報を表示する

例 — システムとソフトウェアのリリース情報を表示する

次の例は、showrev コマンドの出力を示します。

```
$ showrev -a
Hostname: starbug
Hostid: nnnnnnnn
Release: 5.8
Kernel architecture: sun4u
Application architecture: sparc
Hardware provider: Sun_Microsystems
Domain: solar.com
Kernel version: SunOS 5.8 s28_26 February 2000
```

```
OpenWindows version:
OpenWindows Version 3.6.2  9 August 1999

No patches are installed
$
```

▼ 一般的なシステム情報を表示する方法 (uname)

システム情報を表示するには、uname コマンドを使用します。

```
$ uname[-a]
```

-a	オペレーティングシステム名とともに、システムノード名、オペレーティングシステムのリリース、オペレーティングシステムのバージョン、ハードウェア名、プロセッサタイプを表示する

例 — 一般的なシステム情報を表示する

次の例は、uname コマンドの出力を示します。

```
$ uname
SunOS
$ uname -a
SunOS starbug 5.8 Generic sun4u sparc SUNW,Ultra-5_10
$
```

▼ システムのホスト ID 番号を表示する方法

ホスト ID 番号を 16 進形式で表示するには、hostid コマンドを使用します。

```
$ hostid
```

例 — システムのホスト ID 番号を表示する

次の例は、hostid コマンドの出力を示します。

```
$ hostid
80a5d34c
```

▼ システムにインストールされているメモリーを表示する方法

システムにインストールされているメモリー量を表示するには、prtconf コマンドを使用します。

```
$ prtconf [| grep Memory]
```

grep Memory　　コマンド出力をメモリー情報だけに限定する

例 — システムにインストールされているメモリーを表示する

次の例は、prtconf コマンドの出力例を示します。

```
# prtconf | grep Memory
Memory size: 128 Megabytes
```

▼ 日付と時刻を表示する方法

システムクロックに従った現在の日付と時刻を表示するには、date コマンドを使用します。

```
$ date
```

例 — 日付と時刻を表示する

次の例は、date コマンドの出力例を示します。

```
$ date
Thu Sep 16 14:06:44 MDT 1999
$
```

コマンドを使用したシステム情報の変更

表 27–2 に、一般的なシステム情報を変更できるコマンドのマニュアルページと説明を示します。

表 27–2 システム情報を変更するためのコマンド

コマンド	変更できるシステム情報
`rdate(1M)`	日付と時刻を別のシステムの日付と時刻に合わせる
`date(1)`	日付と時刻を自分の指定に合わせる

これらのコマンドを使用することにより、システムの日付と時刻を設定して、サーバーなどの別のシステムの日付と時刻に同期させることができます。また、新しい日付と時刻を指定して、システムの日付と時刻を変更することもできます。

その日のメッセージ (MOTD) は `/etc/motd` に置かれています。この機能を使用すると、ログイン時のシステムメッセージによりすべてのユーザーに通知や問い合わせを送ることができます。ただし、この機能を使用するときは、常に必要なメッセージだけを送ります。メッセージファイルは定期的に編集し、無用になったメッセージを削除するようにしてください。

`/etc/system` ファイルを編集することにより、次の作業が行えます。

- ユーザー当たりのプロセス数を変更する
- ロック要求数を増加する
- 共有メモリーセグメント数を増加する

ネットワークでの Network Time Protocol (NTP) の使用

Solaris 2.6 以降、Solaris ソフトウェアには Delaware 大学の Network Time Protocol (NTP) 公開ドメインソフトウェアが添付されています。

NTP を使用すると、ネットワーク環境における正確な時間やネットワーク時間の同期を管理できます。`xntpd` デーモンは、UNIX システムの時間を インターネット標準時間サーバーの時間と合うように調整し、保守します。`xntpd` デーモンは、RFC

1305 に規定されている Network Time Protocol バージョン 3 標準を完全に実装しています。

xntpd デーモンは、システムの起動時に /etc/inet/ntp.conf ファイルを読み取ります。構成オプションの詳細は、xntpd(1M) のマニュアルページを参照してください。NTP サーバーとクライアントの設定手順については、次の節を参照してください。

ネットワークで NTP を使用する場合、次のことを考慮してください。

- xntpd デーモンは最小限のシステム資源だけしか使用しません。
- NTP クライアントはブート時に自動的に同期します。同期が取れなくなった場合は、タイムサーバーにアクセスしたときに再度同期を取ります。

▼ NTP サーバーを設定する方法

1. スーパーユーザーになります。
2. /etc/inet ディレクトリに移動します。
3. ntp.server ファイルを ntp.conf ファイルにコピーします。

```
# cp ntp.server ntp.conf
```

4. /etc/init.d ディレクトリに移動します。
5. xntpd デーモンを起動します。

```
# ./xntpd start
```

▼ NTP クライアントを設定する方法

1. スーパーユーザーになります。
2. /etc/inet ディレクトリに移動します。
3. ntp.client ファイルを ntp.conf ファイルにコピーします。

```
# cp ntp.client ntp.conf
```

4. /etc/init.d ディレクトリに移動します。

5. xntpd デーモンを起動します。

```
# ./xntpd start
```

▼ 他のシステムの日付と時刻に同期させる方法

1. スーパーユーザーになります。

2. 日付と時刻を設定し直して他のシステムと同期させるには、rdate コマンドを使用します。

```
# rdate another-system
```

another-system　　別のシステム名

3. date コマンドを使用してシステムの日付と時刻を調べ、システムの日付と時刻が正しく変更できたことを確認します。

   出力は同期させたシステムの日付と時刻に一致します。

例 — 他のシステムの日付と時刻に同期させる

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

```
earth# date
Thu Sep 16 11:08:27 MDT 1999
earth# rdate starbug
Thu Sep 16 14:06:37 1999
earth# date
Thu Sep 16 14:06:40 MDT 1999
```

▼ システムの日付と時刻を手作業で設定する方法

1. スーパーユーザーになります。
2. 次のように新しい日付と時刻を入力します。

```
# date mmddHHMM[[cc]yy]
```

mm 月。2 桁を使用

dd 日。2 桁を使用

HH 時。2 桁で 24 時間制を使用

MM 分。2 桁を使用

cc 世紀。2 桁を使用

yy 年。2 桁を使用

3. オプションを指定せずに `date` コマンドを実行し、システムの日付と時刻をチェックして、システムの日付と時刻が正しくリセットされていることを確認します。

   出力は、他のシステムと同じ日付と時刻を示します。

例 — システムの日付と時刻を手作業で設定する

次の例は、`date` コマンドを使用して手作業でシステムの日付と時刻を設定する方法を示します。

```
# date
Thu Sep 16 14:00:00 MDT 1999
# date 0916141099
Thu Sep 16 14:10:00 MDT 1999
```

▼ その日のメッセージを設定する方法

1. スーパーユーザーになります。

2. エディタを使って /etc/motd ファイルを開き、必要なメッセージを追加します。

   テキストを編集して、スペース、タブ、復帰改行を含めて、ユーザーログインプロセスの一部として表示されるメッセージを挿入します。

3. /etc/motd の内容を表示して、変更結果を確認します。

```
$ cat /etc/motd
Welcome to the UNIX Universe. Have a nice day.
```

例 — その日のメッセージを設定する

Solaris ソフトウェアのインストール時に、デフォルトのその日のメッセージが設定されます。メッセージの内容は次のような SunOS バージョン情報です。

```
$ cat /etc/motd
Sun Microsystems Inc.   SunOS 5.8       Generic  February 2000
```

次の例は、編集後の /etc/motd ファイルの内容を示します。このファイルは、ログインする各ユーザーに対してシステムの利用度に関する情報を提供します。

```
$ cat /etc/motd
The system will be down from 7:00 a.m to 2:00 p.m.on
Saturday, July 10, for upgrades and maintenance.
Do not try to access the system during those hours.
Thank you...
```

▼ ユーザー当たりのプロセス数を設定する方法

1. スーパーユーザーになります。

2. エディタを使って /etc/system ファイルを開き、次の行を追加します。

```
set maxuprc=value
```

value　　1 人のユーザーが同時に実行できるプロセス数

3. maxuprc の値が変更されていることを確認します。

```
# grep maxuprc /etc/system
set maxuprc=100
```

4. システムをリブートします。

例 — ユーザー当たりのプロセス数を設定する

次の例は、各ユーザーが 100 プロセスを実行できるようにする場合に、/etc/system ファイルに追加する行を示します。

```
set maxuprc=100
```

▼ 共有メモリーセグメント数を増加する方法

1. スーパーユーザーになります。

2. エディタを使って /etc/system ファイルを開き、次の変数を追加して共有メモリセグメントを増やします。

```
set shmsys:shminfo_shmmax=value
set shmsys:shminfo_shmmin=value
set shmsys:shminfo_shmmni=value
set shmsys:shminfo_shmseg=value
set semsys:seminfo_semmap=value
set semsys:seminfo_semmni=value
set semsys:seminfo_semmns=value
set semsys:seminfo_semmsl=value
set semsys:seminfo_semmnu=value
```

(続く)

```
set semsys:seminfo_semume=value
```

shmsys:shminfo_shmmax	共有メモリーセグメントの最大サイズ
shmsys:shminfo_shmmin	共有メモリーセグメントの最小サイズ
shmsys:shminfo_shmmni	共有メモリー識別子数
shmsys:shminfo_shmseg	プロセスごとのセグメント数
semsys:seminfo_semmap	セマフォマップ中のエントリ数
semsys:seminfo_semmni	セマフォ識別子数
semsys:seminfo_semmns	システム中のセマフォ数
semsys:seminfo_semmsl	ID ごとの最大セマフォ数
semsys:seminfo_semmnu	undo 機能を使用するプロセス数
semsys:seminfo_semume	プロセスごとの最大 undo 構造数

3. 共有メモリーの値が変更されていることを確認します。

```
# grep shmsys /etc/system
```

4. リブートします。

```
# init 6
```

例 — 共有メモリーセグメントを増加する

次の共有メモリー値は、大きなデータベースアプリケーションを実行するために、大容量のメモリー (たとえば 128M バイト) を搭載したシステムに適用されます。

```
set shmsys:shminfo_shmmax=268435456
set shmsys:shminfo_shmmin=200
set shmsys:shminfo_shmmni=200
set shmsys:shminfo_shmseg=200
set semsys:seminfo_semmap=250
set semsys:seminfo_semmni=500
set semsys:seminfo_semmns=500
set semsys:seminfo_semmsl=500
set semsys:seminfo_semmnu=500
set semsys:seminfo_semume=100
```

第 28 章

ディスク使用の管理

この章では、使用していないファイルや大きなディレクトリを見つけることにより、ディスク容量を最適化する方法を示します。この章で説明する手順は次のとおりです。

使用されているブロックとファイルの表示

df コマンドとそのオプションを使用して、空きディスクブロック数とファイル数のレポートを得ることができます。詳細は、df(1M) のマニュアルページを参照してください。

▼ ブロック、ファイル、およびディスク容量に関する情報を表示する方法

次のように df コマンドを入力して、ディスク容量の利用状況に関する情報を表示します。

```
$ df [directory] [-F fstype] [-g] [-k] [-t]
```

df	オプションを指定しない場合、マウントされている全ファイルシステム、それらの装置名、使用されている 512 バイトのブロックの合計数、ファイル数のリストを表示する
directory	ファイルシステムを確認したいディレクトリ。装置名、使用ブロック数、ファイル数を表示する
–F *fstype*	マウントされていないファイルシステム、それらの装置名、使用されている 512 バイトのブロック数、タイプ *fstype* のファイルシステム上のファイル数のリストを表示する
–g	マウントされている全ファイルシステムの statvfs 構造を表示する
–k	ファイルシステム、使用されている K バイト数、空き K バイト数、容量の利用率、マウントポイントのリストを表示する
–t	マウントされている全ファイルシステムの合計ブロック数と使用されているブロック数を表示する

例 — ブロック、ファイル、およびディスク容量に関する情報を表示する

次の例では、/usr/local を除き、すべてのファイルシステムがローカルでマウントされています。/usr/local はシステム mars からリモートにマウントされています。

```
$ df
/                  (/dev/dsk/c0t0d0s0 ):  287530 blocks    92028 files
/usr               (/dev/dsk/c0t0d0s6 ): 1020214 blocks   268550 files
/proc              (/proc             ):       0 blocks      878 files
/dev/fd            (fd                ):       0 blocks        0 files
/etc/mnttab        (mnttab            ):       0 blocks        0 files
/var/run           (swap              ):  396016 blocks     9375 files
/tmp               (swap              ):  396016 blocks     9375 files
/opt               (/dev/dsk/c0t0d0s5 ):  381552 blocks    96649 files
/export/home       (/dev/dsk/c0t0d0s7 ):  434364 blocks   108220 files
/usr/dist          (venus:/usr/dist   ):14750510 blocks  2130134 files
```

次の例では、ファイルシステム、合計 K バイト数、使用されている K バイト数、使用可能な K バイト数、容量の利用率、マウントポイントが表示されています。

```
$ df -k
Filesystem            kbytes    used   avail capacity  Mounted on
/dev/dsk/c0t0d0s0     192807   49042  124485    29%    /
/dev/dsk/c0t0d0s6    1190551  680444  450580    61%    /usr
/proc                      0       0       0     0%    /proc
fd                         0       0       0     0%    /dev/fd
mnttab                     0       0       0     0%    /etc/mnttab
swap                  198056       0  198056     0%    /var/run
swap                  198064       8  198056     1%    /tmp
/dev/dsk/c0t0d0s5     192807    2031  171496     2%    /opt
/dev/dsk/c0t0d0s7     217191       9  195463     1%    /export/home
venus:/usr/dist     20612581 13237326 6963005   66%    /usr/dist
```

次の例は、上の例と同じシステムに関する情報を示していますが、表示されているのは UFS ファイルシステムの情報だけです。

```
$ df -F ufs
/                  (/dev/dsk/c0t0d0s0 ):  287530 blocks    92028 files
/usr               (/dev/dsk/c0t0d0s6 ): 1020214 blocks   268550 files
/opt               (/dev/dsk/c0t0d0s5 ):  381552 blocks    96649 files
/export/home       (/dev/dsk/c0t0d0s7 ):  434364 blocks   108220 files
```

注 - /proc と /tmp はローカルファイルシステムですが、UFS ファイルシステムではありません (/proc は PROCFS ファイルシステム、/var/run および /tmp は TMPFS ファイルシステム、 /etc/mnttab は MNTFS ファイルシステムです)。

次の例は、マウントされているすべてのファイルシステム、装置名、使用されている 512 バイトブロックの合計数、ファイル数を示しています。2 行構成の各エントリの 2 行目は、それぞれのファイルシステムに割り当てられているブロックの合計数とファイルの合計数を示します。

```
$ df -t
/                  (/dev/dsk/c0t0d0s0 ):   287530 blocks    92028 files
                                 total:   385614 blocks    96832 files
/usr               (/dev/dsk/c0t0d0s6 ):  1020214 blocks   268550 files
                                 total:  2381102 blocks   300288 files
/proc              (/proc             ):        0 blocks      879 files
                                 total:        0 blocks      924 files
/dev/fd            (fd                ):        0 blocks        0 files
                                 total:        0 blocks       72 files
/etc/mnttab        (mnttab            ):        0 blocks        0 files
                                 total:        0 blocks        1 files
/var/run           (swap              ):   396112 blocks     9375 files
                                 total:   396112 blocks     9395 files
/tmp               (swap              ):   396112 blocks     9375 files
                                 total:   396128 blocks     9395 files
/opt               (/dev/dsk/c0t0d0s5 ):   381552 blocks    96649 files
                                 total:   385614 blocks    96832 files
/export/home       (/dev/dsk/c0t0d0s7 ):   434364 blocks   108220 files
                                 total:   434382 blocks   108224 files
/usr/dist          (venus:/usr/dist   ): 14750510 blocks  2130134 files
                                 total: 41225162 blocks  2482176 files
```

ファイルサイズの確認

ls コマンドを使用して、ファイルサイズを調べたりソートしたりできます。また、find コマンドを使用して、サイズの制限を超えているファイルを探すことができます。詳細は、ls(1) と find(1) のマニュアルページを参照してください。

▼ ファイルサイズを表示する方法

1. 確認したいファイルがあるディレクトリに移動します。
2. 次のように入力して、ファイルのサイズを表示します。

```
$ ls [-l] [-s]
```

-l　長形式でファイルとディレクトリのリストを表示し、それぞれのサイズをバイト単位で示す

-s　ファイルとディレクトリのリストを表示し、それぞれのサイズをブロック単位で示す

例 — ファイルサイズを表示する

次の例は、lastlog と messages が /var/adm ディレクトリ内のその他のファイルよりも大きいことを示します。

```
$ cd /var/adm
$ ls -l
total 144
drwxrwxr-x   5 adm      adm          512 Sep  1 14:11 acct/
-rw-------   1 uucp     bin            0 Sep  1 14:08 aculog
-r--r--r--   1 root     root      350700 Sep  3 10:37 lastlog
drwxr-xr-x   2 adm      adm          512 Sep  1 14:08 log/
-rw-r--r--   1 root     root       14619 Sep  2 16:11 messages
-rw-r--r--   1 adm      adm         8200 Sep  3 14:35 pacct
-rw-r--r--   1 adm      adm          920 Sep  3 10:47 pacct1
drwxr-xr-x   2 adm      adm          512 Sep  1 14:08 passwd/
drwxrwxr-x   2 adm      sys          512 Sep  1 14:11 sa/
drwxr-xr-x   2 root     sys          512 Sep  1 14:36 sm.bin/
-rw-rw-rw-   1 root     bin            0 Sep  1 14:08 spellhist
-rw-------   1 root     root         420 Sep  3 14:17 sulog
-rw-r--r--   1 root     bin         4092 Sep  3 10:37 utmpx
-rw-r--r--   1 root     root         122 Sep  1 15:39 vold.log
-rw-r--r--   1 adm      adm        11904 Sep  3 10:47 wtmpx
```

次の例は、lpsched.1 が 2 ブロックを使用していることを示します。

```
$ cd /var/lp/logs
$ ls -s
total 2            0 lpsched       2 lpsched.1
```

▼ サイズの大きなファイルを見つける方法

1. サイズの大きなファイルを探したいディレクトリに移動します。
2. 次のように入力して、ファイルのサイズをブロック単位に、最も大きいものから降順に表示します。

```
$ ls -s | sort -nr | more
```

sort -nr	ファイルのリストをブロックサイズの最も大きなものから降順に並べる

例 — サイズの大きなファイルを見つける

次の例では、lastlog と messages が /var/adm ディレクトリ内で最も大きなファイルです。

```
$ cd /var/adm
$ ls -s | sort -nr | more
  48 lastlog
  30 messages
  24 wtmpx
  18 pacct
   8 utmpx
   2 vold.log
   2 sulog
   2 sm.bin/
   2 sa/
   2 passwd/
   2 pacct1
   2 log/
   2 acct/
   0 spellhist
   0 aculog
total 144
```

▼ 指定されたサイズ制限を超えるファイルを見つける方法

次のように find コマンドを使用して、指定したサイズを超えるファイルを見つけてファイル名を表示します。

```
$ find directory -size +nnn
```

directory	ファイルを探したいディレクトリ
-size +*nnn*	512 バイトブロック数。指定したサイズを超えるファイルがリストに表示される

例 — 指定されたサイズ制限を超えるファイルを見つける

次の例は、作業中のカレントディレクトリ内の 400 ブロックを超えるファイルをどのように見つけるかを示します。

```
$ find . -size +400 -print
./Howto/howto.doc
./Howto/howto.doc.backup
./Howto/howtotest.doc
./Routine/routineBackupconcepts.doc
./Routine/routineIntro.doc
./Routine/routineTroublefsck.doc
./.record
./Mail/pagination
./Config/configPrintadmin.doc
./Config/configPrintsetup.doc
./Config/configMailappx.doc
./Config/configMailconcepts.doc
./snapshot.rs
```

ディレクトリサイズの確認

du コマンドとそのオプションを使用してディレクトリのサイズを表示できます。さらに quot コマンドを使用すれば、ユーザーアカウントによって占められるローカル UFS ファイルシステム上のディスク容量のサイズを知ることができます。これらのコマンドの詳細は、du(1M) と quot(1M) のマニュアルページを参照してください。

▼ ディレクトリ、サブディレクトリ、およびファイルのサイズを表示する方法

次のように du コマンドを入力して、1 つ以上のディレクトリ、サブディレクトリ、ファイルのサイズを表示します。サイズは 512 バイトブロック単位で表示されます。

```
$ du [-as] [directory ...]
```

`du`	指定した各ディレクトリとそれらの下の各サブディレクトリのサイズを合わせて表示する
`–a`	指定したディレクトリ内の各ファイルと各サブディレクトリのサイズと合計ブロック数を表示する
`–s`	指定したディレクトリ内の合計ブロック数を表示する
directory ...	調べたい 1 つ以上のディレクトリを指定する

例 — ディレクトリ、サブディレクトリ、およびファイルのサイズを表示する

次の例は、2 つのディレクトリとそれらのすべてのサブディレクトリのサイズを表示します。

```
$ du -s /var/adm /var/spool/lp
130     /var/adm
40      /var/spool/lp
```

次の例は、2 つのディレクトリとそれらのすべてのサブディレクトリとファイルのサイズ、および各ディレクトリ内の合計ブロック数を表示します。

```
$ du /var/adm /var/spool/lp
2       /var/adm/log
2       /var/adm/passwd
2       /var/adm/acct/fiscal
2       /var/adm/acct/nite
2       /var/adm/acct/sum
8       /var/adm/acct
2       /var/adm/sa
2       /var/adm/sm.bin
130     /var/adm
4       /var/spool/lp/admins
2       /var/spool/lp/fifos/private
2       /var/spool/lp/fifos/public
6       /var/spool/lp/fifos
2       /var/spool/lp/requests/starbug
4       /var/spool/lp/requests
2       /var/spool/lp/system
2       /var/spool/lp/tmp/starbug
2       /var/spool/lp/tmp/.net/tmp/starbug
4       /var/spool/lp/tmp/.net/tmp
2       /var/spool/lp/tmp/.net/requests/starbug
4       /var/spool/lp/tmp/.net/requests
```

(続く)

```
10      /var/spool/lp/tmp/.net
14      /var/spool/lp/tmp
40      /var/spool/lp
```

▼ ローカル UFS ファイルシステムのユーザー割り当てを表示する方法

1. スーパーユーザーになります。

2. 次のように入力して、ユーザー、ディレクトリまたはファイルシステム、**1024** バイト単位のブロック数を表示します。

```
# quot [-a] [filesystem]
```

–a	マウントされている各 UFS ファイルシステムの全ユーザーと 1024 バイト単位のブロック数を表示する
filesystem	UFS ファイルシステム。ユーザーと使用されているブロック数が表示される

注 - quot コマンドは、ローカル UFS ファイルシステムに対してだけ使用できます。

例 — ローカル UFS ファイルシステムのユーザー割り当てを表示する

次の例では、ルート (/) ファイルシステムのユーザーが表示され、次にマウントされているすべての UFS ファイルシステムのユーザーが表示されます。

```
# quot /
/dev/rdsk/c0t0d0s0:
43340   root
 3142   rimmer
   47   uucp
   35   lp
   30   adm
```

```
    4   bin
    4   daemon
# quot -a
/dev/rdsk/c0t0d0s0 (/):
43340   root
 3150   rimmer
   47   uucp
   35   lp
   30   adm
    4   bin
    4   daemon
/dev/rdsk/c0t0d0s6 (/usr):
460651  root
206632  bin
  791   uucp
   46   lp
    4   daemon
    1   adm
/dev/rdsk/c0t0d0s7 (/export/home):
    9   root
```

古いファイルと使用されていないファイルの検索と削除

ファイルの数が非常に多くなったファイルシステムを整理する場合、最近使用されていないファイルを見つけて削除します。使用されていないファイルは `ls` または `find` コマンドを使用して見つけることができます。詳細は、`ls(1)` と `find(1)` のマニュアルページを参照してください。

ディスク容量を節約するその他の方法としては、`/var/tmp` または `/var/spool` 内にあるような一時ファイルを空にしたり、`core` ファイルやクラッシュダンプファイルを削除したりするなどが含まれます。これらのファイルの詳細は、第 39 章を参照してください。

▼ 最新ファイルのリストを表示する方法

次のように `ls -t` コマンドを使用して、最も最近に作成または変更されたファイルから順番にファイルのリストを表示します。

```
$ ls -t [directory]
```

`-t`	最新タイムスタンプをリストの最初としてソートする
directory	ファイルを探したいディレクトリ

例 — 最新ファイルのリストを表示する

次の例は、`ls -t` をどのように使用して `/var/adm` ディレクトリ内の最新のファイルを見つけるかを示しています。`sulog` が最も新しく作成または変更されたファイルです。

```
$ ls -tl /var/adm
total 134
-rw-------   1 root     root         315 Sep 24 14:00 sulog
-r--r--r--   1 root     other     350700 Sep 22 11:04 lastlog
-rw-r--r--   1 root     bin         4464 Sep 22 11:04 utmpx
-rw-r--r--   1 adm      adm        20088 Sep 22 11:04 wtmpx
-rw-r--r--   1 root     other          0 Sep 19 03:10 messages
-rw-r--r--   1 root     other          0 Sep 12 03:10 messages.0
-rw-r--r--   1 root     root       11510 Sep 10 16:13 messages.1
-rw-r--r--   1 root     root           0 Sep 10 16:12 vold.log
drwxr-xr-x   2 root     sys          512 Sep 10 15:33 sm.bin
drwxrwxr-x   5 adm      adm          512 Sep 10 15:19 acct
drwxrwxr-x   2 adm      sys          512 Sep 10 15:19 sa
-rw-------   1 uucp     bin            0 Sep 10 15:17 aculog
-rw-rw-rw-   1 root     bin            0 Sep 10 15:17 spellhist
drwxr-xr-x   2 adm      adm          512 Sep 10 15:17 log
drwxr-xr-x   2 adm      adm          512 Sep 10 15:17 passwd
```

▼ 古いファイルと使用されていないファイルを見つけて削除する方法

1. スーパーユーザーになります。

2. 次のように入力して、指定した日数の間アクセスのないファイルを見つけて、ファイルにそれらのリストを書き込みます。

```
# find directory -type f[-atime + nnn] [-mtime + nnn] -print > filename
```

directory	ファイルを調べたいディレクトリ。この下のディレクトリも調べられる
`-atime +`*nnn*	指定した日数の間アクセスのないファイルを見つける
`-mtime +`*nnn*	指定した日数の間変更のないファイルを見つける
filename	使用されていないファイルのリストが書き込まれるファイル名

3. 上の手順でリストに書き込んだ使用されていないファイルを削除します。

```
# rm `cat filename`
```

filename	上の手順で作成されるファイルの名前で、使用されていないファイルのリストを含む

例 — 古いファイルと使用されていないファイルを見つけて削除する

次の例では、/var/adm 内の通常のファイルと、最近 60 日間アクセスされていないディレクトリを見つけ、使用されていないファイルのリストを /var/tmp/deadfiles に保存しています。それらのファイルはその後 rm コマンドで削除されます。

```
# find /var/adm -type f -atime +60 -print > /var/tmp/deadfiles &
# more /var/tmp/deadfiles
/var/adm/log/asppp.log
/var/adm/aculog
/var/adm/spellhist
/var/adm/wtmpx
/var/adm/sa/sa13
/var/adm/sa/sa27
/var/adm/sa/sa11
/var/adm/sa/sa23
/var/adm/sulog
/var/adm/vold.log
/var/adm/messages.1
/var/adm/messages.2
/var/adm/messages.3
```

(続く)

```
# rm `cat /var/tmp/deadfiles`
#
```

▼ 一時ディレクトリを一度にクリアする方法

1. スーパーユーザーになります。

2. 次のように入力して、/var/tmp ディレクトリに移動します。

```
# cd /var/tmp
```

注意 - 次の手順を実行する前に、正しいディレクトリにいることを確認してください。次の手順はカレントディレクトリ内のファイルをすべて削除します。

3. カレントディレクトリ内のファイルとサブディレクトリを削除します。

```
# rm -r *
```

4. 不要な一時サブディレクトリやファイル、または古いサブディレクトリやファイルがある他のディレクトリに移動して、上の手順 **3** を繰り返してサブディレクトリやファイルを削除します。

例 — 一時ディレクトリを一度にクリアする

次の例は、どのように /var/tmp ディレクトリを整理するかを示し、すべてのファイルとサブディレクトリが削除されたことを確認しています。

```
# cd /var/tmp
# ls
deadfiles           wxconAAAa0003r:0.0  wxconAAAa000NA:0.0
test_dir            wxconAAAa0003u:0.0  wxconAAAa000cc:0.0
wxconAAAa000zs:0.0
# rm -r *
```

(続く)

```
# ls
#
```

▼ core ファイルを見つけて削除する方法

1. スーパーユーザーになります。
2. core ファイルの探索を始めたいディレクトリに移動します。
3. 次のように入力して、ディレクトリとサブディレクトリ内のすべての core ファイルを見つけて削除します。

```
# find . -name core -exec rm {} \;
```

例 — core ファイルを見つけて削除する

次の例は、どのように find コマンドを使用して jones のユーザーアカウントから core ファイルを見つけて削除するかを示します。

```
# cd /home/jones
# find . -name core -exec rm {} \;
```

▼ クラッシュダンプファイルを削除する方法

クラッシュダンプファイルは非常に大きくなる可能性があります。したがって、必要以上に長期間保存しないでください。

1. スーパーユーザーになります。
2. 次のように入力して、クラッシュダンプファイルが格納されているディレクトリに変更します。

```
# cd /var/crash/system
```

system　　クラッシュダンプファイルを生成したシステム

注意 - 次の手順を実行する前に、正しいディレクトリにいることを確認してください。次の手順はカレントディレクトリ内のすべてのファイルを削除します。

3. クラッシュダンプファイルを削除します。

```
# rm *
```

4. クラッシュダンプファイルが削除されていることを確認します。

```
# ls
```

例 — クラッシュダンプファイルを削除する

次の例は、システム venus からどのようにクラッシュダンプファイルを削除するかと、クラッシュダンプファイルが削除されているかを確認する方法を示します。

```
# cd /var/crash/venus
# rm *
# ls
#
```

第 29 章

ディスク割り当ての管理

この章では、ディスクの割り当てを設定し管理する方法を示します。この章で説明する手順は次のとおりです。

ディスク割り当ての使用

ディスク割り当てを使用することによって、システム管理者は、個々のユーザーが使用できるディスク容量と i ノード数 (おおよそファイルの数に相当) を制限して、UFS ファイルシステムのサイズを制御できます。このため、ディスク割り当ては、特に、ユーザーのホームディレクトリが存在するファイルシステムで便利です。

設定後、ディスク割り当ては、ユーザーが使用するディスク容量や i ノード数に合わせて変更できます。さらに、ディスク割り当ては、システムの要求の変化に応じて、追加または削除できます。ディスク割り当て、またはディスク割り当てを超えることができる時間を変更する手順、個々のディスク割り当てを無効にする手順、あるいはディスク割り当てをファイルシステムから削除する手順については、535ページの「割り当ての変更と削除」を参照してください。

ディスク割り当ての状態を監視できます。ディスク割り当てコマンドを使用すると、管理者は、ファイルシステム上のディスク割り当てについての情報を表示したり、ディスク割り当てを超えているユーザーを検索したりできます。これらのコマンドの使い方については、532ページの「割り当てのチェック」を参照してください。

弱い制限値と強い制限値

弱い制限値と強い制限値の両方を設定できます。システムは、ユーザーが自分の強い制限値を超えることを許可しません。しかし、システム管理者は、ユーザーが一時的に超えることができる、弱い制限値 (「ディスク割り当て」と呼ぶこともある) を設定できます。弱い制限値は、強い制限値より小さくなければなりません。

いったんユーザーが弱い制限値を超えると、タイマーが起動します。タイマーが動いている間、ユーザーは弱い制限値を超えて操作できます。しかし、強い制限値は超えることができません。再びユーザーが弱い制限値を下回ると、タイマーはリセットされます。しかし、タイマーが期限切れになったときに、まだユーザーの使用率が弱い制限値を超えていた場合、弱い制限値は、強い制限値として実施されます。デフォルトでは、弱い制限値のタイマーは 7 日です。

タイマーの値は、`repquota` コマンドと `quota` コマンドを実行したときの `timeleft` フィールドに示されています。

たとえば、あるユーザーの弱い制限値が 10,000 ブロックで、強い制限値が 12,000 ブロックであると仮定します。そのユーザーのブロック使用率が 10,000 ブロックを超えて、タイマーも期限切れになった (7 日を超えた) 場合、そのユーザーは自分の使用率が弱い制限値を下回るまで、それ以上のディスクブロックをそのファイルシステム上に割り当てることはできません。

ディスクブロックとファイル制限の相違

ファイルシステムがユーザーに提供する資源は、2 つあります。(データの) ブロックと (ファイルの) i ノードです。各ファイルは、1 つの i ノードを使用します。ファイルデータは、データブロック内に格納されます (通常は、1K バイトブロック)。

ディレクトリがないと仮定すると、ユーザーは、ブロックを使用しなくても、すべて空のファイルを作成することによって、自分の i ノードディスク割り当てを超えることができます。また、ユーザーは単純に、ユーザーのディスク割り当て中のすべてのデータブロックを消費するぐらいの大きなファイルを 1 つだけ作成することによって、自分のブロックディスク割り当てを超えることができます。

ディスク割り当ての設定

ディスクの割り当てを設定することにより、ユーザーが利用できるディスク容量と、i ノードの数 (おおよそファイルの数に相当) を制限できます。これらの割り当ては、ファイルシステムがマウントされるたびに自動的に有効になります。この節ではファイルシステム用にディスク割り当てを構成し、設定し、有効にする手順を説明します。

ディスク割り当ての設定には、次の一般的な手順が含まれます。

1. システムをリブートしてファイルシステムをマウントするごとにディスク割り当てが実施されるように、一連のコマンドを使用して、ファイルシステムでディスク割り当てを利用できるようにするための準備をします。エントリは、`/etc/vfstab` ファイルに追加しなければなりません。また、`quotas` ファイルは、ファイルシステムの一番上のディレクトリで作成しなければなりません。
2. 1 人のユーザーに対してディスク割り当てを作成した後、そのディスク割り当てをプロトタイプとして、他のユーザーのディスク割り当てを設定できます。
3. ディスク割り当てを実際に有効にする前に、他のコマンドを使用して、整合性をチェックします。このチェックでは、提案したディスク割り当てと現在のディスク使用率を比較して、矛盾しないことを確認します。
4. 最後に、他のコマンドを使用して、1 つまたは複数のファイルシステム全体のディスク割り当てを有効にします。

上記の手順により、ファイルシステムがマウントされるたびに、そのファイルシステム上でディスク割り当てが自動的に有効になります。上記手順の特定の情報については、527ページの「割り当ての設定」を参照してください。

表 29–1 で、ディスク割り当てを設定するコマンドを説明します。

表 29–1 割り当てを行うコマンド

コマンド	機能
`edquota(1M)`	各ユーザーに対する i ノード数とディスク容量のハード制限とソフト制限を設定する
`quotacheck(1M)`	マウントされている各 UFS ファイルシステムを調べ、ファイルシステムのディスク割り当てファイルにある情報と比較し、矛盾があれば報告する
`quotaon(1M)`	指定したファイルシステムの割り当てを有効にする
`quota(1M)`	マウントされているファイルシステムのユーザーの割り当てを表示し、割り当てが正しく設定されていることを確認する

割り当て設定のガイドライン

ユーザーの割り当てを設定する前に、各ユーザーに割り当てるディスク容量の大きさとファイル数を決定する必要があります。ファイルシステムの合計領域サイズを超えないようにする場合は、ファイルシステムの合計サイズをユーザー数に等分すればよいでしょう。たとえば、3 人のユーザーが 100M バイトのスライスを共有し、それぞれが同じディスク容量のサイズを必要とする場合は、各ユーザーに 33M バイトずつ割り当てます。すべてのユーザーがそれぞれに割り当て制限を押し上げることがないような環境では、割り当ての合計がファイルシステムの合計サイズを超えるように個々の割り当てを設定することも可能です。たとえば、3 人のユーザーが 100M バイトのスライスを共有する場合は、それぞれに 40M バイトを割り当ててもよいということです。

あるユーザーについて `edquota` コマンドを使用して割り当てを決定すると、同じファイルシステム上の他のユーザーにも同じ割り当てプロトタイプとして利用できます。

UFS ファイルシステムの割り当てを構成し、各ユーザーに対する割り当てを終了したら、実際に割り当てを有効にする前に、`quotacheck` コマンドを使用して整合性をチェックしてください。システムがリブートされる機会がそれほど多くない場合、`quotacheck` を定期的に実行するようお勧めします。

edquota により設定した割り当ては、quotaon コマンドを使用して有効にしなければ強制的に設定されません。割り当てファイルを正しく構成したら、システムがリブートし、そのファイルシステムがマウントされるたびに、割り当ては自動的に有効になります。

割り当ての設定

表 29–2 作業マップ: 割り当ての設定

作業	説明	手順の説明
1. ファイルシステムの割り当ての構成	/etc/vfstab を編集して、ファイルシステムがマウントされるたびに割り当てが有効になるようにする。また、quotas ファイルを作成する	527ページの「割り当て用にファイルシステムを構成する方法」
2. 1 ユーザー用の割り当ての設定	edquota を使用して 1 ユーザーアカウント用にディスクと i ノードの割り当てを行う	528ページの「1 ユーザー用の割り当てを設定する方法」
3. 複数ユーザー用の割り当ての設定	省略可能。edquota を使用して、その他のユーザーアカウント用にプロトタイプの割り当てを適用する	529ページの「複数ユーザーに対して割り当てを設定する方法」
4. 整合性のチェック	quotacheck を使用して、1 つまたは複数のファイルシステムの整合性について、現在の使用状況とディスクの割り当てを比較する	530ページの「割り当ての整合性を確認する方法」
5. 割り当てを有効にする	quotaon を使用して、1 つまたは複数のファイルシステムの割り当てを有効にする	531ページの「割り当てを有効にする方法」

▼ 割り当て用にファイルシステムを構成する方法

1. スーパーユーザーになります。

2. /etc/vfstab ファイルを編集します。割り当てを設定しようとする各 **UFS** ファイルシステムの「mount options」フィールドに rq を追加します。

3. 割り当てを格納しようとするファイルシステムの最上位ディレクトリに変更します。

4. 次のように入力して、quotas というファイルを作成します。

```
# touch quotas
```

5. **root** のみ、読み取り権／書き込み権を与えます。

```
# chmod 600 quotas
```

例 — 割り当て用にファイルシステムを構成する

次の例は /etc/vfstab の内容で、システム pluto の /export/home ディレクトリが NFS ファイルシステムとして、割り当てが有効 (mount options 列の rq エントリで示される) なローカルシステム上のマウントポイントにマウントされていることを示しています。

```
#device           device   mount       FS    fsck   mount   mount
#to mount         to fsck  point       type  pass   at boot options
#
pluto:/export/home -       /export/home nfs    -     yes    rq
```

次の例は /etc/vfstab の内容で、割り当てが有効 (mount options 列の rq エントリで示される) なローカル UFSファイルシステムが /work ディレクトリにマウントされていることを示しています。

```
#device           device            mount  FS   fsck mount   mount
#to mount         to fsck           point  type pass at boot options
#
/dev/dsk/c0t4d0s0 /dev/rdsk/c0t4d0s0 /work ufs  3    yes     rq
```

▼ 1 ユーザー用の割り当てを設定する方法

1. スーパーユーザーになります。

2. 次のように入力して割り当てエディタを使用して、quotas ファイルが最上位ディレクトリにある各マウント済み **UFS** ファイルシステムに対して、**1** 行の割り当て情報を含む一時ファイルを作成します。

```
# edquota username
```

username　　割り当てを設定しようとするユーザー名

3. **1K** バイトディスクブロック数の弱い制限値と強い制限値、および **i** ノード数の弱い制限値と強い制限値を、それぞれ **0 (**デフォルト**)** から各ファイルシステム用に指定されている割り当て値に変更します。

4. ユーザーの割り当てを設定できたかどうかを確認するには、次のように quota コマンドを使用します。

```
# quota -v username
```

–v　　ディスク割り当てがある、マウント済みのファイルシステム上の、ユーザーのディスク割り当て情報を表示する

username　　ディスク割り当て制限を表示するユーザー名を指定する

例 — 1 ユーザー用の割り当てを設定する

次の例は、/files だけがマウント済みファイルシステムで、edquota によって開かれた一時ファイルの内容を示しています。このファイルシステムの最上位ディレクトリに quotas ファイルが含まれています。

```
fs /files blocks (soft = 0, hard = 0) inodes (soft = 0, hard = 0)
```

次の例は、割り当て設定後の一時ファイルの上と同じ行を示しています。

```
fs /files blocks (soft = 50, hard = 60) inodes (soft = 90, hard = 100)
```

▼ 複数ユーザーに対して割り当てを設定する方法

1. スーパーユーザーになります。

2. 次のように割り当てエディタを使用して、すでにプロトタイプユーザー用に設定した割り当てを指定するその他のユーザーに適用します。

```
# edquota -p prototype-user  username  ...
```

prototype-user すでに割り当てを設定してあるアカウントのユーザー名

username .. 1 人以上の追加アカウントのユーザー名を指定する

例 — 複数ユーザーに対してプロトタイプ割り当てを設定する

次の例は、ユーザー bob に対して設定されている割り当てをユーザー mary と john に適用します。

```
# edquota -p bob mary john
```

▼ 割り当ての整合性を確認する方法

注 - ディスクのデータの正確さを保つには、quotacheck コマンドを実行するとき、チェックするファイルシステムが他のユーザーによって使用できないようにしてください。システムをリブートするとき、quotacheck コマンドが自動的に実行されます。

1. スーパーユーザーになります。

2. 次のように **UFS** ファイルシステム上の整合性チェックを実行します。

```
# quotacheck [-v] filesystem
```

–v (省略可能) 特定のファイルシステム上の各ユーザーのディスク割り当てを示す

–a /etc/vfstab ファイルに rq エントリがある全ファイルシステムをチェックする

filesystem チェックするファイルシステムを指定する

詳細は、quotacheck(1M) のマニュアルページを参照してください。

例 — 割り当ての整合性を確認する

次の例は、スライス /dev/rdsk/c0t0d0s7 上の /export/home ファイルシステムのディスク割り当てをチェックしています。/export/home ファイルシステムは、/etc/vfstab ファイルに rq エントリを持つ、唯一のファイルシステムです。

```
# quotacheck -va
*** Checking quotas for /dev/rdsk/c0t0d0s7 (/export/home)
```

▼ 割り当てを有効にする方法

1. スーパーユーザーになります。
2. 次のように quotaon コマンドを使用して、ファイルシステムの割り当てを有効にします。

```
# quotaon [-v] -a filesystem ...
```

–v	詳細オプション
–a	/etc/vfstab ファイル内に rq エントリがある全ファイルシステムの割り当てを有効にする
filesystem ...	指定する 1 つ以上のファイルシステムの割り当てを有効にする

例 — 割り当てを有効にする

次の例は、スライス /dev/dsk/c0t4d0s2 と /dev/dsk/c0t3d0s2 上のファイルシステムのディスク割り当てを有効にしています。

```
# quotaon -v /dev/dsk/c0t4d0s7 /dev/dsk/c0t3d0s7
/dev/dsk/c0t4d0s7: quotas turned on
/dev/dsk/c0t3d0s7: quotas turned on
```

割り当てのチェック

ディスクとハードの割り当てを設定して有効にしたら、それらの割り当てを超過して使用しているユーザーをチェックできます。また、ファイルシステム全体の割り当て情報をチェックすることもできます。

表 29–3 でディスク割り当てをチェックするのに使用するコマンドを説明します。

表 29–3 割り当てチェック用コマンド

コマンド	作業
`quota(1M)`	ユーザー割り当てと現在のディスク使用量を表示する。ユーザーの割り当て超過使用量も表示可能
`repquota(1M)`	指定されたファイルシステムの割り当て、ファイル、および所有しているディスク容量を表示する

▼ 割り当てを超過したかどうかを確認する方法

`quota` コマンドを使用して、割り当てが適用されているファイルシステム上の個々のユーザーの割り当てとディスク使用量を表示できます。

1. スーパーユーザーになります。
2. 次のように入力して、割り当てが有効にされているマウント済みファイルシステムのユーザー割り当てを表示します。

```
# quota [-v] username
```

–v	割り当てが設定されているマウント済みファイルシステムすべてについてユーザー割り当てを表示する
username	ユーザーアカウントのユーザー名またはユーザー識別番号 (UID)

例 — 割り当てを超過したかどうかを確認する

次の例は、UID 301 によって識別されるユーザーアカウントに 1K バイトの割り当てが設定されているが、ディスク容量をまったく使用していないことを示しています。

```
# quota -v 301
Disk quotas for bob (uid 301):
Filesystem  usage  quota limit timeleft files quota  limit timeleft
/export/home   0      1     2             0      2      3
```

Filesystem	ファイルシステムのマウントポイント
usage	現在のブロック使用数
quota	弱いブロック制限値
limit	強いブロック制限値
timeleft	ディスク割り当てタイマーの残り時間 (日単位)
files	現在の i ノード使用数
quota	弱い i ノード制限値
limit	強い i ノード制限値
timeleft	ディスク割り当てタイマーの残り時間 (日単位)

▼ ファイルシステムの割り当てを確認する方法

repquota コマンドを使用して 1 つ以上のファイルシステム上のすべてのユーザーの割り当てとディスク使用量を表示します。

1. スーパーユーザーになります。
2. ディスクがまったく使用されていなくても、1 つまたはすべてのファイルシステムのすべての割り当てを表示します。

```
# repquota [-v]  -a filesystem
```

–v	資源を消費していないユーザーも含めて、すべてのユーザーのディスク割り当てを報告する
–a	すべてのファイルシステムについて報告する
filesystem	指定したファイルシステムについて報告する

例 — ファイルシステムの割り当てを確認する

次の例は、割り当てが 1 つのファイルシステム (/export/home) だけに対して有効なシステムでの repquota コマンドからの出力を示しています。

```
# repquota -va
/dev/dsk/c0t3d0s7 (/export/home):
                  Block limits               File limits
User      used   soft   hard  timeleft  used  soft   hard  timeleft
#301  --             0      1   2.0 days         0      2      3
#341  --    57     50     60   7.0 days  2      90    100
```

Block Limits

used	現在のブロック使用数
soft	弱いブロック制限値
hard	強いブロック制限値
timeleft	ディスク割り当てタイマーの残り時間 (日単位)

File Limits

used	現在の i ノード使用数
soft	弱い i ノード制限値

hard	強い i ノード制限値
timeleft	ディスク割り当てタイマーの残り時間 (日単位)

割り当ての変更と削除

割り当てを変更して、ユーザーが使用するディスク容量と i ノード数を調整できます。または、必要に応じて各ユーザーから、あるいはファイルシステム全体から割り当てを削除できます。

表 29–4 で、割り当てを変更または削除するのに使用するコマンドを示します。

表 29–4 割り当てを変更または削除するコマンド

コマンド	機能
edquota(1M)	各ユーザーについて i ノード数またはディスク容量の強い制限値と弱い制限値を変更する。また、任意のユーザーが弱い制限値を超えることが許される期間の長さを変更する
quotaoff(1M)	指定したファイルシステムの割り当てを無効にする

▼ 期間の弱い制限値のデフォルトを変更する方法

デフォルトでは、ユーザーは 1 週間、割り当ての期間の弱い制限値を超えることができます。1 週間以上期間の弱い制限値を超えると、システムはそのユーザーに対し、i ノードとディスクブロックの使用を禁止します。

edquota コマンドを使用すると、この割り当ての期間制限を変更できます。

1. スーパーユーザーになります。
2. 次のように割り当てエディタを使用して、期間の弱い制限値を含む一時ファイルを作成します。

```
# edquota -t
```

3. 期間制限を、**0 (**デフォルト**)** から数値とキーワード
month、week、day、hour、min、または sec を使用して指定する値に変更します。

注 - この手順は、現在のディスク割り当て違反者には影響しません。

例 — 期間の弱い制限値のデフォルトを変更する

次の例は、/export/home がただ 1 つのマウント済みファイルシステムであるシステムで edquota によって開かれた一時ファイルの内容を示しています。値 0 (デフォルト) は、デフォルトで、1 週間の期間制限値が使用されることを意味します。

```
fs /export/home blocks time limit = 0 (default), files time limit = 0 (default)
```

次の例は、ブロック割り当ての超過に対する期間制限値が 1 週間に変更され、ファイル数の超過に対する期間制限値が 10 日に変更された後の、上の例と同じ一時ファイルの内容を示しています。

```
fs /export/home blocks time limit = 2 weeks, files time limit = 16 days
```

▼ 1 ユーザーの割り当てを変更する方法

1. スーパーユーザーになります。

2. 次のように割り当てエディタを使用して、quotas ファイルがそれぞれの最上位ディレクトリにある各マウント済みファイルシステムに対して **1** 行ずつエントリが入っている一時ファイルを開きます。

```
# edquota username
```

username　　割り当てを変更したいユーザー名

注意 - edquota コマンドの引数として複数のユーザーを指定できますが、表示される情報にはどのユーザーのものなのか示されないので、混乱を招く恐れがあります。

---

3. **1K** バイトディスクブロック数の弱い制限値と強い制限値、および **i** ノード数の弱い制限値と強い制限値を入力します。

4. ユーザーの割り当てが正しく変更できたか確認するには、次のように quota コマンドを使用します。

```
# quota -v username
```

–v	ディスク割り当てが有効にされている、すべてのマウント済みのファイルシステムについて、ユーザーのディスク割り当て情報を表示します
username	割り当てを確認したいユーザー名

例 — 1 ユーザーの割り当てを変更する

次の例は、/files だけがマウント済みファイルシステムで、edquota によって開かれた一時ファイルの内容を示しています。このファイルシステムの最上位ディレクトリに quotas ファイルが含まれています。

```
fs /files blocks (soft = 0, hard = 0) inodes (soft = 0, hard = 0)
```

次の例は、上と同じ一時ファイルの割り当て変更後の内容を示しています。

```
fs /files blocks (soft = 0, hard = 500) inodes (soft = 0, hard = 100)
```

次の例は、ユーザー smith の強い制限値の変更結果を確認する方法と、1K バイトブロック数と i ノード数の強い制限値がそれぞれ 500 と 100 に変更されていることを示しています。

```
# quota -v smith
Disk quotas for smith (uid 12):
Filesystem  usage  quota  limit  timeleft  files  quota  limit  timeleft

  /files     1       0     500              1       0     100
```

▼ 1 ユーザーの割り当てを無効にする方法

1. スーパーユーザーになります。

2. 次のように割り当てエディタを使用して、quotas ファイルがその最上位ディレクトリにある各マウント済みファイルシステムに対して 1 行の割り当て情報を含む一時ファイルを作成します。

```
# edquota username
```

username　　割り当てを無効にしようとするユーザー名

注意 - edquota コマンドの引数として複数のユーザーを指定できますが、表示される情報にはどのユーザーのものなのか示されないので、混乱を招く恐れがあります。

3. 1K バイトディスクブロック数の弱い制限値と強い制限値、および i ノード数の弱い制限値と強い制限値を 0 (ゼロ) に変更します。

注 - 必ずこれらの値を 0 (ゼロ) に変更してください。テキストファイルから行を削除してはいけません。

4. ユーザーの割り当てを無効にできたかどうかを確認するには、次のように quota コマンドを使用します。

```
# quota -v username
```

–v　　ディスク割り当てが有効にされている、すべてのマウント済みのファイルシステムについて、ユーザーのディスク割り当て情報を表示します

username　　割り当てを確認しようとするユーザー名またはユーザー識別番号 (UID)

例 — 1 ユーザーの割り当てを無効にする

次の例は、/files だけがマウント済みファイルシステムで、edquota によって開かれた一時ファイルの内容を示しています。このファイルシステムの最上位ディレクトリに quotas ファイルが含まれています。

```
fs /files blocks (soft = 50, hard = 60) inodes (soft = 90, hard = 100)
```

次の例は、割り当てを無効にした後の上と同じ一時ファイルの内容を示しています。

```
fs /files blocks (soft = 0, hard = 0) inodes (soft = 0, hard = 0)
```

▼ 割り当てを無効にする方法

1. スーパーユーザーになります。
2. 次のように入力して、割り当てを無効にします。

```
# quotaoff [-v] -a filesystem ...
```

–v	割り当てが無効にされた場合、各ファイルシステムからメッセージを表示する
–a	全ファイルシステムの割り当てを無効にする
filesystem	指定する 1 つ以上のファイルシステムの割り当てを無効にする

例 — 割り当てを無効にする

次の例は、/export/home ファイルシステムの割り当てを無効にします。

```
# quotaoff -v /export/home
/export/home: quotas turned off
```

第 30 章

システムイベントのスケジュール設定

この章では、crontab および at コマンドを使用して、ルーチンまたは 1 度限りのシステムイベントをスケジュールする方法を示します。さらに、cron.deny、cron.allow、at.deny の各ファイルを使用して、これらのコマンドへのアクセスを制御する方法も説明します。

この章で説明する手順は次のとおりです。

システムイベントのスケジューリング用コマンド

システムイベントは、crontab コマンドを使用して定期的に繰り返し実行するようにスケジュールできます。また、at コマンドを使用して、特定のシステムイベントを指定の時刻に実行するようにスケジュールすることもできます。表 30–1 に crontab、at の他に、これらのコマンドへのアクセスを制御できるファイルを示します。

表 30–1 システムイベントのスケジューリング用コマンド

コマンド	スケジューリングの対象	ファイルの格納場所	アクセス制御用ファイル
crontab	一定間隔で実行する複数のシステムイベント	/var/spool/cron/crontabs	/etc/cron.d/cron.allow /etc/cron.d/cron.deny
at	1 つのシステムイベント	/var/spool/cron/atjobs	/etc/cron.d/at.deny

繰り返されるシステムイベントのスケジューリング (cron)

以降の各項で、crontab ファイルをどのように作成、編集、表示、削除するか、さらに、それらのファイルへのアクセスをどのように制御するかを説明します。

crontab ファイルの内容

cron デーモンは、各 crontab ファイル内にあるコマンドに従ってシステムイベントをスケジュールします。crontab ファイルには、それぞれ一定間隔で実行されるコマンドが 1 行に 1 つずつ入っています。各行の先頭は cron デーモンが各コマンドを実行する日時情報です。

たとえば、SunOS ソフトウェアのインストール時に root という名前の crontab ファイルが提供されますが、このファイルの内容は次のとおりです。

```
10 3 * * 0,4 /etc/cron.d/logchecker
10 3 * * 0   /usr/lib/newsyslog
15 3 * * 0 /usr/lib/fs/nfs/nfsfind
1 2 * * * [ -x /usr/sbin/rtc ] && /usr/sbin/rtc -c > /dev/null 2>&1
30 3 * * * [ -x /usr/lib/gss/gsscred_clean ] && /usr/lib/gss/gsscred_clean
```

最初のコマンド行は、日曜日と木曜日の午前 3 時 10 分に logchecker を実行するようシステムに指示します。2 番目のコマンド行は、毎日曜日の午前 3 時 10 分に newsyslog を実行するようにシステムをスケジュールします。3 番目のコマンド行は毎日曜日午前 3 時 15 分に nfsfind を実行するようにシステムに指示します。4 番目のコマンド行は、毎日夏時間をチェックして、必要であれば修正するようにシステムに指示します。RTC タイムゾーンも /etc/rtc_config ファイルもない場合、このエントリは何もしません。 5 番目のコマンド行では、Generic Security Service テーブル /etc/gss/gsscred_db に重複エントリがないかどうかを調べ、あれば削除します。

crontab ファイル内のコマンド行の構文の詳細は、544ページの「crontab ファイルエントリの構文」を参照してください。

crontab ファイルは /var/spool/cron/crontabs) に格納されます。SunOS ソフトウェアのインストール時には、root 以外にもいくつかの crontab ファイルが提供されます (表 30–2 を参照してください)。

表 30–2 デフォルトの crontab ファイル

crontab ファイル	機能
adm	アカウンティング
lp	印刷
root	一般的なシステム機能とファイルシステムの整理
sys	性能情報の収集
uucp	一般的な uucp の整理

デフォルトの crontab ファイルの他に、ユーザーは crontab ファイルを作成してユーザー自身のシステムイベントをスケジュールできます。

その他の crontab ファイルは、それらの中に作成されるユーザーのアカウントに基づいて、bob、mary、smith、jones などのように命名されます。

root または他のユーザーが所有する crontab ファイルにアクセスするには、スーパーユーザーの特権が必要です。

crontab ファイルを作成、編集、表示、削除する手順については、542ページの「システムイベントのスケジューリング用コマンド」で説明します。

cron デーモンのスケジューリング

cron デーモンは crontab コマンドの自動スケジューリングを行います。このデーモンの機能は、通常 15 分おきに、/var/spool/cron/crontabs を調べて crontab ファイルがないか確認します。新しい crontab ファイルがないか、または既存の crontab が変更されていないかを確認し、いずれかがあった場合は、ファイル内のリストから実行時刻を読み取り、コマンドが正しい時刻に実行されるよう指示します。

ほとんど同様に、cron デーモンは /var/spool/cron/atjobs ディレクトリ内の at ファイルのスケジューリングを制御します。

crontab ファイルエントリの構文

crontab ファイルは、1 行に 1 つのコマンドが入った構成になっています。これらのコマンド行の最初の 5 つのフィールドには、コマンドが実行される時刻を指定し、それぞれスペースで区切ります。表 30–3 にこれら 5 つのフィールドを示します。

表 30–3 crontab 時刻フィールドの値

時刻フィールド	値
分	0-59
時	0-23
日	1-31
月	1-12
曜日	0 - 6 (0 は日曜日)

表 30–3　crontab 時刻フィールドの値　続く

次に、crontab 時刻フィールドで特殊文字を使用する際のガイドラインを示します。

- 各フィールドはスペースで区切る
- 複数の値の間はコンマで区切る
- 値の範囲はハイフンを使用して指定する
- 取り得るすべての値を含むには、ワイルドカードとしてアスタリスクを使用する
- コメントまたは空白行を示すには、行の先頭にコメント記号 (#) を使用する

たとえば、次の crontab コマンドエントリの例は、毎月 1 日と 15 日の午後 4 時に、ユーザーのコンソールウィンドウに注意を促すメッセージを表示します。

```
0 16 1,15 * * echo Timesheets Due > /dev/console
```

crontab ファイル内の各コマンドは、長くても 1 行内に入れなければなりません。これは、crontab が 2 番目以降のキャリッジリターンを認識しないからです。(つまり、複数行とした場合は、同じコマンドとはみなされないということです。) crontab のエントリとコマンドオプションの詳細は、crontab(1) のマニュアルページを参照してください。

crontab ファイルの作成と編集

crontab ファイルの最も単純な作成方法は、crontab -e コマンドを使用して、EDITOR 環境変数によって定義されているテキストエディタを呼び出すものです。この変数を設定していない場合は、crontab はデフォルトのエディタ ed を使用します。EDITOR 環境変数に使い慣れたエディタを定義します。次の例は、エディタが定義されているかどうかを確認する方法と、vi をデフォルトとして設定する方法を示しています。

```
$ which $EDITOR
$
$ EDITOR=vi
$ export EDITOR
```

crontab ファイルを作成すると、自動的に /var/spool/cron/crontabs ディレクトリ内に格納され、作成者のユーザー名が命名されます。スーパーユーザー特権があれば、他のユーザーや root の crontab ファイルを作成または編集できます。

crontab コマンドエントリは、544ページの「crontab ファイルエントリの構文」の説明に従って入力してください。

▼ crontab ファイルを作成または編集する方法

1. **(省略可能) root または他のユーザーが所有する crontab ファイルを作成または編集する場合は、スーパーユーザーになります。**

2. **次のように入力して、新しい crontab ファイルを作成するか、既存の crontab ファイルを編集します。**

```
$ crontab -e [username]
```

username	他のユーザーのアカウント名。作成または編集するにはスーパーユーザーの権限が必要

注意 - 誤ってオプションを指定しないで crontab コマンドを入力した場合は、使用しているエディタの割り込みキーを押してください。割り込みキーを押すと、変更結果を保存しないでエディタを終了できます。この時点でファイルの編集を終了して変更結果を保存すると、既存の crontab ファイルが空のファイルで上書きされてしまいます。

3. **544ページの「crontab ファイルエントリの構文」で説明している構文に従って、コマンド行を crontab ファイルに追加します。**

   crontab ファイルは /var/spool/cron/crontabs に格納されます。

4. **crontab -l コマンドを使用して、crontab ファイルを確認します。**

```
# crontab -l [username]
```

例 — crontab ファイルを作成または編集する

次の例は、他のユーザーのための crontab ファイルをどのように作成するかを示します。

```
# crontab -e jones
```

新しい crontab ファイルに次のコマンドエントリを追加すると、毎日曜日の午前 1 時に、ユーザー jones のホームディレクトリから、すべてのログファイルが自動的に削除されます。このコマンドエントリは出力先を変更しないので、出力先変更文字がコマンド行の *.log の後に追加されて、そのコマンドが正しく実行されるようにしています。

```
# This command helps clean up user accounts.
1 0 * * 0 rm /home/jones/*.log > /dev/null 2>&1
```

▼ crontab ファイルを確認する方法

特定のユーザーの crontab ファイルがあるかどうかを確認するには、/var/spool/cron/crontabs ディレクトリで ls -l コマンドを使用します。たとえば、次の表示はユーザー smith と jones の crontab ファイルがあることを示しています。

```
$ ls -l /var/spool/cron/crontabs
-rw-r--r--  1 root     sys          190 Feb 26 16:23 adm
-rw-------  1 root     staff        225 Mar  1  9:19 jones
-rw-r--r--  1 root     root        1063 Feb 26 16:23 lp
-rw-r--r--  1 root     sys          441 Feb 26 16:25 root
-rw-------  1 root     staff         60 Mar  1  9:15 smith
-rw-r--r--  1 root     sys          308 Feb 26 16:23 sys
```

ユーザーの crontab ファイルの内容を確認するには、crontab -l を使用します。548ページの「crontab ファイルを表示する方法」を参照してください。

crontab ファイルの表示

crontab -l コマンドは、cat コマンドが他のファイルタイプの内容を表示するのとまったく同様に、使用しているユーザーの crontab ファイルの内容を表示しま

す。このコマンドを使用するために、ディレクトリを (crontab ファイルが入っている) /var/spool/cron/crontabs に変更する必要はありません。

デフォルトでは、crontab -l コマンドは自分自身の crontab ファイルを表示します。他のユーザーの crontab ファイルは、スーパーユーザーでなければ表示できません。

▼ crontab ファイルを表示する方法

1. **(省略可能) root または他のユーザーの crontab ファイルを表示する場合は、スーパーユーザーになります。**

2. **次のように入力して、crontab ファイルを表示します。**

```
$ crontab -l [username]
```

username	他のユーザーのアカウント名。作成または編集するにはスーパーユーザーの権限が必要

注意 - 誤ってオプションを指定しないで crontab コマンドを入力した場合は、使用しているエディタの割り込みキーを押してください。割り込みキーを押すと、変更結果を保存せずに crontab コマンドを終了できます。この時点でファイルの編集を終了して変更結果を保存すると、既存の crontab ファイルが空のファイルで上書きされてしまいます。

例 — crontab ファイルを表示する

次の例で、どのように crontab -l を使用してデフォルトユーザー、デフォルト root、他のユーザーの crontab ファイルを表示するかを示します。

```
$ crontab -l
13 13 * * * chmod g+w /home1/documents/*.book > /dev/null 2>&1
$ su
Password:
# crontab -l
#ident  "@(#)root       1.19    98/07/06 SMI"   /* SVr4.0 1.1.3.1       */
#
# The root crontab should be used to perform accounting data collection.
```

(続く)

```
#
# The rtc command is run to adjust the real time clock if and when
# daylight savings time changes.
#
10 3 * * 0,4 /etc/cron.d/logchecker
10 3 * * 0   /usr/lib/newsyslog
15 3 * * 0 /usr/lib/fs/nfs/nfsfind
1 2 * * * [ -x /usr/sbin/rtc ] && /usr/sbin/rtc -c > /dev/null 2>&1
30 3 * * * [ -x /usr/lib/gss/gsscred_clean ] && /usr/lib/gss/gsscred_clean
# crontab -l jones
13 13 * * * cp /home/jones/work_files /usr/backup/. > /dev/null
2>&1
```

crontab ファイルの削除

デフォルトでは、rm コマンドを使用して誤って crontab ファイルを削除してしまうことがないように、crontab ファイルは保護されています。crontab ファイルを削除する場合は、rm コマンドではなく crontab -r コマンドを使用してください。

デフォルトでは、crontab -r は、このコマンドを実行したユーザーの crontab を削除します。root またはその他のユーザーの crontab を削除するには、スーパーユーザーにならなければなりません。

このコマンドを使用するには、ディレクトリを (crontab ファイルが入っている) /var/spool/cron/crontabs に変更する必要はありません。

▼ crontab ファイルを削除する方法

1. **(省略可能) root または他のユーザーの crontab ファイルを削除するには、スーパーユーザーになります。**

2. **次のように入力して、crontab ファイルを削除します。**

```
$ crontab -r [username]
```

username	他のユーザーのアカウント名。作成または編集するにはスーパーユーザーの権限が必要

注意 - 誤ってオプションを指定しないで crontab コマンドを入力した場合は、使用しているエディタの割り込みキーを押してください。割り込みキーを押すと、変更結果を保存せずに crontab コマンドを終了できます。この時点でファイルの編集を終了して変更結果を保存すると、既存の crontab ファイルが空のファイルで上書きされてしまいます。

3. crontab ファイルが削除されていることを確認します。

```
# ls /var/spool/cron/crontabs
```

例 — crontab ファイルを削除する

次の例では、ユーザー smith が crontab -r コマンドを使用して自分の crontab ファイルを削除します。

```
$ ls /var/spool/cron/crontabs
adm     jones     lp     root     smith     sys     uucp
$ crontab -r
$ ls /var/spool/cron/crontabs
adm     jones     lp     root     sys     uucp
```

crontab へのアクセスの制御

cron.deny と cron.allow という /etc/cron.d ディレクトリ内の 2 つのファイルを使用して、crontab へのアクセスを制御できます。これらのファイルによって、指定したユーザーだけが、それぞれ自分の crontab ファイルの作成、編集、表示、または削除などの crontab 関連操作を実行できるようにします。

cron.deny および cron.allow ファイルは、それぞれ 1 行に 1 ユーザー名が入ったリストからなります。これらのアクセス制御用ファイルは、次のように連携して機能を果たします。

- cron.allow が存在する場合は、このファイルにリストされているユーザーだけが crontab ファイルを作成、編集、表示、または削除できます。
- cron.allow が存在しない場合は、cron.deny にリストされているユーザーを除くすべてのユーザーが crontab ファイルの実行を依頼できます。
- cron.allow も cron.deny も存在しない場合は、root 以外は crontab を実行できません。

cron.deny と cron.allow ファイルを編集または作成するには、スーパーユーザーの権限が必要です。

SunOS ソフトウェアのインストール時に、デフォルトで次の cron.deny ファイルが提供されます。

```
$ cat /etc/cron.d/cron.deny
daemon
bin
smtp
nuucp
listen
nobody
noaccess
```

上記のすべてのユーザーが、crontab コマンドにアクセスできません。このファイルを編集すれば、crontab コマンドへのアクセスを拒否したいユーザー名を追加できます。

デフォルトでは、cron.allow ファイルは提供されません。つまり、Solaris ソフトウェアのインストール後には、デフォルトの cron.deny ファイルにリストされているユーザー以外のユーザーすべてが crontab にアクセスできます。cron.allow ファイルを作成した場合、そのユーザーだけが crontab コマンドにアクセスできます。

▼ crontab へのアクセスを拒否する方法

1. スーパーユーザーになります。

2. /etc/cron.d/cron.deny ファイルを編集し、crontab コマンドを使用させないユーザー名を次のように 1 行に 1 つずつ追加します。

```
daemon
bin
smtp
nuucp
listen
nobody
noaccess
username1
username2
username3
.
.
.
```

3. /etc/cron.d/cron.deny ファイルを確認します。

```
# cat /etc/cron.d/cron.deny
```

▼ crontab へのアクセスを特定のユーザーに限定する方法

1. スーパーユーザーになります。

2. /etc/cron.d/cron.allow ファイルを作成します。

3. 次のように、crontab コマンドの使用を許可するユーザー名を 1 行に 1 つずつ入力します。

```
root
username1
username2
username3
.
.
.
```

このリストには必ず root を追加してください。追加しなければ、スーパーユーザーからの crontab コマンドへのアクセスが拒否されてしまいます。

例 — crontab へのアクセスを特定のユーザーに限定する

次は、ユーザー visitor、jones、temp に crontab をアクセスさせない cron.deny ファイルの例です。

```
$ cat /etc/cron.d/cron.deny
daemon
bin
smtp
nuucp
listen
nobody
noaccess
jones
temp
visitor
```

次は cron.allow ファイルの例です。ユーザー smith、jones、lp、root だけが crontab にアクセスできます。

```
$ cat /etc/cron.d/cron.allow
root
jones
lp
smith
```

▼ 制限された crontab へのアクセスを確認する方法

特定のユーザーが crontab にアクセスできるかどうかを確認するには、そのユーザーのアカウントにログインして、crontab -l コマンドを実行します。

```
$ crontab -l
```

そのユーザーが crontab にアクセスできて、すでに crontab ファイルを作成している場合は、その crontab ファイルが表示されます。そのユーザーが crontab にアクセスできるが、crontab ファイルがない場合は、次のようなメッセージが表示されます。

(crontab: crontab ファイルを開けません)

このユーザーは、cron.allow (が存在する場合) に含まれているか、cron.deny に含まれていません。

ユーザーが crontab にアクセスできない場合は、上記の crontab ファイルの有無に関わらず、次のメッセージが表示されます。

(crontab: cron を使用許可されていません)

これは、ユーザーが cron.allow (が存在する場合) に含まれていないか、cron.deny ファイルに含まれていることを意味します。

1 つのシステムイベントのスケジューリング (at)

以降の項では、at(1) コマンドを使用してコマンドやスクリプトなどのジョブを後で特定の時刻に実行するようにスケジュールする方法、それらのジョブを削除または表示する方法、および at コマンドへのアクセスを制御する方法について説明します。

デフォルトでは、ユーザーはそれぞれ自分の at ジョブファイルを作成、表示、または削除できます。root または他のユーザーの at ファイルにアクセスするには、スーパーユーザーの権限が必要です。

at ジョブの実行を依頼すると、at ジョブにジョブ識別番号と .a 拡張子が与えられ、それがファイル名になります。

at コマンドの説明

at ジョブファイルを実行するには、次の手順に従います。

1. コマンド実行時刻を指定して at ユーティリティを起動します。
2. 後で実行させるコマンドまたはスクリプトを入力します。

注 - このコマンドまたはスクリプトからの出力が重要な場合は、後で調べられるように必ずファイルに書き込むようにしてください。

たとえば、次の at ジョブは、7 月 31 日の真夜中に smith のユーザーアカウントから core ファイルを削除します。

```
$ at 11:45pm July 31
at> rm /home/smith/*core*
at> Press Control-d
```

(続く)

```
commands will be executed using /bin/csh
job 933486300.a at Sat Jul 31 23:45:00 1999
```

at コマンドのセキュリティ

特定のユーザーだけがそれぞれの at ジョブに関する待ち行列情報を作成、削除、または表示できるように、at コマンドへのアクセスを制御するファイルを設定できます。at へのアクセスを制御するファイルは /etc/cron.d/at.deny です。ここにはユーザー名が列挙 (1 行に 1 人) されています。このファイルに列挙されているユーザーは、at コマンドにアクセスできません。

Solaris ソフトウェアのインストール時に作成される at.deny ファイルには、次のユーザー名が含まれます。

```
daemon
bin
smtp
nuucp
listen
nobody
noaccess
```

スーパーユーザーの特権があれば、このファイルを編集して、at へのアクセスを制限したい他のユーザー名を追加できます。

▼ at ジョブを作成する方法

1. 次のように、自分のジョブを実行したい時刻を指定して at コマンドを入力し、Return キーを押します。

```
$ at [-m] time [date]
```

-m	ジョブ終了後にメールを送る
time	ジョブをスケジュールしたい時刻の時単位の値。24 時間制を使用しない場合は、am または pm を追加する。midnight、noon、now はキーワードとして使用可能。分単位の値の指定は省略可能
date	月または曜日の名前の最初の 3 英字以上、またはキーワード today または tomorrow

2. at プロンプトに、実行したいコマンドまたはスクリプトを **1** 行に **1** つずつ入力します。各行の終わりで Return キーを押すことにより、複数のコマンドを入力できます。

3. at ユーティリティを終了し、Control-d キーを押して at ジョブを保存します。

   作成できた at ジョブは待ち行列番号を割り当てられ、それがそのファイル名にもなります。この番号は at ユーティリティの終了時に表示されます。

例 — at ジョブを作成する

次の例は、ユーザー jones が彼女のバックアップファイルを 7:30 pm に削除するように作成した at ジョブを示しています。彼女は、ジョブの終了後にメールメッセージを受け取れるように、-m オプションを使用しています。

```
$ at -m 1930
at> rm /home/jones/*.backup
at> Press Control-d
job 897355800.a at Mon Jul  12 19:30:00 1999
```

jones は次のメールメッセージを受け取りました。このメッセージは at ジョブが終了したことを確認しています。

```
Your ``at'' job ``rm /home/jones/*.backup''
completed.
```

次の例は、jones が大きな at ジョブをどのように土曜日の朝 4:00 にスケジュールしているかを示します。その出力は、big.file に送られます。

```
$ at 4 am Saturday
at> sort -r /usr/dict/words > /export/home/jones/big.file
```

▼ at 待ち行列を表示する方法

at 待ち行列で実行を待っているジョブを確認するには、次に示すように atq コマンドを使用します。このコマンドは、その使用者が作成した at ジョブに関する状態情報を表示します。

```
$ atq
```

▼ at ジョブを確認する方法

at ジョブが作成できたかどうかを確認するには、atq コマンドを使用します。次の atq コマンドは、jones に属する at ジョブが待ち行列に入っていることを確認しています。

```
$ atq
  Rank   Execution Date       Owner  Job          Queue   Job Name
  1st    Jul 12, 1999 19:30   jones  897355800.a      a     stdin
  2nd    Jul 14, 1999 23:45   jones  897543900.a      a     stdin
  3rd    Jul 17, 1999 04:00   jones  897732000.a      a     stdin
```

▼ at ジョブを表示する方法

自分の at ジョブの実行時刻に関する情報を表示するには、次のように at -l コマンドを使用します。

```
$ at -l [job-id]
```

-l *job-id*　　状態を確認したいジョブの識別番号

例 — at ジョブを表示する

次の例は、at -l コマンドからの出力を示しています。このコマンドは、特定のユーザーが依頼したすべてのジョブに関する状態情報を得ることを目的としています。

```
$ at -l
897543900.a Wed Jul 14 23:45:00 1999
897355800.a Mon Jul 12 19:30:00 1999
897732000.a Sat Jul 17 04:00:00 1999
```

次の例は、at -l コマンドに 1 つのジョブを指定して表示された出力を示しています。

```
$ at -l 897732000.a
897732000.a  Sat Jul 17 04:00:00 1999
```

▼ at ジョブを削除する方法

1. **(省略可能) root** または他のユーザーの at ジョブを削除する場合は、スーパーユーザーになります。

2. 次のように入力して、at ジョブを実行される前に待ち行列から削除します。

```
$ at -r [job-id]
```

-r *job-id*　　削除したいジョブの識別番号

3. at ジョブを削除できたかどうかを確認するには、at -l **(**または atq**)** コマンドを使用して at 待ち行列に残っているジョブを表示します。識別番号を指定したジョブは、このリストに現れてはなりません。

```
$ at -l [job-id]
```

例 — at ジョブを削除する

次の例では、ユーザーが 7 月 17 日の午前 4 時に実行されるようにスケジュールした at ジョブを削除しようとしています。まず、このユーザーは at 待ち行列を表示してそのジョブの識別番号を探します。次に、そのジョブを at 待ち行列から削除します。最後に、at 待ち行列をもう一度表示して上記のジョブが削除されていることを確認します。

```
$ at -l
897543900.a Wed Jul 14 23:45:00 1999
897355800.a Mon Jul 12 19:30:00 1999
897732000.a Sat Jul 17 04:00:00 1999
$ at -r 897732000.a
$ at -l 897732000.a
at: 858142000.a: No such file or directory
```

at へのアクセスの制御

at.deny ファイルに含まれているユーザーは、at を使用してジョブをスケジュールすることも、at 待ち行列の状態を調べることもできません。

at.deny ファイルは、Solaris ソフトウェアのインストール時にディレクトリ /etc/cron.d に格納されます。そのときに、同じユーザーがこのファイルとデフォルトの cron.deny ファイルの両方に含まれます。

```
daemon
bin
smtp
nuucp
listen
nobody
noaccess
```

このファイルを編集するには、スーパーユーザー特権が必要です。

▼ at へのアクセスを拒否する方法

1. スーパーユーザーになります。

2. /etc/cron.d/at.deny ファイルを開きます。at コマンドを使用させないようにするユーザー名を 1 行に 1 つずつ追加または削除します。

```
daemon
bin
smtp
nuucp
listen
nobody
noaccess
username1
username2
username3
.
.
.
```

例 — at へのアクセスを拒否する

次は、ユーザー smith と jones が at コマンドにアクセスできないように編集された at.deny ファイルの例です。

```
$ cat at.deny
daemon
bin
smtp
nuucp
listen
nobody
noaccess
jones
smith
```

▼ at アクセスの拒否を確認する方法

特定のユーザー名が正しく /etc/cron.d/at.deny に追加されているかどうかを確認するには、そのユーザー名でログインして、at -l コマンドを使用します。そのユーザーが at コマンドにアクセスできない場合は、次のメッセージが表示されます。

```
# su smith
Password:
$ at -l
at: 使用許可されていません
```

同様に、そのユーザーが at ジョブの実行を依頼しようとした場合は、次のメッセージが表示されます。

```
$ at 2:30pm
at: 使用許可されていません
```

これで、そのユーザーが at.deny ファイルに含まれていることが確認されます。

at がコマンドにアクセスできる場合、at -l コマンドは何も返しません。

第 31 章

アカウンティングの設定と管理作業

この章では、アカウンティングを設定し、管理する手順について説明します。

この章で説明する手順は次のとおりです。

システムアカウンティングの設定

システムアカウンティングは、システムがマルチユーザーモード (システム状態 2) のときに実行されるように設定できます。システムアカウンティングには、次の内容が含まれます。

1. /etc/rc0.d/K22acct および /etc/rc2.d/S22acct ファイルの作成
2. /var/spool/cron/crontabs/adm および /var/spool/cron/crontabs/root ファイルの変更

アカウント用スクリプトのほとんどは、/var/spool/cron/crontabs/adm データベースファイルに追加されます。表 31–1 に、デフォルトのアカウンティング管理スクリプトを説明します。

表 31–1 デフォルトのアカウンティング管理スクリプト

アカウンティングスクリプト	目的	実行方法
ckpacct(1M)	/usr/adm/pacct ログファイルのサイズをチェックする	定期的
runacct(1M)	接続、ディスク、および料金のアカウンティング情報を処理する	日次
monacct(1M)	会計レポートを生成する。定期的に実行される	会計期間に基づく

これらのデフォルトは変更できます。上記エントリをデータベースに追加して、アカウンティングプログラムをインストールした後、アカウンティングは自動的に実行されるようになります。

▼ システムアカウンティングを設定する方法

1. スーパーユーザーになります。
2. 必要な場合は、pkgadd コマンドを使用して、システムに SUNWaccr と SUNWaccu パッケージをインストールします。
3. 次のように入力して、/etc/init.d/acct を実行レベル **2** の起動スクリプトとしてインストールします。

```
# ln /etc/init.d/acct /etc/rc2.d/S22acct
```

4. /etc/init.d/acct を実行レベル **0** の停止スクリプトとしてインストールします。

```
# ln /etc/init.d/acct /etc/rc2.d/S22acct
```

5. プログラム ckpacct、runacct、および monacct が自動的に起動するように、adm ユーザーの crontab ファイルに次の行を追加します。

```
# EDITOR=vi; export EDITOR
# crontab -e adm
0 * * * * /usr/lib/acct/ckpacct
30 2 * * * /usr/lib/acct/runacct 2> /var/adm/acct/nite/fd2log
30 7 1 * * /usr/lib/acct/monacct
```

6. プログラム dodisk が自動的に起動するように、root の crontab ファイルに次の行を追加します。

```
# crontab -e
30 22 * * 4 /usr/lib/acct/dodisk
```

7. /etc/acct/holidays を編集して祭日と休日を取り入れます。

8. システムをリブートするか、次のように入力します。

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

例 — アカウンティングを設定する

次の例は、/usr/lib/acct/ckpacct、/usr/lib/acct/runacct、および /usr/lib/acct/monacct を実行する crontab エントリを /var/spool/cron/crontabs/adm に追加する方法を示します。

```
#ident  "@(#)adm        1.5     92/07/14 SMI"    /* SVr4.0 1.2   */
#
# The adm crontab file should contain startup of performance
# collection if the profiling and performance feature has been
# installed.
0 * * * * /usr/lib/acct/ckpacct
30 2 * * * /usr/lib/acct/runacct 2> /var/adm/acct/nite/fd2log
30 7 1 * * /usr/lib/acct/monacct
```

次の例は、/usr/lib/acct/dodisk を実行する crontab エントリを /var/spool/cron/crontabs/root に追加する方法を示します。

```
#ident  "@(#)root       1.16    98/04/28 SMI"   /* SVr4.0 1.1.3.1       */
#
# The root crontab should be used to perform accounting data collection.
#
# The rtc command is run to adjust the real time clock if and when
# daylight savings time changes.
#
10 3 * * 0,4 /etc/cron.d/logchecker
10 3 * * 0   /usr/lib/newsyslog
15 3 * * 0 /usr/lib/fs/nfs/nfsfind
1 2 * * * [ -x /usr/sbin/rtc ] && /usr/sbin/rtc -c > /dev/null 2>&1
30 3 * * * [ -x /usr/lib/gss/gsscred_clean ] && /usr/lib/gss/gsscred_clean
30 22 * * 4 /usr/lib/acct/dodisk
```

次に /etc/acct/holidays ファイルの例を示します。

```
* @(#)holidays January 1, 1999
*
* Prime/Nonprime Table for UNIX Accounting System
*
* Curr Prime Non-Prime
* Year Start Start
*
  1999 0800 1800
*
* only the first column (month/day) is significant.
*
* month/day Company
*   Holiday
*
1/1  New Years Day
7/4  Indep. Day
12/25  Christmas
```

ユーザーへの課金

ファイルの復元、リモート印刷などの特別なユーザーサービスを要求時に提供する場合は、chargefee(1M) という機能を使用してユーザーに対する課金処理を行うことができます。chargefee は、課金をファイル /var/adm/fee に記録します。

次に、runacct ユーティリティが実行されるたびに、新しいエントリが拾い出されて、全体のアカウンティングレコードにマージされます。

▼ ユーザーに課金する方法

1. スーパーユーザーになります。

2. ユーザーに提供されたサービスに対して課金するように設定します。

```
# chargefee username amount
```

username 課金したいユーザーアカウント

amount ユーザーに対する課金の単位数

例 — ユーザーに課金する

次の例は、ユーザーがアカウント print_customer にログインするたびに 10 単位を課金します。

```
# chargefee print_customer 10
```

アカウンティング情報の管理

この節では、アカウンティング情報を保守する方法について説明します。

壊れたファイルと wtmpx エラーを修復する

UNIX のアカウンティングシステムは堅固ではなく、ファイルが壊れたり失われることがあります。そのようなファイルにも、単に無視してよいものや、バックアップから復元できるものがあります。しかし、特定のファイルは、アカウンティングシステムの完全性を維持するために修復しなければなりません。

wtmpx(4) ファイルは、アカウンティングシステムを日常的に運用する上で発生する大部分の問題の原因になっています。日付が変更され、システムがマルチユー

ザーモードになると、1 組の日付変更レコードが /var/adm/wtmp に書き込まれます。wtmpfix(1M) ユーティリティは、日付変更されたときの wtmpx レコード内のタイムスタンプの調整用として用意されています。ただし、日付変更とリブートとの組み合わせによっては、wtmpfix のチェックから漏れて、acctcon の処理を失敗させることがあります。wtmpx の問題を解決する手順については、568ページの「wtmpx ファイルを修復する方法」を参照してください。

▼ wtmpx ファイルを修復する方法

1. スーパーユーザーになります
2. ディレクトリ /var/adm/acct/nite に変更します。
3. 次のように、バイナリファイル wtmp.*MMDD* を **ASCII** ファイル xwtmp に変換します。

```
# fwtmp wtmp.MMDD xwtmp
```

MMDD　　2 桁の数値で指定される月日

4. xwtmp を編集します。壊れたファイルを削除するか、始めから日付変更までのすべてのレコードを削除します。
5. **ASCII** ファイル xwtmp をバイナリファイルに変換し、壊れたファイルを上書きします。

```
# fwtmp -ic xwtmp wtmp.MMDD
```

tacct エラーを修復する

/var/adm/acct/sum/tacct の完全性は、システム資源に対してユーザーに課金している場合は重要です。負の番号、重複したユーザー ID、または 65535 のユーザー ID など、不可思議な tacct レコードが現れることがありますが、その場合は prtacct で /var/adm/acct/sum/tacctprev を印刷して、チェックしてください。内容が正しい場合は、最新の /var/adm/acct/sum/tacct.*MMDD* ファイ

ルを使用して、/var/adm/acct/sum/tacct ファイルを作成し直してください。次の手順は、簡単な修復手順の概要を説明しています。

▼ tacct エラーを修復する方法

1. スーパーユーザーになります。
2. ディレクトリ /var/adm/acct/sum に変更します。
3. 次のように、tacct.***MMDD*** の内容をバイナリから **ASCII** 形式に変換します。

```
# acctmerg -v tacct.MMDD xtacct
```

MMDD　　2 桁の数字で指定される月日

4. xtacct ファイルを編集して、不良レコードを削除し、重複レコードを別のファイルに書き込みます。
5. xtacct ファイルを **ASCII** 形式からバイナリに変換します。

```
# acctmerg -i xtacct tacct.MMDD
```

MMDD　　2 桁の数字で指定される月日

6. ファイル tacct.prv と tacct.***MMDD*** をマージしてファイル tacct を生成します。

```
# acctmerg tacctprev tacct.MMDD tacct
```

runacct を再起動する

runacct プログラムはいろいろな原因で失敗することがあります。一番多い原因は、システムがクラッシュする、/var がディスク容量を使い果たす、wtmpx ファイルが壊れたなどです。active.*MMDD* ファイルが存在する場合、まずエラー

メッセージがないか調べます。active ファイルとロックファイルが存在する場合、異常なメッセージがないかどうか fd2log ファイルを調べます。

runacct は、引数を指定しないで実行すると、その日の最初の起動とみなします。runacct を起動し直し、もう一度 runacct にアカウンティングをやり直させる月日を指定する場合は、引数 *MMDD* が必要です。処理のエントリポイントは statefile の内容に基づきます。statefile を無効にするには、次のように処理を開始したい状態をコマンド行に指定します。

注意 - runacct プログラムを手動で実行するときは、ユーザー adm として実行していることを確認してください。

▼ runacct を再起動する方法

1. lastdate ファイルと lock* ファイル **(**もしあれば**)** を削除します。

```
$ cd /var/adm/acct/nite
$ rm lastdate lock*
```

2. runacct プログラムを再起動します。

```
$ runacct MMDD [state] 2> /var/adm/acct/nite/fd2log &
```

MMDD	月日を数値で指定する
state	runacct 処理を開始させたい状態または開始点を指定する

システムアカウンティングの停止と無効

システムアカウンティングは、一時的に停止することも、永久に無効にすることもできます。

▼ 一時的にシステムアカウンティングを停止する方法

1. スーパーユーザーになります。

2. 適切な行をコメントアウトすることによって、プログラム ckpacct、runacct、および monacct の実行が停止するように、ユーザー adm の crontab ファイルを変更します。

```
# EDITOR=vi; export EDITOR
# crontab -e adm
#0 * * * * /usr/lib/acct/ckpacct
#30 2 * * * /usr/lib/acct/runacct 2> /var/adm/acct/nite/fd2log
#30 7 1 * * /usr/lib/acct/monacct
```

3. 適切な行をコメントアウトすることによって、プログラム dodisk の実行が停止するように、ユーザー root の crontab ファイルを変更します。

```
# crontab -e
#30 22 * * 4 /usr/lib/acct/dodisk
```

4. 次のように入力して、アカウンティングプログラムを停止します。

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

システムアカウンティングプログラムを再び有効にするには、新たに追加したコメント記号を crontab ファイルから削除して、次のように入力します。

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

▼ システムアカウンティングを永久に無効にする方法

1. スーパーユーザーになります。

2. ユーザー adm の crontab ファイルを編集して、プログラム ckpacct、runacct、および monacct 用のエントリを削除します。

```
# EDITOR=vi; export EDITOR
# crontab -e adm
```

3. ユーザー root の crontab ファイルを変更して、プログラム dodisk 用のエントリを削除します。

```
# crontab -e
```

4. 実行レベル **2** 用の起動スクリプトのリンクを取り外します。

```
# unlink /etc/rc2.d/S22acct
```

5. 実行レベル **0** 用の停止スクリプトのリンクを取り外します。

```
# unlink /etc/rc0.d/K22acct
```

6. 次のように入力して、アカウンティングプログラムを停止します。

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

第 32 章

システムアカウンティングの参照情報

この章のリファレンス情報の内容は、次のとおりです。

日次アカウンティングの種類

日次アカウンティングによって、「接続アカウンティング」、「プロセスアカウンティング」、「ディスクアカウンティング」、「料金計算」の 4 種類のアカウンティングを簡単に実行できます。

接続アカウンティング

接続アカウンティングでは、次のデータを調べることができます。

- 特定のユーザーがログインしていた時間
- tty 回線の利用状況
- システムのリブート回数
- アカウンティングソフトウェアが有効または無効に設定された頻度

これらの情報を提供するために、システムは期間調整、リブート回数、アカウンティングソフトウェアが有効または無効に設定された回数、実行レベルの変更回数、ユーザープロセス (login プロセスと init プロセス) の作成、プロセスの消滅などの情報のレコードを格納します。これらのレコード (date、init、login、ttymon、acctwtmp などのシステムプログラムの出力によって生成されます) は /var/adm/wtmpx ファイルに格納されます。wtmpx ファイルのエントリには、ユーザーのログイン名、装置名、プロセス ID、エントリタイプ、エントリがいつ作られたかを示すタイムスタンプなどの情報を含めることができます。

プロセスアカウンティング

プロセスアカウンティングでは、システムで実行される各プロセスに関する次のようなデータを追跡できます。

- プロセスを使用するユーザーおよびグループのそれぞれのユーザー ID とグループ ID
- プロセスの開始時刻と経過時間
- プロセスの CPU 時間 (ユーザー時間とシステム時間)
- 使用されるメモリーの量
- 実行されるコマンド
- プロセスを制御する tty

プロセスが終了するたびに、exit プログラムは上記のデータを収集し、/var/adm/pacct ファイルに書き込みます。

ディスクアカウンティング

ディスクアカウンティングでは、各ユーザーがディスク上にもっているファイルについて次のデータを収集し、それらの書式を指定できます。

- ユーザーの名前と ID
- ユーザーのファイルが使用しているブロック数

これらのデータはシェルスクリプト /usr/lib/acct/dodisk によって収集されますが、収集周期は /var/spool/cron/crontabs/root ファイルに追加する cron コマンドによって決定されます。dodisk は、acctdusg および diskusg コマンドを起動して、システム内の各ファイルについての情報を収集させます。

dodisk の設定の詳細は、564ページの「システムアカウンティングを設定する方法」を参照してください。

acctdusg(1M) コマンドは、すべてのディスクアカウンティング情報を収集します。起動されるたびに、このコマンドは最高 3000 ユーザー分の情報を処理できます。

注意 - dodisk(1M) を実行して収集された情報は、/var/adm/acct/nite/disktacct ファイルに格納されます。これらの情報は、次に dodisk を実行したときに上書きされます。したがって、dodisk は同じ日に 2 回以上実行しないでください。

ランダムなアクセスにより書き込まれ、その結果ディスク上で連続していないファイルに対して diskusg コマンドは、より多く課金します。これは、diskusg がファイルのサイズを判定するときにファイルの間接ブロックを読み取らないためです。このコマンドは、i ノードの *di_size* の値を調べてファイルのサイズを判定します。

ユーザー料金の計算

chargefee ユーティリティは、ユーザーに提供した特別なサービス (ファイルの復元など) に対する課金を、/var/adm/fee ファイルに格納します。このファイルの各エントリは、ユーザーのログイン名、ユーザー ID、および料金から構成されています。このファイルは、runacct プログラムによって毎日チェックされて、新しいエントリが全体のアカウンティングレコードにマージされます。chargefee を実行してユーザーに課金する命令については、567ページの「ユーザーに課金する方法」を参照してください。

日次アカウンティング機能の動作

次に、SunOS の日次アカウンティング機能がどのように動作するかを要約して示します。

1. システムをマルチユーザーモードに切り替えると、/usr/lib/acct/startup プログラムが実行されます。この startup プログラムは、それぞれアカウンティング機能を呼び出す他のプログラムを実行します。
2. acctwtmp プログラムは /var/adm/wtmpx に「ブート」レコードを追加します。このレコードには、システム名が wtmpx レコード内のログイン名として示されます。表 32–1 に、raw アカウンティングデータがどのように収集され、どこに格納されるかをまとめて示します。

表 32–1 raw アカウンティングデータ

/var/adm 内のファイル	情報	ファイルを書くプログラム	書式定義ヘッダ
wtmpx	接続セッション数	login, init	utmpx.h
	日付変更回数	date	
	リブート回数	acctwtmp	
	シャットダウン回数	shutacct シェル	
pacctn	プロセス数	カーネル (プロセス終了時)	acct.h
		turnacct switch (古いファイルの内容が 500 ブロックに達すると、新しいファイルが作成される)	
fee	特別料金	chargefee	acct.h
acct/nite/disktacct	使用ディスク領域	dodisk	tacct.h

3. turnacct プログラムが –on オプションで起動されて、プロセスアカウンティングを開始します。turnacct は、特に /var/adm/pacct 引数を使用して accton プログラムを実行します。
4. remove シェルスクリプトが、runacct によって sum ディレクトリに保存されている pacct および wtmpx ファイルを「整理」します。

5. login および init プログラムが、/var/adm/wtmpx にレコードを書き込み、接続セッションを記録します。すべての日付変更 (引数を指定して date を使用) も /var/adm/wtmpx に書き込まれます。acctwtmp を使用したリブートとシャットダウンも /var/adm/wtmpx に記録されます。
6. プロセスが終了すると、カーネルが /var/adm/pacct ファイルにプロセスごとに 1 レコードを acct.h 形式で書き込みます。

   cron は、1 時間ごとに ckpacct プログラムを実行して /var/adm/pacct のサイズを調べます。このファイルが 500 ブロック (デフォルト) よりも大きくなった場合は、turnacct による切り替えが実行されます。(このプログラムは今までの pacct ファイルを他に移して新しいファイルを作成します。) pacct ファイルを小さく分けることの利点は、それらのアカウントレコードを処理するときに障害が発生し、runacct を起動し直そうとしたときに明らかになります。
7. runacct が毎晩 cron によって実行されます。runacct は /var/adm/pacct*n*、/var/adm/wtmpx、/var/adm/fee、/var/adm/acct/nite/disktacct などのアカウンティングファイルを処理して、ログイン別のコマンド要約と利用状況要約を生成します。
8. /usr/lib/acct/prdaily プログラムが runacct によって 1 日に 1 回実行され、runacct が収集した日次アカウンティング情報 (ASCII 形式) を /var/adm/acct/sum/rprt.*MMDD* に書き込みます。
9. monacct プログラムが月に 1 回 (または毎会計期の終わりなど、ユーザーが決めた周期で) 実行されます。monacct プログラムは、sum ディレクトリに格納されているデータに基づいてレポートを作成します。これらのデータは runacct によって毎日更新されています。このレポートを作成後、monacct は sum ディレクトリを「整理」して、新しい runacct データを格納するためのファイルを準備します。

システムがシャットダウンしたときの動作

shutdown を使用してシステムをシャットダウンした場合は、shutacct プログラムが自動的に実行されます。shutacct プログラムは /var/adm/wtmpx に理由レコードを書き、アカウンティングプロセスを無効に設定します。

アカウンティングレポート

この節では、アカウンティングソフトウェアによって生成される様々なレポートについて説明します。

日次アカウンティングレポート

runacct(1M) シェルスクリプトは、呼び出されるたびに基本的な 4 種類のレポートを生成します。これらのレポートは、接続アカウンティング、毎日のログイン別利用状況、日次および月次合計によって報告されるコマンド利用状況の 3 種類を対象とするレポートと、ユーザーの最後のログイン時刻のレポートです。4 つの基本レポートは次のとおりです。

表 32–2 日次アカウンティングレポート

レポートの種類	説明
日次レポート	tty 番号別の回線の利用状況を示します
日次利用状況レポート	ユーザー別のシステム資源の利用状況を示します。UID 順に表示されます
日次コマンド要約	コマンド別のシステム資源の利用状況を示します。使用したメモリーの大きさの降順に、つまりメモリーを最も多く使用したコマンドから先に表示されます。これらと同じ情報が月次コマンド要約では 1 ヶ月分について報告されます
最終ログインレポート	各ユーザーが最後にログインした日付を示します。日付順に表示されます

日次レポート

このレポートは、使用された各端末回線に関する情報を示します。次に例を示します。

```
Jul  7 02:30:02 1999  DAILY REPORT FOR mercury Page 1


from Wed Jul 07 02:30:02 1999
to   Thu Jul 08 02:30:02 1999
1       system boot
1       run-level 3
1       acctg on
1       runacct
1       acctcon

TOTAL DURATION IS 1384 MINUTES
LINE         MINUTES  PERCENT  # SESS  # ON  # OFF
```

(続く)

```
/dev/pts/5    0         0         0         0      0
/dev/pts/6    0         0         0         0      1
/dev/pts/7    0         0         0         0      0
console       1337      97        1         1      1
pts/3         0         0         0         0      1
pts/4         0         0         0         0      1
pts/5         3         0         2         2      3
pts/6         232       17        5         5      5
pts/7         54        4         1         1      2
pts/8         0         0         0         0      1
pts/9         0         0         0         0      1
TOTALS        1625      --        9         9      16
```

from および to 行はこのレポートで反映される時間帯、つまりこの前のアカウンティングレポートが生成されてから現在のアカウンティングレポートが生成されるまでの時間を指定します。その次はシステムのリブート、シャットダウン、電源異常からの回復と、acctwtmp プログラムによって /var/adm/wtmpx にダンプされたその他のすべてのレコードです。詳細は、acct(1M) のマニュアルページを参照してください。

このレポートの第 2 部は回線利用状況の内訳です。TOTAL DURATION は、システムがどれだけの時間マルチユーザーモード (端末回線を通してアクセス可能です) であったかを示します。この部分を構成しているカラムを表 32–3 で説明します。

表 32–3 日次レポート

カラム	説明
LINE	回線。端末回線またはアクセスポート
MINUTES	回線使用分。アカウンティング期間を通じてこの回線が使用中であった合計分
PERCENT	回線利用率。この回線が使用中であった MINUTES の合計値を TOTAL DURATION で割ったパーセント値
# SESS	セッション数。このポートが login セッション向けにアクセスされた回数

表 32–3 日次レポート 続く

カラム	説明
# ON	ログイン回数。SESS と同じ。(このカラムにはそれ以上の意味はなし。ポートがユーザーのログインに使用された回数を表示する)
# OFF	ログアウト回数。このカラムは、この回線でユーザーがログアウトした回数と発生した割り込みを表す。割り込みは一般にシステムがマルチユーザーモードにされてから ttymon が初めて起動されたときに発生する。# OFF が大きな割合で # ON を上回る場合は、マルチプレクサ、モデム、ケーブルに障害があるか、どこかに接触の問題がある可能性がある。一番考えられる原因は、マルチプレクサからのケーブルの接続が外れたままになっていることである

マシンの稼動中は、/var/adm/wtmpx ファイルから接続アカウンティングが準備されるので、このファイルを監視する必要があります。wtmpx ファイルが急速に大きくなる場合は、acctcon -l *file* < /var/adm/wtmpx を実行してどの tty 回線の使用頻度が最も大きいかを調べてください。割り込みが頻繁に発生する場合は、全般的なシステムの性能が影響を受けることになります。さらに、wtmpx が壊れることもあります。この問題を解決するには、568ページの「wtmpx ファイルを修復する方法」を参照してください。

日次利用状況レポート

このレポートは、システム資源の利用状況のユーザー別の内訳を示します。次に例を示します。

```
Jul  7 02:30:02 1999  DAILY USAGE REPORT FOR mercury Page 1

      LOGIN    CPU (MINS) KCORE-MINS  CONNECT (MINS) DISK  # OF  # OF # DISK FEE
UID   NAME   PRIME NPRIME PRIME NPRIME PRIME NPRIME  BLOCKS PROCS SESS SAMPLES
0     TOTAL  1     1      2017  717    785   840    660361  1067  9     7    20
0     root   1     1      1833  499    550   840    400443  408   2     1    0
1     daemon 0     0      0     0      0     0      400     0     0     1    0
2     bin    0     0      0     0      0     0      253942  0     0     1    0
3     sys    0     0      0     0      0     0      2       0     0     1    0
4     adm    0     0      46    83     0     0      104     280   0     1    0
5     uucp   0     0      74    133    0     0      1672    316   0     1    0
71    lp     0     0      0     2      0     0      3798    1     0     1    0
8198  ksm    0     0      8     0      0     0      0       6     1     0    0
52171 pjm    0     0      56    0      234   0      0       56    6     0    20
```

日次利用状況レポートで示される各データを表 32–4 で説明します。

表 32–4 日次利用状況レポート

カラム	説明
UID	ユーザー ID 番号
LOGIN NAME	ユーザーのログイン名。複数のログイン名をもつユーザーを識別する
CPU-MINS	CPU 使用時間 (分単位)。ユーザーのプロセスが CPU を使用した時間を表す。このカテゴリの情報は、PRIME (プライムタイム時間帯) と NPRIME (プライムタイム時間帯外) に分けられる。アカウンティングシステムのこれらのデータのバージョンは、/etc/acct/holidays ファイルに格納されている
KCORE-MINS	プロセスが実行中に使用する累積メモリー量を表す。表示される値は、毎分当たりに使用される K バイトメモリーセグメント数を表す。この計量値も PRIME と NPRIME に分けられる
CONNECT-MINS	ユーザーがシステムにログインしていた時間を表す。「実時間」とも呼ぶ。PRIME と NPRIME に分けられる。たとえば、この時間の値が大きく # OF PROCS の数値が小さい場合は、ログインの所有者がまず朝にログインし、その後はその日の終わりまで端末にほとんど触れていないと考えられる
DISK BLOCKS	ディスクブロック数。ディスクアカウンティングプログラムが実行された後は、出力が合計アカウンティングレコード (daytacct) にマージされ、このカラムに表示される。このディスクアカウンティングは acctdusg プログラムによってなされる。アカウンティングの目的ではブロックは 512 バイト
# OF PROCS	ユーザーが起動したプロセス数を表す。数値が大きい場合は、ユーザーのシェルプロシージャが制御できなくなった可能性がある
# OF SESS	ユーザーがシステムにログインした回数
# DISK SAMPLES	平均ディスクブロック数 (DISK BLOCKS) を得るためにディスクアカウンティングが何回実行されたかを示す
FEE	chargefee によってユーザーに課金される累積合計額を表す。使用されない場合が多い

日次コマンド要約

このレポートはコマンド別のシステム資源の利用状況を示します。このレポートでは、最も使用率の高いコマンドがわかり、それらコマンドがどのようにシステム資源を利用しているかに基づいて、どのようにしたらシステムの最適チューニングが可能かを知ることができます。日次レポートも月次レポートも見た目には同じですが、日次要約レポートは当日だけについてのレポートであるのに対して、月次要約レポートは、会計期の初めから当日までについてのレポートです。つまり、月次レポートは、monacct が最後に実行されたときからの累積データの累積要約を表します。

これらのレポートは TOTAL KCOREMIN によってソートされます。TOTAL KCOREMIN は任意の基準ですが、システムでのドレーンの計算にはすぐれた指標です。

次に日次コマンド要約レポートの例を示します。

```
Jul  7 02:30:02 1999  DAILY COMMAND SUMMARY Page 1

                                    TOTAL COMMAND SUMMARY
COMMAND NUMBER    TOTAL     TOTAL     TOTAL    MEAN     MEAN     HOG    CHARS  BLOCKS
NAME     CMDS  KCOREMIN  CPU-MIN  REAL-MIN  SIZE-K  CPU-MIN FACTOR TRNSFD   READ

TOTALS   1067   2730.99    2.01    1649.38 1361.41   0.00  0.00    6253571  2305
sendmail   28   1085.87    0.05       0.24  23865.20  0.00  0.19     101544    39
admintoo    3    397.68    0.12    1132.96  3443.12   0.04  0.00     680220    83
sh        166    204.78    0.31     161.13  651.80    0.00  0.00     598158    20
nroff      12    167.17    0.14       0.24  1205.55   0.01  0.59     709048    22
find       10    151.27    0.27       2.72  563.40    0.03  0.10     877971  1580
acctdusg    3     87.40    0.13       2.74  698.29    0.04  0.05     883845   203
lp         10     74.29    0.05       0.22  1397.38   0.01  0.24     136460    57
expr       20     67.48    0.02       0.06  3213.24   0.00  0.34       6380     1
mail.loc    3     65.83    0.01       0.04  11285.60  0.00  0.15      24709    15
cmdtool     1     37.65    0.02      20.13  2091.56   0.02  0.00     151296     1
uudemon.  105     37.38    0.09       0.32  435.46    0.00  0.27      62130    17
csh         6     35.17    0.05      57.28  756.30    0.01  0.00     209560    13
col        12     31.12    0.06       0.26  523.00    0.00  0.23     309932     0
ntpdate    22     27.55    0.05      11.18  599.00    0.00  0.00      22419     0
uuxqt      44     18.66    0.04       0.06  417.79    0.00  0.74      32604     3
man        12     15.11    0.03       7.05  503.67    0.00  0.00      85266    47
.
.
.
```

表 32–5 で、日次コマンド要約レポートに示されるデータをカラム別に説明します。

表 32–5　日次コマンド要約レポート

カラム	説明
COMMAND NAME	コマンド名。プロセスアカウンティングシステムではオブジェクトモジュールしか報告されないので、シェルプロシージャはすべて sh という名前で取り扱われる。 a.out または core と呼ばれるプログラム、またはその他の、適切とは思われない名前のプログラムの使用頻度を監視すると良い。acctcom を使用して、名前に疑問があるコマンドを誰が実行したか、スーパーユーザー特権が使用されたかどうかを知ることができる
NUMBER CMNDS	プライムタイム時間帯にこのコマンドが呼び出された回数
TOTAL KCOREMIN	実行時の毎分当たりにプロセスが使用した K バイトメモリーセグメント数という計量値の累積合計
TOTAL CPU-MIN:	このプログラムのプライムタイム時間帯の累積合計処理時間
TOTAL REAL-MIN	このプログラムのプライムタイム時間帯の累積合計実時間 (壁掛け時計)。分単位
MEAN SIZE-K	NUMBER CMDS で表される呼び出し回数に対する TOTAL KCOREMIN の平均
MEAN CPU-MIN	NUMBER CMDS に対する TOTAL CPU-MIN の平均
HOG FACTOR	合計 CPU 時間を経過時間で割った値。システム利用可能時間とシステム使用時間との比であり、プロセスがその実行中に消費する合計利用可能 CPU 時間の相対値を示す
CHARS TRNSFD	読み取りおよび書き込みシステムコールによってプッシュされた文字数の合計カウント。オーバフローのために負の値になることがある
BLOCKS READ	プロセスが実行した物理ブロックの読み取りおよび書き込みの合計カウント

月次コマンド要約

この月次コマンド要約は、次のただ 1 点の相違を除いて日次コマンド要約と同じです。つまり、月次コマンド要約は monacct が最後に起動されたときからの累積合計を示します。次に例を示します。

```
Jul  6 02:30:03 1999  MONTHLY TOTAL COMMAND SUMMARY Page 1

                                    TOTAL COMMAND SUMMARY
COMMAND NUMBER     TOTAL    TOTAL     TOTAL    MEAN   MEAN   HOG    CHARS  BLOCKS
NAME      CMDS  KCOREMIN   CPU-MIN REAL-MIN SIZE-K CPU-MIN FACTOR TRNSFD   READ

TOTALS     771    483.70   0.94    8984.09  515.12   0.00  0.00   2248299   179

sh         105    155.41   0.23     429.58  667.94   0.00  0.00    491870     1
uudemon.    85     29.39   0.07       0.29  434.28   0.00  0.23     49630    14
acctcms      5     27.21   0.04       0.04  752.41   0.01  0.90    218880     1
ntpdate     17     21.30   0.04      14.10  605.73   0.00  0.00     18192     0
dtpad        1     19.69   0.01      10.87 2072.70   0.01  0.00     46992     8
sendmail    17     16.75   0.02       0.02  859.04   0.00  0.91      1965     0
acctprc      1     14.92   0.03       0.03  552.69   0.03  0.95    115584     0
uuxqt       34     14.78   0.03       0.04  426.29   0.00  0.92     25194     0
uusched     34     10.96   0.03       0.03  363.25   0.00  0.91     25194     0
sed         40     10.15   0.03       0.09  315.50   0.00  0.36     64162     2
man          5     10.08   0.02      57.58  555.05   0.00  0.00     25773     2
getent       1      7.68   0.01       0.02  921.60   0.01  0.40     20136     0
in.rlogi     5      7.65   0.01    4331.67  611.73   0.00  0.00     87440     0
cp          37      7.28   0.03       0.05  280.08   0.00  0.50      1739    36
date        27      7.24   0.02       0.03  329.12   0.00  0.65     23443     1
ls          15      7.05   0.01       0.02  503.33   0.00  0.79     14123     0
awk         19      6.94   0.02       0.06  372.04   0.00  0.32       666     0
rm          29      6.83   0.02       0.04  301.32   0.00  0.60      2348    17
```

各データ項目については、582ページの「日次コマンド要約」を参照してください。

最終ログインレポート

このレポートは、特定のログインが最後に使用された日付を示します。この情報を使用して、使用されていないログインやログインディレクトリを見つけることができます。それらのログインやログインディレクトリは保存して削除できます。次に例を示します。

```
Jul  7 02:30:03 1999  LAST LOGIN Page 1
.
.
.
00-00-00  arimmer       00-00-00  lister        99-06-27  pjm
00-00-00  reception     00-00-00  smithe        99-06-27  ksm
00-00-00  release       00-00-00  smsc          99-06-27  root
00-00-00  resch         00-00-00  datab
```

acctcom による pacct ファイルの確認

/var/adm/pacct*n* ファイル、または acct.h 形式の任意のファイルの内容は、acctcom プログラムを使用していつでも調べることができます。このコマンド

を実行するときに、ファイルも標準入力も指定しなければ、acctcom は pacct ファイルを読み取ります。acctcom は、終了したプロセスに関する情報を示します (実行中のプロセスは ps コマンドで調べることができます)。acctcom のデフォルト出力は次に示す情報を示します。

- コマンド名 (スーパーユーザー特権を使用して実行された場合は # 記号)
- ユーザー
- tty 名 (未知の場合は ? として表示)
- 開始時刻
- 終了時刻
- 実時間 (秒単位)
- CPU 時間 (秒単位)
- 平均サイズ (K バイト単位)

acctcom にオプションを使用することにより、次の情報を得ることができます。

- fork/exec フラグの状態 (exec を使用しない fork の場合は 1)
- システム終了状態
- hog 係数
- 合計 kcore 分
- CPU 係数
- 転送文字数
- 読み取りブロック数

表 32–6 で acctcom のオプションを説明します。

表 32–6 acctcom のオプション

オプション	説明
–a	選択したプロセスに関する特定の平均統計を表示する (統計は出力が記録された後に表示される)
–b	ファイルを逆読みし、最後のコマンドから先に表示する (標準入力の読み込みには関係しない)
–f	fork/exec フラグおよびシステム終了状態カラムを出力する (出力は 8 進数)

表 32–6 acctcom のオプション 続く

オプション	説明
–h	平均メモリーサイズに代えて hog 係数を表示する。hog 係数は、経過時間とプロセスが実行中に消費した合計 CPU 利用可能時間との比 (合計 CPU 利用時間/経過時間)
–i	入出力カウントを含むカラムを出力する
–k	メモリーサイズの代わりに、キロバイト/分ごとのコアサイズの合計値を表示する
–m	平均コアサイズ (デフォルト) を表示する
–q	平均統計だけを出力する。出力レコードは出力しない
–r	CPU 係数 (システム使用時間 / (システム使用時間 + ユーザー使用時間)) を表示する
–t	システムおよびユーザー CPU 時間を表示する
–v	出力からカラム見出しを除外する
–C *sec*	合計 (システム + ユーザー) CPU 時間が *sec* 秒を超えたプロセスだけを表示する
–e *time*	*time* 以前に存在したプロセスを *hr[:min[:sec]]* の書式で表示する
–E *time*	*time* 以前に開始されたプロセスを *hr[:min[:sec]]* の書式で表示する。同じ *time* を –S と –E の両方に使用すれば、そのときに存在していたプロセスを表示する
–g *group*	*group* に属しているプロセスだけを表示する
–H *factor*	*factor* を超えるプロセスだけを表示する。ただし、*factor* は「hog 係数」(–h オプションを参照)
–I *chars*	*chars* によって指定されるカットオフ数を超える文字数を転送したプロセスだけを表示する
–l *line*	端末 /dev/*line* に属しているプロセスだけを表示する

表 32–6　acctcom のオプション　続く

オプション	説明
–n *pattern*	*pattern* 「+」が 1 回以上現れることを意味する以外は、一般的な 正規表現に一致するコマンドだけを表示する
–o *ofile*	レコードを出力しないで、レコードを acct.h 形式で *ofile* にコピーする
–O *sec*	CPU システム時間が *sec* 秒を超えるプロセスだけを表示する
–s *time*	*time* 以後に存在したプロセスを *hr*[:*min*[:*sec*]] の書式で表示する
–S *time*	*time* 以後に開始されたプロセスを *hr*[:*min*[:*sec*]] の書式で表示する
–u *user*	*user* に属しているプロセスだけを表示する

runacct プログラム

メインの日次アカウンティングシェルスクリプトである runacct は、通常はプライムタイムつまり最も忙しい時間帯を避けて cron により起動されます。この runacct シェルスクリプトは、接続、料金、ディスク、プロセス用の各アカウンティングファイルを処理します。さらに、課金目的で prdaily と monacct が使用する日次および累積要約ファイルも準備します。

runacct シェルスクリプトは、エラーが発生した場合、ファイルを壊さないよう注意します。一連の保護機構により、エラーを認識し、インテリジェント診断を提供し、最小のユーザー介入で runacct が起動し直せるように処理を完了しようとします。runacct は、説明メッセージを active というファイルに書き込み、進捗状況を記録します。(runacct が使用するファイルは、特にことわりのないかぎり、ディレクトリ /var/adm/acct/nite にあります。) runacct の実行中、すべての診断出力は fd2log に書き込まれます。

runacct は起動されると lock と lock1 というファイルを作成します。これらのファイルは、runacct が同時に実行されるのを防ぎます。runacct プログラムは、起動時にこれらのファイルが存在すれば、エラーメッセージを出力します。lastdate ファイルには runacct が最後に起動されたときの月日が入っており、このファイルは runacct を 1 日に 1 回しか実行できないようにするために使

用されます。runacct がエラーを検出した場合は、エラーメッセージがコンソールに出力され、メールが root と adm に送られ、ロックが解除され、診断ファイルが保存され、実行が終了されます。runacct を再び起動する手順については、570 ページの「runacct を再起動する方法」を参照してください。

runacct を再起動可能にするために、処理は再入可能な別々の状態に分割されます。statefile というファイルを使用して、最後に終了した状態が追跡されます。各状態が終了するたびに、statefile は更新されて次の状態に反映されます。1 つの状態の処理が終了すると、statefile が読み取られて次の状態が処理されます。runacct は、CLEANUP 状態に達すると、ロックを解除して実行を終了します。各状態は表 32–7 に示すように実行されます。

表 32–7 runacct 状態

状態	説明
SETUP	turnacct switch コマンドが実行されて新しい pacct ファイルが作成される。/var/adm/pacct*n* 内の (pacct ファイル以外の) プロセスアカウンティングファイルが /var/adm/Spacctn.*MMDD* に移される。/var/adm/wtmpx ファイルは (最後に現在時のレコードが追加されて) /var/adm/acct/nite/wtmp.*MMDD* に移され、新しい /var/adm/wtmp が作成される。closewtmp と utmp2wtmp は、現在ログインしているユーザーに課金処理を行うため、wtmp.*MMDD* と新しい wtmpx にレコードを追加する
WTMPFIX	wtmpfix プログラムが nite ディレクトリ内の wtmp.*MMDD* ファイルを調べて誤りがないか確認する。データの変更によっては、acctcon を失敗させることがあるので、wtmpx ファイルに日付変更のレコードが現れた場合は、wtmpfix は関係するタイムスタンプを調整しようとする。さらに、wtmpx ファイルからのエントリが壊れていた場合、壊れたエントリをすべて削除する。問題が解決された後のバージョンの wtmp.*MMDD* が tmpwtmp に書き込まれる
CONNECT	acctcon プログラムが使用されて、ctacct.*MMDD* ファイルに接続アカウントレコードが記録される。これらのレコードは tacct.h 形式になっている。acctcon は、さらに lineuse および reboots ファイルを作成する。reboots ファイルは、wtmpx ファイルで見つかったすべてのブートレコードを記録する
PROCESS	acctprc プログラムを使用して、プロセス課金ファイル /var/adm/Spacct*n*.*MMDD* が ptacct*n*.*MMDD* の合計アカウントレコードに変換される。runacct が失敗した場合に、Spacct ファイルが処理されないように、Spacct ファイルと ptacct ファイルは番号で照合される

表 32–7 runacct 状態 続く

状態	説明
MERGE	acctmerg プログラムが、プロセスアカウントレコードを接続アカウントレコードとマージして daytacct を作成する
FEES	acctmerg プログラムが、fee ファイルからの ASCII tacct レコードを daytacct にマージする
DISK	dodisk プロシージャが実行されて disktacct ファイルが生成されている場合は、DISK プログラムがこのファイルを daytacct にマージし、disktacct を /tmp/disktacct.*MMDD* に移す
MERGETACCT	acctmerg プログラムが、daytacct を累積合計課金ファイルである sum/tacct とマージする。daytacct が毎日 sum/tacct.*MMDD* に保存される。したがって、sum/tacct は壊れたり失われたりしても、作成し直すことができる
CMS	acctcms プログラムが数回実行される。acctcms は、まず Spacct*n* ファイルを使用してコマンド要約を生成し、sum/daycms に書き込む。次に、sum/daycms を累積コマンド要約ファイル sum/cms とマージする。そして、最後に、sum/daycms および sum/cms ファイルからそれぞれ ASCII のコマンド要約ファイル nite/daycms と nite/cms を生成する。lastlogin プログラムを使用してログファイル /var/adm/acct/sum/loginlog が作成される。これは、各ユーザーが最後にログインした時刻を示すレポートである。(runacct が真夜中を過ぎてから実行された場合は、いずれかのユーザーが最後にログインした時刻を示す日付が 1 日分狂うことになる)
USEREXIT	インストールに依存しない任意の (ローカル) 課金プログラムをこの時点で取り入れることができる。runacct はそのプログラムを /usr/lib/acct/runacct.local と想定する
CLEANUP	一時ファイルが整理され、prdaily が実行され、その出力が sum/rpt.*MMDD* に保存され、ロックが解除され終了する

注意 - runacct を CLEANUP 状態で起動し直すときは、最後の ptacct ファイルが不完全であるため、このファイルを削除してください。

アカウンティングファイル

/var/adm ディレクトリ構造は、使用中のデータ収集ファイルを含みます。

表 32–8 に、/var/adm ディレクトリにあるアカウンティング関連のファイルを示します。

表 32–8 /var/adm ディレクトリ内のファイル

ファイル	説明
dtmp	acctdusg プログラムからの出力
fee	chargefee プログラムからの出力。ASCII tacct レコード
pacct	有効なプロセスアカウンティングファイル
pacct*n*	turnacct を使用して切り替えられたプロセスアカウンティングファイル
Spacct*n.MMDD*	runacct の実行中に生成された *MMDD* 日付のプロセスアカウンティングファイル

/var/adm/acct ディレクトリには、nite、sum、fiscal の各ディレクトリが設けられ、それぞれに実際のデータ収集ファイルが格納されます。たとえば、nite ディレクトリは runacct プロシージャが毎日繰り返して使用するファイルを格納しています。表 32–9 で、/var/adm/acct/nite ディレクトリ内の各ファイルを簡単に説明します。

表 32–9 /var/adm/acct/nite ディレクトリ内のファイル

ファイル	説明
active	runacct が進捗状況の記録用、警告メッセージ、エラーメッセージの出力用として使用する
active*MMDD*	runacct がエラーを検出した後の active と同じ

表 32–9　/var/adm/acct/nite ディレクトリ内のファイル　続く

ファイル	説明
cms	prdaily が使用する ASCII の合計コマンド要約
ctacct.*MMDD*	tacct.h 形式の接続アカウンティングレコード
ctmp	acctcon1 プログラムの出力。ctmp.h 形式の接続セッションレコード (acctcon1 と acctcon2 は互換性を保証するために用意)
daycms	prdaily が使用する ASCII 日次コマンド要約
daytacct	tacct.h 形式の 1 日分の合計アカウンティングレコード
disktacct	tacct.h 形式のディスクアカウンティングレコード。dodisk プロシージャが作成する
fd2log	runacct の実行中の診断出力
lastdate	runacct が最後に実行された日 (date +%m%d 書式)
lock	runacct の逐次使用の制御用に使用
lineuse	prdaily が使用する tty 回線利用状況レポート
log	acctcon からの診断出力
log.*MMDD*	runacct がエラーを検出後の log と同じ
owtmp	前日の wtmpx ファイル
reboots	wtmpx からの開始および終了日付とリブートのリスト
statefile	runacct の実行中の現在状態の記録用に使用
tmpwtmp	wtmpfix が修復した結果の wtmpx ファイル
wtmperror	wtmpfix エラーメッセージの格納用の場所

表 32–9 /var/adm/acct/nite ディレクトリ内のファイル 続く

ファイル	説明
wtmperror.*MMDD*	runacct がエラーを検出した後の wtmperror と同じ
wtmp.*MMDD*	wtmpx ファイルの runacct 用コピー

sum ディレクトリは、runacct が更新し、monacct が使用する累積要約ファイルを格納します。表 32–10 に、/var/adm/acct/sum ディレクトリ内のファイルを簡単に説明します。

表 32–10 /var/adm/acct/sum ディレクトリ内のファイル

ファイル	説明
cms	内部要約書式による、会計期の合計コマンド要約ファイル
cmsprev	最新の更新がなされていないコマンド要約ファイル
daycms	内部要約書式による、当日の利用状況を表すコマンド要約ファイル
loginlog	各ユーザーが最後にログインした日付のレコード。lastlogin によって作成され、prdaily プログラム内で使用される
rprt.*MMDD*	prdaily プログラムの保存出力
tacct	会計期の累積合計アカウンティングファイル
tacctprev	最新の更新がない点を除いて tacct と同じ
tacct.*MMDD*	*MMDD* 日付分の合計アカウンティングファイル

fiscal ディレクトリは monacct が作成する定期的要約ファイルを格納します。表 32–11 で、/var/adm/acct/fiscal ディレクトリ内の各ファイルを説明します。

表 32–11 /var/adm/acct/fiscal ディレクトリ内のファイル

ファイル	説明
cms*n*	内部要約書式の、会計期 *n* の合計コマンド要約ファイル
fiscrpt*n*	会計期 *n* の rprt*n* と同じレポート
tacct*n*	会計期 *n* の合計アカウンティングファイル

runacct が生成するファイル

次に示す runacct により生成されるファイル (/var/adm/acct にあります) には特に注意する必要があります。

表 32–12 runacct が生成するファイル

ファイル	説明
nite/lineuse	runacct は acctcon を呼び出して、/var/adm/acct/nite/tmpwtmp から端末回線の利用状況に関するデータを収集し、それらのデータを /var/adm/acct/nite/lineuse に書き込む。prdaily はこれらのデータを使用して回線利用状況を報告する。このレポートは特に不良回線の検出に有効となる。ログアウト回数とログイン回数との比率が 3:1 を超える場合は、回線に障害がある確率が大きい
nite/daytacct	tacct.h 形式の当日の合計課金ファイル
sum/tacct	このファイルは、毎日の nite/daytacct の累積であり、課金の目的に使用できる。このファイルは、毎月または毎会計期ごとに monacct プロシージャによって新たに累積が開始される
sum/daycms	runacct は acctcms を呼び出して、当日に使用されたコマンドに関するデータを処理する。これらの情報は /var/adm/acct/sum/daycms に格納される。このファイルの内容は毎日のコマンド要約。このファイルの ASCII バージョンは /var/adm/acct/nite/daycms
sum/cms	毎日のコマンド要約の累積。monacct が実行されることによって新たに累積が開始される。ASCII バージョンは nite/cms

表 32–12 runacct が生成するファイル 続く

ファイル	説明
sum/loginlog	runacct は lastlogin を呼び出して、/var/adm/acct/sum/loginlog のログインのうち最後にログインした日付を更新する。lastlogin は、さらにこのファイルから有効でなくなったログインを削除する
sum/rprt.*MMDD*	runacct が実行されるたびに、prdaily によって印刷された日次レポートのコピーが保存される

第 33 章

システム性能の管理

ここでは、システム性能を管理する方法について説明します。次の章が含まれます。

第 34 章	システム性能についての概要を説明します。
第 35 章	プロセスコマンドを使用してシステム性能を向上する手順を説明します。
第 36 章	vmstat、sar、およびディスク利用状態をチェックするコマンドを使用して性能を監視する手順を説明します。

第 34 章

システム性能の概要

コンピュータやネットワークの性能を十分に引き出すことは、システム管理における重要な作業です。この章では、コンピュータシステムの性能の維持と管理に影響する要素について簡単に説明します。

この章の内容は次のとおりです。

システム性能の管理に関する新機能

この節では、Solaris 8 リリースのシステム性能を管理するための新しい機能について説明します。

SPARC: busstat

新しいシステム監視ツール `busstat` では、システムのバスに関連したハードウェア性能カウンタにコマンド行からアクセスできます。このツールを使用するとシステム全体のバス性能統計をシステムハードウェアから直接収集できます。現在サポートされているハードウェアは Sbus デバイス、AC デバイス、PCI デバイス

です。これらはすべて SPARC システムデバイスです。現在のところ IA デバイスはサポートされていません。

busstat コマンドでは、メモリーバンクの読み取りおよび書き込み数、クロックサイクル数、割り込み数、ストリーム DVMA 読み取りおよび書き込み転送数など、システム全体の統計を測定できます。

スーパーユーザーは、busstat を使用してこれらのカウンタを設定できます。しかし、他のユーザーは、スーパーユーザーが指定したカウンタしか読むことができません。

busstat コマンドでは、これらのハードウェア性能カウンタをサポートするシステムのデバイスだけが表示されます。サポートされるデバイスがシステムにないと、次のメッセージが表示されます。

```
busstat: No devices available in system.
```

この監視ツールの使用方法の詳細は、busstat(1M) のマニュアルページを参照してください。

cpustat コマンドと cputrack コマンド

新しい cpustat コマンドと cputrack コマンドを使用して、システムやプロセスの性能を監視できます。

cpustat コマンドはシステム全体の CPU 情報を収集します。このコマンドはスーパーユーザーしか実行できません。cputrack コマンドは、アプリケーションやプロセスの情報を表示する truss コマンドに似ています。このコマンドは通常のユーザーでも実行できます。

開発者は、cpustat コマンドの作成に使用されているのと同じライブラリ API を使用すればこのような監視ツールを独自に作成できます。

詳細は、cpustat(1M) と cputrack(1) のマニュアルページを参照してください。

prstat

prstat コマンドは、システムの動作中のプロセス情報を表示します。コマンドでは、どのプロセス、UID、CPU ID、またはプロセッサセットの情報を表示するのかを指定できます。デフォルトで prstat は、すべてのプロセスの情報を CPU 使用率順にソートして表示します。

prstat -m を指定すると、プロセスの詳しいアカウント情報が表示されます。この情報には、プロセスがシステムトラップ、テキストページフォルト、データページフォルトに消費した時間、および CPU を待っていた時間 (CPU 応答時間ともいう) がパーセントで示されます。

詳細は、prstat(1M) のマニュアルページを参照してください。

廃止された Interprocess Communications パラメータ

Solaris 8 では rmalloc(9F) の代わりに kmem_alloc(9F) を使用してメッセージテキストを割り当てるため、Interprocess Communications (IPC) Message 機能が以前よりも拡張できるようになりました。

そのためこのリリースでは、これまで文書化されていた、rmalloc ベース実装の影響である msginfo_msgssz と msginfo_msgseg 調整可能パラメタは廃止になりました。

システム性能についての参照先

システム性能を監視する手順については、次の各章を参照してください。

- 第 35 章
- 第 36 章

システム性能とシステム資源

コンピュータシステムの性能は、システムがその資源をどのように使用して割り当てるかによって左右されます。したがって、通常の条件下でどのように動作するかを知るために、システム性能を定期的に監視することが重要になります。期待できる性能についてよく把握し、問題が発生したときに分析できなければなりません。

性能に影響を及ぼすシステム資源は次のとおりです。

システム資源	説明
中央処理デバイス (CPU)	CPU は、命令をメモリーからフェッチして実行します。
入出力 (I/O) デバイス	I/O デバイスは、コンピュータとの間で情報をやりとりします。この種のデバイスには、端末とキーボード、ディスクドライブ、プリンタなどがあります。
メモリー	物理 (またはメイン) メモリーは、システム上のメモリー (RAM) の容量を示します。

コンピュータシステムの動作と性能に関する統計情報を表示するツールについては、第 36 章を参照してください。

性能の調整に関連する情報

性能の問題は多岐にわたる要素が含まれるため、ここではすべてを詳しく説明できません。Sun では、性能調整コースやオンライン性能調整情報を提供しています。さらに、性能の改善とシステムやネットワークの調整についてさまざまな側面を網羅したいくつかの書籍が出版されています。

目的	参照サイト
性能調整に関するクラス	`http://suned.sun.com`
オンランの性能調整情報	`http://www.sun.com/sun-on-net/performance`
『*Resource Management*』など、Sun Microsystems Press が出版する性能調整に関する書籍の注文	`http://www.sun.com/books/blueprints.series.html`

システムやネットワークの性能調整については、次の書籍を参照してください。

- 『*Resource Management*』Adrian Cockcroft、Evert Hoogendoorn、Enrique Vargas、Tom Bialaski 共著、Sun Microsystems Press 発行、ISBN 0-13-025855-5
- 『*Sun Performance and Tuning: SPARC and Solaris*』, Adrian Cockcroft 著, SunMicrosystems Press/PRT Prentice Hall 発行, ISBN 0-13-149642-3

- 『*System Performance Tuning*』, Mike Loukides 著, O'Reilly & Associates, Inc. 発行
- 『*Managing NFS and NIS*』, Hal Stern 著, O'Reilly & Associates, Inc. 発行

プロセスとシステムの性能

表 34–1 に、プロセスに関連する用語を示します。

表 34–1 プロセスに関連する用語

用語	説明
プロセス	実行中のプログラムの実体
軽量プロセス (LWP)	仮想 CPU または実行資源。LWP は、利用できる CPU 資源をスケジュールクラスと優先順位に基づいて使用するように、カーネルによってスケジュールされる。LWP には、メモリーに常駐する情報が入ったカーネルスレッドと、スワップ可能な情報が入った LWP が含まれる
アプリケーションスレッド	ユーザーのアドレス空間内で独立して実行できる別個のスタックを持った一連の命令。LWP の最上部で多重化できる

1 つのプロセスは、複数の LWP と複数のアプリケーションスレッドで構成できます。カーネルはカーネルスレッド構造をスケジュールします。この構造は、SunOS 環境内をスケジュールする実体です。表 34–2 に各種プロセス構造体を示します。

表 34–2 プロセス構造体

構造体	説明
`proc`	プロセス全体に関連し、メインメモリーに常駐しなければならない情報が入っている
`kthread`	1 つの LWP に関連し、メインメモリーに常駐しなければならない情報が入っている

表 34–2　プロセス構造体　続く

構造体	説明
user	スワップ可能な、プロセス単位の情報が入っている
klwp	スワップ可能な、LWP プロセス単位の情報が入っている

図 34–1 に、これらの構造体の関係を示します。

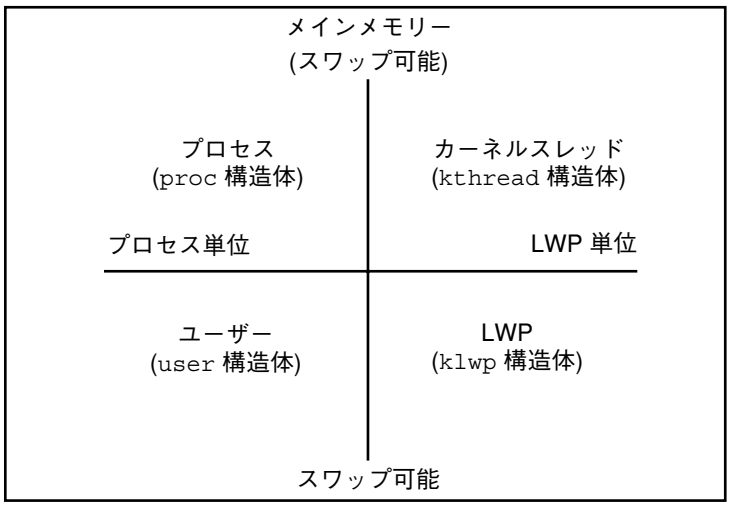


図 34–1　プロセス構造体

プロセス内のすべてのスレッドは、ほとんどのプロセス資源にアクセスできます。ほとんどすべてのプロセスの仮想メモリーが共有されます。あるスレッドが共有データを変更すると、その変更結果をプロセス内の他のスレッドが利用できます。

プロセス管理コマンド

表 34–3 に、プロセスを管理するためのコマンドとその説明を示します。

表 34–3　プロセスを管理するためのコマンド

使用するコマンド	目的
ps(1)、 pgrep(1)、 prstat(1M)	システム上のアクティブなプロセスの状態をチェックする。また、プロセスについての詳細な情報を表示する
dispadmin(1M)	デフォルトのスケジューリングポリシーをリストする
priocntl(1)	プロセスに優先順位クラスを割り当てて、プロセスの優先度を管理する
nice(1)	タイムシェアリングプロセスの優先度を変更する

別の機能を使うと、プロセッサセットに関してプロセスグループを制御できます。プロセッサセットを使用するということは、プロセスグループを 1 つのプロセサだけではなく、プロセッサグループにバインドできるということです。システム管理者は、/usr/sbin/psrset コマンドを使用することにより、プロセッサセットの作成と管理を制御できます。詳細は、psrset(1M) のマニュアルページを参照してください。

プロセスを管理するためのコマンドの詳細は、第 35 章を参照してください。

性能の監視

コンピュータの稼働中は、各種のシステム動作を追跡するためにオペレーティングシステムのカウンタが増分されます。追跡されるシステム動作は次のとおりです。

- 中央処理デバイス (CPU) の使用状況
- バッファーの使用状況
- ディスクとテープの入出力 (I/O) 動作
- 端末デバイスの動作
- システムコールの動作
- コンテキスト切替え
- ファイルアクセス

- 待ち行列の動作
- カーネルテーブル
- プロセス間通信
- ページング
- 空きメモリーとスワップ空間
- カーネルメモリー割り当て (KMA)

監視ツール

Solaris ソフトウェアには、システム性能を追跡できるように複数のツールが提供されています。次のような監視ツールがあります。

表 34–4 性能監視ツール

ツール	目的	参照先
`sar` ユーティリティと `sadc` ユーティリティ	システム動作データを収集および報告する	第 36 章
`ps` コマンドと `prstat` コマンド	活動中のコマンドについての情報を表示する	第 35 章
パフォーマンスメーター	ネットワーク上のシステムの状態をグラフィカルに表示する	第 36 章
`vmstat` コマンドと `iostat` コマンド	システム動作データの要約。仮想メモリーの統計、ディスクの使用率、CPU の動作など	第 36 章
`swap` コマンド	ユーザーのシステムで利用可能なスワップ領域についての情報を表示する	『*Solaris* のシステム管理 *(*第 *1* 巻*)*』の「追加スワップ空間の構成 (手順)」

表 34–4 性能監視ツール 続く

ツール	目的	参照先
`netstat` コマンドと `nfsstat` コマンド	ネットワーク性能についての情報を表示する	
Sun Enterprise SyMON	Sun Enterprise™ レベルのシステム上で、システム動作データを収集する	『*Sun Enterprise SyMON 2.0.1 Software User's Guide*』

第 35 章

プロセスの管理手順

この章では、システムプロセスを管理する手順について説明します。この章で説明する手順は次のとおりです。

プロセスに関する情報の表示

この節では、プロセス情報を管理するために使用されるコマンドについて説明します。

ps コマンド

ps コマンドを使用すると、システム上で活動中のプロセスの状態をチェックできます。また、プロセスについての技術的な情報も表示できます。このデータは、プロセスの優先順位をどのように設定するか判断するなどの管理作業に利用できます。

ps コマンドを使用すると、システム上で活動中のプロセスの状態をチェックできます。使用するオプションに応じて、次の情報が表示されます。

- プロセスの現在の状態
- プロセス ID
- 親プロセス ID
- ユーザー ID
- スケジューリングクラス
- 優先順位
- プロセスのアドレス
- 使用したメモリー
- 使用した CPU 時間

表 35–1 は、ps コマンドで表示されるフィールドの一部を示しています。どのフィールドが表示されるかは、選択するオプションによって異なります。使用可能なすべてのオプションについては、ps(1) のマニュアルページを参照してください。

表 35–1 ps により出力されるフィールド

フィールド	説明
UID	プロセス所有者の実効ユーザー ID
PID	プロセスの識別番号
PPID	親プロセスの識別番号
C	スケジューリングのためのプロセッサ使用率。このフィールドは –c オプションを使用すると表示されない

表 35–1 ps により出力されるフィールド 続く

フィールド	説明
CLS	プロセスが所属するスケジューリングクラス。リアルタイム、システム、またはタイムシェアリングのいずれか。このフィールドは、-c オプションを指定した場合にのみ表示される
PRI	カーネルスレッドのスケジューリング優先順位。番号が大きいほど優先順位が高い
NI	プロセスの nice 番号。これは、スケジュール優先順位に影響する。プロセスの nice 番号を大きくすると、その優先順位が下がる
ADDR	proc 構造体のアドレス
SZ	プロセスの仮想アドレスサイズ
WCHAN	プロセスが休眠中のイベントまたはロックのアドレス
STIME	プロセスの起動時刻 (時、分、秒)
TTY	プロセス (またはその親プロセス) が起動された端末。疑問符は、制御端末がないことを示す
TIME	プロセスの起動以降に使用した合計 CPU 時間
CMD	プロセスを生成したコマンド

▼ プロセスを表示する方法

システム上で実行中のすべてのプロセスを表示するには、ps コマンドを使用します。

```
$ ps [-ef]
```

ps　　ログインセッションに関連するプロセスのみを表示する

–ef　　システム上で実行中のすべてのプロセスに関する詳細情報を表示する

例 — プロセスを表示する

次の例は、オプションを指定しないときの ps コマンドからの出力を示します。

```
$ ps
  PID  TTY      TIME COMD
  1664 pts/4    0:06 csh
  2081 pts/4    0:00 ps
```

次の例は、ps –ef の出力を示します。この例は、システムのブート時に最初に実行されたプロセスが sched (スワッパ) であり、それに続いて init プロセス、pageout の順に実行されたことを示しています。

```
$ ps -ef
     UID   PID  PPID  C    STIME TTY      TIME CMD
    root     0     0  0   May 05 ?       0:04 sched
    root     1     0  0   May 05 ?      10:48 /etc/init -
    root     2     0  0   May 05 ?       0:00 pageout
    root     3     0  0   May 05 ?      43:21 fsflush
    root   238     1  0   May 05 ?       0:00 /usr/lib/saf/sac -t 300
    root   115     1  0   May 05 ?       0:10 /usr/sbin/rpcbind
    root   158     1  0   May 05 ?       0:00 /usr/lib/autofs/autom...
    root   134     1  0   May 05 ?       0:12 /usr/sbin/inetd -s
    root   107     1  0   May 05 ?      11:49 /usr/sbin/in.routed -q
    root   117     1  5   May 05 ?     899:32 /usr/sbin/keyserv
    root   125     1  0   May 05 ?       0:00 /usr/sbin/kerbd
    root   123     1  0   May 05 ?       4:17 /usr/sbin/nis_cachemgr
    root   137     1  0   May 05 ?       0:00 /usr/lib/nfs/statd
    root   139     1  0   May 05 ?       0:02 /usr/lib/nfs/lockd
    root   159     1 50   May 05 ?    8243:36 /usr/sbin/automount
    root   162     1  0   May 05 ?       0:07 /usr/sbin/syslogd
    root   181     1  0   May 05 ?       0:03 /usr/sbin/nscd...
    root   169     1  0   May 05 ?       5:09 /usr/sbin/cron
    root   191     1  0   May 05 ?       0:00 /usr/lib/lpsched
    root   210     1  0   May 05 ?       0:01 /usr/sbin/vold
    root   200     1  0   May 05 ?       0:08 /usr/lib/sendmail -bd -q1h
    root  4942     1  0   May 17 console 0:00 /usr/lib/saf/ttymon...
    root   208     1  0   May 05 ?       0:00 /usr/lib/utmpd
    root   241   238  0   May 05 ?       0:00 /usr/lib/saf/ttymon
    root  5748   134  0 17:09:49 ?       0:01 in.rlogind
    root  5750  5748  0 17:09:52 pts/0   0:00 -sh
    root  5770  5750  2 17:23:39 pts/0   0:00 ps -ef
```

/proc ファイルシステムとコマンド

/usr/proc/bin ディレクトリ内のプロセスツールを使用すると、/proc にあるプロセスに関して詳細情報を表示できます。/proc ディレクトリは、プロセスファイルシステム (PROCFS) とも呼ばれます。アクティブなプロセスのイメージは、そのプロセス ID 番号を使って /proc に格納されます。

プロセスツールは ps コマンドの一部のオプションに似ていますが、このツールから提供される出力の方が詳細です。一般に、プロセスツールには次の機能があります。

- fstat や fcntl 情報、作業ディレクトリ、親プロセスと子プロセスからなるツリーなど、プロセスに関する詳細情報を表示します。
- ユーザーが停止または再開できるように、プロセスに対する制御を提供します。

プロセスに関する情報の表示 (/proc ツール)

/usr/proc/bin に入っているプロセスツールコマンドを使用すると、活動中のプロセスに関する詳細な技術情報を表示できます。表 35–2 は、これらのプロセスツールを示しています。詳細は、proc(1) のマニュアルページを参照してください。

表 35–2 情報を表示する /usr/proc/bin のプロセスツール

プロセスツール	表示される内容
pcred	資格
pfiles	プロセス内で開いているファイルに関する fstat 情報と fcntl 情報
pflags	/proc 追跡フラグ、保留状態のシグナルと保持状態のシグナル、他の状態情報
pldd	プロセスにリンクされた動的ライブラリ
pmap	アドレス空間マップ
psig	シグナルの動作
pstack	16 進 + シンボリックスタックトレース

表 35–2　情報を表示する /usr/proc/bin のプロセスツール　続く

プロセスツール	表示される内容
ptime	microstate アカウンティングを使用するプロセス時間
ptree	プロセスが入っているプロセスツリー
pwait	プロセス終了後の状態情報
pwdx	プロセスの現在の作業ディレクトリ

注 - 長いコマンド名を入力しなくてもすむように、プロセスツールディレクトリを PATH 変数に追加してください。これにより、各ファイル名の最後の部分 (たとえば、/usr/proc/bin/pwdx ではなく pwdx) を入力するだけで、プロセスツールを実行できます。

▼ プロセスに関する情報を表示する方法

1. **(省略可能)** pgrep コマンドからの出力を使用して、詳細情報を表示したいプロセスの識別番号を調べます。

```
# pgrep process
```

process　詳細情報を表示したいプロセス名

プロセス識別番号は、出力の第 1 列目に表示されます。

2. 適切な /usr/proc/bin コマンドを使用して必要な情報を表示します。

```
# /usr/proc/bin/pcommand PID
```

pcommand　実行したいプロセスツールコマンド。表 35–2 を参照

PID　プロセスの識別番号

例 — プロセスに関する情報を表示する

次の例は、プロセスツールコマンドを使用して lpsched プロセスに関する詳細情報を表示する方法を示しています。まず、長いプロセスツールコマンドを入力しなくてもすむように、/usr/proc/bin パスが定義されています。次に、lpsched の識別番号が表示されています。最後に、3 つのプロセスツールコマンドからの出力が表示されています。

```
# PATH=$PATH:/usr/proc/bin
# export PATH
# ps -e | grep lpsched
207 ?        0:00 /usr/lib/lpsched
# pwdx 207
207:    /
# ptree 207
207   /usr/lib/lpsched
# pfiles 207
207:    /usr/lib/lpsched
  Current rlimit: 4096 file descriptors
   0: S_IFIFO mode:0000 dev:179,0 ino:70 uid:0 gid:0 size:0
      O_RDWR
   1: S_IFIFO mode:0000 dev:179,0 ino:70 uid:0 gid:0 size:0
      O_RDWR
   3: S_IFCHR mode:0666 dev:32,8 ino:11446 uid:0 gid:3 rdev:21,0
      O_WRONLY FD_CLOEXEC
   4: S_IFDOOR mode:0444 dev:183,0 ino:59515 uid:0 gid:0 size:0
      O_RDONLY|O_LARGEFILE FD_CLOEXEC  door to nscd[201]
   5: S_IFREG mode:0664 dev:32,9 ino:1330 uid:71 gid:8 size:0
      O_WRONLY
```

1. /usr/proc/bin ディレクトリを PATH 変数に追加します。
2. lpsched のプロセス識別番号を表示します。.
3. lpsched の現在の作業ディレクトリを表示します。.
4. lpsched が入っているプロセスツリーを表示します。.
5. fstat と fcntl の情報を表示します。

次の例は、pwait コマンドからの出力を示しています。このコマンドは、プロセスが終了するまで待ってから、発生した処理に関する情報を表示します。次の例は、コマンドツールウィンドウを閉じた後の pwait コマンドからの出力を示しています。

```
$ ps -e | grep cmdtool
  273 console 0:01 cmdtool
  277 console 0:01 cmdtool
  281 console 0:01 cmdtool
$ pwait -v 281
```

(続く)

```
281: terminated, wait status 0x0000
```

プロセスの制御 (/proc ツール)

/usr/proc/bin に入っているプロセスツールを使用すると、プロセスの一部を制御できます。表 35–3 に、これらのプロセスツールを示します。詳細は、proc(1) のマニュアルページを参照してください。

表 35–3 プロセスツール

ツールの種類	ツールの機能または表示内容
/usr/proc/bin/pstop *pid*	プロセスを停止する
/usr/proc/bin/prun *pid*	プロセスを再開する
/usr/proc/bin/ptime *pid*	microstate アカウントを使用してプロセスの時間を測定する
/usr/proc/bin/pwait [-v] *pid*	指定されたプロセスが終了するのを待つ
プロセスの詳細を表示するツール	
/usr/proc/bin/pcred *pid*	資格
/usr/proc/bin/pfiles *pid*	開いたファイルの fstat 情報と fcntl 情報
/usr/proc/bin/pflags *pid*	/proc の追跡フラグ、保留シグナルと保持シグナル、lwp ごとの他の状態情報
/usr/proc/bin/pldd *pid*	各プロセスにリンクされた動的ライブラリ
/usr/proc/bin/pmap *pid*	アドレス空間マップ

表 35–3 プロセスツール 続く

ツールの種類	ツールの機能または表示内容
`/usr/proc/bin/psig` *pid*	シグナルの動作
`/usr/proc/bin/pstack` *pid*	`lwp` ごとの 16 進数 + 記号スタックトレース
`/usr/proc/bin/ptree` *pid*	指定した *pid* が入ったプロセスツリー
`/usr/proc/bin/pwdx` *pid*	現在の作業ディレクトリ

上記の表 35–3 で、*pid* はプロセス識別番号です。この番号は `ps -ef` コマンドを使用して表示できます。

第 35 章では、プロセスツールコマンドを使用して、プロセスの詳細表示や、プロセスの起動および終了などのシステム管理作業を実行する方法を説明します。プロセスツールのさらに詳細な説明は、`proc(1)` のマニュアルページを参照してください。

プロセスが無限ループ内でトラップされた場合や、実行時間が長すぎる場合は、プロセスを終了 (`kill`) できます。`pkill` コマンドを使用してプロセスを終了する方法については、第 35 章を参照してください。

以前のフラットな `/proc` ファイルシステムは、状態情報と制御機能のためのサブディレクトリが追加されたディレクトリ階層に再構築されました。

`/proc` ファイルシステムは、ウォッチポイント機能も提供します。この機能は、プロセスのアドレス領域の個々のページの読み取り権または書き込み権を再マップするために使用されます。この機能は制限がなく、MT-safe です。

新しい `/proc` ファイル構造は、古い `/proc` インタフェースと完全なバイナリ互換を提供します。ただし、新しいウォッチポイント機能は、古いインタフェースでは使用できません。

デバッグ用ツールは、`/proc` の新しいウォッチポイント機能を使用するように変更されています。つまり、ウォッチポイントプロセス全体がより高速になったためです。

`dbx` デバッグ用ツールを使用してウォッチポイントを設定するときの次の制限は取り除かれました。

- SPARC レジスタウィンドウのため、スタック上のローカル変数にウォッチポイントを設定する。
- マルチスレッド化されたプロセスにウォッチポイントを設定する。

詳細は、proc(4)、core(4)、および adb(1) のマニュアルページを参照してください。

注 - 長いコマンド名を入力しなくてもすむように、プロセスツールディレクトリを PATH 変数に追加してください。これにより、各ファイル名の最後の部分 (たとえば、/usr/proc/bin/prun ではなく prun) を入力するだけで、プロセスツールを実行できます。

▼ プロセスを制御する方法

1. **(省略可能)** ps コマンドからの出力を使用して、詳細情報を表示したいプロセスの識別番号を調べます。

```
# pgrep process
```

process 詳細情報を表示したいプロセス名

プロセス識別番号は、出力の第 1 列目に表示されます。

2. 適切な /usr/proc/bin コマンドを使用してプロセスを制御します。

```
# /usr/proc/bin/pcommand PID
```

pcommand 実行したいプロセスツールコマンド。これらのコマンドについては、表 35–3 を参照

PID プロセスの識別番号

3. ps コマンドで、プロセスの状態を確認します。

```
# ps | grep PID
```

例 — プロセスを制御する

次の例は、プロセスツールを使用して印刷ツールを停止したり、再起動したりする方法を示しています。

```
# PATH=$PATH:/usr/proc/bin
# export PATH
# ps -e | grep 'print*'
264 console 0:03 printtool
# pstop 264
# prun 264
# ps | grep 264
264 console 0:03 printtool
#
```

1. /usr/proc/bin ディレクトリを PATH 変数に追加します。
2. 印刷ツールのプロセス識別番号を表示します。
3. 印刷ツールプロセスを停止します。
4. 印刷ツールプロセスを再開します。

プロセスの終了 (pkill)

プロセスを強制的に終了 (kill) させなければならない場合があります。プロセスが無限ループに入っていたり、大きいジョブを開始したが完了する前に停止したい場合があります。所有しているプロセスであれば、どれでも終了できます。また、スーパーユーザーはプロセス ID が 0、1、2、3、4 のものを除き、システム上のどんなプロセスでも終了できます。

詳細は、pkill(1) のマニュアルページを参照してください。

▼ プロセスを終了させる方法

1. **(省略可能) 別のユーザーが所有するプロセスを終了するには、スーパーユーザーになります。**

2. **(省略可能) pgrep コマンドからの出力を使用して、詳細情報を表示したいプロセスの識別番号を表示します。**

```
$ pgrep process
```

process　　詳細情報を表示したいプロセス名

プロセスの識別番号は、出力の第 1 列目に表示されます。

3. pkill コマンドを使用してプロセスを終了します。

```
$ pkill [-9] PID ...
```

-9　　プロセスを確実に終了させる

PID . . .　　停止する 1 つ以上のプロセスの ID

4. pgrep コマンドを使用して、プロセスが停止したことを確認します。

```
$ pgrep PID ...
```

プロセスクラス情報の管理

次のリストは、システム上で構成されるクラスと、タイムシェアリングクラスのユーザー優先順位の範囲です。クラスの種類は次のとおりです。

- システム (SYS)
- 対話型 (IA)
- リアルタイム (RT)
- タイムシェアリング (TS)
  - ユーザーが与える –20 から +20 までの優先順位の範囲
  - プロセスの優先順位は、親プロセスから継承されます。これを「ユーザーモード」の優先順位と呼びます。
  - システムは、ユーザーモードの優先順位をタイムシェアリングディスパッチパラメタテーブル内で検索し、nice または priocntl (ユーザー提供) 優先順

位に追加し、0 から 59 までの範囲を確保して「グローバル」優先順位を作成します。

priocntl を使用してプロセスのスケジュール優先順位を変更する

プロセスのスケジュール優先順位とは、プロセススケジューラによって割り当てられる優先順位のことです。これらの優先順位は、スケジューラのスケジュールポリシーに従って割り当てられます。dispadmin コマンドを使用すると、デフォルトのスケジュールポリシーを表示できます。dispadmin コマンドの使用方法については、620ページの「プロセスの優先順位を指定する方法」を参照してください。

priocntl(1) コマンドを使用すると、プロセスを優先順位クラスに割り当て、プロセスの優先順位を管理できます。プロセスを管理するための priocntl コマンドの使用方法については、620ページの「プロセスの優先順位を指定する方法」を参照してください。

▼ プロセスクラスに関する基本情報を表示する方法

priocntl -l コマンドを使用すると、プロセスクラスとスケジューリングパラメタを表示できます。

```
$ priocntl -l
```

例 — プロセスクラスに関する基本情報を表示する

次の例に priocntl -l コマンドからの出力を示します。

```
# priocntl -l
CONFIGURED CLASSES
==================

SYS (System Class)

TS (Time Sharing)
        Configured TS User Priority Range: -60 through 60

IA (Interactive)
        Configured IA User Priority Range: -60 through 60

RT (Real Time)
        Maximum Configured RT Priority: 59
```

▼ プロセスのグローバル優先順位を表示する方法

ps コマンドを使用して、プロセスのグローバル優先順位を表示できます。

```
$ ps -ecl
```

グローバル優先順位は、PRI カラムの下に表示されます。

例 — プロセスのグローバル優先順位を表示する

次の例は、ps -ecl からの出力を示しています。PRI カラム内のデータは、pageout が最上位の優先順位を持ち、sh が最下位の優先順位であることを示しています。

```
$ ps -ecl
 F S UID PID  PPID CLS PRI  ADDR      SZ  WCHAN    TTY      TIME   COMD
19 T 0   0    0    SYS 96   f00d05a8   0           ?        0:03  sched
 8 S 0   1    0    TS  50   ff0f4678 185  ff0f4848 ?       36:51   init
19 S 0   2    0    SYS 98   ff0f4018   0  f00c645c ?        0:01 pageout
19 S 0   3    0    SYS 60   ff0f5998   0  f00d0c68 ?      241:01 fsflush
 8 S 0   269  1    TS  58   ff0f5338 303  ff49837e ?        0:07    sac
 8 S 0   204  1    TS  43   ff2f6008  50  ff2f606e console  0:02     sh
```

▼ プロセスの優先順位を指定する方法

1. スーパーユーザーになります。

2. 指定した優先順位を持つプロセスを起動します。

```
# priocntl -e -c class -m userlimit -p pri command_name
```

-e	コマンドを実行する
-c *class*	プロセスを実行する範囲のクラスを指定する。デフォルトのクラスは TS (タイムシェアリング) または RT (リアルタイム)
-m *userlimit*	-p オプションを使用するときに、優先順位を上下できる最大範囲を指定する
-p *pri command_name*	リアルタイムスレッド用に RT クラス内で相対優先順位を指定できるようにする。タイムシェアリングプロセスの場合は、-p オプションを使用すると -20 から +20 までのユーザー提供の優先順位を指定できる

3. ps -ecl コマンドで、プロセスの状態を確認します。

```
# ps -ecl | grep command_name
```

例 — 優先順位を指定する

次の例では、ユーザーが提供する最上位の優先順位を使用して find コマンドを起動します。

```
# priocntl -e -c TS -m 20 -p 20 find . -name core -print
# ps -ecl | grep find
```

▼ タイムシェアリングプロセスのスケジューリングパラメタを変更する方法

1. スーパーユーザーになります。
2. 実行中のタイムシェアリングプロセスのスケジューリングパラメタを変更します。

```
# priocntl -s -m userlimit [-p userpriority] -i idtype idlist
```

`-s`	ユーザー優先順位の範囲の上限を設定し、現在の優先順位を変更できる
`-m` *userlimit*	`-p` オプションを使用するときに、優先順位を上下できる最大範囲を指定する
`-p` *userpriority*	優先順位を指定できる
`-i` *idtype idlist*	*idtype* と *idlist* の組み合わせを使用してプロセスを識別する。*idtype* では `PID` や `UID` など、`ID` のタイプを指定する

3. `ps -ecl` コマンドで、プロセスの状態を確認します。

```
# ps -ecl | grep idlist
```

例 — タイムシェアリングプロセスのスケジューリングパラメタを変更する

次の例では、500 ミリ秒のタイムスライス、クラス RT 内の優先順位 20、グローバル優先順位 120 を指定して、コマンドを実行します。

```
# priocntl -e -c RT -t 500 -p 20 myprog
# ps -ecl | grep myprog
```

▼ プロセスのクラスを変更する方法

1. (省略可能) スーパーユーザーになります。

注 - プロセスをリアルタイムプロセスに変更したり、リアルタイムプロセスから変更したりするには、ユーザーはリアルタイムシェル内でスーパーユーザーであるか、作業中でなければなりません。

---

2. プロセスのクラスを変更します。

```
# priocntl -s -c class -i idtype idlist
```

`-s`	ユーザー優先順位の範囲について上限を設定し、現在の優先順位を変更できる
`-c` *class*	クラス `TS` またはクラス `RT` を指定して、プロセスのクラスを変更する
`-i` *idtype idlist*	*idtype* と *idlist* の組み合わせを使用してプロセスを識別する。*idtype* では `PID` や `UID` など、`ID` のタイプを指定する

3. `ps -ecl` コマンドで、プロセスの状態を確認します。

```
# ps -ecl | grep idlist
```

例 — プロセスのクラスを変更する

次の例では、ユーザー `15249` が所有するすべてのプロセスをリアルタイムプロセスに変更します。

```
# priocntl -s -c RT -i uid 15249
# ps -ecl | grep 15249
```

注 - スーパーユーザーとしてユーザープロセスをリアルタイムクラスに変更すると、その後ユーザーはリアルタイムのスケジューリングパラメタを (`priocntl -s` を使用して) 変更できません。

---

nice を使用してタイムシェアリングプロセスの優先順位を変更する

`nice(1)` コマンドは、SunOS の旧バージョンとの下位互換性を保つためにのみサポートされます。`priocntl` コマンドを使用する方がプロセスを柔軟に管理できます。

プロセスの優先順位は、そのスケジュールクラスポリシーと *nice number* 番号によって決定されます。各タイムシェアリングプロセスは、ユーザーが与えた優先順位を加算して計算されるグローバル優先順位を持っています。これは、`nice` コマンドま

たは priocntl コマンド、およびシステムによって計算される優先順位の影響を受けます。

プロセスの実行優先順位番号は、オペレーティングシステムによって割り当てられ、スケジュールクラス、使用される CPU 時間、nice 値 (タイムシェアリングプロセスの場合) などの複数の要素によって決定されます。

各タイムシェアリングプロセスは、親プロセスから継承したデフォルトの nice 番号で起動します。nice 値は、ps レポートの NI カラムに表示されます。

ユーザーは、自分が与える nice 番号優先順位を大きくしてプロセスの優先順位を下げることができます。ただし、nice 番号を小さくしてプロセスの優先順位を上げることができるのは、スーパーユーザー (または root) だけです。これは、ユーザーが各自のプロセスの優先順位を大きくして CPU の独占比率を高めるのを防ぐためです。

nice 番号の範囲は 0 から +40 までで、0 は最上位の優先順位を与えます。デフォルト値は 20 です。nice コマンドには利用できるバージョンが 2 つあり、一方は標準バージョンの /usr/bin/nice で、他方は C シェルの一部となっているバージョンです。

▼ プロセスの優先順位を変更する方法

nice 番号を変更して、コマンドやプロセスの優先順位を変更できます。プロセスの優先順位を下げるには、次のコマンドを使用します。

/usr/bin/nice *command_name*	nice 番号を 4 単位で増やす (デフォルト)
/usr/bin/nice +4 *command_name*	nice 番号を 4 単位で増やす
/usr/bin/nice -10 *command_name*	nice 番号を 10 単位で増やす

第 1 と第 2 のコマンドは、nice 番号を 4 単位で増やします (デフォルト)。第 3 のコマンドは、nice を 10 単位で増やしていますが負数の増分なので、プロセスの優先順位を下げます。

次のコマンドは、nice 番号を小さくしてコマンドの優先順位を上げます。

プロセスの優先順位を上げるには、次のコマンドを使用します。

`/usr/bin/nice -10` *command_name*	`nice` 番号を小さくしてコマンドの優先順位を上げる
`/usr/bin/nice - -10` *command_name*	`nice` 番号を小さくしてコマンドの優先順位を上げる。最初のマイナス記号はオプションの記号で、第 2 のマイナス記号は負の数を示す

上記のコマンドでは、`nice` 番号を小さくしてコマンド *command_name* の優先順位を上げます。第 2 のコマンドでは、2 つのマイナス記号が必要なので注意してください。

プロセスの問題解決方法

すでに判明している問題の解決方法のヒントを次に示します。

- 同じユーザーが所有する複数の同じジョブがないかどうかを調べます。ジョブが終了するまで待たずに多数のバックグラウンドジョブを起動するスクリプトを実行した場合に、この問題が発生することがあります。
- CPU 時間が大量に増えているプロセスがないかどうかを調べます。そのためには、`TIME` フィールドを調べます。そのプロセスが無限ループに入っている可能性があります。
- 実行中のプロセスの優先順位が高すぎないかどうかを調べます。`ps -c` と入力して `CLS` フィールドを調べると、各プロセスのスケジューラクラスが表示されます。リアルタイム (`RT`) プロセスとして実行中のプロセスが CPU を独占している可能性があります。また、`nice` 値の大きいタイムシェアリング (`TS`) プロセスがないかどうかを調べます。スーパーユーザー特権を持つユーザーが、このプロセスの優先順位を上げすぎた可能性があります。システム管理者は、`nice` コマンドを使用して優先順位を下げることができます。
- 制御がきかなくなったプロセス、つまり CPU 時間の使用が継続的に増加しているプロセスがないかどうかを調べます。プロセスが起動 (`STIME`) されたときに調べるか、またはしばらくの間 CPU 時間が累計されるのを見守っていると (`TIME`)、この問題が発生したのがわかることがあります。

第 36 章

性能の監視手順

この章では、vmstat、iostat、df、または sar コマンドを使用してシステム性能を監視する手順について説明します。この章で説明する手順は次のとおりです。

- 650ページの「未使用のメモリーをチェックする方法 (sar -r)」
- 651ページの「CPU の使用状況をチェックする方法 (sar -u)」
- 652ページの「システムテーブルの状態をチェックする方法 (sar -v)」
- 654ページの「スワップ動作をチェックする方法 (sar -w)」
- 655ページの「端末動作をチェックする方法 (sar -y)」
- 656ページの「システム全体の性能をチェックする方法 (sar -A)」
- 660ページの「自動データ収集を設定する方法」

仮想メモリーの統計情報の表示 (vmstat)

vmstat コマンドを使用すると、仮想メモリーの統計情報と、CPU の負荷、ページング、コンテキスト切替え数、デバイス割り込み、システムコールなどのシステムイベントに関する情報を表示できます。また、vmstat コマンドを使用すると、スワップ、キャッシュフラッシュ、および割り込みに関する統計情報も表示できます。

詳細は、vmstat(1M) のマニュアルページを参照してください。

▼ 仮想メモリーの統計情報を表示する方法 (vmstat)

時間間隔を指定して vmstat コマンドを使用すると、仮想メモリーの統計情報を収集します。

```
$ vmstat n
```

n　　レポート間の間隔を秒単位で表した値

表 36–1 に vmstat の出力内のフィールドを示します。

表 36–1 vmstat コマンドからの出力

カテゴリ	フィールド名	説明
procs		次の状態を報告する
	r	ディスパッチ待ち行列内のカーネルスレッド数

表 36–1 vmstat コマンドからの出力 続く

カテゴリ	フィールド名	説明
	b	資源を待機中のブロックされたカーネルスレッド
	w	資源処理の完了を待機中のスワップアウトされた軽量プロセス数
memory		実メモリーと仮想メモリーの使用状況を表示する
	swap	使用可能なスワップ空間
	free	空きリストのサイズ
page		ページフォルトとページング動作を 1 秒当りの単位数として表示する
	re	回収されたページ数
	mf	軽度のフォルトと重大なフォルト
	pi	ページインされたキロバイト数
	po	ページアウトされたキロバイト数
	fr	解放されたキロバイト数
	de	最後にスワップインされたプロセスに必要だと予想されるメモリー
	sr	ページデーモンによって走査されたページ数 (現在は使用されていない)。sr が 0 以外の値であれば、ページデーモンは実行されている
disk		最高 4 台のディスク上のデータを示す、1 秒当りのディスク処理数を表示する
faults		トラップ/割り込み率 (1 秒当り) を表示する
	in	1 秒当りの割り込み数
	sy	1 秒当りのシステムコール数

表 36–1　vmstat コマンドからの出力　続く

カテゴリ	フィールド名	説明
	cs	CPU のコンテキスト切替え率
cpu		CPU 時間の使用状況を表示する
	us	ユーザー時間
	sy	システム時間
	id	アイドル時間

例 — 仮想メモリーの統計情報を表示する

次の例に、5 秒間隔で収集された統計情報に関する vmstat の表示を示します。

```
$ vmstat 5
 procs     memory            page            disk      faults      cpu
r b w   swap free re  mf  pi  po  fr de sr f0 s3 -- --  in  sy  cs us sy  id
0 0 8 28312  668  0   9   2   0   1  0  0  0  1  0  0  10  61  82  1  2  97
0 0 3 31940  248  0  10  20   0  26  0 27  0  4  0  0  53 189 191  6  6  88
0 0 3 32080  288  3  19  49   6  26  0 15  0  9  0  0  75 415 277  6 15  79
0 0 3 32080  256  0  26  20   6  21  0 12  1  6  0  0 163 110 138  1  3  96
0 1 3 32060  256  3  45  52  28  61  0 27  5 12  0  0 195 191 223  7 11  82
0 0 3 32056  260  0   1   0   0   0  0  0  0  0  0  0   4  52  84  0  1  99
```

▼ システムイベント情報を表示する方法 (vmstat -s)

vmstat -s を実行すると、システムを前回ブートした後に発生した各種システムイベントの合計が表示されます。

```
$ vmstat -s
        0 swap ins
        0 swap outs
        0 pages swapped in
        0 pages swapped out
   392182 total address trans. faults taken
    20419 page ins
      923 page outs
```

(続く)

```
    30072 pages paged in
     9194 pages paged out
    65167 total reclaims
    65157 reclaims from free list
        0 micro (hat) faults
   392182 minor (as) faults
    19383 major faults
    85775 copy-on-write faults
    66637 zero fill page faults
    46309 pages examined by the clock daemon
        6 revolutions of the clock hand
    15578 pages freed by the clock daemon
     4398 forks
      352 vforks
     4267 execs
 12926285 cpu context switches
109029866 device interrupts
   499296 traps
 22461261 system calls
   778068 total name lookups (cache hits 97%)
    18739 user   cpu
    34662 system cpu
 52051435 idle   cpu
    25252 wait   cpu
```

▼ スワップの統計情報を表示する方法 (vmstat -S)

vmstat -S を実行すると、スワップの統計情報が表示されます。

```
$ vmstat -S
 procs     memory            page            disk          faults      cpu
 r b w   swap  free  si  so pi po fr de sr f0 s0 s6 --   in   sy   cs us sy id
 0 0 0 200968 17936   0   0  0  0  0  0  0  0  0  0  0  109   43   24  0  0 100
```

表 36–2 にフィールドを示します。

表 36–2 vmstat -S コマンドからの出力

フィールド	説明
si	1 秒当りにスワップされた平均軽量プロセス数
so	スワップアウトされた全プロセス数

注 - vmstat コマンドは、これらの両フィールドを出力しません。スワップ統計情報の詳細情報を表示するには、sar コマンドを使用してください。

▼ キャッシュフラッシュの統計情報を表示する方法 (vmstat -c)

vmstat -c を実行すると、仮想キャッシュのキャッシュフラッシュ統計情報が表示されます。

```
$ vmstat -c
usr     ctx     rgn     seg     pag     par
  0   60714       5  134584 4486560 4718054
```

このコマンドを実行すると、前回のブート後に発生したキャッシュフラッシュの合計数が表示されます。表 36–3 にキャッシュタイプを示します。

表 36–3 vmstat -c コマンドからの出力

キャッシュ名	キャッシュタイプ
usr	ユーザー
ctx	コンテキスト
rgn	領域
seg	セグメント
pag	ページ
par	ページの一部

▼ 各デバイス当りの割り込み数を表示する方法 (vmstat -i)

vmstat -i を実行すると、各デバイス当りの割り込み数が表示されます。

```
$ vmstat -i
```

例 — 各デバイス当りの割り込み数を表示する

次の例は、vmstat -i コマンドからの出力を示します。

```
$ vmstat -i
interrupt          total     rate
--------------------------------
clock           52163269      100
esp0             2600077        4
zsc0               25341        0
zsc1               48917        0
cgsixc0              459        0
lec0              400882        0
fdc0                  14        0
bppc0                  0        0
audiocs0               0        0
--------------------------------
Total           55238959      105
```

ディスク使用状況の表示 (iostat *n*)

iostat コマンドを使用すると、ディスクの入出力に関する統計情報を表示し、スループット、使用率、待ち行列の長さ、トランザクション率、サービス時間の計測結果を表示できます。このコマンドの詳細は、iostat(1M) のマニュアルページを参照してください。

▼ ディスクの使用状況を表示する方法 (iostat)

時間間隔を指定して iostat コマンドを使用すると、ディスク動作情報を表示できます。

```
$ iostat 5
     tty          fd0           sd3           nfs1         nfs31          cpu
tin tout kps tps serv  kps tps serv  kps tps serv  kps tps serv  us sy wt id
  0    1   0   0  410    3   0   29    0   0    9    3   0   47   4  2  0 94
```

出力の 1 行目は、前回のブート以降の統計情報を示します。2 行目以降は、時間間隔ごとの統計情報を示します。デフォルトでは、端末 (tty)、ディスク (fd と sd)、CPU (cpu) の統計情報が表示されます。

表 36–4 に iostat コマンド出力内のフィールドを示します。

表 36–4 iostat *n* コマンドからの出力

統計情報の対象	フィールド	説明
端末		
	tin	端末の入力待ち行列内の文字数
	tout	端末の出力待ち行列内の文字数
ディスク		
	bps	1 秒当りのブロック数
	tps	1 秒当りのトランザクション数
	serv	ミリ秒単位の平均サービス時間
CPU		
	us	ユーザーモード
	sy	システムモード
	wt	入出力待機中
	id	アイドル状態

例 — ディスクの使用状況を表示する

次の例は、5 秒間隔で収集されるディスク統計情報を示します。

```
$ iostat 5
tty         sd0           sd6           nfs1          nfs49           cpu
tin tout kps tps serv  kps tps serv  kps tps serv  kps tps serv  us sy wt id
 0    0   1   0   49    0   0    0    0   0    0    0   0   15   0  0  0 100
 0   47   0   0    0    0   0    0    0   0    0    0   0    0   0  0  0 100
 0   16   0   0    0    0   0    0    0   0    0    0   0    0   0  0  0 100
 0   16   0   0    0    0   0    0    0   0    0    0   0    0   0  0  0 100
 0   16  44   6  132    0   0    0    0   0    0    0   0    0   0  0  1 99
 0   16   0   0    0    0   0    0    0   0    0    0   0    0   0  0  0 100
 0   16   0   0    0    0   0    0    0   0    0    0   0    0   0  0  0 100
```

(続く)

```
  0   16   0   0    0    0   0    0    0   0    0    0   0    0   0  0  0 100
  0   16   0   0    0    0   0    0    0   0    0    0   0    0   0  0  0 100
  0   16   0   0    0    0   0    0    0   0    0    0   0    0   0  0  0 100
  0   16   3   1   23    0   0    0    0   0    0    0   0    0   0  0  1 99
  0   16   0   0    0    0   0    0    0   0    0    0   0    0   0  0  0 100
  0   16   0   0    0    0   0    0    0   0    0    0   0    0   0  0  0 100
  0   16   0   0    0    0   0    0    0   0    0    0   0    0   0  0  0 100
```

▼ 拡張ディスク統計情報を表示する方法 (iostat -xtc)

iostat -xtc を実行すると、拡張ディスク統計情報が表示されます。

```
$ iostat -xtc
                 extended device statistics                  tty         cpu
device       r/s    w/s   kr/s   kw/s wait actv  svc_t  %w  %b  tin tout  us sy wt id
fd0          0.0    0.0    0.0    0.0  0.0  0.0    0.0   0   0    0    0   0  0  0 100
sd0          0.0    0.0    0.4    0.4  0.0  0.0   49.5   0   0
sd6          0.0    0.0    0.0    0.0  0.0  0.0    0.0   0   0
nfs1         0.0    0.0    0.0    0.0  0.0  0.0    0.0   0   0
nfs49        0.0    0.0    0.0    0.0  0.0  0.0   15.1   0   0
nfs53        0.0    0.0    0.4    0.0  0.0  0.0   24.5   0   0
nfs54        0.0    0.0    0.0    0.0  0.0  0.0    6.3   0   0
nfs55        0.0    0.0    0.0    0.0  0.0  0.0    4.9   0   0
```

このコマンドは、ディスクごとに 1 行ずつ出力を表示します。表 36–5 は、出力フィールドを示します。

表 36–5 iostat -xtc コマンドからの出力

フィールド名	説明
r/s	1 秒当りの読み取り数
w/s	1 秒当りの書き込み数
Kr/s	1 秒当りの読み取りキロバイト数
Kw/s	1 秒当りの書き込みキロバイト数
wait	サービス (待ち行列の長さ) を待機中の平均トランザクション数

表 36–5　iostat -xtc コマンドからの出力　続く

フィールド名	説明
actv	サービス中の平均トランザクション数
svc_t	ミリ秒単位で表した平均サービス時間
%w	待ち行列が空でない時間の割合
%b	ディスクがビジーである時間の割合

ディスク使用統計の表示 (df)

df コマンドを使用すると、マウントされている各ディスク上の空きディスク容量が表示されます。レポート用の統計情報では使用可能容量の合計の内先頭に 10% の空き容量を残しておくので、df から報告される「使用可能」ディスク容量は全容量の 90% のみに相当します。この先頭の空き容量は、性能を高めるために常に空になっています。

実際に df からレポートされるディスク容量の割合は、使用済み容量を使用可能容量で割った値です。

ファイルシステムの容量が 90% を超える場合は、cp を使用して空いているディスクにファイルを転送するか、tar または cpio を使用してテープに転送するか、ファイルを削除してください。

このコマンドの詳細は、df(1M) のマニュアルページを参照してください。

▼ ファイルシステム情報を表示する方法 (df)

df -k コマンドを使用すると、ファイルシステム情報がキロバイト単位で表示されます。

```
$ df -k
Filesystem            kbytes    used   avail capacity  Mounted on
/dev/dsk/c0t3d0s0     192807   40231  133296    24%    /
```

表 36–6 は、df -k コマンドの出力を示します。

表 36–6 df -k コマンドからの出力

フィールド名	説明
kbytes	ファイルシステム内の使用可能容量の合計
used	使用されている容量
avail	使用可能容量
capacity	使用されている容量が全容量に占める割合
mounted on	マウントポイント

例 — ファイルシステム情報を表示する

次の例に、df -k コマンドからの出力を示します。

```
$ df -k
/dev/dsk/c0t0d0s0     192807   49043  124484    29%    /
/dev/dsk/c0t0d0s6    1190551  680444  450580    61%    /usr
/proc                      0       0       0     0%    /proc
fd                         0       0       0     0%    /dev/fd
mnttab                     0       0       0     0%    /etc/mnttab
swap                  198056       0  198056     0%    /var/run
swap                  198064       8  198056     1%    /tmp
/dev/dsk/c0t0d0s5     192807    2031  171496     2%    /opt
/dev/dsk/c0t0d0s7     217191       9  195463     1%    /export/home
venus:/usr/dist    20612581 13237316 6963015    66%    /usr/dist
```

システム動作の監視 (sar)

sar コマンドは、次の目的で使用します。

- システム動作についてのデータを編成し表示する
- 特殊な要求に基づいて、システム動作データにアクセスする

- システム性能を測定および監視するレポートを自動的に生成する。また、特定の性能障害を正確に突き止めるための、特殊な要求レポートも生成する。657ページの「システム動作データの自動収集 (sar)」を参照

このコマンドの詳細は、sar(1) のマニュアルページを参照してください。

▼ ファイルアクセスをチェックする方法 (sar -a)

sar -a コマンドを使用すると、ファイルアクセス操作の統計情報が表示されます。

```
$ sar -a
SunOS venus 5.8 Generic sun4u    09/07/99

00:00:00  iget/s namei/s dirbk/s
01:00:00       0       0       0
02:00:02       0       0       0
03:00:00       0       1       0
04:00:00       0       0       0
05:00:01       0       0       0
06:00:00       0       0       0

Average        0       1       0
```

表 36–7 に、レポートされるオペレーティングシステムのルーチンを示します。

表 36–7 sar -a コマンドからの出力

フィールド名	説明
iget/s	ディレクトリ名検索キャッシュ (dnlc) 内に入っていない i ノードに対して出された要求数
namei/s	1 秒当りのファイルシステムパスの検索数。namei で dnlc 内にディレクトリ名が見つからない場合は、iget が呼び出され、ファイルまたはディレクトリの i ノードが取得される。したがって、ほとんどの igets は dnlc が欠落した結果である
dirbk/s	1 秒間に実行されたディレクトリブロックの読み取り回数

表示される値が大きいほど、カーネルはユーザーファイルへのアクセスに長い時間を費やしています。この時間には、プログラムとアプリケーションによるファイルシステムの使用量が反映されます。-a オプションを使用すると、アプリケーションのディスク依存度を表示できるので便利です。

▼ バッファー動作をチェックする方法 (sar -b)

sar -b コマンドを使用すると、バッファー動作の統計情報が表示されます。

バッファーは、i ノード、シリンダグループブロック、間接ブロックなどのメタデータをキャッシュに書き込むために使用されます。

```
$ sar -b
00:00:00 bread/s lread/s %rcache bwrit/s lwrit/s %wcache pread/s pwrit/s
01:00:00       0       0     100       0       0      55       0       0
```

表 36–8 は、-b オプションを指定したときに表示されるバッファー動作を示します。最も重要なエントリは、キャッシュヒット率 %rcache と %wcache です。この 2 つのエントリは、システムバッファーリングの効率を測定します。%rcache が 90 未満の場合や、%wcache が 65 未満の場合は、バッファー領域を大きくすれば性能を改善できる可能性があります。

表 36–8 sar -b コマンドからの出力

フィールド名	説明
bread/s	ディスクからバッファーキャッシュに投入された 1 秒当りの平均読み取り数
lread/s	バッファーキャッシュからの 1 秒当りの平均論理読み取り数
%rcache	バッファーキャッシュ内で見つかった論理読み込み数の小数部 (lread/s に対する bread/s の比を 100% から差し引いた値)
bwrit/s	バッファーキャッシュからディスクに書き込まれた 1 秒当りの平均物理ブロック数 (512 ブロック)
lwrite/s	バッファーキャッシュへの 1 秒当りの平均論理書き込み数
%wcache	バッファーキャッシュ内で見つかった論理書き込み数の小数部 (lwrit/s に対する bwrit/s の比を 100% から差し引いた値)
pread/s	キャラクタ型デバイスインタフェースを使用する 1 秒当りの平均物理読み取り数
pwrit/s	キャラクタ型デバイスインタフェースを使用する 1 秒当りの平均物理書き込み要求数

例 — バッファー動作をチェックする

次の sar -b 出力の例は、すべてのデータは許容範囲に収まっているので、%rcache バッファーと %wcache バッファーが処理速度低下の原因ではないことを示します。

```
$ sar -b
SunOS venus 5.8 Generic sun4u    09/07/99

00:00:00 bread/s lread/s %rcache bwrit/s lwrit/s %wcache pread/s pwrit/s
01:00:00       0       0     100       0       0      55       0       0
02:00:02       0       0     100       0       0      55       0       0
03:00:00       0       0     100       0       0      72       0       0
04:00:00       0       0     100       0       0      56       0       0
05:00:01       0       0     100       0       0      55       0       0
06:00:00       0       0     100       0       0      55       0       0

Average        0       0      94       0       0      64       0       0
```

▼ システムコールの統計情報をチェックする方法 (sar -c)

sar -c コマンドを使用すると、システムコールの統計情報が表示されます。

```
$ sar -c
00:00:00 scall/s sread/s swrit/s  fork/s  exec/s rchar/s wchar/s
01:00:00      38       2       2    0.00    0.00     149     120
```

表 36–9 に、-c オプションでレポートされる次のシステムコールのカテゴリを示します。一般に、reads と writes はシステムコール合計の約半分ですが、割合はシステムで実行中の動作によって大幅に変動します。

表 36–9 sar -c コマンドからの出力

フィールド名	説明
scall/s	1 秒当りのすべてのタイプのシステムコール数 (通常は、ビジーな 4 ないし 6 ユーザーのシステム上で 1 秒当り約 30)
sread/s	1 秒当りの read システムコール数
swrit/s	1 秒当りの write システムコール数

表 36–9　sar -c コマンドからの出力　続く

フィールド名	説明
fork/s	1 秒当りの fork システムコール数 (4 ないし 6 ユーザーのシステム上で毎秒約 0.5)。この数値は、シェルスクリプトの実行中は大きくなる
exec/d	1 秒当りの exec システムコール数。exec/s を fork/s で割った値が 3 より大きい場合は、効率の悪い PATH 変数を調べる
rchar/s	read システムコールによって転送される 1 秒当りの文字 (バイト) 数
wchar/s	write システムコールによって転送される 1 秒当りの文字 (バイト) 数

例 — システムコールの統計情報をチェックする

次の例に、sar -c コマンドからの出力を示します。

```
$ sar -c
SunOS venus 5.8 Generic sun4u    09/07/99

00:00:00 scall/s sread/s swrit/s  fork/s  exec/s rchar/s wchar/s
01:00:00      38       2       2    0.00    0.00     149     120
02:00:02      38       2       2    0.00    0.00     149     120
03:00:00      42       2       2    0.05    0.05     218     147
04:00:00      39       2       2    0.01    0.00     155     123
05:00:01      38       2       2    0.00    0.00     150     120
06:00:00      38       2       2    0.01    0.00     149     120

Average       50       4       3    0.02    0.02     532     238
```

▼ ディスク動作をチェックする方法 (sar -d)

sar -d コマンドを使用すると、ディスク動作の統計情報が表示されます。

```
$ sar -d
00:00:00   device        %busy   avque   r+w/s  blks/s  avwait  avserv

01:00:00   fd0               0     0.0       0       0     0.0     0.0
```

表 36–10 は、-d オプションを使用したときに表示されるディスクデバイス動作を示します。待ち行列内に何かがあるときは、待ち行列の長さと待ち時間が計測される

ので注意してください。%busy の値が小さい場合に、待ち行列とサービス時間が大きければ、変更されたブロックをディスクに随時書き込むために、システムが定期的に処理していることを示す場合があります。

表 36–10 sar -d コマンドからの出力

フィールド名	説明
device	監視中のディスクデバイス名
%busy	デバイスが転送要求のサービスに費やす時間の割合
avque	平均待ち時間と平均サービス時間の合計
r+w/s	デバイスへの 1 秒当りの読み取り転送数と書き込み転送数
blks/s	デバイスに転送される 1 秒当りの 512 バイトブロック数
avwait	待ち行列内でアイドル状態で待機中の要求を転送する平均ミリ秒数 (待ち行列に要求が入っているときのみ計測)
avserv	デバイスが転送要求を完了するまでの平均ミリ秒数 (ディスクの場合は、この値にシークタイム、回転待ち時間、データ転送時間が含まれる)

例 — ディスク動作をチェックする

次の例は、sar -d コマンドからの一部省略した出力を示します。

```
$ sar -d
SunOS venus 5.8 Generic sun4u    09/07/99

00:00:00   device        %busy   avque   r+w/s  blks/s  avwait  avserv

01:00:00   fd0               0     0.0       0       0     0.0     0.0
           nfs1              0     0.0       0       0     0.0     0.0
           sd0               0     0.0       0       0     0.0    39.6
           sd0,a             0     0.0       0       0     0.0    39.6
           sd0,b             0     0.0       0       0     0.0     0.0
           sd0,c             0     0.0       0       0     0.0     0.0
           sd0,f             0     0.0       0       0     0.0     0.0
           sd0,g             0     0.0       0       0     0.0     0.0
           sd0,h             0     0.0       0       0     0.0     0.0
```

(続く)

```
          sd6               0     0.0       0       0     0.0     0.0
```

▼ ページアウトとメモリーをチェックする方法 (sar -g)

sar -g オプションを使用すると、ページアウトとメモリー解放動作が (平均値として) 表示されます。

```
$ sar -g
00:00:00  pgout/s ppgout/s pgfree/s pgscan/s %ufs_ipf
01:00:00     0.00     0.00     0.00     0.00     0.00
```

sar -g で表示される出力は、より多くのメモリーが必要かどうかを判断するのに役立ちます。ps -elf コマンドを使用すると、ページデーモンに使用される CPU サイクル数が表示されます。サイクル数が大きく、pgfree/s と pgscan/s の値が大きければ、メモリー不足を示します。

また、sar -g を使用すると、i ノードの再利用間隔が短すぎるために、再利用可能なページが失われているかどうかも表示されます。

表 36–11 に、-g オプションからの出力を示します。

表 36–11 sar -g コマンドからの出力

フィールド名	説明
pgout/s	1 秒間にページアウトされた要求数
ppgout/s	1 秒間に実際にページアウトされたページ数 (1 つのページアウト要求で複数のページがページアウトされることがある)
pgfree/s	空きリストに配置された 1 秒当りのページ数

表 36–11 sar -g コマンドからの出力 続く

フィールド名	説明
pgscan/s	ページデーモンによって走査された 1 秒当りのページ数。この値が大きい場合は、ページデーモンが空きメモリーのチェックに大量の時間を費やしている。これは、メモリーを増やす必要があることを示す
%ufs_ipf	ufs がそれに関連付けられた再使用可能ページを持つ iget によって空きリストから取り出された割合。これらのページはフラッシュされ、プロセスが回収できなくなる。したがって、これはページフラッシュを伴う igets の割合である。値が大きければ、i ノードの空きリストがページ境界であり、ufs の i ノード数を増やす必要があることを示す

例 — ページアウトとメモリーをチェックする

次の例に、sar -g コマンドからの出力を示します。

```
$ sar -g
SunOS venus 5.8 Generic sun4u    09/07/99

00:00:00  pgout/s ppgout/s pgfree/s pgscan/s %ufs_ipf
01:00:00     0.00     0.00     0.00     0.00     0.00
02:00:02     0.00     0.00     0.00     0.00     0.00
03:00:00     0.00     0.01     0.01     0.00     0.00
04:00:00     0.00     0.00     0.00     0.00     0.00
05:00:01     0.00     0.00     0.00     0.00     0.00
06:00:00     0.00     0.00     0.00     0.00     0.00

Average      0.01     0.12     0.21     0.66     0.00
```

▼ カーネルメモリーの割り当てをチェックする方法 (sar -k)

sar -k コマンドを使用すると、Kernel Memory Allocator (KMA) に関して次の動作が表示されます。

KMA を使用すると、カーネルサブシステムは必要に応じてメモリーを割り当て、解放できます。最大量のメモリーを静的に割り当てるのではなく、ピークを下回る負荷を要求するのが予想されるため、KMA はメモリー要求を「小」(256 バイト未満)、「大」(512 バイト～ 4K バイト)、「サイズ超過」(4K バイト超) という 3 つのカテゴリに分けます。また、2 つのメモリープールを管理して、「小」要求と

「大」要求を満たします。「サイズ超過」要求は、システムページアロケータからメモリーを割り当てることで満たされます。

KMA 資源を使用するドライブや STREAMS の作成に使用中のシステムを調査する場合は、sar -k を使用すると便利です。それ以外の場合は、このコマンドで提供される情報は不要です。KMA 資源を使用するが、終了前には特に資源を返さないドライバやモジュールがあると、メモリーのリークが生じることがあります。メモリーリークが発生すると、KMA によって割り当てられるメモリーは時間が経つにつれて増大します。したがって、sar -k の alloc フィールドの値が時間が経つにつれ増える場合は、メモリーリークの可能性があります。メモリーリークのもう 1 つの兆候は、要求が失敗することです。この問題が発生した場合は、メモリーリークのために KMA がメモリーを予約したり割り当てたりできなくなっている可能性があります。

メモリーリークが発生した場合は、KMA からメモリーを要求したが返していないドライバや STREAMS がないかどうかをチェックする必要があります。

```
$ sar -k
00:00:00 sml_mem   alloc  fail  lg_mem   alloc   fail  ovsz_alloc  fail
01:00:00 2523136 1866512     0 18939904 14762364    0      360448     0
02:00:02 2523136 1861724     0 18939904 14778748    0      360448     0
```

表 36–12 に、–k オプションからの出力を示します。

表 36–12 sar -k コマンドからの出力

フィールド名	説明
sml_mem	KMA が小メモリー要求プール内で使用できるメモリーのバイト数 (小要求は 256 バイト未満)
alloc	KMA が小メモリー要求プールから小メモリー要求に割り当てたメモリーのバイト数
fail	小量のメモリーで失敗した要求数
lg_mem	KMA が大メモリー要求プール内で使用できるメモリーのバイト数 (大要求は 512 バイトから 4K バイトまで)
alloc	KMA が大メモリー要求プールから大メモリー要求に割り当てたメモリーのバイト数

表 36–12 sar -k コマンドからの出力 続く

フィールド名	説明
fail	大メモリーで失敗した要求数
ovsz_alloc	サイズ超過要求 (4K バイトを超える要求) に割り当てられたメモリーの容量。これらの要求はページアロケータによって満たされるので、プールはない
fail	サイズ超過メモリーで失敗した要求数

例 — カーネルメモリーの割り当てをチェックする (sar)

次の例は、sar -k 出力を示します。

```
$ sar -k
SunOS venus 5.8 Generic sun4u    09/07/99

00:00:00 sml_mem   alloc  fail  lg_mem   alloc  fail  ovsz_alloc  fail
01:00:00 2523136 1866512     0 18939904 14762364     0      360448     0
02:00:02 2523136 1861724     0 18939904 14778748     0      360448     0
03:00:00 2523136 1865664     0 18939904 14745884     0      360448     0
04:00:00 2523136 1867692     0 18939904 14746616     0      360448     0
05:00:01 2523136 1867208     0 18939904 14763700     0      360448     0
06:00:00 2523136 1867772     0 18939904 14779444     0      360448     0

Average  2724096 1791806     0 20089344 15434591     0      360448     0
```

▼ プロセス間通信をチェックする方法 (sar -m)

sar -m コマンドを使用すると、プロセス間通信の動作が表示されます。

```
$ sar -m
00:00:00   msg/s  sema/s
01:00:00    0.00    0.00
```

通常、これらの数字は、メッセージやセマフォを使用するアプリケーションを実行していない限りゼロ (0.00) です。

表 36–13 に、–m オプションからの出力を示します。

表 36–13　sar -m コマンドからの出力

フィールド名	説明
msg/s	1 秒当りのメッセージ処理 (送受信) 数
sema/s	1 秒当りのセマフォ処理数

例 — プロセス間通信をチェックする

次の例は、sar -m コマンドからの一部省略した出力を示します。

```
$ sar -m
SunOS venus 5.8 Generic sun4u    09/07/99

00:00:00   msg/s  sema/s
01:00:00    0.00    0.00
02:00:02    0.00    0.00
03:00:00    0.00    0.00
04:00:00    0.00    0.00
05:00:01    0.00    0.00
06:00:00    0.00    0.00

Average     0.00    0.00
```

▼ ページイン動作をチェックする方法 (sar -p)

sar -p コマンドを使用すると、保護フォルトや変換フォルトを含むページイン動作が表示されます。

```
$ sar -p
00:00:00  atch/s  pgin/s ppgin/s  pflt/s  vflt/s slock/s
01:00:00    0.07    0.00    0.00    0.21    0.39    0.00
```

表 36–14 は、–p オプションを指定したときに表示される統計情報を示します。

表 36–14 sar -p コマンドからの出力

フィールド名	説明
atch/s	現在メモリーに入っているページを回収して満たされる 1 秒当りのページフォルト数 (1 秒当りの付加数)。この例には、空きリストから無効なページを回収し、別のプロセスに現在使用中のテキストページを共有する処理が含まれる (たとえば、複数のプロセスが同じプログラムテキストにアクセスしている場合など)
pgin/s	ファイルシステムがページイン要求を受信する 1 秒当りの回数
ppgin/s	ページインされる 1 秒当りのページ数。ソフトロック要求 (slock/s を参照) などの 1 つのページイン要求や、大型ブロックサイズでは、複数のページがページインされることがある
pflt/s	保護エラーによるページフォルト数。保護フォルトの例には、ページへの不正なアクセスや、「書き込み時コピー」などがある。通常、この数値は主に「書き込み時コピー」からなっている
vflt/s	1 秒当りのアドレス変換ページフォルト数。これは、有効性フォルトと呼ばれ、所定の仮想アドレスに有効なプロセステーブルエントリが存在しないときに発生する
slock/s	物理入出力を要求するソフトウェアロック要求によって発生する 1 秒当りのフォルト数。ソフトロック要求の発生例には、ディスクからメモリーへのデータ転送などがある。システムはデータを受信しないページをロックするので、別のプロセスはそれを回収して使用できない

例 — ページイン動作をチェックする

次の例は、sar -p からの一部省略した出力を示します。

```
$ sar -p
SunOS venus 5.8 Generic sun4u    09/07/99

00:00:00  atch/s  pgin/s ppgin/s  pflt/s  vflt/s slock/s
01:00:00    0.07    0.00    0.00    0.21    0.39    0.00
02:00:02    0.07    0.00    0.00    0.21    0.39    0.00
03:00:00    0.32    0.00    0.00    1.10    2.48    0.00
04:00:00    0.09    0.00    0.00    0.32    0.57    0.00
05:00:01    0.07    0.00    0.00    0.21    0.39    0.00
06:00:00    0.07    0.00    0.00    0.21    0.39    0.00
```

(続く)

```
Average      0.26     0.20     0.30     0.92     1.78     0.00
```

▼ 待ち行列動作をチェックする方法 (sar -q)

sar -q コマンドを使用すると、待ち行列に要求が入っている平均待ち行列の長さと、その間の時間の割合が表示されます。

```
$ sar -q
00:00:00 runq-sz %runocc swpq-sz %swpocc
01:00:00
```

注 - システムに空きメモリーが十分ない場合でも、スワップアウトされた軽量プロセス数が 0 より大きい場合があります。この状態は、休眠中の軽量プロセスがスワップアウトされ処理されない場合 (プロセスや軽量プロセスが休眠中であり、キーボードやマウスの入力を待機中の場合など) に発生します。

表 36–15 は、–q オプションを指定する場合の出力を示します。

表 36–15 sar -q コマンドの出力

フィールド名	説明
runq-sz	CPU を実行するためにメモリー内で待機中のカーネルスレッド数。通常、この値は 2 未満になる。値が常に 2 より大きい場合は、システムが CPU の限界に到達している可能性がある
%runocc	ディスパッチ待ち行列が使用されている時間の割合
swpq-sz	スワップアウトされた平均軽量プロセス数
%swpocc	軽量プロセスがスワップアウトされた時間の割合

例 — 待ち行列動作をチェックする

次の例は、sar -q コマンドからの一部省略した出力を示します。%runocc の値が大きく (90 パーセント超)、runq-sz が 2 より大きい場合は、CPU の負荷が大きく、応答速度が低下しています。この場合は、CPU の容量を増やしてシステムの応答速度を適正化する必要があります。

```
$ sar -q
SunOS venus 5.8 Generic sun4u    09/07/99

00:00:00 runq-sz %runocc swpq-sz %swpocc
01:00:00
02:00:02
03:00:00     1.0       0
04:00:00
05:00:01     1.0       0
06:00:00

Average      1.3       0
```

▼ 未使用のメモリーをチェックする方法 (sar -r)

sar -r コマンドを使用すると、現在使用されていないメモリーページ数とスワップファイルのディスクブロック数が表示されます。

```
$ sar -r
00:00:00 freemem freeswap
01:00:00    2135   401922
```

表 36–16 は、–r オプションを使用する場合の出力を示します。

表 36–16 sar -r コマンドからの出力

フィールド名	説明
freemem	コマンドによるサンプル収集間隔の間にユーザープロセスに利用できる平均メモリーページ数。ページサイズはマシンに応じて異なる
freeswap	ページスワップに使用可能な 512 バイトのディスクブロック数

例 — 未使用のメモリーをチェックする

次の例は、sar -r コマンドからの出力を示します。

```
$ sar -r
SunOS venus 5.8 Generic sun4u    09/07/99

00:00:00 freemem freeswap
01:00:00    2135   401922
02:00:02    2137   401949
03:00:00    2137   402006
04:00:00    2139   401923
05:00:01    2138   402033
06:00:00    2137   401919

Average     2500   399914
```

▼ CPU の使用状況をチェックする方法 (sar -u)

sar -u コマンドを使用すると、CPU の使用状況が表示されます。

```
$ sar -u
00:00:00    %usr    %sys    %wio   %idle
01:00:00       0       0       0     100
```

(オプションを指定しない sar コマンドは、sar -u と同じです)。所定の瞬間に、プロセッサはビジー状態またはアイドル状態になっています。ビジー状態のときは、プロセッサはユーザーモードまたはシステムモードになっています。アイドル状態のときは、プロセッサは入出力の完了を待っているか、何も処理することがないので「待機」している状態です。

表 36–17 は、–u オプションを使用する場合の出力を示します。

表 36–17 sar -u コマンドからの出力

フィールド名	説明
%sys	プロセッサがシステムモードになっている時間の割合が表示される
%user	プロセッサがユーザーモードになっている時間の割合が表示される

表 36–17　sar -u コマンドからの出力　続く

フィールド名	説明
%wio	プロセッサがアイドル状態で入出力の完了を待っている時間の割合が表示される
%idle	プロセッサがアイドル状態で入出力を待っていない時間の割合が表示される

一般に、%wio の値が大きい場合は、ディスクの処理速度が低下していることを意味します。

例 — CPU の使用状況をチェックする

次の例は、sar -u コマンドからの出力を示します。

```
$ sar -u
SunOS venus 5.8 Generic sun4u    09/07/99

00:00:00    %usr    %sys    %wio   %idle
01:00:00       0       0       0     100
02:00:02       0       0       0     100
03:00:00       0       0       0     100
04:00:00       0       0       0     100
05:00:01       0       0       0     100
06:00:00       0       0       0     100
07:00:00       0       0       0     100
08:00:01       0       0       0     100
08:20:00       0       0       0     100
08:40:00       0       0       0     100
09:00:00       0       0       0     100
09:20:00       0       0       0     100
09:40:00       0       0       0     100
10:00:00       0       0       0     100
10:20:00       0       0       0     100
10:40:01       0       0       0     100
11:00:00       5       2      10      82

Average        0       0       0     100
```

▼ システムテーブルの状態をチェックする方法 (sar -v)

sar -v コマンドを使用すると、プロセステーブル、i ノードテーブル、ファイルテーブル、および共有メモリーレコードテーブルの状態が表示されます。

```
$ sar -v
00:00:00  proc-sz    ov  inod-sz    ov  file-sz    ov   lock-sz
01:00:00   43/922     0 2984/4236    0  322/322     0    0/0
```

表 36–18 は、–v オプションを使用する場合の出力を示します。

表 36–18 sar -v コマンドからの出力

フィールド名	説明
proc-sz	現在カーネル内で使用されているか、割り当てられているプロセスエントリ (proc 構造) の数
inod-sz	メモリー内の合計 i ノード数とカーネル内で割り当て済みの最大 i ノード数の比。これは厳密な上限ではなく、超えることもできる
file-sz	開いているシステムファイルテーブルのサイズ。ファイルテーブルには領域が動的に割り当てられるので、sz は 0 として表示される
ov	現在カーネル内で使用されているか割り当てられている共有メモリーレコードテーブルのエントリ数。共有メモリーレコードテーブルには領域が動的に割り当てられるので、sz は 0 として表示される
lock-sz	現在カーネル内で使用されているか割り当てられている共有メモリーレコードテーブルのエントリ数。共有メモリーレコードテーブルには領域が動的に割り当てられるので、sz は 0 として表示される

例 — システムテーブルの状態をチェックする

次の例は、sar -v コマンドからの一部省略した出力を示します。この例は、すべてのテーブルに十分なサイズがあり、オーバーフローは発生しないことを示します。これらのテーブルには、いずれも物理メモリーの容量に基づいて領域が動的に割り当てられます。

```
$ sar -v
SunOS venus 5.8 Generic sun4u    09/07/99

00:00:00  proc-sz    ov  inod-sz    ov  file-sz    ov   lock-sz
01:00:00   43/922     0 2984/4236    0  322/322     0    0/0
02:00:02   43/922     0 2984/4236    0  322/322     0    0/0
03:00:00   43/922     0 2986/4236    0  323/323     0    0/0
04:00:00   43/922     0 2987/4236    0  322/322     0    0/0
```

(続く)

```
05:00:01   43/922      0 2987/4236     0  322/322      0     0/0
06:00:00   43/922      0 2987/4236     0  322/322      0     0/0
```

▼ スワップ動作をチェックする方法 (sar -w)

sar -w コマンドを使用すると、スワッピングと切り替え動作が表示されます。

```
$ sar -w
00:00:00 swpin/s bswin/s swpot/s bswot/s pswch/s
01:00:00    0.00     0.0    0.00     0.0       22
```

表 36–19 は、目標の値と規則を示します。

表 36–19 sar -w コマンドからの出力

フィールド名	説明
swpin/s	メモリーに転送される 1 秒当りの軽量プロセス数
bswin/s	メモリーからスワップアウトされる 1 秒当りの平均プロセス数。この数値が 1 より大きい場合は、メモリーを増やす必要がある
swpot/s	メモリーからスワップアウトされる 1 秒当りの平均プロセス数。この数値が 1 より大きい場合は、メモリーを増やす必要がある
bswot/s	スワップアウト用に転送される 1 秒当りのブロック数
pswch/s	1 秒当りのカーネルスレッド切り替え数

注 - すべてのプロセスのスワップインには、プロセスの初期化が含まれます。

例 — スワップ動作をチェックする

次の例は、sar -w コマンドからの出力を示します。

```
$ sar -w
SunOS venus 5.8 Generic sun4u    09/07/99

00:00:00 swpin/s bswin/s swpot/s bswot/s pswch/s
01:00:00    0.00     0.0    0.00     0.0      22
02:00:02    0.00     0.0    0.00     0.0      22
03:00:00    0.00     0.0    0.00     0.0      22
04:00:00    0.00     0.0    0.00     0.0      22
05:00:01    0.00     0.0    0.00     0.0      22
06:00:00    0.00     0.0    0.00     0.0      22
07:00:00    0.00     0.0    0.00     0.0      22
08:00:01    0.00     0.0    0.00     0.0      22
08:20:00    0.00     0.0    0.00     0.0      22
08:40:00    0.00     0.0    0.00     0.0      22
09:00:00    0.00     0.0    0.00     0.0      22
09:20:00    0.00     0.0    0.00     0.0      22
09:40:00    0.00     0.0    0.00     0.0      22
10:00:00    0.00     0.0    0.00     0.0      22
10:20:00    0.00     0.0    0.00     0.0      22
10:40:01    0.00     0.0    0.00     0.0      23
11:00:00    0.00     0.0    0.00     0.0     144

Average     0.00     0.0    0.00     0.0      24
```

▼ 端末動作をチェックする方法 (sar -y)

sar -y コマンドを使用すると、端末デバイスの動作を監視できます。

```
$ sar -y
00:00:00 rawch/s canch/s outch/s rcvin/s xmtin/s mdmin/s
01:00:00       0       0       0       0       0       0
```

大量の端末入出力がある場合は、このレポートを使用して不良な行がないかどうかを判別できます。表 36–20 は、記録される動作の定義を示します。

表 36–20 sar -y コマンドからの出力

フィールド名	説明
rawch/s	1 秒当りの入力文字数 (raw 待ち行列)
canch/s	標準待ち行列で処理される 1 秒当りの文字数
outch/s	1 秒当りの出力文字数 (出力待ち行列)
rcvin/s	1 秒当りの受信側ハードウェア割り込み数

表 36–20　sar -y コマンドからの出力　続く

フィールド名	説明
xmtin/s	1 秒当りの送信側ハードウェア割り込み数
mdmin/s	1 秒当りのモデム割り込み数

1 秒当りのモデム割り込み数 (mdmin/s) は 0 に近く、1 秒当りの送受信割り込み数 (xmtin/s と rcvin/s) は、それぞれ着信または発信文字数以下になるはずです。そうでない場合は、不良回線がないかどうかをチェックしてください。

例 — 端末動作をチェックする

次の例は、sar -y コマンドからの一部省略した出力を示します。

```
$ sar -y
SunOS venus 5.8 Generic sun4u    09/07/99

00:00:00 rawch/s canch/s outch/s rcvin/s xmtin/s mdmin/s
01:00:00       0       0       0       0       0       0
02:00:02       0       0       0       0       0       0
03:00:00       0       0       0       0       0       0
04:00:00       0       0       0       0       0       0
05:00:01       0       0       0       0       0       0
06:00:00       0       0       0       0       0       0
07:00:00       0       0       0       0       0       0
08:00:01       0       0       0       0       0       0
08:20:00       0       0       0       0       0       0
08:40:00       0       0       0       0       0       0
09:00:00       0       0       0       0       0       0
09:20:00       0       0       0       0       0       0
09:40:00       0       0       0       0       0       0
10:00:00       0       0       0       0       0       0
10:20:00       0       0       0       0       0       0
10:40:01       0       0      20       0       0       0

Average        0       0       3       0       0       0
```

▼ システム全体の性能をチェックする方法 (sar -A)

sar -A コマンドを使用すると、システム全体の性能が表示されます。

このコマンドを使用すると、全体像を把握できます。複数のタイムセグメントからのデータが表示される場合は、レポートに平均値が含まれます。

システム動作データの自動収集 (sar)

システム動作データを自動的に収集するには、sadc、sa1、sa2 という 3 つのコマンドを使用します。

sadc データ収集ユーティリティは、システム動作に関するデータを定期的に収集し、24 時間ごとに 1 つのファイルに 2 進形式で保存します。sadc を定期的に (通常は 1 時間ごとに) 実行するだけでなく、システムがマルチユーザーモードでブートするときにも実行するように設定できます。データファイルは、ディレクトリ /usr/adm/sa に格納されます。各ファイルには sa*dd* という名前が与えられます。この場合、*dd* は現在の日付です。このコマンドの書式は次のとおりです。

```
/usr/lib/sa/sadc [t n] [ofile]
```

このコマンドは、*t* 秒 (*t* は 5 秒より長くする必要があります) 間隔でサンプルデータを *n* 回収集します。次に、ファイル *ofile* または標準出力に 2 進形式で書き込みます。*t* と *n* を省略すると、特殊ファイルに 1 度だけ書き込まれます。

ブート時に sadc を実行する

カウンタが 0 にリセットされるときから統計情報を記録するために、sadc コマンドをシステムのブート時に実行する必要があります。sadc をブート時に確実に実行するには、日ごとのデータファイルにレコードを書き込むコマンド行を /etc/init.d/perf ファイルに入れなければなりません。

コマンドエントリの書式は次のとおりです。

```
su sys -c "/usr/lib/sa/sadc /usr/adm/sa/sa`date +5d`"
```

sa1 を使用して sadc を定期的に実行する

定期的にレコードを生成するには、sadc を定期的に実行する必要があります。そのためには、シェルスクリプト sa1 を呼び出す 1 行を /var/spool/cron/sys ファイルに挿入するのが最も簡単な方法です。このスクリプトは sadc を起動し、日ごとのデータファイル /var/adm/sa/sa*dd* に書き込みます。書式は次のとおりです。

```
/usr/lib/sa/sa1 [t n]
```

引数 *t* と *n* を指定すると、レコードは *t* 秒間隔で *n* 回書き込まれます。この 2 つの引数を省略すると、レコードは 1 度しか書き込まれません。

sa2 を使用してレポートを生成する

もう 1 つのシェルスクリプト sa2 は、2 進データファイルではなくレポートを生成します。sa2 コマンドは sar コマンドを呼び出して、レポートファイルに ASCII 出力を書き込みます。

システム動作データを収集する (sar)

sar コマンドを使用すると、システム動作データそのものを収集するか、sadc で作成された日ごとの動作ファイルに収集された情報をレポートできます。

sar コマンドの書式は次のとおりです。

```
sar [-aAbcdgkmpqruvwy] [-o file] t [n]
```

```
sar [-aAbcdgkmpqruvwy] [-s time] [-e time] [-i sec] [-f file]
```

次の sar コマンドは、オペレーティングシステム内の累積動作カウンタから *t* 秒間隔で *n* 回データを収集します (*t* が 5 秒ないし 5 秒以上の値でないと、コマンドそのものがサンプルに影響を与えることがあります)。サンプルの収集間隔を指定しなければなりません。指定しないと、このコマンドは第 2 の書式に従って動作します。*n* のデフォルト値は 1 です。次の例では、10 秒間隔で 2 つのサンプルが収集されます。-o オプションを指定すると、サンプルは 2 進形式でファイルに保存されます。

```
$ sar -u 10 2
```

その他に、sar では次の点に注意する必要があります。

- サンプル間隔またはサンプル数を指定しなければ、sar はデータを以前に記録されたファイルから抽出します。その場合は、-f オプションで指定したファイル、または、デフォルトでは最新日付分の標準の日ごとの動作ファイル /var/adm/sa/sa*dd* から抽出されます。
- -s オプションと -e オプションでは、レポートの開始時刻と終了時刻を定義します。開始時刻と終了時刻の書式は *hh*[*:mm*[*:ss*]] です (この場合、*h*、*m*、*s* は、それぞれ時間、分、秒を表します)。
- -i オプションでは、レコードの選択間隔を秒単位で指定します。-i オプションを指定しなければ、日ごとの動作ファイル内で見つかったすべての間隔がレポートされます。

表 36–21 に sar コマンドのオプションとその動作を示します。

表 36–21 sar コマンドのオプション

オプション	動作
–a	ファイルアクセス操作をチェックする
–b	バッファー動作をチェックする
–c	システムコールをチェックする
–d	各ブロックデバイスの動作をチェックする
–g	ページアウトとメモリーの解放をチェックする
–k	カーネルメモリーの割り当てをチェックする
–m	プロセス間通信をチェックする
-p	スワップとディスパッチ動作をチェックする
–p	待ち行列動作をチェックする
–r	未使用メモリーをチェックする
–u	CPU の使用率をチェックする
–nv	システムテーブルの状態をチェックする
–w	ボリュームのスワッピングと切り替えをチェックする
–y	端末動作をチェックする
–A	システム全体の性能をリポートする (すべてのオプションを入力した場合と同じです)

オプションを使用しなければ、–u オプションを指定してコマンドを呼び出すのと同じです。

▼ 自動データ収集を設定する方法

1. スーパーユーザーになります。

2. /etc/init.d/perf ファイルを編集してすべての行のコメント指定を解除します。

   このバージョンの sadc コマンドは、カウンタが 0 にリセットされる時間 (ブート時) を示す特殊なレコードを書き込みます。sadc の出力はファイル sa*dd* に格納されます (この場合、*dd* は現在の日付です)。このファイルは、毎日のシステム動作の記録となります。

3. /var/spool/cron/crontabs/sys ファイル (システムの crontab ファイル) を編集します。次の行をコメント解除します。

```
# 0 * * * 0-6 /usr/lib/sa/sa1
# 20,40 8-17 * * 1-5 /usr/lib/sa/sa1
# 5 18 * * 1-5 /usr/lib/sa/sa2 -s 8:00 -e 18:01 -i 1200 -A
```

   第 1 のエントリは、時間に関するレコードを、1 週 7 日間の 1 時間ごとに /var/adm/sa/sa*dd* に書き込みます。

   第 2 のエントリは、月曜から金曜の午前 8 時から午後 5 時までのピーク作業時間中に、各正時から 20 分過ぎと 40 分過ぎの 2 度に渡り、レコードを /var/adm/sa/sa*dd* に書き込みます。

   したがって、この 2 つの crontab エントリによって、月曜から金曜までは午前 8 時から午後 5 時まで 20 分ごとに、それ以外の曜日は 1 時間ごとに、レコードが /var/adm/sa/sa*dd* に書き込まれます。これらのデフォルトは、必要に応じて変更できます。

第 37 章

Solaris ソフトウェアで発生する問題の解決

ここでは、Solaris ソフトウェアで発生する問題を解決する手順を説明します。次の章が含まれます。

第 38 章	一般的なソフトウェアの問題を解決する方法の概要と、システムクラッシュを解決する手順を説明します。
第 39 章	クラッシュダンプを保存する手順とシステムエラー記録をカスタマイズする手順を説明します。
第 40 章	一般的なソフトウェアの問題 (システムがハングする、システムがブートしないなど) の状況と可能な解決策を説明します。
第 41 章	一般的なファイルアクセスについての問題 (正しくないコマンド検索パスやファイルのアクセス権など) の解決策を説明します。
第 42 章	一般的なプリンタの問題 (出力が出ない、出力がおかしいなど) の解決策を説明します。
第 43 章	ファイルシステムに関連する問題について、特定の `fsck` エラーメッセージとその解決策を説明します。
第 44 章	ソフトウェアパッケージを追加および削除するときに発生する問題について、特定のエラーメッセージと可能な解決策を説明します。

第 38 章

ソフトウェアの問題解決の概要

この章では、ソフトウェアの問題の解決についての概要を説明します。システムクラッシュの問題の解決とシステムメッセージの表示などが含まれます。

この章の内容は次のとおりです。

ソフトウェアの問題の解決方法の参照先

ソフトウェアの問題の解決手順については、次の章を参照してください。

- 第 39 章
- 第 40 章
- 第 41 章
- 第 42 章
- 第 43 章
- 第 44 章

システムの問題解決に関する新機能

この節では、Solaris 8 リリースで使用できるようになった、システムの問題解決に関する新機能について説明します。

apptrace

新しいアプリケーションデバッグツール apptrace を使用して、アプリケーション開発者やシステムサポート担当者が、Solaris 共有ライブラリの呼び出しを追跡することによって、アプリケーションやシステムの問題をデバッグできます。この追跡では、障害の発生場所に至るまでの一連のイベントを表示できます。

apptrace ツールの呼び出し追跡機能は、以前の sotruss コマンドよりも信頼性が高くなっています。さらに、apptrace ツールでは、Solaris ライブラリインタフェースに対する関数の引数、戻り値、エラー状況の表示が改善されています。

デフォルトでは、apptrace は、コマンド行に指定した実行可能オブジェクトからそのオブジェクトが依存する各共有ライブラリへの直接呼び出しを追跡します。

詳細は、apptrace(1) のマニュアルページを参照してください。

コアファイル管理の改善

coreadm コマンド

このリリースでは coreadm コマンドが新しく導入されました。coreadm コマンドでは、コアファイルの命名規則が柔軟になり、コアファイルの保存方法が改善されます。たとえば、coreadm コマンドでは、すべてのプロセスコアファイルを同じシステムディレクトリに置くようにシステムを構成できます。そのため、Solaris のプロセスやデーモンが異常終了した場合に、特定のディレクトリにあるコアファイルを調べればよくなり問題の追跡が容易になります。

構成可能な 2 つの新しい core ファイルパス (プロセス別パスとグローバルパス) を、別々に有効にしたり無効にしたりできます。プロセスが異常終了すると、以前の Solaris リリースと同様に core ファイルが現在のディレクトリに作成されます。ただし、グローバルのコアファイルパスが有効で /corefiles/core に設定され

ている場合、プロセスが異常終了するたびに 2 つのコアファイルが、1 つは現在の作業ディレクトリに、もう 1 つは /corefiles ディレクトリに作成されます。

デフォルトでは Solaris のコアパスとコアファイルの保存方法は従来と同じです。

詳細は、682ページの「コアファイルの管理 (coreadm)」と coreadm(1) のマニュアルページを参照してください。

proc ツールによるコアファイルの調査

一部の proc ツールが拡張されてプロセスのコアファイルやライブプロセスが調べられるようになりました。proc ツールは、/proc ファイルシステムの機能を操作するユーティリティです。

現在、コアファイルを処理できるツールは /usr/proc/bin/pstack、pmap、pldd、pflags、pcred です。これらのツールを使用するには、プロセス ID を指定するように、コアファイルの名前をコマンド行に指定します。たとえば、次のように指定します。

```
$ ./a.out
Segmentation Fault(coredump)
$ /usr/proc/bin/pstack ./core
core './core' of 19305: ./a.out
 000108c4 main     (1, ffbef5cc, ffbef5d4, 20800, 0, 0) + 1c
 00010880 _start   (0, 0, 0, 0, 0, 0) + b8
```

proc ツールを使ってコアファイルを調べる方法の詳細は、proc(1) のマニュアルページを参照してください。

新しいリモートコンソールメッセージング機能

新しいリモートコンソール機能により、リモートシステムの問題を解決しやすくなりました。

詳細は、674ページの「リモートコンソールメッセージングを有効にする」と consadm(1M) のマニュアルページを参照してください。

システムクラッシュの問題の解決

Solaris オペレーティング環境が動作しているシステムがクラッシュした場合は、クラッシュダンプファイルを含む、可能なかぎりの情報を購入先に提供してください。

システムがクラッシュした場合の対処方法

最も重要なことは、次のとおりです。

1. システムのコンソールメッセージを書き取ります。

   システムがクラッシュした場合は、システムをリブートする前に、まずコンソール画面にメッセージが表示されていないか確認してください。このようなメッセージは、クラッシュした原因を解明するのに役立ちます。システムが自動的にリブートして、コンソールメッセージが画面から消えた場合でも、システムエラーログファイルを表示すれば、これらのメッセージをチェックできます。システムエラーログファイルは、/var/adm/messages (または /usr/adm/messages) に自動的に生成されます。システムエラーログファイルを表示する方法の詳細は、670ページの「システムメッセージを表示する方法」を参照してください。

   クラッシュが頻繁に発生して、その原因を特定できない場合は、システムのコンソールや /var/adm/messages ファイルから得られるすべての情報を収集して、購入先に問い合わせください。購入先に問い合わせるときに必要な問題解決のための情報の完全なリストについては、666ページの「システムクラッシュの問題の解決」を参照してください。

   システムのクラッシュ後にリブートが失敗する場合は、第 40 章を参照してください。

2. 次のように入力してディスクとの同期をとり、リブートします。

```
ok sync
```

   システムのクラッシュ後にリブートが失敗する場合は、第 40 章を参照してください。

3. savecore コマンドを実行して、スワップ領域に書き込まれたクラッシュ情報を保存します。

```
# savecore
```

クラッシュダンプを自動的に保存する方法については、第 39 章を参照してください。

問題の解決に使用するデータの収集

システムの問題を特定するために、次の質問に答えてください。クラッシュしたシステムの問題を解決するためのデータを収集するには、668ページの「システムクラッシュを解決するためのチェックリスト」を参照してください。

表 38–1 システムクラッシュに関するデータの収集

質問	説明
問題を再現できるか	この質問は、再現可能なテストケースは実際のハードウェア問題をデバッグするために重要であることが多いために重要である。購入先では、特殊な計測機構を使用してカーネルを構築して問題を再現し、バグを引き起こし、診断、および修正できる
Sun 以外のドライバを使用しているか	ドライバは、カーネルと同じアドレス空間で、カーネルと同じ特権で動作する。したがって、ドライバにバグがあると、システムクラッシュの原因となることがある
クラッシュの直前にシステムは何を実行していたか	システムが通常でないこと (新しい負荷テストの実行など) を行なったり、通常よりも高い負荷がシステムにかかったりした場合、クラッシュの原因となることがある
クラッシュ直前に、異常なコンソールメッセージが表示されたか	システムは、実際にクラッシュする前に問題の兆候を示すことがある。この情報は役立つことが多い
`/etc/system` ファイルに調整パラメタを追加したか	調整パラメタは、システムクラッシュの原因となることがある。たとえば、共有メモリーセグメントを増やした結果、システムが限度以上の多くのメモリーを割り当てようとした
問題は最近発生するようになったか	そうであれば、問題の原因は、システムの変更 (たとえば、新しいドライバ、新しいソフトウェア、作業負荷の変化、CPU のアップグレード、メモリーのアップグレードなど) にある可能性がある

システムクラッシュを解決するためのチェックリスト

クラッシュしたシステムの問題を解決するためのデータを収集するときは、次のチェックリストを使用します。

項目	ユーザーのデータ
コアファイルが生成されているか	
オペレーティングシステムのリリースと適切なソフトウェアアプリケーションのリリースレベルを確認する	
システムのハードウェアを確認する sun4d システムの `prtdiag` 出力を含める	
パッチはインストールされているか。そうであれば、`showrev -p` 出力を含める	
問題を再現できるか	
Sun 以外のドライバをシステムで使用しているか	
クラッシュ直前のシステムの動作は	
クラッシュ直前に、異常なコンソールメッセージが表示されたか	
`/etc/system` ファイルにパラメタを追加したか	
問題は最近発生するようになったか	

システムメッセージの表示

システムのメッセージはコンソールデバイスに表示されます。ほとんどのシステムメッセージは次の形式で表示されます。

[ID *msgid facility.priority*]

次に例を示します。

```
[ID 672855 kern.notice] syncing file systems...
```

カーネルから出されるメッセージには、カーネルモジュール名が次のように表示されます。

```
Oct 1 14:07:24 mars ufs: [ID 845546 kern.notice] alloc: /: file system full
```

システムがクラッシュすると、システムのコンソールに次のようなメッセージが表示されることがあります。

```
panic: error message
```

error message は、`crash(1M)` のマニュアルページに説明されているパニックエラーメッセージの 1 つです。

パニックメッセージより頻度は少ないですが、パニックメッセージではなく次のメッセージが表示されることがあります。

```
Watchdog reset !
```

エラー記録デーモン `syslogd` は、自動的に様々なシステムの警告やエラーをメッセージファイルに記録します。デフォルトでは、これらのシステムメッセージの多くは、システムコンソールに表示されて、`/var/adm` ディレクトリに格納されます。システム記録を設定することによって、これらのメッセージを格納する場所を指示できます。詳細は、673ページの「システムのメッセージ記録をカスタマイズする方法」を参照してください。これらのメッセージは、失敗の予兆のあるデバイスなど、システム障害をユーザーに警告できます。

`/var/adm` ディレクトリには、いくつかのメッセージファイルが含まれています。最も新しいメッセージは、`/var/adm/messages` (および `messages.0`) にあり、最も古いメッセージは、`messages.3` にあります。一定の期間 (通常は 10 日) ごとに、新しい `messages` ファイルが作成されます。`messages.0` のファイル名は

messages.1 に、messages.1 は messages.2 に、messages.2 は messages.3 にそれぞれ変更されます。その時点の /var/adm/messages.3 は削除されます。

/var/adm ディレクトリは、メッセージやクラッシュダンプなどのデータを含んでいる大きなファイルを格納するため、多くのディスク容量を消費します。/var/adm ディレクトリが大きくならないようにするために、そして将来のクラッシュダンプが保存できるようにするために、不要なファイルを定期的に削除しなければなりません。crontab を使用すれば、この作業は自動化できます。この作業を自動化する方法については、520ページの「クラッシュダンプファイルを削除する方法」と第 30 章を参照してください。

▼ システムメッセージを表示する方法

システムクラッシュまたはリブートによって生成された最近のメッセージを表示するには、dmesg コマンドを使用します。

```
$ dmesg
```

あるいは、more コマンドを使用して、メッセージを 1 画面ごとに表示します。

```
$ more /var/adm/messages
```

詳細は、dmesg(1M) のマニュアルページを参照してください。

例 — システムメッセージを表示する

次の例は、dmesg コマンドからの出力を示しています。

```
$ dmesg
date starbug genunix: [ID 540533 kern.notice] SunOS Release 5.8 Version 64-bit
date starbug genunix: [ID 223299 kern.notice] Copyright (c) 1983-
1999 by Sun Microsystems, Inc.
date starbug genunix: [ID 678236 kern.info] Ethernet address = xx:xx:xx:xx:xx:xx
date starbug unix: [ID 389951 kern.info] mem = 131072K (0x8000000)
date starbug unix: [ID 930857 kern.info] avail mem = 122134528
date starbug rootnex: [ID 466748 kern.info] root nexus = Sun Ultra 5/10 UPA/
PCI (UltraSPARC-IIi 333MHz)
date starbug rootnex: [ID 349649 kern.info] pcipsy0 at root: UPA 0x1f 0x0
date starbug genunix: [ID 936769 kern.info] pcipsy0 is /pci@1f,0
date starbug pcipsy: [ID 370704 kern.info] PCI-device: pci@1,1, simba0
date starbug genunix: [ID 936769 kern.info] simba0 is /pci@1f,0/pci@1,1
date starbug pcipsy: [ID 370704 kern.info] PCI-device: pci@1, simba1
date starbug genunix: [ID 936769 kern.info] simba1 is /pci@1f,0/pci@1
date starbug simba: [ID 370704 kern.info] PCI-device: ide@3, uata0
date starbug genunix: [ID 936769 kern.info] uata0 is /pci@1f,0/pci@1,1/ide@3
```

(続く)

```
.
.
.
```

システムのメッセージ記録のカスタマイズ

/etc/syslog.conf ファイルを変更すると、様々なシステムプロセスが生成するエラーメッセージを記録できます。デフォルトでは、/etc/syslog.conf は、多くのシステムプロセスのメッセージが /var/adm メッセージファイルに格納されるように指示しています。クラッシュとブートのメッセージも、同様にこのファイルに格納されます。/var/adm メッセージを表示する方法については、670ページの「システムメッセージを表示する方法」を参照してください。

/etc/syslog.conf ファイルは、タブで区切られた 2 つの列から構成されています。

facility.level ...	*action*

facility.level	機能またはメッセージや状態のシステムでの出所。コンマで区切られた機能のリスト。機能の値については表 38–2 を参照。*level* は、記録する状態の重要度や優先順位を示す。優先レベルについては表 38–3 を参照
action	動作フィールドは、メッセージが転送される場所を示す

次は、デフォルトの /etc/syslog.conf ファイルの例です。

```
user.err                                        /dev/sysmsg
user.err                                        /var/adm/messages
user.alert                                      'root, operator'
```

(続く)

```
user.emerg                                      *
```

この例は、次のユーザーメッセージが自動的に記録されることを意味します。

- ユーザーエラーはコンソールに出力され、/var/adm/messages ファイルにも記録されます。
- 早急な対応が必要なユーザーメッセージ (alert) は、root ユーザーと operator ユーザーに送信されます。
- ユーザー緊急メッセージは、各ユーザーに送信されます。

最も一般的なエラー状態の出所を表 38–2 に示します。最も一般的な優先順位を、重要度順に表 38–3 に示します。

表 38–2 syslog.conf メッセージのソース機能

出所	説明
kern	カーネル
auth	認証
daemon	すべてのデーモン
mail	メールシステム
lp	スプールシステム
user	ユーザープロセス

注 - Solaris 2.6 リリース以降、/etc/syslog.conf ファイルで有効化できる syslog 機能の数の制限は解除されます。以前のリリースでは、機能の数は 20 個に制限されていました。

表 38–3 syslog.conf メッセージの優先レベル

優先順位	説明
emerg	システムの緊急事態
alert	すぐに修正が必要なエラー
crit	致命的なエラー
err	その他のエラー
info	情報メッセージ
debug	デバッグ用の出力
none	この設定は出力を記録しない

▼ システムのメッセージ記録をカスタマイズする方法

1. スーパーユーザーになります。
2. 任意のエディタで、/etc/syslog.conf ファイルを編集します。
   syslog.conf(4) のマニュアルページで説明している構文に従って、メッセージの出所、優先順位、およびメッセージの記録場所を追加または変更します。
3. 変更を保存して編集を終了します。

例 — システムのメッセージ記録をカスタマイズする

次の /etc/syslog.conf の user.emerg 機能の例は、ユーザー緊急メッセージを root ユーザーと個別のユーザーに送信します。

```
user.emerg                                        'root, *'
```

リモートコンソールメッセージングを有効にする

次の新しいリモートコンソール機能を使うと、リモートシステムの問題を解決しやすくなります。

- consadm コマンドでは、補助 (またはリモート) コンソールとしてシリアルデバイスを選択できます。consadm コマンドを使用して、システム管理者は 1 つまたは複数のシリアルポートを構成して、出力先が変更されたコンソールメッセージを表示したり、システムの実行レベルが変わったときに sulogin セッションをサポートしたりできます。この機能を使用して、モデム付きのシリアルポートにダイヤルインしてコンソールメッセージを監視し、init 状態の変更を表示できます。(詳細は、sulogin(1M) のマニュアルページと次の詳しい手順を参照してください。)

  補助コンソールとして構成されたポートからシステムにログインすることもできますが、このポートは主に、デフォルトコンソールに表示される情報を表示する出力デバイスです。ブートスクリプトやその他のアプリケーションがデフォルトコンソールに対して読み書きを行う場合、書き込み出力はすべての補助コンソールに出力されますが、入力はデフォルトコンソールからだけ読み込まれます。(対話型ログインセッションでの consadm コマンドの使い方については、676ページの「対話型ログインセッション中に consadm コマンドを使用する」を参照してください。)

- コンソール出力はカーネルメッセージと syslog メッセージからなり、新しい仮想デバイス /dev/sysmsg に書き込まれます。さらに、rc スクリプト起動メッセージが /dev/msglog に書き込まれます。以前のリリースでは、これらのメッセージはすべて /dev/console に書き込まれていました。

  スクリプトメッセージを補助コンソールに表示したい場合は、コンソール出力を /dev/console に出力しているスクリプトで出力先を /dev/msglog に変更する必要があります。メッセージ出力を補助デバイスに出力変更したい場合は、/dev/console を参照しているプログラムで syslog() または strlog() を使用するように明示的に変更してください。

- consadm コマンドは、デーモンを実行して補助コンソールデバイスを監視します。補助コンソールに指定された表示デバイスがハングアップしたりキャリア信号がなくなって切り離されると、そのデバイスは補助コンソールデバイスのリストから削除され、アクティブでなくなります。1 つまたは複数の補助コンソールを有効にしても、メッセージがデフォルトコンソールに表示されなくなるわけではありません。メッセージは引き続き /dev/console に出力されます。

実行レベルの変更中に補助コンソールメッセージングを使用する

実行レベルの変更中に補助コンソールメッセージングを使う場合は、次の点に注意してください。

- システムのブート時に実行する rc スクリプトにユーザーの入力がある場合は、補助コンソールから入力を行うことはできません。入力はデフォルトコンソールから行う必要があります。
- 実行レベルの変更中に、スーパーユーザーパスワード入力を要求するために sulogin プログラムが init によって呼び出されます。このプログラムは、デフォルトのコンソールデバイスだけでなく各補助デバイスにもスーパーユーザーパスワードの入力要求を送信するように変更されています。
- システムがシングルユーザーモードで動作し、1 つまたは複数の補助コンソールが consadm コマンドによって有効になっていると、最初のデバイスでコンソールログインセッションが実行され、正確なスーパーユーザーパスワードを要求する sulogin プロンプトが表示されます。コンソールデバイスから正しいパスワードを受け取ると、sulogin は他のすべてのコンソールデバイスからの入力を受信できないようにします。
- コンソールの 1 つがシングルユーザー特権を取得すると、デフォルトコンソールとその他の補助コンソールにメッセージが出力されます。このメッセージは、どのデバイスから正しいスーパーユーザーパスワードが入力され、コンソールになったかを示します。シングルユーザーシェルが動作する補助コンソールのキャリア信号が失われると、次のどちらかのアクションが起ることがあります。
  - 補助コンソールが実行レベル 1 のシステムを表している場合は、システムはデフォルトの実行レベルに移行します。
  - 補助コンソールが実行レベル S のシステムを表している場合は、シェルから init s または shutdown コマンドが入力されたデバイスに「ENTER RUN LEVEL (0-6, s or S):」というメッセージが表示されます。このデバイスのキャリア信号も失われている場合は、キャリア信号を復活して正確な実行レベルを入力する必要があります。init や shutdown コマンドを実行しても、実行レベルプロンプトが再表示されることはありません。
- シリアルポートを使用してシステムにログインしている場合には、init または shutdownコマンドを使用して別の実行レベルに移行すると、このデバイスが補助コンソールかどうかに関係なくログインセッションは失われます。この状況は、補助コンソール機能がない Solaris リリースと同じです。

- consadm コマンドを使って補助コンソールにするデバイスを選択すると、システムをリブートするか補助コンソールの選択を解除するまで、そのデバイスは補助コンソールとして有効です。ただし、consadm コマンドには、複数のシステムリブートにまたがってデバイスを補助コンソールとして使用するオプションがあります。(詳しい手順については、次の内容を参照してください。)

対話型ログインセッション中に consadm コマンドを使用する

シリアルポートに接続されている端末からシステムにログインしてから consadm コマンドを使ってこの端末にコンソールメッセージを表示して対話型ログインセッションを行う場合、次の点に注意してください。

- この端末で対話型ログインセッションを行う場合、補助コンソールがアクティブだと、コンソールメッセージは /dev/sysmsg デバイスまたは /dev/msglog デバイスに送られます。
- この端末からコマンドを発行すると、入力はデフォルトコンソール (/dev/console) ではなく対話型セッションに送られます。
- init コマンドを実行して実行レベルを変更すると、リモートコンソールソフトウェアは対話型セッションを終了し、sulogin プログラムを実行します。この時点では、入力はこの端末からだけ可能で、入力はコンソールデバイスから行われたかのように扱われます。そのため、675ページの「実行レベルの変更中に補助コンソールメッセージングを使用する」の説明のとおりに、sulogin プログラムにパスワードを入力できます。

  次に、(補助) 端末から正しいパスワードを入力すると、補助コンソールは、対話型 sulogin セッションを実行し、デフォルトコンソールおよび競合する補助コンソールを使えなくします。つまり、その端末は実質的にシステムコンソールとして機能します。
- この端末から実行レベル 3 または別の実行レベルに変更できます。実行レベルを変更すると、すべてのコンソールデバイスで sulogin が再び実行されます。終了したり、システムが実行レベル 3 で起動されるように指定すると、どの補助コンソールからも入力を行えなくなります。すべての補助コンソールはコンソールメッセージを表示するだけのデバイスに戻ります。

  システムが起動する際には、デフォルトのコンソールデバイスから rc スクリプトに情報を入力する必要があります。システムが再び起動すると login プログラムがシリアルポートで実行されるため、別の対話型セッションを開始できま

す。そのデバイスを補助コンソールに指定していれば、コンソールメッセージはその端末に引き続き出力されます。ただし、端末からの入力はすべて対話型セッションに送られます。

▼ 補助 (リモート) コンソールを有効にする方法

consadm デーモンは、consadm コマンドで補助コンソールを追加するまでポートの監視を開始しません。セキュリティ機能として、コンソールメッセージは、キャリア信号が失われるまでか、補助コンソールデバイスの選択が解除されるまでの間だけ出力変更されます。そのため、consadm コマンドを使うには、そのポートでキャリア信号が確立されている必要があります。

補助コンソールを有効にする方法の詳細は、consadm(1M) のマニュアルページを参照してください。

1. スーパーユーザーとしてシステムにログインします。
2. 補助コンソールを有効にします。

```
# consadm -a devicename
```

3. 現在の接続が補助コンソールであることを確認します。

```
# consadm
```

例 — 補助 (リモート) コンソールを有効にする

```
# consadm -a /dev/term/a
# consadm
 /dev/term/a
```

▼ 補助コンソールのリストを表示する方法

1. スーパーユーザーとしてシステムにログインします。
2. 次のどちらかの手順に従います。
   a. 補助コンソールのリストを表示します。

```
# consadm
/dev/term/a
```

b. 持続的補助コンソールのリストを表示します。

```
# consadm -p
/dev/term/b
```

▼ システムリブート後も補助 (リモート) コンソールを有効にする方法

1. スーパーユーザーとしてシステムにログインします。
2. 複数のシステムリブート後も補助コンソールを有効にします。

   ```
   # consadm -a -p devicename
   ```

   このデバイスが持続的な補助コンソールのリストに追加されます。
3. デバイスが持続的な補助コンソールのリストに追加されているか確認します。

   ```
   # consadm
   ```

例 — システムリブート後も補助 (リモート) コンソールを有効にする

```
# consadm -a -p /dev/term/a
# consadm
/dev/term/a
```

▼ 補助 (リモート) コンソールを無効にする方法

1. スーパーユーザーとしてシステムにログインします。

2. 次のどちらかの手順に従います。

   a. 補助コンソールを無効にします。

   ```
   # consadm -d devicename
   ```

   b. 補助コンソールを無効にし、持続的な補助コンソールのリストから削除します。

   ```
   # consadm -p -d devicename
   ```

3. 補助コンソールが無効になっていることを確認します。

```
# consadm
```

例 — 補助コンソールが無効になっていることを確認します。

```
# consadm -d /dev/term/a
# consadm
```

第 39 章

システムクラッシュ情報の生成と保存

この章では、クラッシュダンプを有効または無効にする方法と、システムメッセージを表示または収集する方法について説明します。

この章で説明する手順は次のとおりです。

システムクラッシュ

ハードウェアの障害、入出力の問題、ソフトウェアエラーなどが原因でシステムがクラッシュすることがあります。システムがクラッシュすると、システムはエラーメッセージをコンソールに表示し、物理メモリーのコピーをダンプデバイスに書き

込みます。その後、システムは自動的にリブートします。システムがリブートすると、savecore コマンドが実行され、ダンプデバイスのデータを取り出して保存されたクラッシュダンプを savecore ディレクトリに書き込みます。このクラッシュダンプファイルは、サポートプロバイダにとって、問題を診断する上で貴重な情報となります。

システムクラッシュファイルとコアファイル

システムクラッシュの後で自動的に実行される savecore コマンドは、ダンプデバイスからクラッシュダンプ情報を取り出し、unix.*X* と vmcore.*X* という 1 対のファイルを作成します。*X* はダンプの通し番号です。これらのファイルは 2 つで、保存されたシステムクラッシュダンプの情報を表します。

クラッシュダンプファイルはコアファイルと混同されることがあります。コアファイルは、アプリケーションが異常終了したときに書き込まれるユーザーアプリケーションのイメージです。

クラッシュダンプファイルは、あらかじめ決められたディレクトリに保存されます。これはデフォルトでは /var/crash/*hostname* です。以前の Solaris リリースでは、システムを手動で有効にして物理メモリーのイメージをクラッシュダンプファイルに保存しない限り、システムがリブートされた時にクラッシュダンプファイルが上書きされていました。このリリースでは、クラッシュダンプファイルの保存がデフォルトで有効です。

システムクラッシュ情報は dumpadm コマンドで管理します。詳細は、688ページの「システムクラッシュダンプ情報の管理 (dumpadm)」を参照してください。

コアファイルは coreadm コマンドで管理します。詳細は、682ページの「コアファイルの管理 (coreadm)」を参照してください。

コアファイルの管理 (coreadm)

コアファイルの管理は coreadm コマンドを使用して行います。たとえば、coreadm コマンドを使用して、プロセスコアファイルをすべて同じシステムディレクトリに置くようにシステムを構成できます。Solaris のプロセスやデーモンが異常終了した場合に、特定のディレクトリにあるコアファイルを調べればよいため問題の追跡が容易になります。

Solaris のこれまでのプロセスコアダンプ機能には次の制約がありました。

- プロセス core ファイルはそれぞれの現在の作業ディレクトリに置かれるため、すべての Solaris デーモンが core ファイルを互いに上書きします (一般に Solaris デーモンは初期化の過程で chdir により作業ディレクトリをルート (/) ディレクトリに変更します)。
- statd など多くのシステムデーモンは setuid 操作を行いますが、セキュリティ上の理由から、問題が起きてもコアファイルを生成しません。

構成可能なコアファイルの設定

次の 2 つの構成可能な新しいコアファイルの設定は、個別に有効または無効にすることができます。

- プロセス別コアファイルの設定にはデフォルトで core が使用されます。この設定はデフォルトで有効になっています。プロセス別コアファイルの設定が有効になっていると、プロセスが異常終了したときに core ファイルが生成されます。プロセス別の設定は、親プロセスから新しいプロセスに継承されます。

  プロセス別コアファイルは生成されるとプロセスの所有者によって所有され、所有者には読み取り／書き込み権が与えられます。所有者だけがこのファイルを表示できます。

- グローバルコアファイルの設定にはデフォルトで core が使用されます。この設定はデフォルトで無効になっています。この設定が有効になっていると、プロセス別コアファイルの設定と同じ内容のコアファイルがグローバルコアファイルの設定に追加で作成されます。

  グローバルコアファイルは生成されるとスーパーユーザーによって所有され、スーパーユーザーだけに読み取り／書き込み権が与えられます。アクセス権のないユーザーはこのファイルを表示できません。

プロセスが異常終了すると、以前の Solaris リリースと同じように core ファイルが現在のディレクトリに作成されます。しかし、たとえば、グローバルコアファイルの設定が有効で /corefiles/core に設定されていると、プロセスが終了するたびにコアファイルが 2 つ、1 つは現在の作業ディレクトリに、1 つは /corefiles ディレクトリにそれぞれ作成されます。

デフォルトでは Solaris のコアの設定とコアファイルの保存方法は従来と同じです。

- setuid プロセスは、グローバルの設定やプロセス別の設定を使ってコアファイルを生成することはありません。

拡張されたコアファイル名

グローバルコアファイルディレクトリが有効な場合、次の表に示す変数を使って core ファイルを相互に区別できます。

変数名	変数の定義
%p	プロセス ID
%u	実効ユーザー ID
%g	実効グループ ID
%f	実行可能ファイル名
%n	システムノード名。uname -n の出力と同じ
%m	マシン名。uname -m での出力と同じ
%t	time(2) システム呼び出しの 10 進数
%%	リテラル %

たとえば、グローバルコアファイル設定が次のように設定されている場合、

```
/var/core/core.%f.%p
```

PID 12345 の sendmail プロセスが異常終了すると、次の core ファイルが作成されます。

```
/var/core/core.sendmail.12345
```

コアファイル名パターンの設定

コアファイル名パターンは、グローバルに設定したりプロセス単位で設定したりできます。さらに、この設定をシステムリブート後も有効になるように保存するかどうかを指定できます。

たとえば、次の coreadm コマンドでは、init プロセスで起動されたすべてのプロセスに対しグローバルのコアファイルパターンを設定します。このパターンは複数のシステムリブート後も有効です。

```
$ coreadm -i /var/core/core.%f.%p
```

グローバルのコア値は /etc/coreadm.conf ファイルに格納されるため、この設定値はシステムリブート後も有効になるように保存されます。

次の coreadm コマンドでは、すべてのプロセスに対しプロセス別コアファイル名パターンを設定します。

```
$ coreadm -p /var/core/core.%f.%p $$
```

$$ 記号には、現在実行中のシェルのプロセス ID を指定します。プロセス別コアファイル名パターンは、すべての子プロセスに継承されます。

グローバルまたはプロセス別のコアファイル名パターンを設定したら、これを coreadm -e コマンドで有効にする必要があります。詳細は、次の手順を参照してください。

このコマンドをユーザーの $HOME/.profile または .login ファイルに入れておけば、ユーザーのログインセッションで実行するすべてのプロセスに対しコアファイル名パターンを設定できます。

setuid プログラムを有効にしてコアファイルを作成する

coreadm コマンドを使って setuid プログラムを有効または無効にすれば、次の設定を行うことによって、すべてのシステムプロセスに対して、または各プロセスに対してコアファイルを作成できます。

- グローバル setuid オプションが有効になっていると、グローバルコアファイル設定に従って、システムのすべての setuid プログラムが core ファイルを作成します。
- プロセス別 setuid オプションが有効になっていると、プロセス別コアファイル設定に従って、特定の setuid プロセスが core ファイルを作成します。

デフォルトでは、両方のフラグが無効になっています。セキュリティ上の理由により、グローバルコアファイル設定は、/ で始まるフルパス名であることが必要です。スーパーユーザーがプロセス別コアファイルを無効にすると、個別のユーザーがコアファイルを得ることはできなくなります。

setuid コアファイルはスーパーユーザーによって所有され、スーパーユーザーだけに読み取り／書き込み権が与えられます。通常ユーザーは、たとえ setuid コア

ファイルを生成したプロセスを所有していても、それらのファイルにアクセスできません。

詳細は、coreadm(1M) のマニュアルページを参照してください。

▼ 現在のコアダンプ構成を表示する方法

現在のコアダンプ構成を表示するには、オプションを指定しないで coreadm コマンドを実行します。

```
$ coreadm
               global core file pattern: /var/core/core.%f.%p
                 init core file pattern: core
                      global core dumps: enabled
                 per-process core dumps: enabled
                global setid core dumps: enabled
           per-process setid core dumps: disabled
               global core dump logging: disabled
```

▼ コアファイル名パターンを設定する方法

1. プロセス別コアファイルを設定するのか、グローバルコアファイルを設定するのかを決めて、次のどちらかの手順に従います。

   a. プロセス別コアファイル名パターンを設定します。

   ```
   # coreadm -p $HOME/corefiles/%f.%p $$
   ```

   b. グローバルコアファイル名パターンを設定します。

   まずスーパーユーザーになる必要があります。

   ```
   # coreadm -g /var/corefiles/%f.%p
   ```

▼ コアファイル名パターンを表示する方法

現在のプロセスのコアファイル設定を知るには、次の coreadm コマンドを使用します。$$ 記号には、実行されているシェルのプロセス ID を指定します。

```
$ coreadm $$
278:    core.%f.%p
```

スーパーユーザーは coreadm *process ID* を指定すれば、どのユーザーのコアファイル設定でも照会できます。通常のユーザーは、自分のプロセスのコアファイル設定だけを照会できます。

▼ プロセス別コアファイル設定を有効にする方法

1. スーパーユーザーになります。
2. プロセス別コアファイル設定を有効にします。

```
# coreadm -e process
```

3. 現在のプロセスのコアファイル設定を表示して構成を確認します。

```
$ coreadm $$
1180:   /home/kryten/corefiles/%f.%p
```

▼ グローバルのコアファイル設定を有効にする方法

1. スーパーユーザーになります。
2. グローバルのコアファイル設定を有効にします。

```
# coreadm -e global -g /var/core/core.%f.%p
```

3. 現在のプロセスのコアファイル設定を表示して構成を確認します。

```
# coreadm
      global core file pattern: /var/core/core.%f.%p
        init core file pattern: core
             global core dumps: enabled
        per-process core dumps: enabled
       global setid core dumps: disabled
  per-process setid core dumps: disabled
      global core dump logging: disabled
```

コアファイルの問題解決

エラーメッセージ

```
NOTICE: 'set allow_setid_core = 1' in /etc/system is obsolete
NOTICE: Use the coreadm command instead of 'allow_setid_core'
```

原因

setuid コアファイルを許容する古いパラメータが /etc/system ファイルにあります。

解決方法

/etc/system ファイルから allow_setid_core=1 を削除し、coreadm コマンドを使ってグローバル setuid コアファイルの設定を有効にします。

システムクラッシュダンプ情報の管理 (dumpadm)

この節では、Solaris 環境でシステムクラッシュ情報を管理する方法を説明します。

システムクラッシュダンプ機能

この節では、Solaris 環境でシステムクラッシュダンプ情報を管理する方法を説明します。

- 新しい dumpadm コマンドを使用すると、システム管理者はオペレーティングシステムのクラッシュダンプを構成できます。dumpadm 構成パラメタでは、ダンプ内容、ダンプデバイス、クラッシュダンプファイルが保存されるディレクトリなどを指定します。dumpadm コマンドの詳細は、689ページの「dumpadm コマンド」を参照してください。
- ダンプデータは、圧縮した形式でダンプデバイスに格納されます。カーネルのクラッシュダンプイメージは 4G バイトを超える場合があります。データを圧縮することにより、ダンプが速くなり、ダンプデバイスのディスク領域も少なくてすみます。
- 専用のダンプデバイス (スワップ領域ではなく) がダンプ構成の一部にあると、クラッシュダンプファイルの保存はバックグラウンドで行われます。つまり、ブートシステムは、savecore コマンドが完了するのを待たなくても、次の手順に進むことができます。大容量のメモリーを搭載したシステムでは、savecore コマンドが完了する前にシステムが使用可能になります。
- savecore コマンドで生成されるシステムクラッシュダンプファイルは、デフォルトで保存されます。
- savecore -L コマンドは、移動中の Solaris オペレーティング環境でクラッシュダンプを取得できる新しい機能です。たとえば、性能に問題が発生しているときやサービスが停止しているときなどにメモリーのスナップショットをとって、実行中のシステムの問題を解決するのに使用します。システムが実行中で、一部のコマンドがまだ使用できる場合は、savecore -L を使用してシステムのスナップショットをダンプデバイスに保存し、クラッシュダンプファイルをただちに savecore ディレクトリに書き込むことができます。システムが実行中であるため、専用のダンプデバイスを構成してあれば、savecore -L を使用するだけでダンプを作成できます。

dumpadm コマンド

/usr/sbin/dumpadm コマンドは、システムのクラッシュダンプ構成パラメタを管理するコマンドです。次の表で dumpadm の構成パラメタを説明します。

ダンプパラメタ	説明
ダンプデバイス	システムがクラッシュしたときにダンプデータを一時的に保存するデバイス。ダンプデバイスがスワップ領域でない場合は、savecore がバックグラウンドで実行されるため、ブートプロセスの速度が上がる
savecore ディレクトリ	システムのクラッシュダンプファイルを保存するディレクトリ
ダンプ内容	ダンプするデータの種類、つまりカーネルメモリーとすべてのメモリーのどちらをダンプするかを指定する
最小空き容量	クラッシュダンプファイルを保存した後で savecore ディレクトリに必要な最小空き容量。空き容量を指定しないと、デフォルトで 1M バイトになる

詳細は、dumpadm(1M) のマニュアルページを参照してください。

dumpadm コマンドで管理するダンプ構成パラメタは、/etc/dumpadm.conf ファイルに保存されます。

注 - /etc/dumpadm.conf は、手作業で編集しないでください。システムダンプ構成の整合性が失われる恐れがあります。

---

dumpadm コマンドの動作

dumpadm コマンドは、システム起動時に /etc/init.d/savecore スクリプトによって呼び出され、/etc/dumpadm.conf ファイルの情報に基づいてクラッシュダンプパラメタの構成を行います。

このコマンドは、/dev/dump インタフェースを通してダンプデバイスとダンプ内容を初期化します。

ダンプ構成が完了すると、savecore スクリプトは、/etc/dumpadm.conf ファイルの内容を解析してクラッシュダンプファイルのディレクトリの場所を探します。次に savecore を呼び出してクラッシュダンプがあるかどうかを調べます。さらに、クラッシュダンプディレクトリにある minfree ファイルの内容も調べます。

クラッシュダンプの保存

制御構造体、アクティブなテーブル、動作中またはクラッシュしたシステムカーネルのメモリーのイメージなど、カーネルの動作状況についての情報を調べるには、crash または adb ユーティリティを使用します。crash または adb を完全に使いこなすには、カーネルについての詳細な知識が必要ですが、このマニュアルでは説明を省きます。crash ユーティリティの詳細は、crash(1M) または adb(1) のマニュアルページを参照してください。

savecore で保存したクラッシュダンプを購入先に送って、システムがクラッシュした原因を解析してもらうことも可能です。購入先にクラッシュダンプファイルを送る場合は、691ページの「システムクラッシュ情報の管理 (作業マップ)」にリストされている最初の 2 つの作業を実行してください。

次の節では、dumpadm コマンドを使ってシステムクラッシュ情報を管理する方法を説明します。

システムクラッシュ情報の管理 (作業マップ)

表 39–1 作業マップ : クラッシュダンプの管理

作業	説明	手順の説明
1. 現在のクラッシュダンプ構成を表示する	dumpadm コマンドを使用して、現在のクラッシュダンプ構成を表示する	692ページの「現在のクラッシュダンプ構成を表示する方法」
2. クラッシュダンプ構成を変更する	dumpadm コマンドを使用して、ダンプするデータの種類、システムが専用のダンプデバイスを使用するかどうか、クラッシュダンプファイルを保存するディレクトリ、およびクラッシュダンプファイルが書き込まれた後に残っていなければならない容量を指定する	693ページの「クラッシュダンプ構成を変更する方法」
3. クラッシュダンプファイルを調べる	crash コマンドを使用して、クラッシュダンプファイルを表示する	694ページの「クラッシュダンプを検査する方法」

表 39–1 作業マップ : クラッシュダンプの管理 続く

作業	説明	手順の説明
4. 完全なクラッシュダンプディレクトリから復元する	(省略可能) システムがクラッシュしたが、savecore ディレクトリに空き容量がない。それでも、一部の重要なシステムクラッシュダンプ情報を保存したい	696ページの「フルクラッシュダンプディレクトリから復元する方法 (省略可能)」
5. クラッシュダンプファイルの保存を有効または無効にする	(省略可能) dumpadm コマンドを使用して、クラッシュダンプファイルの保存を有効または無効にする。デフォルトでは、クラッシュダンプファイルは保存される	696ページの「クラッシュダンプの保存を有効または無効にする方法 (省略可能)」

▼ 現在のクラッシュダンプ構成を表示する方法

1. スーパーユーザーになります。
2. dumpadm コマンドをオプションなしで実行して、現在のクラッシュダンプ構成を表示します。

```
# dumpadm
      Dump content: kernel pages
       Dump device: /dev/dsk/c0t3d0s1 (swap)
Savecore directory: /var/pluto
  Savecore enabled: yes
```

上記の出力の意味は次のとおりです。

- ダンプの内容は、カーネルメモリーページである
- カーネルメモリーがスワップデバイス /dev/dsk/c0t3d0s1 にダンプされる。swap -l コマンドにより、すべてのスワップ領域を識別できる
- システムクラッシュダンプファイルは /var/crash/venus ディレクトリに保存される

- システムクラッシュダンプファイルの保存は有効に設定されている

▼ クラッシュダンプ構成を変更する方法

1. スーパーユーザーになります。

2. dumpadm コマンドで、現在のクラッシュダンプ構成を確認します。

```
# dumpadm
      Dump content: kernel pages
       Dump device: /dev/dsk/c0t3d0s1 (swap)
Savecore directory: /var/crash/pluto
  Savecore enabled: yes
```

上記の構成は、Solaris 8 リリースを実行するシステムのデフォルトダンプ構成です。

3. dumpadm コマンドでクラッシュダンプ構成を変更します。

```
# dumpadm -c content -d dump-device -m nnnk | nnnm | nnn% -n -s savecore-dir
```

–c *content*	ダンプするデータの種類、つまり、カーネルメモリーまたはすべてのメモリーのいずれかを指定する。デフォルトはカーネルメモリー
–d *dump-device*	システムがクラッシュしたときに、ダンプデータを一時的に保存するデバイスを指定する。デフォルトのダンプデバイスは 1 次スワップデバイス
–m *nnnk* | *nnnm* | *nnn%*	現在の savecore ディレクトリに minfree ファイルを作成することにより、クラッシュダンプファイルを保存する最小限の空き容量を指定する。このパラメタは K バイト (nnnk)、M バイト (nnnm)、またはファイルシステムサイズのパーセント (nnn%) で指定できる。savecore コマンドは、クラッシュダンプファイルを書き込む前にこのファイルを調べる。クラッシュダンプファイルを書き込むと空き容量が minfree の値より少なくなる場合、ダンプファイルは書き込まれず、エラーメッセージが記録される。このような問題を解決するには、696ページの「フルクラッシュダンプディレクトリから復元する方法 (省略可能)」を参照

-n	システムがリブートするときに、savecore を実行しないように指定する。このダンプ構成は推奨できない。システムクラッシュ情報がスワップデバイスに書き込まれているときに、savecore が実行されないと、クラッシュダンプ情報はシステムがスワップを開始すると上書きされる
-s	クラッシュダンプファイルを保存する別のディレクトリを指定する。デフォルトのディレクトリは /var/crash/hostname で、hostname は uname -n コマンドの出力

例 —クラッシュダンプ構成を変更する

次の例は、すべてのメモリーを専用のダンプデバイス /dev/dsk/c0t1d0s1 にダンプします。また、クラッシュダンプファイルを保存した後に残っていなければならない最小空き容量は、ファイルシステム容量の 10% です。

```
# dumpadm
      Dump content: kernel pages
       Dump device: /dev/dsk/c0t3d0s1 (swap)
Savecore directory: /var/crash/pluto
  Savecore enabled: yes
# dumpadm -c all -d /dev/dsk/c0t1d0s1 -m 10%
      Dump content: all pages
       Dump device: /dev/dsk/c0t1d0s1 (dedicated)
Savecore directory: /var/crash/pluto (minfree = 77071KB)
  Savecore enabled: yes
```

▼ クラッシュダンプを検査する方法

1. スーパーユーザーになります。

2. crash ユーティリティを使用して、クラッシュダンプを検査します。

```
# /usr/sbin/crash [-d crashdump-file] [-n name-list] [-w output-file]
```

–d *crashdump-file*	システムのメモリーイメージを格納するファイルを指定する。デフォルトのクラッシュダンプファイルは /dev/mem
–n *name-list*	システムのメモリーイメージへのシンボリックアクセスを調べる場合、シンボルテーブル情報を格納するテキストファイルを指定する。デフォルトのファイル名は /dev/ksyms
–w *output-file*	クラッシュセッションからの出力を格納するファイルを指定する。デフォルトは標準出力

3. クラッシュ状態情報を表示します。

```
# /usr/sbin/crash
dumpfile = /dev/mem, namelist = /dev/ksyms, outfile = stdout
> status
   .
   .
   .
> size buf proc queue
   .
   .
   .
```

例 — クラッシュダンプを検査する

次の例は、crash ユーティリティからのサンプル出力を示します。状態とバッファーについての情報、プロセス、および待ち行列のサイズが表示されます。

```
# /usr/sbin/crash
dumpfile = /dev/mem, namelist = /dev/ksyms, outfile = stdout
> status
system name: SunOS
release: 5.8
node name: earth
version: s28_25
machine name: sun4m
time of crash: Wed Jun 30 16:02:31 1999
age of system: 18 min.
panicstr:
panic registers:
 pc: 0      sp: 0
> size buf proc queue
120
```

(続く)

```
1808
96
```

▼ フルクラッシュダンプディレクトリから復元する方法 (省略可能)

ここでは、システムがクラッシュしてもメモリーイメージを格納する十分な空き容量が savecore ディレクトリにないが、それでも、一部の重要なシステムクラッシュダンプ情報を保存するものとします。

1. システムがリブートした後で、スーパーユーザーとしてログインします。

2. すでにサービスプロバイダに送ってある既存のクラッシュダンプファイルを削除して、**savecore** ディレクトリ **(**通常は /var/crash/***hostname*****)** を整理します。あるいは、savecore コマンドを実行し、十分な容量を持つ別のディレクトリを指定します **(**次の手順を参照してください**)**。

3. 手作業で savecore コマンドを実行し、必要なら別の **savecore** ディレクトリを指定します。

```
# savecore [ directory ]
```

▼ クラッシュダンプの保存を有効または無効にする方法 (省略可能)

1. スーパーユーザーになります。

2. dumpadm コマンドにより、システム上にクラッシュダンプを保存するかどうかを指定します。

例 — クラッシュダンプの保存を無効にする

次の例は、システムでのクラッシュダンプの保存を無効にします。

```
# dumpadm -n
      Dump content: all pages
       Dump device: /dev/dsk/c0t1d0s1 (dedicated)
Savecore directory: /var/crash/pluto (minfree = 77071KB)
  Savecore enabled: no
```

例 — クラッシュダンプの保存を有効にする

次の例は、システムでのクラッシュダンプの保存を有効にします。

```
# dumpadm -y
      Dump content: all pages
       Dump device: /dev/dsk/c0t1d0s1 (dedicated)
Savecore directory: /var/crash/pluto (minfree = 77071KB)
  Savecore enabled: yes
```

第 40 章

ソフトウェアで発生するさまざまな問題の解決

この章では、ときどき発生するが修正しやすい、さまざまなソフトウェアの問題について説明します。特定のソフトウェアアプリケーションや内容に関連しない問題 (リブートの失敗やファイルシステムがフルになるなど) の解決方法も含みます。これらの問題の解決方法は、この後の節で説明します。

この章の内容は次のとおりです。

リブートが失敗した場合の対処方法

システムがリブートに失敗した場合またはリブートしたがクラッシュした場合は、システムのブートを妨害しているソフトウェアまたはハードウェアの障害があると考えられます。

問題 — システムがブートしない理由	解決方法
システムが /platform/`uname -m`/kernel/unix を見つけられない	SPARC システムの PROM 内の boot-device 設定を変更しなければならない。デフォルトブートデバイスを変更するには、『*Solaris* のシステム管理 *(*第 *1* 巻*)*』の「SPARC: システムのブート (手順)」または「Intel: システムのブート (手順)」を参照
IA システムで、デフォルトのブートデバイスが存在しない。「Not a UFS filesystem.」というメッセージが表示される	Configuration Assistant/Boot (構成用補助) フロッピーディスクを使用してシステムをブートし、ブートするディスクを選択する
/etc/passwd ファイル内に無効なエントリが存在する	無効な passwd ファイルから復元する方法については、『*Solaris* のシステム管理 *(*第 *1* 巻*)*』の「システムのシャットダウンとブート (概要)」を参照
ディスクなどのデバイスに、ハードウェアの問題がある	ハードウェアの接続を確認する ■ 装置が接続されていることを確認する ■ すべてのスイッチが適切に設定されていることを確認する ■ すべてのコネクタおよびケーブル (Ethernet ケーブルも含む) を検査する ■ すべて異常がなければ、システムの電源を切り、10 秒 ～ 20 秒ほど待って、もう一度電源を投入する

上記のリストで問題が解決できない場合は、ご購入先にお問い合わせください。

SPARC: 64 ビット Solaris のブートで発生する問題の解決

64 ビット Solaris リリースを UltraSPARC システムにインストールすると、次のどの条件も該当しない場合は、64 ビットカーネルが自動的にブートされます。

- 64 ビットカーネルをブートするのに、UltraSPARC システムに FLASH PROM アップグレードが必要な場合があります。UltraSPARC システムにファームウェアのアップグレードが必要かどうかを知るには、ハードウェアメーカーの資料を参照してください。
- Open Boot PROM の boot-file パラメータに kernel/unix が設定されています。64 ビットカーネルのブートができない場合に、このパラメータがそのように設定されているなら設定を解除してシステムをリブートします。

- 64 ビット Solaris のすべての構成要素がシステムに完全にインストールされ、正しいファームウェアがインストールされていても、UltraSPARC システムによっては、デフォルトで 64 ビット Solaris カーネルがブートされない場合があります。64 ビット Solaris カーネルがブートされなければ、64 ビットアプリケーションを実行することはできません。

  この問題の詳細や 64 ビット Solaris カーネルのブートをデフォルトで有効にする方法については、`boot(1M)` のマニュアルページを参照してください。

  システムがどの Solaris カーネルを実行しているかを知るには、`isainfo -kv` コマンドが常に使用できます。

```
$ isainfo -kv
64-bit sparcv9 kernel modules
```

  この例では、64 ビット Solaris カーネルが実行されています。

32 ビット Solaris システムで 64 ビット Solaris オペレーティング環境をブートすることはできません。

システムがハングした場合の対処方法

ソフトウェアプロセスに問題がある場合、システムは完全にクラッシュせずに凍結、つまりハングすることがあります。ハングしたシステムから回復するには、次の手順に従ってください。

1. システムがウィンドウ環境を実行していたかどうかを調べて、次のリストの推奨事項に従ってください。これらのリストで問題が解決できなかった場合は、手順 2 に進みます。

   - コマンドを入力しているウィンドウの中に、ポインタがあることを確認します。
   - 間違って `Control-s` キー (画面を凍結する) を押した場合は、`Control-q` キーを押します。`Control-s` キーはウィンドウだけを凍結し、画面全体は凍結しません。ウィンドウが凍結している場合は、他のウィンドウを試します。
   - 可能であれば、ネットワーク上の他のシステムからリモートでログインします。`pgrep` コマンドを使用して、ハングしているプロセスを見つけま

す。ウィンドウシステムがハングしている場合は、そのプロセスを特定して強制終了します。

2. Control-\ キーを押して、動作しているプログラムを強制終了します。core ファイルが書き出されることがあります。
3. Control-c キーを押して、動作している可能性があるプログラムに割り込みをかけます。
4. リモートからログインして、システムをハングさせているプロセスを特定して強制終了します。
5. リモートからログインしてスーパーユーザーになり、システムをリブートします。
6. システムがまだ応答しない場合は、強制的にクラッシュダンプしてリブートします。強制的にクラッシュダンプしてリブートする方法については、第 39 章を参照してください。
7. システムがまだ応答しない場合は、電源を切ってから数分待ち、もう一度電源を入れます。
8. システムがまったく応答しない場合は、ご購入先にお問い合わせください。

ファイルシステムがフルになった場合の対処方法

ルート (/) ファイルシステムや他のファイルシステムがフルになると、次のようなメッセージがコンソールウィンドウに表示されます。

```
.... file system full
```

ファイルシステムがフルになる原因はいくつかあります。次の節では、フルになったファイルシステムを回復する方法をいくつか説明します。ファイルシステムがフルにならないように、古い使用されていないファイルを日常的に整理する方法については、第 28 章を参照してください。

大規模ファイルまたはディレクトリを作成したために、ファイルシステムがフルになる

エラーの原因	解決方法
ファイルかディレクトリを間違った場所にコピーした。これは、アプリケーションがクラッシュして、大きな core ファイルをファイルシステムに書き込んだときにも発生する	スーパーユーザーとしてログインし、特定のファイルシステムで ls -tl コマンドを使用し、新しく作成された大きなファイルを特定して削除する。core ファイルを削除する方法については、520ページの「core ファイルを見つけて削除する方法」を参照

システムのメモリーが不足したために、tmpfs ファイルシステムがフルになる

エラーの原因	解決方法
これは、TMPFS に許可されているよりも多く書き込もうとした、または現在のプロセスがメモリーを多く使用している場合に発生する	tmpfs 関連のエラーメッセージから回復する方法については、tmpfs(7FS) のマニュアルページを参照

コピーまたは復元後にファイルの ACL が消失した場合の対処方法

エラーの原因	解決方法
ACL を持つファイルまたはディレクトリを /tmp ディレクトリにコピーすると、ACL 属性が消失する。/tmp ディレクトリは、通常、一時ファイルシステムとしてマウントされ、ACL などの UFS ファイルシステム属性はサポートしない	代わりに、/var/tmp ディレクトリにファイルをコピーまたは復元する

バックアップ時の問題の解決

この節では、データをバックアップまたは復元するときのいくつかの基本的な問題の解決方法について説明します。

ファイルシステムのバックアップ中に、ルート (/) ファイルシステムがフルになる

ファイルシステムをバックアップしている際に、ルート (/) ファイルシステムがフルになります。このとき、媒体には何も書き込まれていなく、ufsdump コマンドは、媒体の 2 番目のボリュームを挿入するようにプロンプトを表示します。

エラーの原因	問題の解決方法
-f オプションに無効な宛先デバイス名を使用した場合、ufsdump コマンドはファイルをルート (/) ファイルシステムの /dev ディレクトリに書き込み、このファイルシステムをフルにする。たとえば、/dev/rmt/0 ではなく /dev/rmt/st0 と入力した場合、バックアップファイルはテープドライブには送信されず、/dev/rmt/st0 がディスクに作成される	/dev ディレクトリで ls -tl コマンドを使用して、新しく作成された異常に大きなファイルを特定して削除する

バックアップコマンドと復元コマンドが対応していることを確認する

ufsrestore を使用できるのは、ufsdump でバックアップしたファイルを復元するときだけです。tar でバックアップした場合は、tar で復元します。他のコマンドで書き込まれたテープを ufsrestore コマンドを使用して復元しようとした場合、テープが ufsdump フォーマットでないことを知らせるエラーメッセージが表示されます。

現在のディレクトリが間違っていないことを確認する

ファイルを復元する場合に、間違った場所に復元してしまうことがよくあります。ufsdump コマンドは、常にファイルシステムのルートからのフルパス名でファイルをコピーします。したがって ufsrestore を実行する前に、ファイルシス

テムのルートディレクトリに移動しなければなりません。それよりも下のディレクトリでファイルを復元すると、そのディレクトリの下に完全なファイルツリーが作成されます。

古い restore コマンドを使用して、複数ボリュームのフロッピーディスクのバックアップを復元する

dump コマンドで作成した複数ボリュームのフロッピーディスクのバックアップセットからファイルを復元するには、ufsrestore コマンドは使用できません。このようなファイルは、SunOS 4.1 システムで復元しなければなりません。

対話型コマンド

対話型コマンドを使用すると、次の例のような ufsrestore> プロンプトが表示されます。

```
# ufsrestore ivf /dev/rmt/0
Verify volume and initialize maps
Media block size is 126
Dump   date: Mon Jul 12 14:06:54 1999
Dumped from: the epoch
Level 0 dump of a partial file system on venus:/var/adm/acct
Label: none
Extract directories from tape
Initialize symbol table.
ufsrestore >
```

ufsrestore> プロンプトでは、『*Solaris* のシステム管理 *(*第 *1* 巻*)*』の「ufsdump コマンドと ufsrestore コマンド (参照情報)」にリストされているコマンドを使用して、ファイルの検索や、復元するファイルのリストを作成でき、ファイルを復元することもできます。

第 41 章

ファイルアクセスでの問題の解決

この章で説明する手順は次のとおりです。

- 707ページの「検索パスに関連する問題を解決する (コマンドが見つかりません)」
- 710ページの「ファイルアクセスの問題を解決する」
- 711ページの「ネットワークアクセスで発生する問題の把握」

以前は使用できていたプログラム、ファイル、またはディレクトリにアクセスできなくなる (システム管理者に問い合わせる) 場合があります。このようなときは、次の 3 点を調べてください。

- ユーザーの検索パスが変更されているか、または検索パス中のディレクトリが適切な順序であるか
- ファイルまたはディレクトリに適切なアクセス権や所有権があるか
- ネットワーク経由でアクセスするシステムの構成が変更されているか

この章では、これらの 3 点を確認する方法を簡単に説明して、可能な解決策を提案します。

検索パスに関連する問題を解決する (コマンドが見つかりません)

「コマンドが見つかりません」というメッセージは、次のいずれかを示しています。

- コマンドがそのシステムに存在しない

■ コマンドのディレクトリが検索パスに存在しない

検索パスの問題を解決するには、コマンドが格納されているディレクトリのパス名を知る必要があります。

間違ったバージョンのコマンドが見つかってしまうのは、同じ名前のコマンドを持つディレクトリが検索パスにある場合です。この場合、正しいディレクトリが検索パスの後ろの方にあるか、まったく存在しない可能性があります。

現在の検索パスを表示するには、echo $PATH コマンドを使用します。

```
$ echo $PATH
/home/kryten/bin:/sbin:/usr/sbin:/usr/bin:/usr/dt:/usr/dist/exe
```

間違ったバージョンのコマンドを実行しているかどうかを調べるには、which コマンドを使用します。

```
$ which maker
/usr/doctools/frame5.1/bin/maker
```

注 - which コマンドは、.cshrc ファイルの中のパス情報を調べます。.cshrc ファイルに which コマンドの認識する別名を定義している場合に、Bourne シェルか Korn シェルから which コマンドを実行すると、間違った結果が返される場合があります。正しい結果を得るために、which コマンドは C シェルで使用してください。Korn シェルの場合は、whence コマンドを使用します。

▼ 検索パスの問題を診断し、解決する方法

1. **現在の検索パスを表示して、コマンドが入っているディレクトリがユーザーのパス内に存在しない (あるいはスペルが間違っている) ことを確認します。**

   ```
   $ echo $PATH
   ```

2. **次の項目を確認します。**
   - 検索パスは正しいか
   - 検索パスは、コマンドの他のバージョンが存在する他の検索パスの前に指定されているか

- 検索パスのいずれかにコマンドが存在するか

パスを修正する必要がある場合は、手順 3 に進みます。修正する必要がない場合は、手順 4 に進みます。

3. 次の表に示すように、適切なファイルでパスを追加します。

シェル	ファイル	構文	注
Bourne と Korn	`$HOME/.profile`	`$ PATH=`*`$HOME/bin:/sbin:/usr/local/bin ...`* `$ export PATH`	パス名はコロンで区切る

C	`$HOME/.cshrc`
	または
	`$HOME/.login`
		hostname`% set path=(~bin /sbin /usr/local/bin ...)`	パス名は空白文字で区切る

4. 次のように、新しいパスを有効にします。

シェル	パスが指定されているファイル	パスを有効にするコマンド
Bourne と Korn	`.profile`	`$ . ./.profile`
C	`.cshrc`	*hostname*`% source .cshrc`
	`.login`	*hostname*`% source .login`

5. 次のコマンドを使用して、パスを確認します。

```
$ which command
```

例 — 検索パスの問題を診断および修正する

この例は、which コマンドを使用して、mytool の実行可能ファイルが検索パス中のどのディレクトリにも存在しないことを示しています。

```
venus% mytool
mytool: Command not found
venus% which mytool
no mytool in /sbin /usr/sbin /usr/bin /etc /home/ignatz/bin  .
venus% echo $PATH
/sbin /usr/sbin /usr/bin /etc /home/ignatz/bin
venus% vi ~.cshrc
(Add appropriate command directory to the search path)
venus% source .cshrc
venus% mytool
```

コマンドを見つけることができなかった場合は、マニュアルページでそのディレクトリパスを調べます。たとえば、lpsched コマンド (lp プリンタデーモン) を見つけることができなかった場合、lpsched(1M) のマニュアルページを調べると、そのパスが /usr/lib/lp/lpsched であることが解かります。

ファイルアクセスの問題を解決する

以前はアクセスできていたファイルまたはディレクトリにアクセスできない場合は、そのファイルまたはディレクトリのアクセス権または所有権が変更されていることがあります。

ファイルとグループの所有権の変更

誰かがスーパーユーザーとしてファイルを編集したために、ファイルやディレクトリの所有権が変更されていることがあります。新しいユーザーのためにホームディレクトリを作成するときは、そのホームディレクトリのドット (.) ファイルの所有者をそのユーザーにしてください。ユーザーが「.」を所有していない場合、そのユーザーは自分のホームディレクトリにファイルを作成できません。

アクセスに関する問題は、グループの所有権が変更されたとき、またはユーザーがメンバーであるグループが /etc/group データベースから削除されたときにも発生します。

表 41–1 に、アクセスに問題があるファイルのアクセス権や所有権の変更方法を示します。

表 41–1 ファイルアクセスの問題を解決する

変更内容	使用するコマンド	参照箇所
ファイルのアクセス権	`chmod(1)` コマンド	331ページの「アクセス権を絶対モードで変更する方法」
ファイルの所有権	`chown(1)` コマンド	326ページの「ファイルの所有者を変更する方法」
ファイルのグループ所有権	`chgrp(1)` コマンド	327ページの「ファイルのグループ所有権を変更する方法」

ネットワークアクセスで発生する問題の把握

リモートコピーコマンド `rcp` を使用してネットワーク上でファイルをコピーするときに問題が発生した場合、リモートシステム上のディレクトリやファイルは、アクセス権の設定によりアクセスが制限されている可能性があります。他に考えられる問題の原因は、リモートシステムとローカルシステムがアクセスを許可するように構成されていないことです。

ネットワークアクセスで発生する問題と AutoFS 経由でシステムにアクセスする場合に発生する問題については、『*Solaris* のシステム管理 *(*第 *3* 巻*)*』を参照してください。

第 42 章

印刷時の問題の解決

この章では、印刷サービスの設定または管理の際に発生する可能性のある印刷上の問題を解決する方法について説明します。

この章では次の手順を説明します。

- 721ページの「プリンタに出力されない問題を解決する方法」
- 736ページの「出力が正しくない場合の問題を解決する方法」
- 743ページの「LP 印刷サービスのハングを解除する方法」
- 744ページの「アイドル状態になった (ハングした) プリンタの問題を解決する方法」
- 747ページの「矛盾したプリンタ状態メッセージを解決する方法」

印刷と LP 印刷サービスの概要については、第 2 章を参照してください。

印刷時の問題解決のヒント

プリンタを設定後に、何も印刷されないことがあります。また、若干は処理されるものの、何か印刷しても正しく出力されない、読みづらいなど、期待どおりの結果が得られないことがあります。このような問題が発生すると、他にも次のような問題が発生することがあります。

- LP コマンドがハングする
- プリンタがアイドル状態になる
- ユーザーが矛盾したメッセージを受け取る

注 - この章の推奨事項の多くはパラレルプリンタに関連しますが、より一般的なシリアルプリンタにも当てはまります。

出力されない (印刷されない) 場合の解決方法

何も印刷されないときは、次の部分をチェックします。

- プリンタハードウェア
- ネットワーク
- LP 印刷サービス

バナーページは印刷されるのに他には何も印刷されない場合は、不正な出力の特殊ケースです。716ページの「出力が正しくない場合の解決方法」を参照してください。

ハードウェアのチェック

ハードウェアは、最初にチェックすべきポイントです。プリンタが電源に接続され、電源がオンになっているかどうかを確認してください。また、ハードウェア付属のマニュアルを参照して、ハードウェアの設定値を調べてください。コンピュータによっては、プリンタポートの特性を変更するハードウェアスイッチが付いているものがあります。

プリンタハードウェアには、プリンタ、コンピュータへの接続ケーブル、ケーブルの先端を接続するポートが含まれます。一般的なアプローチとしては、プリンタからコンピュータへと順番に調べてください。まず、プリンタをチェックします。次に、ケーブルがプリンタに接続される箇所をチェックします。次に、ケーブルをチェックします。最後に、ケーブルがコンピュータに接続されている箇所をチェックします。

ネットワークのチェック

よく問題が発生するのは、印刷クライアントからプリンタサーバーに送られるリモート印刷要求です。プリンタサーバーと印刷クライアント間でネットワークアクセスが使用可能になっているかどうかを確認してください。

ネットワークがネットワーク情報サービスプラス (NIS+) を実行している場合は、システム間のアクセスを使用可能にする方法について、『*Solaris* ネーミングの管理』を参照してください。ネットワークがネットワーク情報サービス (NIS) または NIS+ を実行していない場合は、プリンタサーバーと印刷クライアントを設定する前に、プリンタサーバー上の `/etc/hosts` ファイルに各クライアントシステムのインターネットアドレスとシステム名を組み込んでください。また、プリンタサーバーのインターネットアドレスとシステム名を、各印刷クライアントシステムの `/etc/hosts` ファイルに組み込まなければなりません。

LP 印刷サービスのチェック

正常に印刷するには、プリンタサーバーと印刷クライアント上で LP スケジューラが動作していなければなりません。動作していない場合は、`/usr/lib/lp/lpsched` コマンドを使用して起動する必要があります。スケジューラの起動に問題がある場合は、102ページの「印刷スケジューラを再起動する方法」を参照してください。

スケジューラが動作している他に、出力する前にプリンタが使用可能になっていて、印刷要求を受け付けられる状態になっていなければなりません。LP 印刷サービスがプリンタへの要求を受け付けなければ、依頼した印刷要求は拒否されます。その場合、一般にユーザーは印刷要求を依頼すると警告メッセージを受け取ります。LP 印刷サービスがプリンタで使用可能になっていないと、印刷要求はプリンタが使用可能になるまでシステム上の待ち行列に残ります。

通常は、次の手順で印刷時の問題を分析してください。

- 手順ごとに印刷要求の経路を追跡します。
- 手順ごとに LP 印刷サービスの状態を調べます。
    - 構成は正しいか
    - プリンタは要求を受け付けるか
    - プリンタは要求を処理できるか
- 要求が転送時にハングしている場合は、`syslog.conf` 内の `lpr.debug` を設定して、転送状況を表示します。
- 要求がローカルでハングしている場合は、`lpsched` ログ (`/var/lp/logs/lpsched`) を調べます。
- 要求がローカルでハングしている場合は、プリンタデバイスエラー (障害) の通知を送らせ、プリンタを再度使用可能にします。

720ページの「印刷時の問題の解決」に掲載されている手順では、この方法を使用して LP 印刷サービスに関する各種の問題に対処する方法を説明します。

LP 印刷サービスの基本的な問題解決の手順で問題を解決できない場合は、当てはまる特定のクライアント/サーバーのケースごとに問題解決の手順を実行する必要があります。

- SunOS 5.8 または互換バージョンのプリンタサーバーを使用する、SunOS 5.8 または互換バージョンの印刷クライアント (操作については、728ページの「SunOS 5.8 または互換バージョンのクライアントから SunOS 5.8 または互換バージョンのプリンタサーバーへの印刷をチェックするには」を参照)
- SunOS 4.1 プリンタサーバーを使用する SunOS 5.8 または互換バージョンの印刷クライアント (操作については、729ページの「SunOS 5.8 または互換バージョンの印刷クライアントから SunOS 4.1 プリンタサーバーへの印刷をチェックするには」を参照)
- SunOS 5.8 または互換バージョンのプリンタサーバーを使用する SunOS 4.1 印刷クライアント (操作については、733ページの「SunOS 4.1 クライアントから SunOS 5.8 または互換バージョンのプリンタサーバーへの印刷をチェックするには」を参照)

出力が正しくない場合の解決方法

プリンタと印刷サービスソフトウェアが正しく構成されていない場合は、プリンタで印刷されても、期待どおりに出力されないことがあります。

プリンタタイプとファイル内容形式のチェック

LP 印刷サービスでプリンタを設定するときに間違ったプリンタタイプを使用すると、不適切なプリンタ制御文字がプリンタに送られる可能性があります。その結果は予測できません。何も印刷されない、出力が読みづらい、正しい文字セットやフォントで印刷されないなどの結果となります。

間違ったファイル内容形式を指定した場合、バナーページは印刷できますが、他には何も印刷されません。プリンタに指定されたファイル内容形式は、プリンタがフィルタなしで直接印刷できるファイル形式を示します。ユーザーがプリンタにファイルを送信すると、ファイルはフィルタなしでプリンタに直接送信されます。プリンタがその形式を処理できないときは、問題が発生します。

印刷クライアントの設定時には、ファイル内容形式がプリンタサーバーと印刷クライアントの両方で正しくなければならないので、間違いをおかす機会が多くなります。推奨する方法は、印刷クライアントのファイル内容形式を any に設定することです。こうすると、ファイルはサーバーに直接送信され、フィルタが必要かどうかはサーバー側で決定されます。したがってファイル内容形式は、サーバー側だけで正しく指定すればよいことになります。

印刷クライアント側でファイル内容を指定し、フィルタリングの負荷をサーバーからクライアントに移すことができますが、内容の形式はプリンタサーバー側でサポートしなければなりません。

stty 設定値のチェック

デフォルトの stty (標準端末) 設定値がプリンタから要求される設定値と一致しないと、多数のフォーマット上の問題が生じる可能性があります。この後の節では、設定値の一部が間違っているときに発生する問題について説明します。

ボーレート設定値が正しくない場合

コンピュータのボーレート設定値がプリンタのボーレート設定値と一致しないときは、通常何か出力されますが、希望する出力は得られません。特殊文字や不要なスペースが異常に混じったランダムな出力が表示されます。LP 印刷サービスのデフォルトは 9600 ボーレートです。

注 - プリンタがパラレルポートで接続されている場合、ボーレート設定値は関係ありません。

パリティ設定値が正しくない場合

プリンタによっては、パリティビットを使用して、印刷用に受け取ったデータに伝送中に誤りがなかったことを確認するものがあります。コンピュータとプリンタのパリティビットの設定値は一致しなければなりません。一致しない場合、文字によってはまったく印刷されないか、他の文字で置き換えられることもあります。その出力は文字間隔が正しく、ほとんどの文字が正しい位置にあるので、一見正しいように見えます。LP 印刷サービスの場合、デフォルトではパリティビットは設定されません。

タブ設定値が正しくない場合

ファイルにタブが含まれていても、プリンタがタブを予期していなければ、印刷出力にはファイルの内容が完全に印刷されますが、テキストは右マージンに対して正確に配置されないことがあります。また、プリンタのタブ設定が間違っていると、テキストに左マージンがない、テキストがつながってしまう、テキストがページの一部分に集中する、間違ってダブルスペースになってしまうなどの問題が発生します。デフォルトでは、タブは 8 スペースごとに設定されます。

Return 設定値が正しくない場合

出力がシングルスペースのはずなのにダブルスペースになる場合は、プリンタのタブ設定値が間違っているか、プリンタが Return の後に 1 行追加されています。LP 印刷サービスは、改行の前に 1 つ Return を追加するので、その組み合わせによって 2 行の改行が発生します。

ジグザグに印刷される場合は、改行の前に Return を送る `stty` オプションの `onlcr` が設定されていません。`stty=onlcr` オプションはデフォルトで設定されますが、他の印刷問題を解決しようとしたときに、それを消去した可能性があります。

ハングした **LP** コマンドの解決方法

`lp` コマンド (`lpsystem`、`lpadmin`、`lpstat` など) を入力しても何も発生しない (エラーメッセージ、状態情報、またはプロンプトが表示されない) 場合は、LP スケジューラに問題が発生した可能性があります。このような問題は、通常は LP スケジューラを停止して再起動すれば解決できます。操作手順については、102ページの「印刷スケジューラを停止する方法」を参照してください。

アイドル状態になった (ハングした) プリンタの解決方法

プリンタが印刷要求を待ち行列に入れているのに、アイドル状態になっていることがあります。プリンタがアイドル状態になっている場合は、次の原因が考えられます。

- 現在の印刷要求にフィルタがかけられている
- プリンタに障害がある
- ネットワーク上の問題が原因で、印刷処理が中断されている

印刷フィルタのチェック

低速印刷フィルタは、プリンタを拘束しないようにバックグラウンドで実行されます。フィルタリングが必要な印刷要求は、フィルタリングが終わるまで印刷されません。

プリンタ障害のチェック

LP 印刷サービスが障害を検出すると、印刷はすぐにではありませんが自動的に再開されます。LP 印刷サービスは約 5 分間待機し、要求が正常に印刷されるまで試行し続けます。プリンタを使用可能にすると、すぐに再試行できます。

ネットワーク上の問題のチェック

ネットワーク経由でファイルを印刷するときには、次の問題が発生することがあります。

- プリンタサーバーに送られた要求が、クライアントシステム (ローカル) の待ち行列で停止する
- プリンタサーバーに送られた要求が、プリンタサーバー (リモート) の待ち行列で停止する

ローカル待ち行列で停止する印刷要求

プリンタサーバーに依頼された印刷要求は、次の原因でクライアントシステムの待ち行列で停止することがあります。

- プリンタサーバーがダウンしている
- プリンタがプリンタサーバー側で使用不可にされている
- 印刷クライアントとプリンタサーバー間のネットワークがダウンしている
- ベースになる互換バージョンのネットワークソフトウェアが適切に設定されていない

問題の原因を突き止めるときには、新しい要求を待ち行列に追加しないでください。詳細は、120ページの「プリンタへの印刷要求を受け付けるまたは拒否する方法」 を参照してください。

リモート待ち行列で停止する印刷要求

印刷要求がプリンタサーバーの待ち行列で停止する場合は、プリンタが使用不可になっている可能性があります。プリンタが要求を受け付けても処理しないとき、その要求は印刷するために待ち行列に入れられます。プリンタを使用可能にすると、それ以外に問題がなければ、待ち行列内の印刷要求は印刷されます。

矛盾した状態メッセージの解決方法

ユーザーが印刷要求を入力すると、クライアントシステムからは受け付けられたことが通知され、プリンタサーバーからは印刷要求が拒否されたことを示すメールを受け取ることがあります。これらの矛盾したメッセージは、次の原因で発生することがあります。

- 印刷クライアントは要求を受け付けることができても、プリンタサーバーは要求を拒否している場合
- 印刷クライアント側のプリンタの定義が、プリンタサーバー側のプリンタの定義と一致しない場合。特に、フィルタ、文字セット、印字ホイール、フォームなど、印刷ジョブコンポーネントの定義が、クライアントとサーバーシステムの間で一致していない場合

ローカルユーザーがプリンタサーバー上でプリンタにアクセスできるように、これらのジョブコンポーネントの定義が印刷クライアントとプリンタサーバーの両方で登録されているかどうかを確認してください。

印刷時の問題の解決

この節では、次の手順について説明します。

- 出力されない問題を解決する方法
- 出力が正しくない問題を解決する方法
- LP コマンドのハングを解除する方法
- アイドル状態になった (ハングした) プリンタの問題を解決する方法
- 矛盾した状態メッセージを解決する方法

▼ プリンタに出力されない問題を解決する方法

この作業には、次の問題解決の手順が含まれています。印刷要求をプリンタに出したのに何も印刷されない場合は、これらの手順を試してください。

- ハードウェアをチェックする (721ページの「ハードウェアをチェックするには」を参照)
- ネットワークをチェックする (723ページの「ネットワークをチェックするには」を参照)
- LP 印刷サービスの基本機能をチェックする (723ページの「LP 印刷サービスの基本機能をチェックするには」を参照)
- SunOS 5.8 または互換バージョンの印刷クライアントから SunOS 5.8 または互換バージョンのプリンタサーバーへの印刷をチェックする (728ページの「SunOS 5.8 または互換バージョンのクライアントから SunOS 5.8 または互換バージョンのプリンタサーバーへの印刷をチェックするには」を参照)
- SunOS 5.8 または互換バージョンの印刷クライアントから SunOS 4.1 プリンタサーバーへの印刷をチェックする (729ページの「SunOS 5.8 または互換バージョンの印刷クライアントから SunOS 4.1 プリンタサーバーへの印刷をチェックするには」を参照)
- SunOS 4.1 印刷クライアントからSunOS 5.8 または互換バージョンのプリンタサーバーへの印刷をチェックする (733ページの「SunOS 4.1 クライアントから SunOS 5.8 または互換バージョンのプリンタサーバーへの印刷をチェックするには」を参照)

該当する印刷クライアント/サーバーのケースに進む前に、上記のうち最初の 3 つの手順をリストの順に試してください。ただし、バナーページは印刷されるが他に何も印刷されない場合は、736ページの「出力が正しくない場合の問題を解決する方法」の説明に進んでください。

ハードウェアをチェックするには

1. プリンタがコンセントに接続され、電源がオンになっているか確認します。
2. ケーブルがプリンタのポートと、システムまたはサーバーのポートに接続されているか確認します。
3. そのケーブルが正しいケーブルであり、欠陥がないことを確認します。

詳細は、ハードウェア付属のマニュアルを参照してください。プリンタがシリアルポートに接続されている場合は、そのケーブルでハードウェアフロー制御がサポートされることを確認してください。NULL モデムアダプタでは、この機能がサポートされます。表 42–1 は、NULL モデムケーブル用のピン構成を示しています。

表 **42–1** NULL モデムケーブル用のピン構成

	ホスト	プリンタ
Mini-Din-8	**25-Pin D-sub**	**25-Pin D-sub**
-	1(FG)	1(FG)
3(TD)	2(TD)	3(RD)
5(RD)	3(RD)	2(TD)
6(RTS)	4(RTS)	5(CTS)
2(CTS)	5(CTS)	4(RTS)
4(SG)	7(SG)	7(SG)
7(DCD)	6(DSR)、8(DCD)	20(DTR)
1(DTR)	20(DTR)	6(DSR)、8(DCD)

4. ポート用のハードウェアスイッチが正しく設定されていることを確認します。
   正しい設定については、プリンタのマニュアルを参照してください。

5. プリンタが動作するか確認します。
   プリンタにセルフテスト機能が付いている場合は、その機能を使用します。プリンタのセルフテストの詳細は、プリンタのマニュアルを参照してください。

6. コンピュータとプリンタのボーレートの設定値が正しいか確認します。
   コンピュータとプリンタのボーレートの設定値が一致しなければ、何も印刷されないことがあり、さらに正しく出力されない場合もあります。詳細は、736ページの「出力が正しくない場合の問題を解決する方法」を参照してください。

ネットワークをチェックするには

1. ping コマンドを使用すると、プリンタサーバーと印刷クライアント間のネットワークが正しく設定されているか確認できます。

```
print_client# ping print_server
print_server is alive
print_server# ping print_client
print_client not available
```

システムが動作していることを示すメッセージが表示されれば、そのシステムにアクセスできることがわかるので、そのネットワークは正常です。また、このメッセージは、入力したホスト (システム) 名が、ネームサーバーまたはローカルの /etc/hosts ファイルによって IP アドレスに変換されたことを示します。変換されていない場合は、IP アドレスを入力する必要があります。

「not available」というメッセージが表示された場合は、次の 3 点を確認してください。まず、NIS または NIS+ はサイトでどのように設定されているか。次に、プリンタサーバーと印刷クライアントが相互に通信できるように付加的な作業が必要か。最後に、サイトが NIS または NIS+ を実行していない場合、各印刷クライアントの /etc/hosts ファイルにプリンタサーバーの IP アドレスを入力し、プリンタサーバーの /etc/hosts ファイルにすべての印刷クライアントの IP アドレスを入力したか確認します。

2. **(SunOS 5.0 – 5.1** プリンタサーバーのみ**)** listen ポートモニターが正しく構成されているか確認します。

3. **(SunOS 5.0 – 5.1** プリンタサーバーのみ**)** ネットワーク待機サービスがプリンタサーバー上のポートモニターに登録されているか確認します。

LP 印刷サービスの基本機能をチェックするには

この手順では、基本 LP 印刷サービス機能をチェックする例として、プリンタ luna を使用しています。

1. プリンタサーバー上と印刷クライアント上で、**LP** 印刷サービスが動作していることを確認します。

   a. このコマンドは、**LP** スケジューラが動作しているか表示します。

```
# lpstat -r
scheduler is running
```

b. スケジューラが動作していない場合は、スーパーユーザーまたは lp になり、スケジューラを起動します。

```
# /usr/lib/lp/lpsched
```

スケジューラを起動できない場合は、743ページの「LP 印刷サービスのハングを解除する方法」を参照してください。

2. プリンタサーバー上と印刷クライアント上で、プリンタが要求を受け付けていることを確認します。

a. プリンタが要求を受け付けていることを確認します。

```
# lpstat -a
mars accepting requests since Jul 12 14:23 1999
luna not accepting requests since Jul 12 14:23 1999
unknown reason
```

このコマンドは、LP システムがシステム用に構成された各プリンタの要求を受け付けているか確認します。

b. プリンタが要求を受け付けていない場合は、スーパーユーザーまたは lp になり、プリンタが印刷要求を受け付けるようにします。

```
# accept luna
```

これで、指定したプリンタは要求を受け付けます。

3. プリンタサーバー上と印刷クライアント上で、プリンタが依頼された印刷要求の印刷で使用可能になっているか確認します。

a. プリンタが使用可能になっていることを確認します。

```
# lpstat -a
mars accepting requests since Jul 12 14:23 1999
luna not accepting requests since Jul 12 14:23 1999
unknown reason
```

このコマンドは、プリンタの状態に関する情報を表示します。プリンタ名を省略すると、システム用に設定されたすべてのプリンタに関する情報を表示できます。次の例は、使用不可になっているプリンタを示しています。

b. プリンタが使用不可になっている場合は、スーパーユーザーまたは lp になり、プリンタを使用可能にします。

```
# enable luna
printer "luna" now enabled.
```

指定したプリンタが、印刷要求の処理に使用可能になります。

4. プリンタサーバー上で、プリンタが正しいシリアルポートに接続されていることを確認します。

a. プリンタが正しいシリアルポートに接続されていることを確認します。

```
# enable luna
printer "luna" now enabled.
```

「device for *printer-name*」というメッセージは、ポートアドレスを示します。LP 印刷サービスの接続先のポートにケーブルが接続されているか確認します。ポートが正しければ、726ページの手順 5 に進みます。

b. スーパーユーザーまたは lp になります。

c. ポートを表すデバイスファイルのファイル所有権を変更します。

```
# chown lp device-filename
```

このコマンドは、特殊なユーザー lp をデバイスファイルの所有者として割り当てます。このコマンドで、*device-filename* はデバイスファイル名です。

d. プリンタポートのデバイスファイルのアクセス権を変更します。

```
# chmod 600 device-filename
```

このコマンドにより、root または lp だけがプリンタポートデバイスファイルにアクセスできます。

5. プリンタサーバー上と印刷クライアント上で、プリンタが正しく構成されていることを確認します。

a. プリンタが適切に設定されていることを確認します。

```
# lpstat -p luna -l
printer luna is idle. enabled since Jul 12 14:24 1999. available
        Content types: postscript
        Printer types: PS
```

上の例は、正しく設定された PostScript プリンタと、そのプリンタを印刷要求の処理に利用できることを示しています。プリンタタイプとファイル内容形式が正しい場合は、726ページの手順 6 に進みます。

b. プリンタタイプまたはファイル内容形式が違っている場合は、印刷クライアント上で、プリンタタイプを unknown に設定し、内容形式を any に設定してください。

```
# lpadmin -p printer-name -T printer-type -I file-content-type
```

6. プリンタサーバー上で、プリンタがプリンタ障害のために待機していないことを確認します。

a. プリンタ障害のためにプリンタが待機していないことを確認します。

```
# lpadmin -p printer-name -F continue
```

このコマンドは LP 印刷サービスに対して、障害のために待機していない場合は続行するように指示します。

b. プリンタを再び使用可能にすることによって、すぐに再試行させます。

```
# enable printer-name
```

c. **(省略可能)** プリンタ障害をすぐに通知するように、**LP** 印刷サービスに指示します。

```
# lpadmin -p printer-name -A 'write root'
```

このコマンドは LP 印刷サービスに対して、プリンタが障害を起こした場合に、root に書き込むというデフォルトポリシーを設定し、root がログインした端末にプリンタ障害メッセージを送るように指示します。これにより、問題を修正するときに障害通知をすぐに受け取れます。

7. プリンタがログイン端末として間違った設定になっていないか確認します。

注 - ログイン端末としてプリンタを設定する作業では誤りをおかしやすいので、当てはまらないと思われる場合にも、必ず設定値を確認してください。

---

a. `ps -ef` コマンドの出力で、プリンタポートのエントリを探します。

```
# ps -ef
    root   169   167  0   Apr 04 ?        0:08 /usr/lib/saf/listen tcp
    root   939     1  0 19:30:47 ?        0:02 /usr/lib/lpsched
    root   859   858  0 19:18:54 term/a   0:01 /bin/sh -c \ /etc/lp/
interfaces/luna
luna-294 rocket!smith ``passwd\n##
#
```

このコマンドの出力で、プリンタポートのエントリを探します。上の例で、ポート `/dev/term/a` はログイン端末として間違って設定されています。この行の最後に `"passwd\n##` 情報が付いているのでわかります。ポートが正しく設定されている場合は、この手順の最後を飛ばしてください。

b. 印刷要求を取り消します。

```
# cancel request-id
```

このコマンドで、*request-id* は取り消したい印刷要求の要求 ID 番号です。

c. プリンタポートをログインデバイス以外のものとして設定します。

```
# lpadmin -p printer-name -h
```

d. ps -ef コマンドからの出力をチェックして、プリンタポートがログインデバイスではなくなったことを確認します。

基本的な LP 印刷サービス機能に印刷時の問題の原因が見つからない場合は、次の中から該当するクライアント/サーバーの手順に進んでください。

SunOS 5.8 または互換バージョンのクライアントから SunOS 5.8 または互換バージョンのプリンタサーバーへの印刷をチェックするには

1. まだチェックしていなければ、プリンタサーバー上で **LP** 印刷サービスの基本機能をチェックします。

   基本機能をチェックする手順については、723ページの「LP 印刷サービスの基本機能をチェックするには」を参照してください。印刷クライアントから要求が出されたときに何も印刷されない原因を探す前に、プリンタがローカルで正しく動作することを確認してください。

2. まだチェックしていなければ、印刷クライアント上で **LP** 印刷サービスの基本機能をチェックします。

   基本機能をチェックする手順については、723ページの「LP 印刷サービスの基本機能をチェックするには」を参照してください。クライアントからの要求が印刷される前に、印刷クライアント上で LP スケジューラが動作していなければならず、またプリンタが使用可能であり、要求を受け付けられる状態になっていなければなりません。

注 - 次の手順のほとんどは、root または lp としてログインして実行しなければなりません。

---

3. プリンタサーバーがアクセス可能であることを確認します。

   a. 印刷クライアント上で、**ping** ***print-server*** と入力して Return キーを押します。このコマンドにより、プリンタサーバーに応答を求める要求が送られます。

```
print_client# ping print_server
```

「*print_server* not available」というメッセージを受け取った場合は、ネットワークに問題があります。

4. **SunOS 5.1** 印刷クライアント上でのみ、**Admintool** の「プリンタの変更 **(Modify Printer)**」ウィンドウを表示して、プリンタサーバーのタイプが s5 になっていることを確認します。

5. プリンタサーバーが正常に動作しているか確認します。

```
# lpstat -t luna
scheduler is running
system default destination: luna
device for luna: /dev/term/a
luna accepting requests since Jul 12 14:26 1999
printer luna now printing luna-314. enabled since Jul 12 14:26 1999.
available.
luna-129            root               488   Jul 12 14:32
#
```

上記の例は、プリンタサーバーが動作していることを示します。

6. プリンタサーバーが正常に動作していない場合は、手順 **1** に戻ります。

SunOS 5.8 または互換バージョンの印刷クライアントから SunOS 4.1 プリンタサーバーへの印刷をチェックするには

1. まだチェックしていなければ、印刷クライアント上で **LP** 印刷サービスの基本機能をチェックします。

   手順については、723ページの「LP 印刷サービスの基本機能をチェックするには」を参照してください。

2. プリンタサーバーがアクセス可能であることを確認します。

   a. 印刷クライアント上で、**ping** ***print-server*** と入力して Return キーを押します。このコマンドにより、プリンタサーバーに応答を求める要求が送られます。

```
print_client# ping print_server
```

「*print_server* not available」というメッセージを受け取った場合は、ネットワークに問題があります。

3. プリンタサーバー上で lpd デーモンが動作していることを確認します。

   a. 次のコマンドを実行して、プリンタサーバー上で lpd デーモンが動作していることを確認します。

```
$ ps -ax | grep lpd
  126 ?  IW    0:00 /usr/lib/lpd
  200 p1 S     0:00 grep lpd
$
```

   lpd デーモンが動作している場合は、上記の例のような 1 行が表示されます。動作していなければ、プロセス情報は表示されません。

   b. lpd がプリンタサーバー上で動作していない場合は、プリンタサーバー上でスーパーユーザーになり、lpd を再起動します。

```
# /usr/lib/lpd &
```

4. プリンタサーバーの lpd デーモンが正しく構成されていることを確認します。

   a. プリンタサーバー上でスーパーユーザーになり、lpc コマンドを入力します。

```
# /usr/etc/lpc
lpc>
```

   b. **LP** 状態情報を取得します。

```
lpc> status
luna:
queuing is enabled
printing is enabled
no entries
no daemon present
lpc>
```

状態情報が表示されます。上記の例では、デーモンは動作していないので再起動する必要があります。

c. デーモンが存在しない場合は、デーモンを再起動します。

```
lpc> restart luna
```

デーモンが再起動されます。

d. lpd デーモンが起動されていることを確認します。

```
lpc> status
```

e. lpc コマンドを終了します。

```
lpc> quit
```

シェルプロンプトが再表示されます。

5. 印刷クライアントがプリンタサーバーにアクセスできることを確認します。

a. SunOS 4.1 プリンタサーバー上に /etc/hosts.lpd ファイルがあるか確認します。

SunOS 4.1 プリンタサーバー上では、このファイルが存在する場合、着信印刷要求を受け付けられるかどうかの判定に使用されます。このファイルが存在しない場合、すべての印刷クライアントシステムがアクセスできるため、次の手順の b と c は省略します。

b. ファイルが存在する場合、印刷クライアントがファイルにリストされるか調べます。

ファイルにリストされていないクライアントシステムからの要求は、プリンタサーバーに転送されません。

c. クライアントがリストされていない場合は、印刷クライアントをファイルに追加します。

注 - ここまでで特に問題点が見つからない場合、SunOS 4.1 システムは正常に設定され、機能しているはずです。

---

6. 印刷クライアントからリモート `lpd` 印刷デーモンへの接続が正しく行われていることを確認します。

   a. 印刷クライアント上でスーパーユーザーになり、`lpsched` デーモンが実行されていることを確認します。

```
# ps -ef | grep lp
   root   154     1 80   Jan 07 ?        0:02 /usr/lib/lpsched
```

   上記の例のように、`lpsched` デーモンは動作しているはずです。

   b. **LP** 印刷サービスを停止します。

```
# lpshut
```

   c. **LP** 印刷サービスを再起動します。

```
# /usr/lib/lp/lpsched
```

7. リモートプリンタサーバーが **SunOS 4.1** システムとして正しく識別されていることを確認します。

SunOS 4.1 クライアントから SunOS 5.8 または互換バージョンのプリンタサーバーへの印刷をチェックするには

1. まだチェックしていなければ、プリンタサーバー上で **LP** 印刷サービスの基本機能をチェックします。

   手順については、723ページの「LP 印刷サービスの基本機能をチェックするには」を参照してください。印刷クライアントから要求が出されたときに何も印刷されない原因を調べる前に、プリンタがローカルで動作していることを確認してください。

   注 - 次の手順で指定されているシステムでは、スーパーユーザーまたは `lp` としてログインする必要があります。

   ---

2. 印刷クライアントにアクセスできることを確認します。

   a. **SunOS 5.8** プリンタサーバー上で、**`ping`** ***print-client*** と入力して `Return` キーを押します。

```
print_server# ping print_client
print_client is alive
```

   「*print_client* `not available`」というメッセージが表示された場合は、ネットワークに問題があります。

3. 印刷クライアント上で、プリンタが正しく設定されていることを確認します。

```
# lpr -P luna /etc/fstab
lpr: cannot access luna
#
```

   このコマンドでは、印刷クライアントが動作しているか表示されます。上記の例は、印刷クライアントが正常に動作していないことを示します。

4. 印刷クライアント上で `lpd` デーモンが動作していることを確認します。

   a. `lpd` デーモンが動作していることを確認します。

```
# ps -ax | grep lpd
  118 ?  IW    0:02 /usr/lib/lpd
#
```

このコマンドでは、lpd デーモンが印刷クライアント上で動作しているか表示されます。上記の例は、デーモンが動作していることを示します。

b. 印刷クライアント上で、lpd デーモンを起動します。

```
# /usr/lib/lpd &
```

5. 印刷クライアント上で、プリンタサーバーを識別する printcap エントリが存在することを確認します。

a. プリンタが認識されていることを確認します。

```
# lpr -P mercury /etc/fstab
lpr: mercury: unknown printer
#
```

上記の例は、指定したプリンタのエントリが /etc/printcap ファイルに入っていないことを示します。

b. エントリがない場合は、/etc/printcap ファイルを編集して次の情報を追加します。

```
printer-name|print-server:\
:lp=:rm=print-server:rp=printer-name:br#9600:rw:\
:lf=/var/spool/lpd/printer-name/log:\
:sd=/var/spool/lpd/printer-name:
```

次の例は、プリンタサーバー neptune に接続されたプリンタ luna のエントリを示します。

```
luna|neptune:\
        :lp=:rm=neptune:rp=luna:br#9600:rw:\
        :lf=/var/spool/lpd/luna/log:\
        :sd=/var/spool/lpd/luna:
```

c. プリンタのスプーリングディレクトリ (/var/spool/lpd/*printer-name*) を作成します。

6. 再試行を強制し、印刷クライアント lpd が待機状態になっていないことを確認します。

   プリンタサーバーが動作し応答している場合、印刷クライアント lpd は再試行する前に待ち状態になっている可能性があります。

   a. 印刷クライアント上でスーパーユーザーとなり、lpc コマンドを起動します。

      lpc> プロンプトが表示されます。

   b. プリンタを再起動します。

   c. lpc コマンドを終了します。

      シェルプロンプトが再表示されます。

```
# lpc
lpc> restart luna
luna:
       no daemon to abort
luna:
      daemon started
# quit
$
```

7. プリンタサーバーへの接続を調べます。

a. 印刷クライアント上でスーパーユーザーになり、プリンタのログファイルを調べます。

```
# more /var/spool/lpd/luna/log
```

通常、何も表示されません。

b. プリンタ状態ログも調べます。

```
# more /var/spool/lpd/luna/status
waiting for luna to come up
#
```

c. 接続が正常な場合は、プリンタサーバー上でプリンタサーバーが正しく設定されているかを確認します。

```
# lpstat -t
scheduler is running
system default destination: luna
device for luna: /dev/term/a
luna accepting requests since Jul 12 14:29 1999
luna accepting requests since Jul 12 14:29 1999
printer luna is idle. enabled since Jul 12 14:29 1999. available.
#
```

上記の例は、プリンタサーバーが起動され、動作していることを示します。プリンタサーバーが動作していない場合は、先に進む前に 733ページの手順 1 に戻ってください。

▼ 出力が正しくない場合の問題を解決する方法

1. スーパーユーザーまたは `lp` としてログインします。
2. プリンタタイプが正しいことを確認します。

プリンタタイプが正しくないと、正しく出力されないことがあります。たとえば、プリンタタイプ PS を指定してページを逆順に印刷する場合は、プリンタタイプ PSR を試してください (この 2 つのタイプ名は大文字で指定しなければなりません)。また、プリンタタイプが正しくないと、テキストの欠落、読みづらいテキスト、または間違ったフォントのテキストが出力されることがあります。プリンタタイプを判別するには、terminfo データベース内のエントリを調べます。terminfo データベースの構造については、60ページの「プリンタタイプ」を参照してください。

a. プリンタサーバー上で、プリンタの特性を表示します。

```
$ lpstat -p luna -l
printer luna is idle. enabled since Mon Jul 12 15:02:32 MDT 1999. available.
        Form mounted:
        Content types: postscript
        Printer types: PS
        Description:
        Connection: direct
        Interface: /usr/lib/lp/model/standard
        After fault: continue
        Users allowed:
                (all)
        Forms allowed:
                (none)
        Banner not required
        Character sets:

        Default pitch:
        Default page size: 80 wide 66 long
        Default port settings:
$
```

b. プリンタのマニュアルを参照して、プリンタのモデルを調べます。

c. プリンタタイプが正しくない場合は、**Admintool** の「プリンタの変更 **(Modify Printer)**」オプションを使用して変更するか、次の lpadmin コマンドを使用します。

```
# lpstat -p printer-name -T printer-type
```

印刷クライアント上では、プリンタタイプを unknown にしてください。プリンタサーバー上では、プリンタタイプは使用するプリンタのモデルをサポートするように定義された terminfo エントリと一致しなければなりません。使用するプリンタのタイプに関する terminfo エントリがない場合は、171

ページの「サポートされていないプリンタの terminfo エントリを追加する方法」を参照してください。

3. バナーページは印刷されるが文書の本文が印刷されない場合は、ファイル内容形式を確認します。

   プリンタに指定したファイル内容形式は、プリンタがフィルタなしで直接印刷できるファイル形式を示します。ファイル内容形式が正しくなければ、必要なときにフィルタリングがバイパスされることがあります。

   a. 前の手順の lpstat コマンドで表示されたファイル内容形式に関する情報をメモします。

      印刷クライアント上では、1 つ以上の明示的な内容形式を指定する理由がない限り、ファイル内容形式を any にしてください。クライアント上で内容を指定すると、プリンタサーバー上ではなく印刷クライアント上でフィルタリングが実行されます。また、クライアント上の内容形式は、プリンタサーバー上で指定した内容形式と一致しなければならず、プリンタサーバー上の内容形式はプリンタの機能を反映していなければなりません。

   b. プリンタのマニュアルを参照し、プリンタで直接印刷できるファイルのタイプを判別します。

      これらのファイル形式を参照するために使用する名前は、プリンタメーカーが使用している名前と一致しなくてもかまいません。ただし、使用する名前は LP 印刷サービスに認識されるフィルタで使用する名前と一致しなければなりません。

   c. ファイル内容形式が正しくない場合は、**Admintool** の「プリンタの変更 **(Modify Printer)**」オプションで変更するか、次の lpadmin コマンドを使用します。

      ```
      # lpadmin -p printer-name -I file-content-type(s)
      ```

      必要に応じて、このコマンドを印刷クライアント上、プリンタサーバー上、またはその両方で実行します。印刷クライアント上で -I any を試し、プリンタサーバー上で -I "" を試してください。-I "" は、NULL のファイル内容形式リストを指定します。これは、プリンタはそのプリンタタイプと正確に一致するファイルしか直接印刷できないので、すべてのファイルをフィルタにかけることを意味します。

ファイルが印刷されないときは、まずこの組み合わせを選択してみるとよいでしょう。それで成功したら、プリンタサーバー上で明示的な内容形式を指定し、不要なフィルタリングを減らすことができます。ローカルの PostScript プリンタでは、プリンタでサポートされている場合は、postscript または postscript,simple を使用してください。PS と PSR はファイル内容形式ではなく、プリンタタイプなので注意してください。

-I を省略すると、ファイル内容のリストはデフォルトの simple になります。-I オプションを使用し、simple 以外にもファイル内容形式を指定したい場合は、リストに simple を含めなければなりません。

複数のファイル内容形式を指定するときは、名前をコンマで区切ります。また、名前をスペースで区切り、リストを引用符で囲むこともできます。ファイル内容形式として any を指定すると、フィルタリングは行われないので、プリンタで直接印刷できるファイルタイプのみを送信する必要があります。

4. フォントのダウンロードに必要なフィルタリングを、印刷要求がバイパスしていないかどうかをチェックします。

   ユーザーがコマンド lp -T PS を使用して印刷要求を PostScript プリンタに依頼すると、フィルタリングは実行されません。フィルタリングを強制するコマンド lp -T postscript を使用して要求を依頼しようとすると、文書に必要な非常駐フォントがダウンロードされることがあります。

5. プリンタポートの stty 設定値が正しいことを確認します。

   a. プリンタのマニュアルを参照して、プリンタポートに合った stty 設定値を判別します。

   注 - プリンタがパラレルポートで接続されている場合、ボーレートの設定値は無関係です。

   ---

   b. 現在の設定値を調べるには、stty コマンドを使用します。

```
# stty -a < /dev/term/a
speed 9600 baud;
rows = 0; columns = 0; ypixels = 0; xpixels = 0;
eucw 1:0:0:0, scrw 1:0:0:0
intr = ^c; quit = ^|; erase = ^?; kill = ^u;
eof = ^d; eol = <undef>; eol2 = <undef>; swtch = <undef>;
```

(続く)

```
start = ^q; stop = ^s; susp = ^z; dsusp = ^y;
rprnt = ^r; flush = ^o; werase = ^w; lnext = ^v;
parenb -parodd cs7 -cstopb -hupcl cread -clocal -loblk -parext
-ignbrk brkint -ignpar -parmrk -inpck istrip -inlcr -igncr icrnl -iuclc
ixon -ixany -ixoff imaxbel
isig icanon -xcase echo echoe echok -echonl -noflsh
-tostop echoctl -echoprt echoke -defecho -flusho -pendin iexten
opost -olcuc onlcr -ocrnl -onocr -onlret -ofill -ofdel tab3
#
```

このコマンドでは、プリンタポートの現在の stty 設定値が表示されます。LP 印刷サービスの標準プリンタインタフェースプログラムで使用されるデフォルトの stty オプションを表 42–2 に示します。

表 42–2 標準インタフェースプログラムで使用されるデフォルトの stty 設定値

オプション	意味
–9600	ボーレートを 9600 に設定
–cs8	8 ビットバイトを設定
–cstopb	1 バイト当たり 1 ストップビットを送信
–parity	パリティを生成しない
–ixon	XON/XOFF (START/STOP または DC1/DC3 ともいう) を使用可能にする
–opost	以下にリストされた設定値をすべて使用して「処理後出力」を実行する
–olcuc	小文字を大文字に割り当てない
–onlcr	改行をキャリッジリターン/改行に変更する
–ocrnl	キャリッジリターンを改行に変更しない

表 42–2 標準インタフェースプログラムで使用されるデフォルトの stty 設定値 続く

オプション	意味
–onocr	カラム 0 でもキャリッジリターンを出力する
–nl0	改行後の遅延なし
–cr0	キャリッジターン後の遅延なし
–tab0	タブ後の遅延なし
–bs0	バックスペース後の遅延なし
–vt0	垂直タブ後の遅延なし
–ff0	用紙送り後の遅延なし

c. stty 設定値を変更します。

```
# lpadmin -p printer-name -o "stty= options"
```

表 42–3 を使用して、印刷出力に影響する様々な問題を解決する stty オプションを選択します。

表 **42–3** 印刷出力の問題を解決する stty オプション

stty 値	結果	間違った設定から起こり得る問題
110, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200, 38400	ボーレートを指定した値に設定する (ボーレートを 1 つだけ入力する)	ランダム文字と特殊文字が印刷され、間隔がバラバラになることがある
oddp	奇数パリティを設定する	文字が欠落または間違った文字がランダムに表示される
evenp	偶数パリティを設定する	
-parity	パリティを設定しない	
-tabs	タブを設定しない	テキストが右マージンにくっついてしまう
tabs	8 スペースごとにタブを設定する	テキストに左マージンがなく、つながってしまうか、くっついてしまう
-onlcr	行頭でキャリッジリターンを設定しない	間違ったダブルスペース
onlcr	行頭でキャリッジリターンを設定する	ジグザグに印刷される

オプションをスペースで区切り、オプションリストを単一引用符で囲むと、複数のオプションの設定を変更できます。たとえば、奇数パリティを使用可能にし、7 ビットの文字サイズを設定する必要のあるプリンタを仮定します。そのためには、次の例のようなコマンドを入力します。

```
# lpadmin -p neptune -o "stty='parenb parodd cs7'"
```

stty オプション parenb でパリティチェック/生成を使用可能にし、parodd で奇数パリティの生成を設定し、cs7 で文字サイズを 7 ビットに生成します。

6. 文書が正しく印刷されることを確認します。

```
# lp -d printer-name filename
```

▼ LP 印刷サービスのハングを解除する方法

1. スーパーユーザーまたは lp としてログインします。

2. **LP** 印刷サービスを停止します。

```
# lpshut
```

このコマンドがハングする場合は、Control-c キーを押して次の手順に進みます。このコマンドが正常に実行された場合は、手順 4 に進みます。

3. **LP** のプロセス **ID** を確認します。

```
# ps -el | grep lp
   134 term/a   0:01 lpsched
#
```

次の手順の *pid* には、最初のカラムのプロセス ID 番号 (PID) を使用します。

4. kill -15 コマンドを使用して、**LP** プロセスを停止します。

```
# kill -15 134
```

これで LP 印刷サービスプロセスが停止します。プロセスが停止しない場合は、最後の手段として手順 5 に進みます。

5. 最後の手段として、プロセスを強制終了します。

```
# kill -9 134
```

すべての lp プロセスが終了します。

6. 次のコマンドでは、**LP** 印刷サービスを再起動できるように、SCHEDLOCK ファイルが削除されます。

```
# rm /usr/spool/lp/SCHEDLOCK
```

7. **LP 印刷サービスを再起動します。**

```
# /usr/lib/lp/lpsched
```

LP 印刷サービスが再起動されます。スケジューラが再起動されない場合は、102ページの「印刷スケジューラを再起動する方法」を参照してください。

▼ アイドル状態になった (ハングした) プリンタの問題を解決する方法

この作業には、プリンタがアイドル状態であってはならないのにアイドル状態になるときに使用する多数の手順が含まれています。通常は各手順を順番に試しますが、順番どおりでなくてもかまいません。

プリンタの準備ができているかチェックするには

1. **プリンタ状態情報を表示します。**

```
# lpstat -p printer-name
```

表示される情報は、プリンタがアイドル状態かアクティブ状態か、使用可能か使用不可か、または印刷要求を利用できるか受け付けていないかを示します。すべて正常と思われる場合は、この節の他の手順に進んでください。`lpstat` コマンドを実行できない場合は、743ページの「LP 印刷サービスのハングを解除する方法」を参照してください。

2. **プリンタが利用できない (要求を受け付けていない) 場合は、プリンタが要求を受け付けるようにします**

```
# accept printer-name
```

プリンタは、その印刷待ち行列に要求を受け付け始めます。

3. **プリンタが使用不可になっている場合は、再び使用可能にします。**

```
# enable printer-name
```

このコマンドでは、待ち行列にある要求を処理するように、プリンタを再び使用可能にします。

印刷のフィルタリングをチェックするには

`lpstat -o` コマンドを使用して、印刷のフィルタリングをチェックします。

```
$ lpstat -o luna
luna-10           fred         1261   Mar 12 17:34 being filtered
luna-11           iggy         1261   Mar 12 17:36 on terra
luna-12           jack         1261   Mar 12 17:39 on terra
$
```

待機している最初の要求がフィルタリングされているかどうかを調べます。上の例のような出力になる場合は、ファイルがフィルタリングされています。プリンタはハングせず、要求の処理に少し時間がかかっているだけです。

プリンタ障害の後に印刷を再開するには

1. プリンタ障害に関するメッセージがある場合は、その障害を解決してください。

   プリンタ障害の警告がどのように指定されているかに応じて、メッセージを電子メールで root に送らせるか、root がログインした端末に書き出すことができます。

2. プリンタを再び使用可能にします。

   ```
   # enable printer-name
   ```

   プリンタ障害によって要求がブロックされた場合は、このコマンドで強制的に再試行します。このコマンドが動作しない場合は、この節の他の手順を続行します。

ローカル待ち行列で停止している印刷要求をリモートプリンタに送信するには

1. 印刷クライアント上で、プリンタサーバーへの印刷要求を、それ以上待ち行列に入れないようにします。

```
# reject printer-name
```

2. 印刷クライアント上で、プリンタサーバーに **ping** 要求 **(**存在をチェックする要求**)** を送信します。

```
print_client# ping print_server
print_server is alive
```

「*print_server* not available」というメッセージが表示される場合は、ネットワークに問題があります。

3. 問題を解決したら、新しい印刷要求を待ち行列に入れられるようにします。

```
# accept printer-name
```

4. 必要であれば、再びプリンタを使用可能にします。

```
# enable printer-name
```

プリンタサーバーの待ち行列で停止する印刷クライアントからの印刷要求を使用可能にするには

1. プリンタサーバー上で、印刷クライアントからプリンタサーバーへの印刷要求を、それ以上待ち行列に入れないようにします。

```
# reject printer-name
```

2. lpsched ログファイルを表示します。

```
# more /var/lp/logs/lpsched
```

表示される情報を参考にして、印刷クライアントからプリンタサーバーへの印刷要求が印刷されない原因を正確に把握できます。

3. 問題を解決したら、新しい印刷要求を待ち行列に入れられるようにします。

```
# accept printer-name
```

4. 必要であれば、プリンタサーバー上で再びプリンタを使用可能にします。

```
# enable printer-name
```

▼ 矛盾したプリンタ状態メッセージを解決する方法

1. プリンタサーバー上でプリンタが使用可能になっており、要求を受け付けているかどうかを確認します。

```
# lpstat -p printer-name
```

印刷クライアントが要求を受け付けているのに、プリンタサーバーが要求を拒否しているときは、矛盾した状態メッセージが表示されます。

2. プリンタサーバー上で、印刷クライアント上のプリンタの定義が、プリンタサーバー上のプリンタの定義と一致するかどうかを確認します。

```
# lpstat -p -l printer-name
```

印刷フィルタ、文字セット、印字ホイール、フォームなど、印刷ジョブコンポーネントの定義を調べて、印刷クライアントとサーバー上で一致し、ローカルユーザーがプリンタサーバーシステムのプリンタにアクセスできることを確認します。

第 43 章

ファイルシステムで発生する問題の解決

この章で説明する情報は次のとおりです。

fsck プログラムと、fsck プログラムを使用してファイルシステムの整合性をチェックする方法については、『*Solaris* のシステム管理 *(*第 *1* 巻*)*』の「ファイルシステムの整合性チェック」を参照してください。

fsck エラーメッセージ

通常、システムが異常終了し、ファイルシステムの最新の変更がディスクに書き込まれなかった場合に、fsck が非対話形式で実行され、ファイルシステムが修復されます。修復されると、ファイルシステムの基本的な非整合状態は自動的に修正されますが、より重大なエラーは修復されません。ファイルシステムを修復する間

に、fsck はこの種の異常終了から予想される非整合状態を修正します。より重大な状況の場合は、エラーが表示されて終了します。

fsck を対話形式で実行すると、fsck は見つかった各非整合状態を表示して小さなエラーを修正します。ただし、より重大なエラーの場合は、非整合状態を表示し、応答を選択するように促します。-y または -n オプションを指定して fsck を実行する場合、fsck が提案するデフォルト応答には、ユーザー側の応答がエラー条件ごとに yes または no にあらかじめ定義されています。

修正処置によっては、若干のデータが失われます。失われるデータの量は、fsck の診断出力から判断できます。

fsck はマルチパスファイルシステムのチェックプログラムです。パスごとに、異なるメッセージセットを使用して fsck プログラムの異なるフェーズが呼び出されます。初期化後に、fsck はファイルシステムごとに連続パスを実行して、ブロックとサイズ、パス名、接続状態、参照数、空きブロックマップをチェックします (再構築することもあります)。また、何らかのクリーンアップも実行します。

UFS バージョンの fsck によって実行されるフェーズ (パス) は次のとおりです。

- 初期化
- フェーズ 1 – ブロックとサイズのチェック
- フェーズ 2 – パス名のチェック
- フェーズ 3 – 接続状態のチェック
- フェーズ 4 – 参照数のチェック
- フェーズ 5 – シリンダグループのチェック

この後の各節では、各フェーズで検出できるエラー条件、表示されるメッセージとプロンプト、および応答できる内容について説明します。

複数のフェーズで表示されるメッセージについては、751ページの「fsck の一般エラーメッセージ」を参照してください。それ以外の場合、メッセージは発生するフェーズのアルファベット順に掲載されています。

多くのメッセージには、表 43–1 に示す省略形が含まれています。

表 43–1　エラーメッセージの省略形

省略形	意味
BLK	ブロック番号
DUP	重複ブロック番号
DIR	ディレクトリ名
CG	シリンダグループ
MTIME	ファイルの最終変更時刻
UNREF	非参照

また、多くのメッセージには、i ノード番号などの変数フィールドが含まれています。このマニュアルでは、i ノード番号を *inode-number* のようにイタリック体で掲載してあります。たとえば、次の画面メッセージは、

```
INCORRECT BLOCK COUNT I=2529
```

次の例のように掲載されています。

```
INCORRECT BLOCK COUNT I=inode-number
```

fsck の一般エラーメッセージ

この節のエラーメッセージは、初期化後のどのフェーズでも表示されることがあります。処理を続けるかどうかのオプションは表示されますが、通常は、致命的だと見なすのが最善の処置です。これらのエラーメッセージは重大なシステム障害を反映しており、ただちに処理する必要があります。この種のメッセージが表示された場合は、n(o) を入力してプログラムを終了してください。問題の原因を判断できない場合は、ご購入先に問い合わせてください。

```
CANNOT SEEK: BLK block-number (CONTINUE)
```

エラーの発生原因

ファイルシステム内で、指定されたブロック番号 *block-number* へ移動させるという要求に失敗した。このメッセージは重大な問題、おそらくハードウェア障害を示す。

ファイルシステムのチェックを続けると、fsck は移動を再び行い、移動できなかったセクタ番号のリストを表示する。このブロックが仮想メモリーバッファーキャッシュの一部であれば、fsck は致命的なエラーメッセージを表示して終了する。

解決方法

ディスクにハードウェア障害があると、この問題は解決しない。もう一度 fsck を実行してファイルシステムをチェックする。

このチェックでも解決しない場合、購入先に問い合わせる。

```
CANNOT READ: BLK block-number (CONTINUE)
```

エラーの発生原因

ファイルシステム内で指定されたブロック番号を読み込むという要求に失敗した。このメッセージは重大な問題、おそらくハードウェア障害を示す。

ファイルシステムのチェックを続けたい場合、fsck は読み取りを再試行して、読み込めなかったセクター番号のリストを表示する。ブロックが仮想メモリーバッファーキャッシュの一部であれば、fsck は致命的な入出力エラーメッセージを表示して終了する。fsck が読み取りに失敗したブロックのいずれかに書き込もうとすると、次のメッセージが表示される。fsck が読み取りに失敗したブロックのいずれかに書き込もうとすると、次のメッセージが表示される。

WRITING ZERO'ED BLOCK *sector-numbers* TO DISK

解決方法

ディスクにハードウェア障害が発生していると、この問題は継続する。もう一度 fsck を実行して、ファイルシステムをチェックし直す。このチェックでも解決しない場合、購入先に問い合わせる。

```
CANNOT WRITE: BLK block-number (CONTINUE)
```

エラーの発生原因

ファイルシステム内で、指定されたブロック番号 *block-number* への書き込みに失敗した。

ファイルシステムのチェックを続けると、fsck は書き込みを再度実行し、書き込めなかったセクタ番号のリストを表示する。ブロックが仮想メモリーバッファーキャッシュの一部であれば、fsck は致命的な入出力エラーメッセージを表示して終了する。

解決方法

ディスクが書き込み保護されている可能性がある。ドライブ上で書き込み保護ロックをチェックする。ディスクにハードウェア障害がある場合、問題は解決しない。もう一書き込み保護が原因でない場合、あるいはファイルシステムを再チェックしても問題が解決しない場合は、購入先に問い合わせる度 fsck を実行してファイルシステムをチェックする。 書き込み保護が原因でない場合、あるいはファイルシステムを再チェックしても問題が解決しない場合は、購入先に問い合わせる。

初期化フェーズでの fsck メッセージ

初期化フェーズでは、コマンド行構文がチェックされます。ファイルシステムのチェックを実行する前に、fsck はテーブルを設定してファイルを開きます。

この節のメッセージは、コマンド行オプション、メモリー要求、ファイルのオープン、ファイルの状態、ファイルシステムのサイズチェック、およびスクラッチファイルの作成によるエラー条件に関するものです。ファイルシステムを修復する間に、どんな初期化エラーが発生した場合も、fsck は終了します。

```
bad inode number inode-number to ginode
```

エラーの発生原因

inode-number が存在しないため、内部エラーが発生した。fsck は終了する。

解決方法

ご購入先に問い合わせる。

```
cannot alloc size-of-block map bytes for blockmap
cannot alloc size-of-free map bytes for freemap
cannot alloc size-of-state map bytes for statemap
cannot alloc size-of-lncntp bytes for lncntp
```

エラーの発生原因

内部テーブル用のメモリー要求に失敗した。fsck は終了する。このメッセージは、即座に処理しなければならない重大なシステム障害を示す。他のプロセスが大量のシステム資源を使用していると、このエラー条件が発生することがある。

解決方法

他のプロセスを終了すると問題を解決できることがある。解決できない場合は、ご購入先に問い合わせる。

```
cannot alloc size-of-block map bytes for blockmap
cannot alloc size-of-free map bytes for freemap
cannot alloc size-of-state map bytes for statemap
cannot alloc size-of-lncntp bytes for lncntp
```

エラーの発生原因

内部テーブル用のメモリー要求に失敗した。fsck は終了する。このメッセージは、即座に処理しなければならない重大なシステム障害を示す。他のプロセスが大量のシステム資源を使用していると、このエラー条件が発生することがある。

解決方法

他のプロセスを終了すると問題を解決できることがある。解決できない場合は、ご購入先に問い合わせる。

```
Can't open checklist file: filename
```

エラーの発生原因

ファイルシステムのチェックリストファイル *filename* (通常は /etc/vfstab) を開いて読み込めない。fsck は終了する。

解決方法

ファイルの有無と、そのアクセスモードで読み取りが可能かどうかをチェックする。

```
Can't open filename
```

エラーの発生原因

fsck はファイルシステム *filename* を開けなかった。対話形式で実行している場合、fsck はこのファイルシステムを無視し、次に指定されたファイルシステムのチェックを続ける。

解決方法

そのファイルシステムの row デバイスファイルに読み取り、または書き込みができるかどうかをチェックする。

```
Can't stat root
```

エラーの発生原因

fsck はルートディレクトリに関する統計情報要求に失敗した。fsck は終了する。

解決方法

このメッセージは、重大なシステム障害を示す。ご購入先に問い合わせる。

```
Can't stat filename
Can't make sense out of name filename
```

エラーの発生原因

fsck はファイルシステム *filename* に関する統計情報要求に失敗した。対話形式で実行している場合、fsck はこのファイルシステムを無視し、次に指定されたファイルシステムのチェックを続ける。

解決方法

ファイルシステムの有無とそのアクセスモードをチェックする。

```
filename: (NO WRITE)
```

エラーの発生原因

-n オプションが指定されているか、fsck はファイルシステム *filename* を書き込み用に開けなかった。fsck を非書き込みモードで実行中であれば、すべての診断メッセージが表示されるが、fsck は何も修正しようとしない。

解決方法

-n を指定しなかった場合は、指定したファイルのタイプをチェックする。通常ファイル名の可能性がある。

```
IMPOSSIBLE MINFREE=percent IN SUPERBLOCK (SET TO DEFAULT)
```

エラーの発生原因

スーパーブロックの最小容量が 99 パーセントを超えているか、0 パーセント未満である。

解決方法

minfree パラメタをデフォルトの 10 パーセントに設定し、デフォルトプロンプトから y と入力する。エラー条件を無視するには、デフォルトプロンプトから n と入力する。

```
filename: BAD SUPER BLOCK: message
USE AN ALTERNATE SUPER-BLOCK TO SUPPLY NEEDED INFORMATION;
e.g., fsck[-f ufs] -o b=# [special ...]
where # is the alternate superblock.  See fsck_ufs(1M)
```

エラーの発生原因

スーパーブロックが破損している

解決方法

次のいずれかのメッセージが表示される場合は、ご購入先に問い合わせる。

```
CPG OUT OF RANGE
FRAGS PER BLOCK OR FRAGSIZE WRONG
INODES PER GROUP OUT OF RANGE
INOPB NONSENSICAL RELATIVE TO BSIZE
MAGIC NUMBER WRONG
NCG OUT OF RANGE
NCYL IS INCONSISTENT WITH NCG*CPG
NUMBER OF DATA BLOCKS OUT OF RANGE
NUMBER OF DIRECTORIES OUT OF RANGE
ROTATIONAL POSITION TABLE SIZE OUT OF RANGE
SIZE OF CYLINDER GROUP SUMMARY AREA WRONG
SIZE TOO LARGE
BAD VALUES IN SUPERBLOCK
```

代替スーパーブロックを使用して fsck を再実行してみる。手始めにブロック 32 を指定するとよい。スライス上で newfs -N コマンドを実行すると、スーパーブロックの代替コピーの位置を調べることができる。-N を指定しないと、newfs は既存のファイルシステムを上書きするので注意する。

```
UNDEFINED OPTIMIZATION IN SUPERBLOCK (SET TO DEFAULT)
```

エラーの発生原因

スーパーブロックの最適化パラメタが OPT_TIME でも OPT_SPACE でもない。

解決方法

ファイルシステム上で処理の実行時間を最小限度まで短縮するには、SET TO DEFAULT プロンプトから y を入力する。このエラー条件を無視するには、n を入力する。

フェーズ 1: ブロックとサイズに関するメッセージのチェック

このフェーズでは、i ノードリストをチェックします。次の処理中に検出されたエラー条件が表示されます。

- i ノードのタイプをチェックする
- ゼロリンク数テーブルを設定する
- 不良ブロックまたは重複ブロックの有無を i ノードブロック番号で検査する
- i ノードのサイズをチェックする
- i ノードの形式をチェックする

ファイルシステムの修復 (preen) 中は、INCORRECT BLOCK COUNT、 PARTIALLY TRUNCATED INODE、 PARTIALLY ALLOCATED INODE、および UNKNOWN FILE TYPE を除き、このフェーズのどのエラーが発生した場合も、fsck が終了します。

フェーズ 1 では、次のメッセージ (アルファベット順) が発生する可能性があります。

```
block-number BAD I=inode-number
```

エラーの発生原因

i ノード *inode-number* に、ファイルシステム内の最初のデータブロックより小さい番号または最後のデータブロックより大きい番号が付いたブロック番号 *block-number* が入っている。i ノード *inode-number* 内にファイルシステムの範囲外のブロック番号が多すぎると、このエラー条件のためにフェーズ 1 で「EXCESSIVE BAD BLKS」エラーメッセージが生成されることがある。フェーズ 2 と 4 では、このエラー条件が原因で「BAD/DUP」エラーメッセージが生成される。

解決方法

ない

```
BAD MODE: MAKE IT A FILE?
```

エラーの発生原因

指定された i ノードの状態がすべて、ファイルシステムの損傷を示す 1 に設定されている。このメッセージは、fsck -y が実行された後で繰り返し表示される場合以外は、物理的なディスクの損傷を示すものではない。

解決方法

y と入力して i ノードを妥当な値に初期化し直す。

```
BAD STATE state-number TO BLKERR
```

エラーの発生原因

内部エラーによって fsck の状態マップが破壊されたため、不可能な値 *state-number* を示す。fsck は即座に終了する。

解決方法

ご購入先に問い合わせる。

```
block-number DUP I=inode-number
```

エラーの発生原因

i ノード *inode-number* には、同じ i ノードまたは別の i ノードがすでに取得したブロック番号 *block-number* が入っている。このエラー条件が発生した場合に、i ノード *inode-number* 内にこの種のブロック番号が多すぎると、フェーズ 1 では

「EXCESSIVE DUP BLKS」エラーメッセージが生成されることがある。このエラー条件によってフェーズ 1B が呼び出され、フェーズ 2 と 4 で「BAD/DUP」エラーメッセージが生成される。

解決方法

ない

```
DUP TABLE OVERFLOW (CONTINUE)
```

エラーの発生原因

fsck の内部テーブルには、重複するブロック番号が入る余地がない。-o p (preen、修復) オプションを指定すると、プログラムが終了する。

解決方法

プログラムを続行するには、CONTINUE プロンプトから y と入力する。このエラーが発生すると、ファイルシステムを完全にチェックできない。別の重複ブロックが見つかると、このエラー条件が再発する。使用可能な仮想メモリーの容量を (プロセスを終了し、スワップ空間を拡張して) 大きくし、もう一度 fsck を実行してファイルシステムをチェックし直す。プログラムを終了するには、n と入力する。

```
EXCESSIVE BAD BLOCKS I=inode-number (CONTINUE)
```

エラーの発生原因

i ノード *inode-number* に関連付けられたファイルシステム内の最初のデータブロックより小さい番号か、最後のブロックより大きい番号を持つブロックが多すぎる (通常は 10 以上)。-o p (preen、 修復) オプションを指定すると、プログラムは終了する

解決方法

プログラムを続行するには、CONTINUE プロンプトから y と入力する。このエラーが発生すると、ファイルシステムを完全にチェックできない。もう一度 fsck を実行してファイルシステムをチェックし直す必要がある。プログラムを終了するには、n と入力する。

```
EXCESSIVE DUP BLKS I=inode-number (CONTINUE)
```

エラーの発生原因

同じ i ノード、別の i ノード、または空きリストが取得するブロック数が多すぎる (通常は 10 以上)。-o p (preen、修復) オプションを指定すると、プログラムは終了する。

解決方法

プログラムを続行するには、CONTINUE プロンプトから y と入力する。このエラーが発生すると、ファイルシステムを完全にチェックできない。もう一度 fsck を実行してファイルシステムをチェックし直す必要がある。プログラムを終了するには、n と入力する。

```
INCORRECT BLOCK COUNT I=inode-number (number-of-BAD-DUP-or-missing-blocks should be
 number-of-blocks-in-filesystem) (CORRECT)
```

エラーの発生原因

i ノード *inode-number* のブロック数は *number-of-BAD-DUP-or-missing-blocks* であるが、*number-of-blocks-in-filesystem* でなければならない。修復 (preen) の場合、fsck は数を訂正する。

解決方法

i ノード *inode-number* のブロック数を *number-of-blocks-in-filesystem* に置き換えるには、CORRECT プロンプトから y と入力する。プログラムを終了するには、n と入力する。

```
LINK COUNT TABLE OVERFLOW (CONTINUE)
```

エラーの発生原因

fsck の内部テーブルには、リンク数が 0 の割り当て済み i ノードが入る余地がない。-o p (preen、修復) オプションを指定すると、プログラムは終了するので、fsck を手作業で終了する必要がある。

解決方法

プログラムを続行するには、CONTINUE プロンプトから y と入力する。リンク数が 0 の別の割り当て済みブロックが見つかると、このエラー条件が再発する。このエラーが発生すると、ファイルシステムを完全にチェックできない。もう一度 fsck

を実行してファイルシステムをチェックし直す必要がある。プロセスをいくつか終了するか、スワップ領域を拡張して、使用可能な仮想メモリーを増やしてから、fsck を実行し直す。プログラムを終了するには、n と入力する。

```
PARTIALLY ALLOCATED INODE I=inode-number (CLEAR)
```

エラーの発生原因

i ノード *inode-number* は割り当て済みでも未割り当てでもない。-o p (preen、修復) オプションを指定すると、この i ノードは消去される。

解決方法

i ノード *inode-number* の内容を消去して割り当てを解除するには、y と入力する。これにより、この i ノードを指すディレクトリごとに、フェーズ 2 でエラー条件 UNALLOCATED が生成されることがある。このエラー条件を無視するには、n と入力する。応答しなくてよいのは、この問題を他の手段で解決しようとする場合だけである。

```
PARTIALLY TRUNCATED INODE I=inode-number (SALVAGE)
```

エラーの発生原因

fsck で、割り当てられたブロック数よりも短い i ノード *inode-number* が見つかった。この条件が発生するのは、ファイルの切り捨て中にシステムがクラッシュした場合だけである。ファイルシステムを修復しているとき、fsck は指定されたサイズへの切り捨てを完了する。

解決方法

i ノード内で指定したサイズへの切り捨てを完了するには、SALVAGE プロンプトから y と入力する。このエラー条件を無視するには、n と入力する。

```
UNKNOWN FILE TYPE I=inode-number (CLEAR)
```

エラーの発生原因

i ノード *inode-number* のモードのワードは、この i ノードがパイプ、特殊文字 i ノード、特殊ブロック i ノード、通常 i ノード、シンボリックリンク、FIFO ファイル、またはディレクトリ i ノードでないことを示す。-o p (preen、修復) オプションを指定すると、この i ノードは消去される。

解決方法

i ノード *inode-number* の内容を消去して割り当て解除するには、CLEAR プロンプトから y と入力する。これにより、この i ノードを指すディレクトリエントリごとに、フェーズ 2 でエラー条件 UNALLOCATED が生成される。このエラー条件を無視するには n と入力する。

フェーズ 1B : 走査し直して DUPS メッセージを表示する

ファイルシステム内で重複ブロックが見つかると、次のメッセージが表示されます。

block-number DUP I=*inode-number*

エラーの発生原因

i ノード *inode-number* には、すでに同じ i ノードまたは別の i ノードによって取得されたブロック番号 *block-number* が入っている。このエラー条件によって、フェーズ 2 で BAD/DUP エラーメッセージが生成される。重複ブロックを持つ i ノードは、このエラー条件とフェーズ 1 の DUP エラー条件を検査すれば判断できる。

解決方法

重複ブロックが見つかると、ファイルシステムが再び走査され、以前にそのブロックを取得した i ノードが検索される。

フェーズ 2: パス名メッセージのチェック

このフェーズでは、フェーズ 1 と 1B で見つかった不良 i ノードを指すディレクトリエントリが削除される。次の原因でエラー条件が表示される。

- 不正なルート i ノードモードと状態
- 範囲外のディレクトリ i ノードポインタ
- 不良 i ノードを指すディレクトリエントリ
- ディレクトリ完全性チェック

ファイルシステムを修復している場合は (–o p (preen、修復) オプション)、このフェーズでどのエラーが発生した場合も、fsck が終了します。ただし、ブロック

サイズの倍数でないディレクトリ、重複ブロックと不良ブロック、範囲外の i ノード、過剰なハードリンクに関連するエラーは除きます。

フェーズ 2 では、次のメッセージ (アルファベット順) が表示される可能性があります。

```
BAD INODE state-number TO DESCEND
```

エラーの発生原因

fsck の内部エラーによって、ファイルシステムのディレクトリ構造を継承するルーチンに、無効な状態 *state-number* が渡された。fsck は終了する。

解決方法

このエラーメッセージが表示された場合は、ご購入先に問い合わせる。

```
BAD INODE NUMBER FOR '.' I=inode-number OWNER=UID MODE=file-mode
SIZE=file-size MTIME=modification-time DIR=filename (FIX)
```

エラーの発生原因

「.」の i ノード番号が *inode-number* に等しくないディレクトリ *inode-number* が見つかった。

解決方法

「.」の i ノード番号を *inode-number* に等しくなるように変更するには、FIX プロンプトから y と入力する。「.」の i ノード番号を変更しない場合は、n と入力する。

```
BAD INODE NUMBER FOR '..' I=inode-number OWNER=UID MODE=file-mode
SIZE=file-size MTIME=modification-time DIR=filename (FIX)
```

エラーの発生原因

「..」の i ノード番号が *inode-number* の親に等しくないディレクトリ *inode-number* が見つかった。

解決方法

「..」の i ノード番号を *inode-number* の親に等しくなるように変更するには、FIX プロンプトから y と入力する (ルート i ノード内の「..」は、それ自体を指すので注意する)。「..」の i ノード番号を変更しない場合は、n と入力する。

```
BAD RETURN STATE state-number FROM DESCEND
```

エラーの発生原因

fsck の内部エラーによって、ファイルシステムのディレクトリ構造を継承するルーチンから、不可能な状態 *state-number* が返された。fsck は終了する。

解決方法

このメッセージが表示される場合は、ご購入先に問い合わせる。

```
BAD STATE state-number FOR ROOT INODE
```

エラーの発生原因

内部エラーによって、ルート i ノードに不可能な状態 *state-number* が割り当てられた。fsck は終了する。

解決方法

このメッセージが表示される場合は、ご購入先に問い合わせる。

```
BAD STATE state-number FOR INODE=inode-number
```

エラーの発生原因

内部エラーによって、i ノード *inode-number* に不可能な状態 *state-number* が割り当てられた。fsck は終了する。

解決方法

このメッセージが表示される場合は、ご購入先に問い合わせる

```
DIRECTORY TOO SHORT I=inode-number OWNER=UID MODE=file-mode
SIZE=file-size MTIME=modification-time DIR=filename (FIX)
```

エラーの発生原因

サイズ *file-size* が最小ディレクトリサイズより小さいディレクトリ *filename* が見つかった。所有者 *UID*、モード *file-mode*、サイズ *file-size*、変更時刻 *modification-time*、およびディレクトリ名 *filename* が表示される。

解決方法

ディレクトリのサイズを最小ディレクトリサイズまで大きくするには、FIX プロンプトから y と入力する。このディレクトリを無視するには n と入力する。

```
DIRECTORY filename: LENGTH file-size NOT MULTIPLE OF block-number (ADJUST)
```

エラーの発生原因

サイズ *file-size* がディレクトリブロックのサイズ *block-number* の倍数でないディレクトリ *filename* が見つかった。

解決方法

長さを適切なブロックサイズに切り上げるには、y と入力する。ファイルシステムを修復しているとき (-o p (preen、修復)、オプション) は、fsck は警告のみを表示してディレクトリを調整する。この条件を無視するには n と入力する。

```
DIRECTORY CORRUPTED I=inode-number OWNER=UID MODE=file-mode
SIZE=file-size MTIME=modification-time DIR=filename (SALVAGE)
```

エラーの発生原因

内部状態の整合性がないディレクトリが見つかった。

解決方法

次のディレクトリ境界 (通常は 512 バイトの境界) までのすべてのエントリを放棄するには、SALVAGE プロンプトから y と入力する。この処置によって、最高で 42 個のエントリを放棄できる。この処置は、他の回復作業に失敗した場合にのみ実行する。問題のディレクトリを変更せずに、次のディレクトリ境界までスキップして読み取りを再開するには、n と入力する。

```
DUP/BAD I=inode-number OWNER=O MODE=M SIZE=file-size
MTIME=modification-time TYPE=filename (REMOVE)
```

エラーの発生原因

フェーズ 1 またはフェーズ 1B で、ディレクトリまたはファイルエントリ *filename*、i ノード *inode-number* に関連付けられた重複ブロックまたは不良ブロックが見つかった。所有者 *UID*、モード *file-mode*、サイズ *file-size*、変更時刻 *modification-time*、ディレクトリまたはファイル名 *filename* が表示される。–p (preen、修復) オプションを指定すると、重複または不良ブロックが削除される。

解決方法

ディレクトリまたはファイルのエントリ *filename* を削除するには、REMOVE プロンプトから y と入力する。このエラー条件を無視するには n と入力する。

```
DUPS/BAD IN ROOT INODE (REALLOCATE)
```

エラーの発生原因

フェーズ 1 またはフェーズ 1B で、ファイルシステムのルート i ノード (通常は i ノード番号 2) に、重複ブロックまたは不良ブロックが見つかった。

解決方法

ルート i ノードの既存の内容を消去して割り当てを解除するには、REALLOCATE プロンプトから y と入力する。ルート内で通常検出されるファイルとディレクトリがフェーズ 3 で復元され、lost+found ディレクトリに格納される。ルートの割り当てに失敗すると、fsck は「CANNOT ALLOCATE ROOT INODE」というメッセージを表示して終了する。CONTINUE プロンプトを表示するには、n と入力する。CONTINUE プロンプトに応答するには、y と n のどちらかを入力する。y と入力すると、ルート i ノード内の DUPS/BAD エラー条件を無視して、ファイルシステムのチェックを続行する。ルート i ノードが不正であれば、他の多数のエラーメッセージが生成されることがある。n の場合は、プログラムを終了する。

```
EXTRA '.' ENTRY I=inode-number OWNER=UID MODE=file-mode
SIZE=file-size MTIME=modification-time DIR=filename (FIX)
```

エラーの発生原因

「.」のエントリが複数個入っているディレクトリ *inode-number* が見つかった。

解決方法

「.」の余分なエントリを削除するには、FIX プロンプトから y と入力する。問題のディレクトリを変更しない場合は、n と入力する。

```
EXTRA '..' ENTRY I=inode-number OWNER=UID MODE=file-mode
SIZE=file-size MTIME=modification-time DIR=filename(FIX)
```

エラーの発生原因

「..」(親ディレクトリ) のエントリが複数個入っているディレクトリ *inode-number* が見つかった。

解決方法

「..」(親ディレクトリ) の余分なエントリを削除するには、FIX プロンプトから y と入力する。問題のディレクトリを変更しない場合は、n と入力する。

```
hard-link-number IS AN EXTRANEOUS HARD LINK TO A DIRECTORY filename (REMOVE)
```

エラーの発生原因

fsck によって、ディレクトリ *filename* へのハードリンク *hard-link-number* にエラーが見つかった。修復 (preen) しているとき (-o p オプション)、fsck はエラーのあるハードリンクを無視する。

解決方法

エラーのあるエントリ *hard-link-number* を削除するには、 プロンプトから y と入力する。エラー条件を無視するには n と入力する。

```
inode-number OUT OF RANGE I=inode-number NAME=filename (REMOVE)
```

エラーの発生原因

ディレクトリエントリ *filename* には、i ノードリストの終わりより大きい i ノード番号 *inode-number* が付いている。–p (preen、修復) オプションを指定すると、i ノードが自動的に削除される。

解決方法

ディレクトリエントリ *filename* を削除するには、REMOVE プロンプトから y と入力する。エラー条件を無視するには、n と入力する。

```
MISSING '.' I=inode-number OWNER=UID MODE=file-mode SIZE=file-size
MTIME=modification-time DIR=filename (FIX)
```

エラーの発生原因

最初のエントリ (「.」のエントリ) に未割り当てのディレクトリ *inode-number* が見つかった。

解決方法

i ノード番号が *inode-number* に等しい「.」のエントリを構築するには、FIX プロンプトから y と入力する。問題のディレクトリを変更しない場合は、n と入力する。

```
MISSING '.' I=inode-number OWNER=UID MODE=file-mode SIZE=file-size
MTIME=modification-time DIR=filename CANNOT FIX, FIRST ENTRY IN
DIRECTORY CONTAINS filename
```

エラーの発生原因

最初のエントリが *filename* となっているディレクトリ *inode-number* が見つかった。fsck はこの問題を解決できない。

解決方法

このエラーメッセージが表示される場合は、ご購入先に問い合わせる。

```
MISSING '.' I=inode-number OWNER=UID MODE=file-mode SIZE=file-size
MTIME=modification-time DIR=filename CANNOT FIX, INSUFFICIENT
SPACE TO ADD '.'
```

エラーの発生原因

最初のエントリが「.」でないディレクトリ *inode-number* が見つかった。fsck は問題を解決できない。

解決方法

このエラーメッセージが表示される場合は、ご購入先に問い合わせる。

```
MISSING '..' I=inode-number OWNER=UID MODE=file-mode SIZE=file-size
MTIME=modification-time DIR=filename (FIX)
```

エラーの発生原因

第 2 のエントリが割り当てられていないディレクトリ *inode-number* が見つかった。

解決方法

i ノード番号が *inode-number* の親に等しい「..」のエントリを構築するには、FIX プロンプトから y と入力する (ルート i ノード内の「..」は、それ自体を指すので注意する)。問題のディレクトリを変更しない場合は、n と入力する。

```
MISSING '..' I=inode-number OWNER=UID MODE=file-mode SIZE=file-size
MTIME=modification-time DIR=filename CANNOT FIX, SECOND ENTRY IN
DIRECTORY CONTAINS filename
```

エラーの発生原因

第 2 のエントリが *filename* となっているディレクトリ *inode-number* が見つかった。fsck はこの問題を解決できない。

解決方法

このエラーメッセージが表示される場合は、ご購入先に問い合わせる。

```
MISSING '..' I=inode-number OWNER=UID MODE=file-mode SIZE=file-size
MTIME=modification-time DIR=filename CANNOT FIX, INSUFFICIENT SPACE
TO ADD '..'
```

エラーの発生原因

第 2 のエントリが「..」(親ディレクトリ) でないディレクトリ *inode-number* が見つかった。fsck はこの問題を解決できない。

解決方法

このエラーメッセージが表示される場合は、ご購入先に問い合わせる。

```
NAME TOO LONG filename
```

エラーの発生原因

長すぎるパス名が見つかった。通常、これはファイルシステムの名前空間内のループを示す。特権を持つユーザーがディレクトリへの循環リンクを作成すると、このエラーが発生することがある。

解決方法

循環リンクを削除する。

```
ROOT INODE UNALLOCATED (ALLOCATE)
```

エラーの発生原因

ルート i ノード (通常は i ノード番号 2) に割り当てモードビットがない。

解決方法

i ノード 2 をルート i ノードとして割り当てるには、ALLOCATE プロンプトから y と入力する。通常、ルート内で検出されるファイルとディレクトリがフェーズ 3 で復元され、lost+found ディレクトリに格納される。ルートの割り当てに失敗すると、fsck は「CANNOT ALLOCATE ROOT INODE」というメッセージを表示して終了する。プログラムを終了するには n と入力する。

```
ROOT INODE NOT DIRECTORY (REALLOCATE)
```

エラーの発生原因

ファイルシステムのルート i ノード (通常は i ノード番号 2) はディレクトリ i ノードではない。

解決方法

ルート i ノードの既存の内容を消去して再割り当てを行うには、REALLOCATE プロンプトから y と入力する。一般にルート内で検出されるファイルとディレクトリがフェーズ 3 で復元され、lost+found ディレクトリに格納される。ルートの割り当てに失敗すると、fsck は「CANNOT ALLOCATE ROOT INODE」というメッセージを表示して終了する。fsck に FIX プロンプトを表示させるには、n と入力する。

```
UNALLOCATED I=inode-number OWNER=UID MODE=file-mode SIZE=file-size
MTIME=modification-time type=filename(REMOVE)
```

エラーの発生原因

ディレクトリまたはファイルのエントリ *filename* は、未割り当ての i ノード *inode-number* を指している。所有者 *UID*、モード *file-mode*、サイズ *file-size*、変更時刻 *modification-time*、およびファイル名 *filename* が表示される。

解決方法

ディレクトリエントリ *filename* を削除するには、REMOVE プロンプトから y と入力する。エラー条件を無視するには n と入力する。

```
ZERO LENGTH DIRECTORY I=inode-number OWNER=UID MODE=file-mode
SIZE=file-size MTIME=modification-time DIR=filename (REMOVE)
```

エラーの発生原因

ディレクトリエントリ *filename* のサイズ *file-size* が 0 になっている。所有者 *UID*、モード *file-mode*、サイズ *file-size*、変更時刻 *modification-time*、およびディレクトリ名 *filename* が表示される。

解決方法

ディレクトリエントリ *filename* を削除するには、REMOVE プロンプトから y と入力する。これにより、フェーズ 4 で「BAD/DUP」エラーメッセージが表示される。エラー条件を無視するには n と入力する。

フェーズ 3: 接続性メッセージのチェック

このフェーズでは、フェーズ 2 で検査したディレクトリがチェックされ、次の原因によるエラー条件が表示されます。

- 参照されないディレクトリ
- 欠落しているか、フルになっている lost+found ディレクトリ

フェーズ 3 では、次のメッセージがアルファベット順に表示される可能性があります。

```
BAD INODE state-number TO DESCEND
```

エラーの発生原因

内部エラーによって、ファイルシステムのディレクトリ構造を継承するルーチンに、不可能な状態 *state-number* が渡された。`fsck` は終了する。

解決方法

このエラーが発生した場合は、ご購入先に問い合わせる。

```
DIR I=inode-number1 CONNECTED. PARENT WAS I=inode-number2
```

エラーの発生原因

これは、ディレクトリ i ノード *inode-number1* が `lost+found` ディレクトリに正常に接続されていることを示す。ディレクトリ i ノード *inode-number1* の親 i ノード *inode-number2* は、`lost+found` ディレクトリの i ノード番号に置き換えられる。

解決方法

ない

```
DIRECTORY filename LENGTH file-size NOT MULTIPLE OF block-number (ADJUST)
```

エラーの発生原因

サイズ *file-size* がディレクトリのブロックサイズ B の倍数でないディレクトリ *filename* が見つかった (この条件は、フェーズ 2 で調整しなければ、フェーズ 3 で再発することがある)。

解決方法

長さを適切なブロックサイズまで切り上げるには、`ADJUST` プロンプトから `y` と入力する。修復しているときは、`fsck` は警告を表示してディレクトリを調整する。このエラー条件を無視するには `n` と入力する。

```
lost+found IS NOT A DIRECTORY (REALLOCATE)
```

エラーの発生原因

`lost+found` のエントリがディレクトリではない。

解決方法

ディレクトリ i ノードを割り当てて、それを参照する lost+found ディレクトリを変更するには、REALLOCATE プロンプトから y と入力する。以前に lost+found ディレクトリによって参照されていた i ノードは消去されず、非参照の i ノードとして再び取得されるか、このフェーズの後半でそのリンク数が調整される。lost+found ディレクトリを作成できない場合は、「SORRY. CANNOT CREATE lost+found DIRECTORY」というメッセージが表示され、消失 i ノードへのリンク試行が中止される。これにより、フェーズ 4 で UNREF エラーメッセージが生成される。フェーズ 4 で UNREF エラーメッセージを生成する消失 i ノードへのリンク試行を中止するには、n と入力する。

```
NO lost+found DIRECTORY (CREATE)
```

エラーの発生原因

ファイルシステムのルートディレクトリ内に lost+found ディレクトリがない。修復しているときは、fsck は lost+found ディレクトリを作成しようとする。

解決方法

ファイルシステムのルート内で lost+found ディレクトリを作成するには、CREATE プロンプトから y と入力する。このため、「NO SPACE LEFT IN / (EXPAND)」というメッセージが表示されることがある。lost+found ディレクトリを作成できなければ、fsck は「SORRY. CANNOT CREATE lost+found DIRECTORY」というメッセージを表示して、消失した i ノードへのリンク試行を中止する。これにより、フェーズ 4 の後半で UNREF エラーメッセージが生成される。消失した i ノードへのリンク試行を中止するには、n と入力する。

```
NO SPACE LEFT IN /lost+found (EXPAND)
```

エラーの発生原因

使用可能な領域がないため、ファイルシステムのルートディレクトリ内で、lost+found ディレクトリに別のエントリを追加できない。修復しているときに、fsck は lost+found ディレクトリを拡張する。

解決方法

lost+found ディレクトリを拡張して新しいエントリを追加する余地をつくるには、EXPAND プロンプトから y と入力する。拡張試行に失敗すると、fsck は「SORRY. NO SPACE IN lost+found DIRECTORY」というメッセージを表示し

て、lost+found ディレクトリへのファイルリンク要求を中止する。このエラーによって、フェーズ 4 の後半で UNREF エラーメッセージが生成される。lost+found ディレクトリ内で不要なエントリを削除する。修復が有効な場合は、このエラーに fsck が終了する。消失 i ノードへのリンク試行を中止するには、n と入力する。

```
UNREF DIR I=inode-number OWNER=UID MODE=file-mode SIZE=file-size
MTIME=modification-time (RECONNECT)
```

エラーの発生原因

ファイルシステムの走査中に、ディレクトリ i ノード *inode-number* がディレクトリエントリに接続されなかった。ディレクトリ i ノード *inode-number* の所有者 *UID*、モード *file-mode*、サイズ *file-size*、および変更時刻 *modification-time* が表示される。修復しているときは、ディレクトリサイズが 0 でなければ、fsck は空でないディレクトリ i ノードを接続し直す。それ以外の場合、fsck はディレクトリ i ノードを消去する。

解決方法

ディレクトリ i ノード *inode-number* を lost+found ディレクトリに接続し直すには、RECONNECT プロンプトから y と入力する。ディレクトリが再び正常に接続されると、「CONNECTED」というメッセージが表示される。それ以外の場合は、lost+found エラーメッセージのいずれかが表示される。このエラー条件を無視するには n と入力する。このエラーにより、フェーズ 4 で UNREF エラー条件が発生する。

フェーズ 4: 参照数メッセージのチェック

このフェーズでは、フェーズ 2 と 3 で取得したリンク数情報がチェックされます。次の原因によるエラー条件が表示されます。

- 非参照ファイル
- 見つからないか、lost+found ディレクトリがフル
- ファイル、ディレクトリ、シンボリックリンク、または特殊ファイルの不正なリンク数
- 非参照ファイル、シンボリックリンク、ディレクトリ
- ファイルとディレクトリ内の不良ブロックまたは重複ブロック

■ 不正な合計空き i ノード数

このフェーズのすべてのエラー (lost+found ディレクトリ内の容量不足を除く) は、ファイルシステムを修復するときに解決できます。

フェーズ 4 では、次のメッセージ (アルファベット順) が表示される可能性があります。

```
BAD/DUP type I=inode-number OWNER=UID MODE=file-mode SIZE=file-size
MTIME=modification-time (CLEAR)
```

エラーの発生原因

フェーズ 1 またはフェーズ 1B で、ファイルまたはディレクトリ i ノード *inode-number* に関連付けられた重複ブロックまたは不良ブロックが見つかった。i ノード *inode-number* の所有者 *UID*、モード *file-mode*、サイズ *file-size*、および変更時刻 *modification-time* が表示される。

解決方法

i ノード *inode-number* の内容を消去して割り当てを解除するには、CLEAR プロンプトから y と入力する。このエラー条件を無視するには、n と入力する。

```
(CLEAR)
```

エラーの発生原因

直前の UNREF エラーメッセージで記述された i ノードを再び接続できない。ファイルシステムを修復していると、ファイルを接続し直すには容量が足りないため fsck が終了するので、このメッセージは表示されない。

解決方法

i ノードの内容を消去して割り当てを解除するには、CLEAR プロンプトから y と入力する。直前のエラー条件を無視するには、n と入力する。

```
LINK COUNT type I=inode-number OWNER=UID MODE=file-mode
SIZE=file-size
MTIME=modification-time COUNT link-count SHOULD BE
corrected-link-count (ADJUST)
```

エラーの発生原因

ディレクトリまたはファイル i ノード *inode-number* のリンク数は *link-count* になっているが、*corrected-link-count* でなければならない。i ノード *inode-number* の所有者 *UID*、モード *file-mode*、サイズ *file-size*、および変更時刻 *modification-time* が表示される。-o p (preen、修復) オプションを指定すると、参照数が増えていない限り、リンク数が調整される。この条件は、ハードウェア障害がなければ発生しない。参照数が修復中に増えると、fsck は「LINK COUNT INCREASING」というメッセージを表示して終了する。

解決方法

ディレクトリまたはファイル i ノード *inode-number* のリンク数を *corrected-link-count* に置き換えるには、ADJUST プロンプトから y と入力する。このエラー条件を無視するには、n と入力する。

```
lost+found IS NOT A DIRECTORY (REALLOCATE)
```

エラーの発生原因

lost+found のエントリがディレクトリではない。

解決方法

ディレクトリ i ノードを割り当てて、それを参照する lost+found ディレクトリを変更するには、REALLOCATE プロンプトから y と入力する。lost+found による以前の i ノード参照は消去されない。非参照 i ノードとして再び取得されるか、そのリンク数がこのフェーズの後半で調整される。lost+found ディレクトリを作成できなければ、「SORRY. CANNOT CREATE lost+found DIRECTORY」というメッセージが表示され、消失 i ノードへのリンク試行が中止される。このエラーにより、フェーズ 4 の後半で UNREF エラーメッセージが生成される。消失 i ノードへのリンク試行を中止するには、n と入力する。

```
NO lost+found DIRECTORY (CREATE)
```

エラーの発生原因

ファイルシステムのルートディレクトリ内に lost+found ディレクトリがない。修復するときに、fsck は lost+found ディレクトリを作成しようとする。

解決方法

ファイルシステムのルート内で lost+found ディレクトリを作成するには、CREATE プロンプトから y と入力する。lost+found ディレクトリを作成できなければ、fsck は「SORRY. CANNOT CREATE lost+found DIRECTORY」というメッセージを表示して、消失 i ノードへのリンク試行を中止する。このエラーにより、フェーズ 4 の後半で UNREF エラーメッセージが生成される。消失 i ノードへのリンク試行を中止するには、n と入力する。

```
NO SPACE LEFT IN / lost+found (EXPAND)
```

エラーの発生原因

ファイルシステムのルートディレクトリ内で、lost+found ディレクトリに別のエントリを追加する容量がない。修復するときに、fsck は lost+found ディレクトリを拡張する。

解決方法

lost+found ディレクトリを拡張して新しいエントリを追加する余地をつくるには、EXPAND プロンプトから y と入力する。拡張試行に失敗すると、fsck は「SORRY. NO SPACE IN lost+found DIRECTORY」というメッセージを表示して、lost+found ディレクトリへのファイル数要求を中止する。このエラーにより、フェーズ 4 の後半で UNREF エラーメッセージが生成される。修復 (-o p オプション) が有効なときは、このエラーによって fsck が終了する。消失 i ノードへのリンク試行を中止するには、n と入力する。

```
UNREF FILE I=inode-number OWNER=UID MODE=file-mode SIZE=file-size
MTIME=modification-time (RECONNECT)
```

エラーの発生原因

ファイルシステムを走査するときに、ファイル i ノード *inode-number* がディレクトリエントリに接続されなかった。i ノード *inode-number* の所有者 *UID*、モード *file-mode*、サイズ *file-size*、および変更時刻 *modification-time* が表示される。fsck が修復しているときに、ファイルのサイズまたはリンク数が 0 であれば、そのファイルは消去される。それ以外の場合は、再び接続される。

解決方法

i ノード *inode-number* を `lost+found` ディレクトリ内のファイルシステムに接続し直すには、`y` と入力する。i ノード *inode-number* を `lost+found` ディレクトリに接続できないと、このエラーによってフェーズ 4 で `lost+found` エラーメッセージが生成されることがある。このエラー条件を無視するには、`n` と入力する。このエラーが発生すると、フェーズ 4 で必ず `CLEAR` エラー条件が呼び出される。

```
UNREF type I=inode-number OWNER=UID MODE=file-mode SIZE=file-size
MTIME=modification-time (CLEAR)
```

エラーの発生原因

ファイルシステムを走査するときに、i ノード *inode-number* (その *type* はディレクトリまたはファイル) がディレクトリエントリに接続されなかった。i ノード *inode-number* の所有者 *UID*、モード *file-mode*、サイズ *file-size*、および変更時刻 *modification-time* が表示される。`fsck` が修復しているときに、ファイルのサイズまたはリンク数が 0 であれば、そのファイルは消去される。それ以外の場合は再び接続される。

解決方法

i ノード *inode-number* の内容を消去して割り当てを解除するには、`CLEAR` プロンプトから `y` と入力する。このエラー条件を無視するには、`n` と入力する。

```
ZERO LENGTH DIRECTORY I=inode-number OWNER=UID MODE=file-mode
SIZE=file-size MTIME=modification-time (CLEAR)
```

エラーの発生原因

ディレクトリエントリ *filename* のサイズ *file-size* が 0 になっている。所有者 *UID*、モード *file-mode*、サイズ *file-size*、変更時刻 *modification-time*、およびディレクトリ名 *filename* が表示される。

解決方法

ディレクトリエントリ *filename* のサイズ *file-size* が 0 になっている。所有者 *UID*、モード *file-mode*、サイズ *file-size*、変更時刻 *modification-time*、およびディレクトリ名 *filename* が表示される。

フェーズ 5: シリンダグループメッセージのチェック

このフェーズでは、空きブロックと使用済み i ノードのマップがチェックされます。次の原因によるエラー条件が表示されます。

- 使用済み i ノードマップから欠落している割り当て済み i ノード
- 空きブロックマップから欠落している空きブロック
- 使用済み i ノードマップ内の空き i ノード
- 不正な合計空きブロック数
- 不正な合計使用済み i ノード数

フェーズ 5 では、次のメッセージがアルファベット順に表示される可能性があります。

```
BLK(S) MISSING IN BIT MAPS (SALVAGE)
```

エラーの発生原因

シリンダグループのブロックマップから空きブロックがいくつか欠落している。修復中に、fsck はマップを作成し直す。

解決方法

空きブロックマップを作成し直すには、SALVAGE プロンプトから y と入力する。このエラー条件を無視するには、n と入力する。

```
CG character-for-command-option: BAD MAGIC NUMBER
```

エラーの発生原因

シリンダグループ *character-for-command-option* のマジック番号が間違っている。通常、このエラーはシリンダグループマップが破壊されていることを示す。対話形式で実行している場合は、シリンダグループに再度の作成が必要であることを示すマークが付けられる。ファイルシステムを修復している場合は、fsck が終了する。

解決方法

このエラーが発生する場合は、ご購入先に問い合わせる。

```
FREE BLK COUNT(S) WRONG IN SUPERBLK (SALVAGE)
```

エラーの発生原因

空きブロック数の実際の数が、ファイルシステムのスーパーブロック内の空きブロック数と一致しない。-o p (preen、修復) オプションを指定した場合は、スーパーブロック内の空きブロック数が自動的に修正される。

解決方法

スーパーブロックの空きブロック情報を作成し直すには、SALVAGE プロンプトから y と入力する。このエラー条件を無視するには、n と入力する。

```
SUMMARY INFORMATION BAD (SALVAGE)
```

エラーの発生原因

集計情報が間違っている。修復していると、fsck は集計情報を計算し直す。

解決方法

集計情報を作成し直すには、SALVAGE プロンプトから y と入力する。このエラー条件を無視するには、n と入力する。

クリーンアップフェーズのメッセージ

ファイルシステムのチェックが終わると、クリーンアップ処理がいくつか実行されます。クリーンアップフェーズでは、次の状態メッセージが表示されます。

number-of files, *number-of-files* used, *number-of-files* free (*number-of* frags, *number-of* blocks, *percent* fragmentation)

上記のメッセージは、チェックされたファイルシステムに、フラグメントサイズの *number-of* 個のブロックを使用中の *number-of* 個のファイルが入っていることと、ファイルシステム内でフラグメントサイズのブロックが *number-of* 個空いていることを示します。括弧内の数は、空いている数を *number-of* 個の空きフラグメント、*number-of* 個の完全サイズの空きブロック、および *percent* のフラグメントに分割したものです。

```
***** FILE SYSTEM WAS MODIFIED *****
```

上記のメッセージは、ファイルシステムが fsck によって変更されたことを示します。このファイルシステムがマウントされているか、現在のルート (/) ファイルシステムの場合はリブートします。ファイルシステムがマウントされている場合は、マウント解除して再び fsck を実行する必要があります。そうしないと、fsck によって実行された処理がテーブルのインコアコピー (カーネル内のコピー) によって取り消されます。

```
filename FILE SYSTEM STATE SET TO OKAY
```

上記のメッセージは、ファイルシステム *filename* に安定を示すマークが付けられたことを示します。-m オプションを指定して fsck を実行すると、この情報を使用して、ファイルシステムのチェックが必要かどうかが判断されます。

```
filename FILE SYSTEM STATE NOT SET TO OKAY
```

上記のメッセージは、ファイルシステム *filename* に安定を示すマークが付いていないことを示します。-m オプションを指定して fsck を実行すると、この情報を使用して、ファイルシステムにチェックが必要かどうかが判断されます。

第 44 章

ソフトウェア管理の問題の解決

この章では、ソフトウェアパッケージをインストールまたは削除するときに発生する問題について説明します。この章には、2 つの節があります。「特定のソフトウェア管理エラー」では、パッケージのインストールエラーと管理エラーについて説明します。「一般的なソフトウェア管理障害」では、特定のエラーメッセージを出さない障害について説明します。

この章の内容は次のとおりです。

ソフトウェアパッケージの管理については、『*Solaris* のシステム管理 *(*第 *1* 巻*)*』の「ソフトウェア管理 (概要)」を参照してください。

ソフトウェア管理の問題解決における新しい機能

以前の Solaris リリースでは、ソフトウェアパッケージを作成するときに、シンボリックリンクのリンク先を pkgmap ファイルに指定できませんでした。そのため、pkgadd コマンドでパッケージを追加する場合、パッケージまたはパッチ関連のシンボリックリンクには、シンボリックリンクのリンク先ではなくシンボリックリンクのソースが使用されていました。したがって、パッケージやパッチパッケージのアップグレードの際にシンボリックリンクのリンク先を別のものに変更する必要があるという問題がありました。この Solaris リリースでは、パッケージのシンボリックリンクのリンク先を変更する必要がある場合、デフォルトで、シンボリックリンクのソースの代わりにリンク先が pkgadd コマンドによって調べられます。

しかしながら、パッケージによっては pkgadd のこの新しい動作に準拠していない場合があります。

pkgadd シンボリックリンクの新旧動作に対応するために PKG_NONABI_SYMLINKS 環境変数が使用できます。この環境変数が真に設定されていると、pkgadd はシンボリックリンクのソースを使用します。

pkgadd コマンドを使ってパッケージを追加する前に管理者がこの変数を設定すれば、新しい動作に対応していないパッケージを以前の動作で処理できます。

pkgadd コマンドを使って既存のパッケージを追加する場合、pkgadd シンボリックリンクの新しい動作が原因でパッケージを追加できないことがあります。その場合には、次のエラーメッセージが表示されます。

```
unable to create symbolic link to <path>
```

この問題のためにパッケージをインストールできない場合は、次の手順に従います。

1. Sun 提供のパッケージの場合は、ご購入先に新動作に対応していないパッケージ名をお知らせください。
2. PKG_NONABI_SYMLINKS 環境変数を設定し、pkgadd コマンドを使ってパッケージを再び追加してください。

```
# PKG_NONABI_SYMLINKS=true
# export PKG_NONABI_SYMLINKS
# pkgadd pkg-name
```

特定のソフトウェア管理エラー

```
WARNING: filename <not present on Read Only file system>
```

エラーの発生原因	解決方法
このエラーメッセージは、パッケージの一部のファイルがインストールできなかったことを示す。このエラーは、通常、pkgadd を使用してパッケージをクライアントにインストールするときに発生する。この場合、pkgadd は、サーバーからマウントしているファイルシステムにパッケージをインストールしようとする。しかし pkgadd は、そのためのアクセス権を持っていない	パッケージのインストール中にこの警告メッセージが表示された場合、パッケージをサーバーにもインストールしなければならない。詳細は、『*Solaris* のシステム管理 *(*第 *1* 巻*)*』の「ソフトウェアの管理 (概要)」を参照

一般的なソフトウェア管理時の問題

エラーの発生原因	解決方法
Solaris 2.5 およびその互換バージョンより前に開発された一部のパッケージの追加と削除に関連して、既知の問題が存在する。このようなパッケージを追加または削除すると、ユーザーとの対話中にインストールが失敗するか、ユーザーとの対話のためにプロンプトが出されるが、ユーザーの応答は無視されることがある	次の環境変数を設定して、パッケージを追加し直す。 `NONABI_SCRIPTS=TRUE`

索引

B

C

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

き

く

け

こ

さ

し

す

せ

そ

た

ち

つ

て

と

へ

ほ

ま

む

め

も

れ

ろ

わ