

SMTP,POP

久保 正明

Jul/04/2002

1 メールに必要なプログラム

1.1 MTA(Mail Transfer Agent)

計算機間でメールの送受信を行います。MTA は、送信しようとするメールの宛先から、受信先のリモートホストを調べ、受信先リモートホスト上で動いている MTA と通信を行ってメールを送受信します。

なお、他の計算機の MTA から受信したメールの宛先によっては、自分自身の計算機が受け取るべきメールでない場合があります。その場合、第 3 者の計算機にメールを送信する必要があります。

Unix で代表的な MTA が Sendmail です。

1.2 MDA(Mail Delivery Agent)

MTA によって受信したメールの宛先を解釈し、そのメールを受け取るべき適切なユーザのメール受信ファイル（メールスプールファイル）に書き込みます。

1.3 MUA(Mail User Agent)

ユーザがメールを読んだり、メールの文章を書いたりするプログラムです。メールを出すときには MTA と通信を行い、MTA にメールの送信を依頼します。そしてメールを読むときには、MDA が書き込んだメールスプールファイルからメールを呼び出します。

2 メールを扱うためのプロトコル

2.1 SMTP(Simple Mail Transfer Protocol)

SMTP は電子メールの転送を行うためのプロトコルです。このプロトコルは、MTA 間で電子メールを配送したり、MTA と MUA 間で電子メールを差し出すときに使われます。TCP ポート番号は 25 です。sendmail では拡張 SMTP プロトコル (ESMTP) が使用されています。

2.2 POP/APOP/IMAP

POP/APOP/IMAP はいずれもメールをサーバから受信するときに使うプロトコルです。POP3(Post Office Protocol Version3) はメールサーバからクライアントへメールを持ってくるためのプロトコルで、これによりメールクライアントは、電子メールをスプールしているサーバからメッセージを取得します。TCP

ポート番号は 110 です。メールプールからメールを読むためには、メールプールを読み書きできる計算機で MUA を動かす必要がありますが、POP を利用するとメールプールを持たない計算機から、NFS を使わずにメールを読み出すことができます。APOP(Authenticated POP) は、「認証を強化した POP」です。POP3 では受信サーバにアクセスする際に必要なパスワードは平文で流されているので、その安全性を強化するため、APOP ではパスワードを暗号化して送ります。IMAP(Internet Message Access Protocol) は POP3/APOP のようにメールをクライアントにダウンロードして読まれるのではなく、常にサーバ上に存在し、ユーザがそこからメールを読み出します。

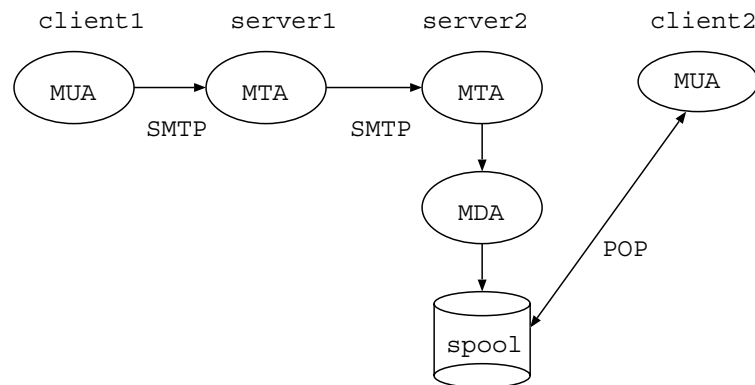


図 1: メールの流れ

3 sendmail

sendmail はメール送信要求が来ると、sendmail.cf(/etc/mail/sendmail.cf) の指示に従って動作します。sendmail.cf の記述に照らして、それが自ホスト宛てのメールかどうか判定します。自ホスト宛てなら、メールプールにメールを格納します。自ホスト宛てでないときは、sendmail.cf のルールにしたがって、そのメールをどこにむけて転送するか決め、配送します。

3.1 sendmail.cf

sendmail は sendmail.cf の内容にしたがって動作します。しかし、sendmail.cf の中身は難しく、これを直接書くのは困難です。そのため、sendmail.cf を作るための CF というツールを使用します。

4 課題:CF を使った sendmail.cf の作成

1. su になる
2. backup をとっておく
 - cd /etc/mail
 - cp sendmail.cf sendmail.cf.back
3. cd /tmp

4. ソースファイルを取得

- `cp -r /u/kubo/work/rinkou/sendmail/CF* .`
- 以下の3つのファイルがあることを確認
- `CF-3.7Wpl2.tar.gz`
- `CF-3.7Wpl2-smtptest.patch1`
- `CF-3.7Wpl2-smtptest.patch2`

5. `cd CF`

6. 展開する

- `gunzip CF-3.7Wpl2.tar.gz`
- `tar xvf CF-3.7Wpl2.tar`

7. `cd CF-3.7Wpl2/Master`

8. `patch` をあてる

- `patch < /tmp/CF/CF-3.7Wpl2-smtptest.patch1`
- `patch < /tmp/CF/CF-3.7Wpl2-smtptest.patch2`

9. `cd ..`

10. `make cleantools`

11. `make tools`

12. `make samples`

13. `sendmail.cf` を作る元になるファイルをコピーする

- `cp Standards/sendmail-v8.def ./sendmail.def`

14. `vi` で `sendmail.def` を編集

- `OS_TYPE=bsd4.5`
- `MX_SENDMAIL=yes`
- `MY_DOMAIN=4f.db.is.kyushu-u.ac.jp`
- `MY_NAME=(自分のマシン名)`
- `FROM_ADDRESS='$m'`
- `RECIPIENT_GENERIC=yes`
- `REWRITE_GENERIC_FROM=lower`
- `REWRITE_GENERIC_TO=yes`
- `REWRITE_LOCAL_ON_RELAY=generic`
- `ACCEPT_ADDRS='$m'`
- `SPOOL_HOST='popple.db.is.kyushu-u.ac.jp'`

- BITNET=mx
 - REJECT_JUNET=yes
 - ALIAS_FILE_PATH='/etc/mail/aliases'
 - PRIVACY_FLAGS='noexpn,novrfy,authwarnings'
 - MAIL_RELAY_RESTRICTION=yes
 - LOCAL_HOST_IPADDR=/etc/mail/sendmail.localip
 - LOCAL_HOST_DOMAIN=/etc/mail/sendmail.localdomain
15. 編集した後 , make sendmail.cf
16. sendmail をとめる
- ps -ax | grep sendmail
 - kill sendmail のプロセス番号
17. sendmail.cf をコピーする
- cp sendmail.cf /etc/mail/sendmail.cf
18. sendmail をデバッグモードで動かす
- sendmail -bt
 -) 0 ユーザ名
(最後に...returns \$# local \$: ユーザ名と出れば OK)
 -) 0 ユーザ名@マシン名.4f.db.is.kyushu-u.ac.jp
(最後に...returns \$# local \$: ユーザ名と出れば OK)
 -) 0 外部のメールアドレス (自宅の PC のアドレスなど)
(最後に...returns \$# smtp \$: 外部アドレスと出れば OK)
19. C-d でデバッグモード終了
20. sendmail を動かす
- sendmail -bd
21. mail コマンドで自分にメール送信
- mail 自分のメールアドレス
 - subject:(テストメールの題名)
 - テストメールの文章入力 (日本語不可)
 - . を入力し、送信。
22. exit で su から戻る
23. minako で受信確認
- ssh minako
 - mail

- 先ほど送ったメールを受信していることを確認
- 確認後は `q` で終了

ここまで出来れば OK です

あとは、編集前の状態に戻しましょう

- `su`
- `cd /etc/mail`
- `ps -ax | grep sendmail`
- `kill sendmail` のプロセス番号
- `mv sendmail.cf.back sendmail.cf`