特集 2

## ウイルス，スパムをしっかり排除

# 実践 メール・サーバー再入門

「独自のメール・アドレスを使いたい」，「どこからでも受信メールを確認したい」，「迷惑メールは自動的に捨てたい」――。日ごろ当たり前のようにメールを使っている人でも，このような“一歩進んだ”使い方をしたいと思っているのではないだろうか。それを実現するのが，自宅に設置するメール・サーバー。メールの仕組みからメール・サーバーの導入方法まで，一から解説する。

（ライター　福田 和宏）


ILLUSTRATION ＊SHIGEKO NAKAYAMA

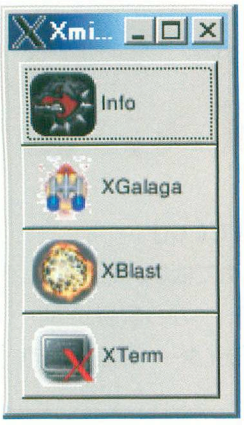


写真13
BlackDogのアプリケーション・ランチャ

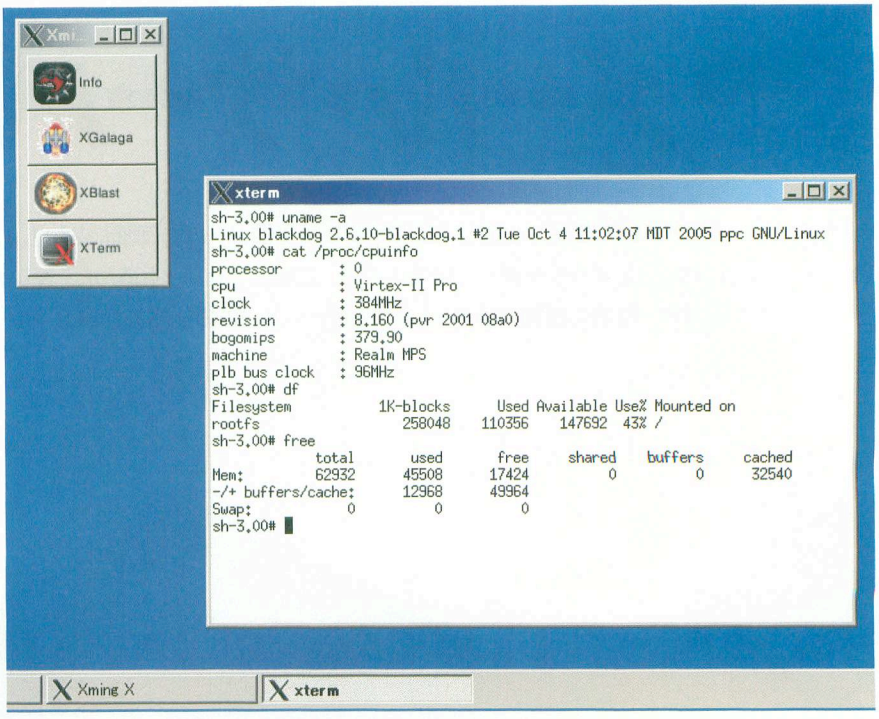


写真14 ターミナルから動作環境を表示
unameコマンドの出力から，CPUはPowerPC系のVirtex-II Proプロセッサであること，dfコマンドの出力から256MバイトのUSBメモリーであること，freeコマンドからメモリーが64Mバイトであることが分かる。

えるわけだ。

## 内蔵LinuxはDebianベース

BlackDogが内蔵するLinuxは，Debian GNU/Linuxベースである。各種コマンドに「BusyBox」が使われるなど，通常のDebianと互換性はないが，aptを採用しているため，BlackDog用のapt-lineを設定することで，簡単にアプリケーションをインストールできる。

例えば，Firefoxも簡単にインストールできる（写真17）。このFirefoxを使えば，いつでもどこでも全く同じ環境でWebブラウジングが可能だ。BlackDog上で動作するアプリケーションは，カスタマイズも自由自在に行える。ただし，BlackDogには日本語フォントや日本語ロケールは入っていない。DebianのPPCパッケージから持ってくるとよいだろう。

また，XmingはWindows IMEからの日本語入力をサポートしていない。「Anthy」などを用いてX上から日本語を入力する必要がある。

（ライター　増井 雄一郎）

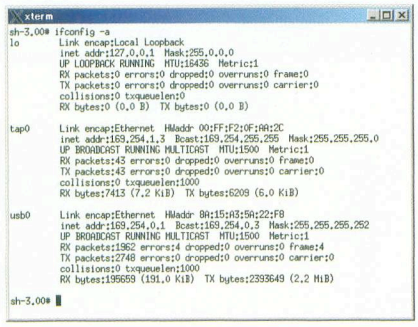


写真15 BlackDogから見たネットワーク・インタフェースの状態

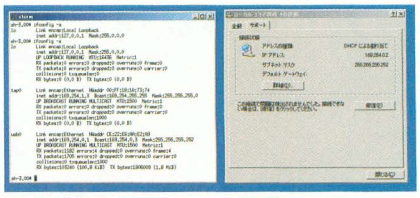

写真16 ネットワークの状態
BlackDog側から（左）とWindows側から（右）見た状態を示した。

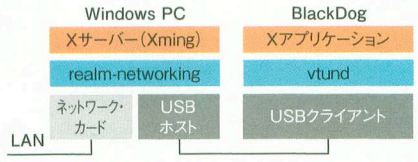


図2 Windows PCとBlackDogのソフトウエア構成図

写真17 BlackDog上でFirefoxが動作

＊1 BlackDogは米Cisco Systems社の一部門であるLinksysが製造したネットワーク・インタフェースLSI「ADM8511」をFPGAでエミュレートしている。
＊2 BlackDogの指紋認証装置は企業内利用を想定して設けられた。例えば，VPN（Virtural Private Network）クライアントを内蔵すれば，BlackDogを紛失した場合の安全性を高められる。

```
smtp-amavis unix -       -       n       -       2       smtp
        -o smtp_data_done_timeout=1200
        -o disable_dns_lookups=yes

127.0.0.1:10025 inet n  -       n       -       -       smtpd
        -o content_filter=
        -o local_recipient_maps=
        -o relay_recipient_maps=
        -o smtpd_restriction_classes=
        -o smtpd_client_restrictions=permit_mynetworks,reject
        -o smtpd_helo_restrictions=
        -o smtpd_sender_restrictions=
        -o smtpd_recipient_restrictions=permit_mynetworks,reject
        -o mynetworks=127.0.0.0/8
        -o smtpd_error_sleep_time=0
        -o smtpd_soft_error_limit=1001
        -o smtpd_hard_error_limit=1000
```

**図11　Postfixのmaster.cfの設定**
PostfixがAMaViSdを呼び出す際の設定をmaster.cfファイルの末尾に追記する。

sinを起動するだけだ。

```
# /etc/init.d/spamassassin start ↵
# /sbin/chkconfig spamassassin on ↵
```

## Postfixの設定

これで各アプリケーションの設定ができたが，このままではPostfixが届いたメールをAMaViSdに渡さないため，メールの検査は行われない。そこで，Postfixの設定を変更してAMaViSdにメールを渡すようにする。

まず，/etc/postfix/main.cf設定ファイルの変更を行う。設定ファイルの末尾にp.69の図10のような項目を追記して，AMaViSdに届いたメールを渡すように設定する。

次に，/etc/postfix/master.cf設定ファイルの変更を行う。master.cfファイルではPostfixが起動するデーモンの設定を行うファイルである。今回は，Postfixが読み出すAMaViSdの設定を行う。設定ファイルの末尾に図11の内容を追記する。

これで設定完了だ。Postfixを再起動して設定を読み込もう。

```
# /etc/init.d/postfix restart ↵
```

## ウイルス除去のテスト

では，実際にウイルス・メールが除去されるかを確かめてみよう。ウイルスのテストにはEICAR（http://www.eicar.com/）で公開しているテスト用のウイルスを利用するとよい。このウイルスはウイルス・チェッカーで認識されるが，実際には無害なウイルスだ。EICARウイルスはhttp://www.eicar.com/anti_virus_test_file.htmから入手できる。ウイルス・メールは数種類用意されており，目的によって利用するウイルスが選べる。

今回はテキストで記載されたウイルス「eicar.com.txt」を利用する。ウイルスのリンクをクリックすると，1行で記載された文字列が表示される。この文字列をコピーし，メールの本文に張り付けて自分あてに送ってみよう。

正常にウイルスと判断されたら，メールはユーザーには届かない。ログ・ファイルには，ウイルス・メールを検知したため削除したと表示される（図12）。

```
          :
Jul 18 20:32:07 kazumail amavis[4813]: (04813-01) Blocked INFECTED (Eicar-Test-Signature), [192.168.1.4]
<fukuda@kazumail.homelinux.net> -> <fukuda@kazumail.homelinux.net>, quarantine: virus-CDXGSUkQBazx,
Message-ID: <44BCAAAE.5050105@kazumail.homelinux.net>, mail_id: CDXGSUkQBazx, Hits: -, 384 ms
Jul 18 20:32:07 kazumail postfix/smtp[5054]: A5E904C8C6: to=<fukuda@kazumail.homelinux.net>, relay=127.0.0.1
[127.0.0.1], delay=1, status=sent (254 2.7.1 Ok, discarded, id=04813-01 - VIRUS: Eicar-Test-Signature)
                                                                    ↑ Eicar が見つかったと記載されている
Jul 18 20:32:07 kazumail postfix/qmgr[5025]: A5E904C8C6: removed
```

**図12　ウイルス・メールの検知をログに記載**
/var/log/maillogファイルにウイルス・メールが検知されて削除したことが記載されている。

から8行の先頭にある「#」一つを削除する。さらに，「/var/run/clamav/clamd」を「/var/run/clamav/clamd.sock」に変更しておく。

これで，設定完了だ。設定を保存し，次のようにしてAMaViSdを実行する。

```
# /etc/init.d/amavisd st
art
# /sbin/chkconfig amavisd
 on
```

### Fedora Core向けに設定

Fedora CoreのClam AntiVirusのパッケージを利用する場合は，/etc/clamd.d/amavisd.confファイルを編集する。設定ファイルにはメールを検査するようファイルの末尾に「Scan Mail」と追記すればよい。

設定が終わったら，Clam AntiVirusを起動する。

```
# /etc/init.d/clamd.amav
isd start
```

```
# /sbin/chkconfig clamd.
amavisd on
```

Clam AntiVirusのウイルス定義ファイルは日々更新されている。そのため，定期的にアップデートすることを勧める。アップデートはfreshclamで行われるが，freshclamの設定ファイルが無効になっているため，設定ファイルを書き換える必要がある。/etc/freshclam.confファイルを編集し，「Example」の行頭に「#」を付けて保存する。これでfreshclamコマンドを実行すると定義ファイルが更新される。

```
# freshclam
```

### CentOS向けに設定

Cent OSの場合は，/etc/clamd.confファイルを設定する（図9）。

まず，「直接ソケット」を利用して，AMaViSdとデータのやり取りを行うようにする。そのため，「LocalSocket」の先頭にある「#」を取り除く。この際，TCPソケットのやり取りは無効にしておく。「TCPSocket」と「TCP Addr」の先頭に「#」を付ければよい。

AMaViSdの実行権限でClam AntiVirusを読み出すので，「User」を「amavis」に変更する。Clam AntiVirusが作成するファイルのディレクトリについても所有者を変更しておく。

```
# chmod amavis.amavis /
var/run/clamav
```

設定が完了したら，Clam AntiVirusを起動する。

```
# /etc/init.d/clamd st
art
# /sbin/chkconfig clamd
 on
```

また，Clam AntiVirusで利用する定義ファイルはfreshclamコマンドで更新できる。

```
freshclam
```

これで，Clam AntiVirusの準備が完了した。

### SpamAssassinを起動する

SpamAssassinの設定は特に必要ない。そのため，そのままSpamAssas

```
# Example                                    ←先頭に「#」を付け無効にする
LocalSocket /var/run/clamav/clamd.sock
# TCPSocket 3310      ┐          ↑ソケット・ファイルの保存先を指定する
# TCPAddr 127.0.0.1   ┘ ← TCPを利用したソケットの利用は無効にしておく
```

**図9 Clam AntiVirusの設定**
CentOSの場合はClam AntiVirusとAMaViSdがデータのやり取りができるよう設定する。

```
content_filter = smtp-amavis:[127.0.0.1]:10024
```

**図10 Postfixのmain.cfの設定**
メールをAMaViSdに渡すようにファイルの末尾に設定する。

スパム検知能力を学習できる。

また，Postfixとこれらの対策ソフトの間でメールを受け渡すアプリケーションが必要だ。これがメール・ゲートウェイだ。メール・ゲートウェイには「AMaViSd」を利用する。

では，これらのパッケージをインストールし，設定しよう。

## CentOSでのレポジトリの設定

Fedora Core向けのパッケージは，yumコマンドで入手できる。しかし，CentOSではこれらのパッケージは用意されていないため，「Red Hat Enterprise Linux」用のソフトを提供しているサイト「DAG」から入手する。CentOSでもこのパッケージが利用可能だ。

パッケージを入手するにはyumのレポジトリを追加する必要がある。DAGではRPMパッケージとしてレポジトリを配布している。CentOSでもこのパッケージを利用すればよい。パッケージは「http://dag.wieers.com/packages/rpmforge-release/rpmforge-release-0.3.4-1.el4.rf.i386.rpm」を入手する。入手できたら，

```
# rpm -ivh rpmforge-release-0.3.4-1.el4.rf.i386.rpm
# rpm --import http://dag.wieers.com/packages/RPM-GPG-KEY.dag.txt
```

と実行する。これで，yumを利用してパッケージを入手できる。

## 必要なパッケージのインストール

次に必要なパッケージをインストールしよう。ただし，Fedora CoreとCentOSではインストールするパッケージが異なる。Fedora Core用のパッケージをインストールするには次のように実行する。

```
# yum install amavisd-new clamav clamav-server clamav-update spamassassin perl perl-Archive-Tar perl-IO-String
```

CentOSでインストールする場合は次のように実行する。

```
# yum install amavisd-new clamav clamd spamassassin perl perl-Archive-Tar perl-IO-String
```

続いて，各アプリケーションの設定を行う。

## AMaViSdの設定

AMaViSdは，Fedora Coreの場合は「/etc/amavisd/amavisd.conf」，CentOSの場合は「/etc/amavisd.conf」ファイルに設定する（図8）。

まず「$mydomain」にダイナミックDNSで設定したホスト名を入力する。CentOSの場合は，308行目付近にある「### http://www.clamav.net/」

```
$mydomain = 'kazumail.homelinux.net'      ←ドメインの指定をする。

↓ CentOSの場合はClam AntiVirusを利用できるように各行にある先頭の「#」を外す
### http://www.clamav.net/
['ClamAV-clamd',
  \&ask_daemon, ["CONTSCAN {}\n", "/var/run/clamav/clamd.sock"],    ←ソケット・ファイル名を変更する
  qr/\bOK$/, qr/\bFOUND$/,
  qr/^.*?: (?!Infected Archive)(.*) FOUND$/ ],
# NOTE: the easiest is to run clamd under the same user as amavisd; match the
# socket name (LocalSocket) in clamav.conf to the socket name in this entry
# When running chrooted one may prefer: ["CONTSCAN {}\n","$MYHOME/clamd"],
```

図8　AMaViSdの設定
ドメインの設定を行う。また，CentOSの場合はClam AntiVirusが利用できるように設定する。

rrelmailの基本設定を行う。設定は，/usr/share/squirrelmail/config/conf.plファイルを実行すればよい。

```
# /usr/share/squirrelmai
l/config/conf.pl ↵
```

設定は対話的に進む。ターミナルの背景が白だと，入力した文字などが見えなくなってしまうので，「C ↵」と入力し，文字色を変更しよう。

まず，サーバーの設定を行う。「2 ↵」と入力するとサーバー関連の設定画面が表示される。「1. Domain」にダイナミックDNSで設定したホスト名を記述する。設定は「1 ↵」と入力し，続いてホスト名を入力すればよい。「A Change IMAP Server Settings」を選択すると，IMAPサーバーの設定が表示される。この中の「9. Delimiter」を「.」に変更する。設定ができたら「r ↵」と入力しメイン・メニューに戻る。

次に利用する言語を選択する。「10 ↵」と入力したら，「1. Default Language」を「ja_JP」に変更する。

これで設定は完了だ。「q ↵」と入力すると記録するかを聞かれるので「y ↵」と入力する。

インストールと設定が完了したらApacheを起動する。システムを再起動したときにもApacheが自動起動するようにしておくとよいだろう。


**写真5　SquirrelMailでのメール操作画面**
ログインに成功するとサーバー上のメールが一覧される。

```
# /etc/init.d/httpd star
t ↵
# /sbin/chkconfig httpd
on ↵
```

次にWebブラウザで，Squirrelmailのアクセスしてみよう*3。IPアドレスが192.168.1.10ならば，「http://192.168.1.10/webmail」にアクセスすればよい。すると，ログイン画面が表示される。メール・サーバーに登録されているユーザー名とパスワードを入力すると，メールの一覧画面が表示されるはずだ（写真5）。

受信箱にはテスト用に送ったメールが一覧されている。「メッセージ作成」でメールが送信できるかをテストしておこう。

## ウイルス，スパム・メールの除去

最後にウイルスとスパム・メールを自動的に排除する仕組みを導入しよう。ウイルスを除去するにはウイルス・チェッカーを利用する。ここでは無償で利用できる「Clam AntiVirus」を使う。Clam AntiVirusは有志によりパターン定義ファイルが作成されており，多くのウイルスを検知・除去できる。

スパム対策には，「SpamAssassin」を使うことにしよう。SpamAssassinは登録しておいたルールに従ってスパムかどうかを判断して除去してくれる。ベイズ・フィルタが利用でき，

*3　Fedora Coreを利用している場合，Squirrelmailにログインしようとしてもエラー・メッセージが表示されてログインできない場合がある。これは，SELinuxがWeb上で動作するスクリプトの実行を禁止しているためだ。そこでスクリプトが実行できるように設定を変更する必要がある。デスクトップ・メニューにある［管理］-［セキュリティレベルとファイヤーウォールの設定］で，「SELinux」タブをクリックし，「SELinuxポリシーの修正」にある「HTTPDサービス」-「ネットワークに接続するためにHTTPDスクリプトとモジュールを許可する。」にチェックを入れる。

「protocols」に指定する。今回はIMAPサーバを利用するので，「imap」を指定しておけばよい。これだけで設定完了だ。

設定が完了したら，Dovecotを起動する。

```
# /etc/init.d/dovecot st
art
```

システムを再起動した場合も自動的にDovecotが起動するように設定しておく。

```
# /sbin/chkconfig dovecot
 on
```

これで，IMAPサーバーの設定が完了した。

## IMAPサーバーに接続する

次に，メール・クライアントからIMAPサーバーに接続してみよう。今回はメール・クライアントに「Thunderbird」を利用する例を紹介する。

設定はメニューの［編集］-［アカウントの編集］を選択し，「アカウント設定」ダイアログで「アカウントの追加」ボタンをクリックする。

すると，アカウント・ウィザードが起動する。あとは，POPでメール・アカウントを作成するように設定していく。ただし，「サーバ情報」では「IMAP」を選択し，「メール受信サーバ」および「メール送信サーバ」には作成しているメール・サーバーのIPアドレスを記載する。


写真4　IMAPサーバーへの接続
メール・クライアントがメール・サーバーにアクセスすると，サーバー上のメールが一覧される。

設定が完了したら，IMAPサーバーの受信トレイを選択してみよう。すると，パスワードを聞かれるので，サーバーに登録しているユーザーのパスワードを入力する。すると，メールが一覧される（写真4）。メール・ボックスにはPostfixを送信テストに利用したメールが届いているはずだ。

さらに，新規メールを作成し，メール・サーバーに登録されている自分あてにメールを送信してみよう。続いていつも使っているプロバイダのメール・アドレスにもテスト・メールを送ってみよう。

## 外出先からメールを送受信可能に

次に外出先からメールを送受信できるようにする。外出先からメールを閲覧する方法としてWebメールを利用する。Webメールのソフトには，Fedora CoreやCent OSにも標準で組み込まれている「Squirrelmail」を利用する。

SquirrelmailはWebサーバー上で動作するアプリケーションなので，メール・サーバー上でWebサーバーを動作させておく必要がある。そこで，Apache HTTP Serverを導入しよう。

Apacheのインストールは次のように実行する。

```
# yum install httpd
```

これだけで，Webサーバーが利用できる。試しにWebブラウザでメール・サーバーのIPアドレスを指定すると，初期状態のWebページが表示されるはずだ。

Apacheが動作したら，Squirrelmailの導入を行う。インストールは次のようにする。

```
# yum install squirrelmai
l php php-mbstring
```

インストールが完了したら，Squi

## メール送信のテスト

では，実際にメールが送信できることを確かめてみよう。メール送信にはmailコマンドが利用できる（図6）。コマンドの後にあて先を指定する。初めはサーバーに登録されているユーザーあてに送ってみるとよいだろう。

コマンドを実行すると，「Subject:」と聞かれるので，件名を適当に入力する。次にメールの内容を適当に入力する。入力が完了したら［Ctrl］+［D］を押す。「Cc:」と同報メールのあて先を聞かれるが，今回は必要ないので［Enter］キーを押す。これで，メールが配信された。

では，メールが届いているかを確かめてみよう。メールは送信したユーザーのホーム・ディレクトリにある「Maildir」ディレクトリ内に保存される。新しいメールは「Maildir/new」ディレクトリに保存される。lsコマンドでディレクトリ内を確認すると1つのファイルがあるはずだ（写真3）。このファイルをcatコマンドなどで表示すると先ほど送ったメールの内容が読める。これで，ローカル内のメール送信はきちんと行えた。

次に実際に自分が利用しているメール・アドレスあてに送ってみよう。送る際はmailコマンドの後に自分のメール・アドレスを指定すればよい。あとはローカルにメールを送るときと同じだ。

```
$ mail fukuda@localhost ↵
Subject: Test Mail from My Mail Server
This is a test mail from my mail server.
入力が済んだら [Ctrl]+[D] を押す
Cc: ↵
```

図6　メール送信のテスト
実際にメールが送信できるかテストする際には，mailコマンドを利用する。（1）サーバー内のユーザーあて，（2）プロバイダのメール・アドレス向けの順にテストをしてみるとよい。

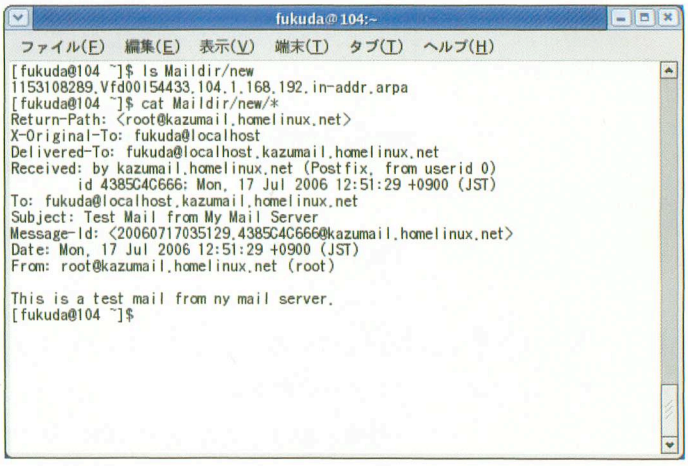


写真3　テスト・メールを確認する
Maildir方式の場合は届いたメールはユーザーのホーム・ディレクトリ以下にある「Maildir」ディレクトリに保存される。

表3　Linuxで利用できる主なIMAPサーバー

| IMAPサーバー名 | URL |
|---|---|
| Dovecot | http://www.dovecot.org/ |
| UW IMAP | http://www.washington.edu/imap/ |
| Cyrus IMAP | http://cyrusimap.web.cmu.edu//imapd/ |
| Courier IMAP | http://mjm4u.net/oss/courier-imap/ |

```
protocols = imap                    ← IMAPを利用するように設定する
```

図7　Dovecotの設定
/etc/dovecot.confファイルにIMAPを利用するよう設定を行う。

送信できたらプロバイダのメール・サーバーからメールを受信してみよう。送信がうまくいっていればテスト・メールが届いているはずだ。

## IMAPサーバーを設置しよう

Postfixを設置したことでメールの送受信が行えるようになった。次に，クライアントから閲覧できるように，IMAPサーバーを導入する。

Linuxでは表3のようなIMAPサーバーが利用できる。今回はFedora CoreやCentOSが標準装備している「Dovecot」を使うことにしよう。Dovecotのインストールは，

```
# yum install dovecot ↵
```

と実行すればよい。インストールが完了したら/etc/dovecot.confファイルに設定を行う（図7）。

ここでは，受け付けるプロトコルを

の際は行頭に付いている「#」を取り除く。設定を無効にしたい場合は行頭に「#」を付ければよい。

まず，「myhostname」に自身のホスト名，「mydomain」に自身が管理するドメイン名を設定する。DynDNSで登録したホスト名およびドメイン名を記載すればよい。

「myorigin」ではローカルからメールを送る際に「From:」の「@」以降に付加するドメインなどを指定する。ダイナミックDNSを利用している場合は，メール・アドレスにホスト名＋ドメイン名を使う必要があるので，「$myhostname」を指定しておく。

次にメール受信の制限を指定する。「inet_interfaces」ではどのあて先のメールを受信するかを指定する。初期設定では「localhost」が設定されているが，これでは外部からのメールは受け取れない。そこで，「all」を有効にする。また「localhost」は行頭に「#」を付加しておく。

「mydestination」では，このサーバーが最終転送となるメールを指定する。ここに記述していないドメインのメールを受け取ったら，他のメール転送サーバーに転送する。ここでは，DynDNSで登録したホスト名の「kazumail.homelinux.net」と記述すればよいので，「$myhostname, localhost」とする。

Postfixでは，踏み台にされないよう，外部のマシンから受信したメールを原則的に他に転送しない。ただしこれでは，メール・クライアントから受信したメールを転送しようとしても，拒否されてしまう。そこで，信頼できるネットワークを「mynetworks」に指定する。例えば，プライベート・ネットワーク（例えば，192.168.1.0から192.168.1.255）からのメール送信を受け付けるには「192.168.1.0/24」と記述する。ループバックを表す「127.0.0.0/8」も合わせて記述しておく。このIPアドレスから受け取ったメールは他に転送できる。

最後に受信したメールの保存方法を指定する。Postfixでは届いたメールを1つのファイルに保存するmbox方式と，各メールを個別のファイルに保存するMaildir方法がある。POPなどでメールをサーバーに保存しない場合は，mboxでも支障はないが，今回のように届いたメールをサーバー上に残す場合は，Maildirの方が向く。mboxを利用するとファイル内からメールを探し出す必要があるのに加え，1つのファイルを誤って削除すると，すべてが消えてしまうからだ。さらに，Maildirを利用すればツリー構造でメールを保存できる。

Maildir方式を利用するには「home_mailbox」を「Maildir/」にする。これで設定が完了だ。設定ファイルを保存したら，

```
# /etc/init.d/postfix re
start ↵
```

のようにPostfixを再起動して設定を有効にする。

```
myhostname = kazumail.homelinux.net          ←ホスト名を設定する
mydomain = kazumail.homelinux.net            ←ドメイン名を設定する。ダイナミック DNS の場合はホスト名と同じでもよい
myorigin = $myhostname                       ←ローカルから送信の際にメール・アドレスの「@」以降に付加するドメイン名またはホスト名
inet_interfaces = all                        ←メールの受信制限を設定する。all にするとどこからでも受信できる
#inet_interfaces = localhost                 ← all を設定したので，localhost は無効にしておく
mydestination = $myhostname, localhost       ←「To:」のメール・アドレス「@」以降がこれと同じなら，自身の担当のユーザーと判断する
#mydestination = $myhostname, localhost.$mydomain, localhost          ←上に設定したので無効にしておく
mynetworks = 192.168.1.0/24, 127.0.0.0/8     ←この IP アドレス以外のホストから届いたメールは他に転送できない
home_mailbox = Maildir/                      ←メールの保存方法を指定する
```

**図5　Postfixの設定**
/etc/postfix/main.cfファイルを編集する。図では変更が必要な行のみを記述した。

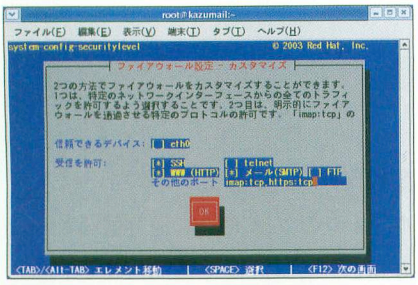


**写真2 ファイアウォールの設定**
外部からメール・サーバーやWebサーバー，IMAPサーバーに接続できるようにポートを開放する。

起動する。次に「カスタマイズ」を選択する。すると許可するサービスを設定する画面が表示される（写真2）。「受信を許可」で「WWW（HTTP）」，「メール（SMTP）」をチェックし，その他のポートに「imap:tcp,https:tcp」と記述する。これでツールを終了すると，指定したポートが開放される。

## メール・サーバーを設置しよう

実際にメール・サーバーを構築していこう。まず，初めにメールを送受信できるメール転送サーバー（MTA：Mail Tranfer Agent）をインストールする。表2のようなメール・サーバーがLinuxで利用可能である。

sendmailは古くからあるMTAで多くのディストリビューションで標準的に利用されている。しかし，古くからのソースを受け継いできたこともあってか，比較的頻繁にセキュリティ・ホールが見つかる。設定ファイルも複雑である。

一方，Postfixではセキュリティを重視して設計されている。設定ファイルもApacheなどのように比較的分かりやすい。

| 名称 | URL | 特徴 |
|---|---|---|
| sendmail | http://www.sendmail.org/ | 古くからあるMTA。設定方法が難しい |
| Postfix | http://www.postfix.org/ | セキュリティを考えて設計されている。設定は比較的簡単。sendmailのコマンドと互換がある |
| Exim | http://www.exim.org/ | 設定方法は比較的簡単。sendmailのコマンドと互換がある |
| qmail | http://www.postfix.org/ | 効率的でシンプルな構造に設計されている。sendmailのコマンドとは互換がない |

表2 Linuxで利用可能な主なメール転送サーバー（MTA）

```
# /usr/sbin/alternatives --config mta
2 プログラムがあり 'mta' を提供します。
  選択          コマンド
-----------------------------------------------
*+ 1            /usr/sbin/sendmail.sendmail
   2            /usr/sbin/sendmail.postfix
Enter を押して現在の選択 [+] を保持するか、選択番号を入力します: 2
```

**図4 sendmailからPostfixにMTAを切り替える**
MTAの切り替えにはalternativesコマンドを実行し，sendmail.postfixを選択する。図の場合は「2」と入力する。

このような理由から，MTAにはPostfixを利用することにする。

### Postfixをインストールする

Fedora CoreやCentOSではMTAとしてsendmailが標準インストールされる。そこで，Postfixを導入し，通常利用するMTAの切り替えを行う。Fedora Core，CentOSのいずれもyumを利用してインストールが行える。インストールは，

```
# yum install postfix
```

と実行すればよい。しばらくするとインストールが完了する。

しかし，インストールをしただけではPostfixを利用するように切り替わらず，sendmailでメールの処理が行われてしまう。そこで，alternativesコマンドを利用してPostfixに切り替える（図4）。すると，どちらのMTAを利用するか聞かれるので，sendmail.postfixの番号を選択する。

次に，sendmailを停止し，Postfixを起動する。

```
# /etc/init.d/sendmail stop
# /etc/init.d/postfix start
```

これで，Postfixのインストールが完了した。

次に/etc/postfix/main.cfファイルでPostfixの設定を行う（図5）。設定

＊1 ddclientで更新を行う場合は，DynDNSサービスからログアウトしておく。
＊2 システムを再起動をせずにネットワークを再起動すると，突然ホスト名が変更されてしまうため，現在動作していたアプリケーションが停止してしまうことがある。そのため，ホスト名を変更したら再起動した方が安全だ。

```
127.0.0.1       localhost.localdomain localhost
127.0.0.1       kazumail.homelinux.net kazumail     ← 追加する
```

**図3　/etc/hostsファイルにIPアドレスとホスト名の関連を記載する**
/etc/hostsにIPアドレスとホスト名の関連を記載しておくことで，サーバーのIPアドレスをループバック・アドレスと関連付けられる。

| サービス名 | プロトコル名（ポート番号） |
|---|---|
| メール・サーバー | smtp（25） |
| Webサーバー | http（80），https（443） |
| IMAPサーバー | imap（143） |

**表1　開放するポート**
それぞれのサービスを外部から利用するにはポートを開放する必要がある。

DNSサーバーのMXレコードに登録するメール・サーバーを知らせるのが，「mx」である。メール・サーバーのホスト名を指定しておく。登録したホスト名は「server」項目の最後の行にも記述する（図2の最下行）。

設定が完了したら，ddclientを起動する*1。

```
# /etc/init.d/ddclinet
start
```

これで，IPアドレスが定期的に更新されるようになったほか，「xxx@kazumail.homelinux.net」あてのメールが自宅メール・サーバーに配送されるようになった。

ダイナミックDNSに登録したら，メール・サーバー上での設定もそのホスト名に合わせて変更しておくとよい。ホスト名の変更は，/etc/sysconfig/networkファイルの「HOSTNAME」に設定する。例えば，kazumail.homelinux.netならば，

```
HOSTNAME=kazumail.homeli
nux.net
```

と書き換えればよい。

内部LANでは，サーバーで利用されるIPアドレスがプライベートIPアドレスであるため，DynDNSで配布されるIPアドレスと実際のIPアドレスが異なる。このままだと，アプリケーションによっては正常に動作できなくなってしまう。そこで，サーバー内で利用するホスト名とIPアドレスの関係を/etc/hostsファイルに記載しておく。自分自身を表すループバック・アドレスとDynDNSで登録したホスト名を関連付けるだけだ。例えば，kazumail.homelinux.netならば，図3のように追加する。

設定が完了したらシステムを再起動しよう。すると，ホスト名が変更される*2。

ブロードバンド接続サービスを利用している場合は，ブロードバンド・ルーターによって内部のマシンに外部から接続できない場合が多い。ブロードバンド接続の場合，1つのグローバルIPアドレスを複数のマシンで共有するNAPTを利用している。そのため，メールがルーターを通過しようとしても，どのマシンにメールを送ってよいか分からない。セキュリティの面でも，外部から内部のマシンに接続できるのは良くない。

そこで，メール・サーバーにメールが届くよう，ブロードバンド・ルーターの設定を変更しておく必要がある。ルーターの設定では，ポート・フォワーディングやNAPT，アドレス変換などの名称を用いることが多い。外部からSMTPのポートである25ポートに届いた場合，メール・サーバーのプライベートIPアドレスに転送するように設定しておこう。

設定方法はブロードバンド・ルーターによって異なるので，マニュアルなどを参照してほしい。

これだけではまだ，外部からメールを読めない。Fedora CoreやCentOSではファイアウオール機能が働いており，初期状態ではSSH以外のプロトコルは受け付けないようになっている。今回のサーバーのようにメール・サーバー，IMAPサーバー，Webサーバーを利用する場合はそれぞれのポートを開放しておく必要がある。ポート番号は表1のようになっている。これらのポートを開放する。

ファイアウオールの設定はグラフィカルな設定ツールの「system-config-securitylevel」および，コマンド「system-config-securitylevel-tui」が利用できる。コマンドを利用する場合は，

```
# su
# /usr/bin/system-con
fig-securitylevel-tui
```

と実行すると，CUIベースのツールが

ダイナミックDNSサービスは，IPアドレスの変化を検知してくれない。そのままにしておくと，IPアドレスが変更された途端，メールを一切受け取れなくなってしまう。

そこで，IPアドレスが変わったらダイナミックDNSサービスに新しいIPアドレスを知らせるようにする。IPアドレスの変化を検知してダイナミックDNSに伝えるソフトが，ダイナミックDNSクライアントだ。常時メール・サーバー上で稼働させておく。

Linux向けのダイナミックDNSクライアントはいくつか存在するが，今回は「ddclient」を利用することにする。ddclientはhttp://ddclient.sourceforge.net/から入手できる。入手したら，図1のようにインストールする。

次に設定を変更する。管理者権限で/etc/ddclient/ddclient.confファイルをテキスト・エディタで開き，図2のように編集する。サンプルの設定が記述されているので，この中から必要な設定を書き換えればよい。有効にする設定は，行頭にある「#」マークを取り除く。

「use」には現在のIPアドレスを取得する方法を選択する。「use=web」とすると，DynDNSで公開されているCGIでIPアドレスを取得する。「login」と「password」にはDynDNSで登録したユーザー名とパスワードを入力する。

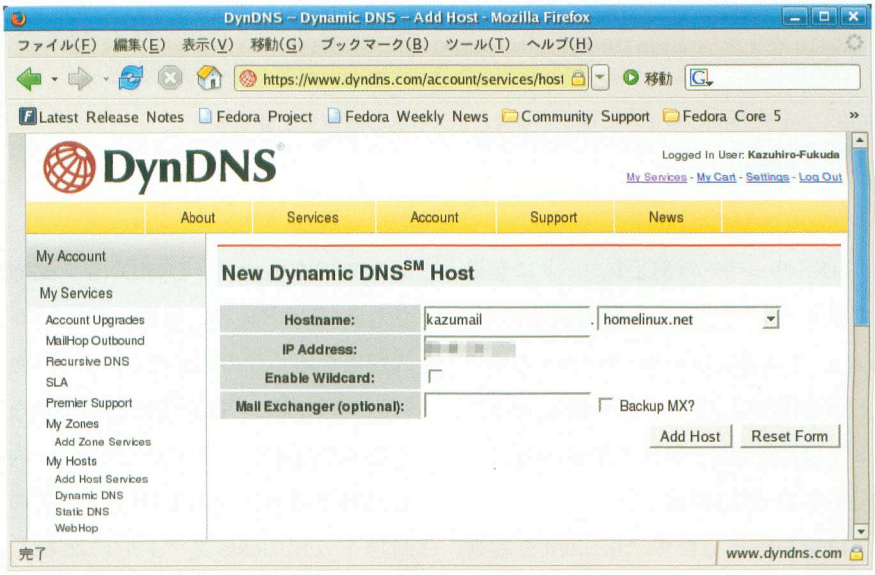


**写真1　ダイナミックDNSの登録**
ホスト名を入力し，利用したいドメインを一覧から選択する。

```
$ tar zxvf ddclient-3.7.0.tar.gz ↵
$ cd ddclient-3.7.0 ↵
$ su ↵
# cp ddclient /usr/sbin ↵
# mkdir /etc/ddclient ↵
# cp sample-etc_ddclient.conf /etc/ddclient/ddclient.conf ↵
# chmod 600 /etc/ddclient/ddclient.conf ↵
# cp sample-etc_rc.d_init.d_ddclient.redhat /etc/init.d/ddclient ↵
# /sbin/chkconfig --add ddclient ↵
```

**図1　ddclientのインストール手順**

↵このマークで改行

```
deamon=300                    指定した秒数後にIPアドレスを確認する
syslog=yes                    システム・ログに保存する
mail=root                     ddclientが出力したメッセージの送信先
mail-failure=root             アップデートができない場合のメッセージの送信先
pid=/var/run/ddclient.pid     プロセス番号を登録する
# ssl=yes                     SSLを利用した通信を利用しないようにする
      :
use=web, web=checkip.dyndns.org/, web-skip='IP Address'
                              IPアドレスの取得方法を記述する。webの場合は
                              DynDNSで公開しているCGIを利用して取得する
      :
login=Kazuhiro-Fukuda         DynDNSのユーザー名を入力する
password=kazupasswd           DynDNSのパスワードを入力する
mx=kazumail.homelinux.net     DNSサーバーに登録するメール・サーバーはここで指定
backupmx=yes                  バックアップ用メール・サーバーを利用するか設定する
wildcard=yes                  ワイルド・カード表記を許す
      :
server=members.dyndns.org, \
protocol=dyndns2, \
kazumail.homelinux.net        取得したホスト名およびドメイン名を記述する
```

**図2　ddclientの設定**
コピーした/etc/ddclient/ddclient.confファイルを編集する。図は変更した行のみ記述した。

# PART 2
# 実践編 ～自宅にメール・サーバーを導入する～

**いよいよメール・サーバーを導入する。メール転送サーバーには「Postfix」を用いる。ウイルスやスパムも除去し，インターネットを介してWebでメールが送信できるようにする。悪意のある第三者から踏み台にされない強固なサーバーを目指す。**

Part 1で見たように，メール・サーバーを立ち上げるためには，DNSサーバーにメール・サーバーを登録する必要がある。しかし個人ユーザーの場合は，独自のドメインを登録してDNSサーバーを設置することはあまりないだろう。プロバイダから固定IPアドレスをもらわなければならないし，DNSサーバーをレンタルするか設置するとなると，運用コストがかさんでしまう。

そこで，「ダイナミックDNS」を利用するとよい。ダイナミックDNSとは，IPアドレスが変わっても同じドメインを使い続けられるようにしたDNSサービスである。多くは無償で利用できる。このDNSサーバーに，自分が立ち上げたメール・サーバーを登録するわけだ。

その仕組みはこうだ。メール・サーバーのIPアドレスが変化したときには，すぐにダイナミックDNSサーバーに新しいIPアドレスを伝える。すると，ダイナミックDNSサーバーは，ドメインに相対するIPアドレスの情報を更新する。これで，IPアドレスが変わる環境でも，専用ドメインを持てるし，そのドメインをメール・アドレスに使えるわけだ。

では，ダイナミックDNSサービスにメール・サーバーを登録してみよう。

現在，多くのダイナミックDNSサービスが存在する。今回は，世界的によく利用されている「DynDNS」を使ってみる。

まず，DynDNSのWebページ（http://www.dyndns.com/）にアクセスする。Webページが表示されたら，右上にある「Sign Up Now」リンクをクリックする。すると，ユーザーの情報を入力する画面が表示される。入力が必要なのは「Username」，「Email Address」，「Password」だ。「Acceptable Use Policy」にある2つのチェック・ボックスにもチェックを入れておく。

入力が完了したら「Create Account」ボタンをクリックする。すると，しばらくして，登録したメール・アドレスあてに確認のメールが届く。メール内に記述されているURLにアクセスすると登録完了だ。DynDNSの画面右上にある「User」と「Pass」に登録したユーザー名とパスワードを入力すると，サービスにログインできる。

続いてダイナミックDNSサービスにドメイン名などを登録する。DynDNSの画面右にある「My Account」をクリックし，続いて「My Services」-「My Host」-「Add Host Services」の順にクリックする。

次に，「Add Host Services」にある「Add Dynamic DNS Host」をクリックすると，ダイナミックDNSの登録画面が表示される（写真1）。Host nameで利用したいホスト名およびドメインを選択する。dyndns.orgやhomelinux.netなど70近いドメインから選択できる。IPアドレスにはホストのIPアドレスが自動的に入力される。「Add Host」ボタンをクリックするとダイナミックDNSへの登録が完了する。

これで，インターネットからホスト名およびドメインで自宅のホストに接続できるようになる。メール・サーバーの登録方法は後述する。

### 定期的にIPアドレスを変更

ADSLやFTTHを使ってインターネットに常時接続していても，プロバイダから割り当てられているIPアドレスは変わる場合がある。こうしたサービスでは，PPPoEなどのプロトコルでIPアドレスを動的に取得する。パソコンまたはルーターの電源を投入するたびにIPアドレスを取得するが，毎回同じIPアドレスがアサインされるとは限らない。

ており，それぞれの階層でそのドメインを管理するDNSサーバーが存在する（図4）。

最上位のDNSサーバーを「ルートDNS」と呼ぶ。その下に，「jp」や「com」など「トップ・レベル・ドメイン」といわれるドメインを管理しているDNSサーバーがある。ルートDNSは，こうしたトップ・レベル・ドメインを管理しているDNSサーバーのIPアドレスを知っている。また，トップ・レベル・ドメインを管理するDNSサーバーは，セカンド・レベル・ドメインである「co」や「ne」を管理するDNSサーバーのIPアドレスを所持している。さらに「co」などのDNSサーバーが，「nikkeibp」といった企業や団体のDNSサーバーのIPアドレスを管理している。

fukuda@xxxx.co.jpに関する情報を知りたいときは，xxxxを管理するDNSサーバーに問い合わせればよい。xxxxを管理するDNSサーバーを見つけるには，ルートDNSに最初に問い合わせる。そうすると上の階層からたらい回しにされて，xxxxのDNSサーバーにたどり着ける。

DNSではメール用に「メール・エクスチェンジャ（MX）」と呼ぶ機能が用意されている。MXはDNSにIPアドレスなどを登録する「レコード」といわれる書式の1種である。

DNSサーバーのMXレコードにドメインとメール転送サーバーのホスト名の対応関係を登録しておき，メール・サーバーから問い合わせがあった場合に，該当ドメインに対するメール転送サーバーのホスト名を返す。

例えば，「xxxxx@nikkeibp.co.jp」にメールを送るとき，メール転送サーバーは，「nikkeibp」ドメインのDNSサーバーに「nikkeibp.co.jp」のMXレコードを要求する。すると，DNSサーバーはMXレコードに保存されているメール転送サーバーのホスト名「bpns1.nikkeibp.co.jp」を返してくる。メール転送サーバーはこのホスト名を利用して，送信先のメール・サーバーのIPアドレスを割り出せる（図5）。

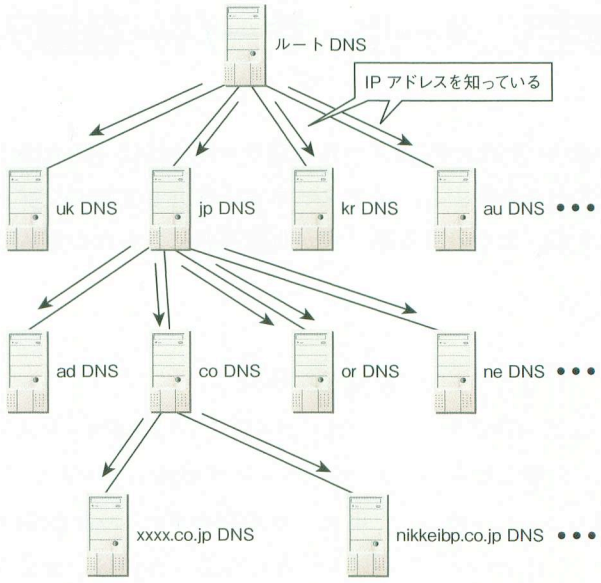


**図4 DNSサーバーの階層構造**
上から順々に問い合わせていけば目的のホストが探し出せる。

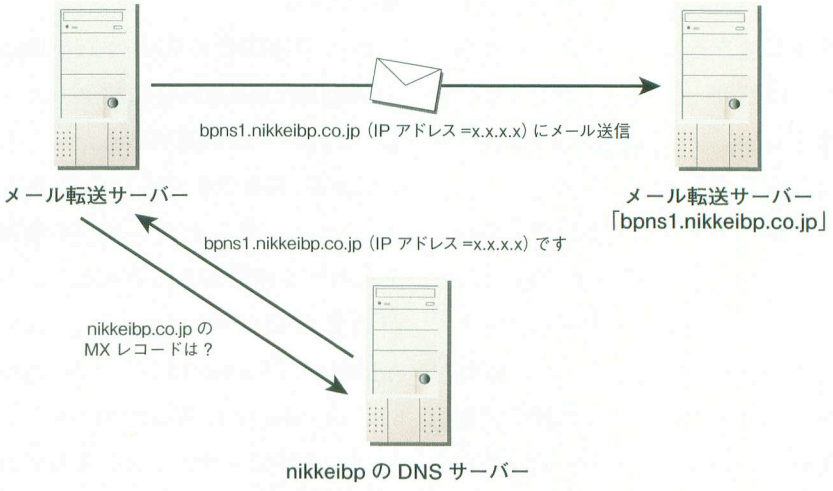


**図5 メール・アドレスからのメール送信方法**
メール・アドレスにドメインのみが記載されている場合は，DNSサーバーにMXレコードを問い合わせて，目的のメール・サーバーを知る。

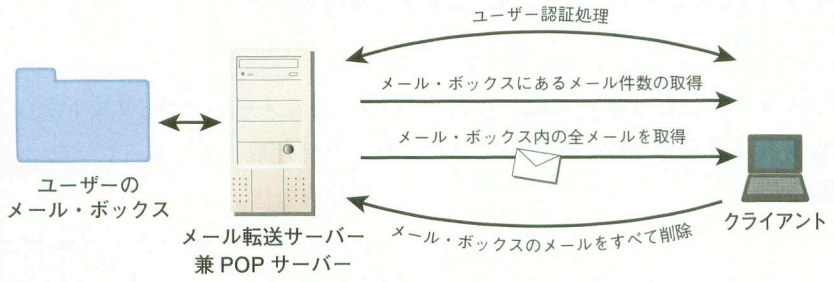


**図2　POPサーバーを利用したメールの取得**
POPサーバーを利用した場合，通常は全メールをメール・クライアントに取り込む。

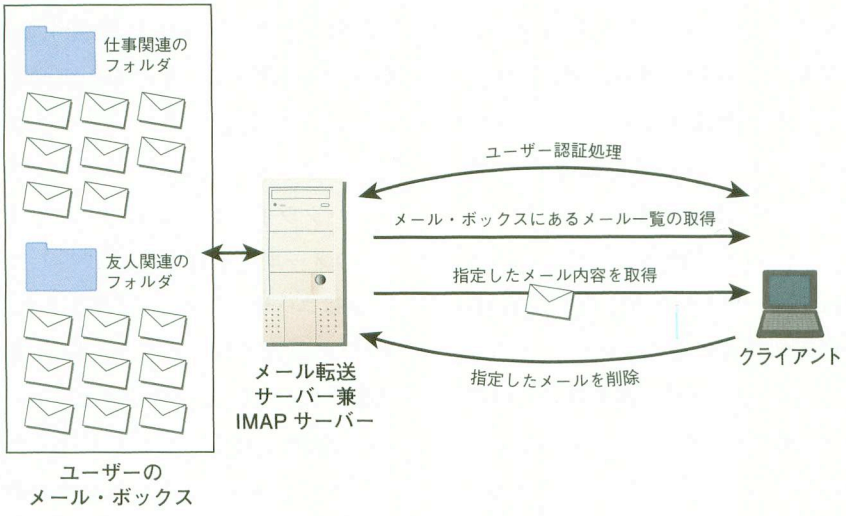


**図3　IMAPサーバーを利用したメールの閲覧**
IMAPサーバーではサーバー上でメールを管理する。メール・クライアントはIMAPサーバーに接続したままの状態でメールの操作を行う。

プロバイダのように多くのユーザーを抱える場合に利用される。

一方，IMAPサーバーは届いたメールをサーバー上で管理する。主に次のような管理機能がある。

- **メール・ボックス内のメールの一覧表示**
- **メール内容の閲覧**
- **メールの削除**
- **メール・ボックスの階層化**
- **メールの検索**
- **既読，返信済みなどフラグの付加**

IMAPサーバーでは図4のようになっている。メール・クライアントからIMAPサーバーに接続すると，ユーザー認証が行われる。その後はIMAPサーバーとの接続を保ったまま，サーバー上のフォルダ内にあるメールの一覧を取得したり，メール内容を閲覧したり，新しいフォルダを作成したり，といった操作を行う。どこからでも閲覧できるし，メール・ボックスを階層化できるメリットがある。

欠点は，過去のメールを見るときにもサーバーへのアクセスが必要になることだ。サーバーがダウンしているときにはメールが読めないし，配下のユーザーが多い場合は，サーバーの処理性能によってはクライアント側の操作が遅くなることもある。そのため，ユーザーが多い場合は向かない。

## 送り先メール・サーバーの見つけ方

次に，あて先のメール・アドレスからどのようにして送信先のメール転送サーバーを決めるのかを説明する。

インターネットに接続されているサーバーやパソコンにはIPアドレスと呼ばれる数字が付与されている。例えば，「192.168.0.1」といったものだ。SMTPでメールを送るときは，送り先のメール転送サーバーのIPアドレスを見つける必要がある。クライアントから送るときは通常，メール・アドレスには関係なく，メール転送サーバーのホスト名またはIPアドレスをあらかじめ設定しておく。

メール転送サーバーが別のメール転送サーバーに送るときは，メール・アドレスを基に見つける。

これには，DNS（Domain Name System）という仕組みを用いる。

### DNSを利用したIPアドレスの取得

メール・アドレス「fukuda@xxxx.co.jp」の「xxxx」や「co」をドメインと呼ぶ。ドメインは階層構造になっ

# PART 1
# 学習編 ～メールとDNSの仕組みをおさらいしよう～

**メール・サーバーを導入する前に，そもそもメールはどうやって配信されるのかをおさらいしておこう。メールが配信される仕組みやDNSを使ってサーバーを見つけるメカニズムを，初歩から解説する。**

メール・サーバーを個人で設置すれば，出先からメールをインターネット経由で送受信したり，届いた不要なメールを取り除くといったことが可能になる。

Linuxディストリビューションの多くは，メール・サーバーのパッケージを標準装備する。以下では，Fedora Core 5またはCentOS 4.3を利用して，自宅にメール・サーバーを構築する手順を紹介する。今回設置するサーバーは，左ページのイラストのような機能を持つ。

## メール送受信の仕組み

最初に，メールが送受信される仕組みを理解しよう。

メール・クライアントはSMTP（Simple Mail Transfer Protocol）と呼ぶプロトコルを利用して，メール・サーバーにメールを受け渡す（図1）。メール・サーバーはユーザーからメールを受け取ると，メール・アドレスを見て，相手のユーザーが所属するメール・サーバーを見つけ，そのサーバーにSMTPでメールを転送する。このように，メールを受け取って転送するサーバーを「メール転送サーバー」または「MTA」（Message Transfer Agent）と呼ぶ。

メール転送サーバーが他のメール転送サーバーからメールを受信すると，「To：」のメール・アドレスの「@」以降を見て，自分が管理するアドレスかどうかを確認する。もし，自分の担当アドレスでなければ他のメール転送サーバーに転送する。

もし，自分の担当であるにもかかわらずユーザーが存在しない場合は，送信元にエラーを返したり，場合によっては何もせずに破棄する。目的のユーザーが存在すると，そのユーザーのメール・ボックスにメールを保存する。メール転送サーバーとなっているマシンに直接ログインできるユーザーは，メール・ボックスを閲覧することで届いたメールの内容を見ることができる。

メール転送サーバーにログインする手段を持たないユーザーはメール・クライアントを使う。メール・クライアントを用いて，ネットワーク越しにメール転送サーバーのメール・ボックスからメールをダウンロードする。

ダウンロードの際に用いるのが，POP（Post Office Protocol）やIMAP（Internet Message Access Protocol）といったプロトコルだ。POPやIMAPサーバーはメール・クライアントからの要求を受け取ると，メール・ボックスの内容を返信する。

POPサーバーとIMAPサーバーではメールの取り扱いに違いがある。POPの場合は，認証に成功すると，メール・ボックスにあるメールのサイズや件数をクライアントに送信する。POPの仕組みでは，その中から必要なメールだけを選んでメッセージ本体をダウンロードできるが，ほとんどのメール・クライアントは届いているすべてのメールを取得する。取得したメールはユーザーのマシンに保存され，いつでも閲覧できるようになる（P.58の図2）。メール・サーバー側は，メールを一時的に保存しておくだけでよいので，メール・ボックスの容量を小さくしても影響は少ない。そのため，

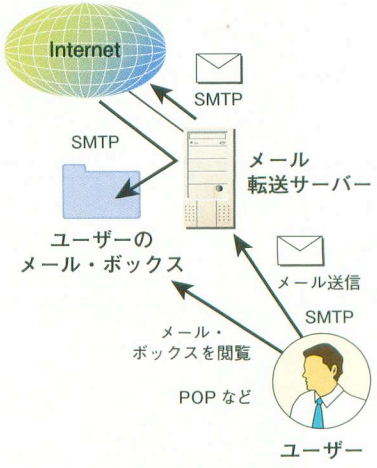


**図1　メール送受信の仕組み**
メール転送サーバーはメールを受信すると，ユーザーのメール・ボックスに保存する。