

メールで遊ぼう

松本徳真@lilo (matsu@netfort.gr.jp)

Mar. 28, 2003

概要

GUIを使ったメールソフトしか知らない人を対象に、インターネットメールの面白さを伝えることを目標にします。

「メールはどのように届けられるのか?」の概略を理解し、「メールを送信/受信する方法」を学び、「ShellScript, Ruby を使ったメールアプリケーション」を作成します。

本書で扱うのは、メールサーバの管理者のための知識ではありません。主にユーザーとして役立つ知識を扱います。

cgi でメール出したいどうすれば良いの?、メールが届くと自動的に返事をするようにしたい!、等々。Shell, ruby, perl, C 等なんらかの言語でプログラムを作成できるとより楽しめます。

1 はじめに

タイトルで、「メールで遊ぼう」と書いといてこういうのもなんですが、インターネットの世界での失敗は、洒落にならない結果をもたらします。まずその点をしっかり肝に銘じておいて下さい。

最初はインターネット (The Internet) から隔離したマシンで、練習しましょう。Linux 等の UNIX 的な OS のインストールされたマシンが一台あれば十分に遊べるでしょう。2, 3 台のマシンをネットワークで接続すれば色々な実験が出来て良い勉強になります。また多少高度な知識が要求されるかもしれませんが、VMware, UML (User Mode Linux) 等を利用できれば、一台の Linux マシンで、複数のマシンからなるネットワークを仮想的に構築できます。利用できる方は、是非活用して下さい。

実験環境で自信がつけば、家庭内ネットワークとか職場のイントラネット環境に応用してみましよう。その際、ネットワーク管理者と良好な関係を結び、協力が得られるように努力しましょう。

本書で紹介する内容を実際に役立てようとするならば、若干のプログラミングの知識と、UNIX 的物の考え方が要求されるでしょう。これは、書籍だけではなかなか得られない種類の物です。もし身近に UNIX グルがいるならば、是非有効に活用しましょう。普通は、そういう便利な人があるとは思えないので、各種ユーザグループに参加する事を検討してください。例えば <http://lilo.linux.or.jp/> .

これらの経験で得られるスキルは、インターネットの世界を生きて行く上できっと役に立つでしょう。

2 電子メールはどのように配達されるの?

電子メールは、MTA (Mail Transfer Agent) から、MTA にメールを転送しながら、最後にあなたのメールアカウントのある MTA に到着します (図 1)。MTA は、電子メールを支える一番大切なものです。MTA には多くの種類がありますが、sendmail, postfix, qmail 等が特に有名です。MTA から MTA へのメールの転送は、SMTP, ESMTP 等のプロトコルで転送します。

普段、メールの送受信に使用するアプリケーションプログラムを MUA (Mail User Agent) といいます。このプログラムは、メールの送受信のお手伝いをします。メールを送信するときは、ユーザーの書いた文書に、Date:, Message-ID: 等必要な情報を追加したり、バイナリファイルの MIME エンコードをしたりし

た後, SMTP プロトコルを使用して MTA にメールを届けます. Outlook Express, Sylpheed, Mew, mutt, Eudra, 等々たくさんの種類が存在します.

図 1 では, A という PC で書かれたメールを E という PC で読む場合を示しています. ここで, 矢印はメールの転送される方向ではなく, アクセスの方向を示していることに注意してください. サーバ/クライアントの関係でいうと, クライアントからサーバへ矢印を向けています.

ここで, MTA は, SMTP サーバと SMTP クライアントの機能を持たなければならないことがわかるでしょう. 一方 MUA は一般に, SMTP クライアントの機能しか持ちません.

矢印に従って行くと, A を出発したメールは, B, C の MTA を経由して, D の受信箱¹ が終点になります. D から, MTA の動作していない E の PC にメールが転送されることはありません. E から D へ接続して, なんらかの方法で受信箱を見なければなりません. 昔は, E から D へ telnet 等で login して受信箱を読んでいましたが, 最近では POP, IMAP 等と呼ばれるプロトコルを使って受信箱に届いたメールを取得することが一般的です. E で動作して, D の受信箱を読みに行くのも MUA の仕事です. D の受信箱は, 現実の郵便で言うところの私書箱のようなものです.

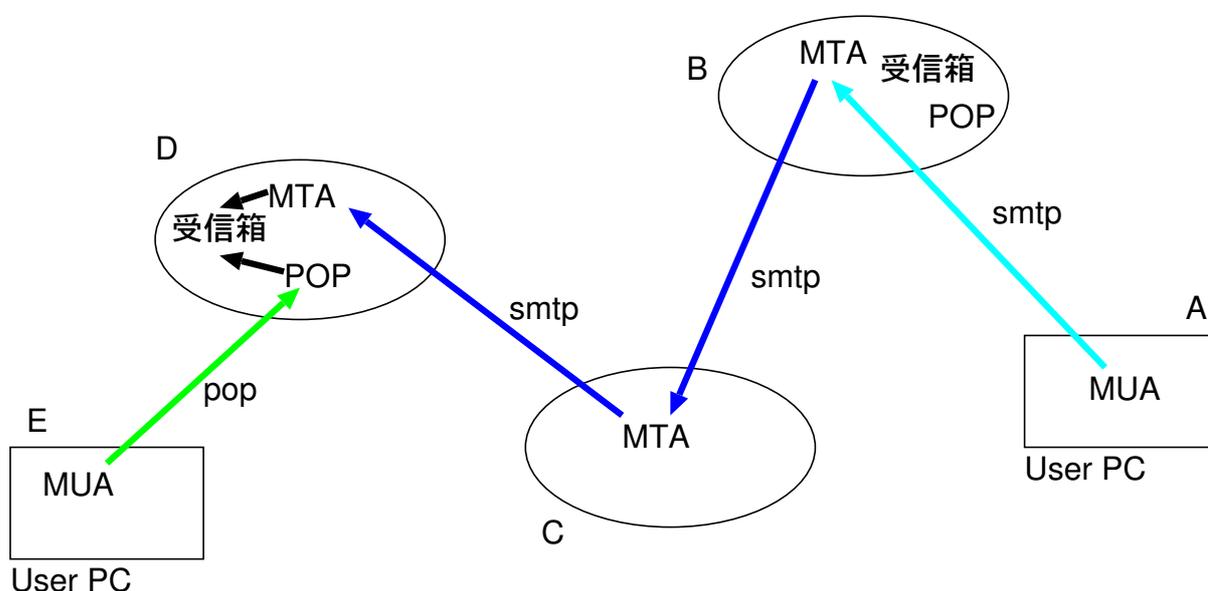


図 1: メール配送の概念

図 1 で, C のマシンはちょっと見た感じ無駄に見えます. 昔々, UUCP を使ってメールをパケツリレーしていた時代には多くの MTA を経由してメールが配送されていました. その後, 多くの MTA が TCP/IP による常時接続をするようになってからは, かなりシンプルになったのですが, その後セキュリティ上の要請から再び多くの MTA を介するようになりました.

C の MTA は, FireWall で守られた D への窓口になっているかも知れませんが, メールを介して繁殖するウイルスチェックの機能を持ったマシンかも知れません.

太古の昔, メールを送受信は, 図 1 の, B のマシンに login してメールを送信して, D のマシンに login して受信するのが普通でした. その際利用したのが “mail” コマンドです.

今回は, この B, D のマシンでメールを送受信する方法を説明します. 単にメールを送受信するだけなら, 現在では便利なメールプログラム (MUA) がたくさん存在するのに意味ないやと言わずに最後までおつき合い下さい.

¹ mailbox が正しい名前? sendmail 関連の和訳本ではメールボックスなどと書かれている.

3 MTA のお仕事

さて、本題に入るまえに、もう少し詳しく MTA の仕事を見てみましょう。図 2 は、図 1 のマシン D をさらに拡大してみたものです。

MTA は、メールを受け入れる機能と、メールを送り出す機能を持っています。受け入れ側としては、SMTP サーバとして、ネットワーク越しにメールを受け付けるほか、同じマシン内からならば、直接メールを受け付ける事も出来ます (ただし全ての MTA がこの機能を持っているとは限らない)。

なんらかの方法で受け付けたメールは、まず自分のユーザ宛てかどうかチェックします。もし他の MTA のユーザ宛てならば、どの MTA に送れば良いか DNS(Domain Name System) サーバに問い合わせ、その結果に従います。他の MTA から無関係な MTA 宛てのメールを受け付けた場合は受け取りを拒否するかもしれません。

もし、自分のユーザ宛てのメールならば、“alias” “.forward(.qmail)” の指示に従います。“alias” “.forward(.qmail)” によって、受信箱への保存、転送、コマンドの実行等が可能です。

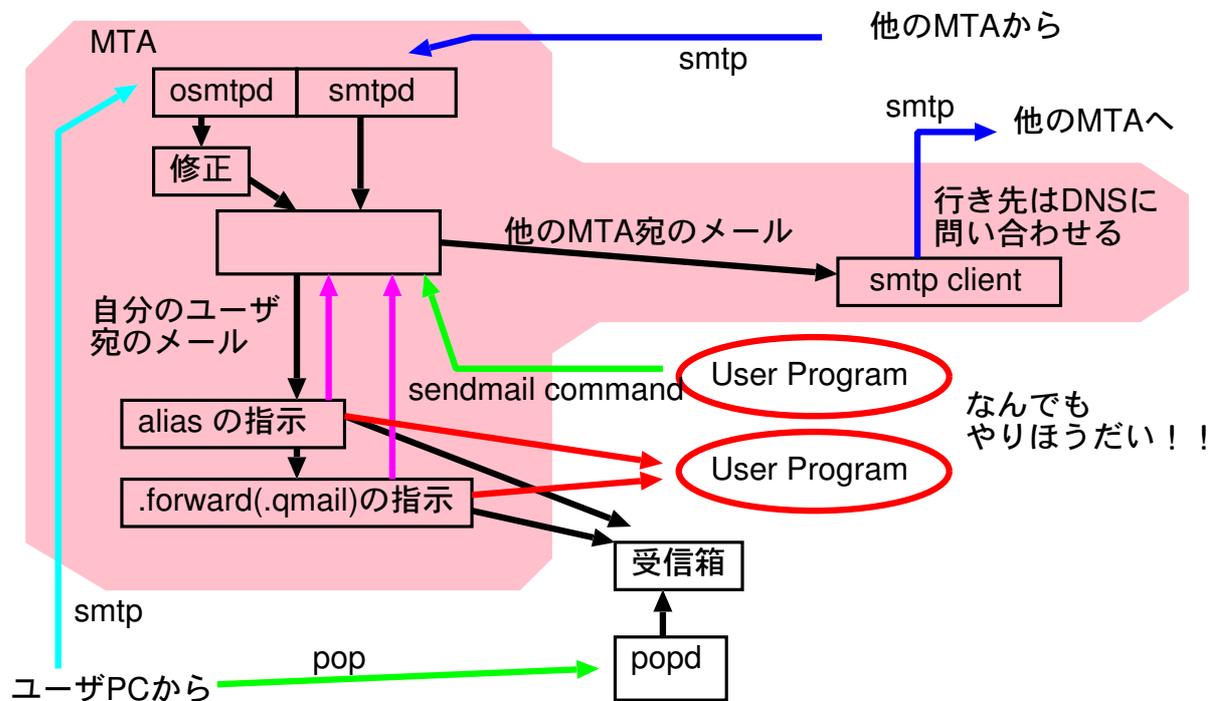


図 2: MTA の仕事

本書では、既に MTA が動作しているとして、ユーザー権限で実行できる内容として、mail(または sendmail) コマンドを使ったメールの送信、.forward を使用した受信メールの処理を扱います。

4 .forward

先に説明したとおり、多くの MTA は、自分のユーザ宛てのメールを受け取ると alias, .forward の指示に従って処理を行います。

.forward の場合、各ユーザがそのユーザ宛てのメールの配送制御を行う事が出来ます。

alias は、.forward と同様の機能を持ちます。通常 root 権限を必要としますが、.forward と異なり、全て

のユーザの配送制御を行う事ができます。また、設定の仕方は、MTA に強く依存します。alias の設定は、MTA の管理者 (つまりメールサーバの管理者) の仕事ですのでここでは扱いません。

ここから、イメージを得やすいよう具体的な名前を出して説明します。MTA の動作しているマシンは仮りに、viper2.netfort.gr.jp とします。ユーザ名は、matsu です。そして、メールアドレスは matsu@netfort.gr.jp になります。

どこかの MTA が、matsu@netfort.gr.jp にメールを送りたいと考えたとき DNS の MX レコードを調べます。例えば shell 上で以下のコマンド² を実行する事で調べることが出来ます。

```
$ host -t mx netfort.gr.jp
netfort.gr.jp mail is handled by 5 viper2.netfort.gr.jp.
netfort.gr.jp mail is handled by 10 decaton.netfort.gr.jp.
```

この結果から、まず最初に viper2.netfort.gr.jp の MTA にメールを転送したら良いとわかるわけです。

さて、viper2.netfort.gr.jp は、matsu@netfort.gr.jp 宛てのメールを自分のユーザ宛てのメールとして受け取りました。実際に@netfort.gr.jp のついたメールアドレスは全て、viper2.netfort.gr.jp の管理下にあったのです。viper2.netfort.gr.jp は、まず alias データベースのチェックを行いました。matsu@netfort.gr.jp に関する記述はありませんでした。次に /etc/passwd を調べると、幸い matsu というユーザが存在しました。そこで、~matsu/.forward³ を調べてその指示に従うことにしました.... と、こんな感じで続いて行くわけです。sendmail, postfix では、~matsu/.forward を見に行きます。qmail の場合は、~matsu/.qmail によって制御されます。

それでは、これから .forward(.qmail) を記述してみましょう。

4.1 受信箱への配送

一番の基本は受信箱への配送です。viper2.netfort.gr.jp では、POP サーバが動作していますので、他のマシン (Windows, MacOS, Linux なんでも良い) で動作する MUA を使い POP 経由でメールを読むことが出来ます。この POP サーバが扱うのは受信箱に届いたメールですので、ここに配送しない限り、POP 経由でのメールの読み込みは出来ないことになります。

ところで、今まで特に断りなく受信箱という呼び方を続けてきましたが、その実態はただのファイルです。その形式は、MTA の種類によって異なります。sendmail で伝統的に使われてきたのは、通常 /var/mail/<user name> というファイルです。ユーザ名が matsu ならば /var/mail/matsu となります。メールが届く度に、このファイルに追加されて行きます。この受信箱は mbox 形式とも呼ばれています。qmail では、Maildir が強く推奨されています。ユーザ名が matsu の場合、~matsu/Maildir/new/ の下に、一メールファイルとして保存されます。sendmail 互換の mbox 形式も可能です。postfix では、sendmail 形式 (mbox), qmail 形式 (Maildir) どちらも使用できます。

さて、forard に戻ります。sendmail の場合、~/forward⁴ に次のように記述します。

² host コマンドの他に dig も使用できます。djbdns をインストールしているなら dnsmx を使用できます。従来使用されて来た nslookup は使用すべきではありません。

³ ~matsu という表記は、ユーザ matsu のホームディレクトリを意味します。多くの場合 /home/matsu 等を意味します。

⁴ ここで、~ という記述は、ホームディレクトリを示します。matsu で login しているならば、先ほどの ~matsu と同じ場所を指します。

```
matsu
```

ところで, sendmail の場合, .forward が存在しなければ, 上記記述をした場合と同様, 受信箱に配送されます。

postfix の場合は, /var/mail/matsu に sendmail 互換の形式で配送したい時は,

```
/var/mail/matsu
```

qmail 互換の Maildir 形式にしたい時は,

```
~/Maildir/
```

となります。もし POP サーバを使用する時は, POP サーバがどの形式の受信箱を前提にしているかをサーバ管理者によく確認して選択しないと, POP 経由でメールが読めないと言う事態になります。

qmail の場合は ~/.qmail に,

```
./Maildir/
```

と記述してください。 .qmail の処理時には, この場合ですとユーザ matsu のホームディレクトリ ~/matsu に cd されていますのでこんな感じです。ところで, qmail の場合も postfix 同様どの形式の受信箱を使用すべきかサーバ管理者に確認する必要があるわけですが, qmail を使うような管理者が mbox 形式を使う訳が無いので上記設定以外はずり無いです。

4.2 メールの転送

そもそも “forward” という名前が付いているくらいですから, 元々はメールの転送が本来の目的だったのでしょう。

メールを別のアドレスに転送する指定はとても簡単で, メールアドレスをただ書くだけです。 .qmail の場合も同じです。

例えば matsu@keitai.jp というアドレスにメールを転送したいなら,

```
matsu@keitai.jp
```

sendmail を例にして, matsu@myhome.jp, matsu@myoffice.jp に転送して, なおかつ POP 経由でメールが読めるように, 受信箱にもメールを残したいなら,

```
matsu  
matsu@myhome.jp  
matsu@myoffice.jp
```

とても、簡単ですね。postfix, qmail の場合、最初の一行だけ、受信箱への配送を参考に変更してください。ところで、もし matsu@myhome.jp の方で、matsu@netfort.gr.jp への転送が指定されていればどうなるでしょうか？ とても恐ろしいことが起きることが想像できるでしょう。実は最近の MTA では、このようなメールループと呼ばれる事故を防ぐための工夫が施されているので、ちょっとした失敗なら防ぐことが出来るので、たぶん大事に至ることは無いと思います。しかし用心に越したことはありません。十分に注意してください。

特にメーリングリストの投稿用アドレス等は、.forward には記述しないようにして下さい。メールの事故が発生した際に、多くの人に迷惑を掛けることとなります。

最近では、インターネットメールを扱うことの出来る携帯電話が、普及しています。そこで携帯電話のメールアドレスに転送すると、すぐにメールが読めて便利です。そのための設定はこれまでの説明で既に理解されたことと思いますが、これも実は色々と問題があります。後でまた詳しく触れたいと思います。

4.3 コマンドの実行

.forward(.qmail) の指定でコマンドを実行することが出来ます。この機能を使いこなせると、色々便利なこと、楽しいこと、邪悪な事が出来るようになります(邪悪な事はしないでね)。

ところで、lilo-ml(lilo メーリングリスト)には既に参加していますか？ lilo-ml 参加時には lilo-ctl@lilo.linux.or.jp に subscribe Norimasa Matsumoto 等と書いたメールを出したはずです。lilo-ctl@lilo.linux.or.jp に help 等と書いたメールを出した事の有る方もいるでしょう。これらのメールの返事は誰が書いているのでしょうか？ ml 管理者？ そんな訳は無いですよ。この節を理解すればそのしくみがわかるでしょう⁵。

ところで、いきなり復習ですが、みなさんは標準入出力をご存知でしょうか。UNIX のプログラム(特に shell 上で実行するプログラム)は、標準入力から入力したデータを処理して標準出力に出力するというモデルで作成されています。また、エラーメッセージなどの例外出力のために標準エラー出力も用意されています。何も指定しないで、shell からコマンドを実行した場合、標準入力は端末からのキー入力、標準出力と、標準エラー出力は端末の表示に割り当てられています。これらは、<, > 等の記号を使ってファイルなどに接続しなおす事が出来ます(リダイレクト)。また|を使用してあるプログラムの出力を、他のプログラムの入力に結びつける事も出来ます(パイプ)。

なにはともあれ、例題です。

sendmail

```
matsu
|"/bin/cat ->~/lastmail"
```

postfix で /Maildir/ へ配送する場合.

```
~/Maildir/
|"/bin/cat ->~/lastmail"
```

qmail の場合.

⁵ 実際の lilo-ml のコマンド実行の指定は、alias を使用しています。基本的に alias と .forward は同様の機能を持ちます。

```
./Maildir/  
|/bin/cat ->~/lastmail
```

一行目は既に紹介した受信箱への配送です。二行目はこれから紹介するコマンドの実行です。このように | で始まる行はコマンドの実行を指示しています。sendmail, postfix では、コマンドを " (double quotation) で括っていますが、qmail では " を使用しません。実は、sendmail, postfix でも、コマンド行に空白を含んでいなければ、" は必要ありません。

ここで実行しているのは /bin/cat コマンドです。cat コマンドは、指定したファイル読み込んでを標準出力に出力します。ファイル名として - を指定した場合は特別で、標準入力から読み込む事を指示しています。>~/lastmail で標準出力をファイルにリダイレクトしているので、/bin/cat ->~/lastmail で、標準入力から読み込んだ内容を、ホームディレクトリの lastmail というファイルに書き出すという意味になります。

さて、ここで標準入力はなんでしょう？ 端末のキーボードって事は無いですよね。実はパイプで MTA とつながっています。そして MTA は、このパイプ経由で受信したメールその物を送り込んでいます。

結局、この .forward の指示で、受信したメールを通常通り受信箱に配達すると同時に、ホームディレクトリの lastmail というファイルに記録する事になります。メールを受信するたびに lastmail は上書きされていき、最新メールの内容が記録される事になります。

それでは、このアドレスに適当なメールを出して確認してください。ところで、もし日本語のメールを受け取った場合 cat ~/lastmail で中身を見ようとする、環境によってまともに読めたり、理解できない記号の羅列になったりするかも知れません。これは漢字コードの問題です。今の所日本語 EUC を読める環境が多いと思いますが、その場合 nkf -em ~/lastmail の様にすれば良いでしょう。

Windows 等で動作しているメールアプリケーション (MUA) に慣れた方には、メールの本文の前に大量に訳のわからない文字列の羅列があつてちょっとめんどくだったかも知れませんが、これが本来のメールの姿だと思って下さい。

このメールの先頭に付いている物をヘッダと呼びます。Subject:, To:, From: 等比較的馴染深いものもいくつか見付かるでしょう。メールの種類によっては Received: が大量に付いたメールもあるかも知れません。Received: は、MTA がメールを受け付ける度に追加されて行きますので、Received: がたくさんあると言う事は、たくさんの MTA を経由してきた事を示します。またこれを辿って行けば、どの MTA を経由してきたか調べる事も出来ます。

5 拡張アドレス

postfix, qmail には、拡張アドレスと言う概念があります。この機能だけで sendmail から乗り換える価値があると思える程便利です。

先程の例同様、私は、viper2.netfort.gr.jp に matsu というアカウントを持っていて matsu@netfort.gr.jp というメールアドレスが使えるとします。ここで viper2.netfort.gr.jp で動作している MTA が qmail または、拡張アドレスの機能を有効にした postfix だった場合、matsu-ext1@netfort.gr.jp, matsu-test@netfort.gr.jp 等、matsu-*@netfort.gr.jp⁶ というメールアドレスが自由に使えるのです。

matsu-ext1@netfort.gr.jp, matsu-test@netfort.gr.jp というアドレスを使いたい時、qmail の場合は ~/.qmail-ext1, ~/.qmila-test に配送の指定をするだけです。postfix の場合は、そのまま ~/.forward の指示に従います。また、~/.forward-ext1, ~/.forward-test を記述する事で ~/.forward

⁶ qmail では例の通り区切り文字は-, postfix の場合は、設定ファイルで自由に選択できる

と異なる指示を与える事も出来ます。

もし、拡張アドレスを使用できる MTA を使用しているなら、~/forward を利用するプログラムを作成する時は、積極的に利用してください。

6 メールを送信

次にメールの送信に付いて見て行きましょう。

6.1 mail

英文のメールを簡単に出すには mail コマンドが使用できます。ただ、最近は mail コマンドが使用できない環境もあるようですので注意して下さい。

```
$ mail matsu@netfort.gr.jp
Subject: test mail
This is a test mail.
.
Cc:
$
```

以上のように簡単なものです。上記例では matsu@netfort.gr.jp 宛てにメールを出します。メールの宛先は ,(カンマ) で区切って複数指定できます。

Subject: の入力後は本文の入力ですが、メール本文の入力が終わると、.(ピリオド) 一文字だけ入力した行を入力して下さい。Cc: は、必要なければそのまま Enter キーを押して下さい。mail コマンドのバージョンによっては Cc: の入力が無いかも知れません。

余程の事が無い限り、このようにメールを出す事は無いでしょう(しんどいだけです)。使うとすれば次のような場合でしょうか。

```
$ df | mail -s "disk check" matsu@netfort.gr.jp
```

df は、ディスクの空き容量の情報を、標準出力に出力するコマンドです。このようにパイプで mail の入力につないでやる事で、df の出力を、メールで送る事が出来ます。この例では、-s オプションで “disk check” という表題 (Subject) を付けて、matsu@netfort.gr.jp にメールを出しています。

このような使い方ならば色々な応用が考えられるでしょう。cron を使って一定時間毎にサーバの状態をメールで報告という使い方も考えられるでしょう。また、サーバで何かトラブルが発生したらメールで報告と言う利用法もあるかも知れません。

6.2 sendmail

次は、sendmail コマンドを使ってメールを出してみましょう。

sendmail は、ややこしい事に起動時のオプションによって色々な顔を持ちます。-bd オプションを付ければ、daemon モードで起動し MTA として機能します(一般ユーザーはこの操作は出来ません)。オプション無しで起動するとメールの送信コマンドになります。

sendmail コマンドを使ってメールを送る動作をした時は、実際にはキューに送信したいメールを追加するだけですので、MTA としての sendmail が動作していなくても実行出来ます。MTA が機能しはじめた時に実際に送信されます。

postfix, qmail にも sendmail というコマンドが用意されています。これらの sendmail コマンドは、メールの送信機能だけを持っています。

sendmail コマンドは、通常 PATH の通っていない所にあるので、フルパスで実行する必要があります。どこにあるかは環境依存ですので探してもらわなければなりません。最近の Linux ディストリビューションでは /usr/sbin/sendmail にある事が多いようです。また qmail を使用するシステムでは /var/qmail/bin/sendmail にあるでしょう。

sendmail を使ってメールを出すにはある程度メールに関する知識が要求されますが、他人が読むメールでなければそれ程難しく考えなくてもかまいません。

```
$ /usr/sbin/sendmail matsu@netfort.gr.jp
Subject: test mail
From: matsu@myhome.jp
To: matsu@netfort.gr.jp

This is a test mail.
^D
$
```

使用方法は、mail コマンドと同様宛先を指定して起動します。複数の宛先アドレスを指定する事も可能です。mail コマンドと異なり、本文を書く前にヘッダを自分で指定しなければなりません。ヘッダ部が終われば、空行を入れて続いて本文を入力します。本文が終われば ^D([Ctrl] キーを押しながら [D] キーを押す) で終了します。ここで注意しなければならないのは、メールの配送は、あくまでもコマンド起動時の引数として与えられたアドレスである事です。To: や Cc: を書いてもそのアドレスに配送される事はありません。

もし手元に lilo-ml のメールがあれば良く見て下さい。To: lilo@atom.good-day.net と書いてあるのに、あなたのアドレスに届いていますよね?

さて、わざわざ面倒な sendmail コマンドを使うのですから、それにふさわしい内容(?) という事で漢字を含むメールを送信してみます。

最初に次の様なシェルスクリプトを作って下さい。漢字コードは全て EUC で作成して下さい。ファイル名は test.sh としましょうか。

```
#!/bin/sh

echo "MIME-Version: 1.0"
echo "Content-Type: text/plain; charset=ISO-2022-JP"
echo "Content-Transfer-Encoding: 7bit"
echo "Subject: 日本語の表題" |nkf -jM
echo "From: 松本徳真 <matsu@netfort.gr.jp>" |nkf -jM
echo "To: matsu@myhome.jp"
echo
cat <<EOF |nkf -j
漢字を含むメールのテストです。
松本
EOF
```

このスクリプトは日本語を含むメールを作成する物です. `sh test.sh` を実行してどのような出力になるか試しに見てみると良いでしょう. この内容を `matsu@myhome.jp` に送ってみます.

```
$ sh test.sh | /usr/sbin/sendmail matsu@myhome.jp
```

普段使っているメールソフト (MUA) で正しく読めるメールが送られたでしょうか?

ここで, 漢字コードの変換に `nkf` を使用していますが, `-M` オプションは新しいバージョンでしか利用できません. 手元のバージョンでは 1.7 はだめで 1.9 では使用できました. `-M` を使用できない環境では, ヘッダの `Subject:`, `From:` で漢字を使用するのをあきらめるか, 別なツールを探すか新しい `nkf` を導入してください.

6.3 cron

ちょっと本題からはずれませんが, `cron` は利用していますか? `mail`, `sendmail` を使用して, `cron` で一定時間毎にメールを出す事が出来るのは既に説明した通りです.

ところで `cron` で実行されるコマンドの標準出力は, どこに行くか考えた事はありますか? 実は `cron` で実行されたコマンドの出力は, メールになって実行したユーザに届きます.

```
$ crontab -l
0 * * * * /bin/df
```

と設定されていると, 毎時 0 分に `df` を実行してその出力がメールで届く事になります. あとは `.forward` に, 実際に届けたいアドレスに転送するよう指定しておくとも良いでしょう.

7 .forward 利用時の注意

ここでは, `.forward` でコマンド実行する際に, 気をつけるべき点を少し紹介したいと思います.

7.1 .forward のパーミッションは？

コマンドを実行する場合に限りませんが、書き込み許可は所有者自身に限定しておいたほうが良いでしょう。また転送先アドレス等に、プライベートな携帯メールアドレス等を書く事を考えると、他のユーザから読まれないようにしておくのが良いでしょう。

```
$ chmod go-rwx ~/.forward
$ ls -l ~/.forward
-rw----- 1 matsu matsu 11 May 1 2002 /home/matsu/.forward
```

7.2 転送先のアドレスでエラーが発生したら？

メールの配送先でエラーが発生した場合、エラーメールが返ってくる場合がありますが、このエラーメッセージはどのアドレスに送られるのでしょうか？ 通常メールの先頭にある、Return-Path: に書かれたアドレスに返されます。

matsu@myhome.jp から matsu@netfort.gr.jp にメールを出したら .forward の指定で matsu@keitai.jp に転送されるとします。ここで matsu@keitai.jp でエラーが発生した場合、そのエラーメッセージは matsu@myhome.jp に返される事になります。

普段はうまく動作していても、たまに発生した障害が原因でメールを転送している事実がばれてしまいます。

転送先がプライベートな携帯メールのアドレスであった場合、エラーメッセージが原因で、秘密にしていた携帯メールのアドレスや、場合によっては電話番号がばれてしまいます。携帯メールの場合、長期間圏外にいたり電源を切っているなどが原因でエラーメッセージが返ってくる可能性が高いので十分注意してください。

この手の問題の解決は比較的難しいので、最初のうちはこんなものだとあきらめるのも一つの手です。

Return-Path: を書き換えるプログラムを書くのは簡単ですが、悲惨なメールループの原因になりかねませんので、次のような工夫をして下さい。

- メール転送、コマンドの実行は matsu-keitai@netfort.gr.jp 等拡張アドレスを利用する。
- Return-Path: は、matsu@netfort.gr.jp とする。
- matsu@netfort.gr.jp には、転送指定や、コマンド実行の指定は絶対に書かない。

7.3 PATH の内容は？

通常の login 時には PATH を含め各種環境変数が設定されていると思いますが、.forward で起動されるコマンドは、かなり限られたものしか設定されていないのが普通です。よって ./forward にコマンドを記述する時には、フルパスで記述するのが安全です。また、コマンドが起動されたときのディレクトリも MTA によってまちまちです。ファイル名などもフルパスで記述するか、所定のディレクトリに cd しておくのが良いでしょう。

```
matsu
|"/usr/bin/env ->~/env"
```

といった設定をして、どのような環境変数が設定されているか確認しておくとい良いでしょう。postfix では、便利そうな環境変数がたくさん用意されています。

余談ですが、cron 等から実行する場合にも同様の問題が発生します。シェルから実行している時は、正常に動作しているのに cron で実行すると何故か思ったような動作をしない! という時は、「環境変数は正しく設定されているか」を疑ったり、「標準入出力が端末である事を前提にしていないか?」などを一度疑ってみてください。

7.4 処理が終わる前に次のメールが来たら?

.forwrd で起動したコマンドが終了する前に次のメールを受信すると、複数のコマンドが同時に起動した状態になります。特定のファイルを扱うタイプのコマンドだと破綻してしまいます。前のコマンドが終了するまで、次のコマンドの実行を待たせるようなプログラムを作る必要があります。

性能を要求しなければ、ロックファイルを使った排他処理が単純で私の好みです。

それぞれのプログラムで、ロックファイル処理をするのは面倒なので、私は、exec1 <lock file> <command> という風に行えば <command> の実行の排他処理を簡便化するプログラムを作っています。参考にしたいかたは <http://www.netfort.gr.jp/~matsu/lms200303/exec1-20030328.tar.gz> を見て下さい。

一時ファイルを作成したいだけの目的なら、一時ファイルの名前を時刻とプロセス ID の組合せなどにして、各プロセスで使用するファイルが重ならないようにするののも一つの手です。

```
#!/bin/sh

tmp_name=~ /tmpdir/`date +%Y%m%d%H%M%S`. $$
echo $tmp_name
```

8 例題

```
matsu-lms1@netfort.gr.jp
```

というメールアドレスを用意しました。モバイル環境のある方は、お試しください。このアドレスにメールを出すと、お礼のメールが返ってきます。そして送ったメールの本文が、

```
http://www.netfort.gr.jp/~matsu/lms1/index.cgi
```

にアクセスする事で参照できます。まるでメンバーの一人もないメーリングリストのようです。あくまでもサンプルです。投稿数が増えて来ると破綻します。

これを実現している MTA は qmail を利用しています。
.qmail-lms1 は、次のようになっています。

```
|~/bin/sample1.sh
```

~/bin/sample1.sh の内容は、

```
#!/bin/sh

file_name=~/public_html/lms1/ `date + "%Y%m%d%H%M%S" `_$$

cat -> $file_name
~/bin/mail2html.rb $file_name >$file_name.html
/bin/chmod a+r $file_name.html
~/bin/reply.rb $file_name
/bin/rm $file_name
```

となっています。~/bin/mail2html.rb が、送られて来たメールからサブジェクトと本文を抜き出して html に変換します。

~/bin/reply.rb は、送られて来たメールの Reply-to: または From: を元に返事のメールを送ります。それぞれのコードはこの資料の最後にあります。また

```
http://www.netfort.gr.jp/~matsu/lms200303/
```

から参照できます。cgi のソースもここで参照できます。ブラウザでそのまま参照できるようファイル名は変更してあります。

9 最後に

とりあえず、メールを利用したアプリケーションの作成の入口までは解説したつもりです。既存のメール関係のアプリケーションは、vacation のような物から procmail のような超多機能ソフト、また各種メーリングリストマネージャ等たくさん見付かるはずですが。

今回扱った内容は、たとえ自分でプログラムを作成をしなくても、これら既存のソフトを利用する際にも役立つはずですが。

また、現在ではお手軽に利用できるスクリプト言語も多いので、なんだ、自分で作ったほうがよっぽど気の効いた物が作れるなと感じた人もいるかも知れません。そういう方は、是非プログラミングにも挑戦してみてください。