Linux
Professional
Institute

# Linuxサーバー構築標準教科書

（Ver.2.0.1）

LPI-JAPAN

# Linux サーバー構築標準教科書

2012-06-20 版　　LPI-Japan　発行

# まえがき

このたび、特定非営利活動法人エルピーアイジャパンは、Linux 技術者教育に利用していただくことを目的とした教材、「Linux サーバー構築標準教科書」を開発し、インターネット上にて公開し、提供することとなりました。

この「Linux サーバー構築標準教科書」は、多くの教育機関から、Linux によるサーバーの構築を「基礎」から学習するための教材や学習環境の整備に対するご要望があり、開発したものです。

公開にあたっては、「Linux サーバー構築標準教科書」に添付されたライセンス（クリエイティブ・コモンズ・ライセンス）の下に公開されています。

本教科書は、最新の技術動向に対応するため、随時アップデートを行っていきます。また、テキスト作成やアップデートについては、意見交換のメーリングリストで、誰でもオープンに参加できます。

メーリングリストの URL http://list.ospn.jp/mailman/listinfo/linux-text

## 執筆者・制作者紹介


### 岡田 賢治（バージョン 1 執筆担当）

UNIX/Linux を初めて触ってから 15 年、ユーザー、管理者そして育成に携わってきました。本テキストは、実際に用いた説明方法などを使い、今までのノウハウを集約して執筆したつもりです。LPIC を通しての Linux 技術者の発展と、育成に関わる先生方のお役に立てれば幸いです。

### 川井 義治（バージョン 1 執筆担当）

Linux の講義をするとき、多肢に渡る知識と解説が必要で、前後の内容が複雑に絡み合うむため、教材選定に苦労したり、自作教材を使っていました。多くの内容の中から実践として使える項目を選び、使いやすい並びを目指して書きました。学校教材の選択肢の一つや、個人教材として使って頂ければ幸いです。

### 宮原 徹（バージョン 2 執筆担当）

本教科書は、Linux/オープンソースソフトウェアをこれから勉強する皆さんと、熱心に指導に当たられている先生方の一助になればと思い、作成いたしました。バージョン 2 の改訂にあたっては、新しいディストリビューションへの対応だけでなく、より分かりやすい実習になるように調整を行ってみました。

遠山 洋平（校正・図版作成担当）

Linux サーバー構築標準教科書のバージョン 1 のリリースから 3 年がたちました。本教科書は CentOS6.2 を使ったサーバー構築のノウハウが集約されています。これからサーバー構築を実践してみたい！ そんなチャレンジ精神のある皆さんのお役に立てれば幸いです。

田口 貴久（バージョン 2 技術検証担当）

改版にあたり、わかりやすい教科書になるよう、構築でつまづきやすい部分を重点的に検証いたしました。Linux 技術者を目指す方のお役に立てれば幸いです。

高橋 征義（PDF ／ EPUB 版制作担当）

本教科書の PDF・EPUB 作成のお手伝いをいたしました。サーバーエンジニアの方々の技術力向上に貢献できれば幸いです。

## 著作権




## 使用に関する権利

●表示

本教科書は、特定非営利活動法人エルピーアイジャパンに著作権が帰属するものであることを表示してください。

●改変禁止

本教科書は、改変せず使用してください。本教科書に対する改変は、特定非営利活動法人エルピーアイジャパンまたは特定非営利活動法人エルピーアイジャパンが認める団体により行われています。フィードバックは誰でも参加できるメーリングリストで行われていますので、積極的にご参加ください。http://list.ospn.jp/mailman/listinfo/linux-text

●非営利

本教科書は、営利目的（※）以外で教材として自由に利用することができます。営利目的での利用は、特定非営利活動法人エルピーアイジャパンによる許諾が必要です。本教科書を利用した教育において、本教科書自体の対価を請求しない場合は、営利目的の教育であっても基本的に使用できます。その場合も含め、LPI-Japan 事務局までお気軽にお問い合わせください。

（※）営利目的の利用とは以下のとおり規定しております。

- 営利企業において、本教科書の複製を用いた研修や講義を行うこと、または非営利団体において有料セミナー等に利用すること

本教科書の使用に関するお問合せ先
特定非営利活動法人エルピーアイジャパン（LPI-Japan）事務局
〒102-0082　東京都千代田区一番町 15　一番町コート 6F
TEL：03-3261-3660
FAX：03-3261-3661
E-Mail：info@lpi.or.jp

## 本教科書の目的

本教科書の目的は、LPIC レベル 2 の 201 試験と 202 試験の学習範囲に含まれるサーバー構築の知識を、構築の実習を通しながら学習することにあります。サーバーを構築した環境で、実際に Web アクセスをしたりメールの送受信をしたりすることで、サーバーの動作原理やプロトコルの仕組みを理解することも可能です

## 想定している実習環境

本教科書での実習環境として、以下の環境を構築しています。

**●講師と受講生**

講師 1 名と受講生が 2 名以上存在すること前提とします。これは、実習のなかで受講生同士 2 名でペアを組み、お互いの設定したサーバーにアクセスをする作業を行うためです。

**●教室と割当**

実習はコンピューター実習室のような教室で行うことを想定してます。講師の指示に従いながら、受講生が実習を行う形式になります。マシンは、講師、受講生に各名 1 台ずつを想定しています。

**●１名で学習する場合**

1 名で学習する場合は、マシンは最低 3 台必要になります。講師用マシン 1 台と受講生用マシン 2 台です。後述する仮想マシン環境を活用すれば、1 台で実習を行うことも可能です。

**●仮想マシン環境**

仮想マシン環境を利用すると、Windows や Mac OS X 上の仮想マシンに Linux をインストールし、動作させることができます。仮想マシンは複数同時に動作させることができるので、3 台必要となる実習環境を 1 台のマシンでまかなうこともできます。仮想マシン環境を実現するソフトウェアとして、たとえば VMware 社の VMware Workstation(Windows) や VMware Fusion(Mac OS X)、Parallels 社の Parallels Desktop(Mac OS X) や Oracle 社の VirtualBox（Windows、Linux、Mac OS X）などが挙げられます。

●マシンの構成とハードディスク

マシンの構成は、市販されている一般的な構成の PC を想定しています。その PC に Linux をインストールします。ハードディスクの内容は完全に消去されるので、ハードディスクの内容を消去しても良いマシンを用意するか、必要に応じてハードディスクの内容をあらかじめバックアップしておく必要があります。

● OS

本教科書では、Linux ディストリビューションとして CentOS のバージョン 6.2（32 ビット版）を利用します。

●ネットワーク

実習で利用するマシンは、すべて 1 つのネットワークで接続されていることを前提とします。インターネットへの接続は任意です。

## 全体の流れ

本教科書では、以下の通りに実習を進めます。

- 1 章 Linux のインストール準備と事前学習
- 2 章 Linux のインストールと設定を行う
- 3 章 ネットワークの設定と確認を行う
- 4 章 DNS サーバーのインストールと設定を行う
- 5 章 Web サーバーのインストールと設定を行う
- 6 章 メールサーバーのインストールと設定を行う
- 7 章 ファイルサーバーのインストールと設定を行う
- 8 章 サーバーのセキュリティの設定を行う

# 目次

# 第1章

# Linuxのインストール準備と事前学習

本教科書では、Linuxをインストールし、サーバー環境を構築する実習をしながら、LPICレベル2の範囲の理解と取得を目指します。第1章では、2章以降に行う実習に必要な環境の確認と知識の確認を行います。

## 1.1 用語集

### Linux

Linus Torvalds氏により開発された、UNIX互換を目指したOSの総称をLinuxといいます。ソースコードは公開されており、世界中の開発者の協力により、日々開発が継続されています。

### ディストリビューション

Linuxは狭い意味ではOSの中心部（＝カーネル）のみを指しますが、Linuxカーネルだけではシステムは動作しません。カーネル以外のさまざまなソフトウェアやインストーラを追加して、利用できるようにしたのがディストリビューションです。ディストリビューションごとに開発方針があり、それに沿ってソフトウェアがまとめられたり、リリースが行われています。

### CentOS

Linuxのディストリビューションの1つです。Red Hat Enterprise Linuxという商用のディストリビューション互換の環境を無償で提供しているディストリビューションで、CentOSコミュニティによってリリースされています。

### ネットワークアドレス

IPネットワークを小さく分割して利用するときに、ネットワークアドレス部とホストアドレス部に分かれます。ネットワークアドレスの識別に利用されるのが、ネットワークアドレス部です。

### IPアドレス

インターネットにおいて、IPで通信が行われる場合、端末一つ一つにIPアドレスが割り当てられます。IPアドレスとはインターネット上での端末の所在地を示す”住所”にあたります。

### サブネットマスク

IP アドレスのうちネットワークアドレスとホストアドレスを識別するための数値のことをいいます。通常 8 ビット毎に.(ドット) で区切って入力されます。

### DNS サーバーアドレス

IP アドレスと FQDN（＝ホスト名＋ドメイン名）の変換を行うのが DNS サービスであり、そのサービスを提供する DNS サーバーの IP アドレスのことです。

### ホスト名

ネットワークに接続されたコンピューターに割り当てられた名称のことをいいます。特定のホストを識別するために使われます。

### ドメイン名

インターネット上に存在するコンピューターやネットワークを識別するために付けられている名前の一種のことをいいます。ドメイン名はアルファベット、数字、一部の記号の組み合わせで構成されますが、日本語.jp のような国際化ドメインも使われるようになっています。

### DVD

光学メディアの 1 種類で、ビデオ再生での利用で普及し、現在ではデータ記録の用途でも利用されています。約 700MB の CD-ROM に比べ、約 4.7GB と大容量でも利用できることから、OS のインストールディスクとしても利用されています。

### ハードディスク

磁気を用いた記憶媒体であり、パソコンの記憶媒体の他、音楽プレーヤー、ビデオなどの記憶媒体としても用いられています。

### SSD

半導体メモリであるフラッシュメモリを用いた記憶媒体であり、ハードディスクよりも読み書きの処理性能に優れています。

### LVM

LVM（Logical Volume Manager）とは，複数のハードディスクやパーティションにまたがった記憶領域を一つの論理的なディスクとして扱うことのできるディスク管理の機能のことです。Linux をはじめとした UNIX 系 OS 上で利用できます。

### RAID

複数のハードディスクをまとめて 1 台のハードディスクとして管理する技術のことです。RAID を使うことによりデータを分散して記録するため、高速化や安全性の向上が期待できます。RAID の方法には、専用のハードウェアを使う方法（ハードウェア RAID）とソフトウェアで実現する方

法（ソフトウェア RAID）があり、高速性や安全性のレベルにより、RAID 0 や RAID 5 などいくつかの種類があります。

## 1.2 実習で利用するハードウェア

本教科書の実習では、市販されているような一般的な構成の PC に Linux を導入して、その環境上に様々なサーバーを導入し実際に動作させます。この実習で必要な、ハードウェアの仕様は次の通りです。

### マシン本体

Windows や Linux が動作する、いわゆる「パソコン」を想定しています。また、「VirtualBox」のような仮想マシンソフトウェアを利用して実習環境を用意することもできます。

### 実装メモリ

CentOS 6 では 1024MB のメモリが推奨されています。実装メモリが少ない場合はテキストインストールが実行されることがあります。

### DVD 光学ドライブ

本教科書の実習では、インストール用 DVD を利用するので、その光学ドライブが DVD を読み取りできる必要があります。また一部のノートパソコン等で、光学ドライブが無いマシンもあります。そういったときは、USB 等で接続する DVD ドライブを用意します。それを利用することにより、インストール DVD を起動することができます。

### ハードディスク

Linux をインストールするためには記憶装置が必要です。ここでは記憶装置としてハードディスクを使います。インストールにはハードディスクに約 10GB の空き領域があれば十分なので、一般的な構成の PC では十分満たしていると思います。またハードディスクをフォーマット（初期化）して Linux をインストールします。従って、ハードディスクの中身は全部消去されます。その為、ハードディスクを削除していい PC を利用するか、バックアップを取ってから作業を行ってください。

### その他周辺機器

本体や DVD、光学ドライブ、ハードディスクドライブの他にも、一般的に利用するためにはキーボード、マウス、ディスプレイ等の周辺機器が必要です。キーボードは、日本語か英語かで設定が異なりますので、日本語キーボード、英語キーボードの区別を「確認シート」に記述します。

## 1.3 利用する Linux のディストリビューション

本教科書では、CentOS のバージョン 6.2、32 ビット版を利用します。

CentOS 公式サイト

http://www.centos.org/

CentOS は、商用ディストリビューションである Red Hat Enterprise Linux の互換ディストリビューションとして提供されています。本家 Red Hat とのバイナリ互換を保ちながら、サポートも同等を目指すという方針で開発されています。利用に際し費用が発生することはない、無償で提供されているディストリビューションです。

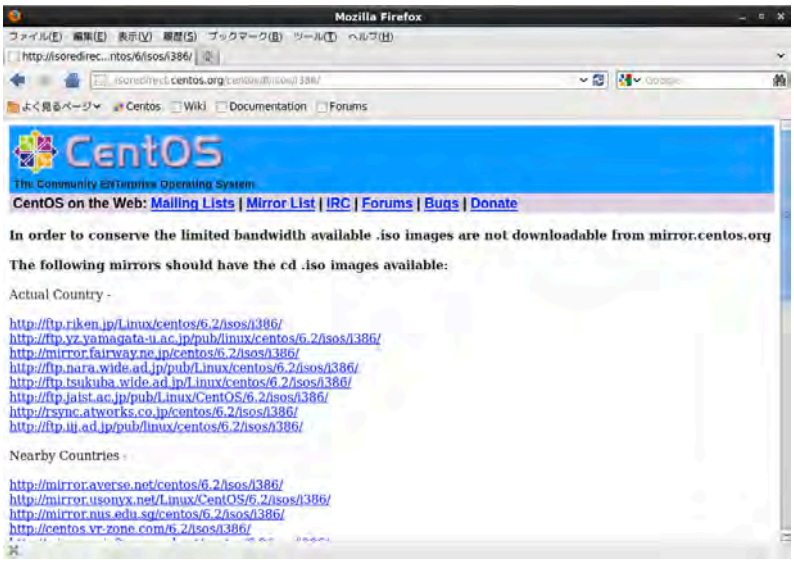
### 1.3.1 インストール DVD の入手方法

今回インストールには、DVD のインストーラーを利用します。CentOS のインストール用 DVD の入手方法には次の 2 通りが存在します。

● ISO イメージをダウンロードする

CentOS が配布している ISO イメージを、ダウンロードします。ダウンロード元の URL は以下のとおりです。CentOS の最新版の、ISO イメージのページへのリンクになっています。

http://isoredirect.centos.org/centos/6.2/isos/i386/




この URL をクリックすると、多くのミラーサイトが表示されます。その中でバージョン 6.2 の ISO イメージをダウンロードします。例えば、riken（理化学研究所）が提供しているミラーサイトの URL であれば次のようになります。

ダウンロードサイト

```
http://ftp.riken.jp/Linux/centos/6.2/isos/i386/
```

ダウンロードするインストール DVD の ISO イメージ

```
http://ftp.riken.jp/Linux/centos/6.2/isos/i386/CentOS-6.2-i386-bin-DVD1.iso
```

DVD イメージは 2 枚組になっていますが、今回の実習では 1 枚目の DVD のみ使いますので CentOS-6.2-i386-bin-DVD1.iso をダウンロードします。ISO イメージは合計で 3.6GB あり、転送に時間がかかります。ミラーサイトによっては、複数枚の CD の ISO イメージはあるものの、DVD の ISO イメージを置いていないサイトも存在しますので、ダウンロードするサイトを選ぶときは注意してください。

また、BitTorrent を使ってのダウンロードも行えます。ダウンロードサイトに.torrent という拡張子のファイル名が置かれているので、このファイルをダウンロード後、BitTorrent に対応したソフトウェアを使ってダウンロードが行えます。

ダウンロードした ISO イメージは、DVD のライティングソフトウェアを使って DVD に書き込んでください。データとしてではなく、イメージとして書き込む点に注意してください。

**●雑誌や解説書の付録**

CentOS は雑誌に付属していたり、CentOS の解説書が多く出版されています。それらに付属しているインストール DVD を利用してもかまいません。

### 1.3.2 バージョン

本教科書では、本教科書の作成時点で最新であった CentOS 6.2 を利用した構築方法について解説しています。雑誌や解説書などの付録など、入手の方法によっては 6.2 ではない、より新しいバージョンの CentOS を手にすることがあるかもしれません。しかし、バージョン 6.x 系であれば大きな差は無いようです。従って CentOS の 6.x 系であれば、rpm コマンドを使ったパッケージの追加時のファイル名などに注意する必要はあるものの、同様の手順でサーバーの構築ができるようになっています。

## 1.4 ネットワーク環境について

本教科書での実習ではネットワークを利用します。ネットワークの設定項目は複数ありますので、あらかじめ別紙「確認シート」を作成した上で設定を行いましょう。

利用するネットワークの各種設定情報が組織のネットワーク管理担当者から指示されている場合は、その内容を「確認シート」に記述します。本教科書では特定の IP アドレスを用いて設定しますが、それを適宜指示された内容に読み替えてください。

ネットワークを自由に設定できる場合は、本教科書で用いているネットワークの設定を利用して下さい。本教科書では、ネットワーク環境としてコンピューター実習教室を想定しています。教室には PC が 3 台以上あり、講師用 PC が 1 台と受講生用 PC が 2 台以上あるとします。ネットワークは、講師用、受講生用 PC の区別無く、すべての PC が 1 つのネットワークに接続されているこ

とを想定しています。

### 1.4.1 ネットワークの設定項目

ネットワークの設定には、ドメイン名やホスト名、IP アドレスなど、いくつかの設定項目が必要です。

#### ドメイン名

ドメイン名は、DNS サーバーを設定するときに必要になります。受講生同士が同じドメイン名にならなければ、各自自由なドメイン名をつけてかまいません。このドメイン名は、あくまでこのネットワーク内のみで有効なドメイン名で、外部の DNS とは隔離された状態にあります。本教科書では、受講生用に alpha.jp と beta.jp の 2 つを使用します。

#### ホスト名

自分の PC に設定するホスト名です。今回は host ＋受講生番号とします。確認のため「確認シート」に記述します。

#### IP アドレス

IP アドレスは、PC の IP アドレスです。本教科書では、講師用 PC の IP アドレスを 192.168.1.10、受講生用 PC の IP アドレスを 192.168.1.101 と 192.168.1.102 としています。

#### サブネットマスク

サブネットマスクは、IP アドレスのネットワーク部とホスト部を分ける値です。本教科書では 255.255.255.0(/24) とします。

#### ネットワークアドレス

ネットワークアドレスは、PC が含まれているネットワーク全体を示すアドレスです。本教科書では 192.168.1.0 とします。

#### デフォルトゲートウエイ

異なるサブネットとの通信に必要な値です。本教科書では 192.168.1.1 とします。

#### DNS サーバーアドレス

ホスト名と IP アドレスの対応を解決する、DNS（ドメインネームシステム）という機構があります。DNS を利用するためには DNS サーバーの IP アドレスが必要です。本教科書では 4 章で、実際に DNS サーバーを設定し動作させます。実習ではまず自分自身で動作させている DNS サーバーを参照するため、DNS サーバーアドレスを自分の IP アドレスとします。

## 1.5　高度なストレージ管理

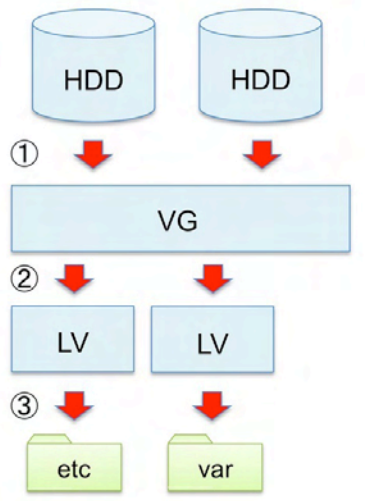
ここでは、高度なストレージ管理として LVM(Logical Volume Manager) と RAID(Redundant Arrays of Inexpensive Disks) について説明します。インストールを開始すると、LVM の設定が施されるところがあり、それについての説明です。少々高度な内容になるため、インストール作業後に読んでもかまいません。

### 1.5.1　LVM

Logical Volume Manager（LVM）という機構は、一言で言えば「ディスク管理操作を非常に便利にしてくれる機構」と言えます。ハードディスクを利用する際には、いくつかの「パーティション」に分割します。Windows のみならず、Linux でもパーティションを分割する作業を行います。パーティション分割作業は、容量を決めるのに非常に困難が伴います。「思ったより利用が多く、足りなくなってしまった」「念のため多めにパーティションを割り当てたら、あまり利用されず、大半が未使用になってしまった」といった具合です。だからといって、パーティションを再分割することは非常に手間がかかります。ハードディスクの内容を、一度全部消してしまうからです。再度インストール作業を行なった後、設定を行い、データを復元する作業は、相当な時間や手間を使います。LVM を用いると、パーティションを柔軟に取り扱うことができます。

### 1.5.2　LVM の仕組み

LVM の仕組みは、次のようになっています。




流れ：
① 複数パーティションをボリュームグループ（VG）にまとめる
② VGから論理ボリューム（LV）に切り出す
③ 論理ボリュームを初期化し、各ディレクトリにマウントして利用する

PV(Physical Volume)

ディスクの物理領域です。パーティションの 1 区画であったり、ディスク 1 台丸ごと PV ということもありえます。

VG(Volume Group)

一つ以上の PV の集まりが Volume Group です。

LV(Logical Volume)

VG から、LV 領域を切り出して利用します。LV は自由に容量を増やしたり減らしたりできます。LV の容量の合計が、切り出し元の VG より大きくなることはありません。LV の領域にファイルシステムを作成して、ファイルやディレクトリなどのデータが格納されます。

### 1.5.3 LVM の利点

LVM を用いると、どんな点が便利なのでしょうか?実際にケーススタディで学習してみましょう。

ディスク領域が不足した場合にディスク容量の増加が容易

LV に保存したデータが増加し、割り当てた LV の容量では足りなくなった場合は、LV の大きさを増やすことで対応可能です。使用しているファイルシステムによっては、OS を止めることなく容量を増やすこともできます。逆に LV を大きく切り出しすぎて余ってしまった場合には、ファイルシステムを縮めた後に LV を小さくします。これで VG の未使用領域が増えるので、他の LV を増やしたり、新しく LV を切り出したりするときに利用できます。

ハードディスクを増設するのも容易

LV の利用率も増え、その元である VG の空き容量が少なくなったとします。そのときは、ハードディスクを増設しますが、LVM を用いると作業は簡単です。ハードディスクを取り付けて、そのハードディスクを PV とします。その PV を VG に追加すると、VG 全体の容量が増えます。増えた VG から新しく LV を切り出したり、既存の LV のサイズを増やしたりすることに利用できる領域を増やすことができます。

## 1.6 RAID

### 1.6.1 RAID とは

RAID とは Redundant Arrays of Inexpensive Disks の略で、ディスクの耐障害性を高めたり、機能を高めたりすることに用いられます。ディスクのアクセス性能を上げたい場合や、重要なデータを置いておく場合に利用します。

### 1.6.2 RAID の種類

RAID にはその機能でさまざまな種類があります。ここでは広く用いられる RAID 0,1,5 について説明します。

#### RAID 0(ストライピング)

2 台以上のディスクを用意し、書き込み時にそれぞれのディスクに分散書き込みを行います。ディスクの処理が分散するので、読み書きの速度が高速になります。また、使用できるディスク容量はすべてのディスクの容量の合計となります。欠点は、1 台でもディスクが壊れるとすべてのデータが読み書きできなくなることです。

#### RAID 1(ミラーリング)

ディスクを 2 台用意し、それぞれに同じ内容を書き込みます。一方のディスクが壊れてももう一方のディスクが正常であれば、データは失われませんので、ディスク障害に強い構成を実現できます。欠点は、利用できる容量が総容量の半分になってしまうことです。例えば容量 500GB のディスクを 2 台用意しても、使用できる容量は 500GB のままです。

#### RAID 5(パリティ分散)

ディスクを 3 台以上用意し、パリティという特別な仕組みを一緒に書き込むことでディスクの冗長化を図っています。ディスクが 1 台故障してもデータを失うことはありません。RAID 5 は 1 台あたりのディスク容量　×　(台数-1) の容量が使えますので、ディスクの利用効率も良いことになります。たとえば 500GB のディスクを 3 台用意すれば、合計 1TB のディスク容量になります。欠点は、データ書き込み時のパリティ計算の負荷が高いため書き込み性能が高くないことや、故障に耐えられるディスクが 1 台までなので、運悪く 2 台以上同時に壊れると元データの復元ができないといった点が挙げられます。

#### RAID 6

パリティを 2 重に計算し書き込むことで、ディスクが 2 台まで故障しても大丈夫にした RAID 構成です。欠点は、より高度なパリティ計算を高速に行うために専用のハードウェアが必要となる点です。

#### RAID1+0

RAID 1+0 はミラーリング (RAID 1) したディスクをストライピング (RAID 0) する RAID の構成です。ストライピングはディスクが 1 台でも壊れるとすべてのデータが失われてしまいますが、RAID 1+0 ではミラーリングが行われているので、ディスクが 1 台壊れてもデータは失われません。ただし、ミラーリングされた両方のディスクが壊れてしまうと、通常の RAID 0 と同様にデータは失われてしまいます。欠点は、ディスクが最低でも 4 台必要であることと、ミラーリングされているため容量が半分になってしまうという点です。

#### その他の RAID

RAID には、他にも 2,3,4 がありますが、あまり使われていません。

### 1.6.3 ハードウェア RAID とソフトウェア RAID

RAID ではハードウェア RAID とソフトウェア RAID が存在します。

#### ハードウェア RAID

ハードウェア RAID は、RAID の処理をハードウェアが行います。従って、OS、マザーボード側から見ると、ディスクが 1 台存在しているように見えるだけです。RAID コントローラは OS、マザーボードにディスクが 1 台と「見せかけながら」、その背後で RAID の処理を行っています。ハードウェア RAID を使う利点は、OS は一般的なハードディスクとして認識されるために特別なドライバーがを導入する必要がないことと、OS やハードウェアに負荷がかからないことです。欠点として特別なハードウェア（RAID コントローラ）が必要であるため、費用がかかる点が挙げられます。

#### ソフトウェア RAID

ソフトウェア RAID は、OS やドライバーが RAID 作業を行います。ソフトウェア RAID は特別なハードウェア（RAID コントローラ）が必要ではないため、コストを抑えて RAID を組むことができますが、欠点として OS の対応やドライバーの対応が必要であることや、ハードウェア RAID と比べて OS、ハードウェア（特に CPU）に負荷がかかる点が挙げられます。

### 1.6.4 高度なストレージの利用

高度なストレージとして、LVM と RAID を紹介しました。ではどのような場面で利用するのが好ましいでしょうか？

Linux では（後に紹介しますが)、ディスクを使用するためにパーティションに対して特定のディレクトリをマウントさせます。デフォルトの構成ではパーティションが一つ作成され、そこにすべてのディレクトリの大元となるルートディレクトリ（”/”）がマウントされ、そのサブディレクトリとして/var や/home といったディレクトリが作成されます。

/var には、ログファイルなどシステムが様々なデータを書き出します。/home は、ユーザーが作成したデータが置かれます。この 2 つのディレクトリは非常に重要であり、なおかつ利用量が非常に変化しやすいディレクトリなので、このようなディレクトリはパーティションを分け、ルートディレクトリと切り離してマウントし、そのパーティションを LVM を用いて可変としたり、RAID を用いて冗長化されるよう構成することが望ましいと言えます。

# 第2章

# Linux のインストール

本章では、実際に Linux のインストールと設定を行います。ネットワークの設定や、インストールするソフトウェアの選択など、次章以降に影響する重要な内容ですので、しっかり学習しましょう。

## 2.1 用語集

### メディアの整合性

作成されたメディアが、配布されたオリジナルの内容と違いが無いかどうかはメディアの整合性が取れているかどうかで確認できます。何らかの原因により整合性が取れていない場合、ソフトウェアのインストールに失敗してしまいます。CentOS ではインストール手順の開始時、メディアの整合性が取れているかどうかチェックが行えるようになっています。

### BIOS

PC の周辺機器を制御するプログラムのことをいいます。PC には必ずこの BIOS が内蔵され、BIOS が起動後、OS が起動します。内蔵の時計や、起動デバイスの選択等を設定できます。設定は、マザーボード上のフラッシュメモリに保存されています。

### 起動順序

どの記憶装置から OS を起動するか、起動デバイスの優先順位をつけることをいいます。BIOS で設定することができます。ハードディスクの他、CD/DVD 等の光学ドライブ、USB のストレージデバイス、FDD 等を選ぶことができます。

### Timezone/時間帯

Linux の動作時に時刻を設定します。通常の時刻を設定するほか、そのマシンが起動している場所の時間帯を設定できます。日本で動作させるときは、日本標準時 (=JST) に設定します。

### フォーマット

ハードディスク等を OS で読み書きできる状態にすることで初期化ともいいます。フォーマットを実行すると、ディスクのデータはすべて削除されます。

#### ファイアウォール

インターネットにコンピューターを直接接続すると不正にアクセスされるおそれがあるため、ファイアウォールを構築します。ファイアウォールを動作させることで、ネットワークのセキュリティ機能を高めることができます。通常の利用では有効化することが推奨されます。

#### SELinux

Linux 上に特別なセキュリティ機能を導入し、Linux の標準機能よりも高度なセキュリティを機能させることができます。ファイアウォール同様、有効化することが推奨されます。

## 2.2 インストールの前に用意するもの

#### 確認シート

インストールの前に、1 章で記入した「確認シート」を手元に用意します。この内容を確認しながら、インストール作業を行います。

#### インストール DVD

CentOS 6.2 のインストール DVD を用意します。

#### マシンの設定

インストールを開始するにあたり、マシンの設定を確認します。確認する内容は、BIOS で設定する「起動順序」です。起動順序を、必ず光学ドライブ優先にします。光学ドライブよりハードディスクの優先度が高いと、ハードディスクにインストールされている OS が起動してしまいます。

#### ハードディスク

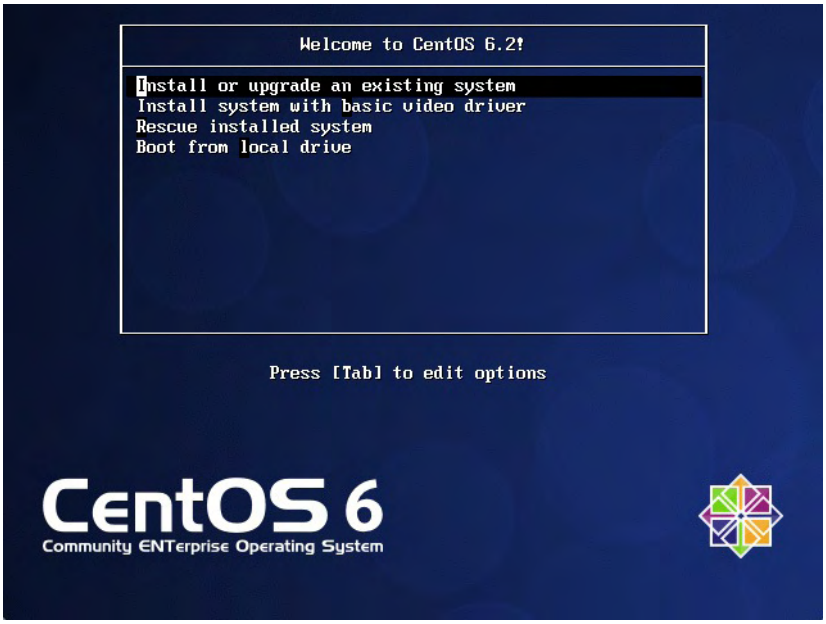
本教科書では、CentOS をインストールする際にハードディスクの中身を消去します。従ってハードディスクの中身を消去しても良い PC を利用するか、ハードディスクの中身のバックアップを取ってから作業を行います。

## 2.3 インストールの開始

それでは、インストールを開始します。

1. インストール DVD を光学式ドライブにセットし、マシンを起動します。

2. 起動画面が現れます。「Install or upgrade an existing system」と表示されるので、Enter キーを押します。60 秒間そのままにしても、自動的にインストールが開始します。

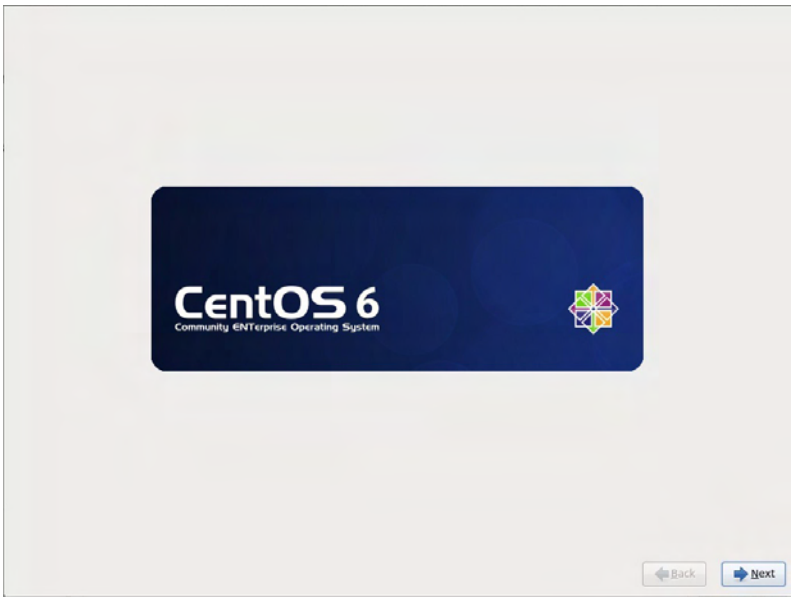



一部のマシンでは内蔵されているグラフィック性能の不足のため、正しくインストーラが起動しないことがあります。この場合は「Install system with basic video driver」を選択してください。

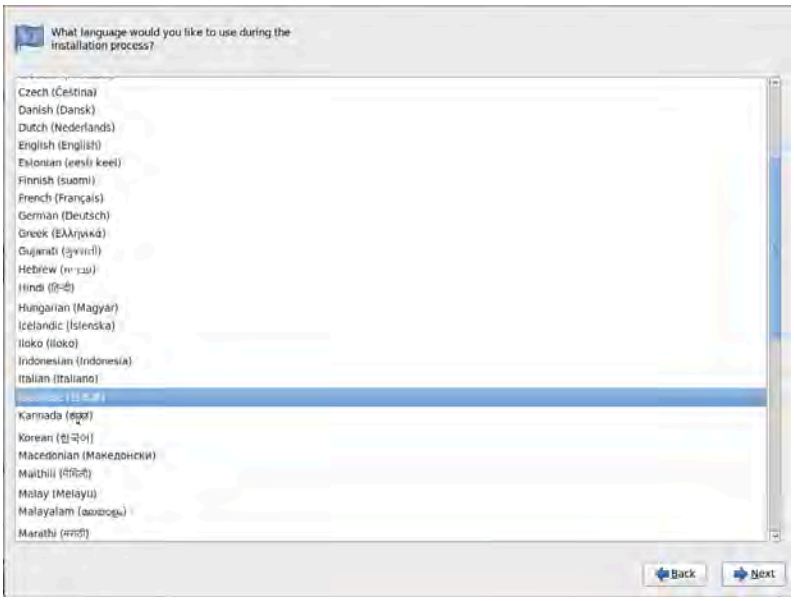
3. 「Disc Found」のダイアログが現れます。メディアの内容の整合性をチェックします。チェックにはしばらく時間がかかります。チェックをする場合は「OK」、しない場合は「Skip」を選択します。

選択項目はカーソルキーか TAB キーで変更できます。選択は Enter キーです。

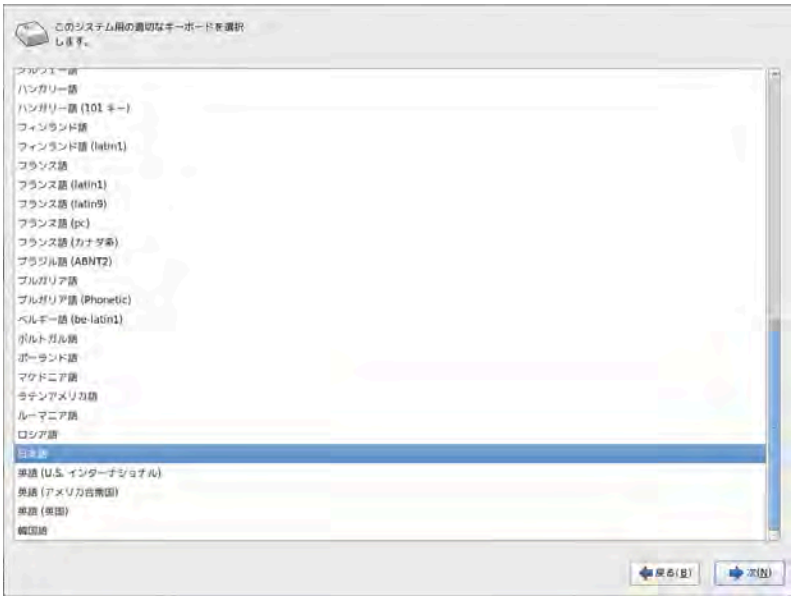
4. インストーラーのトップ画面が表示されるので、右下の「Next」をクリックします。




5. インストール時に利用する言語を選択します。「Japanese(日本語)」を選択し、右下の「Next」をクリックします。



6. キーボードの選択画面が現れます。日本語キーボードのときは「日本語」、英語キーボードのときは「英語（アメリカ合衆国)」を選択し、右下の「次」をクリックします。

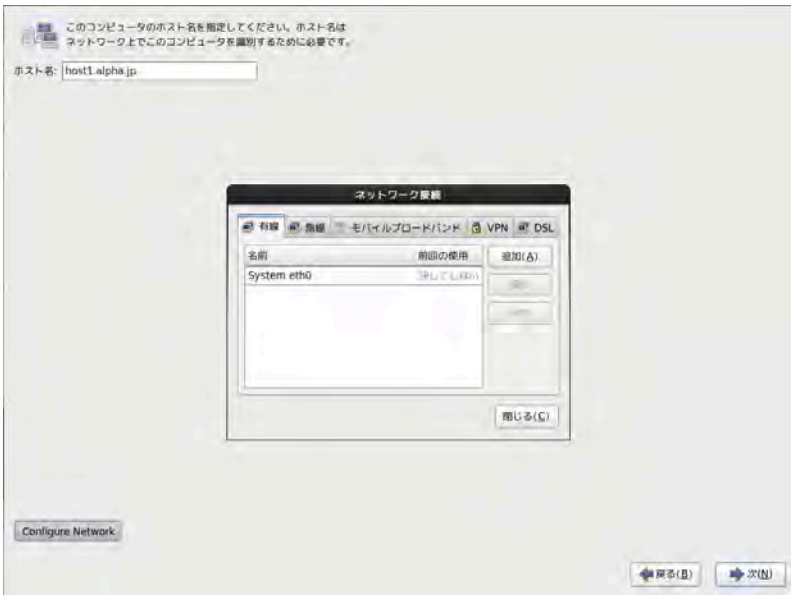



7. ストレージの選択画面が現れます。「Basic Storage Devices」を選択し、右下の「次」をクリックします。

8.「ストレージデバイスの警告」のダイアログが現れます。「はい。含まれていません。どのようなデータであっても破棄してください。」をクリックします。


9. ホスト名の指定の画面が現れます。ホスト名を入力します。

10. 左下の「Configure Network」をクリックし、ネットワークの設定画面を呼び出します。
11. 「ネットワーク接続」のダイアログが現れます。設定したいネットワークインターフェース（例では「System eth0」）を選択し、「編集」をクリックします。



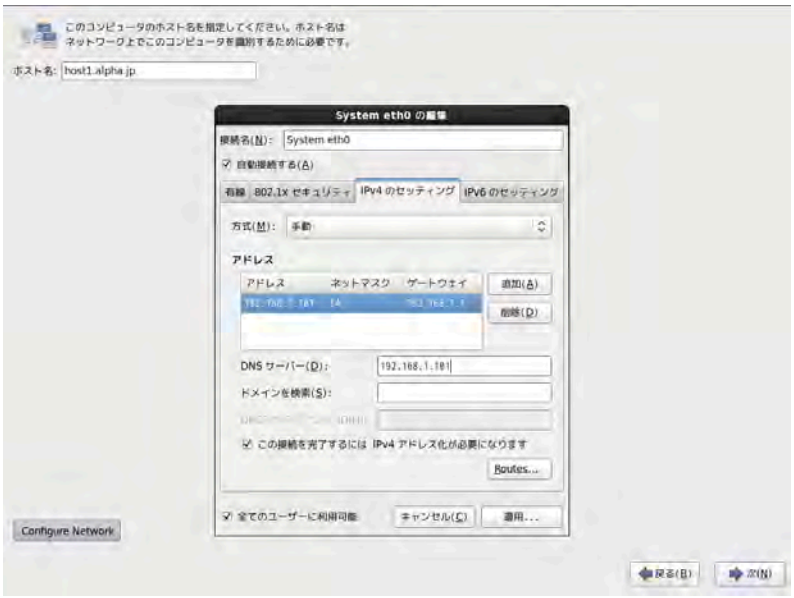

12. ネットワークインターフェースの設定ダイアログが現れるので、確認シートに書いた内容で以下の設定を行います。
13. 「自動接続する」をチェックします。


14. 「IPv4 のセッティング」タブをクリックします。

15.「方式」を「手動」に変更します。


16.「追加」をクリックし、「アドレス」、ゲートウェイを入力します。「ネットマスク」は自動的に入力されます。

17.「DNS サーバー」を入力します。

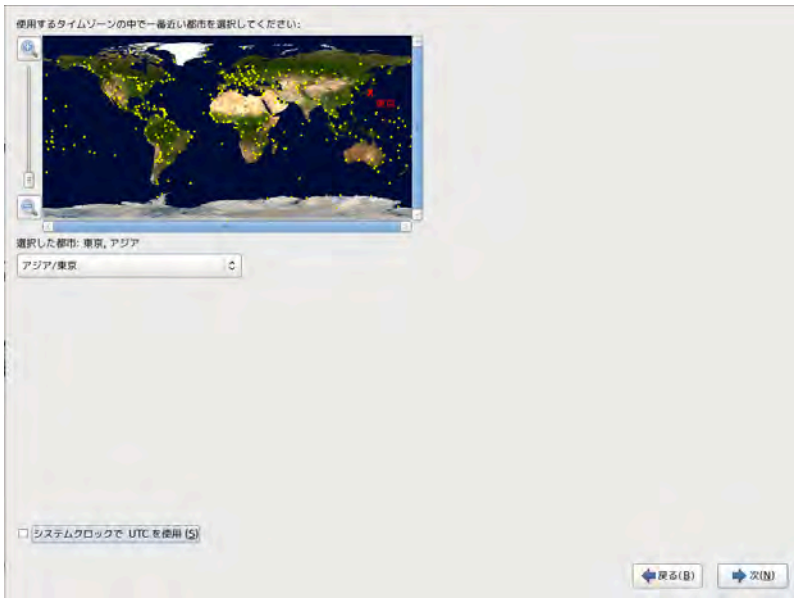



18. 設定が終わったら、「適用」をクリックします。

19.「閉じる」をクリックして「ネットワーク接続」ダイアログを閉じます。

20. 右下の「次」をクリックします。

21. 時間帯を設定します。「アジア／東京」が選択されていることを確認し、「システムクロックで UTC を使用」のチェックを外して[*1]、右下の「次」をクリックします。



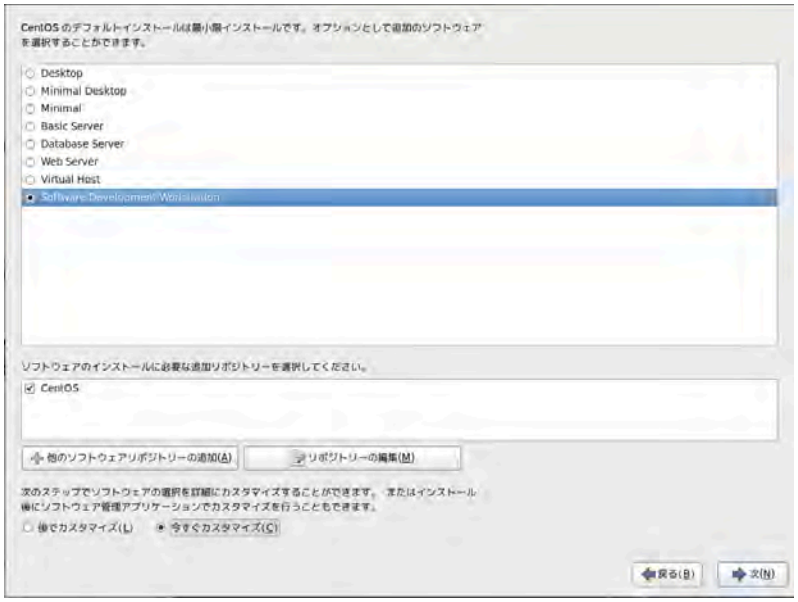

22. root のパスワードを入力して、右下の「次」をクリックします。パスワードが短かったりした場合、「パスワードが弱すぎます。」のダイアログが表示されます。「キャンセル」をクリックしてもっと複雑なパスワードを再設定するか、「とにかく使用する」をクリックします。


[*1] 「システムクロックで UTC を使用」でチェックをはずすのは、PC が持つ内部時計（= BIOS の時計）を UTC（= 世界標準時）とみなしてしまうためです。多くの場合、内部時計は現在の時刻を現在の時間帯（=日本では JST）で設定されているので、UTC と見る機能を OFF にするため、チェックをはずします。

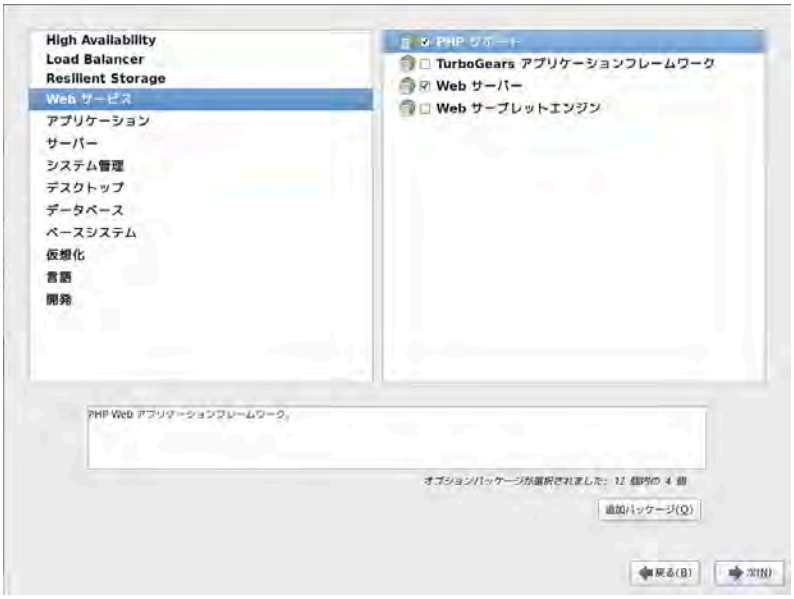
23. ハードディスクをフォーマットする方法を尋ねられます。「Use All Space」を選択し、右下の「次」をクリックします。


「Writing storage configuration to disk」ダイアログが表示されたら、「Write changes to disk」をクリックします。

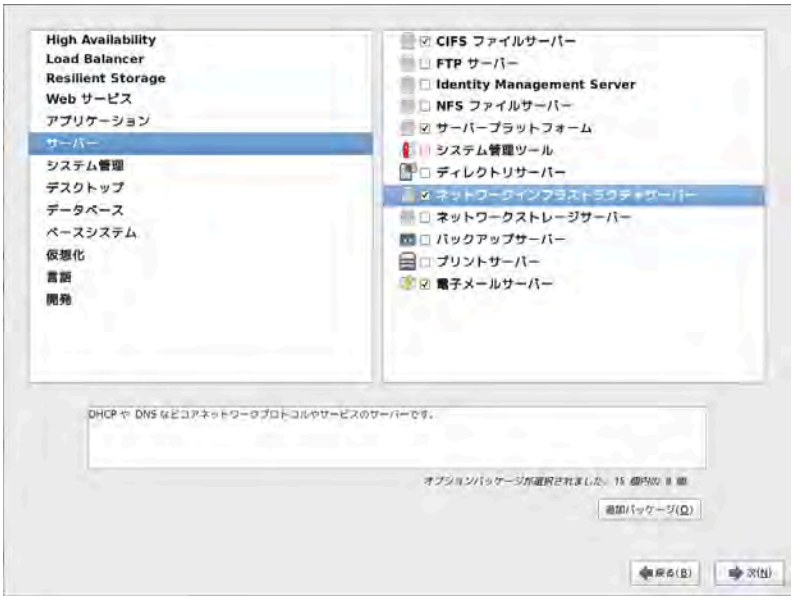
24. インストールするソフトウェアのグループを設定します。「Software Development Workstation」を選択し、左下の「今すぐカスタマイズ」を選択して、右下の「次」をクリックします。



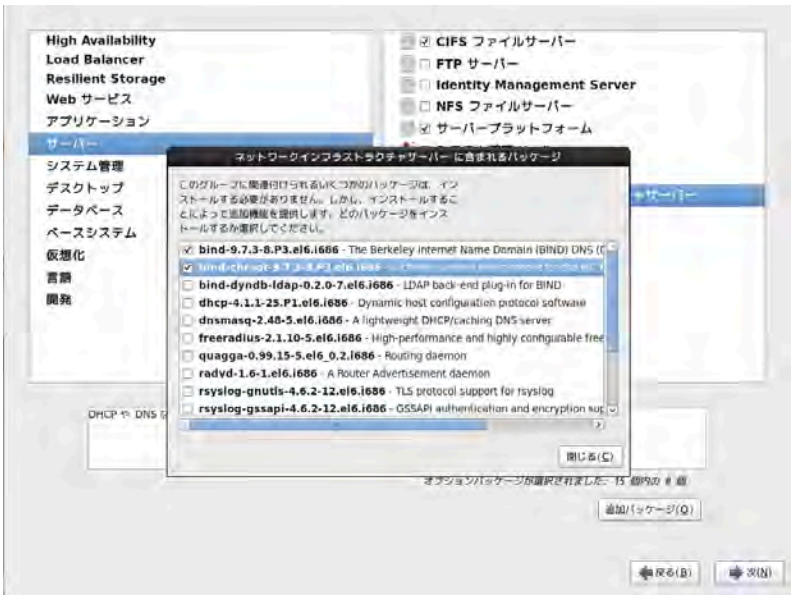

25. 追加でインストールするソフトウェアのグループを設定します。以下の通りに選択します。
26.「Web サービス」を選択し、「PHP サポート」と「Web サーバー」をチェックします。



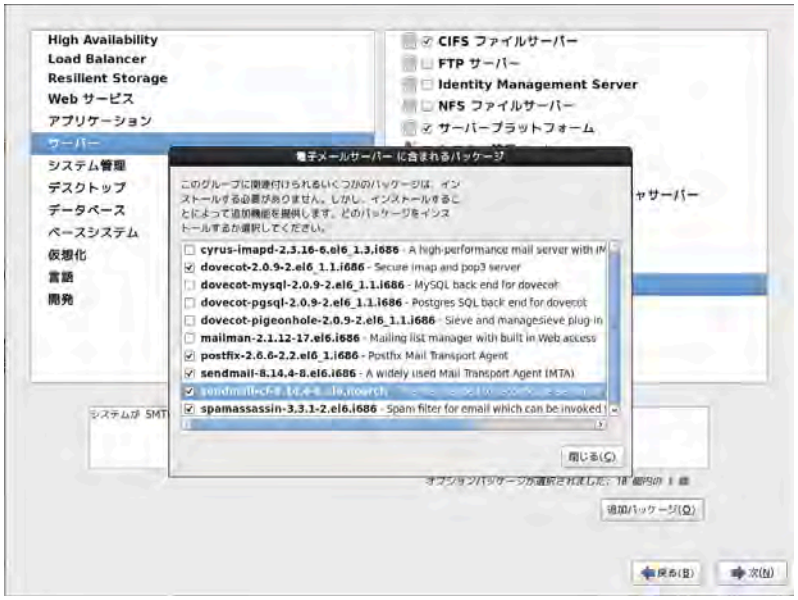
27.「サーバー」を選択し、「CIFS ファイルサーバー」をチェックします。
28.「ネットワークインフラストラクチャサーバー」をチェックし、「追加パッケージ」をクリックします。




29.「bind」と「bind-chroot」をチェックし、「閉じる」をクリックします。



30. 電子メールサーバーをチェックし、「追加パッケージ」をクリックします。
31.「sendmail」と「sendmail-cf」をチェックし、「閉じる」をクリックします。




32. 追加ソフトウェアをすべてを選択したら、右下の「次」をクリックします。
33. インストールが開始されます。
34. インストールが終了すると、再起動が要求されます。右下の「再起動」をクリックします。

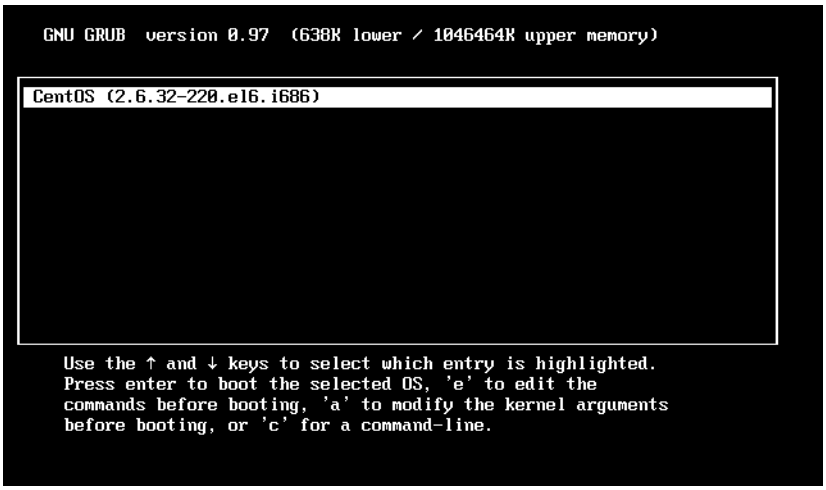

再起動時する前に、DVD を取り出します。もしくは、BIOS を起動して起動順序の優先順位をハードディスクを最初にします。そうしないと、次回の起動時もインストール DVD から起動して

しまいます。

## 2.4 インストール直後の初期設定

インストール直後、最初の起動時に様々な設定を求められます。次に、その設定を行いましょう。

1. インストールしたマシンを起動します。インストール作業から引き続いて作業するときは、再起動になります。ブートローダーが起動する OS の選択を求めるので、そのまま待つか、Enter キーを押します。

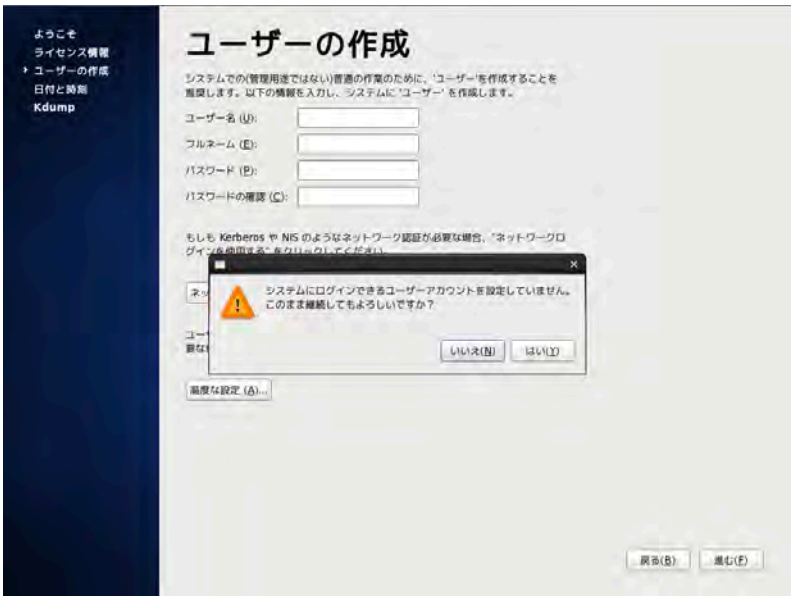



2. 起動プロセスが進み「ようこそ」の画面が現れます。右下の「進む」をクリックします。

3.「ライセンス情報」の画面が現れます。「はい、ライセンス同意書に同意します」を選択して、右下の「進む」をクリックします。


4. ユーザーの作成の画面が現れます。ここでは作成せず、右下の「進む」をクリックします。

「システムにログインできるユーザーアカウントを設定しません。このまま継続してもよろしいですか？」とダイアログが表示されますが、「はい」をクリックして継続します。



5. 日付と時刻の設定画面が現れます。日付と時刻を正しく設定し、右下の「進む」をクリックします。インターネットへの接続が使用できる場合には、「ネットワーク上で日付と時刻を同期化します」をチェックします。

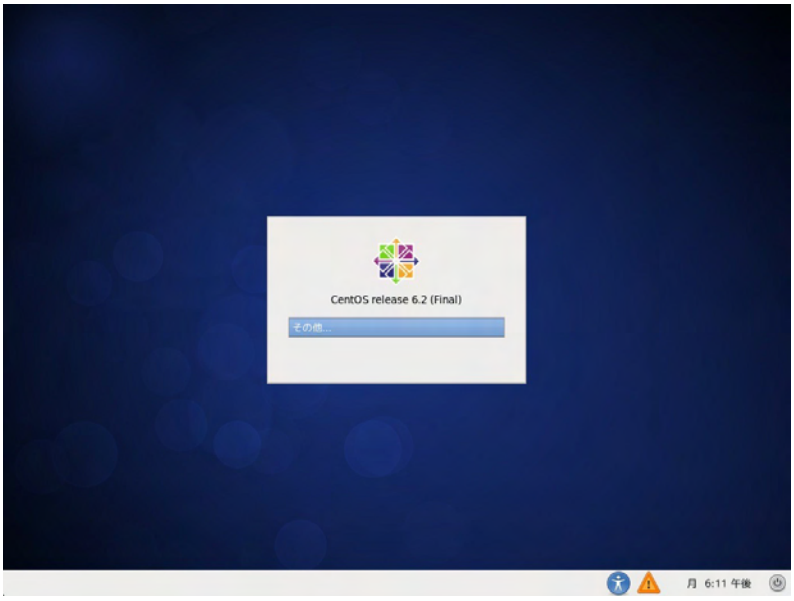

6. Kdump の設定画面が現れます。メモリが少ない場合には「kdump を設定するにはメモリーが足りません。」のダイアログが表示されるので、「OK」をクリックします。ここでは Kdump を設定せず、右下の「終了」をクリックします。


これでインストール直後の初期設定は終了です。

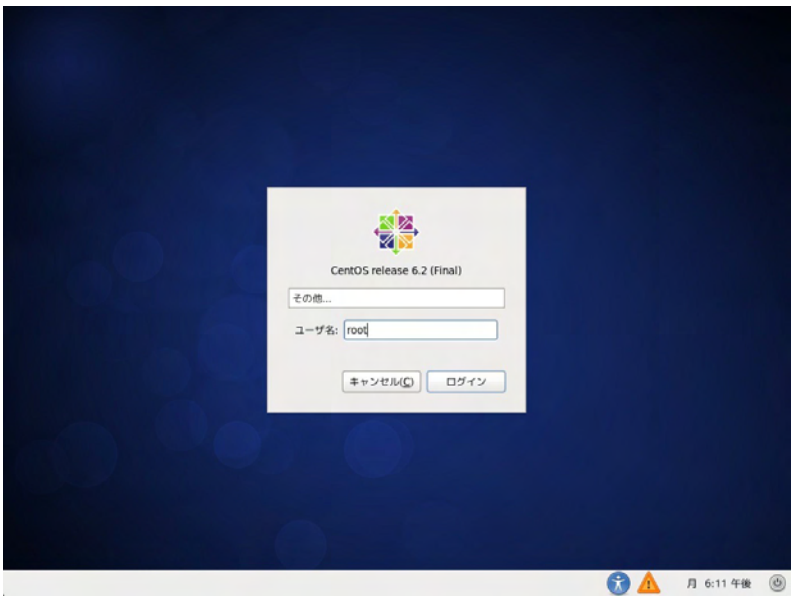
## 2.5 ログインする

ログイン画面が表示されるので、管理者ユーザー root でログインします。

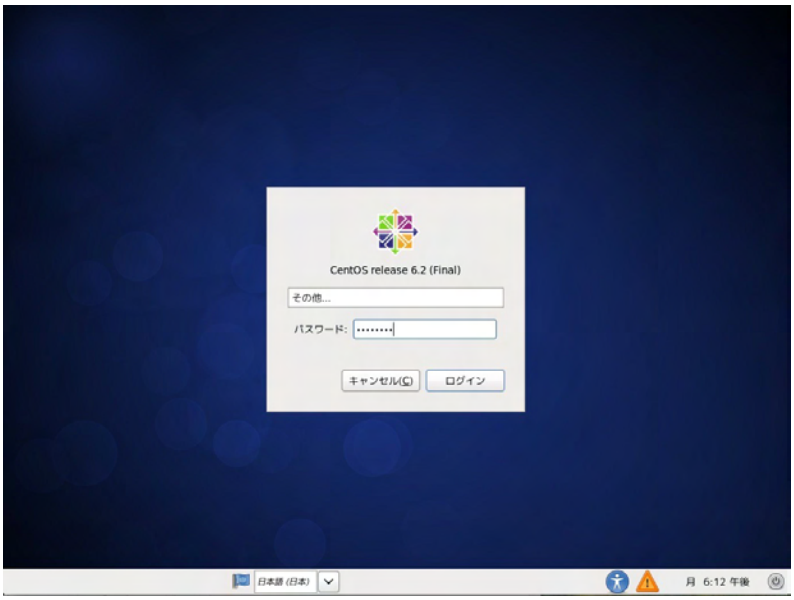
1.「その他…」をクリックします。




2.「ユーザ名」に「root」を入力し、「ログイン」をクリックします。



3. 「パスワード」にインストール時に設定した root パスワードを入力し、「ログイン」をクリックします。




4. 初めての root でのログインの際には、警告ダイアログが表示されます。「再度表示しない」をチェックし、「閉じる」をクリックします。

## 2.6 コマンドの実行

GUI でログインすると、Windows などと同じように様々なグラフィカルなアプリケーションが利用できますが、「端末」を実行すると Linux のコマンドを実行できます。端末を起動するには以下の方法があります。

- 「アプリケーション」メニューから「システムツール」、「端末」を選択する
- デスクトップ上を右クリックし、ポップアップメニューから「端末の中に開く」を選択する

「端末」ウインドウは複数起動できるので、一方で操作をしながら一方でログを表示したり、ユーザーを切り替えて操作することもできます。

## 2.7 セキュリティの設定

インストール直後はセキュリティを高める設定となっていますが、実習をスムーズに進めるためにファイアウォールと SELinux を無効化します。

### 2.7.1 ファイアウォール

ファイアウォールはネットワークにおいて様々なアクセス制限を行い、ネットワークからの攻撃や不正なアクセス等を防ぐ機能です。

従って、実運用で利用する場合は基本的にファイアウォールは有効にすべきです。しかし、本教科書ではサーバー構築の方法をよりわかりやすく説明するためファイアウォールは無効にして構築作業を解説し、最後に基本的なセキュリティの設定について解説しています。

### 2.7.2 ファイアウォールの無効化

ファイアウォールを無効にして、外部からの接続が可能なように設定します。

1. iptables サービスを停止します。

```
# /etc/init.d/iptables stop
iptables: ファイアウォールルールを消去中:                    [  OK  ]
iptables: チェインをポリシー ACCEPT へ設定中 filter          [  OK  ]
iptables: モジュールを取り外し中:                            [  OK  ]
```

2. iptables コマンドでポリシーが ACCEPT になっていることを確認します。

```
# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
```

```
Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

3. chkconfig コマンドで Linux 起動時に iptables サービスを自動起動しないように設定します。

```
# chkconfig iptables off
# chkconfig --list iptables
iptables        0:off   1:off   2:off   3:off   4:off   5:off   6:off
```

### 2.7.3 SELinux

Linux のファイルには、ユーザー権限等でアクセス制御を設定することができます。SELinux では、さらに高度なセキュリティモデルを導入し、Linux のアクセス権以上の高度なアクセス制御を実現します。セキュリティを考慮すると、ファイアウォールと同様に SELinux を有効にして運用すべきですが、本教科書では無効にして作業を行います。

### 2.7.4 SELinux の無効化

SELinux を無効にします。

1. エディタで/etc/sysconfig/selinux を開きます。

```
# vi /etc/sysconfig/selinux
```

2. 以下のように設定を変更します。

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
SELINUX=disabled ← disabled に変更
# SELINUXTYPE= can take one of these two values:
#     targeted - Targeted processes are protected,
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

3. 再起動します。
4. getenforce コマンドを実行して、SELinux が無効にされているか確認します。「Disabled」と応答があった場合、設定が正しく行なわれています。

```
# getenforce
Disabled
```

# 第3章

# ネットワーク

インストール後の設定作業として、ネットワーク環境を利用するために必要な設定を確認し、必要であれば再設定を行います。確認のために、ネットワークインターフェースを調べるコマンドと、設定ファイルの変更やネットワーク設定スクリプトを利用します。ネットワーク機能を利用できるようにする設定は、サーバー構築の前提条件となるため、必要不可欠な知識です。

## 3.1 用語集

**ネットワークインターフェース**

LAN ケーブルを接続して、外部のマシンとの間でデータをやり取りするための物理的なインターフェースです。

**ループバックインターフェース**

マシン内部でデータをやり取りするための仮想的なインターフェースです。

**IP(Internet Protocol)**

IP は、ネットワークに接続したコンピューター間でデータをやり取りするためのプロトコルです。

**IP アドレス**

IP アドレスは、IP 通信で各コンピューターに割り当てられる値です。データの送り先として IP アドレスを指定すると、その IP アドレスが割り当てられたコンピューターに送信されたデータが届きます。

**IPv4(Internet Protocol version 4)**

現在のインターネットで利用されている通信プロトコルです。IPv4 では、IP アドレスを 4 バイト（32 ビット）で表します。本来は 32 個の 2 進数（0 と 1）の羅列ですが、人間に分かりやすくするために 1 バイトごとに 10 進数に変換して．(ドット) で区切って「192.168.1.1」の様に表記します。次世代の IP である IPv6 では、IP アドレスを 128 ビットで表します。

**ネットワークアドレス**

ホストが属しているネットワーク自体を指し示す IP アドレスです。

ブロードキャスト
ホストが属しているネットワーク全体を指し示す IP アドレスです。

ネットマスク
IP アドレスのうち、どこまでがネットワーク部で、どこまでがホスト部かを示すための値です。IP アドレスとネットマスクの 2 つの値から、ネットワークアドレス、ブロードキャストアドレスを割り出すことができます。

デフォルトゲートウェイ
インターネットは、小さなネットワークが相互に接続したネットワークです。小さなネットワーク間を接続する機器としてルーター (ゲートウェイ) が使われます。ゲートウェイは 1 つのネットワークに複数設置することができますが、特に指定が無い場合にはデフォルトゲートウェイを使って外部のネットワークとの通信を行います。

DHCP(Dynamic Host Configuration Protocol)
IP アドレスなどのネットワーク設定を自動的に割り当てるプロトコルです。

TCP(Transmission Control Protocol)
TCP は、コネクション方式で通信するプロトコルです。IP と組み合わせた TCP/IP がインターネットの標準的な通信プロトコルです。TCP の特長として、届かなかった通信パケットを再送信して確実に通信を行う仕組みがあります。

UDP(User Datagram Protocol)
UDP は、コネクションレス方式で通信するプロトコルです。TCP とは異なり、データの再送信が行われないので通信の確実性は劣りますが、セッションを確立するための「3 ウェイハンドシェイク」の手間が不要なためシンプルな通信に適しています。たとえば、大量に問い合わせが行われる DNS への名前解決の問い合わせは UDP となっています。

ポート番号
ポート番号は、TCP と UDP が通信する際に使用する値です。たとえば Web サーバーはポート番号 80 番を使用して動作しているので、Web ブラウザーは目的の Web サーバーのポート番号 80 番に接続します。ポート番号は 0 番から 65535 番まで使用できますが、0〜1023 番は WELL KNOWN PORT、1024〜49151 番は REGISTERED PORT として予約されています。

ICMP(Internet Control Message Protocol)
データの転送エラーやデータ転送量などの情報を通知するためのプロトコルです。

ping コマンド

ping コマンドは、ICMP を使って宛先に指定したホストに到達することができるかどうかを確認するコマンドです。

## 3.2 NetworkManager サービスの停止と network サービスの起動

CentOS 6.2 では、ネットワークインターフェースの管理を NetworkManager サービスが行っています。NetworkManager は高機能ですが、サーバーでの使用には向いていないので、NetworkManager は停止し、従来の network サービスでネットワークインターフェースの管理を行うように変更します。

### 3.2.1 NetworkManager サービスの無効化と停止

chkconfig コマンドで Linux 起動時に NetworkManager サービスを自動起動しないように設定し、現在動作している NetworkManager サービスをスクリプトで停止します。

```
# chkconfig NetworkManager off
# chkconfig --list NetworkManager
NetworkManager  0:off   1:off   2:off   3:off   4:off   5:off   6:off
# /etc/init.d/NetworkManager stop
NetworkManager デーモンを停止中:                          [  OK  ]
```

### 3.2.2 network サービスの有効化と起動

chkconfig コマンドで Linux 起動時に network サービスを自動起動するように設定し、現在停止している network サービスをスクリプトで起動します。

```
# chkconfig network on
# chkconfig --list network
network         0:off   1:off   2:on    3:on    4:on    5:on    6:off
# /etc/init.d/network start
ループバックインターフェイスを呼び込み中                    [  OK  ]
インターフェース eth0 を活性化中:                          [  OK  ]
```

## 3.3 ネットワークインターフェースの確認

Linux をインストールしたマシンが正常にネットワークに接続できるかどうか、設定を確認します。

### 3.3.1 ネットワークインターフェースの確認

インストール時に設定したネットワークインターフェースの状態を ifconfig コマンドで確認します。

```
# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:1C:42:47:04:C9
          inet addr:192.168.1.101  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::21c:42ff:fe47:4c9/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:90 errors:0 dropped:0 overruns:0 frame:0
          TX packets:83 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:11599 (11.3 KiB)  TX bytes:11862 (11.5 KiB)

# ifconfig lo
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:184 errors:0 dropped:0 overruns:0 frame:0
          TX packets:184 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:15172 (14.8 KiB)  TX bytes:15172 (14.8 KiB)
```

ifconfig コマンドで表示された eth0 が物理的なインターフェース、lo が仮想的なループバックインターフェースです。Linux では数字を数えるときに 0 から数える場合があります。eth0 は、「1 番目の Ethernet のネットワークインターフェース」という意味です。

### 3.3.2 ネットワークインターフェースの設定ファイルの確認

ループバックインターフェース lo の設定ファイルである/etc/sysconfig/network-scripts/ifcfg-lo と、ネットワークインターフェース eth0 の設定ファイルである/etc/sysconfig/network-scripts/ifcfg-eth0 を参照します。

```
# cat /etc/sysconfig/network-scripts/ifcfg-lo
DEVICE=lo
IPADDR=127.0.0.1
NETMASK=255.0.0.0
NETWORK=127.0.0.0
# If you're having problems with gated making 127.0.0.0/8 a martian,
# you can change this to something else (255.255.255.255, for example)
BROADCAST=127.255.255.255
ONBOOT=yes
NAME=loopback

# cat /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE="eth0"
```

```
NM_CONTROLLED="yes"
ONBOOT=yes
HWADDR=00:1C:42:97:1C:DA
TYPE=Ethernet
BOOTPROTO=none
IPADDR=192.168.1.101
PREFIX=24
GATEWAY=192.168.1.1
DNS1=192.168.1.101
DEFROUTE=yes
IPV4_FAILURE_FATAL=yes
IPV6INIT=no
NAME="System eth0"
UUID=5fb06bd0-0bb0-7ffb-45f1-d6edd65f3e03
```

ifcfg-lo 設定ファイルはどのマシンでも同じ設定内容となりますが、ifcfg-eth0 設定ファイルはマシン毎に設定が違うファイルとなります。ifcfg 設定ファイルは、設定する項目名と設定内容が=で結ばれた形 (「項目名=設定内容」という記述) で複数の設定が並べられています。主な設定項目と内容は次の表の様になります。

表 3.1　ネットワークインターフェースの設定項目

| 項目 | 内容 | 設定例 |
|---|---|---|
| DEVICE | ネットワークインターフェース名 | eth0 |
| IPADDR | インターフェースの IP アドレス | 192.168.1.101 |
| NETMASK | 所属ネットワークを区切るためのサブネットマスク | 255.255.255.240 |
| NETWORK | 所属ネットワークを指定するネットワークアドレス | 192.168.2.0 |
| ONBOOT | 起動時にネットワーク・インターフェースを有効とするか否か | yes…有効、no…無効 |
| BOOTPROTO | 他機器から設定を受け取るプロトコル | none…プロトコルを利用しない、dhcp…DHCP を利用、BOOTP…BOOTP を利用 |

ネットワークインターフェースの設定ファイル以外に、ネットワークの Red Hat 系の基本設定ファイルである/etc/sysconfig/network も見てみてください。

```
# cat /etc/sysconfig/network
NETWORKING=yes
HOSTNAME=host1.alpha.jp
GATEWAY=192.168.1.1
```

/etc/sysconfig/network 設定ファイルで書き換えなどが必要な項目と内容は次の表の様になります。

設定ファイルがなければファイルを作る必要があり、設定内容と PC を接続するネットワークの整合性が取れていなければ設定を修正する必要があります。Red Hat 系の Linux には、設定をする

表 3.2　ネットワークの設定項目

| 項目 | 内容 | 設定例 |
|---|---|---|
| NETWORKING | ネットワーク機能を有効にするか否か | yes…有効、no…無効 |
| GATEWAY | 他のネットワークへの出入り口となるアドレス | 192.168.1.1 |

setup スクリプトも用意されており、setup スクリプトは CUI(キャラクタベースのインターフェス)で用意された項目を設定することもできます。

### 3.3.3　ネットワークインターフェースの再設定

インストール時に IP アドレスの設定を間違えた時などは、ネットワークインターフェースを再設定します。

ネットワークインターフェースを再設定する作業として、前の節で説明した /etc/sysconfig/network-scripts/ifcfg-eth0 ファイルと/etc/sysconfig/network ファイルの書き換えが必要です。

設定した内容をネットワークインターフェースに反映させるため、/etc/init.d/network スクリプトが用意されています。/etc/init.d/network スクリプトには次の様なオプションがあり、ここでは restart オプションを使い、サービスを再起動します。ネットワークを開始したら、ifconfig コマンドでネットワークインターフェースの状態を再確認します。

| オプション | 内容 |
|---|---|
| start | サービスを開始 |
| stop | サービスを停止 |
| restart | サービスを再起動 (停止してから開始) |
| status | サービスの状況を表示 |

次の例は、eth0 の IP アドレスが誤って 192.168.1.100 に設定されていたので、192.168.1.101（第 4 オクテットを 101 に変更）に変更する場合の例です。変更するには/etc/sysconfig/network-scripts/ifcfg-eth0 ファイルをエディタで編集し、ネットワークを再開したら、ifconfig コマンドでネットワークインターフェースの状態を再確認します。

```
# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:1C:42:47:04:C9
          inet addr:192.168.1.100  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::21c:42ff:fe47:4c9/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:140 errors:0 dropped:0 overruns:0 frame:0
          TX packets:110 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:16215 (15.8 KiB)  TX bytes:17180 (16.7 KiB)
```

```
# vi /etc/sysconfig/network-scripts/ifcfg-eth0
```

```
DEVICE="eth0"
NM_CONTROLLED="yes"
ONBOOT=yes
HWADDR=00:1C:42:47:04:C9
TYPE=Ethernet
BOOTPROTO=none
IPADDR= 192.168.1.101 ← IP アドレスを書き換え
PREFIX=24
GATEWAY=192.168.1.1
DNS1=192.168.1.101
DEFROUTE=yes
IPV4_FAILURE_FATAL=yes
IPV6INIT=no
NAME="System eth0"
UUID=5fb06bd0-0bb0-7ffb-45f1-d6edd65f3e03
```

```
# /etc/init.d/network restart
インターフェース eth0 を終了中:                            [  OK  ]
ループバックインターフェースを終了中                       [  OK  ]
ループバックインターフェイスを呼び込み中                   [  OK  ]
インターフェース eth0 を活性化中:                          [  OK  ]

# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:1C:42:47:04:C9
          inet addr:192.168.1.101  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::21c:42ff:fe47:4c9/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:140 errors:0 dropped:0 overruns:0 frame:0
          TX packets:110 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:16215 (15.8 KiB)  TX bytes:17180 (16.7 KiB)
```

### 3.3.4 ネットワークインターフェースの動作確認

ネットワークインターフェースが機能しているかは ping コマンドで確認します。ping コマンドで確認する IP アドレスとして自分の物理ネットワークインターフェース (eth0) の IP アドレス、講師のマシンの IP アドレス (192.168.1.10) やその他のマシンの IP アドレスなどを指定します。ping コマンドは [Control]+[c] で中止できます。

```
# ping 192.168.1.101 ← 自分の IP アドレス
PING 192.168.1.101 (192.168.1.101) 56(84) bytes of data.
64 bytes from 192.168.1.101: icmp_seq=1 ttl=64 time=0.044 ms
64 bytes from 192.168.1.101: icmp_seq=2 ttl=64 time=0.051 ms
64 bytes from 192.168.1.101: icmp_seq=3 ttl=64 time=0.051 ms
^C ←  [Control]+[c] で中止
--- 192.168.1.101 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2324ms
rtt min/avg/max/mdev = 0.044/0.048/0.051/0.008 ms
```

```
#   ping 192.168.1.10 ← その他のホストの IP アドレス
PING 192.168.1.10 (192.168.1.10) 56(84) bytes of data.
64 bytes from 192.168.1.10: icmp_seq=1 ttl=64 time=0.096 ms
64 bytes from 192.168.1.10: icmp_seq=2 ttl=64 time=0.055 ms
64 bytes from 192.168.1.10: icmp_seq=3 ttl=64 time=0.048 ms
[Control]+[c]
--- 192.168.1.10 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 0.048/0.066/0.096/0.022 ms
```

### 3.3.5 物理ネットワークインターフェースの IP アドレスと名前の対応を確認

/etc/hosts ファイルに PC のネットワークインターフェース (eth0) の IP アドレス (192.168.1.101) と対応する名前の定義を追加します。ここではホスト名として host1 を、ドメイン名として alpha.jp を使います。

インストール時に host1.alpha.jp を入力して 127.0.0.1 で始まる行に host1.alpha.jp の定義が記述されている場合は、127.0.0.1 が定義されている行から host1.alpha.jp と host1 を取り除いてください。

変更前の/etc/hosts ファイル

```
#  cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
```

変更後の/etc/hosts ファイル

```
#  cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.1.101 host1.alpha.jp host1
```

ping コマンドに host1.alpha.jp と host1 を指定してネットワークインターフェースが認識されているか確認します。ping コマンドは [Control]+[c] で止められます。

```
#  ping host1.alpha.jp
PING host1.alpha.jp (192.168.1.101) 56(84) bytes of data.
64 bytes from host1.alpha.jp (192.168.1.101): icmp_seq=1 ttl=64 time=0.016 ms
64 bytes from host1.alpha.jp (192.168.1.101): icmp_seq=2 ttl=64 time=0.032 ms
64 bytes from host1.alpha.jp (192.168.1.101): icmp_seq=3 ttl=64 time=0.036 ms
^C
--- host1.alpha.jp ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3013ms
rtt min/avg/max/mdev = 0.016/0.029/0.036/0.010 ms
```

```
# ping host1
PING host1.alpha.jp (192.168.1.101) 56(84) bytes of data.
64 bytes from host1.alpha.jp (192.168.1.101): icmp_seq=1 ttl=64 time=0.019 ms
64 bytes from host1.alpha.jp (192.168.1.101): icmp_seq=2 ttl=64 time=0.031 ms
64 bytes from host1.alpha.jp (192.168.1.101): icmp_seq=3 ttl=64 time=0.029 ms
^C
--- host1.alpha.jp ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2279ms
rtt min/avg/max/mdev = 0.019/0.026/0.031/0.006 ms
```

### 3.3.6 サービスのポート番号を確認

どんなネットワークサービスが自分の PC で動いているかを、netstat コマンドと lsof コマンドで確認します。

netstat -at コマンドを実行すると、現在の TCP 通信の状態をすべて表示します。

```
# netstat -at
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address               Foreign Address             State
tcp        0      0 *:sunrpc                    *:*                         LISTEN
tcp        0      0 192.168.122.1:domain        *:*                         LISTEN
tcp        0      0 *:ssh                       *:*                         LISTEN
tcp        0      0 localhost:ipp               *:*                         LISTEN
tcp        0      0 localhost:smtp              *:*                         LISTEN
tcp        0      0 *:60323                     *:*                         LISTEN
tcp        0      0 *:amqp                      *:*                         LISTEN
tcp        0      0 *:sunrpc                    *:*                         LISTEN
tcp        0      0 *:ssh                       *:*                         LISTEN
tcp        0      0 localhost:ipp               *:*                         LISTEN
tcp        0      0 localhost:smtp              *:*                         LISTEN
tcp        0      0 *:55720                     *:*                         LISTEN
```

lsof -i コマンドを実行すると、現在開かれているすべてのポートを表示します。

```
# lsof -i
COMMAND    PID    USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
rpcbind   1391     rpc    6u  IPv4  11920      0t0  UDP *:sunrpc
rpcbind   1391     rpc    7u  IPv4  11924      0t0  UDP *:718
rpcbind   1391     rpc    8u  IPv4  11925      0t0  TCP *:sunrpc (LISTEN)
rpcbind   1391     rpc    9u  IPv6  11927      0t0  UDP *:sunrpc
rpcbind   1391     rpc   10u  IPv6  11929      0t0  UDP *:718
rpcbind   1391     rpc   11u  IPv6  11930      0t0  TCP *:sunrpc (LISTEN)
avahi-dae 1515   avahi   13u  IPv4  12373      0t0  UDP *:mdns
avahi-dae 1515   avahi   14u  IPv4  12374      0t0  UDP *:46908
rpc.statd 1556 rpcuser    5r  IPv4  12544      0t0  UDP *:884
rpc.statd 1556 rpcuser    8u  IPv4  12551      0t0  UDP *:53198
rpc.statd 1556 rpcuser    9u  IPv4  12555      0t0  TCP *:60323 (LISTEN)
rpc.statd 1556 rpcuser   10u  IPv6  12559      0t0  UDP *:43370
rpc.statd 1556 rpcuser   11u  IPv6  12563      0t0  TCP *:55720 (LISTEN)
cupsd     1622    root    6u  IPv6  12808      0t0  TCP localhost:ipp (LISTEN)
```

```
cupsd     1622    root    7u  IPv4  12809      0t0  TCP localhost:ipp (LISTEN)
cupsd     1622    root    9u  IPv4  12812      0t0  UDP *:ipp
sshd      1733    root    3u  IPv4  13376      0t0  TCP *:ssh (LISTEN)
sshd      1733    root    4u  IPv6  13378      0t0  TCP *:ssh (LISTEN)
ntpd      1741     ntp   16u  IPv4  13406      0t0  UDP *:ntp
ntpd      1741     ntp   17u  IPv6  13407      0t0  UDP *:ntp
ntpd      1741     ntp   18u  IPv6  13411      0t0  UDP localhost:ntp
ntpd      1741     ntp   19u  IPv6  21146      0t0  UDP [fe80::21c:42ff:fe47:4c9]:ntp
ntpd      1741     ntp   20u  IPv4  13415      0t0  UDP localhost:ntp
ntpd      1741     ntp   21u  IPv4  21727      0t0  UDP host1.alpha.jp:ntp
ntpd      1741     ntp   23u  IPv4  15920      0t0  UDP 192.168.122.1:ntp
master    1822    root   12u  IPv4  13651      0t0  TCP localhost:smtp (LISTEN)
master    1822    root   13u  IPv6  13654      0t0  TCP localhost:smtp (LISTEN)
qpidd     1865   qpidd   10u  IPv4  13944      0t0  TCP *:amqp (LISTEN)
dnsmasq   2027  nobody    5u  IPv4  14578      0t0  UDP *:bootps
dnsmasq   2027  nobody    6u  IPv4  14584      0t0  TCP 192.168.122.1:domain (LISTEN)
dnsmasq   2027  nobody    7u  IPv4  14585      0t0  UDP 192.168.122.1:domain
```

netstat コマンドは-a オプションでサービスの状態を表示、-t オプションで TCP(Transmission Control Protocol) のサービスが使うポートなどの情報のみを表示します。lsof コマンドは-i オプションでサービスを受けているポートと対応するプログラムの情報を表示します。ポート番号とサービスの対応 (WELL KNOWN PORT NUMBERS:0〜1023 や REGISTERED PORT NUMBERS:1024〜49151) が定義されている/etc/services ファイルも確認してみてください。

```
# cat /etc/services
（略）
tcpmux          1/tcp                           # TCP port service multiplexer
tcpmux          1/udp                           # TCP port service multiplexer
rje             5/tcp                           # Remote Job Entry
rje             5/udp                           # Remote Job Entry
echo            7/tcp
echo            7/udp
discard         9/tcp           sink null
discard         9/udp           sink null
（略）
```

## 3.4 Web サーバーの動作確認

ネットワークインターフェースが動いたので Web サーバーを動かし、ネットワークインターフェースが本当に動作しているのかを確認しておきましょう。

### 3.4.1 必要なパッケージを確認

rpm コマンドで必要なパッケージがインストールされているかを確認します。

```
# rpm -q httpd
httpd-2.2.15-15.el6.centos.i686
```

パッケージがインストールされていない場合は次の様に表示されます。

```
# rpm -q httpd
パッケージ httpd はインストールされていません。
```

### 3.4.2 必要なパッケージをインストール

Web サーバーの構築に必要なパッケージがインストールされていないときは、rpm コマンドでインストールをします。

ここでは、自動マウント機能が動作していて DVD-ROM が自動的に/media/CentOS_6.2_Final にマウントされた状態でインストール作業を進めています。DVD-ROM の公開鍵を登録してからインストールすると警告メッセージが出ないので、インストールをする前に rpm コマンドに–import オプションと公開鍵のファイルを指定して登録しています。最後に、DVD-ROM のディレクトリ上で作業をしたので、cd コマンドでホームディレクトリへ移動しています。

```
# rpm --import /media/CentOS_6.2_Final/RPM-GPG-KEY-CentOS-6
# cd /media/CentOS_6.2_Final/Packages
# rpm -ivh httpd-2.2.15-15.el6.centos.i686.rpm
準備中...                ########################################### [100%]
   1:httpd                  ########################################### [100%]
# cd
```

### 3.4.3 Web サーバーを起動

パッケージが導入されているか確認したら、Web サービスである Apache を開始してみます。/etc/init.d/httpd スクリプトに start オプションを付けて Apache を起動します。

```
# /etc/init.d/httpd start
httpd を起動中:                                            [  OK  ]
```

### 3.4.4 ブラウザーで確認

Apache の動作を確認するために、Web ブラウザーで自分のアドレスにアクセスしてみます。

```
http://192.168.1.101/
```

自分のアドレスを確認した後は、周りの人のアドレスにアクセスして確認してみてください。

```
http://192.168.1.102/
```

# 第4章

# DNS サーバーの構築

ネットワークサービスを使うための土台となる名前解決のサービス (DNS) を設定します。自分の DNS サーバーを他のコンピューターから参照できるように設定をします。DNS に問い合わせを行うコマンドに慣れ、ドメインを管理する BIND プログラムの設定ファイルを扱います。

## 4.1 用語集

### ドメイン名とゾーン

組織に割り当てられてインターネットで使用する名前をドメイン名と呼びます。ドメイン名は ICANN(The Internet Corporation for Assigned Names and Numbers) により管理されています。DNS でドメイン名を設定するときは、ドメインではなく「ゾーン」と呼びます。

### FQDN

ドメイン名表記で、一番右に「.」（ドット）でルートドメインまでを記述する方式を FQDN と呼びます。

### DNS

DNS(Domain Name System) は、IP アドレスと対応するホスト名を登録しておき、プログラムからの問い合わせに応じて IP アドレスやホスト名を返答するシステムです。

### BIND

BIND(Berkeley Internet Name Domain) は、Linux と組み合わせて多く使用されている DNS サーバーのソフトウェアです。

### リゾルバ

ドメイン名をもとに IP アドレス情報の検索をしたり、IP アドレスからドメイン情報の検索を行う、名前解決を行うプログラムのことです。

### キャッシングネームサーバー

プログラムからの名前問い合わせを代行して DNS サーバーへ名前問い合わせをおこなって結果をキャッシュしておき、次回問い合わせ時にキャッシュの情報を返す DNS サーバーのことです。

グルーレコード
管理を委任しているサブドメインについての問合せに対して、DNS サーバーが委任先サブドメインの DNS サーバーを返す際に、追加情報として必要となる委任先 DNS サーバーの A レコードをグルーレコードといいます。

A レコード
名前に対して IP アドレスを指定するためのレコードです。

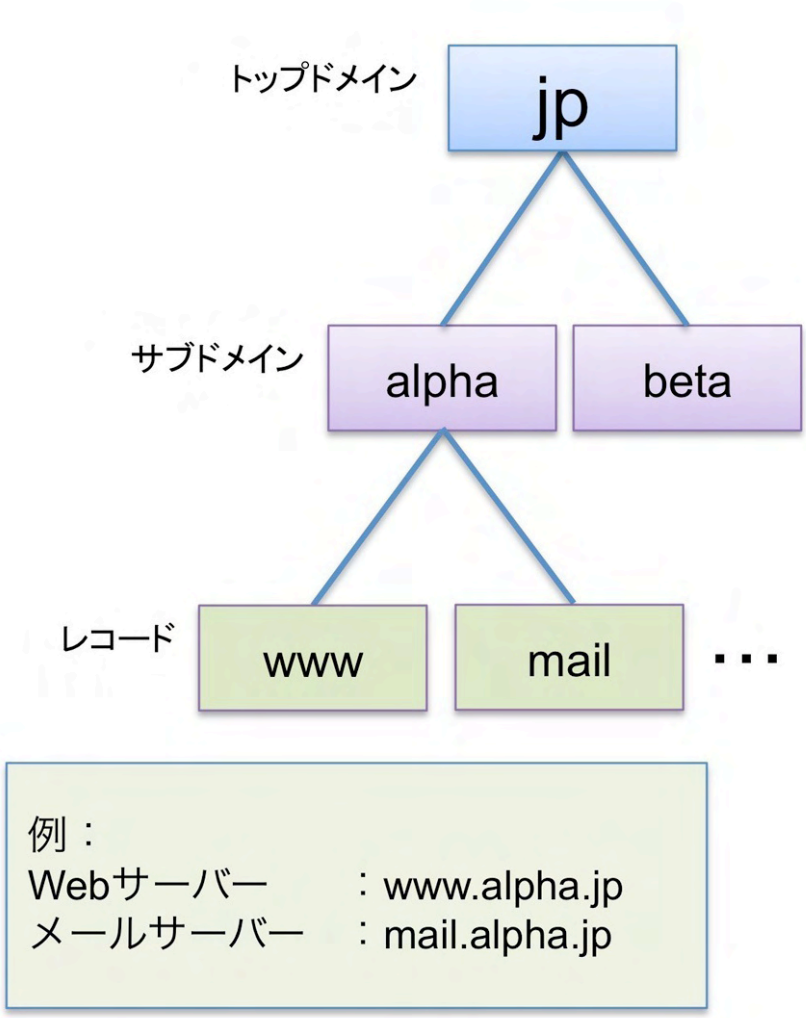
NS(Name Server) レコード
ゾーン権威を持つ DNS サーバーを指定するためのレコードです。

MX(Mail eXchange) レコード
メールアドレスに利用するドメイン名を定義するためのレコードです。メールサーバーの障害にも対応するために、複数個のメールサーバーを記述でき、プリファレンス値の低いサーバーにメール配信が優先されます。

## 4.2 DNS の仕組み

インターネットでのコンピューター同士の通信は、IP(Internet Protocol) を使って行われています。IP 通信には相手の IP アドレスが必要ですが、インターネット上の大量のコンピューターを IP アドレスで識別するのは困難です。そこでドメイン名やホスト名という考え方が導入されました。ドメイン名は組織を表し、ホスト名はその組織が管理しているコンピューターです。表記するときは「ホスト名. ドメイン名」とドット区切りで表記しますが、両方を合わせてホスト名と呼ぶこともあります。

インターネットの研究が始まった当初は IP アドレスが割り当てられたコンピューターの数も数えるほどだったので、ホスト名と IP アドレスの対応関係はファイルに記述されて、定期的に更新されていました。この仕組みは今でも残っており、Linux では/etc/hosts がそのファイルです。しかし、インターネットが広まるに従って、ホストファイルでは管理しきれなくなってきました。そこで登場したのが DNS(Domain Name System) です。




DNS サーバーは、ドメイン名を割り当てられた組織毎に用意します。DNS サーバーの管理者は、そのドメインに所属しているホスト名と割り当てられた IP アドレスを DNS サーバー登録します。ホストにアクセスしたい利用者は、そのホストが所属するドメインの DNS サーバーに問い合わせを行うことで、IP アドレスを得ることができます。

DNS の仕組みでは、ドメインの管理権限がそれぞれの DNS 管理者に権限委譲されているので、ホストファイルのような一元管理ではなく、分散管理となります。管理作業が分担されていて更新も頻繁に行われるので、リアルタイムにホスト名と IP アドレスの対応関係を調べることができる仕組みとなっています。

## 4.3 ドメインの構造

DNS が取り扱うドメイン名は設計上、ルートドメインを頂点とした階層型のツリー構造となっています。ちょうど、コンピューターのファイルシステムがルートディレクトリを頂点としたツリー構造になっているのと同じだと考えてよいでしょう。ドメインは、以下のいくつかの要素で構成されています。

### 4.3.1 ルートドメイン

ルートドメインは、ドメイン名の開始点です。通常は省略されますが、DNS の設定を記述する際には「.」（ドット）で表されます。トップレベルドメイントップレベルドメインには、.com や.org のような組織別ドメインや、.jp のような国別ドメインがあります。また、日本の場合には.co.jp のような組織種別型ドメインと、example.jp のような汎用 JP ドメインなどがあります。

### 4.3.2 ドメイン名の記述

ドメイン名の記述は、右側から記述していきます。FQDN(Fully Qualified Domain Name) であれば一番右にルートドメイン、そしてトップレベルドメインを記述し、さらに左側に各組織毎に割り当てられたドメイン名を記述していきます。各要素の間は「.」（ドット）で区切っていきます。

ドメイン名の記述例

- example.com.
- example.jp.
- example.co.jp.

トップレベルドメイン以降のドメイン名は、ドメイン取得者が独自にドメイン名を決めることができます。上記の例では example の部分が独自のドメイン名にあたります。

### 4.3.3 サブドメイン

記述例のようにドメイン名の左側にさらにドメイン名を記述していくことを「サブドメイン化」と呼びます。たとえば、example.co.jp ドメインをさらに東京と大阪の 2 つに分けて表記したいような場合には、以下の例のように記述します。

- tokyo.example.co.jp.
- osaka.example.co.jp.

サブドメイン化は、上位の（右側の）ドメインを管理している管理者が行います。たとえば、tokyo.example.co.jp ドメインまでのサブドメインの階層は次のようになっています。

1. jp ドメインはルートドメインのサブドメイン
2. co.jp ドメインは jp ドメインのサブドメイン
3. example.co.jp ドメインは co.jp ドメインのサブドメイン

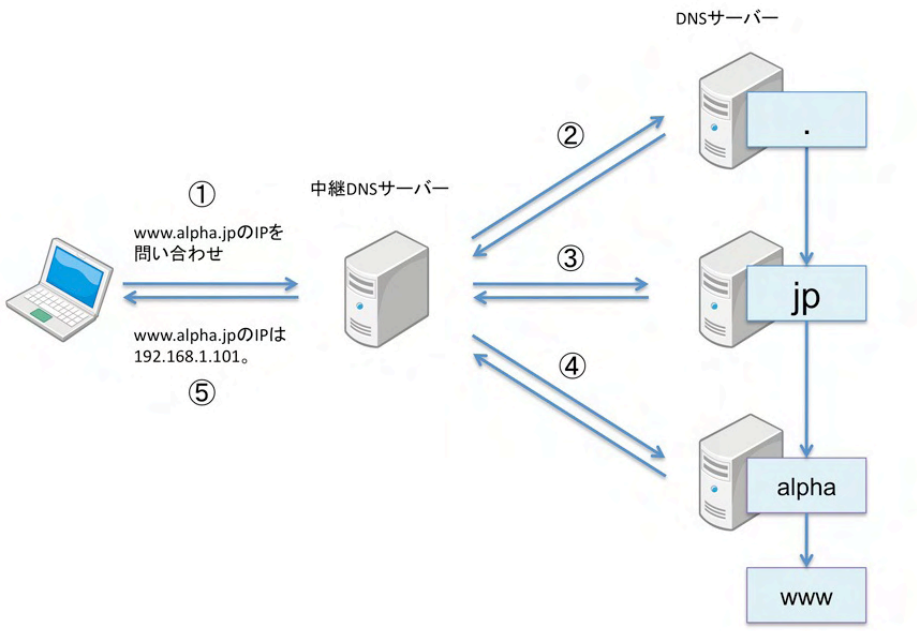
4. tokyo.example.co.jp ドメインは example.co.jp ドメインのサブドメイン

### 4.3.4 ドメイン名の取得

ドメイン名を取得するということは、上位のドメイン名の管理者にサブドメインを作ってもらい、管理権限を委譲してもらうということになります。短いドメイン名を取得したいのであればトップレベルドメインを管理している管理組織からサブドメイン化してもらうことになりますが、既にドメイン名を取得している管理者からサブドメインの管理権限を委譲してもらうこともできます。

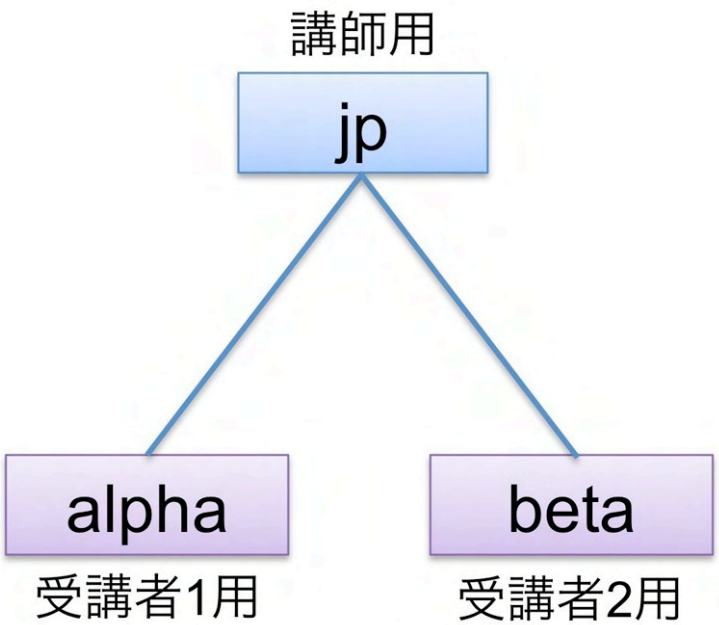
## 4.4 DNS を使った名前解決

DNS を使って名前を解決する、すなわち名前から IP アドレスを調べるには、マシンが自分の組織やプロバイダーで中継する DNS サーバーへ問い合わせ、中継する DNS サーバーが対応するトップレベルドメインの DNS サーバーへ問い合わせます。トップレベルの DNS サーバーは所属するサブドメインの DNS サーバーのアドレスを中継の DNS サーバーへ返します。中継する DNS サーバーは、サブドメインの DNS サーバーへ問い合わせ、マシンへ結果を返します。



## 4.5 これから構築する DNS の概略

各自が jp. ドメインのサブドメインを管理する DNS サーバーを作る演習を進めてもらうので、ドメインを管理する次の 3 台以上のマシンがある環境が望ましいです。




各マシンには、以下のような役割を割り当てます。

- 講師のマシン jp ドメインを受け持つ DNS サーバー、および再帰問い合わせのためのキャッシングネームサーバー
- 受講生 A のマシン alpha.jp ドメインを受け持つ DNS サーバー
- 受講生 B のマシン beta.jp ドメインを受け持つ DNS サーバー

講師マシンが JP ドメインを管理するので、教室の環境はインターネットに接続されている必要はありません。

以降の章 (特にメール) では DNS サーバーが正しく設定されていることを前提としているので、この章の演習内容が完全に終わっている必要があります。

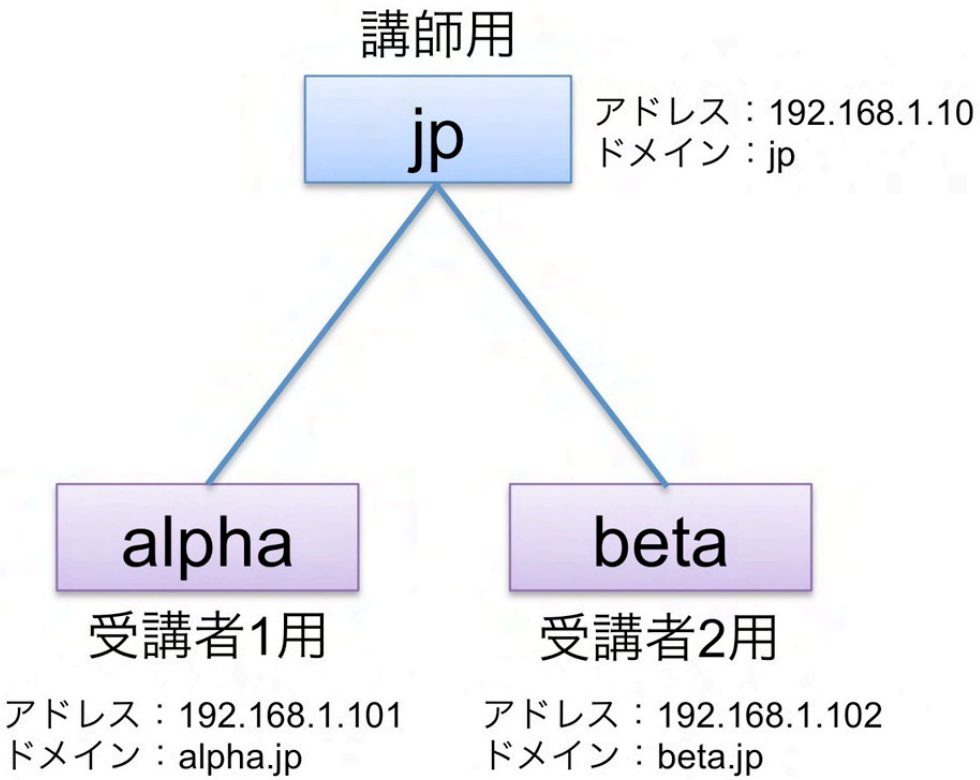



表 4.1　講師

| ドメイン名 | jp. |
|---|---|
| IP アドレス | 192.168.1.10 |

表 4.2　受講生 A

| ドメイン名 | alpha.jp. |
|---|---|
| IP アドレス | 192.168.1.101 |

表 4.3　受講生 B

| ドメイン名 | beta.jp. |
|---|---|
| IP アドレス | 192.168.1.102 |

### 4.5.1 アドレス解決の流れ

受講生 A のマシン (192.168.1.101) がホスト www.beta.jp を解決するときの動きを追ってみましょう。

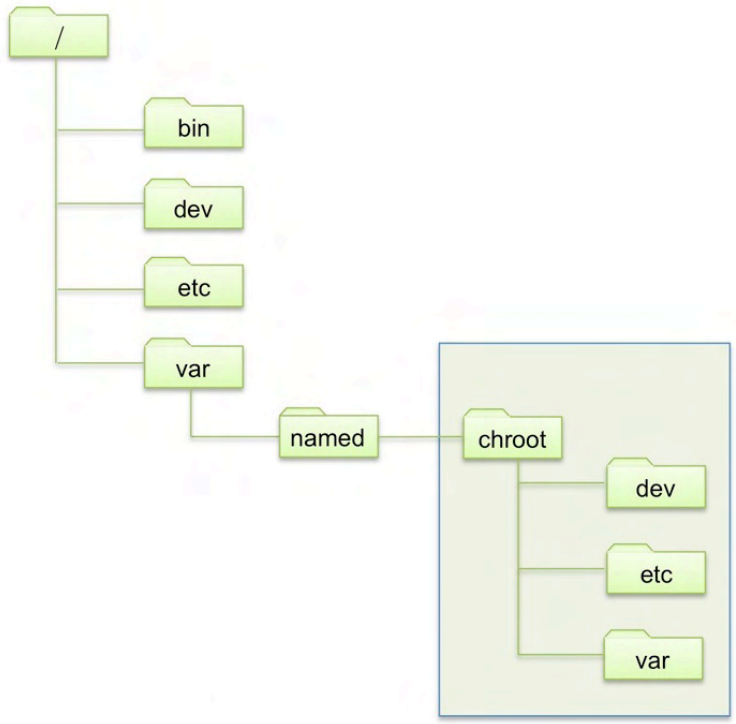
Web ブラウザーで Web ページを表示させるとき、DNS サーバーのことは特に意識せず Web サイトのアドレスを入力しています。ここでは Web アドレスを入力してリクエストしてページが表示されるまでの流れを例に、DNS がどのように動くのか簡単に説明します。

1. 受講生 A は、受講生 A のマシンの Web ブラウザーにアドレスとして www.beta.jp を入力します
2. Web ブラウザーは、Linux のリゾルバに問い合わせします
3. リゾルバは、/etc/resolv.conf ファイルで指定されている DNS サーバー（この場合は講師用サーバー:192.168.1.10）へ問い合わせます
4. 講師のマシンは、講師の DNS サーバーを参照します
5. 講師の DNS サーバーは beta.jp ドメインの DNS サーバーとその IP アドレス（ns.beta.jp → 192.168.1.102）を講師のマシンに返します
6. 講師のマシンは、受講生 B の DNS サーバーに問い合わせします
7. 受講生 B の DNS サーバーは、www.beta.jp ホストの IP アドレス（www.beta.jp → 192.168.1.102）を講師のマシンに返します
8. 講師のマシンは、結果を受講生 A のマシンへ返します
9. 受講生 A のマシンは、www.beta.jp に HTTP でアクセスし、Web ページを受け取って表示します

## 4.6 chroot 機能を利用した BIND のセキュリティ

chroot 機能はプログラムに対して特定のディレクトリ以外にはアクセスできないようにするための機能です。

chroot 機能を使って BIND を実行すると、bind プロセスは/var/named/chroot ディレクトリを/（ルート）ディレクトリとして動作します。たとえば、bind プロセスが/etc ディレクトリにアクセスしても、実際にアクセスされるのは/var/named/chroot/etc ディレクトリになります。




BINDのchroot機能を利用すると、namedからは枠線で囲った部分までしか
アクセスできなくなります。

DNS というサービスを提供している関係上、BIND はインターネット上の数多くのサーバーで実行されており、セキュリティの攻撃を受けやすくなっています。万が一、BIND がセキュリティ攻撃を受けて乗っ取られてしまったとしても、chroot 機能のおかげで bind プロセスがアクセスできるディレクトリを限定することができるので、システムのその他のファイルへのアクセスを妨げ、被害を最小限に食い止めることができます。

# 4.7 BIND のインストール

## 4.7.1 必要なパッケージを確認

DNS サーバーの構築に必要なパッケージを確認します。DNS の機能を提供するプログラムとして BIND があり、bind パッケージと bind-chroot パッケージが必要です。rpm コマンドに-q オプションとパッケージ名を指定し、2 つのパッケージがインストールされているか確認します。パッケージがインストールされている場合は次の様に表示されます。

```
# rpm -q bind bind-chroot
bind-9.7.3-8.P3.el6.i686
bind-chroot-9.7.3-8.P3.el6.i686
```

パッケージがインストールされていない場合は次の様に表示されます。

```
# rpm -q bind bind-chroot
パッケージ bind はインストールされていません。
パッケージ bind-chroot はインストールされていません。
```

## 4.7.2 必要なパッケージをインストール

DNS サーバーの構築に必要なパッケージがインストールされていないときは、rpm コマンドでインストールをします。

ここでは、自動マウント機能が動作していて DVD-ROM が自動的に/media/CentOS_6.2_Final にマウントされた状態でインストール作業を進めています。DVD-ROM の公開鍵を登録してからインストールすると警告メッセージが出ないので、インストールをする前に rpm コマンドに–import オプションと公開鍵のファイルを指定して登録しています。最後に、DVD-ROM のディレクトリ上で作業をしたので、cd コマンドでホームディレクトリへ移動しています。

```
# rpm --import /media/CentOS_6.2_Final/RPM-GPG-KEY-CentOS-6
# cd /media/CentOS_6.2_Final/Packages
# rpm -ivh bind-9.7.3-8.P3.el6.i686.rpm bind-chroot-9.7.3-8.P3.el6.i686.rpm
準備中...                ########################################### [100%]
   1:bind                ########################################### [ 50%]
   2:bind-chroot         ########################################### [100%]
# cd
```

### 4.7.3 chkconfig で起動時の設定

BIND をインストールして設定が終わると、Linux の起動時に BIND を必ず起動する必要が出てきます。Red Hat 系のディストリビューションの BIND パッケージには BIND を起動するための/etc/init.d/named スクリプトが用意されています。/etc/init.d/named スクリプトは chkconfig コマンドにより Linux 起動時に開始するか否かを設定できます。chkconfig コマンドに–list オプションを付けると、起動するか否かが確認できます。

```
# chkconfig --list named
named           0:off   1:off   2:off   3:off   4:off   5:off   6:off
```

named の後に出てくる数字はランレベルという起動するときのモードで、通常は 3 または 5 が使われます。chkconfig コマンドに named スクリプトの名前と on を指定し、BIND を起動するようにします。

```
# chkconfig named on
# chkconfig --list named
named           0:off   1:off   2:on    3:on    4:on    5:on    6:off
```

## 4.8 ドメインを設定する流れ

ドメイン (以降ではゾーンと記述) を追加するために必要な作業は次となります。

1. named.conf ファイルにゾーンを追加
2. ゾーンファイルを記述

ドメインに含まれるゾーン (ゾーン⊆ドメイン) を DNS サーバーである BIND で取り扱うために、BIND の基本的な設定ファイルである/etc/named.conf ファイルがあります。/etc/named.conf に基本的な設定とゾーンの定義を追加したら、ゾーンの詳細を定義するゾーンファイルを/var/named ディレクトリに作ります。

### 4.8.1 正引きのゾーンの追加

BIND の設定ファイルの 1 つとなる named.conf に、基本的な設定とゾーン定義を追加します。

```
# vi /etc/named.conf
```

```
//
// named.conf
//
// Provided by Red Hat bind package to configure the ISC BIND named(8) DNS
// server as a caching only nameserver (as a localhost DNS resolver only).
//
// See /usr/share/doc/bind*/sample/ for example named configuration files.
//

options {
        listen-on port 53 { 127.0.0.1;  192.168.1.101; };
        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";
         allow-query     { any; };
        recursion yes;

 //    dnssec-enable yes; ← コメントアウト
 //    dnssec-validation yes; ← コメントアウト
 //    dnssec-lookaside auto; ← コメントアウト

        /* Path to ISC DLV key */
        bindkeys-file "/etc/named.iscdlv.key";
};

logging {
        channel default_debug {
                file "data/named.run";
                severity dynamic;
        };
};

zone "." IN {
        type hint;
        file "named.ca";
};

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

 zone "alpha.jp" IN {
          type master;
          file "alpha.jp.zone";
          allow-update { none; };
 };
```

named.conf ファイルは 127.0.0.1(ローカルループバックインターフェース) への問い合わせにしか返答しない設定なので、外部からの問い合わせを受けられるように、192.168.1.101;(物理的なネットワークインターフェースの eth0) を listen-on に追加します。

受講生ドメインの alpha.jp のゾーン定義は zone "alpha.jp" IN {〜}; を追記します。alpha.jp ゾーンの詳細な情報は、ゾーンファイルとして/var/named/alpha.jp.zone 記述します。ゾーンファイルを編集時、よくあるミスとしては、括弧の不足、セミコロンの不足などがあります。編集後、BIND を起動する前に編集したゾーンファイルに間違いがないかよく確認しましょう。

なお、named.conf で「allow-query { any; };」と設定しましたが、この設定はどこから問い合わせがあったとしても DNS 問い合わせに応答するという設定になっており（実際はサーバーが外部公開されているか否かによって問題にならない場合もありますが）、実際の運用ではセキュリティの観点から修正が必要です。詳しくは後述する「allow-query の設定について」をご覧ください。

### 4.8.2 ゾーンファイルの作成

named.conf で定義したゾーンの内容を記述するゾーンファイルの作成を行います。

1. ゾーンファイルのお手本となる/var/named/named.empty ファイルをコピーします。

```
# cd /var/named
# cp -p named.empty alpha.jp.zone
# ls -l alpha.jp.zone
-rw-r----- 1 root named 317  4 月  9 15:19 2012 alpha.jp.zone
```

2. コピーした/var/named/alpha.jp.zone ファイルを修正します。

```
# vi /var/named/alpha.jp.zone
```

```
$TTL 3H
 $ORIGIN alpha.jp.
@       IN SOA   host1 root (
                                        2012040601      ; serial
                                        1D      ; refresh
                                        1H      ; retry
                                        1W      ; expire
                                        3H )    ; minimum
     NS     host1.alpha.jp.
     MX 10  host1.alpha.jp.

 host1   A     192.168.1.101
 www     A     192.168.1.101
 mail    A     192.168.1.101
 vhost1  A     192.168.1.101
 vhost2  A     192.168.1.101
```

変更が発生するゾーンファイルは SOA レコードのシリアルナンバー (serial) を、西暦 (4 桁の年) と月日 (2 桁ずつ) の後に 01 から 99 までの数字 (2 桁) が付いた 10 桁の数字で指定します。

MX レコードは受講生ドメインのメールサーバーを定義します。

NS レコードや MX レコードの定義では、右側に FQDN を入れるので、最後に必ず「.」を付けてください。

A レコードで名前と IP アドレスの対応を定義する箇所は、左側にホスト名、右側に IP アドレスが入ります。受講生マシンのホスト名である host1 や、以降の章で使うサーバーの名前である www や mail、vhost1、vhost2 と IP アドレスへの対応を記述しました。最後に「.」が付かない名前には、$ORIGIN で定義しているゾーン名 (ここでは alpha.jp.) が自動的に追加されます。

### 4.8.3 BIND の IPv6 の無効化

BIND は IPv6 を使用している際の名前解決にも対応していますが、本教科書では IPv4 のみ使用しているので、IPv6 対応の機能を無効にしておきます。以下のように、/etc/sysconfig/named に BIND を IPv4 のみ使用するオプションを設定します。

```
# echo 'OPTIONS="-4"' >> /etc/sysconfig/named
```

## 4.9 BIND を起動

各自のドメインを定義したら BIND を起動してみましょう。BIND の起動は/etc/init.d/named スクリプトに start オプションを指定します。

```
# /etc/init.d/named start
named を起動中:                                            [  OK  ]
```

既に BIND を起動してあるのに止めないで BIND を開始するとエラーとなるので、この場合は/etc/init.d/named スクリプトに restart オプションを付けて BIND を再起動します。

```
# /etc/init.d/named restart
named を停止中:                                            [  OK  ]
named を起動中:                                            [  OK  ]
```

### 4.9.1 BIND 起動の確認

BIND が正しく起動したか確認するには、ログファイルを参照すると早いでしょう。/var/log/messages ログファイルを見てください。

```
# tail /var/log/messages
（略）
Apr  9 09:57:03 host1 named[19971]: automatic empty zone: 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0
Apr  9 09:57:03 host1 named[19971]: automatic empty zone: D.F.IP6.ARPA
Apr  9 09:57:03 host1 named[19971]: automatic empty zone: 8.E.F.IP6.ARPA
Apr  9 09:57:03 host1 named[19971]: automatic empty zone: 9.E.F.IP6.ARPA
Apr  9 09:57:03 host1 named[19971]: automatic empty zone: A.E.F.IP6.ARPA
```

```
Apr  9 09:57:03 host1 named[19971]: automatic empty zone: B.E.F.IP6.ARPA
Apr  9 09:57:03 host1 named[19971]: automatic empty zone: 8.B.D.0.1.0.0.2.IP6.ARPA
Apr  9 09:57:03 host1 named[19971]: command channel listening on 127.0.0.1#953
Apr  9 09:57:03 host1 named[19971]: command channel listening on ::1#953
Apr  9 09:57:03 host1 named[19971]: zone 0.in-addr.arpa/IN: loaded serial 0
Apr  9 09:57:03 host1 named[19971]: zone 1.0.0.127.in-addr.arpa/IN: loaded serial 0
Apr  9 09:57:03 host1 named[19971]: zone 1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.
Apr  9 09:57:03 host1 named[19971]: zone alpha.jp/IN: loaded serial 2012040601
Apr  9 09:57:03 host1 named[19971]: zone localhost.localdomain/IN: loaded serial 0
Apr  9 09:57:03 host1 named[19971]: zone localhost/IN: loaded serial 0
Apr  9 09:57:03 host1 named[19971]: managed-keys-zone ./IN: loaded serial 36
Apr  9 09:57:03 host1 named[19971]: running
（略）
```

ログファイルの各行の先頭は日時で始まっており、日時の後の zone がある箇所に続けて追加したドメイン名がある行でシリアル番号 (serial) までが表示されていればドメインの登録は正常です。シリアル番号が出ていないときはエラーなので対処します。

### 4.9.2 BIND 起動がエラーになった場合

次はゾーンファイル名を beta.jp.zone と間違えて、ファイルがないというエラーが出た場合です。

```
# /etc/init.d/named restart
named を停止中:                                            [  OK  ]
named を起動中:
Error in named configuration:
zone localhost.localdomain/IN: loaded serial 0
zone localhost/IN: loaded serial 0
zone 1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.ip6.arpa/IN: loade
zone 1.0.0.127.in-addr.arpa/IN: loaded serial 0
zone 0.in-addr.arpa/IN: loaded serial 0
zone alpha.jp/IN: loading from master file beta.jp.zone failed: file not found
zone alpha.jp/IN: not loaded due to errors.
_default/alpha.jp/IN: file not found
                                                           [失敗]

# tail /var/log/messsages
（略）
Apr  9 10:08:16 host1 named: zone alpha.jp/IN: loading from master file beta.jp.zone fa
Apr  9 10:08:16 host1 named: zone alpha.jp/IN: not loaded due to errors.
Apr  9 10:08:16 host1 named: _default/alpha.jp/IN: file not found
```

他にも listen-on port の{〜}にアドレスを追加するときに、IP アドレスの後にセミコロンを入れ忘れたり、allow-query { trust; }; のように、{ }の中と外にセミコロンを入れ忘れたりしないよう注意します。

## 4.10 名前解決の確認

BIND が起動したら、名前解決が正常に行われるかを確認します。名前解決の確認には、nslookup コマンドと dig コマンドが使用できます。

### 4.10.1 nslookup コマンドで名前を確認

nslookup コマンドで名前から IP アドレスを確認します。

```
# cat /etc/resolv.conf
# Generated by NetworkManager
search alpha.jp
nameserver 192.168.1.101

# nslookup host1.alpha.jp
Server:         192.168.1.101
Address:        192.168.1.101#53

Name:   host1.alpha.jp
Address: 192.168.1.101

# nslookup www.alpha.jp
Server:         192.168.1.101
Address:        192.168.1.101#53

Name:   www.alpha.jp
Address: 192.168.1.101

# nslookup mail.alpha.jp
Server:         192.168.1.101
Address:        192.168.1.101#53

Name:   mail.alpha.jp
Address: 192.168.1.101
```

ここでは/etc/resolv.conf ファイルの定義をリゾルバ経由で使い、自分自身へ問い合わせています。DNS サーバーを指定して問い合わせたい場合は nslookup コマンドにホスト名を指定した後に問い合わせたい DNS サーバーの IP アドレスを付けて問い合わせてみてください。

```
# nslookup host1.alpha.jp 192.168.1.101
Server:         192.168.1.101
Address:        192.168.1.101#53

Name:   host1.alpha.jp
Address: 192.168.1.101

# nslookup host2.beta.jp 192.168.1.102
Server:         192.168.1.102
Address:        192.168.1.102#53
```

```
Name:   host2.beta.jp
Address: 192.168.1.102
```

### 4.10.2 dig コマンドでドメインを確認

dig コマンドでゾーン情報を確認してみてください。

```
# dig alpha.jp axfr

; <<>> DiG 9.7.3-P3-RedHat-9.7.3-8.P3.el6_2.2 <<>> alpha.jp axfr
;; global options: +cmd
alpha.jp.               10800   IN      SOA     host1.alpha.jp. root.alpha.jp. 20120406
alpha.jp.               10800   IN      NS      host1.alpha.jp.
alpha.jp.               10800   IN      MX      10 host1.alpha.jp.
host1.alpha.jp.         10800   IN      A       192.168.1.101
mail.alpha.jp.          10800   IN      A       192.168.1.101
vhost1.alpha.jp.        10800   IN      A       192.168.1.101
vhost2.alpha.jp.        10800   IN      A       192.168.1.101
www.alpha.jp.           10800   IN      A       192.168.1.101
alpha.jp.               10800   IN      SOA     host1.alpha.jp. root.alpha.jp. 20120406
;; Query time: 0 msec
;; SERVER: 192.168.1.101#53(192.168.1.101)
;; WHEN: Mon Apr  9 10:27:55 2012
;; XFR size: 9 records (messages 1, bytes 242)
```

ドメイン名の後に axfr を指定するとゾーンに登録されている全ての情報が表示されます。

dig コマンドの結果に、ANSWER SECTION があれば正常であり、ANSWER SECTION が無ければ結果が返らない状態のエラーです。

```
# dig alpha.jp ns

; <<>> DiG 9.7.3-P3-RedHat-9.7.3-8.P3.el6_2.2 <<>> alpha.jp ns
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 47234
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; QUESTION SECTION:
;alpha.jp.                      IN      NS

;; ANSWER SECTION:
alpha.jp.               10800   IN      NS      host1.alpha.jp.

;; ADDITIONAL SECTION:
host1.alpha.jp.         10800   IN      A       192.168.1.101

;; Query time: 0 msec
;; SERVER: 192.168.1.101#53(192.168.1.101)
;; WHEN: Mon Apr  9 10:28:31 2012
;; MSG SIZE  rcvd: 62
```

ドメイン名の後に ns を指定するとドメインに登録されている NS レコード (ネームサーバーの情報) が表示されます。

```
# dig alpha.jp mx

; <<>> DiG 9.7.3-P3-RedHat-9.7.3-8.P3.el6_2.2 <<>> alpha.jp mx
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 22254
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; QUESTION SECTION:
;alpha.jp.                      IN      MX

;; ANSWER SECTION:
alpha.jp.               10800   IN      MX      10 host1.alpha.jp.

;; AUTHORITY SECTION:
alpha.jp.               10800   IN      NS      host1.alpha.jp.

;; ADDITIONAL SECTION:
host1.alpha.jp.         10800   IN      A       192.168.1.101

;; Query time: 0 msec
;; SERVER: 192.168.1.101#53(192.168.1.101)
;; WHEN: Mon Apr  9 10:28:58 2012
;; MSG SIZE  rcvd: 78
```

ドメイン名の後に mx を指定するとドメインに登録されている MX レコード (メールサーバーの情報) が表示されます。

ここでは/etc/resolv.conf ファイルの定義を使い自分自身へ問い合わせています。DNS サーバーを指定して問い合わせたい場合は、dig コマンドに@と DNS サーバーのアドレスを付けて問い合わせてください。

```
# dig alpha.jp @192.168.1.101 axfr
# dig beta.jp @192.168.1.102 axfr
```

ここまでは、以降へ続く設定となるので、必ず名前が解決できる必要があります。受講生ドメインのゾーン情報を nslookup コマンドと dig コマンドまで確認できたら、次へ進んでください。

## 4.11 ドメイン情報を公開

各受講生が設定した DNS サーバーのドメインは JP ドメインの下位ドメインとなる alpha.jp と beta.jp です。他の受講者からドメイン情報の問い合わせを受けるには、親となる JP ドメインを管理する DNS サーバー (講師サーバー) に受講生ドメインを登録してもらう必要があります。この時に上位 DNS に登録するのが、ドメイン名と IP アドレスをつなげる「グルーレコード」です。

### 4.11.1 講師マシンの DNS 設定

受講生マシンの/etc/named.conf と同じく、講師マシンの/etc/named.conf に JP ドメインの定義と問い合わせに応じる設定を修正します。

講師マシンの/etc/named.conf への追加

```
//
// named.conf
//
// Provided by Red Hat bind package to configure the ISC BIND named(8) DNS
// server as a caching only nameserver (as a localhost DNS resolver only).
//
// See /usr/share/doc/bind*/sample/ for example named configuration files.
//

options {
        listen-on port 53 { 127.0.0.1;  192.168.1.10; };
        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";
         allow-query    { any; };
        recursion yes;

 //    dnssec-enable yes; ← コメントアウト
 //    dnssec-validation yes; ← コメントアウト
 //    dnssec-lookaside auto; ← コメントアウト

        /* Path to ISC DLV key */
        bindkeys-file "/etc/named.iscdlv.key";
};

logging {
        channel default_debug {
                file "data/named.run";
                severity dynamic;
        };
};

zone "." IN {
        type hint;
        file "named.ca";
};

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

 zone "jp" IN {
          type master;
          file "jp.zone";
          allow-update { none; };
 };
```

JP ドメインのゾーンファイルも/var/named/named.empty をコピーして作ります。受講生マシンのゾーンファイルと違うところは、受講生ドメインと受講生ドメインの DNS サーバーの組を複数定義するところです。

```
# cd /var/named
# cp -p named.empty jp.zone
# ls -l jp.zone
-rw-r-----. 1 root named 152 12 月 15 21:27 2009 jp.zone
```

ゾーンファイルをコピーした時に-p オプションをつけるのは、BIND がファイルの所有権により起動できなくなるのを回避するためにグループ所有権である named を引き継がせるためです。ゾーンファイルを作成したら、講師のゾーンファイルに生徒のドメインを追加します。JP ドメインに登録することで、同じネットワーク中にあるマシンから設定したアドレスでアクセスできるようになります。

/var/named/jp.zone ファイルの変更、追加

```
$TTL 3H
$ORIGIN jp.
@       IN SOA  host0 root (
                                        2012040601; serial
                                        1D      ; refresh
                                        1H      ; retry
                                        1W      ; expire
                                        3H )    ; minimum
                NS      host0.jp.
alpha           NS      host1.alpha.jp.
beta            NS      host2.beta.jp.

host0           A       192.168.1.10
host1.alpha.jp. A       192.168.1.101
host2.beta.jp.  A       192.168.1.102
```

### 4.11.2 JP ドメインの DNS サーバーの再起動

講師マシンに JP ドメインと下位ドメインの定義を追加したら、JP ドメインを管理する講師マシンの BIND を再起動します。

```
# /etc/init.d/named restart
named を停止中:                                            [  OK  ]
named を起動中:                                            [  OK  ]

# tail /var/log/messages
Apr  9 10:47:02 host0 named[12950]: none:0: open: /etc/rndc.key: file not found
Apr  9 10:47:02 host0 named[12950]: couldn't add command channel ::1#953: file not foun
Apr  9 10:47:02 host0 named[12950]: zone 0.in-addr.arpa/IN: loaded serial 0
Apr  9 10:47:02 host0 named[12950]: zone 1.0.0.127.in-addr.arpa/IN: loaded serial 0
```

```
Apr  9 10:47:02 host0 named[12950]: zone 1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.
 Apr  9 10:47:02 host0 named[12950]: zone jp/IN: loaded serial 2012040601
Apr  9 10:47:02 host0 named[12950]: zone localhost.localdomain/IN: loaded serial 0
Apr  9 10:47:02 host0 named[12950]: zone localhost/IN: loaded serial 0
Apr  9 10:47:02 host0 named[12950]: managed-keys-zone ./IN: loaded serial 3
Apr  9 10:47:02 host0 named[12950]: running
```

ログファイルの日時の後の zone がある箇所に続けて追加したドメイン名がある行でシリアル番号 (serial) までが表示されていればドメインの登録は正常です。シリアル番号が出ていないときはエラーなので対処します。

### 4.11.3 参照する DNS サーバーの変更

JP ドメインの DNS サーバーを作ったので、受講生マシンから講師マシンを経由すれば全ての下位ドメインを問い合わせできます。受講生マシンの/etc/resolv.conf ファイルに JP ドメインの DNS サーバーである講師マシンのを指定します。

/etc/resolv.conf の修正

```
# Generated by NetworkManager
search alpha.jp
nameserver 192.168.1.10
```

resolv.conf を編集したら、named サービスを再起動します。

```
#  /etc/init.d/named restart
```

### 4.11.4 ネットワークインターフェースの DNS 設定の削除

CentOS 6.2 では、/etc/resolv.conf の他に各ネットワークインターフェースの設定ファイルも参照する DNS の情報を持っています。ネットワークインターフェースの設定の方が/etc/resolv.conf の設定よりも優先されるので、再起動時などに/etc/resolv.conf の内容が上書きされてしまいます。このようなことを防ぐために、以下のように、/etc/sysconfig/network-scripts/ifcfg-eth0 にある DNS1 の設定を削除します。

```
#  vi /etc/sysconfig/network-scripts/ifcfg-eth0
```

```
DEVICE="eth0"
NM_CONTROLLED="yes"
ONBOOT=yes
```

```
HWADDR=00:1C:42:97:1C:DA
TYPE=Ethernet
BOOTPROTO=none
IPADDR=192.168.1.101
PREFIX=24
GATEWAY=192.168.1.1
 DNS1=192.168.1.101 ← この行を削除
DEFROUTE=yes
IPV4_FAILURE_FATAL=yes
IPV6INIT=no
NAME="System eth0"
UUID=5fb06bd0-0bb0-7ffb-45f1-d6edd65f3e03
```

### 4.11.5 名前解決の確認

nslookup コマンドで JP ドメインサーバーへ自分のドメインの NS レコードや MX レコードを問い合せてみてください。

```
# nslookup
> host0.jp
Server:         192.168.1.10
Address:        192.168.1.10#53

Name:   host0.jp
Address: 192.168.1.10
> host1.alpha.jp
Server:         192.168.1.10
Address:        192.168.1.10#53

Non-authoritative answer:
Name:   host1.alpha.jp
Address: 192.168.1.101

> set q=soa
> alpha.jp

Server:         192.168.1.10
Address:        192.168.1.10#53

Non-authoritative answer:
alpha.jp
origin = host1.alpha.jp
mail addr = root.alpha.jp
serial = 2009032401
refresh = 10800
retry = 900
expire = 604800
minimum = 86400

Authoritative answers can be found from:
alpha.jp        nameserver = host1.alpha.jp.
host1.alpha.jp   internet address = 192.168.1.101
```

```
> set q=ns
> alpha.jp

Server:         192.168.1.10
Address:        192.168.1.10#53

Non-authoritative answer:
alpha.jp        nameserver = host1.alpha.jp.

Authoritative answers can be found from:
host1.alpha.jp   internet address = 192.168.1.101

> set q=mx
> alpha.jp

Server:         192.168.1.10
Address:        192.168.1.10#53

Non-authoritative answer:
alpha.jp        mail exchanger = 10 host1.alpha.jp.

Authoritative answers can be found from:
alpha.jp        nameserver = host1.alpha.jp.
host1.alpha.jp   internet address = 192.168.1.101
> exit
```

例のように、正しい結果が出たでしょうか? DNS 機能は以降の章に少なからず必要となり、メール機能は特に DNS とリンクして動きます。DNS の設定は、ファイルの編集如何によって予想できないエラーが出るかもしれませんが、確実に動作する様にしてから進んでください。

## 4.12 rndc の設定

rndc の設定を行うと、ローカル、またはリモートから BIND のコントロールが行えるようになります。設定が行われていない時点で起動スクリプトに status コマンドを与えて実行すると、以下のように表示されます。

```
# /etc/init.d/named status
rndc: connect failed: 127.0.0.1#953: connection refused
named (pid  9222) を実行中...
```

設定は rndc-confgen コマンドを実行して行います。

```
# rndc-confgen -a -c /etc/rndc.key
# chown named.named /etc/rndc.key
# chmod 600 /etc/rndc.key
# cat /etc/rndc.key
key "rndc-key" {
        algorithm hmac-md5;
        secret "OcSjv2NYvE9kOViZujQw9g==";
```

```
};
```

/etc/rndc.key ファイルが生成されたことを確認したら、BIND を再起動し、rndc が正しく動作することを確認します。

```
# /etc/init.d/named restart
named を停止中:                                            [  OK  ]
named を起動中:                                            [  OK  ]
# /etc/init.d/named status
version: 9.7.3-P3-RedHat-9.7.3-8.P3.el6
CPUs found: 1
worker threads: 1
number of zones: 20
debug level: 0
xfers running: 0
xfers deferred: 0
soa queries in progress: 0
query logging is OFF
recursive clients: 0/0/1000
tcp clients: 0/100
server is up and running
named (pid  4600) を実行中...
```

## 4.13 DNS のセキュリティ

本教科書ではサーバー構築の方法をよりわかりやすく説明するため、DNS のセキュリティについていくつかの設定を緩めて設定していますが、実際の本番環境を構築する際にはよりセキュリティを高めた設定が必要となります。ここでは DNS のセキュリティに関係する基本的な設定について解説します。

### 4.13.1 allow-query の設定

すべての IP からの DNS 問い合わせに対して応答するように、「allow-query { any; };」と設定しました。しかし、DNS サーバーを実運用で動かす場合、いかなる場合も内部情報を表示できてしまうのは好ましい状態ではないため、named.conf 設定ファイルでアドレス許可リストを設定して DNS 問い合わせに対する結果表示を規制することが推奨されます。

named.conf を開いて、以下のように修正します。

```
options{
 (略)
        //allow-query { any; } ; ← コメントアウト}
        allow-query { trust; } ; ← 許可リスト trust で指定した端末からは問合わせ許可}
 (略)
};
```

```
acl trust {
        192.168.1.0/24;
        localhost;
};
（略）
```

このように設定すると、ネットワークアドレス 192.168.1.0 配下にある端末と DNS サーバーが動いている localhost（127.0.0.1）以外からの外部からのアクセスから来たクエリには応答しなくなります。また、localhost と特定の IP からのみ DNS 問い合わせを実行できるようにするには、アクセス許可リストに以下のように許可したい IP を列挙していきます。

```
acl trust {
        192.168.1.102;
        localhost;
};
```

上記のように設定した場合は、たとえば同じネットワークアドレス上にある 192.168.1.101 の端末からの DNS 問い合わせについてはエラーとなります。

### 4.13.2 allow-recursion の設定

再帰的なクエリを許可するホストを指定します。デフォルトは、すべてのホストからの再帰的クエリを許可されます。下記のように設定することで、再帰的なクエリを localhost と ネットワークアドレス 192.168.1.0 以下にある端末だけからのみ許可できます。allow-recursion の設定は options {}; の中に記述します。

```
options {
        allow-recursion {
                localhost;
                192.168.1.0/24;
        ;
};
```

### 4.13.3 allow-transfer の設定

ゾーン転送を許可するホストを指定します。デフォルトでは、すべてのホストへのゾーン転送が許可されます。zone ステートに allow-transfer オプションが記述された場合、options ステートメントの設定を上書きできます。下記のように設定することで、ゾーン転送を localhost とネットワークアドレス 192.168.1.0 以下にある端末だけからのみ許可できます。

```
options {
        allow-transfer {
```

```
                localhost;
                192.168.1.0/24;
        };
};
```

# 第5章

# Web サーバーの構築

ホームページや Web システムを公開するための Web サービスを設定します。基本としては、アクセス制限をかける設定と動作を確認します。応用としては、Web システムの開発に広く使われている PHP 言語のインストールと動作確認、サーバーとして広く使われているバーチャルホスト機能にも触れてもらいます。

## 5.1 用語集

### HTML(HyperText Markup Language)

Web サーバー用のドキュメントを書くためのタグを使って文章を構造的に記述できるマークアップ言語です。他ドキュメントへのハイパーリンクを書いたり、画像利用したり、リストや表などの高度な表現も可能です。

### HTTP(HyperText Transfer Protocol)

Web ブラウザーと Web サーバーの間で HTML などのコンテンツ (データ) 送受信に使われる通信手順です。ファイルのリクエスト (要求) とファイルのレスポンス (返送) が組でセッションになります。

### Apache Web サーバー

世界中でもっとも使われている Web サーバーであり、大規模な商用サイトから自宅サーバーまで幅広く利用されています。Apache ソフトウェア財団の Apache HTTP サーバープロジェクトで行われている、オープンソースソフトウェアです。

### セッション

通信の接続を確立してから切断するまでを一つのセッションといいます。

### ディレクティブ

Apache の設定ファイルで Apache の動作を設定する項目名です。

### BASIC 認証

ユーザー名とパスワードを使い、HTTP で定義された認証方式です。ユーザー名とパスワードの組みを:でつなぎ、Base64 でエンコードして送信します。盗聴や改竄が簡単であるという欠点があ

りますが、ほぼ全ての Web サーバーおよび Web ブラウザーが対応しており、広く使われています。

### DIGEST 認証

ユーザー名とパスワードを使う認証方式です。ユーザー名とパスワードを MD5 でハッシュ化して送るため、盗聴や改竄を防げるため推奨される認証方式ですが、Web ブラウザーや端末によって対応していないものがあるために注意が必要です。

### スクリプト言語

コードの作成や修正が容易と見なされるプログラミング言語です。CGI や Web アプリケーションの開発につかわれており、例えば Perl, PHP, Python, Ruby, JavaScript などがあります。

### モジュール

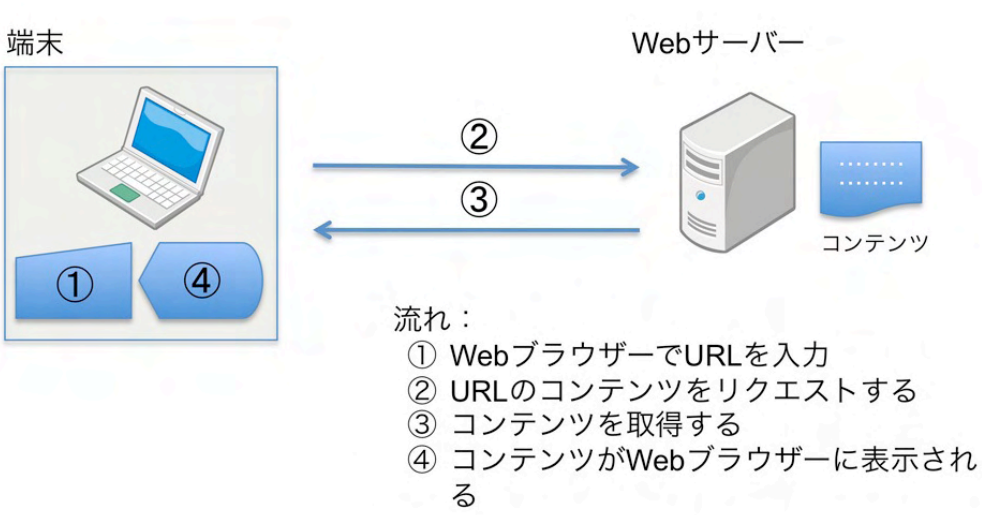
Apache に様々な機能を組み込むことができる拡張プログラムです。mod_○○.so という名前のライブラリとなっています。

### URL(Uniform Resource Locator)

インターネット上のリソースを指定するための記述方法で、ホームページのアドレスやメールのアドレスなどを指定できます。リソースを特定するスキーム名とアドレスを://でつないで書きます。

## 5.2 Web サーバーの仕組み

Web システムとは、インターネット環境で最も代表的なクライアントサーバーシステムで、クライアントの Web ブラウザーと Web サーバーから構成されます。Web サーバーは要求されたファイルを Web クライアントに提供し、クライアントは受け取ったファイルを表示します。提供される情報はテキストから画像やムービーと幅広く、クライアントが対応しているデータならば広く扱えます。Web システムの文章データとしては XML ベースの規格である XHTML や従来の HTML などが一般的に使われています。Web サーバーとして Apache が使われている割合はかなり高いでしょう。




Web サーバーと Web ブラウザーは HTTP 形式の通信手順でデータを転送します。転送されるデータはテキストや画像などのファイルで、HTML 形式 (フォーマット) のファイルが標準形式として使われています。HTML 形式のファイル内には他のページや他のファイルや他の Web サーバーへのリンクを持ち、HTML 形式のデータはクモの巣 (web) 状にデータがつながるのも特徴です。

## 5.3 これから構築する Web サーバーの概略

各自が受講生マシンで Web サーバーを起ち上げ、Web ブラウザーから隣の受講生マシンの Web サーバーへのアクセスを試します。名前の解決には 4 章で構築した講師の DNS サーバーと隣の受講生マシンの DNS サーバーを利用します。Web システムはクライアントとサーバーが同一のマシンに混在しても、2 台以上で構成されても動作に違いはありませんが、テスト確認を曖昧としないために 2 台ペアでの演習を行いましょう。まず、実習を始める前に、Web サーバーのデフォルト設定に問題がなく、Web サーバーとして着実に動くかを確認します。必要なパッケージが導入されているか、設定ファイルが正しく設定されているかなどを確認するため、Web サーバーを起動して Web ブラウザーでアクセスしたり、Apache のログファイルを確認するなどして、Web サーバーが正常に起動していることを確認します。

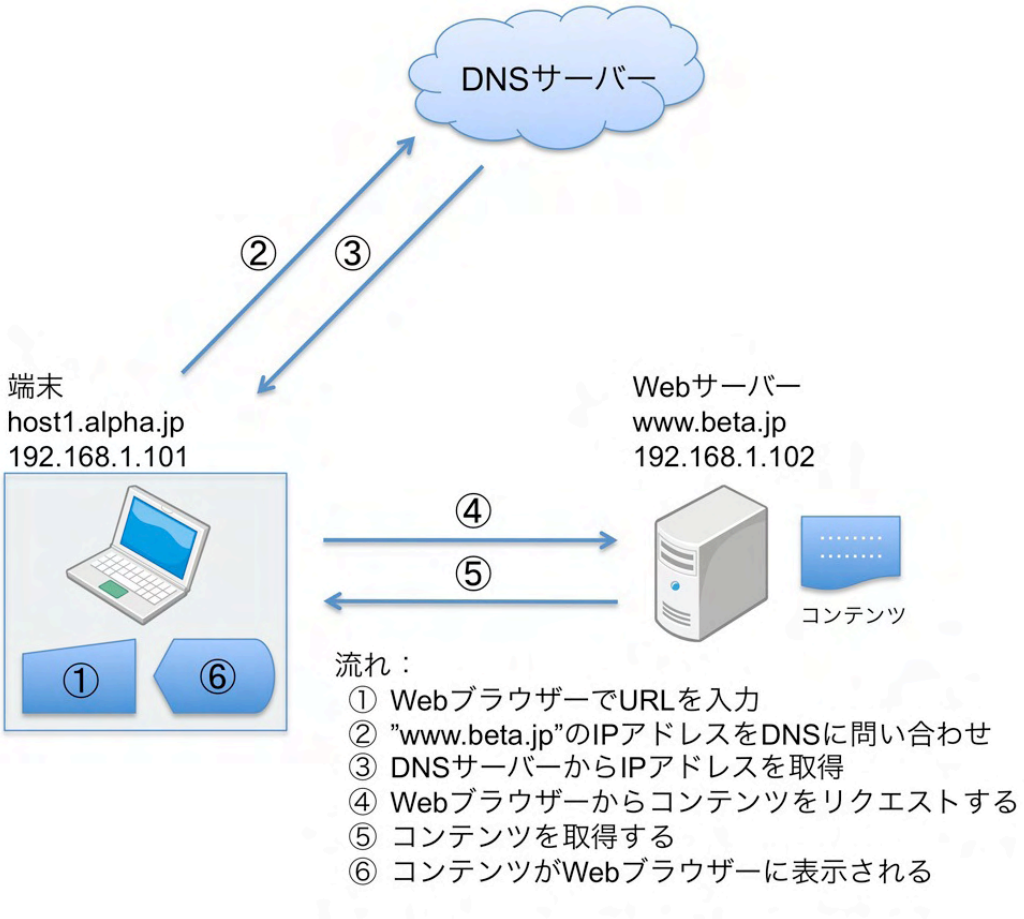


表 5.1　講師

| | |
|---|---|
| **ホスト名** | host0.jp |
| **IP アドレス** | 192.168.1.10 |

表 5.2　受講生 A

| | |
|---|---|
| **役割** | Web サーバー |
| **ホスト名** | host1.alpha.jp |
| **IP アドレス** | 192.168.1.101 |

表 5.3　受講生 B

| | |
|---|---|
| **役割** | Web ブラウザー |
| **ホスト名** | host2.beta.jp |
| **IP アドレス** | 192.168.1.102 |

## 5.4　Web サーバーの設定

Web サーバーの動作に必要なパッケージおよび設定ファイルを確認し、サービスを起動して動作を確認します。

### 5.4.1　必要なパッケージを確認

rpm コマンドで必要なパッケージがインストールされているかを確認します。

```
# rpm -q httpd
httpd-2.2.15-15.el6.centos.i686
```

パッケージがインストールされていない場合は次の様に表示されます。

```
# rpm -q httpd
パッケージ httpd はインストールされていません。
```

### 5.4.2　必要なパッケージをインストール

Web サーバーの構築に必要なパッケージがインストールされていないときは、rpm コマンドでインストールをします。

ここでは、自動マウント機能が動作していて DVD-ROM が自動的に/media/CentOS_6.2_Final にマウントされた状態でインストール作業を進めています。DVD-ROM の公開鍵を登録してからイ

ンストールすると警告メッセージが出ないので、インストールをする前に rpm コマンドに–import オプションと公開鍵のファイルを指定して登録しています。最後に、DVD-ROM のディレクトリ上で作業をしたので、cd コマンドでホームディレクトリへ移動しています。

```
# rpm --import /media/CentOS_6.2_Final/RPM-GPG-KEY-CentOS-6
# cd /media/CentOS_6.2_Final/Packages
# rpm -ivh httpd-2.2.15-15.el6.centos.i686.rpm
準備中...                ########################################### [100%]
   1:httpd                  ########################################### [100%]
# cd
```

### 5.4.3 chkconfig で起動時の設定

インストールされている Apache が確認できたら、Linux の起動時に Apache を必ず起動する必要が出てきます。Red Hat 系のディストリビューションの httpd パッケージには Apache を起動するための/etc/init.d/httpd スクリプトが用意されています。/etc/init.d/httpd スクリプトは chkconfig コマンドにより Linux 起動時に開始するか否かを設定できます。chkconfig コマンドに–list オプションを付けると、起動するか否かが確認できます。

```
# chkconfig --list httpd
httpd           0:off   1:off   2:off   3:off   4:off   5:off   6:off
```

httpd の後に出てくる数字はランレベルという起動するときのモードで、通常は 3 または 5 が使われます。chkconfig コマンドに httpd スクリプトの名前と on を指定し、Linux がブートする時に Apache を起動するようにします。

```
# chkconfig httpd on
# chkconfig --list httpd
httpd           0:off   1:off   2:on    3:on    4:on    5:on    6:off
```

### 5.4.4 設定ファイルを確認

Apache の設定ファイルは/etc/httpd/conf/httpd.conf ファイルと/etc/httpd/conf.d ディレクトリにある拡張子が conf のファイルです。設定ファイルで、先頭が#の行はコメント、ディレクティブの後にはスペースやタブで区切った設定内容の文字列を書きます。httpd.conf ファイルを見てみてください。

```
# cat /etc/httpd/conf/httpd.conf
（略）
```

```
#
# ServerRoot: The top of the directory tree under which the server's
# configuration, error, and log files are kept.
#
# NOTE!  If you intend to place this on an NFS (or otherwise network)
# mounted filesystem then please read the LockFile documentation
# (available at <URL:http://httpd.apache.org/docs/2.2/mod/mpm_common.html#lockfile>);
# you will save yourself a lot of trouble.
#
# Do NOT add a slash at the end of the directory path.
#
ServerRoot "/etc/httpd"

#
# PidFile: The file in which the server should record its process
# identification number when it starts.  Note the PIDFILE variable in
# /etc/sysconfig/httpd must be set appropriately if this location is
# changed.
#
PidFile run/httpd.pid

#
# Timeout: The number of seconds before receives and sends time out.
#
Timeout 60
 (略)
```

ディレクティブでデフォルトの設定と同じ内容の多くはコメントになっています。いくつかの基本的なディレクティブを次の表にまとめました。このデフォルトで設定されているディレクティブは最小限必要な項目なので、これらのディレクティブの設定値があれば (ディレクトリなどがあれば)、Apache は動作します。

表 5.4　基本的なディレクティブ

| ディレクティブ名 | 内容 | 設定例 |
|---|---|---|
| Listen | サービスを受けるポート番号 | 80 |
| DocumentRoot | 公開するディレクトリ | /var/www/html |
| ServerName | サーバーの名前 | www.alpha.jp |
| DirectoryIndex | /をアクセスした時にアクセスするファイル | index.html |
| AddDefaultCharset | レスポンスに使われる文字コード | off…なし、UTF-8 |

### 5.4.5　テストファイルを作成

Web サーバーの機能を一文に要約すると、クライアントが要求するドキュメントや画像などのデータを転送する働きをするのが Web サーバーです。Web サーバーが動作するかチェックするためには/var/www/html ディレクトリに index.html ファイルを作成します。

```
#  echo 'This is TEST Page on `hostname` > /var/www/html/index.html
#  cat /var/www/html/index.html
```

```
This is TEST Page host1.alpha.jp
```

‘hostname‘の前後はバッククォートで括ります。hostname コマンドを実行し、その結果に置き換えています。

### 5.4.6 Apache を起動

設定とテスト用ファイルができたので Apache を起動してみましょう。Apache の起動は/etc/init.d/httpd スクリプトに start オプションをつけて Apache を起動します。

```
# /etc/init.d/httpd start
httpd を起動中:                                            [  OK  ]
```

Apache が起動しているかを、/etc/init.d/httpd スクリプトに status オプションを付けて確認します。Apache が起動している場合、下記のようにステータスが表示されます。

```
# /etc/init.d/httpd status
httpd (pid  7269) を実行中...
```

### 5.4.7 Web ブラウザーで自分のアドレスを確認

Web システムはコマンドで確認する代わりに、Web ブラウザーを使いグラフィカルに結果を確認できます。Web ブラウザーである Firefox から自分のアドレスを確認します。

```
http://www.alpha.jp/
```


自分のサーバーのアクセスができたら、隣の受講生マシンの Apache へアクセスします。

```
http://www.beta.jp/
```


## 5.5 ページが見つからないとき

テストファイルを作る前に Apache を起動し、Web ブラウザーである Firefox から用意されていないページをアクセスすると、どんな結果が出るでしょうか？ 実際に無いページにアクセスしてみてください。ファイルが無い旨のエラー (Not Found) が出ます。エラーページが表示されるのは Apache で対応するページが用意されているからです。エラーページは/var/www/error ディレクトリにあり、トップページが無い場合は特別に用意されたテストページが表示されています。


```
# ls /var/www/error
HTTP_BAD_GATEWAY.html.var                HTTP_REQUEST_TIME_OUT.html.var
HTTP_BAD_REQUEST.html.var                HTTP_REQUEST_URI_TOO_LARGE.html.var
HTTP_FORBIDDEN.html.var                  HTTP_SERVICE_UNAVAILABLE.html.var
HTTP_GONE.html.var                       HTTP_UNAUTHORIZED.html.var
HTTP_INTERNAL_SERVER_ERROR.html.var      HTTP_UNSUPPORTED_MEDIA_TYPE.html.var
HTTP_LENGTH_REQUIRED.html.var            HTTP_VARIANT_ALSO_VARIES.html.var
HTTP_METHOD_NOT_ALLOWED.html.var         README
HTTP_NOT_FOUND.html.var                  contact.html.var
HTTP_NOT_IMPLEMENTED.html.var            include
HTTP_PRECONDITION_FAILED.html.var        noindex.html
HTTP_REQUEST_ENTITY_TOO_LARGE.html.var
```

「アクセスしたページが見つからない」などの理由によって、それに対応するエラーページが表示されます。

### 5.5.1 Apache のエラーコードについて

Apache はアクセス情報とエラー情報をログに記録しています。また、エラーメッセージに対応するエラーページを表示します。ここでは Aapche がエラーを表示したときのエラーコードに対する意味を説明します。

表 5.5 Apache のエラーコード

| 番号 | 理由 | 意味 |
|---|---|---|
| 400 | BAD_REQUEST | 要求されたコードを理解できません |
| 401 | UNAUTHORIZED | アクセスする権利があることを確証できませんでした |
| 403 | FORBIDDEN | 要求するディレクトリーにアクセスする為の許可がありません |
| 404 | NOT_FOUND | 要求されたファイルが見つかりません |
| 405 | METHOD_NOT_ALLOWED | 許可されていないメソッドを受け取りました |
| 408 | REQUEST_TIME_OUT | 指定された時間以内にリクエストを終えなかったのでネットワーク接続を閉じました |
| 410 | GONE | 要求された URL はサーバー利用できず、転送先アドレスも理解できません |
| 411 | REQUIRED | Content-Length メソッドが不正です |
| 412 | PRECONDITION_FAILED | URL に対してリクエストの必要条件は、明確な評価に失敗しました |
| 413 | REQUEST_ENTITY_TOO_LARGE | 要求されたデータ量が容量限度を超えました |
| 414 | REQUEST_URI_TOO_LARGE | 要求された URL の長さは、このサーバーの容量限度を超えた為、処理することができません |
| 415 | UNSUPPORTED_MEDIA_TYPE | サーバーは、メディアタイプがリクエストで送ったことをサポートしません |
| 500 | INTERNAL_SERVER_ERROR | サーバーは内部エラーの為、要求を完了することができませんでした |
| 501 | NOT_IMPLEMENTED | サーバーはリクエストを実行する為の必要な機能をサポートしていません |
| 502 | BAD_GATEWAY | ゲートウェイやプロキシとして動作しているサーバーが、無効な応答を上位のサーバーから受け取りました |
| 503 | SERVICE_UNAVAILABLE | サーバーが一時的な過負荷によるものか保守時間の為、要求を受け付けられませんでした |
| 506 | VARIANT_ALSO_VARIES | 要求するエンティティの値は、そのもの自体で交渉できるリソースです |

### 5.5.2 ログファイルの確認

Apache が起動すると、アクセスされたファイルの履歴が/var/log/httpd/access_log ログファイルに記録され、エラーメッセージが/var/log/httpd/error_log ログファイルに記録されます。ログ

ファイルを確認してみてください。

```
# tail /var/log/httpd/access_log
（略）
192.168.1.202 - - [16/Apr/2012:14:06:40 +0900] "GET / HTTP/1.1" 403 5039 "-" "Mozilla/5
192.168.1.202 - - [16/Apr/2012:14:06:42 +0900] "GET / HTTP/1.1" 403 5039 "-" "Mozilla/5
192.168.1.202 - - [16/Apr/2012:14:06:42 +0900] "GET /icons/apache_pb.gif HTTP/1.1" 304
192.168.1.202 - - [16/Apr/2012:14:06:42 +0900] "GET /icons/poweredby.png HTTP/1.1" 304

# tail /var/log/httpd/error_log
（略）
[Mon Apr 16 14:06:29 2012] [error] [client 192.168.1.202] File does not exist: /var/www
[Mon Apr 16 14:06:40 2012] [error] [client 192.168.1.202] Directory index forbidden by
[Mon Apr 16 14:06:42 2012] [error] [client 192.168.1.202] Directory index forbidden by
```

ログファイルの名前は httpd.conf 設定ファイルで設定されています。エラーログの出力ファイル名は ErrorLog ディレクティブで、アクセスログの出力ファイル名は CustomLog ディレクティブです。ログファイルに記録されるクライアント情報はデフォルトが IP アドレスとなり、HostnameLookups ディレクティブを on にすると IP アドレスから調べたクライアントの名前を保存することも可能です。DNS サーバーから名前を引くと時間がかかるので、デフォルトでは off となっています。

表 5.6　ログ関係のディレクティブ

| ディレクティブ名 | 内容 | 設定例 |
|---|---|---|
| ErrorLog | エラーのログファイル | /var/log/httpd/error_log |
| CustomLog | アクセスのログファイルと形式 | /var/log/httpd/access_log |
| HostnameLookups | DNS への問い合わせ | on…問い合わせ、off…問い合わせない |

## 5.6　アクセス制御

通常、多くの人から自由に見てもらいたい情報を Web サービスで公開します。その一方で、特定の権限がある人だけに見せる会員制のホームページを作りたいときがあると思います。特定の個人だけアクセスできるように限定する手段として Apache には BASIC 認証と DIGEST 認証という機能が備えられています。この節では DocumentRoot ディレクティブで指定されている/var/www/html ディレクトリにある secret ディレクトリを特定の個人だけアクセスできるよう設定するために、セキュリティ上で安心な DIGEST 認証を設定してみましょう。

### 5.6.1　テストファイルを作成

認証をかけたディレクトリとファイルを作成します。

```
# mkdir /var/www/html/secret
# echo 'This is Secret Page on `hostname` > /var/www/html/secret/index.html
```

アクセス制御を設定する前であれば、制限をかけたいディレクトリやファイルであっても見えてしまいます。Web ブラウザーで以下のアドレスにアクセスできることを確認します。

```
http://www.alpha.jp/secret/
```


## 5.6.2 アクセス制御を設定

Apache でアクセス制御を設定するには設定ファイルにアクセス制御の設定を追加し、コマンドでパスワードファイルを作ります。/etc/httpd/conf/httpd.conf 設定ファイルに /var/www/html/secret ディレクトリの設定を追記します。

/etc/httpd/conf/httpd.conf 設定ファイルの追記 (345 行目</Directory>の次の行)

```
#
# Controls who can get stuff from this server.
#
    Order allow,deny
    Allow from all

</Directory>

 <Directory "/var/www/html/secret">
 AuthType Digest
 AuthName "Secret Zone"

 AuthUserFile /etc/httpd/.htdigest
 Require user lpic
 </Directory>

#
# UserDir: The name of the directory that is appended onto a user's home
# directory if a ~user request is received.
```

Directory ディレクティブ (<Directory>〜</Directory>) は/var/www/html/secret ディレクトリに認証をかける複数のディレクティブを囲みます。AuthType は認証の種類を Digest とし、AuthName は認証のダイアログに表示される認証名を"Secret Zone"とし、AuthUserFile でパスワードファイルを指定し、Require user でユーザー名を lpic としました。

表 5.7　アクセス制御関係のディレクティブ

| ディレクティブ名 | 内容 | 設定例 |
|---|---|---|
| Directory | ディレクトリ | /var/www/html |
| AuthType | 認証タイプ | Digest |
| AuthName | 認証ドメイン | Secret Zone |
| AuthUserFile | パスワードファイル | /etc/httpd/.htdigest |
| Require user | ユーザー指定 | lpic |

Require user で認証できるユーザーを指定でき、Require group で認証できるグループを指定できます。認証にはユーザーとパスワードの情報が必要です。パスワードファイルを作成します。パスワードファイルは重要なファイルなので、他のユーザーから見えないように設定します。ファイル作成後、Apache を動かしている apache ユーザーからのみ読みこめる様に、ファイルの所有者とモードを変更します。

```
# htdigest -c /etc/httpd/.htdigest 'Secret Zone' lpic
Adding password for lpic in realm Secret Zone.
New password: lpic ← 実際には表示されない
Re-type new password: lpic ← 実際には表示されない
# ls -l /etc/httpd/.htdigest
-rw-r--r-- 1 root root 50  4 月 16 14:22 2012 /etc/httpd/.htdigest
# chown apache.apache /etc/httpd/.htdigest
# chmod 400 /etc/httpd/.htdigest
# ls -l /etc/httpd/.htdigest
-r-------- 1 apache apache 50  4 月 16 14:22 2012 /etc/httpd/.htdigest
```

### 5.6.3　Apache を再起動

設定を変更したので、Apache を/etc/init.d/httpd スクリプトで再起動します。

```
# /etc/init.d/httpd restart
httpd を停止中:                                            [  OK  ]
httpd を起動中:                                            [  OK  ]
```

### 5.6.4　Web ブラウザーで自分のアドレスを確認

Web ブラウザーで自分のアドレス（http://www.alpha.jp/secret）を確認します。認証のダイアログが表示されるので、初めに登録していない適当なユーザー名と適当なパスワードを入力すると何度も認証のダイアログが表示されること (認証エラー) を確認します。次に登録した lpic ユーザーとパスワードを入力すると作成したファイルが表示されます。

正しいユーザー名、パスワードを入力した場合は、secret フォルダの中の index.html ファイルにアクセスできるようになります。認証をかけたディレクトリにあるディレクトリの中もリカーシブ

(再帰的) にユーザー名とパスワードの組み合わせを知っているユーザーのみがアクセスできます。


### 5.6.5 ログファイルの確認

正しく起動したかログファイルを確認します。

```
# tail /var/log/httpd/access_log
（略）
192.168.1.202 - lpic [16/Apr/2012:14:27:45 +0900] "GET /secret/ HTTP/1.1" 304 - "-" "Mo
192.168.1.202 - lpic [16/Apr/2012:14:27:49 +0900] "GET /secret/ HTTP/1.1" 304 - "-" "Mo
192.168.1.202 - lpic [16/Apr/2012:14:28:01 +0900] "GET /secret/ HTTP/1.1" 200 38 "-" "M

# tail /var/log/httpd/error_log
（略）
[Mon Apr 16 14:27:26 2012] [notice] caught SIGTERM, shutting down
[Mon Apr 16 14:27:26 2012] [notice] suEXEC mechanism enabled (wrapper: /usr/sbin/suexec
[Mon Apr 16 14:27:26 2012] [notice] Digest: generating secret for digest authentication
[Mon Apr 16 14:27:26 2012] [notice] Digest: done
[Mon Apr 16 14:27:26 2012] [warn] ./mod_dnssd.c: No services found to register
[Mon Apr 16 14:27:26 2012] [notice] Apache/2.2.15 (Unix) DAV/2 configured -- resuming n
[Mon Apr 16 14:27:42 2012] [error] [client 192.168.1.202] Digest: user `lpickun' in rea
```

認証に失敗するとエラーログファイルにユーザーがみつからないというエラーメッセージが残ります。

## 5.7 PHP 言語を使えるようにする

Web サーバーは静的な HTML ファイルや画像ファイルを Web ブラウザーに転送する他に、アプリケーションを実行することもできます。Web アプリケーションを作るための言語として PHP 言語が多く使われています。この節では php5 モジュールを組み込んで、PHP 言語を使える状態にしてみましょう。

### 5.7.1 必要なパッケージを確認

PHP は Apache の拡張モジュールとして用意されています。rpm コマンドで必要なパッケージがインストールされているかを確認します。

```
# rpm -q php php-cli php-common
php-5.3.3-3.el6_1.3.i686
php-cli-5.3.3-3.el6_1.3.i686
php-common-5.3.3-3.el6_1.3.i686
```

パッケージがインストールされていない場合は次の様に表示されます。

```
# rpm -q php php-cli php-common
パッケージ php はインストールされていません。
パッケージ php-cli はインストールされていません。
パッケージ php-common はインストールされていません。
```

### 5.7.2 php5 モジュールをインストール

PHP5 と最小限必要なパッケージを rpm コマンドでインストールします。

PHP 言語を使うために必要なパッケージがインストールされていないときは、rpm コマンドでインストールをします。

ここでは、自動マウント機能が動作していて DVD-ROM が自動的に/media/CentOS_6.2_Final にマウントされた状態でインストール作業を進めています。DVD-ROM の公開鍵を登録してからインストールすると警告メッセージが出ないので、インストールをする前に rpm コマンドに–import オプションと公開鍵のファイルを指定して登録しています。最後に、DVD-ROM のディレクトリ上で作業をしたので、cd コマンドでホームディレクトリへ移動しています。

```
# cd /media/CentOS_6.2_Final/Packages/
# rpm -ivh php-5.3.3-3.el6_1.3.i686.rpm php-cli-5.3.3-3.el6_1.3.i686.rpm php-common-5.3.3-3.el6_1.3.i686.rpm
準備中...                ########################################### [100%]
1:php-common             ########################################### [ 33%]
2:php-cli                ########################################### [ 67%]
```

```
3:php                    ########################################## [100%]
# cd
```

### 5.7.3 設定ファイルを確認

PHP5 のパッケージでインストールすると、PHP 言語を使うために必要な設定が /etc/httpd/conf.d/php.conf ファイルとして用意されます。php.conf ファイルを確認します。

```
# cat /etc/httpd/conf.d/php.conf
```

```
#
# PHP is an HTML-embedded scripting language which attempts to make it
# easy for developers to write dynamically generated webpages.
#
<IfModule prefork.c>
   LoadModule php5_module modules/libphp5.so
</IfModule>
<IfModule worker.c>
  LoadModule php5_module modules/libphp5-zts.so
</IfModule>

#
# Cause the PHP interpreter to handle files with a .php extension.
#
AddHandler php5-script .php
 AddType text/html .php

#
# Add index.php to the list of files that will be served as directory
# indexes.
#
 DirectoryIndex index.php

#
# Uncomment the following line to allow PHP to pretty-print .phps
# files as PHP source code:
#
#AddType application/x-httpd-php-source .phps
```

php.conf ファイルには PHP5 モジュールの関連項目だけが書かれているので、確認しておくべきディレクティブはわずか 3 つだけです。LoadModule ディレクティブは PHP5 のプログラムである php5_module として modules/libphp5.so を読み込み、AddType ディレクティブは.php 拡張子のファイルを text/html 形式のファイルと解釈する指定で、DirectoryIndex ディレクティブは/をアクセスした時に index.php ファイルがアクセスされる様に指定しています。

表 5.8　PHP 関係のディレクティブ

| ディレクティブ名 | 内容 | 設定例 |
|---|---|---|
| LoadModule | モジュール指定 | php5_module modules/libphp5.so |
| AddType | ファイルタイプ登録 | text/html .php |
| DirectoryIndex | ファイル名省略時に探すファイル名 | index.php |

### 5.7.4　Apache を再起動

拡張モジュールを追加するために設定ファイルが変更されたので、Apache を再起動します。

```
# /etc/init.d/httpd restart
httpd を停止中:                                            [  OK  ]
httpd を起動中:                                            [  OK  ]
```

### 5.7.5　サンプルプログラムを作成

php5 モジュールを追加できたか確認するために、PHP の設定情報や機能を表示してくれる phpinfo 関数を使ったサンプルを作成して確認してみましょう。

```
# echo "<?php phpinfo(); ?>" > /var/www/html/info.php
```

### 5.7.6　Web ブラウザーで自分のアドレスを確認

Web ブラウザーで自分の Web サーバーにアクセスします。PHP のバージョンから設定情報、利用できる機能が表示されれば動作は正常です。

```
http://localhost/info.php
```


なお、phpinfo は PHP だけでなく、様々なサーバー情報などが確認できるものです。したがって、外部公開するサーバーの場合は PHP の動作を確認次第、速やかに info.php ファイルを削除するようにしましょう。

## 5.8 バーチャルホストを作成する

バーチャルホスト機能を利用すると、1 台の Web サーバーで別々のホームページを見せることができます。バーチャルホスト機能を利用するには、以下の設定が必要です。

1. DNS で 1 つの IP アドレスに対して別々のホスト名を割り当てる
2. Apache でバーチャルホストの設定を行う
3. Linux にそれぞれのバーチャルホスト用のディレクトリを作成する

ここでは、vhost1.alpha.jp と vhost2.alpha.jp という 2 つのバーチャルホストを作成します。それぞれの名前で Web ブラウザーからアクセスした時に、別々のホームページが見えるように設定を行います。

### 5.8.1 IP アドレスと名前の確認

2 つの名前を 1 つの IP アドレスに割り振っている設定を、nslookup コマンドで確認します。

```
# nslookup vhost1.alpha.jp
Server:         192.168.1.10
Address:        192.168.1.10#53

Non-authoritative answer:
Name:   vhost1.alpha.jp
Address: 192.168.1.101

# nslookup vhost2.alpha.jp
Server:         192.168.1.10
Address:        192.168.1.10#53

Non-authoritative answer:
Name:   vhost1.alpha.jp
Address: 192.168.1.101
```

### 5.8.2 バーチャルホストの設定

複数の URL アドレスに対応するバーチャルホストは NameVirtualHost ディレクティブと VirtualHost ディレクティブを使います。vhost1.alpha.jp と vhost2.alpha.jp という 2 つの URL アドレスを定義します。

表 5.9 バーチャルホスト関係のディレクティブ

| ディレクティブ | 内容 | 具体例 |
|---|---|---|
| NameVirtulaHost | バーチャルホストに使う IP アドレス | *:80 |
| VirtualHost | バーチャルホストに適用するディレクティブをまとめる | 指定のみ |

/etc/httpd/conf/httpd.conf に追加

```
NameVirtualHost *:80 ← 行頭の#を削除

※以下を追加
<VirtualHost *:80>
   ServerName www.alpha.jp
   DocumentRoot /var/www/html
   ServerAdmin webmaster@www.alpha.jp
   ErrorLog /var/log/httpd/error_log
   CustomLog /var/log/httpd/access_log common
</VirtualHost>

<VirtualHost *:80>
   ServerName vhost1.alpha.jp
   DocumentRoot /var/www/vhost1.alpha.jp
   ServerAdmin webmaster@vhost1.alpha.jp
   ErrorLog /var/log/httpd/vhost1.alpha.jp-error_log
   CustomLog /var/log/httpd/vhost1.alpha.jp-access_log common
</VirtualHost>

<VirtualHost *:80>
```

```
    ServerName vhost2.alpha.jp
    DocumentRoot /var/www/vhost2.alpha.jp
    ServerAdmin webmaster@vhost2.alpha.jp
    ErrorLog /var/log/httpd/vhost2.alpha.jp-error_log
    CustomLog /var/log/httpd/vhost2.alpha.jp-access_log common
</VirtualHost>
```

NameVirtualHost ディレクティブでバーチャルホスト機能を使う IP アドレスとポート番号を宣言しています。また、<VirtualHost>内でドメイン名、ドメインにアクセスしたときに表示するコンテンツの参照先、サーバーの管理者のメールアドレス、ログの出力先を定義します。上記のように設定した場合は、www.alpha.jp にアクセスした場合は /var/www/html 配下のコンテンツが表示され、ログが /var/log/httpd/access_log に保存されます。エラーがあった場合はエラーログが /var/log/httpd/error_log に保存されます。

### 5.8.3 テストファイルを作成

同じマシンで 2 つの URL に対応するので、全く違った内容を表示するようにテストファイルを作成します。2 カ所の DocumentRoot ディレクティブで指定した 2 つのディレクトリを作成し、各ディレクトリにサンプルファイルを用意します。

```
#  mkdir /var/www/vhost1.alpha.jp
#  mkdir /var/www/vhost2.alpha.jp
#  echo 'This is Virtual Page 1 on "`hostname` > /var/www/vhost1.alpha.jp/index.html
#  echo 'This is Virtual Page 2 on "`hostname` > /var/www/vhost2.alpha.jp/index.html
```

### 5.8.4 Apache の再起動

バーチャルホストの定義を書き加えて設定ファイルを変更したので、Apache を再起動します。

```
#  /etc/init.d/httpd restart
httpd を停止中:                                            [  OK  ]
httpd を起動中:                                            [  OK  ]
```

### 5.8.5 Web ブラウザーで自分のアドレスを確認

それでは、vhost1.alpha.jp や vhost2.alpha.jp にアクセスして別々のコンテンツが表示されるか早速確認してみましょう。

```
http://vhost1.alpha.jp/
http://vhost2.alpha.jp/
```



2 つの URL が表示されないときは、DNS の設定が正しいかどうかを nslookup コマンド等で再確認します。

### 5.8.6 ログファイルの確認

正しく起動したかをログファイルが作られているかで確認してみてください。

```
# ls -l /var/log/httpd
合計 20
-rw-r--r-- 1 root root 149  4 月 16 20:23 2012 access_log
-rw-r--r-- 1 root root 563  4 月 16 20:23 2012 error_log
-rw-r--r-- 1 root root 146  4 月 16 20:23 2012 vhost1.alpha.jp-access_log
-rw-r--r-- 1 root root 109  4 月 16 20:23 2012 vhost1.alpha.jp-error_log
-rw-r--r-- 1 root root  70  4 月 16 20:24 2012 vhost2.alpha.jp-access_log
-rw-r--r-- 1 root root   0  4 月 16 20:23 2012 vhost2.alpha.jp-error_log
```

# 第6章

# メールサーバーの構築

メールのやり取りが行えるよう、メールサーバーを設定します。まずは Sendmail を使って、メールサーバー同士でメールのやり取りが行えるように設定します。さらに POP/IMAP サーバーとメールクライアントを使って、より実践的なメール環境を構築します。

## 6.1 用語集

**メールサーバー**

電子メールのサービスを行います。クライアントよりメールを受け取り、バケツリレーの方式で相手先のメールサーバーまで送ります。また、受信用のメールサーバーでは、送ってきたメールを蓄積しておいて、クライアントの要求に応じて応答します。

**MTA**

Mail Transfer Agent。メールの転送を行うプログラムです。Sendmail や Postfix などが代表例です。

**SMTP(Simple Mail Transfer Protocol)**

電子メールの送信、転送のときに利用されるプロトコルのことです。

**SMTP 認証**

SMTP でのメールを送信する際に認証を行う機構です。迷惑メール対策としてのメール中継の制限を、この認証機能で許可する、といった利用方法があります。

**POP3(Post Office Protocol version3)**

クライアントが電子メールを取り寄せるときに利用されるプロトコルです。シンプルな設計で、IMAP4 と比べて機能が少ないです。

**IMAP4**

クライアントが電子メールを取り寄せるときに利用されるプロトコルです。メールのフォルダ機能サポート等、多機能です。

Sendmail
古くからある、UNIX 系 OS で動作する MTA プログラムです。

Dovecot
POP3 や IMAP4 のサーバー機能を提供するプログラムです。

Thunderbird
Mozilla Project が配布している、高機能なメールクライアントソフトウェアです。

## 6.2 メールサーバー実習の説明

メールサーバーの設定と動作確認を行います。メールは、インターネットにおいて、Web に並んで重要なサービスです。メールサーバーを設定し、実際にメールをやり取りすることで、動作原理を確認してみましょう。

### 6.2.1 メールとメールサーバー

メールは、メールサーバーを介してやり取りが行われます。メールサーバーがメールを受け取ると、宛先のメールアドレスを担当しているメールサーバーまでバケツリレー方式で送られます。これは、Thunderbird/Outlook の様なメールクライアントソフトからのメールでも、Gmail/Hotmail/Yahoo メールのような Web メールからのメールでも、動作原理は同じです。

MTA(Mail Transfer Agent)
メールがバケツリレー方式で運ばれることは、前述の通りです。このバケツリレーをするプログラムのことを MTA といいます。本教科書では Sendmail という MTA を利用します。他に有名な MTA として Postfix などがあります。

SMTP(Simple Mail Transfer Protocol)
メールサーバー間は、SMTP というプロトコルでやり取りされています。SMTP はかなり昔に設計、定義されたプロトコルのため、認証やアクセス制限などが無く、勝手にメールサーバーを利用されて迷惑メールを送られてしまうなどの問題がありました。そこでこのような問題を解決するために、ESMTP（拡張 SMTP）が定義されました。SMTP と呼ぶ場合、この ESMTP で定義された機能も含んでいることがあります。

SMTP 認証 (SMTP Authentication)
正規の SMTP 接続では、メールの中継を行います。ところが前述の通り、SMTP には認証機能が無いため、多くの場合、特定の場所以外からの中継を拒否します。SMTP 認証は、SMTP 上に認証 (Authentication) 機能を加え、その中継要求が正規のものか不正なものかを判断します。SMTP 認証は ESMTP の機能のうちの 1 つです。

#### POP3(Post Office Protocol version 3)

電子メールは、送受信でプロトコルが異なります。POP3 は電子メールを受信するときに利用するプロトコルです。非常にシンプルなプロトコルで、ユーザー名、パスワードを利用して接続し、メールの内容を受信します。

#### IMAP4

IMAP4 も POP3 同様、メールを受信するときに利用するプロトコルです。IMAP4 は POP3 に比べて機能が豊富で、大きな特徴としてフォルダ機能をサポートしていることが挙げられます。そのため、メール管理が非常に楽になります。

### 6.2.2 メールのやり取り

インターネット上で、沢山の人が電子メールを利用しています。電子メールは以下の手順でやり取りされます。

1. 送信側のメールクライアントからメールを送信します
2. メールは送信用メールサーバーを経由して相手のメールサーバーに配信されます
3. 相手の受信側のメールサーバーにメールが届きます
4. 受信側のメールクライアントで受信側のメールサーバーに接続します
5. メールが受信され、メールを見ることができます

前述の構成で実習を行う場合、サーバー、クライアントをそれぞれ 2 台ずつ、計 4 台必要になります。本実習では二人一組のペアを組んで 2 台で実習を行うため、以下のような構成を取ります。

- 1 台のマシンで、メールサーバーとメールクライアントの 1 台 2 役とします
- 自分のメールサーバーに、自分のメールアカウントを作成します
- 相手のメールサーバーに、相手のメールアカウントが作成されます
- 自分のメールクライアントは、自分のメールサーバーを送受信用サーバーとして設定します

ポイントは、自分のマシンは 1 台ですが、メールサーバーとメールクライアントの 1 台 2 役であることです。

### 6.2.3 実習の進め方

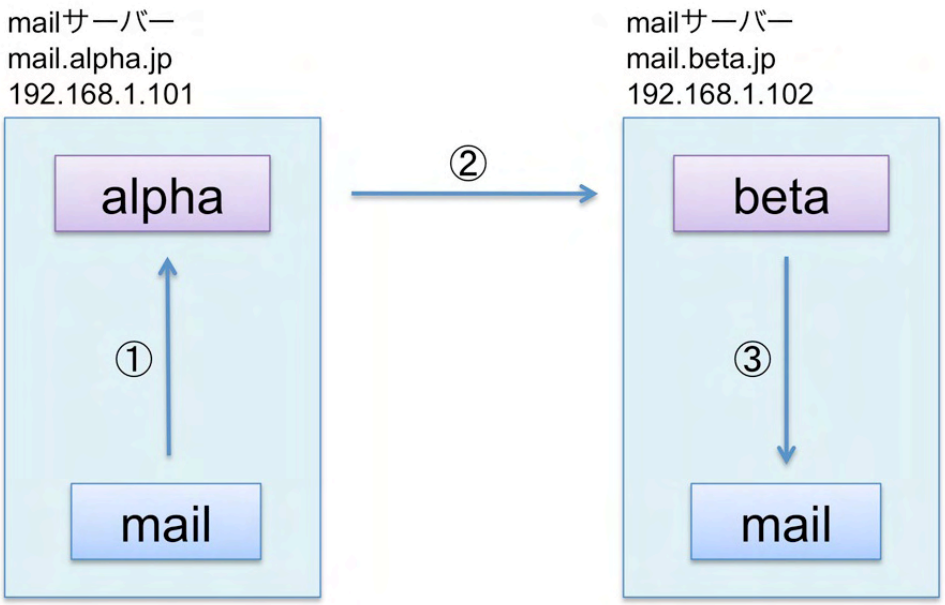
本章では、次の手順で実習を行います。実習でのメールの送受信は、大きく分けて 2 回あります。

#### mail コマンドを利用する

端末で mail コマンドを使って相手にメール送信します。以下の手順に従ってください。

1. 二人一組になります。それぞれ host1.alpha.jp を利用する A さん (usera@alpha.jp) と、host2.beta.jp を利用する B さん (userb@beta.jp) とします。
2. Sendmail の設定ファイルを作成し、Sendmail を起動します。

3. 送受信テスト用の自分のアカウント（usera@alpha.jp、userb@beta.jp）を、それぞれのホストに作成します。
4. A さんが host1.alpha から mail コマンドを使って、B さんにメールを送ります。
5. B さんは mail コマンドを使って、到着を確認します。
6. 逆に B さんから A さんにメールを送り、確認します。

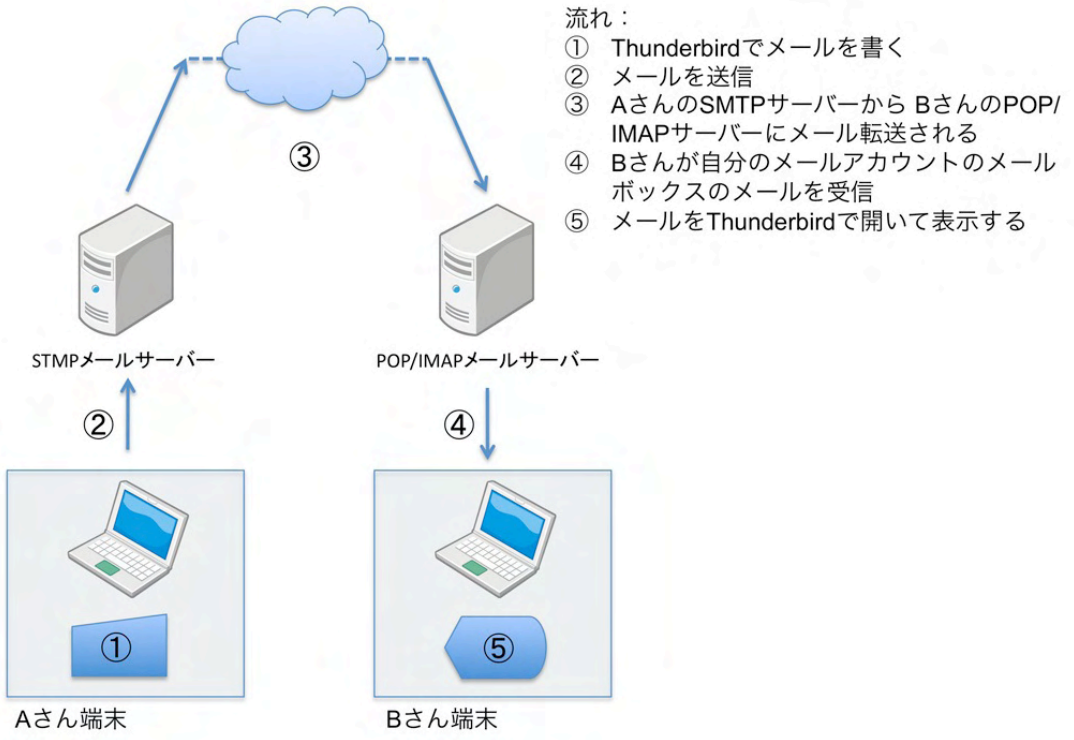



流れ：
① mailコマンドでメールを作成
② メールを転送
③ mailコマンドでメールを受信

メールクライアントソフトを利用する

メールクライアントを使って相手にメール送信します。本実習では Thunderbird を使います。以下の手順に従ってください。

1. POP/IMAP サーバーとして Dovecot の設定を行い、POP/IMAP サーバーを起動します。
2. メールクライアントとして Thunderbird を設定します。
3. A さんが host1.alpha からメールクライアントを使って、B さんにメールを送ります。
4. B さんはメールクライアントを使って、到着を確認します。
5. 逆に B さんから A さんにメールを送り、確認します。




### 6.2.4 実習後の注意点

設定したメールサーバーは、あくまで実習用にメールのやり取りをするために設定されています。ですが、セキュリティ等決して頑丈に設定してある訳ではないので、以下の点に留意します。

- 決してグローバル環境には接続しない。
- 実習後はメールサーバーを止める、もしくは本体自体を止める。LAN の中に不正中継を探すマシンがあった場合、それに利用される可能性があるからです。

### 6.2.5 実習で使用するソフトウェアについて

#### Sendmail

今回の実習では、MTA として Sendmail を利用します。

#### sendmail.cf と sendmail.mc ファイル

Sendmail は/etc/mail/sendmail.cf を設定ファイルとして動作します。その他、sendmail.cf から参照するためのファイルがいくつか有り、それらはすべて/etc/mail ディレクトリの中に格納されています。sendmail.cf は記述方法が難しいため、記述方法が易しい sendmail.mc を修正し、変換する事で sendmail.cf を生成します。

#### mail コマンド

mail コマンドは、標準でインストールされている、メールを操作するコマンドです。mail コマンドには、メールを送る以外に、届いたメールを読む機能もあります。本実習では、メールで送るときと届いたメールを読むときに利用します。

#### Dovecot

Dovecot は、POP3 や IMAP4 を機能させるサーバーのソフトウェアです。実習では、メールのクライアントソフトから IMAP4 でメールを受信します。そのときに IMAP4 のサーバーとして動作させます。標準ではインストールされていないので、実習時にパッケージを追加し、設定ファイルを書き換えます。

#### Thunderbird

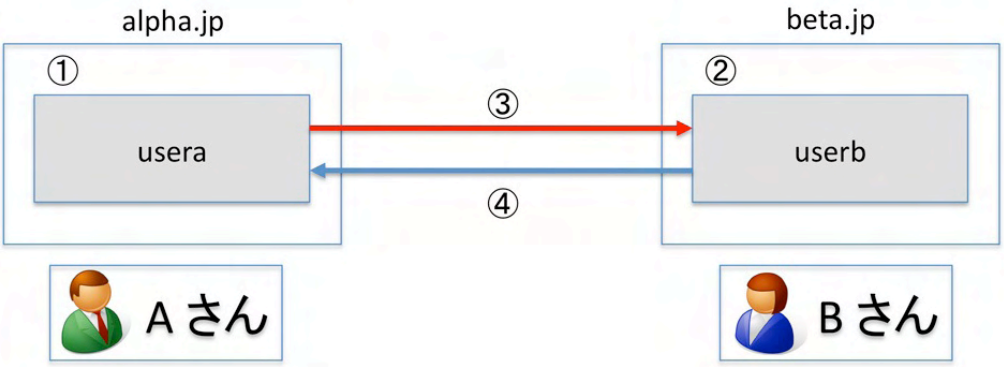
Mozilla Project により開発されている、フリーのメールクライアントソフトがこの Thunderbird です。Windows, Mac OS X, Linux 等と、動作環境は多岐に渡っており、各国語版も用意されております。機能も必要十分な内容がそろっています。本実習では、メールのクライアントソフトとして Thunderbird を使って実習を行います。実際に Thunderbird をインストールしメールの送受信を行うことで、メール設定が正しいか確認を行います。

#### saslauthd

saslauthd は、SMTP 認証の認証機構です。Sendmail は設定ファイルで SMTP 認証の機能を有効にできますが、Sendmail 自体は認証の機能を持っていません。設定ファイルで SMTP 認証の機能を有効にすると、Sendmail は saslauthd に認証を依頼してその結果を受け取ります。

### 6.2.6 実習環境

実習では、受講生二人一組で実習を行います。本章では、それを便宜的に「A さん」「B さん」と呼びます。特に明示がない場合は、両者とも作業を行います。どちらか片方の方が作業をするときは、「次は A さんの作業です」といったように、作業する方を明示します。設定は、alpha.jp のマシン上で行うものとします。ホスト名は、各自読み替えてください。




流れ:
① サーバーalpha.jpにuseraアカウントを作成
② サーバーbeta.jpにuserbアカウントを作成
③ Aさんがusera@alpha.jpからuserb@beta.jpにメール送信
④ Bさんがuserb@beta.jpからusera@alpha.jpにメール送信

## 6.3 Sendmail のインストール

### 6.3.1 必要なパッケージを確認

Sendmail パッケージ、sendmail.mc から設定ファイルである sendmail.cf を作成する sendmail.cf パッケージと procmail パッケージが必要です。rpm コマンドで必要なパッケージがインストールされているかを確認します。

```
# rpm -q sendmail sendmail-cf procmail
sendmail-8.14.4-8.el6.i686
sendmail-cf-8.14.4-8.el6.noarch
procmail-3.22-25.1.el6.i686
```

パッケージがインストールされていない場合は次の様に表示されます。

```
# rpm -q sendmail sendmail-cf procmail
パッケージ sendmail はインストールされていません。
パッケージ sendmail-cf はインストールされていません。
パッケージ procmail はインストールされていません。
```

### 6.3.2 必要なパッケージをインストール

Sendmail によるメールサーバーの構築に必要なパッケージがインストールされていないときは、rpm コマンドでインストールをします。

ここでは、自動マウント機能が動作していて DVD-ROM が自動的に/media/CentOS_6.2_Final にマウントされた状態でインストール作業を進めています。DVD-ROM の公開鍵を登録してからインストールすると警告メッセージが出ないので、インストールをする前に rpm コマンドに–import オプションと公開鍵のファイルを指定して登録しています。最後に、DVD-ROM のディレクトリ上で作業をしたので、cd コマンドでホームディレクトリへ移動しています。

```
# rpm --import /media/CentOS_6.2_Final/RPM-GPG-KEY-CentOS-6
# cd /media/CentOS_6.2_Final/Packages/
# rpm -ivh sendmail-8.14.4-8.el6.i686.rpm sendmail-cf-8.14.4-8.el6.noarch.rpm procmail-3.22-25.1.el6.i686.rpm
準備中...                ########################################### [100%]
   1:procmail            ########################################### [ 33%]
   2:sendmail            ########################################### [ 67%]
   3:sendmail-cf         ########################################### [100%]
# cd
```

### 6.3.3 後から Sendmail をインストールした場合の注意点

CentOS 6.2 では、デフォルトの MTA として Postfix がインストールされます。ディストリビューションのインストール時に Sendmail を追加でインストールするのを忘れた場合、Postfix が自動起動しています。もし、上記手順で Sendmail を後からインストールした場合には、システムを再起動するか、Postfix を停止した後、Sendmail を起動してください。

以下は、Postfix を手動で停止し、Sendmail を起動するスクリプトの実行方法です。

```
# /etc/init.d/postfix stop
# /etc/init.d/sendmail start
sendmail を起動中:                                         [  OK  ]
sm-client を起動中:                                        [  OK  ]
```

### 6.3.4 sendmail.mc の変更

それでは、実際に sendmail.mc を変更し、sendmail.cf を作成します。sendmail.mc の変更点は、次の 3 行です。

- TRUST_AUTH_MECH(‘EXTERNAL... と、define(‘confAUTH_M...、2 つの行の先頭にある"dnl "（半角スペースまで 4 文字）を削除します。これは、SMTP 認証の機能を有効にするための設定です。
- DAEMON_OPTIONS... の行の先頭に dnl をつけてください。設定内容をコメントアウトした＝無効にした、ことになります。この行では、ループバックアドレスからのアクセスのみを許可していましたが、無効にした事でその他のネットワークインターフェースのアドレスからのアクセスを許可します。

```
# cd /etc/mail
# vi sendmail.mc
```

```
（略）
TRUST_AUTH_MECH(`EXTERNAL DIGEST-MD5 CRAM-MD5 LOGIN PLAIN')dnl
先頭の” dnl”を削除します。
  define(`confAUTH_MECHANISMS', `EXTERNAL GSSAPI DIGEST-MD5 CRAM-MD5 LOGIN PLAIN')
先頭の” dnl”を削除します。
（略）
dnl # address restriction to accept email from the internet or intranet.
dnl #
dnl DAEMON_OPTIONS(`Port=smtp,Addr=127.0.0.1, Name=MTA')dnl
先頭に” dnl”を付けてください。
dnl #
dnl # The following causes sendmail to additionally listen to port 587 for
（略）
```

### 6.3.5 sendmail.cf の作成

次に sendmail.cf を作成し、Sendmail を再起動します。sendmail.cf を作成するには、/etc/mail ディレクトリで make コマンドを実行します。

```
# pwd ← 現在の作業ディレクトリを確認
/etc/mail
# make
# ls -l sendmail.cf ← タイムスタンプが make した時間に更新されていることを確認
-rw-r--r-- 1 root root 58677  4 月 16 20:51 2012 sendmail.cf
```

### 6.3.6 受信ドメインの設定

次に、local-host-names を編集します。local-host-names は、メールサーバーが受け付けるドメイン名（＝@以降の部分）を記述します。今回の場合は、ユーザー名@ドメイン名を受け付けるようにするため、alpha.jp を記述しました。host1.alpha.jp や mail.alpha.jp でも受け付けたい場合は、2 行目以降に追加します。

```
# vi local-host-names
```

```
# local-host-names - include all aliases for your machine here.
alpha.jp ← 自マシンで受信するメールのドメイン名を設定
```

### 6.3.7 Sendmail の再起動

sendmail サービスを再起動します。

```
# /etc/init.d/sendmail restart
sm-client を停止中:                                        [  OK  ]
sendmail を停止中:                                         [  OK  ]
sendmail を起動中:                                         [  OK  ]
sm-client を起動中:                                        [  OK  ]
```

sendmail サービスの自動起動を確認します。

```
# chkconfig --list sendmail
sendmail        0:off   1:off   2:on    3:on    4:on    5:on    6:off
```

### 6.3.8 saslauthd サービスの起動

SMTP 認証用の saslauthd サービスを起動します。

```
# /etc/init.d/saslauthd start
saslauthd を起動中:                                        [  OK  ]
# chkconfig saslauthd on
# chkconfig --list saslauthd
saslauthd       0:off   1:off   2:on    3:on    4:on    5:on    6:off
```

## 6.4 アカウントの作成

それでは、実際にメールを送信する前に、まず、宛先となるアカウントを作成します。

### 6.4.1 host1.alpha.jp に usera を作成

host1.alpha.jp で usera というアカウントを作成します。このアカウントは usera@alpha.jp というメールアドレスになります。

```
# useradd usera
# passwd usera
New UNIX password: userapass ← 入力文字は非表示
Retype new UNIX password: userapass ← 入力文字は非表示
```

### 6.4.2 host2.beta.jp に userb を作成

host2.beta.jp で userb というアカウントを作成します。このアカウントは userb@beta.jp というメールアドレスになります。

```
# useradd userb
# passwd userb
New UNIX password: userbpass ← 入力文字は非表示
Retype new UNIX password: userbpass ← 入力文字は非表示
```

## 6.5 メールの送受信

次にメールを送信します。メールの送受信は作成した一般ユーザーで行います。一般ユーザーで操作できるよう別の端末を起動し、su コマンドを使ってユーザーを切り替えます。メールの送信は mail コマンドを使用します。

### 6.5.1 ログの確認用端末の設定

メールの送受信の状況はログファイルに記録されていきます。動作を確認するために、ログを確認するための端末を起動し、tail コマンドを使ってログ出力をリアルタイムで表示できるようにしておきます。

1.「端末」を起動します
2. tail コマンドを実行して、/var/log/maillog を表示します。-f オプションを付けて実行すると、ログが書き込まれる度に再読み込みされて最新のログを閲覧できます。

```
# tail -f /var/log/maillog
```

### 6.5.2 メール送受信用端末の起動とユーザー切り替え

メール送受信用の端末を起動し、su コマンドでユーザーの切り替えを行います。

1.「端末」を起動します
2. su コマンドでユーザーを切り替えます

host1.alpha.jp で usera に切り替え

```
[root@host1 ~]# su - usera
[usera@host1 ~]$ id ← 一般ユーザーはプロンプトが$と表示
uid=500(usera) gid=500(usera) 所属グループ=500(usera)
```

host2.beta.jp で userb に切り替え

```
[root@host2 ~]# su - userb
[userb@host2 ~]$ id
uid=500(userb) gid=500(userb) 所属グループ=500(userb)
```

### 6.5.3 usera@alpha.jp から userb@beta.jp へメール送信

mail コマンドを使って、host1.alpha.jp の usera から userb@beta.jp へメールを送信します。

```
[usera@host1 ~] $ mail userb@beta.jp ← mail コマンドの引数に宛先のアドレスを指定
Subject: Test mail from usera ← Subject を入力
This is Test Mail from usera ← メッセージ本文を入力
. ← メッセージ本文入力が終わったらピリオドを入力
EOT
```

### 6.5.4 userb のメール着信確認

mail コマンドを使って、host2.beta.jp の userb にメールが届いているかを確認します。

```
[userb@host2 ~]$ mail
Heirloom Mail version 12.4 7/29/08.  Type ? for help.
"/var/spool/mail/userb": 1 message 1 new
>N  1 usera@host1.alpha.jp  Mon Apr 23 10:49  23/904   "Test mail from usera"
& 1
```

```
Message  1:
From usera@host1.alpha.jp  Mon Apr 23 10:49:58 2012
Return-Path: <usera@host1.alpha.jp>
From: usera@host1.alpha.jp
Date: Mon, 23 Apr 2012 10:43:46 +0900
To: userb@beta.jp
Subject: Test mail from usera
User-Agent: Heirloom mailx 12.4 7/29/08
Content-Type: text/plain; charset=us-ascii
Status: R

This is Test Mail from usera

& q
Held 1 message in /var/spool/mail/userb
メールが /var/spool/mail/userb にあります
```

このように、host1.alpha.jp から host2.beta.jp にメールが送られていることがわかります。以上で、A さんによる実習が終了です。次に、今度は B さんが A さんに対してメールを送ってみましょう。

## 6.6 メールのスパム対策

今回の実習では言及されていませんが、/etc/mail/access を適切に編集することによって、特定の IP やドメイン、アドレスからの接続を拒否することができます。なお、設定変更後は make を実行することで access.db が更新されます。

```
# cd /etc/mail
# make
```

make を実行する以外に、下記のコマンドを実行することでも変更を反映できます。

```
# makemap hash /etc/mail/access.db </etc/mail/access
```

### IP アドレスによる拒否

```
1.2.3.4 REJECT
1.2.3   REJECT
1.2     REJECT
```

一行目の設定では 1.2.3.4 からの接続を拒否、二行目の設定では 1.2.3.0/24 からの接続を拒否、三行目の設定では 1.2.0.0/16 からの接続を拒否できます。

### ドメインによる拒否

```
From:example.com        REJECT
```

xxx@example.com から送信されるメールをすべて拒否できます。

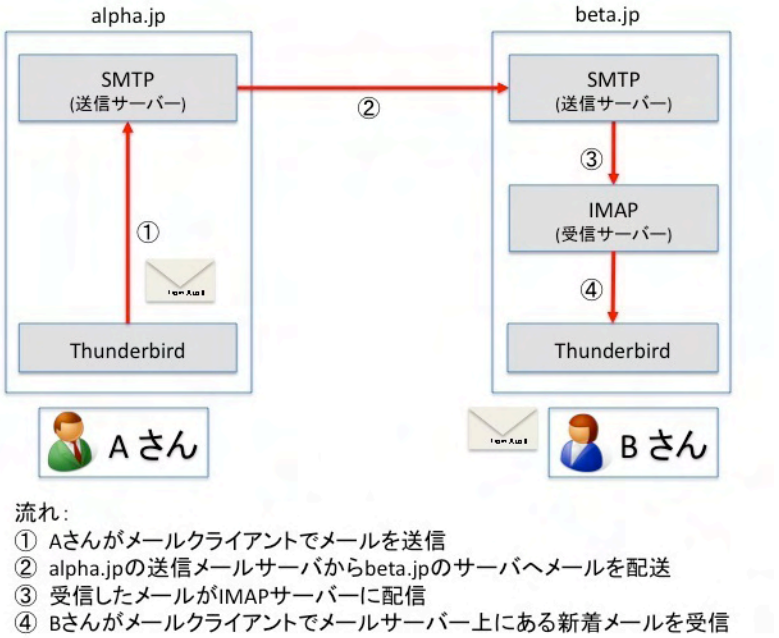
### メールアドレスによる拒否

```
From:foo@example.com    REJECT
foo@example.com REJECT
```

上記のいずれかの記法でアドレスを設定すると、設定したメールアドレスからの送信を拒否できます。

## 6.7 メールクライアントソフトでのメールの送受信

通常のメールサーバーの運用では、メールの利用者はメールクライアントを使用してメールの送受信を行います。送信は SMTP、受信は IMAP や POP3 をプロトコルとして使用します。IMAP サーバーを利用してメールを受信できるよう、IMAP サーバーである Devecot と、メールクライアントとして Thunderbird をインストールしてメールを送受信してみます。




### 6.7.1 Dovecot パッケージの追加

それでは早速、必要なパッケージを追加して、クライアントでメールを送受信できるように設定してみましょう。まずは IMAP サーバーである Dovecot をインストールします。

```
# cd /media/CentOS_6.2_Final/Packages/
# rpm -ivh dovecot-2.0.9-2.el6_1.1.i686.rpm
準備中...                ########################################### [100%]
   1:dovecot             ########################################### [100%]
# cd
```

### 6.7.2 Dovecot の設定

次に、IMAP サーバーである Dovecot の設定を行います。設定ファイルは /etc/dovecot/dovecot.conf と/etc/dovecot/conf.d ディレクトリ以下に分かれています。

/etc/dovecot/dovecot.conf

全体的な設定ファイルです。デフォルトの設定がコメントアウトで記述されています。特に変更は必要ありません。

```
# vi /etc/dovecot/dovecot.conf
```

```
（略）
# Protocols we want to be serving.
 #protocols = imap pop3 lmtp ← IMAP/POP3/LMTP が使用可能
（略）
# A comma separated list of IPs or hosts where to listen in for connections.
# "*" listens in all IPv4 interfaces, "::" listens in all IPv6 interfaces.
# If you want to specify non-default ports or anything more complex,
# edit conf.d/master.conf.
#listen = *, ::  ← ホストのすべての IP アドレスから接続を受け付ける
（略）
```

/etc/dovecot/conf.d/10-mail.conf

メールボックスの位置などを設定するファイルです。今回は mbox 形式のメールボックスを指定します。

```
# vi /etc/dovecot/conf.d/10-mail.conf
```

```
#mail_location =
 mail_location = mbox:~/mail:INBOX=/var/mail/%u ← この設定行を追加
```

/etc/dovecot/conf.d/10-auth.conf

認証を設定するファイルです。今回は暗号化していない平文での認証を許可し、Linux のログイン情報を認証に利用できるように設定します。

```
# vi /etc/dovecot/conf.d/10-auth.conf
```

```
#disable_plaintext_auth = yes
 disable_plaintext_auth = no ← この設定行を追加
（略）
auth_mechanisms = plain  login ← login を追加
```

設定が終わったら、Dovecot を起動します。

```
# /etc/init.d/dovecot start
Dovecot Imap を起動中:                                     [  OK  ]
# chkconfig dovecot on
# chkconfig --list dovecot
dovecot         0:off   1:off   2:on    3:on    4:on    5:on    6:off
```

### 6.7.3 Thunderbird のインストール

メールクライアントとして Thunderbird をインストールします。

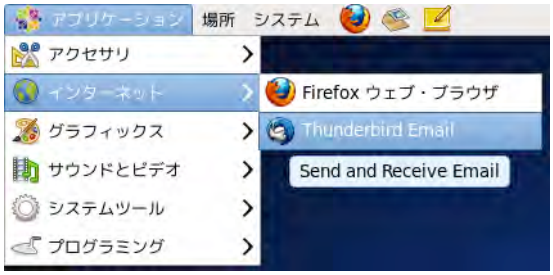
```
# /media/CentOS_6.2_Final/Packages/
# rpm -ivh thunderbird-3.1.16-2.el6.centos.i686.rpm
準備中...                ########################################### [100%]
   1:thunderbird            ########################################### [100%]
# cd
```

### 6.7.4 Thunderbird の設定

次に Thunderbird の設定を行います。以下の手順は受講者 A の場合です。

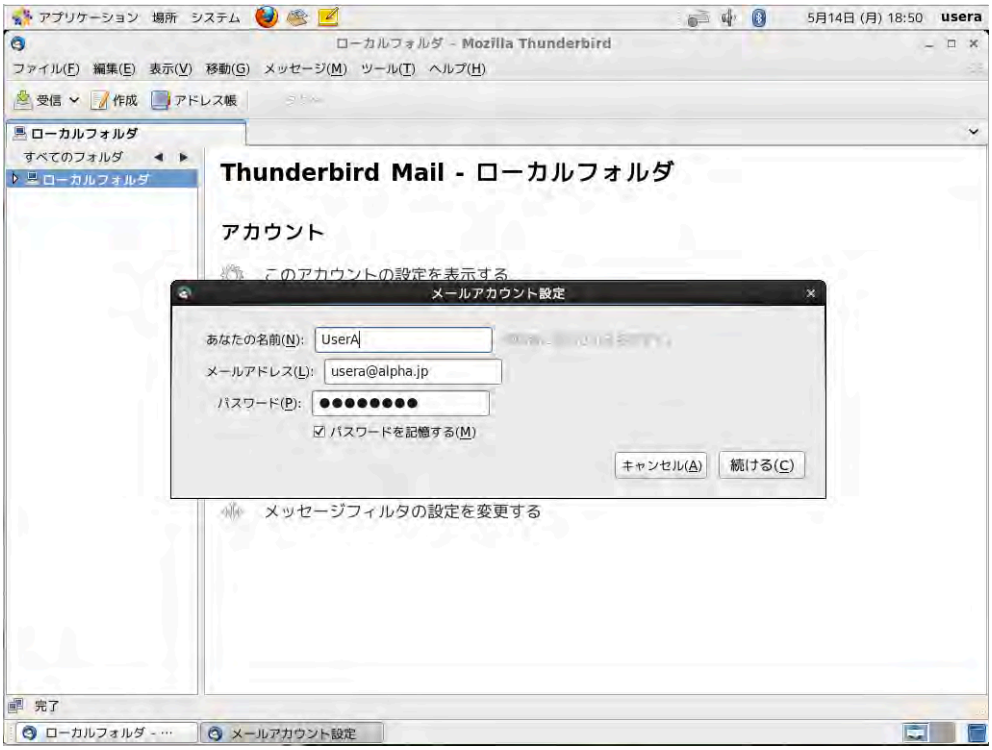
1. root でログインしている場合にはログアウトします。ログアウトは「システム」メニューから「root のログアウト…」を選択します。
2. メールの送受信テスト用に作成したユーザーアカウント usera でログインします。パスワードは userapass です。正しく設定されていない場合には、再度 root でログインし、passwd コマンドで設定し直して下さい。このパスワードがメールの送受信にも使用されます。

3.「アプリケーション」メニューから「インターネット」→「Thunderbird Email」を選択します。

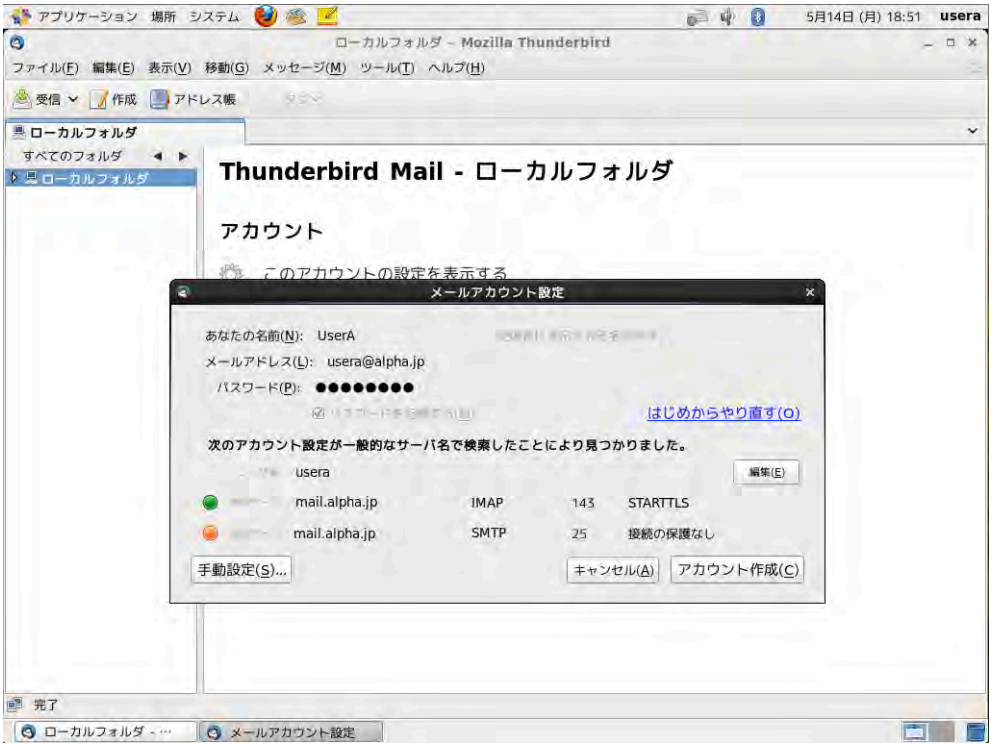



4.「メールアカウント設定」ダイアログが表示されます。以下のように設定して「続ける」をクリックします。

| | |
|---|---|
| **あなたの名前** | UserA |
| **メールアドレス** | usera@alpha.jp |
| **パスワード** | userapass |
| **パスワードを記憶する** | チェックしておく |

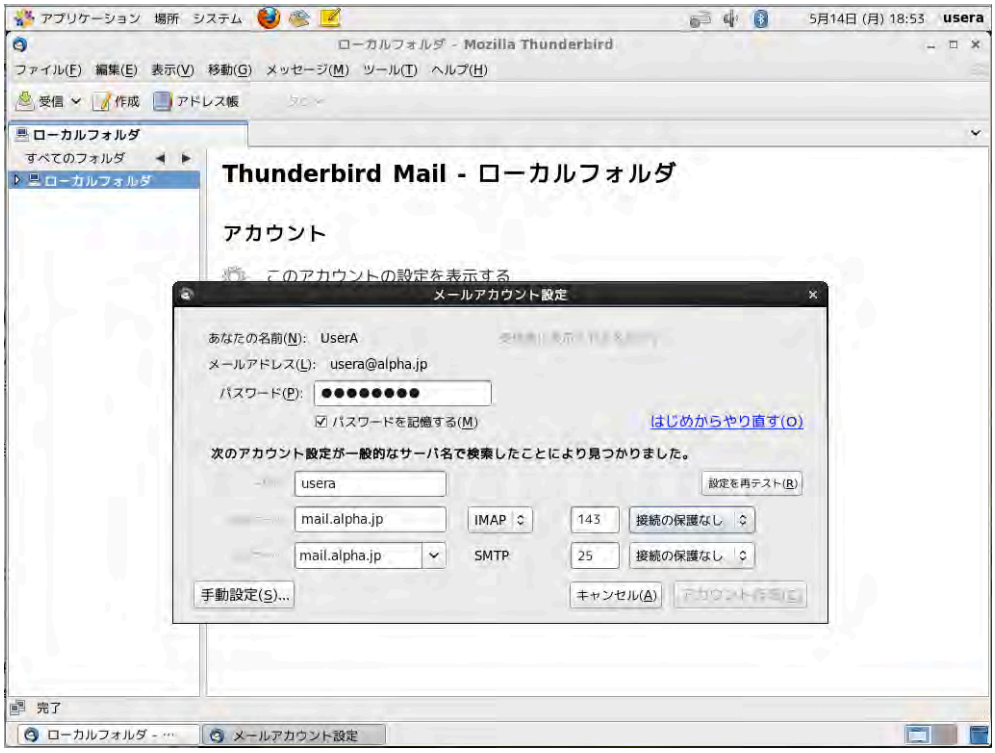


5.「アカウント設定を Mozilla ISP データベースから検索しています。」と表示されます。検索はしばらく時間がかかります。

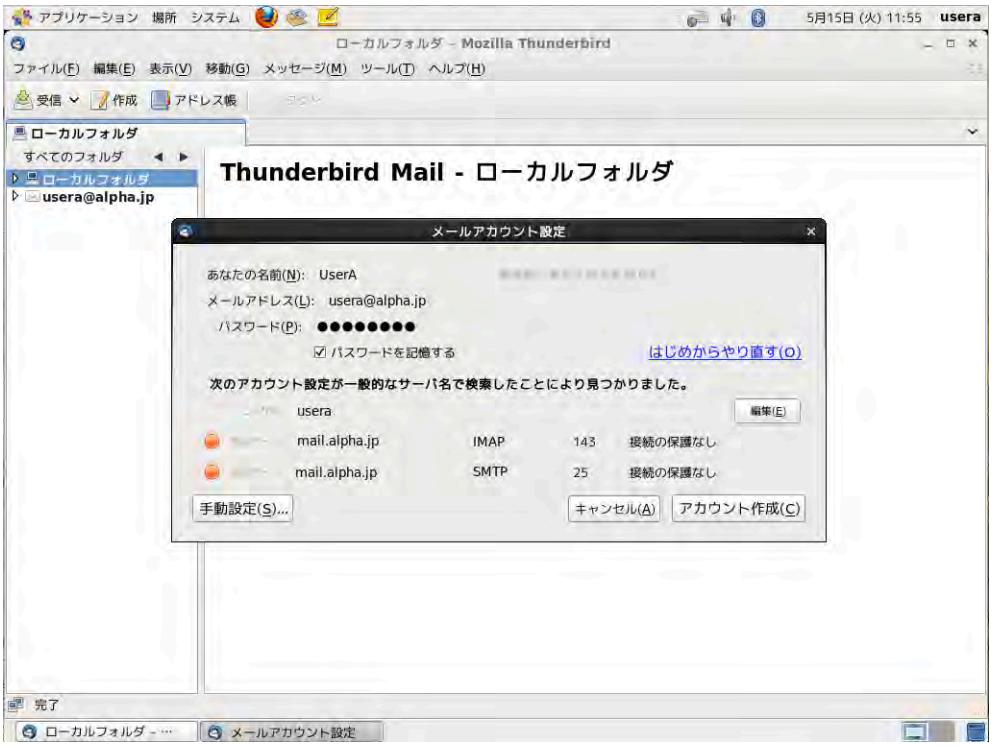


実習環境は正しく自動設定されないので、「編集」ボタンが表示されたらクリックして、以下のように設定します。

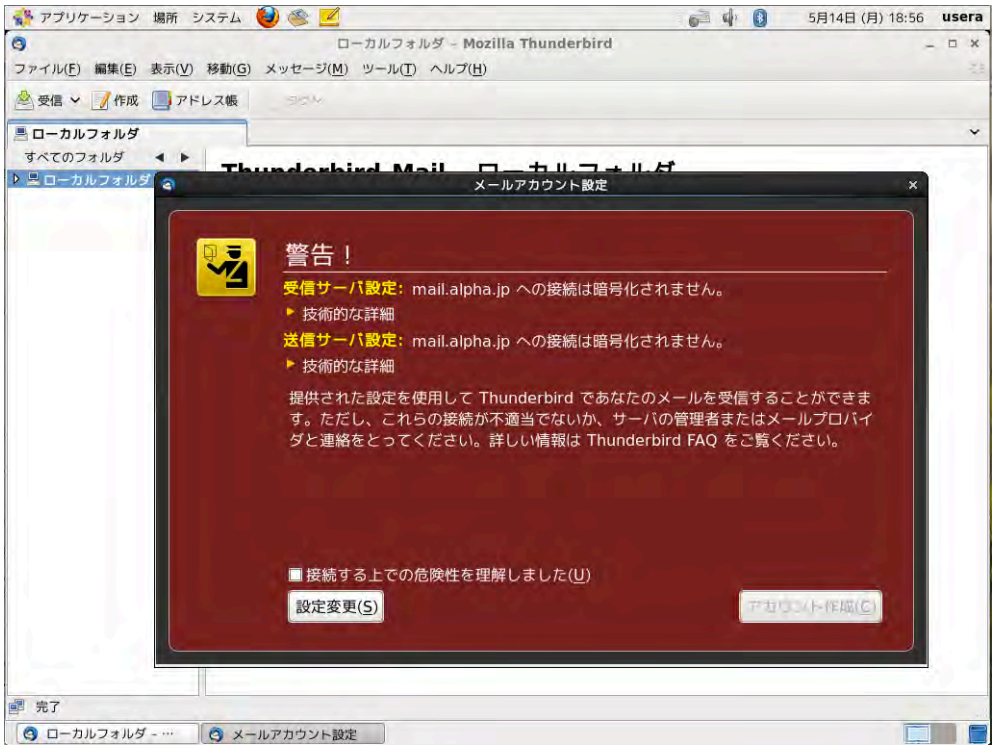
| | |
|---|---|
| **ユーザ名** | usera |
| **受信サーバー** | mail.alpha.jp |
| **プロトコル** | IMAP |
| **受信ポート番号** | 143 |
| **接続の保護** | 接続の保護なし（STARTTLS から変更） |
| **送信サーバー** | mail.alpha.jp |
| **送信ポート番号** | 25 |
| **接続の保護** | 接続の保護なし |



6. 設定が終わったら「設定を再テスト」をクリックします。受信サーバー、送信サーバー共に設定が確認されたら、「アカウント作成」ボタンをクリックします。



7. 接続が暗号化されないため、警告が表示されます。「接続する上での危険性を理解しました」をチェックし、「アカウント作成」ボタンをクリックします。



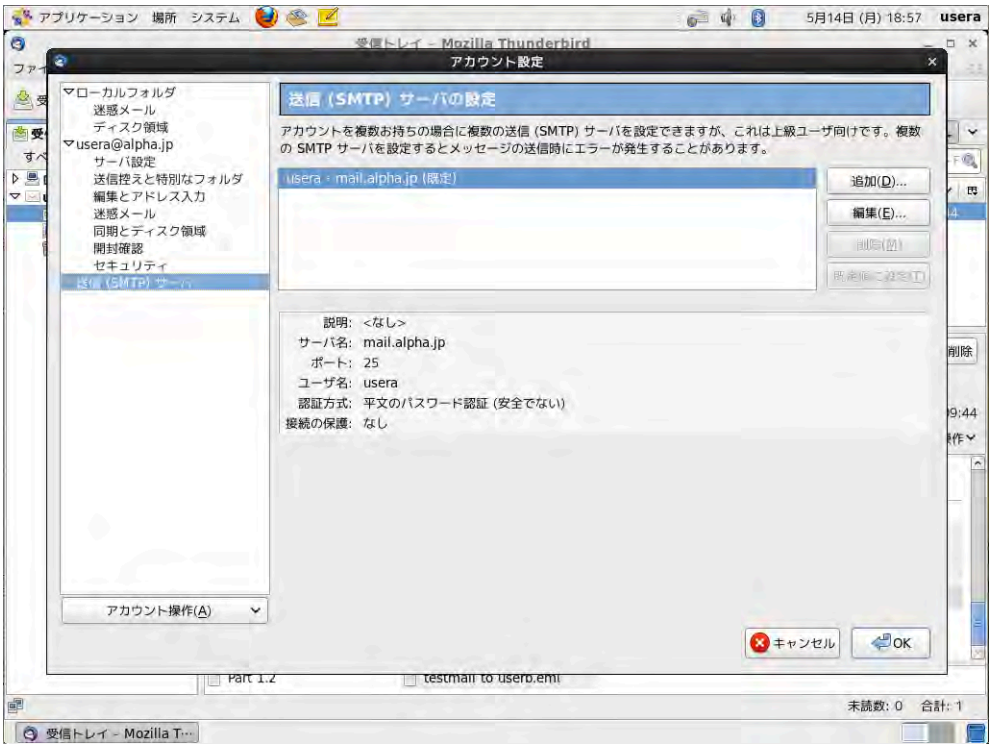

8. 右上に警告が表示されますが、初回起動時のみ表示される警告です。またインターネットに接続できない環境で Thunderbird 起動時に「サーバーが見つかりませんでした」のエラーが表示されますが、どちらも無視して構いません。

### 6.7.5 メール送信時の認証設定

自動設定ではメール送信時の認証設定が間違って設定されているので、修正しておきます。

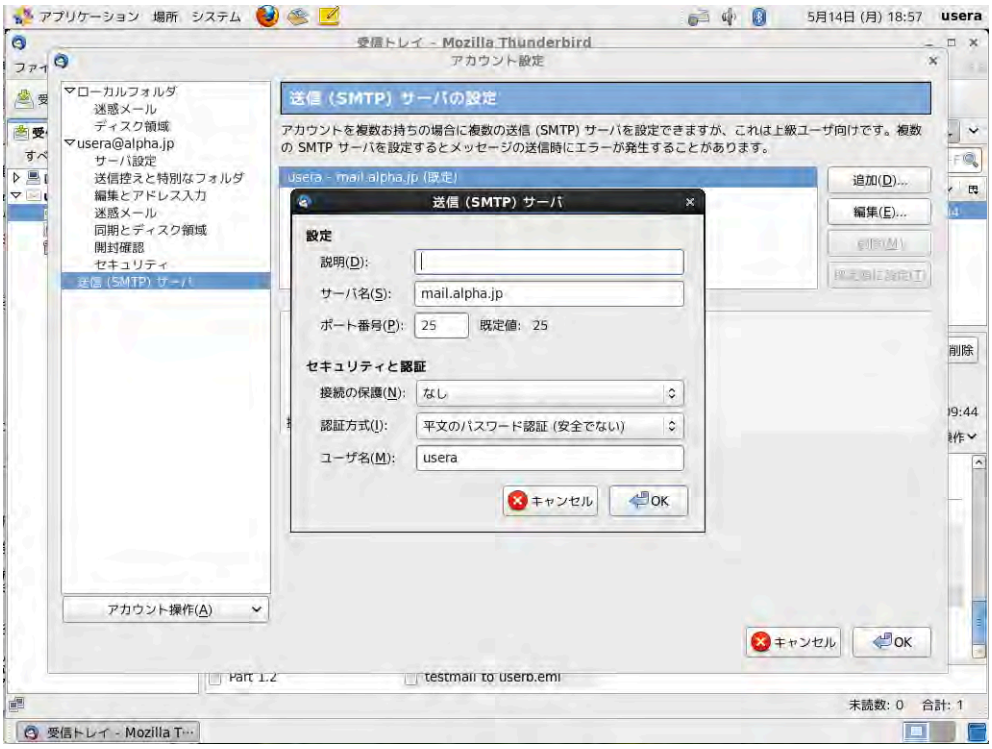
1.「編集」メニューから「アカウント設定」を選択します。
2. 左側のリストから「送信 (SMTP) サーバー」を選択します。

3.「usera - mail.alpha.jp(既定)」を選択し、「編集」ボタンをクリックします。



4.「送信 (SMTP) サーバー」設定ダイアログが表示されるので、以下のように設定します。設定が終わったら、「OK」ボタンを 2 回クリックして設定を終了します。

| | |
|---|---|
| **サーバー名** | mail.alpha.jp |
| **ポート番号** | 25 |
| **接続の保護** | なし |
| **認証方式** | 平文のパスワード認証 (安全でない) |
| **ユーザ名** | usera |




## 6.7.6 メールの送信

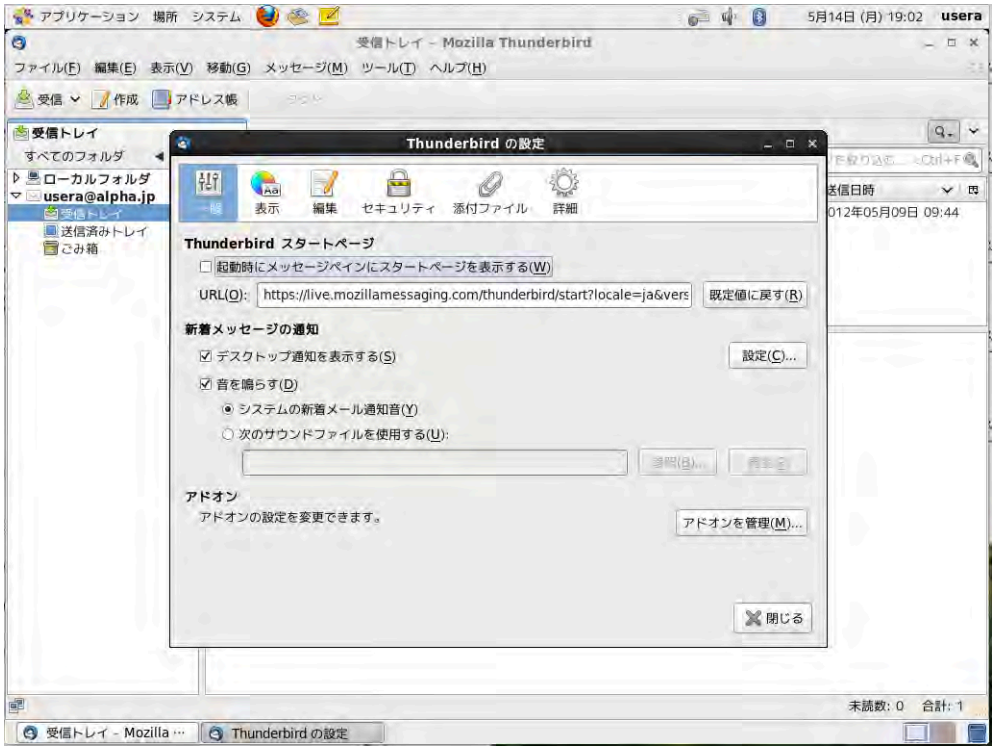
メールを送信するには、「作成」ボタンをクリックしてメール作成ウインドウを呼び出します。

1. 宛先に自分のメールアドレス（usera@alpha.jp）を指定して、メールを作成、送信してみます。
2.「受信」ボタンをクリックして、メールが受信できることを確認します。
3. 宛先に他の受講生のメールアドレス（userb@beta.jp）を指定して、メールを作成、送信してみます。
4. 相手がメールを受信できたこと、相手からのメールを受信できることを確認します。

### 6.7.7　起動時のスタートページ表示の設定

起動時の「サーバーが見つかりませんでした」のエラーが表示されないようにするには、以下の手順で設定を修正します。

1.「編集」メニューから「設定」を選択します。
2.「Thunderbird スタートページ」の「起動時にメッセージペインにスタートページを表示する」のチェックを外して、「閉じる」をクリックします。




## 6.8　まとめ

本章では、電子メールに関する学習を行いました。また、実際にメールサーバーを設定し、mail コマンドや Thunderbird を利用してメールの送受信の確認を行いました。メールサーバーの設定は、メールサーバーが正しく設定され起動していたとしても、DNS サーバーが正しく動いていなければ利用できないなどの理由から難しかったと思います。設定ファイルの記述に問題がないのに、メールがどうしても送られない、受信できない場合は、まず DNS が正しく動いているか、nslookup コマンドや dig コマンドを実行して確認します。また、ログ（/var/log/messages）を見て、エラーが出ていないか確認することも大切です。

# 第7章

# ファイル共有

職場や家庭には多くの Windows マシンがあり、Windows マシンとのファイル共有は大きな需要があります。ファイル共有のサービスを提供する Samba を見ていきます。

## 7.1 用語集

### Samba

UNIX および UNIX 互換マシンをファイルサーバー/プリントサーバーにする、Windows の SMB（Server Message Block）プロトコル互換のオープンソースソフトウェアです。

### SMB サービス

SMB/CIFS ファイル共有をクライアントへ提供する smbd サービスと、クライアントへ NetBIOS ネームサーバー機能を提供する nmbd サービスにより構成されています。

### smb.conf

Samba の設定を行うファイルです。Samba によるファイル共有などの設定をこのファイルに書き込みます。

### smbclient

ftp 接続クライアントに似た Samba サーバーに接続するクライアントツールです。

### smbpasswd

Samba のユーザーアカウントとマシンアカウントの管理のためのツールの一つです。

## 7.2 Samba とは

オープンソースでファイル共有の機能を提供するためのサービスとして、Samba が使われています。Samba は資源を共有するファイルサーバー機能とプリンタサーバー機能、マシンやユーザーを管理する Windows ドメインコントローラ機能と、Active Directory ドメインメンバ機能を提供します。Samba の機能の中で需要が多いのは、ファイルサーバー機能です。ファイルサーバー機能は Linux のディレクトリを共有フォルダとして Windows マシンへ提供します。

## 7.3 Samba パッケージの追加

Samba を動作させるには samba パッケージと samba-common パッケージが必要です。必要なパッケージがインストールされているか、rpm コマンドで確認してください。インストールされていない場合は、rpm コマンドでインストールしてください。

```
# rpm -q samba samba-common
パッケージ samba はインストールされていません。
samba-common-3.5.10-114.el6.i686
# cd /media/CentOS_6.2_Final/Packages/
# rpm -ivh samba-3.5.10-114.el6.i686.rpm
準備中...                ########################################### [100%]
   1:samba                  ########################################### [100%]
# cd
```

## 7.4 Samba サービスの設定ファイル

Samba の設定ファイルは/etc/samba/smb.conf ファイルです。

smb.conf ファイルには [global] と [homes] と [printers] という特殊なセクションがあります。

| セクション | 意味 |
|---|---|
| [global] | 全体の設定 |
| [homes] | 各ユーザーのホームディレクトリ共有の設定 |
| [printers] | プリンタ共有の設定 |

インストール後の時点ではホームディレクトリ共有以外のファイル共有は定義されていないので、必要に応じて [セクション名] という書き出しで、新しいセクションを定義することができます。

各セクションの中には「パラメータ=設定」という記述をします。

パラメータ = 設定

主な設定項目として、次表のようなパラメータがあります。

## 7.5 誰でも読み書きできるファイル共有の作成

誰でも読み書きできるファイル共有を、以下の手順で作成します。

### 7.5.1 設定ファイルの変更

エディタで/etc/samba/smb.conf ファイルを開き、[global] セクションで「security = share」（アクセス制限なし）に設定を変更します。また、共有の設定となる「[lpic] セクション」を作ります。

1. 共有するディレクトリを作成します。

```
# mkdir /home/lpicshare
# chmod 777 /home/lpicshare
# touch /home/lpicshare/testfile
```

2. /etc/samba/smb.conf を編集します。

```
# vi /etc/samba/smb.conf
```

```
[global]
 (略)
# ----------------------- Standalone Server Options ------------------------
#
# Scurity can be set to user, share(deprecated) or server(deprecated)
#
# Backend to store user information in. New installations should
# use either tdbsam or ldapsam. smbpasswd is available for backwards
# compatibility. tdbsam requires no further configuration.

        security = share
        passdb backend = tdbsam
 (略)
#============================ Share Definitions ==============================

[lpic]
        comment = LPIC File Share
        browseable = yes
        writable = yes
        guest ok = yes
        guest only = yes
        path = /home/lpicshare
```

3. smb サービスを起動します。自動機能も有効にしておきます。

```
# /etc/init.d/smb start
SMB サービスを起動中:                                      [  OK  ]
# chkconfig smb on
# chkconfig --list smb
smb             0:off   1:off   2:on    3:on    4:on    5:on    6:off
```

### 7.5.2 Samba が提供する共有の確認

Samba が提供する共有を確認するには、smbclient コマンドに-L オプションをつけて実行します。次のように-L オプションの後に確認したいホスト名を記述します。

```
# smbclient -L localhost
Enter root's password: ← パスワードは指定せず Enter キーを押す
Domain=[MYGROUP] OS=[Unix] Server=[Samba 3.5.10-114.el6]

        Sharename       Type      Comment
        ---------       ----      -------
        lpic            Disk      LPIC File Share
        IPC$            IPC       IPC Service (Samba Server Version 3.5.10-114.el6)
Domain=[MYGROUP] OS=[Unix] Server=[Samba 3.5.10-114.el6]

        Server               Comment
        ---------            -------

        Workgroup            Master
        ---------            -------
```

### 7.5.3 共有への接続

lpic 共有へ接続します。smbclient コマンドの引数にホスト名と共有名を指定します。smbclient の仕様上パスワードの入力が促されますが、Samba 側で security = share に設定しているのでパスワードを入力する必要はありません。

```
# smbclient //localhost/lpic
Enter root's password:  ← パスワードは指定せず Enter キーを押す
Domain=[MYGROUP] OS=[Unix] Server=[Samba 3.5.10-114.el6]
Server not using user level security and no password supplied.
smb: ¥>
```

通常のコマンドライン同様、ls コマンドでファイル一覧が表示できます。

```
smb: ¥> ls
  .                                   D        0  Wed May  9 14:40:08 2012
  ..                                  D        0  Wed May  9 14:39:58 2012
  testfile                            0  Wed May  9 14:40:08 2012

                55176 blocks of size 524288. 52006 blocks available
smb: ¥> exit
# ls -la /home/lpicshare/
合計 8
drwxrwxrwx  2 root root 4096  5 月  9 14:40 2012 .
drwxr-xr-x. 6 root root 4096  5 月  9 14:39 2012 ..
-rw-r--r--  1 root root    0  5 月  9 14:40 2012 testfile
```

### 7.5.4 Windows マシンからの共有へのアクセス

Samba は、デフォルトでホスト名が Windows ネットワークのコンピューター名となります。Windows マシンの「ネットワークコンピュータ」(Windows XP の場合)「ネットワーク」(Windows 7 の場合) などから Samba が動作するコンピューターを見つけるか、Explorer に以下のアドレスを入力して共有にアクセスします。

¥¥ホスト名または IP アドレス ¥共有名

たとえば、host1.alpha.jp に作成した lpic 共有にアクセスするには、以下のように入力します。

```
¥¥host1¥lpic
```

## 7.6 Samba のパスワード認証の設定

Samba によるファイル共有へのアクセスにパスワード認証を設定するには、smb.conf の設定を修正するほか、アクセスを許可するユーザーとパスワードの登録が必要です。

### 7.6.1 Samba のパスワード認証の設定

Samba のパスワード認証を有効にするには、/etc/samba/smb.conf の [global] セクションに「security = user」と設定します。

```
# vi /etc/samba/smb.conf
```

```
[global]
 (略)
        security = user
```

[global] セクションの設定を変更した場合には、smb サービスの再起動が必要です。

```
# /etc/init.d/smb restart
SMB サービスを停止中:                                      [  OK  ]
SMB サービスを起動中:                                      [  OK  ]
```

### 7.6.2 Samba ユーザーとパスワードの登録

Samba サービスのファイル共有でユーザー認証をするには Samba 用のユーザーとパスワードを登録する必要があります。Samba 用のユーザーとパスワードを登録するには smbpasswd コマンド

を使います。はじめてユーザーを登録する場合には-a オプションをつけて smbpasswd コマンドを実行します。

```
# smbpasswd -a usera
New SMB password: userapass ← 入力文字は非表示
Retype new SMB password: userapass ← 入力文字は非表示
Added user usera.
```

### 7.6.3 Samba が提供する共有の確認

smbclient コマンドで Samba が提供する共有を確認します。su コマンドを使って、あらかじめ Samba に登録してあるユーザーに変更してから smbclient コマンドを実行することで、Samba に接続するためのユーザー名を暗黙のうちに指定できます。

```
# su - usera
$ smbclient -L localhost
Enter usera's password: userapass ← 入力文字は非表示
Domain=[MYGROUP] OS=[Unix] Server=[Samba 3.5.10-114.el6]

        Sharename       Type      Comment
        ---------       ----      -------
        lpic            Disk      LPIC File Share
        IPC$            IPC       IPC Service (Samba Server Version 3.5.10-114.el6)
        usera           Disk      Home Directories
Domain=[MYGROUP] OS=[Unix] Server=[Samba 3.5.10-114.el6]

        Server               Comment
        ---------            -------

        Workgroup            Master
        ---------            -------
```

usera のパスワードが要求されていることと、usera という名前の共有ががユーザー名で共有として利用できることが確認できます。

### 7.6.4 共有への接続

smbclient コマンドで usera 共有に接続します。

```
$ smbclient //localhost/usera
Enter usera's password:
Domain=[MYGROUP] OS=[Unix] Server=[Samba 3.5.10-114.el6]
smb: ¥> ls
  .                                   D        0  Thu Apr 26 11:30:42 2012
  ..                                  D        0  Tue Apr 24 17:41:12 2012
  ドキュメント                    D        0  Tue Apr 24 17:19:56 2012
  .gnupg                             DH        0  Tue Apr 24 17:19:56 2012
```

```
 （略）
  .spice-vdagent                      DH        0  Tue Apr 24 17:19:56 2012
  ビデオ                             D        0  Tue Apr 24 17:19:56 2012
  .ICEauthority                       H      310  Tue Apr 24 17:19:56 2012

                55176 blocks of size 524288. 52005 blocks available
smb: ¥> exit
$ ls -la
合計 172
drwx------  29 usera usera  4096  4 月 26 11:30 2012 .
drwxr-xr-x.  5 root  root   4096  4 月 24 17:41 2012 ..
-rw-------   1 usera usera   310  4 月 24 17:19 2012 .ICEauthority
-rw-------   1 usera usera  1158  5 月  7 17:59 2012 .bash_history
-rw-r--r--   1 usera usera    18 12 月  2 23:40 2011 .bash_logout
-rw-r--r--   1 usera usera   176 12 月  2 23:40 2011 .bash_profile
 （略）
drwxr-xr-x   2 usera usera  4096  4 月 24 17:19 2012 ビデオ
drwxr-xr-x   2 usera usera  4096  4 月 24 17:19 2012 音楽
drwxr-xr-x   2 usera usera  4096  4 月 24 17:19 2012 画像
drwxr-xr-x   2 usera usera  4096  4 月 24 17:19 2012 公開
```

usera のホームディレクトリが共有されていることが確認できます。

# 第8章

# セキュリティ

インターネットに公開するサーバーはセキュリティについても考慮しておく必要があります。本章では、メンテナンスのためのリモートログインを安全に行うための SSH、アクセス制御、パケットフィルタリング（ファイアウォール）について解説します。

## 8.1 SSH によるリモートログイン

SSH はネットワーク経由でリモートにある Linux サーバーにログインするために使用するプロトコルです。通信が暗号化されているため、覗き見されてもパスワードや作業内容が分からない他、公開鍵を使った認証を行うことでパスワードをネットワークに流すことなくログインすることができます。Linux では、OpenSSH のサーバーとクライアントが用意されています。

### 8.1.1 TELNET との違い

SSH と同様のリモートログインに TELNET が使用できますが、TELNET は通信が暗号化されていないためパスワードや作業内容などを覗き見することができてしまうという問題があります。SSH は特別な設定をしないでも TELNET と同様のパスワードによるリモートログインが行えるので、通信が暗号化されている SSH が標準的に使われており、現在では TELNET を使用する必要は無くなっています。CentOS 6.2 では、デフォルトでは telnet コマンドはインストールされていません。

### 8.1.2 必要なパッケージを確認

OpenSSH のサーバーとクライアントに必要なパッケージを確認します。

```
# rpm -qa | grep ssh
openssh-5.3p1-70.el6_2.2.i686
openssh-server-5.3p1-70.el6_2.2.i686
libssh2-1.2.2-7.el6_1.1.i686
openssh-clients-5.3p1-70.el6_2.2.i686
openssh-askpass-5.3p1-70.el6_2.2.i686
```

openssh-server パッケージがサーバー、openssh-clients パッケージがクライアントです。インストールされていない場合には、それぞれのパッケージをインストールしてください。

### 8.1.3 chkconfig で起動時の設定

chkconfig コマンドで Linux 起動時に OpenSSH サーバーが開始するか否かを確認、設定できます。

```
# chkconfig --list sshd
sshd            0:off   1:off   2:on    3:on    4:on    5:on    6:off
```

もし、自動的に起動しない設定（off）になっていた場合には、自動起動するように設定しておきます。また、起動スクリプトで起動しておきます。

```
# chkconfig sshd on
# /etc/init.d/sshd start
sshd を起動中:                                             [  OK  ]
# /etc/init.d/sshd status
openssh-daemon (pid  16404) を実行中...
```

### 8.1.4 パスワード認証による接続

ssh コマンドは特別な設定を行わなくても、パスワード認証でリモートログインすることができます。

以下のようにして、自分自身に SSH で接続してみます。初めての接続の場合には、SSH サーバーの電子証明書が送られてきて接続してもよいか訪ねられるので「yes」と入力します。パスワード認証が可能だと、パスワードの入力が要求されます。

```
# ssh usera@localhost ← ユーザー usera として localhost に接続します。
The authenticity of host 'localhost (::1)' can't be established.
RSA key fingerprint is af:24:60:1c:9c:ed:5e:f0:e0:73:38:ad:5b:a1:f3:22.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'localhost' (RSA) to the list of known hosts.
usera@localhost's password: userapass ← 実際には非表示
$ exit ← リモートログインを終了
logout
Connection to localhost closed.
# ← 元のユーザー root に復帰
```

### 8.1.5 公開鍵認証による接続

パスワード認証は、通信経路が SSH で暗号化されているといっても、パスワードがネットワークを流れていること、またパスワードを自動的に生成して順番に試していく「総当たり攻撃」を受けた場合不正にログインされてしまう可能性があるので、インターネット上に公開されているサーバー

で使用するには相応しくありません。

公開鍵認証は、あらかじめサーバーに設置した公開鍵と対になっている秘密鍵を持っているユーザーしかリモートログインできない認証方法です。

以下の手順で公開鍵認証を設定します。

1. 公開鍵と秘密鍵を生成する

ssh-keygen コマンドを使用して一対の公開鍵 (id_dsa.pub) と秘密鍵 (id_dsa) を生成します。鍵のファイルはホームディレクトリに作られた.ssh ディレクトリに保存されます。秘密鍵には不正利用を防止するためのパスフレーズを設定します。接続時にパスフレーズを正しく入力できないと、秘密鍵は利用できないので、公開鍵認証による接続はできません。このパスフレーズは SSH クライアント側で秘密鍵に対して処理されるので、ネットワーク上には情報は流れません。

```
[root@host1 ~]# su - usera ← ユーザー usera にユーザーを切り替え
[usera@host1 ~]$ ssh-keygen -t dsa ← DSA 暗号形式で鍵を生成
Generating public/private dsa key pair.
Enter file in which to save the key (/home/usera/.ssh/id_dsa): ← Enter キーを入力
Enter passphrase (empty for no passphrase): ← パスフレーズを入力（非表示）
Enter same passphrase again: ← パスフレーズを再度入力（非表示）
Your identification has been saved in /home/usera/.ssh/id_dsa.
Your public key has been saved in /home/usera/.ssh/id_dsa.pub.
The key fingerprint is: ↓鍵についた指紋。鍵の照合に使用可能
8f:10:6c:bb:b8:e8:de:30:26:fc:e9:ae:fd:b0:61:b0 usera@host1.alpha.jp
The key's randomart image is:
+--[ DSA 1024]----+
|                 |
|     .           |
|      +          |
|     . o         |
| .    o S        |
|. o  . o o       |
|.E++. . . .      |
| o+==.           |
| +BB+.           |
+-----------------+
[usera@host1 ~]$
```

2. 接続先に authorized_keys を作成する

ユーザーに SSH での接続を許可するには、ユーザーアカウントを作成し、そのユーザーのホームディレクトリに~/.ssh/authorized_keys ファイルを作成しておきます。~/.ssh/のパーミッションは 700(drwx——)、authorized_keys ファイルのパーミッションは 600(-rwx——) に設定する必要があります。

```
[usera@host1 ~]$ ls -ld .ssh
drwx------ 2 usera usera 4096  4 月 26 10:57 2012 .ssh
[usera@host1 ~]$ cd .ssh
[usera@host1 .ssh]$ cat id_dsa.pub >> authorized_keys
```

```
[usera@host1 .ssh]$ chmod 600 authorized_keys
[usera@host1 .ssh]$ ls -l authorized_keys
-rw------- 1 usera usera 610  4 月 26 11:05 2012 authorized_keys
```

3. 公開鍵認証で接続する

公開鍵認証で接続します。ssh コマンドの使用法自体はパスワード認証と同じですが、パスワードの代わりに秘密鍵に設定したパスフレーズの入力が必要です。

```
[usera@host1 ~]$ ssh usera@localhost
The authenticity of host 'localhost (::1)' can't be established.
RSA key fingerprint is af:24:60:1c:9c:ed:5e:f0:e0:73:38:ad:5b:a1:f3:22.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'localhost' (RSA) to the list of known hosts.
Enter passphrase for key '/home/usera/.ssh/id_dsa': ← パスフレーズを入力（表示されません）
Last login: Thu Apr 26 10:53:12 2012 from localhost
[usera@host1 ~]$ exit
logout
Connection to localhost closed.
```

### 8.1.6 パスワード認証の禁止

パスワード認証が有効になっていると、パスワードの総当たり攻撃により不正にリモートログインできてしまいます。公開鍵認証で接続できるようになった後には、OpenSSH サーバーの設定を変更してパスワード認証を禁止しておきます。

1. パスワード認証で接続できることを確認します。

```
# ssh root@localhost
root@localhost's password: ← パスワードを入力（非表示）
Last login: Mon May  7 18:05:57 2012 from 192.168.1.101
# exit
logout
Connection to localhost closed.
```

2. 設定ファイル/etc/ssh/sshd_config を修正します。

```
# vi /etc/ssh/sshd_config
```

```
（略）
# To disable tunneled clear text passwords, change to no here!
#PasswordAuthentication yes
 PasswordAuthentication no ← no に変更
（略）
```

3. 設定を変更後、sshd サービスを再起動します。

```
# /etc/init.d/sshd restart
sshd を停止中:                                             [  OK  ]
sshd を起動中:                                             [  OK  ]
```

4. 公開鍵認証を設定していないユーザーで OpenSSH サーバーに接続します。

```
# ssh root@localhost
Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
```

公開鍵認証が行えないので、接続が行えません。

## 8.2 TCP Wrapper によるアクセス制御

TCP Wrapper を使うと、サーバーに対するアクセスを制御できます。アクセス制御はアクセス先のサービスと、アクセス元の IP アドレスの組み合わせでアクセスの許可、拒否を設定できます。

### 8.2.1 TCP Wrapper の確認

TCP Wrapper が使用できるかどうかは ldd コマンドで確認できます。ldd コマンドは指定されたプログラムのバイナリがリンクしているライブラリの一覧を表示します。TCP Wrapper が使用できる場合には libwrap.so.0 がリンクされています。

以下の例では、OpenSSH サーバー（sshd）は TCP Wrapper でアクセス制御ができることが分かります。

```
# ldd /usr/sbin/sshd | grep libwrap
        libwrap.so.0 => /lib/libwrap.so.0 (0x001b8000)
```

### 8.2.2 /etc/hosts.allow と/etc/hosts.deny の設定

TCP Wrapper の設定ファイルは/etc/hosts.allow（許可）と/etc/hosts.deny（拒否）の 2 つのファイルで行います。それぞれのファイルに、制御したいサービスのプログラム名とホスト名、または IP アドレスを:（セミコロン）で区切って記述します。

```
サービス名 : ホスト名または IP アドレス
```

設定は以下の順番で評価されます。

1. /etc/hosts.allow にマッチする記述があればアクセスを許可
2. /etc/hosts.deny にマッチする記述があればアクセスを拒否
3. どちらにも記述がなければアクセスを許可

### 8.2.3 OpenSSH サーバーへのアクセス制御

OpenSSH サーバーへのアクセス制御を行ってみます。

1. OpenSSH サーバーへのアクセス拒否を設定

/etc/hosts.deny に以下のように記述し、OpenSSH サーバーへのアクセスを拒否します。

```
# vi /etc/hosts.deny
```

```
（略）
sshd : ALL
```

2. OpenSSH サーバーに接続できないことを確認

OpenSSH サーバーに接続できないことを確認します。

```
# ssh usera@localhost
ssh_exchange_identification: Connection closed by remote host
```

TCP Wrapper により、OpenSSH サーバーへのアクセスが拒否されました。

3. OpenSSH サーバーへのアクセス許可を設定

/etc/hosts.allow に以下のように記述し、OpenSSH サーバーへのアクセスを許可します。

```
# vi /etc/hosts.allow
```

```
sshd : ALL
```

4. OpenSSH サーバーに接続できることを確認

OpenSSH サーバーに接続できることを確認します。

```
# ssh usera@localhost
usera@localhost's password: ← パスフレーズを入力（非表示）
Last login: Thu Apr 26 11:46:47 2012 from localhost
```

hosts.allow の設定が優先されたため、OpenSSH サーバーへのアクセスが許可されました。

### 8.2.4 TCP Wrapper の使い方

/etc/hosts.allow と/etc/hosts.deny が評価される順番を考慮すると、TCP Wrapper を活用するには、以下のように設定するのがよいでしょう。

1. /etc/hosts.allow でアクセスを許可したいサービスおよびホストを指定する
2. /etc/hosts.deny に「ALL : ALL」と指定し、すべてのホストからのすべてのサービスへのアクセスを拒否する

## 8.3 iptables によるパケットフィルタリング

iptables を使用すると、サーバーに送られてくるパケットを、送信元や送信先の IP アドレス、ポート番号などを元に、受け入れるか否かなどの制御を行うことができます。このような制御を「パケットフィルタリング」と呼びます。

### 8.3.1 iptables の概要

iptables は、Linux カーネルに組み込まれているパケットフィルタリングの機能です。サーバーの受信パケットおよび送信パケット、さらにネットワークインターフェースが複数ある場合にはインターフェース間のパケットの転送について、許可、または拒否のルールを設定します。

### 8.3.2 iptables の設定

1. setup コマンドを実行します。

2.「ファイアウォールの設定」を選択し、Enter キーを押します。
ツールの選択はカーソルキーの上下で行えます。選択項目は TAB キーで移動できます。
3.「ファイアウォール: [ ] 有効」でスペースキーを押して、チェック [*] を入れます。
4.「OK」を選択し、Enter キーを押します。
選択項目は TAB キーで移動できます。
5. 警告を確認し、「はい」を選択して Enter キーを押します。
この設定方法では、iptables サービスが有効になっている必要があります。この後の手順で結果が正しくない場合は「service iptables start」コマンドを実行して iptables サービスを開始します。
6.「終了する」を選択して、Enter キーを押します。

### 8.3.3 iptables の設定確認

iptables の設定状態を確認します。確認するには iptables コマンドに-L オプションを付けて実行します。

```
# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere            state RELATED,ESTABLISHED
ACCEPT     icmp --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere
ACCEPT     tcp  --  anywhere             anywhere            state NEW tcp dpt:ssh
REJECT     all  --  anywhere             anywhere            reject-with icmp-host-proh

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
REJECT     all  --  anywhere             anywhere            reject-with icmp-host-proh

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

「Chain」とは、パケットの受信（INPUT）、転送（FORWARD）、送信（OUTPUT）のそれぞれの設定ルールをまとめたものです。「policy」はその Chain が基本的にパケットを送受信、転送を許可 (ACCEPT) するのか拒否 (REJECT) するのかの設定となります。
各ルールは、それぞれ以下のようにな意味となります。

- ACCEPT all – anywhere anywhere state RELATED,ESTABLISHED

TCP 通信で自分から通信を始めたやり取りのパケットに関するものは受け入れる。

- ACCEPT icmp – anywhere anywhere

ICMP はすべて受け入れる。

- ACCEPT all – anywhere anywhere

lo インターフェースはすべて受け入れる。この設定については iptables -L の表示だけでは判断できません。設定ファイルである/etc/sysconfig/iptables に以下のように-i lo とインターフェースの指定が記述されていることで分かります。

```
-A INPUT -i lo -j ACCEPT
```

- ACCEPT tcp – anywhere anywhere state NEW tcp dpt:ssh

ssh のポートへの接続は受け入れます。

- REJECT all – anywhere anywhere reject-with icmp-host-prohibited

その他のパケットはすべて拒否 (REJECT) します。

### 8.3.4 iptables の動作の確認

iptables の動作を確認します。現在の設定では、OpenSSH サーバーへの接続は許可されるが、その他のサービスへの接続は拒否されます。

1. OpenSSH サーバーへの接続

以下のように、他の受講者の OpenSSH サーバーへの接続を確認します。

```
# ssh root@host2.beta.jp
The authenticity of host 'host2.beta.jp (192.168.1.102)' can't be established.
RSA key fingerprint is cf:91:cc:b2:69:a9:c0:74:6e:df:3e:f1:6a:0b:85:dd.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'host2.beta.jp,192.168.1.102' (RSA) to the list of known host
root@host2.beta.jp's password: ← 接続できたので Ctrl+C キーを押して終了
```

2. Web サーバーへの接続

Web ブラウザーで、他の受講者の Web サーバーに接続できないことを確認します。

```
http://www.beta.jp
```

### 8.3.5 iptables の許可ルールの設定

iptables に許可ルールを設定します。設定情報は/etc/sysconfig/iptables に記述します。以下のように、Web サーバーにアクセスする HTTP 用にポート 80 番を、DNS サーバーに対する問い合わせ用にポート 53 番のルールを追加します。DNS サーバーに対する問い合わせは UDP が使用されるので、TCP と UDP の両方を設定する必要があります。

```
# vi /etc/sysconfig/iptables
```

```
# Firewall configuration written by system-config-firewall
# Manual customization of this file is not recommended.
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 80 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 53 -j ACCEPT
-A INPUT -m state --state NEW -m udp -p udp --dport 53 -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT
```

設定が完了したら、iptables サービスを再起動して新しい設定を適用します。

```
# service iptables restart
iptables: ファイアウォールルールを消去中:                  [  OK  ]
iptables: チェインをポリシー ACCEPT へ設定中 filter        [  OK  ]
iptables: モジュールを取り外し中:                          [  OK  ]
iptables: ファイアウォールルールを適用中:                  [  OK  ]
# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere            state RELATED,ESTABLISHED
ACCEPT     icmp --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere
ACCEPT     tcp  --  anywhere             anywhere            state NEW tcp dpt:ssh
ACCEPT     tcp  --  anywhere             anywhere            state NEW tcp dpt:http
ACCEPT     tcp  --  anywhere             anywhere            state NEW tcp dpt:domain
ACCEPT     udp  --  anywhere             anywhere            state NEW udp dpt:domain
REJECT     all  --  anywhere             anywhere            reject-with icmp-host-proh

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
REJECT     all  --  anywhere             anywhere            reject-with icmp-host-proh

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

再度、他の受講者の Web サーバーにアクセスして、接続できるようになったことを確認します。

# Linux サーバー構築標準教科書

2012 年 6 月 1 日 v2.0.0 版発行
2012 年 6 月 20 日 v2.0.1 版発行

発行所　LPI-Japan

Linux
Professional
Institute

www.lpi.or.jp