Asianux Server 3

サーバー構築・運用ガイド

Asianux Server 3 サーバー構築・運用ガイド

目次

第1章　システムの起動と終了

この章で説明する内容

目的	システムの起動と終了のしくみを理解する
機能	root のパスワードを忘れたときの対処方法や、システム起動時に利用可能となるようにサービスの設定を行う
必要な RPM	initscripts――基本システムスクリプト
設定ファイル	/boot/grub/grub.conf /etc/inittab
章の流れ	1 システムの起動と終了 2 ログインとログアウト 3 システムのシャットダウン(停止)とリブート（再起動） 4 ランレベル 5 サービスの起動制御
関連 URL	**Linux と Windows のデュアルブートの設定方法** http://www.linux.or.jp/JF/JFdocs/Linux%2BWin9x%2BGrub-HOWTO/

1.1　システムの起動

Linux OS はマシンの電源を投入すると起動を開始します。ハードウェア（CPU、メモリ、ディスクなど）の自己診断が終了すると、GRUB（GRand Unified Bootloader）が起動して、図 1-1 のようなブート画面が表示されます。

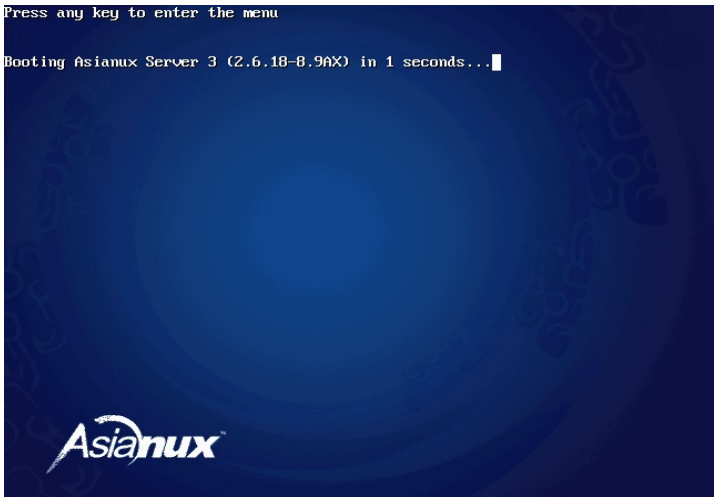


図 1-1　ブート画面

図 1-1 の画面が表示されている間に何れかのキーを押すと図 1-2 のような OS 選択メニューが表示されます。何も押さなければデフォルト設定の OS がそのまま起動されます。

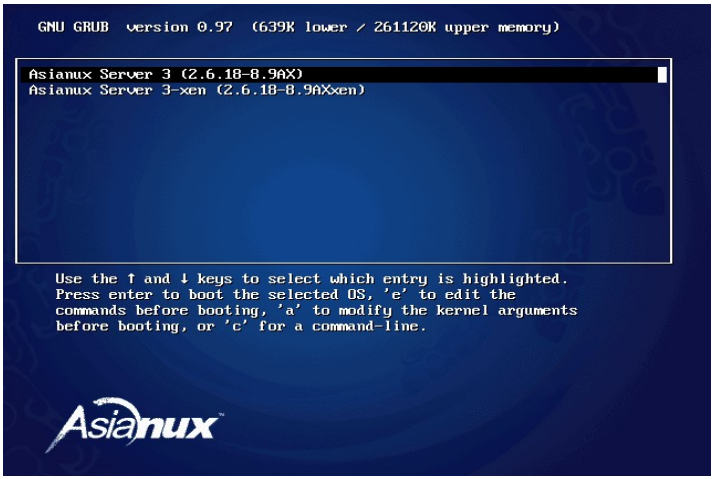


図 1-2　OS 選択メニュー

マシンに Linux だけがインストールされている場合には、選択肢は「Asianux Server 3」（Linux）のみとなります。一方、同じマシンに Linux だけではなく Microsoft 社の Windows XP などがインストールされている場合には、Windows を指す「DOS」という選択肢が増えて、どの OS を起動するかを選択できるようになります（このとき表示される「DOS」や「Asianux Server 3」といったラベルは GRUB をインストールするときに指定できます）。

Linux と Windows との切り替えは矢印キー（[↓]、[↑]）で行い、[Enter] キーを押すと選択した OS が起動されます。キー入力をせずに一定の時間が経過すると、デフォルトで Linux が起動します。

1.2　ログインとログアウト

システムを利用するためには、ログインする必要があります。本節では、GUI モード時および CUI モード時それぞれのログインとログアウトの方法について説明します。

1.2.1　GUI モード時のログイン

GUI ログインの場合、システム起動後は図 1-3 のようなログイン画面が表示されます。この画面で[ユーザ名(U)]にユーザのアカウントを、[パスワード(P)]にユーザのパスワードを入力すると、ログインすることができます。

※ユーザーを作成していない場合は、ユーザー名に root と入力してログインしてください。

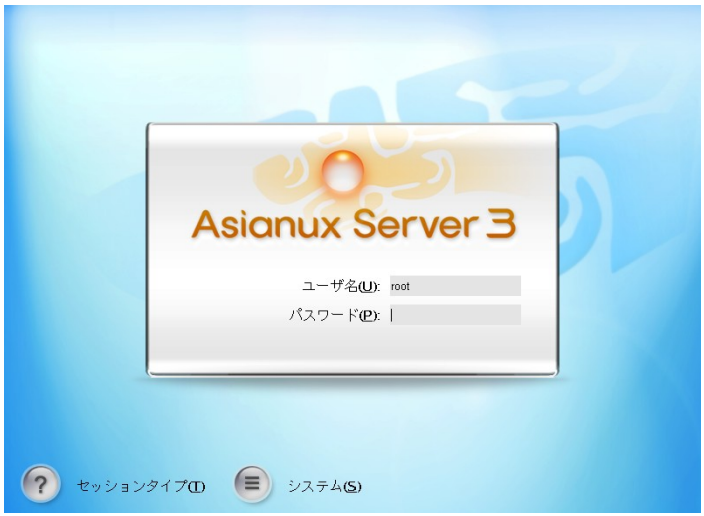


図 1-3　GUI モードのログイン画面

1.2.2　CUI モード時のログイン

CUI ログインの場合、システム起動後は図 1-4 のようなログイン画面が表示されます。この画面で「ホスト名 : login」にユーザのアカウント名を、次にパスワードを入力すると、システムにログインですることができます。

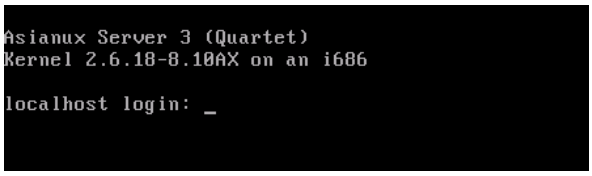


図 1-4　CUI モードのログイン画面

1.2.3 GUI モード時のログアウト

GUI モードでログアウトする場合、画面左下の「Start」ボタンをクリックし、メニューから[ログアウト]を選択します

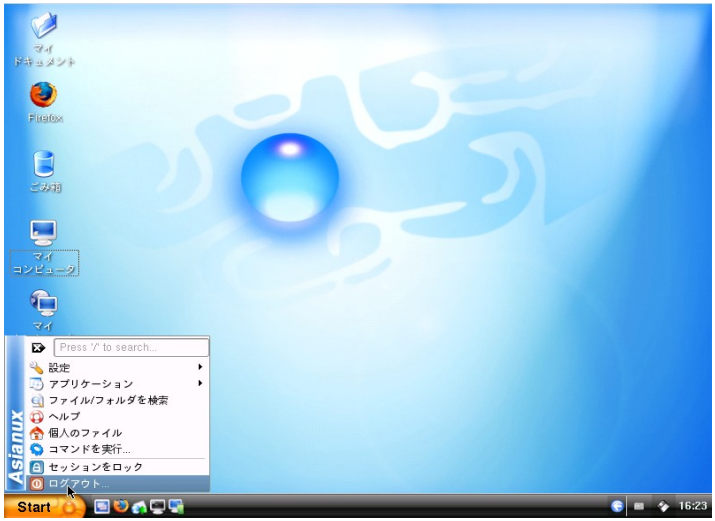


図 1-5 GUI モードのログアウト画面（1）

次に表示される画面で[ログアウト]ボタンをクリックします。

図 1-6 GUI モードのログアウト画面（2）

1.2.4　CUI モード時のログアウト

CUI モードでログアウトする場合、コンソールから **logout** コマンドを入力します。

```
# logout
```

1.2.5　システムのシャットダウン（停止）とリブート（再起動）

システムを停止してマシンの電源を切るためには、シャットダウン（停止）する必要があります。またシステムを再起動するためには、リブート（再起動）する必要があります。本節では、GUI モード時および CUI モード時それぞれのシャットダウンとリブートの方法について説明します。

1.2.6　GUI モード時のシャットダウン/リブート

GUI モードでシャットダウン/リブートする場合、1.2.3 と同様の手順で以下の画面を表示します。

この画面で、シャットダウンする場合は[シャットダウン]ボタンを、再起動する場合は「再起動」ボタンをクリックします。

図 1-7　GUI モードのシャットダウン画面

なお、root ユーザ以外の一般ユーザが、シャットダウンまたは再起動を実行する場合、root 認証が必要です。

root ユーザのパスワードを要求する画面が表示されますので、入力してください。

1.2.7 CUI モード時のシャットダウン

以下のコマンドを root ユーザーで実行します。

```
# /sbin/shutdown -h now
```

1.2.8 CUI モード時のリブート

以下のコマンドを root ユーザーで実行します。

```
# /sbin/shutdown -r now
```

1.3 ランレベル

1.3.1 ランレベルの種類

Linux の実行状態には以下の 7 つの状態（ランレベル）が存在します。

通常はランレベル 3 かランレベル 5 の状態でシステムを運用します。

ランレベル	状態	説明
0	停止	マシンの電源を切って問題ない状態。
1	シングルユーザーモード	ネットワークが使用できず、root がコンソールでのみ使用できる状態。
2	マルチユーザーモード	ネットワークは起動しているが、NFS が使用できない状態。
3	マルチユーザーモード	ネットワークが起動していて、NFS などが使用できる状態。 コンソールはテキストログイン。
4	コンフィギュレーションモード	インストール後の状態。
5	マルチユーザーモード	ネットワークが起動していて、NFS などが使用できる状態。 コンソールはグラフィカルログイン。
6	再起動	システムを再起動する。

1.3.2 ランレベルの変更

システムをどのランレベルで起動するかは**/etc/inittab** で指定します。inittab のエントリ行は、以下のような書式で記述します。

```
id:runlevels:action:process
```

エントリ行の"runlevel"部分に、起動するランレベルを記述します。inittab の詳細については、「**man 5 inittab**」を参照してください。次の例は、ランレベル 3 で起動する場合の記述例です。

```
id:3:initdefault:
```

システム運用中にランレベルを動的に変更するには、**telinit** コマンドを使用します。次の例は、ランレベル 3 に変更する場合の実行例です。

```
# /sbin/telinit 3
```

1.4 サービスの起動制御

サービスの起動を制御する場合、システム起動時には **chkiconfig** コマンドを使用します。また、システム運用中にサービスの起動/停止などの制御を行う場合は、**service** コマンドを使用します。

1.4.1 chkconfig

chkconfig コマンドを使用することで、システム起動時にどのようなサービス(デーモン)を起動するかを指定できます。用途別の chkconfig コマンドの使用方法について以下に示します。

➢ システム起動時にサービスを起動させる

```
# /sbin/chkconfig サービス名 on
```

この書式では、指定したサービスが起動スクリプトで指定したランレベルで起動するようになります。たとえば、システムの起動時に Samba を起動させたい場合には次のコマンドを実行します。

```
# /sbin/chkconfig smb on
```

➢ システム起動時にサービスが起動しないようにする

```
# /sbin/chkconfig サービス名 off
```

たとえば、システム起動時に Apache が起動しないようにするには、次のコマンドを実行します。

```
# /sbin/chkconfig httpd off
```

➢ あるサービスを特定のランレベルで起動させる

```
# /sbin/chkconfig --level=ランレベル サービス名 on
```

たとえば、sendmail をランレベル 2 でも起動させるには、次のコマンドを実行します。

```
# /sbin/chkconfig --level=2 sendmail on
```

➢ サービスの起動設定状態を表示する

```
# /sbin/chkconfig --list
```

1.4.2　service

service コマンドは、システム運用中にサービス制御スクリプトを操作するためのコマンドです。

➢ システム運用中にサービスを起動する

```
# /sbin/service サービス名 start
```

たとえば、システム運用中に httpd を起動させるには、次のコマンドを実行します。

```
# /sbin/service httpd start
```

➢ システム運用中にサービスを停止する

```
# /sbin/service サービス名 stop
```

たとえば、システム運用中に CUPS（Common UNIX Printing System）を停止させるには、次のコマンドを実行します。

```
# /sbin/service cups stop
```

➢ システム運用中にサービスを再起動する

```
# /sbin/service サービス名 restart
```

たとえば、**xinetd.conf** や **xinetd.d/*** を変更した場合に xinetd を再起動させるには、次のコマンドを実行します。

```
# /sbin/service xinetd restart
```

第2章　パッケージ管理

この章で説明する内容

目的	RPM を用いたパッケージ管理について理解する
機能	パッケージに関する情報の閲覧、パッケージの導入、削除、アップグレードを行う
必要な RPM	rpm――パッケージ管理ツール
設定ファイル	/etc/sysconfig/kernel
章の流れ	1 RPM の概要 2 RPM コマンドの使用法 3 kernel パッケージの管理
関連 URL	

2.1 RPM の概要

Asianux Server 3 に含まれる標準的なパッケージは RPM（Red Hat Package Manager）によって管理されます。RPM は Red Hat Linux や Red Flag などの多くのディストリビューションで採用されているパッケージ管理ツールで、次のような機能を提供します。

- **容易なアップグレード**
  パッケージを再インストールすることなく、個別にパッケージを安全にアップグレードできます。アップグレードに必要な他のパッケージがある場合には教えてくれます。
- **高度な問い合わせ機能**
  システム全体からパッケージを検索したり、特定のパッケージ群のみを検索したりできます。あるファイルがどのパッケージによってインストールされたのかを簡単に検索することもできます。
- **パッケージの検証**
  パッケージに関する重要ファイルを削除してしまった可能性がある場合に、パッケージを検証することで矛盾の有無を調べることができます。
- **ソースの利用とパッケージの再構築**
  パッケージのソフトウェアソースをユーザーが利用できます。もし、パッケージがシステムの環境に適合しない場合でも、パッケージを再コンパイルしたり再構築したりすることで、他システムのパッケージを移植することが容易になります。

RPM はシステムに変更を加えるため、RPM パッケージのインストール、削除、アップグレードは root ユーザーとして実行してください。

この章で表記している用語の意味は以下のとおりです。**samba-3.0.24-6AX.i386.rpm** の場合を例にして説明します。

- **パッケージ**名 —— RPM パッケージのバージョンを除いた名前を指します。
  [例] **samba**
- パッケージ**ファ**イル名 —— RPM パッケージ自体のファイル名を指します。
  [例] **samba-3.0.24-6AX.i386.rpm**
- ファイル名 —— RPM パッケージに含まれ、インストールされるファイル名を指します。
  [例] **/usr/sbin/smbd** 他

2.2 RPM の使用法

本節では、パッケージのインストール、アンインストール、アップグレード、問い合わせ、検証の 5 つの基本操作モードについて説明します。パッケージの作成については他の文献を参照してください。詳細な説明およびオプションについては、**rpm --help** または **man rpm** を実行して表示される情報を参照してください。

2.2.1 RPM の検索

RPM を使用する前に、パッケージがどこにあるかを調べる必要があります。ほとんどは製品インストール DVD の Asianux/RPMS ディレクトリの中にあります。また、製品発売後に修正したパッケージは Asianux Technical Support Network で提供いたします。

2.2.2 RPM の問い合わせ（-q）

rpm コマンドに**-q** オプションを付けて実行することで、パッケージの問い合わせができます。

RPM パッケージには通常 **samba-3.0.24-6AX.i386.rpm** というようなファイル名が付けられています。このファイル名は、パッケージ名（samba）、バージョン（3.0.24）、リリース（6AX）、アーキテクチャ（i386）で構成されています。rpm コマンドを使用するときは、引数に十分注意してください。

- **インストール済みの 1 つのパッケージを表示する**

```
# /bin/rpm -q パッケージ名
```

- **インストール済みのすべてのパッケージを表示する**

```
# /bin/rpm -qa
```

- **パッケージの名前、説明、リリースなどのパッケージ情報を表示する**

```
# /bin/rpm -qi インストール済みパッケージ名
```

- **指定したファイルが含まれるパッケージについて問い合わせる**

```
# /bin/rpm -qf ファイル名
```

ファイルを指定するときは、そのファイルの絶対パスを指定する必要があります。
たとえば、**/etc/samba/smb.conf** がどのパッケージに含まれるかを調べたい場合は、次のようにします。

```
# /bin/rpm -qf /etc/samba/smb.conf
```

- **パッケージファイルについて問い合わせる**

-p を指定すると、まだインストールされていないパッケージファイルについても問い合わせができます。

```
# /bin/rpm -qip パッケージファイル名
```

- **パッケージに含まれるファイルの一覧を表示**

```
# /bin/rpm -ql インストール済みパッケージ名
# /bin/rpm -qlp パッケージファイル名
```

2.2.3　インストール（-i）

インストールされていないパッケージをインストールします（すでにインストールされているとエラーになります）。

```
# /bin/rpm -ivh パッケージファイル名
パッケージ名 ##################################
```

RPM はパッケージ名を出力して、パッケージのインストール状況をシャープ記号（#）を使って表示します。

2.2.4　アンインストール（-e）

インストールされているパッケージを削除します。引数にはパッケージのファイル名ではなく、パッケージの名前を指定することに注意してください。

```
# /bin/rpm -e パッケージ名
```

2.2.5 アップグレード（-U）

インストールされていないものは新規インストールされ、古いパッケージがインストールされているとバージョンアップされます。

```
# /bin/rpm -Uvh パッケージファイル名
```

古いバージョンのパッケージは自動的にアンインストールされます。 アップグレードで以下のようなメッセージが表示されることがあります。

```
警告: /etc/sample.conf は/etc/sample.conf.rpmsave として保存されます
```

このメッセージは、既存の設定ファイルが新しいファイルによって置き換えられたことを意味します。2 つのファイルの違いを調査することで、引き続きシステムが正しく動作することを確認する必要があります。

2.2.6 アップグレード（-F）

インストールされているものだけを最新の状態にします。**-U** と違い、パッケージがインストールされていなければ何もしません。

```
# /bin/rpm -Fvh パッケージファイル名
```

多数のパッケージの中からシステムにインストール済みのパッケージのみをアップグレードする場合には、次のコマンドを使用します。

```
# /bin/rpm -Fvh *.rpm
```

2.2.7 検証（-V）

- **パッケージに含まれるすべてのファイルがインストール時と同じ状態かどうかを検証する**

```
# /bin/rpm -V パッケージ名
```

- **特定のファイルを含むパッケージを検証する**

```
# /bin/rpm -Vf ファイル名
```

このとき、ファイル名は、**/bin/vi** のようにフルパスで記述してください。

- **インストール済みのすべてのパッケージについて検証を実行する**

```
# /bin/rpm -Va
```

- **インストール済みパッケージと RPM パッケージファイルとを照合する**

```
# /bin/rpm -Vp パッケージファイル名
foo-1.0-1.i386.rpm
```

RPM データベースが破損した疑いがある場合に、このコマンドが役に立ちます。すべてが正常に検証された場合は何も出力されません。何らかの矛盾が見つかった場合はその内容が表示されます。

2.2.8　エラー時の例外処理

パッケージの操作でエラーが出た場合は、原因を調査し問題を解決してください。しかし、原因が明らかな場合や、開発者やサポートから例外的な操作を指示されている場合にのみ、次のようにして回避できます。

- **RPM からインストールされたファイルが誤って削除されてしまった場合**

  必要なファイルが削除されてしまった場合や、RPM からオリジナルの設定ファイルをインストールしたい場合には、**--replacepkgs** オプションを使用すると、インストール済みのものと同じバージョンであってもエラーを無視して再インストールできます。

```
# /bin/rpm -ivh --replacepkgs パッケージファイル名
```

- **ファイルの競合が発生した場合**

  別のパッケージや同じパッケージの古いバージョンによってインストールされたファイルと新しいファイルが競合する場合には、次のメッセージが表示されます。

```
依存性の欠如: xxx と競合します
```

RPM にこのエラーを無視するよう指示するには、**--replacefiles** オプションを使用します。

```
# /bin/rpm -ivh --replacefiles パッケージファイル名
```

- **バージョンの古いパッケージをインストール**

  新しいパッケージをインストールした結果不具合が発生したなどの理由により古いパッケージに戻す場合には、**--oldpackage** オプションを使用します。

```
# /bin/rpm -Uvh --oldpackage パッケージファイル名
```

2.3 kernel パッケージの管理

kernel パッケージは、Linux OS の中枢をなすパッケージです。kernel パッケージは、通常のパッケージとは異なり、新旧パッケージの共存が可能です。kernel は、次のパッケージで構成されています。

➢ x86 用

- **kernel**
  通常のカーネルパッケージ。最大 4GB までメモリを利用できます。
- **kernel-PAE**
  最大 64GB までメモリを利用できるカーネルパッケージ。
- **kernel-PAE-devel**
  kernel-PAE 用のカーネルモジュールをビルドするために必要なヘッダを含みます。

➢ x86-64 用

- **kernel**
  通常のカーネルパッケージ。

➢ 共通

- **kernel-xen**
  仮想化対応したカーネルパッケージです。
- **kernel-xen-devel**
  kernel-xen 用のカーネルモジュールをビルドするために必要なヘッダを含みます。
- **kernel-doc**
  カーネル関連のドキュメントが入っています。
- **kernel-devel**
  kernel 用のカーネルモジュールをビルドするために必要なヘッダを含みます。

共存可能なパッケージは、**kernel**、**kernel-PAE**、**kernel-xen**、**kernel-devel**、**kernel-PAE-devel**、**kernel-xen-devel** の 6 種類です。doc パッケージの共存はできません。

kernel パッケージを共存させるには、インストールオプション**-ivh** を使います。

```
# /bin/rpm -ivh kernel-2.6.18-8.9AX.i686.rpm
# /bin/rpm -qa | /bin/grep kernel
kernel-2.6.18-8.7AX
kernel-2.6.18-8.9AX
```

kernel パッケージのインストール完了後に、ブートローダの GRUB(**/boot/grub/grub.conf**)には、新しくインストールされた kernel のエントリが自動的に追加されます。

- kernel パッケージインストール前の**/boot/grub/grub.conf** ファイルの内容例

```
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title Asianux Server 3 (2.6.18-8.7AX)
        root (hd0,0)
        kernel /vmlinuz-2.6.18-8.7AX ro root=LABEL=/
        initrd /initrd-2.6.18-8.7AX.img
```

- kernel パッケージインストール後の**/boot/grub/grub.conf** ファイルの内容例

```
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title Asianux Server 3 (2.6.18-8.9AX)
        root (hd0,0)
        kernel /vmlinuz-2.6.18-8.9AX ro root=LABEL=/
        initrd /initrd-2.6.18-8.9AX.img
title Asianux Server 3 (2.6.18-8.7AX)
        root (hd0,0)
        kernel /vmlinuz-2.6.18-8.7AX ro root=LABEL=/
        initrd /initrd-2.6.18-8.7AX.img
```

➢ 注意事項

kernel パッケージを追加インストールすると、デフォルトで起動されるカーネルとして設定されます。

kernel パッケージを追加インストールしても、デフォルトのカーネルが変更されないようにするためには、**/etc/sysconfig/kernel** の"UPDATEDEFAULT"パラメータを **no** に変更してください。

また、デフォルトで起動されるカーネルを変更する場合は、**/boot/grub/grub.conf** の"**default**"パラメータに、適切なエントリ番号(エントリ番号は 0 から開始)を設定してください。

第3章　ユーザー／グループ管理

この章で説明する内容

目的	システムを使用するユーザーとグループの管理について理解する
機能	ユーザーの作成、変更、削除　　パスワードの設定 グループの作成、変更、削除
必要な RPM	shadow-utils――ユーザー／グループアカウントおよび shadow パスワードの管理ユーティリティ passwd――パスワードユーティリティ coreutils――GNU シェルユーティリティ kdeadmin-kuser――KDE ユーザマネージャ
設定ファイル	/etc/passwd /etc/group /etc/shadow
章の流れ	1 ユーザー／グループ管理の概要 2 ユーザーの作成、変更、削除 3 パスワードの変更 4 グループの作成、変更、削除 5 ログインユーザーの変更 6 KUser を用いたユーザ／グループ管理
関連 URL	**The Linux Japanese FAQ Project** http://www.linux.or.jp/JF/index.html

3.1 ユーザー／グループ管理の概要

Linux などの UNIX 系 OS では、ユーザーはまずシステムにログインすることから作業を始めます。これにより、システム管理者（root）は、認証したユーザーのみにシステムの使用権限を与えることができます。また各ユーザーは、専用 ID とパスワードを使用して、他のユーザーからの不正なアクセスを受けることなく、システム内の自身のリソースを使用できます。

ユーザーは少なくとも 1 つの主グループに所属し、同一グループのユーザーは、そのグループ内でリソースを共有することができます。また、ユーザーは、複数の補助グループに所属することも可能です。

ログイン後、ユーザが作成したファイルは主グループがセットされますが、リソースへのアクセス等については、主グループも補助グループも同様に共有することができます。

3.2 ユーザーの作成、変更、削除

- **新規ユーザーを作成する**

  新規ユーザーの作成は、root ユーザーが useradd を使用して行います。たとえばユーザー foo を作成するには、次のコマンドを実行します。

```
# /usr/sbin/useradd foo
```

  グループを指定せずに新規ユーザを作成した場合、ユーザ名と同名のグループが一緒に作成されます。ここで作成されたユーザ／グループは、それぞれ**/etc/passwd** と**/etc/group** に保存されます。確認するには、次のコマンドを実行します。

```
# grep foo /etc/passwd
foo:x:512:512::/home/foo:/bin/bash

# grep foo /etc/group
foo:x:512:
```

- **グループを指定して新規ユーザを作成する**

  useradd に**-g** をオプションを指定して、次のコマンドを実行します。ここでは users グループを指定して、bar ユーザを作成しています。

```
# /usr/sbin/useradd -g users bar
```

- **複数グループを指定して新規ユーザを作成する**

  useradd に**-g**/**-G**をオプションを指定して、次のコマンドを実行します。ここでは foo ユーザーの主グループとして users グループ、補助グループとして wheel, ftp, apache グループを指定しています。**-g** を指定しなければ、ユーザ名と同名のグループが主グループとしてセットされます。

```
# /usr/sbin/useradd -g users -G wheel,ftp,apache foo
```

- **作成したユーザのユーザ名を変更する**

  以下の例では、foo のユーザ名を hoge に変更しています。

```
# /usr/sbin/usermod -l hoge foo
```

- **作成したユーザのユーザ ID を変更する**

  以下の例では、foo のユーザー ID を 1000 に変更しています。

```
# /usr/sbin/usermod -u 1000 foo
```

- **ユーザー ID やグループ ID を確認する**

```
# /usr/bin/id foo
uid=1000(foo) gid=100(users) 所属グループ=100(users),10(wheel),50(ftp),48(apache)
```

- **既存のユーザーを削除する**

  root ユーザーが **userdel** を使用して行います。たとえばユーザー foo を削除するには、次のコマンドを実行します。

```
# /usr/sbin/userdel foo
```

  既存ユーザーの削除を行う際、ユーザのホームディレクトリも同時に削除したい場合、**userdel** に**-r** オプションを指定して、次のコマンドを実行します。

```
# /usr/sbin/userdel -r foo
```

3.3 パスワードの変更

ユーザーが自分のパスワードを変更するには、次のコマンドを実行します。

```
$ /usr/bin/passwd
```

すると、現在のパスワード、新しいパスワード、確認用に再度新しいパスワードと、合計 3 回の入力が要求されます。

パスワードには、辞書にある英単語など、他人が予想しやすいものは設定できません。たとえば、「password」という単語を新しいパスワードとして入力した場合には次のようなエラーとなります。

```
BAD PASSWORD: it is based on a dictionary word
```

また、パスワードの文字数が 6 文字に満たないときには、次のようなエラーとなります。

```
BAD PASSWORD: it is WAY too short
```

root ユーザーは、次のコマンドを実行して、他の一般ユーザーのパスワードを変更できます。

```
# /usr/bin/passwd ユーザー名
```

3.4 グループの作成、変更、削除

- **新規グループを作成する**

  新規グループの作成は、root ユーザーが **groupadd** を使用して行います。たとえば asianux というグループを作成するには、次のコマンドを実行します。

```
# /usr/sbin/groupadd asianux
```

作成したグループを確認するには、次のコマンドを実行します。

```
# grep asianux /etc/group
asianux:x:513:
```

- **作成したグループのグループ名を変更する**

```
# /usr/sbin/groupmod -n miraclelinux asianux
```

- **作成したグループのグループ ID を変更する**

```
# /usr/sbin/groupmod -g 1000 asianux
```

- **既存グループを削除する**

  グループの削除は、root ユーザーが **groupdel** を使用して行います。たとえば asianux というグループを削除するには、次のコマンドを実行します。

```
# /usr/sbin/groupdel asianux
```

3.5 ログインユーザーの変更

ログイン中に他のユーザーに代わりたいときには、**su** コマンドを使用します。たとえば root でログイン中にユーザー foo で作業をしたいときには、次のコマンドを実行します。

```
# /bin/su - foo
```

id コマンドを使用すると、以下のように現在の自分のログイン ID とグループ ID を確認することができます。

```
# /usr/bin/id
uid=0(root) gid=0(root) 所属グループ
=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel)
# /bin/su - foo
# /usr/bin/id
uid=500(foo) gid=500(asianux) 所属グループ=500(asianux)
```

su に「**-**」(ハイフン) を付けると、切り替え後のユーザーに直接ログインした場合と同じ環境になりますが、付けない場合は切り替え前のユーザー環境を引き継ぎます。

一般ユーザーが他のユーザーに変わるときには、su 実行時にパスワードの入力を要求されるので、不正に他のユーザーになりすますことはできません。

3.6　KUserを用いたユーザ／グループ管理

KUserとは、コマンドやコマンドオプションを入力することなく、GUI でユーザ／グループ管理を簡単に行うことができるユーティリティです。

KUser を起動するには、GUI デスクトップ画面左下の「Start」ボタンをクリックし、メニューから[アプリケーション][ユーザマネージャ]を選択するか、または次のコマンドを実行します。

```
# /usr/bin/kuser
```

KUser が起動すると、以下のような画面が表示されます。

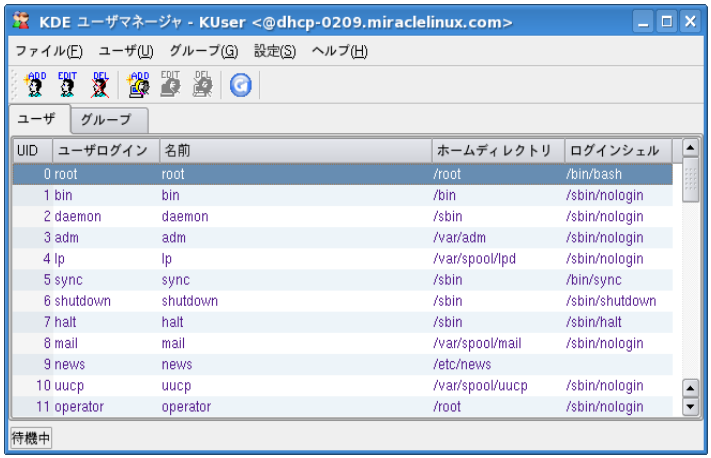


図 3-1 KUser 起動画面

ユーザの追加、変更、削除は、[ユーザ]メニューから選択するか、またはアイコンをクリックすることで操作できます。グループの追加、変更、削除も同様にタブを切り替えることで一覧表示し、操作できます。

[設定]メニューから[KUser を設定]を選択すると、図 3-2 の画面が表示されます。

ユーザ／グループ情報を、ローカルホスト上か NIS（図 3-3）、または LDAP のいずれかで管理する設定を行うことができます。KUser では、ユーザ ID／グループ ID はデフォルトで 500 から ID 番号が割り当てられます。

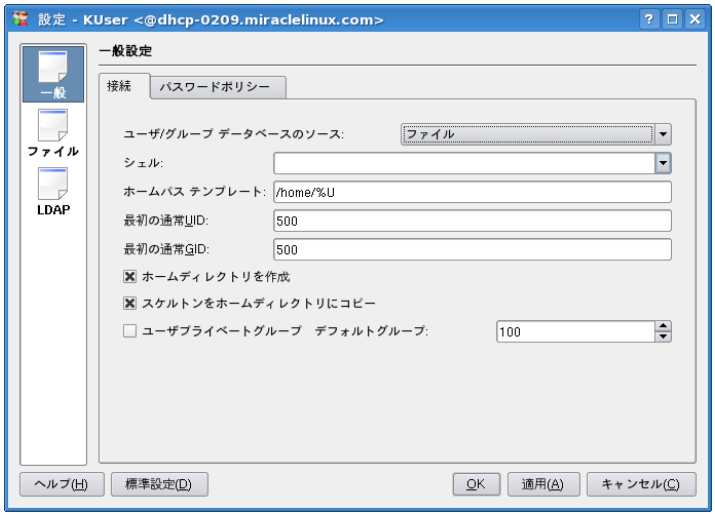


図 3-2 KUser の設定画面（一般）

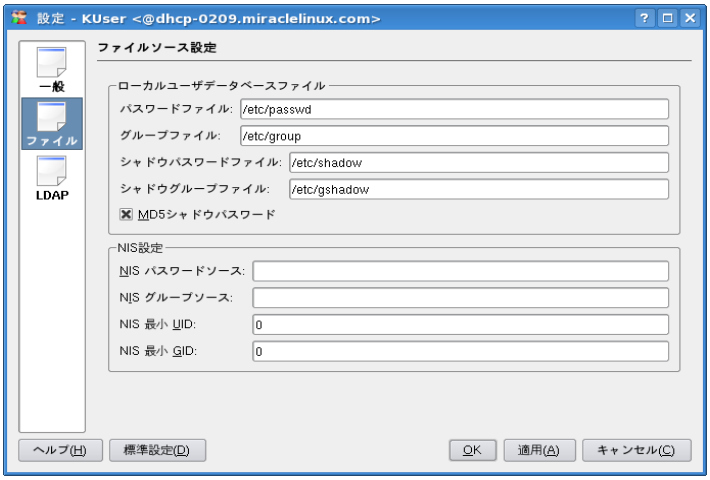


図 3-3 KUser の設定画面（ファイル）

3.6.1　LDAP/Samba 連携におけるユーザ／グループ管理

KUser を用いて、LDAP/Samba ユーザ／グループを管理することができます。LDAP/Samba サーバは、**smbdcsetup** ツールを用いて構築します。smbdcsetup ツールの詳細については、本書「10.8 ドメインコントローラーの構築」を参照してください。

ここでは、LDAP/Samba サーバ構築後に、KUser を使用してユーザ／グループ管理の設定する方法について説明します。

KUser を起動して、[設定]メニューから[KUser を設定]を選択し、「一般設定」の「ユーザ／グループ データベースのソース」設定を LDAP へ変更します（図 3-4）。

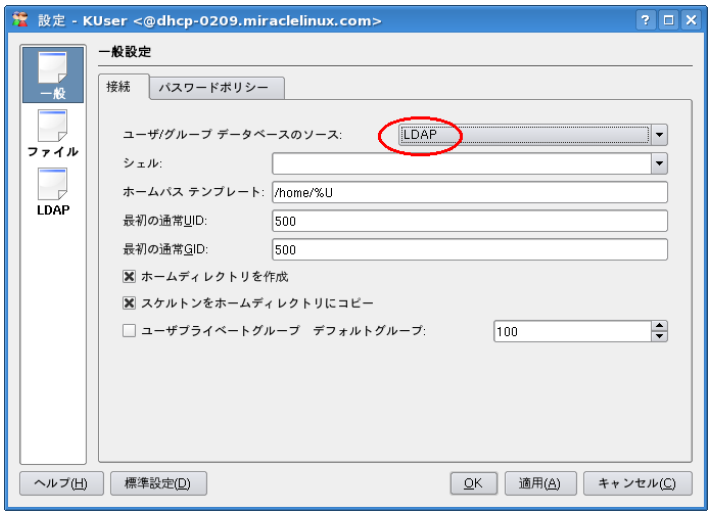


図 3-4 KUser の設定画面（一般[LDAP]）

smbdcsetup ツールで設定した内容を、LDAP 設定項目へ入力します。ここでは例として、（図 3-5～図 3-7）の内容を設定します。

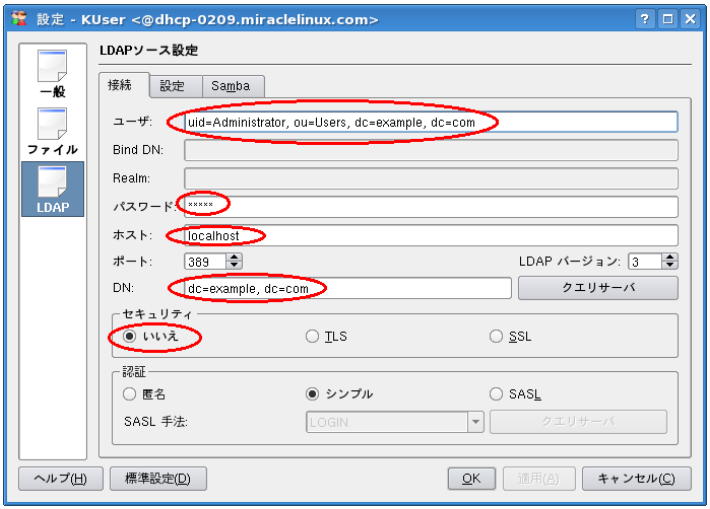


図 3-5 KUser の設定画面（LDAP[接続]）

- ユーザ : uid=Administrator, ou=Users, dc=example, dc=com
- パスワード : ********
- ホスト : localhost
- DN : dc=example, dc=com
- セキュリティ : <いいえ>を選択

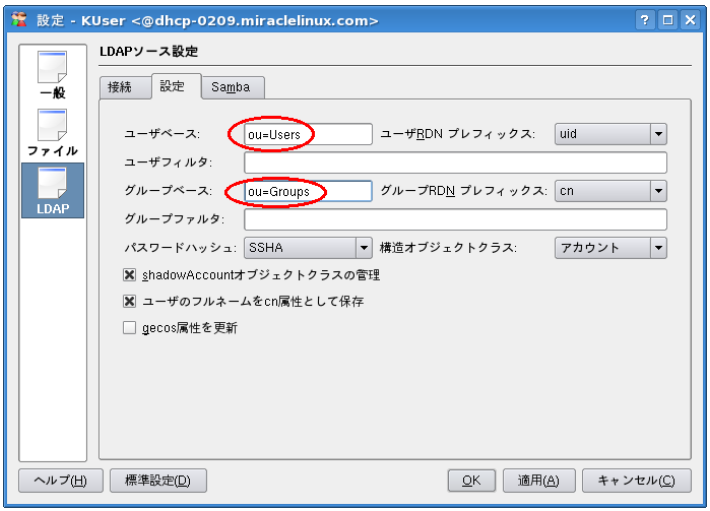


図 3-6 KUser の設定（LDAP[設定]）

- ユーザベース : ou=Users
- グループベース : ou=Groups

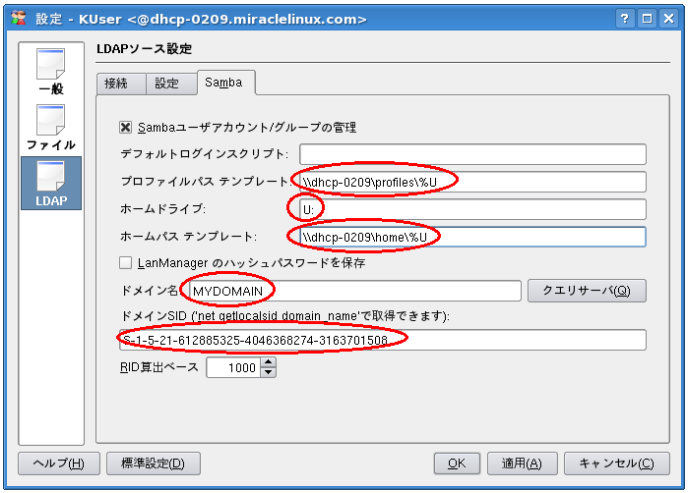


図 3-7 KUser の設定（LDAP[Samba]）

- プロファイルパステンプレート : \ \dhcp-0209\profiles\%U
- ホームドライブ : U:
- ホームパステンプレート : \\dhcp-0209\ home\%U
- ドメイン名 : MYDOMAIN
- ドメイン SID :下記参照

Samba のドメイン SID を調べるには、以下のコマンドを実行します。

```
# /usr/bin/ldapsearch -x -LLL -b "dc=example, dc=com" "objectClass=sambaDomain"
dn: sambaDomainName=MYDOMAIN,dc=example,dc=com
objectClass: sambaDomain
objectClass: sambaUnixIdPool
sambaDomainName: MYDOMAIN
sambaSID: S-1-5-21-612885325-4046368274-3163701508
uidNumber: 1000
gidNumber: 1000
sambaPwdHistoryLength: 0
sambaMaxPwdAge: -1
```

LDAP サーバへの接続に成功すると smbdcsetup ツールで設定したドメイン情報が取得されます（図 3-8、図 3-9）。また、KUser でユーザ／グループを新規追加する際、デフォルト設定の場合、ユーザ ID／グループ ID は 500 から ID 番号が割り当てられます。

LDAP/Samba ユーザ／グループを管理するコマンドラインツールとして smbldap-tools がありますが、それらは 1000 から ID 番号が割り当てられます。KUser と smbldap-tools を併用される場合、スタートする ID 番号の違いにご注意ください。

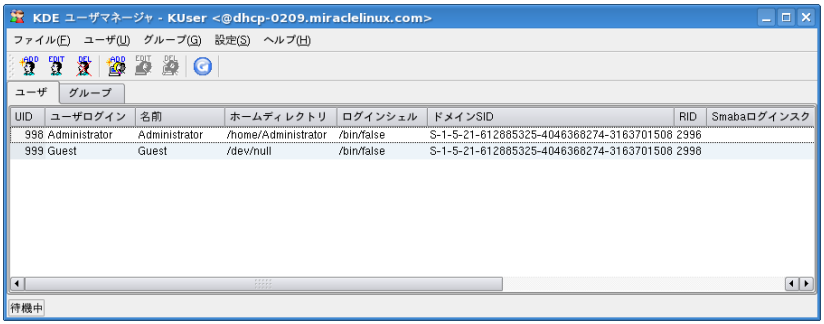


図 3-8 KUser のユーザ管理画面（LDAP）

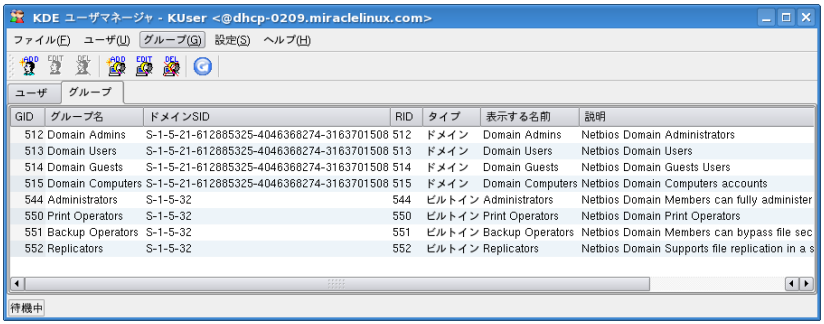


図 3-9 KUser のグループ管理画面（LDAP）

第4章　ディスク管理

この章で説明する内容

目的	ハードディスクの領域の管理方法について理解する	
機能	ディスクパーティションの操作 EXT3 ファイルシステムの利用 RAW デバイスの利用	ソフトウェア RAID の利用 LVM の利用 quota の設定
必要な RPM	coreutils util-linux mount e2fsprogs	mdadm lvm quota
設定ファイル	/etc/fstab /etc/mdadm.conf /etc/lvmtab	
章の流れ	1 ディスク管理の概要 2 パーティションの操作 3 ファイルシステム（ext3） 4 RAW デバイス 5 ソフトウェア RAID 6 LVM について 7 quota の設定	
関連 URL	http://www.linux.or.jp/JF/JFdocs/INDEX-diskmanage.html	

4.1　ディスク管理の概要

ハードディスクや SAN（Storage Area Network）などの、ストレージの領域管理は、Linux サーバー管理者の最も重要な仕事の 1 つです。ここでは、最も基本的なディスクの管理手順に関して説明します。Linux サーバーに新しくディスクを増設して、実際に使用するためには、一般的に次のような作業手順となります。

（1）ハードディスクの物理的な接続
（2）パーティションの作成――fdisk
（3）ファイルシステムの作成――mkfs
（4）マウント――mount

最初に行う作業は、物理的にハードディスクをサーバーに接続する作業です。この作業は、各ハードウェアの取り扱い説明書に従って作業してください。

4.1.1　デバイスファイル

Linux で、ハードディスクや、CD-ROM、フロッピーディスクなどのデバイスを指定するときには、デバイスファイルを指定します。デバイスファイルは、デバイスを抽象化してファイルとして表現したものです。通常のファイルはデータを格納するために利用されますが、デバイスファイルは各種デバイスにアクセスするために利用されます。標準的なデバイスファイルは、OS のインストール時に **/dev** ディレクトリ配下に作成されます。**/dev** ディレクトリ配下には、ディスクデバイス以外のデバイス用のデバイスファイルも作成されています。

デバイスファイルは、major 番号と minor 番号を持っており、OS はこの番号を使ってアクセス対象のデバイスを特定します。major／minor 番号はデバイスドライバによってデバイスごとに決められています。デバイスファイルに関連付けられた major／minor 番号は、ls コマンドを使って確認できます。また、デバイスファイルはデバイスの種類によって 2 種類に分かれていて、ディスクのようにブロック単位でアクセスして、ランダムアクセス可能な**ブロックデバイス**と、端末のようにキャラクタ単位でアクセスする**キャラクタデバイス**があります。

```
# /bin/ls -l /dev/sda
brw-rw----    1 root     disk       8,   0  3月 19 19:09 /dev/sda
↑                                    ↑    ↑
block/characterデバイスの種類        major  minor
```

一般的なディスク装置のデバイスファイルとして、次のものが頻繁に利用されます。

➢ SCSI デバイス

- **/dev/sda**、**/dev/sdb**、**/dev/sdc** など

  SCSI コントローラや、SCSI RAID コントローラに接続された SCSI ディスクデバイスを表します。また、Fibre Channel に接続されたストレージ装置のディスクや、USB 接続のディスク装置や、SATA ディスクデバイスなども、この形式で表されます。1 つのディスクが**/dev/sda** といった形式で表され、そのディスク内のパーティションは、パーティション番号に従って、**/dev/sda1**、**/dev/sda2** といった形式で表されます。

  SCSI デバイスのデバイスファイルの割り当ては、システム起動時にロードされる SCSI デバイス用のドライバが、ロードされる順番に基づいて SCSI デバイスを探索して、ディスクを発見した順番で決まります。同一の SCSI チャンネルに接続された SCSI ディスクの場合、SCSI ID の小さなものから探索が行われます。そして、最初に発見したディスク装置が**/dev/sda**、次に発見したディスク装置が**/dev/sdb** というように割り当てられます。したがって、新規に SCSI コントローラや、SCSI デバイスを追加した場合、デバイスファイルの割り当て順が変更される可能性があることに注意が必要です。
- **/dev/scd0**

  最近では少なくなりましたが、SCSI 接続の CD-ROM ドライブを利用する場合に使用します。

➢ IDE デバイス

- **/dev/hda**、**/dev/hdb**、**/dev/hdc**、**/dev/hdd**

  IDE デバイスは SCSI デバイスと異なり、接続されているチャンネルとモードによって、デバイスファイルが決まります。通常、PC/AT 互換機にはプライマリとセカンダリの 2 つのチャンネルがあり、さらにそれぞれのチャンネルごとにマスター、スレーブの 2 台の装置を接続できます。

  - **/dev/hda** —— プライマリ IDE のマスター
  - **/dev/hdb** —— プライマリ IDE のスレーブ
  - **/dev/hdc** —— セカンダリ IDE のマスター
  - **/dev/hdd** —— セカンダリ IDE のスレーブ

  各ディスク上のパーティションを表す場合には、SCSI デバイスの場合と同様に、パーティション番号に基づいて**/dev/hda1**、**/dev/hda2** というように指定します。

  IDE 接続の CD-ROM も、同じルールに基づいてデバイスファイルが割り当てられます。ただし、通常は OS のインストール時に**/dev/cdrom** のシンボリックリンクファイルが実際のデバイスファイルを示すように作成

されているため、CD-ROM を指定する場合には、デバイスファイルとして**/dev/cdrom** を指定することが一般的です。

➢ フロッピーデバイス

- **/dev/fd0**

  一般的なフロッピーデバイスを利用する場合には、**/dev/fd0** を指定します。**/dev/fd0** は一般的な 1.44MB フォーマットのフロッピーのために利用されますが、特殊な用途向けに**/dev/fd0h1660** などのデバイスファイルも用意されています。

通常、デバイスファイルを新たに作成することはあまりありませんが、ストレージデバイスなどで大量にディスク装置を追加した場合や、特殊なハードウェアのために、デバイスファイルが新たに必要になった場合には、**mknod** コマンドで、デバイスファイルを作成します。なお、下記のコマンドの「b」は、ブロックデバイスを意味します。キャラクタデバイスの場合は「c」を指定します。

```
# /bin/mknod /dev/newdev b [major番号] [minor番号]
```

4.2　パーティション

1 つのハードディスク上で、論理的に分割された各領域のことを**パーティション**と呼びます。個々のパーティションは、それぞれ 1 つのハードディスクのように利用できます。パーティションはディスクの管理を容易にしたり、1 台のコンピュータを複数の OS を切り替えながら使用したりするために作成されます。

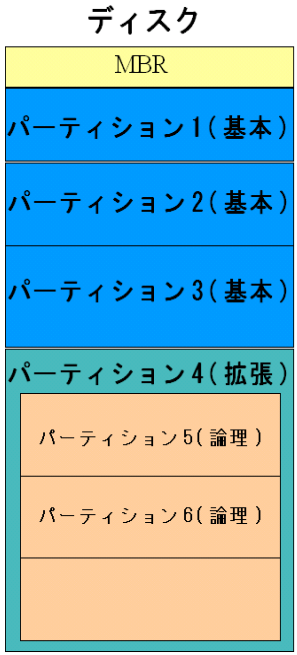
PC/AT 互換機では、1 つのハードディスクを最大 4 つのパーティションに分割できます。これらのパーティション情報は **MBR**（Master Boot Record:ディスクの一番先頭のセクタ）中のパーティションテーブルに格納されます。

このパーティションテーブルに登録されているパーティションを「基本パーティション」と呼びます。4 つ以上のパーティションが必要な場合は、この 4 つの基本パーティションのうち、1 つを「拡張パーティション」にすることができます。拡張パーティションの中には、複数の「論理パーティション」を作成でき、パーティションの合計最大数は、IDE ディスクの場合は 63 個、SCSI ディスクの場合は 15 個となります。ただし、ディスクアレイを使用する場合には、作成できるパーティションの数が制限されることがあります。詳細については、ディスクアレイコントローラの各メーカーに問い合わせてください。

DOS や Windows 系のシステムでは、「MS-DOS 領域」や「論理 MS-DOS ドライブ」などの言葉を使用しますが、これらとパーティションは次のように対応すると考えていいでしょう。

- **基本 MS-DOS 領域** —— 基本パーティション
- **拡張 MS-DOS 領域** —— 拡張パーティション
- **論理 MS-DOS ドライブ** —— 論理パーティション

パーティションの作成例を右に示します。



4.2.1 パーティション分割のメリット

1 つのディスクを単一のパーティションではなく、わざわざ複数のパーティションに分割することにはどのような利点があるのでしょうか。ここでは、Linux では一般的に行われているパーティション分割に対するメリットについて説明します。

- **ファイルシステム障害の局所化**
  システム運用中に不意にシステムのトラブルに遭遇することは珍しいことではありません。原因はさまざまですが、システムのトラブルによってファイルシステムの一部が破壊されることがあります。また、ディスクの不調や故障により特定のブロックが読めなくなる場合もあります。このような場合に備えて、ディスクを複数のパーティションに分割することで、障害発生時の被害を特定のパーティションだけに抑えられる場合があります。

- **ディスク容量不足によるトラブルの防止**
  パーティションを分割していないシステムで、あるユーザーが自分のホームディレクトリに、巨大なファイル(たとえば CD イメージなど)を複数個置いたとします。それが原因で、ファイルシステムの空き領域がなくなり、新しいファイルを作成できない状況が発生したとしましょう。

システムプログラムには、**/etc** や **/var** 配下のファイルを修正したり、**/tmp** などに一時ファイルを作成したりするものがあるため、ファイルシステムに空き領域がなくなると、システムの運用に支障をきたすことがあります。もし仮に **/home** を独立したパーティションに割り当てていたら、その被害は **/home** だけにとどまることになるため、システム全体に影響を与えずに済みます。つまり、ディスクを適切なパーティションに分割することは、ファイルシステムの空き領域がなくなった場合に発生するシステムの異常を最小限に抑えられるメリットがあります。

- **性能劣化の防止**

  システムが使用するディレクトリの中には、そのディレクトリ内にあるファイルの生存期間（ファイルが作成されてから削除されるまでの期間）に特徴を持つものがあります。たとえば、**/var** は数多くのファイルが作成・削除・修正される場所なので、生存期間が短いファイルが集まっているディレクトリだと言えます。**/usr** の場合は逆に、一度アプリケーションをインストールすれば、そのアプリケーションをアップデートするまでの比較的長い間関連ファイルが存在する（大半のプログラムはアップデートされずインストールしたときのままの状態で存在する）ので、ファイルの生存期間が長いといえるでしょう。

  Linux で従来から使用されてきたファイルシステム ext2 は、ファイルに対して連続したブロックを割り当てることで、ファイルアクセス性能を向上させます。しかし、ファイルの作成や削除が頻繁に起こるような状況で長期間運用を続ける場合、フラグメンテーションと呼ばれる領域の「虫食い状態」が発生して、連続したブロックを割り当てられなくなり、性能の劣化が発生します。

  よって、性能を重要視するシステムでは、ファイルの生存期間を考慮してパーティション分割を行うことが推奨されます。たとえば、ファイルの生存期間が異なる **/var** と **/usr** は、それぞれ独立したパーティションに分割するのがいいでしょう。

- **複数 OS の共存**

  パーティションを分割することによって、1 台のハードディスク上に複数の OS（たとえば、Linux と Windows）を共存させることができます。1 つの OS には、最低 1 つのパーティションを割り当てる必要があります。

  Asianux Server 3 の主な用途はサーバーシステムのため、複数の OS を共存させて運用することは推奨していません。しかし、Asianux Server 3 を試験的にインストールする場合などには、別のサーバーを準備する必要がなくなるので有用です。

パーティション分割に関する詳細な説明は、JF のウェブサイトなどを参考にしてください。

http://www.linux.or.jp/JF/JFdocs/INDEX-diskmanage.html

4.2.2 パーティション分割候補のディレクトリと分割例

Linuxをサーバーとして運用する場合、どのようにパーティションを分割するのがいいのでしょうか。また、そのサイズはどれだけ確保すればいいのでしょうか。一般的には、次に示すようなディレクトリがパーティション候補として挙げられます。

- **swap**
- **/boot**
- **/（ルート）**
- **/usr**
- **/home**
- **/var**
- **/tmp**
- **/opt**

Linuxで一般的に利用されるディレクトリは、FHS（Filesystem Hierarchy Standard）[1]によって定義されています。ここでは、パーティション候補として挙げた各ディレクトリの役割について説明します。

※/etc、/lib、/bin、/sbin、は/(root)パーティションに含めてください。/usrも可能であれば/(root)パーティションに含めてください。

- **swap**

  Linuxのスワップパーティションです。Linuxのメモリ管理システムは、ページと呼ばれる単位（Intel 386系CPUの場合、1ページは通常4KB）で、メモリを管理しています。システムに搭載しているメモリよりも多くのメモリが必要になる場合、参照頻度の低いページをスワップパーティションに移動します。よって、システムに搭載しているメモリのサイズと、スワップパーティションとして用意したサイズの合計が、仮想記憶領域（OSが利用できるメモリ領域）のサイズになります。Linuxでは、ファイルの生存期間の観点から、通常スワップパーティションは他のファイルシステムとは別のパーティションに確保します。実際にスワップパーティションがどれぐらい必要になるかは、システム設計の範疇に含まれます。運用するシステムの高負荷時に必要な仮想記憶領域のサイズを想定し、メモリサイズとスワップサイズの合計がその範囲より大きければ問題ないでしょう。

- **/boot**

  Linuxの起動に必要なファイルがこのディレクトリ配下に存在します。したがって、ブートローダからこの配下のファイルが確実に読めなければ、システムを起動できません。BIOSの仕様や不具合によってブートに必要なデータを読めないこともありますので、**/boot** は別パーティションに確保して、なるべくディスクの先頭に位置させておくことを推奨します。またソフトウェアRAIDを使用して、**/boot** パーティションもソフトウェアRAIDの対象範囲に含めた場合、正常にブートできないことがあります。このようなことを考慮して、**/boot** を独立したパーティションとして、ソフトウェアRAIDの制御対象外にするのがいいでしょう。

1 http://www.pathname.com/fhs/index.html

Asianux Server 3 では、**/boot** パーティションには、ブートに必要なデータとカーネルイメージがインストールされるので、カーネルのアップデートなどに備えて、32MB 以上の容量を確保しておくことを推奨します。※現在は/boot パーティションは 2TB を越える容量は利用できません。

- **/（ルート）**
  「ルートディレクトリ」と呼ばれる、システム全体のファイルシステムの最上位のディレクトリです。Linux では、このパーティションが必ずどこかに確保されている必要があり、システム起動時にマウントされます。システムに必要な情報ファイルや、システムの起動に必要なコマンドがこのパーティション内に存在します。ファイルシステムの最上位に位置するので、他のパーティションとの関係によって、パーティションとして必要な容量が変わります。※現在は/(ルート)パーティションは 2TB を越える容量は利用できません。

- **/home**
  通常、ユーザーのホームディレクトリがこの配下に置かれて、ユーザーのデータファイルなどがここに置かれることになります。ファイルの生存期間は中程度でしょう。このディレクトリも独立したパーティションに確保することが望ましいでしょう。ユーザーが作成したファイルが格納されるので、運用期間が経過するにつれて、ディスク使用量が増加していくことが一般的です。

- **/usr**
  OS のプログラムやライブラリがこのディレクトリ配下にインストールされます。ファイルの生存期間が比較的長いものが集まっており、性能の観点から独立したパーティションを確保することが望まれます。必要なパーティションサイズは、インストールするパッケージによって変わりますが、Asianux Server 3 のインストール時にすべてのパッケージをインストールした場合、4GB 程度が必要となります。しかし、システム運用時には、年々新たなパッケージやプログラムをインストールすることになるため、**/home** と同様に **/usr** も増加していくことが一般的です。よって、あらかじめ余裕を持って領域を確保しておきましょう。また、**/usr/local** ディレクトリには、ユーザーが独自にインストールしたソフトウェアのプログラムやライブラリが置かれるので、システム構成によっては、**/usr/local** を別パーティションとして確保してもいいかもしれません。

- **/var**
  システムのログやスプールファイルなどが、ここに作られます。プログラムによっては、**/var/cache** にキャッシュファイルを作ったり、**/var/tmp** に一時ファイルを作ったりするものもあります。ファイルの生存期間が比較的短いものが集まっており、システムの運用状態によっては、ファイルが次々に作られることもあるので、ルートパーティションとは分けて確保しておくことが望ましいでしょう。パーティションのサイズは、最低でも 1GB 程度確保することを推奨しますが、メールサーバーや HTTP サーバーのデフォルト設定では、このディレクトリ配下にメールや HTML ファイルを配置するので、システムの運用規模に応じた領域を確保しておく必要があります。

- **/tmp**

  /tmp は特殊なディレクトリで、だれもが書き込み可能なディレクトリです。一時ファイルをこのディレクトリ配下に作成するプログラムも数多く存在します。だれもが書き込み可能なディレクトリのため、便利である一方で危険な一面もあります。悪意のあるユーザーや、プログラムのバグによって、自由に大きなファイルを **/tmp** ディレクトリに作成できるため、**/tmp** ディレクトリが **/**（ルート）と同じパーティション内に存在する場合、**/**（ルート）パーティションの空き領域がなくなり、システムに異常をきたす可能性があります。よって、システムをより安全に運用するためには、**/**（ルート）パーティションとは別のパーティションにすることを推奨します。

- **/opt**

  /opt は、アプリケーションのインストール先として利用されるディレクトリです。商用アプリケーションの多くは、**/opt** ディレクトリにアプリケーションのプログラムやデータなどをインストールします。したがって、確保すべき容量は、インストールするアプリケーションに依存します。OS のインストール時には何もインストールされないので、商用アプリケーションなどを使わないのであれば、別パーティションを確保する必要はありません。

システム構築時には、表 4-1 に示す典型的なパーティション分割の設定例を参考にしてください。メールサーバーなどを構築する場合には、**/var** により大きな領域を割り当てる必要があるでしょう。

表 4-1 パーティションの設定例

パーティション	容量	
	メモリ 512MB、ディスク 36GB を備えるファイルサーバー用のシステム	メモリ 1GB、ディスク 72GB を備えるデータベースサーバー用のシステム
/boot	128MB	128MB
swap	1GB	2GB
/	5GB	5GB
/var	5GB（ファイル共有に利用）	1GB
/tmp	1GB	1GB
/home	残り全部（ユーザー用ファイル共有に利用）	1GB
/opt	なし	残り全部（データベースに利用）

4.2.3 パーティションの作成

パーティションの管理には、**fdisk** を使用します。ここでは、**fdisk** の基本的な操作を説明します。

パーティションの作成を行う前に理解しておかなければいけないことは、「既存のパーティションサイズを変更することは容易ではない」という点です。パーティションサイズを変更するという作業は、実質的には古いパーティションを削除して、新規にパーティションを作るという作業と等価です。ファイルシステムの内容を保持したまま、パーティションを拡張、縮小するという作業は難しいため、必ずデータのバックアップを他のデバイスなどに取って、新規パーティションとして確保し直してから、バックアップしていたデータをリストアするという作業が必要になります。

次に大事なことは、「新規パーティションは、ディスク上の連続した未使用領域に対してしか作成できない」ということです。このため、ディスク上に複数の未使用領域が存在していても、連続していない領域をまとめて 1 つのパーティションとすることはできません。したがって、通常、パーティションを追加する場合には、ディスクの最後尾の未使用領域に新たなパーティションを作成することになります。

このように一度使い始めたパーティションを変更することは非常に大変なため、システムインストール前にパーティション割り当てを十分検討することが大切です。

※現在は fdisk では 2TB を越える容量を管理できません。後に記載されている parted をご利用ください。

4.2.4 fdisk によるパーティション操作

fdisk コマンドは、伝統的に Linux のパーティション操作に用いられてきたコマンドで、ハードディスクのパーティションを新規に作成したり、あるいは既存のパーティションを削除したりできます。**fdisk** の引数にはデバイスファイルを指定します。たとえば、**/dev/sda**（1 つ目の SCSI ディスク）に関する操作を行う場合には、次のコマンドを実行します。

```
# /sbin/fdisk /dev/sda
```

起動後に、「m」を入力して[Enter]キーを押すとヘルプが表示されるので、必要なコマンドを入力します。

「p」を入力すると、現在操作中のディスクのパーティション情報が表示されます。

```
コマンド (m でヘルプ): p

Disk /dev/sda: 16.1 GB, 16106127360 bytes
255 heads, 63 sectors/track, 1958 cylinders
Units = シリンダ数 of 16065 * 512 = 8225280 bytes

 デバイス   ブート      始点        終点     ブロック   ID  システム
/dev/sda1   *           1          26      208813+  83  Linux
/dev/sda2              27          91      522112+  82  Linux スワップ
```

```
/dev/sda3            92         665     4610655   83  Linux
/dev/sda4           666        1958    10386022+   f  Win95 拡張領域 (LBA)
/dev/sda5           666        1958    10385991   83  Linux
```

新規にパーティションを作成する場合は、「n」を入力して[Enter]キーを押し、対話式にパーティションの作成を行います。最初に、基本パーティションか拡張パーティションか聞かれるので、4 つ目以降のパーティションであれば、拡張パーティションを選択します。通常は、基本領域から作成していきます。

```
# /sbin/fdisk /dev/sdd
コマンド (m でヘルプ): n
コマンドアクション
   e   拡張
   p   基本領域 (1-4)
p
領域番号 (1-4): 1
```

続いて、パーティションのサイズを指定しますが、最初にパーティションの開始位置が聞かれます。デフォルトでは、未使用領域の先頭が初期値になっているので、特に変更がなければ[Enter]キーを押します。

続いて、パーティションの最後尾、またはパーティションのサイズを指定します。サイズを指定するときには、「+500M」のように指定します(MB 単位の場合)。

```
最初 シリンダ (1-1024, 初期値 1):              ←<Enter>を入力
初期値 1 を使います
終点 シリンダ または +サイズ または +サイズM または +サイズK (1-1024, 初期値 1024):+500M
```

以上で、メモリ上にパーティション情報が作成されました。

```
コマンド (m でヘルプ): p

Disk /dev/sdd: 1073 MB, 1073741824 bytes
64 heads, 32 sectors/track, 1024 cylinders
Units = シリンダ数 of 2048 * 512 = 1048576 bytes

 デバイス ブート  始点      終点   ブロック  ID システム
/dev/sdd1       1         478      489456  83 Linux
```

この時点では、ディスクにはパーティションの情報が反映されていないので、サイズを間違えたりした場合には、「d」を入力してパーティションを削除してから、再度パーティションを作成します。

スワップパーティションを作成する場合には、パーティションのシステム ID を 82 に変更しなければいけません。システム ID の変更は「t」を入力します。また、Windows で使われている VFAT などのパーティションを作る場合にも、システム ID を変更します。システム ID 一覧は、システム ID の入力時に「L」を入力することで表示されます。

```
コマンド (m でヘルプ): t
領域番号 (1-4): 2
16進数コード (L コマンドでコードリスト表示):82
領域のシステムタイプを 2 から 82 (Linux スワップ) に変更しました

コマンド (m でヘルプ): p

Disk /dev/sdd: 1073 MB, 1073741824 bytes
64 heads, 32 sectors/track, 1024 cylinders
Units = シリンダ数 of 2048 * 512 = 1048576 bytes

 デバイス ブート      始点        終点   ブロック    ID  システム
/dev/sdd1               1         478      489456   83  Linux
/dev/sdd2             479         670      196608   82  Linux スワップ  <- システムID変更済
```

パーティションの作成を完了したら、「w」を入力して、変更をディスクに反映させます。いったんパーティション情報をディスクに反映したら元の状態に戻すことはできないので、既存のパーティション情報を変更する場合には、間違いがないことを十分確認してから、パーティション情報を反映させましょう。

```
コマンド (m でヘルプ): w
領域テーブルは交換されました！

ioctl() を呼び出して領域テーブルを再読込みします。
ディスクを同期させます。
```

以上で、パーティションの作成は完了です。

4.2.5 parted によるパーティション操作

GNU Parted は、fdisk よりも操作が簡単な上に、パーティションのリサイズやコピー機能も備えた優れたパーティショニングツールです。

Parted でパーティション設定を行うには、次のようなコマンドを実行します。ここでは、**/dev/sdc**（3 つ目の SCSI ディスク）を開きます。

```
# /sbin/parted /dev/sdc
```

すると、**(parted)**というプロンプトが現れます。**help**コマンドを入力すると、使用できるコマンド一覧が表示されます。

現在のパーティション情報を表示するには、**print**コマンドを入力します。

```
(parted) print

モデル: Virtual HDD [2] (ide)
ディスク /dev/hdc: 1049MB
セクタサイズ (論理/物理): 512B/512B
パーティションテーブル: msdos

番号  開始    終了    サイズ  タイプ      ファイルシステム  フラグ
 1    32.3kB  1045MB  1045MB  プライマリ
```

新規にパーティションを作る場合は、**mkpart**コマンドを使用します。**mkpart**コマンドは、次のように引数を指定して作成するか、引数を指定せずに実行し、対話的に作成を行うかを選択することができます。

```
(parted) mkpart primary ext3 32.3KB 1045MB
```

対話的に作成を進める場合は、次のような流れになります。

```
(parted) mkpart
パーティションの種類? primary/プライマリ/extended/拡張? primary
ファイルシステムの種類? [ext2]? ext3
開始? 32.3KB
終了? 1045MB
```

その他のコマンドの使用法については、「**man parted**」を実行し、参照してください。

パーティションの作成を完了したら、**quit**コマンドを入力して、変更をディスクに反映させます。いったんパーティション情報をディスクに反映したら元の状態に戻すことはできないので、既存のパーティション情報を変更する場合には、間違いがないことを十分確認してから、パーティション情報を反映させましょう。

4.3 ファイルシステム

fdiskなどでパーティションを作成しただけでは、そのパーティションを利用することはできません。OS がそのパーティションを利用するためには、そのパーティション上に**ファイルシステム**を作成しなければいけません。ファイルシステムとは、OS がファイルを管理するための枠組みであり、Asianux Server 3 では ext2、ext3、などのファイルシステムを利用できます。

新しいパーティション上にファイルシステムを作成すると、メタデータと呼ばれる管理情報がパーティション内に作成されて、そのパーティションを利用することが可能となります。

ext3 ファイルシステムは、**ジャーナリングファイルシステム**と呼ばれ、ジャーナリング機能を持っています。ジャーナリング機能は、ファイルシステムの信頼性を向上させるための機能の 1 つです。ジャーナリングファイルシステムにおける「ジャーナル（Journal）」とは、ファイルシステムの変更に対する操作を、あらかじめ準備された領域にログとして記録することを意味します。ジャーナリングファイルシステムは、障害からの復旧時にジャーナルの情報を利用してファイルシステムの復旧を行い、ファイルシステムの一貫性を保つことができます。

一方、Linux の初期の頃から利用されてきた ext2 ファイルシステムは、ジャーナリング機能を持っておらず、システム障害時などファイルシステムを正常にアンマウントできなかった場合、再起動後のマウント時に、**fsck** コマンドによるファイルシステムの検査が行われます。この検査は、ファイルシステム内のすべてのファイルの一貫性を検査するので、ファイルシステムが大きくなると、検査に必要な時間も延び、サービスの停止時間を延ばす要因となります。したがって、現在では ext2 以外のジャーナリングファイルシステムを用いて、システムを運用することが一般的になっています。

4.3.1　ext3 ファイルシステム

ext3 ファイルシステムは、Linux の初期段階から利用されてきた ext2 ファイルシステムに、　　ジャーナリング機能を追加したファイルシステムです。ext3 ファイルシステムは、ext2 ファイルシステムと上位互換であり、既存の ext2 ファイルシステムを ext3 ファイルシステムに変更したり、ext3 ファイルシステムを ext2 ファイルシステムとして利用したりすることが簡単にできます。

ext3 ファイルシステムの操作は、**e2fsprogs** パッケージに含まれているツールを用います。また、ext2 ファイルシステムの操作も ext3 と同じ操作で行うことができます。

➢ ext3 ファイルシステムの作成

ext3 ファイルシステムを新規に構築するには、**mkfs** のオプションとして、ファイルシステムの種類を表す**-t ext3** オプションと、ext3 ファイルシステムを作成するパーティションのデバイスファイルを指定します。

```
# /sbin/mkfs -t ext3 /dev/sdd1
```

また、既存の ext2 ファイルシステムを ext3ファイルシステムに変換できます。ext2 ファイルシステムを ext3 ファイルシステムに変換するためには、**tune2fs** の**-j** オプションを使用します。ファイルシステムの変換はマウント中でも行うことができます。次の例は、**/dev/sda3** 上に作成された ext2 ファイルシステムを ext3 ファイルシステムに変換します。このとき、**/dev/sda3** 上のデータはすべて保持されます。

```
# /sbin/tune2fs -j /dev/sda3
```

➢ ext3 ファイルシステムのマウント

作成した ext3 ファイルシステムは **mount** で、ファイルツリー上にマウントします。次の例では、**/dev/sda3** を **/mnt/asianux1** にマウントします。

```
# /bin/mount -t ext3 /dev/sda3 /mnt/asianux1
```

➢ ext3 ファイルシステムのラベル設定

ext3 ファイルシステムには、ラベルを設定できます。ラベルを用いることの利点は、デバイスの指定時にデバイスファイルではなく、ラベルによってファイルシステムを特定できることです。この機能により、SCSI デバイスを用いて運用しているシステムで、SCSI デバイスの追加・削除などによってデバイスファイルの割り当てが変更されても、システムの運用に影響を与えなくなります。

ext3 ファイルシステムにラベルを指定するためには、**e2label** を利用します。次の例は、**/dev/sda3** にラベル「asianux1」を指定しています。

```
# /sbin/e2label /dev/sda3 "asianux1"
```

現在、ファイルシステムに設定されているラベルを確認したいときには、ラベル名を付けずに **e2label** を実行します。

```
# /sbin/e2label /dev/sda3
asianux1
```

➢ /etc/fstab の変更

作成したファイルシステムをシステムの再起動時に自動的にマウントするためには、**/etc/fstab** に記述を追加します。次の例は **/dev/sda3** デバイスを **/mnt/asianux1** ディレクトリにマウントするための設定例です。

```
/dev/sda3         /mnt/asianux1    ext3    defaults   0   0
```

また、ラベルを利用して指定する場合には、次のように設定します。

```
LABEL=asianux1     /mnt/asianux1    ext3    defaults   0    0
```

4.4 RAW デバイス

通常のファイルシステムは、ディスクに対する I/O 処理の際に、カーネル内部の**ページキャッシュ**と呼ばれるキャッシュにいったんデータをコピーしてから、ページキャッシュ内のデータを I/O します。ページキャッシュにデータをコピーしておくことで、読み込み要求に対しては同じデータを何度もディスクから読む必要がなくなり、書き込み要求に対しては実際のディスクに対する書き込みを遅延させたりできるため、I/O 性能の向上に効果を発揮します。

一方で、商用データベースなどでは、ディスクに対する I/O データは、データベースのメモリ管理システム内のバッファにおいて管理されていて、データベースプログラムとカーネルの 2 箇所でバッファを管理することによってオーバーヘッドが発生してしまいます。

そこで、特定のデバイスに対して行われる I/O 要求はページキャッシュを経由しない方法が実装され、この機能が RAW デバイスとして提供されています。

4.4.1 RAW デバイスの利用

RAW デバイスはパーティション単位で管理します。したがって、RAW デバイスを利用したい場合には、まず RAW デバイス用のパーティションを作成します。

たとえば、**/dev/sdd1**～**/dev/sdd4** までのパーティションを RAW デバイス用に作成したとします。これらのパーティションを RAW デバイスとして利用するためには、**raw** を利用して、それぞれのパーティションを、RAW デバイスにバインドします。

```
# /bin/raw /dev/raw/raw1 /dev/sdd1
/dev/raw/raw1:  bound to major 8, minor 49
# /bin/raw /dev/raw/raw2 /dev/sdd2
/dev/raw/raw2:  bound to major 8, minor 50
# /bin/raw /dev/raw/raw3 /dev/sdd3
/dev/raw/raw3:  bound to major 8, minor 51
# /bin/raw /dev/raw/raw4 /dev/sdd4
/dev/raw/raw4:  bound to major 8, minor 52
```

以上の操作によって、**/dev/raw/raw1**～**/dev/raw/raw4** までが、RAW デバイスとして利用可能になりました。

RAW デバイスのバインド状況を確認したいときには、**raw** の**-qa** オプションを使います。

```
# /bin/raw -qa
```

```
/dev/raw/raw1:  bound to major 8, minor 49
/dev/raw/raw2:  bound to major 8, minor 50
/dev/raw/raw3:  bound to major 8, minor 51
/dev/raw/raw4:  bound to major 8, minor 52
```

なお、RAW デバイスとして利用しているパーティション（今回の例の場合、**/dev/sdd1**～**/dev/sdd4**）に、ファイルシステムを作成して利用してはいけません。ファイルの不整合が発生する可能性があります。

4.4.2 RAW デバイスの起動設定

システム起動時に自動的に RAW デバイスをバインドするためには、**/etc/udev/rules.d/60-raw.rules** ファイルを設定します。次の設定例は、**/dev/sdd1**～**/dev/sdd4** を、**/dev/raw/raw1**～**/dev/raw/raw4** に自動的に設定するためのものです。

```
ACTION=="add", KERNEL=="sdd1", RUN+="/bin/raw /dev/raw/raw1 %N"
ACTION=="add", KERNEL=="sdd2", RUN+="/bin/raw /dev/raw/raw2 %N"
ACTION=="add", KERNEL=="sdd3", RUN+="/bin/raw /dev/raw/raw3 %N"
ACTION=="add", KERNEL=="sdd4", RUN+="/bin/raw /dev/raw/raw4 %N"
```

設定が完了したら再起動を行い、raw コマンドでデバイスが自動でバインドされているかを確かめます。

4.5 ソフトウェア RAID

RAID は、「Redundant Array of Inexpensive Disks」の略で、安価なディスクを組み合わせて、信頼性の高い大容量ディスクアレイを形成しようというものです。実際の機能としては、パーティションを組み合わせて RAID 構成を作りますが、1 台のディスクを複数パーティションに分けて組み合わせても意味がありません。高性能、高信頼性を得るためには、複数台のディスクで RAID システムを構成してください。

ここで説明するソフトウェア RAID は、RAID の機能をカーネルで実現します。したがって RAID コントローラなどのハードウェアがない場合にも利用することが可能です。ただし、RAID コントローラで RAID の機能を実現しているハードウェア RAID と比較すると、ソフトウェア RAID では I/O 処理において RAID 機能の処理のために余分に CPU を使用しますので、I/O パフォーマンスが低下することがあります。

RAID の種類にはリニアモードと RAID レベルというものがあります。以下に Linux のソフトウェア RAID でサポートしているものについて説明します。

- **リニアモード**
  複数のディスクを単純に結合して、大容量のディスクを作成します。1 台のディスクが壊れるとすべてのデータを失う恐れがあるので、信頼性は低く、また性能向上もほとんど望めません。

- **RAID-0**
  「ストライピング」と呼ばれ、データを分割して複数ディスクに書き込みます。これにより I/O 性能が向上しますが、1 台のディスクが壊れるとすべてのデータを失うことになります。

- **RAID-1**
  「ミラーリング」と呼ばれ、1 台のディスクの完全なコピーを他のディスクに保持します。信頼性は高くなりますが、I/O の性能は 1 台のディスクよりも低下します。

- **RAID-5**
  データの書き込み時に、データのパリティ情報も書き込み、データとパリティ情報を複数のディスクに分散して書き込みます。3 台以上のディスクで構成され、1 台のディスクが壊れてもデータを復旧できます。

4.5.1 ソフトウェア RAID の作成

まず少なくとも 2 つのパーティションを用意します。RAID 5 であれば、最低 3 つのパーティションが必要となります。実際の運用では、それらのパーティションが、それぞれ別のディスク上になければ意味がありませんが、ソフトウェア RAID のテストであれば、同じディスク上にあっても特に問題はありません。それぞれのパーティションサイズが同じである必要はありませんが、異なるサイズのパーティションを利用した場合、利用されない領域が発生するので、ディスク領域を効率よく利用するためには等しいことが望ましいでしょう。

ソフトウェア RAID として利用するパーティションは、パーティションの ID を **0xFD** に設定しておくと、起動時に自動的にソフトウェア RAID 用のパーティションとして認識されます。**fdisk** でパーティションを作成する時点で、パーティションの ID を **0xFD** に設定しておきましょう。

```
# /sbin/fdisk /dev/sdd

コマンド (m でヘルプ): p

Disk /dev/sdd: 1073 MB, 1073741824 bytes
64 heads, 32 sectors/track, 1024 cylinders
Units = シリンダ数 of 2048 * 512 = 1048576 bytes

 デバイス ブート    始点       終点   ブロック    ID  システム
/dev/sdd1             1      102     104432  83  Linux

コマンド (m でヘルプ): t
Selected partition 1
16進数コード (L コマンドでコードリスト表示): fd
領域のシステムタイプを 1 から fd (Linux raid 自動検出) に変更しました

コマンド (m でヘルプ): p

Disk /dev/sdd: 1073 MB, 1073741824 bytes
64 heads, 32 sectors/track, 1024 cylinders
Units = シリンダ数 of 2048 * 512 = 1048576 bytes

 デバイス ブート    始点       終点   ブロック    ID  システム
/dev/sdd1             1      102     104432   fd  Linux raid 自動検出
```

次に**/etc/mdadm.conf** ファイルに RAID システムの構成を記述します。

次の例は、**/dev/sdb1**、**/dev/sdc1**、**/dev/sdd1** を使用して RAID を構成するための設定です。

```
DEVICE /dev/sdb1 /dev/sdc1 /dev/sdd1
ARRAY /dev/md0 devices=/dev/sdb1,/dev/sdc1,/dev/sdd1
```

ソフトウェア RAID として構成したいデバイスのデバイスファイル名は、通常は**/dev/md0**、**/dev/md1** という順番で割り当てていきます。 設定内容の詳細に関しては **mdadm.conf** のオンラインマニュアルを参照してください。

/etc/mdadm.conf の記述が完了したら、**mdadm** コマンドを使用し RAID デバイスを作成します。

次の例は、**/dev/sdb1**、**/dev/sdc1**、**/dev/sdd1** を使用して RAID-5 を構成するためのコマンドです。

```
/sbin/mdadm -C /dev/md0 -l 5 -n 3 /dev/sdb1 /dev/sdc1 /dev/sdd1
```

各パラメータの詳細に関しては、**mdadm** のオンラインマニュアルを参照してください。

ソフトウェア RAID の状態を調べるためには、次のように**/proc/mdstat** の参照および、**mdadm** コマンドの**-D** オプションを利用します。

```
# /bin/cat /proc/mdstat
Personalities : [raid5]
md0 : active raid5 sdd1[2] sdc1[1] sdb1[0]
     196352 blocks level 5, 64k chunk, algorithm 2 [3/3] [UUU]

unused devices: <none>
```

```
# /sbin/mdadm -D /dev/md0
/dev/md0:
        Version : 00.90.01
  Creation Time : Tue Jul 19 18:17:28 2005
     Raid Level : raid5
     Array Size : 196352 (191.75 MiB 201.06 MB)
    Device Size : 98176 (95.88 MiB 100.53 MB)
   Raid Devices : 3
  Total Devices : 3
Preferred Minor : 0
    Persistence : Superblock is persistent

    Update Time : Tue Jul 19 18:18:54 2005
          State : clean
 Active Devices : 3
Working Devices : 3
 Failed Devices : 0
  Spare Devices : 0

         Layout : left-symmetric
     Chunk Size : 64K

    Number   Major   Minor   RaidDevice State
       0       3       65        0      active sync   /dev/sdb1
       1       3       66        1      active sync   /dev/sdc1
       2       3       67        2      active sync   /dev/sdd1
           UUID : 5a9fe0cd:e89a279b:29b60829:ec3065d2
         Events : 0.64
```

次のコマンドで RAID デバイスを停止し、すべてのリソースを開放することができます。

```
# /sbin/mdadm -S /dev/md0
```

また、次のコマンドで定義済みの RAID デバイスを編成し、起動することができます。

```
# /sbin/mdadm -A /dev/md0
```

4.5.2 RAID の運用

mdadm で初期化が完了した RAID デバイスは、通常のパーティションのように扱うことができます。次のコマンドは、**/dev/md0** に割り当てられた RAID デバイスに、ext3 ファイルシステムを構築しています。

```
# /sbin/mkfs -t ext3 /dev/md0
```

Asianux Server 3 では、カーネルの起動時に自動的に RAID デバイスを検出して、ソフトウェア RAID を起動します。ただし、パーティションの ID を **0xFD** にしておくことを忘れないようにしてください。

システム起動時に、パーティションを自動的にマウントするためには、**/etc/fstab** にソフトウェア RAID デバイスの設定を追加します。

```
/dev/md0        /mnt/raid       ext3    defaults  1  2
```

4.6 LVM（Logical Volume Manager）

LVM（Logical Volume Manager）は、ユーザーが扱うパーティションとして、**論理ボリューム**と呼ばれる単位でパーティションを提供して、物理的なディスクの存在を隠蔽します。その結果、物理的なディスクの増設や変更などが、ユーザーやアプリケーションに対して隠蔽されて、ディスクデバイス管理の柔軟性を向上させます。

LVM は、**物理ボリューム**（physical volume）、**ボリュームグループ**（volume group）、**論理ボリューム**（logical volume）から構成され、図 4-1 のような構成で管理されます。

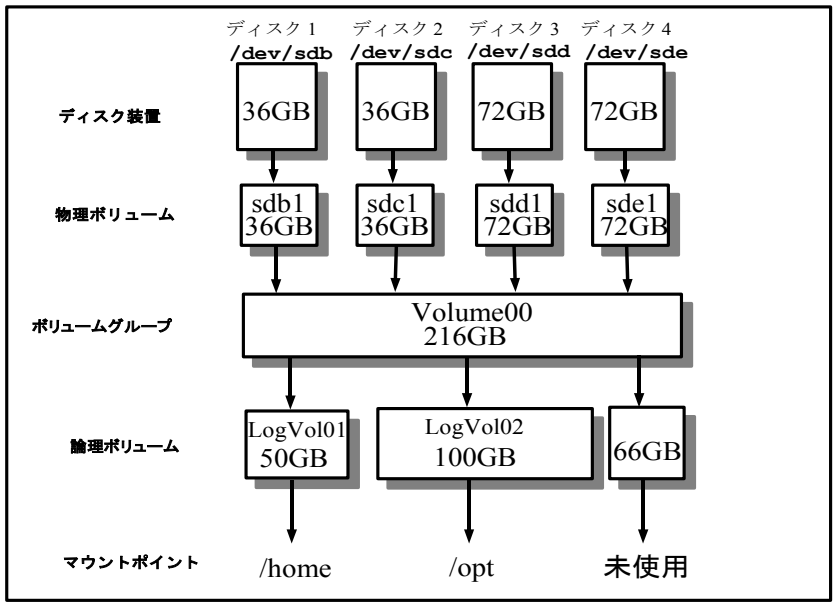


図 4-1 LVM の構成例

LVM の操作は、**lvm** パッケージに含まれるツールによって行われます。

4.6.1 物理ボリュームの作成

物理ボリュームは、パーティション単位で管理されます。したがって、1 つのディスクの全体を 1 パーティションとし、1 つの物理ボリュームとしても構わないですし、一部分だけを LVM 用のパーティションとして確保して 1 つの物理ボリュームとしても構いません。もちろん、1 つのディスクを複数のパーティションに分割して、複数の物理ボリュームを作成することもできます。

LVM 用のパーティションとするためには、最初に **fdisk** を使用して、作成したパーティションの ID を **0x8E** に設定します。

```
# /sbin/fdisk /dev/sdc
コマンド (m でヘルプ): p

Disk /dev/sdc: 1073 MB, 1073741824 bytes
64 heads, 32 sectors/track, 1024 cylinders
Units = シリンダ数 of 2048 * 512 = 1048576 bytes

 デバイス ブート    始点      終点   ブロック    ID  システム
/dev/sdc1          1        96    98288   83  Linux

コマンド (m でヘルプ): t
Selected partition 1
16進数コード (L コマンドでコードリスト表示): 8e
領域のシステムタイプを 1 から 8e (Linux LVM) に変更しました

コマンド (m でヘルプ): p

Disk /dev/sdc: 1073 MB, 1073741824 bytes
64 heads, 32 sectors/track, 1024 cylinders
Units = シリンダ数 of 2048 * 512 = 1048576 bytes

 デバイス ブート    始点      終点   ブロック    ID  システム
/dev/sdc1          1      96      98288   8e  Linux LVM
```

続いて、**pvcreate** でパーティションを物理ボリュームとして初期化します。

```
# /usr/sbin/pvcreate /dev/sdc1
  Physical volume "/dev/sdc1" successfully created
```

LVM の領域として利用するすべての物理ボリュームに対して、初期化を行います。初期化が完了した物理ボリュームは、**pvscan** で確認できます。

```
# /sbin/pvscan
  PV /dev/sdb1          lvm2 [95.76 MB]
  PV /dev/sdc1          lvm2 [95.79 MB]
  Total: 2 [191.55 MB] / in use: 0 [0   ] / in no VG: 2 [191.55 MB]
```

4.6.2 ボリュームグループの作成

すべての物理ボリュームの準備が完了したら、次にそれらの物理ボリュームをもとにボリュームグループを作成します。ボリュームグループは、仮想的なディスクに相当すると考えればよいでしょう。

ボリュームグループの作成は **vgcreate** で行います。**vgcreate** には、ボリュームグループ名と、そのボリュームグループを構成する物理ボリュームを指定します。

```
# /usr/sbin/vgcreate Volume00 /dev/sdb1 /dev/sdc1
  Volume group "Volume00" successfully created
```

ボリュームグループを作成するときに、Physical Extent（PE）サイズを指定できます。PE とは、LVM でデータを管理する単位で、1 つのボリュームグループは 64K 個の PE を管理できます。デフォルトの PE のサイズは 4MB のため、最大で 256GB のボリュームグループを作成できます。もしそれ以上のサイズのボリュームグループを作成したい場合は、**vgcreate** に**-s** オプションで PE のサイズを指定します。たとえば、PE のサイズを 32M として指定した場合、最大 2TB のボリュームグループを作成できることになります。

作成したボリュームグループの情報は、**vgscan** で確認できます。

```
# /sbin/vgscan
  Reading all physical volumes.  This may take a while...
  Found volume group "Volume00" using metadata type lvm2
```

4.6.3 論理ボリュームの作成

作成したボリュームグループの領域を利用して、論理ボリュームを作成します。論理ボリュームは、通常のパーティションに相当するもので、ボリュームグループ全体を 1 つの論理ボリュームとすることもできるし、複数の論理ボリュームに分割して利用することもできます。

論理ボリュームの作成は **lvcreate** で行います。**lvcreate** には、論理ボリュームのサイズ（MB 単位）、論理ボリューム名、論理ボリュームを作成するボリュームグループ名を指定します。

次の例は、ボリュームグループ **Volume00** 上に、50MB の論理ボリューム **LogVol01** を作成しています。

```
# /usr/sbin/lvcreate -L 50M -n LogVol01 Volume00
  Rounding up size to full physical extent 52.00 MB
  Logical volume "LogVol01" created
```

作成した論理ボリュームの情報は、**lvdisplay** で確認できます。

```
# /usr/sbin/lvdisplay /dev/Volume00/LogVol01
```

```
  --- Logical volume ---
  LV Name                /dev/Volume00/LogVol01
  VG Name                Volume00
  LV UUID                wchfwk-6V2n-wrKb-v8D7-LVdS-JXPB-01uzRb
  LV Write Access        read/write
  LV Status              available
  # open                 0
  LV Size                52.00 MB
  Current LE             13
  Segments               1
  Allocation             inherit
  Read ahead sectors     0
  Block device           253:0
```

4.6.4 論理ボリュームの利用

作成した論理ボリュームを利用するためには、**lvcreate** 時に表示された論理ボリューム用のデバイスファイルを利用します。通常は、**/dev/[ボリュームグループ名]/[論理ボリューム名]** にデバイスファイルが作成されます。

通常のファイルシステムとして利用する場合には、ファイルシステムを作成してからマウントします。次の例は、論理ボリューム **/dev/Volume00/LogVol01** を ext3 ファイルシステムとして/hoge にマウントして利用する例です。

```
# /sbin/mkfs -t ext3 /dev/Volume00/LogVol01
# /bin/mount -t ext3 /dev/Volume00/LogVol01 /hoge
```

RAW デバイスとして利用する場合には、RAW デバイスへのバインドを行ってから、**/dev/raw/raw[n]** のデバイスファイル経由で利用してください。

```
# /bin/raw /dev/raw/raw1 /dev/Volume00/LogVol01
/dev/raw/raw1:  bound to major 253, minor 0
```

システム起動時に自動的にマウントするためには、**/etc/fstab** に LVM の論理ボリュームをマウントするための設定を追加してください。

次の設定は、ext3 ファイルシステムとして作成された論理ボリューム **/dev/Volume00/LogVol01** を **/hoge** ディレクトリにマウントするための設定例です。

```
/dev/Volume00/LogVol01   /hoge                    ext3     defaults        1 2
```

4.6.5 スナップショットの取得

LVM には**スナップショット**と呼ばれる機能が備えられています。これは、論理ボリュームのデータをスナップショットとして取得した時点で、読み取り専用の別の論理ボリュームとしてコピーしておく機能です。スナップショットは論理ボリューム上のすべてのデータのコピーを行うのではなく、元データへのリンク情報のみを作成するため、非常に高速に動作します。スナップショットの取得後に、元データが更新された場合、更新される前のデータをスナップショット領域に保存します。そのため、通常は元データの領域よりも少ない領域（一般的には 10%～20%程度）があればスナップショットとして利用できます。

さらに、スナップショットに対しては読み込みしかできないため、データが更新されるといった危険がありません。このため、ファイルシステムのバックアップを取得する際に、まずスナップショットを取得して、バックアップ操作はスナップショットに対して行うことで、バックアップ実行中のデータ更新も発生しなくなるため、バックアップデータの一貫性が保たれます。このようにスナップショット機能はファイルシステムのバックアップを取得する目的に非常に合致します。

スナップショットを取得するためには、ボリュームグループにスナップショットを取得できるための空き領域が必要です。スナップショットに必要な最大サイズはスナップショットの取得対象となる論理ボリュームのサイズですが、通常はそれよりも少ない領域でも問題ありません。スナップショットに必要な領域のサイズは、対象となる論理ボリュームの更新頻度にも依存しています。更新されるデータが多いほどスナップショット用の領域が必要になることを覚えておきましょう。

スナップショットの取得は、**lvcreate** に**－snapshot** オプションを指定して行います。スナップショットにアクセスするためのデバイスファイル名を**-n** オプションで指定します。また、スナップショット領域として利用するサイズを**-L** オプションで指定します。また、**lvcreate** 実行前に、**modprobe** コマンドで **snapshot**ドライバを読み込む必要があります。

```
# /sbin/modprobe dm_snapshot
# /usr/sbin/lvcreate --snapshot -L 50M -n snap1 /dev/Volume00/LogVol01
  Rounding up size to full physical extent 52.00 MB
  Logical volume "snap1" created
```

4.6.6 ディスクの追加

LVM の利点が特に発揮されるのは、ディスクの追加や削除といった状況が発生したときです。直接パーティションを利用していた場合、あるパーティションの容量を拡大したいと思っても、新しいディスクに容量を拡大したパーティションを準備し、すべてのデータをコピーしてから、新しいディスクに交換するといった手順が必要となってしまいます。これに対して、LVM を利用していた場合、既存のディスクを残したまま、新たなディスクを追加して、パーティションを拡大できます。

ここでは、前述の環境に新たなディスク（**/dev/sdd**）を追加する手順を説明します。

最初に新しいディスクにパーティション（**/dev/sdd1**）を作成して、物理ボリュームを作成します。

```
# /usr/sbin/pvcreate /dev/sdd1
  Physical volume "/dev/sdd1" successfully created
```

次に、**vgextend** でこの物理ボリュームを既存のボリュームグループに追加します。

```
# /usr/sbin/vgextend Volume00 /dev/sdd1
  Volume group "Volume00" successfully extended
```

新しい物理ボリュームが追加されたことを **pvscan** で確認します。

```
# /sbin/pvscan
  PV /dev/sdb1   VG Volume00   lvm2 [88.00 MB / 36.00 MB free]
  PV /dev/sdc1   VG Volume00   lvm2 [88.00 MB / 88.00 MB free]
  PV /dev/sdd1   VG Volume00   lvm2 [92.00 MB / 92.00 MB free]
  Total: 3 [268.00 MB] / in use: 3 [268.00 MB] / in no VG: 0 [0   ]
```

次に、**lvextend** で、容量を拡大したい論理ボリュームのサイズを拡大します。下記の例では、論理ボリューム LogVol01 に 50MB の容量を追加しています。**-L** オプションに指定する単位には、「M」（メガバイト）のほか、「G」（ギガバイト）、「T」（テラバイト）も使えます。また、「**+**」を指定することを忘れないようにしましょう。

```
# /usr/sbin/lvextend -L +50M /dev/Volume00/LogVol01
  Rounding up size to full physical extent 52.00 MB
  Extending logical volume LogVol01 to 104.00 MB
  Logical volume LogVol01 successfully resized
```

また、オプションとしてパーティションを指定することで、拡大する領域を確保するディスクを指定できます。

```
# /usr/sbin/lvextend -L +20M /dev/Volume00/LogVol01 /dev/sdd1
  Extending logical volume LogVol01 to 124.00 MB
  Logical volume LogVol01 successfully resized
```

最後に、実際のファイルシステムのサイズを変更する必要があります。ファイルシステムのサイズ変更は、ファイルシステムの情報を変更するので、作業の際には安全のため論理ボリュームはアンマウントしてから行いましょう。

- **ext3 ファイルシステムの場合**

  ext3 ファイルシステムのサイズ変更は **resize2fs** で行うことができます。論理ボリュームのサイズに合わせたパーティションとするためには、特にサイズを指定する必要はありません。作業に先だって、ファイルシステムの整合性をチェックするために、**e2fsck** を実行しておきます。

```
# /sbin/e2fsck -f /dev/Volume00/LogVol01
# /sbin/resize2fs -p /dev/Volume00/LogVol01
resize2fs 1.35 (28-Feb-2004)
Resizing the filesystem on /dev/Volume00/LogVol01 to 106496 (1k) blocks.
Begin pass 1 (max = 6)
Extending the inode table     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
The filesystem on /dev/Volume00/LogVol01 is now 106496 blocks long.
```

  以上で、ext3 ファイルシステムのサイズ拡大は完了です。再度マウントしてから利用してください。

- **RAW デバイスの場合**

  RAW デバイスには、ファイルシステムのサイズのような情報はありません。パーティションのサイズがそのまま RAW デバイスで利用可能な最大サイズとなります。したがって、論理ボリュームを拡大すれば、そのまま拡大したサイズを利用できます。

4.6.7 ディスクの交換

システム運用中には、現在利用中のディスクを別のディスクに変更したいことがあります。LVM を使って運用している場合、LVM 用に利用中の物理ボリュームのデータを、他の空いている物理ボリュームに移すことができます。したがって、利用中のディスクを交換したい場合には、まず利用中の物理ボリュームのデータを他の物理ボリュームに移動させます。そして、未使用状態になったディスクを LVM の管理から切り離します。

物理ボリュームの利用状況は **pvdisplay** で確認できます。PE（Physical Extent）の Allocated PE の項目が利用中のエクステントの数です。この利用中のエクステントを、他の物理ボリュームに移すことが最初の作業です。

```
# /usr/sbin/pvdisplay /dev/sdb1
  --- Physical volume ---
  PV Name               /dev/sdb1
  VG Name               Volume00
  PV Size               88.00 MB / not usable 0
  Allocatable           yes (but full)
```

```
  PE Size (KByte)       4096
  Total PE              22
  Free PE               0
  Allocated PE          22
  PV UUID               z806zf-lCTk-RZrP-hEVn-l983-11Rm-H8OmCk
```

最初に安全のため、利用中の論理ボリュームのファイルシステムをアンマウントします。

次に **pvmove** でその他の空いている物理ボリュームに利用中のエクステントを移動させます。**pvmove** は、さまざまなオプションによって、移動させる物理ボリュームのエクステントを指定できます。

今回は、**/dev/sdb1** に割り当てられているすべてのエクステントを**/dev/sdd1** に移動させます。

```
# /usr/sbin/pvmove /dev/sdb1 /dev/sdd1
  /dev/sdb1: Moved: 86.4%
  /dev/sdb1: Moved: 100.0%
```

この時点で、エクステントの移動が完了したので、再度ファイルシステムをマウントして利用できます。

割り当て済みのエクステントがなくなった物理ボリュームは、ボリュームグループから解放できます。ボリュームグループから解放するために、**vgreduce** で物理ボリュームをボリュームグループから解放します。

```
# /usr/sbin/vgreduce Volume00 /dev/sdb1
  Removed "/dev/sdb1" from volume group "Volume00"
```

以上で、物理ボリュームが解放されたので、**pvscan** で状態が **inactive** になっていることを確認してから、ディスクを取り外します。

```
# /sbin/pvscan
  PV /dev/sdc1   VG Volume00   lvm2 [88.00 MB / 52.00 MB free]
  PV /dev/sdd1   VG Volume00   lvm2 [92.00 MB / 4.00 MB free]
  PV /dev/sdb1                 lvm2 [88.00 MB]
  Total: 3 [268.00 MB] / in use: 2 [180.00 MB] / in no VG: 1 [88.00 MB]
```

4.6.8 ディスクの削除

LVM で利用中のディスクをすべて解放するには、次の手順で行います。

最初に論理ボリューム上で利用中のファイルシステムをアンマウントします。

続いて、**lvchange** で論理ボリュームの利用を停止し、**lvremove** で論理ボリュームを解放します。論理ボリュームの利用を停止するためには、**lvchange** の**-a** オプションに「**n**」を指定します。

```
# /usr/sbin/lvchange -a n /dev/Volume00/LogVol01
# /usr/sbin/lvremove /dev/Volume00/LogVol01
  Logical volume "LogVol01" successfully removed
```

ボリュームグループ上で使用しているすべての論理ボリュームを削除すれば、ボリュームグループを削除できるようになります。ボリュームグループの削除は **vgchange** でボリュームグループの利用を停止して、**vgremove** でボリュームグループを解放します。ボリュームグループの利用を停止するためには、**vgchange** の**-a** オプションに「**n**」を指定します。

```
# /sbin/vgchange -a n Volume00
  0 logical volume(s) in volume group "Volume00" now active
# /usr/sbin/vgremove Volume00
  Volume group "Volume00" successfully removed
```

以上で、関連する物理ボリュームも利用停止状態になったので、ディスクを自由に変更することが可能です。通常のパーティションとして利用したい場合には、**fdisk** でパーティションの ID を変更してから利用してください。

4.7　quota の設定

4.7.1　quota とは

quota を制御するツールを用いると、ユーザーごとやグループごとに、ファイルシステムの使用可能領域を制限できます。制限するのはブロック数と i ノード数です。1 ブロックは 1KB です。i ノードとは、ファイルの情報を格納する領域で、通常は 1 ファイルに 1 つ使用されます。

4.7.2　quota の設定方法

➢ /etc/fstab の修正

まず**/etc/fstab** を編集して、**mount** のオプションを加えます。quota 設定をしたいファイルシステムの記述の第 4 フィールドに **usrquota** または **grpquota** の記述を追加します。次の例では、**/home** にマウントするファイルシステムにユーザー quota、グループ quota の両方のオプションを指定しています。

```
LABEL=/home /home ext3 defaults,usrquota,grpquota 1 2
```

/etc/fstab を編集したら、該当するファイルシステムをマウントし直します。

```
# /bin/mount -o remount /
```

➢ quota ファイルの作成

次のコマンドにより、**/etc/fstab** に記述されている quota を設定するファイルシステムを自動的にチェックして、該当するファイルシステムのトップディレクトリに quota ファイル（**aquota.user**、**aquota.group**）を作ります。

```
# /sbin/quotacheck -vaug
```

quota ファイルはテキストエディタなどで編集できないので、注意してください。
また、ルートパーティションの場合は-m オプションが必要です。

➢ quota ファイルの編集

edquota コマンドで quota ファイルを編集して、各ユーザー、各グループに quota を設定します。次の例では、ユーザー foo の設定を行っています。

```
# /usr/sbin/edquota -u foo
```

グループ asianux の設定を行いたい場合には、次のコマンドを実行します

```
# /usr/sbin/edquota -g asianux
```

edquota コマンドを実行すると、デフォルトでは **vi** が起動します（エディタは環境変数 EDITOR で変更できます）。以下は、**edquota** の実行後にエディタに表示される内容です。

```
Disk quotas for user foo (uid 500):
Filesystem  blocks  soft  hard  inodes  soft  hard
/dev/sda5   200     300   500   51      0     0
```

変更するのは、soft と hard に対応する数値です。

hard は、絶対に超えることのできない最大の制限値です。

soft は、制限時間が設定されている場合に動作する制限値です。ユーザーの使用量が soft の値を超えるとユーザーに警告メッセージが出され、猶予期間に入ります。猶予期間中は hard 制限値まで使用可能ですが、猶予期間が過ぎると書き込みができなくなります。

猶予期間の設定は次のコマンドで行います。

```
# /usr/sbin/edquota -t
```

上の場合と同様にエディタが起動するので、設定を変更してください。

```
Grace period before enforcing soft limits for users:
Time units may be: days, hours, minutes, or seconds
  Filesystem             Block grace period     Inode grace period
  /dev/sda5                     1days                  1days
```

➢ quota の有効化

上記の設定後にシステムを再起動すれば有効になります。手動で quota を有効にするには、**quotaon** を使用します。次の例では、**/etc/fstab** に quota の記述がされているすべてのファイルシステムで、ユーザー quota とグループ quota を有効にします。

```
# /sbin/quotaon -vaug
/dev/sda5 [/home]: group quotas turned on
/dev/sda5 [/home]: user quotas turned on
```

無効にするには、次のコマンドを実行します。

```
# /sbin/quotaoff -vaug
/dev/sda5 [/home]: group quotas turned off
/dev/sda5 [/home]: user quotas turned off
```

➢ quota の確認

quota の設定内容を確認するには、**quota** コマンドを使用します。次の例では、ユーザー foo に対する設定内容を確認できます。

```
# /usr/bin/quota -u foo
Disk quotas for user foo (uid 500):
  Filesystem  blocks  quota  limit  grace  files  quota  limit  grace
  /dev/sda5   312*    300    500    24:00  52     0      0
```

第5章　バックアップ／リストア

この章で説明する内容

目的	システムのバックアップの取得およびシステムの復旧の方法について理解する
機能	バックアップ
必要な RPM	dump ―― バックアップ／リストアコマンド afio ―― アーカイブユーティリティ tar ―― アーカイブユーティリティ star ―― ACL 対応 アーカイブユーティリティ
設定ファイル	`/etc/dumpdates`
章の流れ	1 バックアップの必要性 2 バックアップの方法 3 バックアップ／リストアの実行 4 ACL に関連したバックアップ、リストア 5 ディザスタリカバリーのための手段
関連 URL	**Linux: dump and restore mini-HOWTO** http://www.linux.or.jp/JF/JFdocs/dump-restore-mini-HOWTO.html **Backup-mini-HOWTO** http://www.linux.or.jp/JF/JFdocs/Backup-mini-HOWTO/index.html

5.1　バックアップの必要性

ハードディスクなどの記憶媒体は、時として障害を引き起こして、保存してある情報を読み出せなくなります。障害が発生した場合でもすみやかに復旧できるように、**バックアップ**を取得するなど、事前に対策を講じておく必要があります。

現在は RAID が普及したことにより記憶媒体の可用性が向上していますが、RAID を使用した場合でも障害の発生を確実に阻止できるわけではありません。また、ユーザーの操作ミスによるファイル削除や、故意のデータ破壊などのデータ自体の損傷などに関しては、RAID では対応ができません。

必ず定期的にバックアップを取得して、障害が発生しても確実に復旧できるように日頃から備えるようにしてください。

5.2　バックアップの方法

バックアップの方法は、**フルバックアップ法**、**差分バックアップ法**、**累積差分バックアップ法**の 3 種類に大別できます（図 5-1 を参照）。これらの間には、バックアップに要する時間、リストアに要する時間、保持しなければならないバックアップ媒体の数などのトレードオフが存在します。

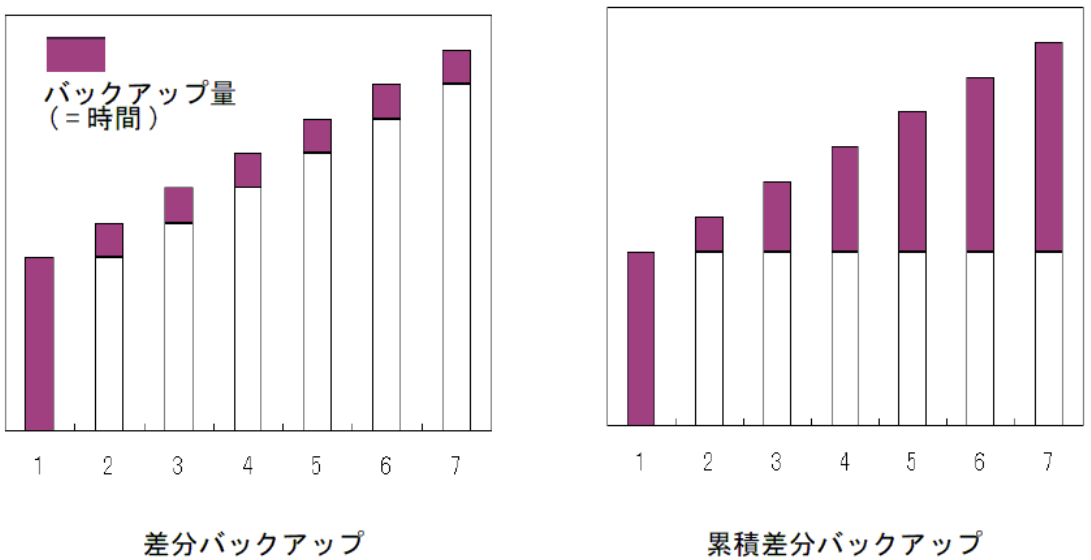


図 5-1　バックアップ方法の種類

- **フルバックアップ法**は、バックアップを取得する際に毎回非常に長い時間を必要としますが、最新の状態へは最新のバックアップ媒体のみでリストアすることが可能です。
- **差分バックアップ法**は、初回のフルバックアップ以外は非常に短い時間でバックアップの取得が可能ですが、場合によっては、最新の状態へ戻すために、初回のバックアップおよびすべての差分をリストアしなければなりません。
- **累積差分バックアップ法**は、バックアップ取得に差分バックアップ法より長い時間を必要としますが、最新の状態へ戻すために最大でも 2 つのバックアップ媒体で済ませることが可能です。

これらのバックアップ取得方法から、稼働中のシステムに対して最も適切な方法を選択して運用してください。ただし、差分バックアップ／累積差分バックアップともに、しかるべき期間をもって差分取得期間を終了して、再度フルバックアップを取得する必要があります。差分または累積差分のみを取得し続けるのは非常に危険です。

5.3 バックアップ／リストアの実行

バックアップを取得する際に使用されるものとして知られているコマンドにはいくつかありますが、ここでは、**dump**（**restore**）コマンド、**afio** コマンド、**tar** コマンドを使用して、SCSI 接続のテープデバイス（次ページの解説を参照）へ保存する方法についてその典型例を元に説明します。

これらのうち、**tar** コマンドと **afio** コマンドはファイルを単位としてバックアップを取得しますが、**dump** コマンドは基本的にファイルシステムを単位としてバックアップを取得します。この違いに注意して実行してください。

dump は対話モードがあって使いやすく、また**インクリメンタルバックアップ**（前回のバックアップ時との差分のみバックアップすること）の機能に関して優れています。**dump** のバックアップ対象は基本的にファイルシステムです。**afio** でバックアップしたデータをリストアした場合には atime（アクセス時間）が変わってしまいますが、**dump** の場合は raw デバイス経由で処理するので変わりません（管理者が意図的に atime のチェックを行っていなければ、atime が変わっても問題ないと思われます）。**tar** でデータのリストアを行う場合、標準では atime が変わりますが、**tar** 実行時に**--atime-preserve** オプションを加えることで変わらないようにすることが可能です。

afio は **gzip** と組み合わせて圧縮バックアップをとれる点で便利です。**dump** にも **gzip** と組み合わせるオプションがありますが、バックアップファイルが壊れてしまったときに **dump** ではまったく復旧できません。**afio** の場合は、壊れている箇所をスキップして最後までファイルをリストアできます。

tar コマンドも、比較的使いやすいため利用者は多いと思いますが、圧縮バックアップしたものの一部が壊れてしまうと、やはりその箇所以降のファイルをリストアできなくなるという問題点があります。

いずれのコマンドを使用する場合でも、バックアップの取得対象としているファイルまたはファイルシステムがバックアップ取得中に他のプロセスによって書き換えられることがないよう注意する必要があります。

バックアップを差分や累積差分で取得している場合には、リストアに注意が必要です。

フルバックアップをまずリストアして、差分の取得日付の古い順にリストアすることでバックアップを取得している中での最新の状態に修復することが可能となります。また、実際にリストアする前にそれぞれのコマンドで用意されているオプションを用いてバックアップファイルの内容を確認することを推奨します。

テープデバイス

デフォルトは**/dev/tape** です。

Asianux Server 3 の場合は、テープデバイスは**/dev/st***または**/dev/nst***として指定します。

- **/dev/st***を指定すると、テープドライブはコマンド実行後にテープの先頭まで巻き戻します。

BOT	今回のバックアップ	EOF		EOF

- **/dev/nst***を指定すると、巻き戻しません。メディアに追記していく場合は**/dev/nst***を利用します。

BOT	前回のバックアップ	EOF	今回のバックアップ	EOF

常に同じテープデバイスにバックアップする場合には、次のようにシンボリックリンクを作成しておくと、毎回の指定を省略できて便利です。

```
# /bin/ln -s /dev/nst0 /dev/tape
```

次にそれぞれのコマンドについて解説していきます。

5.3.1 dump コマンド

➢ dump によるバックアップ対象

バックアップをするファイルシステムの選択には**/etc/fstab** が使用されます。**fstab** の第 5 フィールドが「1」となっているファイルシステムがバックアップの対象となります。ただし「/」ファイルシステムは該当しません。

```
LABEL=/                 /                       ext3    defaults        1 1
LABEL=/boot             /boot                   ext3    defaults        1 2
LABEL=/home             /home                   ext3    defaults        1 2
LABEL=/var              /var                    ext3    defaults        1 2
```

また **dump** コマンドは、バックアップ対象のファイルシステムが ext2 または ext3 でなければ理解できません。つまり **dump** コマンドでは、ReiserFS などのファイルシステムはサポートされていません。

➢ dump の使用方法

dump コマンドは root 権限で実行します。

次に示す例では、バックアップ対象は **/home** です。このバックアップ対象は一般に **/etc/fstab** に記述されているマウントポイント（ディレクトリ）か、ディスクパーティションを指定します。

/etc/fstab に記述のないファイルシステム上のファイルを指定する場合には、**-u** オプションは指定できず、フルバックアップのみ実行可能です。**/home/hoge** のようなサブディレクトリを指定する場合も同様です。

```
# /sbin/dump -0u -b 20 -f /dev/nst0 /home
```

以下に主なオプションの説明を示します。詳細は、「**man 8 dump**」を参照してください。

- **-0～9 ——** バックアップレベルを指定します。
  0 を指定すると、フルバックアップを取ります。1 以上の数字を指定すると、インクリメンタルバックアップを取ります。インクリメンタルバックアップは、**/etc/dumpdates** を参照して、指定されたレベルより小さいレベルのバックアップの日付を探し、その日から更新のあったもののみバックアップします。

- **-u ——** **/etc/dumpdates** ファイルに記録します。
  /etc/dumpdates をエディタで編集してすべての行を削除した場合、その後 **dump** コマンドに **-u** を指定して実行したときにコアダンプする場合があります。そのときは以下のようにして **/etc/dumpdates** ファイルを初期化してください。

```
# /bin/rm /etc/dumpdates; /bin/touch /etc/dumpdates
```

- **-b ——** 一度の I/O でやりとりするブロックのサイズを KB 単位で指定します。デフォルトは 10KB です。

- **-f ——** ダンプ出力先のファイルまたはデバイスを指定します。

➢ dump での差分バックアップ

dump での差分バックアップは、フルバックアップを取得した後に実行可能となります。あるダンプレベルが指定されると、それより小さい数のダンプレベルで **dump** が実行された時刻より後に更新されたファイルだけをバックアップの対象にします。

```
# /sbin/dump -1u -b 20 -f /dev/nst0 /home
```

上記の例のようにレベルを 1 で続けていけば累積差分バックアップとなり、レベルを日ごとに増やしていけば差分バックアップとなります。
差分バックアップをテープなどのメディアに取る際には**/dev/st***では上書きしてしまうので、**/dev/nst***で追記することを忘れないように注意が必要です。また、**/home/hoge** のようなサブディレクトリのみの差分バックアップは、**/etc/dumpdates** に記述されない為取得できません。

5.3.2　restore コマンド

➢ restore の使用方法

dump コマンドで取得したバックアップは **restore** コマンドでリストアします。**restore** コマンドは root 権限で実行します。
バックアップ媒体からのフルリストアを行う一般的な使用方法は、次のとおりです。

```
# /sbin/restore -r -f /dev/nst0
```

restore コマンドを実行すると、バックアップがカレントディレクトリにリストアされます。
以下に、主なオプションの説明を示します。詳細は、「**man 8 restore**」を参照してください。
-r、**-i**、**-x**、**-t** は、どれか 1 つを指定します。

- **-r** ―― バックアップファイルの内容すべてを一括でリストアします。
- **-i** ―― 対話モードです。「restore > 」のプロンプトが表示されて、**ls**、**add**、**extract** などのコマンドでファイルの選択、リストアを行います。
- **-x** ―― 指定したファイルのみリストアします。
- **-t** ―― バックアップファイルの内容をリストします。
- **-f** ―― バックアップファイルまたはテープデバイスを指定します。
- **-s** ―― テープのアーカイブの位置を指定します。

-i の対話モードでは、次のようなプロンプトが出ます。

```
# /sbin/restore>
```

この状態で、次のコマンドを使用して展開するファイルを選択、展開を実行します。

- **ls** —— ファイルを表示します。
- **cd** —— バックアップファイル内でディレクトリを移動します。
- **add** —— ファイルを選択します。
- **extract** —— ファイルを展開します。
- **help** もしくは **?** —— コマンドヘルプを表示します。

差分バックアップを取得している際、テープの操作が必要になります。方法は **restore** の**-s** オプションを使用する方法と、**mt** コマンドでテープの位置を指定する方法の 2 種類があり、**restore** コマンドは両方使用することが可能です。

➢ restore の-s オプションを使用する方法

restore の**-s** オプションでは、現在の位置から何番目のバックアップファイルかを数字で指定します。指定する数字は 1 以上です。たとえば 3 つのバックアップを取得済みで、現在テープの先頭にいるときに 3 つ目のバックアップファイルをリストアする場合には、次のように 3 を指定します。

```
# /sbin/restore -r -f /dev/nst0 -s 3
```

➢ mt コマンドでテープの位置を指定する方法

この方法は **restore** コマンドだけではなく、**afio** コマンドや **tar** コマンドでのリストアでも使用する方法です。

mt コマンドで現在の位置から何番目の EOF に移動するかを指定します。

mt コマンドのオプションには、次のようなものがあります。詳細は、「**man 8 mt**」を参照してください。

```
# /bin/mt -f /dev/nst0 status -- テープのステータスを表示します。
# /bin/mt -f /dev/nst0 rewind -- テープの先頭の位置に戻ります。
# /bin/mt -f /dev/nst0 offline -- テープをイジェクトします。
# /bin/mt -f /dev/nst0 fsf n -- テープを先送りし、n番目のEOFを読み込んだところで停止します。
# /bin/mt -f /dev/nst0 bsf n -- テープを巻き戻し、n番目のEOFを読み込んだところで停止します。
```

たとえば、3 つのバックアップを取得済みで、現在テープの先頭にいるときに 3 つ目のバックアップファイルをリストアする場合には、次のように 2 つ先の EOF を指定します。

```
# /bin/mt -f /dev/nst0 fsf 2
```

この時、テープの位置は図 5-2～図 5-3 のように移動します。

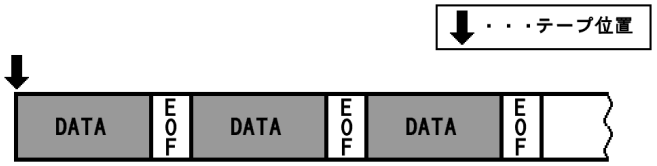


図 5-2　コマンド実行前

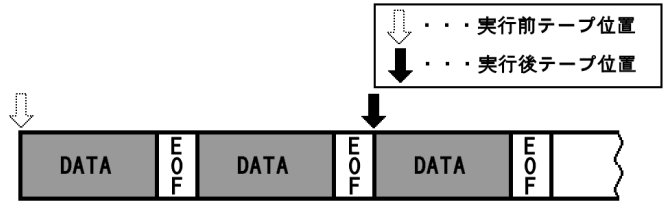


図 5-3　”/bin/mt -f /dev/nst0 fsf 2” 実行後

また別の例として、3 つ目のバックアップの終わりの EOF の位置にいて、2 つ目のバックアップをリストアしたい場合には、次のように実行します。

```
# /bin/mt -f /dev/nst0 bsf 3
# /bin/mt -f /dev/nst0 fsf 1
```

この時、テープの位置は図 5-4～図 5-6 のように移動します。バックアップデータの先頭に移動するため、一旦 bsf で 3 戻り、fsf で 1 進めていることにご注意ください。fsf で 1 進めることにより、EOF の前から EOF の後へ移動しています。

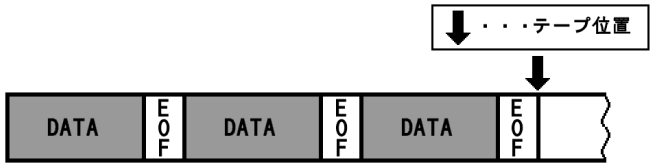


図 5-4 コマンド実行前

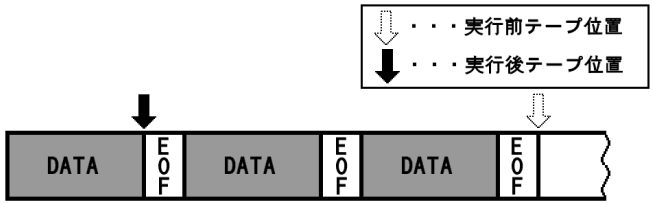


図 5-5 ”**/bin/mt -f /dev/nst0 bsf 3**” 実行後

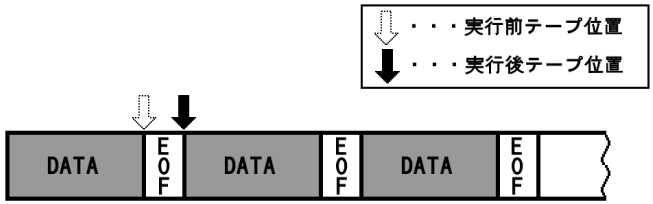


図 5-6 ”**/bin/mt -f /dev/nst0 fsf 1**” 実行後

上記のようにテープの位置を移動した後に、次のように実行することでリストアすることが可能となります。

```
# /sbin/restore -r -f /dev/nst0
```

5.3.3　afio コマンド

➢ afio の使用方法

afio コマンドはバックアップ対象がファイル単位で、一般ユーザーからも利用できます（ただし、一般ユーザーから使用する場合は、テープデバイス(/dev/nst*)へのアクセス権が必要です）。
下の例でバックアップ対象は**/home** です。バックアップ時には、バックアップするファイルのリストを **afio** コマンドに標準入力で渡します。一般的に使われるのは **find** コマンドです。**ls** でリストを出力させたり、エディタで作成したバックアップリストファイルの内容を **cat** で出力させたりもできます。
出力先はコマンド行の最後に記述します。

```
# /usr/bin/find /home | /bin/afio -ovZ -L /tmp/full.log /dev/nst0
```

以下に、主なオプションの説明を示します。詳細は、「**man 1 afio**」を参照してください。
-o、**-i**、**-t**、**-p** は、どれか 1 つを指定します。

- **-o** ―― バックアップファイルのパス名を標準入力から受け取ります。
- **-i** ―― リストアを行います。
- **-t** ―― バックアップファイルの内容を表示します。
- **-p** ―― コピーします。
- **-v** ―― リストや詳細な情報を出力します。
- **-Z** ―― **gzip** により圧縮バックアップを行います。
- **-L** ―― ログを指定ファイルに出力します。

バックアップの対象から除外する場合には、次のように実行します。

```
# /usr/bin/find /home | /bin/grep -v "jpg" | \
                                /bin/afio -ovZ -L /tmp/full.log /dev/nst0
```

この場合は、**/home** 以下でファイルパスに「jpg」を含むファイルをバックアップ対象から除外しています。
バックアップしたファイルの内容を確認するには以下のように-t オプションを指定します。-v オプションを指定することで、「**ls -l**」の情報と同等の詳細なリストが表示されます。

```
# /bin/afio -tvZ /dev/nst0
```

➢ afio での差分バックアップ

afio での差分バックアップは、フルバックアップで作成したログファイルを元に実行します。

```
# /usr/bin/find /home -cnewer /tmp/full.log | \
                                /bin/afio -ovZ -L /tmp/sabun1.log /dev/nst0
```

上記を繰り返すと、フルバックアップからの累積差分バックアップとなります。差分バックアップを取得したい場合には、「**-cnewer**」で指定するファイルとして、前日のログファイルを指定します。

➢ afio でのリストア

バックアップ媒体からのフルリストアを行う一般的な使用方法は下記のとおりです。実行すると、バックアップ媒体に格納されているファイルがすべてカレントディレクトリに展開されます。なおバックアップ取得時に**-z** オプションを指定しなかった場合には、リストア時にも同様に**-z** オプションを指定しないでください。また、注意として、**mt** コマンドで目的の位置までテープを移動させておくことを忘れないでください。

```
# /bin/afio -ivZ /dev/nst0
```

-y オプションを利用することで、部分リストアを行うことができます。**-y** オプションの引数には、**"パターン"** を指定でき、**"パターン"**にマッチしたファイルやディレクトリがリストアされます。しかし、バックアップ取得時に「**/**」からのパスを指定した場合、最初の「**/**」がパスから取り除かれているので注意が必要です。
たとえば、**/home/hoge/test.txt** ファイルを指定するには次のように実行します。

```
# /bin/afio -ivZ -y "home/hoge/test.txt" /dev/nst0
```

ディレクトリ**/home/hoge** を指定するには次のように実行します。

```
# /bin/afio -ivZ -y "home/hoge*" /dev/nst0
```

5.3.4 tar コマンド

➢ tar の使用方法

tar コマンドは、バックアップ対象がファイル単位で、一般ユーザーからも利用できます（ただし、一般ユーザーから使用する場合は、テープデバイス(/dev/nst*)へのアクセス権が必要です）。取得対象となるファイルまたはディレクトリのリストを入力します。複数のファイルまたはディレクトリを併記する場合には、それらの名前の間を

スペースで区切ります。ディレクトリを指定した場合には、そのディレクトリ配下のファイルとディレクトリを再帰的にバックアップするよう動作します。
下の例でバックアップ対象は**/home** です。

```
# /bin/tar cvf /dev/nst0 /home
```

以下に主なオプションの説明を示します。詳細は、「**man 1 tar**」を参照してください。

- **-A** —— アーカイブに **tar** ファイルを追加します。
- **-c** —— 新しいアーカイブを作成します。
- **-d** —— アーカイブとファイルシステムとの差分を取ります。
- **-r** —— アーカイブの最後にファイルを追加します。
- **-t** —— アーカイブ内容の一覧を表示します。
- **-u** —— アーカイブ内の同名のファイルより新しいものだけを追加します。
- **-x** —— アーカイブからファイルを抽出します。
- **-C** —— 指定したディレクトリに移動します。
- **-f** —— アーカイブ・ファイルまたはデバイスを指定します（デフォルトは**"-"** すなわち標準入出力）。
- **-v** —— 処理したファイルの一覧を詳しく出力します。
- **-z** —— アーカイブを **gzip** でフィルターします。
- **-N** —— 指定した日付より新しいファイルだけ格納します。

なお、バックアップの取得を目的として **tar** コマンドを使用する場合は、**-z** オプションを指定しての圧縮アーカイブの作成は推奨できません。圧縮アーカイブに損傷が生じた場合に、損傷箇所以降に格納されたファイルがすべてリストア不可能となる為です。バックアップ媒体の大きさとの兼ね合いで圧縮を行わなければならない場合には、先に説明した **afio** コマンドを使用することを推奨します。

➢ tar での差分バックアップ

tar での差分バックアップは日付を指定することで、その日付以降に更新された対象をバックアップすることで可能となります。

```
# /bin/tar cvf /dev/nst0 -N "2004-01-01 18:00:00" /home
```

上記の例では、2004 年 1 月 1 日 18:00:00 以降に**/home** で変更されたファイルのバックアップを取ります。設定日付を一定にすることで累積差分バックアップが取得可能となり、設定日付としてバックアップを取得する前日を指定することで、差分バックアップの取得が可能となります。

➢ tar でのリストア

バックアップ媒体からのフルリストアを行う一般的な使用方法は、次のとおりです。実行すると、バックアップ媒体に格納されているファイルがすべてカレントディレクトリに展開されます。なお展開時には、バックアップ取得時のディレクトリ構成がそのまま反映されます。
注意として、**mt** コマンドで目的の位置までテープを移動させておくことを忘れないでください。

```
# /bin/tar xvf /dev/nst0
```

またカレントディレクトリ以外にリストアを行う場合は、C オプションを指定します。**dest** にはリストアしたいディレクトリへのパスを指定してください。下の例では**/tmp/test** へリストアします。

```
# /bin/tar xvf /dev/nst0 -C /tmp/test
```

部分リストアを行う場合には、引数としてリストア対象のファイルまたはディレクトリの、バックアップ取得時のパスを指定します。複数を列挙する場合には、スペースで区切ります。
なお、バックアップ取得時に「**/**」からのパスを指定した場合、最初の「**/**」がパスから取り除かれているので注意が必要です。
たとえば、/home のバックアップを取るには次のように実行します。

```
# /bin/tar cvf /dev/nst0 /home
```

/home/hoge をリストアしたい場合には次のように実行します。

```
# /bin/tar xvf /dev/nst0 home/hoge
```

5.4 ACL に関連したバックアップ、リストア

ACL（Access Control Lists）に対応したバックアップ、リストア方法としては、アーカイブユーティリティである tar の ACL 対応版である star があります。ここでは **star** コマンドの使用方法について説明します。

5.4.1 star でのバックアップ

star コマンドは、**tar** コマンド同様バックアップ対象がファイル単位で、一般ユーザーからも利用できます。
（ただし、一般ユーザーから使用する場合は、テープデバイス(/dev/nst*)へのアクセス権が必要です。）

取得対象となるファイルまたはディレクトリのリストを入力します。複数のファイルまたはディレクトリを併記する場合には、それらの名前の間をスペースで区切ります。ディレクトリを指定した場合には、そのディレクトリ配下のファイルとディレクトリを再帰的にバックアップするよう動作します。

下の例でバックアップ対象は/home です。

```
# /usr/bin/star -cv -acl artype=exustar f=/dev/nst0 /home
```

以下に主なオプションの説明を示します。詳細は、「**man 1 star**」を参照してください。

- **-c** —— 新しいアーカイブを作成します。
- **-t** —— アーカイブ内容の一覧を表示します。
- **-x** —— アーカイブからファイルを抽出します。
- **-v** —— 処理したファイルの一覧を詳しく出力します。
- **-z** —— アーカイブを **gzip** にフィルターします。
- **f** —— アーカイブ・ファイルまたはデバイスを指定します。
- **-C** —— 指定したディレクトリに移動します。
- **-acl artype=exustar** —— ACL 情報をアーカイブすることを指定する。リストア時には**-acl のみ指定**

5.4.2 star でのリストア

バックアップ媒体からのフルリストアを行う一般的な使用方法は、次のとおりです。実行すると、バックアップ媒体に格納されているファイルがすべてカレントディレクトリに展開されます。なお展開時には、バックアップ取得時のディレクトリ構成がそのまま反映されます。

```
# /usr/bin/star -xv -acl f=/dev/nst0
```

またカレントディレクトリ以外にリストアを行う場合は、C オプションを指定します。**dest** にはリストアしたいディレクトリへのパスを指定してください。下の例では**/tmp/test** へリストアします。

```
# /usr/bin/star -xv -acl f=/dev/nst0 -C /tmp/test
```

部分リストアを行う場合には、引数としてリストア対象のファイルまたはディレクトリの、バックアップ取得時のパスを指定します。複数を列挙する場合には、スペースで区切ります。

なお、バックアップ取得時に「**/**」からのパスを指定した場合、最初の「**/**」がパスから取り除かれているので注意が必要です。

たとえば、/honme のバックアップを取るには次のように実行します。

```
# /usr/bin/star -cv -acl artype=exustar f=/dev/nst0 /home
```

/home/hoge をリストアしたい場合には次のように実行します。

```
# /usr/bin/star -xv -acl f=/dev/nst0 home/hoge
```

5.5 ディザスタリカバリーのための手段

ディザスタリカバリーとは、サーバーのハードウェア障害などにより、OS やミドルウェアの再インストールを余儀なくされる障害から、再インストールすることなく、バックアップ時の状態まで復旧することです。

5.5.1 バックアップ

今回の例では、dump/restore を使用した方法を説明しますが、**afio** や **tar** など他のバックアップツールでも同様のバックアップ／リカバリーが可能です。

ファイルシステム構成は手動で作成する必要がありますので、**/etc/fstab** および **fdisk -l** コマンドの出力を別サーバーへバックアップするかプリントアウトしておきます。

OS を含めた全てのファイル、ディレクトリをバックアップするため、シングルユーザーモードで起動します。シングルユーザーモードで起動する方法については、第 35 章「35.3 シングルユーザーモード」を参照してください。

今回の例では以下のようなファイルシステム構成を想定しています。

```
/dev/sda3    /      ext3
/dev/sda2           swap
/dev/sda1    /boot  ext3
```

（1）/（ルート）ファイルシステムにマウントされている、/boot からバックアップを取得します。

今回の例では 1 本のテープに全てのファイルシステムをバックアップします。

テープメディアをテープドライブに挿入し、次のコマンドを実行して巻き戻します。

```
#  /bin/mt -f /dev/nst0 rewind
```

（2）dump コマンドで **/boot** のバックアップを取得します。

```
# /sbin/dump -0uf /dev/nst0 /boot
```

（3）**mt** コマンドでテープのヘッド、ファイル位置を確認し記録します。この例では File number が[1]まで利用されました。

```
# /bin/mt -f /dev/nst0 status
SCSI 2 tape drive:
File number=1, block number=0, partition=0.
省略
```

(4)引き続き同じテープの続きに/(ルート)のバックアップを取得します。

```
# /sbin/dump -0uf /dev/nst0 /
```

(5)**/boot**と同じように**mt**コマンドにてテープのヘッド、ファイル位置を確認し記録します。

この例ではFile numberが[2]まで利用されました。

```
# /bin/mt -f /dev/nst0 status
SCSI 2 tape drive:
File number=2, block number=0, partition=0.
省略
```

(6)以上でバックアップは終了です。**mt**コマンドでテープを巻き戻して保管します。

```
# /bin/mt -f /dev/nst0 rewind
```

5.5.2 リストア

restoreツールではext3でフォーマットされたパーティションにファイルをリストアするため、事前にバックアップ前の容量かそれ以上のパーティションを作成し、ext3でフォーマットしておく必要があります。ハードディスクに十分な容量がある場合には fdiskが使用できます。またsfdiskを使用してパーティションを分割することも可能です。それぞれの使用法はマニュアルで調べることができます。man fdiskおよびman sfdisk

ハードウェアの交換が終了した時点で、Asianux Server 3インストールDVDからレスキューモードで起動します。

レスキューモードで起動する場合はインストールDVDで起動直後に以下のように入力します。

```
boot: linux rescue
```

※このときインストール時にドライバディスクを利用していた場合は、[dd]を追加して以下のように実行します。

```
boot: linux dd rescue
```

言語選択、キーボード選択、ネットワーク設定の後、パーティションを**/mnt/sysimage** にマウントするか確認されるので、[Skip]を選択します。

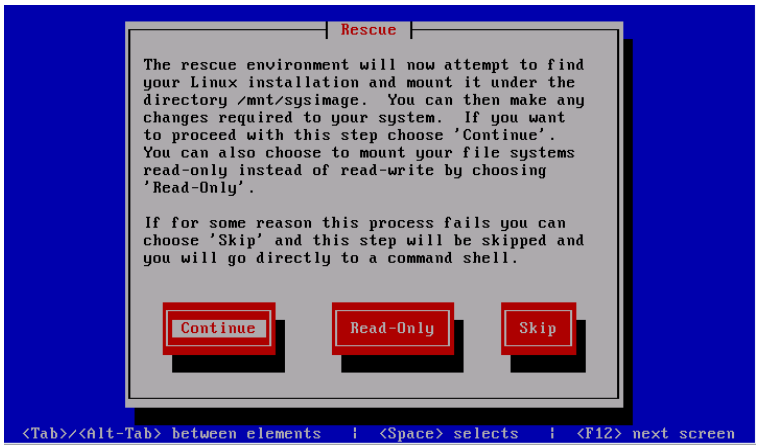


図 5-7　rescue モード画面

今回の例では以下のようなファイルシステム構成を想定しています。

```
/dev/sda3    /      ext3
/dev/sda2           swap
/dev/sda1    /boot  ext3
```

(1) **fdisk** を利用して保存してあった fstab の通りパーティションを作成します。

```
# fdisk /dev/sda
```

(2) パーティション作成が終了したら、ext でフォーマットします。

```
# mkfs.ext3 /dev/sda1
# mkfs.ext3 /dev/sda3
```

(3) /(ルート)からリストアするため、**/mnt/sysimage** 以下に/(ルート)をマウントします。

```
# mkdir /mnt/sysimage
# mount /dev/sda3 /mnt/sysimage
```

(4) ここからテープからのリストア作業に入ります。

まず、テープのヘッド位置を/(ルート)のバックアップ開始位置に移動します。
fsf の後の数字はバックアップ時のメモを参考に/boot のバックアップ終了時(/のバックアップ開始時)とし、今回は[1]です。

```
# mt -f /dev/nst0 fsf 1
```

(5)バックアップ開始位置に移動したら、リカバリ先のディレクトリに移動し、**restore** コマンドを実行します。

```
# cd /mnt/sysimage/
# restore -rf /dev/nst0
```

(6) /(ルート)のリストアが終了した時点で**/mnt/sysimage** 以下に**/boot** が作成されています。
/boot をリストアするパーティションをマウントします。

```
# mount /dev/sda1 /mnt/sysimage/boot
```

(7)**mt** にてヘッド位置を**/boot** のバックアップ開始位置(先頭)にします。

```
# mt -f /dev/nst0 rewind
```

(8)boot ディレクトリに移動し、**restore** でデータをリストアします。

```
# cd /mnt/sysimage/boot
# restore -rf /dev/nst0
```

この時点で、必要なデーターは全てテープからディスクへ書き出されました。

➢ リストア後のファイルシステムラベル付与

通常のインストールでは**/etc/fstab** 内のデバイス名は LABEL を利用しているため、新規に作成したパーティションでは、起動時に fstab 内の LABEL が利用できないためにマウントエラーが発生し、起動しません。
ラベルを設定するには **e2label** という ext3 パーティションに LABEL を付与するためのツールを使用します。
保存してある fstab を参考に、次のようにラベルをつけます。(ラベルを利用していな場合は不要です。)

```
# /usr/sbin/e2label /dev/sda1 /boot
```

➢ リストア後の GRUB のインストール

GRUB ブートローダーのデータはディスクの物理位置に依存しているため、リストアしただけでは起動できない可能性があります。

この問題に対処するため、いったん再起動し再度レスキューモードで起動します。

この際は書き込みが必要なため、レスキュー画面の確認では「Continue」を選択します。

マウントされたリストア済みのパーティションに **chroot** します。

```
# /usr/sbin/chroot /mnt/sysimage
```

grub コマンドを利用し、MBR にデータを書き込みます。

※デバイス名などは例なので環境に合わせてください。

```
# /sbin/grub
grub> device (hd0) /dev/sda
grub> root (hd0,0)
 Filesystem type is ext2fs, partition type 0x83
grub> setup (hd0)
 Checking if "/boot/grub/stage1" exists... no
 Checking if "/grub/stage1" exists... yes
 Checking if "/grub/stage2" exists... yes
 Checking if "/grub/e2fs_stage1_5" exists... yes
 Running "embed /grub/e2fs_stage1_5 (hd0)"...  15 sectors are embedded.
succeeded
 Running "install /grub/stage1 (hd0) (hd0)1+15 p (hd0,0)/grub/stage2
/grub/grub.conf"... succeeded
Done.
grub> quit
```

exit を 2 回実行し、**chroot** とレスキューモードを抜けます。

この時点でサーバーは再起動されますので、インストール DVD-ROM を抜いてサーバーが正常に起動することを確認してください。

正常に起動され、利用できることが確認できましたらリカバリ完了です。

第6章　ネットワーク設定

この章で説明する内容

目的	システムをネットワークに接続する方法について理解する
機能	ネットワーク上の他のシステムとの通信
必要な RPM	initscript —— 基本システムスクリプト net-tools —— ネットワーク設定の基本ツール
設定ファイル	/etc/sysconfig/network /etc/sysconfig/network-scripts/ifcfg-* /etc/hosts /etc/resolv.conf /etc/modprobe.conf /etc/modules.conf
章の流れ	1 ネットワーク設定の概要 2 ネットワークの起動と停止 3 ネットワークの設定 4 ネットワークの状況の確認 5 ボンディングインターフェイスの設定 6 ジャンボフレームの設定
関連 URL	**The Linux Japanese FAQ Project** http://www.linux.or.jp/JF/JFdocs/INDEX-network.html

6.1　ネットワーク設定の概要

Linux システムでは、ほとんどの運用ケースにおいて TCP/IP ネットワークに接続します。この章では Linux システムを LAN に接続するときの設定方法、設定内容、また簡単な設定の確認方法などを説明します。

設定手順は、接続するネットワーク環境によって異なります。たとえば、DHCP サーバーがあるネットワーク環境では、ほとんどのネットワーク情報を自動的に設定できるので、設定作業は非常に簡単です（ただし、Linux システムをサーバーとして運用する場合は、IP アドレスやホスト名を固定で設定するケースが一般的です）。

6.2　ネットワークの起動と停止

ネットワークの起動スクリプトは**/etc/rc.d/init.d/network** です。通常はシステムの起動と同時に実行されますが、このスクリプトは以下のように手動で実行することも可能です。

- ネットワークを起動するには、次のコマンドを実行します。

```
# /sbin/service network start
```

- ネットワークを停止するには、次のコマンドを実行します。

```
# /sbin/service network stop
```

- ネットワークを再起動するには、次のコマンドを実行します。

```
# /sbin/service network restart
```

- ネットワークの現在の状況を確認するには、次のコマンドを実行します。

```
# /sbin/service network status
設定されたデバイス:
lo eth0
現在活動中のデバイス:
lo eth0
```

一般にネットワークの設定を修正した場合には、他のサービスとの関連を考慮して、システムを再起動したほうがいいでしょう。

6.3 ネットワークの設定

6.3.1 設定方法

通常は、Asianux Server 3 のネットワークに関する設定は、システムのインストール時にインストーラ内で行います。詳細は本製品に同梱されている「Asianux Server 3 インストレーションガイド」のインストール手順を参照してください。

インストール終了後にネットワークを再設定するには、CUI の場合は **system-config-network-tui** コマンド、X Window 環境の場合は **system-config-network-gui** コマンドを実行します。

```
# /usr/sbin/system-config-network-tui
もしくは
# /usr/sbin/system-config-network-gui
```

設定を変更した後に[OK] ボタンを押すと、変更をファイルに書き込みます。[OK] ボタンを押さずに[取り消し] ボタンを押すと、変更を書き込まずに終了します。設定を変更した場合には、システムを再起動して設定内容を反映してください。

6.3.2 設定ファイル

ネットワークの設定に関連するファイルには次のようなものがあります。

➢ /etc/sysconfig/network

このファイルには、接続するネットワークに関する定義を記述します。設定内容の例を次に示します。

```
NETWORKING=yes
NETWORK_IPV6=no
HOSTNAME=host1.your.domain.name
DOMAINNAME=your.domain.name
GATEWAY=192.168.0.1
GATEWAYDEV=eth0
```

各変数の意味は次のとおりです。

- NETWORKING —— ネットワークを使用するかどうか（yes／no）
- NETWORKING_IPV6 —— IPv6 ネットワークを使用するかどうか（yes／no）
- HOSTNAME —— このシステムのホスト名

- DOMAINNAME —— ネットワークのドメイン名
- GATEWAY —— ゲートウェイマシンの IP アドレス
- GATEWAYDEV —— ネットワークインターフェイス名

➢ /etc/sysconfig/network-scripts/ifcfg-eth0

このファイルには、そのシステムのネットワークインターフェイスに関する定義を記述します。**eth0** は 1 つ目のネットワークインターフェイスを指します。2 枚のイーサネットカードが装着されている場合には、2 枚目のネットワークインターフェイスは **eth1** になります。このファイルの設定内容の例を次に示します。

```
DEVICE=eth0
BOOTPROTO=static
BROADCAST=192.168.0.255
HWADDR=xx:xx:xx:xx:xx:xx
IPADDR=192.168.0.2
IPV6ADDR=
IPV6PREFIX=
NETMASK=255.255.255.0
NETWORK=192.168.0.0
ONBOOT=yes
```

各変数の意味は以下のとおりです。

- DEVICE —— ネットワークインターフェイス名
- BOOTPROTO —— IP アドレスの割り当て方の設定。BOOTP 、DHCP 、static などを記述可能
- BROADCAST —— ブロードキャストアドレス
- HWADDR —— インターフェイスの MAC アドレス
- IPADDR —— このシステムの IP アドレス
- IPV6ADDR —— IPv6 の IP アドレス（無効にしている場合は何も書かれていません）
- IPV6PREFIX —— IPv6 のネットマスク
- NETMASK —— ネットマスク
- NETWORK —— このシステムが属するネットワーク
- ONBOOT —— 起動時にネットワークインターフェイスを有効にするかどうか（yes／no）

➢ /etc/hosts

このファイルには、ネットワーク内のシステムの IP アドレスとホスト名の対応を記述します。このファイルの設定内容の例を次に示します。

```
192.168.0.197 host2.your.domain.name host2
127.0.0.1 localhost.localdomain localhost
```

➢ /etc/resolve.conf

このファイルには、ホスト名から IP アドレスを調べるために利用するネームサーバーの IP アドレスや、ホストを探すためのドメイン名などを記述します。このファイルの設定内容の例を次に示します。

```
domain your.domain.name
search your.domain.name
nameserver 192.168.1.11
```

domain 行には、接続している LAN のローカルドメイン名を、search 行にはホスト名を調べるために使うドメイン名を記述します。ネームサーバーが複数あるときには、nameserver 行を 3 つまで記述できます。

6.4 ネットワークの状況の確認

本節では、ネットワークの設定や状況の確認を行うためのコマンドをいくつか紹介します。

6.4.1 ifconfig

ifconfigコマンドはネットワークインターフェイスの起動、設定内容の確認などで使用されます。このコマンドを実行すると、以下のようにすべてのネットワークインターフェイスの設定内容、状態を確認できます。

```
# /sbin/ifconfig
eth0  Link encap:Ethernet HWaddr xx:xx:xx:xx:xx:xx
      inet addr:xxx.xxx.xxx.xxx Bcast:xxx.xxx.255.255 Mask:255.255.0.0.
      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
      RX packets:19910 errors:0 dropped:0 overruns:1 frame:0
      TX packets:819 errors:0 dropped:0 overruns:0 carrier:1
      collisions:0 txqueuelen:100
      Interrupt:11 Base address:0xdc00
lo    Link encap:Local Loopback
      inet addr:127.0.0.1 Mask:255.0.0.0.
      UP LOOPBACK RUNNING MTU:16436 Metric:1
      RX packets:10 errors:0 dropped:0 overruns:0 frame:0
      TX packets:10 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:100
```

6.4.2 netstat

このコマンドは、ネットワークシステムに関する多くの情報を表示できます。実行例を次に示します。**-er** オプションは、IP 経路テーブルを表示します。

```
# /bin/netstat -er
Kernel IP routing table
Destination     Gateway       Genmask       Flags Metric Ref Use Iface
xxx.xxx.xxx.xxx *             255.255.255.0 U     0      0     0 eth0
127.0.0.0       *             255.0.0.0     U     0      0     0 lo
default         192.168.100.1 0.0.0.0       UG    0      0     0 eth0
```

-ei オプションは、ネットワークインターフェイスの設定を **ifconfig** と同様に表示します。

```
# /bin/netstat -ei
```

```
eth0  Link encap:Ethernet HWaddr xx:xx:xx:xx:xx:xx
      inet addr:xxx.xxx.xxx.xxx Bcast:xxx.xxx.xxx.255 Mask:255.255.255.0.
      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
      RX packets:19910 errors:0 dropped:0 overruns:1 frame:0
      TX packets:819 errors:0 dropped:0 overruns:0 carrier:1
      collisions:0 txqueuelen:100
      Interrupt:11 Base address:0xdc00
lo    Link encap:Local Loopback
      inet addr:127.0.0.1 Mask:255.0.0.0.
      UP LOOPBACK RUNNING MTU:16436 Metric:1
      RX packets:10 errors:0 dropped:0 overruns:0 frame:0
      TX packets:10 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:100
```

6.4.3 ping

このコマンドは、ネットワーク上のホストへパケットを送信し、通信が行われていることを確認するものです。実行例を次に示します。

```
# /bin/ping 192.168.2.1
```

このコマンドは、正常に接続されているときには、パケット通信の状況を標準出力に表示します。引数にはホスト名を指定することもできます。

6.5 ボンディングインターフェイスの設定

ネットワークの冗長化を行う方法に、ボンディングインターフェイスを利用する方法があります。ここではそのボンディングインターフェイスを使い、アクティブ－バックアップ構成のネットワーク設定について紹介します。この節では、2 つのネットワークインターフェイスをボンディング化する方法を説明しますが、3 枚以上での構成も可能です。

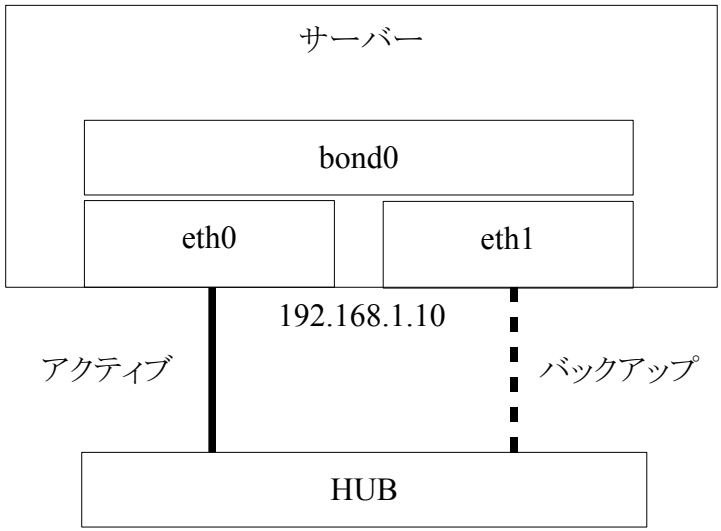


図 6-1　ネットワークの構成例

6.5.1 設定ファイル

➢ /etc/modprobe.conf

/etc/modprobe.conf ファイルに bonding ドライバが自動的にロードするための設定を行います。

```
alias bond0 bonding
```

➢ /etc/sysconfig/network-scripts/ifcfg-＊

- **bond0 の設定（ifcfg-bond0）**

  ifcfg-bond0 では、bonding デバイスを作成するときのオプション、およびネットワーク設定を記述します。

```
DEVICE=bond0
BOOTPROTO=none
BONDING_OPTS="mode=1 miimon=100"
BROADCAST=192.168.1.255
IPADDR=192.168.1.10
NETMASK=255.255.255.0
NETWORK=192.168.1.0
ONBOOT=yes
```

各パラメータの意味は次のとおりです。

bond0 ——ボンディングインターフェイス名

miimon —— MIIリンク監視を行う間隔(ミリ秒単位)

mode —— ボンディングモード(1:アクティブ－バックアップ、0:ラウンドロビンなど、全7種類)

MIIリンクによる監視の換わりにARPモニタリングを使用するには、BONDING_OPTSを以下のようにします。

```
BONDING_OPTS=”mode=1 arp_interval=100 arp_ip_target=+XXX.XXX.XXX.XXX”
```

各パラメータの意味は次のとおりです。

arp_interval —— ARPモニタリングを行う間隔(ミリ秒単位)

arp_ip_target —— ARPリクエストを送るターゲットのIPアドレス

- **eth0の設定(ifcfg-eth0)**

  ネットワーク設定の大半はbond0に記述したので、eth0およびeth1では、残りのカード固有の設定及びボンディング特有の設定を記述します。

```
DEVICE=eth0
BOOTPROTO=none
HWADDR=xx:xx:xx:xx:xx:xx
ONBOOT=yes
MASTER=bond0
SLAVE=yes
```

各パラメータの意味は次のとおりです。

MASTER ——結合されるボンディングインターフェイス名

SLAVE —— ボンディングインターフェイスで制御されるかどうか(yes／no)

- **eth1 の設定（ifcfg-eth1）**

```
DEVICE=eth1
BOOTPROTO=none
HWADDR=XX:XX:XX:XX:XX:XX
ONBOOT=yes
MASTER=bond0
SLAVE=yes
```

設定が終わったら、次のコマンドを実行してネットワークを再起動し、システムに反映させます。

```
# /sbin/service network restart
```

6.5.2 設定確認

ボンディングインターフェイスの動作状況は**/proc/net/bonding/bond***で確認することができます。

```
# /bin/cat /proc/net/bonding/bond0
Ethernet Channel Bonding Driver: v3.0.3 (March 23, 2006)

Bonding Mode: fault-tolerance (active-backup)
Primary Slave: None
Currently Active Slave: eth0
MII Status: up
MII Polling Interval (ms): 100
Up Delay (ms): 0
Down Delay (ms): 0

Slave Interface: eth0
MII Status: up
Link Failure Count: 0
Permanent HW addr: 00:40:26:97:15:ab

Slave Interface: eth1
MII Status: up
Link Failure Count: 0
Permanent HW addr: 00:90:27:3c:82:ff
```

ネットワークインターフェイスの動作状況は **ifconfig** コマンドで確認することができます。

```
# /sbin/ifconfig
bond0   Link encap:Ethernet  HWaddr 00:0C:29:01:65:4B
        inet addr:192.168.1.10  Bcast:192.168.1.255  Mask:255.255.255.0
        inet6 addr: fe80::240:26ff:fe97:15ab/64 Scope:Link
        UP BROADCAST RUNNING MASTER MULTICAST  MTU:1500  Metric:1
        RX packets:23698 errors:0 dropped:0 overruns:0 frame:0
        TX packets:31143 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:2667097 (2.5 MiB)  TX bytes:3996717 (3.8 MiB)

eth0    Link encap:Ethernet  HWaddr 00:0C:29:01:65:4B
        inet addr:192.168.1.10  Bcast:192.168.1.255  Mask:255.255.255.0
        inet6 addr: fe80::240:26ff:fe97:15ab/64 Scope:Link
        UP BROADCAST RUNNING SLAVE MULTICAST  MTU:1500  Metric:1
        RX packets:23699 errors:0 dropped:0 overruns:0 frame:0
        TX packets:31149 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:2667525 (2.5 MiB)  TX bytes:3997925 (3.8 MiB)
        Interrupt:10 Base address:0x1080

eth1    Link encap:Ethernet  HWaddr 00:0C:29:01:65:4B
        inet addr:192.168.1.10  Bcast:192.168.1.255  Mask:255.255.255.0
        inet6 addr: fe80::240:26ff:fe97:15ab/64 Scope:Link
        UP BROADCAST RUNNING NOARP SLAVE MULTICAST  MTU:1500  Metric:1
        RX packets:8 errors:0 dropped:0 overruns:0 frame:0
        TX packets:3 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:480 (480.0 b)  TX bytes:210 (210.0 b)
        Interrupt:9 Base address:0x1400
```

6.5.3 複数のボンディングインターフェースの設定

複数のボンディングインターフェース(bond1,bond2...)を設定するには、次のようにファイルを追加・修正後、ネットワークを再起動します。

- **modprobe.conf** —— **alias bond1 bonding** などと、必要な分だけ定義を追加します。
- **ifcfg-bondX** —— X の部分を 1 または 2 として、必要な分だけファイルを追加します。
- **ifcfg-ethX** —— eth3、eth4...とファイルを必要なだけ作成します。その時、どのネットワークカードとどのネットワークカードを組で利用するかを HWADDR パラメタで明確に指定します。

6.6 ジャンボフレームの設定

従来、イーサネットのデータ転送で 1 度に転送できるフレームサイズは 1,518 バイトと定められていましたが、100Mbps や 1Gbps のイーサネット規格が普及し、1,518 バイトでは転送効率が悪くなりました。そこで、1 度に転送できるフレームサイズを拡張したのがジャンボフレームです。

ジャンボフレームの設定を行う前に、**/etc/sysconfig/network-scripts/ifcfg-eth0**（eth0 ではない場合は他のネットワークインターフェイス名）をエディタで開き、次の 1 行を追加もしくは編集します。

```
MTU=9000
```

上記の例ではフレームサイズを 9,000 バイトに設定していますが、デバイスによってフレームサイズの設定上限値が異なるため、事前にデバイスのフレームサイズ上限値を調べておくと良いでしょう。

設定した内容を反映させるには、network サービスを再起動します。

```
# /sbin/service network restart
```

また、一時的にフレームサイズの変更を行う場合は、ifconfig コマンドを使用します。この方法で設定した場合、再起動すると設定が変更前に戻ります。

```
# /sbin/ifconfig eth0 mtu 9000
```

第7章　プリンタの管理

この章で説明する内容

目的	プリンタを管理する方法について理解する
機能	ドキュメントをプリンタに出力する
必要な RPM	cups —— プリンタ管理ツール a2ps —— ポストスクリプトコンバーター mpage —— 複数ページ印刷ツール
設定ファイル	/etc/cups/classes.conf　　/etc/cups/printers.conf /etc/cups/cupsd.conf　　/etc/cups/client.conf
章の流れ	1 プリンタ管理の概要　　4 設定項目の詳細 2 プリンタデーモンの起動と停止　　5 ドキュメントの印刷 3 プリンタデバイスの設定
関連 URL	**Common UNIX Printing System** http://www.cups.org/ **OpenPrinting - The Linux Foundation** http://www.linux-foundation.org/en/OpenPrinting **Linux Printing Usage HOWTO** http://www.linux.or.jp/JF/JFdocs/Printing-Usage-HOWTO.html **OPFC プロジェクト**　http://opfc.sourceforge.jp/

7.1 プリンタ管理の概要

Asianux Server 3 では、プリンタシステムとして、CUPS（Common UNIX Printing System）を採用しています。CUPS は、従来の UNIX で採用されていた LPD（Line Printer Daemon）システムと比べ、柔軟な設定が可能となっています。本章では、CUPS を使用してドキュメントを印刷するための設定等について説明します。

7.2 プリンタデーモンの起動と停止

プリンタから印刷するには、**cupsd** デーモンをあらかじめ起動しておく必要があります。**cupsd** の起動／停止スクリプトは、**/etc/rc.d/init.d/cups** です。起動スクリプトのオプションでは、起動（start）、停止（stop）、再起動（restart）、または現在の状況を確認（status）できます。

- **cupsd** を起動するには、次のコマンドを実行します。

```
# /sbin/service cups start
```

- **cupsd** を停止するには、次のコマンドを実行します。

```
# /sbin/service cups stop
```

- **cupsd** を再起動するには、次のコマンドを実行します。

```
# /sbin/service cups restart
```

- **cupsd** の状態を確認するには、次のコマンドを実行します。

```
# /sbin/service cups status
```

また、**chkconfig** を使用することで、**cupsd** サーバーをマシン起動時に自動的に立ち上げるか、立ち上げないかを選択できます。

- 現在の設定を確認するには、次のコマンドを実行します。

```
# /sbin/chkconfig --list cups
```

- サーバーマシンの起動時に **cupsd** サーバーを立ち上げるようにするには、次のコマンドを実行します。

```
# /sbin/chkconfig cups on
```

- サーバーマシンの起動時に **cupsd** を立ち上げないようにするには、次のコマンドを実行します。

```
# /sbin/chkconfig cups off
```

7.3　プリンタデバイスの設定

Asianux Server 3 では、KDE の「プリンタ構成ツール」でプリンタの設定を行うことができます（図 7-1）。プリンタ構成ツールを使用するには、cupsd が起動していることを確認してから、デスクトップの左下にある[Start]ボタンをクリックし、メニューから[設定]-[ハードウェア]-[プリンタ]を選択します。root ユーザー以外でこのツールを使う場合、画面下の[管理者モード(M)...]ボタンをクリックし、root ユーザーのパスワードを入力します（図 7-2）。

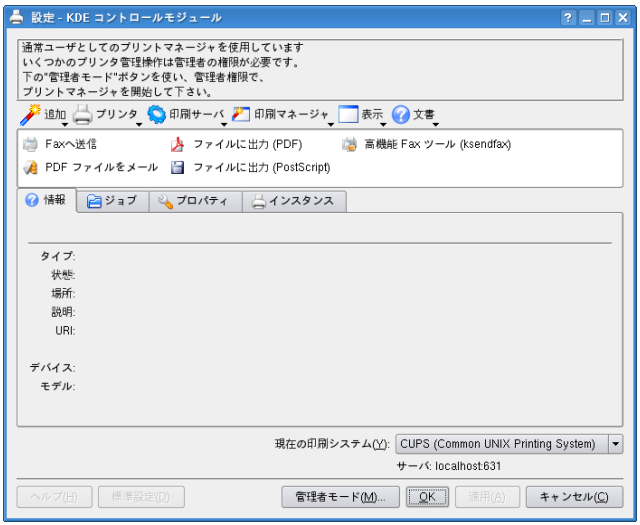


図 7-1　プリンタ構成ツール画面

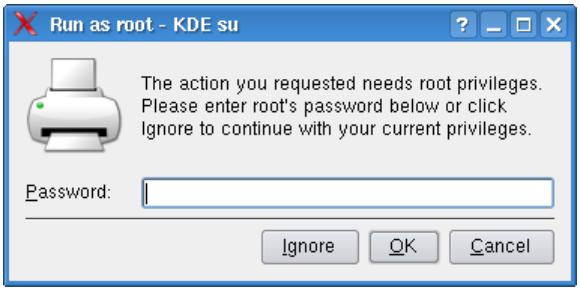


図 7-2　root パスワード要求画面

新たにプリンタを追加するには、図 7-1 の画面左上[追加]ボタンをクリックし、[プリンタ/クラスを追加(P)]を選択します。

最初に、表示されるウィザード初期画面（図 7-3）の下部にある[次(N)]をクリックすると、図 7-4 のような画面が表示されます。ここでは対象となるプリンタの種類を選択します。以下の例ではネットワークプリンタを導入するので、[ネットワークプリンタ(TCP)(T)]にチェックを入れて[次(N)]をクリックします。

図 7-3 ウィザード初期画面

図 7-4 プリンタ追加ウィザード（1）

次に、接続先のプリンタのアドレスを入力します（図 7-5）。この例では、対象アドレスとして[remotehost]を指定しています。

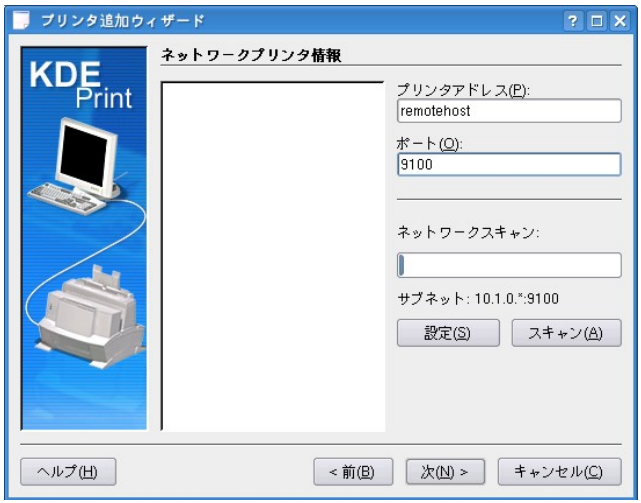


図 7-5　プリンタ追加ウィザード（2）

プリンタの接続先を指定したら、そのプリンタのメーカー名（製造者）と型番（モデル）を選択します（図 7-6）。この情報を元にインストールされるドライバが決定されます。もし該当するエントリが存在しないポストスクリプトプリンタの場合は、[製造者]から「Generic」を選び、[モデル]では「Postscript Printer」を選びます。

図 7-6　プリンタ追加ウィザード（3）

ウィザードが選択したドライバを適当だと判断すると、図 7-7 の画面が表示されるので、問題がなければ[次(N)]をクリックします。

図 7-7 プリンタ追加ウィザード(4)

次に、バナーの挿入を選択します。必要に応じてバナーを選択し、[次(N)]をクリックします。

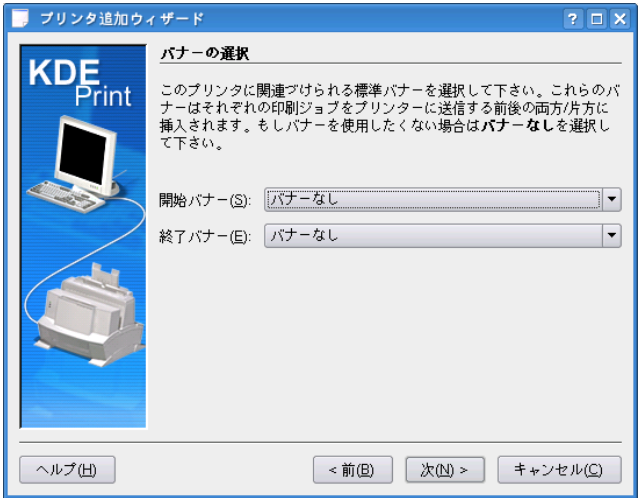


図 7-8 プリンタ追加ウィザード(5)

次に、プリンタの利用制限を設定するクォータ設定を行います。期間や、サイズ、ページの制限をすることができます。必要に応じて設定できたら、[次(N)]をクリックします。

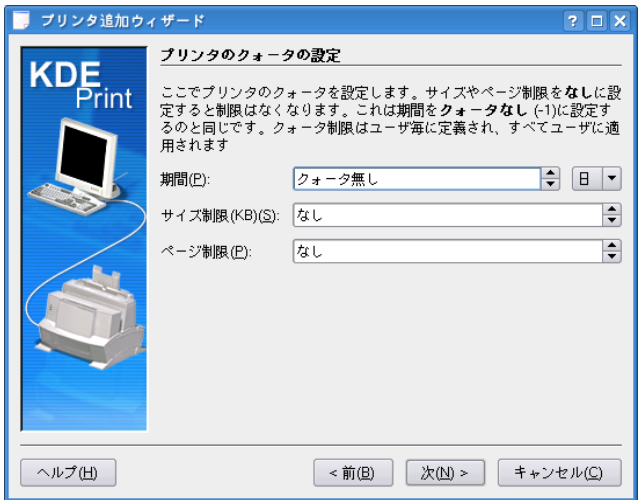


図 7-9　プリンタ追加ウィザード（6）

次に、ユーザーアカウントの設定を行います。タイプは「許可されたユーザー」と「拒否されたユーザー」から選択することができます。必要に応じて設定したら、[次(N)]をクリックします。

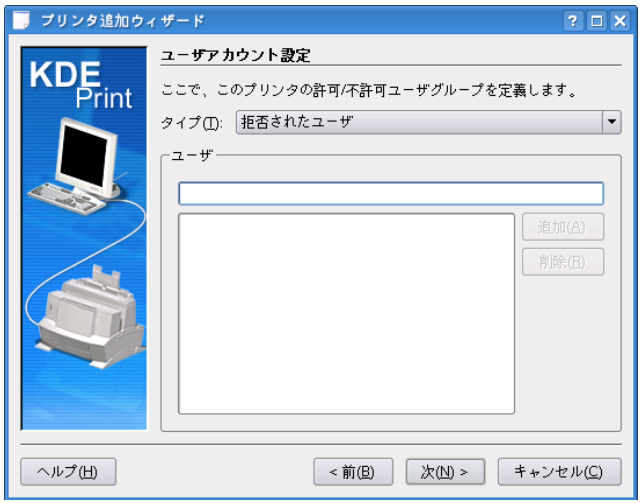


図 7-10　プリンタ追加ウィザード（7）

最後に、設定したプリンタに名前を付けます（図 7-11）。[名前]以外のフィールドは空欄でも問題ありません。[次(N)]をクリックすると、設定内容の確認画面が表示されます（図 7-12）。

図 7-11 プリンタ追加ウィザード（8）

図 7-12 プリンタ追加ウィザード（9）

設定内容に問題がなければ[完了(E)]をクリックすると、プリンタが追加されます（図 7-13）。

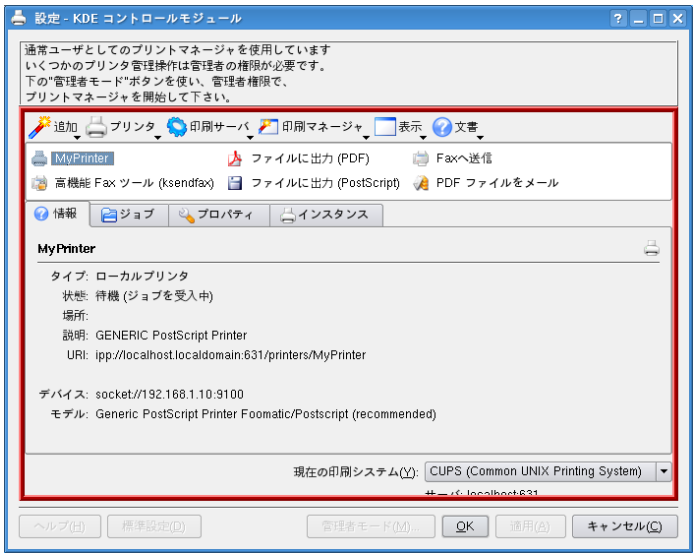


図 7-13　プリンタ構成ツール画面

7.4 設定項目の詳細

導入されたプリンタは、プリンタ設定によって様々なオプション項目の変更が可能です。

プリンタ構成ツールで追加されたプリンタのアイコンを選択し、マウスの右ボタンをクリックしメニューの中から「設定」を選択すると、次のようなプリンタ設定画面（図 7-14）が表示されます。

この画面を利用することで、用紙サイズ、印刷方向等の印刷方法について詳細な設定を行うことができます。

図 7-14 プリンタ設定画面

7.5 ドキュメントの印刷

Asianux Server 3 付属のほとんどの GUI ソフトウェアには印刷機能があり、[印刷]メニューから表示中のドキュメントの印刷を行うことができますが、コマンドラインから直接印刷したいファイルを指定して印刷することもできます。

コマンドラインからドキュメントを印刷するには、**lpr** コマンドを使います。<file> には基本的にポストスクリプト形式のファイルを指定します。ただし、Asianux Server 3 ではテキストファイルや代表的な画像ファイルはポストスクリプト形式に自動的に変換するため、これらも指定することができます。

```
$ /usr/bin/lpr <file>
```

lpr で印刷する時に**-o** でプリンタのオプションを指定することもできます。次のコマンド実行例では、プリンタに「MyPrinter」、解像度は 1200dpi、プリンタのトレイ 1 を使用するように指定しています。(なお、オプションを指定しないときの初期値を変更するには **lpoptions** コマンドを使います。)

```
$ /usr/bin/lpr -P MyPrinter -o Resolution=1200 -o InputSlot=Tray1 <file>
```

テキストファイルを印刷するときに、自動変換を使わず明示的にポストスクリプト形式に変換してから印刷する方法もあります。**a2ps** や **ghostscript** が導入されている場合は、次のコマンドを実行することで印刷することができます。

```
$ /usr/bin/a2ps <file> | /usr/bin/lpr
```

a2ps コマンドの代わりに、**mpage** コマンドを使用することもできます。特に複数のページを 1 枚に印刷したい場合に便利です。次のコマンド実行例では、2 ページ分を 1 枚にまとめて印刷しています。

```
$ /usr/bin/mpage -2 <file> | /usr/bin/lpr
```

第8章　DNS サーバーの構築

この章で説明する内容

目的	DNS サーバーを構築する方法について理解する
機能	ネットワーク上のホスト間の名前解決
必要な RPM	bind —— DNS サーバー本体 caching-nameserver —— キャッシングネームサーバー用設定ファイル bind-chroot —— chroot 環境ファイル
設定ファイル	/var/named/chroot/etc/named.caching-nameserver.conf /var/named/chroot/etc/named.conf /etc/resolv.conf /etc/rndc.conf
章の流れ	1 DNS サーバーの概要 2 DNS サーバーの起動と停止 3 名前解決のしくみ 4 DNS サーバーの種類と設定 5 RNDC 6 DNS サーバーのテスト
関連 URL	**Internet Systems Consortium - BIND** http://www.isc.org/products/BIND/ **DNS HOWTO** http://www.linux.or.jp/JF/JFdocs/DNS-HOWTO.html

8.1 DNS サーバーの概要

DNS（Domain Name System）サーバーは、ホスト情報を分散データベースによって提供するしくみです。DNS サーバーが提供する機能には、ホスト名を元に IP アドレスを検索したり、IP アドレスを元にホスト名を検索したりする機能があります。

Asianux Server 3 では DNS 機能の代表的な実装である、**BIND**（Berkeley Internet Name Domain）を採用しています。本章では、この BIND を使って DNS サーバーを構築する方法について説明します。

8.2 DNS サーバーの起動と停止

DNS サーバー（BIND）を使用するには、BIND の実体であるデーモンの **named** を起動する必要があります。**named** の起動／停止スクリプトは、**/etc/rc.d/init.d/named** です。起動スクリプトのオプションでは、起動（start）、停止（stop）、再起動（restart）、または現在の状況を確認（status）を指定できます。

- DNS サーバーを起動するには次のコマンドを実行します。

```
# /sbin/service named start
```

- DNS サーバーを停止するには、次のコマンドを実行します。

```
# /sbin/service named stop
```

- DNS サーバーを再起動するには、次のコマンドを実行します。

```
# /sbin/service named restart
```

- DNS サーバーの現在の状況を確認するには、次のコマンドを実行します。

```
# /sbin/service named status
```

8.3 名前解決のしくみ

あるホストが、自分自身もしくは他のホストの名前・IP アドレスの検索を行うには、次の 3 つの方法があります。

- **/etc/hosts** ファイルによる解決
- DNS サーバーへの問い合わせ
- NIS ドメインサーバーによる解決

/etc/hosts ファイルによる解決では、すべてのホストで同じ **hosts** ファイルを保持する必要があり、大規模なネットワークでは特に維持管理が困難です。複数台のマシンがある環境では、できる限り DNS を導入することを推奨します。

8.3.1 リゾルバ

クライアントが名前解決を行うには、DNS サーバーに検索の実行を要求して、その結果を受け取るクライアントプログラムが必要です。このクライアントプログラムは、「リゾルバ」と呼ばれ、次の 2 つの設定ファイルを必要とします。

- **/etc/host.conf**
- **/etc/resolv.conf**

➢ /etc/host.conf

/etc/host.conf は、名前解決の順番を設定するファイルです。このファイルでは、DNS サーバーによる解決、NIS ドメインサーバーによる解決、**/etc/hosts** ファイルによる解決の 3 つの方法を記述でき、記述した順番によって名前解決を行います。

Asianux Server 3 のデフォルトでは、**/etc/host.conf** は以下の内容になっています。

```
order hosts,bind
```

「order」に続けて名前解決の方法を記述することで、順番を指定できます。上記の例では、最初に **hosts** ファイルによる名前解決を試み、解決できなかった場合に DNS サーバー（BIND）を利用するという内容の設定となっています。

➢ /etc/resolv.conf

/etc/resolv.conf は、ドメイン名や DNS サーバーについて記述しています。

```
domain your.domain.name
nameserver 192.168.1.2
```

- **domain**
  サーバーが属しているドメイン名を指定します。最後にドット(.)の付いたドメイン名の形式で問い合わせを行った場合に、ここに記述したドメイン内でホスト名を探します。

- **search**
  リゾルバが検索するドメインの一覧を定義します。複数個のドメインを指定でき、ドメイン形式で問い合わせが行われなかった場合には、ここに記述した順序で各ドメインに問い合わせを実行します。
  domain と search の両方を指定することはできません。domain と search の両方を指定した場合には、最後に指定したほうが有効となります。

- **nameserver**
  使用する DNS サーバーを IP アドレスで指定します。**/etc/resolv.conf** には、最大で nameserver を 3 つまで指定できます。

8.4 DNS サーバーの種類と設定

8.4.1 DNS サーバーの種類

DNS サーバーは、役割・機能によって次の 3 種類に分けられます。

- **キャッシュオンリーサーバー**

  キャッシュオンリーサーバーは、自身で名前解決を行わず、クライアントから問い合わせがあると、指定された DNS サーバーに問い合わせを転送します。この際、一度行われた問い合わせを、一定期間キャッシュします。このため、次回からの問い合わせではキャッシュを参照することにより、名前解決にかかる時間を短縮し、また、マスター／スレーブサーバー（後述）の負荷を軽減する働きがあります。

- **マスターサーバー（プライマリネームサーバー）**

  ドメイン内にあるすべてのホスト名情報の管理、メールサーバーへの転送経路の確保、スレーブサーバーへのドメインネーム情報の提供、他のドメインのネームサーバーとの情報交換などの機能を有する重要なサーバーです。

- **スレーブサーバー（セカンダリネームサーバー）**

  定期的にマスターサーバーからデータをコピーして、マスターサーバーと同様に機能します。ドメイン申請が承認されるためには、マスターサーバー以外に、スレーブサーバーが 1 台以上必要となります。

8.4.2 設定ファイルについて

Asianux Server 3 では、named の各設定ファイルは**/var/named/chroot** 配下にあります。(**bind-chroot** パッケージがインストールされている必要があります。) 特に指定が無い場合はこのディレクトリ下のファイルを編集します。

デフォルトでは、キャッシュオンリーサーバー用に **named.caching-nameserver.conf** のみ用意されており、DNS サーバー起動時にこの設定ファイルが使用されます。

キャッシュオンリーサーバーではなく、マスターサーバーやスレーブサーバーとして運用したい場合には、**/var/named/chroot/etc/named.conf** を作成し、設定を記述します。

named.caching-nameserver.conf と **named.conf** が混在している場合、**named.conf** が優先して読み込まれます。

➢ 設定ファイルの書式

BIND の設定ファイルには、ステートメントごとに、以下のような書式で記述します。

1 つのステートメントは、{...} で括られており、最後にはすべて ;(セミコロン)を記述します。そしてこのステートメント内にサブステートメントを列挙して記述していきます。

```
<ステートメント> [<パラメータ>] {
      <サブステートメント> [<パラメータ>]{
      };
};
```

8.4.3 キャッシュオンリーサーバーの設定

DNS サーバーをキャッシュオンリーサーバーとして動作させるには、**named.caching-nameserver.conf** をエディタで開き、必要に応じて編集します。

Asianux Server 3 では、デフォルトで以下のような設定になっています。

```
options {
        listen-on port 53 { 127.0.0.1; };
        listen-on-v6 port 53 { ::1; };
        directory       "/var/named";
        dump-file       "/var/named/data/cache_dump.db";
        statistics-file "/var/named/data/named_stats.txt";
        memstatistics-file "/var/named/data/named_mem_stats.txt";
        query-source    port 53;
        query-source-v6 port 53;
        allow-query     { localhost; };
};
logging {
        channel default_debug {
                file "data/named.run";
                severity dynamic;
        };
};
view localhost_resolver {
        match-clients      { localhost; };
        match-destinations { localhost; };
        recursion yes;
```

- **options ステートメント**

  options ステートメントには、DNS サーバ全体の動作に関する設定を記述します。

いくつかのサブステートメントについて

listen-on サブステートメントには、named が問い合わせを受け付けるインタフェースとポートを指定します。

directory サブステートメントには、DNS データベース用のファイルを格納するディレクトリを設定します。デフォルトでは/var/named が設定されていますが、chroot 環境では、ディレクトリの実体は、/var/named/chroot/var/named です。

allow-query サブステートメントには、問い合わせを許可するホストを指定します。allow-query は、zone ステートメントの中でも指定できます。その場合は、options ステートメント内の allow-query よりも優先されます。指定がない場合は、デフォルトですべてのホストからの問い合わせを許可します。

- **logging ステートメント**

  logging ステートメントには、named のログ出力に関する指定をします。

  デフォルトでは、サーバのデバッグレベルに応じたログが、/var/named/chroot/var/named/data/named.run ファイルに出力されます。

- **view ステートメント**

  view ステートメントでは、問合せ元に応じて異なる名前解決を行うことができます。例えば外部向けと内部向けのネームサーバを 1 台で構築することができます。

  match-clients----発信元 IP アドレスで振り分け

  match-destinations----宛先 IP アドレスで振り分け

  デフォルトの view ステートメントは、キャッシングネームサーバとして動作する設定になっています。

named.caching-nameserver.conf ファイルの記述方法は、named.conf と同様です。詳細な設定については、「**man 5 named.conf**」および **usr/share/doc/bind-9.3.3/**ディレクトリ以下のドキュメントを参照してください。

8.4.4 マスターサーバー（プライマリネームサーバー）の設定

マスターサーバーを設定するのに必要なファイルは以下の 6 ファイルです。

- named 設定ファイル —— **/var/named/chroot/etc/named.conf**
- リゾルバファイル —— **/etc/resolv.conf**
- キャッシュファイル —— **/var/named/chroot/var/named/named.ca**
- ループバックファイル —— **/var/named/chroot/var/named/localhost.zone**
- 正引きファイル（ゾーンファイル） —— **/var/named/chroot/var/named/your-domain.zone**

- 逆引きファイル（リバースファイル） —— **/var/named/chroot/var/named/your-domain.rev**

キャッシュファイル、ループバックファイル、正引きファイル、逆引きファイルの 4 つは任意の名前を付けることができます。これらのファイル名は、**named.conf** で指定します。

➢ ネットワークの条件等

本節では、以下のような条件で設定を行っています。

- **ネットワーク**

  IP アドレス範囲 —— 192.168.1.1 ～ 192.168.1.255

  サブネットマスク —— 255.255.255.0

  ドメイン名 —— your.domain.name

  スレーブサーバー —— secondary.name.server（172.16.1.2）

- **IP アドレス割り当て**

IP アドレス	ホスト名	用途
192.168.1.0	－	ネットワークアドレス
192.168.1.1	－	デフォルトゲートウェイ（ルーター）
192.168.1.2	ns.your.domian.name	DNS サーバー（マスター）
192.168.1.3	host1.your.domain.name	通常のホスト
192.168.1.255	－	ブロードキャスト

- **その他**

  ホスト名 www.your.domain.name、mail.your.domain.name、ftp.your.domain.name を、ns.your.domain.name（192.168.1.2）の別名として設定します。

➢ named.conf の設定

Asianux Server 3 では、デフォルトでは、キャッシュオンリーサーバー用に named.caching-nameserver.conf のみ用意されており、named.conf は用意されていません。

マスターサーバーの設定を行う場合には、/var/named/chroot/etc/named.conf を作成し、設定を記述する必要があります。以下は named.conf ファイルの記述例です。

```
options {
      directory "/var/named";
};
```

```
controls {
inet 127.0.0.1 allow { localhost; } keys { rndckey; };
};

include "/etc/named.rfc1912.zones";

zone "your.domain.name" IN {
      type master;
      file "your-domain.zone";
};

zone "1.168.192.in-addr.arpa" IN {
      type master;
      file "your-domain.rev";
};

include "/etc/rndc.key";
```

上記の設定では、view ステートメントを使用していませんので、すべての zone は"default"view に含まれ、すべてのクライアントがマッチします。

named.conf に view ステートメントを記述する場合は、すべての zone はいずれかの view に属する必要があります。その場合、zone ステートメントは、view ステートメントの内部に記述してください。

- **include** "ファイル名";
  include ステートメントは、指定のファイルを挿入します。
  上記の例では、etc/named.rfc1912.zones を挿入しています。

- **zone "your.domain.name"**
  ゾーン名としてドメイン名を指定します。さらに、そのドメインで使用される正引き DNS データベースファイルの名称を指定します。

- **zone "1.168.192.in-addr.arpa"**
  ゾーン名として、使用を許されたネットワークの逆引き名を指定します。さらに、そのゾーンで使用される逆引き DNS データベースファイルの名称を指定します。

詳細な設定については、「**man 5 named.conf**」および **usr/share/doc/bind-バージョン/**ディレクトリ以下のドキュメントを参照してください。

➢ DNS ゾーンデータベースファイルの設定

ループバックファイル、正引きファイル、逆引きファイルは、DNS ゾーンデータベースファイルです。ここでは各ファイルで共通に使用する設定項目について説明します。

- **$TTL** … Time To Live、キャッシュの有効期限
- **SOA** … Start Of Authority、ゾーンに対する管理情報を設定します。
- **IN** … InterNet、インターネットレコードを表します。
- **NS** … NameServer、ネームサーバー。ホスト名＋ドメイン名の形式で記述して、最後にピリオドを入力します。
- **A** … Address、アドレスレコード。ホスト名から IP アドレスへの変換時に使用します。
- **PTR** … PoinTer Record、ポインタレコード。IP アドレスからホスト名への変換時に使用します。
- **MX** … Mail eXchanger、メールエクスチェンジャ。どのホストが外部からのメールを受信するかを記述します。"MX" の後に数字を書き、複数ホストを設定したときの優先順位を指定します。
- **CNAME** … Canonical NAME、ホストの別名。特定のホストに、別名を付けます。

➢ 正引きファイル（/var/named/chroot/var/named/your-domain.zone）の設定

```
$TTL 86400
@          IN SOA ns.your.domain.name. root.your.domain.name. (
              2001092300 ; Serial
              10800 ; Refresh
              3600 ; Retry
              604800 ; Expire
              86400 ) ; Minimum
            IN NS ns.your.domain.name.
            IN NS secondary.name.server.
            IN MX 10 ns.your.domain.name.
localhost IN A 127.0.0.1
ns         IN A 192.168.1.2
host1      IN A 192.168.1.3
mail       IN CNAME ns
www        IN CNAME ns
ftp        IN CNAME ns
```

- **$TTL**

  ファイルの先頭に$TTL 行を記述します。単位は秒で指定するので、86400 秒は、24 時間を表します。

- **SOA**

  $TTL の次行の先頭に「@」を記述して書き始めます。最初の行には、SOA レコードが権限を持つマスターサーバーの名称と、ゾーンの管理者のメールアドレスを記述します。また、メールアドレスの「@」はピリオドに置き換えて記述します。

  かっこの中にある数値は、それぞれ次の項目の値をあらわします。

項目	説明
シリアル値	レコードを更新したら、この数値を上げます。2001092300 のように、年月日+2 桁の連番のように記述する方法が一般的です。
更新期間	スレーブサーバーに対しレコードが更新されたか確認する間隔を指定します。
リトライ間隔	スレーブサーバーが、マスターサーバーに接続できなかったときにリトライする間隔を指定します。
データが無効になるまでの期間	マスターサーバーに接続できない場合に、ここで設定した期間を過ぎるとゾーンのデータを破棄します。この値は 1 週間～ 2 週間と長めに設定します。
キャッシュ期間	キャッシュしたレコードを保持する期間を指定します。

- **NS レコード**

  ネームサーバーのホスト名を記述します。ホスト名の最後にはピリオドを入力します。

- **PTR レコード**

  IP アドレスに対応したホスト名を記述します。

- **A レコード**

  ホスト名に対する、IP アドレスを記述します。"localhost" に対する IP アドレスは"127.0.0.1" となります。

- **MX レコード**

  メールサーバーの優先順位を示す数値であるプリファレンス値と、メールサーバーのホスト名を設定します。プリファレンス値は符号なし 16bits 数値で設定し、小さいほど優先度が高いと判断されます。ここでは、優先度が比較的高い"10" を"ns.your.domain.name." に設定します。この結果、your.domain.name 宛てのメールが ns.your.domain.name に届くようになります。

  ※ MX レコードに設定するホスト名は、必ず A レコードで設定しているホスト名を指定してください。

- **CNAME レコード**

  ホストの別名を記述します。前述の例は、ns.your.domain.name の別名として以下の 3 つを設定しています。

  - www.your.domain.name
  - mail.your.domain.name
  - ftp.your.domain.name

この結果、ns.your.domain.name が www.your.domain.name 、
mail.your.domain.name、ftp.your.domain.name で名前解決できるようになります。

➢ 逆引きファイル（/var/named/chroot/var/named/your-domain.rev）の設定

```
$TTL 86400
@   IN SOA ns.your.domain.name. root.your.domain.name. (
            2001092300 ; Serial
            10800 ; Refresh
            3600 ; Retry
            604800 ; Expire
            86400 ) ; Minimum
    IN NS ns.your.domain.name.
    IN NS secondary.name.server.
    IN PTR your.domain.name.
    IN A 255.255.255.0
2   IN PTR ns.your.domain.name.
3   IN PTR host1.your.domain.name.
```

IN PTR your.domain.name. でドメイン名の対応付けを行います。

IN A 255.255.255.0 では、サブネットマスクの設定を行います。

それ以降には、IP アドレスの末尾を記述して、対応するホスト名を記述します。上記の例では、次に示す記述がホスト名の対応付け設定となります。

2 IN PTR ns.your.domain.name.

3 IN PTR host1.your.domain.name.

以上の設定を行い、DNS サーバーを起動、もしくは再起動します。

8.4.5 スレーブサーバー（セカンダリネームサーバー）の設定

DNS サーバーをスレーブサーバーとして動作させるためには、**named.conf** を次のように設定します（マスターサーバーを 192.168.1.2 としている場合の設定）。

```
options {
directory "/var/named";
};
controls {
inet 127.0.0.1 allow { localhost; } keys { rndckey; };
};

include "/etc/named.rfc1912.zones";

zone "your.domain.name" IN {
type slave;
file "slaves/your-domain.zone";
masters { 192.168.1.2; };
};

zone "1.168.192.in-addr.arpa" IN {
type slave;
file "slaves/your-domain.rev";
masters {192.168.1.2; };
};

include "/etc/rndc.key";
```

スレーブサーバーでは、zone ステートメント内で「**type slave;**」を指定します。これにより、IP アドレス 192.168.1.2 のプライマリサーバーからゾーンファイルとリバースファイルを取得して、ネームサービスを実施します。

8.5 RNDC

BIND4、BIND8 では、DNS サーバーの制御に NDC（Name Daemon Control）が使用されてきましたが、Asianux Server 3 で採用している BIND9 からは、RNDC（Remote Name Daemon Control）を使用するように変更されています。

RNDC は、従来、ファイルシステムソケットやシグナルを使って BIND と通信していた NDC と異なり、ネットワークを介して BIND と通信します。また、通信を安全に行うため、認証には暗号化鍵を使用します。

暗号化鍵の情報など、RNDC の設定は、**/etc/rndc.conf** で行っています。また、**/etc/rndc.conf** の中で指定している暗号化鍵は、**named.conf** の controls ステートメントで指定する鍵と共通のものです。Asianux Server 3 のデフォルトでは、named 側の暗号化鍵を**/etc/rndc.key** に用意しています。

➢ /etc/rndc.conf

RNDC の設定情報は、**/etc/rndc.conf** に記述します。**/etc/rndc.conf** は、**named.conf** のサブセットとなっており、options ステートメント、key ステートメント、server ステートメントが使用できます。

```
options {
        default-server  localhost;
        default-key     "rndckey";
};

server localhost {
        key     "rndckey";
};

include "/etc/rndc.key";
```

➢ /etc/rndc.key

/etc/rndc.key には named で使用する暗号化鍵の記述があり、**named.conf** の最後で取り込みます。

8.5.1 RNDC の確認

RNDC が正しく設定されているかを確認するには、以下のコマンドを入力します。

```
# /usr/sbin/rndc status
```

正しく設定されていれば、以下のようなメッセージが表示されます。表示されなかった場合は、設定ファイルの内容を確認してください。

```
number of zones: 3
debug level: 0
xfers running: 0
xfers deferred: 0
soa queries in progress: 0
query logging is OFF
recursive clients: 0/1000
tcp clients: 0/100
server is up and running
```

8.6 DNS サーバーのテスト

bind が正常に機能しているかどうかをテストするには、次のツールが便利です。

- **ping**
- **host**
- **nslookup**
- **dig**

8.6.1 ping によるテスト

ping ではネットワークが正常に機能しているか確認できます。

ping を、自ホスト、自ドメイン内の別ホスト、外部のネットワークのホストに対してそれぞれ実行して、反応が返ってくるかどうかチェックします。**ping** は途中で中断しない限り、相手のホストにパケットを送り続けます。[Ctrl] + [C] キーを入力して中断するか、**-c** オプションで送信回数を指定してください。

自ホスト(192.168.1.2)に対するテストと、ネットワークは正常に機能している場合の結果を次に示します。

```
# /bin/ping -c 5 192.168.1.2
PING 192.168.1.2 (192.168.1.2) 56(84) bytes of data.
64 bytes from 192.168.1.2: icmp_seq=0 ttl=64 time=0.035 ms
64 bytes from 192.168.1.2: icmp_seq=1 ttl=64 time=0.026 ms
64 bytes from 192.168.1.2: icmp_seq=2 ttl=64 time=0.028 ms

--- 192.168.1.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 0.026/0.029/0.035/0.007 ms
```

8.6.2 host によるテスト

host コマンドは、ホスト名から IP アドレス(正引き)、または IP アドレスからホスト名(逆引き)を確認するためのプログラムです。詳細な使い方については、「**man 1 host**」を参照してください。

- テスト実行例(正引き)

```
# host www.your.domain.name
www.your.domain.name is an alias for ns.your.domain.name.
ns.your.domain.name has address 192.168.1.2
```

- テスト実行例（逆引き）

```
# host 192.168.1.2
2.1.168.192.in-addr.arpa domain name pointer ns.your.domain.name.
```

8.6.3 nslookup によるテスト

nslookup コマンドでも、ゾーンデータベースの内容を確認することができます。

nslookup には対話モードと非対話モードの 2 つのモードがあります。詳細な使い方については、「**man 1 nslookup**」を参照してください。

nslookup コマンドを対話モードで実行する場合には、以下のように実行します。終了するには **exit** と入力します。

```
# /usr/bin/nslookup
> ホスト名又はIPアドレス
       |
       |
> exit
#
```

- 外部のネットワークに対するテスト実行例（正引き/逆引き）

```
> www.miraclelinux.com
Server:         192.168.1.2
Address:        192.168.1.2#53

Non-authoritative answer:
www.miraclelinux.com    canonical name = ns.miraclelinux.com.
Name:   ns.miraclelinux.com
Address: 219.118.163.66
> 219.118.163.66
Server:         192.168.1.2
Address:        192.168.1.2#53

Non-authoritative answer:
66.163.118.219.in-addr.arpa     canonical name = 66.64.163.118.219.in-addr.arpa.
66.64.163.118.219.in-addr.arpa  name = ns.miraclelinux.com.

Authoritative answers can be found from:
64.163.118.219.in-addr.arpa     nameserver = ns1.bit-drive.ne.jp.
```

```
64.163.118.219.in-addr.arpa     nameserver = ftp01.miraclelinux.com.
64.163.118.219.in-addr.arpa     nameserver = ns.miraclelinux.com.
ns.miraclelinux.com     internet address = 219.118.163.66
ns1.bit-drive.ne.jp     internet address = 211.9.32.227
```

8.6.4 dig によるテスト

dig コマンドは、nslookup と同様に名前解決のテストを行うことができ、より詳細なテストを実行することも可能です。詳細な使い方については、「**man 1 dig**」を参照してください。

以下のテストは、dig のシンプルな実行例です。

- 外部のネットワークに対するテスト(正引き)

```
# /usr/bin/dig www.miraclelinux.com
; <<>> DiG 9.3.3rc2 <<>> www.miraclelinux.com
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 33528
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 2, ADDITIONAL: 1

;; QUESTION SECTION:
;www.miraclelinux.com.          IN      A

;; ANSWER SECTION:
www.miraclelinux.com.   3600    IN      CNAME   ns.miraclelinux.com.
ns.miraclelinux.com.    3600    IN      A       219.118.163.66

;; AUTHORITY SECTION:
miraclelinux.com.       3600    IN      NS      ns.miraclelinux.com.
miraclelinux.com.       3600    IN      NS      ns1.bit-drive.ne.jp.

;; ADDITIONAL SECTION:
ns1.bit-drive.ne.jp.    77511   IN      A       211.9.32.227

;; Query time: 34 msec
;; SERVER: 192.168.100.20#53(192.168.100.20)
;; WHEN: Wed Sep 12 16:50:55 2007
;; MSG SIZE  rcvd: 134
```

- 外部のネットワークに対するテスト(逆引き)

```
# dig -x 219.118.163.66
; <<>> DiG 9.3.3rc2 <<>> -x 219.118.163.66
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 40798
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 2, ADDITIONAL: 2

;; QUESTION SECTION:
;66.163.118.219.in-addr.arpa.   IN      PTR

;; ANSWER SECTION:
66.163.118.219.in-addr.arpa. 604800 IN  CNAME   66.64.163.118.219.in-addr.arpa.
66.64.163.118.219.in-addr.arpa. 86400 IN PTR    ns.miraclelinux.com.

;; AUTHORITY SECTION:
64.163.118.219.in-addr.arpa. 86400 IN   NS      ns.miraclelinux.com.
64.163.118.219.in-addr.arpa. 86400 IN   NS      ns1.bit-drive.ne.jp.

;; ADDITIONAL SECTION:
ns.miraclelinux.com.    3558    IN      A       219.118.163.66
ns1.bit-drive.ne.jp.    77469   IN      A       211.9.32.227

;; Query time: 381 msec
;; SERVER: 192.168.100.20#53(192.168.100.20)
;; WHEN: Wed Sep 12 16:51:37 2007
;; MSG SIZE  rcvd: 177
```

第9章　DHCP サーバーの構築

この章で説明する内容

目的	コンピュータのネットワーク設定の一元管理や自動設定について理解する
機能	IP アドレスなどのネットワークパラメータ自動設定
必要な RPM	dhcp —— DHCP サーバー dhclient —— DHCP クライアント
設定ファイル	/etc/dhcpd.conf
章の流れ	1 DHCP の概要 2 DHCP サーバーの起動と停止 3 DHCP サーバーの設定 4 DHCP クライアント
関連 URL	**DHCP mini-HOWTO** http://www.linux.or.jp/JF/JFdocs/DHCP/index.html

9.1 DHCP の概要

DHCP（Dynamic Host Configuration Protocol）は、IP ネットワーク上の個々の機器が、自分自身のネットワーク設定情報（IP アドレス、サブネットマスク、ブロードキャストアドレスなど）を DHCP サーバーから得られるようにするプロトコルで、その主な目的は大規模なネットワークの管理を容易にすることです。

Asianux Server 3 では DHCP クライアントとして **dhclient** を採用しています。

9.2 DHCP サーバーの起動と停止

DHCP の起動スクリプトは、**/etc/init.d/dhcpd** です。起動スクリプトのオプションでは、起動（start）、停止（stop）、再起動（restart）、または現在の状況を確認（status）を指定できます。

DHCP の設定を変更した場合は、変更を反映するために DHCP を再起動する必要があります。

- DHCP を起動するには、次のコマンドを実行します。

```
# /sbin/service dhcpd start
```

- DHCP を停止するには、次のコマンドを実行します。

```
# /sbin/service dhcpd stop
```

- DHCP を再起動するには、次のコマンドを実行します。

```
# /sbin/service dhcpd restart
```

- DHCP の現在の状況を確認するには、次のコマンドを実行します。

```
# /sbin/service dhcpd status
```

- システムが起動したときに自動的に dhcpd が起動するようにするには、次のコマンドを実行します。

```
# /sbin/chkconfig dhcpd on
```

- システムが起動したときに dhcpd が起動しないようにするには、次のコマンドを実行します。

```
# /sbin/chkconfig dhcpd off
```

9.3 DHCP サーバーの設定

DHCP サーバーの設定は**/etc/dhcpd.conf** を記述することで行います。詳細は、「**man 5 dhcpd.conf**」を参照してください。

➢ サブネットと共有ネットワーク

リースするアドレスプールを設定します。主な設定項目は次のとおりです。

- ネットワークアドレス —— subnet
- ネットマスク —— netmask
- アドレスの範囲 —— range
- デフォルトリース時間 —— default-lease-time（秒単位で設定します）
- 最大リース時間 —— max-lease-time（秒単位で設定します）

➢ クライアントオプション

各 DHCP クライアントの設定をします。設定はサブネット単位、もしくはすべてのサブネット共通で設定することが可能です。主な設定項目は次のとおりです。

- デフォルトルータ —— option routers
- DNS ドメイン名 —— option domain-name
- DNS サーバー —— option domain-name-servers
- WINS（Windows Internet Name Service）サーバー —— option netbios-name-servers

➢ ホスト

特定のホストに固定のアドレスを設定します。主な設定項目は次のとおりです。

- ホスト名 —— host
- ハードウェアアドレス —— hardware ethernet
- 固定の IP アドレス —— fixed-address

サブネット 192.168.0.0/24 のネットワークで 192.168.0.10-254 をリースして、デフォルトのリース期間を 3 日（最大を 6 日）とし、DNS サーバーを指定し、MAC アドレス（12:34:56:78:9A:BC）のホストに固定アドレス 192.168.0.5 を指定した場合の**/etc/dhcpd.conf** の例を以下に示します。

```
ddns-update-style ad-hoc;
subnet 192.168.0.0 netmask 255.255.255.0 {
        option domain-name-servers 192.168.0.1;
        option domain-name "your.domain.name";
        option routers 192.168.0.1;
        max-lease-time 518400;
        default-lease-time 259200;
        range 192.168.0.10 192.168.0.254;
        host server {
              hardware ethernet 12:34:56:78:9A:BC;
              fixed-address 192.168.0.5;
        }
}
```

➢ リース情報データベース

DHCP でリースされた情報は**/var/lib/dhcpd/dhcpd.leases** で管理されており、各クライアントのリース情報を見ることができます。

```
lease 192.168.0.12 {
  starts 5 2007/08/10 06:48:51;
  ends 1 2007/08/13 06:48:51;
  binding state active;
  next binding state free;
  hardware ethernet xx:xx:xx:xx:xx:xx;
}
```

リース情報を削除したい場合は、**dhcpd.leases** から「lease IP アドレス{」で始まるブロックを削除します。

すべてのリース情報を削除しようとして **dhcpd.leases** ファイル自体を削除した場合、DHCP サーバーが起動しないことがあります。DHCP サーバーが起動時にこのファイルの存在を確認しているために生じる現象です。ファイルを削除した場合には、次のようにして空のファイルを作成しておいてください。

```
# /bin/touch /var/lib/dhcpd/dhcpd.leases
```

9.4 DHCP クライアント

Asianux Server 3 では、DHCP クライアントとして **dhclient** を採用しています。

- DHCP の更新または動的アドレス取得には次のコマンドを使います。

```
# /sbin/dhclient
```

- DHCP を解放するには次のコマンドを使います。

```
# /sbin/dhclient -r
```

上記のコマンドを実行すると、インターフェイスもダウンするので注意してください。インターフェイスがダウンしたときには、**dhclient** コマンドを実行してインターフェイスをアップさせてください。

第10章　Samba サーバーの構築

この章で説明する内容

目的	Samba サーバーの構築方法について理解する
機能	Windows ファイルサーバー機能、Windows プリントサーバー機能、Windows ドメインコントローラ機能
必要な RPM	samba　samba-common　samba-client samba-swat　smbldap-tools　smbdcsetup
設定ファイル	/etc/samba/smb.conf /etc/samba/smbldap_bind.conf /etc/samba/smbldap.conf
章の流れ	1 Samba の概要　5 ファイルサーバーの構築 2 Samba の起動と停止　6 プリントサーバーの構築 3 Samba サーバーの基本設定　7 winbind 連携 4 ユーザー管理　8 ドメインコントローラの構築
関連 URL	**Samba.org**　http://www.samba.org/ **日本 Samba ユーザー会**　http://www.samba.gr.jp/

10.1 Samba の概要

Samba は、現在高い評価を受けているオープンソースの Windows 互換のファイル／プリントサーバーソフトウェアです。

Samba を使用することで、Linux などの UNIX 系のサーバーを、Windows と共に利用することが可能になり、安価にファイルサーバーやプリントサーバーを構築できます。また、Samba に備えられたクライアント機能を活用することで、Linux 側から Windows のリソースを利用したり、管理操作を行うことも可能です。

本章では、Samba の基本的な使い方を説明します。さらに詳しい使用方法については、オンラインドキュメントや市販の書籍、ウェブサイトなどを参照してください。Samba.org 公式サイトや日本 Samba ユーザー会のサイトでは、Samba に関する最新情報や Samba の最新バージョンを入手できます。

10.2 Samba の起動と停止

Samba の起動スクリプトは**/etc/rc.d/init.d/smb** と **/etc/rc.d/init.d/winbind** です。winbind 連携機能を利用しない場合には、**/etc/rc.d/init.d/winbind** は利用しません。起動スクリプトのオプションでは、起動（start）、停止（stop）、再起動（restart）、または現在の状態を確認（status）を指定できます。Samba の設定を変更した場合は、変更を反映させるために、必ず Samba を再起動する必要があります。

- Samba を起動するには、root ユーザーになって次のコマンドを実行します。

```
# /sbin/service smb start
```

- Samba を停止するには、次のコマンドを実行します。

```
# /sbin/service smb stop
```

- Samba を再起動するには、次のコマンドを実行します。

```
# /sbin/service smb restart
```

- Samba の現在の接続状況を確認するには、次のコマンドを実行します。

```
# /sbin/service smb status
```

winbind の起動、終了については、10.7「winbind 連携」を参照してください。

10.3 Samba サーバーの基本設定

本節では Samba の設定ファイル **smb.conf** について説明します。Samba の設定は、**smb.conf** ファイルをエディタを利用して直接変更する方法と、SWAT（Samba Web Administration Tool：Samba ウェブ管理ツール）を用いて変更する方法があります。

samba-common パッケージがインストールされると、デフォルトの **smb.conf** ファイルが **/etc/samba** 配下に自動的に作成されます。初期状態の **smb.conf** は基本的な設定のみが行われているので、必要な設定を追加してから Samba を起動してください。

smb.conf ファイルの書式は、[ ]で囲まれた名前を持つセクションで構成され、それぞれのセクションが Samba が提供する共有やプリンタなどに対応します。セクションのうち、[global]セクションや[homes]セクションは特別な機能を設定するためのセクションです。日本語環境で利用するための標準的な設定内容は次のとおりです。

```
[global]
   unix charset = UTF-8
   dos charset  = CP932
   display charset = UTF-8
   workgroup    = WORKGROUP
   server string = Samba Server
   dos filetimes = Yes
   dos filetime resolution = Yes
[homes]
   read only = No
   browseable = No

[printers]
   comment = All Printers
   path = /var/spool/samba
   print ok = Yes
   browseable = No

[public]
  comment = Public Space; anyone can write any files
  path = /var/samba/public
  guest ok = Yes
  read only = No
  force group = public
  force create mode = 0666
  force directory mode =  0777
```

続いて、各セクションの内容、および主要なパラメータについて説明します。

10.3.1　[global]セクション

[global]セクションは、**smb.conf** ファイルの先頭に記述し、Samba 全体の設定を行います。

最初に行う設定は文字コードの設定です。文字コードに関するパラメータは、表 10-1 のとおりです。

表 10-1　文字コード関連パラメータ

項目	説明
unix charset	Samba サーバーに作成するファイルの文字コードを指定します。Samba サーバーの言語設定に合わせて設定しなければなりません。
display charset	Samba の管理コマンドやクライアントコマンドが表示するメッセージの文字コードを指定します。通常は unix charset と同じコードを設定しておきます。
dos charset	Windows 側で用いられる文字コードです。日本語版 Windows では CP932 を指定します。その他の言語環境では、Windows の利用するコードページにあわせて指定します。

表 10-2 が推奨の文字コード設定です。Samba サーバーの環境に合わせて下記のパラメータを **smb.conf** の [global]セクションに設定してください。インストール直後の日本語環境では、システムの言語設定は ja_JP.UTF-8 です。

表 10-2　文字コードパラメータの推奨値

システムの言語設定	unix charset	display charset	dos charset
ja_JP.UTF-8	UTF-8	UTF-8	CP932

その他の主要な設定項目は表 10-3 のとおりです。

表 10-3　[global]セクションの主な設定項目

項目	説明
workgroup	Samba サーバーが所属するドメイン名、もしくはワークグループ名を設定します。 インストール直後の既定値は MYGROUP です。
netbios name	Samba サーバーのコンピュータ名を設定します。何も設定しない場合、コンピュータ名として Samba サーバーのホスト名が使われます。
server string	Samba サーバーに関する説明を記述します。この値は Windows クライアントの「マイネットワーク」で見た場合に、コンピュータのコメントとして表示される文字列になります。既定値は「Samba Server」です。
passdb backend	Samba サーバーのユーザー管理データベースを指定します。詳細は 10.3.3「passdb backend」を参照してください。既定値は smbpasswd です。

各種サーバー機能に必要な設定内容に関しては、機能説明にあわせて説明します。

10.3.2 セキュリティモード

Samba には 5 つのセキュリティモードがあり、ユーザー認証の方法がこのモードの設定によって切り替わります。このセキュリティモードは、[global]セクションの **security** パラメータの設定で決まります（表 10-4）。

表 10-4 security パラメータに指定する値

パラメータ	説明
user	ユーザー単位で認証を行います。認証は Samba サーバーが行うので、ユーザーは事前に Samba サーバーに登録されている必要があります。
server	動作モードは user と同じですが、ユーザー認証を password server パラメータに指定された別のサーバーに依頼します。認証に失敗した場合のみ、user モードと同様に Samba サーバーのユーザー情報を使って認証を行います。
domain	Samba サーバーを既存の Windows ドメインのメンバーサーバーとして設定する場合に指定します。したがって、ユーザー認証はドメインコントローラで行います。Samba サーバーにアクセスするユーザーはそのドメインのユーザーとして登録されている必要があります。
ads	Samba サーバーを既存の Active Directory ドメインのメンバーサーバーとして設定する場合に指定します。ユーザー認証は Active Directory ドメインのドメインコントローラで行います。Samba サーバーにアクセスするユーザーは Active Directory ドメインに登録されている必要があります。
share	共有に接続するたびにユーザー認証が行われます。Win9x と同様で共有単位にパスワードを指定できる方法です。

security パラメータに何も指定しない場合、セキュリティモードの既定値は user として動作します。

10.3.3 passdb backend

Samba は、Windows 用のユーザー情報を格納するために、独自のユーザー情報データベースを持ちます。このユーザー情報データベースは **passdb** と呼ばれ、さまざまな形式でユーザー情報を格納できます。

passdb のデータ格納方式は、passdb backend パラメータで指定します（表 10-5）。passdb backend パラメータは、security=user の設定時に有効となります。passdb backend パラメータには、複数のバックエンドデータベースを指定できます。

表 10-5 passdb backend パラメータ

パラメータ	説明
smbpasswd	従来より用いられてきたユーザー管理データベースで、ユーザー情報がすべてテキストファイルに格納されます。passdb backend パラメータを指定しない場合の既定値です。パスワードファイルを指定しない場合、**/etc/samba/smbpasswd** にユーザー情報が格納されます。

パラメータ	説明
tdbsam	Samba 3.0 で新しく導入されたユーザー管理データベースです。ユーザー情報はバイナリ形式で格納され、既定値では**/etc/samba/passdb.tdb** にユーザー情報が格納されます。Samba3.0 では smbpasswd 形式よりも tdbsam 形式の利用が推奨されます。
ldapsam	Samba に必要なユーザー情報をすべて LDAP サーバーで管理するための設定です。ドメイン運用で PDC、BDC を必要とする場合や、ユーザー数が 250 人以上のサイトではこの方式の利用が推奨されます。また、LDAP サーバーに Linux のユーザー情報もあわせて登録することで、Linux のユーザーアカウントと Windows のユーザーアカウントを統合して管理できます。

passdb backend パラメータは運用方式に大きく関わりますが、スタンドアロンサーバーでは tdbsam の利用を、ドメイン構成では ldapsam の利用を推奨します。

passdb backend には、次のようにパラメータを設定します。

- 通常の設定

```
        passdb backend = tdbsam
```

```
        passdb backend = ldapsam:ldap://ldapserver.example.com
```

- ファイル名を指定する場合

```
        passdb backend = smbpasswd:/etc/samba/smbpasswd
```

```
        passdb backend = tdbsam:/etc/samba/passdb.tdb
```

passdb backend パラメータを指定しない場合は、既定値として smbpasswd 形式が用いられます。

10.4 ユーザー管理

Sambaサーバーの管理者にとって、ユーザー管理は非常に重要な仕事です。ここでは、Sambaのユーザー管理に関する基本的な操作方法について説明します。

Sambaのユーザー管理では、次の3つのアカウント情報に関して管理を行います。

- **ユーザーアカウント**
  Sambaサーバーを利用するユーザーそれぞれの情報を管理します。ユーザーIDやパスワードなど、Sambaサーバーに必要なすべての情報を含んでいます。
- **マシン信頼アカウント**
  Sambaサーバーをドメインコントローラとして構築した場合に、ドメインに参加するクライアントマシンの情報を管理します。
- **グループアカウント**
  Samba サーバー上でユーザーをまとめて扱うためのグループ情報を管理します。また、Windowsに初期設定されているいくつかのグループは、Sambaの初期グループとして登録されています。

Sambaのユーザー管理のほとんどは、**pdbedit**と**smbpasswd**で行います。これらのコマンドは、Sambaサーバー上で実行しなければなりません。一方、ユーザー管理機能を持つコマンドとして、**net**があり、このコマンドは本来リモートのWindowsサーバーの管理を行うためのコマンドですが、その一部の機能を用いてユーザー管理を行うこともできます。

本節では、**pdbedit**と**smbpasswd**の操作方法について説明します。

10.4.1 ユーザーの追加

Sambaサーバーを利用するためには、あらかじめSamba用のユーザーアカウントを作成しておかなければなりません。通常は、次の順序でSamba用のユーザーアカウントを作成することになります。

（1）Linuxのグループアカウントの作成
（2）Linuxのユーザーアカウントの作成
（3）Sambaのユーザーアカウントの作成

なお、Linuxのユーザーやグループアカウントの作成の詳細に関しては、第3章「ユーザー／グループ管理」を参照してください。

（1）Linux のグループアカウントの作成

ユーザーは必ず 1 つ以上のグループに属します。ファイルの操作権限などは、グループ単位での設定を行うことが多いので、アクセス管理の観点からも、グループ単位でユーザーを管理することを推奨します。そこで、まずは Linux のグループアカウントを作成します。

下記の例は、GID（グループ ID）を 20001 番のグループ project を作成しています。

```
# /usr/sbin/groupadd -g 20001 project
```

必要なグループの数だけ、グループの作成を繰り返してください。

（2）Linux のユーザーアカウントの作成

グループアカウントの作成を終えたら、次に、そのグループに所属するユーザーを作成します。次の実行例は、UID（ユーザー ID）が 20101 番で、project グループに所属するユーザー tanaka を作成しています。

```
# /usr/sbin/useradd -u 20101 -g project tanaka
```

また、ユーザーは複数のグループに所属することもできます。次の実行例は、project グループに加えて、manager グループにも所属するユーザー yamada を作成しています。

```
# /usr/sbin/useradd -u 20202 -g project -G manager yamada
```

作成したユーザーが Linux サーバーに **ssh** や **telnet** でログインする必要がある場合には、**passwd** でユーザーのパスワードを設定してください。ユーザーが Samba サーバーのみにログインする場合には、この時点でパスワードを作成する必要はありません。

なお、作成したユーザーの ID や、所属グループの確認は **id** で行うことができます。

```
# /usr/bin/id yamada
uid=20201(yamada) gid=20001(project) 所属グループ=20001(project),20002(manager)
```

（3）Samba のユーザーアカウントの作成

Linux のユーザーアカウントが作成できたら、**pdbedit** で Samba のユーザーアカウントを作成します。ユーザー情報を新規に作成するときは、**-a** オプションを指定します。**pdbedit** を実行すると Samba 用のパスワードの入力が求められるので、パスワードを入力してください。

```
# /usr/bin/pdbedit -a tanaka
new password:********
retype new password:********
Unix username:        tanaka
NT username:
Account Flags:        [U          ]
User SID:             S-1-5-21-1722631489-2624286973-3194339827-41202
Primary Group SID:    S-1-5-21-1722631489-2624286973-3194339827-41003
Full Name:
Home Directory:       \\asianux3\tanaka
HomeDir Drive:
Logon Script:
Profile Path:         \\asianux3\tanaka\profile
Domain:               ASIANUX3
Account desc:
Workstations:
Munged dial:
Logon time:           0
Logoff time:          Tue, 14 Jan 2038 12:14:07 GMT
Kickoff time:         Tue, 14 Jan 2038 12:14:07 GMT
Password last set:    Thu, 16 Aug 2007 16:48:29 GMT
Password can change:  Thu, 16 Aug 2007 16:48:29 GMT
Password must change: Tue, 14 Jan 2038 12:14:07 GMT
Last bad password   : 0
Bad password count  : 0
Logon hours         : FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
```

以上でユーザーの作成は完了です。

作成したユーザーアカウントに関する情報を確認するときには、**pdbedit** の**-L** オプションを利用します。**-L** オプションのみ指定した場合には、最小限の情報のみが表示されるので、次の実行例のように**-v** オプションもあわせて指定して、すべての情報を確認してください。

```
# /usr/bin/pdbedit -L -v tanaka
```

Samba サーバーが稼動していれば、次のコマンドでログインできることを確認できます。パスワードなどに間違いがなければ、認証に成功し、Samba サーバーの情報が表示されます。

```
# /usr/bin/smbclient //localhost/共有名 -U ユーザー名
Password:
Domain=[ASIANUX3] OS=[Unix] Server=[Samba 3.0.28-0.4.3AXS3]
smb: \>
```

10.4.2 ユーザーアカウントの変更、削除

登録済みのユーザーアカウントの変更、削除も **pdbedit** で行います。ユーザーアカウントには多くの項目が登録されていますが、変更可能な項目に関してはそれらの 1 つ 1 つに対応したオプションが用意されています。

pdbedit で変更できる項目の詳細に関しては man データを参照してください。**--help** オプションを指定すると、簡単なヘルプメッセージも表示されます。次の実行例は、ユーザーに設定された Full Name の項目を変更しています。

```
# /usr/bin/pdbedit -f "T.Tanaka" tanaka
```

ユーザーアカウントの削除は、**pdbedit** の**-x** オプションで行います。

```
# /usr/bin/pdbedit -x tanaka
```

10.4.3 パスワード管理

ユーザーのパスワード設定、変更は **smbpasswd** で行います。root 管理者のみがユーザー名を指定して他のユーザーのパスワード変更を行うことができます。

```
# /usr/bin/smbpasswd tanaka
New SMB password: ********
Retype new SMB password: ********
```

各ユーザーも、**smbpasswd** を用いて自分の Samba 用パスワードを変更できます。ユーザーがパスワードを変更するときには現在利用中のパスワードも入力する必要があります。

```
$ /usr/bin/smbpasswd
Old SMB password: ********
New SMB password: ********
Retype new SMB password: ********
```

Samba 用のパスワードの変更にあわせて、Linux 用のパスワードの変更も行いたい場合には、次の設定を[global]セクションに行っておきます。

```
[global]
  ...
  unix password sync = yes
  pam password change = yes
```

（1） Windows クライアントからのパスワード変更

前述のようにユーザーが Linux サーバーにログオンすれば、ユーザー自身がパスワードを変更できます。しかしこの方法は Windows を利用しているユーザーにとってはわずらわしいものです。そこで、Windows ユーザーがより簡単にユーザー自身のパスワードを変更するための方法を紹介します。

Windows XP の場合は、パスワードの変更の画面を表示するために、[コントロールパネル]-[ユーザーアカウント]-[ユーザーのログオンやログオフの方法を変更する]を選択して、図 10-1 の画面を表示します。[ようこそ画面を使用する]が選択されている場合は、チェックを外しておきます。この操作は、各クライアントで行う必要があります。Windows 2000 の場合はこの操作は不要です。

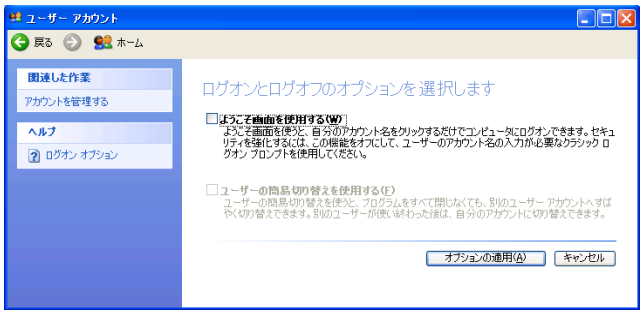


図 10-1 Windows XP のログオン方法の変更画面

続いて、Windows 2000/XP などの画面上で、[CTRL]+[ALT]+[Delete]キーを同時に押して、図 10-2 の画面を表示させます。

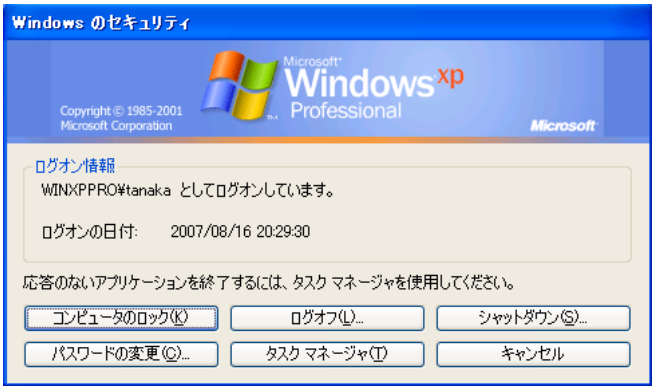


図 10-2　Windows のセキュリティ画面

図 10-2 の画面で、[パスワードの変更(C)]ボタンをクリックすると、図 10-3 の画面が表示されます。パスワードを変更するユーザー名を入力して、ログオン先に Samba サーバー名を入力します。そして、現在利用中のパスワードと、変更後のパスワードを入力します。すべての項目を入力してから、[OK]をクリックします。

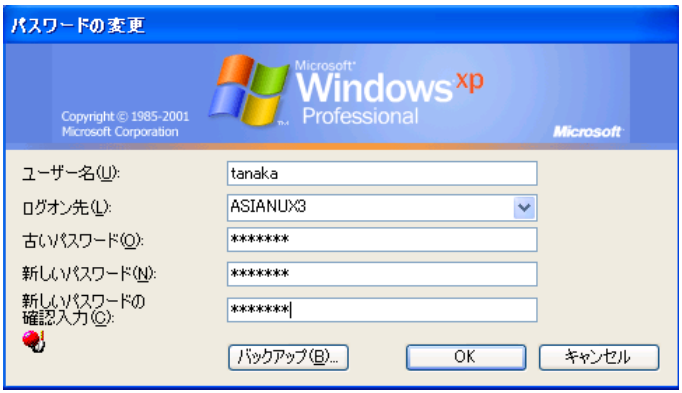


図 10-3　Samba サーバーのパスワード変更

パスワードの変更に成功すれば、図 10-4 のメッセージが表示されます。

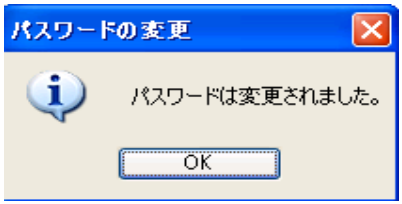


図 10-4 パスワードの更新成功

10.5 ファイルサーバーの構築

ファイルサーバー機能は、Samba の一番基本となる機能です。Samba を利用するユーザーの多くがこの機能を利用するでしょう。ファイルサーバー機能には数多くの機能が実装されているので、ここでは頻繁に利用する基本的な機能について紹介します。

10.5.1 ファイル共有の作成

ファイルサーバーとして構築するためには、ユーザーが利用できる**ファイル共有**を作成しなければなりません。ファイル共有を作成するためには、**smb.conf** ファイルに、ファイル共有セクションを追加します。

典型的なファイル共有は次のような形式です。

```
[project]
   path = /var/samba/project
   read only = no
   browseable = yes
```

最初に[共有名] を書きます。これが、共有セクションの始まりを意味します。この共有セクション内で設定したパラメータは、この共有にのみ有効となります。続いて、この共有に置かれたファイルが、Samba サーバー上のどこに置かれるかを指定するために **path** パラメータを指定します。path を指定しない場合、既定値として **/tmp** ディレクトリが用いられますが、**/tmp** ディレクトリはシステムによっては定期的なクリーンアップが行われていることもありますので、共有用のディレクトリとしては不適切です。必ず、共有用のディレクトリを作成し、path に指定するようにしましょう。

ディレクトリ作成時の注意事項として、Samba がファイルを操作するときにはユーザーの権限で操作するので、root しかファイル操作ができないような権限にしないように注意してください。アクセス制限に関しては、Samba の機能として別途設定できます。

read only パラメータは、その共有上のファイルの新規作成や更新を許可するかどうかのパラメータです。yes を指定した場合、その共有上のファイルを更新することはできなくなります。既定値では yes が指定されています。

browseable パラメータを yes に指定すると、Windows の「マイネットワーク」にファイル共有が表示されます。このパラメータで制御できるのは、あくまでも「マイネットワーク」での表示上の動作に限られるので、表示していなくても、共有名がわかっていれば直接 UNC を指定することで、アクセスすることは可能です。既定値は yes です。

最低限、以上の設定を行えば、ファイル共有として利用することが可能です。Samba サーバーを起動して、ファイル共有が表示されることを確認してみましょう。Windows から確認する場合には、「マイネットワーク」から確認するか、エクスプローラのアドレスバーに「¥¥Samba サーバー名」あるいは、「¥¥Samba サーバーの IP アドレス」を入力します。Samba サーバーが稼動していれば、ユーザー認証のダイアログが表示されるので、登録済みのアカウントを使って認証してください（図 10-5）。

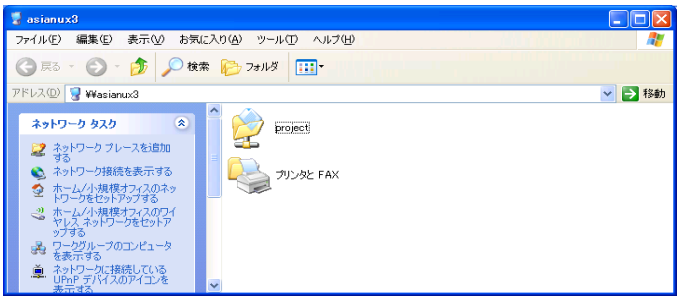


図 10-5　ファイル共有の確認

Samba サーバー上から確認する場合には、**smbclient** を使ってください。ファイル共有サービスが正常に動作していれば、Samba サーバーの情報が表示されてログインに成功します。

```
# /usr/bin/smbclient //localhost/project -U ユーザー名
Password: ******
Domain=[ASIANUX3] OS=[Unix] Server=[Samba 3.0.28-0.4.3AXS3]
smb: \>
```

10.5.2 homes 共有機能

Samba は、ユーザーのホームディレクトリを共有としてユーザーごとに提供する機能を持っています。ユーザーのホームディレクトリとは、Linux のユーザーアカウントを作成したときに、**/home/ユーザー名**などの形でユーザーごとに用意されているディレクトリのことを意味します。ユーザーのホームディレクトリを確認したい場合には、**getent** を利用します。次の実行例では、ユーザー tanaka のホームディレクトリが**/home/tanaka** であることを確認できます。

```
# /usr/bin/getent passwd tanaka
tanaka:x:20101:20001::/home/tanaka:/bin/bash
```

Samba でこの機能を利用するためには、**smb.conf** ファイルに[homes]セクションを作成します。

典型的な[homes]セクションの設定例は次のようになります。

```
[homes]
      read only = No
      browseable = No
      comment = %U
```

homes 共有では、接続ユーザーのユーザー名の共有が提供されるため、「homes」という名称の共有は必要ないため、**browseable** パラメータを No とします。通常は、ユーザー用の共有は書き込み可能とするために **read only** パラメータを No としています。**comment** パラメータは必須ではありませんが、今回は設定例として追加しています。パラメータの値に**%U** を使うと、実行時には変数の置換が行われて%U が接続ユーザー名に変換されます。図 10-6 では、ユーザー tanaka で接続中のため、共有名「tanaka」の共有が利用可能となっています。

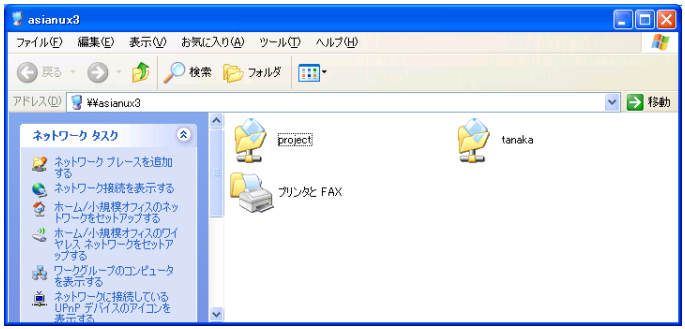


図 10-6 homes 共有機能によるファイル共有

10.5.3　共有レベルのアクセス管理

Samba をファイルサーバーとして運用するときには、複数の共有を作成して、グループ単位でアクセス可能な共有を制限することが一般的なアクセス管理手法です。Samba では、さまざまなパラメータを用いて共有へのアクセスを管理できます。また、共有内で作成されるファイルやディレクトリに関するルールもあわせて設定することで、柔軟なアクセス管理を実現しています。

（1）共有へのアクセス制限

smb.conf ファイルの各共有セクションごとに、アクセス管理のためのパラメータを設定できます。主要なパラメータについて説明します。

- **write list**
  このパラメータに設定されたユーザーとグループは共有上のファイルに対して、更新権と参照権が与えられます。このパラメータに設定されたユーザーは、read only パラメータが yes に設定されている共有に対しても、更新権を持ちます。グループ名で指定するときには、「@グループ名」の形式で指定します。
- **read list**
  このパラメータに指定されているユーザーは、read only パラメータの設定に関係なく参照権しか与えられません。write list パラメータと同様にグループ名での指定も可能です。なお、write list パラメータと read list パラメータの両方に指定されたユーザーは、write list パラメータの設定が優先されます。
- **invalid users**
  このパラメータに設定されたユーザーは、この共有にアクセスできなくなります。
- **valid users**
  既定値では、何も指定されていません。このパラメータが設定されていない状態であれば、どのユーザーでも共有にアクセスできます。いったん、このパラメータに値が設定されると、このパラメータに設定されていないユーザーは、共有にアクセスできなくなります。invalid users パラメータと valid users パラメータの両方に同じユーザーが指定された場合、invalid users の設定が優先されます。
- **admin users**
  このパラメータに設定したユーザーは、この共有内では root 権限を持ってファイル操作を行うことが可能になります。つまりすべての操作が許可されることになるため、設定や使用には細心の注意が必要です。

次の例は、共有[project]に対して、projectグループのメンバーとmanagerグループのメンバーが参照・更新可能で、testグループのメンバーは参照のみが可能になるように設定をしています。その他のユーザーはアクセスが拒否されます。

```
[project]
    path = /var/samba/project
    read only = no
    browseable = yes
    write list = @project, @manager
    read list = @test
    valid users = @project, @manager, @test
```

(2) ファイル・ディレクトリ作成時の権限制限

ユーザーが共有上に新しくファイルを作成するとき、その他のユーザーとのアクセス権の兼ね合いで、ある属性を強制したいことがあります。このような場合に備えて、共有単位でファイルやディレクトリの新規作成時の属性を管理できます。

- **create mask**
  ファイルを作成する際のファイル属性のマスクを4桁の8進数で指定します。既定値は0744です。このマスク値として設定されていないビットは、新規に作成するファイルの属性から削除されます。したがって、既定値の0744のマスクであれば、所有者の参照権・更新権・実行権、グループの参照権、その他の参照権以外の属性は必ず削除されます。
- **directory mask**
  ディレクトリを作成する際のディレクトリ属性のマスクを4桁の8進数で指定します。既定値は0755です。このマスク値として設定されていないビットは、新規に作成するディレクトリの属性から削除されます。したがって、既定値の0755のマスクであれば、グループの更新権、その他の更新権は必ず削除されます。グループ内のメンバーが自由にディレクトリ内を操作するためには、既定値を0775に変更しておかなければならないでしょう。
- **force create mode**
  ファイルに対して強制的に付与したいビットを指定します。create maskパラメータの処理の後でビットが追加されるので、必ず指定したい属性がある場合に利用します。
- **force directory mode**
  ディレクトリに対して強制的に付与したいビットを指定します。directory maskパラメータの処理の後でビットが追加されるので、必ず指定したい属性がある場合に利用します。

- **force user**
  この共有上でファイル操作を行う際のユーザー権限として、このパラメータに指定したユーザーが強制的に利用されます。したがって、共有内のファイルの所有者が、パラメータに指定したユーザーのみとなります。ただし、共有への接続時には、通常どおり接続を行ったユーザー権限で認証が行われます。
- **force group**
  force user と同様に、ファイル操作を行う際のグループ権限を強制的に設定できます。

（3）Guest 接続

共有に接続する際、ユーザー認証に失敗した場合に、Guest（ゲスト）接続と呼ばれる形態で共有に接続することが可能です。Guest 接続を可能にするためには、まず[global]セクションの **map to guest** パラメータを設定しなければなりません。map to guest パラメータには次の値を設定できます。

- **Never**
  パスワードが不正な場合の接続を許可しません。したがって、Guest 接続を行うことができません。この値が既定値です。
- **Bad User**
  ユーザーが存在して、かつパスワードが間違っている場合には、接続を拒否します。一方、ユーザーが存在しない場合には、Guest 接続として接続します。
- **Bad Password**
  パスワードが一致しない場合には、Guest 接続として接続します。このときに、ユーザー側にはパスワードを間違えたことが伝えられないため、Guest として接続していることを判断できません。その結果、通常と異なる権限でファイル操作を行い、操作が拒否されることがあるので、利用の際には注意が必要です。

また、[global]セクションでは、Guest 接続時に利用するユーザーアカウントを **guest account** パラメータで指定できます。既定値は nobody となっています。

```
map to guest  = Bad User
guest account = nobody
```

続いて、各共有ごとに Guest 接続を許可するかどうか指定します。

- **guest ok**
  ゲスト接続を許可する場合に Yes を指定します。map to guest パラメータが有効な場合のみ、有効となります。

- **guest only**
  Yes を指定した場合、接続要求をすべて Guest 接続として処理します。map to guest パラメータが有効な場合のみ、有効となります。

10.5.4 ネットワークレベルのアクセス制限

Samba をネットワークレベルでアクセス制限することも可能です。
次に共有ごとに設定可能なネットワークレベルでのアクセス制限です。

- **hosts allow**
  共有へのアクセスを許可するコンピュータのリストを指定します。[global]セクションで指定された場合、すべての共有に対して有効な設定となります。
- **hosts deny**
  共有へのアクセスを禁止するコンピュータのリストを指定します。hosts allow と矛盾した設定を行った場合には、hosts allow の設定が優先されます。

10.6 プリントサーバーの構築

Samba のプリントサーバー機能を用いて、Samba サーバーに接続されたプリンタや、ネットワーク上のプリンタに対して、ユーザーの Windows クライアントから、ドキュメントを印刷することが可能になります。

プリントサーバーとして動作する場合の Samba サーバーの役割は、クライアントからの印刷要求を受け取って、印刷のデータをプリンタへと転送することです。プリンタへの印刷データは、Windows クライアント上で該当のプリンタ用のデータとして変換済みのため、Samba サーバーは Windows のプリンタドライバの情報などを必要としません。したがって、Windows 用にドライバが提供されているプリンタであれば、Samba のプリントサーバー機能を利用できます。

プリントサーバー機能を利用するためには、あらかじめシステム上でプリンタの設定を行っておかなければなりません。プリンタの設定方法に関しては、第 7 章「プリンタの管理」で説明されているので、参照しながらプリンタの設定を行ってください。

Samba のプリントサーバー機能のための注意点として、プリンタキューに投入されたデータをそのままプリンタに渡す必要があるため、プリンタドライバとして「RAW タイプ」を選択することに注意してください。RAW タイプのドライバは、ドライバとして、何も処理を行わないことを意味します。

10.6.1 smb.conf の設定

プリンタを利用するためには、[global]セクションに次のパラメータを指定します。

- **printing**
  Asianux Server 3 では、印刷システムとして CUPS を採用しているので、値として **CUPS** を設定します。

10.6.2 printers セクションの設定

smb.conf ファイルの printers セクションは、プリンタ用の特別なセクションです。**smb.conf** ファイルに printers セクションを作成することで、CUPS によって作成されたプリンタを自動的に Samba の共有プリンタとして扱うことが可能になります。典型的な printers セクションの設定は次のようになります。

```
[printers]
       comment = All Printers
       path = /var/spool/samba
       printable = Yes
       browseable = No
```

10.6.3 プリンタのアクセス管理

プリンタ共有もファイル共有と同様に、共有レベルやネットワークレベルでのアクセスを管理できます。設定できるパラメータは、ファイル共有と同じなので、ファイル共有の設定を参照してください。

プリンタ共有独自のパラメータとして、**printer admin** があります。このパラメータは、ファイル共有の **admin users** に相当するパラメータで、このパラメータに指定したユーザーが、プリンタのジョブ管理などの操作を行うことが可能になります（図 10-7）。また、root ユーザーは、常にこの権限を持っています。次の設定例は、admin ユーザーと staff グループのユーザーがプリンタのジョブ管理を可能にするための設定です。

```
printer admin = admin, @staff
```

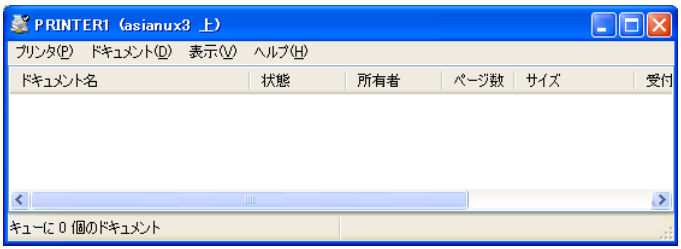


図 10-7 プリンタのステータス画面

10.7　winbind 連携

winbind 機能は、Windows のアカウント情報を Linux でも利用可能にするための機能です。winbind を利用することで、Windows のユーザーアカウントを Linux 上で利用することが可能になります。winbind 機能は、Windows サーバーとの連携が必要なドメインコントローラ構成やドメインメンバーサーバー構成で、非常に役立ちます。

winbind 機能は、winbindd、libnss_winbind ライブラリ、pam_winbind ライブラリから構成されます。**winbindd** は、smbd、nmbd に次ぐ Samba の第 3 のデーモンで、winbind 機能を提供する中心的な役割を担います。**libnss_winbind ライブラリ**は、Linux の NSS（Name Service Switch）機能で winbind 機能を利用可能にするために利用されます。NSS 機能とは、Linux のユーザーアカウントや、グループアカウントの情報を、さまざまなバックエンドから収集してユーザーに提供するための枠組みです。**pam_winbind ライブラリ**は、Linux の PAM（Pluggable Authenticate Module）機能の一機能として、ユーザー認証を Windows サーバー側で行うことを実現するためのライブラリです（図 10-8）。

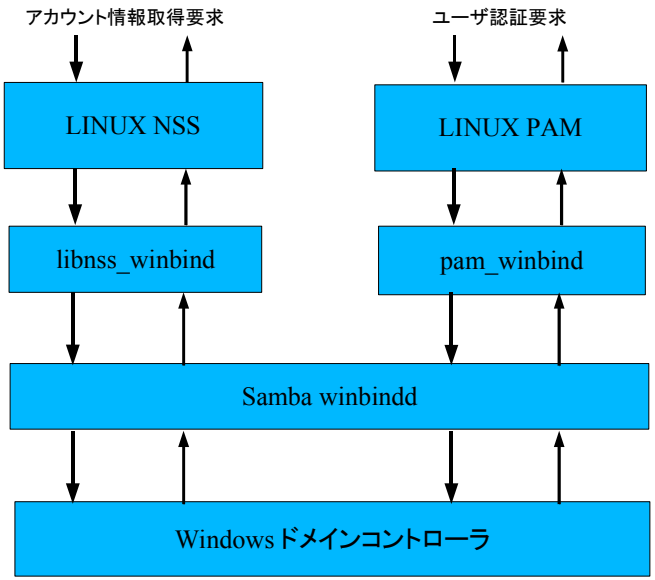


図 10-8　winbind 機能の構成

10.7.1 NSS、PAM の設定

winbind 機能を利用することで、**getpwent** や **getgrent** といったシステム標準のライブラリを呼び出してユーザー情報やグループ情報の取得を試みるときに、Windows ドメインコントローラからこれらのアカウント情報を取得することが可能になります。システム標準のライブラリ内部で動作が切り替わるので、これらのライブラリを利用するアプリケーションは、その詳細を知る必要はありません。

winbind 機能を有効にするためには、**authconfig-tui** コマンドを利用して設定を行います。

```
# /usr/sbin/authconfig-tui
```

authconfig-tui を実行すると、図 10-9 が表示されますので、ユーザー情報の[Winbind を使用]と認証の[Winbind 認証を使用]を有効にします。

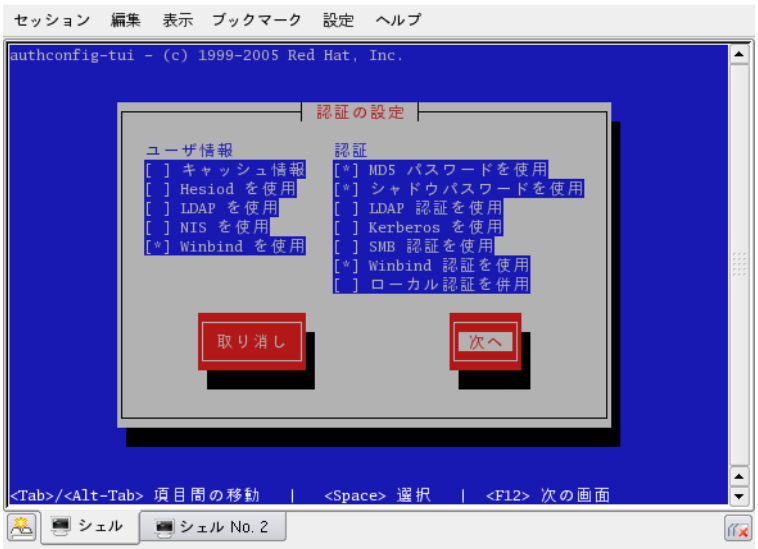


図 10-9 authconfig による winbind の設定

[次へ]を選択すると、winbind の設定の画面に移ります（図 10-10）。

図 10-10　Samba の winbind 設定

図 10-10 では、winbind を有効にするために必要な Samba のパラメータを設定します。

- セキュリティモデル
  ads…Active Directory 形式でドメインに参加する方法です。
  domain…NT ドメイン形式でドメインに参加する方法です。
- ドメイン
  参加するドメインのドメイン名を指定します。
- ドメインコントローラ
  ドメインコントローラのコンピュータ名、もしくは IP アドレスを指定します。
- ADS レルム（Realm）
  Active Directory 形式でドメインに参加するときに、Realm 名を指定します。
- テンプレートシェル
  ユーザーのデフォルトのシェルを設定します。ログインを許可する場合には、/bin/bash などを選択します。

以上の入力を完了したら、[ドメイン参加]を選択します。

図 10-11 の画面が表示されますので、ドメイン管理者のユーザー名とパスワードを入力してから、[OK]を選択します。

図 10-11 ドメイン管理者のパスワード入力

Windows サーバーが適切に動作している環境であれば、ドメインへの参加が完了し、図 10-10 の画面に戻ります。ドメインへの参加が成功しているかどうかは、wbinfo コマンドで確認することができます。

```
# /usr/bin/wbinfo -u
DOMAIN\administrator
DOMAIN\guest
```

ドメインの参加に成功している場合は、Windows サーバーに登録されているユーザー名が表示されます。

wbinfo が失敗する場合には、ドメインの参加に失敗していますので、各パラメータの設定値、ドメイン管理者のパスワードなどを再度確認してください。

10.7.2 smb.conf の設定

winbind 機能を利用するためには、いくつかのパラメータを設定する必要があります。そのうち、idmap uid と idmap gid は必須のパラメータです。これらのパラメータが設定されていない場合、winbindd は起動しません。

- **idmap uid**
  winbindd は Windows ドメインコントローラから取得した情報を元に、Linux 用のユーザー情報を作成します。このときに、このパラメータに指定した範囲の UID を順番に割り当てます。既存の Linux のユーザーアカウントに割り当てられていない UID の範囲を指定しなければなりません。
- **idmap gid**
  idmap uid と同様の機能で、GID に関する範囲を指定します。
- **winbind enum users**
  Windows ドメインコントローラに非常に多数のユーザーが登録されている場合、ユーザーエントリの取得に非常に時間がかかることがあります。このような環境でユーザー一覧の取得時に、一覧の取得を抑制するためのパラメータです。no を設定すると、ユーザー一覧の取得を抑制します。ただし、一部のアプリケーションでは、取得できるはずのユーザー情報が取得できないことにより、異常な動作を引き起こす可能性もあります。
- **winbind enum groups**
  winbind enum users と同様のパラメータで、no を指定した場合、グループ一覧の取得を抑制します。
- **template homedir**
  winbind によって、自動的に作成されたユーザーのホームディレクトリの初期パラメータとして割り当てられます。既定の設定値は **/home/%D/%U** なので **/home/ドメイン名/ユーザー名** がホームディレクトリとして割り当てられます。
- **template shell**
  winbind によって、自動的に作成されたユーザーのログインシェルの初期パラメータとして割り当てられます。既定では **/bin/false** が設定されているため、winbind によって作成されたユーザーは Linux システムにログインできません。これらのユーザーに Linux システムへのログインを許可するためには、**/bin/bash** などを設定しておきます。
- **winbind enable local accounts**
  このパラメータを yes に設定すると、winbindd は、ユーザー作成要求が発生したときに、winbind 独自のデータベース内にユーザーアカウントを自動的に作成します。ユーザーエントリ一覧の取得時などには、winbindd 経由で、作成されたユーザーアカウント情報などが提供されます。

- **winbind separator**

  winbind 機能を用いると、Windowsドメインコントローラから取得したユーザー情報は「ドメイン名\ユーザー名」の形式であらわされます。Linux システムにおいて、「\」(バックスラッシュ)は、シェル上などで特別な意味を持つので、運用上好ましくありません。そこで、「\」を他の文字に置き換えたい場合に、このパラメータに置き換える文字の設定を行います。

- **winbind use default domain**

  このパラメータに yes を指定すると、winbind によって作成されたユーザー名から、ドメイン名が取り除かれます。既定値は no です。

次の設定は、典型的な winbind のための設定例です。winbind 関連パラメータ以外は除いています。

```
idmap uid = 30000-40000
idmap gid = 30000-40000
winbind enum users  = yes
winbind enum groups = yes
template homedir = /home/%U
template shell   = /bin/bash
winbind enable local accounts = yes
winbind use default domain = yes
```

10.7.3 winbindd の起動・停止

smb.conf、NSS、PAM の設定を完了したら、winbindd を起動して、winbind 機能を有効にします。ただし winbind 機能は Samba の構成によってその動作が切り替ります。winbind 機能によって Windows ドメインコントローラのユーザー情報が取得できるのは、次のような構成の場合です。

- Samba がドメインコントローラとなっており、他のドメインと信頼関係を構築している場合、winbind 機能によって、信頼関係を構築しているドメインのユーザー情報を取得できます。
- Samba が NT ドメイン、もしくは Active Directory ドメインのドメインメンバーとしてドメインに参加している場合、ドメインコントローラのユーザー情報を取得できます。

したがって、winbind の設定が完了した後で、winbind を起動させる前に、Samba のドメインコントローラ設定やメンバーサーバー設定を行ってください。

- winbind の起動は、次のように行います。

```
# /sbin/service winbind start
```

- winbind の停止は、次のように行います。

```
# /sbin/service winbind stop
```

なお、winbind 機能が有効な状態で winbindd を起動すると、最初にドメインコントローラからユーザー情報一覧の取得を行います。ユーザー数やグループ数が多い場合、この処理に時間がかかることがありますので、注意してください。

winbind 機能の動作を確認するために **getent** を利用します。システムで利用可能なユーザー一覧を **getent** で取得して、ユーザー情報に Windows サーバーのユーザーアカウントが含まれていることを確認します。グループに関しても同様です。

```
# /usr/bin/getent passwd
  ... 省略 ...
# /usr/bin/getent group
  ... 省略 ...
```

10.8 ドメインコントローラの構築

ドメインとは、Windows ネットワークの管理の枠組みで、Windows NT 4.0 Server が提供していた NTドメインと、Windows 2000 Server 以降で提供されている Active Directoryドメインがあります。Samba は NTドメインを管理するドメインコントローラの機能を提供できます。ドメインを構築する利点は、ドメインに所属するユーザー情報の管理をドメインコントローラに一元化でき、ドメイン特有のさまざまな機能を提供できることです。本節では、Samba LDAP 連携を使ったドメインコントローラ構築について説明します。

ドメインコントローラには、サービスを提供する中心となるプライマリドメインコントローラ（PDC）と、PDC の障害に備えたり、負荷分散をはかったりするために利用されるバックアップドメインコントローラ（BDC）の 2 種類があります。ドメインコントローラは、PDC に格納されているユーザー情報を使って、ドメインに所属しているクライアントからのユーザー認証要求などに対応します。BDC は PDC からユーザー情報を複製し、ユーザー認証要求に応えます。

10.8.1　smbdcsetup での PDC の設定

Samba をドメインコントローラとして構築するためには構築用の GUI ツールを利用して行うことができます。

smbdcsetup ツールの起動は、次のコマンドで行います。

```
# /usr/sbin/smbdcsetup
```

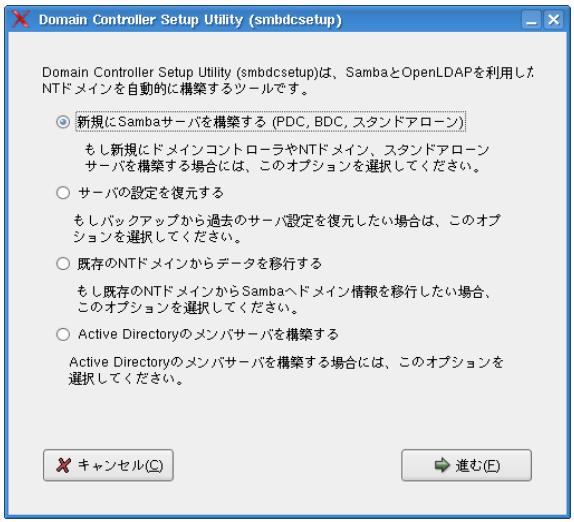


図 10-12　smbdcsetup 画面

「新規に Samba サーバーを構築する (PDC, BDC, スタンドアローン)」を選択し、[進む(F)]ボタンを押します（図 10-12）。

図 10-13 smbdcsetup 画面

「新規にプライマリドメインコントローラ(PDC)を作成する」を選択し、[進む(F)]ボタンを押します（図 10-13）。

図 10-14 smbdcsetup 画面

「ドメイン名」、「NetBIOS 名」を入力し、[進む(F)]ボタンを押します（図 10-14）。

図 10-15　smbdcsetup 画面

「ドメインの DNS」を入力し、[進む(F)]ボタンを押します（図 10-15）。

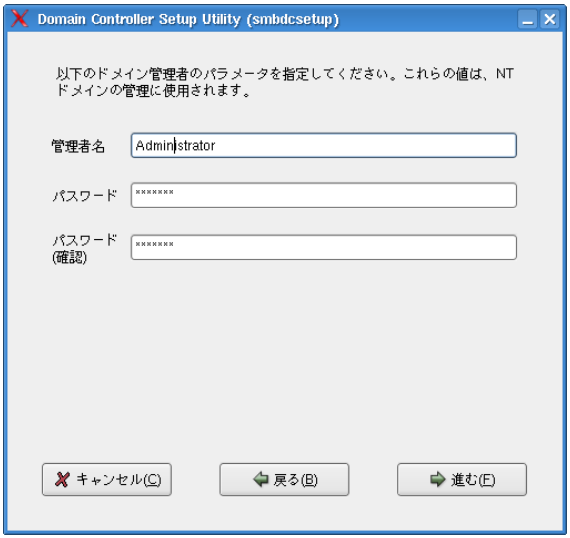


図 10-16　smbdcsetup 画面

「管理者名」、「パスワード」を入力し、[進む(F)]ボタンを押します（図 10-16）。

図 10-17　smbdcsetup 画面

参加させたいバックアップドメインコントローラを設定し、[進む(F)]ボタンを押します（図 10-17）。

図 10-18　smbdcsetup 画面

更新するファイルを選択し、[進む(F)]ボタンを押します（図 10-18）。

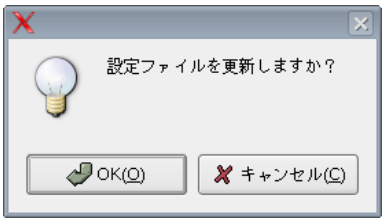


図 10-19　smbdcsetup 画面

設定ファイル更新確認で、[OK(O)]ボタンを押します（図 10-19）。

図 10-20　smbdcsetup 画面

設定ファイルバックアップ確認で、[OK(O)]ボタンを押します（図 10-20）。

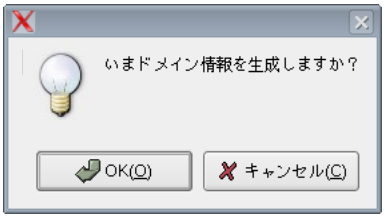


図 10-21　smbdcsetup 画面

ドメイン情報生成確認で、[OK(O)]ボタンを押します（図 10-21）。

図 10-22　smbdcsetup 画面

ドメインコントローラの構築完了画面が表示されるので、[終了(Q)]ボタンを押します（図 10-22）。

図 10-23　smbdcsetup 画面

smbdcsetup ツールの終了確認で、[OK(O)]ボタンを押します（図 10-23）。
以上でドメインコントローラの構築は完了です。

第11章　Oracle データーベースサーバーへの対応

この章で説明する内容

目的	商用データベースサーバーの Oracle に関する情報を知る
機能	
必要な RPM	特になし
設定ファイル	特になし
章の流れ	1 Oracle データベースの概要 2 Install Navigator for Oracle について 3 Oracle の自動起動／停止設定 4 Oracle の動作確認 5 Oracle に関する情報
関連 URL	日本オラクル株式会社　http://www.oracle.co.jp/ Oracle Technology Network Japan　http://otn.oracle.co.jp/

11.1 Oracle データベースの概要

Oracle データベースは、トップクラスのシェアを持つ商用データベースです。またデータベース以外にも、ウェブアプリケーションサーバーの Oracle Application Server やグループウェアの Oracle Collaboration Suite、ERP ソフトウェアの Enterprise Business Suite（EBS）などがあります。

Asianux Server 3 は、Oracle 製品に最適化されています。そのため、カーネルパラメータを変更したり、追加でパッケージをインストールしなくても、Oracle 製品をインストールできます。またインストール支援ツールとして、Install Navigator for Oracle を提供しています。

11.2 Install Navigator for Oracle（oranavi）の概要

Asianux Server 3 には、Oracle 製品（現在は Database に対応）のインストールを支援するツール Install Navigator for Oracle（以下 oranavi と記述）がバンドルされています。

一般的に Oracle 製品のインストールは、次の手順で行ないます。

1. swap 領域の確保
2. カーネルパラメータの調整
3. シェル制限の設定
4. インストール先のディスクにディレクトリ作成
5. Oracle 用 Linux ユーザー、Linux グループの作成
6. Oracle ユーザー用 環境変数の設定
7. Oracle Universal Installer の起動
8. Oracle 製品のインストール

Asianux Server 3 では OS で 2 と 3 に関して再設定の必要がないように最適化されています。これに加えて oranavi を使用すれば、4 から 7 の作業を GUI で簡単に設定できます。また、それぞれの Oracle 製品のインストール手順が解説された HTML ドキュメントを提供しています。

oranavi は GUI アプリケーションです。root ユーザーでログインし、X Window を起動させた後、[Start]メニューから「アプリケーション」－「ユーティリティ」－「インストールナビゲータ for Oracle」を選択するか、又は次のコマンドを実行します。

```
# /usr/sbin/oranavi
```

oranavi の起動画面（図 11-1）が表示されます。

図 11-1 oranavi 起動画面

oranavi の起動と同時に、各バージョンにおけるインストール手順が解説された HTML ドキュメント（図 11-2）が表示されます。oranavi 操作や Oracle インストール手順については HTML ドキュメントの方でより詳細に解説しています。ここでは oranavi 操作の流れについて簡単に紹介します。

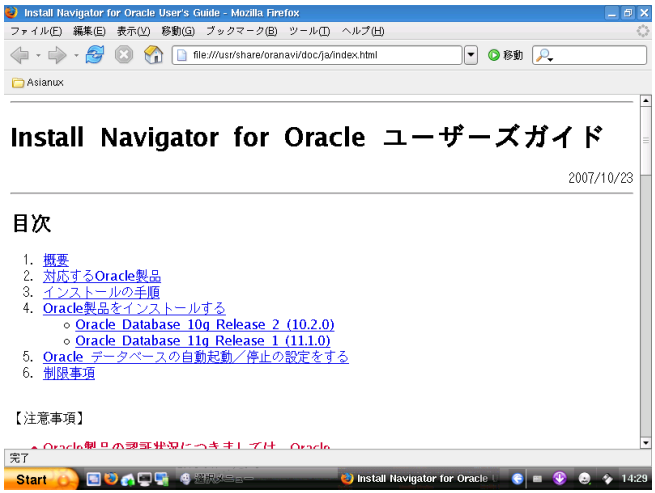


図 11-2　ドキュメントページ画面

oranavi の起動画面（図 11-1）で「Oracle Database をインストールする」を選択して[進む(F)]をクリックします。「ソースメディアの選択」画面（図 11-3）が表示されます。

図 11-3　ソースメディアの選択

Oracle のインストールメディアを指定します。CD-ROM/DVD-ROM からインストールする方法と、ローカルマシンのハードディスク上からインストールする方法の 2 種類が選択できます。

- CD/DVD-ROM : メディアを CD/DVD-ROM ドライブに入れ「進む」を選択します。
- ハードディスク : ローカルマシンのハードディスク上の runInstaller が存在するディレクトリを指定して、「進む」を選択します。

oranavi が対応していない Oracle のバージョンの場合、次の「未対応エラー画面」（図 11-4）が表示されます。

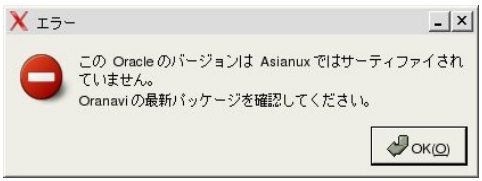


図 11-4 未対応エラー画面

最新版の oranavi を利用されているかご確認ください。また、「11.5 Oracle に関する情報」をご参照ください。

対応していない Oracle のバージョンチェックを無視して oranavi を使用したい場合、オプションとして -f を指定して、次のコマンドを実行します。

```
# /usr/sbin/oranavi -f
```

但し、oranavi による設定後、幾つかのパラメータをお客様で適切な値へ修正して頂く可能性があります。設定が不十分であると Oracle を正常にインストールできない可能性もあります。対応していないバージョンへの oranavi 動作についてはサポート対象外であることをご了承ください。

対応している Oracle のメディアを検出すると、バージョンが表示されます（図 11-5）。

図 11-5 バージョンの確認

正しいバージョンであれば[OK(O)]を選択します。

Oracle の「ユーザーアカウント情報の入力」画面(図 11-6）が表示されます。Oracle をインストール、実行する上で必要なローカルマシン上の Oracle 用 Linux ユーザー、Linux グループを作成します。

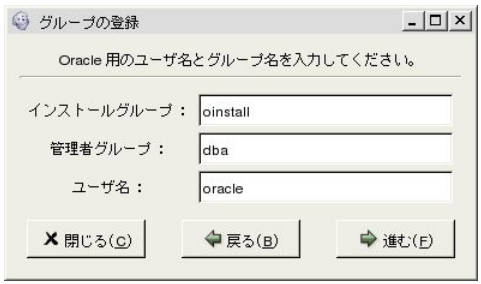


図 11-6　ユーザーアカウント情報の入力

「ユーザ属性情報の入力」画面（図 11-7）が表示されます。作成するユーザのパスワードとホームディレクトリを設定します。

図 11-7　ユーザー属性情報の入力

「Oracle 用の環境変数の設定」画面(図 11-8）が表示されます。

図 11-8　Oracle 用の環境変数の設定

図（11-8）に表示されている項目の意味は以下のとおりです。

- ORACLE_BASE : Oracle プロダクトを格納するトップディレクトリです
- ORACLE_HOME : Oracle データベースを格納するディレクトリです
- ORACLE_SID : 同一マシン上でデータベースをユニークに認識するための名前です

NLS_LANG 等、他の環境変数を独自に設定したい場合、「上記以外の環境変数にデフォルト値を使用する。」のチェックを外して「進む」を選択します。NLS_LANG は、Oracle において、表示する文字コードやメッセージの言語、日付の表示形式などを決定する重要な環境変数です。デフォルトでは、”Japanese_Japan.AL32UTF8 ” がセットされます。

たとえば Oracle DB 11gR1 をインストールする場合、デフォルト値は次の通りです。環境変数は oranavi の設定を最後まで行った後.bash_profile に保存されます。もし、後で環境変数を変更したくなった場合、 .bash_profile を編集します。

```
# /bin/su - oracle
$ /bin/cat .bash_profile | /bin/sed -n "/^# added by oranavi/,/^$/p"
# added by oranavi
export ORACLE_BASE=/opt/app/oracle
export ORACLE_HOME=/opt/app/oracle/product/11.1.0/db_1
export ORACLE_SID=orcl
export NLS_LANG=Japanese_Japan.AL32UTF8
export ORA_NLS33=$ORACLE_HOME/ocommon/nls/admin/data
export PATH=$PATH:$ORACLE_HOME/bin
export ORACLE_DOC=$ORACLE_HOME/doc
CLASSPATH=$ORACLE_HOME/jre:$ORACLE_HOME/JRE
CLASSPATH=$CLASSPATH:$ORACLE_HOME/jlib
CLASSPATH=$CLASSPATH:$ORACLE_HOME/rdbms/jlib
CLASSPATH=$CLASSPATH:$ORACLE_HOME/jdbc/lib/classes12.jar
CLASSPATH=$CLASSPATH:$ORACLE_HOME/jdbc/lib/nls_charset12.jar
export CLASSPATH
LD_LIBRARY_PATH=$ORACLE_HOME/lib:$ORACLE_HOME/ctx/lib
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORACLE_HOME/jdbc/lib
export LD_LIBRARY_PATH
```

ここではデフォルト値を使用して進みます。デフォルト値を使用しない場合の詳細については、oranavi の付属 HTML ドキュメントをご参照ください。

Apache 用の環境変数の設定」画面(図 11-9）が表示されます。「はい」を選択すると、前画面で設定した環境変数を Apache の設定ファイルへ追加します。

図 11-9　Apache 用の環境変数の設定

「Oracle のインストール確認」画面(図 11-10）が表示されます。

[OK(O)]を選択すると Oracle Universal Installer が起動します。 Oracle のインストールについては Oracle 社のマニュアルや技術情報等をご参照ください。また参考までに oranavi 付属の HTML ドキュメントに、デフォルト設定によるインストール方法について紹介しています。

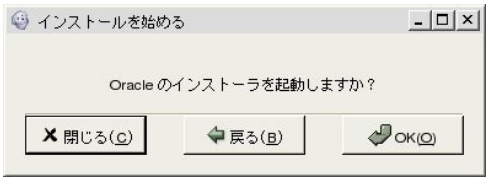


図 11-10　Oracle のインストール確認

11.3 Oracle の自動起動／停止設定

Oracle が正常にインストールされた後にサーバの起動／停止時に Oracle データベースを自動的に起動／停止させるための設定ができます。oranavi の起動画面（図 11-1）で「Oracle データベース自動起動／停止の設定」を選択して[進む(F)]をクリックします。

「SID 選択(変更前)」画面(図 11-11）が表示されます。インストールされた Oracle の SID を選択して、[適用(A)]をクリックします。

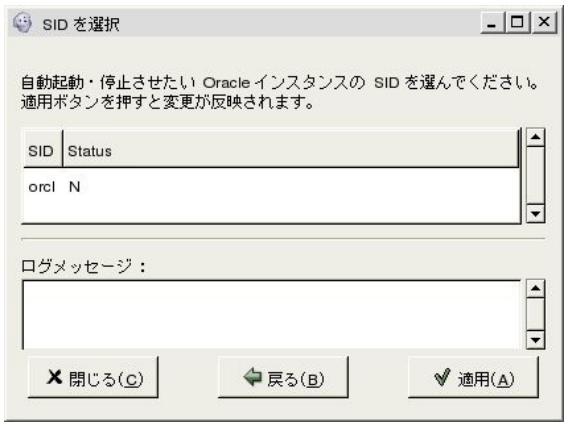


図 11-11 SID 選択（変更前）

「SID 選択(変更後)」画面(図 11-12）のように、状態が「Y」に変更され、ログメッセージに履歴が表示されます。

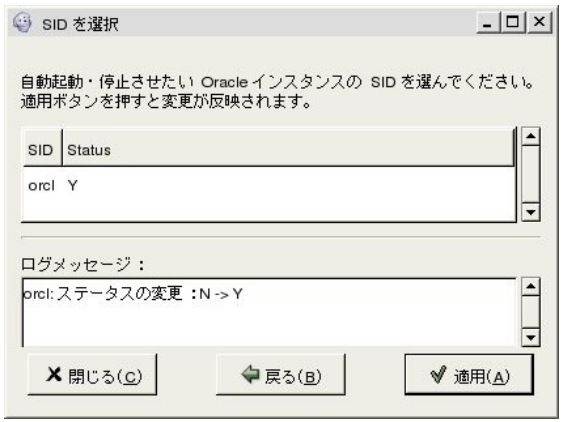


図 11-12 SID 選択（変更後）

これで自動起動／停止の設定は完了です。また、自動起動／停止の設定を取りやめたい場合、同様に、再度「適用」ボタンをクリックすることで状態が「N」に変更されます。設定が完了した後、「閉じる」ボタンを選択します。

「終了確認」画面(図 11-13）が表示されます。

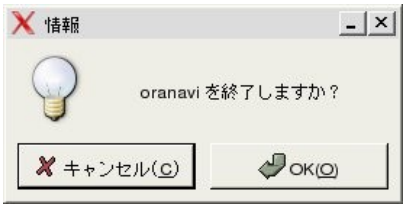


図 11-13　終了確認

※Oranavi 付属の自動起動／停止スクリプトは ASM（Auto Storage Management）には対応しておりません。ASM を導入される場合はご注意ください。

11.4 Oracle の動作確認

Oracle が正常にインストールされた後に Oracle を手動で起動するには、dbora コマンドを実行します。

```
# /sbin/service dbora start
LSNRCTL for Linux: Version 11.1.0.6.0 - Production on 14-5月 -2008 07:49:05
Copyright (c) 1991, 2007, Oracle.  All rights reserved.
/opt/app/oracle/product/11.1.0/db_1/bin/tnslsnrを起動しています。お待ちください...
TNSLSNR for Linux: Version 11.1.0.6.0 - Production
システム・パラメータ・ファイル
は/opt/app/oracle/product/11.1.0/db_1/network/admin/listener.oraです。
ログ・メッセージを/opt/app/oracle/diag/tnslsnr/hostname/listener/alert/log.xmlに書き込
みました。
リスニングしています: (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))
リスニングしています: (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=hostname.example.com)
(PORT=1521)))

(DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC1521)))に接続中
リスナーのステータス
------------------------
別名                      LISTENER
バージョン                TNSLSNR for Linux: Version 11.1.0.6.0 - Production
開始日                    14-5月 -2008 07:49:05
稼働時間                  0 日 0 時間 0 分 0 秒
トレース・レベル          off
セキュリティ              ON: Local OS Authentication
SNMP                        OFF
パラメータ・ファイル
/opt/app/oracle/product/11.1.0/db_1/network/admin/listener.ora
ログ・ファイル           /opt/app/oracle/diag/tnslsnr/hostname/listener/alert/log.xml
リスニング・エンドポイントのサマリー...
(DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=hostname.example.com)(PORT=1521)))
リスナーはサービスをサポートしていません。
コマンドは正常に終了しました。
ORACLE_HOME_LISTNER is not SET, unable to auto-start Oracle Net Listener
Usage: /opt/app/oracle/product/11.1.0/db_1/bin/dbstart ORACLE_HOME
Processing Database instance "orcl": log file
/opt/app/oracle/product/11.1.0/db_1/startup.log
```

たとえば、Oracle DB 11gR1 をインストールした場合、起動時のログは

/opt/app/oracle/product/11.1.0/db_1/startup.log に保存されます。

Oracle を手動で停止するには次のコマンドを実行します。

```
# /sbin/service dbora stop
ORACLE_HOME_LISTNER is not SET, unable to auto-stop Oracle Net Listener
Usage: /opt/app/oracle/product/11.1.0/db_1/bin/dbshut ORACLE_HOME
Processing Database instance "orcl": log file
/opt/app/oracle/product/11.1.0/db_1/shutdown.log
LSNRCTL for Linux: Version 11.1.0.6.0 - Production on 14-5月 -2008 07:48:54
Copyright (c) 1991, 2007, Oracle.  All rights reserved.

(DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC1521)))に接続中
コマンドは正常に終了しました。
```

停止時のログは **/opt/app/oracle/product/11.1.0/db_1/shutdown.log** に保存されます。

Oracle の動作確認のためにデータベースへ接続します。oracle ユーザへ切り替えて sqlplus を実行します。

```
# /bin/su - oracle
$ sqlplus / as sysdba

SQL*Plus: Release 11.1.0.6.0 - Production on 火 5月 20 08:43:04 2008
Copyright (c) 1982, 2007, Oracle.  All rights reserved.
Oracle Database 11g Enterprise Edition Release 11.1.0.6.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options に
接続されました。

SQL> select * from v$version;
BANNER
--------------------------------------------------------------------------------
Oracle Database 11g Enterprise Edition Release 11.1.0.6.0 - 64bit Production
PL/SQL Release 11.1.0.6.0 - Production
CORE    11.1.0.6.0      Production TNS for Linux: Version 11.1.0.6.0 -
Production
NLSRTL Version 11.1.0.6.0 - Production
```

11.5 Oracle に関する情報

Oracle 製品はバージョンによってインストール方法や操作方法が異なります。最新情報は、ミラクル・リナックス社の下記のページを参照してください。Asianux Server 3 と Oracle に関する最新情報が掲載されています。

- 技術フォーラム　http://www.miraclelinux.com/technet/index.html

Oracle 製品に関する技術情報は Oracle Technology Network Japan が充実しています。製品マニュアルやトライアル版、会議室、各種技術情報など、さまざまなコンテンツが掲載されています。

- Oracle Technology Network Japan　http://otn.oracle.co.jp/

第12章　MySQL データーベース サーバーの構築

この章で説明する内容

目的	オープンソースデータベースの MySQL に関する情報を知る
機能	リレーショナルデータベース管理システム
必要な RPM	mysql ――――― MySQL クライアント標準プログラム mysql-server ―― MySQL サーバー mysql-bench ―― テスト、ベンチマーク用プログラム mysql-devel ――MySQL クライアントアプリケーション作成に必要なライブラリ、ヘッダーファイル
設定ファイル	/etc/my.cnf
章の流れ	1 MySQL の概要 2 サーバーの起動／停止 3 データベースの初期化 4 データベースの作成と操作 5 ユーザの管理
関連 URL	MySQL 本家　　http://www.mysql.com/ 日本 MySQL ユーザ会　　http://www.mysql.gr.jp/

12.1 MySQL の概要

MySQL データベースは、オープンソースデータベースの中でも広く利用されているデータベースの一つで、高速性と堅牢性を追及したマルチユーザ・マルチスレッドの SQL データベースです。

12.2 サーバーの起動と停止

MySQL の起動スクリプトは、**/etc/init.d/mysqld** です。起動スクリプトのオプションでは、起動（start）、停止（stop）、再起動（restart）、または現在の状況を確認（status）を指定できます。

MySQL の設定を変更した場合は、変更を反映するために MySQL を再起動する必要があります。

- MySQL を起動するには、次のコマンドを実行します。

```
# /sbin/service mysqld start
```

- MySQL を停止するには、次のコマンドを実行します。

```
# /sbin/service mysqld stop
```

- MySQL を再起動するには、次のコマンドを実行します。

```
# /sbin/service mysqld restart
```

- MySQL の現在の状況を確認するには、次のコマンドを実行します。

```
# /sbin/service mysqld status
```

- システムが起動したときに自動的に MySQL が起動するようにするには、次のコマンドを実行します。

```
# /sbin/chkconfig mysqld on
```

- システムが起動したときに MySQL が起動しないようにするには、次のコマンドを実行します。

```
# /sbin/chkconfig mysqld off
```

12.3 データベースの初期化

MySQLの最初の起動時に自動で初期化が行われます。

```
# /sbin/service mysqld start
MySQL データベースを初期化中:                              [  OK  ]
MySQL を起動中:                                            [  OK  ]
```

データベース初期化後のデータベース構成は次のようになっています。
どのようなデータベースがあるかは**mysqlshow**コマンドで確認することができます。

```
# /usr/bin/mysqlshow
+--------------------+
|     Databases      |
+--------------------+
| information_schema |
| mysql              |
| test               |
+--------------------+
```

初期に作成されるデータベースの内容は次のとおりです。

- **information_schema** －－ MySQLサーバーが保持する他のすべてのデータベースに関する情報を保存しているデータベースです。
- **mysql** －－ MySQLデーモンが使用する設定テーブルが格納されているデータベースです。データベースにアクセスするユーザー、ホストに関する情報を管理しています。
- **test** －－ テスト用（ベンチマークの測定などに利用）のデータベースです。必要のない場合は削除しても構いません。

データベースの初期化を手動で行うには、**mysql_install_db** スクリプトを使用します。

```
# /usr/bin/mysql_install_db
Installing all prepared tables
Fill help tables

To start mysqld at boot time you have to copy support-files/mysql.server
to the right place for your system

PLEASE REMEMBER TO SET A PASSWORD FOR THE MySQL root USER !
To do so, start the server, then issue the following commands:
/usr/bin/mysqladmin -u root password 'new-password'
/usr/bin/mysqladmin -u root -h asianux3srv.example.com password 'new-password'
See the manual for more instructions.

NOTE:  If you are upgrading from a MySQL <= 3.22.10 you should run
the /usr/bin/mysql_fix_privilege_tables. Otherwise you will not be
able to use the new GRANT command!

You can start the MySQL daemon with:
cd /usr ; /usr/bin/mysqld_safe &

You can test the MySQL daemon with the benchmarks in the 'sql-bench' directory:
cd sql-bench ; perl run-all-tests
```

12.4 データベースの作成と操作

新たにデータベースを作成する方法はいくつかありますが、今回はSQL文の**create database**コマンドを使用する例を次に示します。SQL文を実行するには、MySQLのCUIクライアントである**mysql**コマンドを利用します。

以下の例では、データベース名**sample**、文字コード**UTF-8**のデータベースを作成しています。

```
# /usr/bin/mysql -D mysql
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3 to server version: 5.0.22

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> create database sample default character set utf8;
Query OK, 1 row affected (0.00 sec)
mysql> exit
```

データベースの確認を行います。

```
# /usr/bin/mysqlshow
+--------------------+
|     Databases      |
+--------------------+
| information_schema |
| mysql              |
| sample             |
| test               |
+--------------------+
```

sample データベースが作成されましたので、**mysql** コマンドを利用して接続確認を行います。以下の例では、**status** コマンドにより接続中のデータベース情報を表示しています。

```
# /usr/bin/mysql -D sample
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 5 to server version: 5.0.22

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> status
--------------
mysql  Ver 14.12 Distrib 5.0.22, for asianux-linux-gnu (i686) using readline 5.0

Connection id:          5
Current database:       sample
Current user:           root@localhost
SSL:                    Not in use
Current pager:          stdout
Using outfile:          ''
Using delimiter:        ;
Server version:         5.0.22
Protocol version:       10
Connection:             Localhost via UNIX socket
Server characterset:    latin1
Db     characterset:    utf8
Client characterset:    latin1
Conn.  characterset:    latin1
UNIX socket:            /var/lib/mysql/mysql.sock
Uptime:                 11 min 43 sec

Threads: 1  Questions: 31  Slow queries: 0  Opens: 0  Flush tables: 1  Open
tables: 17  Queries per second avg: 0.044
--------------
```

後は作成したデータベースに対し、ユーザー、テーブル、インデックスなどのオブジェクトを作成することで、データベースとして利用することができます。

➢ テーブルの作成

テーブルの作成には、CREATE TABLE コマンドを使用します。以下の例では、作成した sample データベースに、テーブルを作成しています。

```
mysql> USE sample;

mysql> CREATE TABLE EMP (EMPNO INT(4) ,
    ->          ENAME VARCHAR(10),
    ->          JOB VARCHAR(9),
    ->          MGR FLOAT(4),
    ->          SAL FLOAT(7,2),
    ->          COMM FLOAT(7,2),
    ->          DEPTNO FLOAT(2) );
Query OK, 0 rows affected (0.00 sec)

mysql> SHOW tables;
+------------------+
| Tables_in_sample |
+------------------+
| EMP              |
+------------------+
1 row in set (0.00 sec)
```

➢ テーブルにデータを格納する

データの格納には INSERT INTO コマンドを使用します。以下の例では、作成した EMP テーブルにデータを格納しています。

```
mysql> INSERT INTO EMP VALUES
    -> (7369,'SMITH','CLERK',7902,800,NULL,20);
Query OK, 1 row affected (0.01 sec)

mysql> select * from EMP;
+-------+-------+-------+------+--------+------+--------+
| EMPNO | ENAME | JOB   | MGR  | SAL    | COMM | DEPTNO |
+-------+-------+-------+------+--------+------+--------+
|  7369 | SMITH | CLERK | 7902 | 800.00 | NULL |     20 |
+-------+-------+-------+------+--------+------+--------+
1 row in set (0.00 sec)
```

➢ テーブルの削除

テーブルの削除には、DROP TABLE コマンドを使用します。以下の例では、作成した EMP テーブルを削除しています。

```
mysql> SHOW tables;
+------------------+
| Tables_in_sample |
+------------------+
| EMP              |
+------------------+
1 row in set (0.00 sec)

mysql> DROP TABLE EMP;
Query OK, 0 rows affected (0.00 sec)

mysql> SHOW tables;
Empty set (0.00 sec)
```

➢ データベースの削除

データベースの削除には、DROP DATABASE コマンドを使用します。以下の例では、作成した sample データベースを削除しています。

```
mysql> SHOW databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| sample             |
| test               |
+--------------------+
4 rows in set (0.01 sec)

mysql> DROP DATABASE sample;
Query OK, 0 rows affected (0.00 sec)

mysql> SHOW databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
```

```
| test               |
+--------------------+
3 rows in set (0.00 sec)
```

12.5 ユーザの管理

MySQL はユーザをホスト名とユーザー名の組み合わせで認識します。ユーザの管理情報は mysql データベースの user テーブルに格納されています。また、下記のように、デフォルトでは管理者ユーザー（root）と匿名ユーザー（ユーザー名が空白）が存在しています。

```
mysql> SELECT user,host from mysql.user;
+------+------------------+
| user | host             |
+------+------------------+
|      | mysql.example.com |
| root | mysql.example.com |
|      | localhost        |
| root | localhost        |
+------+------------------+
4 rows in set (0.00 sec)
```

➢ ユーザの追加

ユーザの追加には、GRANT コマンドを使用します。

GRANT コマンドの書式

```
GRANT 権限[,権限[,...]] ON データベース名.テーブル名 ユーザー名@接続ホスト
[IDENTIFIED BY "パスワード"];
```

以下の例では、SELECT 権限を付与したユーザ user1 を追加しています。

```
mysql>  GRANT SELECT ON sample.* TO user1@localhost IDENTIFIED BY "12345";
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT user FROM  mysql.user;
+-------+
| user  |
+-------+
```

```
|       |
| root  |
|       |
| root  |
| user1 |
+-------+
5 rows in set (0.01 sec)
```

以下の例では、全ての権限を持ったユーザを追加しています。

```
mysql> GRANT ALL PRIVILEGES ON *.* TO user2@localhost IDENTIFIED BY '12345' WITH
GRANT OPTION;
```

➢ パスワードの設定

パスワードの設定は、**mysqladmin** コマンドもしくは、SET PASSWORD コマンドを使用します。

mysqladmin コマンドの書式

```
# mysqladmin -u ユーザ名 -p password '新しいパスワード'
```

以下の例では、mysqladmin コマンドを使用して、ユーザ root のパスワードを設定しています。

```
# mysqladmin -u root -p password '12345'
```

SET PASSWORD の書式

```
mysql> SET PASSWORD FOR ユーザ名@ホスト名 = PASSWORD('パスワード');
```

以下の例では、SET PASSWORD コマンドを使用して、user1 のパスワードを設定しています。

```
mysql> SET PASSWORD FOR user1@localhost=password('1234');
```

パスワードを設定した場合、データベース接続時パスワードの入力が必要になります。以下のように mysql コマンドで接続する際、-p オプションが必要になります。

```
#  mysql -D mysql -u root -p
Enter password:
```

➢ ユーザの削除

ユーザの削除には、DROP USER を使用します。

以下の例では、ユーザ user1 を削除しています。

```
mysql> DROP USER user1@localhost;
Query OK, 0 rows affected (0.00 sec)

mysql> select user,host from mysql.user;
+------+------------------+
| user | host             |
+------+------------------+
|      | mysql.example.com |
| root | mysql.example.com |
|      | localhost        |
| root | localhost        |
+------+------------------+
4 rows in set (0.00 sec)
```

第13章　PostgreSQL データーベース サーバーの構築

この章で説明する内容

目的	オープンソースデータベースの PostgreSQL に関する情報を知る
機能	オブジェクトリレーショナルデータベース管理システム
必要な RPM	Postgresql ――――― PostgreSQL クライアント postgresql-libs ――― PostgreSQL ライブラリー postgresql-server ―― PostgreSQL サーバー
設定ファイル	/var/lib/pgsql/data/postgresql.conf /var/lib/pgsql/data/pg_hba.conf /var/lib/pgsql/data/pg_ident.conf
章の流れ	1 PostgreSQL の概要 2 PostgresSQL のユーザ 3 サーバーの起動／停止 4 データベースの初期化 5 データベース、データベースロールの作成と操作
関連 URL	PostgreSQL 本家　　http://www.postgresql.org/index.html 日本 PostgreSQL ユーザ会　　http://www.postgresql.jp/

13.1 PostgreSQL の概要

PostgreSQL データベースは、カリフォルニア大学バークレイ校で開発された POSTGRES, Version4.2 をベースとしたオープンソースのオブジェクトリレーショナルデータベース管理システム（ORDBMS）の 一つです。標準 SQL やトリガ、手続き言語等、DBMS として必要な機能が豊富です。また、最新バージョンの翻訳ドキュメントのリリースが早いなど、世界各地におけるユーザ会のコミュニティ活動が盛んなことも特徴の 1 つです。

13.2 PostgreSQL のユーザ

PostgreSQL データベースの起動や操作は、非特権ユーザ(root 以外)で行う必要があります。PostgreSQL をインストールする際、システムには postgres ユーザが作成されます。デフォルトで用意されている起動スクリプトを利用する場合、 postgres ユーザによってデータベースが起動されます。また、手動でデータベースを操作する場合も同様に postgres ユーザでログインする、又は su コマンドで切り替えて実行する必要があります。

13.3 サーバーの起動と停止

PostgreSQL の起動は**/etc/init.d/postgresql** スクリプトまたは、**pg_ctl** コマンドを使用することで可能です。pg_ctl コマンドを使用する場合、環境変数 PGDATA を設定する、または-D オプションを指定して、データベースの格納領域を指定する必要があります。デフォルトの格納領域は /var/lib/pgsql/data になります。

起動スクリプトやコマンドのオプションでは、起動（start）、停止（stop）、再起動（restart）、または現在の状況を確認（status）を指定できます。

PostgreSQL の設定を変更した場合は、変更を反映するために PostgreSQL を再起動する必要があります。

- PostgreSQL を起動するには、次のコマンドを実行します。

```
# /sbin/service postgresql start
または
# /bin/su - postgres
-bash-3.1$ /usr/bin/pg_ctl -D /var/lib/pgsql/data start
```

- /etc/init.d/postgresql スクリプトを使用して、初めてデータベースを起動させる場合のみ、初期データベースが作成されます。

```
# /sbin/service postgresql start
データベースを初期化中:                                    [  OK  ]
postgresql サービスを開始中:                               [  OK  ]
```

- PostgreSQL を停止するには、次のコマンドを実行します。

```
# /sbin/service postgresql stop
または
# /bin/su - postgres
-bash-3.1$ /usr/bin/pg_ctl -D /var/lib/pgsql/data stop
```

- PostgreSQL を再起動するには、次のコマンドを実行します。

```
# /sbin/service postgresql restart
または
# /bin/su - postgres
-bash-3.1$ /usr/bin/pg_ctl -D /var/lib/pgsql/data restart
```

- PostgreSQL の現在の状況を確認するには、次のコマンドを実行します。

```
# /sbin/service postgresql status
または
# /bin/su - postgres
-bash-3.1$ /usr/bin/pg_ctl -D /var/lib/pgsql/data status
```

- システムが起動したときに自動的に PostgreSQL が起動するようにするには、次のコマンドを実行します。

```
# /sbin/chkconfig postgresql on
```

- システムが起動したときに PostgreSQL が起動しないようにするには、次のコマンドを実行します。

```
# /sbin/chkconfig postgresql off
```

13.4 データベースの初期化

データベースの初期化を行うには、**initdb** コマンドを使用します。/etc/init.d/postgresql スクリプトで起動した場合、既に作成されているのでこの作業を行う必要はありません。次の例では、文字コード **UTF-8**、ロケール指定なしでデータベースの初期化を行っています。

```
# /bin/su - postgres
-bash-3.1$ /usr/bin/initdb -E utf8 --no-locale -D /var/lib/pgsql/data/
データベースシステム内のファイルの所有者は"postgres"ユーザでした。
このユーザがサーバプロセスを所有しなければなりません。

データベースクラスタはロケールCで初期化されます。

ディレクトリ/var/lib/pgsql/dataの権限を設定しています ... ok
ディレクトリ/var/lib/pgsql/data/globalを作成しています ... ok
ディレクトリ/var/lib/pgsql/data/pg_xlogを作成しています ... ok
ディレクトリ/var/lib/pgsql/data/pg_xlog/archive_statusを作成しています ... ok
ディレクトリ/var/lib/pgsql/data/pg_clogを作成しています ... ok

  ～ 省 略 ～

警告: ローカル接続向けに"trust"認証が有効です。
pg_hba.confを編集する、もしくは、次回initdbを実行する時に-Aオプショ
ンを使用することで変更することができます。

Success. You can now start the database server using:

   postmaster -D /var/lib/pgsql/data
or
   pg_ctl -D /var/lib/pgsql/data -l logfile start
```

既にデータベースが作成されている場合は、次のメッセージが出力されます。

```
# /bin/su - postgres
-bash-3.1$ /usr/bin/initdb -E utf8 --no-locale -D /var/lib/pgsql/data/
データベースシステム内のファイルの所有者は"postgres"ユーザでした。
このユーザがサーバプロセスを所有しなければなりません。

データベースクラスタはロケールCで初期化されます。
initdb: ディレクトリ"/var/lib/pgsql/data"が空ではありませんでした。
新規にデータベースシステムを作成したいのであれば、ディレクトリ"/var/lib/pgsql/data"
を削除するか空にしてください。または、initdbを"/var/lib/pgsql/data"以外の引数で実行して
ください。
```

初期化が完了したら PostgreSQL を起動します。

```
-bash-3.1$ /usr/bin/pg_ctl -D /var/lib/pgsql/data/ start
postmasterは起動中です。
```

データベース初期化後のデータベース構成は次のようになります。

どのようなデータベースがあるかは postgres ユーザーに切り替えて **psql** コマンドで確認することができます。

```
# /bin/su - postgres
-bash-3.1$ /usr/bin/psql -l
            データベース一覧
  名前    | 所有者 | エンコーディング
-----------+----------+--------------------------
 postgres  | postgres  | UTF8
 template0 | postgres  | UTF8
 template1 | postgres  | UTF8
(3 行)
```

初期状態で、テンプレートデータベースとして postgres、template0、template1 が作成されています。

データベースへ接続するアカウント(postgres)のパスワードを設定します。

```
-bash-3.1$ /usr/bin/psql
Welcome to psql 8.1.11, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help with psql commands
       \g or terminate with semicolon to execute query
       \q to quit
postgres=# alter user postgres encrypted password 'asianux';
ALTER ROLE
postgres=# \q
-bash-3.1$ vi /var/lib/pgsql/data/pg_hba.conf
local   all         all                               md5
host    all         all         127.0.0.1/32          md5
host    all         all         ::1/128               md5
```

psql コマンドに-U オプションを指定しない場合、システムのユーザアカウントでデータベースへ接続します。

上記の例では postgres ユーザで psql コマンドを実行しているので postgres ユーザでデータベースへログインします。初期状態では、パスワードが設定されていないため ALTER USER 文を使用して ‘asianux’ というパスワードを設定しています。

パスワードの設定後、データベース接続時に認証を行うために pg_hba.conf を設定します。

2 つの設定が完了したら PostgreSQL を再起動して、再度データベースへ接続しようとすると、パスワードが尋ねられるようになります。

```
-bash-3.1$ /usr/bin/pg_ctl -D /var/lib/pgsql/data/ restart
postmasterの停止を待機しています....LOG:  logger shutting down
完了
postmasterは停止しました
postmasterは起動中です。
-bash-3.1$ /usr/bin/psql
パスワード:
```

root ユーザからでも psql コマンドに-U オプションを指定することで、次のようにデータベースへ接続することができます。

```
-bash-3.1$ exit
# whoami
root
# psql -U postgres
Password for user postgres: ********
Welcome to psql 8.1.11, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help with psql commands
       \g or terminate with semicolon to execute query
       \q to quit

postgres=#
```

13.5 データベース、データベースロールの作成と操作

新たにデータベースを作成する方法はいくつかありますが、今回は **createdb** コマンドを使用してデータベース名 **sample**、文字コード **UTF-8** のデータベースを作成する例を次に示します。

```
-bash-3.1$ /usr/bin/createdb -E utf8 sample
パスワード:
CREATE DATABASE
```

データベースの確認を行います。

```
-bash-3.1$ /usr/bin/psql -l
パスワード:
            データベース一覧
  名前    | 所有者 | エンコーディング
-----------+----------+--------------------------
 postgres  | postgres  | UTF8
 sample    | postgres  | UTF8
 template0 | postgres  | UTF8
 template1 | postgres  | UTF8
(4 行)
```

sample データベースが作成されましたので **psql** コマンドで接続確認を行います。

```
-bash-3.1$ /usr/bin/psql -d sample
パスワード:
Welcome to psql 8.1.11, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help with psql commands
       \g or terminate with semicolon to execute query
       \q to quit

sample=#
```

PostgreSQL データベースのユーザを作成、削除するには **createuser** コマンドや **dropuser** コマンドを使用すると便利です。ここでは pguser を作成します。最初に pguser 自身のパスワードを 2 回入力します。その後、スーパーユーザ権限、データベース作成権限、ロール作成権限の有無が尋ねられます。最後に postgres のパスワードを入力すると pguser が作成されます。厳密には PostgreSQL では、ユーザではなくロールという概念で管理されていますが、詳細については日本 PostgreSQL ユーザ会のドキュメントをご参照ください。

```
-bash-3.1$ /usr/bin/createuser pguser -P
Enter password for new role: ********
再入力してください: ********
Shall the new role be a superuser? (y/n) n
Shall the new role be allowed to create databases? (y/n) n
Shall the new role be allowed to create more new roles? (y/n) n
パスワード:
CREATE ROLE
```

作成した pguser で sample データベースへ接続します。

```
-bash-3.1$ /usr/bin/psql -U pguser -d sample
Password for user pguser: ********
Welcome to psql 8.1.11, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help with psql commands
       \g or terminate with semicolon to execute query
       \q to quit

sample=#
```

作成した pguser を削除するには、dropuser コマンドを使用します。

```
-bash-3.1$ /usr/bin/dropuser pguser
パスワード:
DROP ROLE
```

テーブルを作成するには 次のように CREATE TABLE 文を使用します。また \d でテーブル構成(列やデータ型)を確認できます。

```
-bash-3.1$ /usr/bin/psql -d sample
パスワード:
Welcome to psql 8.1.11, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help with psql commands
       \g or terminate with semicolon to execute query
       \q to quit

sample=# create table products (
sample(#   product_no serial,
sample(#   product_name text,
sample(#   product_version text,
sample(#   product_date date
sample(# );
NOTICE:  CREATE TABLE will create implicit sequence "products_product_no_seq"
for serial column "products.product_no"
CREATE TABLE
sample=# \d products
                          テーブル "public.products"
```

```
    カラム      |    型    |                                修飾語
-----------------+---------
+-------------------------------------------------------------------
 product_no      | integer | not null default
nextval('products_product_no_seq'::regclass)
 product_name    | text    |
 product_version | text    |
 product_date    | date    |

sample=#
```

テーブルを削除するには DROP TABLE 文を使用して、次のように実行します。

```
sample=# drop table products;
DROP TABLE
sample=# \d products
"products" という名前のリレーションが見つかりません。
```

テーブルへのデータ追加、更新、削除には、次のように DML(Data Manipulation Language)文を使用します。

```
sample=# \d products
                          テーブル "public.products"
    カラム     |   型    |                                修飾語
---------------+---------
+-------------------------------------------------------------
 product_no      | integer | not null default
nextval('products_product_no_seq'::regclass)
 product_name    | text    |
 product_version | text    |
 product_date    | date    |

sample=# insert into products values
sample-# (nextval('products_product_no_seq'), 'Asianux Server', '3',
'2007/09/18');
INSERT 0 1
sample=# insert into products values
sample-# (nextval('products_product_no_seq'), 'Asianux Server', '3 SP1',
'2007/08/25');
INSERT 0 1

sample=# select * from products;
 product_no |  product_name  | product_version | product_date
------------+----------------+-----------------+--------------
          1 | Asianux Server | 3               | 2007-09-18
          2 | Asianux Server | 3 SP1           | 2007-08-25
(2 行)

sample=# update products set product_date='2008/09/24' where product_no='2';
UPDATE 1

sample=# select * from products;
 product_no |  product_name  | product_version | product_date
------------+----------------+-----------------+--------------
          1 | Asianux Server | 3               | 2007-09-18
          2 | Asianux Server | 3 SP1           | 2008-09-24
(2 行)

sample=# delete from products where product_date <= '2008/01/01';
DELETE 1

sample=# select * from products;
 product_no |  product_name  | product_version | product_date
------------+----------------+-----------------+--------------
          2 | Asianux Server | 3 SP1           | 2008-09-24
(1 行)
```

SQL 手続き言語として PL/pgSQL を　　使用します。標準で付属していますが、最初に使用するデータベースへインストールする必要があります。次の例では sample データベースへインストールします。

```
-bash-3.1$ createlang plpgsql sample
パスワード:
```

CREATE FUNCTION 文を使用して、テーブル foo へ空データを追加、更新(初期化)する関数(foo_initialize)を作成します。関数への引数として max を取り、max 行分のデータが初期化されます。PL/pgSQL の仕様や制御構造の詳細については日本 PostgreSQL ユーザ会のドキュメントをご参照ください。

```
sample=# create table foo(id integer primary key, name text, recital text);
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "foo_pkey" for
table "foo"
CREATE TABLE

sample=# create function foo_initialize(max integer) returns int as
sample-# $$
sample$# declare
sample$#   i integer;
sample$# begin
sample$#   for i in 1..max loop
sample$#     begin
sample$#       insert into foo values (i, '', '');
sample$#     exception
sample$#       when unique_violation then
sample$#         update foo set name='', recital='' where id=i;
sample$#     end;
sample$#   end loop;
sample$#   return 0;
sample$# end;
sample$# $$
sample-# language plpgsql;
CREATE FUNCTION
```

作成した関数(foo_initialize)を削除するには、次のように実行します。

```
sample=# drop function foo_initialize(max integer);
DROP FUNCTION
```

作成した foo テーブル構成は次のとおりです。作成後なのでデータは存在しない状態です。

```
sample=# \d foo
   テーブル "public.foo"
 カラム  |   型    |  修飾語
---------+---------+----------
 id      | integer | not null
 name    | text    |
 recital | text    |
インデックス:
   "foo_pkey" PRIMARY KEY, btree (id)

sample=# select * from foo;
 id | name | recital
----+------+---------
(0 行)
```

作成した関数(foo_initialize)で foo テーブルの 3 行のデータを初期化(追加)します。

```
sample=# select foo_initialize(3);
 foo_initialize
----------------
              0
(1 行)

sample=# select * from foo;
 id | name | recital
----+------+---------
  1 |      |
  2 |      |
  3 |      |
(3 行)
```

foo テーブルのデータを更新します。

```
sample=# update foo set name='xxx' where id=2;
UPDATE 1
sample=# update foo set recital='yyy' where id=3;
UPDATE 1
sample=# select * from foo;
 id | name | recital
----+------+---------
  1 |      |
  2 | xxx  |
```

```
  3 |       | yyy
(3 行)
```

作成した関数(foo_initialize)で foo テーブルの 5 行のデータを初期化(更新、追加)します。

```
sample=# select foo_initialize(5);
 foo_initialize
----------------
              0
(1 行)

sample=# select * from foo;
 id | name | recital
----+------+---------
  1 |      |
  2 |      |
  3 |      |
  4 |      |
  5 |      |
(5 行)
```

このように PL/pgSQL を使用することで SQL による操作を自動化したり、SQL へ制御構造を追加することができます。

第14章　NFS によるファイル共有

この章で説明する内容

目的	ネットワーク上にある他のシステムのディスク資源を共有する
機能	他のシステムへディスクのマウントを許可する 他のシステムからディスクをマウントする
必要な RPM	portmap —— RPC（Remote Procedure Call）を使用するプログラムの管理ツール nfs-utils —— NFS サーバーのユーティリティとデーモン autofs —— automount ユーティリティとデーモン
設定ファイル	/etc/exports /etc/fstab
章の流れ	1 NFS の概要 2 NFS サーバー 3 NFS クライアント 4 オートマウントと NFS の組み合わせ
関連 URL	**NFS HOWTO** http://www.linux.or.jp/JF/JFdocs/NFS-HOWTO/

14.1　NFS の概要

NFS（Network File System）はネットワーク上の他のホストとファイル資源を共有するために利用されます。クライアントはサーバーのローカルファイルをマウントして、自分のローカルファイルであるかのように参照できるようになります。Linux システムはクライアントとしてもサーバーとしても設定でき、また両方同時に機能させることもできます。

NFS を機能させるためには、**portmap** というデーモンプログラムを起動しておかなければなりません。**portmap** は RPC プログラム番号を DARPA プロトコルポート番号に変換するサーバーであり、NFS サーバーを起動する前に起動しておく必要があります。また NFS クライアント側でも **portmap** を動作させる必要があります。

また、ここに記述されているとおりの設定を行っても、NFS サーバー側のネットワークファイアーウォールによりアクセスできない場合がありますのでご注意ください。ネットワークファイアーウォールの詳細については、第 26 章「ファイアーウォール」を参照してください。

14.2　NFS サーバー

14.2.1　portmap の起動と停止

portmap の起動スクリプトは、**/etc/rc.d/init.d/portmap** です。起動スクリプトのオプションでは、起動（start）、停止（stop）、再起動（restart）、現在の状況を確認（status）を指定できます。

portmap の設定を変更した場合、その変更内容は portmap を再起動するまで有効とはなりません。設定変更を施した場合、portmap を再起動させる必要があります。

- portmap を起動するには、次のコマンドを実行します。

```
# /sbin/service portmap start
```

- portmap を停止するには、次のコマンドを実行します。

```
# /sbin/service portmap stop
```

- portmap を再起動するには、次のコマンドを実行します。

```
# /sbin/service portmap restart
```

- portmap が稼働中かどうかは次のようにして調べられます。

```
# /sbin/service portmap status
```

14.2.2 portmap へのアクセス制限

portmap は **tcp_wrappers** パッケージに含まれるアクセスコントロールライブラリ（libwrap.a）を使用して、リモートシステムからのアクセスを制御します。

たとえば、**/etc/hosts.deny** に次のように記述します。

```
portmap : ALL
```

そして、**/etc/hosts.allow** に次にように記述します。

```
portmap : 192.168.0.2
```

上記の設定により、そのシステムの portmap へのアクセスはリモートホスト 192.168.0.2 のみ許可することになります。**/etc/hosts.deny**、**/etc/hosts.allow** を編集した場合でも、特に portmap を再起動する必要はありません。

/etc/hosts.deny、**/etc/hosts.allow** の記述についての詳細は **hosts_access** のオンラインマニュアルを参照してください。

14.2.3 NFS サーバーの起動と停止

NFS サーバーの起動スクリプトは、**/etc/rc.d/init.d/nfs** と **/etc/rc.d/init.d/nfslock** です。

起動スクリプトのオプションでは、起動（start）、停止（stop）、再起動（restart）、現在の状況を確認（status）を指定できます。

NFS の設定を変更した場合、その変更内容は NFS を再起動するまで有効とはなりません。設定を変更した場合、NFS を再起動させる必要があります。

- NFS を起動するには、次のコマンドを実行します。

```
# /sbin/service nfs start
# /sbin/service nfslock start
```

- NFS を停止するには、次のコマンドを実行します。

```
# /sbin/service nfs stop
# /sbin/service nfslock stop
```

- NFS を再起動するには、次のコマンドを実行します。

```
# /sbin/service nfs restart
# /sbin/service nfslock restart
```

- NFS が稼働中かどうかは次のようにして調べられます。

```
# /sbin/service nfslock status
```

14.2.4　NFS サーバーの設定

➢ /etc/exports の設定

/etc/exports には、NFS クライアントに対して export が可能なファイルシステムのアクセスコントロールリストを記述します。「ファイルシステムを export する」とはサーバーがクライアントに対してファイルシステムの共有を許可することを指します。**/etc/exports** ファイルの書式は次のとおりです。

```
［ディレクトリ名］［ホスト名 （オプション）］
```

[ディレクトリ名] は共用させるディレクトリの名前です。ホスト名はそのディレクトリをマウントすることを許可するホストの名前です。オプションにはさまざまな指定ができます。

/etc/exports の例を次に示します。

```
/ master(rw) trusty(rw,no_root_squash)
/usr *.your.domain.name(ro)
/pub (ro,all_squash)
/pub/private (noaccess)
```

1 行目は master、trusty というホストに対してのマウントと読み書きを許可しています。

2 行目は your.domain.name というドメイン名を持つホストすべてに読み込みのみ許可しています。

3 行目はすべてのホストに対して/pub の読み込みのみを許可しています。

4 行目はすべてのホストに対して/pub/private ディレクトリへのアクセスを拒否しています。

/etc/exports を編集したときには、nfs を再起動すれば設定は反映されますが、**exportfs** コマンドを次のように実行するだけでも反映されます。

```
# /usr/sbin/exportfs -r
```

➢ NFS サーバーの動作確認

NFS の export の状態を確認するには、NFS サーバ側で、次のように **exportfs** コマンドを実行します。

```
# /usr/sbin/exportfs
```

また export したファイルシステムをどのホストがマウントしているかを調べるには、次のように **showmount** コマンドを実行します。

```
# /usr/sbin/showmount -a
```

exportfs、**showmount** の使用方法に関する詳細は、オンラインマニュアルを参照してください。

14.3 NFS クライアント

14.3.1 NFS クライアントの起動と停止

NFS クライアントの起動には、あらかじめ **/etc/rc.d/init.d/portmap** を起動しておく必要があります。また、システム開始時に **/etc/fstab** に書かれた NFS 領域をマウントするには、**/etc/rc.d/init.d/netfs** が起動するよう設定する必要があります。起動や終了などは 14.2.1「portmap の起動と停止」と同様に操作してください。

14.3.2 NFS クライアントの設定

NFS クライアントが NFS サーバーのファイルシステムをマウントする場合は、**mount** コマンドを使用します。たとえば、ホスト名 server の/pub ディレクトリを/mnt/pub にマウントするときには、次のコマンドを実行します。

```
# /bin/mount -t nfs server:/pub /mnt/pub
```

このとき、/mnt/pub ディレクトリはローカルファイルシステム上にあらかじめ作成しておく必要があります。**mount** コマンドの使用方法に関する詳細は、オンラインマニュアル参照してください。

また、**/etc/fstab** ファイルにあらかじめにマウントするファイルシステムに関する情報を記述しておけば、システムの起動時に自動的にマウントが実行されます。

たとえば上記の server:/pub のマウントは次のような行を**/etc/fstab** に追加することにより実行されます。

```
server:/pub /mnt/pub nfs rw,soft
```

rw は読み書きモードであることを指定しています。**soft** は NFS サーバー側で問題があるなどの理由で応答がない場合にタイムアウトすることを指定しています。**/etc/fstab** の記述に関する詳細は、**nfs** のオンラインマニュアルを参照してください。

新たに**/etc/fstab** に追加したディレクトリのマウントは、**mount** コマンドで簡単にシステムに反映できます。

たとえば上記の/mnt/pub の記述を追加した後、次のコマンドを実行すれば、/mnt/pub のマウントは完了します。

```
# /bin/mount /mnt/pub
```

14.3.3 NFS クライアントの動作確認

NFS のマウントの状態を確認するには、次のように **mount** コマンドを実行します。

```
# /bin/mount
```

14.3.4 NFS クライアントの停止方法

既にマウントされている NFS クライアントをアンマウントする場合は、**umount** コマンドを使用します。例えば、/mnt/pub ディレクトリにマウントされているファイルシステムをアンマウントするには、次のコマンドを実行します。

```
# /bin/umount /mnt/pub
```

14.4 オートマウントと NFS の組み合わせ

オートマウントとは、マウントポイントとなるディレクトリパスにアクセスした際に自動でマウントポイントの作成及びマウントする機能となります。本機能と nfs とを組み合わせた場合の利点は、実際にアクセスが発生した場合にマウントするので、ネットワークの負荷が最小限に抑えられます。Asianux Server 3 のオートマウントは **autofs** というサービス名で実行します。なお、本設定は NFS クライアント側で実施します。

14.4.1 autofs の設定

➢ /etc/auto.master の設定

/etc/auto.master ファイルには、マウントポイントとその詳細について記述された別ファイルへのパスを設定します。書式は以下のとおりです。

```
［マウントポイント］［マップファイル］
```

[マウントポイント]にはオートマウントで使用されるディレクトリの絶対パスを記述します。[マップファイル]には[マウントポイント]で指定したディレクトリのサブディレクトリに関する設定を記述します。
/etc/auto.master の例を次に示します。

```
/share        /etc/auto.nfs
```

/share というディレクトリ下のオートマウントの設定は、/etc/auto.nfs ファイルに記述する、という意味をあらわします。なお、/share というディレクトリは事前に作成しておく必要があります。

➢ マップファイルの設定

マップファイルには、/etc/auto.master に設定したマウントポイント以下に作成するサブディレクトリ名とその設定に関して記述します。
（マップファイル名は/etc/auto.master で設定したパスによって異なりますが、ここでは**/etc/auto.nfs** とします。）
マップファイルの書式は以下のとおりです。マップファイルの詳細は、**autofs** のオンラインマニュアルを参照してください。

```
サブディレクトリ名　　マウントオプション　　ホスト名またはIPアドレス:ディレクトリへのパス
```

サブディレクトリ名「subdir」に対して、ホスト名「server」上の「/shareddir」ディレクトリをマウントする場合は以下のように記述します。以下の例で、マウントオプション部分の **rw** は読み書き可能、**soft** は NFS サーバー側で問題があるなどの理由で応答がない場合にタイムアウトすること、**intr** はキーボードからの割り込みを可能にするかどうか、をあらわします。詳細は、**mount** のオンラインマニュアルを参照してください。

```
subdir    -rw,soft,intr    server:/shareddir
```

➢ autofs サービスの起動設定

autofs サービスが OS 起動時に自動で開始するようにするには次のコマンドを実行します。
※既に実施している場合は不要です

```
# /sbin/chkconfig autofs on
```

➢ autofs サービスの起動

autofs サービスを再起動します。既に起動している場合も、今回の設定を反映させる為には再起動が必要です。

```
# /sbin/service autofs restart
```

➢ 動作確認

cd コマンド等で、オートマウントの設定をしたディレクトリに移動し、自動でマウントされるか確認します。

```
# cd /share/subdir
```

第15章　メールサーバーの構築（Sendmail）

この章で説明する内容

目的	Sendmail を使ったメールサーバーの構築方法について理解する
機能	メールの配送
必要な RPM	sendmail sendmail-cf
設定ファイル	/etc/mail/sendmail.cf
章の流れ	1 Mail Transfer Agent の概要 2 Mail Transfer Agent Switcher の利用方法 3 Sedmail の概要 4 Sendmail の起動と停止 5 Sendmail の設定
関連 URL	**sendmail.org** http://www.sendmail.org/

15.1　MTA (Mail Transfer Agent) の概要

MTA（Mail Transfer Agent）とは、インターネット上で、ホスト間や企業間などのメールの配送を受け持つアプリケーションです。この MTA に対して、実際にクライアント上でメールの読み書きを行うアプリケーションは MUA（Mail Usesr Agent）と呼ばれます。インターネットのメールシステムではこのクライアント（MUA）とサーバー（MTA）が連携して電子メールシステムを構成しています。

15.2　Mail Transfer Agent Switcher の利用方法

Postfix と **Sendmail** など、同じ役割を持つ MTA（Mail Transfer Agent）を 2 種類以上インストールした場合、Mail Transfer Agent Switcher を利用して、Postfix と Sendmail のうち、どちらか一方を MTA として利用するように設定を行う必要があります。Mail Transfer Agent Switcher を起動するには次のコマンドを実行します。

※Asianux Server 3 の初期状態では Postfix が MTA となっています。

- X Window System 環境の場合

```
# /usr/sbin/system-switch-mail
```

- X Window System 環境ではない場合

```
# /usr/sbin/system-switch-mail-nox
```

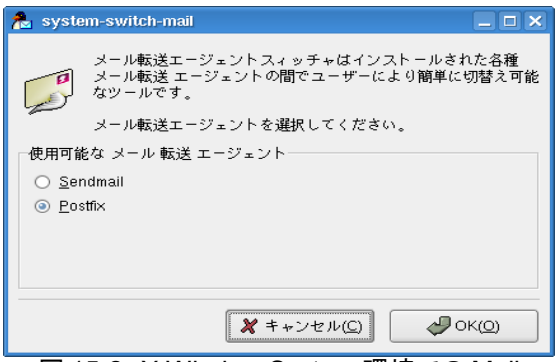


図 15-2　X Window System 環境での Mail Transfer Agent Switcher

図 15-1　非 X Window System 環境での Mail Transfer Agent Switcher

15.3 Sendmail の概要

Sendmail は、世界中で最も広く使用されている MTA プログラムの 1 つです。Sendmail は SMTP（Simple Mail Transfer Protocol）を使用して、他のメールサーバーとの電子メールのやり取りや、配送されてきたメールをユーザーごとのメールスプールへ保存する、といった一連の処理を一括して行います。

Sendmail は非常に高機能なため、任意の運用形態に合わせた設定が可能です。しかしながらその性格上、設定を誤るとシステムに変調をきたしたり、他のサイトへ迷惑をかけてしまったりするので、設定には十分に注意を払う必要があります。さらに昨今では、誤った設定によって、SPAM メールの踏み台として不正中継リレーに利用されてしまう危険性もあり、セキュリティの観点からも注意しておくべきでしょう。

15.4 Sendmail の起動と停止

Sendmail の起動スクリプトは、**/etc/rc.d/init.d/sendmail** です。起動スクリプトのオプションでは、起動（start）、停止（stop）、再起動（restart）、または現在の状況を確認（status）を指定できます。

Sendmail の設定を変更した場合、その変更内容は Sendmail を再起動するまで有効とはなりません。設定を変更した場合、Sendmail を再起動してください。

- Sendmail を起動するには、次のコマンドを実行します。

```
# /sbin/service sendmail start
```

- Sendmail を停止するには、次のコマンドを実行します。

```
# /sbin/service sendmail stop
```

- Sendmail を再起動するには、次のコマンドを実行します。

```
# /sbin/service sendmail restart
```

- Sendmail が稼働中かどうかは次のようにして調べられます。

```
# /sbin/service sendmail status
```

次のように稼働中かどうかを **lsof** を利用して検証することも可能です。

```
# /usr/sbin/lsof -i:smtp
COMMAND   PID USER   FD   TYPE DEVICE SIZE NODE NAME
sendmail 3737 root    3u  IPv4  13071       TCP localhost.localdomain:smtp
(LISTEN)
```

Sendmail が稼動していれば、COMMAND 列に「sendmail」と表示されます。

15.5 Sendmail の設定

Sendmail の設定は **/etc/mail/sendmail.cf** 内にて行われます。この設定ファイルはインストール時に自動的に配置されますが、運用形態に合わせて設定するべきです。

sendmail.cf 内の設定は複雑で非常に難解なため、手動で設定変更を行うにはかなりのスキルが必要となります。Asianux Server 3 には、**sendmail.cf** の設定変更を行うためのツールである **sendmail-cf** パッケージもインストールされているので、そちらを利用した設定について説明します。

15.5.1 準備

まず、**sendmail-cf** 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。次のコマンドを実行することにより確認できます。

```
# /bin/rpm -q sendmail-cf
sendmail-cf-8.13.8-2.2AX
```

設定変更は root アカウントにて行ってください。

不測の事態に備えて、現在の **sendmail.cf** のバックアップを取得しておくことを推奨します。

```
# /bin/cp /etc/mail/sendmail.cf /etc/mail/sendmail.cf.`date +"%Y%m%d%H%M"`
```

15.5.2 基本的な設定

Asianux Server 3 では、**sendmail.cf** の変更を **m4** と呼ばれるプログラムを用いて行います。

```
# cd /usr/share/sendmail-cf/cf
```

cf のディレクトリには、次のようなファイルが配置されています。

```
# /bin/ls
Build            generic-bsd4.4.cf       generic-osf1.cf      mailspool.cs.mc
Makefile         generic-bsd4.4.mc       generic-osf1.mc      python.cs.mc
README           generic-hpux10.cf       generic-solaris.cf   s2k-osf1.mc
chez.cs.mc       generic-hpux10.mc       generic-solaris.mc   s2k-ultrix4.mc
clientproto.mc   generic-hpux9.cf        generic-sunos4.1.cf  submit.cf
cs-hpux10.mc     generic-hpux9.mc        generic-sunos4.1.mc  submit.mc
cs-hpux9.mc      generic-linux.cf        generic-ultrix4.cf   tcpproto.mc
cs-osf1.mc       generic-linux.mc        generic-ultrix4.mc   ucbarpa.mc
cs-solaris2.mc   generic-mpeix.cf        huginn.cs.mc         ucbvax.mc
cs-sunos4.1.mc   generic-mpeix.mc        knecht.mc            uucpproto.mc
cs-ultrix4.mc    generic-nextstep3.3.cf  mail.cs.mc           vangogh.cs.mc
cyrusproto.mc    generic-nextstep3.3.mc  mail.eecs.mc
```

Asianux Server 3 にインストールされている**/etc/mail/sendmail.cf** は、**/etc/mail/sendmail.mc** を元に生成されています。したがって、設定の追加や変更はこの **sendmail.mc** ファイルを元に行うことを推奨します。

ただし、次のように **sendmail.mc** ファイルに別名をつけてコピーし、設定の追加や変更はコピーした **mc** ファイルに対して行うようにしてください。ここでは、**ホスト名.mc** というファイルにコピーしています。

```
# /bin/cp /etc/mail/sendmail.mc `hostname`.mc
```

ファイルのコピーが終了したら、コピーしたファイルをエディタで開きます。

15.5.3 m4 による mc ファイルの設定

m4 は定義されたマクロを展開し、設定値を出力するマクロプロセッサです。

初期状態のマクロと設定値について説明していきます。

- **OSTYPE**

  Sendmail が動作するプラットホームの情報を設定します。Asianux Server 3 では「linux」という値を設定します。この設定は必須です。

```
OSTYPE(`linux')dnl
```

- **DAEMON_OPTIONS**

  Sendmail が起動するときの内部的なパラメータを指定します。

```
DAEMON_OPTIONS(`Port=smtp,Addr=127.0.0.1, Name=MTA')dnl
```

  上記の設定は、sendmail デーモンに対して 127.0.0.1（localhost）からのアクセスだけを許可するもので、一般的なメールサーバーではこの設定は必要ありません。Asianux Server 3 にインストールされている **sendmail.cf** は初期状態でこのマクロが設定されているので、他の一般的なメールサーバーを使用する場合にはこのマクロをコメントアウト（行頭に「dnl 」と入力）して、**sendmail.cf** を生成し直してください。

- **define**

  初期状態では次の値が設定されています。

```
ALIAS_FILE
```

  alias データベースの場所を指定するものです。**/etc/aliases** に設定されています。

```
STATUS_FILE
```

  status ファイルの場所を指定するものです。**/var/log/mail/statistics** に設定されています。

```
PROCMAIL_MAILER_PATH
```

  procmail への絶対パスを設定します。**/usr/bin/procmail** に設定されています。

- **FEATURE**

  Sendmail の機能に関する設定を行います。always_add_domain が設定されていて、ドメイン名省略によるメール発信がなされた場合に自動的にドメイン名が付加されます。

- **EXPOSED_USER――**これによって指定されたユーザーに対するメールは、リレーせずにローカルホスト内のスプールに格納されます。root ユーザーのみが設定されています。
- **MAILER――**メーラーの設定を行います。Asianux Server 3 ではメーラーとして procmail を使用するため、その設定がされています。この設定は mc ファイルの最後で行われる必要があります。

15.5.4 cf ファイルの生成

mc ファイルの変更が完了したら、次のコマンドを実行し **cf** ファイルを生成します。

```
# /usr/share/sendmail-cf/cf/Build `hostname`.cf
```

cf ファイルが生成されるので、これを**/etc/mail/sendmail.cf** にコピーしてから、Sendmail を再起動してください。以上で Sendmail の設定は完了です。

```
# /bin/cp -p `hostname`.cf /etc/mail/sendmail.cf
# /sbin/service sendmail restart
```

第16章　メールサーバーの構築（Postfix）

この章で説明する内容

目的	Postfix を使ったメールサーバーの構築方法について理解する
機能	メールの配送
必要な RPM	postfix cyrus-sasl cyrus-sasl-libs cyrus-sasl-md5 cyrus-sasl-plain
設定ファイル	/etc/postfix/main.cf
章の流れ	1 MTA (Mail Transfer Agent) の概要 2 Mail Transfer Agent Switcher の利用方法 3 Postfix の概要 4 Postfix の起動と停止 5 Postfix の設定
関連 URL	**Postfix のページ** http://www.postfix-jp.info/

16.1　MTA (Mail Transfer Agent) の概要

MTA（Mail Transfer Agent）とは、インターネット上で、ホスト間や企業間などのメールの配送を受け持つアプリケーションです。この MTA に対して、実際にクライアント上でメールの読み書きを行うアプリケーションは MUA（Mail Usesr Agent）と呼ばれます。インターネットのメールシステムではこのクライアント（MUA）とサーバー（MTA）が連携して電子メールシステムを構成しています。

16.2　Mail Transfer Agent Switcher の利用方法

Postfix と **Sendmail** など、同じ役割を持つ MTA（Mail Transfer Agent）を 2 種類以上インストールした場合、Mail Transfer Agent Switcher を利用して、Postfix と Sendmail のうち、どちらか一方を MTA として利用するように設定を行う必要があります。Mail Transfer Agent Switcher を起動するには次のコマンドを実行します。

※Asianux Server 3 の初期状態では Postfix が MTA となっており、Postfix を利用する場合は変更の必要はありません。

- X Window System 環境の場合

```
# /usr/sbin/system-switch-mail
```

- X Window System 環境ではない場合

```
# /usr/sbin/system-switch-mail-nox
```

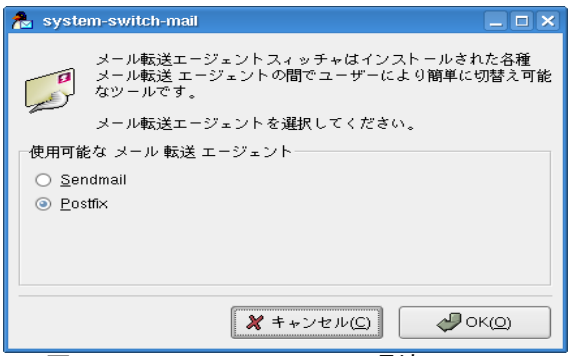


図 16-2　X Window System 環境での Mail Transfer Agent Switcher

図 16-1　非 X Window System 環境での Mail Transfer Agent Switcher

16.3 Postfix の概要

Postfix は、Sendmail、qmail に並んで利用されている代表的な MTA（Mail Transfer Agent）の 1 つです。

Sendmail より安全かつ容易に設定を行えるため、利用者が増えています。また、Sendmail との互換性を考慮して開発されているため、Sendmail を利用していたシステムから置き換えやすいのも特徴の 1 つです。

16.4 Postfix の起動と停止

Postfix の起動スクリプトは、**/etc/rc.d/init.d/postfix** です。起動スクリプトのオプションでは、起動（start）、停止（stop）、再起動（restart）、設定変更の反映（reload）、現在の状況を確認（status）を指定できます。Postfix の設定を変更した場合、その変更内容は reload を実行するか、再起動するまで有効とはなりません。設定を変更した場合、Postfix を再起動してください。

- Postfix を起動するには、次のコマンドを実行します。

```
# /sbin/service postfix start
```

- Postfix を停止するには、次のコマンドを実行します。

```
# /sbin/service postfix stop
```

- Postfix を再起動するには、次のコマンドを実行します。

```
# /sbin/service postfix restart
```

- Postfix が稼働中かどうかは次のようにして調べられます。

```
# /sbin/service postfix status
```

次のように稼働中かどうかを **lsof** を利用して検証することも可能です。

```
# /usr/sbin/lsof -i:smtp
COMMAND  PID USER   FD   TYPE DEVICE SIZE NODE NAME
master  4790 root   11u  IPv4  16325       TCP localhost.localdomain:smtp
(LISTEN)
```

Postfix が稼動している場合は、COMMAND 列に「master」と表示されます。

Postfix の設定内容を確認するは次のコマンドを実行します。

```
# /usr/sbin/postconf -n
```

16.5 Postfix の設定

Postfix の設定ファイルは**/etc/postfix/**以下にあります。
通常の設定では、**main.cf** ファイルを修正します。

16.5.1 インターネットのドメインメールサーバーとしての設定方法

インターネット上に Postfix を動作させてメールサーバーとして設置する場合、最低限 **main.cf** ファイル の次の項目を修正する必要があります。

- **myhostname**
  Postfix が動作しているホストの FQDN を指定します。
  ※設定しているサーバーのネットワーク設定と一致している必要があります。

```
myhostname = mail.example.com
```

- **mydomain**
  ドメインメールサーバーとして運用する場合はメールアドレスのドメイン部分を指定します。

```
mydomain = example.com
```

- **myorigin**
  発信元アドレスの「@」以降に Postfix が動作しているホスト名ではなく、ドメイン名を指定する場合は次のように指定します。

```
myorigin = $mydomain
```

- **inet_interfaces**
  smtp でメールを受け付けるアドレスを指定します。

```
inet_interfaces = localhost, 192.168.1.1
```

- **mydestination**
  ドメインメールサーバーとして運用する場合は**$mydomain** を追加します。

```
mydestination = $myhostname, localhost.$mydomain, $mydomain
```

- **mynetworks**
  メールの中継を許可するネットワークアドレスを指定します。

```
mynetworks = 192.168.1.0/24, 127.0.0.0/8
```

以上で、Postfix の最低限の設定は終了です。

16.5.2 Postfix での SMTPAUTH の利用

インターネットメールサーバーを公開するにあたり、SPAM メールなどの不正中継対策は必須です。

SMTPAUTH は、認証されたホストからのメールのみリレーを許可することで、送信元アドレスを偽った不正中継の踏み台になることを防ぎます。

➢ Postfix の設定方法

(1) **/etc/postfix/main.cf** に次の行を追加します。

```
smtpd_sasl_auth_enable = yes
smtpd_sasl_security_options = noanonymous
smtpd_sasl_local_domain = $myhostname
broken_sasl_auth_clients = yes
smtpd_recipient_restrictions = permit_sasl_authenticated, check_relay_domains
```

(2) ユーザーとパスワードを登録したデーターベースファイル **/etc/sasldb2** を作成します。

```
# /usr/sbin/saslpasswd2 -c -u `postconf -h myhostname` exampleuser
Password:                  ← examleuserのパスワードを入力します。
Again (for verification):  ← examleuserのパスワードをもう一度入力します。
```

(3) 作成した**/etc/sasldb2 ファイル**のパーミッションを変更します。

```
# /bin/chgrp mail /etc/sasldb2
# /bin/chmod g+r /etc/sasldb2
```

以上で Postfix の設定は終わりです。Postfix を再起動し、**saslauthd** サービスを起動してください。

```
# service postfix restart
# service saslauthd start
```

設定の確認は、Telnet で Postfix へ接続して行います。

```
# /usr/bin/telnet localhost 25                              ← ポート25へTelnet接続します。
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 mail.example.com ESMTP Postfix
EHLO localhost                                              ←「EHLO localhost」と入力します
。
<<中略>>
250-AUTH DIGEST-MD5 GSSAPI CRAM-MD5 LOGIN PLAIN NTLM  ← この行が表示されることを確認しま
す。
```

➢ クライアントの設定方法

MUA（クライアントのメールソフト）にて送信時に認証を行うように設定しますが、設定内容はソフトに依存するので、ここでは解説しません。MUA（クライアントのメールソフト）のマニュアルを参照してください。

第17章　POP / IMAP サーバー

この章で説明する内容

目的	POP3/IMAP サーバーの構築方法について理解する
機能	メールクライアントに対して POP3/IMAP プロトコルでメール受信機能を提供
必要な RPM	cyrus-imapd cyrus-imapd-utils cyrus-imapd-perl cyrus-sasl cyrus-sasl-libs cyrus-sasl-md5 cyrus-sasl-plain dovecot
設定ファイル	/etc/cyrus.conf /etc/imapd.conf /etc/dovecot.conf
章の流れ	1 Cyrus IMAP の概要 2 Cyrus IMAP の起動と停止 3 Cyrus IMAP の設定 4 Dovecot の概要 5 Dovecot の起動と停止 6 Dovecot の設定
関連 URL	**Project Cyrus** http://cyrusimap.web.cmu.edu/ **Dovecot: Secure IMAP Server** http://www.dovecot.org/

17.1 Cyrus IMAP の概要

Cyrus IMAP は、Project Cyrus(http://cyrusimap.web.cmu.edu/)によって開発されている IMAP サーバーです。セキュリティの安全性が高く、パフォーマンスも優れています。

17.2 Cyrus IMAP の起動と停止

Cyrus IMAP の起動スクリプトは、**/etc/rc.d/init.d/cyrus-imapd** です。起動スクリプトのオプションでは、起動(start)、停止(stop)、再起動(restart)、または現在の状況を確認(status)を指定できます。

Cyrus IMAP の設定を変更した場合、その変更内容は Cyrus IMAP を再起動するまで有効とはなりません。設定を変更した場合、Cyrus IMAP を再起動させる必要があります。

- Cyrus IMAP を起動するには、次のコマンドを実行します。

```
# /sbin/service cyrus-imapd start
```

- Cyrus IMAP を停止するには、次のコマンドを実行します。

```
# /sbin/service cyrus-imapd stop
```

- Cyrus IMAP を再起動するには、次のコマンドを実行します。

```
# /sbin/service cyrus-imapd restart
```

- Cyrus IMAP が稼働中かどうかは次のようにして調べられます。

```
# /sbin/service cyrus-imapd status
```

次のように稼働中かどうかを **lsof** を利用して検証することも可能です。

```
# /usr/sbin/lsof -i:imap
```

- システムが起動したときに自動的に Cyrus IMAP を起動するようにするには、次のコマンドを実行します。

```
# /sbin/chkconfig cyrus-imapd on
```

- システムが起動したときに Cyrus IMAP が起動しないようにするには、次のコマンドを実行します。

```
# /sbin/chkconfig cyrus-imapd off
```

17.3 Cyrus IMAP の設定

Cyrus IMAP に関する設定は、**/etc/imapd.conf** の各種パラメータを設定することにより行われます。設定ファイルのパラメータの書式は、次のようなものです。

```
オプション名： パラメータ
```

ここでは、基本的な設定項目のみ解説します。詳しくは、「**man imapd.conf**」を参照してください。

- **configdirectory**
  設定ファイルを置くディレクトリを指定します。

```
configdirectory: /var/lib/imap
```

- **partition-default**
  新規メールボックスを作成するデフォルトのパーティション（保存場所）を指定します。

```
partition-default: /var/spool/imap
```

- **admins**
  管理者権限を持つユーザーを指定します。複数指定する場合は、スペースで区切ります。
  admins に設定したユーザーは、IMAP サーバーにログインしてメールを読んではいけません。ログインしてしまうと、メールボックス環境を破壊してしまう可能性があります。

```
admins: cyrus
```

- **sasl_pwcheck_method**
  パスワードの認証方法を設定します。デフォルトでは、ローカルユーザーとそのパスワードを使用する **saslauthd** になっています。認証の設定については後述の認証設定で説明します。

```
sasl_pwcheck_method: saslauthd
```

設定が終わったら、Cyrus IMAP を起動します。

```
# /sbin/service cyrus-imapd start
```

17.3.1 MTA の設定

IMAP を設定する場合は、MTA にも設定の変更を行う必要があります。ここでは、**Postfix** を使用した場合の設定方法を説明します。なお、「第 16 章 メールサーバーの構築 (Postfix)」の設定が一通り済んでいるものとします。

/etc/postfix/main.cf を開き、次の 2 行を追記します。

```
mailbox_transport = cyrus
fallback_transport = cyrus
```

設定ファイルを保存したら、Postfix のサービスを再起動します。

```
# /sbin/service postfix restart
```

17.3.2 認証設定

Cyrus IMAP はさまざまな認証方法が利用可能です。ここでは、デフォルト設定の **saslauthd** を使用した認証方法と、**sasldb** を使用した方法を説明します。

➢ saslauthd を使用した認証

インストール直後の Cyrus IMAP は、ローカルユーザーのユーザー名とパスワードで認証を行うようになっています。この方法で運用を行う場合、saslauthd のサービスを起動させる必要があります。

```
# /sbin/service saslauthd start
```

また正式に運用を行う場合は、サーバー起動時にサービス起動するようにしておくと良いでしょう。

```
# /sbin/chkconfig saslauthd on
```

ユーザーを追加するには、**useradd** コマンドを使用して行います。同時に、パスワードも設定してください。ここでは例として、ユーザー tanaka の作成を行います。

```
# /usr/sbin/useradd tanaka
# /usr/bin/passwd tanaka
```

ユーザーが作成できたら、メール管理プログラムの **cyradm** コマンドで管理コンソールにログインし、ユーザー tanaka のメールボックスを作成します。ログインユーザー名は **/etc/imapd.conf** ファイルで設定した管理ユーザーで行います。管理ユーザーのパスワードについては、あらかじめ **passwd** コマンドで設定しておく必要があります。

```
# /usr/bin/cyradm --user cyrus localhost
```

ログインに成功すると、**cyradm** のプロンプトが表示されます。ユーザーのメールボックスを作成するには、次のコマンドを入力します。

```
asianux.example.com> createmailbox user.tanaka
```

作成できたら、**exit** コマンドを入力して、**cyradm** を終了します。
IMAP アカウントに対応したメーラーで、正しく接続ができるか確認してください。

➢ sasldb による認証設定

sasldb による認証方法では、ローカルユーザーとは別にユーザー管理を行うことができます。sasldb を使用するには、**/etc/imapd.conf** ファイルを開き、sasl_pwcheck_method オプションを次の通り書き換えて保存します。

```
sasl_pwcheck_method: auxprop
```

設定が完了したら、**/etc/imapd.conf** ファイルで指定した管理ユーザーのパスワードを、**saslpasswd2** コマンドを使用して行います。パスワードは確認のため 2 度聞かれます。

```
# /usr/sbin/saslpasswd2 cyrus
Password:
Again (for Verification):
```

ユーザーの作成についても、同じく **saslpasswd2** コマンドを使用して行います。ユーザーを作成するには、引数 **-c** を指定します。ここでは例としてユーザー satoshi を作成します。

```
# /usr/bin/saslpasswd2 -c satoshi
Password:
Again (for Verification):
```

cyrus ユーザーが /etc/sasldb2 を参照できるようにパーミッションの変更を行います。

```
# /bin/chgrp mail /etc/sasldb2
# /bin/chmod g+r /etc/sasldb2
```

ユーザーの作成、/etc/sasldb2 のパーミッション変更ができたら、メール管理プログラムの **cyradm** コマンドで管理コンソールにログインし、ユーザー satoshi のメールボックスを作成します。ログインは管理ユーザーで行い、次のようなコマンドを実行します。

```
# /usr/bin/cyradm --user cyrus localhost
```

ログインに成功すると、**cyradm** のプロンプトが表示されます。ユーザーのメールボックスを作成するには、次のコマンドを入力します。

```
asianux.example.com> createmailbox user.satoshi
```

作成できたら、**exit** コマンドを入力して、**cyradm** を終了します。

IMAP アカウントに対応したメーラーで、正しく接続ができるか確認してください。

17.3.3 ログ取得設定

ログを取得するには、**/etc/syslog.conf** をエディタで開き、次の1行を追加します。

```
local6.debug  /var/log/imapd.log
```

編集が完了したら、**/var/log/imapd.log** の空ファイルを作成し、syslog サービスを再起動します。

```
# /bin/touch /var/log/imapd.conf
# /sbin/service syslog restart
```

17.4 Dovecot の概要

Dovecot とは、POP3/IMAP サーバーの1つで、セキュリティ面で丈夫で、拡張性に富み、軽快に動作するのが特徴です。また、設定も簡単に行うことができます。

17.5 Dovecot の起動と停止

Dovecot の起動スクリプトは、**/etc/rc.d/init.d/dovecot** です。起動スクリプトのオプションでは、起動（start）、停止（stop）、再起動（restart）、または現在の状況を確認（status）を指定できます。

Dovecot の設定を変更した場合、その変更内容は Dovecot を再起動するまで有効とはなりません。

- Dovecot を起動するには、次のコマンドを実行します。

```
# /sbin/service dovecot start
```

- Dovecot を停止するには、次のコマンドを実行します。

```
# /sbin/service dovecot stop
```

- Dovecot を再起動するには、次のコマンドを実行します。

```
# /sbin/service dovecot restart
```

- Dovecot が稼働中かどうかは次のようにして調べられます。

```
# /sbin/service dovecot status
```

- システムが起動したときに自動的に Dovecot を起動するようにするには、次のコマンドを実行します。

```
# /sbin/chkconfig dovecot on
```

- システムが起動したときに Dovecot が起動しないようにするには、次のコマンドを実行します。

```
# /sbin/chkconfig dovecot off
```

17.6 Dovecot の設定

Dovecot の設定ファイルは、**/etc/dovecot.conf** です。設定の書式は次の通りです。

```
オプション名 = パラメータ
```

ここでは、最小限必要な設定について説明します。そのほかのオプション設定方法については、dovecot.conf 内のコメントを参照してください。

- **protocols**
  Dovecot で提供するプロトコルを記述します。

```
protocols = imap imaps pop3 pop3s
```

- **mail_location**
  メールボックスを置くディレクトリを設定します。次の例の場合、各ユーザーのホームディレクトリ下の Maildir ディレクトリに置いています。

```
mail_location = maildir:~/Maildir
```

17.6.1 MTA の設定

Dovecot で maildir 形式のメールボックスを使用するためには、MTA にも設定の変更を行う必要があります。ここでは、Postfix を使用した場合の設定方法を説明します。なお、246 ページからの Postfix の設定が一通りすんでいるものとします。

/etc/postfix/main.cf を開き、次の行を追記します。

```
home_mailbox = Maildir/
```

設定ファイルを保存したら、Postfix のサービスを再起動します。

```
# /sbin/service postfix restart
```

17.6.2 ログ取得設定

ログファイル設定は、デフォルトでは syslog が出力しており、**/var/log/maillog** に記録されています。

また、dovecot.conf の「log_path」オプションおよび「info_log_path」を設定することによって、syslog に頼らないログ出力も可能です。

```
log_path = /var/log/dovecot.log
info_log_path = /var/log/dovecot_mail.log
```

第18章　メーリングリスト管理

この章で説明する内容

目的	メーリングリストの構築・管理方法について理解する
機能	メーリングリストの作成・管理
必要な RPM	postfix mailman httpd
設定ファイル	/etc/mailman/mm_cfg.py /etc/postfix/main.cf /etc/aliases
章の流れ	メーリングリストの概要 メールサーバのエイリアス機能による管理 Mailman によるメーリングリスト管理
関連 URL	**Mailman** http://www.gnu.org/software/mailman/

18.1　メーリングリストの概要

メーリングリストは、複数の受信者に同時にメールを配信する仕組みです。本章では、メールサーバのエイリアス機能を使用して簡単なメーリングリストを作成する方法と、高度な管理のためにメーリングリストマネージャ(Mailman)を使用する例を説明します。

18.2　メールサーバのエイリアス機能による管理

メールサーバ（Postfix/Sendmail）には、メールを別のユーザやプログラムに転送する機能（エイリアス機能）があります。この機能を利用して簡単なメーリングリストを作成することができます。（使用中のメールサーバから、メールが送信できる環境を前提とします。）

➢ /etc/aliases

デフォルトの設定ファイルは**/etc/aliases** です。/etc/aliases には、以下のような書式でエントリを記述します。

```
メーリングリスト名： メールアドレス,メールアドレス,...
```

以下の例では、examplelist 宛のメールを user1@example.com、user2@example.com、user3@example.com に転送します。

```
examplelist : user1@example.com, user2@example.com, user3@example.com
```

/etc/aliases ファイルを修正した場合には、次のコマンドを実行し、エイリアスを更新します。

```
# newaliases
```

18.3 Mailman を使用したメーリングリスト管理

18.3.1 Mailman の概要

Mailman は、コマンドや Web ページからメーリングリストの管理や設定が行えるメーリングリストマネージャです。Mailman は各メーリングリストにウェブページを用意し、ユーザは、ウェブ経由でリストへの参加・脱退を行なうことができます。

Mailman によるメーリングリスト管理は、Postfix 等のメールサーバプログラムが、Mailman のエイリアスファイルを読み込むことによって実現します。また、Mailman は Postfix との連携に最適化されています。本節では Mailman と Postfix との連携について説明します。なお、Mailman を使用する為には crond、Apache HTTP Server、Postfix が使用出来る環境が必要です。

18.3.2 Mailman の起動と停止

Mailman の起動スクリプトは、**/etc/rc.d/init.d/mailman** です。起動スクリプトのオプションでは、起動（start）、停止（stop）、再起動（restart）、または現在の状況を確認（status）を指定できます。

- Mailman を起動するには、次のコマンドを実行します。

```
# service mailman start
```

- Mailman を停止するには、次のコマンドを実行します。

```
# service mailman stop
```

- Mailman を再起動するには、次のコマンドを実行します。

```
# service mailman restart
```

- Mailman が稼働中かどうかは次のようにして調べることができます。

```
# service mailman status
```

18.3.3 Mailman の設定

➢ Mailman と Apache HTTP Server の連携

Mailman の Web ページを利用する為には、Apache HTTP Server に Mailman 設定が必要となります。デフォルトでは、**/etc/httpd/conf.d/mailman.conf** ファイルに以下のように必要な設定がされており、設定を変更する必要はありません。

```
ScriptAlias /mailman/ /usr/lib/mailman/cgi-bin/
<Directory /usr/lib/mailman/cgi-bin/>
AllowOverride None
    Options ExecCGI
    Order allow,deny
    Allow from all
</Directory>

Alias /pipermail/ /var/lib/mailman/archives/public/
<Directory /var/lib/mailman/archives/public>
    Options Indexes MultiViews FollowSymLinks
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>
```

/etc/httpd/conf.d/mailman.conf を変更した場合、設定を反映する為にはサービスの再起動が必要です。

```
# service httpd restart
```

➢ Mailman と Postfix の連携

Postfix との連携の為、設定ファイル **/etc/postfix/main.cf** に以下の設定を行います。

（1）Postfix の alias_maps,alias_database に Mailman のエイリアスファイルを追記します。

```
alias_maps = hash:/etc/aliases, hash:/etc/mailman/aliases
alias_database = hash:/etc/aliases, hash:/etc/mailman/aliases
```

（2）Postfix の -owner と -request のアドレスを特別扱いする機能を無効にします。

```
owner_request_special = no
```

（3）Mailman の VERP 配送のサポートの為、次のように設定します。

```
recipient_delimiter = +
```

（4）配送先ユーザーが不明の場合のエラーコード 550(エラーの際送しない)となるように設定します。

```
unknown_local_recipient_reject_code = 550
```

（5）設定を反映するために、サービスを再起動します。

```
# service postfix reload
```

➢ crond の起動

Mailman はメーリングリストの動作に必要な定期的な処理を行う為、crond を使用しています。
次のコマンドで、crond が動作していることを確認してください。

```
# service crond status
```

crond が起動していない場合は、次のコマンドで起動してください。

```
# service crond start
```

➢ Mailman の初期設定

Mailman の設定ファイル/**etc/mailman/mm_cfg.py** を編集し、以下の操作を行います。
/etc/mailman/mm_cfg.py は編集前にバックアップを採取することをお勧めします。

（1）設定ファイルの末尾に、メールサーバに Postfix を使用し、日本語を使用するための設定を追記します。

```
MTA = 'Postfix'
DEFAULT_SERVER_LANGUAGE = 'ja'
```

（2）次のコマンドを実行し、Mailman のエイリアスファイルを初期化します

```
# /usr/lib/mailman/bin/genaliases
```

（3）次のコマンドを実行し、作成したエイリアスファイルのパーミッションとオーナーを確認・修正します。
実行後、”問題ありません”と表示されるまで繰り返し実行します。

```
# /usr/lib/mailman/bin/check_perms -f
```

（4）管理者用パスワードを設定します。サイトパスワードは Mailman の設定を行う管理者用のパスワードです。

```
# /usr/lib/mailman/bin/mmsitepass
```

（5）管理者用メーリングリストを作成します。
管理者用メーリングリストは、Mailman を使用する際に必要なメーリングリストです。管理者用メーリングリストの名前はデフォルトで、mailman となっています。メーリングリストの新規作成は newlist コマンドを使用します。作成の為、以下のコマンドを実行します。

```
# /usr/lib/mailman/bin/newlist mailman
Enter the email of the person running the list: root@example.com
...
# /usr/lib/mailman/bin/config_list -i /etc/mailman/sitelist.cfg mailman
```

管理者用メーリングリストを作成後、登録したリスト管理者のメールアドレスにメールが届いていることを確認します。

（6）Mailman を起動します。
関連のサービス postfix, httpd, crond が動作していることが必要です。

```
# service mailman start
```

➢ 動作確認・メーリングリストの作成

ブラウザから Mailman の管理用 Web インターフェイスを開きます。接続の際、URL は必ずサーバの FQDN を指定します。FQDN が mailman.example.com の場合の例は以下のとおりです。

```
http://mailman.example.com/mailman/admin
```

図 18-1 のページが表示されたら、”新しいメーリングリストを作成する”のリンクを開きメーリングリストを作成します。
メーリングリストが正常に作成できれば成功です。

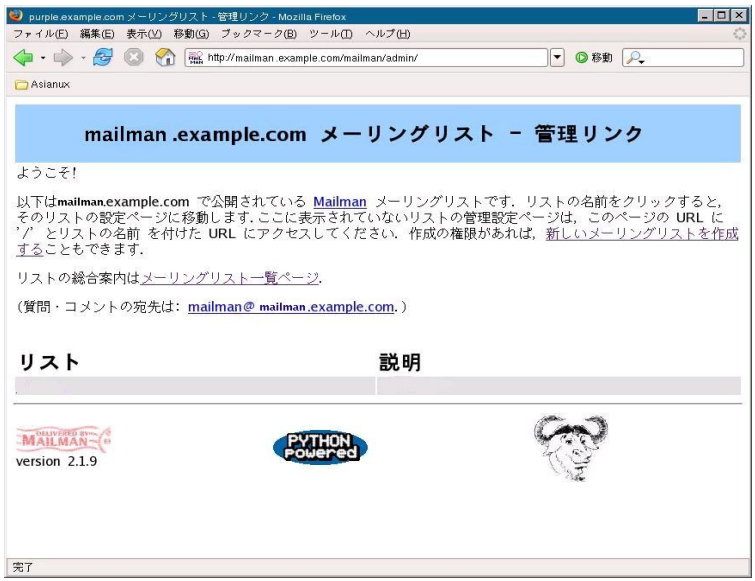


図 18-1 Mailman 管理用 Web インターフェース

第19章　プロキシサーバーの構築

この章で説明する内容

目的	プロキシサーバーの構築方法について理解する
機能	プロキシサーバー
必要な RPM	squid ―― プロキシサーバー本体
設定ファイル	/etc/squid/squid.conf
章の流れ	1 Squid の概要 2 Squid の起動と停止 3 Squid の設定 4 Squid の利用 5 Squid の運用
関連 URL	**Squid Web Proxy Cache** http://www.squid-cache.org/

19.1 Squid の概要

Squid とは、http や ftp などのインターネット上にあるリソースをローカルネットワーク内のストレージ上にキャッシュして、外部とのネットワークトラフィック（情報転送量）を軽減させるためのデーモンプログラムです。

Squid はウェブサーバーとクライアントプログラムとの間に介在して、転送されたデータを Squid が管理するデータ構造内に保持します。Squid を介するウェブブラウザから要求されたウェブオブジェクトと、Squid がキャッシュしているデータが一致した場合には、Squid は外部ネットワークからのデータ転送を行わず、キャッシュ情報をクライアントプログラム側へ転送します。これにより、自ネットワークと外部ネットワーク間のトラフィックを軽減させることができます。

19.2 Squid の起動と停止

Squid の起動スクリプトは、**/etc/rc.d/init.d/squid** です。起動スクリプトのオプションでは、起動（start）、停止（stop）、再起動（restart）、現在の状況を確認（status）を指定できます。

Squid の設定を変更した場合は、変更を反映するために Squid を再起動する必要があります。

- Squid を起動するには、次のコマンドを実行します。

```
# /sbin/service squid start
```

- Squid を停止するには、次のコマンドを実行します。

```
# /sbin/service squid stop
```

- Squid を再起動するには、次のコマンドを実行します。

```
# /sbin/service squid restart
```

- Squid の現在の状況を確認するには、次のコマンドを実行します。

```
# /sbin/service squid status
```

- また、**ps** コマンドを実行すると、Squid プロセスの状況を次のように確認できます。

```
# /bin/ps aux | /bin/grep squid | /bin/grep -v grep
```

```
root      2394  0.0  0.1  7748  1260 ?        Ss   11:24   0:00 squid -D
squid     2396  0.0  0.5  9972  5776 ?        S    11:24   0:00 (squid) -D
squid     2398  0.0  0.0  1492   216 ?        Ss   11:24   0:00 (unlinkd)
```

上記のような表示がない場合は、Squid が動作していない、あるいは Squid のインストールが行われていない可能性があります。Squid の起動か再起動、またはインストールを行ってください。

- システムが起動したときに自動的に Squid を起動させるためには、次のコマンドを実行します。ランレベル 3、4、5 について自動的に起動するように設定されます。

```
# /sbin/chkconfig squid on
```

- システムが起動したときに Squid が自動的には起動しないようにするためには、次のコマンドを実行します。

```
# /sbin/chkconfig squid off
```

19.3 Squid の設定

Squid デーモンに関する設定は、**/etc/squid/squid.conf** で各種パラメータを設定することにより行われます。設定ファイル内の各パラメータの書式は、基本的に次のようなものです。

```
タグ[ オプション...]
```

各タグとそれに指定する値については、**squid.conf** 内により詳しい説明がありますので、それを参考にしてください。また、Squid にはキャッシュサーバー同士の連携を行うなどの高度な機能も実装されていますが、それらの設定方法についても **squid.conf** 内の説明を参照してください。

ここでは、Squid をローカルネットワーク内での単純なキャッシュサーバーとして動作させる場合の設定方法を説明します。

19.3.1　アクセス制御（acl）

Squid へのアクセス許可の制限を制御します（初期設定では localhost からのアクセスのみ許可されます）。

アクセス制御は、「acl」と「http_access」のタグを使用することにより可能となります。

- **acl** タグはアクセスリストを定義するためのものです。書式は次のようになります。

```
acl acl名 acl種別 文字列1 ...
acl acl名 acl種別 "ファイル" ...
```

- **http_access** タグは、acl タグにより定義されたアクセスリストからの要求のアクセスを制御します。書式は次のとおりです。acl 名の前の「!」は否定を意味します。

```
http_access allow|deny [!]acl名
```

次に、ローカルネットワークに存在するホストからのアクセスをすべて許可する場合の設定について説明します。ローカルネットワーク内のホストの IP アドレスは 10.xx.xx.xx であるものとします。

まず、ローカルネットワークの acl 名を定義します。

```
acl localnetwork src 10.0.0.0/255.0.0.0
```

上記の設定の「localnetwork」は発信元 IP アドレスを 255.0.0.0 でマスクした際に、ネットワークアドレスが 10.0.0.0 になるホストであるという設定です。次に定義した「localnetwork」に対して許可を設定します。

```
http_access allow localnetwork
```

以上により、ローカルネットワークに存在するホストからのアクセスがすべて許可されます。

http_access タグの解釈は上から順に行われることに注意してください。たとえば、次のように記述されているとすると、すべての要求は http_access deny all に合致してしまうため、2 行目以下に記述されている Squid に対する要求はすべて却下されてしまいます。

```
http_access deny all
http_access allow manager localhost
http_access deny manager
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
http_access allow localnetwork
```

19.3.2 ポート番号（http_port）

Squid が使用するポート番号を指定します（初期設定ではポート 3128 を使用します）。

書式は次のようになります。

```
http_port ポート番号
```

19.3.3 キャッシュディレクトリとデータサイズ（cache_dir）

Squid がキャッシュを保持するためのディレクトリを指定します。

初期設定では**/var/spool/squid** 配下に 100MB のキャッシュサイズを保持する設定になっています。したがって、初期状態で Squid を利用する場合は**/var** あるいは**/var/spool/squid** パーティションに 100MB 以上の領域を用意する必要があります。

書式は次のようになります。

```
cache_dir Type ディレクトリ名 サイズ 階層1 階層2
```

Type には ufs を指定します。ほかに asyncufs が指定できますが推奨されません。

サイズにはキャッシュ総データサイズを MB 単位で指定します。

ディレクトリ名はキャッシュデータを保持するためのディレクトリを指定します。Squid はここで指定したディレクトリ配下に 2 階層のサブディレクトリを使用します。これらのサブディレクトリの個数は、階層 1、階層 2 で指定します。

19.3.4 ログディレクトリ（access_log / cache_log / cache_store_log）

Squid はその動作状況をログとしてファイルに出力します。ここでは、それらのログの出力場所の設定方法を説明します（初期設定では、ログはすべて**/var/log/squid** 配下に生成されます）。

出力されるログファイルは次のとおりです。

- **access.log** —— クライアントからの要求状況
- **cache.log** —— キャッシュ状況
- **store.log** —— 保持されているキャッシュファイルの情報

これらはそれぞれ **access_log**、**cache_log**、**cache_store_log** タグで指定されます。このときの書式は次のとおりです。

```
access_log ファイル名
```

```
cache_log ファイル名
```

```
cache_log ファイル名
```

19.3.5 メモリ使用量（cache_mem）

Squid が使用するメモリ使用量を設定します。メモリ使用量を大きくした場合、Squid の性能の向上が期待できます（初期設定では 8MB のメモリを使用します）。

書式は次のようになります。メモリ使用量は MB 単位で指定します。

```
cache_mem メモリ使用量
```

19.4 Squid の利用

Squid を起動させただけではその機能を有効利用することはできません。Squid を利用するには、クライアントのウェブブラウザで Squid を利用する設定が必要となります。各ブラウザのプロキシの設定で、Squid を稼働させているホスト名と、http_port で指定したポート番号を指定してください。

19.5 Squid の運用

19.5.1 キャッシュディレクトリの変更

キャッシュディレクトリは、**squid.conf** 内の **chache_dir** タグにディレクトリ名を指定することで変更できます。何らかの理由でキャッシュディレクトリを変更した場合には、次のコマンドを実行してキャッシュディレクトリの初期化を行ってください。ここでは**/localdisk/squid/cache** にキャッシュディレクトリを変更するものとします。

```
# umask 22
# /bin/mkdir -p /localdisk/squid/cache
# /bin/chown squid:squid /localdisk/squid/cache
```

```
# /usr/sbin/squid -z
```

19.5.2 ログのローテーション

Squid が生成する **access.log**、**cache.log**、**store.log** のログファイルは、Squid の利用頻度にも依存しますが、ディスクスペースを圧迫する可能性があります。次のコマンドを定期的に実行することにより、各ログファイルをローテーションすることを推奨します。

```
# /usr/sbin/squid -k rotate
```

これにより、Squid はログファイルの切り替えを行います。古いログを何世代残しておくかは、s**quid.conf** 内の **logfile_rotate** タグに値を設定することにより変更できます。

ログ切り替えの実行は cron により自動実行するように設定しておくことを推奨します。毎日 00:00 に行う場合の crontab エントリは次のようになります。

```
0 0 * * * /usr/sbin/squid -k rotate
```

19.5.3 ダイアルアップ環境での利用

Squid は、起動時に **squid.conf** 内の **dns_testnames** に設定された URL の解決を試みます。ダイアルアップ環境下で Squid を自動実行する場合、Squid は起動時に DNS への問い合わせができません。この場合、Squid の起動オプションに**-D** を指定してください（Asianux Server 3 でインストールされる起動スクリプトには、初期状態で**-D** が指定されています）。

第20章　ウェブサーバーの構築

この章で説明する内容

目的	ウェブサーバーの構築方法について理解する
機能	Web ページの表示、動的 WWW サイト
必要な RPM	**Apache のパッケージ** httpd —— Apache 本体（ウェブサーバー） httpd-manual —— Apache のオンラインマニュアル httpd-devel —— 開発者用パッケージ **関連するパッケージ** mod_perl —— mod_perl を実行するために必要なパッケージ（Perl-CGI では不要） mod_ssl —— セキュアな通信を行うために必要なパッケージ
設定ファイル	/etc/httpd/conf/httpd.conf /etc/sysconfig/httpd
章の流れ	1 Apache サーバーの概要 2 Apache サーバーの起動と停止 3 Apache サーバーの設定
関連 URL	**Apache HTTP SERVER PROJECT**　http://httpd.apache.org/ **日本 Apache ユーザ会**　http://www.apache.jp/

20.1　Apache サーバーの概要

ウェブ システムで表示されるページは大きく分けて 2 種類あります。

1 つは静的ページで、Apache を使えば簡単にパフォーマンスに優れたシステムを構築できます。

もう 1 つは動的ページで、アクセスがあるたびに CGI や SSI を使って新たにページを作成して表示します。また動的ページには、データベースをバックエンドに置いた Web + DB システムというものもあります。

Apache とは、このようなウェブシステムを構築する際に、世界中で広く利用されているウェブサーバープログラムです。

この章では、一般的なウェブサーバーの構築から、動的なページを構築する方法やウェブシステムのパフォーマンス向上のヒントを取り上げます。

なお、Asianux Server 3 ではコンパイル済みの Perl-CGI や PHP などがインストールされるので、インストール直後からすぐにウェブサーバーを稼動させることができます。

20.2　Apache サーバーの起動と停止

Apache の起動スクリプトは、**/etc/rc.d/init.d/httpd** です。起動スクリプトのオプションでは、起動（start）、停止（stop）、再起動（restart）、再読み込み（reload）、現在の状況を確認（status）を指定できます。

Apache の設定を変更した場合は、変更内容をシステムに反映させるために Apache を再起動してください。

- Apache を起動するには、次のコマンドを実行します。

```
# /sbin/service httpd start
```

- Apache を停止するには、次のコマンドを実行します。

```
# /sbin/service httpd stop
```

- Apache 再起動するには、次のコマンドを実行します。

```
# /sbin/service httpd restart
```

- Apache の設定ファイルを再読込みするには、次のコマンドを実行します。

```
# /sbin/service httpd reload
```

- Apache の現在の状況を確認するには、次のコマンドを実行します。

```
# /sbin/service httpd status
```

- また、**ps** コマンドを実行すると、Apache プロセスの状況を次のように確認できます。

```
# /bin/ps aux | /bin/grep /usr/sbin/httpd | /bin/grep -v grep
root      6466 16.2  5.3 28716 13528 ?        Ss   09:04   0:02 /usr/sbin/httpd
apache    6469  0.2  5.5 31092 13944 ?        S    09:04   0:00 /usr/sbin/httpd
apache    6470  0.2  5.5 31092 13944 ?        S    09:04   0:00 /usr/sbin/httpd
apache    6471  0.2  5.5 31092 13944 ?        S    09:04   0:00 /usr/sbin/httpd
apache    6472  0.2  5.5 31092 13944 ?        S    09:04   0:00 /usr/sbin/httpd
apache    6473  0.2  5.5 31092 13944 ?        S    09:04   0:00 /usr/sbin/httpd
apache    6474  0.1  5.5 31092 13944 ?        S    09:04   0:00 /usr/sbin/httpd
apache    6475  0.2  5.5 31092 13944 ?        S    09:04   0:00 /usr/sbin/httpd
apache    6476  0.2  5.5 31092 13944 ?        S    09:04   0:00 /usr/sbin/httpd
```

上記のような表示がない場合は、Apache が動作していないか、あるいは Apache がインストールされていない可能性があります。Apache の起動か再起動、またはインストールを行ってください。

- システムが起動したときに自動的に Apache を自動的に起動するように設定するには、次のコマンドを実行します。

```
# /sbin/chkconfig httpd on
```

- システムが起動したときに Apache が自動的には起動しないようにするには、次のコマンドを実行します。

```
# /sbin/chkconfig httpd off
```

20.3 Apache サーバーの設定

Apache の設定ファイルは **/etc/httpd/conf/httpd.conf** と **/etc/httpd/conf.d/** 配下の ***.conf**（**ssl.conf** など）です。設定の変更は、これら設定ファイル内の**ディレクティブ**とよばれるパラメータごとに値を指定することで行います。設定を変更した場合は設定ファイル編集後に Apache を再起動するか、または、設定ファイルの再読込みの必要があります。各ディレクティブの詳細は Apache 関連の書籍を参照してください。

Apache サーバーの管理では、パフォーマンスとセキュリティを重視する必要があります。ここではその 2 点の基本的な設定について説明します。

20.3.1　Apache のパフォーマンスチューニング

➢ プロセス管理

Apache 2.2 では、子プロセスの管理は、MPM（Multi-Processing Modules）と呼ばれるモジュールグループが担当します。他のモジュールとは異なり、MPM は 3 種類のモジュールから構成されていて、そのうち 1 つのモジュールが Apache サーバーによってロードされます。現在、MPM に同梱されているモジュールは prefork、worker、perchild で、prefork と worker のみ利用可能です。

- **prefork** モジュールは、Linux 版のデフォルトで使用される MPM で、バージョン 1.3 と同様にリクエストの処理に子プロセスを生成します。
- **worker** モジュールは、リクエストの処理にスレッドを使用し、プロセスベースのサーバーよりも少ない資源で多くのリクエストを処理します。worker モジュールを使用するには、システム側の追加設定を伴います。詳細については、関連書籍を参照してください。

prefork モジュールを使用した場合、リクエストごとに httpd の子プロセスを生成して処理を行います。子プロセスの生成にはオーバーヘッドがかかるので、この子プロセス制御の設定によってはパフォーマンスに影響が生じることがあります。子プロセスの制御に関するディレクティブは表 20-1 のとおりです。

表 20-1　子プロセスに関するディレクティブ

ディレクティブ	説明	デフォルト
StartServers	起動時の初期子プロセスの数	8
MinSpareServers	待機している子プロセスの数	5
MaxSpareServers	最大待機子プロセスの数	20
MaxClients	子プロセスの最大数	256

通常はこれらの値を変更する必要はありません。しかし、同時アクセス数が多い場合は **MinSpareServers** や **MaxSpareServers** を増やすとパフォーマンスが上がることがあります。
特に搭載メモリが少ないマシンでは、子プロセスが増えすぎてスワップをする場合に全体のパフォーマンスが大きく低下することがあります。このスワップを防ぐためには httpd が使用するメモリ自体を少なくするか、**MaxClients** 数を減らしてください。また、根本的なスケーラビリティーの向上を行うために、worker モジュールの使用を検討する必要があります。

➢ DNS ルックアップ処理

DNS ルックアップとは、アクセスしてきたクライアントの IP アドレスからホスト名を求める処理のことです。これは便利な機能ですが、DNS への問い合わせは非常にコストの高い処理になります。次のようなディレクティブの設定でこの機能を停止できます（初期設定では Off になっています）。

```
HostnameLookups Off
```

DNS ルックアップを Off にすると、ホスト名ではなく IP アドレスがアクセスログファイルに記述されるというデメリットがありますが、ログファイルの IP アドレスを一括で変換する **logresolve** コマンドを使えば解消できます。**logresolve** コマンドは次のような書式で使用します。

```
# /usr/bin/logresolve < /var/log/httpd/access_log
```

➢ アクセスコントロールファイル

ディレクトリごとにアクセスコントロールを行う方法として、**.htaccess** ファイルを作成する方法があります。しかしこの処理は余分なディスクアクセスが発生することになるので注意が必要です。

たとえば、**http://www.hoge.com/dir1/dir2/dir3/hoge.html** という URL にアクセスがあった場合には、**/.htaccess**、**/dir1/.htaccess**、**/dir1/dir2/.htaccess**、**/dir1/dir2/dir3/.htaccess** と上位のディレクトリのアクセスコントロールファイルまでも読み込むことになり、余分なディスクアクセスが発生します。余計なディレクトリのアクセスを防ぐには、次のディレクティブを設定してください。

```
AllowOverride None
```

➢ CGI

通常最も問題になるのが CGI です。CGI は実行時にプロセスを生成するという重い処理をするため、ボトルネックになることがあります。この場合は、mod_perl や PHP を使うことで、CGI プロセス生成のオーバーヘッドをなくすことができます。mod_perl について Asianux Server 3 では標準で使用するように設定されています。

➢ データベース接続処理

バックグラウンドでデータベースを使うシステムでは、データベースの処理がボトルネックになることがあります。データベースのチューニングなどを行うことによってボトルネックが解消されることがあります。それとは別にデータベースとの接続処理のオーバーヘッドも無視できません。

通常の Web + DB のシステムでは、データベースへのアクセスリクエストがあるたびにデータベースへの接続処理が行われ、アクセスが終了するたびにデータベースとの接続を切断します。このオーバーヘッドを回避するには、データベースの接続状態を保持して使いまわす機能「DB コネクションプーリング」などが必要になります。アプリケーションサーバー（AS など）を導入するか、PHP なら OCILogon の代わりに OCIPLogon を使うことでデータベースとの接続を使いまわすことが可能になります。

20.3.2 Apache のセキュリティ

➢ SSL を使用したセキュアサイトの構築

Apache 2.2 には、SSL（Secure Socket Layer）機能を提供するモジュールとして mod_ssl が標準で組み込まれています。mod_ssl の設定ファイルは **`httpd.conf`** ファイルとは別で、**`/etc/httpd/conf.d/ssl.conf`** になります。デフォルトの設定で、**`httpd.conf`** から **`ssl.conf`** ファイルが読み込まれて SSL 機能を使用できる設定になっています。

mod_ssl が正しく組み込まれているかどうかの確認は、ウェブブラウザで「https://サーバーの IP アドレス」にアクセスして行ってください。接続ができていれば mod_ssl が正しく組み込まれています。

実際に、SSL 機能を使用したセキュアなサイトを構築する場合、セキュアサーバーを運営するには、VeriSign（http://www.verisign.co.jp/）や Thawte（http://www.thawte.com/）などの **CA**（認証局）への鍵の登録が必要になります。鍵の登録方法や、鍵の作成方法については各認証機関の手順にしたがってください。

- **CA（認証機関）**

  CA は、認証局または認証機関とも呼ばれます。電子メールやウェブページなどに電子印鑑（デジタル署名）を付けるときに添付する、電子印鑑証明書（デジタル ID、公開鍵証明書）を発行する機関のことを指します。電子メールなどのメッセージに対して、デジタル署名を付加することにより、メッセージの作成者が正しく本人であることや、メッセージが改ざんされていないことが確認できるようになります。この確認には、一般に公開された作成者固有の鍵（公開鍵）を使用します。デジタル ID は、この公開鍵が正しく本人であることを証明します。

  日本で利用できる CA には、米 VeriSign 社と NTT グループなどが共同で設立した日本ベリサイン株式会社や、株式会社日立製作所や富士通株式会社などが出資して設立した日本認証サービス株式会社があります。

  - 日本ベリサイン株式会社――http://www.verisign.co.jp/
  - サイバートラスト株式会社 ――http://www.cybertrust.ne.jp/
  - 日本認証サービス株式会社――http://www.jcsinc.co.jp/

- **SSL プロトコル**
  SSL（Secure Socket Layer）は、Netscape Communications 社が開発したプロトコルです。SSL は TCP/IP（Transmission Control Protocol/Internet Protocol）の上位で動作するもので、HTTP だけでなくもっと汎用的な設計となっているため、Telnet、FTP、NNTP（Network News Transfer Protocol）、LDAP（Lightweight Directory Access Protocol）といったプロトコルにも適用できます。実際の応用には SSL を実装するアプリケーションが必要です。

20.3.3 Apache 2.0 からの移行

Asianux Server 3 には、Apache 2.2 が含まれています。Apache 2.0 からの移行の際には以下の点をご確認ください。

- mod_cern_meta と mod_asis モジュールがデフォルトでロードされません。
- mod_ext_filter モジュールがデフォルトでロードされます。
- httpd の設定を 2.2 用に更新する必要があります。詳細は次のページを参照してください。
  http://httpd.apache.org/docs/2.2/en/upgrading.html
- Apache 2.0 用にコンパイルされたサードパーティモジュールは、Apache 2.2 用に再ビルドする必要があります。

第21章　PHP

この章で説明する内容

目的	PHP を使って動的サイトを構築する
機能	サーバーサイドで実行されるスクリプト、各種データベースとの接続
必要な RPM	php ―― Apache 用 php モジュール本体 php-manual ―― php のオンラインマニュアル php-oci8 ―― Oracle との連携に必要なモジュール php-pgsql ―― PostgreSQL との連携に必要なモジュール php-ldap ―― LDAP を使用するために必要なモジュール php-imap ―― IMAP を使用するために必要なモジュール php-snmp ―― net-snmp との連携に必要なモジュール php-odbc ―― ODBC との連携に必要なモジュール
設定ファイル	/etc/php.ini
章の流れ	1 PHP の概要 2 PHP の設定 3 Oracle10g との連携 4 PostgreSQL、MySQL、ODBC との連携 5 PHP 4.3 からの移行
関連 URL	関連 URL 日本 PHP ユーザ会　http://www.php.gr.jp/

21.1 PHP の概要

PHP とは HTML ファイル内に記述するタイプのスクリプト言語です。通常の CGI としても使用できますが、PHP モジュールを Apache サーバーに組み込むことにより、CGI と比較して処理速度の高速化・サーバーの負荷低減を実現できます。また Oracle や PostgreSQL といった各種データベースとの連携に優れているという特長があります。

21.2 PHP の設定

Asianux Server 3 には、PHP を利用するための Apache 用 mod_php モジュールが標準で組み込まれています。Apache 用 mod_php の設定ファイルは **/etc/httpd/conf.d/php.conf** に存在し、デフォルトの設定のままで特に変更する必要はありません。PHP 本体の設定ファイルは **/etc/php.ini** に存在し、こちらもデフォルトの設定のまま利用できますが、こちらの設定ファイルは、必要に応じて変更してください。**/etc/php.ini** での設定は、対象サーバで動作している全ての PHP に対して有効になります。ディレクトリ別に設定をする場合には、httpd.conf や.htaccess の Directory ディレクティブ内へ記述することにより設定できます。

➢ php.ini の設定例

PHP で（UTF-8 のファイルを、UTF-8 で表示）を扱うには、**/etc/php.ini** を次のように設定します。

```
output_buffering = Off
output_handler = none
default_charset = UTF-8

[mbstring]
mbstring.language = Japanese
mbstring.encoding_translation = On
mbstring.http_input = auto
mbstring.http_output = UTF-8
mbstring.internal_encoding = UTF-8
mbstring.substitute_character = none
```

PHP で（EUC-JP のファイルを、EUC-JP で表示）を扱うには、**/etc/php.ini** を次のように設定します。

```
output_buffering = Off
output_handler = none
default_charset = EUC-JP

[mbstring]
mbstring.language = Japanese
mbstring.encoding_translation = On
mbstring.http_input = auto
mbstring.http_output = EUC-JP
mbstring.internal_encoding = EUC-JP
mbstring.substitute_character = none
```

➢ httpd.conf または.htaccess の設定例

PHP で日本語（UTF-8 のファイルを、UTF-8 で表示）を扱うには、httpd.conf や.htaccess を次のように設定します。on/off の項目には php_flag、それ以外の項目には php_value を使用して設定します。

```
<Directory "/var/www/html/sample">
   php_flag output_buffering Off
   php_value output_handler none
   php_value default_charset UTF-8

   # [mbstring]
   php_value mbstring.language Japanese
   php_flag mbstring.encoding_translation On
   php_value mbstring.http_input auto
   php_value mbstring.http_output UTF-8
   php_value mbstring.internal_encoding UTF-8
   php_value mbstring.substitute_character none
</Directory>
```

PHP で日本語（EUC-JP のファイルを、EUC-JP で表示）を扱うには、httpd.conf や.htaccess を次のように設定します。on/off の項目には php_flag、それ以外の項目には php_value を使用して設定します。

```
<Directory "/var/www/html/sample">
   php_flag output_buffering Off
   php_value output_handler none
   php_value default_charset EUC-JP

   # [mbstring]
   php_value mbstring.language Japanese
```

```
    php_flag mbstring.encoding_translation On
    php_value mbstring.http_input auto
    php_value mbstring.http_output EUC-JP
    php_value mbstring.internal_encoding EUC-JP
    php_value mbstring.substitute_character none
</Directory>
```

21.3 Oracle10g との連携

php を使うことで比較的簡単に Web + Oracle DB システムを構築できます。

まず、次のコマンドを実行して Oracle10g 用拡張モジュールをインストールします。

```
# /bin/rpm -ivh php-oci8-*.i686.rpm
```

インストールが終了すると、次の内容の**/etc/php.d/oci8.ini** ファイルが作成されて、Oracle10g 用の拡張モジュールの設定が自動的に有効になります。

```
extension=oci8.so
```

次に、**/etc/sysconfig/httpd** に次のような Oracle に必要な環境変数を設定することで、Apache から Oracle を使えるようになります。Install Navigator for Oracle を用いて、Oracle をインストールした場合、このファイルは自動的に作成されます。

```
export ORACLE_BASE=/opt/app/oracle
export ORACLE_HOME=/opt/app/oracle/product/10.1.0/db_1
export ORACLE_SID=orcl
export NLS_LANG=Japanese_Japan.JA16EUC
export PATH=$ORACLE_HOME/bin:$PATH
export ORACLE_DOC=$ORACLE_HOME/doc
CLASSPATH=$ORACLE_HOME/JRE:$ORACLE_HOME/jlib:$ORACLE_HOME/rdbms/jlib
CLASSPATH=$CLASSPATH:$ORACLE_HOME/network/jlib
CLASSPATH=$CLASSPATH:$ORACLE_HOME/jdbc/lib/classes12.zip
CLASSPATH=$CLASSPATH:$ORACLE_HOME/jdbc/lib/nls_charset12.zip
CLASSPATH=$CLASSPATH:$ORACLE_HOME/sqlj/lib/translator.zip
CLASSPATH=$CLASSPATH:$ORACLE_HOME/sqlj/lib/runtime.zip
CLASSPATH=$CLASSPATH:.
export CLASSPATH
```

Oracle10g 用の拡張モジュールが正常に読み込まれているか確認するために、次の内容の **test.php** ファイルを**/var/www/html** に作成します。

```
<? phpinfo(); ?>
```

ウェブブラウザで **http://サーバー名/test.php** にアクセスしてみて、「oci8」の項目があれば、正常に読み込まれています。

また、「oci8」の項目がない場合には、Apache のエラーログファイル **/var/log/httpd/error_log** を確認してください。

➢ 文字化けについて

データベースのキャラクタセットと「NLS_LANG」、**php.ini** の mbstring.internal_encoding の文字コードが違っていると文字化けが起こることがありますが、現在の PHP では mbstring.internal_encoding に「SJIS」を設定することは推奨されていません。データベースのキャラクタセットに Shift JIS を使うときには注意が必要です。

21.4 PostgreSQL、MySQL、ODBC との連携

次のコマンドを実行して PostgreSQL、MySQL、ODBC 拡張パッケージをインストールしてください。

```
# /bin/rpm -ivh php-pgsql-*.i686.rpm
# /bin/rpm -ivh php-mysql-*.i686.rpm
# /bin/rpm -ivh php-odbc-*.i686.rpm
```

インストールが終了すると、それぞれ次の内容で **/etc/php.d/以下に pgsql.ini、mysql.ini、odbc.ini** ファイルが作成されて、拡張モジュールの設定が自動的に有効になります。

- **pgsql.ini**

```
extension=pgsql.so
```

- **mysql.ini**

```
extension=mysql.so
```

- **odbc.ini**

```
extension=odbc.so
```

各拡張モジュールが正常に読み込まれているか確認するには、**/var/www/html** に次の内容の **test.php** ファイルを作り、ウェブブラウザで **http://サーバー名/test.php** にアクセスします。

```
<? phpinfo(); ?>
```

「pgsql」、「mysql」、「odbc」のそれぞれの項目があれば、正常に読み込まれています。項目がない場合には、Apache のエラーログファイル **/var/log/httpd/error_log** を確認してください。

21.5 PHP 4.3 からの移行

Asianux Server 3 には、PHP 5.1 が含まれています。PHP 4.3 からの移行の際には以下の点をご確認ください。

- スクリプトの修正が必要になる場合があります。詳細は以下のページを参照してください。
  http://www.php.net/manual/ja/migration5.php
- **/usr/bin/php** は CLI command-line SAPI を使用してビルドされています。
- CGI SAPI には、**/usr/bin/php-cgi** を使用してください。
- php-cgi は FastCGI をサポートしています。
- 追加されたモジュール

追加モジュール	パッケージ
date, hash, Reflection, SPL, SimpleXML	php パッケージ内に埋め込み
mysqli (MySQL 4.1 用インターフェース)	php-mysql
pdo_mysql	php-mysql
|pdo, pdo_psqlite	php-pdo
pdo_pgsql	php-pgsql
pdo_odbc	php-odbc
soap	php-soap
xmlreader, xmlwriter	php-xml
dom (domxml 拡張の入れ換え)	php-xml

- 削除されたモジュール
  `dbx, dio, yp, overload, domxml`
- PEAR Framework は php-pear パッケージに組み込まれています。php-pear パッケージには次のコンポーネントが含まれています。
  - `Archive_Tar`
  - `Console_Getopt`
  - `XML_RPC`

第22章　drupal の構築

この章で説明する内容

目的	drupal の構築方法について理解する
機能	コンテンツ管理機能を備えたウェブサイトの構築
必要な RPM	**Drupal のパッケージ** drupal ―― Drupal 本体
設定ファイル	/etc/httpd/conf.d/drupal.conf
章の流れ	1 Drupal の概要 2 Drupal サイトの構築
関連 URL	**Drupal.org**　http://drupal.org/

22.1 Drupal の概要

Drupal はウェブサイトの構築・管理を支援する CMF（コンテンツ管理フレームワーク）です。

通常の CMS（コンテンツ管理システム）のようにプログラムを意識することなく新たなウェブサイトを簡単に構築・管理ができる一方、追加で Drupal を拡張するプログラムも簡単に書けるようになっています。

様々なモジュールを組み合わせることにより、すぐに Blog や Forum などの機能を利用することができます。

22.2 Drupal サイトの構築

Drupal は初期状態では無効になっていますので、使用する際にはいくつかの設定をする必要があります。大きく分けて MySQL、Apache、そして Drupal 自体の設定を行ないます。本節ではその 3 ステップについて説明します。

以降の作業は、MySQL サーバーと Apache サーバーが起動していることを前提としています。それぞれの詳しい手順は、第 12 章「MySQL データベースサーバーの構築　12.2 サーバの起動と停止」と 第 20 章「ウェブサーバーの構築 20.2 Apache サーバーの起動と停止」をご参照ください。

22.2.1 データベースの作成

Drupal ではデータの格納にデータベースを利用しており、そのデータベースを作成します。

以下の例では、データベース名 **drupal_db**、ユーザ名 **drupal_user**、パスワード **drupal_password** として作成しています。

```
# /usr/bin/mysql -uroot -p
Enter password: MySQLのrootユーザーのパスワード
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 28 to server version: 5.0.22

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> create database drupal_db default character set utf8;
mysql> grant all privileges on drupal_db.* to drupal_user@localhost identified
by 'drupal_password';
mysql> quit
```

22.2.2 Apache の設定

Drupal の Apache 設定は、**/etc/httpd/conf.d/drupal.conf** ファイルに存在します。初期状態では、コメントアウトされているので、以下のようにコメントをはずし保存します。

```
Alias /drupal /var/www/drupal

<Directory /var/www/drupal>
   Options +FollowSymLinks
   AllowOverride All
   Order allow,deny
   Allow from all
</Directory>
```

設定後 Apache を再起動してください。

```
# service httpd restart
httpd を停止中:                                            [  OK  ]
httpd を起動中:                                            [  OK  ]
```

22.2.3 Drupal の設定

(1) Drupal の設定保存ディレクトリへの書き込み権限付与

Drupal の設定を行なうため、Apache サーバによる Drupal の設定保存ディレクトリへの書き込みを許可します。

```
# chmod o+w /var/www/drupal/sites/default/
```

(2) Drupal のセットアップ

Drupal のセットアップページ(http://www.example.com/drupal/)へアクセスして Drupal のセットアップを開始します。

Choose language より「Japanese(日本語)」を選択します。(図 22-2)

図 22-2 使用言語の設定

図 22-1 データベースの設定

「データベースのタイプ」、「データベース名」、「データベースのユーザ名」、「データベースのパスワード」を入力し、[保存して次へ]ボタンを押します。(図 22-1)

図 22-1 では、例として以下の値を設定しています。

データベースのタイプ	mysql
データベース名	drupal_db
データベースのユーザ名	drupal_user
データベースのパスワード	drupal_password

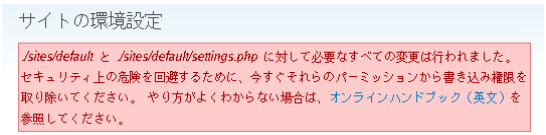


図 22-3 設定保存ディレクトリへの書き込み権限の削除

図 22-3 のメッセージに従い、Apache サーバによる Drupal の設定保存ディレクトリへの書き込み権限を元に戻します。

```
# chmod o-w /var/www/drupal/sites/default/
```

図 22-4 サイト情報 1

「サイト名」、「サイトのメールアドレス」、「ユーザ名」、「メールアドレス」を入力します。
図 22-4 では例として次の値を設定しています。

サイト名	www.example.com
サイトのメールアドレス	drupal@example.com
ユーザ名	drupal_admin
メールアドレス	drupal@example.com

図 22-5 サイト情報 2

「パスワード」、「パスワードの確認」、「デフォルトのタイムゾーン」、「クリーン URL」、「アップデート通知」を入力します。図 22-5 では例として次の値を設定しています。

デフォルトのタイムゾーン	*＜標準時＞*+0900
クリーン URL	有効
アップデート通知	チェックしない

なお、「アップデート通知」の機能は drupal.org のパッケージのアップデート通知であり、Asianux の RPM パッケージのアップデートとは異なりますので、チェックしないことを推奨します。

Drupal のインストール完了画面が表示されたら、インストール作業は終了です。

図 22-6 Drupal のインストール完了

第23章　FTP サーバーの構築

この章で説明する内容

目的	リモートクライアントとの FTP プロトコルによるファイル転送を管理する方法について理解する
機能	ログインユーザーへのファイル転送 不特定多数のユーザーへのファイル配布 FTP サーバーへのアクセス制限
必要な RPM	vsftpd ―― FTP サーバー本体
設定ファイル	/etc/vsftpd/vsftpd.conf /etc/vsftpd/ftpusers /etc/vsftpd/user_list
章の流れ	1 FTP サーバーの概要 2 FTP サーバーの起動と停止 3 FTP サーバーの設定 4 FTP サーバーのトラブルシューティング
関連 URL	**vsftpd**　http://vsftpd.beasts.org/

23.1 FTP サーバーの概要

FTP（File Transfer Protocol）は TCP/IP ネットワークで、ファイル転送を行う手段として使用されます。

FTP サーバーは、他のクライアントとのファイル転送を管理するデーモンプログラムです。マシンアカウントを持っているユーザーにマシン間のファイル転送を許可したり、不特定多数のユーザーにマシン上のファイルを配布したりするために使用されます。不特定多数のユーザーにファイルを転送する手段を匿名（anonymous）FTP と呼びます。

Asianux Server 3 では、FTP サーバーとして **vsftpd** を採用しています。

23.2 FTP サーバーの起動と停止

vsftpd の起動スクリプトは、**/etc/rc.d/init.d/vsftpd** です。起動スクリプトのオプションでは、起動（start）、停止（stop）、再起動（restart）、現在の状況を確認（status）を指定できます。

vsftpd の設定を変更した場合は、変更を反映するために vsftpd を再起動する必要があります。

- vsftpd を起動するには、次のコマンドを実行します。

```
# /sbin/service vsftpd start
```

- vsftpd を停止するには、次のコマンドを実行します。

```
# /sbin/service vsftpd stop
```

- vsftpd を再起動するには、次のコマンドを実行します。

```
# /sbin/service vsftpd restart
```

- vsftpd の現在の状況を確認するには、次のコマンドを実行します。

```
# /sbin/service vsftpd status
```

- インストール直後は、vsftpd は自動起動しない設定になっています。FTP サーバーをマシン起動時に自動起動させるときには、次のコマンドを実行します。

```
# /sbin/chkconfig vsftpd on
```

23.3 FTP サーバーの設定

vsftpd の設定は、**/etc/vsftp** 以下にあります。

- **vsftpd.conf** —— メインの設定ファイル
- **ftpusers** —— ログインを禁止するユーザーの一覧を記述するファイル
- **user_list** —— ログインを制限するユーザーの一覧を記述するファイル

23.3.1 vsftpd.conf の設定

vsftpd に関する設定ファイルのパラメータの書式は、次のようなものです。

```
オプション名=パラメータ
```

ここでは、基本的な設定項目のみ説明します。

オプション名	デフォルト値	内容
anonymous_enable	YES	匿名 FTP サーバーについて設定します。匿名 FTP サーバーにしない場合は NO に変更します。
local_enable	YES	サーバーのローカルユーザーへの接続を許可するか設定します。
write_enable	YES	書き込み権限の設定を行います。YES の場合、FTP サーバーに接続する全てのユーザーの書き込みを許可します。
local_umask	022	ローカルユーザーの umask の設定を行います。通常はこのままで構いません。
anon_upload_enable	YES (コメントアウト)	匿名 FTP サーバーを有効にしている場合、匿名ユーザーのファイルアップロードを許可するか設定します。
anon_mkdir_write_enable	YES (コメントアウト)	匿名 FTP サーバーを有効にしている場合、匿名ユーザーのディレクトリ作成を許可するか設定します。
xferlog_enable	YES	ログの取得に関する設定を行います。YES にすることでログの取得を行います。
connect_from_port_20	YES	データコネクションポートを20番に固定するかどうかを設定します。通常はこのままで構いません。
xferlog_file	/var/log/vsftpd.log (コメントアウト)	ログファイルの保存場所を変更したい場合は、コメントアウトを外し、保存場所を指定します。

オプション名	デフォルト値	内容
ascii_upload_enable	YES (コメントアウト)	ASCII モードでのアップロードについて設定します。 CGI を設置する場合には、コメントアウトを外します。
ascii_download_enable	YES (コメントアウト)	ASCII モードでのダウンロードについて設定します。 CGI を設置する場合には、コメントアウトを外します。
ftpd_banner	(コメントアウト)	ユーザーが FTP サーバーに接続したときに表示されるメッセージを設定します。コメントアウトの状態では、vsftpd のバージョンが表示されます。
chroot_list_enable	YES	上位アクセスディレクトリへのアクセスを制限するためのオプションです。詳しくは後述します。
chroot_list_file	/ etc/vsftpd/chroot_list (コメントアウト)	
chroot_local_user	(デフォルトでは 記述なし)	

詳しくは、「**man vsftpd.conf**」等を参照してください。

23.3.2 ログイン制限の設定

特定のユーザーや、システムユーザーのログインを禁止するには、**/etc/vsftpd/ftpusers** ファイルにユーザー名を記述します。複数ユーザーを記述する場合は改行で区切ります。単純に特定ユーザーのログインを禁止したい場合はこの方法で行います。

また、**/etc/vsftpd/user_list** ファイルを使用したログインの制限も可能です。常にログインを禁止する **ftpusers** ファイルとは違い、**user_list** ファイルに記述したユーザーのみログインを許可するといった制限を行うことができます。

user_list ファイルを使用したユーザーの FTP サーバーへのログインを制限するには、**vsftpd.conf** の「userlist_enable」オプションを YES に設定します。

特定のユーザーのログインを禁止させたい場合は、次の1行を追記します。ログインを禁止にするユーザーの一覧は、**user_list** ファイルに記述します。複数ユーザーを記述する場合は改行で区切ります。

```
userlist_deny=YES
```

一方、特定のユーザーのみログイン可能にする場合は、次の1行を追記します。ログインを許可するユーザーの一覧は、**/etc/vsftpd/user_list** に記述します。複数ユーザーを記述する場合は改行で区切ります。

```
userlist_deny=NO
```

どちらの設定も、ユーザーの一覧を記述するファイル名が同じであるため、途中で設定を変更する場合は、今まで許可されていたユーザーがログインできなくなったり、今までログインを禁止されていたユーザーがログインできるようになったりしないよう、注意する必要があります。

また、「userlist_enable」オプションを NO に設定した場合は、**user_list** ファイルによるユーザー制限は行われません。

23.3.3 上位ディレクトリへのアクセス制限

vsftpd のデフォルトの設定では、ユーザーがログインすると、そのユーザーのホームディレクトリ（ユーザー tanaka がログインした場合、**/home/tanaka**）にアクセスしますが、その後はほとんどのディレクトリやファイルにアクセス可能になってしまい、設定ファイル等を読み取られてしまう可能性があるため、セキュリティ上非常に危険です。

ユーザーがホームディレクトリより上のディレクトリにアクセスできないようにするためには、**vsftpd.conf** の「chroot_list_enable」、「chroot_list_file」オプションのコメントアウトを外します。また、デフォルトの **vsftpd.conf** ファイルにはない「chroot_local_user」オプションを追記する必要があります。

設定ファイルを書き換えた結果は、次のようになります。

```
# You may specify an explicit list of local users to chroot() to their home
# directory. If chroot_local_user is YES, then this list becomes a list of
# users to NOT chroot().
chroot_local_user=YES
chroot_list_enable=YES
# (default follows)
chroot_list_file=/etc/vsftpd/chroot_list
```

/etc/vsftpd/chroot_list については、**touch** コマンドで空ファイルを作成しておく必要があります。

```
# touch /etc/vsftpd/chroot_list
```

設定ができたら、vsftpd を再起動して、ユーザーがホームディレクトリより上のディレクトリにアクセスできないことを確認してください。

なお、上記設定を行った上で **chroot_list** にユーザー名を記述する（複数の場合は Enter で区切る）と、そのユーザーのみがホームディレクトリより上のディレクトリにアクセスできるようになります。

23.4 FTP サーバーのトラブルシューティング

vsftpd の起動時にエラーが出た場合、設定ファイルの記述を間違えている可能性があるので、再度設定を確認してください。次の例は、オプション名を間違えた場合のエラーメッセージです。

```
# service vsftpd start
vsftpd用のvsftpdを起動中: 500 OOPS: unrecognised variable in config file:
local_uswr
```

FTP クライアントから FTP サーバーに接続できない場合は、vsftpd の起動が有効になっていません。vsftpd の起動を行ってください。

```
$ /usr/bin/ftp ftp.your.domain.name
ftp: connect: Connection refused
```

FTP クライアントが接続したときに、次のように表示される場合、FTP サーバーの起動は行われています。

```
$ /usr/bin/ftp ftp.your.domain.name
Connected to ftp.your.domain.name.
220 (vsFTPd 2.0.5)
```

23.4.1 FTP クライアントからログインできないとき

(1) ログインを試みているユーザー名、パスワードが正しいことを確認します。

(2) SSH などの他の手段でログインを試みてログインできることを確認します。

(3) userlist_deny オプションが YES になっている場合、**user_list** ファイルにユーザー名が記述されていないか確認します。

(4) userlist_deny オプションが NO になっている場合、**user_list** ファイルにユーザー名が記述されているか確認します。

(5) サーバーのローカルユーザー名でアクセスを試みている場合、local_enable オプションが YES になっていることを確認します。

第24章　LDAP サーバーの構築

この章で説明する内容

目的	LDAP サーバーの構築方法について理解する
機能	柔軟なディレクトリサービスを提供するサーバー
必要な RPM	openldap ―― LDAP ライブラリ openldap-servers ―― LDAP サーバー openldap-clients ―― LDAP クライアントプログラム nss_ldap ―― nss による LDAP 認証パッケージ
設定ファイル	/etc/ldap.conf /etc/openldap/slapd.conf /etc/openldap/ldap.conf
章の流れ	1 LDAP の概要 2 LDAP に関する基本的な知識 3 LDAP サーバーの起動と停止 4 設定ファイルの編集 5 LDAP クライアントのコマンド 6 LDAP サーバーを利用したユーザー認証 7 アクセス制限 8 インデックス化 9 バックアップ・リストア方法 10（応用編）LDAP サーバーの複製
関連 URL	**Open LDAP Administrator's Guide**　http://www.openldap.org/doc/admin23/ **PADL**　http://www.padl.com/

24.1 LDAP の概要

LDAP（Lightweight Directory Access Protocol）は、ネットワーク上に分散する情報を統合する**ディレクトリサービス**を提供するために生まれたプロトコルです。ディレクトリサービスとは、**ディレクトリ**と呼ばれる情報の蓄積場所から、ある「キー」に関連する情報を取り出す仕組みのことで、たとえば電話帳は名前をキーにして電話番号を取り出すディレクトリサービスの 1 つだと言えます。LDAP は、現在バージョン 3 が RFC で定義されており、LDAP v3 に対応した製品同士ならば情報の交換を行うこともできます。

Asianux Server 3 では、フリーな LDAP の実装である **OpenLDAP** を採用しています。

この章では、OpenLDAP を使用して LDAP サーバーを構築する方法について解説します。LDAP の利用方法はさまざまですが、この文書ではネットワーク内のユーザー情報の統合管理に絞って解説を進めます。

通常、Linux や UNIX のユーザー情報は **/etc/passwd** のようなパスワードデータベースに格納されます。このデータベースに格納できる情報は、次のようなものに限られています。

- ユーザー名
- 所属グループ
- パスワード
- ホームディレクトリ

ユーザーに関連する情報を新たに格納するためにパスワードデータベースを拡張したいと思っても、実際にはシステム的な制約からパスワードデータベースを拡張することは難しくなっています。

しかし、ユーザー情報を LDAP サーバーで管理すれば、ユーザーに関連する情報を自由に拡張して保存し、簡単に取り出すことが可能になります。たとえば、LDAP サーバーではメールアドレスなどのようなデータを簡単に追加できます。つまり、LDAP を活用することで、単にログインアカウントとしてのユーザー管理にとどまらず、それぞれのユーザーに付随する情報を利用してさまざまなサービスを実現することが可能になります。

また、OpenLDAP は、信頼に足るディレクトリサービスを提供するために次のようなさまざまな機能を提供します。

- LDAP v3 対応
- アクセス制御
- 通信経路の暗号化
- レプリケーション
- 分散管理（referral）

もちろん上記がすべての機能ではありませんので、その他の多くの機能に関しては OpenLDAP のドキュメントなどを参照してください。

24.2 LDAP に関する基本的な知識

LDAP でシステムを構築・運用するには、基本的な用語とそれらの役割を理解しておくことが必要不可欠です。このパートではそれぞれの項目について簡単に説明します。

➢ エントリ

エントリとは、LDAP ディレクトリ内でのユニットの単位です。各エントリはユニークな Distinguished Name（**DN**）で識別されます。エントリはオブジェクトクラスという単位（集合体）に属します。
オブジェクトクラスの定義は**/etc/openldap/schema/** ディレクトリ内にある各種のスキーマファイルで確認できるので、参照してみてください。

➢ 属性

属性とは、あるエントリと関連した情報です。たとえば、ある組織を LDAP エントリとして LDAP サーバーに格納するケースを考えると、組織と関連した属性として住所などがあげられます。ただし、LDAP で格納できるデータはなにも組織に限られたものだとは限りません。たとえば、同じサーバーに人もエントリとして格納できます。その場合、人のエントリの属性にはメールアドレスなどの属性が定義されることになります。
属性はエントリを構築するのに必要不可欠なものと、明示的に指定がなくてもエントリを指定できるオプショナルなものの 2 種類に分かれます。オブジェクトクラスごとに、必須の属性とそうでない属性が定義されています。

➢ LDAP のデータ管理

LDAP において、各エントリは階層ツリー形式で管理されます。伝統的に、この階層ツリーの構造には実際の地理や組織での階層の境界が反映されている場合が多々あります。最もわかりやすい方法は、図 24-1 のように、ツリーをインターネットドメイン名を元に構築することです。

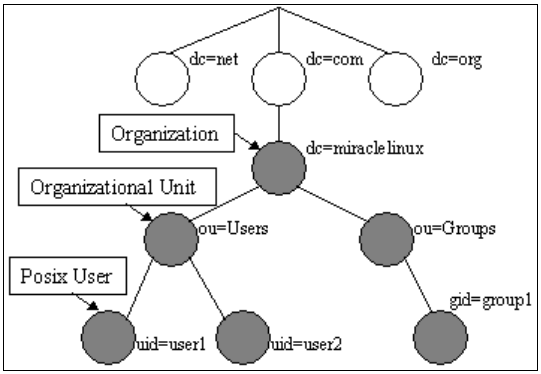


図 24-1 LDAP におけるデータの配置

➢ LDIF

LDAP Data Interchange Format（LDIF）は ASCII テキストのフォーマットを利用した、LDAP エントリの表示です。後述する **ldapadd** コマンドなどで LDAP サーバーへデータをインポートする場合、LDIF 形式のデータファイルを使用します。LDIF のフォーマットは次のようになります。

```
# コメント
dn: <識別名>
<属性記述子>: <属性値>
<属性記述子>: <属性値>
```

上記の LDIF では、エントリの 1 つとそれに関連する属性の定義を行っています。

各エントリは、必要な数の**<属性記述子>: <属性値>** ペアを含み、エントリの定義の終了には空白行が使用されます。「#」で始まる行はコメントとして扱われ、LDAP サーバーからは無視されます。

新しい識別名や属性の定義では、行の左端から記入するようにしてください。行の左端に単一のスペースかタブを入力すると、前の行の続きとみなされます。

```
dn: cn=Asianux Server,dc=asianux, dc=
 com
cn: Asianux
 Server
```

上記の例では、識別名に「cn=Asianux Server, dc=asianux, dc=com」が、属性「cn」には値「Asianux」が割り当てられます。同一の属性値が複数存在する場合は、数行にまたがって定義します。

```
dn: cn=Manager,dc=asianux, dc=com
cn: Manager
cn: Administrator
```

24.3 LDAP サーバーの起動と停止

LDAP サーバーを使用するには、LDAP の実体であるデーモンプログラム **slapd** を起動する必要があります。**slapd** の起動／停止スクリプトは**/etc/rc.d/init.d/ldap** となっています。起動スクリプトのオプションでは、起動（start）、停止（stop）、再起動（restart）、現在の状況を確認（status）を指定できます。

- LDAP サーバーを起動するには、次のコマンドを実行します。

```
# /sbin/service ldap start
```

- LDAP サーバーを停止するには、次のコマンドを実行します。

```
# /sbin/service ldap stop
```

- LDAP サーバーを再起動するには、次のコマンドを実行します。

```
# /sbin/service ldap restart
```

- LDAP サーバーの状態を確認するには、次のコマンドを実行します。

```
# /sbin/service ldap status
```

また、**chkconfig** を使用することで、マシン起動時に LDAP サーバーを自動的に立ち上げるかどうかを設定できます。

- 現在の設定を確認するには、次のコマンドを使用します。

```
# /sbin/chkconfig --list ldap
```

- マシン起動時に LDAP サーバーを立ち上げるように設定するには、次のコマンドを実行します。

```
# /sbin/chkconfig ldap on
```

- マシン起動時に LDAP サーバーを立ち上げないように設定するには、次のコマンドを実行します。

```
# /sbin/chkconfig ldap off
```

24.4 設定ファイルの編集

24.4.1 /etc/openldap/slapd.conf

LDAP サーバーを効果的に使用するには、**/etc/openldap/slapd.conf** を適切に編集する必要があります。主な設定箇所を示しますが、ここには記述されていない設定パラメータがいくつも存在します。これらのパラメータについては、**man slapd.conf** と入力してマニュアルページを参照してください。

➢ スキーマファイルの追加

LDAP のエントリは、1 つか複数の**オブジェクトクラス**という集合体に属しています。オブジェクトクラスは前述のスキーマファイルで定義されていますが、LDAP サーバーがどのスキーマファイルを参照するかは、**slapd.conf** の設定によって決まります。デフォルトではいくつかのスキーマファイルがすでにインクルードされていますが、インクルードされていないスキーマファイルを使用するためには、**slapd.conf** ファイルを手動で修正して、そのスキーマをインクルードする必要があります。次の例では **samba.schema** というスキーマファイルをインクルードするように変更しています。

```
include         /etc/openldap/schema/inetorgperson.schema
include         /etc/openldap/schema/nis.schema
include         /etc/openldap/schema/samba.schema（追加します）
...（省略）...
```

➢ ベースサフィックスの定義

suffix パラメータで**ベースサフィックス**を設定できます。ベースサフィックスとは、その LDAP サーバーが管理する範囲を決定するための指定です。

管理したい範囲が「asianux.com」だった場合、ベースサフィックスは「dc=asianux, dc=com」となり、データは常のこのサフィックスより下位のノードに追加され、データ検索もこのサフィックスの下に存在するデータに対して行われます（たとえば uid=user1, dc=asianux, dc=com など）。このためには、**slapd.conf** の **suffix** パラメータを次のように設定します。

```
suffix              "dc=asianux,dc=com"
```

それぞれの「=」の左側にある文字列は属性の種類に対応しています。LDAP に使用される一般的な文字列とその属性を表 24-1 に示します。

表 24-1 LDAP に使用される一般的な文字列

文字列	X.500 AttributeType
CN:	commonName
L:	localityName
ST:	stateOrProvinceName
O:	organizationName
OU:	organizationalUnitName
C:	countryName
STREET:	streetAddress
DC:	domainComponent
UID:	userid

➢ LDAP サーバー管理者の指定

OpenLDAP では、LDAP サーバー管理者を指定する必要があります。サーバー管理者は、LDAP サーバーに設定されたアクセス制御に関係なく LDAP の操作を行うことができます。サーバー管理者は、**slapd.conf** の **rootdn** パラメータで指定します。次のように設定すると、「cn=Manager, dc=asianux, dc=com」の識別名で表されるユーザーがサーバー管理者に設定されます。もちろん、rootdn に設定する識別名は、「cn=Manager」以外でも問題ありません。

```
rootdn              "cn=Manager,dc=asianux,dc=com"
```

➢ LDAP サーバー管理者のパスワード設定

設定ファイルの中には LDAP サーバーの管理者のためのパスワードを記述する必要がありますが、プレーンテキストで明記してしまうとセキュリティ上問題があります。OpenLDAP では暗号化したパスワードも取り扱えるので、次のコマンドを実行して、MD5 化したパスワードを設定ファイルに記入してください。例では「secret」という文字列を MD5 で暗合化しています。

```
# /usr/sbin/slappasswd -s secret -h {MD5}
{MD5}Xr4ilOzQ4PCOq3aQ0qbuaQ==
```

上記の結果をもとに、**slapd.conf** の **rootpw** エントリーを編集します。

```
rootpw {MD5}Xr4ilOzQ4PCOq3aQ0qbuaQ==
```

➢ LDAP サーバーのログレベル設定

LDAP サーバーのログはデフォルトの設定では収集できません。収集方法については「24.5.1 LDAP サーバーの動作確認」で記述しますが、loglevel パラメータの設定はこのファイルにて行う必要があります。デフォルトの loglevel は 256 です。

```
loglevel 256
```

24.4.2 設定後の注意

LDAP サーバーは ldap ユーザー権限で動くので、ディレクティブで指定されているディレクトリやその内部のファイルに対して ldap ユーザーに書き込み権限がないと、データを更新できません。特に **slapadd** コマンドは root ユーザーで実行する必要があるので、コマンド実行後は directory パラメータで指定されたディレクトリ（デフォルトでは**/var/lib/ldap** ディレクトリ）の所有者を ldap ユーザーに変更してください。

```
# /bin/chown -R ldap /var/lib/ldap
```

24.5 LDAP クライアントのコマンド

openldap-client パッケージには、複数の LDAP クライアントプログラムが含まれています。これらのツールを使用することで、LDAP サーバー内に存在するデータを検索、追加、修正、削除などを行うことが可能になります。

- **ldapsearch** —— データ検索
- **ldapadd** —— データ追加
- **ldapmodify** —— データ更新
- **ldapdelete** —— データ削除
- **ldappasswd** —— データのパスワード変更
- **ldapmodrdn** —— データ名の変更

ここでは、データの検索と追加について解説します。他のコマンドについては、それぞれのマニュアルページを参照してください（たとえば **man ldapdelete** など）。

24.5.1 LDAP サーバーの動作確認

slapd デーモンが動作しているかどうかは前述の **service** コマンドを使用するか、LDAP のログを確認すればわかります。ログを収集したい場合には以下の設定を施して下さい。

/etc/syslog.conf を次のように設定します。

```
local4.*                /var/log/ldap.log
```

/etc/sysconfig/ldap を以下のように作成します。

```
SLAPD_OPTIONS="-l LOCAL4"
```

上記を設定後、次のコマンドを実行することで**/var/log/ldap.log** が作成され、LDAP サーバーへのコネクト時などのログが収集できます。

```
# /sbin/service syslog restart
```

クライアントとしてデータを参照できる状態にあるかは **ldapsearch** コマンドを使用して確認できます。

```
$ /usr/bin/ldapsearch -x -b '' -s base '(objectclass=*)' namingContexts
# extended LDIF
#
# LDAPv3
# base <> with scope baseObject
# filter: (objectclass=*)
# requesting: namingContexts
#

#
dn:
namingContexts: dc=asianux,dc=com

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```

24.5.2 LDAP サーバーへデータの追加

LDAP サーバーにデータを追加するには、**ldapadd** コマンドを使用します。ldapadd は LDIF 形式のファイルを必要とするので、最初に LDIF ファイルを作成して、そのファイルを ldapadd に渡すことでデータを登録します。

example.ldif というファイルを作成する例を示します。

```
dn: dc=asianux,dc=com
objectclass: dcObject
objectclass: organization
dc: asianux
o: ASIANUX SERVER

dn: cn=Manager,dc=asianux,dc=com
objectclass: organizationalRole
cn: Manager

dn: ou=People,dc=asianux,dc=com
objectClass: organizationalUnit
ou: People

dn: ou=Group,dc=asianux,dc=com
objectClass: organizationalUnit
ou: Group
```

ファイルを作成したら、**ldapadd** でこれらのエントリを追加します。

```
$ /usr/bin/ldapadd -x -f example.ldif
adding new entry "dc=asianux,dc=com"
ldap_add: Strong(er) authentication required (8)
       additional info: modifications require authentication
```

上記のコマンドは、どのユーザーでも LDAP サーバーに書き込みが可能な場合にのみ有効です。「24.7 アクセス制限」で記載しているアクセス制限などが LDAP サーバーに設定されている場合、管理者ユーザーなどの書き込み権限のあるユーザーを指定する必要があります。

```
$ /usr/bin/ldapadd -x -D "cn=Manager,dc=asianux,dc=com" -W -f example.ldif
Enter LDAP Password:<slapd.confのrootpwエントリーで指定したパスワード>
adding new entry "dc=asianux,dc=com"

adding new entry "cn=Manager,dc=asianux,dc=com"

adding new entry "ou=People,dc=asianux,dc=com"

adding new entry "ou=Group,dc=asianux,dc=com"
```

-D オプションの後にバインド（接続）するユーザー名を指定します。**-W** オプションはパスワードのプロンプトを表示させるオプションです。**-w <パスワード>** を替わりに指定すると、パスワードプロンプトが現れずに、指定された文字列をパスワードとして使用して認証を行います。

なお、これらのオプションは他の LDAP クライアントツールと共通です。

24.5.3 LDAP サーバーの参照

追加したエントリがディレクトリ中にあるかどうかを確認するには LDAP クライアントが必要ですが、ここでは **ldapsearch** ツールを使うことにします。次の例の「dc=asianux,dc=com」の部分は運用するサイトに合わせて適切な値に書き換えてください。

```
$ /usr/bin/ldapsearch -x -b 'dc=asianux,dc=com' '(objectclass=*)'
# extended LDIF
#
# LDAPv3
# base <dc=asianux,dc=com> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#

# asianux.com
dn: dc=asianux,dc=com
objectClass: dcObject
objectClass: organization
dc: asianux
o: ASIANUX SERVER

# Manager, asianux.com
dn: cn=Manager,dc=asianux,dc=com
objectClass: organizationalRole
cn: Manager

# People, asianux.com
dn: ou=People,dc=asianux,dc=com
objectClass: organizationalUnit
ou: People

# Group, asianux.com
dn: ou=Group,dc=asianux,dc=com
objectClass: organizationalUnit
ou: Group

# search result
search: 2
result: 0 Success

# numResponses: 5
# numEntries: 4
```

上記の例では、LDAP に格納されているすべてのエントリを参照できます。

24.6 LDAP サーバーを利用したユーザー認証

nss_ldap パッケージを使用すると、LDAP サーバーに格納されている Linux ユーザーの情報をユーザー認証に使用することが可能になります。情報を格納するサーバーには **openldap-servers** が、情報を参照するクライアントには上記の **nss_ldap** パッケージのほかに **openldap** と **openldap-clients** の各パッケージが必要となります。

クライアントが LDAP を参照するには **/etc/nsswitch.conf**、**/etc/ldap.conf**、**/etc/openldap/ldap.conf**、**/etc/pam.d/system-auth** の各ファイルを編集する必要があります。

設定方法には、手動で設定する方法と、**authconfig-tui** コマンドを使用して自動で設定する方法があります。

手動で設定するには、**/etc/nsswitch.conf** で、次の passwd、shadow、group の各エントリーに、「ldap」という値を追加します。

```
passwd: files ldap
shadow: files ldap
group:  files ldap
```

/etc/ldap.conf は次のように設定します。

```
host 127.0.0.1
base dc=asianux,dc=com
```

/etc/openldap/ldap.conf も同じように設定します。

```
host 127.0.0.1
base dc=asianux,dc=com
```

/etc/pam.d/system-auth で、次の auth、account、password、session の各エントリーに、「pam_ldap.so」という値を追加します。

```
auth        required      pam_env.so
auth        sufficient    pam_unix.so nullok try_first_pass
auth        sufficient    pam_ldap.so use_first_pass
auth        requisite     pam_succeed_if.so uid >= 500 quiet
auth        required      pam_deny.so

account     required      pam_unix.so
account     sufficient    pam_succeed_if.so uid < 500 quiet
account     [default=bad success=ok user_unknown=ignore service_err=ignore
system_err=ignore authinfo_unavail=ignore] pam_ldap.so
account     required      pam_permit.so

password    requisite     pam_cracklib.so try_first_pass retry=3
password    sufficient    pam_unix.so md5 shadow nullok try_first_pass
use_authtok
password    sufficient    pam_ldap.so use_authtok
password    required      pam_deny.so

session     optional      pam_keyinit.so revoke
session     required      pam_limits.so
session     [success=1 default=ignore] pam_succeed_if.so service in crond quiet
use_uid
session     required      pam_unix.so
session     optional      pam_ldap.so
```

なお、これらの設定はデフォルトの状態なので、設定によってはそれぞれの値を変えたり、設定パラメータを追加したりする必要があります。詳しくは LDAP のドキュメントを参考にしてください。

これらの操作は設定ツールを使用して行うこともできます。**authconfig-tui** を起動して、[LDAP サポートを有効にする]にチェックをいれることで、これらのファイルを簡単に、まとめて設定することが可能です。

authconfig-tui を起動するには、次のように入力します。

```
# /usr/sbin/authconfig-tui
```

図 24-2 のような画面が表示されますので、ユーザー情報の[LDAP を使用]、認証の[LDAP 認証を使用]にチェックを入れて、LDAP による認証を有効にするように設定します。

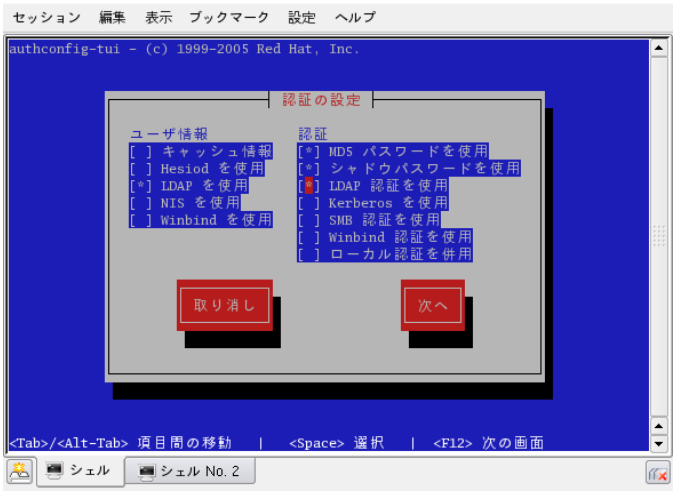


図 24-2 authconfig の設定画面（1）

画面下部の[次へ]ボタンを選択するとLDAP 設定画面が表示されるので、サーバーとベース DN に対して適当な値を入力し、[OK]ボタンを選択します（図 24-3）。

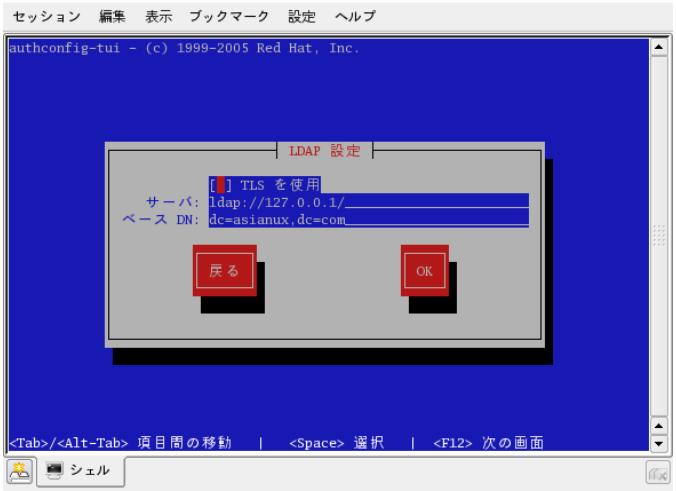


図 24-3 authconfig の設定画面（2）

LDAP サーバーからユーザー情報が取得できるか確認するために、ユーザー satou を追加して、LDAP にユーザー情報を登録します。

```
# /usr/sbin/useradd satou
```

/etc/passwd ファイルを開き、次のように satou のある行（追加直後であれば最終行）をメモします。左側からユーザー名・パスワード（x とある場合は shadow 化されています）・ユーザー ID・グループ ID・（コメントフィールド）・ユーザーのホームディレクトリ・ユーザーのコマンドインタプリタとなっています。

```
satou:x:513:513::/home/satou:/bin/bash
```

適当なディレクトリにファイルを作成し（例：/root/satou.ldif）、次のように入力します。パスワードについては、312 ページで紹介した ldappasswd コマンドを使用して設定すると良いでしょう。

```
dn: uid=satou,ou=People,dc=asianux,dc=com
uid: satou                                          ※ユーザー名
cn: satou                                           ※ユーザー名
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
userPassword: {MD5}oAEw/rV4t24ecopKcwm2yg==         ※パスワード
shadowLastChange: 13744
shadowMax: 99999
shadowWarning: 7
loginShell: /bin/bash                               ※ユーザーのコマンドインタプリタ
uidNumber: 513                                      ※ユーザーID
gidNumber: 513                                      ※グループID
homeDirectory: /home/satou                          ※ユーザーのホームディレクトリ
```

保存したら、LDAP サーバーに登録します。

```
$ /usr/bin/ldapadd -x -D "cn=Manager,dc=asianux,dc=com" -W -f satou.ldif
Enter LDAP Password:
```

LDAP サーバーからユーザー情報が取得できるか確認するには、**getent** コマンドを使用します。slapd サーバーが起動していることと、**/etc/ldap.conf** と**/etc/openldap/ldap.conf** が正常に設定されていることを確認したら、次のコマンドを実行してください。

```
# /usr/bin/getent passwd
```

LDAP サーバーに格納されているユーザーの情報を取得できたら成功です。

同じく、LDAP サーバーに格納されているグループの情報も確認できます。

```
# /usr/bin/getent group
```

24.7 アクセス制限

LDAP に登録されたデータの参照、更新などに関しては、LDAP の機能としてアクセス制限を設定できます。インストール後の設定では、LDAP のコマンドを利用すればだれでも（暗号化はされていますが）パスワードフィルードの内容を確認できるので、パスワードフィールドのアクセス制御を行う必要があります。

パスワードフィールド以外にも、ユーザーのエントリに関してはユーザー自身で編集可能にしておくほうが利便性がよくなります。そこで、LDAP のアクセス制御の方法について、説明します。

今回は、次のルールに従ったアクセス制御を行います。

（1）LDAP 管理者はパスワードの参照、更新が可能
（2）ユーザーは自身のパスワードの参照、更新が可能
（3）ユーザーはパスワード以外の自分のエントリを参照、更新が可能
（4）他人はパスワード以外のエントリを参照可能

アクセス制限は、LDAP サーバーの**/etc/openldap/slapd.conf** ファイルで行うので、次のような設定を**slapd.conf** ファイルに追加します。ここでは、「cn=Manager,dc=asianux,dc=com」エントリを LDAP の管理者として使用しています。

```
access to attrs=userPassword
        by self write
        by anonymous auth
        by * none

access to *
        by self write
        by * read
```

各フィールドの意味は、次のとおりです。

- **access to** の行は、アクセス制限の対象エントリを指定します。
- **by** の行が、アクセス制限の内容です。
- **write** を指定すると更新と参照が可能になります。**read** を指定すると参照のみが可能になります。
- **by anonymous auth** の設定により、認証前のユーザーが、認証のためにエントリを利用することを許可します。

24.8 インデックス化

LDAP では、適切なエントリの**インデックス**を作成することで、検索性能の向上を図ることができます。ただし、LDAP のデータベースにエントリが何も存在しない場合は、インデックスは作成できません。また、データの更新が頻繁にあるようなフィールドをインデックスとして選択してしまうと、逆にパフォーマンスが落ちてしまう場合もあるので注意が必要です。

たとえばユーザー認証のために LDAP を使用している場合、**objectClass**、**uidNumber**、**gidNumber**、**uid**、**cn**、**memberUid** などのエントリがよく参照されると考えられるため、これらのフィールドに対してインデックスを作成することにします。**slapd.conf** には、次のように記述します(既存の設定に含まれていることもあります)。

```
index       objectClass, uidNumber, gidNumber, uid, cn, memberUid       eq
```

応用として、たとえば Samba の認証にも LDAP のユーザーを使用したい場合、**sambaSID** をあわせてインデックス化するのも有効でしょう。

```
index       objectClass, uidNumber, gidNumber, uid, cn, memberUid, sambaSID  eq
```

インデックスの作成は、データベースの一貫性を保つために、slapd が停止している状態で行う必要があります。**slapd** が停止していることを確認してから、次のコマンドを実行すると、現在のデータベースの情報を元に、インデックスが作成されます。もうすでに **slapd** が起動している場合、一度停止してから **slapindex** コマンドを発行してください。

また、**slapindex** を実行するには、**/var/lib/ldap** 下に **DB_CONFIG** ファイルが必要です。**DB_CONFIG** ファイルのサンプルが **/etc/openldap** にあるので、こちらをコピーして使うと良いでしょう。

```
# cp /etc/openldap/DB_CONFIG.example /var/lib/ldap/DB_CONFIG
```

slapindex の実行は次のように行います。

```
# /sbin/service ldap stop
# /usr/sbin/slapindex
# /bin/chown -R ldap /var/lib/ldap
```

大量のユーザー登録を行う場合などには、ユーザーアカウントの登録ごとにインデックスの更新も発生することになり処理に時間がかかるので、すべてのデータ登録完了後にインデックスを設定することを検討しても良いでしょう。また、**slapd.conf** の設定を変更して別のフィールドをインデックス化したい場合も、インデックスの再作成が必要となります。

24.9 バックアップ・リストア方法

ldap データベースのバックアップ・リストアは、ldap サービスの動作状況によって、手順が異なります。(表 24-2) また各手順で作成されたバックアップデータには互換性がありません。リストア作業は各バックアップデータに対応した手順で実施する必要があります。(表 24-3)

表 24-2:ldap サービス動作状況とバックアップ・リストア方法の対応表

	使用するバックアップ・リストア方法	
ldapサービスの動作状況	バックアップ	リストア
動作時	ldapsearch	ldapadd
停止時	slapcat	slapadd
停止時	ldapデータベースファイルをコピー	コピーしたldapデータベースファイルをリストア

表 24-3:バックアップデータとリストア方法の対応表

	リストア方法		
ldap データバックアップ方法	ldapadd	slapadd	コピーしたldapデータベースファイルをリストア
ldapsearch コマンド	○	×	×
slapcat コマンド	×	○	×
ldapデータベースファイルをコピー	×	×	○

24.9.1 ldap サービス動作時のバックアップ方法

ldap サービス動作時は、バックアップコマンドとして **ldapsearch** を使用します。

但し、通常 ldap サーバーは、ldapsearch コマンドでの問い合わせに対して、500 エントリ分しか返答しません。500 件以上のエントリが存在する場合は、slapd.conf に以下の設定を実施する必要があります。設定実施後は ldap サービスの再起動が必要です。

```
sizelimit unlimited
```

ldapsearch でのバックアップは次のように実施します。

```
# /usr/bin/ldapsearch -x -D "ldapサーバーのrootdn" -w パスワード -h IPアドレス -LLL > バックアップファイル名.ldif
```

24.9.2 ldap サービス停止時のバックアップ方法

ldap サービス停止時は、バックアップコマンドとして **slapcat** を使用するか、もしくは ldap データーベースファイルをコピーする方法の 2 種類ございます。

slapcat でのバックアップは次のように実施します。

```
# /usr/sbin/slapcat -l バックアップファイル名.ldif
```

ldap データベースファイルをコピーする場合は、以下の様に ldap データベースのディレクトリ配下を固めてしまうと便利かと思われます。(以下のコマンド例の場合は、/tmp 配下にバックアップファイルが作成されます)

```
# cd /var/lib/ldap/
# tar zcvf /tmp/バックアップファイル名.tgz *
```

24.9.3 ldapsearch で作成したバックアップデータのリストア方法

ldap のサービスを停止します。

```
# /sbin/service ldap stop
```

必要に応じて/var/lib/ldap 内の古い ldap データベースファイルのバックアップを取得します。

```
# tar zcvf ldapdir_back.tgz /var/lib/ldap/*
```

/var/lib/ldap 内の古い ldap データベースファイルを削除します。

```
# rm -rf /var/lib/ldap/*
```

ldap のサービスを起動します。

```
# /sbin/service ldap start
```

データをリストアします。

```
# /usr/bin/ldapadd -x -D "ldapサーバーのrootdn" -w パスワード -h IPアドレス -f
ldapsearchで作成したldifデータ
```

実際にリストアされているかは、ldapsearch コマンドで確認してください。

24.9.4 slapcat で作成したバックアップデータのリストア方法

ldap のサービスを停止します。

```
# /sbin/service ldap stop
```

必要に応じて/var/lib/ldap 内の古い ldap データベースファイルのバックアップを取得します。

```
# tar zcvf バックアップファイル名.tgz /var/lib/ldap/*
```

/var/lib/ldap 内の古い ldap データベースファイルを削除します。

```
# rm -rf /var/lib/ldap/*
```

データをリストアします。

```
# /usr/sbin/slapadd -l slapcatで作成したldifデータ
```

ファイルの所有者、及び所有グループを ldap に変更します。

```
# chown ldap:ldap /var/lib/ldap/*
```

サービスを起動します。

```
# /sbin/service ldap start
```

実際にリストアされているかは、ldapsearch コマンドで確認してください。

24.9.5 コピーした ldap データベースファイルのリストア方法

ldap のサービスを停止します。

```
# /sbin/service ldap stop
```

必要に応じて/var/lib/ldap 内の古い ldap データベースファイルのバックアップを取得します。

```
# tar zcvf バックアップファイル名.tgz /var/lib/ldap/*
```

/var/lib/ldap 内の古い ldap データベースファイルを削除します。

```
# rm -rf /var/lib/ldap/*
```

データをリストアします。（以下は「24.9.2 ldap サービス停止時のバックアップ方法」で作成したバックアップファイルの場合の例です）

```
# cd /var/lib/ldap/
# tar zxvf /tmp/バックアップファイル名.tgz
```

ファイルの所有者、及び所有グループを ldap に変更します。

```
# chown ldap:ldap /var/lib/ldap/*
```

サービスを起動します。

```
# /sbin/service ldap start
```

実際にリストアされているかは、ldapsearch コマンドで確認してください。

24.10 （応用編）ldap サーバーの複製

ldap サーバーを複製するには、2 種類の方法があります。（図 24-4 参照）

複製により、負荷を分散したり、障害時の信頼性向上が可能です。以前のバージョン(MILRACLE LINUX V4.0 以前)は slurpd のみの対応でしたが、Asianux Server 3 からは syncrepl も利用可能になりました。

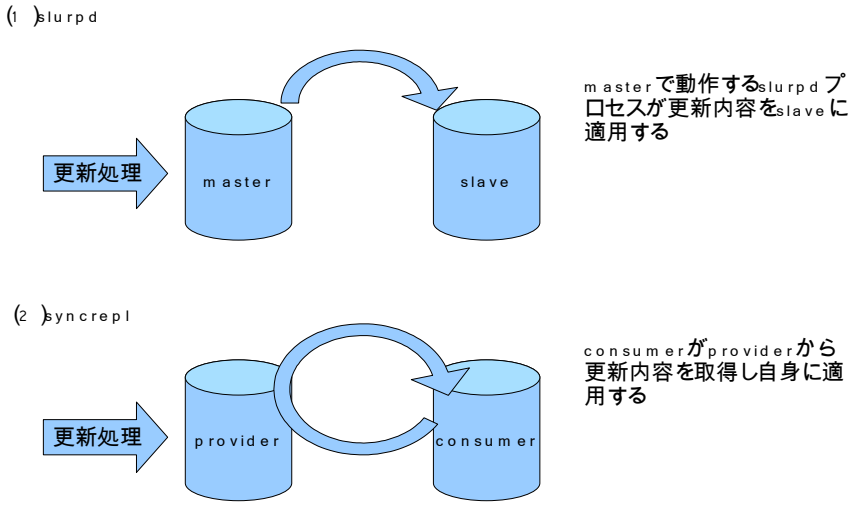


図 24-4:ldap サーバーの複製方式

24.10.1 slurpd を使用した複製

slurpd を使用して複製するには、master・slave 双方の slapd.conf に設定が必要です。

また、作業前に master 及び slave サーバーの ldap サービスを停止しておきます。

master 側の**/etc/openldap/slapd.conf**（以下 slapd.conf）の設定例は以下のとおりです。

slapd.conf ファイルの末尾に設定を追加します。

```
#レプリケーション先のホスト名（すなわちslaveサーバー）
replica host=slave.example.com
#レプリケーション時に使用するslaveサーバーのrootdn
       binddn=cn=replica,dc=example,dc=com
#認証方法（パスワード認証の場合はsimple）
       bindmethod=simple
#パスワード
       credentials=miracle
#レプリケーションログの記述先
replogfile      /var/lib/ldap/master-replica.log
```

slave 側の slapd.conf の設定例は以下のとおりです。master 側と同様に、slapd.conf ファイルの末尾に設定を追加します。

```
#slaveのrootdnを指定
updatedn        "cn=replica,dc=example,dc=com"
#masterサーバーのLDAP URLを記述します
updateref       "ldap://master.example.com/"
```

slave 側のその他の設定については、master 側から複製の設定を除いたものと同じです。ただし、rootdn と rootpw に関しては、master で設定した複製用の rootdn とパスワードが設定されているものとします。

master、slave1 両方の設定完了後、master 側に存在する ldap データを slave 側にもコピーします。

※コピー時は、master を停止する必要がありますので、「24.9.2 ldap サービス停止時のバックアップ方法」を参照して下さい。また、手順と異なるのは、リストア先が slave サーバーとなる点です。

コピー終了後、slave サーバー、master サーバーの順で ldap サービスを開始してください。

```
# /sbin/service ldap start
```

以降 master サーバーの ldap データベースに加えられた変更は、slave サーバーに反映される事になります。

24.10.2 syncrepl を利用した複製

syncrepl を使用して複製する場合も、provider・consumer 双方の slapd.conf に設定が必要です。また、事前に provider・consumer サーバー双方の ldap サービスを停止しておきます。

provider 側の**/etc/openldap/slapd.conf**（以下 slapd.conf）の設定例は以下となります。slapd.conf ファイルの末尾に設定を追加します

```
#チェックポイントを書き込む間隔（以下は100操作もしくは10分経過すると書き込む）
syncprov-checkpoint 100 10
#セッションログの利用を指定（以下は100操作分）
syncprov-sessionlog 100
#syncreplオーバーレイの利用を指定（必須）
overlay syncprov
```

comsumer 側の slapd.conf の設定例は以下のとおりです。slapd.conf ファイルの末尾に設定を追加します。

```
#providerから見て一意の3桁の数字
syncrepl rid=123
#providerのホスト名
 provider=ldap://provider.example.com
#レプリケーションモードを指定
 type=refreshAndPersist
#プロバイダに接続出来ない場合のリトライ回数（以下は5秒間隔で10回アクセスを試み、60秒間隔で無制限に再接続を試みる）
 retry="5 10 60 +"
#指定したldapドメイン配下をレプリケーションする
 searchbase="dc=example,dc=com"
#providerへの認証方法を指定
 bindmethod=simple
#providerへの接続に利用する、providerのrootdnを指定
 binddn="uid=Administrator,ou=Users,dc=example,dc=com"
#上記で指定したproviderのrootdnのパスワード
 credentials=miracle
```

consumer 側のその他の設定については、provider 側の設定から複製の設定部分を除いたものと同じです。

provider、consumer 両方の設定完了後、consumer 側の ldap データベースファイルを削除します。※この作業は、consumer 側の ldap データベースに何も登録されていなければ必要ありません。

```
# rm -rf /var/lib/ldap/*
```

provider、consumer の順で ldap サービスを開始すると、自動で ldap データベースの同期が行われます。

```
# /sbin/service ldap start
```

以降、provider の ldap データベースに加えられた変更は、slave サーバーに反映される事になります。

第25章　セキュリティ対策

この章で説明する内容

目的	システムのセキュリティ対策に必要な情報の取得方法について理解する
機能	セキュリティに関する設定
必要な RPM	xinetd —— inetd に代わるインターネットサービスデーモン acl —— ACL 管理ツール
設定ファイル	/etc/xinetd.conf /etc/xinetd.d/
章の流れ	1 セキュリティ対策の概要 2 セキュリティ対策 3 ネットワークセキュリティ対策 4 システムセキュリティ対策 5 その他の注意点
関連 URL	**Asianux Technical Support Network**　https://tsn.miraclelinux.com/tsn_local/ **JPCERT/CC**　http://www.jpcert.or.jp/

25.1　セキュリティ対策の概要

ネットワークを構成するマシンの中で、サーバーが提供する役割は非常に大きく、サーバーの安定運用のためにセキュリティ対策は欠かせないものです。サーバーのセキュリティ対策が不十分だと、クラッカーの侵入を許してサーバー自身の運用に問題が発生するだけでなく、サービスを受けるクライアントに、さらには他のネットワークに対しても問題が波及します。

以上のような理由から、システム管理者はサーバー構築時から十分にセキュリティ対策に気を付ける必要があります。ただし、セキュリティの強化は、エンドユーザーの利便性と相反する場合もあります。したがって、サーバーを含めたネットワークのセキュリティポリシーを定め、セキュリティの強化がもたらすメリットとデメリットを十分ユーザーに認識してもらうこともセキュリティ対策の重要な要素の 1 つです。

セキュリティ対策とは、どこからもアクセスできないサーバーを構築することではなく、セキュリティ対策によってもたらされるメリットとコストが釣り合うように、サイトに適したセキュリティポリシーを決定して、そのポリシーを徹底することだと言えます。

本章では、Asianux Server 3 を使用する上で、セキュリティ対策のために必要な情報について記載します。

25.2　セキュリティ対策

具体的なセキュリティ対策にはさまざまなものがありますが、最低限の対策として、次のようなことに注意する必要があります。

- 不要なサービスをインストールしたり実行したりしない。
- セキュリティ情報を収集して、セキュリティ問題の修正された最新バージョンのソフトウェアを利用する。
- システムのログを記録して不正アクセスが行われていないことを確認する。
- 万が一不正アクセスが行われた場合に備え、バックアップを準備しておく。

セキュリティ対策は多方面に亘るため、サーバーにインストールされて実行されているソフトウェアが増えれば増えるほど、対策に必要なコストが増加します。まずは OS やソフトウェアのインストール時には、提供するサービスに応じた適切なソフトウェアのみをインストールすることが重要です。

また、不正アクセスを防ぐためのセキュリティアップデートは日々更新されています。DVD-ROM から Asianux Server 3 をインストールした直後であっても、**Asianux Technical Support Network** (https://tsn.miraclelinux.com/tsn_local/) のページを確認し、アップデートされたパッケージがあれば、必ずパッケージのアップデートを行うようにしてください。

root 以外のユーザーによる su の許可を、wheel グループに所属するユーザーのみに限定する場合は、**/etc/pam.d/su** ファイルの以下行から # を削除し、該当行を有効にしてください。

```
#auth            required        pam_wheel.so use_uid
```

本製品は、デフォルトの状態では、サーバーが提供するサービスを限定してあります。つまりシステム管理者は、提供予定のサービスを **chkconfig** コマンドなどで明示的に指定する必要があります。

サーバーの運用開始前には以下のコマンドを実行して、サーバー起動時に開始されるサービスや、xinetd 経由で提供されるサービスを確認しましょう。

```
# /sbin/chkconfig --list
```

chkconfig コマンドの使用方法は 15 ページからの「システムの起動と終了」を参照してください。

次節からネットワーク／システムセキュリティ対策について具体的に説明します。

25.3　ネットワークセキュリティ対策

25.3.1　xinetd

xinetd は、inetd デーモンに代わるインターネットスーパーサーバーです。inetd では、提供するサービスの設定は**/etc/inetd.conf** で行いましたが、xinetd では、**/etc/xinetd.conf** と、**/etc/xinetd.d/**配下のファイルを使って提供するサービスの設定を行います。

25.3.2　xinetd の設定

xinetd の設定は、**/etc/xinetd.conf** と**/etc/xinetd.d/**配下のサービスごとの設定ファイルで行います。**/etc/xinetd.conf** は **xinetd** の基本設定ファイルで、**/etc/xinetd.d/**配下の各ファイルは、それぞれのサービスごとの設定ファイルです。

➢ /etc/xinetd.conf

xinetd.conf では、各サービス共通の設定や、サービスごとの設定ファイルが置かれているパスを記述します。
デフォルトでは、/etc/xinetd.d ディレクトリ配下にサービスごとの設定ファイルが置かれています。
次の設定は、デフォルトの xinetd.conf からの抜粋です。

```
defaults
{
   log_type      = SYSLOG daemon info
   log_on_failure= HOST
   cps           = 50 10
}
includedir /etc/xinetd.d
```

- **log_type**…syslog ログの出力方法を指定します。デフォルトでは、syslog を通して、ファシリティ daemon、プライオリティ info で出力すると設定しています。
- **log_on_failure**…サービスの起動に失敗したときに出力する情報を指定します。HOST は、接続元のアドレスをログに出力する指定です。
- **cps**…1 番目の引数は、1 秒間に接続できる最大コネクション数です。2 番目の引数は、最大コネクション数を超えて接続した場合に、再アクセス可能になるまでの時間を指定します。
- **includedir**…/etc/xinetd.d 配下にある、サービスごとの設定ファイルを読み込む設定をしています。

xinetd.conf の詳細な設定方法については、「**man 5 xinetd.conf**」を参照してください。

➢ /etc/xinetd.d/

/etc/xinetd.d ディレクトリ配下には、サービスごとの設定ファイルが置かれています。これらのファイルは、/etc/xinetd.conf で読み込むように指定されています。

例として、Samba の SWAT を xinetd 経由で起動する際の設定ファイル/etc/xinetd.d/swat を以下に示します。

```
service swat
{
    port = 901
    socket_type = stream
    wait = no
    only_from = 127.0.0.1
    user = root
    server = /usr/sbin/swat
    log_on_failure += USERID
    disable = yes
}
```

- **server**…xinetd から起動するコマンドを指定します。
- **only_from**…接続を許可するホストを限定できます。同様に **no_access** を設定すると接続を許可しないホストを指定できます。
- **disable**…サービスの有効/無効を設定します。**disable=no** にすることで、該当するサービスが有効になります。

サービスごとの設定ファイルの書式は、xinetd.conf と同じです。詳細な設定方法については、「man 5 xinetd.conf」を参照してください。
各サービスを有効にする場合は、以下のコマンドを実行してください。

```
# /sbin/chkconfig サービス名 on
```

設定ファイルを変更した後は、以下のコマンドを実行して、新たな設定を xinetd に反映させてください。

```
# /sbin/service xinetd restart
```

25.3.3　アクセス制御

xinetd の設定ファイルでは、only_from や no_access を使用することで、xinetd 経由で提供されるサービスに対してアクセス制限を行うことができますが、さらに、カーネルレベルでの強力なアクセス制限方法として、ファイアーウォールによるアクセス制限があります。ファイアーウォールの設定方法の詳細は、第 26 章「ファイアーウォール」を参照してください。

25.4　システムセキュリティ対策

25.4.1　ACL（Access Control Lists）

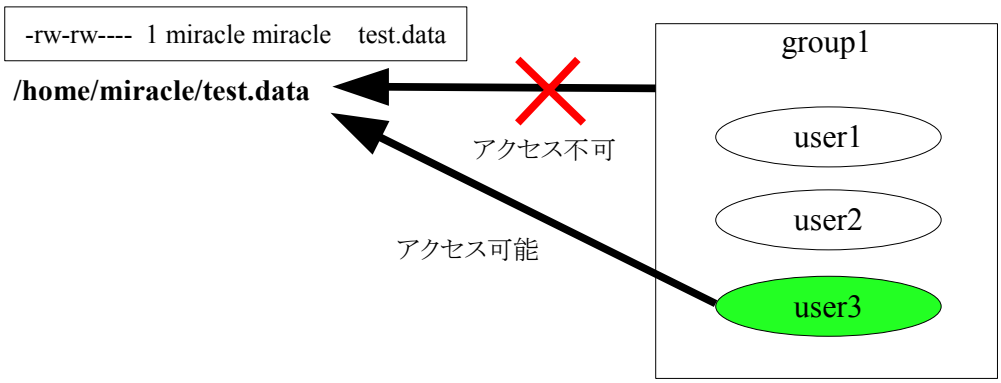
ACL はカーネル 2.6 から標準で採用された機能であり、基本的なパーミッションでは実現が困難なきめ細かなアクセスコントロールが可能となります。

例えば、あるグループに対して読込み・書込み・実行を拒否するが、その中の特定のユーザーにだけ読込み・書込みの権限を与えたい場合などに ACL を利用することで実現できます。

以下の例では miracle ユーザー（miracle グループ）が所有している **test.data** ファイルに対して、miracle ユーザーとそのグループである miracle グループのみ読込み・書込みの許可をパーミッション（-rw-rw----）で与えています。

そのため他グループである group1 グループに所有しているユーザー（user1,user2,user3）からは **test.data** にアクセスできません。

ACL を利用することで group1 グループの user3 ユーザーに対してのみ読込み・書込みを許可することが可能になります。



25.4.2 ACL の使用条件

ACL を使用するためには、ファイルシステムによって以下の設定を行う必要があります。

ここでは、**ext3** で ACL を使用するための例を示します。

- **mount** コマンドのオプションに **acl** を指定します。

```
# /bin/mount -t ext3 -o remount,acl /dev/sda5 /home
```

- **/etc/fstab** の第 4 フィールドにマウントオプション（**acl**）を追加します。

```
LABEL=/home      /home      ext3    acl      1 2
```

- マウント状態（**ACL** 利用可能）の確認をします。

```
# /bin/mount
/dev/sda2 on / type ext3 (rw)
/dev/sda1 on /boot type ext3 (rw)
略
/dev/sda5 on /home type ext3 (rw,acl)
```

25.4.3 ACL の設定

ACL を設定するには **setfacl** コマンド、表示するには **getfacl** コマンドを使用します。

ここでは、25.4.1 の例をもとに ACL の設定について説明します。

- 現在のパーミッションを確認します。
  所有ユーザー、グループに対して読込み・書込みを許可されています。

```
$ /bin/ls -l test.data
-rw-rw----  1 miracle miracle 0  8月 15 13:38 test.data
```

- 確認のため、user1 から user3 ユーザーで **/home/miracle/test.data** にアクセスしてみます。
  「許可がありません」と表示され、アクセスすることはできません。

```
$ /bin/cat /home/miracle/test.data
cat: /home/miracle/test.data: 許可がありません
```

- 次に ACL を利用し、user3 ユーザーに対して読込み・書込みを許可します。

```
$ /usr/bin/setfacl -m u:user3:rw /home/miracle/test.data
```

- ACL の設定が正しくされていることを確認します。

```
$ /bin/ls -l test.data
-rw-rw----+ 1 miracle miracle 5  8月 15 14:08 /home/miracle/test.data
          |
     ACLが設定されるとパーミッションの表示の後ろに"+"が付きます

$ /usr/bin/getfacl /home/miracle/test.data
# file: home/miracle/test.data
# owner: miracle
# group: miracle
user::rw-
user:user3:rw-  ー＞ user3に対して読込み・書込みを許可
group::rw-
mask::rw-
other::---
```

- user3 ユーザーで**/home/miracle/test.data** へのアクセス確認を行います。

```
$ /bin/cat /home/miracle/test.data
test
```

- ACL を利用したグループに対するアクセス権限付与方法

```
$ /usr/bin/setfacl -m g:group1:r /home/miracle/test.data
$ /usr/bin/getfacl test.data
# file: home/miracle/test.data
# owner: miracle
# group: miracle
user::rw-
user:user3:rw-
group::rw-
group:group1:r--  ー＞ group1に対して読込みを許可
mask::rw-
other::---
```

以上のように、ACL を使用することで、今までのパーミッション設定だけではできなかった、ユーザー、グループ単位でのきめ細かなファイルへのアクセス制御が可能になるため、不用意な情報漏洩を防ぎ、システム全体のセキュリティを高めることができます。

25.4.4 Exec-Shield

Exec-Shield とは、バッファオーバーフロー攻撃を防御する機能です。

プログラムのバグを利用してプロセスを乗っ取る攻撃はいくつか存在しますが、バッファオーバーフローはそのような手法の 1 つです。

プログラムのバグを狙った攻撃を回避する一番の方法は、バグを修正したパッチをサーバーに適用することです。しかし、まだパッチが提供されていないバグに対する攻撃は、この方法では回避することができません。このような潜在的な攻撃に対して有効な機能が Exec-Shield です。しかし、Exec-Shield も万能ではありません。パッチの適用をせずにサーバーを稼動し続けるのは、危険ですので行わないようにしてください。

25.4.1 Exec-Shield の設定

インストール後の状態では、Exec-Shield の設定は、明示的に有効にした実行ファイル以外はすべて無効になっています。

- 現在の設定内容を確認するには、**/proc/sys/kernel/exec-shield** ファイルの内容を参照します。

```
# /bin/cat /proc/sys/kernel/exec-shield
1
```

表示された値によって現在の状態がわかります。

0 －－ 常に無効

1 －－ 基本的に無効。(実行ファイルごとに有効にする) (デフォルト値)

2 －－ 常に有効。(実行ファイルごとに無効にする)

- 常に有効にするためには、ブートコマンドの kernel 行のパラメータに **exec-shield=2** を追加します。
  以下の例は、**/etc/grub.conf** を直接修正して、パラメータを追加する方法です。一時的に追加する場合は、GRUB のブートメニューを編集してください。

```
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title Asianux (2.6.18-8.9AX)
        root (hd0,0)
        kernel /vmlinuz-2.6.18-8.9AX ro root=LABEL=/ exec-shield=2
        initrd /initrd-2.6.18-8.9AX.img
```

効率良く設定するには、常に有効に設定し、不具合が起きたプログラムのみ Exec-Shield の対象から外すようにします。プログラム単位で Exec-Shield 有効/無効の設定を行うには、**execstack** コマンドを使用します。

```
# /usr/bin/execstack <オプション> 実行ファイル
```

execstack コマンドのオプション

-c 有効

-s 無効

-q 設定内容の確認

- Apache を Exec-Shield の対象にする。

```
# /usr/bin/execstack -c /usr/sbin/httpd
```

- Apache を Exec-Shield の対象外にする。

```
# /usr/bin/execstack -s /usr/sbin/httpd
```

- Apache の Exec-Shield が有効か無効か確認する。

```
# /usr/bin/execstack -q /usr/sbin/httpd
- /usr/sbin/httpd  ー> 有効の場合、先頭が「 - 」となっている
# /usr/bin/execstack -q /usr/sbin/httpd
X /usr/sbin/httpd  ー> 無効の場合、先頭が「 X 」となっている
```

➢ NX（No eXecute）機能搭載 CPU マシンへの対応について

最近のサーバーに搭載されている NX 機能搭載 CPU を使用すると、Exec-Shield 機能が強化され CPU 上での不正コード実行を阻止することが可能です。

NX 機能は対応 CPU を搭載したサーバーであれば標準で有効になっていますが、Java プログラムの中にはこの機能が有効になっていると実行エラーになってしまうものがありますので、その場合は次の方法でこの機能を無効にしてください。

NX 機能を無効にするには、ブートコマンドの kernel 行のパラメータに **noexec=off** を追加します。
以下の例は、**/etc/grub.conf** を直接修正して、NX 機能を常時無効にする方法です。一時的に追加する場合は、GRUB のブートメニューを編集してください。

```
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title Asianux (2.6.9-11.19AX)
        root (hd0,0)
        kernel /vmlinuz-2.6.9-11.19AX ro root=LABEL=/ noexec=off
        initrd /initrd-2.6.9-11.19AX.img
```

25.5 その他の注意点

Telnet や FTP で使われるパスワードは、ネットワーク上を平文で流れます。そのため、同じネットワークに接続されているマシンからパスワードを読み取ることが可能です。したがって、サーバーへのアクセス手段として、Telnet や FTP の代わりに **ssh** や **scp** を利用することを推奨します。ssh や scp は、openssh-server や openssh-clients のパッケージに含まれています。

セキュリティ対策は、インストール時だけでなく、サーバーの運用中は常に継続的に注意を払うことが重要です。サーバーの運用を停止させるまではセキュリティには十分注意を払い、信頼性の高いサービス提供を心がける必要があります。

第26章　ファイアーウォール

この章で説明する内容

目的	ファイアーウォールを設定する方法について理解する
機能	パケットフィルタリング
必要な RPM	iptables —— ファイアーウォール設定ツール system-config-securitylevel —— 簡易 GUI ファイアーウォール設定ツール guarddog —— GUI ファイアーウォール設定ツール
設定ファイル	/etc/sysconfig/iptables
章の流れ	1 ファイアーウォールの概要 2 iptables 3 guarddog
関連 URL	**Japanese FAQ Project**　http://www.linux.or.jp/JF/JFdocs/netfilter-faq.html

26.1 ファイアーウォールの概要

26.1.1 Netfilter

バージョン 2.6 の Linux カーネルには、パケットフィルタリングの機能を提供する **Netfilter** が組み込まれており、この機能を利用することで、ファイアウォールを構築することができます。異なるネットワーク間のアクセス制御（パケットフィルタリング）や、自ホストへのアクセス制御が可能です。

26.2 iptables

パケットフィルタリングは、カーネルの機能（**Netfilter**）によって提供されますが、その機能を利用するためには、**iptables** というツールが必要です。

本節では、iptables を使用してファイアーウォールを設定する方法について説明します。

26.2.1 iptables の起動と停止

iptables の起動/停止スクリプトは、**/etc/rc.d/init.d/iptables** です。

起動スクリプトのオプションでは、起動（start）、停止（stop）、再起動（restart）、現在の状況を確認（status）を指定できます。

- iptables を起動するには、次のコマンドを実行します。

```
# /sbin/service iptables start
```

- iptables を停止するには、次のコマンドを実行します。

```
# /sbin/service iptables stop
```

- iptables を再起動するには、次のコマンドを実行します。

```
# /sbin/service iptables restart
```

- iptables の状態を確認するには、次のコマンドを実行します。

```
# /sbin/service iptables status
または
# /sbin/iptables -L
```

- 一時的に外部からの接続をすべて遮断するには、次のコマンドを実行します。
  すべてのホストからのアクセスが拒否されることになるので、実行後に何らかの操作を行うためには、ホストに対してアクセスできる状況を用意する必要があります。

```
# /sbin/service iptables panic
```

26.2.2 iptables の概念

➢ チェイン

iptables では、パケットをチェックするポイントをチェインと呼び、チェインにはパケットをどのように処理するかのルールを設定します。役割に応じて以下の 5 種類のチェインに分けられます。

チェイン	役割
INPUT	パケットの受信に関するルールを設定する。
OUTPUT	パケットの送信に関するルールを設定する。
FORWARD	パケットの転送に関するルールを設定する。
PREROUTING	パケットを受信する際に、宛先アドレスを書き換える（DNAT）。
POSTROUTING	パケットを送信する際に、送信元アドレスを書き換える（SNAT）。

➢ テーブル

iptables では、用途に応じて適切なテーブルを指定する必要があります。パケットに対するルールは、チェインで設定しますが、テーブルにより使用できるチェインが異なります。

テーブル	用途	使用できるチェイン
filter	パケットフィルタリング	INPUT, FORWARD, OUTPUT
nat	アドレス変換	PREROUTING, POSTROUTING, OUTPUT
mangle	特殊なパケット操作	PREROUTING, OUTPUT

➢ ターゲット

ターゲットでは、ルールに適合したパケットに対する動作を設定します。ここでは主要なターゲットのみを以下に示します。

ターゲット	動作
ACCEPT	パケットを通過させる。
DROP	パケットを破棄する。
REJECT	パケットを破棄し、送信元に ICMP パケットを送り返して通知する。
MASQUERADE	送信元アドレスを書き換える。POSTROUTING チェインでのみ使用可能。
DNAT	宛先アドレスを書き換える。PREROUTING, OUTPUT チェインで使用可能。

26.2.3　iptables の設定

➢ 自動設定ツールによる簡易設定

system-config-securitylevel-tui コマンドは、iptables の設定を自動で行えるツールです。

```
# /usr/bin/system-config-securitylevel-tui
```

図 26-1 のような画面が表示されるので、「セキュリティレベル」で[有効]にチェックし、[OK]を選択します。

図 26-1　セキュリティレベルの設定

[OK]を選択した後、「setenforce: SELinux is disabled」と表示される場合がありますが、問題はありません。

➢ iptables コマンドの書式

iptables コマンドには多くのオプションがありますが、ルールを追加する際の書式は以下のとおりです。

```
iptables [-t テーブル] <アクション> [ルール] [-j ターゲット]
```

-t オプションには、テーブルを指定します。省略した場合には filter テーブルを使用します。詳細は、「**man 8 iptables**」の「テーブル」を参照してください。

アクション には、どのチェインに何を行うのか、オプションを指定します。アクションの指定は省略できません。詳細は、「**man 8 iptables**」の「オプション」「コマンド」を参照してください。

ルールには、チェインに追加するルールを指定します。ルールには、パケットの送信元アドレス(**-s**)、送信先アドレス(**-d**)、プロトコル(**-p**)などを指定します。詳細は、「**man 8 iptables**」の「オプション」「パラメータ」を参照してください。

-j オプションには、ターゲットを指定します。詳細は、26.2.2 iptables の概念 および、「**man 8 iptables**」の「ターゲット」を参照してください。

➢ iptables コマンドの設定例

iptables の設定ファイルは**/etc/sysconfig/iptables** です。自動設定ツール **system-config-securitylevel-tui** では設定できない詳細なルールを設定する場合には、このファイルに直接ルールを記述します。iptables サービスの起動スクリプトは、このファイルの内容に従ってルールを設定します。

/etc/sysconfig/iptables ファイルにルールを記述する場合は、iptables コマンドのオプション部分から書き始めます。以下にいくつかの記述例を示します。

- 特定のホストまたはネットワークからのアクセスを制限する場合。

```
-A INPUT -s host1.specific.domain.name -j REJECT
```

上記エントリが設定ファイル**/etc/sysconfig/iptables** に記述された状態で iptables サービスを起動すると、ホスト host1.specific.domain.name からのあらゆるアクセスは拒絶されます。**-A** オプションがルールの追加を、**INPUT** が外部からのアクセスに対する設定であることを、**-s** オプションが対象となるソースアドレス(発信元の IP アドレスまたはホスト名)を意味します。

- 10.1.1.1 ～ 254 から ns.your.domain.name の telnet ポート(23)への接続を禁じる。

```
-A INPUT -p tcp -s 10.1.1.0/24 -d ns.your.domain.name --dport 23 -j REJECT
```

上記エントリは、**-d** が接続先のホストを、**-p** がプロトコルを、**--dport** が宛先ポート番号を意味します。

- ns.your.domain.name の telnet ポート(23)への接続について、10.1.2.1 ～ 254 からのみ許可し、他のすべてのネットワークからの接続を禁止する場合。

```
-A INPUT -p tcp -s 10.1.2.0/24 -d ns.your.domain.name --dport 23 -j ACCEPT
-A INPUT -p tcp -s  0.0.0.0/24 -d ns.your.domain.name --dport 23 -j REJECT
```

上記エントリでは、複数のルールを組み合わせることで、特定のホストまたはネットワークからの接続のみを許可しています。最初のエントリで 10.1.2.1 ～ 254 からの ns.your.domain.name への telnet ポートへの接続を許可し、次のエントリですべてのネットワークからの telnet ポートへの接続を禁止しています。この場合、先に設定された接続許可は、続いて設定されたすべてのネットワークからの接続禁止よりも優先されます。

- 特定のホストまたはネットワークへのアクセスを制限する場合。

```
-A OUTPUT -d host2.specific.domain.name -j REJECT
```

上記エントリが設定ファイル **/etc/sysconfig/iptables** に記述された状態で iptables サービスを起動すると、ホスト host2.specific.domain.name へのアクセスを禁止することになります。他のオプションについては、INPUT と同様に指定できます。

また、**iptables** コマンドを直接実行した後に、以下のコマンドを実行することでも、**/etc/sysconfig/iptables** にルールを保存することができます。この場合、iptables サービス再起動後にも、ルールが反映されます。

```
# /sbin/service iptables save
```

/etc/sysconfig/iptables に記述したルールを、システム起動時に反映させるためには、次のコマンドを実行します。

```
# /sbin/chkconfig iptables on
```

逆に、設定ファイルに記述した IP テーブルの設定を、システム起動時に反映させないようにするには、次のコマンドを実行します。

```
# /sbin/chkconfig iptables off
```

26.2.4 パケットの転送

iptables でパケットの転送に関するルールの設定を行う際には、**FORWARD** チェインを指定します。FORWARD チェインは、パケットの転送に関するルールを設定しますが、Asianux Server 3 では、カーネルのパケット転送機能がデフォルトで無効になっています。そのため、事前にパケットの転送機能を有効にしておく必要があります。

コマンドでパケットの転送機能を有効にするためには、**/proc/sys/net/ipv4/ip_forward** ファイルに 1 を書き込みます。このファイルが 1 であればパケット転送が有効、0 であれば無効となります。

```
# /bin/echo 1 > /proc/sys/net/ipv4/ip_forward
```

コマンドで設定した場合には、システムを再起動した際に設定が無効となってしまいます。

恒久的に設定を有効にするためには、**/etc/sysctl.conf** ファイルを変更してください。変更後、以下のコマンドを実行することで、設定を即時反映させることができます。

```
# /sbin/sysctl -p
```

26.2.5 IP マスカレード

IP マスカレードは、プライベートアドレスを持つホストから送られてきたパケットの送信元アドレスとポート番号を書き換え、インターネットに接続できるようにする仕組みです。

IP マスカレードを設定する際には、あらかじめ 26.2.4 の操作でパケットの転送を有効にしてからルールを設定します。

➢ IP アドレスが動的な場合

グローバル IP アドレスが動的な場合（ISP から固定グローバル IP アドレスが割り当てられていないような場合）には、ルールを設定する際に、ジャンプするターゲットとして **MASQUERADE** を指定します。

MASQUERADE ターゲットは、nat テーブルの POSTROUTING チェインのみで有効です。

以下の例では、インターネット側のインターフェイス eth0 に対して、IP マスカレードを設定しています。

```
-t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

➢ IP アドレスが静的な場合

固定グローバル IP アドレスを割り当てられている場合には、ルールを設定する際に、ジャンプするターゲットとして **SNAT** を指定します。

MASQUERADE ターゲットは、nat テーブルの POSTROUTING チェインのみで有効です。

以下の例では、インターネット側のインターフェイス eth0 の IP アドレス XX.XX.YY.YY に対して、IP マスカレードを設定しています。

```
-t nat -A POSTROUTING -o eth0 -j SNAT --to-source XX.XX.YY.YY
```

➢ ヘルパーモジュール

IP マスカレードを行う際には、実現したい機能に応じて、モジュールをロードする必要があります。

システム起動時にロードさせる場合には、**/etc/rc.d/rc.local** へ記述を追加してください。

```
# ftpの転送を許可する場合
/sbin/modprobe ip_nat_ftp

# tftpの転送を許可する場合
/sbin/modprobe ip_nat_tftp

# IRCの転送を許可する場合
/sbin/modprobe ip_nat_irc
```

26.2.6 ステートフルパケットフィルタ

ステートフルパケットフィルタは、現在の接続の状態を把握して、パケットのヘッダ情報だけでなく、接続状態からも判断して、パケットの許可/拒否を決定する方法です。

kernel-2.6 に実装されている netfilter（iptables）は、ステートフルパケットフィルタに対応しています。

ステートフルパケットフィルタでは、パケットの状態が、以下のいずれかであるかを判断することができます。

状態	説明
NEW	新規接続開始パケット
ESTABLISHED	既存の接続の応答パケット
RELATED	既存の接続に関連のあるパケット
INVALID	既存の接続に関連のないパケット

➢ ステートフルパケットフィルタの設定例

iptables コマンドでステートフルパケットフィルタの設定を行う場合、以下のように state モジュールを使用します。詳細は、「**man 8 iptables**」を参照してください。

```
iptables [-t テーブル] -m state --state <state> [-j ターゲット]
```

以下に/etc/sysconfig/iptables に記述する形式で、設定例を示します。

- 内部ネットワーク（192.168.1.0/24 eth0 側）からの、新規 SSH 接続については許可する場合。

```
iptables -A INPUT -m state --state NEW -i eth0 -s 192.168.1.0/24 -p tcp --dport
22 -j ACCEPT
```

- 既に接続を確立しているパケットと、それに関連するパケットについては接続を許可する場合。

```
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
```

26.3　Guarddog

Guarddogとは、ファイアーウォールの複雑な設定を GUI で簡単に行えるユーティリティです。

iptables コマンドを使用してファイアーウォールを構築することが難しい場合でも、Guarddog を使用して簡単に設定を行うことが可能です。

26.3.1　Guarddog の起動

Guarddog を起動するには、GUI デスクトップ画面左下の[Start]メニューから「設定」－「システム管理」－「Guarddog」を選択します。

Guarddog が起動すると、のような画面が表示されます。

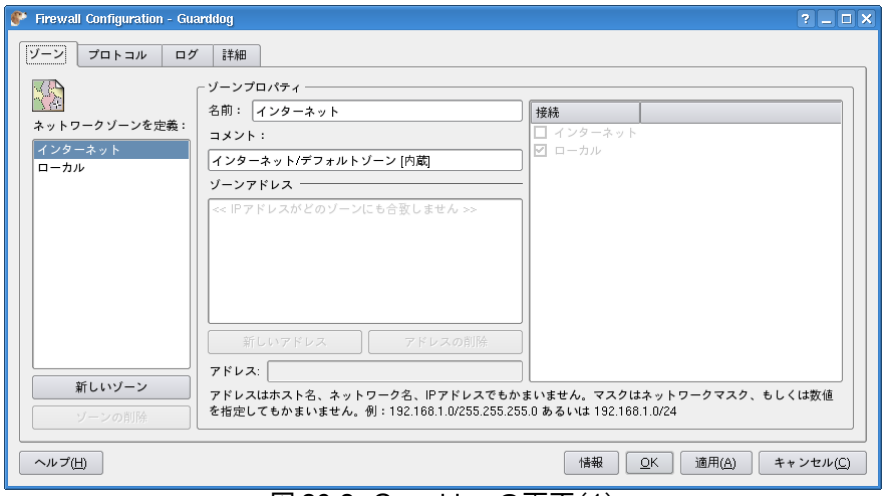


図 26-2　Guarddog の画面（1）

初回起動時に、「Guarddog ファイアーウォールの/etc/rc.firewall を読み込むことが出来ませんでした。」というエラーが表示されます。

Guarddog を使用してファイアーウォールの設定を行う場合は、[OK]をクリックし次項を参考に設定を行ってください。

もし、既に iptables コマンドを使用してファイアーウォールの設定を行っていて、設定を上書きされたくない場合は、この画面で[OK]をクリックした後、図 26-3 の画面で[キャンセル(C)]ボタンをクリックしてください。

26.3.2 Guarddog の設定

この節では例として、外部から Web サーバー(ポート番号 80)および SSH(ポート番号 22)への接続を受け付ける設定を行います。

プロトコルごとの許可設定を行うには、図 26-3 の画面で、[プロトコル]タブをクリックします。

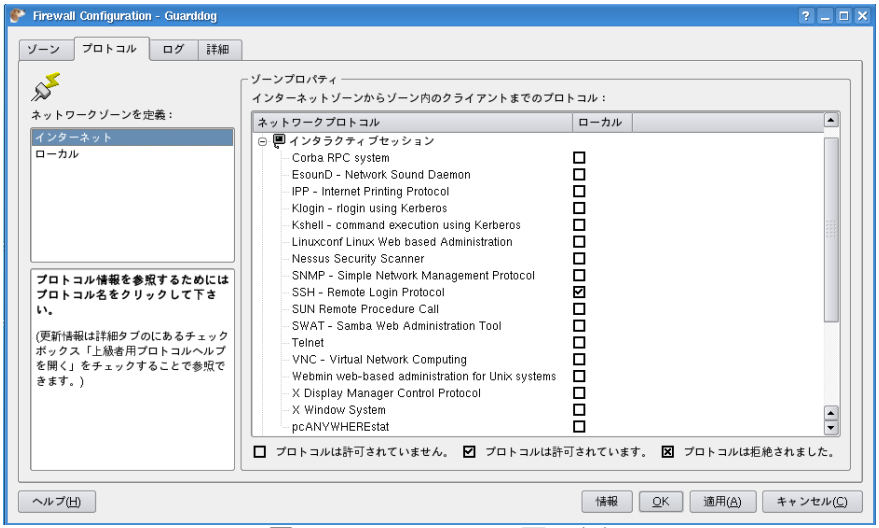


図 26-3 Guarddog の画面（2）

画面左のネットワークゾーンは、デフォルトでは「**インターネット**」と「**ローカル**」の 2 つのゾーンが用意されています。「インターネット」は、外部からのデータ受信について設定し、「ローカル」は、外部へのデータ送信について設定します。

[ネットワークゾーンを定義:]から[ローカル]を選択し、ゾーンプロパティの[インタラクティブセッション]-[SSH – Remote Login Protocol]と、[ファイルの転送]-[HTTP – World Wide Web]のチェックボックスにチェックします。

設定ができたら、[適用(A)]ボタンをクリックし、ファイアーウォール構成を変更します。

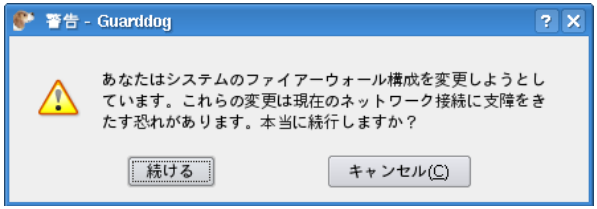


図 26-4 Guarddog の画面（3）

変更後のファイアーウォールの設定は、**iptables** コマンドで確認できます。

```
# /sbin/iptables -L
```

Guarddog によって設定されたローカルゾーンのプロトコル設定は、**f0to1** というチェインに設定されています。

```
Chain f0to1 (3 references)
target     prot opt source      destination
ACCEPT     tcp  --  anywhere    anywhere     tcp spts:1024:65535 dpt:http state
NEW
ACCEPT     tcp  --  anywhere    anywhere     tcp spts:1024:65535 dpt:webcache
state NEW
ACCEPT     tcp  --  anywhere    anywhere     tcp spts:1024:65535 dpt:http-alt
state NEW
ACCEPT     tcp  --  anywhere    anywhere     tcp spts:1024:65535 dpt:irdmi state
NEW
ACCEPT     tcp  --  anywhere    anywhere     tcp spts:1024:65535 dpt:ddi-tcp-1
state NEW
ACCEPT     tcp  --  anywhere    anywhere     tcp spts:1024:65535 dpt:ssh state
NEW
ACCEPT     tcp  --  anywhere    anywhere     tcp spts:0:1023 dpt:ssh state NEW
logdrop    all  --  anywhere    anywhere
```

設定が正しく行われたかどうかを確認するには、Web サーバーと SSH サーバーを起動し、他のマシンからサーバーに接続できるか確認を行います。

また逆に、自サーバーから Web サイトの閲覧や、他のマシンへの SSH 接続が行えないことを確認します。

Guarddog で設定したルールは/etc/rc.firewall に保存されます。Guarddog の設定を適用すると、iptables のルールを上書きしますので、Guarddog と iptables コマンドの設定は併用することはできません。

iptables コマンドで作成したルールを再度適用したい場合には、iptables サービスを再起動する必要があります。

第27章　ログ管理

この章で説明する内容

目的	システムのセキュリティ対策に必要な情報の取得方法について理解する
機能	セキュリティに関する設定
必要な RPM	sysklogd ―― システムロギングデーモン logrotate ―― ログ管理ツール
設定ファイル	/etc/syslog.conf /etc/logrotate.d/ /etc/sysconfig/iptables
章の流れ	1 ログ管理の概要 2 Syslog 3 logrotate
関連 URL	

27.1 ログ管理の概要

サーバーの運用状況や、サーバーへのアクセス状況はシステムのログに記録されます。ほとんどのサービスにおいて、ログの記録は syslogd によって行われます。

また、ログファイルに追加されるファイルは、そのままにしておくとディスクスペースを占有することになりますが、logrotate コマンドを使用して、ログファイルをローテーションすることが可能です。

本章では、**syslogd** によるログ採取と、**logrotate** によるログ管理の設定等について説明します。

27.2 syslog

27.2.1 syslog の概要

syslog とは、システムロギングとカーネルメッセージの確保という 2 つの機能を提供するユーティリティです。情報は、ローカルに記録することも、リモートのログサーバーに記録することも可能です。

27.2.2 syslog の起動と停止

syslog サーバーを使用するには、syslog の実体であるデーモンプログラム **syslogd** を起動する必要があります。syslogd の起動/停止スクリプトは、**/etc/rc.d/init.d/syslog** です。起動スクリプトのオプションでは、起動(start)、停止(stop)、再起動 (restart)、現在の状況を確認(status)を指定できます。

syslog の設定を変更した場合は、変更内容をシステムに反映させるために syslog を再起動してください。

- syslogd を起動するには次のコマンドを実行します。

```
# /sbin/service syslog start
```

- syslogd を停止するには、次のコマンドを実行します。

```
# /sbin/service syslog stop
```

- syslogd を再起動するには、次のコマンドを実行します。

```
# /sbin/service syslog restart
```

- syslogd の現在の状況を確認するには、次のコマンドを実行します。

```
# /sbin/service syslog status
```

27.2.3 syslog の設定

syslog の動作は、**/etc/syslog.conf** で設定します。**syslog.conf** には、ログレベルとそのログの出力先を、以下のような書式で記述します。

```
[ファシリティ].[プライオリティ] [出力先]
```

➢ ファシリティ

ファシリティには、ログを生成するプログラムの種類を設定します。
ファシリティに使える項目は、表 27-1 のとおりです。

表 27-1 syslog 設定ファイルのファシリティ項目

ファシリティ	説明
auth	認証
authpriv	ローカル認証
cron	cron
daemon	デーモン
kern	カーネル
lpr	印刷
mail	メール
news	ニュース
syslog	syslog
user	user
uucp	uucp
local0～local7	ローカルなファシリティ

➢ プライオリティ

プライオリティには、ログの重要度を設定します。
プライオリティに使える項目は、表 27-2 のとおりです。

表 27-2 syslog 設定ファイルのプライオリティ項目

プライオリティ	説明
none	メッセージを無視
debug	デバッグメッセージ
info	一般的なメッセージ
notice	注意を要するメッセージ
warning	警告メッセージ
err	一般的な障害メッセージ
crit	危機的な状況
alert	即時に対処を要するメッセージ
emerg	システムパニック

syslog 以外にも各アプリケーションがログを生成しますが、ほとんどのログは**/var/log** 以下に保存されます。
生成されたログは、ファイルに出力する以外にも、ログインユーザーの端末にメッセージとして出力したり、他のホストの syslogd に転送したりできます。

- priority が info 以上のすべてのログメッセージを**/var/log/messages** に出力するには、次のように設定します。

```
*.info /var/log/messages
```

- kernel 関係のログすべてを、ホスト miracle の syslog に転送するには、以下の手順で設定します。

1) 転送元の syslog.conf には次のように記述します。

```
kern.* @miracle
```

2) 転送先のホスト側でも、リモートからのログの受け入れを可能にしておく必要があります。
ホスト miracle が Asianux Server 3 の場合は、/etc/sysconfig/syslog の SYSLOGD_OPTIONS 変数に、-r を追加し、syslogd を再起動してください。

```
SYSLOGD_OPTIONS="-m 0 -r"
```

```
# service syslog restart
```

syslog.conf の詳細な設定方法については、「**man 5 syslog.conf**」を参照してください。
デフォルトの設定では、**/var/log/secure** にユーザー認証の情報が記録されているので、不正にログインを試みようとした形跡がないか、ログに注意を払いましょう。

27.3 logrotate

27.3.1 logrotate の概要

ログファイルにはログが次々に追加されます。特にサーバーではさまざまなサービスが提供されており、多数のアクセスがあるため、そのままにしておくとログファイルが大きなディスクスペースを占有することになります。

logrotate は、ログファイルを定期的にバックアップ/削除して世代管理を行うコマンドです。

システム管理者は logrotate コマンドを使用して、ログファイルを適切にローテーションする必要があります。

27.3.2 logrotate の設定

logrotate は、定期的に cron（/etc/cron.daily/logrotate）から実行され、その設定は、
/etc/logrotate.conf と **/etc/logrotate.d** 配下のサービスごとの設定ファイルで行います。

➢ /etc/logrotate.conf

全体の設定は、logrotate.conf で行います。

次の設定は、デフォルトの logrotate.conf の内容です。

```
weekly
rotate 4
create
#compress
include /etc/logrotate.d
/var/log/wtmp {
monthly
create 0664 root utmp
rotate 1
}
```

- **weekly**…毎週 1 回、ログのローテーションを行うことを指定しています。同様な設定に、daily、monthly などがあります。
- **rotate 4**…ログのローテーションに 4 つのログファイルを使用することを指定しています。また、**size** を指定することで、1 つのログファイルのサイズを指定することもできます。より詳細な設定方法については、**logrotate**(8)を参照してください。
- **create**…ローテーション実行後に、新たに同名のログファイルを作成することを指定しています。
- **compress**…ローテーションしたファイルを圧縮するかどうかを指定します。デフォルトではコメントアウトされており、圧縮はしません。
- **include**…設定ファイルが置かれているパスを記述します。デフォルトでは、/etc/logrotate.d ディレクトリ配下にサービスごとの設定ファイルが置かれています。
- **/var/log/wtmp** 以降…個別のログファイル名と、それに対する設定を記述しています。
  個別のログファイルに対する設定は、/etc/logrotate.d 以下に設定ファイルを置き、**include** で読み込みますが、logrotate.conf に直接設定を書くこともできます。

➢ /etc/logrotate.d

/etc/logrotate.d ディレクトリには、サービスごとの設定ファイルが置かれています。

次の例は、httpd サービスの設定ファイルの内容です。

```
/var/log/httpd/*log {
    missingok
    notifempty
    sharedscripts
    postrotate
        /bin/kill -HUP `cat /var/run/httpd.pid 2>/dev/null` 2> /dev/null || true
    endscript
}
```

- **missingok**…ログファイルが見つからなくても、エラーを出力せずに次の処理を続けます。
- **notifempty**…ログファイルが空の場合はローテーションしません。
- **sharedscripts**…ローテーションするファイルが複数ある場合でも、prescript や postscript のスクリプトを 1 回しか実行しません。
- **postrotate～endscript**…ローテーション後に実行するコマンドを、postrotate と endscript の間に記述します。

logrotate の詳細な設定方法については、「**man 8 logrotate**」を参照してください。

第28章　SSH

この章で説明する内容

目的	シェルの機能をリモートから使用する方法について理解する
機能	通信路の暗号化による安全性の向上 ホストの認証
必要な RPM	openssh —— SSH 本体 openssh-clients —— SSH サーバープログラム openssh-serve —— SSH クライアントプログラム openssh-ask-pass —— X11 パスフレーズダイアログ
設定ファイル	/etc/ssh/ssh_config /etc/ssh/sshd_config
章の流れ	1 SSH の概要 2 SSH の起動と停止 3 SSH の設定 4 SSH の利用
関連 URL	**OpenSSH**　http://www.openssh.com/

28.1 SSH の概要

Unix の世界では長い間、リモートログインには Telnet が使用されてきました。しかしインターネットが普及してくるにつれて、以下のような 2 つの問題点が指摘されるようになりました。

(1) **通信内容の暗号化ができない**

TELNET では通信路上を流れるデータは何の加工もせずに送られます。このためログインパスワードといった他人に対して秘密にしたい情報も、簡単に盗み見られてしまう可能性があります。

(2) **通信の相手が本物かどうかを確認できない**

通常 Unix では、ユーザーがシステムにログインしようとすると、パスワードの入力を求められます。これで確認できるのは「システムからみてユーザーが本物かどうか」という点だけです。ユーザーからみて「システムが本物かどうか」という点に関しては、確認できません。このような場合でも使用するシステムが目の前にあるうちは問題はおこりません。なぜならば、ユーザーは目でそのシステムの物理的存在を確かめることにより、暗黙のうちに「システムが本物であること」を確認しているからです。しかし、ネットワークを経由してシステムを利用する場合には、ユーザーは目視による確認ができません。画面上に「Login:」という文字列が表示されていても、それがユーザー自身の目標としている本当のシステムかどうかを確認できません。だれか悪意のある第三者が、そのシステムになりすました偽のシステムを用意して、あなたの秘密情報、たとえばパスワードを盗もうとしているのかもしれません。

このような問題を解決する目的で開発されたのが **SSH**(Secure Shell)です。SSH を利用することによって、通信の内容が暗号化されて、また確実に相手先ホストが本物であることが確認できるようになります。

現在のところ、SSH には **SSH1** と **SSH2** という、互換性のない 2 つのプロトコルが存在しています。

➢ SSH1

SSH1 プロトコルでは RSA 暗号が使用されます。

サーバーは、**ホスト鍵**(1024 ビット)と**サーバー鍵**(768 ビット)の 2 種類の RSA 鍵ペア(秘密鍵と公開鍵の組み)を持ちます。ホスト鍵は、初めて SSH サーバーが起動するときに生成されて、ファイルに保存されます。一度生成されたこの鍵は、ほぼ半永久的に同じものが使用されます。サーバー鍵は SSH サーバーが起動するたびに生成され、さらに一定時間経過すると古いものが破棄されて新しいものが生成されます。

SSH のセッションは次の手順で開始されます(図 28-1)。

(1) SSH クライアントから、セッションを開始したいというリクエストが SSH サーバーへ送られます。

(2) リクエストを受け取った SSH サーバーは RSA ホスト公開鍵と RSA サーバー公開鍵の両方をクライアントへ送ります。

(3) 鍵を受け取ったクライアントは、あらかじめ何らかの手段で入手しておいたホスト公開鍵と、通信路経由で送られてきたホスト公開鍵が一致しているかどうかをチェックします。
このとき、これらが一致しなければ、このホストは偽者であるか、あるいは何らかの理由でホストの鍵が変更されたことになります。SSH クライアントは「鍵が一致しない」という警告をユーザーに対して発して、ユーザーがその原因を調査することになります。事前に相手ホストの公開鍵が入手できていない場合は、SSH クライアントは「送られてきた公開鍵を本物とみなし、今後もその鍵を使用するか」という質問をユーザーに対して発して、確認を求めます。了解が得られれば、そのホスト公開鍵を今後目的のホストの公開鍵として、クライアント側のデータベースに登録します。

(4) クライアントは、自身の内部で 256 ビットの乱数を生成します。

(5) これが今後 SSH セッションで使用されるセッション鍵となります。同時に暗号化方式についても 3DES か Blowfish のどちらかが選択されます。クライアントは生成されたセッション鍵を RSA ホスト鍵で暗号化し、さらにその結果を RSA サーバー鍵で暗号化してホストに送ります。

(6) SSH サーバーは、受け取った暗号化データに対し、自分が持つ RSA ホスト秘密鍵と RSA サーバー秘密鍵を用いて復号処理を行い、セッション鍵を取り出します。
これで第 3 者には秘密を保ったまま、サーバーとクライアントでセッション鍵を共有できるようになります。以降のすべての通信はこのセッション鍵を利用した暗号で行われます。ここから後は通常の Unix のログインプロセスと同様で、サーバーからユーザーに対してパスワードの入力が求められ、ユーザーが正しいパスワードを入力すればシェルに制御が渡されます。

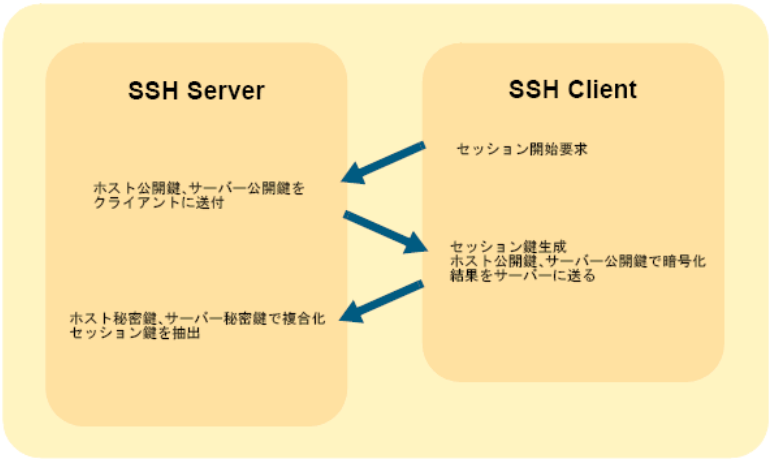


図 28-1 SSH の動作

➢ **SSH2**

SSH2 プロトコルも SSH1 プロトコルとほぼ同様に機能します。違いは、セッション鍵の交換のために RSA ではなく Diffie-Hellman を用いること、さらに使用可能である対称暗号の種類が異なることが挙げられます。またセッションの完全性チェックには SHA1、MD5 などの使用が可能となっています。

28.2 SSH の起動と停止

SSH を手動で起動/停止させる場合は、**service** コマンドで **sshd** を起動してください。起動スクリプトのオプションでは、起動(start)、停止(stop)、再起動(restart)、現在の状況を確認(status)を指定できます。

SSH の設定を変更した場合は、変更を反映するために SSH を再起動する必要があります。

- SSH サーバーを起動するには、次のコマンドを実行します。

```
# /sbin/service sshd start
```

- SSH サーバーを停止するには、次のコマンドを実行します。

```
# /sbin/service sshd stop
```

- SSH サーバーを再起動するには、次のコマンドを実行します。

```
# /sbin/service sshd restart
```

- SSH の現在の状況を確認するには、次のコマンドを実行します。

```
# /sbin/service sshd status
```

- システムが起動したときに自動的に SSH を起動するように設定するには、次のコマンドを実行します。

```
# /sbin/chkconfig sshd on
```

- システムが起動したときに自動的には SSH が起動しないように設定するには、次のコマンドを実行します。

```
# /sbin/chkconfig sshd off
```

28.3 SSH の設定

SSH の設定は**/etc/ssh/sshd_config** の中に記述されています。この内容を変更した場合は、変更内容を反映させるために SSH を再起動してください。

Asianux Server 3 の出荷時の設定では、安全のために、リモートからの root のログインが禁止されています。これを許可するようにしたい場合は**/etc/ssh/sshd_config** を以下のように変更してください。

- 変更前

```
PermitRootLogin   no
```

- 変更後

```
PermitRootLogin   yes
```

28.4 SSH の利用

28.4.1 SSH でリモートホストにログインする

SSH でリモートホストにログインするためには以下のコマンドを実行します。

```
/usr/bin/ssh [ユーザー名@]ホスト名
```

対象となるホストに初めてアクセスした場合は、相手のホストの公開鍵を記録しておくかどうかたずねられるので、yes と答えてください。この内容はホームディレクトリの下の、**.ssh/known_hosts** に保存されます。次にパスワードを聞かれるので、リモートホストでのユーザーパスワードを入力するとログイン完了となります。

ユーザー名 foo がリモートホスト host1 にログインする例を次に示します。

```
The authenticity of host 'host1 (XXX.XXX.XXX.XXX)' can't be established.
RSA key fingerprint is XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'host1' (RSA) to the list of known hosts.
foo@host1's password:
```

28.4.2 パスワードを入力せずにログインする

SSH では認証の強化と安全性の確保のため、パスワードに頼らない認証（公開鍵認証）を可能にしています。28.4.1 の例では、クライアント側からみたサーバーの認証には公開鍵暗号が使われていますが、サーバー側からみたユーザーの認証には、伝統的なパスワードが使われています。

ローカルマシンであらかじめ自分の公開鍵を生成しておいて、それをリモートマシンに登録しておくと、毎回パスワードを入力しなくてもリモートログインできるようになります。この機能を実現するための手順を次に示します。ただし、この設定は非常に危険です。他人がクライアントマシンにアクセスしないように十分注意をしてください。

（1）ローカルマシンで公開鍵と秘密鍵の対の生成を行う

```
$ /usr/bin/ssh-keygen -t dsa
```

上記のコマンドを実行すると、鍵ファイルを保存する場所をたずねられるので、何も入力せずに[Enter]キーを押してください。次にパスフレーズを聞かれますので、このときも何も入力せずに[Enter]キーを押してください。

```
Generating public/private dsa key pair.
Enter file in which to save the key (/home/foo/.ssh/id_dsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/foo/.ssh/id_dsa.
Your public key has been saved in /home/foo/.ssh/id_dsa.pub.
The key fingerprint is:
XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX foo@secific.domain.name
```

するとホームディレクトリの下の **.ssh** というディレクトリの下に、次の 2 つのファイルが生成されます。

- **id_dsa** ———— 秘密鍵
- **id_dsa.pub** —— 公開鍵

パスフレーズを設定すると、SSH を起動するたびにパスフレーズの入力を求められます。パスフレーズを設定していない場合には、秘密鍵の管理を厳重に行い、決して他人に盗み出されないように注意してください。この内容を他人に知られると、その人があなたになりすましてリモートホストにログインできるようになってしまいます。

ssh-keygen コマンドでクライアントの鍵ペアを生成するときにパスフレーズを使うと、SSH クライアントを起動したときにパスフレーズの入力を求められます。これはクライアント側の秘密鍵がこのパスフレーズで暗号化されていて、これを復号化して生の秘密鍵を取り出す必要があるからです。したがって、このパスフレーズと SSH サーバー、クライアント間の認証とは直接の関係はありません。

(2)サーバーに自分の公開鍵を登録する

サーバーにユーザー名 foo でログインする場合は、サーバー上の**/home/foo/.ssh/authorized_keys** ファイルの最後に、生成された **id_dsa.pub** の内容をテキストエディタで追加してください。これにより、SSH サーバーが SSH クライアントを認証できるようになり、リモートログインの際にパスワードの入力をする必要がなくなります。

28.4.3 ssh-agent の利用

SSH キーの生成時には、通常パスフレーズを入力してください。前項で説明したパスフレーズの省略はセキュリティの観点からも推奨されません。ただし、SSH によるログインのたびにパスフレーズを入力するのは手間がかかります。SSH には、これを代替して行ってくれる **ssh-agent** というツールが付属しています。

ssh-agent を利用すると、わずらわしいパスフレーズの入力は最初の 1 回に省略できます。**ssh-agent** を利用するには、次のようにします。

```
$ eval `ssh-agent -s`
Agent pid 4530
$ /usr/bin/ssh-add ~/.ssh/id_dsa
Enter passphrase for /home/foo/.ssh/id_dsa:
Identity added: /home/foo/.ssh/id_dsa (/home/foo/.ssh/id_dsa)
```

上記の一連のコマンドを実行したシェル環境からは、以降の SSH の接続を行う際のパスフレーズ入力は **ssh-agent** が代替して行うこととなり、パスフレーズ入力を省略できます。

Asianux Server 3 では、X Window の起動時に自動的に **ssh-agent** を起動します。

ターミナルを開き、以下のコマンドにより **ssh-agent** が起動していることを確認してください。

```
$ /bin/env | /bin/grep SSH
SSH_AGENT_PID=738
SSH_AUTH_SOCK=/tmp/ssh-EWpvC721/agent.721
SSH_ASKPASS=/usr/libexec/openssh/gnome-ssh-askpass
```

この状態で、次のように実行することで、利用している X Window 環境からの SSH 接続について **ssh-agent** がすべてパスフレーズ入力を代行します。

```
$ /usr/bin/ssh-add ~/.ssh/id_dsa
```

28.4.4　Windows からの SSH の使用

Microsoft 社の Windows 9x/Me/NT/2000/XP は、標準で Telnet クライアントしか同梱されていません。SSH を使用するには、SSH クライアントを別途インストールする必要があります。ここでは、フリーソフトの SSH クライアント **PuTTY** を用いて、Asianux Server 3 へ SSH で接続する方法を紹介します。

PuTTY は、http://www.chiark.greenend.org.uk/~sgtatham/putty/ から入手できます。ダウンロードしたアーカイブを展開して、PuTTY をインストールして、起動します[2]。

PuTTY を起動すると図 28-2 のようなウィンドウが出現します。ログイン先のホスト名を入力し、SSH を指定した後、[Open]ボタンをクリックしてください。

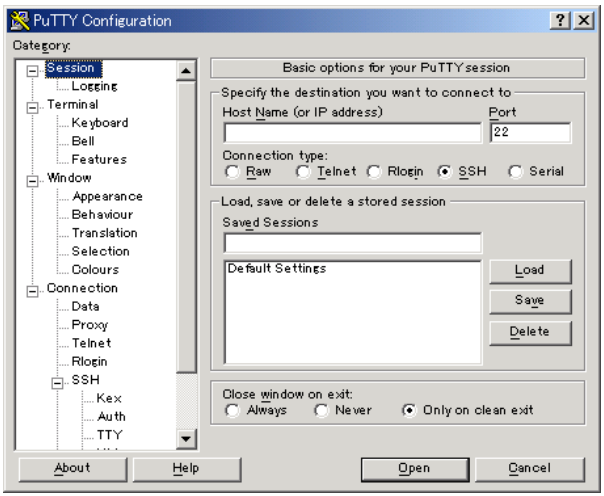


図 28-2　PuTTY の設定画面

対象となるホストに初めてログインする場合、図 28-3 のようなダイアログが出現するので[はい(Y)]をクリックしてください。

2　PuTTY は日本語の表示が正しく行えません。非公式の日本語用パッチを次の場所から入手できます。
http://hp.vector.co.jp/authors/VA024651/

図 28-3 PuTTY のセキュリティ警告画面

この後は、アカウント名およびパスワードを正しく入力することでシステムにログインできます。

また、**PuTTYGEN** を使って秘密鍵と公開鍵の対を作成し、登録することで、公開鍵を使用することも可能です。PuTTYGEN は PuTTY のダウンロードページから入手できます。

http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

PuTTYGEN を起動したら「Generate」ボタンをクリックし、ウィンドウに現れたプログレスバーが 100%になるまでウィンドウ内でマウスカーソルをただ動かします。しばらく動かしていると、公開鍵と秘密鍵が生成されます。「Key passphrase」にパスワードを入力し、「Confirm passphrase」に繰り返しパスワードを入力したら、「Save private key」ボタンをクリックして秘密鍵を保存します。公開鍵は、「Public key for pasting into OpenSSH authorized_keys file」に書かれている内容をコピーし、テキストファイルに書き込みます。

公開鍵のファイルの内容を、サーバーの**`/home/foo/.ssh/authorized_keys`** ファイルの最後に書き込みます。

秘密鍵ファイルは、PuTTY の設定画面（図 28-2）左側のリストの「SSH」－「Auth」にある「Private key file authentication」にファイルの場所を指定します。

以上で、Windows から公開鍵を利用した SSH 接続を行うことができます。

28.4.5 Windows からの SCP の使用

Windows で使える SCP プログラム PSCP は、PuTTY のダウンロードページから入手できます。

http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

ダウンロードした pscp.exe をパスの通っている場所（C:\Windows\Command など）にコピーしたら、MS-DOS プロンプト（コマンドプロンプト）を起動します。

- Windows から Linux へファイルをコピーするには次のコマンドを実行します。

```
pscp filename username@linuxserver:ディレクトリ名
```

- Linux から Windows へファイルをコピーするには次のコマンドを実行します。

```
pscp username@linuxserver:filename ディレクトリ名
```

- ディレクトリごとコピーする場合は**-r** オプションを指定します。

```
pscp -r username@linuxserver:ディレクトリ名 ディレクトリ名
```

たとえば、Linux マシン **asianux** の**/home/user1** ディレクトリに **test.txt** があり、Windows マシンの **c:\data** ディレクトリに **sample.txt** があるとします。Linux のユーザー名が **user1** だとすると、次のようにして PSCP を使ってファイルをコピーできます。

- Linux マシンにある test.txt を Windows マシンにコピーするには、次のように pscp コマンドを実行します。

```
C:\data>pscp user1@asianux:/home/user1/test.txt c:\data
```

あるいは、次のように実行します。

```
C:\data>pscp user1@asianux:test.txt C:\data
```

- Windows マシンにある sample.txt を Linux マシンにコピーするには、次のように実行します。

```
C:\data>pscp C:\data\sample.txt user1@asianux:/home/user1
```

あるいは次のように実行します。

```
C:\data>pscp C:\data\sample.txt user1@asianux:
```

第29章　時刻同期

この章で説明する内容

目的	NTP サーバーの構築方法について理解する
機能	インターネット上の標準時サーバーとの時刻同期
必要な RPM	ntp――NTP 本体
設定ファイル	/etc/ntp.conf /etc/ntp/step-tickers /etc/sysconfig/ntpd
章の流れ	1 NTP サーバーの概要 2 NTP サーバーの設定 3 NTP サーバーの起動と停止 4 NTP サーバーのテスト
関連 URL	**NTP HOWTO** http://www.linux.or.jp/JF/JFdocs/TimePrecision-HOWTO/ntp.html

29.1　NTP サーバーの概要

ネットワーク内に NTP（Network Time Protocol）サーバー（タイムサーバー）を設置することにより、ネットワーク内の時刻をすべて同期することが可能となります。

本章では、外部の上位 NTP サーバーへ時刻同期を取る NTP サーバーを構築する方法について説明します。

29.2　NTP サーバーの設定

まず、ネットワークの時刻合わせに使う、外部の上位サーバーを決定する必要があります。

http://support.ntp.org/bin/view/Servers/WebHome から、stratum 1 のサーバーを選択します。Stratum（階層）は 15 階層まで存在し、上位サーバーの方がより精度が高いと言われます。しかし実際には、上位サーバーは、負荷が集中して応答時間がかかることで時刻の精度も落ちる可能性があるので、一概に上位サーバーならば精度が高いとは限りません。精度の面からも見ても、近い地域の NTP サーバーを指定するのがいいでしょう。

NTP サーバーとして動作させるには、**/etc/ntp.conf** を次のように設定します。

```
restrict default ignore
restrict 127.0.0.1
restrict 192.168.1.0 mask 255.255.255.0 nomodify notrap
restrict 上位サーバーIP noquery nomodify notrap
restrict 上位サーバーIP noquery nomodify notrap
server 上位サーバーIP
server 上位サーバーIP
driftfile /var/lib/ntp/drift
```

- 1 行目――すべての ntp 通信を無視。
- 2 行目――local host のみ許可。
- 3 行目――LAN 内の通信を許可（使用するネットワークに合わせて設定）。
- 4、5 行目――決定した上位サーバーの IP アドレス（上位サーバーのアクセス許可）。
- 6、7 行目――決定した上位サーバーの IP アドレス。
- 8 行目――誤差調整用ファイル（絶対パスで指定）。あらかじめ作成しておく必要があります。

29.3 NTP サーバー起動時のオプション

/etc/sysconfig/ntpd ファイルは **ntpd** デーモンが起動する時に必要な動作設定を記載します。

初期状態では次の行が含まれています。

```
# Drop root to id 'ntp:ntp' by default.
OPTIONS=" -u ntp:ntp -p /var/run/ntpd.pid"

# Set to 'yes' to sync hw clock after successful ntpdate
SYNC_HWCLOCK=no
```

Slew※モードで ntpd を動作させたい場合は OPTIONS 行に”-x”を追加し以下のように設定し、ntpd デーモンを再起動してください。

```
OPTIONS=" -x -u ntp:ntp -p /var/run/ntpd.pid"
```

※Slew モードとは時刻の修正を一回で修正せずに、1 秒に 0.0005 秒 ずつ何度かに分けて、緩やかに修正する動作モードです。このモードを常に利用すると大幅に時刻がずれていた場合に、正常な時刻になるのに非常に時間を要するのでご注意ください。初期状態では 0.128 秒以下の修正が必要な場合は slew モードで動作し、0.128 秒以上の修正が必要な場合は 1 回で時刻を修正する動作モードとなっています。

上位の ntp サーバーと時刻を同期が成功したあとに、サーバー自身のハードウェア時計も同時に修正する場合は、”SYNC_HWCLOCK=yes”の行のコメント(#)をはずして、ntpd デーモンを再起動してください。

```
SYNC_HWCLOCK=yes
```

/etc/ntp/step-tickers ファイルには ntpd デーモン起動直前にあらかじめ時間を合わせるべきサーバーを記述します。このファイルにサーバー名を記載しなくても ntp.conf から server エントリーを探し自動的に時刻を合わせますが、明示的に指定したい場合はサーバー名または IP アドレスを指定してください。

29.4 NTP サーバーの起動と停止

NTP サーバーを使用するには、NTP の実体であるデーモン **ntpd** を起動する必要があります。**ntpd** の起動／停止スクリプトは **/etc/rc.d/init.d/ntpd** となっています。起動スクリプトのオプションでは、起動（start）、停止（stop）、再起動（restart）、現在の状況を確認（status）を指定できます。

- NTP サーバーを起動するには、次のコマンドを実行します。

```
# /sbin/service ntpd start
```

- NTP サーバーを停止するには、次のコマンドを実行します。

```
# /sbin/service ntpd stop
```

- NTP サーバーを再起動するには、次のコマンドを実行します。

```
# /sbin/service ntpd restart
```

- NTP サーバーの現在の状況を確認するには、次のコマンドを実行します。

```
# /sbin/service ntpd status
```

29.5 NTP サーバーのテスト

（1）まず、現時点でのシステムの時刻を表示します。

```
# /bin/date
2008年  8月 13日 水曜日 16:30:48 JST
```

（2）故意にシステムの時刻をずらします。

```
# /bin/date -s "2007/08/01 00:00:00"
2008年  8月  1日 金曜日 00:00:00 JST
```

（3）ntpd を起動します。

```
# /sbin/service ntpd start
ntpdを起動中:                                              [  OK  ]
```

（4）サーバーの動作確認をします。

```
# /usr/sbin/ntpq -pn
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
 XXX.XXX.XXX.XXX .STEP.          16 u   40   64    0    0.000    0.000 4000.00
 XX1.XX1.XX1.XX1 .STEP.          16 u  100   64    0    0.000    0.000 4000.00
```

st（stratum）の値は 15 までなので、16 ということはまだ同期されていないことを示します。この状況が 5 分以上続くようであれば **/var/log/messages** やもう一度設定ファイル（**/etc/ntp.conf**）を確認してください。

通常はしばらくすると下記のように st が実際に使用できる数値へ変わり、徐々に同期が行われていきます。

```
# /usr/sbin/ntpq -pn
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
 XXX.XXX.XXX.XXX .GPS.            1 u    1   64    3   65.832   -1.960   0.809
 XX1.XX1.XX1.XX1 .GPS.            1 u    4   64    3   63.948   -2.286   0.487
```

さらにしばらくすると、次のような表示になります。

```
# /usr/sbin/ntpq -pn
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
+XXX.XXX.XXX.XXX .GPS.            1 u   17   64   77   64.345   -6.797   3.167
*XX1.XX1.XX1.XX1 .GPS.            1 u   24   64   77   63.948   -2.286   3.682
```

一番左の「*」が現在の同期中のサーバーを示し、「+」はいつでも同期可能なサーバーを示します。

ここで再度システムの時刻を表示すると、現在の時間に戻っています。

```
# /bin/date
2007年  8月 13日 水曜日 16:46:29 JST
```

(5)テストが成功したら、NTP サーバーを自動起動させるように設定しておきます。

```
# /sbin/chkconfig ntpd on
```

この設定により、再起動後も **ntpd** は自動的に起動します。

NTP サーバーの設定が完了したら、各クライアントはこの NTP サーバーをタイムサーバーとしてシステム全体の時刻の同期が取れることとなります。

第30章　ジョブスケジューラー

この章で説明する内容

目的	ジョブを自動化する方法について理解する
機能	指定した時間や日付にジョブを自動実行
必要な RPM	crontabs —— 設定ファイル vixie-cron —— cron デーモン at —— at コマンド用 anacron——anacron lime-cron —— GUI 設定用
設定ファイル	/var/spool/cron/ /etc/crontab /etc/cron.hourly /etc/cron.daily /etc/cron.monthly /etc/cron.weekly /etc/cron.d /var/spool/at/ /etc/anacrontab
章の流れ	1 ジョブスケジューラーの概要 2 cron 3 at 4 anacron 5 タスクスケジューラ
関連 URL	**PATH HOWTO: cron、at コマンドについて** http://www.linux.or.jp/JF/JFdocs/Path-10.html

30.1 ジョブスケジューラーの概要

Linux のスケジューラーは、指定された日付や時刻に自動的にジョブを実行するような設定が可能です。システム管理者はタスクの自動化によって、定期的にバックアップを実行するなどが可能になります。

本章では、ジョブを特定の日時で繰り返し実行させる **cron** と一度だけ特定の日時に実行させる **at**、および前回の実行からの日数で実行する **anacron** について説明します。

30.2 cron

本節では繰り返しジョブを実行させる cron について説明します。

30.2.1 cron デーモンの起動と停止

cron を使用するには、cron の実体であるデーモン **crond** を起動する必要があります。

crond の起動／停止スクリプトは **/etc/rc.d/init.d/crond** となっています。起動スクリプトのオプションでは、起動（start）、停止（stop）、再起動（restart）、現在の状況を確認（status）を指定できます。

- crond を起動するには、次のコマンドを実行します。

```
# /sbin/service crond start
```

- crond を停止するには、次のコマンドを実行します。

```
# /sbin/service crond stop
```

- crond を再起動するには、次のコマンドを実行します。

```
# /sbin/service crond restart
```

- crond の現在の状況を確認するには、次のコマンドを実行します。

```
# /sbin/service crond status
```

30.2.2 cron の設定ファイル

cron の設定ファイルには、次に示すものがあります。crond は毎分これらの設定ファイルに変更がないかをチェックして、変更があった場合には変更を反映させて実行します。

- /var/spool/cron/ユーザー名
- /etc/crontab
- /etc/cron.d
- /etc/cron.hourly/
- /etc/cron.daily/
- /etc/cron.weekly/
- /etc/cron.monthly/

➢ /var/spool/cron/ユーザー名

各ユーザーの設定ファイルです。このファイルを作成、削除、閲覧するには **crontab** コマンドを使用します。

```
crontab [-u user] {-e|-l|-r}
```

-u user——user で指定したユーザーの crontab ファイルを操作の対象とします。なお、このオプションは、root ユーザーのみ使用できます。

-e——crontab を対話的に編集します。通常は、vi エディタが起動して、設定ファイルを編集します。

-l——crontab ファイルの内容を表示します。

-r——crontab ファイルを削除します。

このファイルの構文は、「分（0～59）」、「時（0～23）」、「日（1～31）」、「月（1～12）」、「曜日（0～6）」、「コマンド」の 6 つのフィールドで構成されます。コマンドフィールド以外では、「,」、「－」、「/」などの記号が使えます。
たとえば、9 時、11 時、13 時、15 時、17 時のそれぞれ 0 分、15 分、30 分、45 分に XXX というコマンドを実行する設定は、次のように記述します。

```
0,15,30,45 9-17/2 * * * XXX
```

➢ /etc/crontab

通常、このファイルには、「cron.monthly」、「cron.weekly」、「cron.daily」、「cron.hourly」配下のファイルを指定時間ごとに実行する指示が記述されています。
構文は **crontab** コマンドの構文と似ていて、「曜日」と「コマンド」の間に「ユーザー」が入るだけです。

また、環境変数 **MAILTO** で指示されたメールアドレスに対して、実行結果をメールで送ります。MAILTO がない場合は、ファイルの所有者にメールが送られます。メールを受け取りたくない場合には、**MAILTO=''** または **MAILTO=""** と指定することで受け取らないように設定できます。

- **/etc/cron.d**

  cron タスクを毎時間、毎日、毎週、毎月以外の予定で実行する必要がある場合は、そのスクリプトをこのディレクトリに追加できます。このディレクトリ内のファイルの構文はすべて **/etc/crontab** と同じです。

- **/etc/cron.hourly,/etc/cron.daily,/etc/cron.weekly,/etc/cron.monthly**

  これらのディレクトリ配下のファイルは、**/etc/crontab** ファイルによって呼び出されて、指定された時間にジョブを実行します。ディレクトリ配下のファイルはいずれもシェルスクリプトである必要があります。

30.3 at

本節では、一度だけ指定した日時にジョブを実行させる **at** について説明します。

30.3.1 at デーモンの起動と停止

at を使用するには、at の実体であるデーモン **atd** を起動する必要があります。**atd** の起動／停止スクリプトは、**/etc/rc.d/init.d/atd** となっています。起動スクリプトのオプションでは、起動（start）、停止（stop）、再起動（restart）、現在の状況を確認（status）を指定できます。

- atd を起動するには、次のコマンドを実行します。

```
# /sbin/service atd start
```

- atd を停止するには、次のコマンドを実行します。

```
# /sbin/service atd stop
```

- atd を再起動するには、次のコマンドを実行します。

```
# /sbin/service atd restart
```

- atd の現在の状況を確認 するには、次のコマンドを実行します。

```
# /sbin/service atd status
```

30.3.2 atコマンドの使用方法

atコマンドは、以下の構文で実行させることができます。プロンプトが表示されるので、実行するコマンドを入力して、[Ctrl]＋[d]キーでプロンプトを抜けます。この場合、HH:MM に指定したコマンドが実行されます。

```
$ /usr/bin/at HH:MM
>
```

また、シェルスクリプトを用意して、次のようにすることで実行が可能となります。

```
$ /usr/bin/at HH:MM -f file
```

他にも時間の設定として、次のような設定が可能です。

- 実行した日より3 日後の午後 4 時に実行（minutes、hours、weeksも設定可能）

```
$ /usr/bin/at 4pm+3days
```

- 7 月 31 日の午前 10 時に実行

```
$ /usr/bin/at 10am Jul 31
```

- 明日の午前 1 時に実行（todayも設定可能）

```
$ /usr/bin/at 1am tomorrow
```

まだ実行されていないジョブを確認する際は **atq** コマンドを使用します。

```
# /usr/bin/atq
3  2007-08-13 17:00 a ユーザー名
```

まだ実行されていないジョブは**/var/spool/at/**配下に保存されます。

実行待ちのジョブを削除するには **atrm** コマンドを使用します。

```
# /usr/bin/atrm ジョブ番号
```

30.4 anacron

本節では時間間隔を指定して繰り返しジョブを実行させる anacron について説明します。

30.4.1 anacron

anacron を使用するには、anacron の実体であるデーモン **anacron** を起動する必要があります。

anacrond の起動／停止スクリプトは **/etc/rc.d/init.d/anacron** です。起動スクリプトのオプションでは、起動（start）、停止（stop）、再起動（restart）、現在の状況を確認（status）を指定できます。

- anacron を起動するには、次のコマンドを実行します。

```
# /sbin/service anacron start
```

- anacron を停止するには、次のコマンドを実行します。

```
# /sbin/service anacron stop
```

- anacron を再起動するには、次のコマンドを実行します。

```
# /sbin/service anacron restart
```

- anacron の状態を確認するには、次のコマンドを実行します。

```
# /sbin/service anacron status
```

30.4.2 anacrontab の設定ファイル

anacron に定期的にジョブを実行させるには/etc/anacrontab に次のように記載します。

```
間隔（日数）　ディレイ（分）　ジョブの名前　実行するコマンド
```

- **間隔(日数)**:前回実行時からの間隔を日数で指定します。
- **ディレイ(分)**:前回実行時から上の間隔日数実行されていない場合は、このディレイ(分)後にジョブを実行します。
- **ジョブの名前**:任意のジョブ名を一続きで指定します。スペースは利用できません。
- **実行するコマンド:**コマンドラインから実行できるコマンド文字列を指定します。

設定例として2日おきにディスク使用量調査を記録する場合は/etc/anacrontab に次のように記載します。

```
2       5       disk-check      "/bin/df >>/var/log/disk-check.log"
```

30.5 タスクスケジューラ

ここまで **cron** と **at** それぞれのコマンドや設定ファイルを解説してきましたが、Asianux Server 3 では、これらのコマンドでの設定をGUIツール**タスクスケジューラ**で設定できるようになりました。タスクスケジューラの本体である **lime-cron** を実行すると、図 30-1 のウィンドウが表示されます。

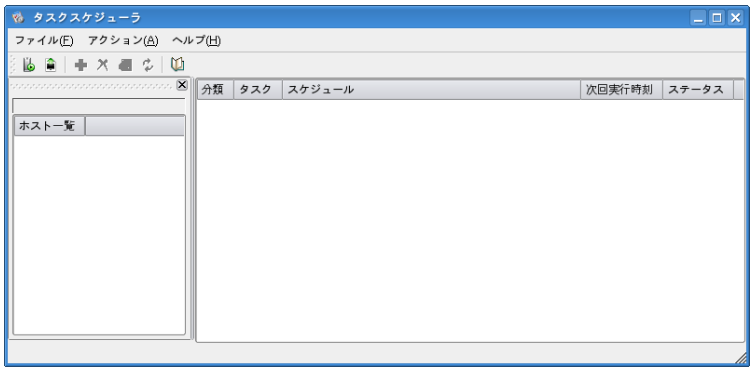


図 30-1 lime-cron のウィンドウ

スケジュールを管理するにはホストに接続する必要があります。メニューの[ファイル(F)]－[ホストを開く(O)]を選択し、[ローカルホスト(L)]を選択して[OK(O)]をクリックします。「CIMOM が起動していないようですが、今すぐ起動しますか？」と尋ねられた場合は、[はい[Y]]をクリックしてください。スケジュールの一覧が表示されます。

スケジュールを追加するには、メニューの[アクション(A)]－[新規タスク(N)]を選択し、表示されたウィザード画面にしたがって登録をします。

第31章　ZABBIX サーバーの構築

この章で説明する内容

目的	ZABBIX の構築方法について理解する
機能	統合監視環境の構築
必要な RPM	**ZABBIX のパッケージ** zabbix、zabbix-server、zabbix-server-mysql —— ZABBIX サーバー zabbix-web、zabbix-web-mysql —— ZABBIX Web フロントエンド zabbix-agent —— ZABBIX エージェント
設定ファイル	/etc/httpd/conf.d/zabbix.conf /etc/zabbix/zabbix.conf.php /etc/zabbix/zabbix_server.conf /etc/zabbix/zabbix_agentd.conf
章の流れ	1 ZABBIX の概要 2 ZABBIX サーバーの構築
関連 URL	**ZABBIX**　http://www.zabbix.com/ **ZABBIX-JP**　http://www.zabbix.jp/

31.1 ZABBIX の概要

ZABBIX とは、様々なアーキテクチャに対応したオープンソースの統合監視ソフトウェアで、サーバやネットワーク機器のリソース監視／設定を一元管理することができます。

ZABBIX は図 31-1 のように、サーバー、エージェント、フロントエンドから構成されます。

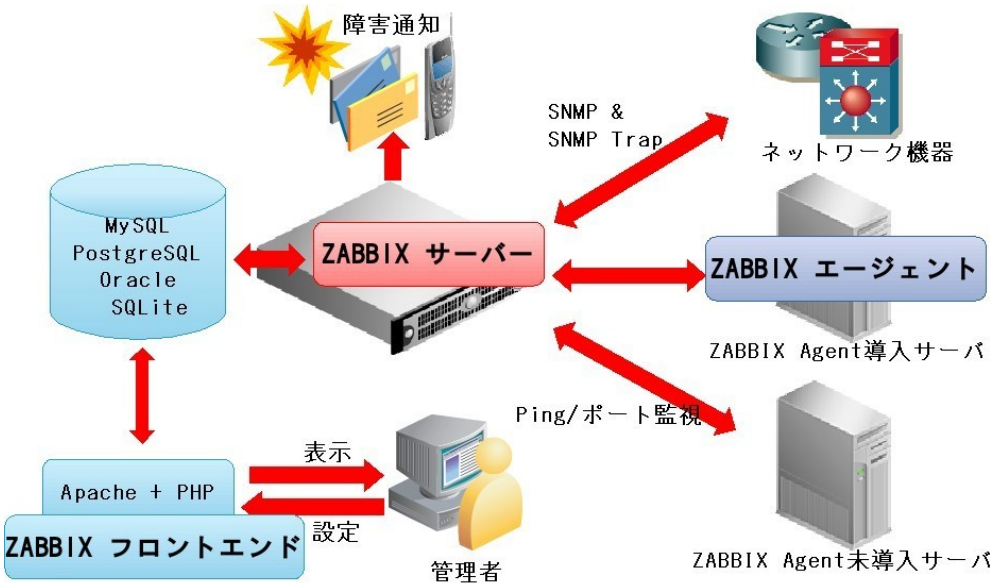


図 31-1 ZABBIX 動作環境

- **ZABBIX サーバー**
  ネットワーク機器や ZABBIX エージェント等から情報を収集し、データベースに監視情報を保存します。
- **ZABBIX エージェント**
  マシンの情報を収集し、サーバーへ監視情報を送ります。
- **ZABBIX フロントエンド**
  監視の設定と収集した情報の表示を行います。

主な特徴や監視項目、対応 OS は表 31-1 のとおりです。

表 31-1

主な特徴	主な監視項目
• オープンソース・ソフトウェア • サポートする OS が豊富 • 簡潔な Web インターフェースから設定、管理、情報表示 • 柔軟に設定可能な監視、障害検知、通知機能 • 簡単に作成できるグラフ、マップ機能 • SQL データベースによるデータ管理 • 高性能な専用エージェント • エージェントレスの監視にも対応 • SNMP v1/v2/v3 に対応 • 監視設定のテンプレート化	• CPU 負荷状況 • メモリ使用状況 • ディスク使用状況 • ネットワーク使用状況 • ポート監視 • ログ監視 • プロセス監視 • ファイル監視
対応 OS（エージェント）	
• AIX • HP-UX • Mac OS X • NetWare • FreeBSD • OpenBSD • Solaris • Tru64/OSF • SCO OpenServer • Windows • Linux	

31.2 ZABBIX サーバーの構築

ZABBIX は初期状態では無効になっていますので、使用する際にはいくつかの設定をする必要があります。データベースとして MySQL を利用する場合、大きく分けて MySQL、Apache、そして ZABBIX 自体の設定を行なっていきます。本章ではその 3 ステップについて説明します。

この作業は、MySQL サーバーと Apache サーバーが起動していることを前提としています。それぞれの起動手順は、第 12 章「MySQL データベースサーバーの構築」および第 20 章「ウェブサーバーの構築」を参照してください。

31.2.1 データベースの設定

ZABBIX ではデータの格納に MySQL データベースを利用します。MySQL データベースサーバは ZABBIX サーバ専用のものを構築し、バックエンド DB には InnoDB の利用を推奨します。以下に ZABBIX サーバ用の MySQL データベースサーバの設定手順を解説します。

(1) MySQL サーバの設定

MySQL サーバは自動で文字エンコード変換を行う機能を有していますが、接続するクライアントの設定や状態によっては適切な変換が行われずに文字化けを起こすことがあります。ZABBIX が利用するデータベースの文字エンコードをあらかじめ UTF-8 に固定し、文字コードの自動変換機能を無効にすることで文字コード関連の問題を事前に回避します。また、InnoDB 利用時に ibdata1 ファイルの肥大化を防ぐためテーブル単位でファイルを分割するように設定を行います。

/etc/my.cnf ファイルに以下の行を追加します。

```
[mysqld]
datadir=/opt/mysql
socket=/var/lib/mysql/mysql.sock
# Default to using old password format for compatibility with mysql 3.x
# clients (those using the mysqlclient10 compatibility package).
old_passwords=1

default-character-set=utf8              ←追加
skip-character-set-client-handshake     ←追加
innodb_file_per_table                   ←追加
```

MySQL データベースサーバを起動します。

```
# service mysqld start
```

(2)ZABBIX 用データベースの作成

以下の例では、データベース名 **zabbix**、ユーザ名 **zabbix**、パスワード **zabbix_password** として作成しています。

MySQL の root ユーザーのパスワードは初期状態では設定されていませんので、何も入力せずに Enter キーを押します。すでに MySQL の root ユーザーパスワードを設定済みの場合は、設定されているパスワードを入力します。

```
# /usr/bin/mysql -u root -p
Enter password: 'MySQLのrootユーザーのパスワード'
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 28 to server version: 5.0.22

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> create database zabbix default character set utf8;
mysql> grant all privileges on zabbix.* to zabbix@localhost identified by
'zabbix_password';
mysql> flush privileges;
mysql> quit
```

(3)ZABBIX 用データベースのデータ作成

作成したデータベース内にデータを作成するため、次のコマンドを実行します。

```
# cat /usr/share/doc/zabbix-server-1.6.4/schema/mysql.sql |mysql -u root -p
zabbix
Enter password: 'MySQLのrootユーザーのパスワード'
# cat /usr/share/doc/zabbix-server-1.6.4/data/data.sql |mysql -u root -p  zabbix
Enter password: 'MySQLのrootユーザーのパスワード'
# cat /usr/share/doc/zabbix-server-1.6.4/data/images_mysql.sql |mysql -u root -p
zabbix
Enter password: 'MySQLのrootユーザーのパスワード'
```

(4)zabbix_server.conf ファイルの設定

作成したデータベース内の以下項目を、**/etc/zabbix/zabbix_server.conf** に設定します。

```
# DBHost=localhost ← データベースがlocalhost以外の場合はコメントを外して設定
```

```
DBName=zabbix ← データベース名
DBUser=zabbix ← データベース接続ユーザ名
DBPassword=zabbix_password ← コメントを外し、データベース接続ユーザのパスワードを設定
# DBSocket=/tmp/mysql.sock ← MySQLの接続にソケットを使用する場合はコメントを外して設定
```

31.2.2 Apache サーバーの起動

ZABBIX の Apache 設定は、**/etc/httpd/conf.d/zabbix.conf** ファイルに存在します。初期状態では、コメントアウトされているので、コメントをはずし、さらに以下の行を <Directory "/usr/share/zabbix"> ディレクティブに追加します。

```
   php_value date.timezone Asia/Tokyo
   php_value memory_limit 64M
   php_value upload_max_filesize 10M
```

最終的に**/etc/httpd/conf.d/zabbix.conf** ファイルの内容は次のようになります。

```
#
# Zabbix monitoring system php web frontend
#

Alias /zabbix /usr/share/zabbix

<Directory "/usr/share/zabbix">
   Options FollowSymLinks
   AllowOverride None
   Order allow,deny
   Allow from all

   php_value max_execution_time 300
   php_value date.timezone Asia/Tokyo
   php_value memory_limit 64M
   php_value upload_max_filesize 10M
</Directory>

<Directory "/usr/share/zabbix/include">
   Order deny,allow
   Deny from all
   <files *.php>
      Order deny,allow
```

```
      Deny from all
   </files>
</Directory>

<Directory "/usr/share/zabbix/include/classes">
   Order deny,allow
   Deny from all
   <files *.php>
      Order deny,allow
      Deny from all
   </files>
</Directory>
```

設定後 httpd サービスを再起動します。

```
# service httpd restart
httpd を停止中:                                            [  OK  ]
httpd を起動中:                                            [  OK  ]
```

31.2.3 ZABBIX のサーバープロセスの起動

zabbix サービスを起動します。

サーバー再起動の際に、自動的に zabbix サービスを起動する場合は chkconfig コマンドで有効にします。

```
# service zabbix-server start
Starting zabbix server:                                    [  OK  ]
# chkconfig zabbix-server on
```

31.2.4　ZABBIX の Web フロントエンドの設定

ZABBIX の監視設定を WEB から行なうための Web フロントエンドを設定します。

(1)ZABBIX の設定ファイルの削除とディレクトリへの書き込み権限付与

ZABBIX Web フロントエンドからインストーラを利用して設定を行うため、Web フロントエンドの設定ファイル/etc/zabbix/zabbix.conf.php の設定を一度削除し、設定ファイルのディレクトリに Apache サーバからの書き込みを許可します。

```
# rm /etc/zabbix/zabbix.conf.php
# chmod o+w /etc/zabbix
```

(2)ZABBIX のセットアップ

ZABBIX のセットアップページ(http://localhost/zabbix/)へアクセスして ZABBIX のセットアップを開始します。(図 31-2)

[Next]を選択して次ページへ進みます。

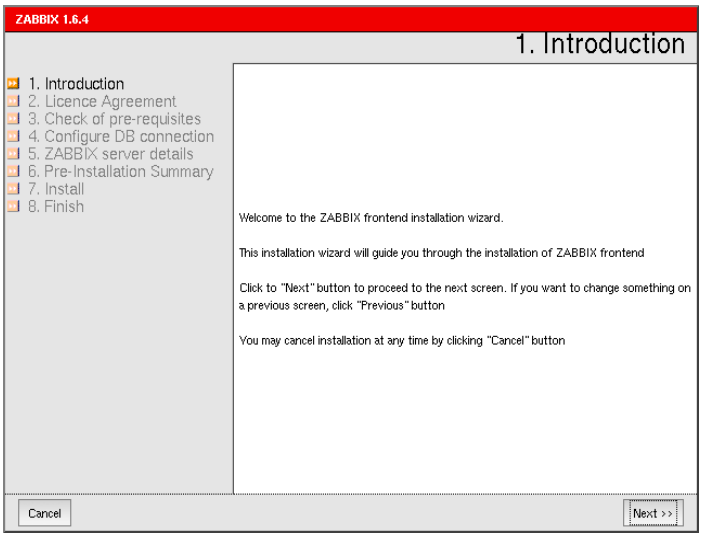


図 31-2 イントロダクション

表示されている GPL ライセンスを読み、ライセンスに同意する場合は[I agree]にチェックを入れ[Next]を選択して次ページへ進みます（図 31-3）。

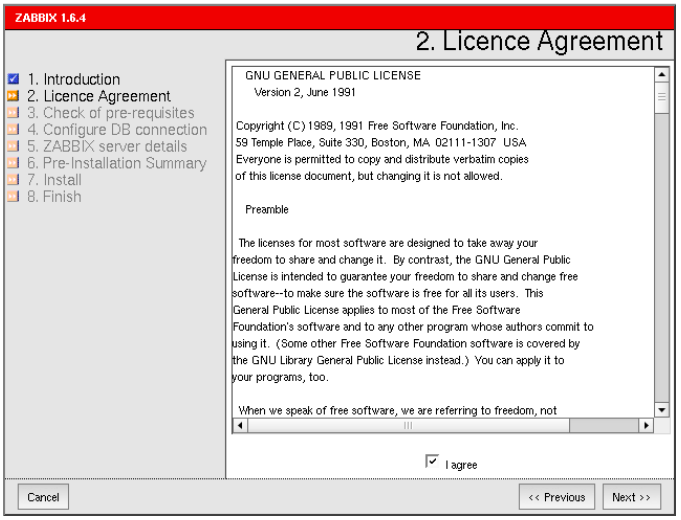


図 31-3 GPL ライセンス画面

全ての項目に OK が表示されていることを確認し、[Next]を選択して次ページへ進みます。（図 31-4）

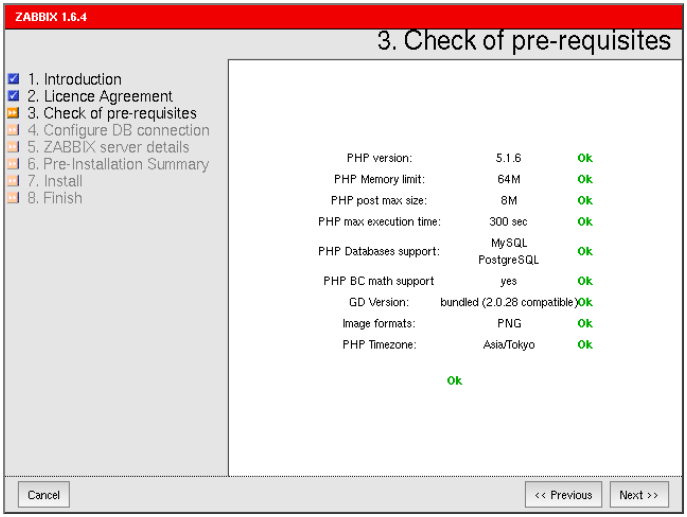


図 31-4 設定前の環境確認

[Type] [Host] [Port] [Name] [User] [Password] を入力し、[Test connection]を押して OK が表示されることを確認して、[Next]を選択して次ページへ進みます。（図 31-5）

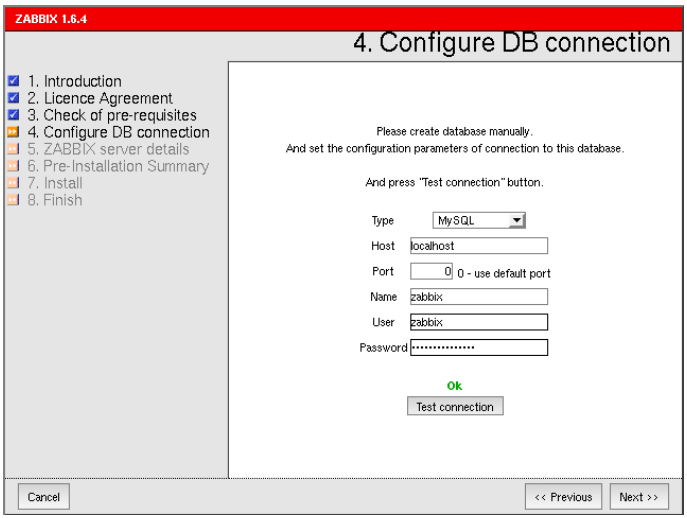


図 31-5 データベース設定

図 31-5 では例として以下の値を設定しています。

Type	MySQL
Host	localhost
Port	0
Name	zabbix
User	zabbix
Password	zabbix_password （31.2.1-(1)で設定した MySQL のユーザー zabbix のパスワード）

Web フロントエンドから ZABBIX サーバの起動状態の確認に利用する設定を入力します。

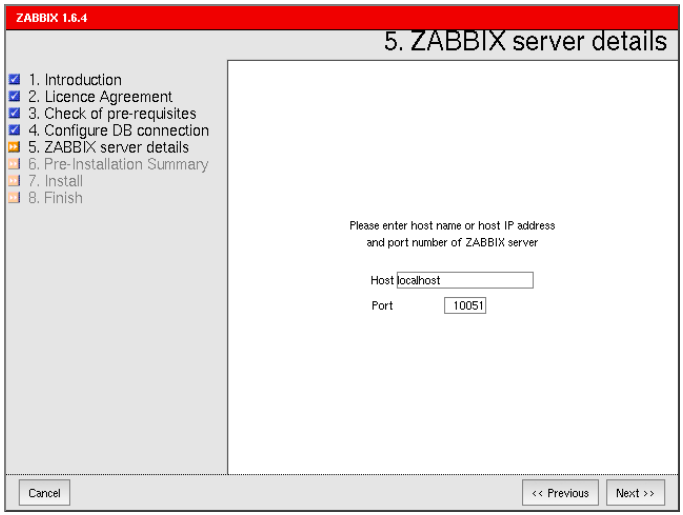


図 31-6 ZABBIX サーバの設定

表示されている情報に間違いが無いことを確認し、[Next]を選択して次ページへ進みます。(図 31-7)

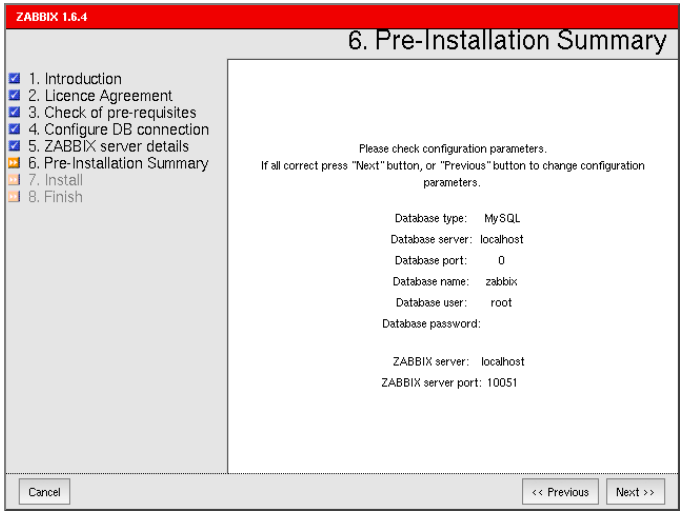


図 31-7 インストールの環境確認

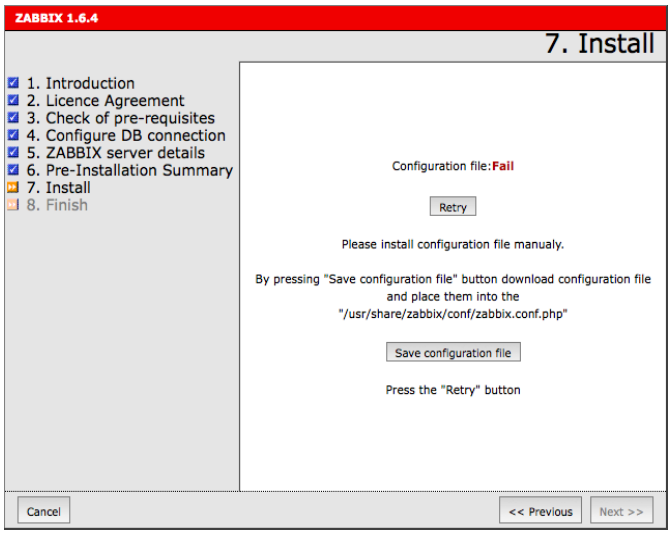


図 31-8 設定ファイルの保存

Web フロントエンドの設定ファイルを保存します。[Retry]をクリックして下記の画面が出れば正常に保存されています。

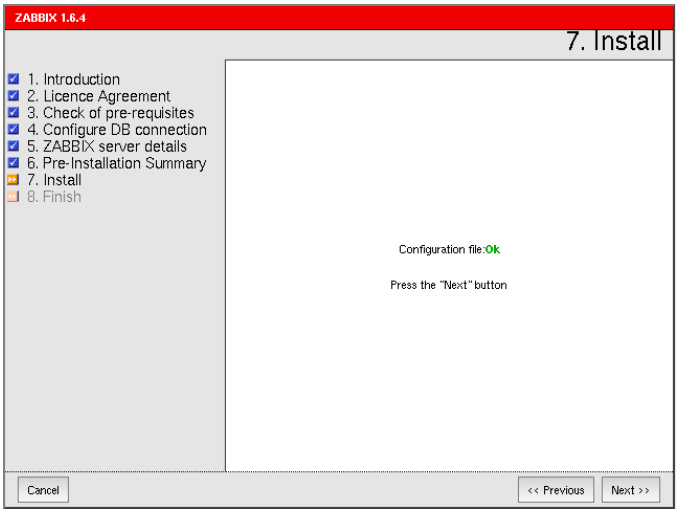


図 31-9 ZABBIX のインストール完了

インストールに問題がないことを確認し、[Next]を選択して次ページへ進みます。(図 31-9)
ZABBIX のインストール完了画面が表示されます。(図 31-10)

図 31-10 ZABBIX のインストール完了

(3)ZABBIX の設定ファイルからの書き込み権限削除

Apache サーバによる ZABBIX の設定ファイル**/etc/zabbix** への書き込み権限を削除します。

```
# chmod o-w /etc/zabbix
```

31.2.5 ZABBIX のエージェントプロセスの設定

ZABBIX サーバー自身を監視するために、ZABBIX エージェントを設定します。

(1)zabbix_agentd.conf の設定

ZABBIX エージェントの設定を行なうため、以下の項目を、**/etc/zabbix/zabbix_agentd.conf** に設定します。

```
Server=127.0.0.1 ← ZABBIXサーバのIPアドレス（注1）
Hostname=localhost ← ZABBIXエージェントを動作させるホスト名
ListenIP=127.0.0.1 ← ListenするIPアドレス（注2）
PidFile=/var/run/zabbix/zabbix_agentd.pid
```

```
LogFile=/var/log/zabbix/zabbix_agentd.log
```

注 1：ZABBIX エージェントはここに設定した IP アドレスからの通信にのみ応答します。

注 2：ZABBIX サーバの監視設定で、監視対象として登録する際に指定する IP アドレスになります。

(2)ZABBIX のエージェントプロセスの起動

zabbix-agent サービスを起動します。

```
# service zabbix-agent start
Starting zabbix agent:                                     [  OK  ]
# chkconfig zabbix-agent on
```

31.2.6　ZABBIX Web フロントエンドへのログイン

以下 URL より Web フロントエンドにアクセスし、ログインすることにより、ZABBIX の監視状況を一覧できます。

初期設定ではユーザ名 **admin**、パスワード **zabbix** でログイン可能です。

```
http://localhost/zabbix
```

図 31-11 Web フロントエンド

第32章　日本語関連

この章で説明する内容

目的	文字コードの設定を行えるようにする 日本語入力を行えるようにする フォントを使用できるようにする
機能	環境変数 LANG によるロケールの設定方法 日本語入力の設定方法 フォントのインストール方法
必要な RPM	
設定ファイル	
章の流れ	1 日本語文字コード 2 文字コードの設定 3 日本語入力設定 4 フォントのインストール 5 ロケールの変更
関連 URL	

32.1　日本語文字コード

日本語の文字コードは、**日本語 EUC**（euc-jp）、**シフト JIS**（shiftjis）、**JIS**（ISO-2022-JP）というように、3 種類の文字コードが使われてきました。最近ではこれらの文字コードに加えて、すべての言語の文字を単一の統一されたコードに割り当てた Unicode（**UTF-8**）が使われるようになってきています。

Asianux Server 3 では、標準の設定で UTF-8 を使うように設定されています。

32.2　文字コードの設定

Asianux Server 3 でファイル名やターミナルで使用する日本語の文字コードを変更するには、次のファイルで環境変数 LANG に使う文字コードを設定します。

後述するロケールの設定で行うと容易に設定することができます。

ファイル	設定が有効な範囲
/etc/sysconfig/i18n	システム全体
$HOME/.i18n	ユーザーごと

日本語文字コードは、環境変数 LANG に次のように設定します。

使用する文字コード	環境変数 LANG の設定
UTF-8（デフォルト）	**LANG=ja_JP.UTF-8**

他のマシンに対してファイル共有を行っている場合は、**$HOME/.smb/smb.conf** を削除します。

文字コードの設定は、ログアウトして再度ログインした時に有効になります。

ファイル名に日本語を使用している場合は、運用途中で使用する文字コードを変更すると、それまで作成されていた日本語のファイル名を正しく表示できなくなります。文字コードの設定は運用前に決めておくか、新たなユーザーを追加して、ユーザーごとの設定で文字コードを指定して利用することが推奨されます。

32.3 日本語入力設定

Asianux Server 3 では、仮名漢字変換サーバーとして **Anthy** が用意されています。日本語入力用 IM（Input Method）としては、X Window 上で利用できる **SCIM**（Smart Common Imput Method platform）が用意されています。

日本語入力を行うには、X Window が立ち上がった状態で、ターミナルやテキストエディタなどを起動しキー入力できる状態にした上で、「Ctrl＋スペースキー」もしくは「半角/全角」キーを押します。

32.3.1 SCIM の設定

SCIM は日本語入力モードになると、Windows の日本語入力のようにツールバー（図 32-1）が表示され、現在の入力モード表示や各種モード変更などが行えるようになっています。

図 32-1 SCIM ツールバー

更に細かな設定を行うために、GUI のセットアップユーティリティが提供されています。SCIM セットアップユーティリティの起動コマンドは次のとおりです。

```
# /usr/bin/scim-setup
```

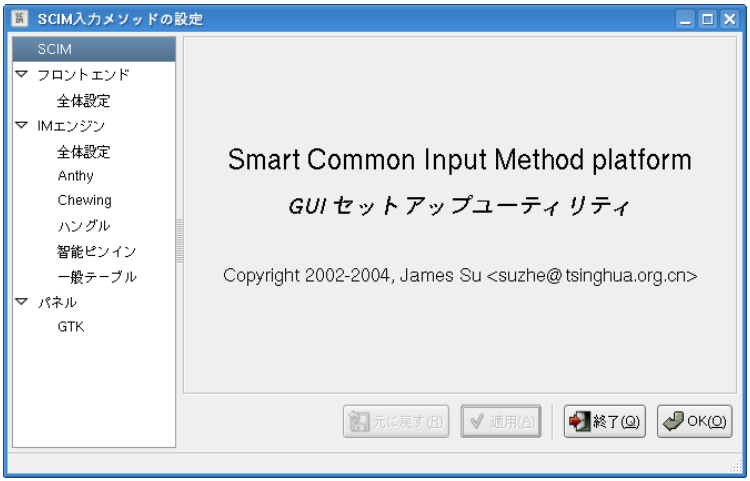


図 32-2 SCIM セットアップユーティリティ初期画面

図 32-3 のような初期画面が表示されるので、左の設定項目を選択し様々な設定を行います。

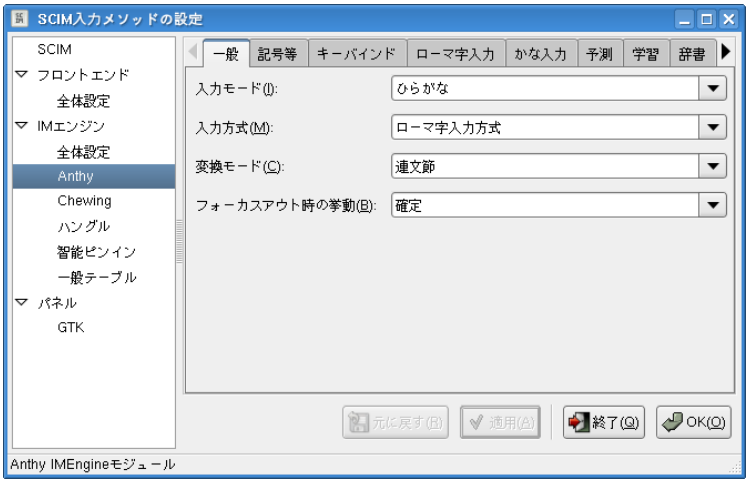


図 32-3　Anthy 項目選択画面

設定変更が終わったら、[OK(O)]ボタンを押しツールを終了させます。

32.4 フォントのインストール

rpm パッケージ化されていない、TrueType フォントや PostScript Type1 フォントをインストールするには、[Start]メニューから「設定」－「システム管理」－「フォントインストーラ」を起動します。

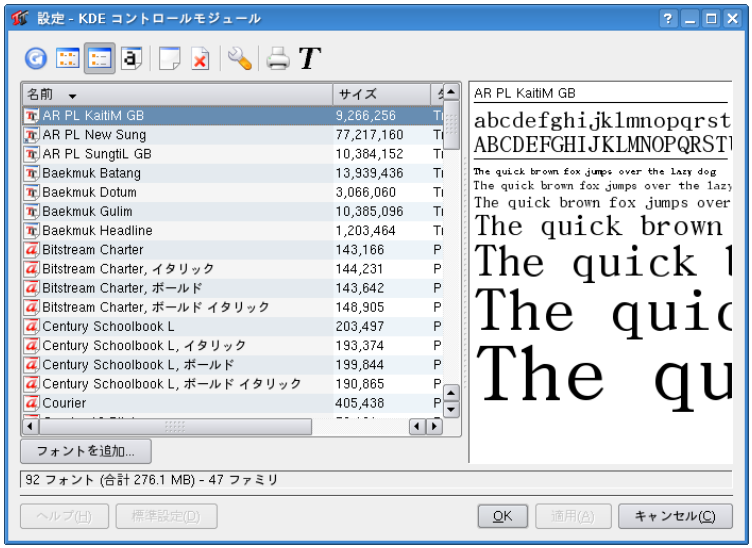


図 32-4 フォントインストーラ画面

フォントのインストールは、次のような手順で行います。

(1) インストールしたいフォントファイルを作業用の任意のディレクトリにコピーしておきます。
(2) フォントインストーラを起動します。
(3) [フォントを追加]ボタンをクリックします。
(4) ファイル選択のダイアログウィンドウが表示されるので、先ほどコピーしておいたフォントファイルを指定します。
(5) [OK]ボタンをクリックすると、フォントがインストールされます。
(6) インストールが終わったら、フォントインストーラを終了させます。

以上で新しくインストールしたフォントが利用可能になります。

32.5　ロケールの変更

Asianux Server 3 では、ロケールの種類として次に示すものが利用できます。

- 簡体中国語（UTF-8）
- 簡体中国語（GB18030）
- 繁体中国語（UTF-8）
- 繁体中国語（big5）
- 英語（UTF-8）
- 日本語（eucJP）
- 日本語（UTF-8）（インストール後の設定値）
- 韓国語（UTF-8）
- 韓国語（eucKR）

ロケールの変更を行いたい場合は、GUI ツールとして提供されている[言語の選択]ツールを利用することで容易に行うことができます。

[Start]メニューから「設定」－「システム設定」－「言語」でツールを起動します。（図 32-5）

図 32-5　ロケール設定ツール画面

リストの中から設定したいロケールを選択します。選択後、[OK(O)]ボタンを押すと更新完了画面（図 32-6）が表示されるので、[OK(O)]ボタンを押し指示通り再度ログインを行うとロケール変更が完了します。

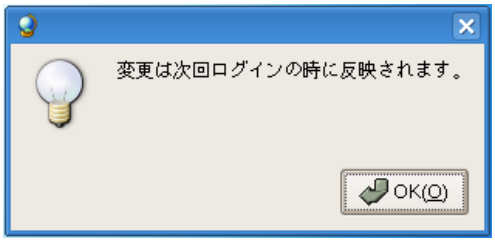


図 32-6 更新完了画面

第33章　パフォーマンス管理

この章で説明する内容

目的	システムパフォーマンスの管理方法について理解する
機能	システム動作統計情報の収集
必要な RPM	sysstat —— システムモニタリングツール procps —— システム＆プロセス モニタリング・ユーティリティ
設定ファイル	なし
章の流れ	1 パフォーマンス管理の概要 2 procps に含まれるコマンドの使い方 3 sysstat に含まれるコマンドの使い方
関連 URL	**Sysstat Home Page**　http://perso.orange.fr/sebastien.godard/ **The Linux Kernel メモリ管理** http://www.linux.or.jp/JF/JFdocs/The-Linux-Kernel-4.html **日本 OSS 推進フォーラム** http://www.ipa.go.jp/software/open/forum/

33.1　パフォーマンス管理の概要

本章では、Linux のパフォーマンス管理に有用なソフトウェアについて説明します。

システムを管理するうえで、CPU の使用率やメモリの使用量、ディスク I/O のビジー率などは重要な要素です。これらのパフォーマンス情報を管理する有用なツールとして、**vmstat**、**iostat**、**sar**、**free**、**top** などがあります。本章では、これらのコマンドの使い方について説明します。

なお、これらのコマンドは単独のパッケージではなく、procps、sysstat というパッケージに含まれています。ここでは、含まれるパッケージごとにコマンドを説明します。

33.2　procps に含まれるコマンドの使い方

procps は、システム統計情報を表示するパッケージで、**ps**、**free**、**top**、**uptime**、**vmstat** など、10 種類以上のコマンドが含まれています。今回はその中でも重要な **free**、**top**、**vmstat** の 3 つについて説明します。

33.2.1　top

top を使うと、CPU の使用率やメモリ使用量、各プロセスの CPU 使用率など、さまざまなパフォーマンス情報を表示できます。リアルタイムに監視できるため、手軽にパフォーマンス情報を知りたいときに便利です。詳しい内容は、オンラインマニュアルで参照できます。

```
$ /usr/bin/man top
```

top を使うには、次のように入力します。

```
$ /usr/bin/top
```

起動すると次のような画面が表示されます。上部には CPU の使用率やメモリの使用量が表示されます。下半分のリストには、CPU の使用率順にプロセスが表示されます。表示する項目やソート順を変更することもできます。

[h]キーを入力するとヘルプが表示されるので、詳しい使い方はヘルプを見てください。

終了するときには[q]キーを入力します。

```
top - 03:07:33 up  2:15,  1 user,  load average: 0.00, 0.00, 0.00
Tasks:  37 total,   1 running,  36 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.3% us,  1.0% sy,  0.0% ni, 97.0% id,  0.0% wa,  0.0% hi,  1.7% si
Mem:    251908k total,   127192k used,   124716k free,    14364k buffers
Swap:  1052248k total,        0k used,  1052248k free,    79572k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 4967 root      17   0  2012  904  744 R  0.7  0.4   0:00.08 top
 3605 root      15   0  3448  496  416 S  0.3  0.2   0:16.29 vmware-guestd
 4237 tama      16   0 10500 2384 1908 S  0.3  0.9   0:00.47 sshd
    1 root      16   0  2620  544  464 S  0.0  0.2   0:01.02 init
    2 root      34  19     0    0    0 S  0.0  0.0   0:00.00 ksoftirqd/0
    3 root       5 -10     0    0    0 S  0.0  0.0   0:00.36 events/0
    4 root       5 -10     0    0    0 S  0.0  0.0   0:00.13 khelper
    5 root      15 -10     0    0    0 S  0.0  0.0   0:00.00 kacpid
   18 root       5 -10     0    0    0 S  0.0  0.0   0:00.27 kblockd/0
   28 root      20   0     0    0    0 S  0.0  0.0   0:00.00 pdflush
   29 root      15   0     0    0    0 S  0.0  0.0   0:01.15 pdflush
   31 root       8 -10     0    0    0 S  0.0  0.0   0:00.00 aio/0
   19 root      15   0     0    0    0 S  0.0  0.0   0:00.00 khubd
   30 root      25   0     0    0    0 S  0.0  0.0   0:00.00 kswapd0
  617 root      25   0     0    0    0 S  0.0  0.0   0:00.00 kseriod
  706 root      15   0     0    0    0 S  0.0  0.0   0:03.06 kjournald
 2713 root       6 -10  2328  440  368 S  0.0  0.2   0:00.08 udevd
 2986 root      20   0     0    0    0 S  0.0  0.0   0:00.00 kjournald
 3525 root      16   0  2044  644  540 S  0.0  0.3   0:00.56 syslogd
```

33.2.2 free

free は、システムの空きメモリと使用メモリを表示するコマンドです。

ただし **free** で表示される項目は、必ずしも空きメモリだとは限りません。

```
$ /usr/bin/free
             total       used       free     shared    buffers     cached
Mem:        770144     739524      30620          0      96028     441740
-/+ buffers/cache:     201756     568388
Swap:      1636072     162404    1473668
```

実行すると、**Mem**、**-/+ buffers/cache**、**Swap** の 3 行が表示されます。

それぞれの値の意味については表 33-1 を参照してください。

表 33-1　free コマンドの出力項目の意味

項目	説明
Mem	ページキャッシュとバッファキャッシュを考慮しないメモリサイズ。
total	OS が認識している物理的なメモリサイズ。RAID カードや NIC などを装着しているときは、それらのためにメモリが使われるので、実際の搭載メモリサイズよりも少なくなります。
used	使用しているメモリサイズ。これにはバッファキャッシュやページキャッシュなど、OS がディスクキャッシュのために使用しているメモリも含まれます。
free	空きメモリサイズ。この値にはバッファキャッシュとページキャッシュが含まれていません。一般に Linux は使い続けるほど、メモリをキャッシュに割り当てます。そのために、使い続けるほど free の値はゼロに近づきます。この値が少なくなったからといって、空きメモリがないわけではないことに注意してください。
shared	共有メモリに割り当てたメモリ。
buffers	バッファキャッシュに割り当てたメモリ。バッファキャッシュはブロックデバイス用のキャッシュです。
cached	ページキャッシュに割り当てたメモリ。ページキャッシュは、ファイルに対するキャッシュです。
-/+ buffers/cache	ページキャッシュとバッファキャッシュを考慮したメモリサイズ
used	1 行目の used からページキャッシュとバッファキャッシュを引いた値。 OS とアプリケーションが純粋に使用しているメモリサイズを表します。
free	1 行目の free にページキャッシュとバッファキャッシュを足した値。 キャッシュに割り当てられているメモリを、自由に割り当て可能なメモリと考えれば、この値が空きメモリサイズになります。
Swap	スワップに割り当てたサイズ
total	スワップに割り当てたディスクサイズ。
used	割り当てた中で使用中のサイズ。
free	割り当てた中で使用していないサイズ。

33.2.3　vmstat

vmstat は、プロセスの状態、メモリの使用状況、ページングの回数、I/O の回数、CPU の使用率などを表示するコマンドです。幅広い情報が取得できて、結果をファイルに出力できるので、データベースのチューニングのような、アプリケーションのチューニングに便利です。詳しい内容は、オンラインマニュアルで参照できます。

```
$ /usr/bin/man vmstat
```

vmstat は、次の構文で使用します。実行回数を省略すると、停止されるまで無限に実行し続けるので、終わらせたいところで[Ctrl]+[C]キーを押します。

```
$ /usr/bin/vmstat [実行間隔（s）[実行回数]]
```

次に、具体的な実行例について説明します。次の例では 1 秒間隔で合計 3 回実行しています。1 行目は、OS を起動してからの平均で、それ以降は実行間隔ごとの値です。

```
$ /usr/bin/vmstat 1 3
procs ----------memory---------- ---swap-- -----io---- --system-- -----cpu------
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 0  0      0  20828  16180 133952    0    0    52     9  518   92  1  0 98  1  0
 0  0      0  20828  16180 133952    0    0     0     0 1016   76  0  0 100  0 0
 0  0      0  20832  16180 133952    0    0     0     0 1023   72  0  0 100  0 0
```

出力結果のそれぞれの項目の意味は表 33-2 を参照してください。

表 33-2 vmstat の出力項目の意味

1 行目	2 行目	説明
procs	r	CPU を割り当て中もしくは割り当て可能なプロセスの数。CPU の個数以下であることが望ましい。
	b	割り込みを禁止しているプロセスの数。I/O 待ちなどで、割り込み不可能なときに発生。ゼロであることが望ましい。
memory	swpd	使用している仮想メモリの量（KB）。
	free	空きメモリの量（KB）。
	buff	バッファキャッシュに使用されているメモリ量（KB）。
	cache	ページキャッシュに使用されているメモリ量（KB）。
swap	si	ディスクからページインされているメモリの量（KB/秒）。
	so	ディスクにページアウトしているメモリの量（KB/秒）。
io	bi	ブロックデバイスに送られたブロック数（blocks/秒）。
	bo	ブロックデバイスから受け取ったブロック数（blocks/秒）。
system	in	1 秒あたりの割り込み回数。クロック割り込みも含む。
	cs	1 秒あたりのコンテキストスイッチの回数。
cpu	us	ユーザー時間。
	sy	システム時間。
	id	アイドル時間。I/O 待ちは含まない。
	wa	I/O 待ち時間。
	st	仮想マシンから盗まれた時間。

33.3　sysstat に含まれるコマンドの使い方

sysstat は Linux 用に作成された、システム統計情報を報告するパッケージです。
主に次のツールによって構成されます。

- **iostat** —— I/O 関連の統計情報を報告します。
- **sar** —— システムの動作統計情報を報告します。
- **mpstat** —— プロセッサ関連の統計情報を報告します。

これらのツールは性能関連の情報をモニタするのに使用できます。
sar、iostat、mpstat は SVR4 系の商用 UNIX（Solaris など）にも実装されているツールで、特に iostat はデータベースのチューニングに欠かすことができないツールといえます。

33.3.1　iostat

iostat は、I/O 統計情報を表示します。**iostat** によって、各ディスクの単位時間あたりの I/O 数を知ることができます。**iostat** の詳しい内容については、オンラインマニュアルで参照できます。

```
$ /usr/bin/man iostat
```

iostat は、次の構文で使用します。実行回数を省略すると、停止されるまで無限に実行し続けるので、終わらせたいところで[Ctrl]+[C]キーを押します。

```
$ /usr/bin/iostat [オプション [実行間隔（s）[実行回数]]]
```

次に **iostat** の具体的な使用方法について説明します。

➢ ディスクに関する I/O 統計情報を 2 秒間隔で出力する

```
$ /usr/bin/iostat -dt 2
Linux 2.6.18-8.9AX (localhost)        2007年08月15日

時間: 15時39分07秒
Device:            tps   Blk_read/s   Blk_wrtn/s   Blk_read   Blk_wrtn
hda               6.66       163.18        45.24     857160     237660

時間: 15時39分09秒
Device:            tps   Blk_read/s   Blk_wrtn/s   Blk_read   Blk_wrtn
hda               1.99         5.30        84.77          8        128

時間: 15時39分11秒
Device:            tps   Blk_read/s   Blk_wrtn/s   Blk_read   Blk_wrtn
hda               0.00         0.00         0.00          0          0
```

- 最初の出力はシステムが起動してからの統計値を表しています。
- 2 回目以降の出力から指定した秒間隔内での統計値となります。
- hp 社製の RAID カードなどのように、デバイス名が**/dev/sda** や**/dev/hda** でないデバイスに関しては、情報が表示されないことがあります。その場合は、「**iostat -dt -x /dev/cciss/c0d0 2**」のように、**-x** オプションを指定して実行します。デバイス名は省略可能です。

➢ ディスクに関する詳細な I/O 統計情報を 2 秒間隔で出力する

```
$ /usr/bin/iostat -d -x /dev/sda 2
Linux 2.6.18-8.9AX (localhost)        2007年08月15日

Device:         rrqm/s   wrqm/s     r/s     w/s   rsec/s   wsec/s avgrq-sz
avgqu-sz   await  svctm  %util
hda               1.42     4.04    4.93    1.55   157.45    44.66    31.22
0.15   23.00   5.22   3.38

Device:         rrqm/s   wrqm/s     r/s     w/s   rsec/s   wsec/s avgrq-sz
avgqu-sz   await  svctm  %util
hda               0.00     0.00    0.00    0.00     0.00     0.00     0.00
0.00    0.00   0.00   0.00

Device:         rrqm/s   wrqm/s     r/s     w/s   rsec/s   wsec/s avgrq-sz
avgqu-sz   await  svctm  %util
➢ hda               0.00     0.00    0.00    0.00     0.00     0.00     0.00
  0.00    0.00   0.00   0.00
```

-x オプションを指定すると、詳細なディスク情報を取得できます。この中でも重要なのは次の項目です。

- **avgqu-sz ——** デバイスごとの I/O リクエストの平均キューの長さ。これが大きいと I/O 待ちが発生している可能性が高いと言えます。
- **await ——** I/O リクエストを発行してから、それが実行されるまでの平均待ち時間（m sec）。
- **%util ——** デバイスのリクエストキューに I/O リクエストがあった時間の割合 。ディスクのビジー率。

33.3.2　sar

sar は、さまざまなシステムの動作統計情報を報告します。システムをチューニングするときは、システムの状況を正確に把握することが肝心です。そのような場合は **sar** を利用しましょう。

sar の詳しい内容については、オンラインマニュアルを参照してください。

```
$ /usr/bin/man sar
```

次に **sar** の具体的な使い方をいくつか紹介します。

➢ I/O 統計情報を 2 秒間隔で 4 回出力する

```
$ /usr/bin/sar -b 2 4
Linux 2.6.18-8.9AX (localhost)        2007年08月15日

15時47分51秒       tps      rtps      wtps   bread/s   bwrtn/s
15時47分53秒     63.85     63.85      0.00   1846.15      0.00
15時47分55秒    347.37    347.37      0.00  11396.49      0.00
15時47分57秒    324.56    267.54     57.02   8203.51    884.21
15時47分59秒    700.88    592.04    108.85  18477.88   1515.04

平均値:      348.41    308.49     39.92   9686.62    577.49
```

➢ ディスクに対する I/O の統計情報（512 バイト単位）を 2 秒間隔で無限に出力する

```
$ /usr/bin/sar -B 2 0
Linux 2.6.18-8.9AX (localhost)        2007年08月15日

15時50分09秒  pgpgin/s pgpgout/s   fault/s  majflt/s
15時50分11秒      0.00      0.00     35.92      0.00
15時50分13秒      0.00      0.00     13.51      0.00
15時50分15秒      0.00      0.00      7.69      0.00
```

➢ プロセス生成の統計情報を 3 秒間隔で無限に出力する

```
$ /usr/bin/sar -c 3 0
Linux 2.6.18-8.9AX (localhost)        2007年08月15日

15時51分49秒    proc/s
15時51分52秒      0.00
15時51分55秒      0.44
```

➢ メモリとスワップの使用統計情報を 2 秒間隔で無限に出力する

```
$ /usr/bin/sar -r 2 0
Linux 2.6.18-8.9AX (localhost)        2007年08月15日

16時11分49秒 kbmemfree kbmemused  %memused kbbuffers  kbcached kbswpfree
kbswpused  %swpused  kbswpcad
16時11分51秒     16944    238808     93.37     10772    136716    524184
96      0.02        12
16時11分53秒     16944    238808     93.37     10772    136716    524184
96      0.02        12
```

➢ スワップ領域に対する統計情報(ページ単位)を 2 秒間隔で無制限に出力させる

```
$ /usr/bin/sar -W 2 0
Linux 2.6.18-8.9AX (localhost)        2007年08月15日

16時15分06秒  pswpin/s pswpout/s
16時15分08秒      0.00      0.00
16時15分10秒      0.00      0.00
16時15分12秒      0.00      0.00
16時15分14秒      0.00      0.00
16時15分16秒      0.00      2.61
```

➢ コンテキストスイッチ統計情報を 3 秒間隔で 3 回出力する

```
$ /usr/bin/sar -w 3 3
Linux 2.6.18-8.9AX (localhost)        2007年08月15日

16時17分50秒   cswch/s
16時17分53秒     73.59
16時17分56秒     62.01
16時17分59秒     68.56

平均値:       68.07
```

sar には、上記以外にも統計情報を出力するオプションは数多く存在します。詳細はオンラインマニュアルを参照してください。

第34章　GUI ツール

この章で説明する内容

目的	システム付属の GUI(TUI)ツールの機能について理解する
機能	各種 GUI 管理ツールの機能一覧 各種 TUI 管理ツールの機能一覧
必要な RPM	
設定ファイル	
章の流れ	1 コントロールパネル/[設定]メニュー 2 テキストモードセットアップユーティリティ
関連 URL	

34.1　コントロールパネル/[設定]メニュー

Asianux Server 3 のコントロールパネルまたは[Start]ボタンをクリックして[設定]メニューを開いたときにアクセスできる、GUI の管理ツールとして提供されているものは次のようになります。

（1）項目「ハードウェア」で提供されている管理ツール一覧を表 34-1 に示します。

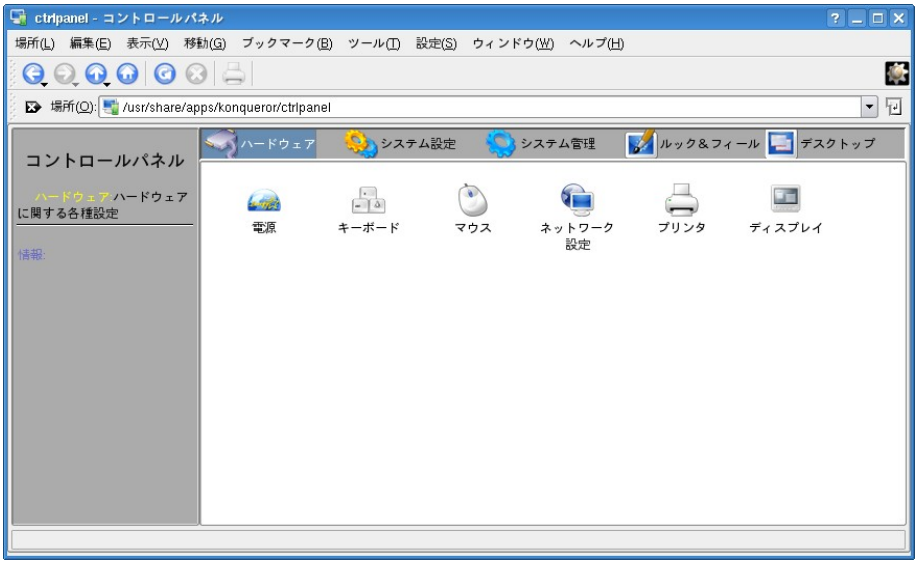


図 34-1　コントロールパネルの「ハードウェア」タブ

表 34-1　項目「ハードウェア」の管理ツール

ツール名	機能概要
電源	スタンバイ設定などの電源管理の設定を行う
キーボード	キーボードの設定を行う
マウス	マウスの設定を行う
ネットワーク設定	ネットワーク設定を行う
プリンタ	プリンタ等の印刷設定を行う
ディスプレイ	画面の設定変更を行う

(2) 項目「システム設定」で提供されている管理ツール一覧を表 34-2 に示します。

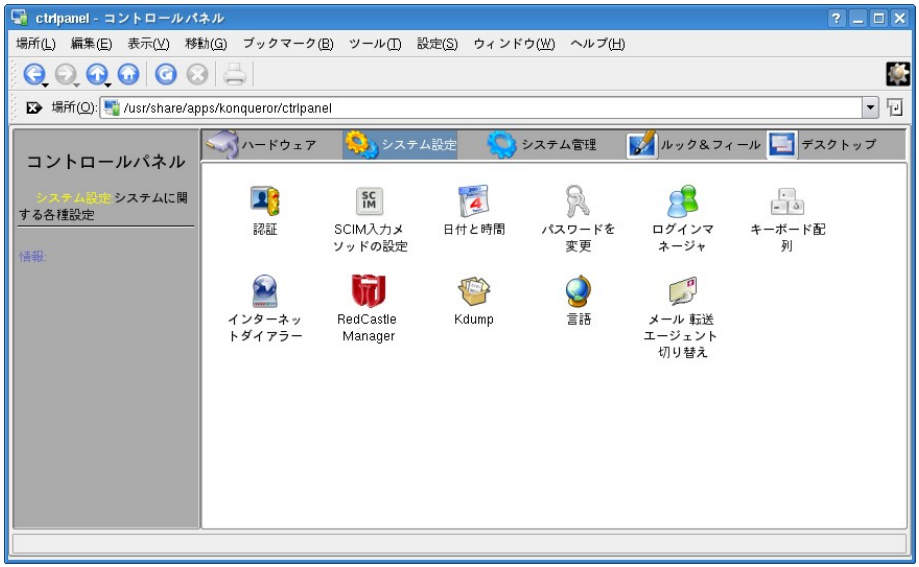


図 34-2 コントロールパネルの「システム設定」タブ

表 34-2 項目「システム設定」の管理ツール

ツール名	機能概要
認証	ログイン時の認証方法について設定を行う
SCIM 入力メソッドの設定	文字入力システムの設定を行う
日付と時間	システム日時の変更を行う
パスワードを変更	パスワードの変更を行う
ログインマネージャ	ログイン画面の表示や自動ログインなどの設定を行う
キーボード配列	キーボードの配列の設定を行う
インターネットダイアラー	インターネットダイヤルアップツール
RedCastle Manager	RedCastle に関する設定を行う
Kdump	カーネルクラッシュダンプの設定を行う
言語	システムで使用する言語の設定を行う
メール転送エージェント切り替え	Sendmail、Postfix のどちらを使用するか設定を行う

(3) 項目「システム管理」で提供されている管理ツール一覧を表 34-3 に示します。

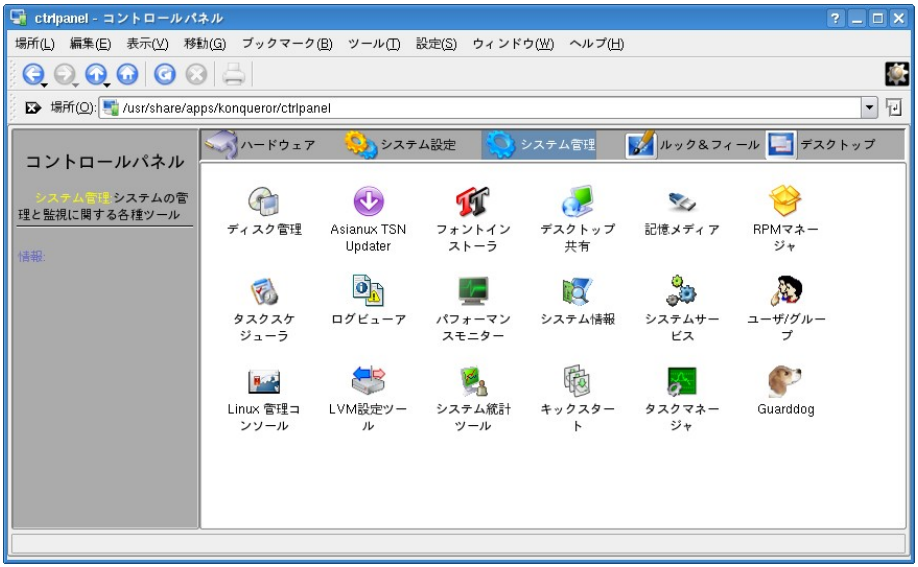


図 34-3 コントロールパネルの「システム管理」タブ

表 34-3 項目「システム管理」の管理ツール

ツール名	機能概要
ディスク管理	ディスクのフォーマット、マウントを行う
Asianux TSN Updater	システムのアップデートを行う
フォントインストーラ	フォントのインストール、アンインストールを行う
デスクトップ共有	デスクトップを遠隔表示・操作するための設定を行う
記憶メディア	メディアを認識した時の動作に関して設定を行う
RPM マネージャ	RPM パッケージのインストール、アンインストールを行う
タスクスケジューラ	ジョブの登録、変更などの管理を行う
ログビューア	各種ログ情報の表示を行う
パフォーマンスモニター	各種稼動状況をグラフ表示する
システム情報	ハードウェアに関する構成情報を表示する
システムサービス	サービス(デーモン)の起動・停止・設定などを行う
ユーザ／グループ	ユーザ／グループの登録、変更などの管理を行う
Linux 管理コンソール	システムのエラーログや情報などを表示する

ツール名	機能概要
LVM 設定ツール	LVM の構成変更を行う
システム統計ツール	sysstat 情報のグラフ表示を行う
キックスタート	キックスタート設定ツール
タスクマネージャ	実行中のアプリケーションの管理を行う
Guarddog	ファイアーウォール設定を行う

（4）項目「ルック＆フィール」で提供されている管理ツール一覧を表 34-4 に示します。

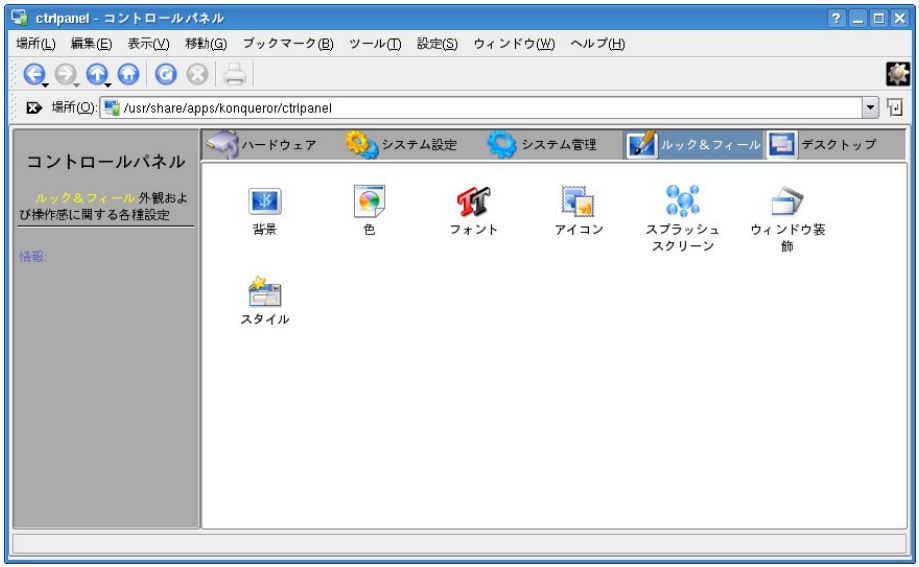


図 34-4　コントロールパネルの「ルック＆フィール」タブ

表 34-4　項目「ルック＆フィール」の管理ツール

ツール名	機能概要
背景	デスクトップの背景の設定を行う
色	ウィンドウの色の設定を行う
フォント	表示フォントの設定を行う
アイコン	アイコンの効果設定を行う
スプラッシュスクリーン	デスクトップ起動時の画面スタイルの設定を行う
ウィンドウ装飾	ウィンドウのスタイル設定を行う
スタイル	ウィンドウの背景、ボタンなどの設定を行う

(5) 項目「デスクトップ」で提供されている管理ツール一覧を表 34-5 に示します。

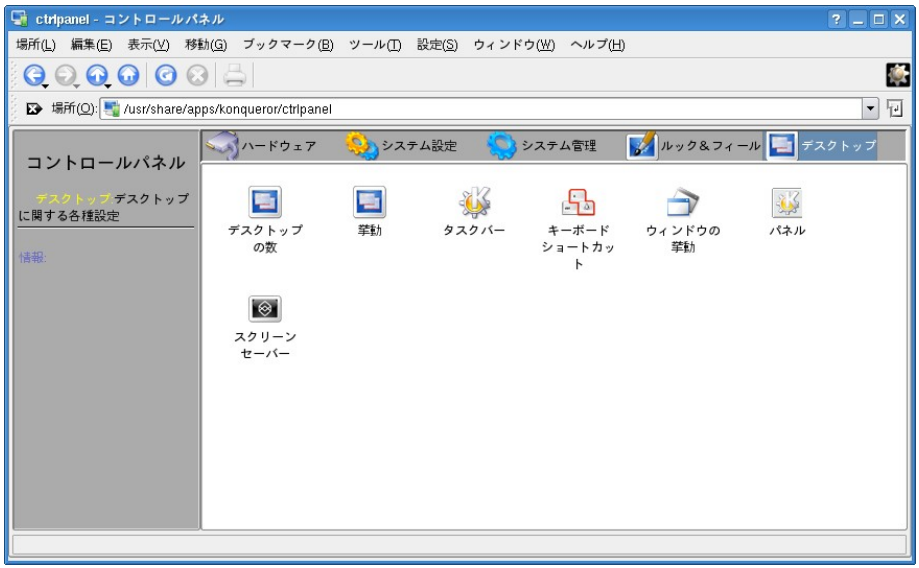


図 34-5 コントロールパネルの「デスクトップ」タブ

表 34-5 項目「デスクトップ」の管理ツール

ツール名	機能概要
デスクトップの数	仮想デスクトップの数の設定を行う
挙動	デスクトップの動作に関する全般的な設定を行う
タスクバー	タスクバーの表示設定を行う
キーボードショートカット	ショートカットキーの定義を行う
ウィンドウの挙動	ウィンドウの表示や挙動に関する設定を行う
パネル	パネルの表示や位置の設定などを行う
スクリーンセーバー	スクリーンセーバーの設定を行う

34.2　テキストモードセットアップユーティリティ

Asianux Server 3 で、テキストモードセットアップユーティリティ(**setup** コマンド)を使用するとテキストモード(TUI)管理ツールにアクセスできます。

TUI 管理ツールの一覧を表 34-6 に示します。括弧内は直接ツールを起動するコマンドです。

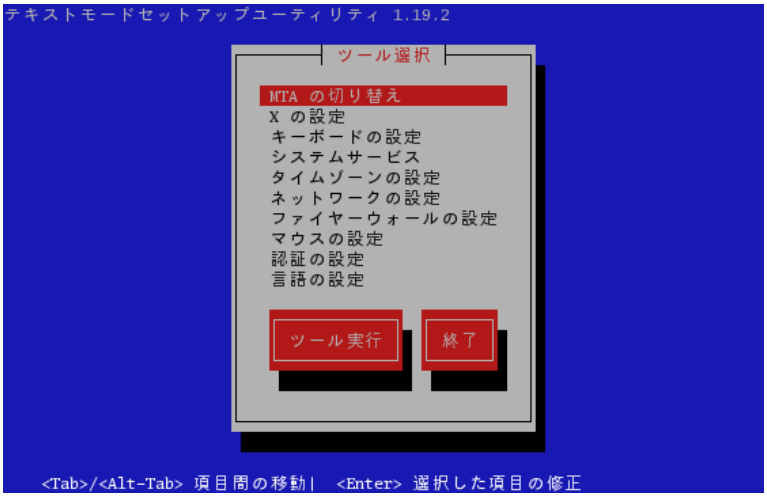


図 34-6　テキストモードセットアップユーティリティ

表 34-6 テキストモード(TUI)の管理ツール

ツール名	機能概要
MTA の切り替え (system-switch-mail-nox)	Sendmail、Postfix のどちらを使用するか設定を行う
X の設定 （system-config-display）	X の設定を行う
キーボードの設定 （system-config-keyboard）	キーボードの設定を行う
システムサービス （ntsysv）	サービス（デーモン）の自動起動設定を行う
タイムゾーンの設定 （system-config-date）	タイムゾーンの設定を行う
ネットワークの設定 （system-config-network-tui）	ネットワークの設定を行う
ファイヤーウォールの設定 （system-config-securitylevel）	ファイヤーウォールの設定を行う
マウスの設定 （mouseconfig）	マウスの設定を行う
認証の設定 （authconfig-tui）	ユーザー認証の設定を行う
言語の設定 （system-config-language）	システムのデフォルトの言語を設定する

第35章　トラブルシューティング

この章で説明する内容

目的	さまざまなトラブルシューティングについて知る
機能	レスキューモードの使用方法 ブートローダのリストア方法 mcinfo の使用方法 fsck の使用方法 kdump の設定方法 crash コマンドの使用方法 障害対応
必要な RPM	support-tools ―― mcinfo コマンド kexec-tools ―― kdump system-config-kdump ―― kdump 設定ツール crash ―― crash コマンド
設定ファイル	
章の流れ	1 レスキューモードの概要 2 レスキューモードの使用方法 3 シングルユーザーモード 4 ブートローダのリストア 5 mcinfo の使用方法 6 fsck ディスク障害対応 7 kdump の設定 8 crash コマンド 9 障害対応
関連 URL	**技術ドキュメント：Linux110 番（ミラクル・リナックス　サポートサイト）** http://www.miraclelinux.com/support/index.php **JM Project**　http://www.linux.or.jp/JM/ **Linux JF（Japanese FAQ）Project**　http://www.linux.or.jp/JF/

35.1 レスキューモードの概要

レスキューモードとは、インストール DVD を使用して Linux を起動する機能で、何らかの原因でハードディスクから Linux を起動できない場合に使用するためのものです。

Linux がハードディスクから起動できない原因には、たとえばルートファイルシステムを読み込めなくなった、MBR が壊れたなどがあります。

インストール DVD からシステムを起動すると、DVD イメージの中の Linux カーネルが起動するので、このカーネルを利用して緊急時の対処を行うことができます。ただし、Linux に対する高度な知識が必要になります。オペレーションミスによってシステム全体を破壊する可能性があるので、慎重に作業を行う必要があります。

また、通常運用の場合ならば、起動に必要な初期化処理で自動的に実行されるものがあります(**/etc/rc.d/** 配下の処理など)。しかし、インストール DVD から起動した場合には、それらの初期化処理は自動的には実行されません。インストール DVD を緊急時の対処として利用する場合は、目的に応じて必要な初期化処理を手動で行わなければなりません。

35.2 レスキューモードでのシステムの起動

レスキューモードで起動する手順は次のとおりです。

(1)インストール DVD を挿入して、マシンを起動します。

(2)boot プロンプトで、**linux rescue** と入力して、レスキューモードでシステムを起動します。

```
boot: linux rescue
```

(3) 日本語の場合は、[Japanese]を選択します。

言語を選択する画面(Choose a Language)が表示されます(図 35-1)。Japanese を選択すると、日本語が表示できない旨のメッセージが表示されるので、[OK]を選択します(図 35-2)。

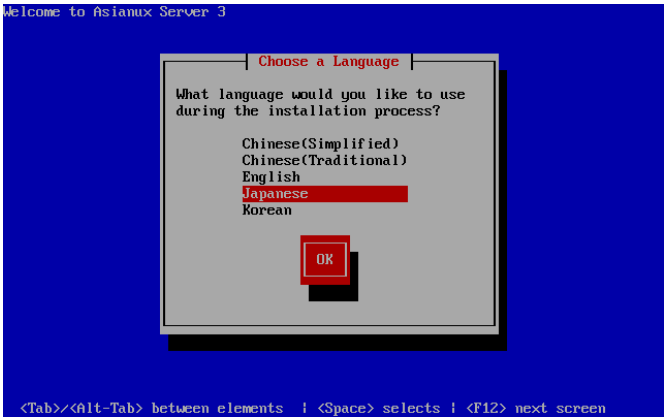


図 35-1 言語の選択

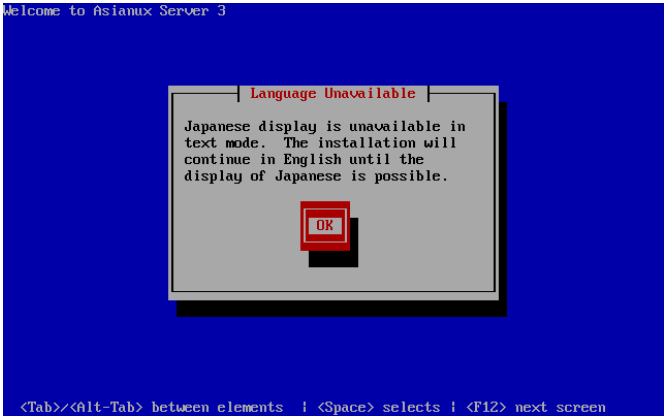


図 35-2 言語の選択

(4)キーボードの種類を選択する画面が表示されます(図 35-3)。
日本語キーボードの場合は、[jp106]を選択します。

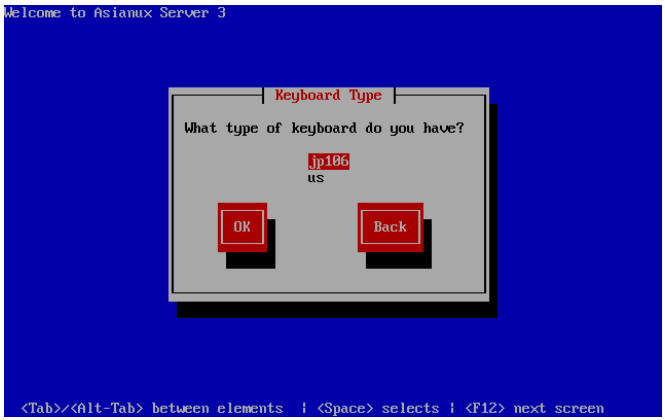


図 35-3　キーボードの種類の選択

(5) ネットワークの設定を行う画面が表示されます(図 35-4)。ネットワーク機能を使用する場合は、DHCP／スタティック IP アドレスの設定をしてください。

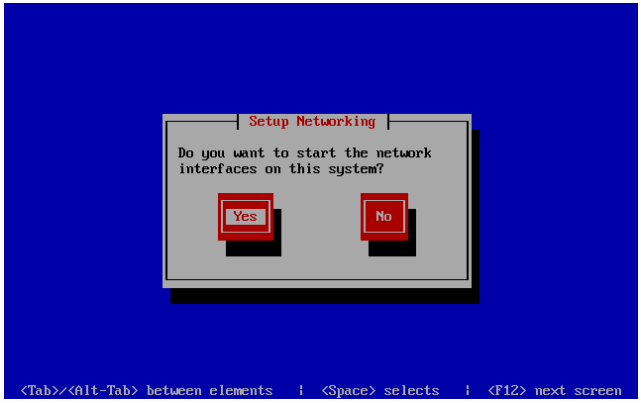


図 35-4　ネットワークの設定

(6) ハードディスクをマウントするか選択する画面(レスキュー)が表示されます(図 35-5、図 35-6)。
ハードディスクをマウントするように選択した場合は、使用可能なハードディスクを**/mnt/sysimage** にマウントします。後からマウントすることも可能です。次の 3 つから選択します。

- [Continue] —— マウントを行います。
- [Read Only] —— 読み取り専用でマウントを行います。
- [Skip] —— マウントを行いません。

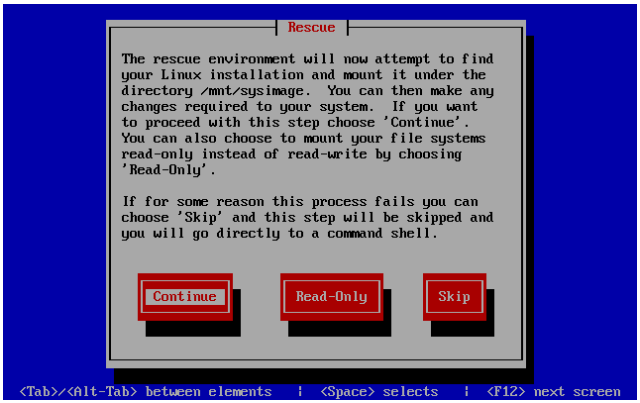


図 35-5 レスキュー画面

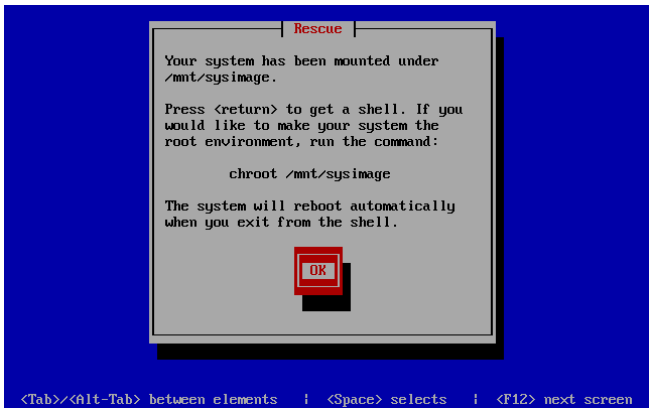


図 35-6 レスキュー画面

（7）シェルプロンプトが表示されます。（図 35-7）

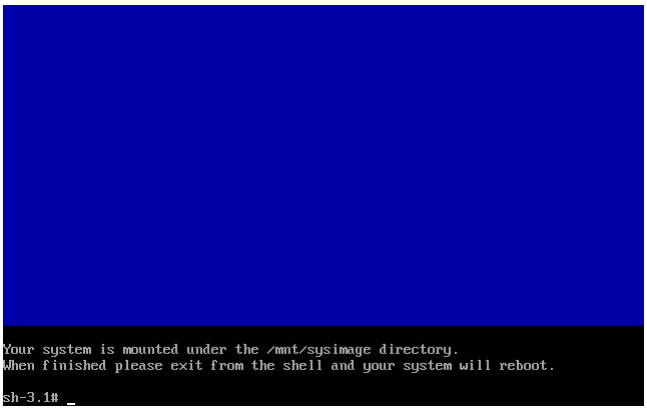


図 35-7 シェルプロンプト

これで、Linux のシェルを通じてシステムの操作を行うことができます。

ハードディスクをマウントした場合で、**/mnt/sysimage** を「**/**」として作業する場合は、次のコマンドを実行します。chroot 環境を終了する場合は、exit を実行します。

```
sh-3.00# chroot /mnt/sysimage
```

レスキューモードを終了する場合は、以下のコマンドを実行します。実行するとシステムは再起動します。

```
sh-3.00# exit
```

35.3 シングルユーザーモード

シングルユーザーモードとは、ログインしているユーザーが常に root ユーザーの状態であって、管理者以外のユーザーがログインしていない状態のことを言います。シングルユーザーモードは次のようなときに使用します。

- root のパスワードを忘れてしまったとき
- ネットワーク障害やディスク障害などにより、通常のモードでは起動しないとき

Linux を**シングルユーザーモード**で起動させるには、次の手順を実行します。

(1) GRUB のブートメニューに複数の OS のラベルが表示されている場合は、「Asianux Server 3」を選択します。
(2) [a]キーを押します。
(3) スペースを入力して引数を区切ってから **single** と入力します。

```
grub append> ro root=/dev/VolGroup00/LogVol00 single
```

(4) Enter] キーを押すと、Linux がシングルユーザーモードで起動します。

35.4　ブートローダのリストア

ブートローダのリストアとは、ディスクなどのトラブルで、MBR の破壊や GRUB の設定変更などにより、マシンが起動しなくなった場合に、MBR の初期化または GRUB の再設定を行う機能です。

ブートローダのリストア機能を使用するには特に特別な準備は不要で、MBR の初期化または GRUB の再設定を行いたいマシンに、Asianux Server 3 のインストール DVD を挿入し起動します。

その後は通常のインストールと同じく、言語選択、使用権許諾、キーボード選択と進んでいき、その後以下の画面（図 35-8）になります。

図 35-8　ブートローダのリストア 画面

通常のインストールであれば、この部分はパーティション設定の画面となりますが、インストール済みのマシンに再度インストールを行うとこの画面（図 35-8）に変わります。

[ブートローダ Asianux Server 3 のリストア]を選択し、[次(N)]ボタンを押します。

ブートローダの復元方法選択画面（図 35-9）が表示されます。

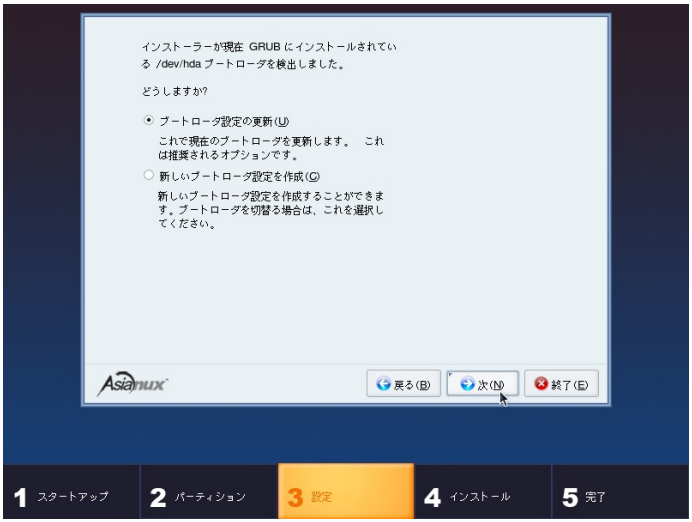


図 35-9 ブートローダの復元方法選択画面

[ブートローダ設定の更新(U)]を選択し、[次(N)]ボタンを押すとブートローダの再インストールが行われ、終了するとインストール完了画面（図 35-10）となりますので、マシン再起動を行います。

[新しいブートローダ設定を作成(C)]を選択し、[次(N)]ボタンを押すとインストール時と同じ、ブートローダの設定画面（図 35-11）となりますので、必要な設定を行います。その後はブートローダのインストール、設定が行われ、インストール完了画面（図 35-10）となりますので、マシン再起動を行います。

図 35-10　インストール完了画面

図 35-11　ブートローダ設定画面

35.5 mcinfo の使用方法

mcinfo は、現在稼動しているホストの各種ログやハードウェア情報、インストールされているパッケージ情報など、さまざまな情報を取得するコマンドです。

取得情報の中には、root ユーザーでしか取得できないものがあるので、**mcinfo** コマンドは必ず root ユーザーで実行してください。

mcinfo コマンドを実行すると、実行した時のパスに以下の形式の圧縮ファイルが作成されます。xxxx にはホスト名が、yyyy にはコマンドを実行した日時が記録されます。

```
mcinfo-xxxx-yyyy.tar.bz2
```

また、リダイレクトを使って **mcinfo** のログをファイルに書き込むことも可能です。

```
# /usr/sbin/mcinfo > mcinfo.log
```

取得される情報の一例を次に示します。

- Asianux Server 3 のバージョン（**/etc/asianux-release**）
- 起動時のメッセージ、デバイスの初期化処理など（**dmesg**）
- CPU の種類と個数（**/proc/cpuinfo**）
- 実メモリと swap の状態（**/proc/meminfo**、**/proc/swaps**）
- マウントしているデバイス（**df**）
- ディスクのドライブ割り当て（**fdisk -l**）
- PCI デバイスのリスト（**lspci**）
- ロードされているモジュールのリスト（**lsmod**）
- インストールされている RPM のリスト（**rpm -qa**）
- 最近の syslog（**messages***）

35.6 fsck

35.6.1 fsck の概要

fsck は、ファイルシステムの不整合をチェックし、修正するツールです。

このファイルシステム不整合の多くは、ハードウェアの障害や突然の電源断などにより、引き起こされます。

※fsck を実行することによりファイルシステム上に不整合が無い場合でも、ファイルデーターの完全性は保証されません、かならずバックアップを前提とした運用を行ってください。

fsck は、指定されたディスクパーティションのファイルシステムにより、自動的にファイルシステムに適合した修復プログラムを呼び出します。

例えば fsck のチェック対象に ext3 ファイルシステムを指定した場合は、fsck を実行すると e2fsck (fsck.ext3)プログラムが呼び出されて実行されます。

35.6.2 fsck のオプションについて

fsck ではその後に呼び出される e2fsck などのプログラムへオプションを渡します。

そのため、実行オプションの詳細を確認する場合は、man e2fsck など、実際のプログラムのオプションを確認してください。

ここでは、標準的な ext3 ファイルシステムの修復作業で使うと思われるオプションのみ紹介します。

-y 修正箇所が見つかった場合に実行者に確認せずに修正を行います。
このオプションを指定しないで fsck を実行した場合、修正箇所が見つかるたびに[y]キーで承認しなければならないため修正に時間と手間がかかります。通常はこのオプションの利用を推奨します。

-f ファイルシステムがクリーンな場合でも全てのチェックを行います。
障害発生時に修正する場合はこのオプションの利用を推奨します。

-p 起動時の fsck 実行時にデフォルトで指定されているオプションで、手動で実行する場合は利用しません。
起動時に手動(Manual)で fsck を実行するように指示の出た場合はこのオプションは利用しないでください。

-n チェックだけで全ての修正をしないまま終了します。事前に修正箇所の確認をする場合に利用しますが、見つかった箇所は修正以外に対応方法が無いため、通常はこのオプションは利用しません。

35.6.3 fsck 修正パーティションの指定

修正パーティションのデバイス名指定は、物理ディスク、論理ディスクなど、環境によって代わります。
ご利用の環境に合わせてデバイス名を指定します。

例えば/home の修正を行いたい場合は最初に mount コマンドで対象パーティションのデバイス名を確認します。

mount コマンドの実行例

```
# mount
/dev/mapper/VolGroup01-LogVol00 on / type ext3 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
/dev/sdb1 on /boot type ext3 (rw)
tmpfs on /dev/shm type tmpfs (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
none on /proc/fs/vmblock/mountPoint type vmblock (rw)
/dev/mapper/VolGroup00-LogVol00 on /home type ext3 (rw)
```

この場合は/home は[/dev/mapper/VolGroup00-LogVol00]がデバイス名となります。

35.6.4 fsck の実行例

※fsck は修正対象のパーティションを必ずマウント解除して行ってください。

fsck プログラム自身や関連ライブラリが含まれる/(root)、/usr ファイルシステムを修正する場合は、インストール DVD-ROM からレスキューモードで起動し、必ずマウント解除(unmount)状態で実行してください。

以下の例では/home パーティションのファイルシステムをチェックします。

（1）unmount コマンドを使用し、/home パーティションのマウントを解除します。

```
# umount /home
```

（2）fsck コマンドに、以下のように[-y]オプションを付加して実行します。これでも修正できない場合はさらに[-f]オプションを追加して、（3）を実行してください。

```
# fsck -y /dev/mapper/VolGroup00-LogVol00
```

（3）fsck コマンドに、以下のように [-f]オプションおよび [-y]オプションを付加して実行します。

```
# fsck -fy /dev/mapper/VolGroup00-LogVol00
```

fsck の実行結果は以下のように表示されます。

```
fsck 1.39 (29-May-2006)
e2fsck 1.39 (29-May-2006)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/mapper/VolGroup00-LogVol00: 200314/9125888 files (0.7% non-contiguous),
8705642/18235392 blocks
```

35.7 kdump の設定

kdump とは、カーネルダンプを取得するためのツールです。ダンプ情報をクラッシュしたカーネルから取得するのではなく、新しく起動したカーネルから取得するため、信頼性の高いカーネルダンプを取得することができます。

システムメモリから kdump 用に最低 64MB 以上確保するため、サーバーにシステムメモリを多く搭載してある必要があるでしょう。

kdump の設定を行うには、**system-config-kdump** コマンドを実行します。

```
# system-config-kdump
```

system-config-kdump コマンドを実行すると、図 35-12 のような画面が表示されます。

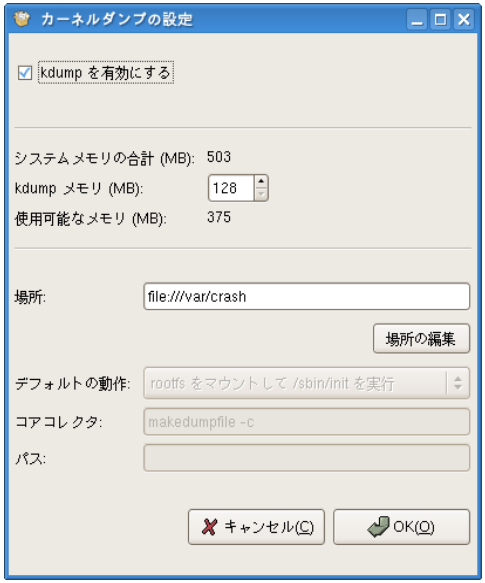


図 35-12 kdump 設定ツール画面

[kdump を有効にする]にチェックを入れ、kdump 用にリザーブするメモリの容量を調節します。[場所:]にはカーネルダンプの保存先を指定します。

必要な設定ができたら[OK(O)]をクリックします。

設定を反映させるためにシステムを再起動する必要があるという内容のダイアログが表示されるので、[OK(O)]をクリックした後、システムの再起動を行います。

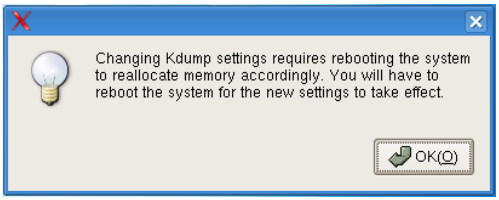


図 35-13 kdump 設定ツールダイアログ

再起動が完了したら、kdump が正しく動作するか確認するため、次のコマンドを実行して、故意にシステムをクラッシュさせます。

```
# echo c > /proc/sysrq-trigger
```

クラッシュ後、kdump カーネルが立ち上がり、クラッシュしたカーネルのダンプを記録した後、通常のカーネルで再起動します。カーネルのダンプは **system-config-kdump** で設定した保存場所に保存されています。

35.8 crash コマンド

crash コマンドは、kdump で取得したカーネルダンプや、現在のカーネルを解析するためのツールです。この項では、**crash** コマンドの使用方法について簡単に説明します。

35.8.1 kernel-debuginfo のインストール

crash コマンドを使用するためには、kernel-debuginfo および kernel-debuginfo-common パッケージが必要です。カーネルのバージョンと同じバージョンのパッケージをダウンロードするようにしてください。各パッケージは下記 URL からダウンロードすることができます。

```
最初のリリースに含まれているパッケージ
  http://ftp.miraclelinux.com/pub/Asianux/Server/3.0/Unsupported/
エラータで更新されたパッケージ
  http://ftp.miraclelinux.com/pub/Asianux/Server/3.0/updates/Unsupported/
```

ダウンロードが完了したら、**rpm** コマンドを使用してパッケージをインストールします。

```
# /bin/rpm -ivh kernel-debuginfo-common-*.rpm
# /bin/rpm -ivh kernel-debuginfo-*.rpm
```

35.8.2 crash コマンドの書式

crash コマンドの書式は、次の通りです。

```
# /usr/bin/crash [System.mapのパス] [vmlinuxのパス] [vmcoreのパス]
```

[vmcore のパス]については省略が可能です。省略した場合、現在稼動中のカーネルのメモリ空間を解析することができます。

```
# /usr/bin/crash /Boot/System.map-2.6.18-8.9AX /usr/lib/debug/lib/modules/2.6.18
-8.9AX/vmlinux
```

また、前項の kdump で作成したカーネルダンプを **crash** コマンドを使って開くには、次のようなコマンドを実行します。

```
# /usr/bin/crash /Boot/System.map-2.6.18-8.9AX
/usr/lib/debug/lib/modules/2.6.18-8.9AX/vmlinux
/var/crash/2007-08-19-15\:48/vmcore
```

35.8.3 解析コマンド

crash コマンドを実行すると、数秒ほど時間がかかって、次のようなメッセージが表示されます。

```
  SYSTEM MAP: /boot/System.map-2.6.18-8.9AX
DEBUG KERNEL: /usr/lib/debug/lib/modules/2.6.18-8.9AX/vmlinux (2.6.18-8.9AX)
   DUMPFILE: /var/crash/2007-08-19-15:48/vmcore
       CPUS: 1
       DATE: Sun Aug 19 15:48:11 2007
     UPTIME: 00:06:48
LOAD AVERAGE: 0.33, 0.31, 0.18
      TASKS: 76
   NODENAME: asianux.example.com
    RELEASE: 2.6.18-8.9AX
    VERSION: #1 SMP Tue Jul 31 08:04:24 EDT 2007
    MACHINE: i686  (2453 Mhz)
     MEMORY: 512 MB
      PANIC: "SysRq : Trigger a crashdump"
        PID: 1950
    COMMAND: "bash"
       TASK: ddabf000  [THREAD_INFO: d3ac6000]
        CPU: 0
      STATE: TASK_RUNNING (SYSRQ)
```

これらの情報は、カーネルがクラッシュした時点のものです。「PANIC」という行が、カーネルパニックを起こした内容になります。

crash コマンド実行中はプロンプトが立ち上がっているので、専用のコマンドを入力して、解析作業を行います。ここでは、代表的なコマンドをいくつか説明します。その他のコマンドについては、**help** コマンドを入力して参照してください。

- **log (dmesg)** コマンド
  カーネルメッセージを表示します。
- **ps** コマンド
  カーネルが停止した当時のプロセス一覧を表示します。
- **bt** コマンド
  カーネルが停止した箇所の実行パスを表示します。ps コマンドで表示されるプロセス ID を引数として指定することもできます。
- **files** コマンド
  引数で指定したプロセス ID がオープンしていたファイルの一覧を表示します。
- **mod** コマンド
  読み込み済みのモジュールの一覧を表示します。
- **mount** コマンド
  マウント情報を表示します。
- **irq** コマンド
  割り込み情報を表示します。

35.9 障害対応

35.9.1 障害の詳細情報の取得

万が一障害が発生した場合、適切に対応するためには、まず情報を集める必要があります。トラブルシューティングに役に立つ情報には次のようなものがあります。

- 障害の情報
  - 現象の内容、何ができて、何ができないのか、再現性、発生頻度
  - コンソールに表示されるエラーメッセージ
  - エラーログ（**/var/log/アプリケーション名/*.log**）
  - Oracle のアラートログ（**alert_${SID}.log**）
- 現象再現手順
  - どのような状態で、何をすれば発生するのか
- 障害発生環境の情報
  - ハードウェアの構成
  - OS の環境（バージョン、カーネル、glibc のバージョンなど）
  - アプリケーションの種類、バージョン

35.9.2 一般的な Linux 環境の調査

トラブルシューティングでは、システムの環境に関する情報も重要な要素となります。

- **/var/log/messages**

  syslog ファイルは、**/var/log/messages** に記録されます。

- **dmesg**

  dmesg は、起動時のメッセージを取得するコマンドです。詳細は、**dmesg** のオンラインマニュアルを参照してください。

- **lsmod**

  lsmod は、現在ロードされているモジュールの一覧を取得するコマンドです。**/proc/modules** と同じ内容です。詳細は、**lsmod** のオンラインマニュアルを参照してください。

```
# /sbin/lsmod
Module                  Size  Used by
md5                     3968  1
ipv6                  232768  12
i2c_dev                11392  0
i2c_core               22400  1 i2c_dev
vmhgfs                 45360  4
iptable_filter          2688  0
ip_tables              16640  1 iptable_filter
dm_mod                 56596  0
button                  6416  0
battery                 8836  0
ac                      4740  0
uhci_hcd               34072  0
snd_ens1371            31528  0
snd_rawmidi            26788  1 snd_ens1371
```

- **lspci**

  lspci は、すべての PCI デバイスの情報を表示します。詳細は、**lspci** のオンラインマニュアルを参照してください。

```
# /sbin/lspci
00:00.0 Host bridge: Intel Corp. 82440MX Host Bridge (rev 01)
00:07.0 ISA bridge: Intel Corp. 82440MX ISA Bridge (rev 01)
00:07.1 IDE interface: Intel Corp. 82440MX EIDE Controller
00:07.2 USB Controller: Intel Corp. 82440MX USB Universal Host Controller
00:07.3 Bridge: Intel Corp. 82440MX Power Management Controller
00:09.0 Modem: PCTel Inc HSP MicroModem 56 (rev 02)
00:0b.0 Ethernet controller: Intel Corp. 82557/8/9 [Ethernet Pro 100] (rev 08)
```

- **free**

  free は、システムの空きメモリと利用メモリの量を表示します。詳細は、**free** のオンラインマニュアルを参照してください。

```
# /usr/bin/free
             total       used       free     shared    buffers     cached
Mem:        190216     185808       4408          0       4780     110276
-/+ buffers/cache:      70752     119464
Swap:       385552      94168     291384
```

- **uname**

  uname コマンドは、システムの情報を表示します。オプションの**-r** を使用することで、現在のカーネルバージョンを調べることができます。詳細は、**uname** のオンラインマニュアルを参照してください。

```
# /bin/uname -r
2.6.18-8.9AX
```

- **top**

  top は、現在の実行しているプロセスの活動推移を調べるコマンドです。CPU／メモリの使用率を調べたり、CPU／メモリを多く使用しているプロセスを特定したりできます。詳細は **top** のオンラインマニュアルを参照してください。

```
top - 02:49:55 up 18 min,  2 users,  load average: 1.32, 1.19, 0.78
Tasks:  46 total,   2 running,  44 sleeping,   0 stopped,   0 zombie
Cpu(s): 94.9% us,  2.2% sy,  0.0% ni,  0.0% id,  0.0% wa,  0.3% hi,  2.5% si
Mem:    251908k total,   243488k used,     8420k free,    13364k buffers
Swap:  1052248k total,      624k used,  1051624k free,   179920k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
10035 root      25   0 11152 6776  484 R 94.7  2.7   0:10.58 prelink
 7397 tama      16   0 10452 2388 1908 S  3.5  0.9   0:00.67 sshd
 3755 root      16   0  8604 5236 1648 S  0.6  2.1   0:06.84 hald
10071 root      17   0  3020  908  740 R  0.6  0.4   0:00.12 top
    1 root      16   0  2868  544  464 S  0.0  0.2   0:01.14 init
    2 root      34  19     0    0    0 S  0.0  0.0   0:00.03 ksoftirqd/0
    3 root       5 -10     0    0    0 S  0.0  0.0   0:00.15 events/0
    4 root       5 -10     0    0    0 S  0.0  0.0   0:00.13 khelper
    5 root      15 -10     0    0    0 S  0.0  0.0   0:00.00 kacpid
   18 root       5 -10     0    0    0 S  0.0  0.0   0:01.64 kblockd/0
```

35.9.3 障害原因の特定と解決

障害の情報や環境の情報がまとまったら、障害原因の特定／解決を行います。

- **既存のバグの有無**

  弊社の Asianux Technical Support Network から既存のバグの有無を調査します。

  https://tsn.miraclelinux.com/tsn_local/

- **ソースコードの調査**

  障害の対象パッケージのソースコードより障害調査を行います。各ソースパッケージは、添付の Asianux Server 3 Source DVD あるいは、次の URL から取得できます。

  http://ftp.miraclelinux.com/pub/Asianux/Server/3.0/updates/src/

- **サポートの利用**

  Asianux Server 3 には購入した製品によって、初年度、「ベーシック」または「スタンダード」のサポートサービスが製品についています。「スタンダード」を購入されている場合は、障害原因の特定／解決の支援を受けることができますので御活用ください。

  サポートサービスに関する詳細情報については下記 URL をご参照ください。

  http://www.miraclelinux.com/service/support/linux/axs3.html

索引

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

さ

た

な

は

ま

や

ら

Asianux Server 3 サーバー構築・運用ガイド
2007 年 9 月 18 日　初版発行
2008 年 9 月 12 日　第二版発行
2009 年 9 月 11 日　第三版発行
発行　ミラクル・リナックス株式会社