



---

第 28 章  
Proxy Server

---

## 第 28 章 Proxy Server

### 28-1 Proxy 代理伺服器概論

我們可以使用 proxy Server 來當作代理伺服器，將遠端的資料存到我們的代理伺服器，這樣檔案的傳輸就不用經過很長的距離，而只要在近端就可以得到資料了。這樣可以縮短存取檔案的時間。

我們使用 `rpm -qalgrep squid` 就可以查尋得到我們 Linux 上面所裝的伺服器是哪一種。

```
[root@flash chaiyen]# rpm -qalgrep squid
squid-2.5.STABLE1-2
```

下面是我們 Proxy 代理伺服器抓取資料的圖。我們經常使用網際網路，從我們的個人電腦讀取遠端的資訊，如果網站在外國很遠的地方，那麼我們讀取資料就要花費很多的時間來讀取，因為資料傳送的路徑很長，如果在台灣就有網站提供，或者在我們的附近就有網站伺服器提供，這樣路徑就可以縮短，而我們讀取的時間就可以大幅度的縮短。Proxy 代理伺服器可以將很遠網站的資料暫時存到我們的 Proxy 代理伺服器中，當我們要讀取很遠網站的資料時，會先經過 Proxy 代理伺服器，當代理伺服器中有我們所要的遠端網站資料，則我們就會直接向 Proxy 代理伺服器請求 Request，則 Proxy 代理伺服器就會將資料送給我們。

我們可以安裝我們的 Proxy 代理伺服器。我們如果再安裝時選取完整安裝，這時 Proxy 就已經安裝在我們作業系統了。

首先我們先將 RedHat Linux Fedora 作業系統的光碟片放入光碟機，再使用掛載

```
#mount /mnt/cdrom
```

我們再到放置 squid 套件的地方，然後安裝 squid，也就是 Proxy 代理伺服器的套件。

```
#cd /mnt/cdrom/RedHat/RPMS
```

```
#rpm -ivh squid-*. *是指 Proxy 代理伺服器的版本。
```

如果我們要啟動 Proxy 代理伺服器，我們可以輸入 `/etc/rc.d/init.d/squid start` 來啟



動 Proxy 代理伺服器。

```
[root@flash etc]# /etc/rc.d/init.d/squid start  
啓動 squid: [ 確定 ]
```

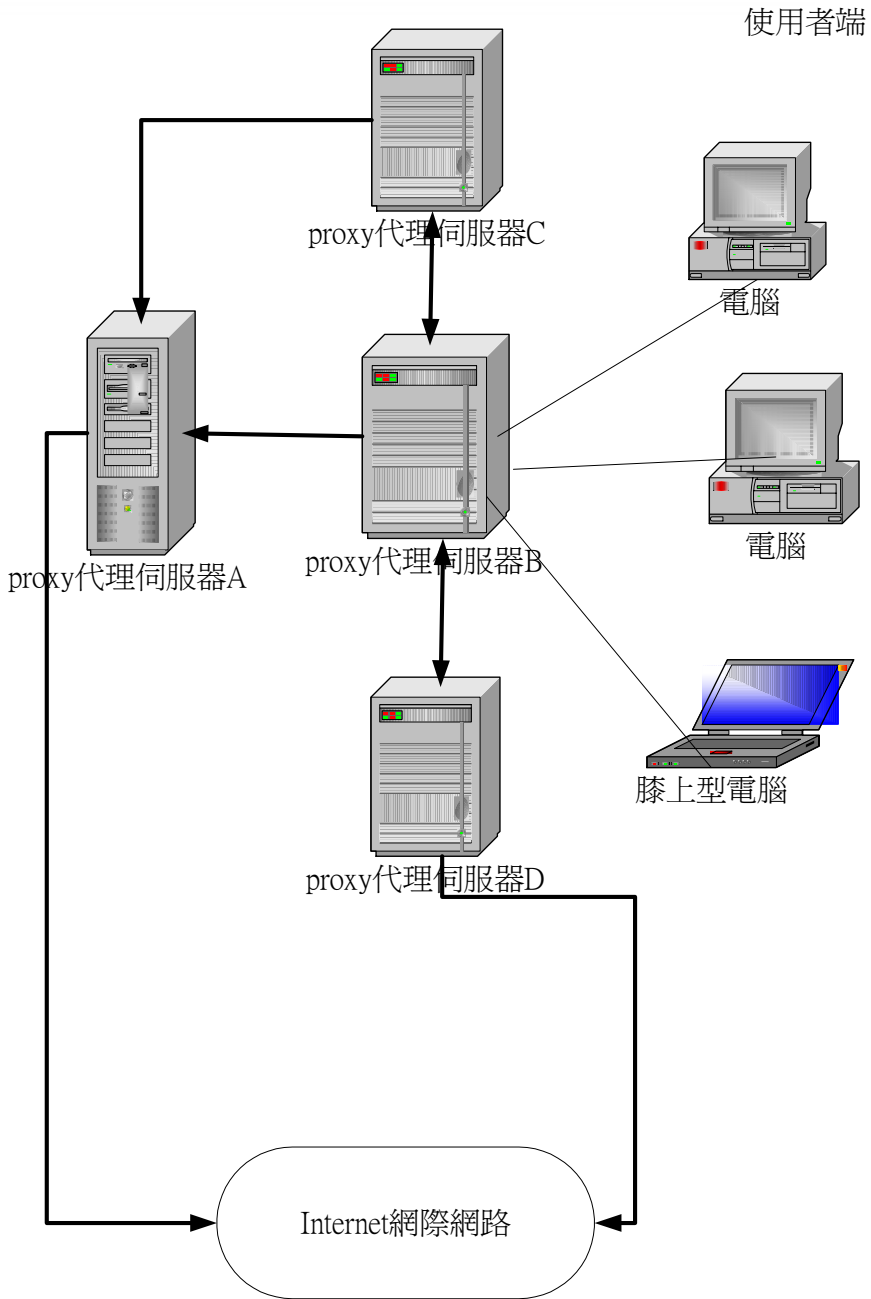
<http://www.squid-cache.org>我們也可以從這個網站下載最新的 Squid 代理伺服器。



我們的個人電腦經過 Proxy 代理伺服器，然後到遠端的網際網路去讀取資料，如果在這我們近端的 Proxy 代理伺服器已經有資料了，則就會直接到近端的 Proxy 伺服器讀取。如果近端的 Proxy 伺服器都沒有，則要到很遠的網站去載入網頁或資料到我們的 Proxy 伺服器中，再將它載入到我們的電腦，這樣我們下次就可以直接到我們的 Proxy 代理伺服器去讀取相關網站的檔案和資料了。Proxy 伺服器就像是 cache 暫存器的功能，它可以暫時存放我們經常讀取的資料，讓我們可以很迅速的就可以讀取到我們所要的檔案和資料。

我們的 Proxy 代理伺服器，可以分為好幾台，而它們是有階層 Hierarchy 的關係。我們的 Proxy 代理伺服器 B 和 Proxy 代理伺服器 C、Proxy 代理伺服器 D 是同一層的兄弟關係，它們可以在同一個網域內。而我們的 Proxy 代理伺服器 A 它是代理伺服器 B、C、D 的上一階層 Proxy 代理伺服器，也叫作父節點，當我們在 B、C、D 代理伺服器都讀不到資料時，再到 Proxy 代理伺服器 A 去尋找，如果再找不到，再到更上層去，直到找到或直接到指定的遠端網際網路網站上去尋找。





我們可以設定開機時自動啟動 Proxy 代理伺服器，我們使用 setup 指令來設定開機時自動啟動。

```
root@flash:~ - Shell - Konsole
工作階段 編輯 檢視 設定 說明
[root@flash root]# setup
```

我們選取 System services 系統服務。

```
root@flash:~ - Shell - Konsole
工作階段 編輯 檢視 設定 說明
Text Mode Setup Utility 1.8 (c) 1999-2001 Red Hat, Inc.
Choose a Tool
Authentication configuration
Firewall configuration
Keyboard configuration
Mouse configuration
Network configuration
Printer configuration
System services
Sound card configuration
Timezone configuration
X configuration
Run Tool Quit
<Tab>/<Alt-Tab> between elements | Use <Enter> to edit a selection
新增(N) Shell
```

我們將 squid 選取，則這樣開機時就能夠啟動 Proxy 代理伺服器了。

```
root@flash:~ - Shell - Konsole
工作階段 編輯 檢視 設定 說明
htsysv 1.3.5 - (C) 2000-2001 Red Hat, Inc.
那些服務要被自動啟動?
[*] smartd
[*] smb
[*] snmpd
[*] snmptrapd
[*] squid
[*] sshd
[*] swat
[*] syslog
確定 取消
請按 <F1> 查看該項服務的更多資訊。
新增(N) Shell
```



Proxy 伺服器所能暫存的資料大都是 http 的網頁、ftp 的檔案、gopher 的資料，而最經常使用的 Proxy 代理伺服器以 Unix 系統下的 Squid 最多人使用，Squid 能夠支援的傳輸協定有 http、ssl、ftp、gopher、pop3、SMTP(郵件傳輸協定)、影音傳輸協定。

Proxy 代理伺服器因為要儲存我們遠端的網頁、檔案和資料，所以它的儲存設備要夠大和存取速度要夠快，這樣我們才能夠快和精準的從 Proxy 代理伺服器存取檔案。以我們自己的規畫，我們是以 PIII866 的 CPU 在跑，記憶體基本是 512MB，當然越多且越快越好。而硬碟是 MAXTOR 40GB，硬碟也是越大越好，存取速度越快越好。

我們可以編輯 Proxy 代理伺服器的設定檔。我們使用 vi 編輯器編輯 /etc/squid/squid.conf。

```
[root@flash etc]# vi /etc/squid/squid.conf
```

這是我們內定放置相關 Proxy 代理伺服器組態檔的目錄是/etc/squid。

```
[root@flash chaiyen]# cd /etc/squid
```

```
.
|-- errors -> /usr/lib/squid/errors/English
|-- mib.txt
|-- mime.conf
|-- mime.conf.default
|-- squid.conf
`-- squid.conf.default
```

errors 是顯示錯誤時的語系，我們可以將 errors 連結到 /usr/lib/squid/errors/Traditional\_Chinese 來改變錯誤時顯示的語系。

Mib.txt 是 SQUID-MIB 定義檔。

Mime.conf 定義 squid 的 mime 型態。

Squid.conf 為 squid 的組態設定檔。

Squid.conf.default 為 squid 的內定組態檔。



這是 Proxy 代理伺服器的組態設定檔/etc/squid/squid.conf，僅列出重要部份。

```
1
2 # WELCOME TO SQUID 2
3 # -----
4 #
5 # This is the default Squid configuration file. You may wish
6 # to look at the Squid home page (http://www.squid-cache.org/)
7 # for the FAQ and other documentation.
8 #
9 # The default Squid config file shows what the defaults for
10 # various options happen to be. If you don't need to change the
11 # default, you shouldn't uncomment the line. Doing so may cause
12 # run-time problems. In some cases "none" refers to no default
13 # setting at all, while in other cases it refers to a valid
14 # option - the comments for that keyword indicate if this is the
15 # case.
16 #
17 #
18 #
19 # NETWORK OPTIONS
20 # -----
21 #
22 # TAG: http_port

23 # Usage: port
24 #         hostname:port
25 #         1.2.3.4:port
26 #
27 # The socket addresses where Squid will listen for HTTP client
28 # requests. You may specify multiple socket addresses.
29 # There are three forms: port alone, hostname with port, and
30 # IP address with port. If you specify a hostname or IP
31 # address, then Squid binds the socket to that specific
32 # address. This replaces the old 'tcp_incoming_address'
33 # option. Most likely, you do not need to bind to a specific
34 # address, so you can use the port number alone.
35 #
36 # The default port number is 3128.
37 #
38 # If you are running Squid in accelerator mode, then you
39 # probably want to listen on port 80 also, or instead.
40 #
41 # The -a command line option will override the *first* port
42 # number listed here. That option will NOT override an IP
43 # address, however.
44 #
45 # You may specify multiple socket addresses on multiple lines.
```



```
46 #
47 #Default:
48 # http_port 3128
49
50 # TAG: icp_port
51 # The port number where Squid sends and receives ICP queries to
52 # and from neighbor caches. Default is 3130. To disable use
53 # "0". May be overridden with -u on the command line.
54 #
55 #Default:
56 # icp_port 3130
57
58 # TAG: htcp_port
59 # Note: This option is only available if Squid is rebuilt with the
60 # --enable-htcp option
61 #
62 # The port number where Squid sends and receives HTCP queries to
63 # and from neighbor caches. Default is 4827. To disable use
64 # "0".
65 #
66 # To enable this option, you must use --enable-htcp with the
67 # configure script.
68 #

69 #Default:
70 # htcp_port 4827
71
72 # TAG: mcast_groups
73 # This tag specifies a list of multicast groups which your server
74 # should join to receive multicasted ICP queries.
75 #
76 # NOTE! Be very careful what you put here! Be sure you
77 # understand the difference between an ICP_query_ and an ICP
78 # _reply_. This option is to be set only if you want to RECEIVE
79 # multicast queries. Do NOT set this option to SEND multicast
80 # ICP (use cache_peer for that). ICP replies are always sent via
81 # unicast, so this option does not affect whether or not you will
82 # receive replies from multicast group members.
83 #
84 # You must be very careful to NOT use a multicast address which
85 # is already in use by another group of caches.
86 #
87 # If you are unsure about multicast, please read the Multicast
88 # chapter in the Squid FAQ (http://www.squid-cache.org/FAQ/).
89 #
90 # Usage: mcast_groups 239.128.16.128 224.0.1.20
91 #
```



```
92 #      By default, Squid doesn't listen on any multicast groups.
93 #
94 #Default:
95 # none
96
97 # TAG: tcp_outgoing_address
98 # TAG: udp_incoming_address
99 # TAG: udp_outgoing_address
100 #      Usage: tcp_incoming_address 10.20.30.40
101 #             udp_outgoing_address fully.qualified.domain.name
102 #
103 #      tcp_outgoing_address    is used for connections made to remote
104 #                               servers and other caches.
105 #      udp_incoming_address    is used for the ICP socket receiving pac
106 #                               kets
107 #                               from other caches.
108 #      udp_outgoing_address    is used for ICP packets sent out to othe
109 #                               r
110 #                               caches.
111 #
112 #      The default behavior is to not bind to any specific address.
113 #
114 #
115 #      A *_incoming_address value of 0.0.0.0 indicates that Squid should
116 #      listen on all available interfaces.
117 #
118 #      If udp_outgoing_address is set to 255.255.255.255 (the default)
119 #      then it will use the same socket as udp_incoming_address. Only
120 #      change this if you want to have ICP queries sent using another
121 #      address than where this Squid listens for ICP queries from other
122 #      caches.
123 #
124 #      NOTE, udp_incoming_address and udp_outgoing_address can not
125 #      have the same value since they both use port 3130.
126 #
127 #      NOTE, tcp_incoming_address has been removed. You can now
128 #      specify IP addresses on the 'http_port' line.
129 #Default:
130 # tcp_outgoing_address 255.255.255.255
131 # udp_incoming_address 0.0.0.0
132 # udp_outgoing_address 255.255.255.255
133 #
134 # OPTIONS WHICH AFFECT THE NEIGHBOR SELECTION ALGORITHM
```



```

134 # -----
-----
135
136 # TAG: cache_peer
137 #   To specify other caches in a hierarchy, use the format:
138 #
139 #       cache_peer hostname type http_port icp_port
140 #
141 #   For example,
142 #
143 #       #           #           #           #
144 #       #           hostname         type       proxy   icp
145 #       #           -----         -----   port    port  options
146 #       cache_peer parent.foo.net     parent    3128   3130  [proxy-onl
147 #       cache_peer sib1.foo.net       sibling    3128   3130  [proxy-onl
148 #       cache_peer sib2.foo.net       sibling    3128   3130  [proxy-onl
149 #
150 #           type: either 'parent', 'sibling', or 'multicast'.
151 #

```

```

152 #   proxy_port: The port number where the cache listens for proxy
153 #               requests.
154 #
155 #   icp_port:   Used for querying neighbor caches about
156 #               objects. To have a non-ICP neighbor
157 #               specify '7' for the ICP port and make sure the
158 #               neighbor machine has the UDP echo port
159 #               enabled in its /etc/inetd.conf file.
160 #
161 #   options:   proxy-only
162 #               weight=n
163 #               ttl=n
164 #               no-query
165 #               default
166 #               round-robin
167 #               multicast-responder
168 #               closest-only
169 #               no-digest
170 #               no-netdb-exchange
171 #               no-delay
172 #               login=user:password
173 #               connect-timeout=nn
174 #               digest-url=url

```



```
1095 # TAG: reference_age
1096 # As a part of normal operation, Squid performs Least Recently
1097 # Used removal of cached objects. The LRU age for removal is
1098 # computed dynamically, based on the amount of disk space in
1099 # use. The dynamic value can be seen in the Cache Manager 'info'
1100 # output.
1101 #
1102 # The 'reference_age' parameter defines the maximum LRU age. For
1103 # example, setting reference_age to '1 week' will cause objects
1104 # to be removed if they have not been accessed for a week or
1105 # more. The default value is one year.
1106 #
1107 # Specify a number here, followed by units of time. For example:
1108 # 1 week
1109 # 3.5 days
1110 # 4 months
1111 # 2.2 hours
1112 #
1113 # NOTE: this parameter is not used when using the enhanced
1114 # replacement policies, GDSH or LFUDA.
1115 #
1116 #Default:
1117 # reference_age 1 year
1118
1119 # TAG: quick_abort_min (KB)
1120 # TAG: quick_abort_max (KB)
1121 # TAG: quick_abort_pct (percent)
1122 # The cache can be configured to continue downloading aborted
1123 # requests. This may be undesirable on slow (e.g. SLIP) links
1124 # and/or very busy caches. Impatient users may tie up file
1125 # descriptors and bandwidth by repeatedly requesting and
1126 # immediately aborting downloads.
1127 #
1128 # When the user aborts a request, Squid will check the
1129 # quick_abort values to the amount of data transfered until
1130 # then.
1131 #
1132 # If the transfer has less than 'quick_abort_min' KB remaining,
1133 # it will finish the retrieval. Setting 'quick_abort_min' to -1
1134 # will disable the quick_abort feature.
1135 #
1136 # If the transfer has more than 'quick_abort_max' KB remaining,
1137 # it will abort the retrieval.
1138 #
1139 # If more than 'quick_abort_pct' of the transfer has completed,
1140 # it will finish the retrieval.
```



```
2748 # none
2749 #
2750 # TAG: ie_refresh onloff
2751 # Microsoft Internet Explorer up until version 5.5 Service
2752 # Pack 1 has an issue with transparent proxies, wherein it
2753 # is impossible to force a refresh. Turning this on provides
2754 # a partial fix to the problem, by causing all IMS-REFRESH
2755 # requests from older IE versions to check the origin server
2756 # for fresh content. This reduces hit ratio by some amount
2757 # (~10% in my experience), but allows users to actually get
2758 # fresh content when they want it. Note that because Squid
2759 # cannot tell if the user is using 5.5 or 5.5SP1, the behavior
2760 # of 5.5 is unchanged from old versions of Squid (i.e. a
2761 # forced refresh is impossible). Newer versions of IE will,
2762 # hopefully, continue to have the new behavior and will be
2763 # handled based on that assumption. This option defaults to
2764 # the old Squid behavior, which is better for hit ratios but
2765 # worse for clients using IE, if they need to be able to
2766 # force fresh content.
2767 #
2768 #Default:
2769 # ie_refresh off
2770 #
```

## 28-2 快速設定 Proxy 代理伺服器

當安裝好 Proxy 代理伺服器後，我們只要修改/etc/squid/squid.conf 組態檔的下列組態，就可以使用我們的 Proxy 代理伺服器了。

我們將第 1457 行的 http\_access deny all 改成 http\_access allow all。這樣就允許大家來使用我們的 Proxy 代理伺服器。

```
1448 # Deny requests to unknown ports
1449 http_access deny !Safe_ports
1450 # Deny CONNECT to other than SSL ports
1451 http_access deny CONNECT !SSL_ports
1452 #
1453 # INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS
1454 #
1455 # And finally deny all other access to this proxy
1456 http_access allow localhost
1457 http_access allow all
```



從 1422 到 1458 行，這是我們設定可以讀取我們 Proxy 代理伺服器的權限。預設是 `http_access deny all` 禁止存取。因此我們將第 1457 行的 `http_access deny all` 改成 `http_access allow all`。這樣就允許大家來使用我們的 Proxy 代理伺服器。

```
1422 # TAG: http_access
1423 #     Allowing or Denying access based on defined access lists
1424 #
1425 #     Access to the HTTP port:
1426 #     http_access allow|deny [!]aclname ...
1427 #
1428 #     NOTE on default values:
1429 #
1430 #     If there are no "access" lines present, the default is to deny
1431 #     the request.
1432 #
1433 #     If none of the "access" lines cause a match, the default is the
1434 #     opposite of the last line in the list.  If the last line was
1435 #     deny, then the default is allow.  Conversely, if the last line
1436 #     is allow, the default will be deny.  For these reasons, it is a
1437 #     good idea to have an "deny all" or "allow all" entry at the end
1438 #     of your access lists to avoid potential confusion.
1439 #
1440 #Default:
1441 # http_access deny all
1442 #
1443 #Recommended minimum configuration:
1444 #
1445 # Only allow cachemgr access from localhost
1446 http_access allow manager localhost
1447 http_access deny manager
1448 # Deny requests to unknown ports
1449 http_access deny !Safe_ports
1450 # Deny CONNECT to other than SSL ports
1451 http_access deny CONNECT !SSL_ports
1452 #
1453 # INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS
1454 #
1455 # And finally deny all other access to this proxy
1456 http_access allow localhost
1457 http_access allow all
1458
```

從 1095 行到 1117 行是設定我們資料在 Proxy 代理伺服器的保存期限。我們 Proxy 代理伺服器是使用 LRU (Least Recently Used)，來將最久沒有使用的資料檔案給移除，我們在第 1117 行設定，當只要資料在四個月內都沒有被使用就會被移除。

```
1117 reference_age 4 months
```



這是從 1095 行到 1117 行是設定我們資料在 Proxy 代理伺服器的資料保存期限

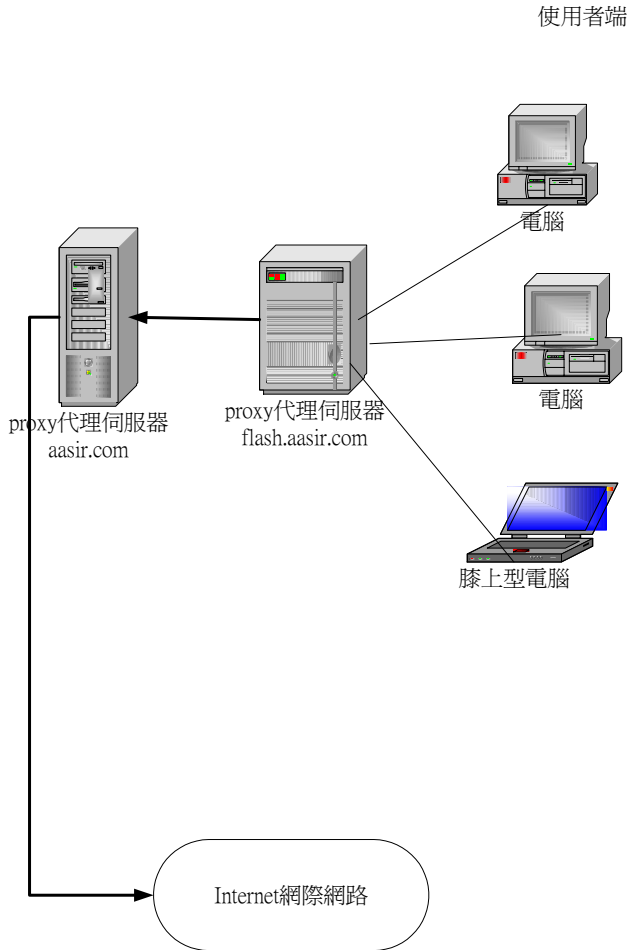
```
1095 # TAG: reference_age
1096 # As a part of normal operation, Squid performs Least Recently
1097 # Used removal of cached objects. The LRU age for removal is
1098 # computed dynamically, based on the amount of disk space in
1099 # use. The dynamic value can be seen in the Cache Manager 'info'
1100 # output.
1101 #
1102 # The 'reference_age' parameter defines the maximum LRU age. For
1103 # example, setting reference_age to '1 week' will cause objects
1104 # to be removed if they have not been accessed for a week or
1105 # more. The default value is one year.
1106 #
1107 # Specify a number here, followed by units of time. For example:
1108 # 1 week
1109 # 3.5 days
1110 # 4 months
1111 # 2.2 hours
1112 #
1113 # NOTE: this parameter is not used when using the enhanced
1114 # replacement policies, GDSH or LFUDA.
1115 #
1116 #Default:
1117 reference_age 4 months
```

我們重新啟動 Proxy 代理伺服器，這樣就可以使用它了。

```
[root@flash squid]# /etc/rc.d/init.d/squid restart
停止 squid:.[ 確定 ]
啓動 squid: [ 確定 ]
```

在實際我們的網站規劃上，我們的 Proxy 代理伺服器，可以分為 2 幾台，而它們是有階層 Hierarchy 的關係。我們的 Proxy 代理伺服器 flash.aasir.com 和 Proxy 代理伺服器 C、Proxy 代理伺服器 D 是同一層的兄弟關係，它們是在同一個網域內。而我們的 Proxy 代理伺服器 aasir.com 它是代理伺服器 flash.aasir.com、C、D 的上一階層 Proxy 代理伺服器，也叫作父節點，當我們在 flash.aasir.com、C、D 代理伺服器都讀不到資料時，再到 Proxy 代理伺服器 aasir.com 去尋找，如果再找不到，再到更上層去，直到找到或直接到指定的遠端網際網路網站上去尋找。





當我們使用瀏覽器時，我們可以設定我們的 Proxy 代理伺服器，這樣當瀏覽器要讀取資料時，會先在近端的 Proxy 代理伺服器去尋找，如果找不到，才會到遠端的網站。這樣讓我們讀取網頁的時間速度加快。我們現在在 IE 來設定 Proxy 代理伺服器。我們的網域是在 flash.aasir.com 上，且在上面有裝設 Proxy 代理伺服器。這樣當我們瀏覽器要讀取資料時，就會到 flash.aasir.com 的 Proxy 代理伺服器去讀取。





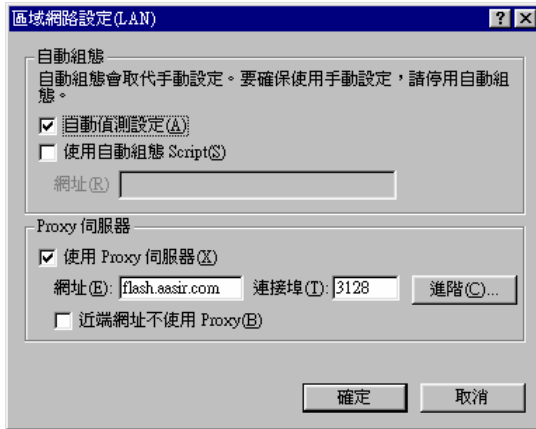
我們選取工具，再選取 Internet 選項。



我們選取連線，再選取區域網路設定。



這是我們將 Proxy 伺服器打勾，並且輸入我們 Proxy 代理伺服器的網址 flash.aasir.com 和輸入連接埠 3128。連接埠 3128 是我們 flash.aasir.com 代理伺服器預設的，設定前請先問網管人員。



我們選取所有通訊協定使用相同的 Proxy。當我們設定完成後，我們就可以使用 Proxy 代理伺服器。當一開始時，因為我們的 Proxy 代理伺服器是空的，裏面沒有資料，所以它會先將資料載入到我們的代理伺服器中，因此一開始需發較長的時間來讀取網頁資料。



### 28-3 Proxy 伺服器的重要組態設定

這是我們 Proxy 伺服器的重要組態設定。

從第 22 行到第 48 行是設定連接埠。我們如果要設定前可以把#給去除。預設的連接埠為 3128。我們在我們的瀏覽器上設定的連接埠就是 Proxy 伺服器上的連接埠。

```
48 # http_port 3128
```

```
22 # TAG: http_port
23 #     Usage: port
24 #           hostname:port
25 #           1.2.3.4:port
26 #
27 #     The socket addresses where Squid will listen for HTTP client
28 #     requests. You may specify multiple socket addresses.
29 #     There are three forms: port alone, hostname with port, and
30 #     IP address with port. If you specify a hostname or IP
31 #     address, then Squid binds the socket to that specific
32 #     address. This replaces the old 'tcp_incoming_address'
33 #     option. Most likely, you do not need to bind to a specific
34 #     address, so you can use the port number alone.
35 #
36 #     The default port number is 3128.
37 #
38 #     If you are running Squid in accelerator mode, then you
39 #     probably want to listen on port 80 also, or instead.
40 #
41 #     The -a command line option will override the *first* port
42 #     number listed here. That option will NOT override an IP
43 #     address, however.
44 #
45 #     You may specify multiple socket addresses on multiple lines.
46 #
47 #Default:
48 # http_port 3128
```

當我們的 flash.aasir.com 代理伺服器中沒有我們需求的資料或網頁，這時 Squid 代理伺服器將向其它兄弟的 Proxy 代理伺服器查詢，或者是向它的父節點的 Proxy 代理伺服器查詢，如果都查詢不到才直接向網際網路上的網站 Request。第 146 行是設定我們的父節點 parent 的網址及連接埠。第 147 行、第 148 行是設定我們的兄弟節點 Proxy 代理伺服器的網址及連接埠。



```

136 # TAG: cache_peer
137 #   To specify other caches in a hierarchy, use the format:
138 #
139 #       cache_peer hostname type http_port icp_port
140 #
141 #   For example,
142 #
143 #   #           hostname           type      proxy  icp
144 #   #           -----           -
145 #   #           parent.foo.net      parent   3128   3130
146 #   #           sib1.foo.net        sibling   3128   3130
147 #   #           sib2.foo.net        sibling   3128   3130
148 #   #
149 #   #           type: either 'parent', 'sibling', or 'multicast'.
150 #

```

我們在第 251 行和第 252 行設定 Proxy 代理伺服器支援的範圍。

我們指定網域可向 Cache 求取支援，或限制或禁止查詢的網域。例如果是 .edu 的網域就會向 parent.foo.net 的 Proxy 代理伺服器作查詢。如果前面加上 !not 符號，表示禁止的意義。

```

247 # TAG: cache_peer_domain
248 #   Use to limit the domains for which a neighbor cache will be
249 #   queried. Usage:
250 #
251 #       cache_peer_domain cache-host domain [domain ...]
252 #       cache_peer_domain cache-host !domain
253 #
254 #   For example, specifying
255 #
256 #       cache_peer_domain parent.foo.net .edu

```

我們在第 405 行是設定我們的快取記憶體。預設是設定 8MB，

```

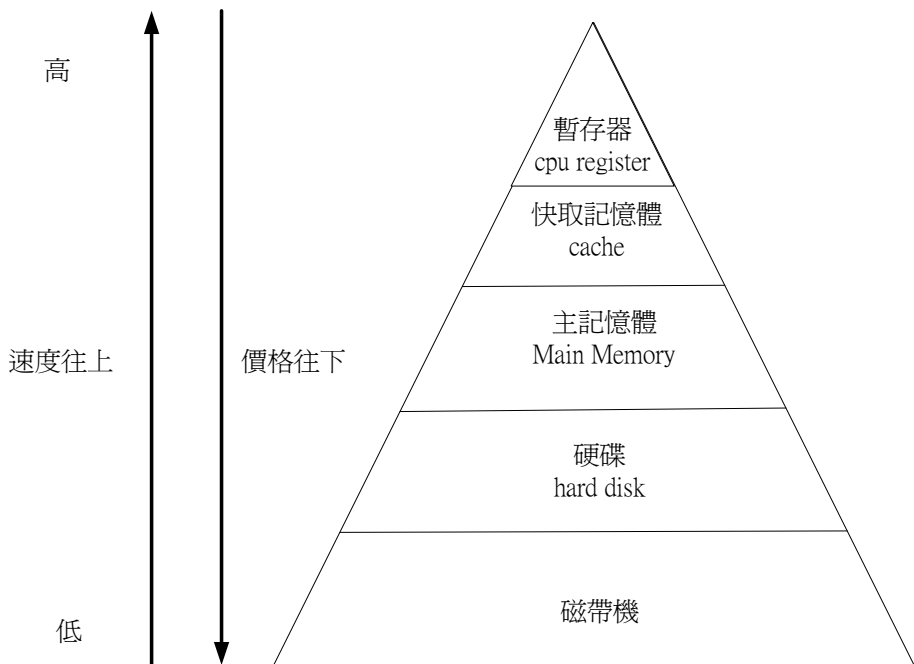
405 #Default:
406 # cache_mem 8 MB

```

我們會將資料放到 CPU 暫存器，再放到快取記憶體中，然後再放到主記憶體，最後是放到硬碟（硬碟、磁帶為非揮發性儲存體，當電源消失時，還是有內部資料儲



存在裏面，不會隨電源的消失而消失，其它的儲存體則為揮發性儲存體，當電源消失時，資料也就消失了。CPU 暫存器的儲存讀取速度最快，但價格也最昂貴。Cache 快取記憶體的速度也是很快，通常為主記憶體速度的數十倍以上，因此有時也會將 cache 內建到 CPU 中，這是可以改善整個 CPU 的效能。現在的 Cache 因為速度之分又有分為 Cache L1 和 Cache L2 以便增加整體 CPU 的效能。而主記憶體就是我們經常使用的記憶體，因此我們的記憶體越大，這樣在記憶體儲存的資料就越多。主記憶體存取的速度快上硬碟存取速度 1000 倍以上。所以我們只要設定記憶體越多，則它的效能和存取的速度就越快，但所花費的費用就越多。



第 423 行和第 424 行是設定虛擬記憶體的大小，一般是 `cache_swap_low` 和 `cache_swap_high` 差距在 5%(數值是以百分比表示)。

```
423 # cache_swap_low 90
424 # cache_swap_high 95
```



第 439 行是設定如果我們要查詢的物件是大過我們設定的檔案大小，則不會存入的硬碟 disk。預設是 4096K。這個設定和我們擊中率(hit ratio)得到我們所要的網頁有關。

```
426 # TAG: maximum_object_size (bytes)
427 # Objects larger than this size will NOT be saved on disk. The
428 # value is specified in kilobytes, and the default is 4MB. If
429 # you wish to get a high BYTES hit ratio, you should probably
430 # increase this (one 32 MB object hit counts for 3200 10KB
431 # hits). If you wish to increase speed more than your want to
432 # save bandwidth you should leave this low.
433 #
434 # NOTE: if using the LFUDA replacement policy you should increase
435 # this value to maximize the byte hit rate improvement of LFUDA!
436 # See replacement_policy below for a discussion of this policy.
437 #
438 #Default:
439 # maximum_object_size 4096 KB
```

第 519 行是設定記憶體取代的策略。預設的是 lru 最近最少使用策略，我們將最近最少使用的頁面放入到硬碟中。除了 lru 策略，還有 heap GDSF 策略、heap LFUDA 策略、heap LRU 策略。

```
478 # lru : Squid's original list based LRU policy
479 # heap GDSF: Greedy-Dual Size Frequency
480 # heap LFUDA: Least Frequently Used with Dynamic Aging
481 # heap LRU : LRU policy implemented using a heap
482 #
483 # TAG: memory_replacement_policy
484 # The memory replacement policy parameter determines which
485 # objects are purged from memory when memory space is needed.
486 #
487 # See cache_replacement_policy for details.
488 #
489 #Default:
490 # memory_replacement_policy lru
```

第 602 行的 `cache_dir ufs /var/spool/squid` 為設定暫存資料的目錄，我們預設的目錄是 `/var/spool/squid`。在這裏參數 100 是設定這個目錄所使用的容量為 100MB，16 表示第一層子目錄為 16 個，256 為第二層子目錄數量。ufs 為 squid 預設的儲存格式，如果我們使用非同步 I/O 的電腦(Linux 或 Solaris)則可以改成 aufs。我們也可以指定多個 `cache_dir` 暫存目錄，來將資料分散到幾個不同的硬碟。



```
601 #Default:
```

```
602 # cache_dir ufs /var/spool/squid 100 16 256
```

這是我們/var/spool/squid 第一層的目錄，總共有 16 個子目錄。

```
[root@flash squid]# ls
```

```
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F swap.state
```

這是我們/var/spool/squid 第二層的目錄 00，總共有 256 個子目錄。這子目錄就是存放我們從遠端下載的網頁檔案資料，來供作代理伺服器的暫存檔。

```
[root@flash 00]# ls
```

```
00 0D 1A 27 34 41 4E 5B 68 75 82 8F 9C A9 B6 C3 D0 DD EA F7
01 0E 1B 28 35 42 4F 5C 69 76 83 90 9D AA B7 C4 D1 DE EB F8
02 0F 1C 29 36 43 50 5D 6A 77 84 91 9E AB B8 C5 D2 DF EC F9
03 10 1D 2A 37 44 51 5E 6B 78 85 92 9F AC B9 C6 D3 E0 ED FA
04 11 1E 2B 38 45 52 5F 6C 79 86 93 A0 AD BA C7 D4 E1 EE FB
05 12 1F 2C 39 46 53 60 6D 7A 87 94 A1 AE BB C8 D5 E2 EF FC
06 13 20 2D 3A 47 54 61 6E 7B 88 95 A2 AF BC C9 D6 E3 F0 FD
07 14 21 2E 3B 48 55 62 6F 7C 89 96 A3 B0 BD CA D7 E4 F1 FE
08 15 22 2F 3C 49 56 63 70 7D 8A 97 A4 B1 BE CB D8 E5 F2 FF
09 16 23 30 3D 4A 57 64 71 7E 8B 98 A5 B2 BF CC D9 E6 F3
0A 17 24 31 3E 4B 58 65 72 7F 8C 99 A6 B3 CO CD DA E7 F4
0B 18 25 32 3F 4C 59 66 73 80 8D 9A A7 B4 C1 CE DB E8 F5
0C 19 26 33 40 4D 5A 67 74 81 8E 9B A8 B5 C2 CF DC E9 F6
```

第 609 行是我們用戶端使用的記錄。它設定快取記錄檔案名稱和所在目錄，並記錄了所有使用者快取的行為記錄。

```
608 #Default:
```

```
609 # cache_access_log /var/log/squid/access.log
```

第 617 行是 cache 緩衝記錄檔。這是我們 cache 緩衝行為的資訊。

```
611 # TAG: cache_log
```

```
612 # Cache logging file. This is where general information about
```

```
613 # your cache's behavior goes. You can increase the amount of data
```

```
614 # logged to this file with the "debug_options" tag below.
```

```
615 #
```

```
616 #Default:
```

```
617 # cache_log /var/log/squid/cache.log
```

第 627 行是記錄我們儲存管理的記錄。顯示什麼物件被拒絕儲存，和什麼物件被儲存多久的時間。如果不要記錄儲存管理的行為只要輸入 none 就好了。

```
626 #Default:
```

```
627 # cache_store_log /var/log/squid/store.log
```



```
1440 #Default:
1441 # http_access deny all
1442 #
1443 #Recommended minimum configuration:
1444 #
1445 # Only allow cachemgr access from localhost
1446 http_access allow manager localhost
1447 http_access deny manager
1448 # Deny requests to unknown ports
1449 http_access deny !Safe_ports
1450 # Deny CONNECT to other than SSL ports
1451 http_access deny CONNECT !SSL_ports
1452 #
1453 # INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS
```

以一般系統的操作, Squid 使用 LRU 移除在 Cache 中最久沒有使用的檔案資料。LRU 隨著時間動態的移除 Cache 中的資料, 是以硬碟的容量為基礎。這 `reference_age` 參考的時間參數, 定義最大的 LRU 時間。例如我們如果設定 1 week, 表示如果資料如果一個禮拜都沒被參考到, 則它就會被移除。預設是 1 year。我們在第 1117 行設定 4 個 months。

```
1116 #Default:
1117 reference_age 4 months
```

```
1095 # TAG: reference_age
1096 # As a part of normal operation, Squid performs Least Recently
1097 # Used removal of cached objects. The LRU age for removal is
1098 # computed dynamically, based on the amount of disk space in
1099 # use. The dynamic value can be seen in the Cache Manager 'info'
1100 # output.
1101 #
1102 # The 'reference_age' parameter defines the maximum LRU age. For
1103 # example, setting reference_age to '1 week' will cause objects
1104 # to be removed if they have not been accessed for a week or
1105 # more. The default value is one year.
1106 #
1107 # Specify a number here, followed by units of time. For example:
1108 # 1 week
1109 # 3.5 days
1110 # 4 months
1111 # 2.2 hours
1112 #
1113 # NOTE: this parameter is not used when using the enhanced
1114 # replacement policies, GDSH or LFUDA.
1115 #
1116 #Default:
1117 reference_age 4 months
```



第 1254 行設定使用者端最長可以連線使用 Proxy 代理伺服器的時間。預設是 1day，也就是 1440 分鐘。

```
1237 # TAG: client_lifetime time-units
1238 # The maximum amount of time that a client (browser) is allowed to
1239 # remain connected to the cache process. This protects the Cache
1240 # from having a lot of sockets (and hence file descriptors) tied u
1241 # p
1242 # in a CLOSE_WAIT state from remote clients that go away without
1243 # properly shutting down (either because of a network failure or
1244 # because of a poor client implementation). The default is one
1245 # day, 1440 minutes.
1246 # NOTE: The default value is intended to be much larger than any
1247 # client would ever need to be connected to your cache. You
1248 # should probably change client_lifetime only as a last resort.
1249 # If you seem to have many client connections tying up
1250 # filedescriptors, we recommend first tuning the read_timeout,
1251 # request_timeout, pconn_timeout and quick_abort values.
1252 #
1253 #Default:
1254 # client_lifetime 1 day
```

第 1300 行是設定收到 timeout 的時間。當 Proxy 代理伺服器關機時，使用者將在指定的時間後收到 timeout 的訊息。

```
1292 # TAG: shutdown_lifetime time-units
1293 # When SIGTERM or SIGHUP is received, the cache is put into
1294 # "shutdown pending" mode until all active sockets are closed.
1295 # This value is the lifetime to set for all open descriptors
1296 # during shutdown mode. Any active clients after this many
1297 # seconds will receive a 'timeout' message.
1298 #
1299 #Default:
1300 # shutdown_lifetime 30 seconds
```

這是設定存取權限(Access Control list,acl)。

第 1320 行是設定從 src(來源端 source)端存取 Proxy 代理伺服器的連線 IP。

第 1321 行是設定從 src(來源端 source)端存取 Proxy 代理伺服器的連線名稱。

第 1322 行是設定從 dst(目的地端 destination)端存取 Proxy 代理伺服器的連線名稱。(指網站主機的 IP 位址)。

aclname 就是定義的名稱，在設定權限時，我們就是以這個名稱來設定。acltype 就是控制形式(access control type)。在這裏是設定指定主機、網路和連接埠連接使用我們 Proxy 代理伺服器的權限。我們先設定存取控制的類別名單(access control



list,acl) , 然後我們再設定每一個 acl 的權限。

```

1306 # TAG: acl
1307 # Defining an Access List
1308 #
1309 # acl aclname acltype stringl ...
1310 # acl aclname acltype "file" ...
1311 #
1312 # when using "file", the file should contain one item per line
1313 #
1314 # acltype is one of src dst srcdomain dstdomain url_pattern
1315 # urlpath_pattern time port proto method browser user
1316 #
1317 # By default, regular expressions are CASE-SENSITIVE. To make
1318 # them case-insensitive, use the -i option.
1319 #
1320 # acl aclname src ip-address/netmask ... (clients IP address)
1321 # acl aclname src addr1-addr2/netmask ... (range of addresses)
1322 # acl aclname dst ip-address/netmask ... (URL host's IP address)
ss)

```

這是我們定義 acl(設定存取權限)的範例。

```

1400 #Examples:
1401 #acl myexample dst_as 1241
1402 #acl password proxy_auth REQUIRED
1403 #acl fileupload req_mime_type -i ^multipart/form-data$
1404 #
1405 #Recommended minimum configuration:
1406 acl all src 0.0.0.0/0.0.0.0
1407 acl manager proto cache_object
1408 acl localhost src 127.0.0.1/255.255.255.255
1409 acl SSL_ports port 443 563
1410 acl Safe_ports port 80 # http
1411 acl Safe_ports port 21 # ftp
1412 acl Safe_ports port 443 563 # https, snews
1413 acl Safe_ports port 70 # gopher
1414 acl Safe_ports port 210 # wais
1415 acl Safe_ports port 1025-65535 # unregistered ports
1416 acl Safe_ports port 280 # http-mgmt
1417 acl Safe_ports port 488 # gss-http
1418 acl Safe_ports port 591 # filemaker
1419 acl Safe_ports port 777 # multiling http
1420 acl CONNECT method CONNECT

```

我們也可以定義從網域 61.218.29.0 為 Proxy 代理伺服器的連線稱為 flash。這是我們自己設定的 acl 類別。在這裏因為遮罩為 255.255.255.248 , 所以總共有 61.218.29.0 到 61.218.29.7 的八個 IP 位址可使用 Proxy 代理伺服器。

```
1421 acl flash src 61.218.29.0/255.255.255.248
```



這是我們 ACL 的定義。

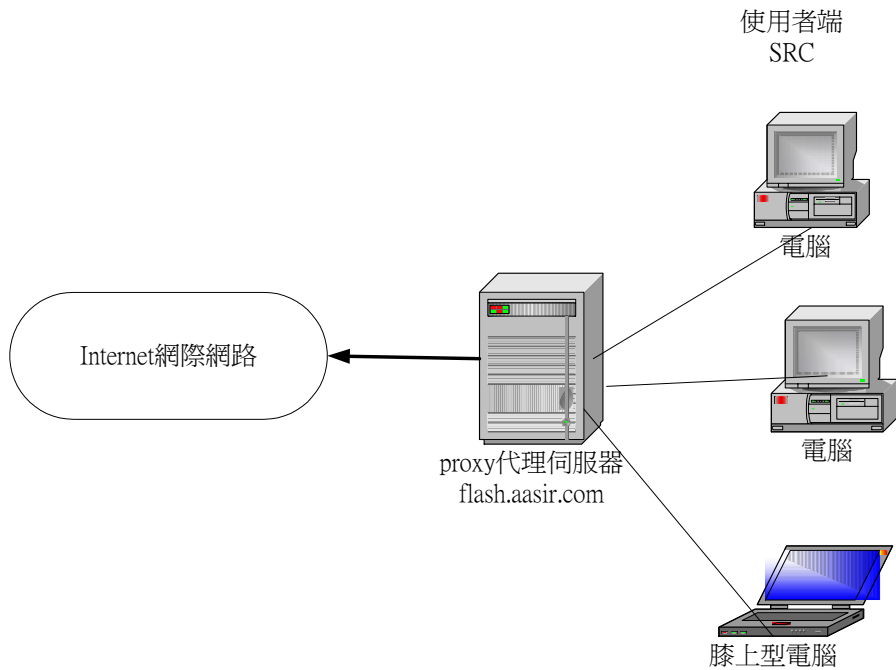
用法：

acl acl 名稱 acl 類型 acl 串列

這是我們 src 的限制類形。我們使用此來對來源的使用者端的 IP 網路來進行存取控管，主要是用來設定可使用 Proxy Server 的來源範圍。

用法：

acl acl 名稱 src 網路 IP / 網路遮罩



從第 1422 行是開始設定 http 存取的權限。預設是 http 禁止存取，http\_access deny all。如果我們將 http\_access deny all 改成 http\_access allow all 則允許任何網域的存取。

```
1422 # TAG: http_access
1423 #     Allowing or Denying access based on defined access lists
1424 #
1425 #     Access to the HTTP port:
1426 #     http_access allow|deny [!]aclname ...
1427 #
1428 #     NOTE on default values:
1429 #
1430 #     If there are no "access" lines present, the default is to deny
1431 #     the request.
1432 #
1433 #     If none of the "access" lines cause a match, the default is the
1434 #     opposite of the last line in the list.  If the last line was
1435 #     deny, then the default is allow.  Conversely, if the last line
1436 #     is allow, the default will be deny.  For these reasons, it is a
1437 #     good idea to have an "deny all" or "allow all" entry at the end
1438 #     of your access lists to avoid potential confusion.
1439 #
1440 #Default:
1441 # http_access deny all

1453 # INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS
1454 #
1455 # And finally deny all other access to this proxy
1456 http_access allow localhost
1457 http_access allow all
```

我們也在第 1458 行可以設定 http\_access allow flash 來允許 flash 類別的存取。也就是由 61.218.29.0/61.218.29.7 網域來的需求皆可存取 我們可以隨自己的需求來設定。

## 1458 http\_access allow flash

第 1539 行是設定當緩衝區出問題時，要將通知 Proxy 代理伺服器的管理者。預設是 Webmaster 網站管理員，也就是 root。

```
1536 # TAG: cache_mgr
1537 #     Email-address of local cache manager who will receive
1538 #     mail if the cache dies.  The default is "webmaster."
1539 #cache_mgr root
```

從第 1639 行開始是設定 httpd 的加速選項。



```
1639 # HTTPD-ACCELERATOR OPTIONS
1640 # -----
1641
1642 # TAG: httpd_accel_host
1643 # TAG: httpd_accel_port
1644 # If you want to run Squid as an httpd accelerator, define the
1645 # host name and port number where the real HTTP server is.
1646 #
1647 # If you want virtual host support then specify the hostname
1648 # as "virtual".
1649 #
1650 # If you want virtual port support then specify the port as "0".
1651 #
1652 # NOTE: enabling httpd_accel_host disables proxy-caching and
1653 # ICP. If you want these features enabled also, then set
1654 # the 'httpd_accel_with_proxy' option.
1655 #
1656 #Default:
1657 # httpd_accel_port 80
```

從第 1704 行開始是雜項設定。第 1713 行是設定 DNS 測試的第一個網站。

```
1704 # MISCELLANEOUS
1705 # -----
1706
1707 # TAG: dns_testnames
1708 # The DNS tests exit as soon as the first site is successfully loo
1709 # ked up
1710 # This test can be disabled with the -D command line option.
1711 #
1712 #Default:
1713 # dns_testnames netscape.com internic.net nlanr.net microsoft.com
```

## 補充一：cache 的取代策略

當 cache 失誤 miss 時，要由下層的主記憶體中取得所要資料區塊，並將它與 cache 中的某一區塊作替換，以符合 CPU 的抓取及使用。常用的替換策略有三種隨機、LRU 和 FIFO。

隨機方式是以隨機方式來選擇 chace 中要被取代的資料區塊。

LRU 是最近較少被使用，選擇那些最近較少被使用的區塊來作取代，這種方法符合時間的區域性(temporal locality)。

先進先出(FIFO)：選擇最早進入 Cache 的資料區塊作更換。



一般寫回的策略又分為兩種，一種是直接寫回，另外一種是間接寫回。

直接寫回是當 cache 要寫入時，除了將資料區塊寫入 cache，同時也將此資料區塊寫入 cache 所對應的下層記憶體的相對位址中。

間接寫回是當 cache 要寫入時，資料區塊只有寫入 cache 中，並未連其下層的記憶體也一起寫入，等到此區塊在 cache 中已被寫入過的資料區塊要被替換出去時，這些已經被修改過的資料區塊才一起寫入到其對應的下層記憶體位置中，用來標記這些已寫入過的區塊位元，我們稱為 dirty bit。當其 dirty bit=1，我們就知到資料區塊已被修改內容。當其 dirty bit =0 時，我們知到此資料區塊還未被修改寫入過。

## 補充二：虛擬記憶體

虛擬記憶體是記憶體階層中很重要的一項技術。

使用虛擬記憶體可讓多個程式之間共同分享記憶體。

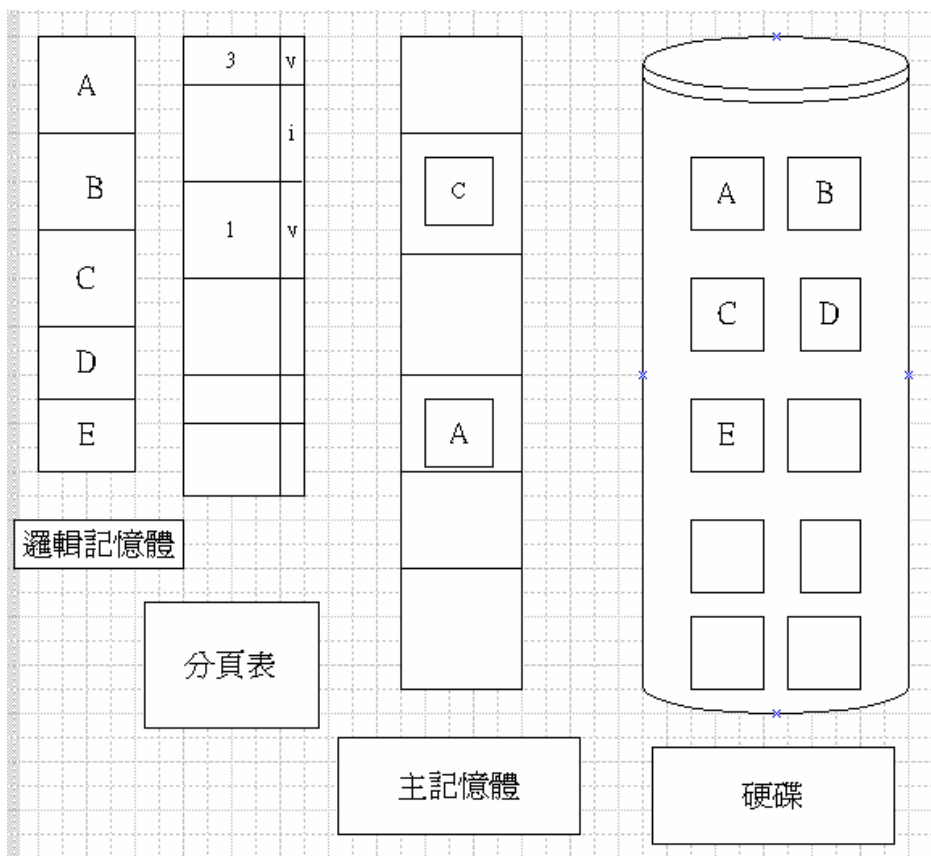
使用虛擬記憶體可讓使用者使用超過主記憶體容量大小的程式，而使用者感覺不出來，這就是虛擬記憶體的優點。由於原來的記憶體容量限制，所以我們可以將目前不用的資料暫時存放在次級記憶體(Secondary storage)，當我們要使用時再將資料從次級記憶體(一般指硬碟)將資料搬入主記憶體。

使用虛擬記憶體可簡化執行的載入程式，也就是重置(relocation)，這樣，我們的程式就可透過虛擬記憶體的對映，在實體記憶體不同的地方執行程式。

當我們要加快虛擬記憶體與主記憶體的位址對應過程，我們可以使用 TLB(translation lookaside buffer)快取來加強虛擬記憶體的效能。

這是我們虛擬記憶體的觀念，在我們作業系統中，邏輯記憶體就是我們程式或使用的記憶體，而我們使用分頁表將記憶體分成一頁一頁的來和主記憶體(實體記憶體)來作對應，因為當我們的資料過多，就可以將部份的頁面放到硬碟，當要使用時再插斷 tarp 通知作業系統，在分頁表圖中，V 代表 VALID 表示分頁有在主記憶體中，i 代表 INVALID 表示分頁不在主記憶體中而在硬碟中，需要從硬碟載入。(v 為 dirty bit=1，I 為 dirty bit =0))





### 補充三：虛擬記憶體的取代策略

當我們要讀取主記憶體資料時，發生資料頁面不在主記憶體時，也就是虛擬記憶體的 miss 失誤(page fault)，當 page fault 發生時，我們要想辦法將資料頁來作替換，讓我們主記憶體具有 CPU 所需要的資料，我們常用的策略有 LRU、FIFO、OPT。

LRU 是最近較少被使用，選擇那些最近較少被使用的區塊來作取代，這種方法符合時間的區域性(temporal locality)。

先進先出(FIFO)：選擇最早進入主記憶體的資料區塊作更換。

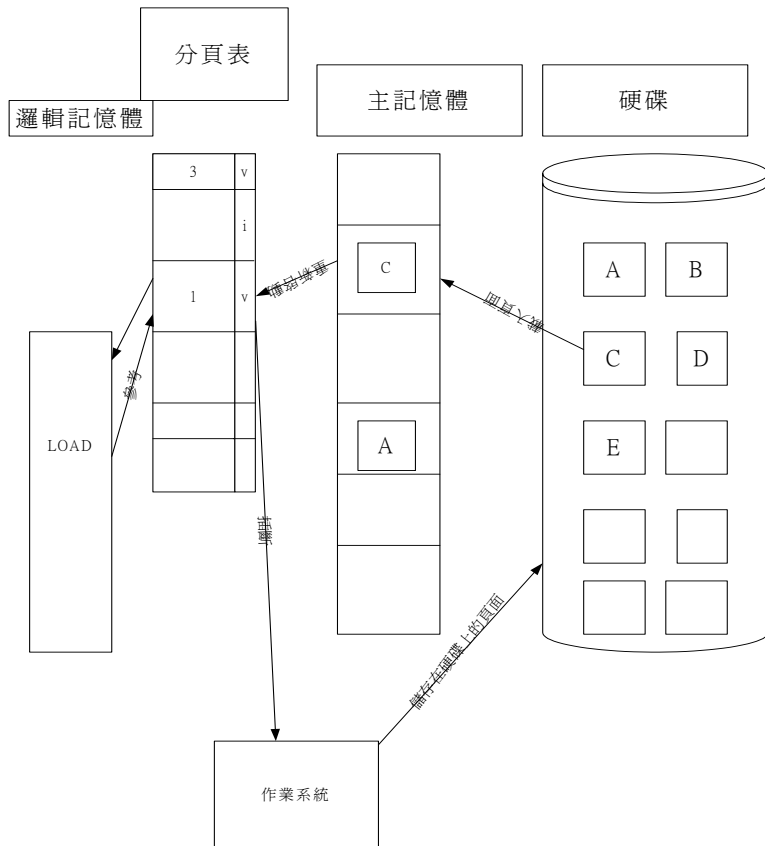
OPT 是最佳化取代策略：我們可以選擇在以後的執行中最少被使用到的資料區

塊被替換出去。OPT 是最好的取代策略，但要知到下一個被取代的頁面要很嚴謹的演算法。

虛擬記憶體的寫回策略為 Write back 間接寫回。

#### .補充四：需求分頁的概念

當我們要載入 D 這個分頁時，因為他不在主記憶體中，因此發生了分頁錯誤的現象(page fault)，因此由作業系統發出 Trap(插斷)去從硬碟載入頁面 D 到主記憶體，再將 D 分頁表的 I 改成 v 來重新設定分頁表，再重新啟動載入。



## .補充五：LRU 的取代策略

假設我們的主記憶體有 3 個資料頁面的記憶體容量。時間 1 和時間 2 的 miss 為強迫性失誤,因為此時一開始,主記憶體沒有資料,所以會將資料存到記憶體中,而產生強迫性失誤,其它的 miss 為 page fault(找不到頁面)。miss 為沒有擊中頁面, Hit 為擊中頁面。

\*代表下次要被取代的頁面 page.在這裏 LRU 總共有兩次 HIT,擊中頁面,因此其 Hit ratio 為  $2/8=0.25$

LRU最近未使用取代策略

經過時間	1	2	3	4	5	6	7	8
資料頁面	2	3	2	1	4	3	1	2
主記憶體	2*	2*	2	2	2*	3	3	3*
		3	3*	3*	4	4	4*	2
				1	1	1*	1	1
	miss	miss	hit	miss	miss	miss	hit	miss

時間1和時間2的miss為強迫性失誤,因為此時一開始,主記憶體沒有資料,所以會將資料存到記憶體中,而產生強迫性失誤,其它的miss為page fault(找不到頁面)。

\*代表下次要被取代的頁面page.在這裏LRU總共有兩次HIT,擊中頁面,因此其Hit ratio為 $2/8=0.25$



## .補充六：FIFO 的取代策略

假設我們的主記憶體有 3 個資料頁面的記憶體容量。時間 1 和時間 2 的 miss 為強迫性失誤,因為此時一開始,主記憶體沒有資料,所以會將資料存到記憶體中,而產生強迫性失誤,其它的 miss 為 page fault(找不到頁面).

\*代表下次要被取代的頁面 page。在這 FIFO 總共有三次 HIT(擊中頁面), 因此其 Hit ratio 為  $3/8=0.375$

FIFO先進先出取代策略

經過時間	1	2	3	4	5	6	7	8
資料頁面	2	3	2	1	4	3	1	2
主記憶體	2*	2* 3	2* 3	2* 3 1	4 3* 1	4 3* 1	4 3* 1	4 2 1*
	miss	miss	hit	miss	miss	hit	hit	miss

時間1和時間2的miss為強迫性失誤,因為此時一開始,主記憶體沒有資料,所以會將資料存到記憶體中,而產生強迫性失誤,其它的miss為page fault(找不到頁面).

\*代表下次要被取代的頁面page.在這FIFO總共有三次HIT,擊中頁面,因此其Hit ratio為 $3/8=0.375$



## .補充：OPT 的取代策略

假設我們的主記憶體有 3 個資料頁面的記憶體容量。

時間 1 和時間 2 的 miss 為強迫性失誤,因為此時一開始,主記憶體沒有資料,所以會將資料存到記憶體中,而產生強迫性失誤,其它的 miss 為 page fault(找不到頁面).

\*代表下次要被取代的頁面 page.在這 opt 總共有三次 HIT,擊中頁面,因此其 Hit ratio 為  $3/8=0.375$

OPT最佳取代策略

經過時間	1	2	3	4	5	6	7	8
資料頁面	2	3	2	1	4	3	1	2
主記憶體	2*	2*	2*	2*	4*	4	4	4
		3	3	3	3	3*	3*	2
				1	1	1	1	1*
	miss	miss	hit	miss	miss	hit	hit	miss

時間1和時間2的miss為強迫性失誤,因為此時一開始,主記憶體沒有資料,所以會將資料存到記憶體中,而產生強迫性失誤,其它的miss為page fault(找不到頁面).

\*代表下次要被取代的頁面page.在這opt總共有三次HIT,擊中頁面,因此其Hit ratio為 $3/8=0.375$



當我們主記憶體的容量越大時，擊中的比率就越高，一般而言 OPT 的擊中比率 (CPU 可以直接從主記憶體取出資料) 大於 LRU，而 LRU 的速度又大於 FIFO。當主記憶體的容量越大我們 CPU 取到資料頁面的 Hit ratio 就越高。

補充快取記憶體和虛擬記憶體的設計比較：

	快取記憶體	虛擬記憶體
映射	快取記憶體和主記憶體的映設方法為直接映射、關聯映射 和集合關聯映射。	關聯映射
搜尋	使用 index 搜尋或是完全搜尋	使用 Page table 分頁表或 segment 分段表
取代策略	隨機、LRU、先進先出	LRU、先進先出、OPT
寫回策略	直接寫回、間接寫回	間接寫回
資料儲存時間	Cache miss 由硬體控制替換過程，當電源消失時，其 cache 所儲存的資料就會消失，所以為暫時儲存。	虛擬記憶體替換由作業系統處理。虛擬記憶體的主要儲存為硬碟，因此可以永久儲存。

在我們的虛擬記憶體映射方面，虛擬記憶體的定址方式為分頁記憶體、分段記憶體和分頁分段記憶體。



## 課後練習

---

1. Proxy 伺服器所能暫存的資料大都是 http 的網頁、ftp 的檔案、gopher 的資料，而最經常使用的 Proxy 代理伺服器以 Unix 系統下的 Squid 最多人使用，Squid 能夠支援的傳輸協定有 http、ssl、ftp、gopher、pop3、SMTP(郵件傳輸協定)、影音傳輸協定。下列何者網路協定不可使用 Squid 伺服器?複選

- (A). News
- (B). HTTP
- (C). DNS
- (D). FTP

---

2. 我們的 Proxy 代理伺服器，可以分為好幾台，而它們是有下列何種的關係? 我們的 Proxy 代理伺服器 B 和 Proxy 代理伺服器 C、Proxy 代理伺服器 D 是同一層的兄弟關係，它們可以在同一個網域內。而我們的 Proxy 代理伺服器 A 它是代理伺服器 B、C、D 的上一階層 Proxy 代理伺服器，也叫作父節點，當我們在 B、C、D 代理伺服器都讀不到資料時，再到 Proxy 代理伺服器 A 去尋找，如果再找不到，再到更上層去，直到找到或直接到指定的遠端網際網路網站上去尋找。

- (A). 上下
- (B). 左右
- (C). 階層 Hierarchy
- (D). 對稱

---

3. 假如我們有一系列的 Squid Proxy 代理伺服器連接在一起，有一些是兄弟關係，有一些是父子關係。假如系統在我們所有的代理伺服器都找不到所要的網頁，它會到下列哪裏去尋找?

- (A). 這原始的代理伺服器會到外面網路搜尋該網頁
- (B). 這 sibling 的代理伺服器會到外面網路搜尋該網頁



- (C). 這 parent 的代理伺服器會到外面網路搜尋該網頁
- (D). 我們的瀏覽器會顯示無法找到該網頁

---

4. 在大大公司，每天一到中午網路就會塞車。剛好老版在這時後要讀取和寄送信件，而他要我們立刻解決此問題，請問我們該怎麼作？

- (A). Upgrade our network
- (B). Route all Web surfing through a Squid server
- (C). Route the boss's mail over a different subnet
- (D). Reconfigure our DNS servers to increase their local cache

---

5. 我們會將資料放到 CPU 暫存器，再放到快取記憶體中，然後再放到主記憶體，最後是放到硬碟（硬碟、磁帶為非揮發性儲存體，當電源消失時，還是有內部資料儲存在裏面，不會隨電源的消失而消失，其它的儲存體則為揮發性儲存體，當電源消失時，資料也就消失了）。CPU 暫存器的儲存讀取速度最快，但價格也最昂貴。Cache 快取記憶體的速度也是很快，通常為主記憶體速度的數十倍以上，因此有時也會將 cache 內建到 CPU 中，這是可以改善整個 CPU 的效能。現在的 Cache 因為速度之分又有分為 Cache L1 和 Cache L2 以便增加整體 CPU 的效能。而主記憶體就是我們經常使用的記憶體，因此我們的記憶體越大，這樣在記憶體儲存的資料就越多。主記憶體存取的速度快上硬碟存取速度 1000 倍以上。所以我們只要設定記憶體越多，則它的效能和存取的速度就越快，但所花費的費用就會如何？

- (A). 越少
- (B). 越多
- (C). 不變
- (D). 免費

---

6. 當我們主記憶體的容量越大時，擊中的比率就會如何？一般而言 OPT 的擊中比率（CPU 可以直接從主記憶體取出資料）大於 LRU，而 LRU 的速度又大於 FIFO。

- (A). 越低



- (B). 越高
- (C). 不變
- (D). 無

---

7. 使用下列何種技術可讓多個程式之間共同分享記憶體? 使用虛擬記憶體可讓使用者使用超過主記憶體容量大小的程式, 而使用者感覺不出來, 這就是虛擬記憶體的優點。由於原來的記憶體容量限制, 所以我們可以將目前不用的資料暫時存放在次級記憶體(Secondary storage), 當我們要使用時再將資料從次級記憶體(一般指硬碟)將資料搬入主記憶體。

- (A). SWAP
- (B). LRU
- (C). MBR
- (D). 虛擬記憶體

---

**【答案】**

1.A,C 2.C 3.C 4.B 5.B 6.B 7.D

