OCNユーザー必見

FreeBSDで最強のインターネットサーバーを作ろう!!

井上尚司
◆
菊地宏明

なぜ、今、PC-UNIXなのか？その答えは簡単だ。OCNの登場で個人でも常時接続のネットワーク環境を持てる時代になったからだ。借り物のIPアドレスではなく常に固定のIPアドレスを持てること。プロバイダーが提供する「~user」ではなく自分のドメインにホームページを置けること。これらはすべて夢ではなくなった。そうなると次はDNSやメールサーバーを自分で運用するという課題が生まれる。ここに高性能で無料で古いマシンでも動作するサーバーOSがあるとしたらどうだろう。それがFreeBSDをはじめとするPC-UNIXなのだ。すでにOCNに加入している人、これから加入しようと考えている人、この記事をきっかけに、ぜひPC-UNIXという選択肢を手に入れてほしい。

FreeBSD vs ウィンドウズNTサーバー

OSは無料で高性能、管理者への道も遠くない

OCNエコノミーの普及でDNSやメールサーバーを個人が運用する機会が多くなった。そして最近、これらのサーバーのOSとして「FreeBSD」や「Linux」などのPCで動くUNIXがにわかに注目を集めている。ここでは、今回の記事で紹介するFreeBSDともう1つの有名なサーバーOS「ウィンドウズNTサーバー」とを比較して、なぜ今PC-UNIXなのかを考えてみたい。

最大の魅力はフリーソフトウェアであること

OCNエコノミーに加入する際の初期費用を考えてみよう。工事費、ドメイン登録料、その月の利用料、ルーター代などを合わせるとおよそ12万円ほどの出費となる。これに、DNSサーバーソフトウェアとメールサーバーソフトウェア、そしてこれらを動かすOSの費用が加わるわけだ。

ウィンドウズNTサーバーを使った場合、OSとサーバーソフトウェアを合わせて12万円から20万円といったところだ（表1）。OCNの初期費用との合計はおよそ30万円になる。

これに対してFreeBSDはフリーソフトウェアだからOSは無料。専用のFTPサイトからダウンロードできるほか、市販のCDパッケージも4,000円から6,000円程度で購入できる。さらに、各種サーバーソフトウェアもすべて無料。FreeBSDなら初期費用プラス1万円以下ですべてがそろってしまうのだ。

眠れるマシンがよみがえる

サーバーソフトウェアを動かすためのPCはどうだろうか。

マイクロソフト社のウェブサイトには、ウィンドウズNTサーバーの動作環境が486／25MHz以上のCPUと16Mバイト以上のメモリーと記述されているが、実際に運用するためには少なくともペンティアム166MHzのCPUに64Mバイトのメモリーは必要だろう。

これに対してFreeBSDの動作環境は驚くほど低スペックであることが分かる（表2）。今回の記事のために編集部で立ち上げたサーバーマシンは、486DX2-66のCPUに16Mバイトのメモリーという古い仕様にもかかわらず、すべてのサーバーが同時に起動しても実に快適にかつ安定して動いた。

もし、ウィンドウズ95が満足に動かないような古いPCが押し入れに眠っていたら、この機会にもう一度電源を入れてみてほしい。FreeBSDでサーバーマシンとしてよみがえる可能性は本当に高いのだ。

信頼と実績はお墨付き

もともとインターネットの技術やプロトコルの多くはUNIXのプラットフォーム上で育ってきた経緯がある。加えて、長い年月の間多くの開発者やユーザーの手によってバグフィックスが施されてきた。実際に、あのYAHOO!でもWWWサーバーの一部にFreeBSDを使っているという。

今回は、実験のため、まったく同じスペックのPCを2台用意して一方にウィンドウズNTサーバー4.0とIIS3.0をインストールし、もう一方にFreeBSD2.2.5とApache1.2.5をインストールした。この2つのサーバーに対して、クライアント側からHTTPプロトコルを使って100個のファイルを読みこむプログラムを同時に10プロセス走らせてみた。これを10回行って全プロセスにかかる時間の平均を出したのが表3だ。結果は、NTサーバー4.0のほうが約1秒ほど速かったが、10万円と無料の差が誤差の範囲ともいえるわずか1秒。性能に関しても十分に信頼のおけるOSであることがお分かりだろう。

UNIXはむずかしくない？

UNIXといえばむずかしいものの代名詞のようで抵抗がある人も多いはずだ。はっきりいって初めてUNIXに触るという人にとっては、FreeBSDでのサーバー設定は楽じゃない。ウィザードが対話的に設定を進めてくれるウィンドウズと比べると確かに「むずかしそう」に見える。

でもちょっと待ってほしい。初めてウィンドウズやマッキントッシュに触ったときのことを思い出してほしい。おそらく何がなんだかさっぱりわからなかったはずだ。そう、UNIXは決してむずかしくはない。ただ、慣れていないだけなのだ。いくつかのコマンドを覚えたとたんにこのOSがいかに使いやすいものかが分かるはずだ。

さらに、サーバーソフトウェアの設定ファイルを自分で編集するという作業は、面倒な反面ネットワークの仕組みが手に取るように分かってくるというメリットがある。実際に設定しながらこの記事を読んでいくうちに、今よりもインターネットが身近に感じるようになるはずだ。

高性能なOSやサーバーソフトウェアが無料で使える。その上、ネットワークが本当に理解できる。これだけでもFreeBSDにチャレンジしてみる価値は大いにあるはずだ。

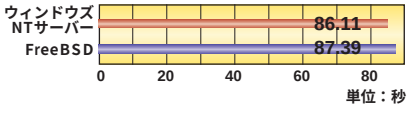
■サーバーソフトウェアの価格　（表1）

ウィンドウズNTサーバーの場合	
OS	約10万円
BIND 4.9.5 for NT(DNS)	無料
Sendmail with POP3(メール)	89,800円
合計	約20万円
FreeBSDの場合	
OS	無料 (CDは4〜6,000円)
各種サーバーソフトウェア	無料
合計	0円〜6,000円

■動作環境　（表2）

ウィンドウズNTサーバー	
CPU	486/25MHz以上
メモリー	16MB以上
ハードディスク	175MB以上
FreeBSD	
CPU	386SX以上
メモリー	5MB以上
ハードディスク	60MB以上

■性能比較　（表3）



サーバーソフトウェアを動かす準備をしよう

これさえ乗り越えればあとはワンダーランド

さて、さっそくサーバーの設定といいたいところだが、その前にFreeBSDで各種のサーバーソフトウェアを動かすための準備をしておこう。FreeBSDのインストールに関してはページ数の関係上ここでは詳細について触れない。その代わりに有益な情報を多く盛り込んだ。困ったとき、何かの設定でつまずいたとき、ここで紹介した情報が必ず役に立つはずだ。

FreeBSD 2.2.5-RELEASE収録先：
ナビゲーターCD【B】
標準的なインストールの手順は410ページ参照

インストールには運不運がある

FreeBSDのインストールがスムーズにいくかどうかは、自分の持っているPCがどんなハードウェア構成になっているかで決まる。インストール時に使用するブートフロッピーで設定されているハードウェアのI/OアドレスやIRQ番号と、自分のマシンのそれとがぴったり一致した場合はなんの問題もなくインストールできる。しかし、一致しない場合は自分のPCのハードウェアを調べて、ブート時にI/OアドレスとIRQ番号を変更しなくてはならない。

特に問題となるのは、CD-ROMドライブとイーサネットカードが認識できない場合だ。CD-ROMが使えなくてはインストールができないし、ネットワークに接続するためのイーサネットカードを認識しなければサーバーとしての機能を果たせないからだ。使用しているハードウェアがFreeBSD2.2.5でサポートされていないという最悪のケースもありえる。

本誌CD-ROMに収録のFreeBSD 2.2.5のインストールの仕方については、410ページで解説しているので、とりあえず手順どおりに進めてみてほしい。これでうまくいった幸運な人は次のページに進もう。うまくいかなかった人は以下に紹介するいずれかの方法を試してみてほしい。

「FreeBSD徹底入門」を入手しよう

FreeBSDのインストールには、これだけで1冊の本ができてしまうほど多くのバリエーションとTIPSが存在する。CD-ROMが使えない場合の対処方法やFTPサイトからネットワーク経由でインストールする方法など、とてもこのページだけでは解説しきれない。

そこで、数あるFreeBSDのインストール解説本の中から編集部おすすめの1冊を紹介しよう。株式会社翔泳社から出版されている「FreeBSD徹底入門」を入手してほしい。FreeBSD 2.2.1のCD-ROMと日本語インストーラーに対応したブートフロッピー付きで3,600円はお徳だ。「単なるインストール本ではない」のキャッチフレーズどおり、これだけでFreeBSDの基本は一通りマスターできるはずだ。

FAQやメーリングリストを利用しよう

「FreeBSD徹底入門」を読んでもまだ解決しない問題がある。そんな場合は、「FreeBSD友の会」のホームページにアクセスしてみよう。ここにはFreeBSDに関するさまざまなテーマについてのメーリングリストや、これまでに寄せられた質問をまとめたFAQなどが用意されている。

特に貴重な情報としては、「FreeBSDが標準でサポートしていないハードウェアを使えるように設定できた」などのユーザーからの投稿だ。たいていのハードウェア名はどこかに掲載されているので検索を駆使して探してみよう。

なお、メーリングリストに入会して質問を送る際には必ずFAQを見て過去に誰かが同じ質問をしていないかをチェックしよう。

FreeBSD友の会ホームページ
URL http://www.jp.freebsd.org/

最新のFreeBSDを入手しよう

2月5日現在のFreeBSD正式版の最新バージョンは本誌CDに収録の「2.2.5-RELEASE」だが、今後も頻繁にバージョンアップしていくはずだ。そして、バージョンが上がるたびにサポートされるハードウェアも増えていく。現バージョンでのインストールがうまくいかなくても、次のバージョンでは問題なくインストールできることがあるのだ。

最新のFreeBSDを入手する方法は2種類。1つは下記のFTPサイトなどからネットワーク経由でダウンロードする方法。もう1つは、最新のCD-ROMを購入する方法だ。

最新のCD-ROMは米国WalnutCreek社のものがhttp://www.cdrom.com/からオンラインで購入できるほか、日本ではこれに加えてパシフィック・ハイテック株式会社製のものが下記のショップなどで発売されている。また、2月21日にLASER5出版局から日本語環境がインストール後すぐに使える「FreeBSD 2.2.5 COMPLETE PACKAGE」が発売されたばかりだ（プレゼントコーナー参照）。

●FreeBSDの入手先
URL ftp://ftp.freebsd.org/pub/FreeBSD/

●秋葉原にあるおもなFreeBSD販売店

■ぷらっとホーム（通信販売あり）
地図：http://www.impress.co.jp/akibamap/map/map_00.html ブロック【B3】
URL http://www.plathome.co.jp/

■LASER5秋葉原店（通信販売あり）
地図：http://www.cdrom.co.jp/shop/
URL http://www.cdrom.co.jp/

FreeBSD徹底入門
（上）PC-AT互換機版　（下）PC98版
発行：株式会社翔泳社
価格：各3,600円（＋税）

portsを使ってソフトウェアをインストールしよう

FreeBSDのソフトウェアのインストールには何種類かの方法がある。ここでは、手軽であることから「ports」というインストールを簡易化するパッケージを使った方法を紹介する。

まず、rootでログインする。FreeBSDのセットアップ時にportsをインストールしていない場合は、

```
# /stand/sysinstall
```

を実行して「Configure」→「Distributions」→「Custom」→「ports」を選び、portsのインストールを済ませておこう。

【/cdromをマウントする】

portsを使ったインストールは、通常「/usr/ports/distfiles」に「tarball」（ソースコードを集めて1つのファイルにしたもの）を入れておき、それぞれのportsディレクトリー内で

```
# make install
```

を実行する。この際に、「/usr/ports/distfiles」にtarballがなければ「/cdrom/ports/distfiles」（CD-ROM内）を探し、ここにもなければ自動的にFTPサーバーからダウンロードする。

なお、今回の記事で紹介するサーバーソフトウェアのインストールに必要なtarballは、「qpopper」を除いてすべてCD-ROMの「/ports/distfiles」に収録してある。これを使うために、CD-ROMを「/cdrom」としてマウントしておこう。次のコマンドを実行する。

```
# mount /cdrom
```

個別のインストール方法は各章に記述してあるのでそれを参考にしてほしい。

【最新のportsを入手しよう】

この記事で紹介したサーバーソフトウェアのソースコードがアップデートされ場合、これをインストールするためのportsも新しくする必要がある。最新のportsは

```
ftp://ftp.jp.freebsd.org/pub/FreeBSD/ports-current/
```

以下からダウンロードできる。インターネットに接続していれば、portsの入手後に先に解説した手順で「# make install」コマンドを実行するだけで、適切なFTPサーバーから最新のソースコードが自動的にダウンロードされる。

今回の記事では、WWWサーバーソフト「apache」とPOP3サーバー「qpopper」はともに最新のバージョンを使用するため、CD-ROMにportsの最新バージョンを収録した。これを既存のものと置き換えておこう。

CD-ROMを「/cdrom」としてマウントしたうえで、リストAのコマンドを実行する。

リストA●

```
# cp  /cdrom/ports-current/apache/  usr/ports/www/apache

# cp  /cdrom/ports-current/popper/  usr/ports/mail/popper
```

「カーネル」を作り直そう

FreeBSDではオペレーティングシステム（OS）の中核となる部分を「カーネル」と呼ぶ。

システムに変更があった際には、管理者はカーネルを変更してコンパイルし、新しいカーネルに作り変えることができる。

今回の記事で紹介するサーバーソフトウェアの中には、カーネルを作り直しておかなければ使えないものがある。ここでこの記事に必要な設定をすべて済ませておこう。まず、rootでログインして作業をする。

【パッチを当てる】

今回の記事ではマック用のファイルサーバー「netatalk」を使うため、カーネルにパッチを当てる必要がある。

CD-ROMを「/cdrom」にマウントして次のコマンドを実行する。

```
# cd /usr/src/sys
# patch <  /cdrom/updates/atalk.diff.2.2
```

【カーネルの再コンパイルをする】

カーネルの再コンパイルには、まずカーネルの設定ファイルの作成から始める。カーネルの設定ファイルは「/usr/src/sys/i386/ conf」ディレクトリーにあるので、ここに移動しよう。

```
# cd /usr/src/sys/i386/conf
```

汎用の設定ファイル「GENERIC」を適当なファイル名でコピーする。通常は自分のホスト名（ここではKURAZONO）などを使う。

```
# cp GENERIC  KURAZONO
```

ファイルの編集は「vi」や「ee」コマンドで行うが、viの使い方が分からない場合は「ee」を使おう。次のコマンドを実行する。

```
# ee KURAZONO（ファイル名）
```

この設定ファイルには、各種デバイスの設定やカーネルオプションの選択が書かれている。これを編集して適切な設定にする。まず、カーネルの識別子が書かれた行（ident）を探し、新しく付けたファイル名に変更しておく。

```
ident       KURAZONO
```

今回の記事では、netatalkとDHCPを使う際に、下に示す行を追加する必要がある。

（netatalkでは、次の1行を追加）

```
options         NETATALK
```

（DHCPでは、次の1行を追加）

```
pseudo-device bpfilter      4
```

この2行を追加したらファイルを保存する（eeなら「Esc」キーを押して「leave editor」→「save changes」）。

次に、設定ファイルに対応したコンパイル処理を制御する「Makefile」を作成するために、次のコマンドを実行する。

```
# config KURAZONO
```

Makefileが作成されるので、コンパイル用ディレクトリーに移ってコンパイルを行う。

```
# cd ../../compile/KURAZONO
# make depend ; make
```

カーネルが作られたら、新しいカーネルと今までのカーネルとを入れ替える。次のコマンドを実行する。

```
# make install
```

以上でカーネルの作り直しは完了だ。次のコマンドを実行してFreeBSDを再起動しよう。

```
# reboot
```

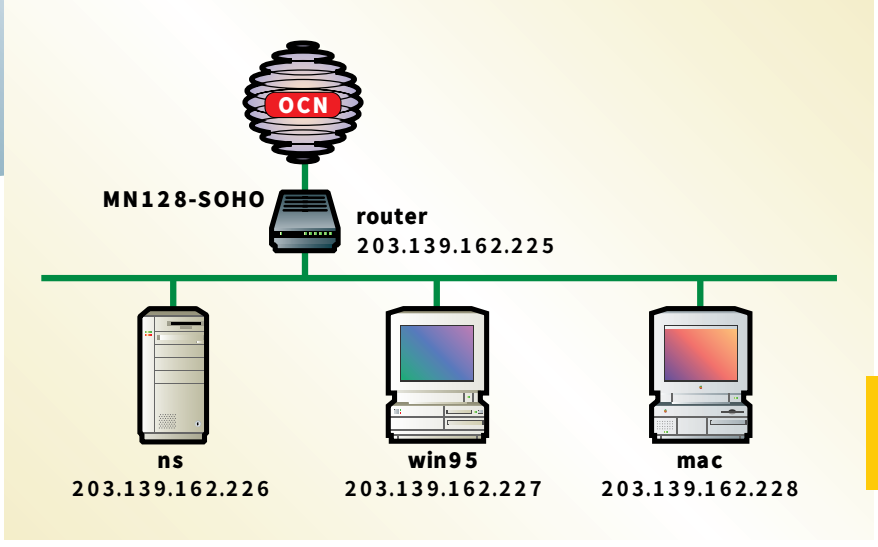
古いカーネルは「/」ディレクトリーに「kernel.old」というファイル名で保存される。再起動の際に新しいカーネルで動作しなくなった場合には、こちらを指定して起動すればよい。システムのブート時に「Boot:」が表示されたら次のコマンドを実行する。

```
Boot: /kernel.old
```

ネットワークプランを立てよう!!

ルーターとホストの設定

OCNエコノミーを申し込んでしばらくすると、NTTから「契約内容のご案内」が送られてくる。そこには、NTTから契約者に割り当てられたIPアドレスやセカンダリーDNSサーバーの情報が書かれている。
最初にしなければならないのは、それらの情報をもとに自分のネットワークの構成を考え、IPアドレスなどの割り当てを行うことだ。



＊MN128-SOHO/DSU
発売元：㈱エヌ・ティ・ティ・テレコムエンジニアリング東京
製造元：㈱ビー・ユー・ジー
価　格：69,800円

自分だけのネットワーク設計図を作る

最初の作業はネットワークの設計をすること。つまり、OCNに接続したあとの「できあがり」の状態を描くことだ。ここでは上の図で示すように、論理的な1つのイーサネット上にFreeBSDのサーバーホスト、ウィンドウズ95のホスト、マッキントッシュのホストがあり、さらにルーターとしてNTT-TE東京のダイアルアップルーター「MN128-SOHO」を使ってOCNへ接続する、という構成を目指すことにする。そして、FreeBSDホストでDNSやメールなどのサーバーソフトウェアを稼動させることにする。

3台のホストとルーター以外に必要な機材は、3台のホストに取り付けるネットワークカードと10Base-Tのイーサネットケーブルぐらいなものだ。ハブに関してはMN128-SOHOは3ポート分のイーサネットハブを内蔵しているので、3台までなら別に用意する必要はない。

ネットワークの設定でもっとも重要なのはIPアドレスである。契約後にNTTから送られてくる「契約内容のご案内」には、割り当てIPアドレスとして「ネットワークアドレス」と「ネットマスク」が指定されている。たとえば、ネットワークアドレスとして「203.139.162.224」、「サブネットマスク」として「255.255.255.240」というように指定されている。

DNSサーバーを自分で用意する場合、サブネットマスクは通常上記に示す「255.255.255.240」が割り当てられる。これは28ビットのサブネットマスクであり、最後の4ビットの部分の「1」・「0」の組み合わせによって、合わせて全部で16個のIPアドレスを生み出すことができるわけだ。上記の例では、最後の4ビットがすべて「0」のときの「203.139.162.224」から、4ビットがすべて「1」のときの「201.139.162.239」までの16個が、利用可能なIPアドレスとなる。

表1.

	IPアドレス	対　象	ホスト名	役　割
1	203.139.162.224	ネットワークアドレス		
2	203.139.162.225	MN128-SOHO ルーター	router	デフォルトゲートウェイとなる
3	203.139.162.226	FreeBSD PC-UNIXサーバー	ns	DNSやメール等のサーバーとなる
4	203.139.162.227	ウィンドウズ95	win95	
5	203.139.162.228	マッキントッシュ	mac	
6	203.139.162.229			
⋮	≀			
15	203.139.162.238			
16	203.139.162.239	ブロードキャストアドレス		

＊上記のIPアドレスは、OCNの「DNSサーバ設定方法」で使用されているものを流用した。実際にはNTTから送付される「契約内容のご案内」を参照して自分に割り当てられたものに変更する。

利用可能なIPアドレスが分かったら、ホストへの割り当てをしよう。ここでは表1に示すように3台のホストにIPアドレスを割り当て、それぞれns、win95、macという名前を付けることにする。

なお、DNSサーバーとなるホストの名前は、OCN申し込み時に記入した名前を用いる必要がある。またこのホストのIPアドレスは、NTTからの「ご案内」に記された「プライマリDNS IPアドレス」を割り当てなければならない。

以降の例では、申請して割り当ててもらったドメイン名を「kurazono.co.jp」と仮定して話を進める。

IPアドレスを割り当てる

ネットワークの設計が終わったら、実際に各ホストの設定作業に移る。ここでは、ルーター、FreeBSD、ウインドウズ95、マッキントッシュの設定作業をそれぞれ見ていこう。なお、それぞれのクライアントマシンにはTCP/IPプロトコルなどの必要なソフトウェアはすでにインストールされていることが前提となる。また、DNSサーバーの指定では1つのDNSサーバーしか指定していないが、OCNから知らされたセカンダリーサーバーのアドレスも指定するほうがよりよい。

＜ウィンドウズ95＞

ウィンドウズ95ホストで必要な作業は、IPアドレス、デフォルトゲートウェイ、DNSの3つを設定することだ。いずれも、「コントロールパネル」の「ネットワーク」の中の「TCP/IPのプロパティ」で行う。

「IPアドレス」タブでは、IPアドレスを自分で指定するほうを選ぶ。そしてたとえば今回の例では、IPアドレスに「201.139.162.227」を、サブネットマスクに「255.255.255.240」を入れる。

「ゲートウェイ」タブでは、ルーターのアドレスを登録する。今回の例では、MN128-SOHOのIPアドレスである「201.139.162.225」を入れる。

最後は「DNS設定」タブだ。「DNSを使う」を選び、ホスト名、ドメイン名、DNSサーバーなどを指定する。以上を設定して再起動すれば、完了だ。

＜MN128-SOHOの設定＞

MN128-SOHOは初期状態でDHCP機能がオンになっているので、この状態でWWWブラウザーから設定するのが手っ取り早い。

すべての作業は「詳細設定」で行う。まず「ISDN設定」で回線種別を専用線128Kにし、「接続/電話登録」の「#0」で接続モードをLAN型接続にする。

次に「IP設定」に行き、オプションの欄へのデフォルトルートの書き込みと、オンになっているDHCPサーバー機能などをオフにする。最後にIPアドレスとサブネットマスク長を書き込む。今回の例では「203.139.162.225/28」となる。

ルーターの設定が終わったら、電源を入れ直しておく。MN128-SOHOをOCNエコノミーで利用する場合の詳しい説明が設定時のオンラインヘルプに載っているので、参考にするとよい。

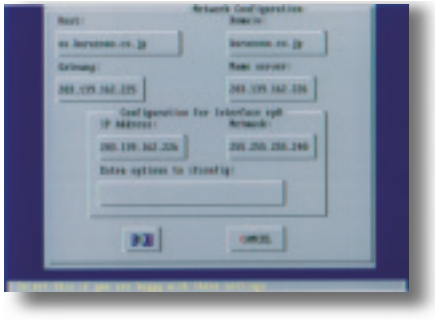
＜マッキントッシュ＞

ここではMacOS 8を例にとって説明する。必要な作業はウィンドウズ95と同様にIPアドレス、ゲートウェイ、DNSの設定である。

「コントロールパネル」の「TCP/IP」で表示されたウィンドウで、「経由先」としてEthernetを選び、さらに「設定方法」として手入力を選ぶ。

IPアドレス、サブネットマスク、ルーターアドレス、ネームサーバーアドレスをそれぞれ入力する。

さて、コントロールパネルにあるのが「TCP/IP」ではなく、「MacTCP」だという方もいるはずだ。表示されるウィンドウは異なるが、行うべき設定はまったく同じだ。アドレス取得を「マニュアルで」にして、IPアドレス、サブネットマスク、ゲートウェイアドレス、ネームサーバー情報を入力すればよい。



＜FreeBSD＞

FreeBSDでのネットワークの設定は、インストール時に行える。また、インストール時に行わなくても、インストール後「/stand/sysinstall」コマンドで行うこともできる。

sysinstallを起動した場合、「Configure」、「Networking」、「Interfaces」、「No」と進む（それぞれ項目を選択して「スペース」キーを押す）と、ネットワークインターフェイスデバイスの選択画面となる。この画面は、FreeBSDのインストール時のネットワーク設定と同様の画面である。この中から、イーサネットインターフェイスのデバイスを選ぶと、そのインターフェイスの設定画面となる。

ここで設定するのは、ホスト名、ドメイン名、ゲートウェイ、ネームサーバー、IPアドレス、ネットマスクなどである。

ホスト名は、ドメイン名を付けた完全な形のものを指定する。設定はGUIベースなので、難なく行えるはずだ。

DNSサーバーを作ろう

IPアドレスとドメイン名のマッピング

さて、ここから本格的なネットワークの設定が始まる。UNIXのコマンドを使って設定ファイルを編集していくわけだ。最初は暗闇の中を明かりもつけずに歩いているような不安を感じるかもしれない。でも、ここでの設定さえ終われば、NTTに「DNSの設定が完了しました」と電話をして、いよいよネットワークが開通する。お楽しみはこれからだ。

DNSサーバーの役割

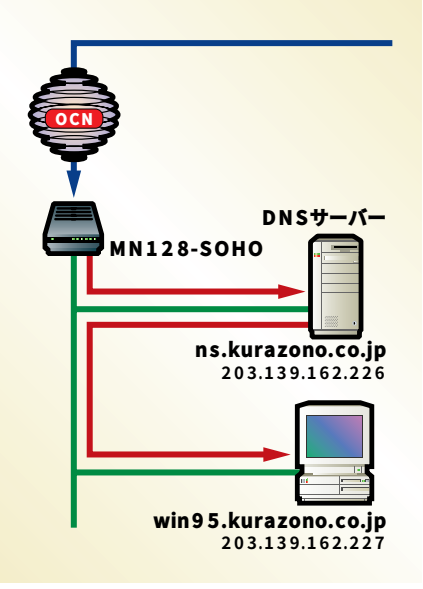
OCNエコノミーで独自ドメインを利用する場合、まず最初に用意しなければいけないのがDNSサーバーだ。DNSは「ドメイン・ネーム・システム」の略。

インターネットに接続するホストには、世界中で一意のIPアドレスを割り当てる。このIPアドレスをもとにデータのやり取りを行っているわけだ。しかし、IPアドレスは数字の並びのため覚えにくく、間違いやすいので、文字による表記のドメイン名が用いられている。たとえば、インプレスのウェブサーバーのIPアドレスは192.218.90.1であるが、ほとんどの人はwww.impress.co.jpしか気にしなくていいはずだ。

この、人間にとって分かりやすいドメイン名からネットワーク上で必須なIPアドレスを検索したり、反対にIPアドレスからドメイン名を検索したりといった作業を行ってくれるのがDNSサーバーである（これをマッピングと呼ぶ）。マッピングには、ドメイン名からIPアドレスを得る正引きと、IPアドレスからドメイン名を得る逆引きがある。DNSサーバーは、自分が管理すべき範囲（これをゾーンと呼ぶ）のすべてのホストの情報を保持し、クライアントからの要求に回答する。

また、DNSサーバーは、単にマッピングをするだけでなく、メールの配送先の指定などにも用いられる。

FreeBSDには、BINDと呼ばれるDNSサーバーが初めから用意されている。ここではBINDの設定を見ていくことにする。



FreeBSDでの設定

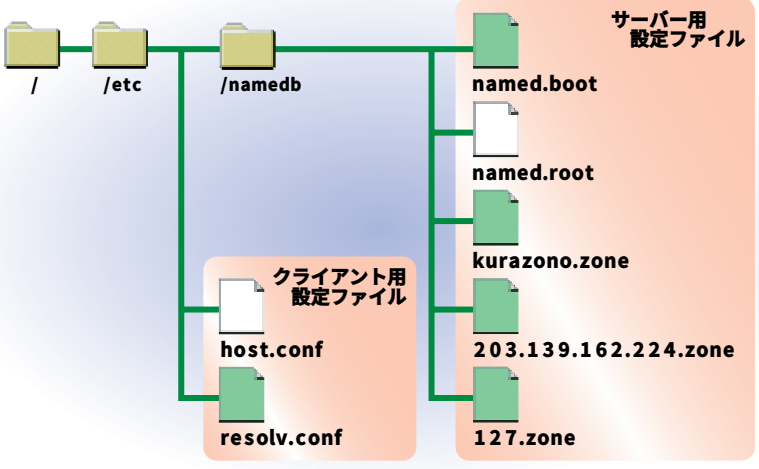
DNSの設定・運用に関係するファイルは、右図に示したとおりである。DNSサーバーとなるホストは、同時にDNSクライアントでもある。そのため、サーバーとして必要なファイルだけでなく、クライアントとして必要なファイルもある。

クライアント用のファイルとしては、まず「/etc/host.conf」がある。これは、ドメイン名とIPアドレスのマッピング方法と、その適応順序を記述するファイルである。もう1つのクライアント用のファイルに「/etc/resolv.conf」がある。これは、問い合わせるDNSサーバーの指定を行うファイルだ。

サーバー用のファイルのうち、設定ファイルはすべて「/etc/namedb」というディレクトリーに置く。このディレクトリーにはあらかじめ「named.boot」のサンプルと、「named.root」が入っている。まず基本となるのが「named.boot」だ。これはDNSサーバーが起動するときに参照するファイルである。次に「named.root」というファイルがある。これはルートネームサーバーと呼ばれる世界中のおおもととなるサーバーを示したファイルだ。

この2つのファイルのほかに管理するゾーンの情報を記述したファイルをいくつか用意する必要がある。今回の例のような比較的単純なネットワーク構成の場合、3つのファイルを用意することになる。それらは、正引きマッピングのためのファイル、逆引きマッピングのためのファイル、ループバックアドレスの逆引きのためのファイルである。

設定ファイル以外に必要なファイルとしては、「/etc/rc.conf」と「/etc/rc.network」ファイルがある。これらは、システムの起動時に利用されるファイルで、ネームサーバーへのフラグの指定や、実際の起動を行っている。



＊緑色のファイルは編集が必要

リスト1●「named.boot」

	1項	2項	3項
1行	directory	/etc/namedb	
2行	cache	.	named.root
3行	primary	kurazono.co.jp	kurazono.zone
4行	primary	224.162.139.203.in-addr.arpa	203.139.162.224.zone
5行	primary	0.0.127.in-addr.arpa	127.zone

named.boot ファイルの編集

DNSサーバーの設定として、まず最初に行う作業は、named.bootファイルの設定である。このファイルは、DNSサーバーの動作を指定する初期設定ファイルで、起動時に読み込まれる。FreeBSDをインストールすると、「/etc/namedb」ディレクトリーの中にDNSサーバー関係のファイルが用意されている。named.bootファイルのサンプルも、ここに入っている。

とりあえずは、リスト1の「named.boot」を見てほしい。用意されているサンプルには、説明のために多くのコメントが入っているが、ここではそれらを省いて必要最小限の5行のみを示している。設定ファイルでは、それぞれの行がDNSサーバーに対する指示を表している。以下に順番に見ていく。

1行目では、これ以降で指定されているファイルがどのディレクトリーに入っているかを示している。初期設定のまま、「/etc/namedb」を指定しておけばよい。

2行目は、キャッシュの指定である。2項目の「.」を忘れてはいけない。3項目の「named.root」はファイル名で、この中に情報が入っていることを示している。このファイルは、すでに用意されているものを用いればよい。ちなみに、中身はリスト2の「named.root」のようになっている。これは、世界中のルートネームサーバーをリストしたもので、全部で13個のサーバーが示されている。この内容は時々変わることがあるので、常に新しいものを用意する必要がある。ファイルの入手は、「ftp://ftp.rs.internic.net/domain/named.root」から行う。

3行目、4行目、5行目が、このDNSサーバーが管理を行うゾーンの指定と、そのゾーンの情報が入っているファイルの指定だ。最初の項目である「primary」は、このサーバーがそのゾーンに対するプライマリー（主、あるいは一次）サーバーとなることを意味している。第2項目の管理するゾーンの指定では、ドメイン名を入れる。

3行目の例では、kurazono.co.jpを指定している。

第3項目がファイル名である。この3行目は、ドメイン名からIPアドレスへのマッピング、すなわち正引き用のものである。

4行目と5行目のドメイン名表記は、見慣れた形式のものとは異なる。これは、逆引き用のゾーン指定の表記を表している。

4行目は、ネットワークアドレス（236ページの表参照）の前後を逆にして、その後ろに「in-addr.arpa」を付けて記述する。

インターネットでは、ループバックネットワークとローカルホスト（localhost）が定義されている。ループバックネットワークのネットワークアドレスは127.0.0で、ローカルホストのIPアドレスは127.0.0.1である。インターネットのどのホスト上でも、127.0.0.1は常にそのホスト自身を意味している。これらのアドレスの逆引き用ゾーン指定が、5行目ということになる。ネットワークアドレスである127.0.0の前後を入れ替えて0.0.127とし、その後ろにin-addr.arpaを付ける。

コマンドワンポイントレッスン 1

「named.bootの編集」

1）rootでログイン
2）# cd　/etc/namedb ‥「namedb」ディレクトリーに移動
3）# ls ‥‥‥‥‥‥‥‥‥「namedb」ディレクトリーの中身を表示
4）# ee　named.boot ‥‥「named.boot」ファイルを編集

リスト2●「named.root」

```
;       This file holds the information on root name servers needed to
;       initialize cache of Internet domain name servers
;       (e.g. reference this file in the "cache  .  <file>"
;       configuration file of BIND domain name servers).
;
;       This file is made available by InterNIC registration services
;       under anonymous FTP as
;           file                /domain/named.root
;           on server           FTP.RS.INTERNIC.NET
;       -OR- under Gopher at    RS.INTERNIC.NET
;           under menu          InterNIC Registration Services (NSI)
;              submenu          InterNIC Registration Archives
;           file                named.root
;
;       last update:    Aug 22, 1997
;       related version of root zone:   1997082200
;
;
; formerly NS.INTERNIC.NET
;
.                        3600000  IN  NS    A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET.      3600000      A     198.41.0.4
;
; formerly NS1.ISI.EDU
;
.                        3600000      NS    B.ROOT-SERVERS.NET.
B.ROOT-SERVERS.NET.      3600000      A     128.9.0.107
;
; formerly C.PSI.NET
;
.                        3600000      NS    C.ROOT-SERVERS.NET.
C.ROOT-SERVERS.NET.      3600000      A     192.33.4.12
;
; formerly TERP.UMD.EDU
;
.                        3600000      NS    D.ROOT-SERVERS.NET.
D.ROOT-SERVERS.NET.      3600000      A     128.8.10.90
;
; formerly NS.NASA.GOV
;
.                        3600000      NS    E.ROOT-SERVERS.NET.
E.ROOT-SERVERS.NET.      3600000      A     192.203.230.10
;
; formerly NS.ISC.ORG
;
.                        3600000      NS    F.ROOT-SERVERS.NET.
F.ROOT-SERVERS.NET.      3600000      A     192.5.5.241
;
; formerly NS.NIC.DDN.MIL
;
.                        3600000      NS    G.ROOT-SERVERS.NET.
G.ROOT-SERVERS.NET.      3600000      A     192.112.36.4
;
; formerly AOS.ARL.ARMY.MIL
;
.                        3600000      NS    H.ROOT-SERVERS.NET.
H.ROOT-SERVERS.NET.      3600000      A     128.63.2.53
;
; formerly NIC.NORDU.NET
;
.                        3600000      NS    I.ROOT-SERVERS.NET.
I.ROOT-SERVERS.NET.      3600000      A     192.36.148.17
;
; temporarily housed at NSI (InterNIC)
;
.                        3600000      NS    J.ROOT-SERVERS.NET.
J.ROOT-SERVERS.NET.      3600000      A     198.41.0.10
;
; housed in LINX, operated by RIPE NCC
;
.                        3600000      NS    K.ROOT-SERVERS.NET.
K.ROOT-SERVERS.NET.      3600000      A     193.0.14.129
;
; temporarily housed at ISI (IANA)
;
.                        3600000      NS    L.ROOT-SERVERS.NET.
L.ROOT-SERVERS.NET.      3600000      A     198.32.64.12
;
; housed in Japan, operated by WIDE
;
.                        3600000      NS    M.ROOT-SERVERS.NET.
M.ROOT-SERVERS.NET.      3600000      A     202.12.27.33
; End of File
```

正引きゾーンファイルの編集

ゾーンファイルの説明に入る前に、重要な点を1つ確認しておこう。ゾーンファイルの記述では、ピリオド1つの有無によってまったく動作が異なってしまうので、注意が必要である。基本的に、ピリオドで終われば記述したままの形で取り扱われるが、最後にピリオドを付けないときは、その時点でのデフォルトのゾーン名が後ろに置かれることになる。

リスト3の「kurazono.zone」が、正引き用のkurazono.zoneゾーンファイルだ。このファイルは新規に作成する。次のコマンドを実行して設定を記述する。

```
# ee kurazono.zone
```

❶ホスト名を書く。最後はピリオドで終わらせる。
❷管理者のメールアドレスの「@」を「.」に変えたものを書く。最後はピリオドで終わらせる。
❸シリアル番号。DNSではこの番号をもとにデータの新旧を判断している。そのため、ゾーンファイルに変更を加えたら、必ずこの値を増加させなければならない。
❹NSレコードと呼ばれ、DNSサーバーの指定を意味する。最初の行は当然ながら今設定中のホスト名となる。この下の行はOCNが知らせてきたセカンダリーDNSサーバーを指定する。
❺アドレスを指定するAレコード。ここでは、DNSサーバーを稼動するこのホストのIPアドレスを指定している。これは、ホスト名を付けずにドメイン名だけを検索したときに返すIPアドレスとなる。この例では、kurazono.co.jpへの参照はns.kurazono.co.jpへの参照とみなすことになる。
❻MXレコードといい、メールの配送先の指定をする。ドメイン名にkurazono.co.jpが指定されたメールは、この指定に従って、ns.kurazono.co.jpに送られる。
❼ローカルホストのアドレス指定。ローカルホストに対する正引きは、ほとんどの場合ゾーンとしては独立させずに、このようにkurazono.co.jpゾーンの中で行う。
❽CNAMEレコード。別名の指定を意味している。ここではloghostという別名を用意し、それが実際にはlocalhostだということを指定している。

これ以降の行では、パターンがだいたい決まっている。まずホスト名に対してIPアドレスを付けるためのAレコードがあり、それに続いてMXレコード、HINFOレコード、CNAMEレコードが続くという形だ。ホスト名まで指定してメールを送ることがないときは、各ホストに対するMXレコードはなくてよい。

HINFOレコードでは、そのホストのハードウェアやソフトウェアの情報を記述する。ここではこの情報はコメントとしてのみ扱うように、行の先頭に「;」を入れている。

逆引き用ゾーンファイルの編集

次は逆引き用ゾーンファイルである。これも新規に作成する。リスト4の「203.139.162.224.zone」を見てほしい。ファイルの最初の部分は、正引き用と同様である。違うのは❾のところで、これはドメイン名ポインターを意味するPTRレコードである。この例では、ゾーン名である224.162.139.203.in-addr.arpaに対して、kurazono.co.jpへのポインタを作っていることになる。

❿は、習慣的にネットマスクを入れているが、これはなくても問題ない。あとは、各ホストのIPアドレスの最下位部とホスト名とのPTRレコードを記述するだけだ。これらのPTRレコードがあるおかげで、IPアドレスからドメイン名へのマッピングが定義されることになる。⓫はこのネットワーク全体を表すドメイン名。通常は「xxx-net.xxx.co.jp」の形式で記述する。

リスト3●「kurazono.zone」

```
@           IN    SOA   ❶ns.kurazono.co.jp. ❷root.ns.kurazono.co.jp. (
                        ❸199802011 ; Serial
                          3600       ; Refresh
                          300        ; Retry
                          3600000    ; Expire
                          86400 )    ; Minimum

            IN    NS    ❹ns.kurazono.co.jp.
            IN    NS      ns-tk011.ocn.ad.jp.

            IN    A     ❺203.139.162.226
            IN    MX      10          ❻ns.kurazono.co.jp.

localhost   IN    A     ❼127.0.0.1
loghost     IN    CNAME ❽localhost

router      IN    A       203.139.162.225
;           IN    HINFO   ISDN-ROUTER  MN128-SOHO

ns          IN    A       203.139.162.226
            IN    MX      10 ns.kurazono.co.jp.
;           IN    HINFO   INTEL        FreeBSD2.2.5
server      IN    CNAME   ns
www         IN    CNAME   ns
ftp         IN    CNAME   ns
mail        IN    CNAME   ns
mailhost    IN    CNAME   ns

win95       IN    A       203.139.162.227
            IN    MX      10 ns.kurazono.co.jp.
;           IN    HINFO   INTEL        Windows95

mac         IN    A       203.139.162.228
            IN    MX      10 ns.kurazono.co.jp.
;           IN    HINFO   MACINTOSH    MacOS8
```

リスト4●「203.139.162.2240.zone」

```
@           IN    SOA   ns.kurazono.co.jp. root.ns.kurazono.co.jp.  (
                        199802011       ; Serial
                        3600            ; Refresh
                        300             ; Retry
                        3600000         ; Expire
                        86400 )         ; Minimum

            IN    NS    ns.kurazono.co.jp.
            IN    NS    ns-tk011.ocn.ad.jp.

            IN    PTR   ❾kurazono.co.jp.
            IN    A     ❿255.255.255.240

224   IN    PTR   ⓫kurazono-net.kurazono.co.jp.
225   IN    PTR   router.kurazono.co.jp.
226   IN    PTR   ns.kurazono.co.jp.
227   IN    PTR   win95.kurazono.co.jp.
228   IN    PTR   mac.kurazono.co.jp.
```

ループバックの逆引き用ゾーンファイルの編集

最後は、ループバックの逆引き用ゾーンファイルである。これも新規に作成する。リスト5の「127.zone」を見てほしい。このゾーンのセカンダリーサーバーになるDNSはないので、ここではNSレコードは自分自身のみで問題ない。

PTRレコードも、必須なのは⑫のlocalhostのみである。例で示してある0のPTRレコード（⑬）は、ループバックネットワークに名前を付ける意味を持つ。

自動スタートの設定

すべての設定ファイルの準備ができたら、システムの立ち上げ時に自動的にDNSサーバーが起動するように、「/stand/sysinstall」コマンドを使って設定する必要がある。sysinstallを起動後「Configure」に進み、さらに「Startup」に進むと、「named」と「namedflags」というメニューが出る。namedをオンにし、namedflagsで「-b /etc/namedb/named.boot」と指定すると、/etc/rc.confファイルの中の変数に値が設定され、システムの立ち上げ時に/etc/rc.network内に書かれたnamedプログラムが起動するようになる。

DNSクライアントとしての設定

DNSクライアントとしての設定は、ファイル2つのみである。

まずはマッピングの方法とその順序を決める「/etc/host.conf」ファイルだ。リスト6の「host.conf」に示す内容のファイルがあらかじめ用意されているので、とくに行う作業はない。

もう1つのファイルは「/etc/resolv.conf」で、DNSサーバーの検索方法を指定するものだ。ネットワークの設定を行うと自動的にファイルが作成されるが、実際の運用ではさらに設定を追加する必要がある。

とりあえずリスト7の「resolv.conf」を見てほしい。最初の「domain」行はローカルドメインを定義している。ピリオドの入っていないホスト名が使われた場合、このドメイン内のホスト名と解釈される。

「nameserver」行は、問い合わせるDNSサーバーの指定である。最初のアドレスはこのホスト自身のIPアドレスであり、2番目のアドレスはOCNが知らせてきたセカンダリーDNSサーバーのIPアドレスである。DNSサーバーへのアクセスは、ここに書かれた順に行われる。なお、自分自身のIPアドレスのところは、ローカルホストを示す127.0.0.1を書いてもよい。

リスト5●「127.zone」

```
@           IN    SOA    ns.kurazono.co.jp. root.ns.kurazono.co.jp.  (
                         199802011      ; Serial
                         3600           ; Refresh
                         300            ; Retry
                         3600000        ; Expire
                         86400 )        ; Minimum

            IN    NS     ns.kurazono.co.jp.

0           IN    PTR    loopback-net.  ⑬
1           IN    PTR    localhost.     ⑫
```

リスト6●「host.conf」

```
# $Id: host.conf,v 1.2 1993/11/07 01:02:57 wollman Exp $
# Default is to use the nameserver first
bind
# If that doesn't work, then try the /etc/hosts file
hosts
# If you have YP/NIS configured, uncomment the next line
# nis
```

リスト7●「resolv.conf」

```
domain      kurazono.co.jp
nameserver  203.139.162.226
nameserver  203.139.160.69
```

動作確認をしてみよう

サーバーとクライアントの設定が終わったら、さっそく動作確認する。DNSの確認には、nslookupコマンドを用いるとよい。必要な確認項目は、以下のとおりである。

（1）自ドメイン内のホストのホスト名を入れて、IPアドレスが返されることを確認（たとえばwin95）。
（2）(1)のホスト名の代わりに完全なドメイン名まで入れてみる（win95.kurazono.co.jp）。
（3）自ドメイン内のホストのIPアドレスを入れて、ホスト名が返されることを確認。
（4）自ドメイン以外のホスト名を入れて、IPアドレスが返されることを確認。
（5）(4)のIPアドレスを入れて、ホスト名が返されることを確認。

ここで「set type=mx」と入力し、以降はMXレコードを検索するように指定する。

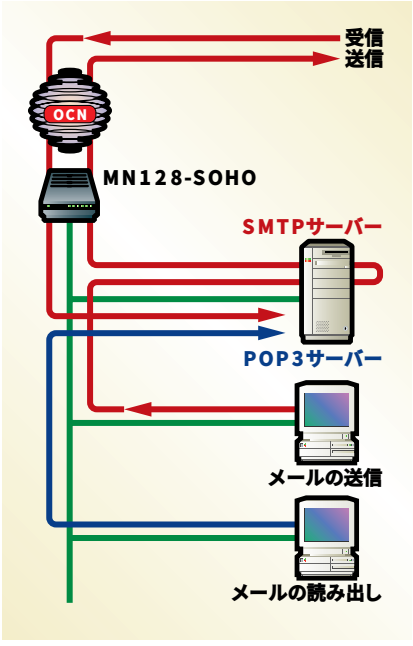
（6）自ドメインの名前を入れてみる（たとえばkurazono.co.jp.）。
（7）さらにホスト名も付けてみる（win95.kurazono.co.jp.）。
（8）自ドメイン以外のドメイン名を入れてみる（最後にピリオドを付ける）。

ここまで確認して問題なければ、一応正しく設定されている考えてよい。

メールサーバーを作ろう

自分のメールボックスを管理する

DNSの設定が終わったら、次はなんといっても電子メールだろう。プロバイダーや会社のメールボックスに送られてきたメッセージをクライアントソフトで読むというのは昨日までのこと。これからは自分のネットワークにメールサーバーが置かれることになる。もちろん、メールアドレスにも取得したドメイン名が使えるのだ。



メールサーバーの役割

電子メールサーバーは、2つの役割を複合して行っている。1つは、クライアントのコンピュータから差し出された電子メールを相手の電子メールサーバーに届ける役割だ。

もう1つは、インターネットから自分のネットワークに届いた電子メールを蓄積して保存し、クライアントが電子メールを読み込むまで管理する役割だ。クライアントは、各種クライアントソフトウェアを使って、この電子メールを取り出して読めるようになる。

クライアントがメールサーバーから電子メールを読み込む方法としては、POP3とIMAP4が有名だ。IMAP4のプロトコルを使うと、届いた電子メールのヘッダーだけを読みこんでダウンロードするメールを自由に選べるなど機能的にも優れているが、現時点ではIMAP4に対応する電子メールソフトはまだ少ないため、今回はPOP3に対応したサーバーを用意する。クライアントとなるPCには使い慣れた電子メールソフトを設定しておく。メールの送信はSMTPサーバー経由でインターネットに送り、インターネットから届いたメールは受け手のSMTPサーバーが蓄積する。クライアントはPOP3サーバーに接続して届いたメールを読む。

SMTPサーバーとPOP3サーバーは別のマシンに設定することも、同一のマシンに設定することもできる。

FreeBSDでの設定

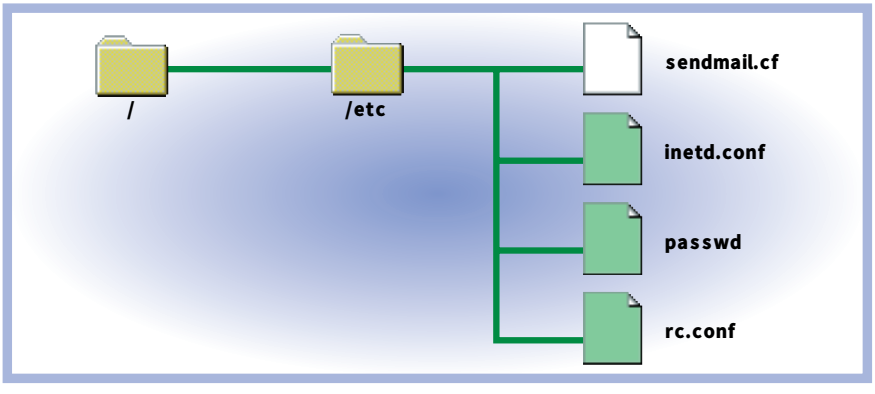
利用する電子メールサーバソフトは、SMTPが「sendmail」、POPが「qpopper」となる。FreeBSD 2.2.5には「sendmail 8.8.7」が組み込まれており、インストールなしに設定作業から行えばよい。

POPサーバーを利用するときには、利用者別にIDとパスワードを用意する。これは、FreeBSDのアカウントに該当するので、FreeBSDの利用者登録を行う。

```
#   /stand/sysinstall
```

このコマンドを実行して、「Configure」→「User Management」→「Add user」を選択していくと、ユーザー登録画面が表示される（図A）。ここでログインID、パスワード、フルネーム、メンバーグループ（省略可）、ホームディレクトリー、ログインシェルを設定する。このログインIDとパスワードがPOP3サーバーのIDとパスワードになる。

＊sendmailの最新版は8.8.8となっている。
入手先：ftp://ftp.iij.ad.jp/pub/network/sendmail/



このままの設定では、通常のユーザーと同様に、FreeBSDにログインすることも可能になる。メールの利用は許可するが、ログインを行わせないようにするには、「vipw」コマンドを使って「/etc/passwd」ファイルを編集すればよい。このファイルには登録利用者ごとにIDなどが「：」で区切って記載されている。該当するIDの最後の項目の部分をリストAのように「/nonexistent」にする。

A

リストA●「passwd」

```
kurazono:*:1001:1001:Keizo Kurazono:/:/nonexistent
```

sendmail.cfの作成

UNIX管理者でも避けて通りたいのが「/etc/sendmail.cf」ファイルの設定だ。SMTPサーバの動作時に参照する設定ファイルで、難解で面倒なことで定評がある。通常は、もう少し分かりやすい形式で設定を記述し、「CF」や「m4」といったマクロ言語処理を行って生成する。ただし、この方法も「perl」の知識などが必要なため、分かりやすくなるものの別の手間は避けて通れない。

しかし、FreeBSDでのsendmail.cfは従来の設定よりも便利になっている。かつてはドメイン名などの情報を書き込む必要はあったものの、ある程度はすでに設定されている情報を参照してくれるようになった。そのため、高度な設定を行わないのであれば、汎用的な設定ファイルを使うことができる。

そこで、今回設定なしですぐに使えるsendmail.cfファイルをCD-ROMに収録した。CD-ROMがマウントされていることを確認してからリストBのコマンドを実行する。次に、リストCのように「/etc/rc.conf」ファイルを修正し、sendmail_enableを「YES」にする。

最後に、設定を反映させるためにFreeBSDを再起動する。

リストB●

```
# cp /cdrom/INTERNETmagazine/sendmail/sendmail-v7.cf /tmp/sendmail.cf
# cd /tmp
# install -c -o root -g wheel -m 644  sendmail.cf /etc
```

リストC●「rc.conf」

```
sendmail_enable="YES"      # Run the sendmail daemon (or NO).
```

popperの設定

FreeBSD用のPOP3サーバソフトウェアとしてqpoperが配布されている。今回は最新のバージョン2.4をインストールしよう。このためのportsは先にCD-ROMからコピーしてあるはずだ。今回、popperのtarballはCD-ROMに収録されていないので、portsを使ってFTPサーバーからtarballをダウンロードする方法に挑戦してみる。といっても、DNSの設定が済んでいれば次のコマンドを実行するだけで自動的に済んでしまう。

```
#  cd /usr/ports/mail/popper
#  make  install
```

画面に表示されるメッセージを見ていると、FTPサーバーにアクセスしてpopperのtarballをダウンロードしているのが分かるはずだ。インストールが終わったら「/etc」ディレクトリーに移動し、リストCの「/etc/inetd.conf」ファイルを編集する。POP3に関する設定がコメントになっている（最初の1文字目に「#」がかかれている）箇所を探し、その行の先頭の「#」を削除する。

設定を反映させるために、次のコマンドを実行する。

```
#  killall -HUP inetd
```

リストC●「inetd.conf」

```
# example entry for the optional pop3 server
#
pop3    stream    tcp     nowait     root     /usr/local/libexec/popper     popper
```

動作を確認しよう

まず、sendmailだけでも動いているかどうかを確認したい。

そのためには、FreeBSDでmailコマンドを実行する。

```
# mail  メールアドレス
```

続いて現れる「Subject」にメールの題名を入力して「Enter」キーを押す。そのあとが本文入力になる。本文の最後の印に「Ctrl」キーを押しながら「d」キーを入力する。これで電子メールが送信される。まず、自分自身に出してみよう。rootでログインしているのなら、次のようになる。

```
# mail root
```

出したメールを読むには、次のコマンドを実行する。

```
# mail
```

メールが来ていれば知らせてくれ、読むことができる。ここでメールが届いてなければ、sendmailの設定手順を確認する。

各クライアントの設定

次に、クライアントになるコンピュータの電子メールソフトを設定しよう。

SMTPサーバー:<サーバーのホスト名>
POP3サーバー:<サーバーのホスト名>
アカウント：＜FreeBSDでの利用者名＞
パスワード：＜FreeBSDの利用者パスワード＞

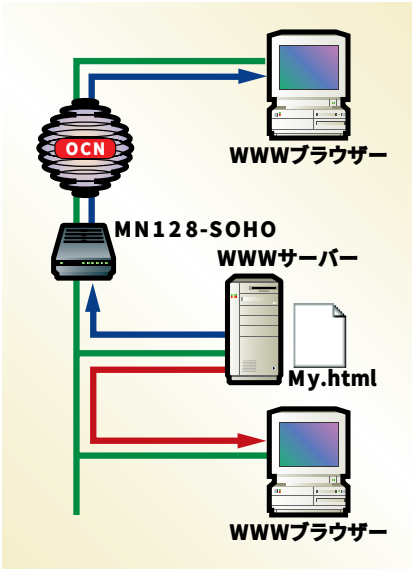
POP3アカウントに、＜FreeBSDでの利用者名＞@<サーバーのホスト名>を書くメールソフトもあるようだ。この詳細は、使用する電子メールソフトのマニュアルを参照してほしい。

クライアントから自分宛に電子メールを送信し、さらに着信するかどうかを確認する。着信しなければFreeBSD上で着信を確認する。ここで問題なく着信を確認できれば、qpopperの設定をもう一度確認しよう。

WWWサーバーを作ろう

ドメイン名を取得した喜びを実感する瞬間

DNSもメールサーバーも動くようになった。と来れば、次はWWWサーバーだろう。自分が取得したドメイン名がそのままウェブコンテンツのURLになる。その喜びを味わえる瞬間がやってきた。ここで紹介するWWWサーバー「Apache」はDNSやメールサーバーと比べると設定はいたって簡単。ここは、ちょっと一休みの気分で気軽に挑戦してみてほしい。



WWWサーバーの役割

WWWサーバーはHTMLファイルを中心にさまざまな種類のコンテンツを保持し、クライアントのWWWブラウザーからの要求に応じてコンテンツをHTTPプロトコルで送信する。

最近のホームページではCGIを使ってサーバーマシンで処理した結果をWWWブラウザーに表示させるといったことが多く行われている。プロバイダーのサービスを使う場合、CGIの利用やホームページの容量に制限を設けられてしまうが、自家製WWWサーバなら制限はほとんどなし。自分のウェブサイトを自由に拡張できるだろう。

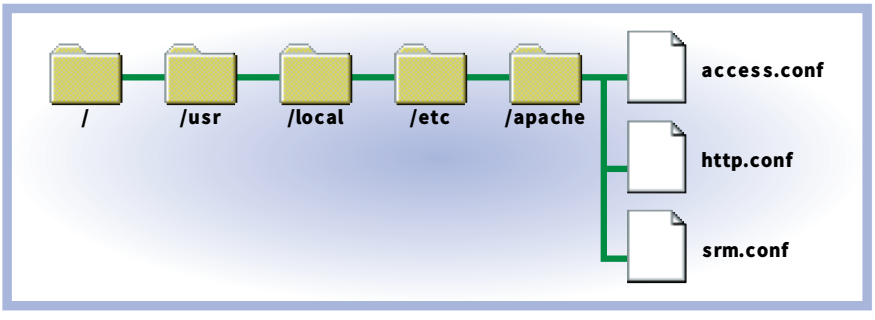
CGIで使われるPerlもFreeBSDで動作するし、ホームページのサイズもハードディスクの許す限り増やすことができる。ホームページを存分に作り込めるのだ。さらに、LAN側の利用者に向けてコンテンツを充実させていけば、立派にイントラネットの中心的サーバーとなるだろう。

Apacheのインストール

WWWサーバーソフトにはApacheを使う。本誌CD-ROMには2月10日の時点での最新版Apache 1.2.5を収録した。このバージョンではセキュリティー関連のバグフィックスもなされている。CD-ROMから最新のportsをコピーしたら、Apacheのディレクトリーに移動して「make install」を行う。これでインストールは完了だ。

```
# mount  /cdrom
# cd  /usr/ports/www/apache
# make  install
```

インストールが終了したら、FreeBSDを再起動する。



FreeBSDでの設定

Apacheの設定ファイルは、/usr/local/etc/apacheディレクトリーに置かれるaccess.conf、httpd.conf、srm.confの3つ。access.conf、httpd.confは変更しなくても大丈夫。リストAのsrm.confの中から以下の記述を見つけてほしい。❶URLでサーバー名だけ指定したときのルートディレクトリーが指定されている。❷利用者のホームディレクトリーを公開するときのサブディレクトリー名。❸URLでファイル名までの指定がなかったときに、初めに読む込むファイルの名前。複数のファイルを指定するときは、スペースで区切る。❹ディレクトリー内をWWWブラウザーで表示させたときにアイコンを使用するかどうかを設定する。この先頭に「#」を付けてコメントアウトするとアイコンを表示しなくなる。❺と❻アイコン表示時のアイコンとファイルタイプの関連付け、アイコンと拡張子の関連付けを定義している。❼アクセス制御命令を記述したファイルの名前を指定する。❽URLの要求で第一引数の文字列を第二引数に置き換えて処理をする。長いURLを縮めて表記するときに役立つ。❾サーバースクリプトの指定で❽と同様の置き換えをする場合、このように指定する。❿このサーバーを示すURLの中に第一引数の文字列を見つけると、第二引数のURLへと移動する。つまり、クライアントが「http://www.kurazono.co.jp/impress/」を参照すると、対象の文字列が見つかるので「http://www.impress.co.jp/」に移動することになる。通常はコメントになっているので、「#」をはずして使う。

リストA●「srm.conf」

```
❶ DocumentRoot      /usr/local/www/
❷ dataUserDir       public_html
❸ DirectoryIndex    index.html
❹ FancyIndexing     on
❺ AddIconByType (TXT,/icons/text.gif) text/*
❻ AddIcon           /icons/binary.gif .bin .exe
❼ AccessFileName    .htaccess
❽ Alias             /icons/ /usr/local/www/icons/
❾ ScriptAlias       /cgi-bin/ /usr/local/www/cgi-bin/
❿ Redirect          fakename url
```

ホームページを置いてみる

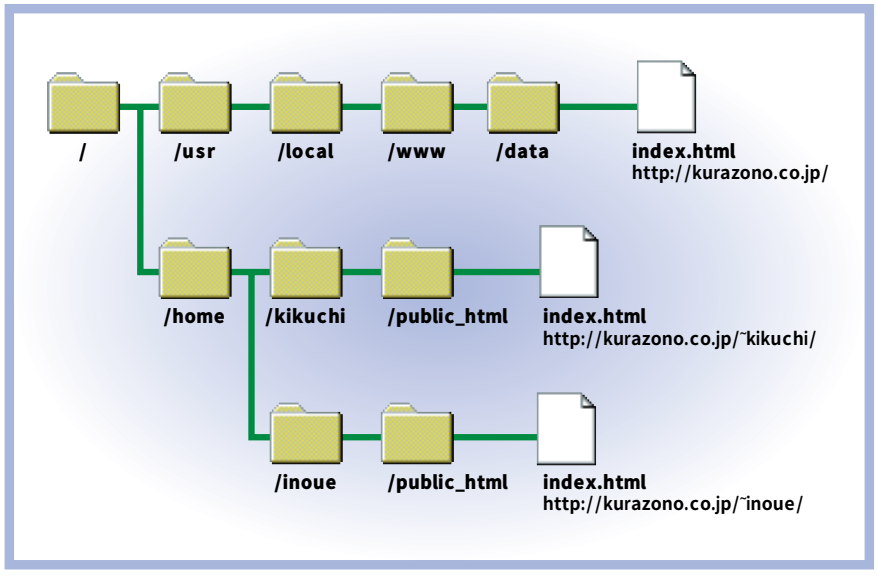
srm.confの「DocumentRoot」で指定された/usr/local/www/dataディレクトリーに「index.html」というファイル名で自分の作成したページを置いてみよう。これがサーバーのホームページ、つまりhttp://www.kurazono.co.jp/とURLを指定した際に表示されるページとなる。

また、サーバー内のそれぞれの利用者のホームディレクトリーにページを置く場合には、それぞれが自分のアカウントでログインしてホームディレクトリーにsrm.confの「UserDir」で設定した名前（先の例ではpublic_html）のディレクトリーを作り、この中に「index.html」を置く。このページを見るためのURLはユーザー名が「kikuchi」なら

http://www.kurazono.co.jp/~kikuchi/

となる（図A）。

A



コマンドワンポイントレッスン2

「public_html」ディレクトリーを作る

1）自分のアカウントでログイン
2）% mkdir　public_html……「public_html」ディレクトリーを作成
3）% cd public_html……「public_html」ディレクトリーに移動

アクセスを制御する

それぞれのユーザーが公開しているホームディレクトリーでもアクセス制御はできる。このためにはsrm.confの「AccessFileName」で指定した名前を付けてアクセス制御を書き記したファイルをそれぞれのホームディレクトリーに置いておけばよい。先の例では、ファイル名を「.htaccess」と設定していたはずだ。アクセス制御の書き方には多くのバリエーションがあり、後述のApacheのマニュアルに詳細が記されている。リストBの「.htaccess」では、index.htmlが存在しないディレクトリーをURLで指定した場合、ディレクトリーの中身をWWWブラウザーに表示しない設定になる。

リストB●「.htaccess」

```
AllowOverride None
Options None
allow from all
```

マニュアルを表示させよう

PORTSを使ってインストールすると、Apacheのマニュアルも組み込まれる。このマニュアルはHTMLで記述されているため、ここで設定したばかりのWWWサーバーで公開すればWWWブラウザーを使って読める。そこで、サーバーのホームディレクトリー（ここでは/usr/local/www/data）の下にマニュアルへのリンクを設定しよう。リストCのコマンドを実行しよう。

あとはWWWブラウザーで、サーバーのホームページのURLに「man/」を付け足す。

http://www.kurazono.co.jp/man/

これでマニュアルを読めるようになる。

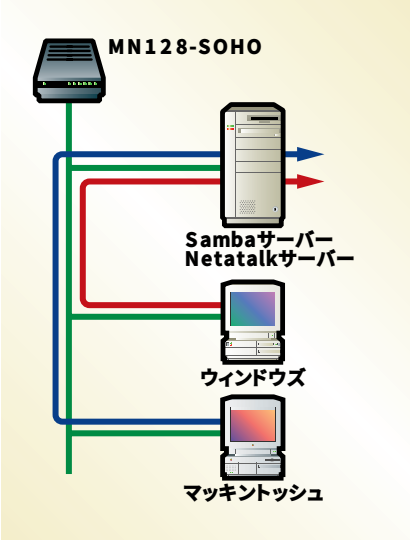
リストC●「srm.conf」

```
# ln -s /usr/ports/www/apache/apache_1.2.5/htdocs /usr/local/www/data/man
```

ファイルサーバーを作ろう

ウィンドウズやマッキントッシュとファイルを共有するために

ここまでで、インターネットサーバーとしての機能はかなり充実したはずだ。そこで、次のステップとしてFreeBSDをインストールしたサーバーをファイルサーバーにして、ウィンドウズやマッキントッシュなどの異なったプラットフォーム間でのファイル共有ができる環境を作ってみよう。自宅のネットワークがさらにパワーアップすること間違いなしだ。



ファイルサーバーの役割

ファイルサーバーは、ネットワーク上の複数の利用者に対してファイル共有の機能を提供する。ウィンドウズでは、ファイルサーバーとしてNTサーバーのファイル共有機能があり、小規模なピアツーピア接続としてウィンドウズ95やNTワークステーションのファイル共有機能がある。マッキントッシュの場合は、ファイルサーバーとしてはAppleShareが標準的で、ピアツーピア接続ならシステムに組み込まれているファイル共有がこれにあたる。

FreeBSDでは、これらと互換性のあるファイル共有サービスを提供するサーバーソフトウェアが用意されている。

ウィンドウズクライアントとのファイル共有を実現するソフトはSambaだ。SambaはFreeBSD上で設定した共有ディレクトリーをウィンドウズ95やNTからアクセスできるようにする。クライアントからの操作は、NTサーバーや共有に設定されたウィンドウズ95のフォルダーにアクセスするときと同じように、ネットワークコンピュータやエクスプローラを使って行う。

マッキントッシュのファイル共有をサポートするソフトはnetatalkになる。マッキントッシュに組み込まれているAppleShareクライアントによって、FreeBSD上の共有ディレクトリーにアクセスできる。

Sambaのインストールと設定

本誌CD-ROMから最新のportsをコピーしたら、Sambaのディレクトリーに移動して「make install」を行う。これでインストールは完了だ。

```
# mount  /cdrom
# cd  /usr/ports/japanese/samba
# make  install
```

インストールが終わったら、/usr/local/etcのディレクトリー内に作られたサンプルファイルを元に「smb.conf」を編集する。次のコマンドを実行しよう。

```
# cd /usr/local/etc
# cp smb.conf.sample smb.conf
```

smb.confはSambaの動作を設定するファイルだ。大きく分けて、[global]、[homes]と、それ以外のエントリーからなる。

[global]エントリーは全体に関係する設定だ(リストA)。

```
workgroup = KURAZONO-HOME
```

にワークグループ名を設定する。さらに

```
guest account = nobody
```

でゲスト接続するときのアカウントを設定している。

[homes]エントリーはFreeBSDにアカウントを持つユーザーのホームディレクトリーにファイル共有ディレクトリーを設置する場合に参照される(リストB)。アクセスするにはFreeBSDにログインするときと同じユーザーIDとパスワードが必要になる。重要なファイルの閲覧などファイルを読むだけの場合は「read only」の項目を「yes」としよう。

ホームディレクトリー以外に共有フォルダーを設定するときは、新しくエントリーを作る(リストC)。

次に、FreeBSDの起動時に自動的にSambaサーバーが立ち上がるようにする。次のコマンドを実行しよう。

```
# cd /usr/local/etc/rc.d
# cp samba.sh.sample samba.sh
```

設定が終わったら、Sambaを起動するためにシステムを再起動する。これで完了だ。

リストA●「smb.conf」

```
[global]
   comment = FreeBSD - Samba %v
   workgroup = KURAZONO-HOME
   printing = bsd
   printcap name = /etc/printcap
   load printers = yes
   guest account = nobody
```

リストB●「smb.conf」

```
[homes]
   comment = Home Directories
   browseable = no
   read only = no
   create mode = 0750
```

リストC●「smb.conf」

```
[tmp]
   comment = Temporary file space
   path = /tmp
   public = yes
   only guest = yes
   writable = yes
```

ウィンドウズ95でアクセスしてみる

ウィンドウズ95側では、TCP/IPでLAN上のNTサーバーやウィンドウズ95のファイル共有が使える設定にする。コントロールパネルの「ネットワーク」を開き、ネットワーク構成に「Microsoftネットワーククライアント」があることを確認する。ユーザー情報の「ワークグループ」はSambaで設定したワークグループと同じにする（右図）。

設定が正しければ、ワークグループにファイルサーバが見つかるはずだ。

デスクトップのネットワークコンピュータやエクスプローラからSambaのフォルダーにアクセスしたときにパスワードを尋ねられたら、FreeBSDにログインする際のパスワードを入力すればいい。もちろん共有フォルダーなら、ゲストでもアクセスできる。

「ワークグループ」にリスト9で指定したものと同じ名前を入れる

netatalkのインストールと設定

初期設定ではFreeBSDのカーネルはマッキントッシュの標準プロトコルであり、AppleShareファイルサーバーにも使われている「AppleTalk」をサポートしていない。インストールの前に235ページに掲載のnetatalkを使うための設定（修正パッチのインストールとカーネルの再作成）を済ませておこう。

本誌CD-ROMから最新のportsをコピーしたら、netatalkのディレクトリーに移動して「make install」を行う。これでインストールは完了だ。

```
#  mount  /cdrom
#  cd  /usr/ports/net/netatalk
#  make  install
```

インストールが終わったらFreeBSD内のどのディレクトリーを共有するかを設定する。/usr/local/etcのディレクトリー内にある「AppleVolumes.default」の最後に共有するディレクトリーを追加しよう（リスト12）。初めの項目は共有するディレクトリー、2番目の項目はその共有ディレクトリーをクライアントから見たときの名前になる。すでに書かれている「~」はFreeBSDのユーザーのホームディレクトリーを表す。リストDでは「/tmp」ディレクトリーを「Public」という名前で共有に設定している。

複数のAppleTalkゾーンがある場合は、サーバーがどのゾーンに属するかを指定する。/usr/ports/net/netatalk/work/netatalk-1.4b2/configディレクトリーにあるatalkd.confを編集し、/usr/local/etcにコピーしておこう。記述するのは1行で、

<インタフェース名> -phase 2 -net <AppleTalkネットレンジ開始番号>-<AppleTalkネットレンジ終了番号> -addr <サーバのAppleTalkネット番号>.<ノード番号> -zone "<サーバの属するゾーン名>"

となる（リストE）。なお、インターフェイス名は237ページの＜FreeBSD＞の設定で使用したイーサネットインターフェイスのデバイス名のこと。

設定が終わったら、netatalkを起動するためにシステムを再起動する。これで完了だ。

リストD●「AppleVolumes.default」

```
~
/tmp "Public"
```

リストE●「atalkd.conf」

```
ep0 -phase 2 -net 0-65534 -addr 65280.226 -zone "AppleTalk Zone"
```

マッキントッシュでアクセスしてみる

クライアントは、イーサネットに接続されたマッキントッシュであれば、なんでもよい。Appleメニューから「セレクタ」を選び、ウインドウ左側のアイコンから「AppleShare」を選ぶ。AppleTalkのネットワークが複数のゾーンを構成している場合は、ゾーンリストからnetatalkサーバーの属するゾーンを指定する。すると、右側にサーバーが現れる。DNSで設定した名前になっているはずだ。

サーバーを選んで「OK」を押すと、IDとパスワードを尋ねてくる。FreeBSDのそれぞれのユーザーのホームディレクトリーにアクセスするのなら、FreeBSDにログインする際のアカウントとパスワードを入力する。それ以外の共有ディレクトリーは、ゲストで利用できる。

ゲストまたは利用者IDとパスワードを入力すると、共有されているディレクトリーがボリュームリストに表示され、最後にマウントしたいボリュームを選ぶと、デスクトップにそのボリュームのアイコンが現れる。

リストDで指定した「Public」が見える

さらに強力なサーバーにしよう

DHCPサーバーとFTPサーバーを運用する

自宅のネットワークにノートパソコンなどの移動可能なPCが接続されているなら、IPアドレスを動的に割り当てるDHCPサーバーがあれば便利だ。また、ネットワークの内外を問わず、ほかのコンピュータと頻繁にファイルのやり取りをするなら、FTPサーバーも絶対に欠かせない。さらに強力なサーバーにするために、これらの機能を追加してみよう。

DHCPサーバーの役割

DHCPサーバーは、DHCPクライアントからの要求に応じて、IPアドレスを動的に割り当てる。ノートパソコンのように、特定のLANに接続されるだけでなく、移動によって数か所のLANに接続し再び切断するような場合に、DHCPサーバから常に適切なIPアドレスを割り当ててもらうようにすれば、ネットワークの設定を手動で書き換える手間がかからない。DHCPサーバーは、IPアドレスはもちろん、ドメイン名やDNSサーバー、デフォルトゲートウェイなどの情報も割り当てられるからだ。

リストA●「dhcpdb.pool」

```
global:!snmk=255.255.255.240:tmof=32400:

kurazononet:tblc=global:dht1=500:dht2=850:brda=203.139.162.239:dnsd=kurazono
.co.jp:dnsv=203.139.162.226:rout=203.139.162.225:albp=true:

node0:     :ipad=203.139.162.233:dfll=3600:maxl=7200:tblc=kurazononet:
node1:     :ipad=203.139.162.234:dfll=3600:maxl=7200:tblc=kurazononet:
node2:     :ipad=203.139.162.235:dfll=3600:maxl=7200:tblc=kurazononet:
node3:     :ipad=203.139.162.236:dfll=3600:maxl=7200:tblc=kurazononet:
node4:     :ipad=203.139.162.237:dfll=3600:maxl=7200:tblc=kurazononet:
node5:     :ipad=203.139.162.238:dfll=3600:maxl=7200:tblc=kurazononet:
```

DHCPサーバーのインストールと設定

本誌CD-ROMから最新のportsをコピーしたら、WIDE DHCPのディレクトリーに移動して「make install」を行う。これでインストールは完了だ。

```
# mount  /cdrom
# cd  /usr/ports/net/wide-dhcp
# make  install
```

WIDE DHCPを使うには、カーネルの変更が必要である。インストールの前に235ページに掲載のWIDE DHCPを使うための設定（カーネルの再作成）を済ませておこう。

次に、カーネルに組み込んだ「Berkeley Packet Filter（BPF）」に対応できるようにBPFで使用するデバイスを作成する。以下のコマンドを入力しよう。

```
#  cd  /dev
#  ./MAKEDEV  bpf0  bpf1  bpf2  bpf3
```

DHCPサーバーの設定ファイルは2つ。まず、1つ目の「/etc/dhcpdb.relay」を作ろう。これは空のファイルだ。

```
# touch /etc/dhcpdb.relay
```

2つ目の設定ファイルは/etc/dhcpdb.poolで、これも新規に作成する。各項目は「:」で区切る。リストAを見てみよう。ファイル前半の設定は以下のとおりだ。

!snmk=＜サブネットマスク＞
brda=＜ブロードキャストアドレス＞
dnsd=＜ドメイン名＞
dnsv=＜DNSサーバーアドレス＞
rout=＜デフォルトルーターアドレス＞
albp=true＜BOOTP対応の古いマッキントッシュでDHCPを使えるようにするためのオプション＞

後半は、割り当てるIPアドレスの個数分だけレコードを追加する。この際にIPアドレスが重複しないように注意しよう。次の書式で記述する。

ipad=＜割り当てるIPアドレス＞

次に、FreeBSDの起動時に自動的にDHCPサーバーが立ち上がるようにする。リストBのコマンドを実行しよう。

次に「wide-dhcps.sh」を開きリストCの「[Interface Name]」の部分を237ページの＜FreeBSD＞の設定で使用したイーサネットインターフェイスのデバイス名に書きかえる。

設定が終わったら、DHCPサーバーを起動するためにシステムを再起動する。これで完了だ。

リストB●

```
# cd /usr/local/etc/rc.d
# cp wide-dhcps.sh.sample winde-dhcps.sh
```

リストC●

```
/usr/local/sbin/dhcps [Interface Name]
                  ↓
/usr/local/sbin/dhcps ep0
```

クライアントでDHCPサーバーの働きを確認しよう

まず、クライアントのコンピュータ側でDHCPを使えるように設定する。

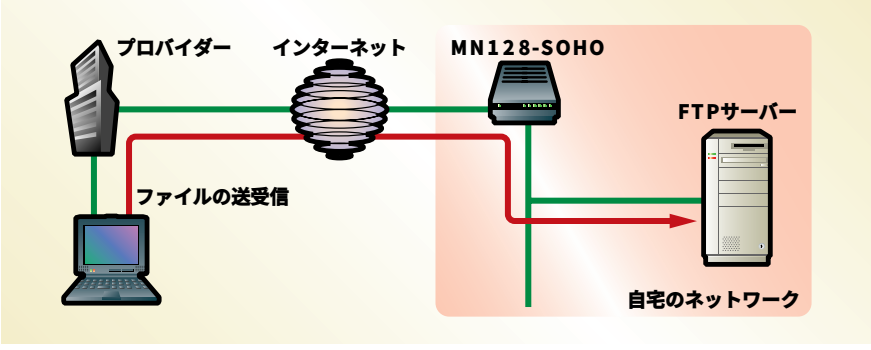
ウィンドウズならば、コントロールパネルの「ネットワーク」設定から「TCP/IP」の「プロパティ」を押し、「IPアドレス」タブで「IPアドレスを自動的に取得」にチェックを付ける。コンピュータを再起動したら「スタート」メニューの「ファイル名を指定して実行」を選んで「winipcfg」と入力する。「IP設定」の各項目に正しいアドレスが表示されていればOKだ。

マッキントッシュならコントロールパネルの「TCP/IP」で「DHCPサーバを使う」を選ぶ。コントロールパネルに「TCP/IP」がなくて「MacTCP」を使っているならば、IPの割り当て方法を選ぶボタンで「サーバから」を選ぶようにする。コンピュータを再起動したら「TCP/IP」（MacTCP）の「ファイル」メニューから「情報を見る」を選ぶ。「TCP/IP」情報の「このコンピュータ」と「ルータアドレス」に正しいアドレスが表示されていればOKだ。

ウィンドウズ95の「IP設定」（上）とマッキントッシュの「TCP/IP情報」（下）

FTPサーバーの役割

インターネットを使ってファイルをやり取りする手段として広く使われているプロトコルがFTP。FTPクライアントでFTPサーバーに接続すればファイルの送信と受信ができるようになる。プロバイダー経由などの方法で外から自宅のFTPサーバーにアクセスしてサーバーにファイルを置いたりサーバーからファイルを取得したりといった使い方ももちろん可能だ。



FTPサーバーのインストールと設定

FreeBSDには標準でFTPサーバーが組み込まれているが、ここでは、匿名FTPサーバーとしてよく使われる「wu-ftpd」という別のFTPサーバーをインストールして使う。

本誌CD-ROMから最新のportsをコピーしたらwu-ftpdのディレクトリーに移動して「make install」を行う。これでインストールは完了だ。

```
#  mount  /cdrom
#  cd  /usr/ports/net/wu-ftpd
#  make  install
```

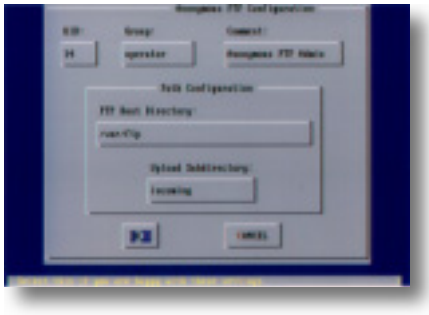
ftpdの起動はinetdというプログラムから行われる。inetdの設定ファイル/etc/inetd.confのftpdの記述を確認しよう。ftpdの置かれたディレクトリーが変更されるので、設定ファイルを編集する。ftpdは下記のインストール前のものとインストール後のものと2つのディレクトリーに置かれる。/etc/inetd.confをインストール後のパスに書き換えよう。

インストール前
/usr/libexec/ftpd
インストール後
/usr/local/libexec/ftpd

システムを再起動すればFTPサーバーが動作する。

この状態でFreeBSDのそれぞれのユーザーはFreeBSDにログインする際のアカウントとパスワードでFTPサーバーにログインできる。匿名FTP（anonymousFTP）を許可するには、「/stand/sysinstall」コマンドを入力して「Configure」→「Networking」→「Anon FTP」にチェックを付ける。図Aの画面が表示されるがこのまま「OK」を押せば設定は完了だ。

匿名FTPにログインした際に、最初に現れるディレクトリーは、匿名FTP用に作られたftpというアカウントのホームディレクトリーになる。つまり、匿名FTPで利用するファイルは「~ftp」のディレクトリーに置けばいいということだ。



コマンドワンポイントレッスン3 システムを再起動せずに、ftpdに新しい設定を反映させる

1）rootでログイン
2）# ps -ax ¦ grep inetd ……現在動いているプロセスをすべて調べて「inetd」だけを表示する
3）2）の結果の先頭の数字をチェックする
4）# kill -HUP 3）でチェックした数字 ……inetdに変更した設定が反映され新しいftpdが起動するようになる

セキュリティーを強化しよう

お金をかけずにここまでできる

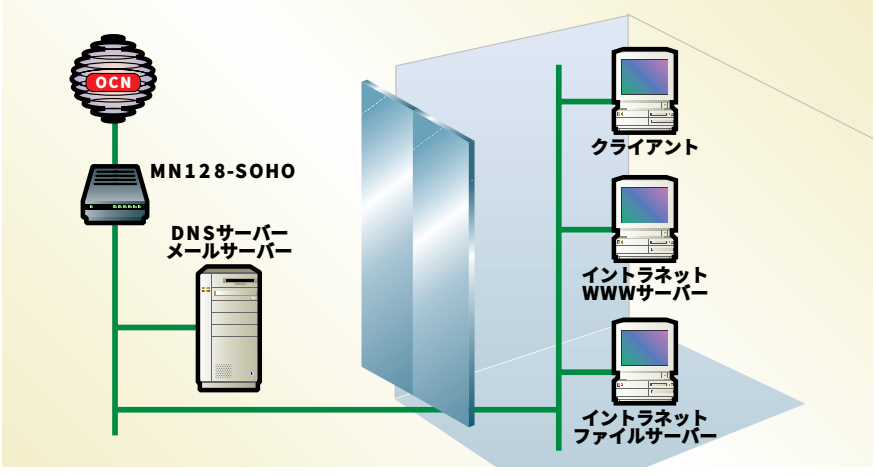
以上でサーバーの設定についての解説は完了だ。でも、実際の運用にあたってはもう1つ忘れてはならない大事な点がある。「セキュリティー」だ。この記事を通してできあがったサーバーは、機能は十分でも「自宅に鍵をかけていない」状態なのだ。そこで、最後の仕上げとしてお金をかけずにちょっとした工夫でネットワークのセキュリティーを強化する方法を解説しよう。

どこを公開してどこを閉じるか

インターネットに常時接続という形態でつながると、セキュリティーを考慮することが必要になってくる。これは、サーバーかクライアントかにかかわらず言えることだ。また、FreeBSDなどのPC-UNIXに限らず、ウィンドウズやマッキントッシュにも言えることである。

セキュリティーの基本は、ドメインの内側と外側との間に、どれだけの制限を加えるかということだ。たとえば、イントラネット用のWWWサービスを提供する場合、ドメイン外からWWWサーバーへのアクセスを一切拒否するといったものがこれにあたる。

しかし、制限を加えてはいけないところもある。DNSサーバーとメールサーバーなどがそれである。たとえばメールサーバーが外から見えない状態になっていると、インターネットから送られてくるメールを受け取れなくなってしまう。重要なのは、外側からも利用できるサービスと内側だけでしか利用できないサービスをはっきり分けることだ。



内側と外側との間で、何らかの方策で制限を加えることを、「ファイアーウォールを構築する」という。このファイアーウォールの構築にはサーバーの特定のサービスを停止するといった単純なことから、ルーター専用機やファイアーウォール専用ソフトウェアの導入などの究極の方法まで、多種多様なやり方がある。

ここでは、FreeBSDとMN128-SOHOを使ってOCNに接続しているときに利用できる、単純なファイアーウォールの例を示す。

MN128-SOHOでパケットフィルターをかける

最初に、誰でもすぐにできる方法として、MN128-SOHOのパケットのフィルタリングを取り上げる。これは、MN128-SOHOを通るTCP/IPプロトコルのパケットに対して、特定の中身のものを通さずに破棄するというものだ。

この設定は、MN128-SOHOのIP設定ページの「オプション」内に、「ip filter」コマンドを記述して行う。コマンドの引数の詳しい意味については、MN128-SOHOのマニュアルを参照してほしい。

たとえば、外側から内側のWWWサーバーへ送られる要求パケットをすべて破棄するには、リストAのように指定する。これで、外側から内側に送られてくるhttpプロトコルのパケットは、すべてMN128-SOHOで破棄される。

また、MN128-SOHOでは、TCPパケットのセッション確立用パケットのみをフィルタリングする機能を持っている。そのため、たとえば、リストBのように指定すれば、外側から内側にTCPセッションを確立することはできないが、内側から外側にはTCPセッションを確立できるという状態を構成できる。

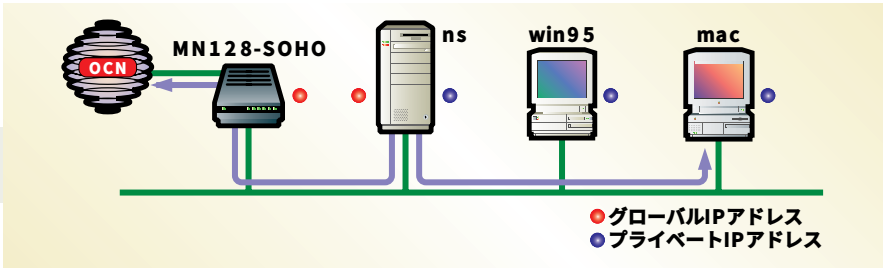
MN128-SOHOのIP設定ページの「オプション」に設定を記述する

リストA

```
ip filter 1 reject in * * * * www remote 0
```

リストB

```
ip filter 1 reject in * * tcpest * * remote 0
```



FreeBSDをゲートウェイホストにする

FreeBSDなどのPC-UNIXが威力を発揮するのは、ゲートウェイホストとなったときである。ルーティング専用機は高速だが、高価でそう簡単には手に入れにくい。しかしPC-UNIXなら非常に安価にゲートウェイホストが完成する。

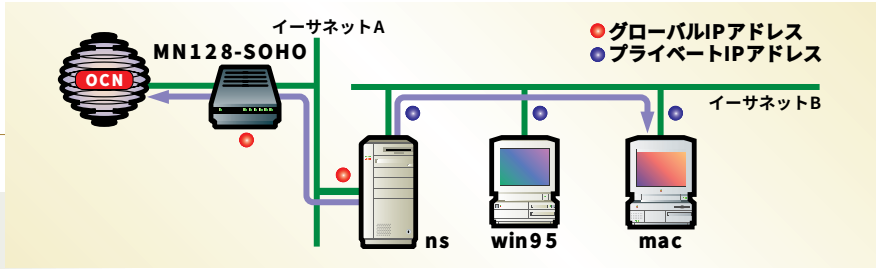
最初の例は、お金を使わずに実現する方法だ。ネットワーク構成はまったく変わらない。これまでどおり、ドメイン内には1つのイーサネットにすべてのホストが接続されている。

FreeBSDではネットワークインターフェイスに対して複数のIPアドレスを割り当てることができる。そこで、今回のネットワークを例にとると、OCNから割り当てられたアドレス、すなわちグローバルIPアドレスをルーターとnsの2つのみに割り当てる。そして外部からアクセスできないプライベートIPアドレスをwin95とmacに割り当て、さらにnsにも割り当てる。すなわちnsにはグローバルとプライベートの2つのIPアドレスが割り当てられ、パケットのルーティングをすることになる。こうすれば、外側からwin95やmacに直接パケットが届くことはなくなる。しかし、たとえばwin95からルーターへは一度nsを経由することになるので、ネットワークトラフィックは倍増することになる。

インターフェイスに2つのIPアドレスを付けるには、ifconfigコマンドでalias引数を用いる。インターフェイスに2つのIPアドレスが付いた状態をリストCに示した。この方式を利用するには、ifconfig以外に各ホストでのIPアドレスの設定や、ルーティング、NATなどの設定が必要となる。

リストC

```
# ifconfig -a
lp0: flags=8810<POINTOPOINT,SIMPLEX,MULTICAST> mtu 1500
ep0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        inet 203.139.162.224 netmask 0xffffff0 broadcast 203.139.162.239
        inet 192.168.0.1 netmask 0xffffff00 broadcast 192.168.0.255
        atalk 65280.226 range 0-65534 phase 2 broadcast 0.255
        ether 00:60:8c:cb:a1:18
tun0: flags=8010<POINTOPOINT,MULTICAST> mtu 1500
sl0: flags=c010<POINTOPOINT,LINK2,MULTICAST> mtu 552
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
        inet 127.0.0.1 netmask 0xff000000
        atalk 0.0 range 0-0 phase 2
#
```



イーサネットカードを2枚挿す

最後の例は、nsにイーサネットカードを2枚挿す方法である。自ドメイン内にイーサネットを2本用意するわけだ。一方のイーサネットにはルーターとnsのみを接続する。これらには、OCNから割り当てられたグローバルIPアドレスを付ける。

そしてもう一方のイーサネットには、nsの2つ目のインターフェイスとwin95、macを接続し、プライベートIPアドレスを付けるわけだ。

パケットの流れは先の例と同じだ。外側と内側とのやり取りは、すべてnsを通る。ただ、ネットワークが完全に分れているので、トラフィックの影響が及ぶことはない。

2枚のイーサネットカードの状態をリストDに示した。

リストD

```
# ifconfig -a
lp0: flags=8810<POINTOPOINT,SIMPLEX,MULTICAST> mtu 1500
ep0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        inet 203.139.162.226 netmask 0xffffff0 broadcast 203.139.162.239
        ether 00:60:8c:cb:a1:18
ep1: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        inet 192.168.0.1 netmask 0xffffff00 broadcast 192.168.0.255
        atalk 65280.226 range 0-65534 phase 2 broadcast 0.255
        ether 00:60:8c:cb:9f:8c
```

FreeBSDでインターネットはもっと身近になる

非常に早足ではあるが、FreeBSDを使ったインターネット兼イントラネットサーバーの構築例を一通り見てきた。FreeBSDによってOCNエコノミーの常時接続環境用のサーバーが安価に用意できるので、自分のドメイン名を持つときの強い味方になることが分かったはずだ。

今回取り上げたソフトウェアは、FreeBSDで用意されているソフトウェアのうちの氷山の一角にすぎない。FreeBSDにはまだまだ有用なソフトウェアが入っている。たとえばセキュリティー関係では、パケットのフィルタリングソフトや、ファイアーウォールを実現するソフトまである。非常に奥が深いことは間違いない。

ただ、商用製品ではないので、手厚いサポートを期待する場合には向かない。ある程度の覚悟と努力は必要だ。その努力が身を結んだとき、インターネットに対するスタンスが変化することは間違いない。挑戦するだけの価値は、必ずある。

［インターネットマガジン　バックナンバーアーカイブ］ ご利用上の注意

このPDFファイルは、株式会社インプレスR&D（株式会社インプレスから分割）が1994年～2006年まで発行した月刊誌『インターネットマガジン』の誌面をPDF化し、「インターネットマガジン　バックナンバーアーカイブ」として以下のウェブサイト「All-in-One INTERNET magazine 2.0」で公開しているものです。

http://i.impressRD.jp/bn

このファイルをご利用いただくにあたり、下記の注意事項を必ずお読みください。

- 記載されている内容（技術解説、URL、団体・企業名、商品名、価格、プレゼント募集、アンケートなど）は発行当時のものです。
- 収録されている内容は著作権法上の保護を受けています。著作権はそれぞれの記事の著作者（執筆者、写真の撮影者、イラストの作成者、編集部など）が保持しています。
- 著作者から許諾が得られなかった著作物は収録されていない場合があります。
- このファイルやその内容を改変したり、商用を目的として再利用することはできません。あくまで個人や企業の非商用利用での閲覧、複製、送信に限られます。
- 収録されている内容を何らかの媒体に引用としてご利用する際は、出典として媒体名および月号、該当ページ番号、発行元（株式会社インプレスR&D）、コピーライトなどの情報をご明記ください。
- オリジナルの雑誌の発行時点では、株式会社インプレス R&D（当時は株式会社インプレス）と著作権者は内容が正確なものであるように最大限に努めましたが、すべての情報が完全に正確であることは保証できません。このファイルの内容に起因する直接的および間接的な損害に対して、一切の責任を負いません。お客様個人の責任においてご利用ください。

このファイルに関するお問い合わせ先
株式会社インプレスR&D
All-in-One INTERNET magazine 編集部
im-info@impress.co.jp