逢　甲　大　學

資訊工程學系專題報告

網路郵局

學生：黃紀豪(四甲)
余世鴻(四甲)
曾嘉祥(四甲)

指導教授：李維斌

中華民國九十二年十二月

目錄

圖 表 目 錄

第一章　緒論

1.1 動機、

在網路日益普及化的現在，電子郵件已經成為了人與人之間溝通的最方便且快速的管道。

在講求效率的現代，重要訊息的正確、迅速、確實地傳達是必須的，但是通常在訊息、文件以及公文的傳遞中，冗長的時間花費似乎是不可避免的。為了避免無謂的時間浪費，以電子郵件作為傳遞媒介無疑是最適當不過的。再非常短的時間之內，所要傳遞的文件以及公文可以透過網路和 Mail Server 到達對方的手中。

但是，總是會有對方遲遲未有反應的情況，導致這種情況發生有幾種原因： e-mail 可能寄錯位置，或是網路的問題而導致信件無法正確地送到 Mail Server；信件已送到 Server 但對方遲遲未觀看郵件；或是對方已經下載郵件了但卻沒有回應。

在此情況下，信件回條（掛號信）的功能可以解滿足需求。當郵件送到對方的手中，會有回應給寄件者，確保郵件確實地送到對方的手中。這功能，在部分的 e-mail 軟體，如 Microsoft 的 Outlook 即有此項功能。但是這樣是不好的，若使用者不使用此類可以回信功能的軟體將無法享有此項功能。

1.2 目的

雖然，Microsoft 的 Outlook 已經有了掛號信功能，但是那並不是完整的回函機制。Outlook 的掛號功能還需要用滑鼠點"確定回信"，回函才會正式寄出，如此並沒有多大的可靠度。但是我們在 Mail Server 上做的掛號信功能並不是由收件者決定回函與否，而是由身為郵局的 Mail Server 擔任此角色，可靠性高。除此之外，脫離了應用程式的規範，使用者不必因為收信軟體的不同而無法使用此功能。

另外，一般的 Mail Server 沒有讓使用者設定個人化的擋信名單功能。所以，我們希望在 Mail Server 上加一個設計，能藉由使用者發出給 Mail Server 的 E-mail 中主旨裡的命令規格（如 Keyword、

對方郵件位置、特定的字串等）來增加個人的擋信名單。在垃圾信充斥的今天，相信這是個必須的功能。

此外，若有重要事件，寄件者可以使用高優先權的信件格式，如此可以確保對方能在第一時間看到寄件者的信，而不是在眾多無關緊要的信件中慢慢地尋找。

這些動作都是由使用者發出郵件到 Mail Server，然後 Mail Server 依照該郵件的指示做出動作。這代表了在這種制度下，使用者可以透過 E-mail 對 Mail Server 下”指令”，若配合更多其他的功能，Mail Server 的使用變化不僅於此。

1.3 系統簡介

經過我們的研討，發現要做出一個外掛程式使得所有的 Mail Server 有我們所需要的功能似乎不太可能。再加上技術上的難度，軟體內部作業方式以及 Source Code 的取得方法，我們將目標鎖定在使用最為廣泛的 Sendmail 系統上。

首先，先說明雙掛號信的功能。當 A 寄一封註明為掛號的信給 B 的時候，首先會先將信寄到 Mail Server，而 Mail Server 收到信件的時候即會發送另一封信回覆給 A，表示 Mail Server 確實地收到了信件，A 可以不用擔心信件遺失了。當 B 成功地下載此封掛號信的時候，Mail Server 也會再回一封信給 A，表示 B 確實地從 Mail Server 那裡下載了此封掛號信。

第二是信件優先權。當有信件註明為掛號信或是急件之類的信件，系統會將此類的信件放在 Mailbox 的最上端，使用者下載信件時為優先被處理的信件，確保重要信件在第一時間由收件者接收。另外還可以由使用者定義依照信件的來源以及標題或內容作為優先順序的依據。

第三是使用者定義的擋信名單。電子郵件發達之後廣告信更是蓬勃的發展，使用者會收到大量的垃圾信件，甚至可能會掩蓋重要訊息。因此我們設計使用者可以自己定義一個阻擋信件的名單，而 Mail Server 將會依此名單來過濾信件，做出直接刪除或是做出其他的處

理。

確定了目標之後，接著就是收集系統所需要的技術資訊以及相關資料，進一步去實行。其後的章節將會漸進式的講解洗統的相關技術，系統架構，以及實作結果。

第二章　背景

在本章將介紹郵件伺服器運作原理以及相關網路與協定的背景知識。

2.1 Mail Server

2.1.1 Mail Server 的相關名詞：

2.2.1.1 MUA

MUA 為 Mail User Agent 縮寫，顧名思義，就是『郵件使用者代理人』，這是由於通常我們 Client 端的電腦都無法直接寄信，所以，需要透過 MUA 來幫我們傳達信件，不論是送信還是收信，Client 端的用戶都需要透過各個作業系統提供的 MUA 才能夠使用郵件系統。舉個例子來說， Windows 裡面的 OutLook Express，Netscape 裡面的 mail 功能與 KDE 裡面的 Kmail ，都是 MUA ，MUA 主要的功能就是收受郵件主機的電子郵件，以及提供使用者瀏覽與編寫郵件的功能。

2.2.1.2 MTA

MTA 為 Mail Transfer Agent 縮寫，MTA 是用在郵件主機上面的軟體，也是主要的郵件伺服器喔，這個 MTA 就是『郵件傳送代理人』的意思。顧名思義，既然是『傳送代理人』，那麼使用者寄出的信，與使用者要收信時，就是找 MTA ，因為他要負責幫我們使用者傳送，MTA 的功能有這些：

1. 外部主機寄來的信件：MTA 最主要的功能就是收受外部來的信件，只要這個信件裡面有 MTA 內部的帳號時，那麼這封信就會被 MTA 收下來。
2. 幫使用者傳送 (寄出) 信件：只要使用者具有合法的使用 MTA 的權力，那麼該使用者就可以利用這部 MTA 將他把信傳送出去，不過需要注意的是， MTA 會將信件送給目的地的 MTA 而不是目的地的 MUA。
3. 讓使用者自己的信可以收回去：使用者可以將放置在郵件主機的信件收到自己的個人電腦上面收看。

大致的功能就是這些，通常我們所說的 Mail server (郵件伺

服器) 就是指 MTA 而言的。

2.2.1.3 MDA

MDA 為 Mail Delivery Agent 的縮寫，『郵件遞送代理人』主要的功能就是將 MTA 所收受的信件，依照信件的流向 (送到哪裡去) 來將該信件放置到本機帳戶下的郵件檔案中 (Mailbox)！或者是再經由 MTA 將這個信件送到下個 MTA 去！而如果信件的流向是到本機當中時，這個郵件代理人的功能可不止是將由 MTA 傳來的郵件放置到每個使用者的 Mailbox 而已，他還可以具有郵件分析 (filtering) 與其他相關的功能。

2.2.1.4 Mailbox

『郵件信箱』，就是在你主機上面的一個目錄下的，某個人『專用』的信件收受檔案，舉例來說，系統管理員 root ，在預設的情況下，他會有個信箱，預設的檔案是在 /var/spool/mail/root 這個檔案就是了，一個帳號都會有一個自己的信箱，然後，當 MTA 收到 root 的信時，就會將該封信件存到 /var/spool/mail/root 這個檔案中，使用者可以透過程式來將這個檔案裡面的信件資料讀取回去。

2.1.2 Mail Server 的運作原理：

Mail Server 如何傳送到目的郵件主機的呢？底下分成『寄信』與『收信』兩個主要的郵件主機使用方式來加以介紹。

2.1.2.1 Mail Server 的寄信流程

參考一下底下的圖示來說明：

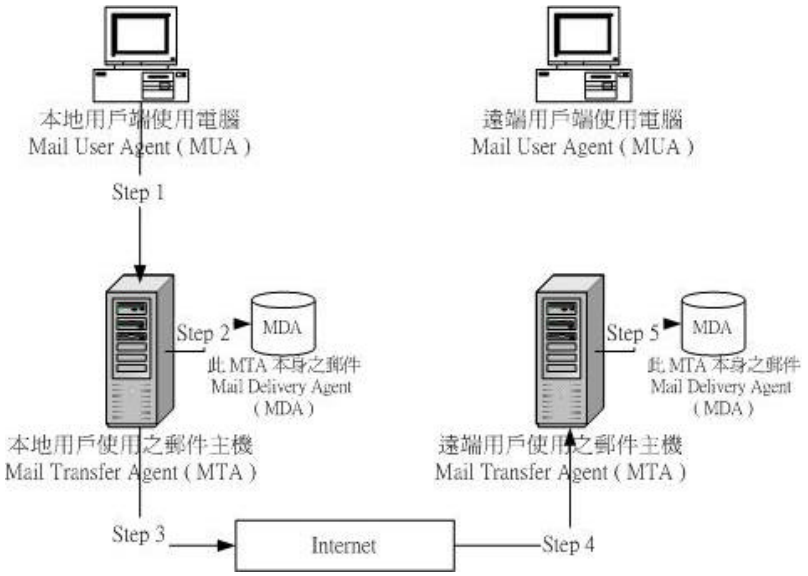


圖 2-1、電子郵件以郵件主機寄送信件示意圖

左上角的 MUA 將信件寄出之後，最後信件將會存放在右邊那部 MTA 主機裡面，還沒有到達目的地電腦 (就是右邊的 MUA 那部電腦)一般來說，將信寄出去可分成幾個步驟：

Step 1 使用者利用 MUA 寄信到 MTA 上面：通常我們使用 MUA (例如 Outlook express) 寫信的時候，你總是要定義出幾個東西：

發信人與發信網站：這個發信網站就是等一下 Step 2 接收信件的那個 MTA。

Step 2 MTA 收到自己的信件，交由 MDA 發送到該帳號的 MailBox 中：如果在 Step 1 所收到的信件中，那個 e-mail.server 就是 MTA 自己，此時 MTA 會將該信件交由 MDA 去處理，將信件放置在收信者的信箱中。

Step 3 如果由 Step 1 來的信件的收件人並不是 MTA 的內部帳號，那麼該封信將會被再轉送出去，由 Step 1 及 Step 3 的動作，我們也稱為 Relay (郵件轉遞) 的功能。

Step 4 遠端 MTA 收受本地的 MTA 所發出的郵件：遠端的 MTA 會收受我們這部 MTA 的信件，並將該信件交給他的 MDA 來處理此時，信件會存放在遠端的 MTA 上面，等待使用者

登錄讀取或者下載回去。

2.1.2.2 Mail Server 的收信流程

底下的圖示為收信的流程圖：

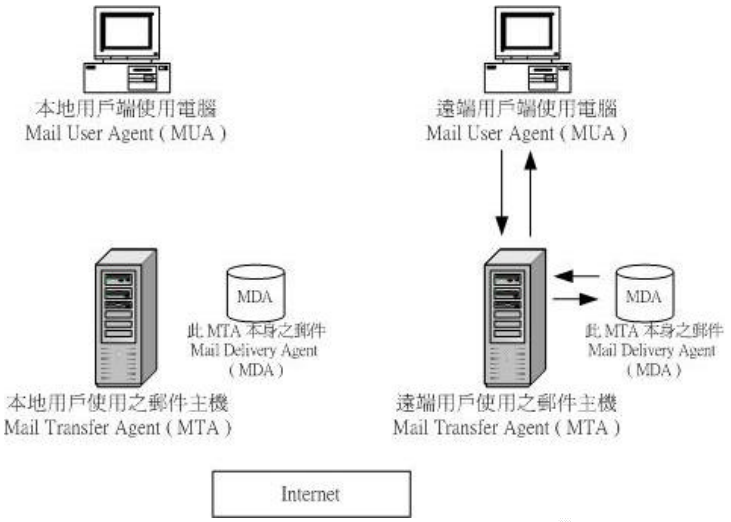


圖 2-2、用戶端收受郵件主機的電子郵件示意圖

遠端用戶使用的電腦直接連接到他的 MTA ，跟 MTA 要求察看自己的 mailbox 是否有信件，而 MTA 透過 MDA 去檢查之後，如果有信件的話，就會將他傳送回使用者的 MUA 中！同時，根據 MUA 的不同設定， MTA 會選擇將該 mailbox 清除掉，或者繼續保留！若繼續保留的話，那麼下次使用者再次的接收信件時，保留的信件會再次的被下載，因此，通常使用者 MUA 都是預設刪除掉 MTA 上面的 Mailbox 內容。

2.2 郵遞通訊協定

下圖為使用協定 POP 與 SMTP 的流程圖，本章節針對常用的郵遞通訊協定 POP3 與 SMTP 分別介紹。

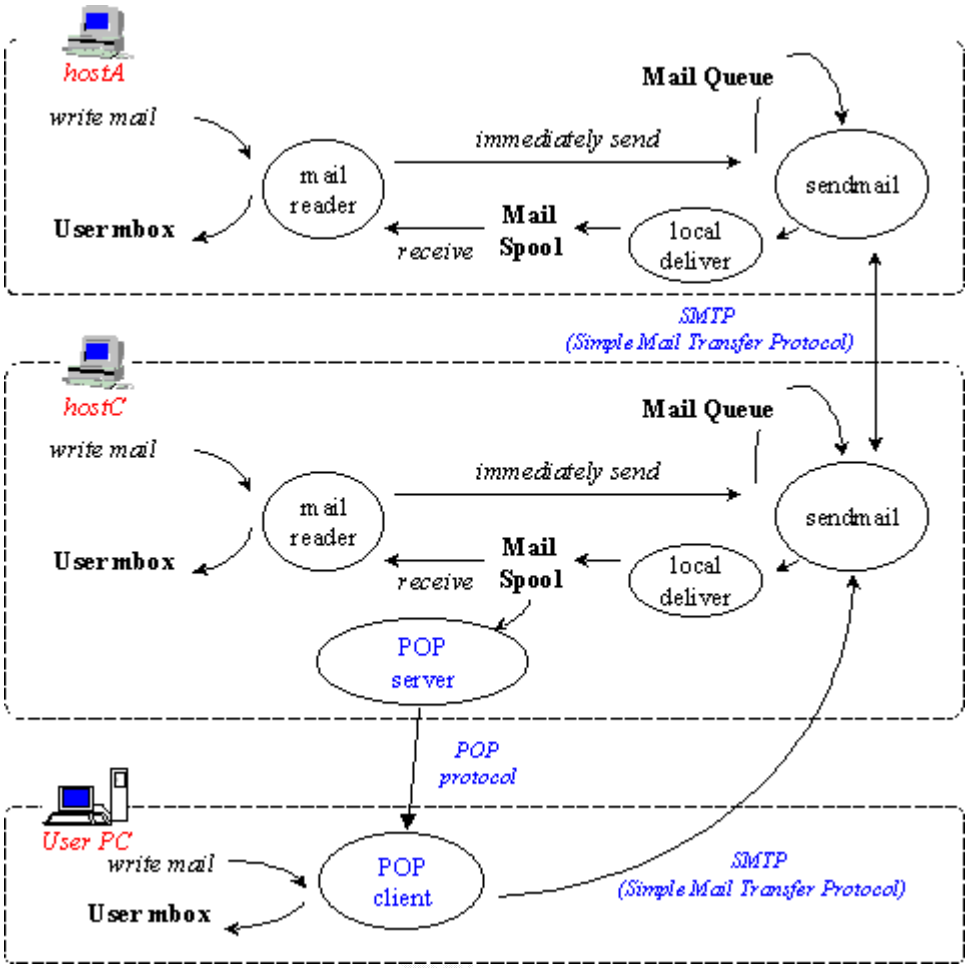


圖 2-3、協定 POP 與 SMTP 的流程圖

2.2.1 POP3

MTA 的使用者 Mailbox，以讀取或者下載使用者在 Mailbox 當中的信件。，目前常用的 POP 協定為 POP3 (Post Office Protocol version 3)，這個協定產生的 port number 為 110 ，所以，MUA 經由 MTA 的 port 110 將信件由 MTA 的 mailbox 當中將信件收到本地端的 MUA 上面以供瀏覽！同樣的，只要 MTA 與 MUA 同時支援 POP3 這個協定，那麼信件就可以自由的收受了。在 pop3 的收信協定中，一般來說，當 client 端收完了主機端的信件之後，則該信件會主動的被主機端所刪除。

POP3 在 RFC 1725 中被定義，它是一個簡單的請求/反應協定:客戶端把命令送給伺服器，而伺服器回答命令,表 13.1 表示在 RFC1725 中定義的 POP3 命令組合。

命令	功能
USER username	登錄需要的使用者名稱
PASS passwird	登錄需要的使用者密碼
STAT	請求不能讀取的訊息/位元組數目
RETR m	恢復訊息的編號 m
DELE m	刪除訊息的編號 m
LAST	請求最後一個訊息存取的編號
LIST [m]	所有訊息 message m 的容量的請求
RSET	回覆所有訊息並重設訊息編號為 1
TOP m n	印出表頭及訊息編號 m 前面 n 行之訊息
NOOP	什麼都不做
QUIT	POP3 任務結束

表 2-1　POP3 的命令

一般而言，要維持與管理一個信息傳遞系統(message transport system，MTS) 是相當複雜的一件工作。所以許多工作站及個人電腦均希望可以收/送電子郵件，但卻不用去維持與管理一個信息傳遞系統。為了解決這個問題，一個 MTS 伺服器可以利用 POP3 這個協定來提供"郵政信箱"(maildrop)的服務給一些不想管 MTS 卻又想有郵件服務的工作站及個人電腦。

基本運作

伺服器在 TCP port 110 上等客戶端連線。所以 POP3 假設傳輸層提供可靠的傳輸。

POP3 是定義在 RFC1725 (November 1994)。

POP 運作時共分成三種狀態：AUTHORIZATION，TRANSACTION，UPDATE Server 回 Client 的回應中，如果是正常的回應則以"+OK"為起頭，不正常的回應則以"-ERR"為起頭。

Server 的回應如有含有一行以上，則會以一行只含有句點的特殊行做結束。(這跟 SMTP 用的方法一樣

POP3 Server

Popper 是在 Unix 系統下 post office protocol server 的實做用於管理 Macintosl 和 Ms-DOS 的電子郵件,此伺服器是由加州柏克萊大學發展,整個說明見 RFC1081 和 RFC1082

執行指令 /usr/local/libexec/popper

[-d]
設定旗標 debugging 並且開啟 debugging 所有訊息將存於 syslog(8)

[-s]
在每個 popper 動作後會將 username, 刪除訊息的數量,刪除位元的數量,存在 server 上訊息和位元的數量記錄在 syslog(8)

[-t trace-file]
開啟 debugging 並儲存 trace 訊息

[-T timeout]
改變編譯時 POP_TIMEOUT 所設定的時間當 Client 端停滯過久時 server 為了避免無窮的等待必須將使用者中斷

[-b bulldir]
開啟公告的選項 bulldir 存放公告路徑

系統公告允許系統管理者發佈重要訊息給使用者,而不須經由 mail **的形式在** bulldir **目錄下每一個公告存成一個個別檔案每一個檔案包含一個單一** mail **的訊息如** header **和** body,**每個公告的第一行必須是** "From"**最簡單的方式建立公告檔是系統管理者**,mail **給自己公告檔案並** copy **一份給想要被回信的使用者**,**將信的內容** copy **到** bulldir **目錄下**,**所有的公告檔以公告的數目為開頭如：**23.pophost_down_Sunday
每次 POP **的** client **端登錄後新的訊息將加到使用者的信箱中**

柏克來 POP server 將被 inetd 啟動,除非同時跑 POP2 和 POP3 你可以簡單的定義 port 110 和 port 109 給 POP server,popper 程式將會檢查對等的 ip address 是否已經在區域的 domain 註冊,當連結的的機器沒有正式的網域名稱時將會 logging 一個警告的訊息,當 client 端必須依據一個再 hostuserid 和 password 來確認自己時似服器將進入 authorization 狀態這個狀態下,系統不會允許任何的交換(exchange)(除了要求結束外),假如認證失敗會被記錄下來,而且這個交易(session)會被中止, 一旦使用者的身份確認,popper(pop3 server 的程式名稱)會更改本身的 user id 和 group id 使其和使用者的相符,然後進入交易狀態,server 會替使用者的郵筒產生一個臨時性的複本,以使用於所有伴隨而來的交易.其中包括找回 mail,移除 mail 反移除 mail 等大量的 POP3 命令.當 Client 端中止時,server 會進入最後的更新狀態 在這個狀態下,網路連結會被中止且使用者的信箱將以已被修改的臨時信箱所更新

LOGGING

POP Server 使用 syslog 去維持本身活動的記錄.在用 BSD 4.3 syslogging 的系統中, 除了"Debugging"的訊息會用"debug"的優先權記錄,其餘的皆以"notice"的優先權 記錄在"local0"的設備中(內定).內定的 log file 是/usr/spool/mqueue/POPlog. 如用需要,這些都能改變.在用 BSD 4.2 syslogging 的系統中,所有的 message 都記錄在本地的 log file 中,通常是 /usr/spool/mqueue/syslog

DEBUGGING

在 popper 程式在 inetd.conf 以 -d 這個參數啟動後,那麼它將會記錄所有的 debugging 訊息.在使用這個選項時,當產生不可忽視的訊息至 log file 時,應特別小心.我們可以用"-t <file name>"這個選項選擇性地將 debugging 資訊用"fprintf"代替"syslog"放到 file "<file name>"

對於 SunOS 3.5 版,popper 程式被 inted 從/etc/service 中啟動(launched).這個檔案並不充許你從命令去修改(需要 root 的權限).然而,假如你想要啟動 debugging,你可以設計一個 shell script(類似一個批次檔)取代 popper,讓/etc/servers 去啟動他,而在 script 中呼叫 popper 所要的參數

2.2.2 SMTP

一般在寄信的時候，亦即由 MUA 將信件發送到 MTA 的過程中，以及 MTA 將信轉遞到下一個 MTA 的功能，目前絕大部分的郵件主機都是使用 SMTP (Simple Mail Transfer Protocol) 這個協定，port number 為 25 ！在寄信的時候，你的 MUA 會主動的連接 MTA 的 port 25 ，然後將信經由 MTA 的 smtp 協定 (port 25) 而送出去！而郵件主機 MTA 在轉遞的時候，也是經由下一部 MTA 的 port 25 來將信送出去的！所以，不論是使用什麼 MUA 或 MTA 郵件架設軟體，只要支援 smtp ，那麼信件都可以順利的流傳。

第三章 系統實作

3.1 系統使用工具

3.1.1 SED

現今的文字編輯器多摹擬人們在紙上寫作的環境，其普遍具備的功能，如：游標可上下左右移動，或快速地向前、向後捲頁；尋找和替代字串或是整段文字的搬移等等。換句話說，編輯器是``被動地''來記錄人們修改的文件。而 Sed 乃是一``主動''的編輯器，當它接受到編輯命令之後，即依命令自行完成文章中大部份的編輯工作。它尤其適合處理``規律性''的文件。

何謂 Sed(Stream EDitor)

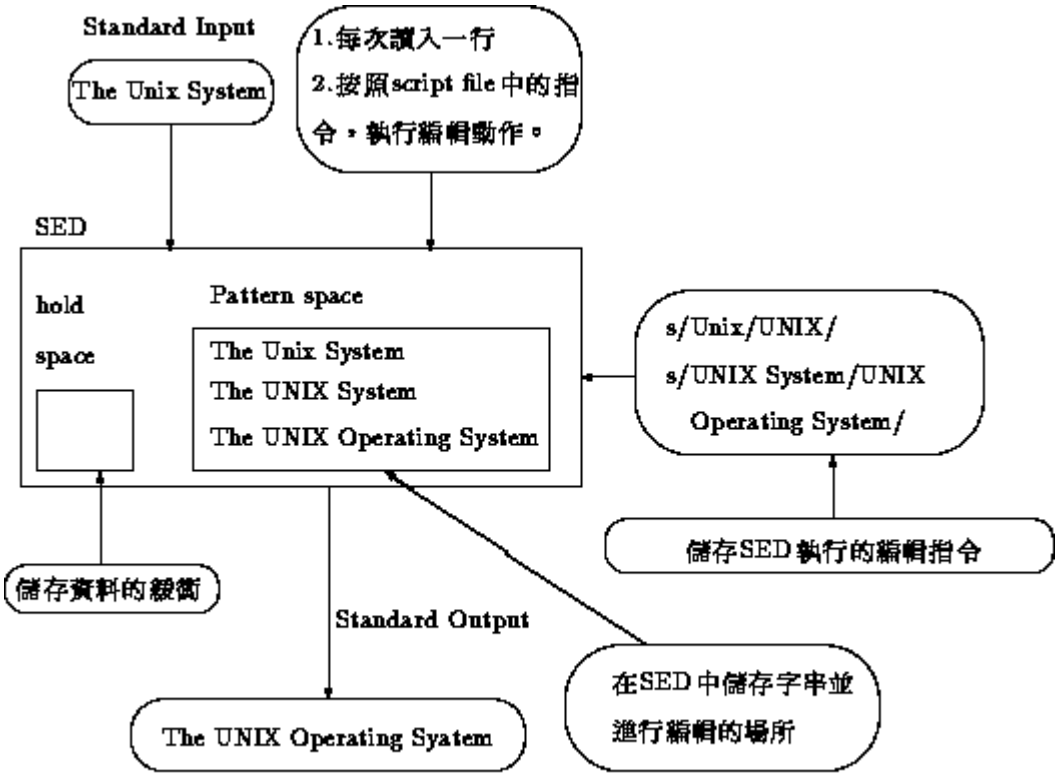
Sed 原為 UNIX 系統上的非交談式文字編輯器(non-interactive stream editor)。當 Sed 讀入待編輯文件，會依編輯命令來進行文件的編輯工作。

如同其他 UNIX 的指令一般，Sed 亦是由標準輸入(standard input)讀入欲編輯的文件，經 Sed 處理後，再由標準輸出(standard output)送出結果。由於 Sed 是採逐行讀入文字，並依據命令來進行編輯，因此可視作檔案中的文字逐行通過 Sed 編輯器。對此，我們可從圖一的組織架構圖中進一步了解 Sed 的概況。

由於 Sed 可依使用者所下的命令自動完成文件編輯工作，因而成為``管線式處理''(pipe line)中文件編輯的重要工具。

由圖一中，可以發現：圖右邊的方塊為 Sed 所執行之指令所在--- script file；上方的 standard input 為標準輸入埠，是 Sed 讀入文字之處；下方是 Sed 標準輸出埠，即主要結果由此送出。此外，Sed 另包含 pattern space 和 hold space 兩部份。其中，pattern space 為其工作場所，當 Sed 讀入一行文字後，會將文字置於此處，再至 script file 讀取編輯指令進行編輯工作；hold space 則為其暫存文字字串的地方，當 Sed 由標準輸入讀入一行文字並放入 pattern space 時，Sed 會依照 script file 所記載的

編輯指令逐一對 pattern space 內資料執行編輯動作，之後，再將 pattern space 內的結果送至標準輸出埠中輸出，接著再由標準輸入埠讀入下一行文字。如此重複執行上述動作，直至讀完所有待編輯文件為止。



圖一 Sed組織架構圖

圖 3-1 Sed 組織架構圖

Sed 主要的用途

以下舉出四類 Sed 的主要用途：

1. 在一檔案中進行多項編輯工作：有一類的編輯動作使用者經常會應用到，例如將電子郵件的信頭刪除、將文件中連續空白行刪減至一行…等等；通常此類的處理，可藉由幾個 sed 命令即可完成使用者所需的編輯動作。因此，使用者可充份應用 Sed 的此一特性，快速完成文件的修改。此外，由程式產生的輸出資料，大都可經由此法改變排列格式或加入相關說明以得出更佳的文件品質。

2. 在檔案中進行搜尋字串和替換字串工作：在一份文件的編輯過程中，字串替換的動作經常發生(諸如：改變字母的大小寫、日期格式或名稱等)，若將這些重複性的動作交由 Sed 處理，將可大為節省所需耗費的人力與時間。

3. 於一檔案中截取或增添部份文稿：對於文件中可能引用其他文稿的處理，Sed 具有可用關鍵字擷取文件中所需的部份，再併入文件中的功能。例如在一資料庫中取得一份文件資料，或使用 Sed 將該文件的大綱（即章、節）擷取出來，製成目錄，最後，您還可在文件中插入自己的說明，完成一份完整的文件。

4. 在執行程式中進行編輯工作：由於 Sed 是採用非交談方式進行，故可在程式執行時，動態地編輯文字檔，包括調整參數、環境變數等，使程式自動地隨外在環境而改變，以達成程式自動化的目的。這是 Sed 所擅長的方式。

Sed 的語法

Sed 的命令格式如下：

```
sed [-n] [-e 'script'] [-f script_file] file...
```

其中，script_file 為儲存編輯指令的檔案。-e 後緊接著編輯指令，是將編輯指令放在命令列中的表示法。此兩者可混和使用。-n 表示由指令控制編輯結果是否輸出，而非在指令執行結束時，逕行輸出結果。

例如：將 text.old 檔案中 Unix 字串以 UNIX 字串替換，並將結果存於 text.new 檔案中的方法為：

```
sed -e 's/Unix/UNIX/' text.old > text.new
```

Sed 亦可由標準輸入讀入資料，依照 command.scr 內的編輯命令處理，然後顯示在螢幕上。其指令為：

```
cat text.old｜ sed -f command.scr ｜more
```

接著, 則是依照 command.scr 內的命令控制輸出。其命令為：

```
sed -n -f command.scr text.old > text.new
```

以上均是使用者經常可見的 Sed 命令形式。

若要使 Sed 能正確無誤地編輯文件，需先認識 Sed 的編輯指令。以下為 Sed 的編輯指令格式：

```
[address [, address]] command [argument]
```

在上面的指令格式中，Sed 將行數視為位址(address)。Sed 指令可以有 0、1 或 2 個位址參數，用以規範指令的作用範圍。因此， Sed 的指令可作用在每一行、特定一行或是某幾行。Sed 允許擁有參數的指令只有``s''(substitute)。

Sed 的位址參數可以是行數或是用字串表示出現字串的行。以刪除指令``d''(delete)為例，當僅有一指令 d 而無位址標示時，是表示刪除所有的文字；欲刪除第一行的方法為 1d（此時 1 表示第一行）；刪除出現``UNIX''所在該行的方法為：/UNIX/d（用/UNIX/表示出現該字串的行數）。例如：欲刪除第 1 行至第 10 行，其方式為：1,10d（此時 1 和 10 均為行數）；而刪除第一行至第一個空白行的方式則為：1,/^$/d (此時 1 為行數，/^$/ 表空白行)。

Sed 的指令

Sed 的指令約略可分成七類，分述如下：

1.處理整行文字的指令(whole-line commands)：此類指令以一行文字為單位, 如刪除或插入一行文字。Sed 提供五種此類指令，其詳細說明如下

指 令 名 稱	位置參數數目	功 能 描 述
(刪除一行) d	0，1，或 2	將符合位置參數的整行文字刪除，使其不輸出。
(讀入下一行) n	0，1，或 2	遇到符合位置參數的該行文字時，輸出該行文字。再由標準輸入讀入一行文字，繼續執行下一個指令。
(加入文字) a\ 文字	0 或 1	將文字輸出在符合位置參數的該行文字之後。
(插入文字) i\ 文字	0 或 1	將文字輸出在符合位置參數的該行文字之前。
(替換文字) c\ 文字	0，1，或 2	將文字輸出在符合位置參數的該行文字，而該行文字並不輸出。 沒有位置參數時，每行均替換。有一個位置參數時，替換該行。有二個位置參數時，以一份文字代替此二位置參數之間的所有文字輸出。

表 3-1 Sed 處理整行文字的指令

2．替換或轉換(substitute，transform)：此類指令改變符合位置參數的該行文字的部份字串，與前處理整行文字的命令相異。此類指令共有 2 個，即替換字串及轉換字元，分述如下：

(a) 替換字串：找到符合位址參數的該行文字中，以新字串替換
舊字串。如：

/UNIX/s/system/System/
是將
The Unix system.
The UNIX system.
改成
The Unix system.
The UNIX System.
將第二行之 system 改成 System

此命令又是 Sed 指令中唯一具有參數的指令。它共包含四個
參數，分別為：

g：在該行文字中，替換所有符合的字串。
n：在同行文字中，替換第 n 個符合的字串。
p：Sed 執行替換及輸出該行文字。
w filename：將替換後的整行文字，寫入檔案名為 filename
的檔案中。

(b) 轉換字元：字元替換命令。例如：

[address[,address]]Y/string1/string2/

在符合位置參數的該行文字中，找尋在 string1 中出現的字元，並以 string2 中相對應 string1 位置的字元替換。例如：

Y/abc/cde/
即：將 adobe 改成 cdode。

3. 輸入及輸出(Input/ Output commands)：此類指令控制讀取或寫入檔案，或輸出至標準輸出的動作。其詳細說明見表二。

4. 同時處理多行文字(process multiple lines commands)：Sed 的 patten space 一次只容納一行文字，故 Sed 一次只處理一行文字。而此組指令提供 Sed 一次處理數行文字的能力。其詳細說明如下。

指 令 名 稱	位置參數數目	功 能 描 述
(送至輸出) p	0, 1, 或 2	遇符合位置參數的該行文字時，印出 pattern space 的內容。
(送至輸出) l	0, 1, 或 2	與 p 相似，不同的是將鍵盤上無法輸出的符號以 8 進位數字印出，且將超 72 個字長度的部份移至下一行。
(寫到檔案) w filename	0, 1, 或 2	將符合位置參數的整行文字寫至 filename 檔案中。
(由檔案讀入) r filename	0, 1, 或 2	遇到符合位置參數時，由 filename 檔案讀入資料，然直接送到到輸出。

表 3-2 輸入及輸出指令

指 令 名 稱	位置參數數目	功 能 描 述
(Next line) N	0, 1, 或 2	遇符合位置參數時，由輸入讀入一行加在目前

		pattern space 之字串之後。
(Delete first part of pattern space) D	No address	將 pattern space 內第一個 newline 字母之前的文字都刪去。若在 patten space 內僅有個 newline 字母，則如 d 作用一般。
P (Print first part of pattern space)	No address	將第一個 newline 字母之前的文字印至輸出裝置。

表 3-3　多行處理指令

3.1.2　Sendmail

基本上 Sendmail 是一個 MTA (Mail Transfer Agent) 的程式，所以它並不負責處理 Mail 和使用者之間的介面，或是 Deliver Mail 的工作。

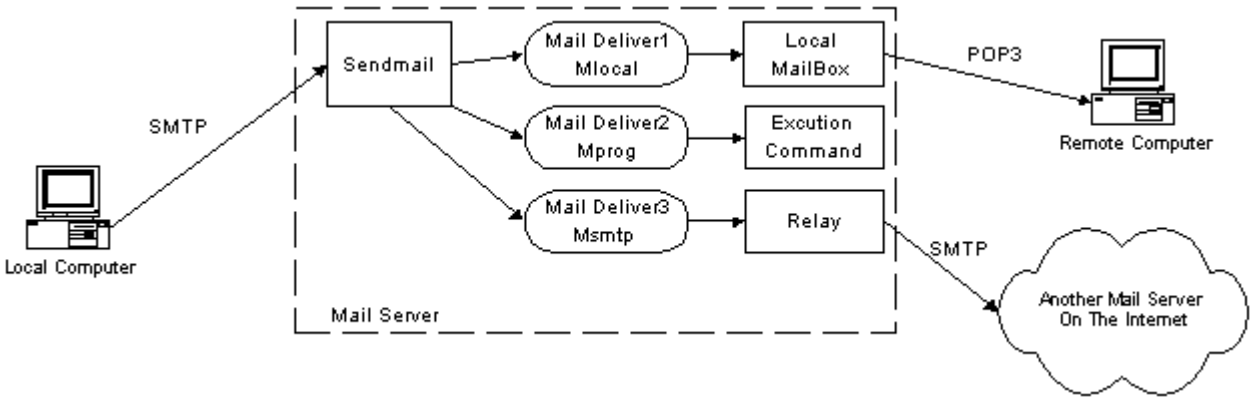


圖 3-2　Sendmail 組織架構圖

它的作法是，依據不同的 e-mail 的地址，將 mail 傳送的不同的 Mail 程式。所以我們可以在 Sendmail.cf (這個是 Sendmail 主要的設定檔，它決定了 Sendmail 是如何工作的) 中找到以下幾行:

```
Mlocal,
P=/usr/libexec/mail.local, F=lsDFMAw5:/|@qrmn9, S=10/30,
R=20/40,
T=DNS/RFC822/X-Unix,
A=mail -d $u
Mprog,
P=/bin/sh, F=lsDFMoqeu9, S=10/30, R=20/40,
D=$z:/,
T=X-Unix,
A=sh -c $u
Msmtp,
P=[IPC], F=mDFMuX, S=11/31, R=21, E=\r\n,
L=990,
T=DNS/RFC822/SMTP,
A=IPC $h
Mesmtp,
p=[IPC], F=mDFMuXa, S=11/31, R=21, E=\r\n,
L=990,
T=DNS/RFC822/SMTP,
A=IPC $h
Msmtp8,
P=[IPC], F=mDFMuX8, S=11/31, R=21, E=\r\n,
L=990,
T=DNS/RFC822/SMTP,
A=IPC $h
Mrelay,

P T=DNS/RFC822/SMTP,
A=IPC $h =[IPC], F=mDFMuXa8, S=11/31, R=61, E=\r\n, L=2040,
```

其中 M 代表是一個命令，用來設定 sendmail 使用的郵件傳送程式 (mailer)，命令語法如下:

Mname, {field=value}

- name 是 sendmail 參考這個 mailer 的名稱，其中有兩個 name 是固定且必要的:
  - local: 傳遞自己系統上的郵件給自已系統上的使用者的郵件程式，如 mail anchor
  - prog: 傳遞郵件給其它程式的郵件程式，如 mail ls@localhost ，其中 ls 在/ etc/aliases 中定義為
    ls: "| /usr/local/bin/ls"
- filed 參數設定
  - P: path 郵件程式的路徑
  - F: Flags 這個郵件程式所使用的旗標
  - S: Sender 發信人的 Rewrite Ruleset
  - R: Recipient 收信人的 Rewrite Ruleset
  - A: Argv 郵件程式的引數
  - E: Eol 一行結束的字串

於是當我們寫好以上的設定之後，sendmail 就可以根據以上的 mailer 設定將適當的 mail 送到指定的郵件程式，不過在這之前它們的 Recipient Adderess 和 Sender Adderess 會先被所指定的 Ruleset 進行 Rewrite 動作。

Ruleset 重寫規則

重寫規則是 sendmail.cf 中的檔案精華，透過這些規則，Sendmail 可以將使用者的郵件地址進行解析，然後加以適當的處理，甚至可以擋掉許多不必要的垃圾信，和不合法的 Request。

Sendmail Server 的檔案架構與基礎說明

Sendmail 幾乎所有的設定檔都安置在 /etc/mail 底下，不過，如果你是以 RPM 安裝的話，那麼還有所謂的 sendmail-cf 的設定檔，這個就是使用 M4 在進行 sendmail.cf 設定的程式！由於 Sendmail Server 所使用到的套件並不少，這包括有 sendmail, imap

以及 m4 等等，我們針對這些套件來談一談每個目錄與檔案下的資料吧！

設定檔：

Sendmail 的設定檔幾乎全部都在 /etc/mail 底下，但是也不一定！因為還需要看當初你建立 sendmail.cf 這個主要設定檔時，將各個檔案放置的地點而定！這部份可以使用 RPM 的方式來反查出你的設定檔案的路徑。Sendmail 與相關套件的設定檔與相關的說明為：

/etc/mail/sendmail.cf 或 /etc/sendmail.cf：

這個就是 sendmail 的主要設定檔，所有的參數都是他在管理的！但是，這個檔案內的各個設定被號稱為『天書』，所謂的天書就是『非一般人看的懂得！』，就連 sendmai 官方網站自行開發出來的設定程式也都『告誡大家不要手動編輯這個檔案』，所以在這裡也不需要討論此檔案的內容！但既然這個是主要設定檔，又不讓 USER 手動編輯，那要進行 sendmail 設定的修改就必須使用別的方法，也就是 M4 指令。m4 可以將簡單的一些環境設定參數，重新以內定的函式庫或者函式定義來『製作』 sendmail.cf 這個設定檔。sendmail 預設的 sendmail.cf 放置在 /etc/mail/sendmail.cf ，但是某些 Linux distributions 則將他改放在 /etc/sendmail.cf 。

/usr/share/sendmail-cf/cf/xxxx.m4 ：

剛剛提過 sendmail.cf ，由於這個檔案最好不要手動修改，所以需要使用到 m4 這個程式。 m4 可以將一個簡單的環境設定檔轉成 sendmail.cf ，那個環境設定檔就是 sendmail-cf 這個套件所提供的。在 Red Hat 7.x 的系統中，主要的環境設定檔就是 /usr/share/sendmail-cf/cf/redhat.mc 這個檔案。不過，在 Red Hat 7.3 以後的所有 Red Hat Linux 版本當中，這個檔案被移動到 /etc/mail/sendmail.mc 。

/etc/mail/local-host-names ：

這個檔案主要用來處理一個主機同時擁有多個主機名稱時候的收發信件主機名稱問題。這個檔案的用途可非常多。當主機擁有多個 HOSTNAME 的時候，例如一個主機擁有三、四個主機名稱，那麼是否每個名稱都可以用來做為收受信件的主機名稱呢？並非如此！如果主機名稱為 test1.your.domain 以及 test2.your.domain ，而且欲將這兩個 hostname 可以用在收受電子郵件，如此，就必需將這兩個名字都寫入 local-host-names 這個檔案當中，一個主機名字佔用一行。注意：沒有寫入這個檔案的主機名稱，那信件將無法正確的寄達這部主機。例如：www.vbird.adsldns.org、vbird.adsldns.org 這兩個主機名稱的 ip 都是相同的，也就是指向同一台機器上。假設這台主機名稱預設為 vbird.adsldns.org，那在預設情況下，寄給 userid@vbird.adsldns.org 都是 ok 沒有問題的！但是寄給 userid@www.vbird.adsldns.org 就會出現錯誤。其中原因是因為沒有告訴 MTA 除了 vbird.adsldns.org 這個主機名稱外，還有 www.vbird.adsldns.org 也是指向這台主機上。所以寄給 userid@www.vbird.adsldns.org 會出現錯誤，通常就是 mail loop to me，要不然就是不允許 relay 的錯誤情況。

/etc/mail/access.db ：

這個是『規定使用本郵件伺服器資料庫的使用者』，要轉成這個資料庫需要藉由 makemap 以及 /etc/mail/access 檔案的配合。這個檔案可以說是 Sendmail 裡面最重要的『使用者權限管理』的資料。在後面會有接續的說明。

/etc/mail/aliases.db 或 /etc/aliases.db ：

這個 aliases.db 是用來設定『信箱別名』的一個檔案！可以藉由這個檔案的設定來規範『群組收信』。不過，還需要藉由 aliases 及 newaliases 來做成這個檔案才行。

/etc/mail/statistics ：

這個檔案在記錄 Sendmail 收發信件的相關資訊，

執行檔：

Sendmail 的執行檔如下：

/usr/sbin/sendmail：

即為 sendmail 的主要執行檔。它會讀取 sendmail.cf 這個檔案的設定內容。在發送信件時，就使用到此程式。啟用此程式之後，預設的啟用的 port 為 25 。

/usr/sbin/ipop3d：

sendmail 的功能是在處理寄信問題，而 ipop3d 就是處理 client 的收信問題。如果 Mail Server 希望提供用戶端使用 Netscape 或 OutLook express 來收信，那麼就需要提供這個服務。這個服務的設定檔在 Red Hat 當中是在 /etc/xinetd.d/ipop3，如果是 Open Linux server 3.1.1 的話，那就會在 /etc/inet.d/imap 這個檔案中。注意： pop3 是由 imap 套件所提供的，並沒有包含在 sendmail 套件之中。

/usr/sbin/makemap：

主要將 access 轉成 access.db 的資料庫製作的執行檔。

/usr/sbin/mailstats：

將 /etc/mail/statistics 檔案讀出來的程式。可以查看到目前為止 Sendmail 工作共傳送、接收多少郵件。

/usr/bin/newaliases：

將 /etc/mail/aliases 轉成 /etc/mail/aliases.db 的執行檔。

/usr/bin/mailq：

用來觀察 /var/spool/mqueue 這個郵件暫存目錄的資料情況的指令！。

/usr/bin/m4：

這個就是將 *.mc 檔案轉成 *.cf 檔案的主要執行檔, 需要搭配 sendmail 原始碼，或者是 sendmail-cf 這個套件。注意： m4 是也需要額外的安裝的一個套件，sendmail 原本套件中並未包含 m4 這個套件！。

郵件相關目錄：

sendmail 接收下來的郵件放置的說明如下：

/var/spool/mail ：

這個是郵件『收受下來之後，每個使用者信件放置的目錄』，一個帳號會使用掉一個檔案，例如你的帳號為 test ，那麼你的信在 Server 中時，就是 /var/spool/mail/test 這個檔案了！此外，你的 POP3 的協定亦是使用這個目錄中的 mailbox 做為預設的郵件取得的檔案資料。

/var/spool/mqueue：

當郵件由於對方主機的問題，或者是網路的問題，而無法送出去時，那麼該封郵件將會暫時的存放在這個目錄下，然後主機會每隔大約 30 ~ 60 分鐘重新嘗試傳送一遍，通常設定在五天內該封信件還寄不出去，將會退給原發信者。

3.1.3 Procmail

Procmail 是 Linux 系統預設的當地郵件傳遞程式， Procmail 提共最強而且最為複雜的的 e-mail 過濾系統給 Linux 使用， Procmail 過濾器被使用者定義在 .procmailc 檔中，此外，系統管

理員可以定義系統使用的過濾器在 /etc/procmailc 檔中，這兩個檔的格式是一樣的，但是由使用者定義的 procmail 過濾器比系統管理員定義的還常用。

.procmailc 檔的單元有兩種型態：環境變數的指定以及郵件過濾規則，這個規則的名稱為 recipes ，環境變數的指定是簡單而且近似 shell 初始化手稿 .bashrc，例如，HOME=/home/craig 是有效的環境變數的指定。指定的敘述很少需要，因為系統的變數通常有正確的值。

.procmailc 檔真正的根據是 recipes，每個 recipes 的語法是：

```
:0 [flag] [:[lockfile]
[* condition]
action
```

每個 recipes 以 :0 開始，用來表示跟只是的敘述不停，:0 後面選擇性的旗標用來更改過濾器如何被處裡，以下為部分的旗標以及意義：

A	假如前導的 recipe 被視為真的時候執行此 reicpe
a	同 A 旗標的意義，除了前導 recipe 必須有完全成功地執行
b	傳遞訊息本文至目的，這是預設值
B	過濾訊息本文
c	建立這封郵件的 carbon 拷貝
D	測試是 case senstive，case 預設是忽略
e	假如前導的 recipe 的執行傳回錯誤時執行這個 recipe
E	假如前導的 recipe 沒有執行，執行這個 recipe
f	透過外部的過濾程式來傳遞資料
H	過濾訊息標頭，這是預設值
h	傳遞序席標頭到目的上，這是預設值
l	忽略這個 recipe 寫入的錯誤
r	在位保證郵件適當格式化時寫出郵件
w	檢視外部過濾程式的 exit 碼
W	同 w 旗標，但沒有顯示錯誤的訊息

在此簡單的介紹如何在 Mail Server 上做信件過濾。我們先選定以帳號 kao 下撰寫.procmail 檔，以下就以幾個在 procmail 檔裡撰寫信件過濾規則為例，說明如何使用 procmail 作過濾信件。

CASE 1：

寄件者是以 center 帳號為開頭的郵件位置，交信件轉擠到 centerwork 郵件信箱（可以是檔案或目錄)。

```
:0                ; L1
* ^From.*center      ; L2
work              ; L3
```

其中，.procmail 的基本撰寫格式是

L1 :0;擷取 sendmail 的 releset0 作為處理，規定上都是以:0 做開頭，其後可直接選項

L2 比對符合的 token；使用正則表示式做樣式比對。

L3 比對後所做的動作，如上例只寫 worl，表示將信件送入此 mailbox。

CASE 2：

若寄件者的帳號是 maillist，寄件主題是有關 perl 的議題，將信件轉寄到 mail.linux.test.edu.tw 的 kao、yu 帳號，而郵件則放置在 prel mailbox 裡。

```
:0 c                          ;L1
* ^From.*maillist             ;L2
* ^Subject.*perl              ;L2
! kau@mail.linux.test.edu.tw   ;L3

:0 c                          ;L1
* ^From.*mailist              ;L2
* ^Subject.*perl              ;L2
! yu@mail.linux.test.edu.tw    ;L3

:0 c                          ;L1
```

```
* ^Form.*maillist          ;L2
* ^Subject.*perl           ;L2
perl                       ;L3
```

注意到上面以 L1 所標示的部分，多了選項 C 字元表示郵件做一份複製。L3 的部分若是以！符號為開頭表示轉記 email。L2、L3 兩部分的敘述不限定只有一條規則。上力是較為簡單但繁複的寫法，事實上可以改寫成下列簡單的形式：

```
:0                         ;L1
* ^From.*maillist          ;L2
* ^Subject.*perl           ;L2
 {
   :0 c
   ! kau@mail.linux.test.edu.tw   ;L3
   ! yu@mail.linux.test.edu.tw    ;L3

   :0
   perl                    ;L3
}
```

將 L3 的部分做變化，即以:0 接 L1、L2 部分處理的結果，再於其下方寫上部分 L3 的規則。另外還可寫成下例：

```
:0 c                       ;L1
* ^From.*maillist          ;L2
* ^Subject.*perl           ;L2
! kau@mail.linux.test.edu.tw      ;L3
! yu@mail.linux.test.edu.tw       ;L3

:0 A                       ;選項 A 表示繼承前面的過濾規則
perl                       ;L3
```

CASE 3：

郵件主題不含有”餐飲特報”等字眼的信件，則將信件標頭附至

一份並且子 縮加入到 HEAD>GZ 檔。

```
:0 hwc                          ;L1
* ^Subject:.*餐飲特報              ;L2
| gzip >> head.gz                ;L3
```

其中，L1 選項 h 是處理標頭。選項 w 是以|作為導向處理郵件的過程中鎖住信件，直到信件處理完畢才解出 LOCK 狀態，以避免信件被移除。hwc 是用來 LOCK 信件並且複製信標頭。

L3 以|作為開頭，表示將信件做導向系統程式或命令處理，在本例中是將信件標頭揪由 gzip 壓縮並附加到 head.gz ，若不進行壓縮則可以改寫如下：

```
|cat >> head      ;L3
```

CASE 4：

當郵件主題含有 mailinglist 字串時，先 LOCK 郵件，再交由 formail 處理，郵件轉成標準格式後，將郵件附加到 list mailbox 裡。

```
:0:                          ;L1
* ^Subject:.*mailinglist     ;L2
| formail +1 –ds >> list     ;L3
```

L1 :0 後的：號表示 LOCK 郵件，以避免郵件被移動或刪除。

L3 formail 的選項 –ds 表示將所承接的 mail 無論如期格式為何，分成一項項的 mail message，再交由 formail 格式或郵件，在本例中 >> lsit 表示將 foramil 格式化郵件的結果輸出到 list 檔案。

3.1.4 Formail

基本上，formail 也是個過濾信件的程式，但它專精於制定(Format)郵件的格式，以符合各種不同的 MUA 所需要的 mailbox 格

式。與 procmail 不同的是，procmail 需要撰寫過濾信件規則，但是 formail 通常只以"formail 參數 1 參數 2 參數 3"的形式處理信件。而在 procmail 裡是允許將經過過濾處理之後的信件刁由 formail 處理。

如果 formail 並沒有加上任何的參數或其他的指令，則它將會把所有的信件轉成 mailbox 的格式，並且略過所由'>'開頭的行。

其餘我們有用到的參數：

-r 制定一個回覆信件的格式。
-i 在-r 制定完回覆信件的格式之後，由-i 來設定該信件德欄位內容，如收件者的地址等。
-k 將指定的訊息放在信件內文中。

3.2 分工情形

工作項目	負責人員
Mail Server 架設	余世鴻、黃紀豪
Procmail 程式	黃紀豪
POP3 程式	余世鴻
系統整合	曾嘉祥

表 3-4 分工表

3.3 甘特圖

	四月	五月	六月	七月	八月	九月	十月	十一月
選擇題目	●	●	●	●				
訂定目標			●	●				
閱讀資料				●	●			
系統建立					●	●		
Procmail 程式						●	●	
POP3 程式							●	●
系統整合								●
測試						●	●	●
撰寫報告						●	●	●

表 3-5 甘特圖

第四章　系統功能及說明

4.1系統功能介紹:

本系統的是利用MailServer模擬郵局的概念，配合Sendmail程式，實作出特殊的信件處理功能，以達成e-mail也具有實際信件往來確認的樣貌。

4.2 系統說明：

4.2.1　掛號信

掛號信回函會有兩封。

通常寄出 mail 後，有可能 Mail Server 並未收到。因此本系統的 Mail Server 收到一封掛號信時，系統會先回信通知寄件者，確認 Mail Server 信件已經收到，此為第一封回函。

而系統第二封回函則是在收件者在下載信件的時候，由 Mail Server 回信通知寄件者“收件者已經下載了信件”，以達到掛號信之目的與意義。

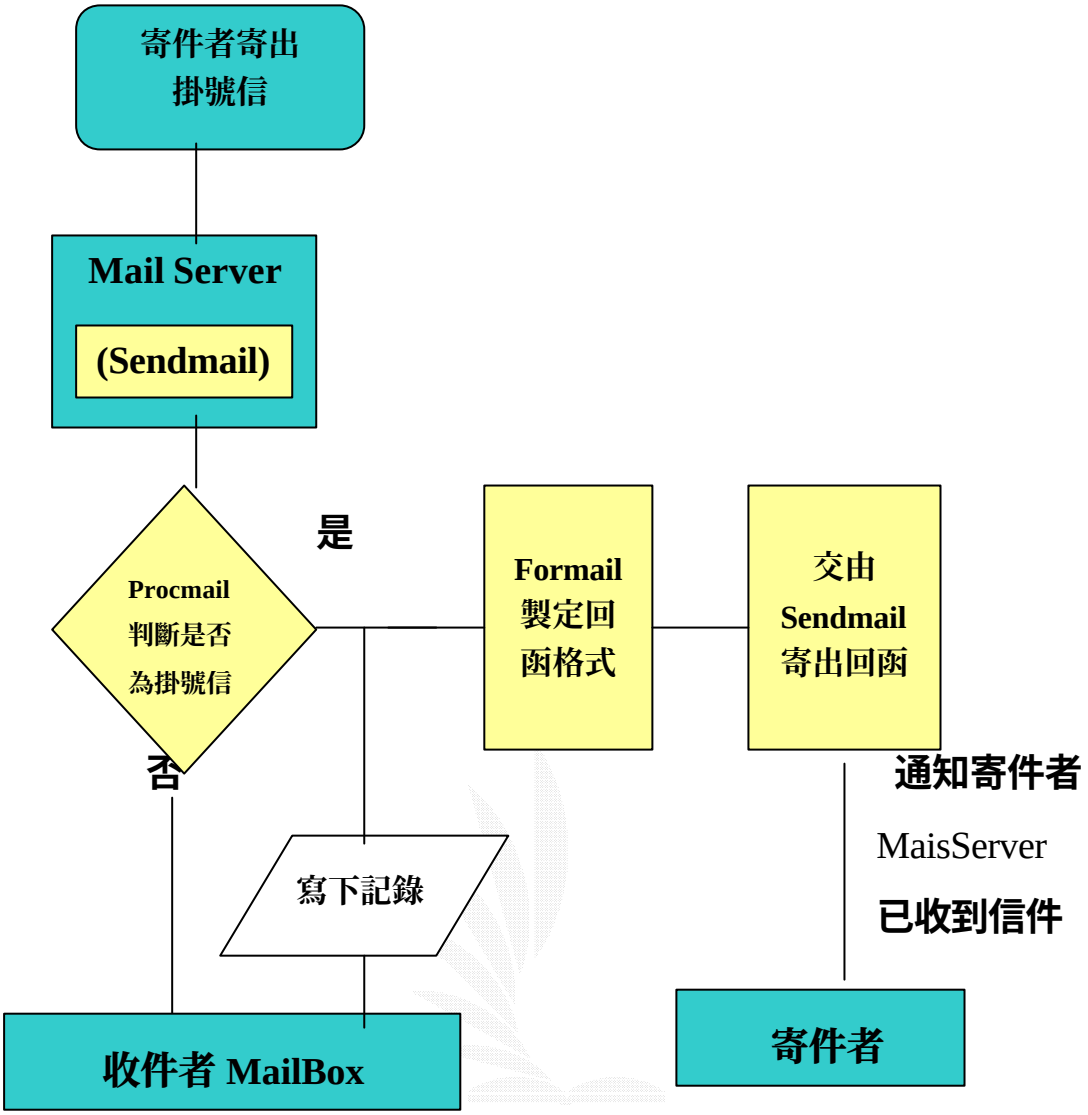


圖 4-1　第一封回函流程圖

系統在處理掛號信時，因為會寄出兩封回函，圖 4-1 為系統處理掛號信第一封回函流程圖，當寄件者要寄一封掛號信時，只要在主旨一開始的欄位加上<<<<<reply>>>>>(如圖 4-2)的字眼，代表此封為掛號信，如此當 MailServer 一收到這封信件時，便會自動回覆一封信件給寄件者，通知寄件者此封信件 MailServer 已經收到，並在 MailServer 裡存下記錄，以便未來查詢之用。

圖 4-2　掛號信寄件格式範例

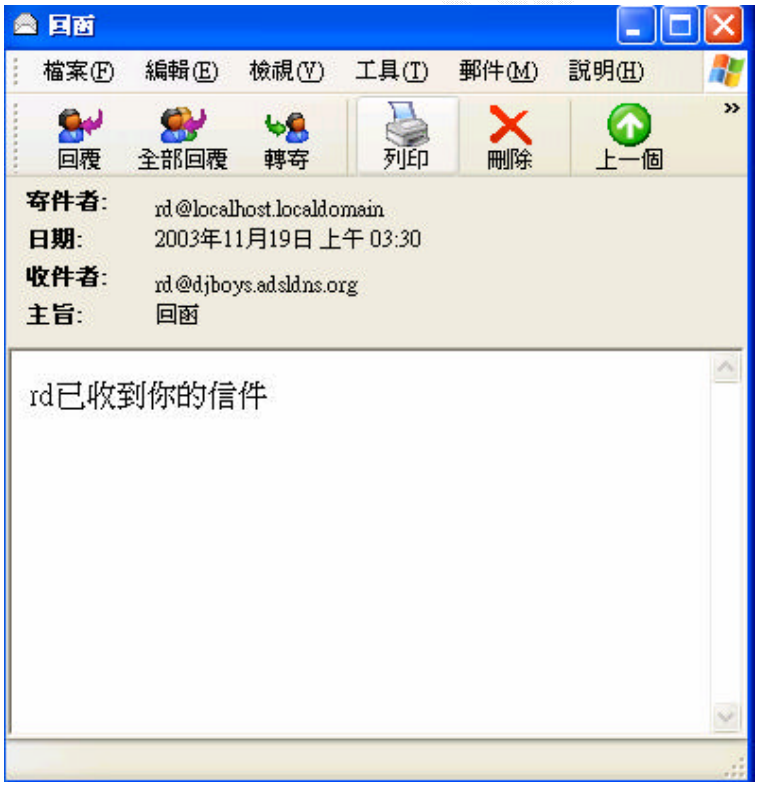


圖 4-3　掛號信回函之格式

圖 4-3 為掛號信回函之格式，當掛號信送達到 MailServer 時，

MailServer 便會自動回覆一封回函，通訊寄件者 MailServer 已收到此掛號信了。

圖 4-4 為系統處理掛號信第二封回函的流程圖，當收件者要下載掛號信閱讀時，MailServer 會自動再寄出一回函，此封為第二封回函，通知寄件者掛號信已被收件者下載，以達到雙重卻認的效果。

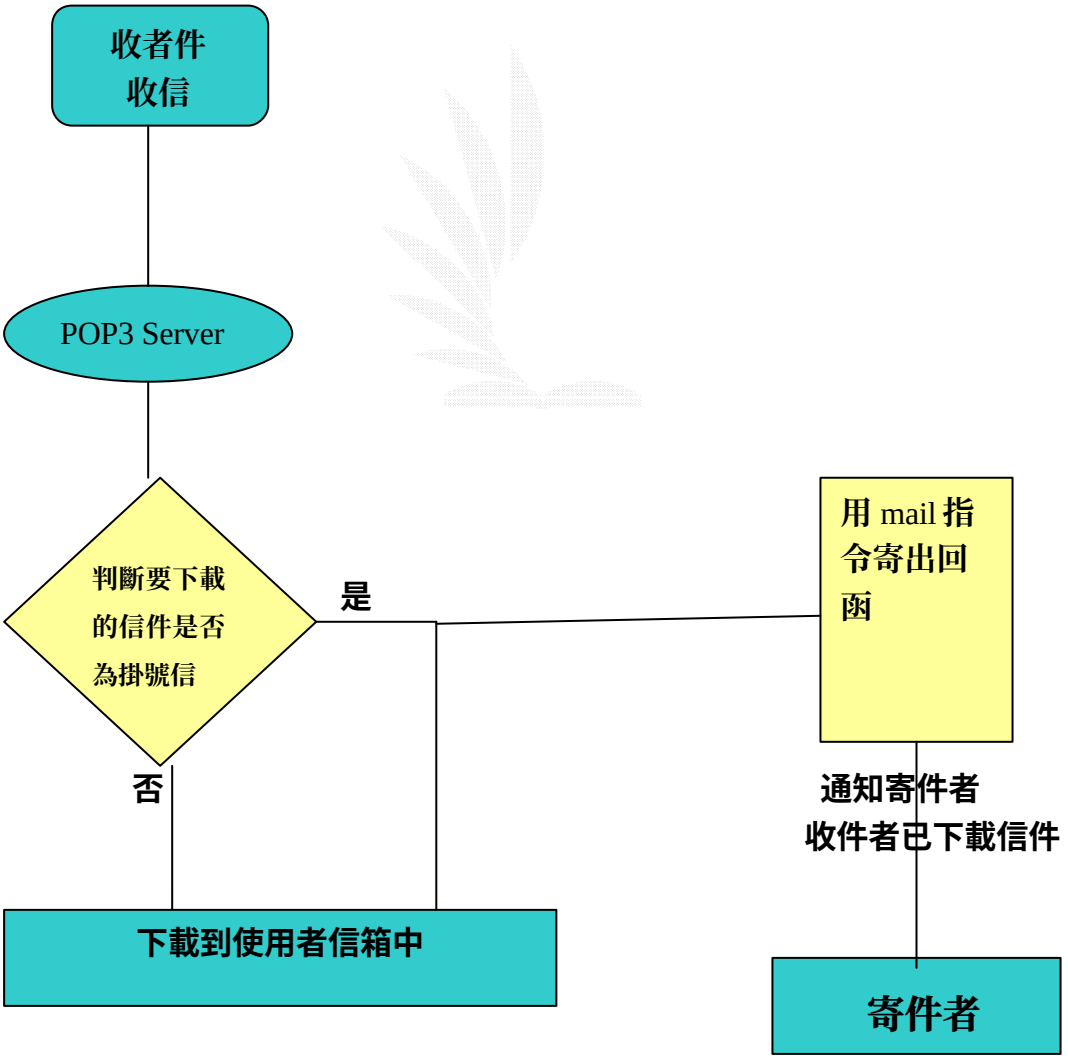


圖 4-4　第二封回函流程圖

4.2.2 垃圾信件的過濾

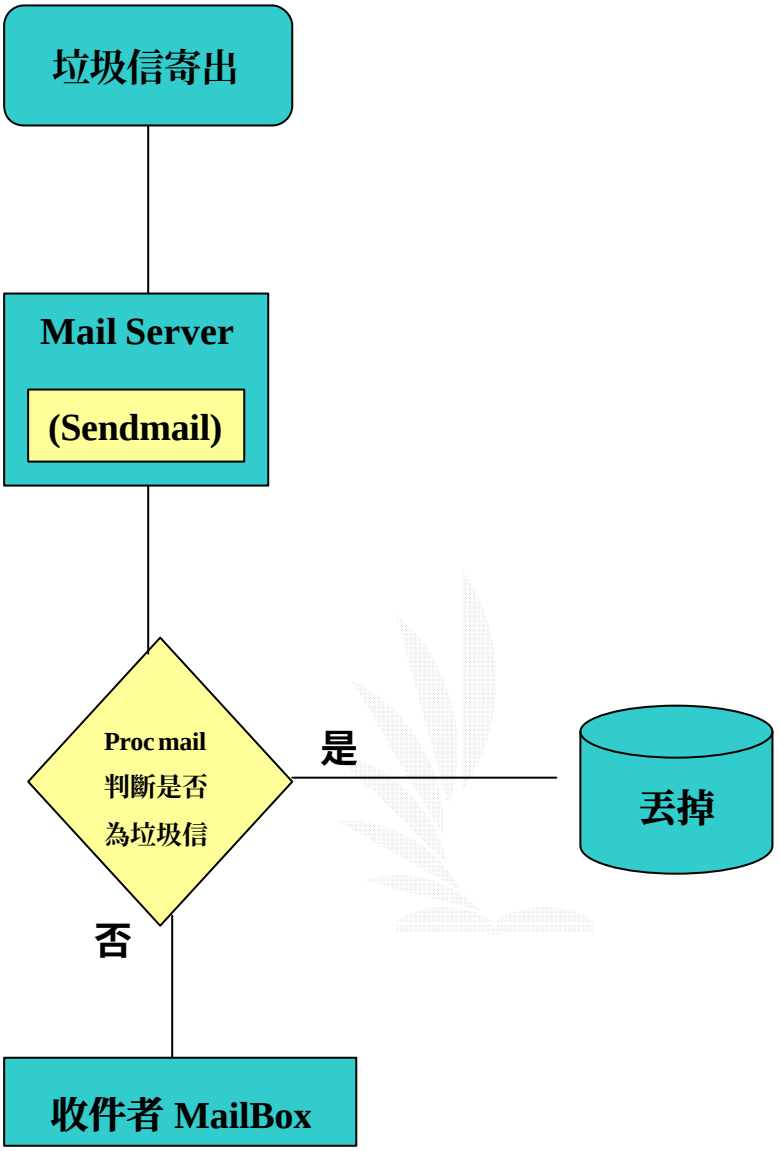


圖 4-5　系統擋垃圾信流程圖

圖 4-5 為系統處理垃圾信的流程圖，我們系統的 MailServer 有內建一些垃圾信件的規則，當的 MailServer 收到一信件時，若符合內建的垃圾信件規則的話，MailServer 便會把此封信件當作是垃圾郵件，直接把此封郵件刪除掉，不會再放到使用者的 MailBox 中，如此達到過濾垃圾信件的目的，使的使用者不會為垃圾信所困擾。

4.2.3 藉由寄信通知 MailServer可擋特殊的信件

圖4-8使用者要求擋特殊垃圾信流程圖，因為系統MailServer內建的垃圾郵件規則有限，且針對每一位使用者所要擋的信件不同，所以我們利用使用者寄信給MailServer，並在主旨一開始上打上<<<<<adsubject>>>>>後面加上要擋信件的主旨名稱(如圖4-6)，如此MailServer未來便會擋掉信件件主旨中有 ” 同樣是家具”的信件，若使用者想擋的是位址的話，同樣的，只要在主旨一開始上打上<<<<<adaddress>>>>>後面加上要擋信件的主旨名稱(如圖4-7)如此MailServer未來只要收到來自cloud@cloudx.adsldns.org的信件的話，也一樣會將它視為垃圾信件過濾掉。

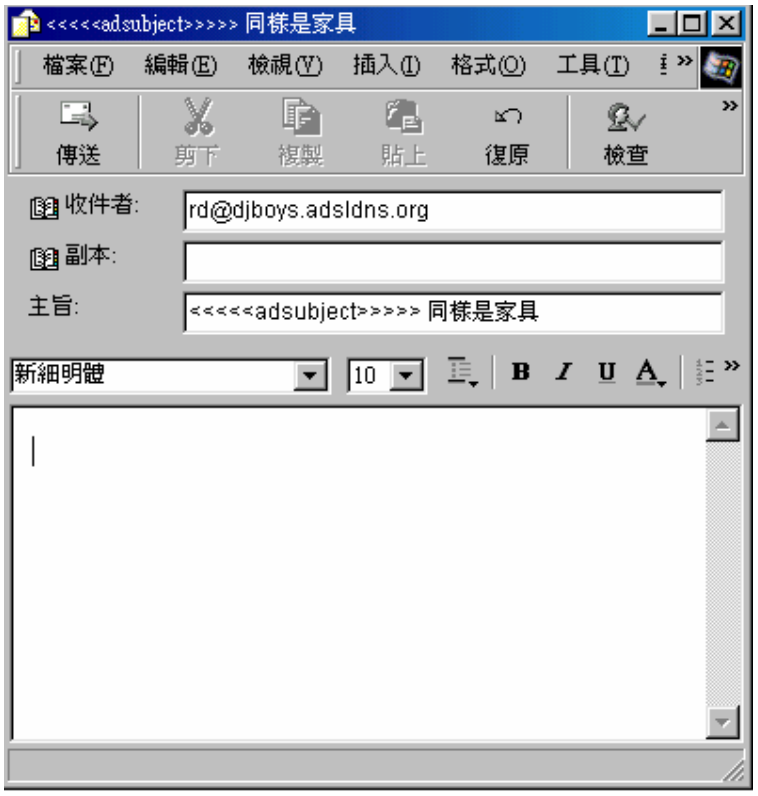


圖 4-6 擋特殊主旨信件的寄件格式

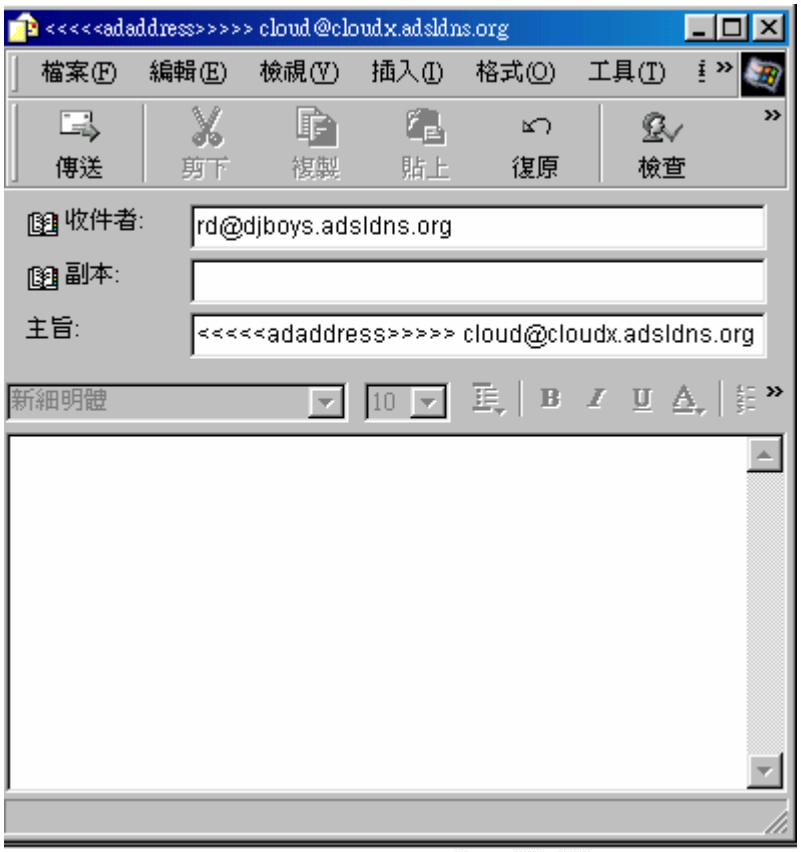


圖 4-7　擋特殊位址信件的寄件格式

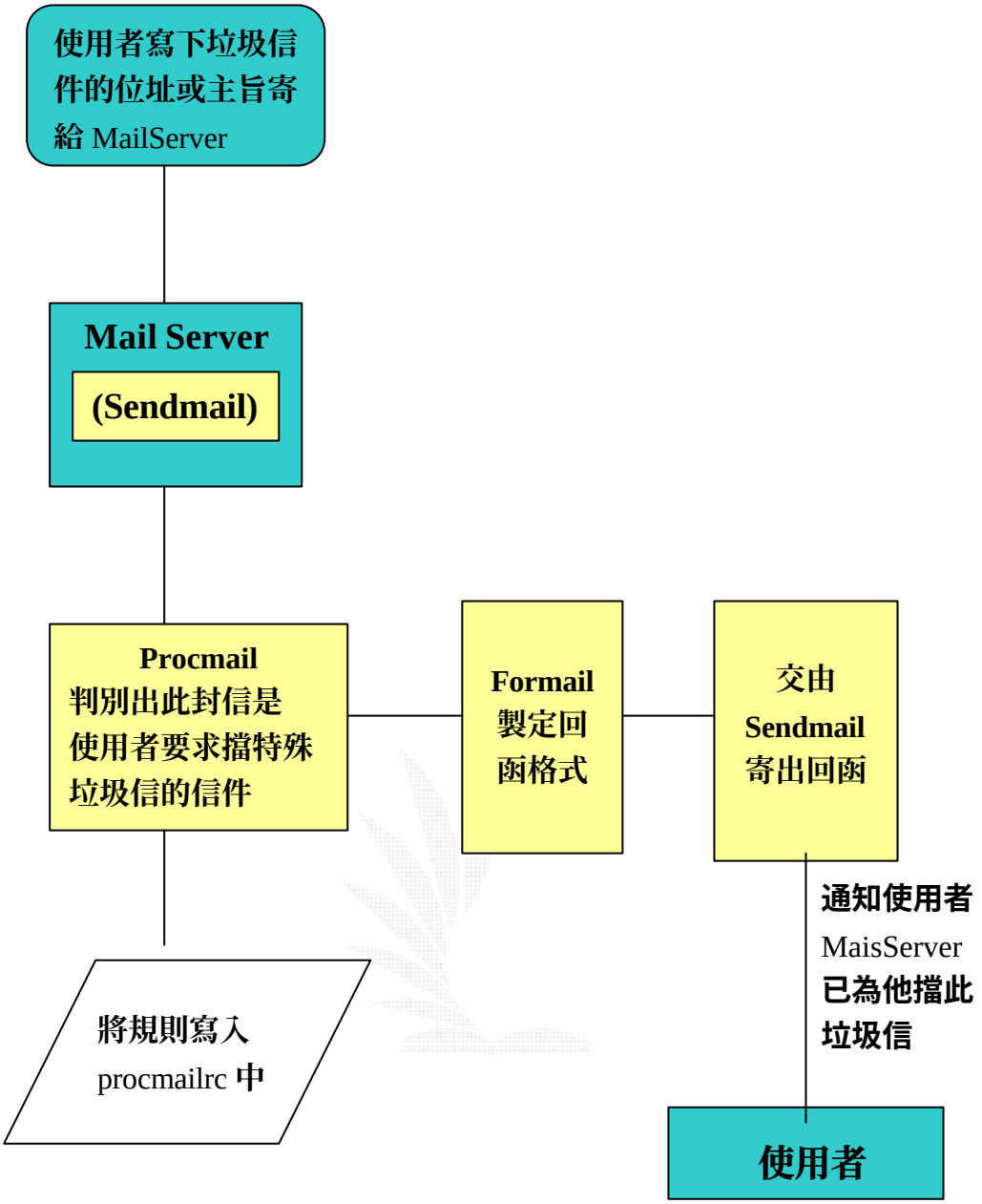


圖 4-8 使用者要求擋特殊垃圾信流程圖

圖 4-9、圖 4-10 為 MailServer 收到使用者要求擋特殊垃圾信的回函，MailServer 收到使用者要求擋特殊垃圾信時，會去更改 procmail 的規則，當規則改完後，便會再寄一封回函通知使用者，以便讓使用者做確認。

圖 4-9 MailServer 收到使用者要求擋特殊主旨信的回函

圖 4-10 MailServer 收到使用者要求擋特殊位址信的回函

4.2.3 高優先權掛號信

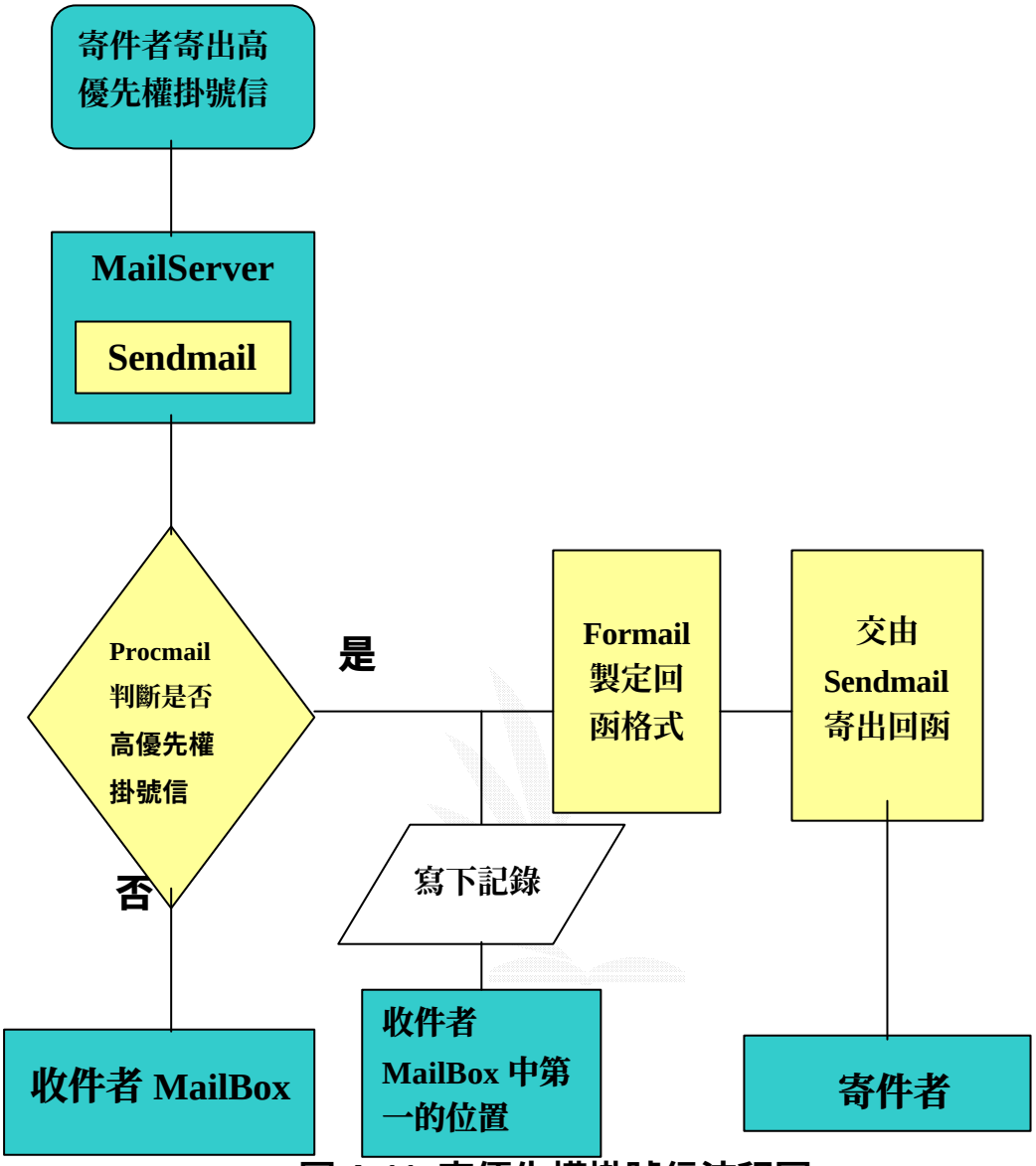


圖 4-11 高優先權掛號信流程圖

圖 4-11 為 MailServer 處理高優先權掛號信的流程圖，當寄件者想要寄一封可以有回信且優先權高的信件，只要在信件的主旨前面打上<<<<<first reply>>>>>的字眼(如圖 4-12 所示)如此收件者就可以在下載郵件時第一個收到此封信件，以避免重要信件被垃圾郵件擋住，以至造成重要信件無法在第一時間閱讀重要信件。

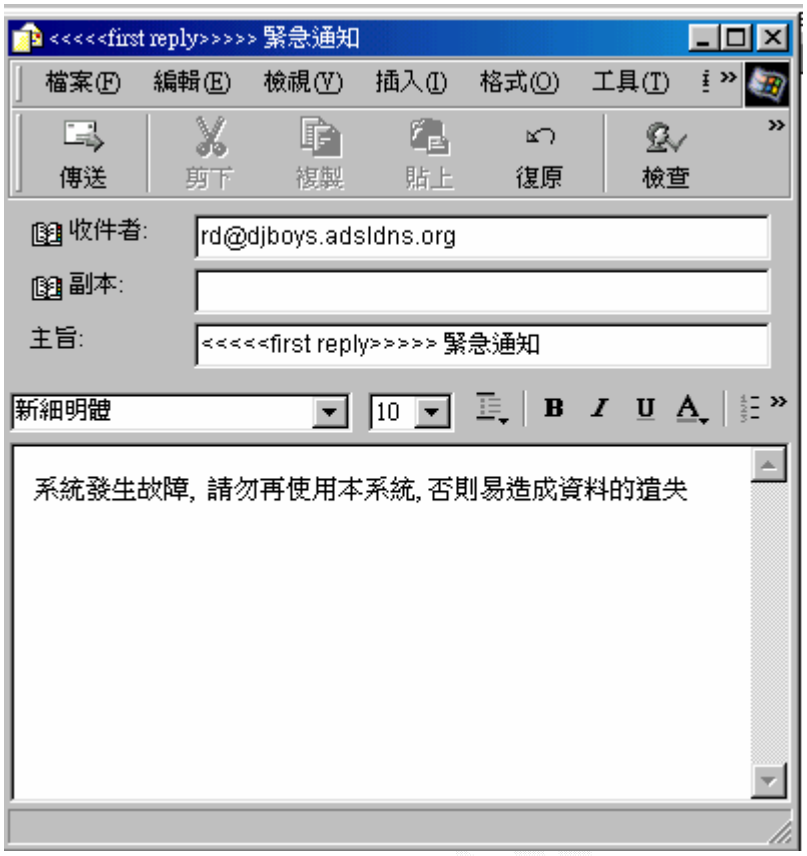


圖 4-12　高優先權掛號信的寄件範例

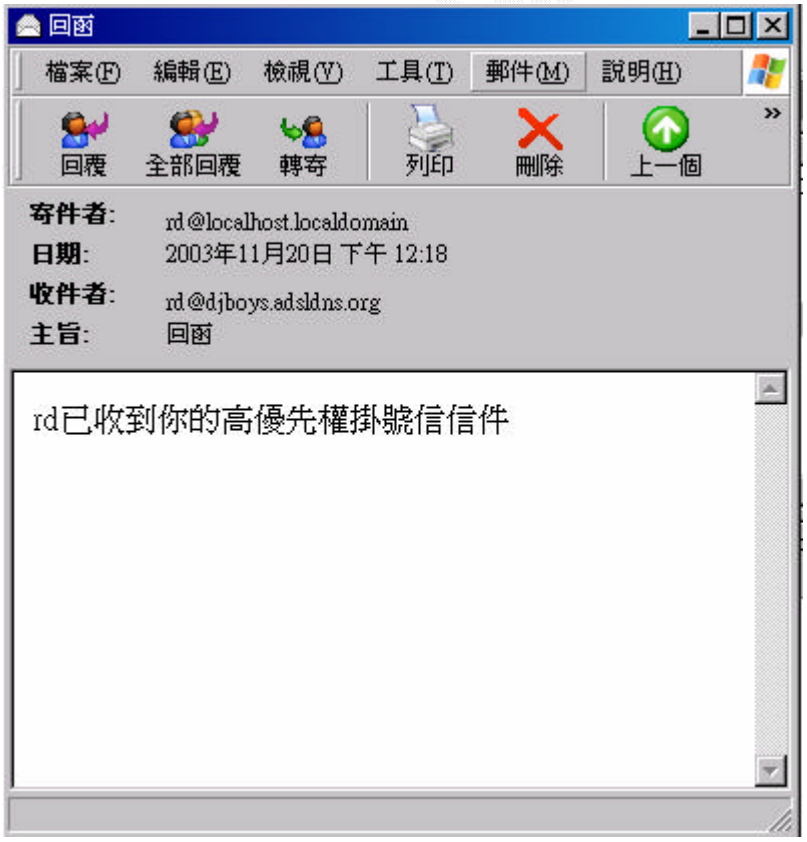


圖 4-13 高優先權掛號信的回函格式

4.3 系統實體架構:

4.3.1 主機配備

AMD 1000MHZ CPU、256MB Memery、Maxtor 60GB HD、10/100 乙太網路卡。

4.3.2 所採用的軟體套件

Item	Open Source Package	程式修改/自行設計
OS	● Chinese GNU/Linux Extensions for Red Hat 9.0 ● Kernel 2.4.20-8	
Mail	● Sendmail – 8.12.8-4 ● Imap – 2001a-18 ● Pop3 – 2001a-18	更改 pop3 原始碼，使得下載信件時得以回信
Check Mail	● Procmail ● Shell	撰寫 procmail 規則以達到特殊功能
其它	● Sed ● Formail	

表 4-1 系統採用的軟體套件

4.3.3 Procmail 的程式內容

我們藉由 procmail 的規則，以時做出我們所要的功能，當 MailServer 收到一封郵件時, procmail 會跟據 procmailrc 內容的規則去判別對此信件做何種處理，以下我 procmailrc 的內容:

```
MAILDIR=/var/spool/mail
VERBOSE=off
PATH="/usr/bin:$PATH:/usr/local/bin"

FROM=`formail -zxFrom:| sed 's/\(.*\) \(.*\)/\2/'`
sub=`formail -zxSubject:| sed 's/\("adsubject"\) \(.*\)/\2/'`
```

```
address=`formail -zxSubject:| sed 's/\("adaddress"\)
\(.*\)/\2/'`
TO=`formail -zxTo:|tr ',' '\n'| sed 's/\(.*\) \(.*\)/\2/'`
TIME=`formail -zxDate:`

#如果這封信給 a b c 三個人,在 to 的地方是寫成 a,b,c

#==================掛號信==============================
:0 c
* ^Subject.*<<<<<reply>>>>>
| ( if [ $LOGNAME = "rd" ] || [ $LOGNAME = "rd2" ]; \
then (formail -r -I"To:$FROM" -I"From:$LOGNAME" -I"Subject:
掛號信回函"; \
echo "$FROM     $sub    $TIME" >> /etc/mailfile \
\
echo "$LOGNAME已收到你的信件") \
| /usr/sbin/sendmail -oi -t ;\
fi )
#加了c代表郵件做一份複製
#所有來自rd且主題包函是"reply"的郵件
#先丟給formail格式化郵件並用-r表示回覆 k表示將原來信件放置
其中
#最後藉/usr/sbin/sendmail -t 來回覆
#=======================================================
#==================高優先權信==========================

:0
* ^Subject.*<<<<<first>>>>>
{
:0 c
/etc/temp

:0
|cat /var/spool/mail/$LOGNAME >> /etc/temp ; cp /etc/temp
/var/spool/mail/$LOGNAME; rm -f /etc/temp
}
```

```
#=========================================================
#================高優先權掛號信=========================
:0
* ^Subject.*<<<<<first reply>>>>>
{
:0c
/etc/temp

:0
| ( if [ $LOGNAME = "rd" ] || [ $LOGNAME = "rd2" ]; \
then (formail -r -I"To:$FROM" -I"From:$LOGNAME" -I"Subject:
掛號信回函"; \
echo "$FROM     $sub    $TIME" >> /etc/mailfile ; \
cat /var/spool/mail/$LOGNAME >> /etc/temp; cp /etc/temp
/var/spool/mail/$LOGNAME ; rm -f /etc/temp \
\
echo "$LOGNAME已收到你的高優先權掛號信") \
| /usr/sbin/sendmail -oi -t ;\
fi ) \
}
#=========================================================

#================標題擋信=================================
:0c
* ^Subject.*<<<<<adsubject>>>>>
|(formail -r \
-I"To: $FROM" \
-I"From: $TO" \
-I"Subject: 擋信回函"; \
echo ":0 \
\
* ^Subject.*$sub \
\
/dev/null \
" >>/etc/procmailrc \
```

```
\
echo "MailServer已為您擋去垃圾信$sub") | /usr/sbin/sendmail
-t -oi
#=========================================================

#==================位址擋信=================================
:0c
* ^Subject.*<<<<<adaddress>>>>>
|(formail -r \
-I"To: $FROM" \
-I"From: $TO" \
-I"Subject: 擋信回函"; \
echo ":0 \
\
* ^From.*$address \
\
/dev/null \:0
* ^Subject.*"first"
{
:0 c
/etc/temp

:0
|cat /var/spool/mail/$LOGNAME >> /etc/temp ; cp /etc/temp
/var/spool/mail/$LOGNAME; rm -f /etc/temp \
}

" >>/etc/procmailrc \
\
echo "MailServer已為您擋去垃圾信$address") |
/usr/sbin/sendmail -t -oi
#=========================================================

#=================擋廣告信=================================
:0
* ^From.*newsletter@condenet.com.tw
```

```
/dev/null
:0
* ^From.*sfene.9gjvr@hotmail.com
/dev/null
:0
* ^From.*OOilsnoopy@ms0.hinet.net
/dev/null
:0
* ^From.*F48TLkcpMypAI@disney.com
/dev/null
:0
* ^From.*yikrpm_OO@406411065246166rrlfenxed.net
/dev/null
:0
* ^From.*jlvsyju@nrxyfcsqnw.net
/dev/null
```

4.3.4 POP3的程式內容

當使用者用outlook等MUA的軟體下載信件時，POP3會自動辨別此信件是否為掛號信，若是的話，POP3會用mail指令寄出一封回函，通知寄件者，此封掛號信已被收件者下載讀取，而為了達到此目標，我們必須知道MUA是如何跟pop3 server溝通的，以下是MUA和pop3 server溝通的方式：

```
//使用者登入，輸入帳號和密碼
USER XXX
PASS XXX
//得到郵件清單及全部郵件大小
LIST
//跟據清單的訊息開始收郵件
RETR 1……RETR X
//將收下來的郵件刪除
DELE 1……DELE X
//作業完畢，登出伺服器
QUIT
```

由這溝通過程可知若想在使用者用MUA收掛號信時，寄一封回函給寄件者，可以利用修改RETR指令來達成，使得當MUA在下達RETR指令時，pop3 server會順便判斷此封信件是否為掛號信，如果是就回信。

但是有一種情況，就是當使用者不藉由MUA收信，而是自己登入pop3 server下達RETR指令看掛號信，使用者每下達一次RETR指令，伺服器就寄一封，萬一使用者一直下RETR指令看同一封掛號信，伺服器就會一直寄回函給寄件者通知他所寄的信件被下載了，這樣會造成寄件者的困擾，而且當信件較為龐大時，網路頻寬也會被佔據，為此，我們想出了一個解決辦法，就是在使用者的目錄底下建立一個紀錄檔，這個紀錄檔只存放使用者信箱內寄過的掛號信的編號，然後在使用者登入pop3 server時載入記憶體，以下我分段解釋程式的流程：

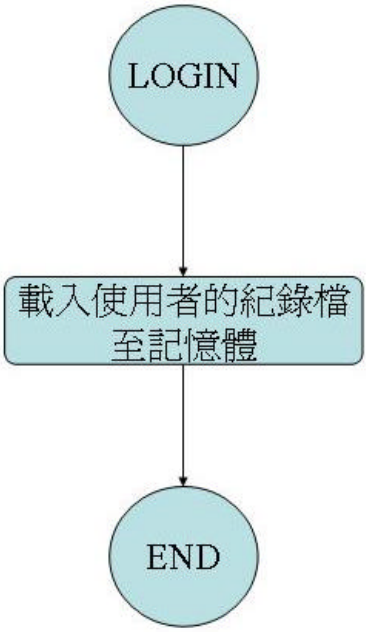


圖4-14 使用者登入pop3 server

如圖4-14所示，當使用者登入到pop3 server時，如果紀錄檔有內容，就將記錄檔載入至記憶體，載入至我所定義的結構中的num裡，然後其它的欄位預設值皆是0。

```
struct maildata
{
 char address[40];      紀錄此封信的寄件者位置
 char subject[40];      此封信的主旨
 long num;              此封信的編號
 long numd;             用來算出郵件最後的編號
 int stat;              郵件編號最後是否需變動
 int dele;              標明此封信是否被刪除
};
```

struct maildata usermail[10];定義一個結構為maildata的陣列，此陣列只存放掛號信的資料

接著當pop3 server收到RETR X的指令時，會如圖4-15的流程圖所示，首先比對maildata陣列每一欄裡的num欄位，看是否有相符合(=X)，如果有，就表示使用者要讀取的信件是掛號信且已經寄過了，就直接顯示信件內容；如果沒有找到相符合的(≠X)，表示這封信是第一次被讀取，接著server就將此封信的標頭(header)取出，然後呼叫我用lex寫的程式來取出此封信的寄件者位址及主旨，並判斷此封信是否為掛號信，如果不是掛號信，就直接顯示信件內容；如果是掛號信，並將此封信的資訊寫入usermail這個陣列中，然後回信。

此外，有一個情況，就是使用者在刪除掉前面一些信後才來讀此封掛號信，如下所示：

```
list
+OK Mailbox scan listing follows
1 644
2 646
4 644          假設此封為掛號信
5 646
```

第四封為掛號信，可是前面的第三封信已經被刪除了，所以這封掛號信正確來說，應該是第三封信，可是我們也不可以將此信直接在usermail裡的num(信件編號)欄位裡標明為第三封，倘若我們這樣

做，萬一server收到了一個RSET指令，要將全部標明為刪除的郵件還原，可是我們已經不知道這封掛號信原本是第幾封了，所以無法在usermail裡將此封掛號信改成第四封，造成了有usermail裡所紀錄的郵件和使用者mailbox裡的郵件不相同的情況發生。

為了解決這個問題，在此封掛號信內容被顯示之前，我們還有一個步驟，就是比對使用者在讀此封掛號信之前所刪除郵件的編號，如果發現到被刪除郵件的編號比掛號信小，那就將numd++，將stat設為1，一直到比對完所有被刪除郵件的編號為止，numd是一個counter，用來計算掛號信之前有多少信被刪除，而stat是設為1是表示此封掛號信前面有信件被刪除，所以最後要使用者登出，server寫紀錄檔時要記得給這封郵件正確的編號。當這一步驟做完，server就顯示此封掛號信的內容。

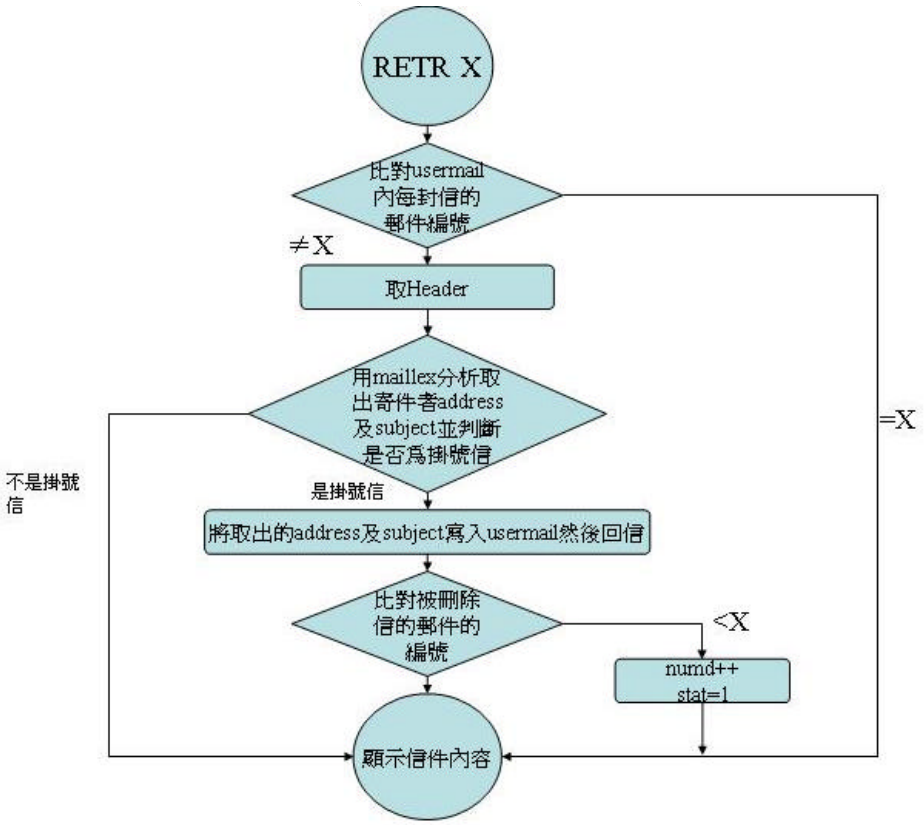


圖4-15 pop3 server收到RETR指令的流程

除了RETR指令需要修改，DELE的指令也同樣需要修改，以配合RETR指令及最後寫紀錄檔的工作。

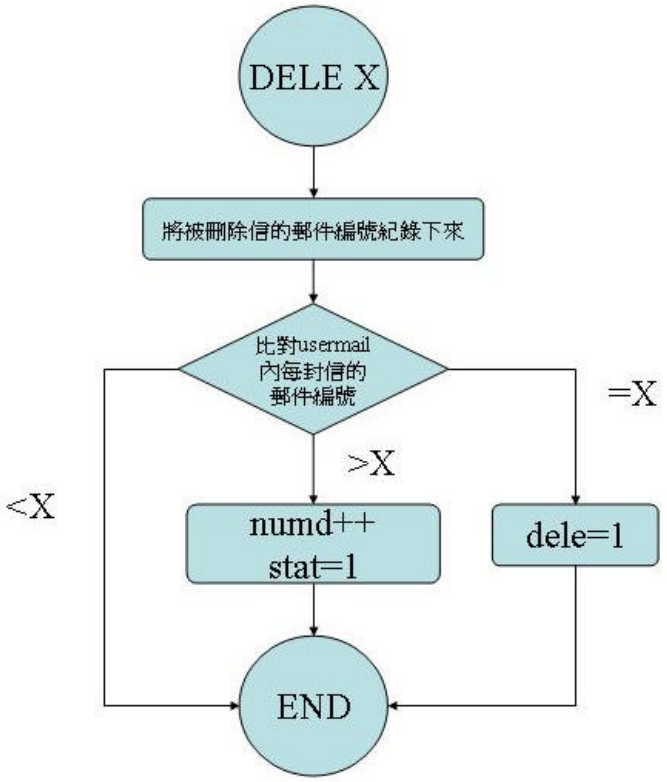


圖4-16 pop3 server收到DELE指令的流程

當DELE指令下達時，pop3 server會將被刪除郵件的編號紀錄起來，讓RETR指令流程裡的比對被刪除郵件的編號使用，接著server會比對usermail裡所有掛號信的的編號，發現有掛號信的編號比這封正被刪除的信件編號大(>X)，就將此封掛號信裡的numd加1(也就是圖中的numd++)，然後將此封掛號信裡stat設為1，原理和先前所解釋的相同，是為了標明此封掛號信的編號被更動了；如果發現刪除的郵件是usermail裡所紀錄著的掛號信(=X)，就將usermail裡此封掛號信的dele旗標設為1，表示使用者mailbox裡的這封掛號信被被標明為刪除了，所以這封掛號信在usermail裡的資料在最後使用者登出時可以不用寫入紀錄檔了；另外，發現usermail裡的掛號信編號小於正被刪除的信件編號(<X)，則甚麼動作都不用做。如此，當DELE指令下達時，pop3 server就一直重覆著上述的動作，直到usermail裡所有掛號信

的編號都被比對完。

接著就是當使用者下達RSET指令，將所有被標明刪除的郵件上的刪除標記取消。

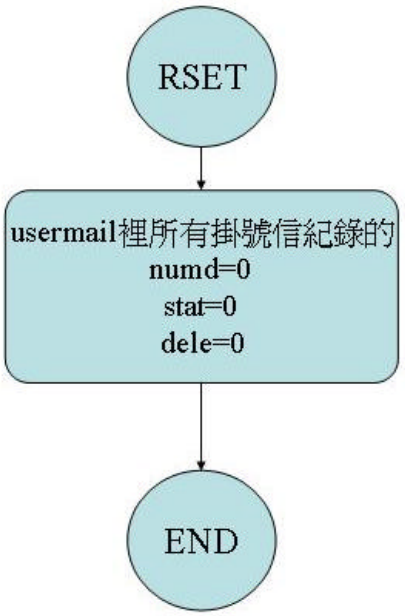


圖4-17 pop3 server收到RSET指令的流程

如圖4-17，當pop3 server收到RSET指令時，會將usermail裡所有掛號信紀錄裡的unmd、stat、dele欄位清為0，表示所有的信件都已還原, 所以先前被刪除的掛號信及需要更動編號的掛號信都不需被刪除、更動了。

最後就是當使用者登出時，下達QUIT指令時，pop3 server所需做的動作。

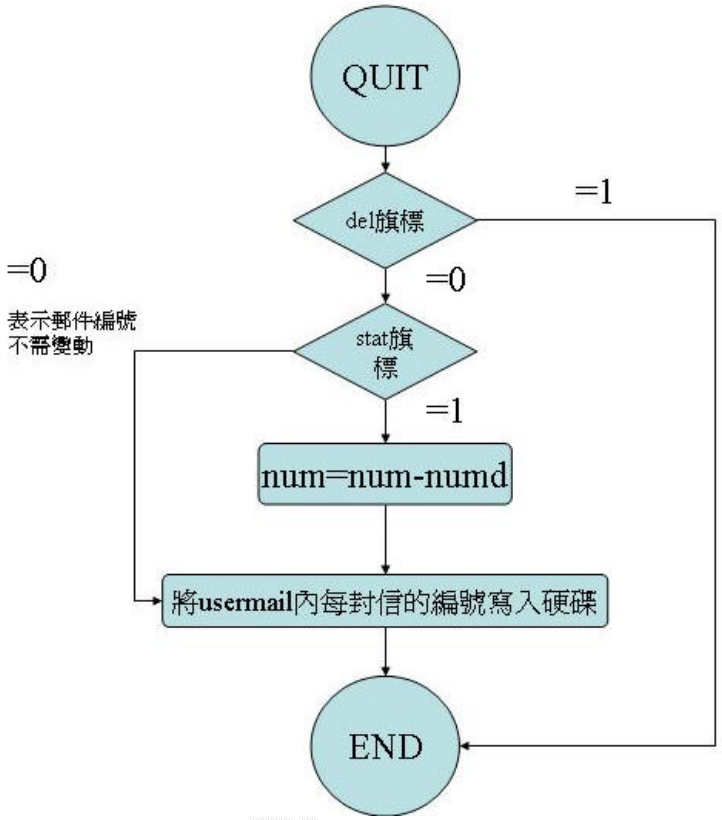


圖4-18 pop3 server收到QUIT指令的流程

當使用者下QUIT指令時，這時就是pop3 server決定usermail內哪些掛號信的編號該寫入紀錄檔內了，圖4-18顯示當伺服器收到QUIT指令時，它會比對usermail裡掛號信紀錄的del旗標，如果旗標為1，表示使用者mailbox裡對應的那封掛號信已被刪除，那pop3 server就不會將此封掛號信的編號寫入紀錄檔，然後就跳下一封信的紀錄檔；如果stat旗標為0，表示郵件編號不需更動，直接就可以寫入紀錄檔；如果stat旗標為1，表示此封信編號有變動，那pop3 server會將num減去numd，算出正確編號，再寫入紀錄檔。

以上就是我的程式流程介紹。

五、心得感想

5.1 遭遇的困難

5.1.1 DNS 取得困難

凡是 Server 都必須要和 DNS 打交道，要不然就像是要求郵差將信封依經緯度投遞到家裡的郵筒一般地困難。而我們最初所面臨的困難就是 DNS 的取得，因為現在網路上的 DNS 幾乎都必須付費，所以幾乎無法取得 DNS。在我們多次嘗試從許多管道是著解決問題之後，最後我們終於在專業的討論區問到了此網頁：
http://www.adsldns.org。雖然此 DNS 的提供網頁所限制的條件實在是嚴苛的不像話，但是看在免費的份上，以及並非長久使用的情況下，我們的第一個問題算是解決了。

5.1.2 資料難以取得

Sendmail、Procmail、SED 等無論在網路上或是及資料上，可供我們使用的雖然不算多，但是取得不算困難，而且中文方面的資料也算是充足。但是關於 POP3 內部運作方式、Formail 等就真的可算的上是鳳毛麟角了。先說 Formail 好了，在網路上所有介紹說明 Sendmail 以及 Procmail 的網頁，對於 Formail 雖然有提到，但皆是稍微的敘述一下其作用，甚至連有哪些參數可以使用都沒有說明就帶過了。書籍也是一樣，比起網頁的情況沒有好多少。於是我們只號硬著頭皮從 Sendmail 所附的 Formail 說明文件下手，當然，全都是英文的。但是若是單純的英文那也沒什麼，硬著頭皮即可將他翻譯出來，而問題是在於，不知道是我們的英文能力不足，或是文中的文句的問題，在我們看來，說明文件有許多關鍵部分用的是模糊的字眼啊！因此我們只好慢慢地一個一個的參數試驗，再配合著說明文件上我們看的懂的部分試出我們需要的參數。即使是如此的辛苦，我們還是解決了此項困擾

接下來就是 POP3 的內部運作問題了。一般而言，POP3 的說明大部分都是概略式的說明方式，但是我們所需要是 POP3 內部詳盡的運作方式，例如參數的傳遞等。令人遺憾的是我們無法取得相關的資料，所以只好從 POP3 程式的 Source Code 下手。這裡是我們所遇到

的最大的困難。

5.1.3 時間的急迫

這就是我們的疏失了。大部分的原因還是在於我們誤解了專題的意義了。大學的專題就是要將所學過的知識以及技巧在有可用的資源下選擇一個可以較為深入探討的主題來發揮。但是我們卻從一開始就往我們不熟悉的領域挺進，導致跌跌撞撞，專題失敗了一個又一個。原本在三年級下學期就應該開始的專題，卻一直到接近暑假才算真正的開始。不過，至少在放棄一個專題題目之前，我們都是已經至少盡我們所能地瞭解過了之後，經過再三的思量，真的覺得不可行才放棄的。雖然在專題上我們是浪費了一些時間，但我們卻在這些失敗中了解了其他更多資訊相關領域。

5.2 心得感想

在這次的研究專題中，首先我們學習到如何使用linux，因為老實說，在我們研究這項專題之前，我們根本都沒碰過linux，後來慢慢的，linux摸熟了，我? 開始學習使用linux架設MailServer，在架設的過程中，我們遇到了許許多多大小的問題，dns的問題，localhost的問題，pop3的問題，這些許許多多的問題，曾經讓我們很氣餒，有求助無門的失落感，後來，我們不放棄，我們開始上網問別人，開始尋求助教的幫忙，慢慢的，我們了解了問題的所在，最後，我們的人生第一個MailServer出現了，它終於可以順利的收信寄信了，但當我們把MailServer架設起來時，時間已經到達了9月，我們開始發現，時間已經所剩不多了，這時，我們開始有警戒心，開始分配工作，開始正式的研究我們的專題，
在研究的過程中，我們發現資料的收集不是那麼的容易，畢竟大部份的人總認為MailServer能收信，寄信就好了，沒有像我們有如此的鴻圖壯志，就這樣我們原地踏步了好一陣子，有一天，我們在圖書館發現了一本書，從那本書中，我們認識了procmail，後來就造就了我們利用procmail來做出我們的專題，也才造就出這份報告。

在這次專題研究中，我們對linux有深一步的認識與體驗，對於架設MailServer我們有很深的認識，這包括了MailServer的運作流程，信件在網路上的傳遞過程，以及POP3的運作，我們都有深刻的體

驗，這些都還只是有形的體驗，在無形的體驗上，我們了解到分工合作的重要行，很多事情是需要大家一起合作才能完成的，我們更了解到事情的成功，資料的收集是佔有很大的一環的，最後，我們的專題能順利趕出來，我們要謝謝我們的指導教授-李維斌老師的督導，以及助教的熱心幫助，藉由他們的教導，讓我們順利完成了我們的專題，也讓我們穫益良多。

5.3 系統未來展望

此系統目前開發只有在本MailServer內的使用者才具有這些功能，若未來每一個MailServer都加上了這些功能，相信這對所有的使用者都能使用到這些功能，如此e-mail就能像傳統信件般具有雙掛號的信件，本系統只是做一先驅示範，希望未來所有MailServer皆能跟進，並再增加信件與使用者的介面程式，如此使得使用者更容易操作，MailServer管理者更易於管理。

參考資料

[1]Sendmail-基礎篇 / Bryan Costales with Eric Allman原著;汪若文 譯

[2]Sendmail-安裝管理 / Bryan Costales with Eric Allman原著;汪若文 譯

[3]Sed & awk程式設計 / Dale Dougherty & Arnold Robbins原著;陳爽 譯

[4]Linux程式設計教學手冊 / Richard Stones & Neil Mathew原著;吳德銘 譯

[5]Linux Red Hat 9 實務應用/ 施威銘研究室 著

[6]鳥哥的Linux私房菜 / 鳥哥 著

[7] http://www.fanqiang.com/a6/b9/20010929/1305001372.html

[8] http://www.imap.org/

[9] http://www.tceb.edu.tw/~aie/linux/mail.htm

[10]http://www.freebsd.org.hk/html/aspac/reports/96/96005/

[11] http://linux.vbird.org/linux_server/0380sendmail.php

[12] http://www.procmail.org

[13] http://www.sektorn.mooo.com/era/procmail/mini-faq.html