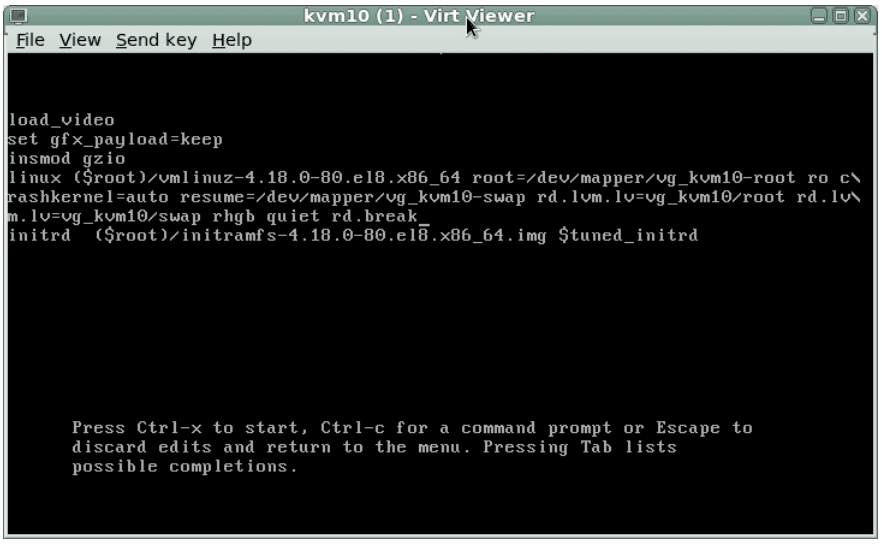
Linux 系統管理 (RHCSA 9)

Department of Computer Science and Information Engineering
Chaoyang University of Technology
Taichung, Taiwan, Republic of China

Instructor: De-Yu Wang (王德譽)
E-mail: dywang@mail.cyut.edu.tw
Phone: (04)23323000 ext 4538
Office: E738



Friday 17th January, 2025

- Instructor: De-Yu Wang
    1. Email: dywang@csie.cyut.edu.tw
    2. Homepage: http://dywang.csie.cyut.edu.tw
    3. 本文件 pdf 檔下載
    4. Phone: (04)23323000 ext 4538
    5. Office: E738
- 參考資料
    1. RedHat Documentation
    2. RedHat Customer Protal
    3. Deploying VDO
    4. Getting started with Tuned

Contents

I 練習環境與基本操作 **1**

1 實機練習系統 **2**

1.1 動機 2
1.2 系統設計 2
1.3 練習系統製作-VirtualBox 3
1.4 練習系統製作-Raw Image 6
1.5 NAT 設計 7
1.6 虛擬機管理界面 8
1.7 虛擬機管理界面 V2 9
1.8 系統管理與評分 10
1.9 系統註冊–上課前先確定三件事 12
1.10 虛擬機設定 13
1.11 系統伺服器管理 14

2 命令列基本操作 **17**

2.1 Linux 指令 17
2.2 線上求助 18
2.3 man page 21
2.4 基礎指令操作 24
2.5 重要的幾個熱鍵與錯誤訊息的察看 37

3 vi 編輯器 **38**

3.1 vi 與 vim 38
3.2 vi 的使用 38
3.3 vim 的額外功能 41
3.4 vi 實機練習題 43

II 系統管理 KVM8 **46**

4 IPv4 網路設定 **47**
4.1 前言 47
4.2 nmcli 命令 47
4.3 修改網卡設定 48
4.4 * 刪除重新設定 49
4.5 SSH 設定與連線 50
4.6 主機名稱設定 51
4.7 實機練習 52

5 YUM 套件管理 **54**
5.1 前言 54
5.2 yum 套件庫設定 55
5.3 *yum 查詢功能 57
5.4 *yum 安裝、升級與移除 59
5.5 實機練習 61

6 systemctl 系統服務控制 **62**
6.1 前言 62
6.2 systemctl 命令 62
6.3 服務啓動 65
6.4 開機自動啓動服務 68
6.5 * 永久關閉服務 69
6.6 * 失效的服務 70
6.7 實機練習 70

7 SELinux **72**
7.1 前言 72
7.2 SELinux Contexts 73
7.3 啓動、關閉與觀察 74
7.4 SELinux 基本用法 75
7.5 Httpd Port 78
7.6 Httpd SELinux Context 80
7.7 實機練習 82

8 * 正規表示法 **83**
8.1 前言 83
8.2 基礎正規表示法 83

8.3 延伸正規表示法 90
8.4 實機練習題 91

9 資料導向與管線處理 93
9.1 爲何要使用命令輸出重導向？ 93
9.2 資料流重導向 93
9.3 管線命令 (pipe) 96
9.4 實機練習 96

10 群組管理 97
10.1 簡介 97
10.2 增加群組 98
10.3 刪除群組 98
10.4 群組更名 98
10.5 實機練習 99

11 帳號與身份管理 100
11.1 簡介 100
11.2 新增帳號 101
11.3 刪除帳號 102
11.4 變更帳號 105
11.5 設定使用者密碼 106
11.6 帳號家目錄問題 107
11.7 帳號家目錄問題 107
11.8 帳號整體參數變更 109
11.9 實機練習 111

12 su 與 sudo 112
12.1 su 切換使用者 112
12.2 sudo 以其他身份執行命令 113
12.3 sudo 用户測試 114
12.4 sudo 群組測試 114
12.5 實機練習 116

13 帳號密碼策略 117
13.1 /etc/shadow 117
13.2 預設密碼過期 118
13.3 變更預設密碼過期策略 119
13.4 設定已存在帳號密碼過期狀態 120

13.5 實機練習 121

14 Linux 檔案權限與屬性 **122**
14.1 檔案權限 122
14.2 改變檔案權限 123
14.3 檔案的預設權限 124
14.4 檔案的特殊屬性 126
14.5 特殊屬性 SGID 練習 128
14.6 特殊屬性 SGID 測試 129
14.7 實機練習 130

15 Access Control Lists **132**
15.1 簡介 132
15.2 ACL 練習 133
15.3 ACL 變更 134
15.4 實機練習 136

16 例行性命令 **137**
16.1 Linux 的例行性命令 137
16.2 僅執行一次的工作排程：at 137
16.3 循環執行的工作排程 139
16.4 實機練習 141

17 Chronyd vs. ntpd 校時 **142**
17.1 前言 142
17.2 chronyd 使用 142
17.3 立即校時 144
17.4 *makestep 校時 145
17.5 實機練習 146

18 Network File System, NFS **147**
18.1 NFS 簡介 147
18.2 NFS Client 端設定 147
18.3 autofs 自動掛載 149
18.4 autofs 自動掛載測試一 150
18.5 autofs 自動掛載測試二 152
18.6 *NFS 掛載非常慢 152
18.7 實機練習 153

19 檔案搜尋 **154**
19.1 find 命令 154
19.2 找屬於某用户的檔案 155
19.3 找特定大小的檔案 156
19.4 找特定屬性的檔案 157
19.5 實機練習 160

20 Shell Script **161**
20.1 前言 161
20.2 執行 shell script 162
20.3 可執行檔 163
20.4 * 可執行檔別名 165
20.5 實機練習 166

21 檔案打包壓縮 **168**
21.1 檔案的壓縮 168
21.2 檔案的壓縮與打包 169
21.3 實機練習 170

22 Podman 簡介 **172**
22.1 前言 172
22.2 Podman 安裝 172
22.3 Podman 命令 172
22.4 Registry 設定 174

23 *Podman Registry 架設 **176**
23.1 建置 Registry 176
23.2 啓用 Registry 177
23.3 上傳 Image 到 Registry 178
23.4 * 修改 Image 181
23.5 *commit 建立 Image 183
23.6 登入 Registry 183

24 Images 下載建立 **184**
24.1 登入搜尋 184
24.2 下載 Images 185
24.3 build 建立 Image 186
24.4 實機練習 187

25 開機啓用 Podman 容器 **188**
25.1 *Systemd 啓動容器 188
25.2 * 刪除容器 191
25.3 容器掛載儲存空間 191
25.4 開機啓動容器 192
25.5 空間掛載測試 194
25.6 開機啓動容器檢查 195
25.7 *podman 版本問題 196
25.8 實機練習 199

III 儲存空間管理 KVM10 **201**

26 開機 **202**
26.1 前言 202
26.2 開機 targets 203
26.3 變更密碼 203
26.4 實機練習 206

27 YUM 套件管理 **207**
27.1 前言 207
27.2 yum 套件庫設定 208
27.3 *yum 查詢功能 210
27.4 *yum 安裝、升級與移除 212
27.5 實機練習 214

28 邏輯卷冊放大縮小 **215**
28.1 簡介 215
28.2 縮小 LV 標準作法 216
28.3 放大 LV 標準作法 217
28.4 單一指令放大縮小 LV 218
28.5 實機練習 220

29 邏輯卷冊管理 **221**
29.1 前言 221
29.2 新增分割區 221
29.3 產生 PV 224
29.4 產生 VG 224
29.5 產生 LV 225

29.6 格式化 227
29.7 開機掛載 FS 227
29.8 開機啓動 Swap 229
29.9 *SWAP LV 放大縮小 229
29.10 實機練習 231

30 邏輯卷冊 – 可獨立修改項目 **232**
30.1 修改 LV 名稱 232
30.2 修改 VG 名稱 233
30.3 修改檔案系統格式 234
30.4 修改掛載目錄 235
30.5 VG 刪除重建 237
30.6 sulogin 解決 237

31 邏輯卷冊 – PE Size 調整 **239**
31.1 前言 239
31.2 改變 PE 大小 – 必須改變 PV 大小 240
31.3 改變 PE 大小 – LV 的 PE 數量 242
31.4 改變 PE 大小 – 必須調整 LV 243
31.5 改變 PE 大小 – 必須調整 SWAP 245

32 Tuned 性能調整 **247**
32.1 前言 247
32.2 Tuned 配置 247
32.3 啓動 Tuned 248
32.4 tuned-adm 248
32.5 實機練習 250

Part I

練習環境與基本操作

Chapter 1

實機練習系統

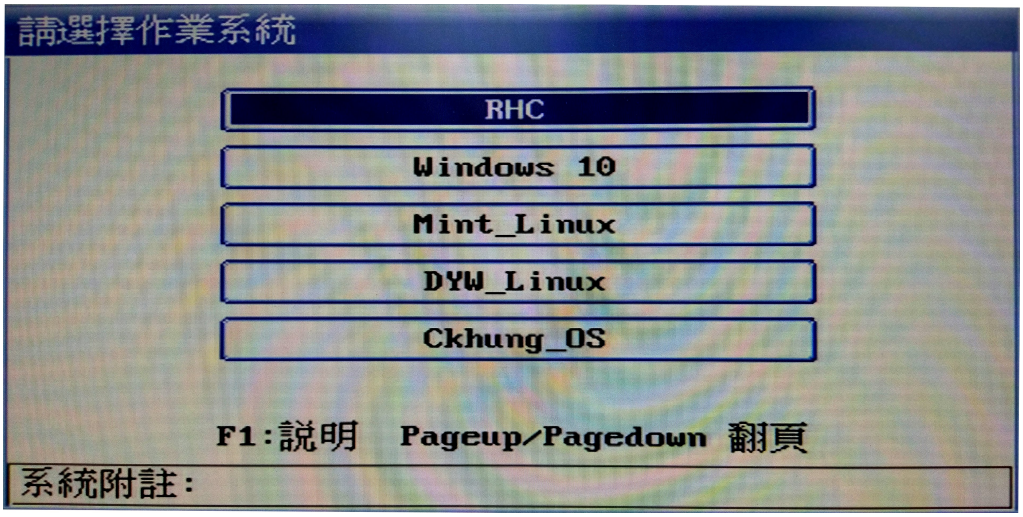
1.1 動機

1. 教學若有實際環境，讓學生實機操作練習，可以大大提昇學習成效。
2. Linux 實機練習，很容易因學生操作錯誤，而造成系統損壞。例如：硬碟分割、格式化及管理的練習。
3. 實機練習結束或操作不當造成當機，系統必須可以迅速還原，才能重複練習。
4. 系統必須可以批次快速修改或更新。
5. 系統必須可以依據學生實際操作狀況，批次進行檢查或評分。
6. 虛擬機練習所需伺服器資源，例如：LDAP 網路帳號、YUM repository 套件庫伺服器、NFS 分享、NTP 校時伺服器、ISCSI 硬碟分享、內部網域 DNS 伺服器、FTP 資源下載... 等環境必須架設好，才能提供虛擬機做各種狀況的練習。

1.2 系統設計

1. Linux 教學系統設計
   (a) 客製化 Linux 系統。
   (b) Linux 伺服器架設。
   (c) VNC 遠端桌面與 KVM 虛擬機。
2. Linux 教學系統安裝

(a) E517 電腦教室：方便學生上課使用。



(b) 個人隨身碟：插上隨身碟，以安裝光碟開機，不需要任何設定，自動安裝於隨身碟，不會影響電腦中的硬碟。

(c) LiveCD：以 LiveCD 光碟開機，開機後即可使用，只是所有操作與設定都是暫存，重新開機後即回到原始狀態。

1.3 練習系統製作-VirtualBox

1. 下載 DYW Linux 練習系統 vdi 檔，下載網址公告於本平台課程內，請登入查看。

```
[root@dywmsi ~]# unxz crt9.vdi.xz
```

2. 依自己使用的 OS (Mac Windows Linux) 下載安裝 VirtualBox。

```
https://www.virtualbox.org/wiki/Downloads
```

3. DYW Linux 練習系統，Host 主機以 virsh 管理 VM，最低硬體因所需 VM 個數不同：

(a) CPU 4 核心以上，若修的課不用內部虛擬機，設定 2 核心也可以。

(b) RAM 8G 以上，若修的課不用內部虛擬機，設定 4G RAM 也可以。

(c) RHCSA, RHCE 請設定 4 核心，8G RAM 以上

(d) CPU 支援虛擬化技術 (virtualization technology)，BIOS 啓用 VT。

4. VirtualBox 新增 VM

```
New(Ctrl+N)
    Name: 自訂，例如 crt9
    Type: Linux
    Version: 只要是 64-bit 就可以，例如 Fedora (64-bit)
    Next >
Memory size
    8192 MB
    Next >
Hard disk
    Use an existing virtual hard disk file
    新增解壓縮的 crt9.vdi，並選用這個硬碟檔。
    Create
```

5. 若要使用內部虛擬機，必須保證巢式虛擬化有開啓，假設匯入的 VM 名稱爲 crt9

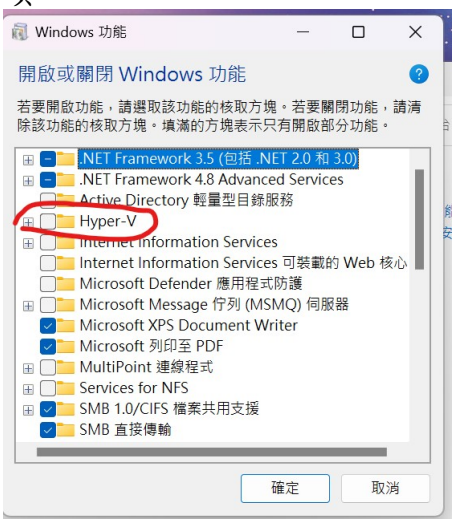
   (a) Host 是 Linux，在終端執行以下命令：

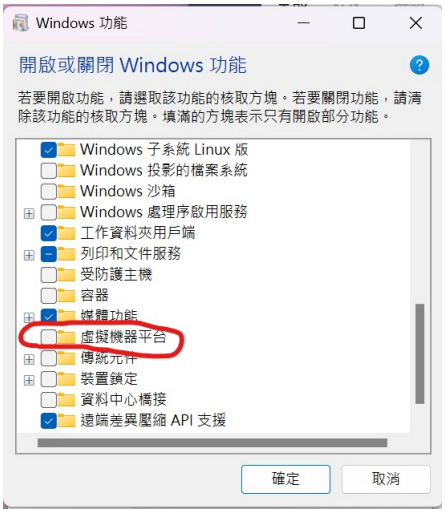
```
[dywang@dywmsi ~]$ VBoxManage modifyvm crt9 --nested-hw-virt on
```

   (b) Host 是 window，在命令提示 cmd 執行以下命令：

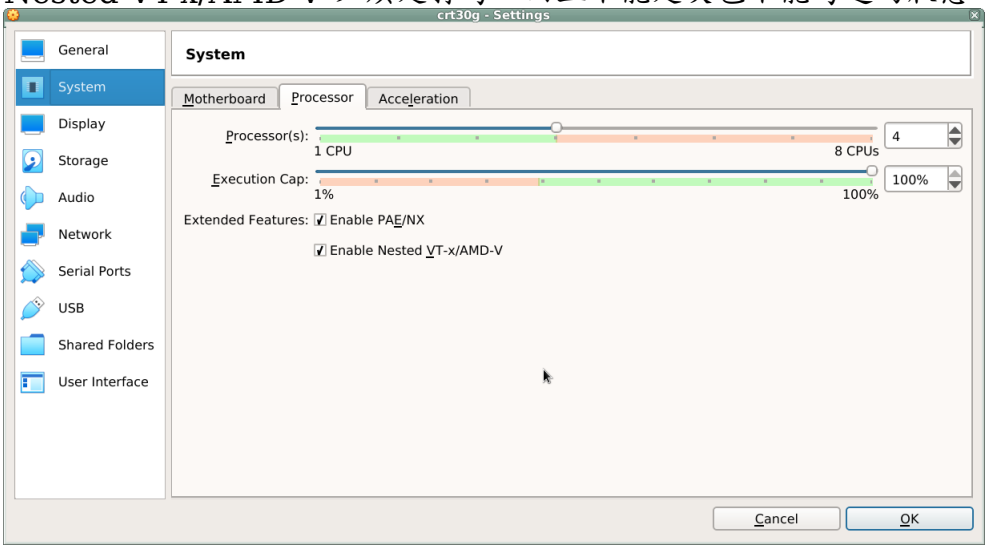
```
cd "c:\Program Files\Oracle\VirtualBox"
VBoxManage.exe modifyvm crt9 --nested-hw-virt on
```

   (c) Host 是 window，還必須關閉「Hyper-V」及「虛擬機器平台」，兩個選項。

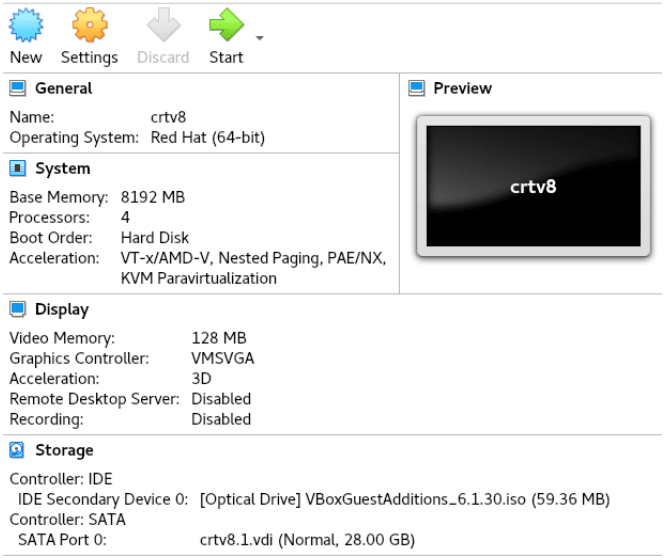




6. 進 VirtualBox 查看 VM crt9 的設定，system -> Processor -> Enable Nested VT-x/AMD-V 必須是打勾，而且不能是灰色不能勾選的狀態。



7. 使用 VirtualBox 或其他虛擬化工具，都要提供 VM crt9 足夠的 CPU 及 RAM，且要開啓巢式虛擬化，只要這三個要求達到，應該不會有□題，其他設定不懂可以先用預設值。若進不了圖形界面，可以打開 Display 的 3D 加速看看，VirtualBox VM 設定 (Settings) 參考值如下：



8. * 爲方便 VirtualBox 使用，建議下載附加套件 Oracle_VM_VirtualBox_Extension_Pack-6.1.30.vbox-extpack，放到你的 VirtualBox 目錄中，並於 VirtualBox 選單中加入附加 CD image。

```
Devices --> Insert Guest Additions CD image...
```

9. 開啓 DYW Linux 後，若虛擬機沒安裝，請先 rebuild 安裝虛擬機，練習後只需要 revert 還原，不用再要重新 rebuild，rebuild 要從學校 server 下載 image，要花較多時間。

1.4 練習系統製作-Raw Image

1. 下載 DYW Linux 練習系統 image 檔，或由 VDI 檔轉換格式，下載網址公告於本平台課程內，請登入查看。

2. 下載的 image 檔是 xz 壓縮檔，先用 unxz 解壓縮，windows 可用 winzip 解壓縮。

```
1 [root@dywmsi ~]# unxz crt9.img.xz
```

3. DYW Linux 練習系統，Host 主機以 virsh 管理 VM，最低硬體因所需 VM 個數不同：

   (a) CPU 4 核心以上，若修的課不用內部虛擬機，設定 2 核心也可以。

   (b) RAM 8G 以上，若修的課不用內部虛擬機，設定 4G RAM 也可以。

(c) RHCSA, RHCE 請設定 4 核心，8G RAM 以上

(d) CPU 支援虛擬化技術 (virtualization technology)，BIOS 啓用 VT。

4. 將 image 檔 dd 寫到 32G 以上硬碟或隨身碟，其中 /dev/sdX 是您要寫入的硬碟或隨身碟。

```
1 [root@dywmsi ~]# dd if=crt9.img of=/dev/sdX bs=100M
```

5. /dev/sdX 整顆硬碟的資料會被覆蓋，請確認要寫入的硬碟無誤再按 Enter 執行。

6. dd 完成後以 /dev/sdX 開機。

7. 開啓 DYW Linux 後，若虛擬機沒安裝，請先 rebuild 安裝虛擬機，練習後只需要 revert 還原，不用再要重新 rebuild，rebuild 要從學校 server 下載 image，要花較多時間。

1.5 NAT 設計

1. NAT 架設：電腦教室網路設置、管控學生上課上網。

2. 在 Linux，一行命令即可完成設定；在 Windows，Honda 老師亦幫忙撰寫批次檔。控制方式：

3. 任何設定皆不管控老師機，老師機可自由上網。

4. 學生機上網方式：

(a) 自由上網

(b) 不能上網

(c) 限制只能上學內網

(d) 限制只能上某一網站

(e) 限制只能上「依老師設定」的部分網站

(f) 只有某一學生機可自由上網

5. 不管老師做任何網路設定，下課後系統自動恢復成所有電腦皆可自由上網，以避免因老師忘了復原，而影響下節上課的老師。

1.6 虛擬機管理界面

1. 設計虛擬機管理 GUI 程式，方便學生管理虛擬機。程式包含虛擬機重安裝、還原、重新啓動及終端界面等四個功能。

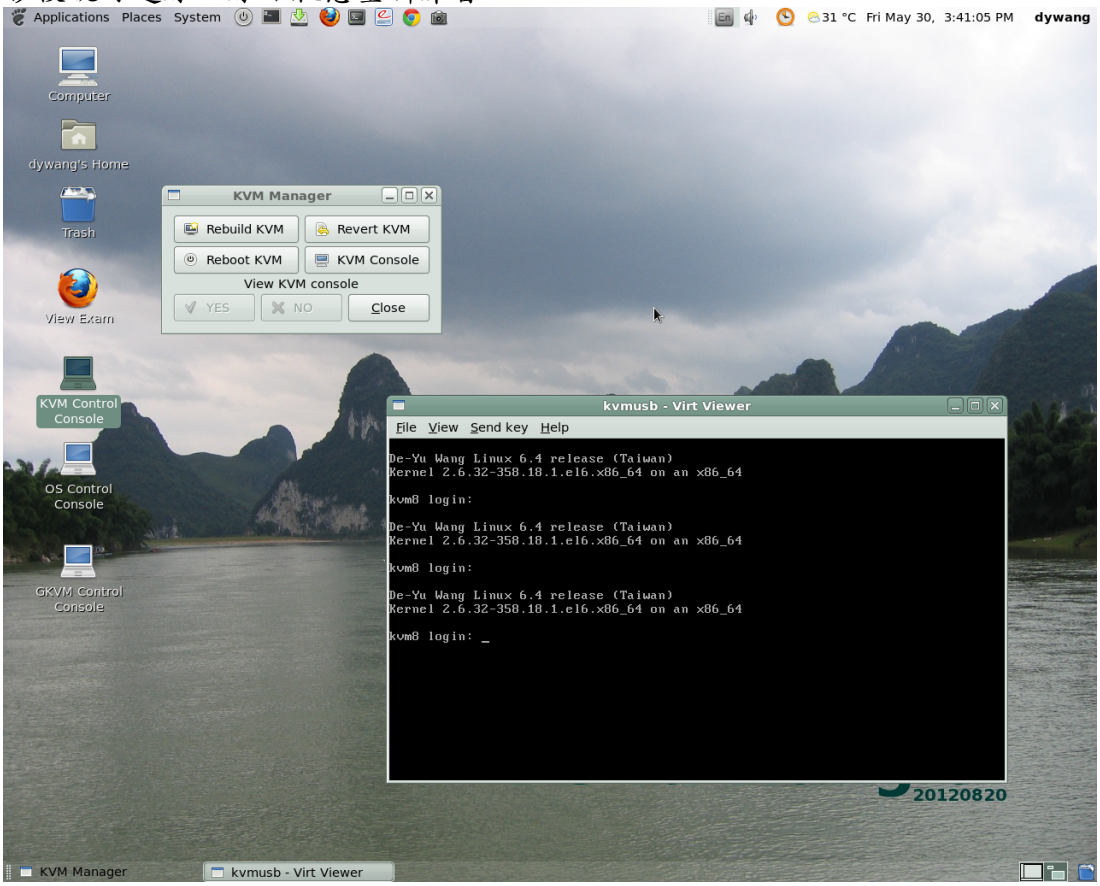
   (a) Rebuild KVM：重新安裝 KVM 虛擬機，約 10 幾分鐘。

   (b) Revert KVM：還原 KVM 虛擬機，約幾秒鐘到幾分鐘，請耐心等侯，不要在還原過程按管理界面的其他功能，以免造成虛擬機損壞。

   (c) Reboot KVM：重新啓動 KVM 虛擬機。

   (d) KVM Console：打開 KVM 虛擬機終端畫面。

2. 系統桌面中黑色背景視窗爲 KVM 虛擬機操作界面。原始 KVM 虛擬機主機名稱爲 111111，因已練習過，主機也已被設定成 kvm8，點擊 Revert KVM，幾秒後就可還原回原始狀態重新練習。



3. 在終端機以指令 virt-view 直接開啓虛擬機，出現錯誤訊息。

```
[dywang@dywH ~]$ sudo virt-viewer kvmusb
No protocol specified
Cannot open display:
Run 'virt-viewer --help' to see a full list of available command line
    options
```

4. 先檢查 Xwindow 認證變數 $XAUTHORITY 存不存在？

```
[dywang@dywH ~]$ echo $XAUTHORITY
```

5. 變數 XAUTHORITY 不存在，必須先設定 XAUTHORITY=~/.Xauthority 後再使用 sudo 啓動虛擬機，才能正常啓動。

```
[dywang@dywH ~]$ XAUTHORITY=~/.Xauthority sudo virt-viewer kvmusb
```

1.7 虛擬機管理界面 V2

1. 設計虛擬機管理 GUI 程式，方便學生管理虛擬機。程式包含虛擬機重安裝、還原、重新啓動、終端界面、啓用實體 USB 隨身碟、解除實體 USB 隨身碟的使用等六個功能。



(a) Rebuild KVM：重新安裝 KVM 虛擬機，直接下載 qcow2 映像檔，產生虛擬機，校內網路約 1 分多鐘。

(b) Revert KVM：還原 KVM 虛擬機，約幾秒鐘到幾分鐘，請耐心等侯，務必等黑色終端出現登入畫面，表示還原完成。練習前先關閉黑色終端畫面讓還原程式拍照，記下還原狀態，以免造成虛擬機損壞。

(c) Reboot KVM：重新啓動 KVM 虛擬機。

(d) KVM Console：打開 KVM 虛擬機終端畫面。

(e) Attach USB：插上 USB 隨身碟，按下此按鈕，KVM 虛擬機啓動實體 USB 隨身碟。

(f) Detach USB：KVM 虛擬機解除實體 USB 隨身碟的使用。

1.8 系統管理與評分

1. 系統管理腳本程式自動掃瞄目前開機的電腦，選擇要進行的工作，輸入密碼後即可自動完成。

```
[root@dywH ~]# e517sshpass.sh
myip=104
hosts=103,105
q(quit) iscsi halt reboot c(clear) l(lvm) rhcsa rhce
>  t(mounttest) k(kvmreboot) v(kvmrevert) s(servicerestart)
>  vi1 c1 lpvar c999 sh999 hp(hostpatch)? c999
keyin remote host root password:
```

2. 預設工作項目：

   (a) q(quit)：退出，無動作。

   (b) iscsi：學生機 ISCSI 硬碟分享架設。

   (c) halt：學生機關機。

   (d) reboot：學生機重新啓動。

   (e) c(clear)：登入學生機後清除部分資料，此部分保留修改彈性，目前入作爲限制學生機互相登入之限制，以防代作答情況。

   (f) l(lvm)：硬碟分割、格式化、自動掛載評分。

   (g) rhcsa：RHCSA 考照練習總評分。

   (h) rhce：RHCE 考照練習總評分。

   (i) t(mounttest)：測試學生機是否有掛載隨身碟或光碟。

   (j) k(kvmreboot)：學生機 KVM 虛擬機重新啓動。

   (k) v(kvmrevert)：學生機 KVM 虛擬機還原。

   (l) s(servicerestart)：啓動學生機主機服務。

   (m) vi1：vi 編輯器實機練習評分。

   (n) c1：C 語言第一支程式練習評分。

   (o) lpvar：shell 變數變化練習評分。

   (p) c999：C 語言程式實機練習評分。

   (q) sh999：Shell script 程式練習評分。

   (r) hp(hostpatch)：學生機網路修正後重新啓動網路。

3. 遠距教學，在家練習，自己評分上傳：

(a) 登入本平台，下載評分腳本及參數檔，以「資料庫」第一次評分爲例。

(b) 切換身分爲 root，root 密碼在平台部落格，請登入。

```
[dywang@deyu ~]$ su -
Password:
[root@deyu ~]#
```

(c) 切換工作目錄到評分程式的下載目錄，用瀏覽器下載預設是放在家目錄的 Downloads。

```
[root@deyu ~]# cd /home/dywang/Downloads
```

(d) 下載的評分程式沒有可執行的權限，先增加評分程式 mariadbeval1.sh.x 可執行屬性，使評分腳本 mariadbeval1.sh.x 成爲可執行檔。

```
[root@deyu Downloads]# chmod +x mariadbeval1.sh.x
```

(e) 查看目前有兩個檔案，評分程式爲 mariadbeval1.sh.x，參數檔爲 mariadbpara1.sh。

```
[root@deyu Downloads]# ll
-rwxrwxr-x. 1 root root 58656 May 21 13:29 mariadbeval1.sh.x
```

(f) 執行評分腳本 mariadbeval1.sh.x

```
[root@deyu Downloads]# ./mariadbeval1.sh.x
20210521-13:46:36
./mariadbeval1.sh.x
163.17.10.123
kvm=kvm19
163.17.10.123 10927000 王大呆
mariadbpara1.sh
mariadbeval1.sh.x_kvm19_1.mysqlversion:0
mariadbeval1.sh.x_kvm19_2.mariadb_enabled:0
mariadbeval1.sh.x_kvm19_3.mariadb_active:0
mariadbeval1.sh.x_kvm19_4.rootpw:0
```

(g) 如果評分參數檔不在同目錄，評分腳本 mariadbeval1.sh.x 無法執行。

```
[root@deyu Downloads]# mv mariadbpara1.sh /tmp/
```

```
[root@deyu Downloads]# ./mariadbeval1.sh
20210521-13:48:51
./mariadbeval1.sh
163.17.10.123
kvm=kvm19
163.17.10.123 10927000 王大呆
ls: cannot access mariadbpara1: No such file or directory
NO mariadbpara1 file
```

1.9 系統註冊-上課前先確定三件事

1. E517 電腦教室

   (a) E517 教室位置，靠窗爲第一排，確認自己電腦位置爲?-?。例如：1-1。

   (b) 在伺服器 (也就是開機進入圖形界面中有虛擬機管理界面的這台主機)，不是虛擬機，確認自己的電腦 IP。以下例子爲 192.168.1.140，IP 尾數爲 140。

```
[dywang@deyu ~]$ ifconfig | grep 192
          inet addr:192.168.1.140  Bcast:192.168.1.255  Mask
              :255.255.255.0
          inet addr:192.168.122.1  Bcast:192.168.122.255  Mask
              :255.255.255.0
```

   (c) 在伺服器 deyu.wang 主機一般帳號 dywang 家目錄，產生學號姓名檔 sid，檔案資訊格式爲「電腦位置 IP 尾數學號姓名」，每一欄位中間以一個空格隔開。以下例子電腦位置 1-11、IP 尾數 199、學號 123456，姓名王大呆。

```
[dywang@deyu ~]$ echo '1-11 199 123456 王大呆' > ~/sid
[dywang@deyu ~]$ cat ~/sid
1-11 199 123456 王大呆
```

2. 在家練習

   (a) 查詢練習系統對外 public ip 有三種方式：

```
[dywang@deyu ~]$ curl -4 ipinfo.io/ip
163.17.10.122[dywang@deyu ~]$
[dywang@deyu ~]$ curl -4 icanhazip.com
163.17.10.122
```

```
5 [dywang@deyu ~]$ curl -4  ifconfig.me
  163.17.10.122[dywang@deyu ~]$
```

(b) 將 ip 學號姓名寫入 dywang 家目錄的 sid 檔，如下範例：

```
[dywang@deyu ~]$ echo '163.17.10.123 10827000 王大呆' > sid
```

(c) 查看 sid，ip 學號姓名務必正確，以免評分錯誤或無法匯入成績。

```
1 [dywang@deyu ~]$ cat sid
  163.17.10.123 10827000 王大呆
```

1.10 虛擬機設定

1. 伺服器(也就是開機進入圖形界面中有虛擬機管理界面的這台主機)，不確定的話，先看命令列提示符號是不是 [dywang@deyu ~]$，或是 ifconfig 查目前所在主機的 IP：

   (a) Server 主機有兩個名稱 deyu.wang 及 server.deyu.wang

   (b) Server 與虛擬機網域 deyu.wang，使用網段 192.168.122.0/24。

   (c) Server IP 爲 192.168.122.1，虛擬機 IP 爲 192.168.122.X ，其中 X 依題目要求設定。

2. 虛擬機環境說明：

   (a) 虛擬機主機名稱 kvmX.deyu.wang IP 192.168.122.X，X 爲虛擬機編號，例如 5 或 7。

   (b) 虛擬機 root 密碼無法取得，統一重新設定爲 123qwe。

   (c) RHCE 練習環境增加
   網域 my111.wang，使用網段 192.168.111.0/24，
   server 主機增加此網域 IP 192.168.111.1。

   (d) 虛擬機網路設定：

```
IP Address:   192.168.122.X
2 Netmask:    255.255.255.0
  Gateway:    192.168.122.1
4 Name server: 192.168.122.1
```

(e) RHCSA/RHCE 6,7 repository 套件庫網址：
http://dywang.csie.cyut.edu.tw/centos7.

(f) RHCSA 8 repository 套件庫網址：

i. BaseOS repo server:
http://dywang.csie.cyut.edu.tw/centos8/BaseOS.

ii. AppStream repo server:
http://dywang.csie.cyut.edu.tw/centos8/AppStream.

(g) kernel packages: 網路下載核心 RPM 檔
http://dywang.csie.cyut.edu.tw/kernel/centos7.

(h) LDAP server 設定：

i. 系統 deyu.wang

ii. base DN 爲 dc=deyu,dc=wang

iii. 證書 ftp://deyu.wang/pub/cacert.pem.

1.11 系統伺服器管理

1. 使用虛擬機練習時，伺服器 (也就是開機進入圖形界面中有虛擬機管理界面的這台主機) 必須正常運作，否則有些驗證無法進行。

2. 練習用到的 server 都已寫成腳本，如果運作有問題，可以執行腳本重啓。

1.11.1 DNS server

(a) 如果確定你的練習虛擬機 DNS 設定沒問題，但 ssh 卻無法使用主機名稱連線到虛擬機，可以在 server 機使用 host 命令看是否可以查到虛擬機的 IP？

(b) 沒有 DNS server 可查。

```
[dywang@deyu ~]$ host kvm7.deyu.wang
;; connection timed out; trying next origin
;; connection timed out; no servers could be reached
```

(c) 在 DNS server 查不到 kvm7.deyu.wang 的 IP。

```
[dywang@deyu ~]$ host kvm7.deyu.wang
Host kvm7.deyu.wang not found: 3(NXDOMAIN)
```

(d) 在伺服器 deyu.wang 主機切換成 root 身份。

```
[dywang@deyu ~]$ su -
Password: 123123
[root@deyu ~]#
```

(e) 執行 setdnsmasq.sh 腳本重啓 dnsmasq DNS 服務。

```
[root@deyu ~]# setdnsmasq.sh restart
```

(f) 再使用 host 查詢 kvm7.deyu.wang，已可查到其 IP 爲 192.168.122.7。

```
[root@deyu ~]# host kvm7.deyu.wang
kvm7.deyu.wang has address 192.168.122.7
kvm7.deyu.wang mail is handled by 1 20.
```

(g) 如果執行 setdnsmasq.sh 腳本重啓 dnsmasq DNS 服務未能解決問題，還可以使用 setdns.sh 腳本啓動 named DNS 服務。

```
[root@deyu ~]# setdns.sh restart
```

1.11.2 LDAP server

(a) 如果確認你的虛擬機 LDAP 設定沒問題，但還是無法以 ldapuser1 登入。在伺服器 deyu.wang 主機 (不是你的練習虛擬機)，先切換成 root 身份。

```
[dywang@deyu ~]$ su -
Password: 123123
[root@deyu ~]#
```

(b) 在伺服器執行 setldap1.sh restart 重新產生 LDAP 憑證。因爲憑證是新產生的，所以虛擬機必須再重新執行認證設定更新。

```
[root@deyu ~]# setldap1.sh restart
```

1.11.3 NFS server

(a) 虛擬機設定自動掛載必須查得到 server 有分享目錄，server 的 NFS 不像 LDAP 有憑證過期問題，所以一般都不需要重啓，但如果要重啓 NFS server，先在伺服器 deyu.wang 主機切換成 root 身份。

```
[dywang@deyu ~]$ su -
Password: 123123
[root@deyu ~]#
```

(b) 執行 setnfs.sh 腳本重新啓動。

```
[root@deyu ~]# setnfs.sh restart
```

Chapter 2

命令列基本操作

2.1 Linux 指令

1. 簡單指令

    - `man` - page 21：查詢指令使用手冊。
    - `date` - page 24：顯示日期及時間。
    - `cal` - page 27：顯示日月曆。
    - `bc` - page 31：簡單好用的計算機。
    - `sync` - page **??**：手動將緩衝於記憶體內的磁碟資料寫入磁碟。
    - `who` - page **??**：查看誰在線上。
    - `shutdown` - page **??**：通知線上使用者關機的時刻後關機。
    - `halt` - page **??**：關機。
    - `reboot` - page **??**：重新開機。
    - `chgrp` - page **??**：改變檔案所屬群組。
    - `chown` - page **??**：改變檔案擁有者。
    - `chmod` - page 123：改變權限。
    - `cd` - page **??**：變換工作目錄。
    - `pwd` - page **??**：顯示目前所在的目錄。
    - `mkdir` - page **??**：建立新目錄。
    - `rmdir` - page **??**：刪除【空】的目錄。
    - `echo` - page **??**：顯示變數內容或文字。
    - `ls` - page **??**：顯示檔案名稱、屬性等。

- cp - page **??**：拷貝檔案或目錄。
- rm - page **??**：移除檔案或目錄。
- mv - page **??**：檔案移動或改名。
- cat - page **??**：由第一行開始連續 (concatenate) 顯示檔案內容。
- touch - page **??**：修改檔案時間，若檔案不存在，則產生新檔案。
- find - page 154：功能強大的檔案搜尋指令。
- tar - page **??**：多用途打包壓縮指令。
- df - page **??**：查看檔案系統空間的使用狀況。
- fdisk - page **??**：對硬碟進行分割。
- mount - page **??**：掛載檔案系統。
- useradd - page 101：增加使用者。
- passwd - page **??**：設定或變更密碼。
- userdel - page **??**：刪除使用者。
- id - page **??**：查詢使用者的 UID、GID 及所擁有的群組。
- groupadd - page **??**：增加群組。
- su - page 112：轉換用户身份。
- crontab - page 139：循環執行的工作排程。

2.2 線上求助

1. 大部分指令都有 --help 的選項，但不是所有指令皆有此選項，例如：

```
[root@mdk-dyw root]# ls --help
Usage: ls [OPTION]... [FILE]...
List information about the FILEs (the current directory by default).
Sort entries alphabetically if none of -cftuSUX nor --sort.

Mandatory arguments to long options are mandatory for short options too.
  -a, --all                  do not hide entries starting with .
  -A, --almost-all           do not list implied . and ..
      --author               print the author of each file
  -b, --escape               print octal escapes for nongraphic
      characters
      --block-size=SIZE      use SIZE-byte blocks
  -B, --ignore-backups       do not list implied entries ending with ~
  -c                         with -lt: sort by, and show, ctime (time of
       last
                               modification of file status information)
```

```
                               with -l: show ctime and sort by name
                               otherwise: sort by ctime
-C                             list entries by columns
    --color[=WHEN]             control whether color is used to
        distinguish file
                               types.  WHEN may be `never', `always', or
                                    `auto'
-d, --directory                list directory entries instead of contents,
                               and do not dereference symbolic links
-D, --dired                    generate output designed for Emacs' dired
    mode
-f                             do not sort, enable -aU, disable -lst
-F, --classify                 append indicator (one of */=@|) to entries
    --format=WORD              across -x, commas -m, horizontal -x, long -
        l,
                               single-column -1, verbose -l, vertical -C
    --full-time                like -l --time-style=full-iso
-g                             like -l, but do not list owner
-G, --no-group                 inhibit display of group information
-h, --human-readable  print sizes in human readable format (e.g., 1K
    234M 2G)
    --si                       likewise, but use powers of 1000 not 1024
-H, --dereference-command-line
                               follow symbolic links listed on the command
                                    line
    --dereference-command-line-symlink-to-dir
                               follow each command line symbolic link
                               that points to a directory
    --indicator-style=WORD append indicator with style WORD to entry
        names:
                               none (default), classify (-F), file-type
                                   (-p)
-i, --inode                    print index number of each file
-I, --ignore=PATTERN           do not list implied entries matching shell
    PATTERN
-k                             like --block-size=1K
-l                             use a long listing format
-L, --dereference              when showing file information for a
    symbolic
                               link, show information for the file the
                                   link
                               references rather than for the link
                                   itself
-m                             fill width with a comma separated list of
    entries
-n, --numeric-uid-gid          like -l, but list numeric UIDs and GIDs
-N, --literal                  print raw entry names (don't treat e.g.
    control
                               characters specially)
-o                             like -l, but do not list group information
-p, --file-type                append indicator (one of /=@|) to entries
-q, --hide-control-chars       print ? instead of non graphic characters
    --show-control-chars       show non graphic characters as-is (default
                               unless program is `ls' and output is a
```

```
                                terminal)
  -Q, --quote-name            enclose entry names in double quotes
      --quoting-style=WORD    use quoting style WORD for entry names:
                                literal, locale, shell, shell-always, c,
                                    escape
  -r, --reverse               reverse order while sorting
  -R, --recursive             list subdirectories recursively
  -s, --size                  print size of each file, in blocks
  -S                          sort by file size
      --sort=WORD             extension -X, none -U, size -S, time -t,
                                version -v
                              status -c, time -t, atime -u, access -u,
                                  use -u
      --time=WORD             show time as WORD instead of modification
                                 time:
                                atime, access, use, ctime or status; use
                                specified time as sort key if --sort=time
      --time-style=STYLE      show times using style STYLE:
                                full-iso, long-iso, iso, locale, +FORMAT
                              FORMAT is interpreted like `date'; if
                                  FORMAT is
                              FORMAT1<newline>FORMAT2, FORMAT1 applies to
                              non-recent files and FORMAT2 to recent
                                  files;
                              if STYLE is prefixed with `posix-', STYLE
                              takes effect only outside the POSIX locale
  -t                          sort by modification time
  -T, --tabsize=COLS          assume tab stops at each COLS instead of 8
  -u                          with -lt: sort by, and show, access time
                                with -l: show access time and sort by
                                    name
                                otherwise: sort by access time
  -U                          do not sort; list entries in directory
                                 order
  -v                          sort by version
  -w, --width=COLS            assume screen width instead of current
                                 value
  -x                          list entries by lines instead of by columns
  -X                          sort alphabetically by entry extension
  -1                          list one file per line
      --help     display this help and exit
      --version  output version information and exit

SIZE may be (or may be an integer optionally followed by) one of
                                  following:
kB 1000, K 1024, MB 1000*1000, M 1024*1024, and so on for G, T, P, E, Z,
                                   Y.

By default, color is not used to distinguish types of files.  That is
equivalent to using --color=none.  Using the --color option without the
optional WHEN argument is equivalent to using --color=always.  With
--color=auto, color codes are output only if standard output is
                                  connected
to a terminal (tty).
```

```
97
Report bugs to <bug-coreutils@gnu.org>.
```

2. 其他基本指令之 -help 功能：

   (a) date - page 24：顯示日期及時間。

   (b) cal - page 27：顯示日月曆。

   (c) bc - page 31：簡單好用的計算機。

3. 一般指令之 -help 選項可能提供：

   (a) 指令使用方法；

   (b) 選項之用途；

   (c) 若選項需要加入參數，則說明參數之格式；

   (d) 簡易說明。

2.3 man page

1. man：查詢指令使用手冊。

```
[dywang@mdk-dyw ~]$ man name
2 參數說明：
name: 要查詢的指令。
4
#範例：查詢 cal(calendar) 指令用法。
6 [dywang@mdk-dyw ~]$ man cal
CAL(1)                    BSD General Commands Manual
    CAL(1)
8
NAME
10      cal - displays a calendar

12 SYNOPSIS
      cal [-smjy13] [[month] year]
14
DESCRIPTION
16      Cal displays a simple calendar.  If arguments are not specified,
          the cur-
      rent month is displayed.  The options are as follows:
18
      -1      Display single month output.  (This is the default.)
20
      -3      Display prev/current/next month output.
22
```

```
     -s      Display Sunday as the first day of the week.  (This is the
             default.)

     -m      Display Monday as the first day of the week.

     -j      Display Julian dates (days one-based, numbered from January
         1).

q    -y      Display a calendar for the current year.

     A single parameter specifies the year (1 - 9999) to be displayed;
        note
     the year must be fully specified: ``cal 89'' will not display a
        calendar
     for 1989.  Two parameters denote the month (1 - 12) and year.  If
        no
     parameters are specified, the current month's calendar is displayed
        .

     A year starts on Jan 1.

     The Gregorian Reformation is assumed to have occurred in 1752 on
        the 3rd
     of September.  By this time, most countries had recognized the
        reforma-
     tion (although a few did not recognize it until the early 1900's.)
         Ten
     days following that date were eliminated by the reformation, so the
         cal-
     endar for that month is a bit unusual.

HISTORY
     A cal command appeared in Version 6 AT&T UNIX.

OTHER VERSIONS
     Several much more elaborate versions of this program exist, with
        support
     for colors, holidays, birthdays, reminders and appointments, etc.
        For
     example, try the cal from http://home.sprynet.com/~cbagwell/
        projects.html
     or GNU gcal.

BSD                                 June 6, 1993
                                    BSD
```

2. man 的相關檔案

   (a) 設定檔 /etc/man.config

   (b) 資料目錄 /usr/share/man

   (c) 文件目錄 /usr/share/doc

3. 數字代號

代號	代表內容
1	使用者可以操作的指令或可執行檔
2	系統核心可呼叫的函式與工具
3	常用函式 (function) 與函式庫 (library)
4	裝置檔案的說明
5	設定檔
6	遊戲
7	慣例與協定
8	系統管理員可用的管理指令
9	跟核心有關的文件

4. 內容代號

代號	代表內容
NAME	簡短指令、資料名稱說明
SYNOPSIS	簡短指令下達語法簡介
DESCRIPTION	較完整的說明
OPTIONS	針對 SYNOPSIS 中列舉的所有參數說明
COMMANDS	當此程式執行中可下達的指令
FILES	此程式所使用、參考或連結的檔案
SEE ALSO	與此指令相關的說明
EXAMPLE	參考範例
BUGS	是否有相關的臭蟲
HISTORY	指令歷史
OTHER VERSIONS	是否有其他版本

5. man 相關按鍵有：

按鍵	進行工作
空白鍵	往下翻一頁。
page up	往上翻一頁。
page down	往下翻一頁。
home	回到最前頁。
end	到最終頁。
/word	向下搜尋 word 這個字。
?word	向上搜尋 word 這個字。
n,N	n 繼續下一個搜尋，N 繼續反向搜尋。
q	離開。

6. man 使用功能

```
[root@mdk-dyw root]# man --help
man, version 1.5m

usage: man [-adfhktwW] [section] [-M path] [-P pager] [-S list]
        [-m system] [-p string] name ...

  a : find all matching entries
  c : do not use cat file
  d : print gobs of debugging information
  D : as for -d, but also display the pages
  f : same as whatis(1)
  h : print this help message
  k : same as apropos(1)
  K : search for a string in all pages
  t : use troff to format pages for printing
  w : print location of man page(s) that would be displayed
      (if no name given: print directories that would be searched)
  W : as for -w, but display filenames only

  C file   : use `file' as configuration file
  M path   : set search path for manual pages to `path'
  P pager  : use program `pager' to display pages
  S list   : colon separated section list
  m system : search for alternate system's man pages
  p string : string tells which preprocessors to run
              e - [n]eqn(1)   p - pic(1)    t - tbl(1)
              g - grap(1)     r - refer(1)  v - vgrind(1)
```

2.4 基礎指令操作

1. date：顯示日期及時間。

```
[dywang@dywHome2 ~]$ date --help
Usage: date [OPTION]... [+FORMAT]
  or:  date [-u|--utc|--universal] [MMDDhhmm[[CC]YY][.ss]]
Display the current time in the given FORMAT, or set the system date.

  -d, --date=STRING         display time described by STRING, not `now'
                          ## 以 STRING 格式顯示時間，不是'現在'時間。

  -f, --file=DATEFILE       like --date once for each line of DATEFILE
  -r, --reference=FILE      display the last modification time of FILE
  -R, --rfc-2822            output date and time in RFC 2822 format
      --rfc-3339=TIMESPEC   output date and time in RFC 3339 format.
                            TIMESPEC=`date', `seconds', or `ns' for
                            date and time to the indicated precision.
  -s, --set=STRING          set time described by STRING
                          ## 以 STRING 格式設定時間。
  -u, --utc, --universal    print or set
```

```
                    ## 顯示目前的格林威治時間。
      --help     display this help and exit
      --version  output version information and exit

FORMAT controls the output.  The only valid option for the second form
specifies Coordinated Universal Time.  Interpreted sequences are:

  %%   a literal %
  %a   locale's abbreviated weekday name (e.g., Sun)
  %A   locale's full weekday name (e.g., Sunday)
  %b   locale's abbreviated month name (e.g., Jan)
  %B   locale's full month name (e.g., January)
  %c   locale's date and time (e.g., Thu Mar  3 23:05:25 2005)
  %C   century; like %Y, except omit last two digits (e.g., 21)
  %d   day of month (e.g, 01)
  %D   date; same as %m/%d/%y
  %e   day of month, space padded; same as %_d
  %F   full date; same as %Y-%m-%d
  %g   last two digits of year of ISO week number (see %G)
  %G   year of ISO week number (see %V); normally useful only with %V
  %h   same as %b
  %H   hour (00..23)
  %I   hour (01..12)
  %j   day of year (001..366)
  %k   hour ( 0..23)
  %l   hour ( 1..12)
  %m   month (01..12)
  %M   minute (00..59)
  %n   a newline
  %N   nanoseconds (000000000..999999999)
  %p   locale's equivalent of either AM or PM; blank if not known
  %P   like %p, but lower case
  %r   locale's 12-hour clock time (e.g., 11:11:04 PM)
  %R   24-hour hour and minute; same as %H:%M
  %s   seconds since 1970-01-01 00:00:00 UTC
  %S   second (00..60)
  %t   a tab
  %T   time; same as %H:%M:%S
  %u   day of week (1..7); 1 is Monday
  %U   week number of year, with Sunday as first day of week (00..53)
  %V   ISO week number, with Monday as first day of week (01..53)
  %w   day of week (0..6); 0 is Sunday
  %W   week number of year, with Monday as first day of week (00..53)
  %x   locale's date representation (e.g., 12/31/99)
  %X   locale's time representation (e.g., 23:13:48)
  %y   last two digits of year (00..99)
  %Y   year
  %z   +hhmm numeric timezone (e.g., -0400)
  %:z  +hh:mm numeric timezone (e.g., -04:00)
  %::z  +hh:mm:ss numeric time zone (e.g., -04:00:00)
  %:::z  numeric time zone with : to necessary precision (e.g., -04,
      +05:30)
  %Z   alphabetic time zone abbreviation (e.g., EDT)

```

```
時間方面 :

 %% : 印出 %
 %n : 下一行
 %t : 跳格
 %H : 小時(00..23)
 %I : 小時(01..12)
 %k : 小時(0..23)
 %l : 小時(1..12)
 %M : 分鐘(00..59)
 %p : 顯示本地 AM 或 PM
 %r : 直接顯示時間 (12 小時制，格式爲 hh:mm:ss [AP]M)
 %s : 以 1970 年 1 月 1 日 00:00:00 UTC 到目前爲止的秒數
 %S : 秒(00..61)
 %T : 直接顯示時間 (24 小時制)
 %X : 相當於 %H:%M:%S
 %Z : 顯示時區

 日期方面 :
 %a : 星期幾 (Sun..Sat)
 %A : 星期幾 (Sunday..Saturday)
 %b : 月份 (Jan..Dec)
 %B : 月份 (January..December)
 %c : 直接顯示日期與時間
 %d : 日 (01..31)
 %D : 直接顯示日期 (mm/dd/yy)
 %h : 同 %b
 %j : 一年中的第幾天 (001..366)
 %m : 月份 (01..12)
 %U : 一年中的第幾周 (00..53) (以 Sunday 爲一周的第一天的情形)
 %w : 一周中的第幾天 (0..6)
 %W : 一年中的第幾周 (00..53) (以 Monday 爲一周的第一天的情形)
 %x : 直接顯示日期 (mm/dd/yy)
 %y : 年份的最後兩位數字 (00.99)
 %Y : 完整年份 (0000..9999)


By default, date pads numeric fields with zeroes.
The following optional flags may follow `%':

  - (hyphen) do not pad the field
  _ (underscore) pad with spaces
  0 (zero) pad with zeros
  ^ use upper case if possible
  # use opposite case if possible

After any flags comes an optional field width, as a decimal number;
then an optional modifier, which is either
E to use the locale's alternate representations if available, or
O to use the locale's alternate numeric symbols if available.

Report bugs to <bug-coreutils@gnu.org>.

[dywang@mdk-dyw ~]$ date +%Y/%m/%d
```

```
2006/10/05

[dywang@dywOffice ~]$ date +%H:%M
09:37

[dywang@dywHome2 ~]$  # 顯示時間後跳行，再顯示目前日期:
[dywang@dywHome2 ~]$ date '+%T%n%D'
18:16:19
08/01/09

[dywang@dywHome2 ~]$  # 顯示月份與日
[dywang@dywHome2 ~]$ date '+%B %d'
August 01

[dywang@dywHome2 ~]$  # 明天
[dywang@dywHome2 ~]$ date --date="1 day" +%F
2009-08-02

[dywang@dywHome2 ~]$  # 後天
[dywang@dywHome2 ~]$  date --date="2 day" +%F
2009-08-03

[dywang@dywHome2 ~]$  # 昨天
[dywang@dywHome2 ~]$  date --date="1 day ago" +%F
2009-07-31

[dywang@dywHome2 ~]$  # 前天
[dywang@dywHome2 ~]$  date --date="2 day ago" +%F
2009-07-30

[dywang@dywHome2 ~]$  # 2009-1-1 的前一天
[dywang@dywHome2 ~]$  date --date="2009-1-1 1 day ago" +%F
2008-12-31

[dywang@dywHome2 ~]$  # 2008-2-28 的後一天
[dywang@dywHome2 ~]$  date --date="2008-2-28 1 day" +%F
2008-02-29

[dywang@dywHome2 ~]$  # 2008-2-28的兩個月前
[dywang@dywHome2 ~]$  date --date="2008-2-28 2 month ago" +%F
2007-12-28

[root@dywHome2 ~]#  # 設定時間爲 12:34:56，必須以 root 身份
[root@dywHome2 ~]# date -s '12:34:56'
Sat Aug  1 12:34:56 CST 2009
```

2. cal：顯示日月曆。

```
[dywang@dywHome2 ~]$ cal --help
cal: invalid option -- -
usage: cal [-13smjyV] [[month] year]

```

```
[dywang@mdk-dyw ~]$ man cal
CAL(1)                    BSD General Commands Manual
    CAL(1)

NAME
     cal - displays a calendar

SYNOPSIS
     cal [-smjy13] [[month] year]

DESCRIPTION
     Cal displays a simple calendar.  If arguments are not specified,
        the cur-
     rent month is displayed.  The options are as follows:

     -1      Display single month output.  (This is the default.)

     -3      Display prev/current/next month output.

     -s      Display Sunday as the first day of the week.  (This is the
             default.)

     -m      Display Monday as the first day of the week.

     -j      Display Julian dates (days one-based, numbered from January
          1).

     -y      Display a calendar for the current year.

     A single parameter specifies the year (1 - 9999) to be displayed;
        note
     the year must be fully specified: ``cal 89'' will not display a
        calendar
     for 1989.  Two parameters denote the month (1 - 12) and year.  If
        no
     parameters are specified, the current month's calendar is displayed
        .

     A year starts on Jan 1.

     The Gregorian Reformation is assumed to have occurred in 1752 on
        the 3rd
     of September.  By this time, most countries had recognized the
        reforma-
     tion (although a few did not recognize it until the early 1900's.)
          Ten
     days following that date were eliminated by the reformation, so the
          cal-
     endar for that month is a bit unusual.

HISTORY
     A cal command appeared in Version 6 AT&T UNIX.

OTHER VERSIONS
```

```
     Several much more elaborate versions of this program exist, with
         support
     for colors, holidays, birthdays, reminders and appointments, etc.
         For
     example, try the cal from http://home.sprynet.com/~cbagwell/
         projects.html
     or GNU gcal.

BSD                                June 6, 1993
                                   BSD

[dywang@dywHome2 ~]$ cal
    August 2009
Su Mo Tu We Th Fr Sa
                    1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30 31
[dywang@dywHome2 ~]$ cal -1
    August 2009
Su Mo Tu We Th Fr Sa
                    1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30 31
[dywang@dywHome2 ~]$ cal -3
     July 2009            August 2009          September 2009
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
          1  2  3  4                     1         1  2  3  4  5
 5  6  7  8  9 10 11   2  3  4  5  6  7  8   6  7  8  9 10 11 12
12 13 14 15 16 17 18   9 10 11 12 13 14 15  13 14 15 16 17 18 19
19 20 21 22 23 24 25  16 17 18 19 20 21 22  20 21 22 23 24 25 26
26 27 28 29 30 31     23 24 25 26 27 28 29  27 28 29 30
                      30 31
[dywang@dywHome2 ~]$ cal -s
    August 2009
Su Mo Tu We Th Fr Sa
                    1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30 31
[dywang@dywHome2 ~]$ cal -m
    August 2009
Mo Tu We Th Fr Sa Su
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
```

```
24 25 26 27 28 29 30
31
[dywang@dywHome2 ~]$ cal -j
        August 2009
Sun Mon Tue Wed Thu Fri Sat
                          213
214 215 216 217 218 219 220
221 222 223 224 225 226 227
228 229 230 231 232 233 234
235 236 237 238 239 240 241
242 243
[dywang@dywHome2 ~]$ cal -y
                                2009

       January               February                March
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
             1  2  3    1  2  3  4  5  6  7    1  2  3  4  5  6  7
 4  5  6  7  8  9 10    8  9 10 11 12 13 14    8  9 10 11 12 13 14
11 12 13 14 15 16 17   15 16 17 18 19 20 21   15 16 17 18 19 20 21
18 19 20 21 22 23 24   22 23 24 25 26 27 28   22 23 24 25 26 27 28
25 26 27 28 29 30 31                          29 30 31

        April                 May                   June
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
          1  2  3  4                   1  2       1  2  3  4  5  6
 5  6  7  8  9 10 11    3  4  5  6  7  8  9    7  8  9 10 11 12 13
12 13 14 15 16 17 18   10 11 12 13 14 15 16   14 15 16 17 18 19 20
19 20 21 22 23 24 25   17 18 19 20 21 22 23   21 22 23 24 25 26 27
26 27 28 29 30         24 25 26 27 28 29 30   28 29 30
                       31
        July                  August                September
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
          1  2  3  4                      1          1  2  3  4  5
 5  6  7  8  9 10 11    2  3  4  5  6  7  8    6  7  8  9 10 11 12
12 13 14 15 16 17 18    9 10 11 12 13 14 15   13 14 15 16 17 18 19
19 20 21 22 23 24 25   16 17 18 19 20 21 22   20 21 22 23 24 25 26
26 27 28 29 30 31      23 24 25 26 27 28 29   27 28 29 30
                       30 31
       October               November              December
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
             1  2  3    1  2  3  4  5  6  7          1  2  3  4  5
 4  5  6  7  8  9 10    8  9 10 11 12 13 14    6  7  8  9 10 11 12
11 12 13 14 15 16 17   15 16 17 18 19 20 21   13 14 15 16 17 18 19
18 19 20 21 22 23 24   22 23 24 25 26 27 28   20 21 22 23 24 25 26
25 26 27 28 29 30 31   29 30                  27 28 29 30 31


[dywang@dywHome2 ~]$ cal -m3 9 2012
    August 2012          September 2012         October 2012
Mo Tu We Th Fr Sa Su  Mo Tu We Th Fr Sa Su  Mo Tu We Th Fr Sa Su
       1  2  3  4  5                   1  2   1  2  3  4  5  6  7
 6  7  8  9 10 11 12   3  4  5  6  7  8  9   8  9 10 11 12 13 14
13 14 15 16 17 18 19  10 11 12 13 14 15 16  15 16 17 18 19 20 21
20 21 22 23 24 25 26  17 18 19 20 21 22 23  22 23 24 25 26 27 28
```

```
152 27 28 29 30 31          24 25 26 27 28 29 30  29 30 31
```

3. bc：簡單好用的計算機。

```
[dywang@dywHome2 ~]$ bc --help
usage: bc [options] [file ...]
  -h  --help         print this usage and exit
  -i  --interactive  force interactive mode
  -l  --mathlib      use the predefine math routnes
  -q  --quiet        don't print initial banner
  -s  --standard     non-standard bc constructs are errors
  -w  --warn         warn about non-standard bc constructs
  -v  --version      print version information and exit

[dywang@mdk-dyw ~]$ man bc
## 以下只整理使用重點，完整使用手冊請自行參考 man bc。

   NUMBERS
       ...省略... The length is the total number of significant  decimal
            digits
       in a number and the scale is the total number of decimal  digits
            after
       the decimal point.  For example:
               .000001 has a length of 6 and scale of 6.
               1935.000 has a length of 7 and a scale of 3.

   VARIABLES
       ...省略... All  letters  must  be  lower case.
       ...省略... The default value of scale is 0.

   COMMENTS
       Comments in bc start with the characters /* and end with the
           characters
       */. ...省略... A single line comment starts at a # character
       and continues to the next end of the line. ...省略...

   EXPRESSIONS
       ...省略...

       - expr The result is the negation of the expression.

       ++ var The variable is incremented by one and  the  new  value
           is  the
              result of the expression.

       -- var The  variable  is  decremented  by  one and the new value
           is the
              result of the expression.

       var ++  The result of the expression is the value of the
           variable  and
              then the variable is incremented by one.
```

```
44      var -- The result of the expression is the value of the
            variable and
            then the variable is decremented by one.
46
        expr + expr
48          The result of the expression is the sum of the two
                expressions.
50      expr - expr
            The result of the expression is the difference of
                the two
52          expressions.
54      expr * expr
            The result of the expression is the product of the two
                expres-
56          sions.
58      expr / expr
            The result of the expression is the quotient of the two
                expres-
60          sions. The scale of the result is the value of the
                variable
            scale.
62
        expr % expr
64          The result of the expression is the "remainder". ...省略
                ...
66      expr ^ expr
            The result of the expression is the value of the first
                raised to
68          the second. The second expression must be an integer.
                ...省略...
70      ( expr )
            This alters the standard precedence to force the
                evaluation of
72          the expression.
74      var = expr
            The variable is assigned the value of the expression.
76
        var <op>= expr
78          This is equivalent to "var = var <op> expr" ...省略...
80      ...省略... relational operators are:
82      expr1 < expr2
            The result is 1 if expr1 is strictly less than expr2.
84
        expr1 <== expr2
86          The result is 1 if expr1 is less than or equal to expr2.
```

```
88      expr1 > expr2
              The result is 1 if expr1 is strictly greater than expr2.
90
        expr1 >= expr2
92            The result is 1 if expr1 is greater than or equal to expr2
                  .
94      expr1 == expr2
              The result is 1 if expr1 is equal to expr2.
96
        expr1 != expr2
98            The result is 1 if expr1 is not equal to expr2.
100     ...省略...  The  boolean  operators are:
102     !expr  The result is 1 if expr is 0.
104     expr && expr
              The result is 1 if both expressions are non-zero.
106
        expr || expr
108           The result is 1 if either expression is non-zero.
110     The expression precedence is as follows: (lowest to highest)
              || operator, left associative
112           && operator, left associative
              ! operator, nonassociative
114           Relational operators, left associative
              Assignment operator, right associative
116           + and - operators, left associative
              *, / and % operators, left associative
118           ^ operator, right associative
              unary - operator, nonassociative
120           ++ and -- operators, nonassociative
122     ...省略...
124     length ( expression )
              The  value  of  the length function is the number of
                  significant
126           digits in the expression.
128     read ( )
              The read function (an extension) will read  a  number
                  from  the
130           standard   input,  ...省略...
132     scale ( expression )
              The value of the scale function is the number  of  digits
                   after
134           the decimal point in the expression.
136     sqrt ( expression )
```

```
            The value of the sqrt function is the square root of the
                expres-
138         sion.  If the expression is negative, a run time error is
                gener-
            ated.
140
    STATEMENTS
142     ...省略...

144     { statement_list }
            This is the compound statement.  It allows  multiple
                statements
146         to be grouped together for execution.

148     if ( expression ) statement1 [else statement2]
            ...省略...
150         If  the  expression  is  non-zero,  statement1  is
                executed.  If
            statement2 is present and the value of the expression is
                0, then
152         statement2 is executed.

154     while ( expression ) statement
            The while statement will execute the statement while the
                expres-
156         sion is non-zero.  ...省略...

158     for ( [expression1] ; [expression2] ; [expression3] ) statement
            The for statement controls repeated execution of the
                statement.
160         ...省略...

162
        break  This statement causes a forced exit of the most recent
            enclosing
164         while statement or for statement.

166     continue
            The continue statement (an extension)  causes  the  most
                recent
168         enclosing for statement to start the next iteration.

170     halt   ...省略...

172     return Return the value 0 from a function.  (See the section  on
             func-
            tions.)
174
        return ( expression )
176         Return  the  value  of the expression from a function. ...
                省略...

178 PSEUDO STATEMENTS
    ...省略...
```

```
180
    FUNCTIONS
182     ...省略...   A function is defined as follows:
                define name ( parameters ) { newline
184                 auto_list   statement_list }
        A function call is just an expression of the form "name(
            parameters)".
186
        The  auto_list  is  an  optional list of variables that are for "
            local"
188     use.  The syntax of the auto list (if present) is "auto name,
            ...  ;".
        ...省略...  The variables are then initialized to  zero ...省略
            ...
190
        For example, the following definitions are legal.
192             define d (n) { return (2*n); }
                define d (n)
194               { return (2*n); }

196 MATH LIBRARY
        ...省略...
198
        s (x)  The sine of x, x is in radians.
200
        c (x)  The cosine of x, x is in radians.
202
        a (x)  The arctangent of x, arctangent returns radians.
204
        l (x)  The natural logarithm of x.
206
        e (x)  The exponential function of raising e to the value x.
208
        j (n,x)
210             The bessel function of integer order n of x.

212 EXAMPLES
        ...省略...
214
        範例：定義階層 x! 函數
216             define f (x) {
                  if (x <== 1) return (1);
218               return (f(x-1) * x);
                }
220
    READLINE AND LIBEDIT OPTIONS
222     ...省略...

224 DIFFERENCES
        ...省略...
226
    LIMITS
228     ...省略...
```

```
ENVIRONMENT VARIABLES
        ...省略...

DIAGNOSTICS
        ...省略...

BUGS
        ...省略...

AUTHOR
       Philip A. Nelson
       philnelson@acm.org

ACKNOWLEDGEMENTS
       The author would like to thank  Steve  Sommars  (Steve.
           Sommars@att.com)
       for  his extensive help in testing the implementation.  Many
           great sug-
       gestions were given.  This is a much better product due to his
           involve-
       ment.



                                        .

                                            bc(1)

[dywang@mdk-dyw ~]$ bc
bc 1.06
Copyright 1991-1994, 1997, 1998, 2000 Free Software Foundation, Inc.
This is free software with ABSOLUTELY NO WARRANTY.
For details type `warranty'.
(3+4)^2
49
4/3
1
scale=3
4/3
1.333

sacle=0
x=read()
100
for(i=0;i<x;i++) {
total+=(i+1)
}
total
5050

var1=10
var2=var1++
var1;var2
11
```

```
10
var2=++var1
var1;var2
12
12

define f (x) {
if (x <== 1) return (1);
return (f(x-1) * x);
}

f(4)
24
```

2.5 重要的幾個熱鍵與錯誤訊息的察看

1. [Tab] 按鍵：命令補全及檔案補齊

```
[dywang@mdk-dyw ~]$ ls[Tab]
ls          lsd         lsnetdrake  lspcidrake  lss16toppm
lsattr      lshal       lspci       lspgpot
[dywang@mdk-dyw ~]$ ls -al ~/.bash[Tab]
.bash_history  .bash_logout   .bash_profile  .bashrc        .bashrc~
```

2. [Ctrl]+c 按鍵：中斷目前程式

```
[dywang@mdk-dyw ~]$ find / -name *mdk*
```

3. [Ctrl]+d 按鍵：鍵盤輸入結束或取代 exit 直接離開文字界面。

4. 藉由螢幕上顯示的錯誤訊息解決問題。

```
[dywang@mdk-dyw ~]$ Date
bash: Date: command not found
[dywang@mdk-dyw ~]$ cal 13 2006
cal: illegal month value: use 1-12
```

Chapter 3

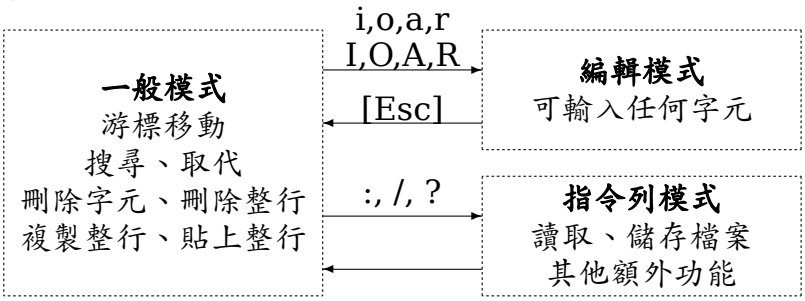
vi 編輯器

3.1 vi 與 vim

1. Linux 的系統中使用文字編輯器來編輯您的 Linux 參數設定檔。
2. Linux 與 Unix 系統中的參數檔幾乎都是 ASCII 碼的『純文字』檔。
3. vi 爲文書編輯器，vim 是 vi 的進階軟體。
4. vim 加入了支援正規表示法的搜尋架構、多檔編輯、區塊複製等功能。
5. vim 官方網站亦稱 vim 爲程式開發工具。
6. 大家來學 VIM（一個歷久彌新的編輯器）

3.2 vi 的使用

1. vi 共分爲三種模式，分別是『一般模式』、『編輯模式』與『指令列命令模式』三種。



2. 簡易範例

   (a) 使用 vi 進入一般模式

```
[dywang@dywOffice tmp]$ vim test.txt
```

(b) 按下 i 進入編輯模式，開始編輯文字。

(c) 按下 [Esc] 按鈕回到一般模式。

(d) 在一般模式中按下:wq 儲存後離開 vi。

3. 常用指令列表

一般模式	移動游標
h 或向左方向鍵	游標向左移動一個字元
j 或向下方向鍵	游標向下移動一個字元
k 或向上方向鍵	游標向上移動一個字元
l 或向右方向鍵	游標向右移動一個字元
[Ctrl]+f	螢幕『向前』移動一頁
[Ctrl]+b	螢幕『向後』移動一頁
[Ctrl]+d	螢幕『向前』移動半頁
[Ctrl]+u	螢幕『向後』移動半頁
+	游標移動到非空白字元的下一行
-	游標移動到非空白字元的上一行
n<space>	按下數字後再按空白鍵，游標會向右移動這一行的 n 個字元。例如 20<space> 則游標會向後面移動 20 個字元距離。
0	這是數字『0 』: 移動到這一行的最前面字元處。
$	移動到這一行的最後面字元處
H	游標移動到這個螢幕的最上方那一行
M	游標移動到這個螢幕的中央那一行
L	游標移動到這個螢幕的最下方那一行
G	游標移動到這個檔案的最後一行
nG	游標移動到這個檔案的第 n 行。例如 20G 則會移動到檔案的第 20 行 (可配合:set nu)
n<Enter>	游標向下移動 n 行

一般模式	搜尋與取代
/word	向游標之後尋找一個字串名稱爲 word 的字串。
?word	向游標之前尋找一個字串名稱爲 word 的字串。
n	重複前一個搜尋。
N	反向進行前一個搜尋動作。
:n1,n2s/word1/word2/g	在第 n1 與 n2 行之間尋找 word1 這個字串，並將該字串取代爲 word2 。

:1,$s/word1/word2/g	從第一行到最後一行尋找 word1 字串，並將該字串取代爲 word2 。
:1,$s/word1/word2/gc	從第一行到最後一行尋找 word1 字串，並將該字串取代爲 word2 且在取代前顯示提示字元給使用者確認是否需要取代。

一般模式	**刪除、複製與貼上**
x, X	x 爲向後刪除一個字元，X 爲向前刪除一個字元
nx	向後刪除 n 個字元
dd	刪除游標所在的那一整行
ndd	刪除游標所在的向下 n 行，例如 20dd 則是刪除 20 行
d1G	刪除游標所在到第一行的所有資料
dG	刪除游標所在到最後一行的所有資料
dw	刪除一個字 (delete word)。不適用於中文。
yy	複製游標所在的那一行（常用）
nyy	複製游標所在的向下 n 行，例如 20yy 則是複製 20 行
y1G	複製游標所在行到第一行的所有資料
yG	複製游標所在行到最後一行的所有資料
p, P	p 爲複製的資料在游標下一行貼上，P 則爲貼在游標上一行。
J	將游標所在行與下一行的資料結合成同一行
u	復原前一個動作。
[Ctrl]+r	反復原前一個動作。
.	重做上一個動作。

進入編輯模式	
i, I	插入：i(I) 在目前的游標所在處 (所在行第一個非空白字元處) 插入輸入之文字；
a, A	增加：a(A) 由目前游標所在的下一個 (最後一個) 字元處開始輸入：
o, O	插入新的一行：o(O) 從游標所在的下 (上) 一行插入新的一行；
r, R	取代：r 會取代游標所在的那一個字元；R 會一直取代游標所在的文字，直到按下 ESC 爲止；
Esc	退出編輯模式，回到一般模式中。

指令列命令模式	
:w	將編輯的資料寫入硬碟檔案中
:w!	若檔案屬性爲『唯讀』時，強制寫入該檔案
:q	離開 vi
:q!	若曾修改過檔案，又不想儲存，使用! 爲強制離開不儲存檔案。

:wq	儲存後離開，若爲:wq! 則爲強制儲存後離開
:x	與 wq 相同
:e!	將檔案還原到最原始的狀態。
ZZ	若檔案沒有更動，則不儲存離開，若檔案已經經過更動，則儲存後離開。
:w [filename]	將編輯的資料儲存成另一個檔案（類似另存新檔）
:r [filename]	在編輯的資料中，讀入另一個檔案的資料。亦即將『filename』這個檔案內容加到游標所在行後面。
:set nu	顯示行號，設定之後，會在每一行的字首顯示該行的行號
:set nonu	與 set nu 相反，爲取消行號。
:n1,n2 w [filename]	將 n1 到 n2 的內容儲存成 filename 這個檔案。
:! command	暫時離開 vi 到指令列模式下執行 command 的顯示結果。例如 [:! ls /home]

4. vi 編輯中回復與暫存檔

   (a) 如果編輯檔案爲 /home/csie/vitest.txt，則同時會有一個暫存檔 /home/csie/.vitest.txt.swp 產生，其爲隱藏檔。

   (b) 結束 vi 編輯時暫存檔也會刪除。

   (c) 若不知因素導致 vi 程式中斷，則在下次編輯該檔案時會出現是否回復的訊息。

3.3 vim 的額外功能

1. 如果使用 vi，在畫面右下角有目前游標所在行列號碼，則 vi 已被 vim 取代了。

```
[dywang@dywOffice tmp]$ ll /bin/vi*
lrwxrwxrwx  1 root root 20 Oct 17 13:11 /bin/vi -> /etc/alternatives/vi*
lrwxrwxrwx  1 root root 21 Oct 17 13:11 /bin/vim -> /etc/alternatives/
    vim*
[dywang@dywOffice tmp]$ ll /etc/alternatives/vi*
lrwxrwxrwx  1 root root 21 Oct 17 13:11 /etc/alternatives/vi ->
/usr/bin/vim-enhanced*
lrwxrwxrwx  1 root root 21 Oct 17 13:11 /etc/alternatives/vim ->
/usr/bin/vim-enhanced*
```

2. vim 具有顏色顯示的功能，並且還支援許多的程式語法。

3. vim /etc/man.config 出現訊息說明：

```
"/etc/man.config" [readonly] 150L, 4900C                    1,1
   Top
```

(a) 編輯檔案名稱爲 /etc/man.config；

(b) 檔案爲唯讀檔；

(c) 檔案共有 150 行，4900 字元；

(d) 目前游標所在位置爲第一行, 第一列；

(e) 目前頁面在最前頁。

4. 區塊選擇

	區塊選擇的按鍵意義
v	字元選擇，會將游標經過的地方反白選擇。
V	行選擇，會將游標經過的反白選擇。
[Ctrl]+v	區塊選擇，可以用長方形的方式選擇資料。
y	將反白的地方複製起來。
d	將反白的地方刪除掉。

5. 多檔案編輯

(a) vi 內使用:r filename 可將檔案 filename 的內容在游標處插入。

(b) 可於 vim 後接多個檔案來同時開啓多個檔案，例如：vim filename1 filename2 filename3。其相關按鍵有：

	多檔案編輯的按鍵
:n	編輯下一個檔案。
:N	編輯上一個檔案。
:files	列出目前開啓的所有檔案。

6. 多視窗功能

(a) 在指令列模式輸入:sp filename。

(b) 如果省略 filename 則兩視窗爲同一檔案。

	多視窗下的按鍵功能
:sp	開啓同一檔案於新視窗。
:sp filename	開啓檔案 filename 於新視窗。
[Ctrl]+wj	游標移動到下方的視窗。按法爲：先按下 [Ctrl] 不放，再下 w 後放開所有的按鍵，然後再按下 j。
[Ctrl]+wk	游標移動到上方的視窗。按法爲同上。
[Ctrl]+wq	結束下方視窗，與 [Ctrl]+w 移動到下方視窗後，再按下:q 離開相同。

7. vim 環境設定

   (a) 個人動作記錄檔案：~/.viminfo。例如：編輯同一檔案時，游標會在上次退出時的位置。

   (b) 整體 vim 的設定值放在 /etc/vimrc。

   (c) 若要更改 vim 設定，建議自行建立 ~/.vimrc。

	vim 的環境設定參數
:set nu	設定行號。
:set nonu	取消定行號。
:set hlsearch	將搜尋的字串反白。
:set autoindent	自動縮排。
:set noautoindent	不自動縮排。
:set backup	自動儲存備份。備份檔名爲 filename~。
:set rule	顯示右下角的狀態說明。
:set showmode	顯示 –insert– 等字眼在左下角的狀態列。
:set backspace=(012)	2 利用 backspace 例退鍵除任意字元；0 或 1 僅可刪除剛剛輸入的字元。
:set all	顯示目前所有的環境參數設定值。
:syntax (off|on)	是否依據程式相關語法顯示不同顏色。

3.4 vi 實機練習題

3.4.1 練習一

1. 在自己的家目錄建立一個新的目錄 zzz。

2. 進入目錄 zzz。

3. 下載檔案 viex1.txt
4. 將 viex1.txt 重新命名爲 vi1.txt
5. 使用 vi 或 vim 編輯 vi1.txt，執行以下動作：
   (a) 先到第 n 行，n 爲學號除 37 取餘數 +10。
   (b) 複製 3 行。
   (c) 再到第 20 行，往下貼上。
   (d) 再到最後一行，往下產生一行空白行方式進入”插入”模式進行編輯。
   (e) 加入文字’ABCDEabcde’。
   (f) 換行再加入’1234567890’。
   (g) 搜尋字串”MANPATH”，並將其取代爲”manpath”。
   (h) 存檔後退出。

3.4.2 練習二

1. 在自己的家目錄建立一個新的目錄 zzz。
2. 進入目錄 zzz。
3. 下載檔案 viex2.txt
4. 將 viex2.txt 重新命名爲 vi2.txt
5. 使用 vi 或 vim 編輯 vi2.txt，執行以下動作：
   (a) 先到第 n 行，n 爲學號除 27 取餘數 +8。
   (b) 刪除 3 行。
   (c) 再到第 30 行，往上貼上。
   (d) 再到第一行，往上產生一行空白行方式進入”插入”模式進行編輯。
   (e) 加入文字’ABCDEabcde’。
   (f) 換行再加入’1234567890’。
   (g) 從第 30 行至最後一行搜尋字串「MANPATH」，並將其取代爲「manpath」。
   (h) 存檔後退出。

3.4.3 練習三

1. 在自己的家目錄建立一個新的目錄 zzz。
2. 進入目錄 zzz。
3. 下載檔案 viex3.txt
4. 將 viex3.txt 重新命名爲 vi3.txt
5. 使用 vi 或 vim 編輯 vi3.txt，執行以下動作：
   (a) 先到第 n 行，n 爲學號除 44 取餘數 +11。
   (b) 將下一行合併至這一行
   (c) 再到第 20 行，刪除這一行到第一行的資料。
   (d) 再到最後一行，往下貼上。
   (e) 再到第 13 行，在這行的最後加入文字'ABCDEabcde'。
   (f) 換行再加入'1234567890'。
   (g) 從第 30 行至第 55 行搜尋字串"man"，並將其取代爲"MAN"。
   (h) 存檔後退出。

Part II

系統管理 KVM8

Chapter 4

IPv4 網路設定

4.1 前言

1. 網路 (網路介面) 卡由系統 daemon NetworkManager 管理，包含：
   (a) device: 網路卡。
   (b) connection: 設定網卡。
   (c) 每一個 connection 都有一個名字及 UUID 做爲識別。
   (d) 目錄 /etc/sysconfig/network-scripts 下 ifcfg-name 爲網卡 name 的設定檔，可以使用命令 nmcli 產生或編輯網卡連線 (connection)。
   (e) /etc/sysconfig/network-scripts/ifcfg-name 網卡設定檔，還可以跟 v6 版本一樣，直接以 vim 手動編輯。

4.2 nmcli 命令

1. 查看 device 狀態，除了 lo 外只有一個網路介面 enp1s0。

```
[root@999999 ~]# nmcli device
DEVICE  TYPE      STATE      CONNECTION
enp1s0  ethernet  connected  enp1s0
lo      loopback  unmanaged  --
```

2. 查詢網路連線狀態，網路連線 enp1s0 使用網路界面 enp1s0。

```
[root@999999 ~]# nmcli connection
NAME    UUID                                  TYPE      DEVICE
enp1s0  2a4b4018-d130-405b-8647-6f3a1744b525  ethernet  enp1s0
```

4.3 修改網卡設定

1. 不必刪除設定，直接使用 modify 選項變更設定。設定 ipv4 位址 192.168.122.8、遮罩 255.255.255.0、GATEWAY 192.168.122.1 及 DNS 192.168.122.1，最後一定要設定 ipv4.method 爲 manual，將網路連線設定爲手動，也就是自行設定 IP，不是由 DHCP 自動取得，否則原 DHCP 取得的 IP 還會存在。

```
[root@999999 ~]# nmcli connection modify enp1s0 \
ipv4.addresses 192.168.122.8/24 \
ipv4.gateway 192.168.122.1 \
ipv4.dns 192.168.122.1 \
ipv4.method manual
```

2. 將網路連線 enp1s0 啓動。

```
[root@999999 ~]# nmcli connection up enp1s0
Connection successfully activated (D-Bus active path: /org/freedesktop/
    NetworkManager/ActiveConnection/5)
```

3. 查看網卡設定，看手動設定是否正確？

```
[root@999999 ~]# nmcli connection show enp1s0 | grep -iE ipv?4
ipv4.method:                            manual
ipv4.dns:                               192.168.122.1
ipv4.dns-search:                        --
ipv4.dns-options:                       --
ipv4.dns-priority:                      0
ipv4.addresses:                         192.168.122.8/24
ipv4.gateway:                           192.168.122.1
ipv4.routes:                            --
ipv4.route-metric:                      -1
ipv4.route-table:                       0 (unspec)
ipv4.routing-rules:                     --
ipv4.ignore-auto-routes:                no
ipv4.ignore-auto-dns:                   no
ipv4.dhcp-client-id:                    --
ipv4.dhcp-iaid:                         --
ipv4.dhcp-timeout:                      0 (default)
ipv4.dhcp-send-hostname:                yes
ipv4.dhcp-hostname:                     --
ipv4.dhcp-fqdn:                         --
ipv4.dhcp-hostname-flags:               0x0 (none)
ipv4.never-default:                     no
ipv4.may-fail:                          yes
ipv4.required-timeout:                  -1 (default)
```

```
ipv4.dad-timeout:                       -1 (default)
ipv4.dhcp-vendor-class-identifier:      --
ipv4.dhcp-reject-servers:               --
IP4.ADDRESS[1]:                         192.168.122.8/24
IP4.GATEWAY:                            192.168.122.1
IP4.ROUTE[1]:                           dst = 192.168.122.0/24, nh =
    0.0.0.0, mt = 100
IP4.ROUTE[2]:                           dst = 0.0.0.0/0, nh =
    192.168.122.1, mt = 100
IP4.DNS[1]:                             192.168.122.1
```

4.4 * 刪除重新設定

1. 刪除網路連線 System enp1s0。

```
[root@999999 ~]# nmcli connection delete enp1s0
Connection 'enp1s0' (f03ca672-c3f6-433a-b336-832566e18e76) successfully
    deleted.
```

2. 如果是用 ssh 連線，這時會斷線，必須回到虛擬機終端機重新「新增」網路連線。

```
[root@999999 ~]# nmcli connection add \
con-name enp1s0 ifname enp1s0 type ethernet \
ipv4.addresses 192.168.122.8/24 \
ipv4.gateway 192.168.122.1 \
ipv4.dns 192.168.122.1 \
ipv4.method manual
Connection 'enp1s0' (a7b3ec85-0fa4-4bf3-b775-da87d43deacc) successfully
    added.
```

3. 查看網路連線，有 enp1s0 及 virbr0。

```
[root@999999 ~]# nmcli connection
NAME    UUID                                  TYPE      DEVICE
enp1s0  2a4b4018-d130-405b-8647-6f3a1744b525  ethernet  enp1s0
```

4. 將網路連線 enp1s0 啓動。

```
[root@999999 ~]# nmcli connection up enp1s0
```

```
Connection successfully activated (D-Bus active path: /org/freedesktop/
    NetworkManager/ActiveConnection/5)
```

5. 查看新增的網路連線設定 enp1s0 是否正確？是否生效？

```
[root@999999 ~]# nmcli connection show enp1s0 | grep -iE ipv?4
ipv4.method:                            manual
ipv4.dns:                               192.168.122.1
ipv4.dns-search:                        --
ipv4.dns-options:                       --
ipv4.dns-priority:                      0
ipv4.addresses:                         192.168.122.8/24
ipv4.gateway:                           192.168.122.1
ipv4.routes:                            --
ipv4.route-metric:                      -1
ipv4.route-table:                       0 (unspec)
ipv4.routing-rules:                     --
ipv4.ignore-auto-routes:                no
ipv4.ignore-auto-dns:                   no
ipv4.dhcp-client-id:                    --
ipv4.dhcp-iaid:                         --
ipv4.dhcp-timeout:                      0 (default)
ipv4.dhcp-send-hostname:                yes
ipv4.dhcp-hostname:                     --
ipv4.dhcp-fqdn:                         --
ipv4.dhcp-hostname-flags:               0x0 (none)
ipv4.never-default:                     no
ipv4.may-fail:                          yes
ipv4.required-timeout:                  -1 (default)
ipv4.dad-timeout:                       -1 (default)
ipv4.dhcp-vendor-class-identifier:      --
ipv4.dhcp-reject-servers:               --
IP4.ADDRESS[1]:                         192.168.122.8/24
IP4.GATEWAY:                            192.168.122.1
IP4.ROUTE[1]:                           dst = 192.168.122.0/24, nh =
    0.0.0.0, mt = 100
IP4.ROUTE[2]:                           dst = 0.0.0.0/0, nh =
    192.168.122.1, mt = 100
IP4.DNS[1]:                             192.168.122.1
```

4.5 SSH 設定與連線

1. sshd 允許 root 登入的參數 PermitRootLogin 設定：

   (a) yes：允許

   (b) no：不允許

(c) prohibit-password (without-password)：不允許以輸入密碼的方式登入

(d) forced-commands-only：只允許在登入命令用公鑰授權登入。

2. PermitRootLogin 預設是 prohibit-password，修改成 yes。

```
[root@999999 ~]# vim /etc/ssh/sshd_config
[root@999999 ~]# grep PermitRoot /etc/ssh/sshd_config
#PermitRootLogin prohibit-password
PermitRootLogin yes
# the setting of "PermitRootLogin without-password".
```

3. 重新啓動 sshd 服務。

```
[root@999999 ~]# systemctl restart sshd.service
```

4. 從 host 主機以 ssh 登入 kvm8。

```
[dywang@deyu ~]$ ssh root@kvm8.deyu.wang
The authenticity of host 'kvm8.deyu.wang (192.168.122.8)' can't be
    established.
ECDSA key fingerprint is SHA256:
    WaRiVLOS0JYkmNagaoEUu3MoCDEzqxougcjB0uJ99QQ.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'kvm8.deyu.wang' (ECDSA) to the list of known
     hosts.
root@kvm8.deyu.wang's password:
Last login: Mon Aug 29 15:35:12 2022 from 192.168.122.1
```

4.6 主機名稱設定

1. 遠端連線 192.168.122.8。

```
[dywang@deyu ~]$ ssh root@192.168.122.8
root@k192.168.122.8's password:
Last login: Mon Sep 19 10:51:26 2016 from deyu.wang
[root@999999 ~]#
```

2. 查看主機名稱爲 999999。

```
[root@999999 ~]# hostname
999999
```

3. 設定主機名稱爲 kvm8.deyu.wang。

```
[root@999999 ~]# hostnamectl set-hostname kvm8.deyu.wang
```

4. 查看主機名稱狀態。

```
[root@999999 ~]# hostname
kvm8.deyu.wang
```

5. hostnamectl 查看狀態。

```
[root@999999 ~]# hostnamectl status
   Static hostname: kvm8.deyu.wang
         Icon name: computer-vm
           Chassis: vm
        Machine ID: 25c85e60dd87ff6e80d795b6a828e0ec
           Boot ID: 4682a4e4a5014ed7b14c92f870b8882d
    Virtualization: kvm
  Operating System:
      ://www.centos.org/CentOS Linux 8 KVM8rchitecture: x86_64
\end{myPE OS Name: cpe:/o:centos:centos:8
            Kernel: Linux 4.18.0-80.el8.x86_64
      Architecture: x86-64
```

4.7 實機練習

1. 查詢目前乙太網路

   (a) 查詢網卡名稱，將網卡名稱寫到 /root/netif。

   (b) 查詢網卡連線名稱，將連線名稱寫到 /root/netcon。

2. 設定目前乙太網路

   (a) ipv4 位址 192.168.122.8

   (b) 遮罩 255.255.255.0

(c) GATEWAY 192.168.122.1

(d) DNS 192.168.122.1

3. 設定主機名稱爲 kvm8.deyu.wang

Chapter 5

YUM 套件管理

5.1 前言

1. 何謂 RPM？
   (a) RPM 全名是『RedHat Package Manager』簡稱爲 RPM。
   (b) 由 Red Hat 公司發展出來，由於 RPM 使用方便，所以成了目前最熱門的套件管理程式。
   (c) RPM 是以資料庫記錄的方式來將所需要的套件安裝到 Linux 主機的一套管理程式。
   (d) RPM 將要安裝的套件先編譯過，並且打包好，安裝時 RPM 會先依照套件裡的紀錄資料查詢相依屬性套件是否滿足，若滿足則予以安裝，若不滿足則不予安裝。
2. RPM 套件的屬性相依
   (a) RPM 打包套件檔案時，會同時加入套件的訊息。例如：
      i. 版本、
      ii. 打包套件者、
      iii. 相依屬性的套件、
      iv. 套件的功能說明、
      v. 套件的所有檔案與目錄紀錄。
   (b) 當要安裝某個以 RPM 型態提供的套件時，如果資料庫顯示其相依套件不存在，則會顯示錯誤訊息。
   (c) 屬性相依的克服方式
      i. 手動下載並安裝好所有相依套件。

ii. 利用 yum 安裝 rpm 套件。

3. YUM

(a) RHEL/AlmaLinux 都提供套件庫 (repository) 供 yum 下載安裝、升級、移除。

(b) 在安裝好 RHEL/AlmaLinux 系統後，都會將 YUM 的套件庫設定到同版本的下載連結點。

(c) 利用 yum 安裝 rpm 套件時，yum 會在設定的套件庫尋找未安裝的相依套件加以安裝。

(d) 也可自行建立套件庫，設定爲 yum 的套件庫，yum 就可以升級，安裝，移除套件庫中的套件。

5.2 yum 套件庫設定

1. 設定 yum 套件庫 (repository) 設定檔，必須在 /etc/yum.repo.d/ 目錄下，建立.repo 檔案，檔案設定方式可查詢 dnf.conf 手冊。

```
[root@kvm8 ~]# man dnf.conf
```

2. 撰寫 /etc/yum.repos.d/dywang.repo，建立 AlmaLinux 9 BaseOS 及 AppStream 兩類套件。

```
[root@kvm8 ~]# vi /etc/yum.repos.d/dywang.repo
[root@kvm8 ~]# cat /etc/yum.repos.d/dywang.repo
[AppStream]
name=App Stream
baseurl=http://dywang.csie.cyut.edu.tw/alma9/AppStream
gpgcheck=0

[BaseOS]
name=Base OS
baseurl=http://dywang.csie.cyut.edu.tw/alma9/BaseOS
gpgcheck=0
```

3. .repo 檔參數說明：

(a) [base]：套件庫的名稱，中刮號一定要存在，名稱可以自訂但不可重複。

(b) name：說明這個套件庫的意義。

(c) mirrorlist=：列出這個套件庫可以使用的映射站台，如果不想使用，可以註解到這行；

(d) baseurl=：指定固定套件庫的實際網址。

(e) enable=1：預設啓動套件庫，可以不設定。如果不想啓動可以使用 enable=0。

(f) gpgcheck=1：指定是否需要查閱 RPM 檔案內的數位簽章。

(g) gpgkey=：數位簽章的公鑰檔所在位置。

4. 列出 yum 使用的所有套件庫

```
[root@kvm8 ~]# dnf repolist
repo id                                     repo name
AppStream                                   App Stream
BaseOs                                      Base OS
```

5. 試安裝 vsftpd？如果出現 Is this ok [y//N]: 表示設定無誤可以取得套件進行安裝。y 安裝、N 不安裝，輸入 n 進行安裝。

```
[root@kvm8 ~]# dnf install vsftpd
Last metadata expiration check: 0:00:41 ago on Tue 04 Oct 2022 11:13:08
    AM CST.
Dependencies resolved.
================================================================================

 Package         Architecture    Version               Repository
             Size
================================================================================

Installing:
 vsftpd          x86_64          3.0.3-49.el9          AppStream
             158 k

Transaction Summary
================================================================================

Install  1 Package

Total download size: 158 k
Installed size: 348 k
Is this ok [y/N]: n
```

6. cache 套件庫資料刪除

```
[root@kvm8 ~]# dnf clean [packages|headers|all]
```

```
選項與參數：
 packages：將已下載的軟體檔案刪除
 headers ：將下載的軟體檔頭刪除
 all     ：將所有套件庫資料都刪除

#刪除已下載過的所有套件庫的相關資料 (含軟體本身與清單)
[root@kvm8 ~]# dnf clean all
```

5.3 *yum 查詢功能

1. 查詢選項：

```
[root@kvm8 ~]# dnf [options] [command] [package ...]
[option]：主要的搜尋選項有：
[查詢工作項目] [相關參數]：這方面的參數有：
  search  ：搜尋某個套件名稱或是描述 (description) 的重要關鍵字；
  list    ：列出目前 yum 所管理的所有的套件名稱與版本，有點類似 rpm -qa
      ;
  info    ：列出套件的相關訊息，類似 rpm -qai 的執行結果；
  provides or whatprovides：搜尋包含某檔案的套件，類似 rpm -qf 的功能。
```

2. 查詢名稱或是描述中包含關鍵字 cjk 的套件

```
[root@kvm8 ~]# dnf search cjk
Last metadata expiration check: 0:16:27 ago on Tue 04 Oct 2022 11:13:08
    AM CST.
========================== Name & Summary Matched: cjk
    ==========================
google-noto-cjk-fonts-common.noarch : Common files for Noto CJK fonts
google-noto-sans-cjk-ttc-fonts.noarch : Sans OTC font files for
                                      : google-noto-cjk-fonts
google-noto-serif-cjk-ttc-fonts.noarch : Serif OTC font files for
                                       : google-noto-cjk-fonts
texlive-cjk.noarch : CJK language support
texlive-xecjk.noarch : Support for CJK documents in XeLaTeX
============================== Summary Matched: cjk
    ==============================
bitmap-fangsongti-fonts.noarch : Selected CJK bitmap fonts for Anaconda
texlive-garuda-c90.noarch : TeX support (from CJK) for the garuda font
texlive-norasi-c90.noarch : TeX support (from CJK) for the norasi font
    in
                          : thailatex
```

3. 查看套件 google-noto-cjk-fonts-common 的相關訊息

```
[root@kvm8 ~]# dnf info google-noto-cjk-fonts-common.noarch
Last metadata expiration check: 0:18:58 ago on Tue 04 Oct 2022 11:13:08
    AM CST.
Installed Packages
Name         : google-noto-cjk-fonts-common
Version      : 20201206
Release      : 4.el9
Architecture : noarch
Size         : 32 k
Source       : google-noto-cjk-fonts-20201206-4.el9.src.rpm
Repository   : @System
From repo    : AppStream
Summary      : Common files for Noto CJK fonts
URL          : https://github.com/googlei18n/noto-cjk
License      : OFL
Description  :
             : Noto CJK fonts, supporting Simplified Chinese,
               Traditional
             : Chinese, Japanese, and Korean. The supported scripts are
               Han,
             : Hiragana, Katakana, Hangul, and Bopomofo. Latin, Greek,
               Cyrllic,
             : and various symbols are also supported for compatibility
               with CJK
             : standards.
             :
             :
             : Common files for Google Noto CJK fonts.
```

4. 列出所有 evince 開頭的套件

```
[root@cs ~]# dnf list evince*
Last metadata expiration check: 21:06:59 ago on Mon 03 Oct 2022 02:25:56
     PM CST.
Installed Packages
evince.x86_64                              40.4-4.el9
    @AppStream
evince-libs.x86_64                         40.4-4.el9
    @AppStream
evince-nautilus.x86_64                     40.4-4.el9
    @AppStream
evince-previewer.x86_64                    40.4-4.el9
    @AppStream
evince-thumbnailer.x86_64                  40.4-4.el9
    @AppStream
Available Packages
evince-libs.i686                           40.4-4.el9
    AppStream
```

5. 列出包含檔案 bin/evince 的所有套件，例如/usr/bin/evince。

```
[root@kvm8 ~]# dnf provides *bin/evince
Last metadata expiration check: 0:21:21 ago on Tue 04 Oct 2022 11:13:08
    AM CST.
evince-40.4-4.el9.x86_64 : Document viewer
Repo        : AppStream
Matched from:
Other       : *bin/evince
```

5.4 *yum 安裝、升級與移除

1. 安裝/升級選項

```
[root@www ~]# yum command [packages]
command:
  install  :安裝
  reinstall:重新安裝
  update   :升級，若要整個系統都升級，後不接套件
  remove   :移除
```

2. 安裝

```
[root@kvm8 ~]# dnf install bzip2
Last metadata expiration check: 0:01:07 ago on Tue 04 Oct 2022 11:53:21
    AM CST.
Dependencies resolved.
================================================================================
 Package         Architecture     Version                 Repository
            Size
================================================================================

Installing:
 bzip2           x86_64           1.0.8-8.el9             BaseOS
                52 k

Transaction Summary
================================================================================

Install  1 Package

Total download size: 52 k
Installed size: 94 k
Is this ok [y/N]: y
Downloading Packages:
```

```
bzip2-1.0.8-8.el9.x86_64.rpm                    721 kB/s |  52 kB
    00:00
--------------------------------------------------------------------------------

Total                                           379 kB/s |  52 kB
    00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                        1/1
  Installing       : bzip2-1.0.8-8.el9.x86_64                               1/1
  Running scriptlet: bzip2-1.0.8-8.el9.x86_64                               1/1
  Verifying        : bzip2-1.0.8-8.el9.x86_64                               1/1

Installed:
  bzip2-1.0.8-8.el9.x86_64

Complete!
```

3. 移除功能：yum [remove] 軟體

```
[root@kvm8 ~]# dnf remove bzip2
Dependencies resolved.
================================================================================
 Package         Architecture     Version                Repository
             Size
================================================================================

Removing:
 bzip2           x86_64           1.0.8-8.el9            @BaseOS
                94 k

Transaction Summary
================================================================================

Remove  1 Package

Freed space: 94 k
Is this ok [y/N]: y
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
```

```
20   Preparing        :                                                 1/1
     Erasing          : bzip2-1.0.8-8.el9.x86_64                        1/1
22   Running scriptlet: bzip2-1.0.8-8.el9.x86_64                        1/1
     Verifying        : bzip2-1.0.8-8.el9.x86_64                        1/1
24
Removed:
26   bzip2-1.0.8-8.el9.x86_64

28 Complete!
```

5.5 實機練習

1. 建立 yum 安裝套件倉庫來源
   (a) http://dywang.csie.cyut.edu.tw/alma9/BaseOS
   (b) http://dywang.csie.cyut.edu.tw/alma9/AppStream
2. 取消 RPM 檔案數位簽章檢查
3. 試著安裝套件 vsftpd

Chapter 6

systemctl 系統服務控制

6.1 前言

1. v6 PID=1 是 init。
2. v7 PID=1 是 systemd，其優點有：
   (a) 平行處理能力，增加開機速度。
   (b) 自動服務相依管理，避免過久的 timeout，例如：網路不通時，就不會像 v6 還試圖連上網路。

6.2 systemctl 命令

1. systemctl 命令用來管理各種不同型態的 systemd 物件。

```
[root@kvm4 ~]# systemctl -t help
Available unit types:
service
socket
target
device
mount
automount
snapshot
timer
swap
path
slice
scope
```

2. systemctl 輔助說明

```
[root@kvm4 ~]# systemctl --help
systemctl [OPTIONS...] {COMMAND} ...

Query or send control commands to the systemd manager.

  -h --help           Show this help
     --version        Show package version
  -t --type=TYPE      List only units of a particular type
     --state=STATE    List only units with particular LOAD or SUB or
         ACTIVE stat
  -p --property=NAME  Show only properties by this name
  -a --all            Show all loaded units/properties, including dead/
      empty
                      ones. To list all units installed on the system,
                          use
                      the 'list-unit-files' command instead.
     --reverse        Show reverse dependencies with 'list-dependencies'
  -l --full           Don't ellipsize unit names on output
     --fail           When queueing a new job, fail if conflicting jobs
         are
                      pending
     --irreversible   When queueing a new job, make sure it cannot be
         implicitly
                      cancelled
     --ignore-dependencies
                      When queueing a new job, ignore all its
                          dependencies
     --show-types     When showing sockets, explicitly show their type
  -i --ignore-inhibitors
                      When shutting down or sleeping, ignore inhibitors
     --kill-who=WHO   Who to send signal to
  -s --signal=SIGNAL  Which signal to send
  -H --host=[USER@]HOST
                      Show information for remote host
  -P --privileged     Acquire privileges before execution
  -q --quiet          Suppress output
     --no-block       Do not wait until operation finished
     --no-wall        Don't send wall message before halt/power-off/
         reboot
     --no-reload      When enabling/disabling unit files, don't reload
         daemon
                      configuration
     --no-legend      Do not print a legend (column headers and hints)
     --no-pager       Do not pipe output into a pager
     --no-ask-password
                      Do not ask for system passwords
     --system         Connect to system manager
     --user           Connect to user service manager
     --global         Enable/disable unit files globally
     --runtime        Enable unit files only temporarily until next
         reboot
```

```
  -f --force          When enabling unit files, override existing
      symlinks
44                    When shutting down, execute action immediately
     --root=PATH      Enable unit files in the specified root directory
46  -n --lines=INTEGER  Number of journal entries to show
  -o --output=STRING  Change journal output mode (short, short-monotonic
      ,
48                    verbose, export, json, json-pretty, json-sse, cat)
     --plain          Print unit dependencies as a list instead of a
        tree
50
Unit Commands:
52  list-units                      List loaded units
  list-sockets                      List loaded sockets ordered by address
54  start [NAME...]                 Start (activate) one or more units
  stop [NAME...]                    Stop (deactivate) one or more units
56  reload [NAME...]                Reload one or more units
  restart [NAME...]                 Start or restart one or more units
58  try-restart [NAME...]           Restart one or more units if active
  reload-or-restart [NAME...]       Reload one or more units if possible,
60                                  otherwise start or restart
  reload-or-try-restart [NAME...] Reload one or more units if possible,
62                                  otherwise restart if active
  isolate [NAME]                    Start one unit and stop all others
64  kill [NAME...]                  Send signal to processes of a unit
  is-active [NAME...]               Check whether units are active
66  is-failed [NAME...]             Check whether units are failed
  status [NAME...|PID...]           Show runtime status of one or more
      units
68  show [NAME...|JOB...]           Show properties of one or more
                                    units/jobs or the manager
70  set-property [NAME] [ASSIGNMENT...]
                                    Sets one or more properties of a unit
72  help [NAME...|PID...]           Show manual for one or more units
  reset-failed [NAME...]            Reset failed state for all, one, or
      more
74                                  units
  list-dependencies [NAME]          Recursively show units which are
      required
76                                  or wanted by this unit or by which
                                        this
                                    unit is required or wanted
78
Unit File Commands:
80  list-unit-files                 List installed unit files
  enable [NAME...]                  Enable one or more unit files
82  disable [NAME...]               Disable one or more unit files
  reenable [NAME...]                Reenable one or more unit files
84  preset [NAME...]                Enable/disable one or more unit files
                                    based on preset configuration
86  is-enabled [NAME...]            Check whether unit files are enabled

88  mask [NAME...]                  Mask one or more units
  unmask [NAME...]                  Unmask one or more units
```

```
90    link [PATH...]                     Link one or more units files into
                                           the search path
92    get-default                        Get the name of the default target
      set-default NAME                   Set the default target
94
    Job Commands:
96    list-jobs                          List jobs
      cancel [JOB...]                    Cancel all, one, or more jobs
98
    Snapshot Commands:
100   snapshot [NAME]                    Create a snapshot
      delete [NAME...]                   Remove one or more snapshots
102
    Environment Commands:
104   show-environment                   Dump environment
      set-environment [NAME=VALUE...] Set one or more environment variables
106   unset-environment [NAME...]        Unset one or more environment
          variables
108
    Manager Lifecycle Commands:
      daemon-reload                      Reload systemd manager configuration
110   daemon-reload                      Reload systemd manager configuration
      daemon-reexec                      Reexecute systemd manager
112
    System Commands:
114   default                            Enter system default mode
      rescue                             Enter system rescue mode
116   emergency                          Enter system emergency mode
      halt                               Shut down and halt the system
118   poweroff                           Shut down and power-off the system
      reboot                             Shut down and reboot the system
120   kexec                              Shut down and reboot the system with
          kexec
      exit                               Request user instance exit
122   switch-root [ROOT] [INIT]          Change to a different root file system
      suspend                            Suspend the system
124   hibernate                          Hibernate the system
      hybrid-sleep                       Hibernate and suspend the system
```

6.3 服務啓動

1. 查看服務 firewalld 狀態，目前爲 active，正在執行中。

```
1   [root@kvm4 ~]# systemctl status firewalld.service
      firewalld.service - firewalld - dynamic firewall daemon
3        Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled;
             vendor>
```

```
     Active: active (running) since Fri 2022-09-23 09:45:32 CST; 1 week
         6 days >
       Docs: man:firewalld(1)
   Main PID: 640 (firewalld)
      Tasks: 2 (limit: 11111)
     Memory: 42.5M
        CPU: 4.410s
     CGroup: /system.slice/firewalld.service
             640 /usr/bin/python3 -s /usr/sbin/firewalld --nofork --
               nopid

Sep 23 09:45:16 999999 systemd[1]: Starting firewalld - dynamic firewall
     daemon>
Sep 23 09:45:32 999999 systemd[1]: Started firewalld - dynamic firewall
    daemon.
[root@kvm4 ~]# systemctl status firewalld.service
  firewalld.service - firewalld - dynamic firewall daemon
     Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled;
          vendor>
     Active: inactive (dead) since Thu 2022-10-06 10:13:55 CST; 2s ago
       Docs: man:firewalld(1)
    Process: 640 ExecStart=/usr/sbin/firewalld --nofork --nopid
        $FIREWALLD_ARGS>
   Main PID: 640 (code=exited, status=0/SUCCESS)
        CPU: 5.179s

Sep 23 09:45:16 999999 systemd[1]: Starting firewalld - dynamic firewall
     daemon>
Sep 23 09:45:32 999999 systemd[1]: Started firewalld - dynamic firewall
    daemon.
Oct 06 10:13:54 kvm4.deyu.wang systemd[1]: Stopping firewalld - dynamic
    firewal>
Oct 06 10:13:55 kvm4.deyu.wang systemd[1]: firewalld.service:
    Deactivated succe>
Oct 06 10:13:55 kvm4.deyu.wang systemd[1]: Stopped firewalld - dynamic
    firewall>
Oct 06 10:13:55 kvm4.deyu.wang systemd[1]: firewalld.service: Consumed
    5.179s C>
```

2. 關閉服務 firewalld。

```
[root@kvm4 ~]# systemctl stop firewalld.service
```

3. 查看服務 firewalld 狀態，目前爲 inactive，關閉中。

```
[root@kvm4 ~]# systemctl status firewalld.service
```

```
  firewalld.service - firewalld - dynamic firewall daemon
     Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled;
          vendor>
     Active: inactive (dead) since Thu 2022-10-06 10:13:55 CST; 2s ago
       Docs: man:firewalld(1)
    Process: 640 ExecStart=/usr/sbin/firewalld --nofork --nopid
        $FIREWALLD_ARGS>
   Main PID: 640 (code=exited, status=0/SUCCESS)
        CPU: 5.179s

Sep 23 09:45:16 999999 systemd[1]: Starting firewalld - dynamic firewall
     daemon>
Sep 23 09:45:32 999999 systemd[1]: Started firewalld - dynamic firewall
    daemon.
Oct 06 10:13:54 kvm4.deyu.wang systemd[1]: Stopping firewalld - dynamic
    firewal>
Oct 06 10:13:55 kvm4.deyu.wang systemd[1]: firewalld.service:
    Deactivated succe>
Oct 06 10:13:55 kvm4.deyu.wang systemd[1]: Stopped firewalld - dynamic
    firewall>
Oct 06 10:13:55 kvm4.deyu.wang systemd[1]: firewalld.service: Consumed
    5.179s C>
```

4. 查看服務 firewalld 目前是否啓動？結果爲 inactive。

```
[root@kvm4 ~]# systemctl is-active firewalld.service
inactive
```

5. 啓動服務 firewalld。

```
[root@kvm4 ~]# systemctl start firewalld.service
```

6. 再查看服務 firewalld 目前是否啓動？結果爲 active。

```
[root@kvm4 ~]# systemctl is-active firewalld.service
active
```

7. 也可以用 restart 選項，重新啓動服務 firewalld。

```
[root@kvm4 ~]# systemctl restart firewalld.service
```

8. 再查看服務 firewalld 目前是否啓動？結果爲 active。

```
1 [root@kvm4 ~]# systemctl is-active firewalld.service
  active
```

6.4 開機自動啓動服務

1. 查看服務 firewalld 開機是否自動啓動？結果爲 enabled。

```
2 [root@kvm4 ~]# systemctl is-enabled firewalld.service
  enabled
```

2. 關閉服務 firewalld 開機自動啓動。

```
2 [root@kvm4 ~]# systemctl disable firewalld.service
  Removed /etc/systemd/system/multi-user.target.wants/firewalld.service.
  Removed /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service.
```

3. 再查看服務 firewalld 開機是否自動啓動？結果爲 disabled。

```
1 [root@kvm4 ~]# systemctl is-enabled firewalld.service
  disabled
```

4. 再啓動服務 firewalld 開機自動啓動。

```
2 [root@kvm4 ~]# systemctl enable firewalld.service
  Created symlink /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.
      service →
```

```
/usr/lib/systemd/system/firewalld.service.
Created symlink /etc/systemd/system/multi-user.target.wants/firewalld.
    service →
/usr/lib/systemd/system/firewalld.service.
```

5. 再查看服務 firewalld 開機是否自動啓動？結果爲 enabled。

```
[root@kvm4 ~]# systemctl is-enabled firewalld.service
enabled
```

6.5 * 永久關閉服務

1. mask 服務 firewalld，使這個服務無論開機或手動都無法啓動。

```
[root@kvm4 ~]# systemctl mask firewalld.service
Created symlink /etc/systemd/system/firewalld.service → /dev/null.
```

2. 啓動服務 firewalld。

```
[root@kvm4 ~]# systemctl enable firewalld.service
Failed to enable unit: Unit file /etc/systemd/system/firewalld.service
    is masked.
```

3. 查看服務 firewalld 是否啓動，結果爲 masked。

```
[root@kvm4 ~]# systemctl is-enabled firewalld.service
masked
```

4. unmask 服務 firewalld，解除這個服務的鎖定。

```
[root@kvm4 ~]# systemctl unmask firewalld.service
Removed /etc/systemd/system/firewalld.service.
```

5. 再查看服務 firewalld 是否啓動，結果爲 enabled。

```
[root@kvm4 ~]# systemctl is-enabled firewalld.service
enabled
```

6.6 * 失效的服務

1. 查看失效的服務。

```
[root@kvm4 ~]# systemctl --failed --type service
  UNIT          LOAD   ACTIVE SUB    DESCRIPTION
  httpd.service loaded failed failed The Apache HTTP Server

LOAD   = Reflects whether the unit definition was properly loaded.
ACTIVE = The high-level unit activation state, i.e. generalization of
   SUB.
SUB    = The low-level unit activation state, values depend on unit type
   .
1 loaded units listed.
```

6.7 實機練習

1. 使用 systemctl 控制服務

   (a) 查看 chronyd 服務是否啓動 (active)？結果導向到 /root/systemd-active。

   (b) 查看 chronyd 服務是否開機啓動 (enable)？結果導向到 /root/systemd-enabled。

   (c) 把關閉 chronyd 服務的完整指令寫到 /root/systemd-stop。

   (d) 把啓動 chronyd 服務的完整指令寫到 /root/systemd-start。

   (e) 把重新啓動 chronyd 服務的完整指令寫到 /root/systemd-restart。

(f) 把查詢 chronyd 服務狀態的完整指令寫到 /root/systemd-status。

Chapter 7

SELinux

7.1 前言

1. 自主式存取控制 (Discretionary Access Control, DAC)
   (a) 傳統的檔案權限與帳號關係：依據程序的擁有者與檔案資源的 rwx 權限來決定有無存取的能力。
   (b) 若該程序屬 root 權限，則其可在系統上進任何資源存取。
   (c) 若某個目錄權限爲 777，則任何帳號的程序都可任意存取及寫入。
2. 系統出現問題
   (a) 內部員工的資源誤用遠高於外部攻擊。
   (b) 員工資源誤用：系統管理員爲了自己方便，將防火牆完全關閉或某個檔案目錄的權限設定爲 777。
   (c) 爲控管員工資源誤用問題，美國國家安全局 (NSA) 開發安全增強式 Linux（SELinux, Security-Enhanced Linux）模組，並整合到 Linux 核心。
   (d) SELinux 是一種強制存取控制（mandatory access control, MAC）的實現。
3. 強制存取控制
   (a) 針對特定的程序與特定的檔案資源進行權限的控管。
   (b) WWW server 啓動的程序爲 httpd，selinux 預設僅能在/var/www 目錄下存取檔案。
4. 開啓 SELinux 出現的現象：
   (a) httpd 配置正常，防火牆也放行，但用户就是無法訪問？

(b) vsftpd 配置正常，防火牆也放行，但客户端訪問提示無法轉換家目錄？

(c) 關閉 SELinux 後一切正常，因此很多人對 SELinux 有了極大的偏見，認爲 SELinux 只有帶來麻煩，於是選擇在安裝系統後或出現問題時關閉 SELinux。

(d) 實際上，SELinux 在很多時候提供關鍵的保障，所以還是要開啓 SELinux，出現無法訪問等問題時，以修改 SELinux 安全上下文、布林值... 等方式來解決。

7.2 SELinux Contexts

1. Linux 執行程序或檔案會以 SELinux 上下文 (Contexts) 做標記，內容包含：SELinux user, role, type, level，以這些資訊來控制存取決定。

2. ls -Z 查詢檔案的 SELinux contexts，列出四個欄位 `user:role:type:level`，分別以冒號隔開。

```
[root@kvm7 ~]# ls -Z anaconda-ks.cfg
-rw-------. root root system_u:object_r:admin_home_t:s0 anaconda-ks.cfg
```

(a) user：使用者身份，以確認 SELinux 策略，Linux user 帳號會被對應到 SELinux user，執行以下命令可以列出兩者的對應，其中 `__default__` 表示登入的用户，一般是使用 useradd 增加的帳號；`system_u` 表示系統程序，由執行使用者可以確認身份。

```
[root@kvm7 ~]# semanage login -l
Login Name           SELinux User         MLS/MCS Range
    Service
__default__          unconfined_u         s0-s0:c0.c1023       *
root                 unconfined_u         s0-s0:c0.c1023       *
system_u             system_u             s0-s0:c0.c1023       *
```

(b) role：SELinux 以 Role-Based Access Control (RBAC) 控制存取，角色 (role) 決定是否可存取 domain，而每個 user 可以有多個角色，但同一時間只能有一個角色。

(c) type：指定資料型別，規則中定義何種程序型別訪問何種檔案。

(d) level：Multi-Level Security (MLS) and Multi-Category Security (MCS)，僅在 MLS 策略下才有用。

3. SELinux Contexts 四個欄位說明：

項目	用户 (user)	角色 (role)	類型 (type)	級別 (level)
程序	system_u 爲系統服務程序，受到管制。unconfined_u 爲不管制程序，通常都是用户自己開啓的，如 bash。	system_r 爲系統服務程序，受到管制。unconfined_r 爲不管制程序，通常都是用户自己開啓的，如 bash。	在 targeted 策略下，只有兩者的類型對應上（不代表相同），程序才能存取檔案。	MLS 安全級別，僅在 MLS 策略下才有意義。
檔案	system_u 爲系統服務創建的檔案。unconfined_u 爲用户自己創建的檔案。	均爲 object_r，表示一個檔案。	在 targeted 模式下，只有兩者的類型對應上了（不代表相同），程序才能存取檔案。	MLS 安全級別，僅在 MLS 模式下才有意義。

7.3 啓動、關閉與觀察

1. SELinux 開機啓動模式

```
[root@kvm7 ~]# vim /etc/selinux/config
[root@kvm7 ~]# cat /etc/selinux/config

# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
SELINUX=enforcing
# SELINUXTYPE= can take one of three two values:
#     targeted - Targeted processes are protected,
#     minimum - Modification of targeted policy. Only selected processes
    are protected.
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

2. 關察目前 SELinux 狀態

```
[root@kvm7 ~]# sestatus
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
```

```
SELinux root directory:          /etc/selinux
Loaded policy name:              targeted
Current mode:                    permissive
Mode from config file:           enforcing
Policy MLS status:               enabled
Policy deny_unknown status:      allowed
Max kernel policy version:       28
```

3. 改變 SELinux 模式

```
[root@kvm7 ~]# getenforce
Permissive
[root@kvm7 ~]# setenforce 1
[root@kvm7 ~]# getenforce
Enforcing
```

7.4 SELinux 基本用法

1. coreutils 的工具如 ps、ls 可以增加 Z 選項的方式取得 SELinux 的資訊。

```
[root@kvm7 ~]# ls -Z anaconda-ks.cfg
-rw-------. root root system_u:object_r:admin_home_t:s0 anaconda-ks.cfg
[root@kvm7 ~]# ps auxZ | grep apache
system_u:system_r:httpd_t:s0  apache  15694  0.0  0.2 215320  3040 ?  S
    Oct14   0:00 /usr/sbin/httpd -DFOREGROUND
```

2. chcon 修改檔案或目錄的 SELinux contexts。

```
[root@kvm7 ~]# chcon --help
Usage: chcon [OPTION]... CONTEXT FILE...
  or:  chcon [OPTION]... [-u USER] [-r ROLE] [-l RANGE] [-t TYPE] FILE
      ...
  -u, --user=USER        set user USER in the target security context
  -r, --role=ROLE        set role ROLE in the target security context
  -t, --type=TYPE        set type TYPE in the target security context
  -l, --range=RANGE      set range RANGE in the target security context
  -R, --recursive        operate on files and directories recursively
  -v, --verbose          output a diagnostic for every file processed
```

3. chcon 範例

```
[root@kvm7 ~]# ls -Z /etc/resolv.conf
-rw-r--r--. root root system_u:object_r:net_conf_t:s0  /etc/resolv.conf
[root@kvm7 ~]# cp /etc/resolv.conf .
[root@kvm7 ~]# ls -Z resolv.conf
-rw-r--r--. root root unconfined_u:object_r:admin_home_t:s0 resolv.conf
[root@kvm7 ~]# chcon -t net_conf_t resolv.conf
[root@kvm7 ~]# ls -Z resolv.conf
-rw-r--r--. root root unconfined_u:object_r:net_conf_t:s0 resolv.conf
```

4. restorecon 恢復檔案或目錄的 SELinux contexts 到預設值。

```
[root@kvm7 ~]# restorecon
usage:  restorecon [-iFnprRv0] [-e excludedir] pathname...
usage:  restorecon [-iFnprRv0] [-e excludedir] -f filename
```

5. restorecon 範例

```
[root@kvm7 ~]# restorecon resolv.conf
[root@kvm7 ~]# ls -Z resolv.conf
-rw-r--r--. root root unconfined_u:object_r:admin_home_t:s0 resolv.conf
```

6. getsebool 查詢 SELinux 布林值型規則的狀態。

```
[root@kvm7 ~]# getsebool
usage:  getsebool -a or getsebool boolean...
```

7. getsebool 範例

```
[root@kvm7 ~]# getsebool -a | grep ssh
fenced_can_ssh --> off
selinuxuser_use_ssh_chroot --> off
sftpd_write_ssh_home --> off
ssh_chroot_rw_homedirs --> off
ssh_keysign --> off
ssh_sysadm_login --> off
```

8. setsebool 開關 SELinux 布林值型規則。

```
[root@kvm7 ~]# setsebool
Usage:  setsebool [ -NPV ] boolean value | bool1=val1 bool2=val2...
```

9. setsebool 範例，沒加 -P 選項只是現在生效，重開機後會恢復到原設定值。

```
[root@kvm7 ~]# setsebool sftpd_write_ssh_home on
[root@kvm7 ~]# getsebool -a | grep ssh
fenced_can_ssh --> off
selinuxuser_use_ssh_chroot --> off
sftpd_write_ssh_home --> on
ssh_chroot_rw_homedirs --> off
ssh_keysign --> off
ssh_sysadm_login --> off
```

10. semanage 設定 SELinux 策略的某些元素，例如：檔案 context、程序埠號等。

```
[root@kvm7 ~]# semanage
usage: semanage [-h]

               {...,port,..,fcontext,...}
               ...
semanage is used to configure certain elements of SELinux policy with-
    out
requiring modification to or recompilation from policy source.

    port              Manage network port type definitions
    fcontext          Manage file context mapping definitions
    boolean           Manage booleans to selectively enable
        functionality
```

11. semanage 範例一：添加目錄的 SELinux context。

```
[root@kvm7 ~]# mkdir /usr/share/www
[root@kvm7 ~]# ls -Z /usr/share/www
[root@kvm7 ~]# ls -Zd /usr/share/www
drwxr-xr-x. root root unconfined_u:object_r:usr_t:s0   /usr/share/www
[root@kvm7 ~]# semanage fcontext -a -t httpd_sys_content_t "/usr/share/
    www(/.*)?"
[root@kvm7 ~]# ls -Zd /usr/share/www
drwxr-xr-x. root root unconfined_u:object_r:usr_t:s0   /usr/share/www
[root@kvm7 ~]# restorecon -Rv /usr/share/www
restorecon reset /usr/share/www context unconfined_u:object_r:usr_t:s0
->unconfined_u:object_r:httpd_sys_content_t:s0
[root@kvm7 ~]# ls -Zd /usr/share/www
drwxr-xr-x. root root unconfined_u:object_r:httpd_sys_content_t:s0 /usr/
    share/www
```

12. semanage 範例二：添加 httpd 程序允許訪問的 port。

```
[root@kvm7 ~]# semanage port -l | grep ^http_port
http_port_t      tcp      80, 81, 443, 488, 8008, 8009, 8443, 9000
[root@kvm7 ~]# semanage port -a -t http_port_t -p tcp 8989
[root@kvm7 ~]# semanage port -l | grep ^http_port
http_port_t      tcp      8989, 80, 81, 443, 488, 8008, 8009, 8443, 9000
```

13. semanage 範例三：刪除 httpd 程序允許訪問的 port。

```
[root@kvm7 html]# semanage port -d -t http_port_t -p tcp 8989
[root@kvm7 html]# semanage port -l | grep ^http_port
http_port_t       tcp      80, 81, 443, 488, 8008, 8009, 8443, 9000
```

7.5 Httpd Port

1. 查看 httpd 服務的狀態，enabled (開機啓動)，Active:failed (啓動失敗)。從狀態說明，查到原因是無法啓動在 82 port。

```
[root@kvm8 ~]# systemctl status httpd.service
  httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled;
vendor preset: disabled)
   Active: failed (Result: exit-code) since Wed 2019-11-06 21:01:25 CST;
1min 53s ago
     Docs: man:httpd.service(8)
  Process: 1961 ExecStart=/usr/sbin/httpd $OPTIONS -DFOREGROUND
(code=exited, status=1/FAILURE)
 Main PID: 1961 (code=exited, status=1/FAILURE)
   Status: "Reading configuration..."

Nov 06 21:01:25 kvm8.deyu.wang systemd[1]: Starting The Apache HTTP
    Server...
Nov 06 21:01:25 kvm8.deyu.wang httpd[1961]: (13)Permission denied:
    AH00072:
make_sock: could not bind to address [::]:82
Nov 06 21:01:25 kvm8.deyu.wang httpd[1961]: (13)Permission denied:
    AH00072:
make_sock: could not bind to address 0.0.0.0:82
Nov 06 21:01:25 kvm8.deyu.wang httpd[1961]: no listening sockets
    available,
shutting down
Nov 06 21:01:25 kvm8.deyu.wang httpd[1961]: AH00015: Unable to open logs
Nov 06 21:01:25 kvm8.deyu.wang systemd[1]: httpd.service: Main process
    exited,
code=exited, status=1/FAILURE
Nov 06 21:01:25 kvm8.deyu.wang systemd[1]: httpd.service: Failed with
    result
```

```
'exit-code'.
Nov 06 21:01:25 kvm8.deyu.wang systemd[1]: Failed to start The Apache HTTP
Server.
```

2. 查詢 http_port_t，82 port 不是 httpd 程序允許訪問的 port。

```
[root@kvm8 ~]# semanage port -l | grep ^http_port
http_port_t                    tcp      80, 81, 443, 488, 8008, 8009, 8443, 9000
```

3. 新增 82 爲 httpd 服務允許訪問的埠號。

```
[root@kvm8 ~]# semanage port -a -t http_port_t -p tcp 82
```

4. 再查詢 http_port_t，82 port 已經是 httpd 程序允許訪問的 port。

```
[root@kvm8 ~]# semanage port -l | grep ^http_port
http_port_t                    tcp      82, 80, 81, 443, 488, 8008, 8009, 8443,
    9000
```

5. 再重新啓動 httpd 服務，成功啓動服務。

```
[root@kvm7 html]# systemctl restart httpd.service
```

6. 再查看 httpd 服務的狀態，enabled (開機啓動)，Active: active (running)。

```
 httpd.service - The Apache HTTP Server
  Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled;
      vendor preset: disabled)
  Active: active (running) since Wed 2019-11-06 21:15:06 CST; 5s ago
    Docs: man:httpd.service(8)
Main PID: 15569 (httpd)
  Status: "Started, listening on: port 82"
   Tasks: 213 (limit: 11512)
  Memory: 30.9M
  CGroup: /system.slice/httpd.service
          15569 /usr/sbin/httpd -DFOREGROUND
          15570 /usr/sbin/httpd -DFOREGROUND
          15571 /usr/sbin/httpd -DFOREGROUND
          15572 /usr/sbin/httpd -DFOREGROUND
          15573 /usr/sbin/httpd -DFOREGROUND
```

```
15
Nov 06 21:15:06 kvm8.deyu.wang systemd[1]: Starting The Apache HTTP
    Server...
17 Nov 06 21:15:06 kvm8.deyu.wang httpd[15569]: Server configured,
    listening on: port 82
Nov 06 21:15:06 kvm8.deyu.wang systemd[1]: Started The Apache HTTP
    Server.
```

7.6 Httpd SELinux Context

1. 查看網頁伺服器根目錄下有 file1, file2 兩個檔案，權限是任何人都可讀取。

```
[root@kvm8 ~]# ll /var/www/html/
2 total 8
-rw-r--r--. 1 root root 10 Nov  6 20:55 file1
4 -rw-r--r--. 1 root root 10 Nov  6 21:18 file2
```

2. 使用 curl 訪問網頁 file1，回應沒有權限存取。

```
[root@kvm8 ~]# curl http://127.0.0.1:82/file1
2 <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
4 <title>403 Forbidden</title>
</head><body>
6 <h1>Forbidden</h1>
<p>You don't have permission to access /file1
8 on this server.<br />
</p>
10 </body></html>
```

3. 使用 curl 訪問網頁 file2，成功回應檔案內容”web test2”。

```
```

4. 查看 SELinux Contexts，file1 是 admin_home_t，file2 是 httpd_sys_contect。

```
[root@kvm8 ~]# ls -Z /var/www/html/
2       system_u:object_r:admin_home_t:s0 file1
system_u:object_r:httpd_sys_content_t:s0 file2
```

5. 將 /var/www/html 目錄下所有檔案，恢復預設的 selinux contexts。

```
[root@kvm8 ~]# restorecon -Rv /var/www/html/
```

6. 再查看 SELinux Contexts，file1 還是 admin_home_t。

```
[root@kvm8 ~]# ls -Z /var/www/html/
      system_u:object_r:admin_home_t:s0 file1
system_u:object_r:httpd_sys_content_t:s0 file2
```

7. 查看 /var/www/html/file1 的 SELinux Contexts type 被設成 admin_home_t。

```
[root@kvm8 ~]# semanage fcontext --list | grep file1
/var/www/html/file1        all files        system_u:object_r:
    admin_home_t:s0
```

8. 客製化的 SELinux Contexts type 設定在 file_contexts.local，此檔案是 libmanage 自動產生，不要直接編輯檔案。

```
[root@kvm8 ~]# cat /etc/selinux/targeted/contexts/files/file_contexts.
    local
# This file is auto-generated by libsemanage
# Do not edit directly.

/var/www/html/file1    system_u:object_r:admin_home_t:s0
```

9. semanage -d 刪除 /var/www/html/file1 自訂的 admin_home_t。

```
[root@kvm8 ~]# semanage fcontext -d -t admin_home_t /var/www/html/file1
```

10. semanage 列出 SELinux Contexts type，已查不到 /var/www/html/file1。

```
[root@kvm8 ~]# semanage fcontext --list | grep file1
```

11. 再查客製化的 SELinux Contexts type 設定檔 file_contexts.local，檔案內已無任何設定。

```
[root@kvm8 ~]# cat /etc/selinux/targeted/contexts/files/file_contexts.
    local
# This file is auto-generated by libsemanage
# Do not edit directly.
```

12. 將 /var/www/html 目錄下所有檔案，恢復預設的 selinux contexts。

```
[root@kvm8 ~]# restorecon -Rv /var/www/html/
Relabeled /var/www/html/file1 from system_u:object_r:admin_home_t:s0 to
    system_u:object_r:httpd_sys_content_t:s0
```

13. 查看 SELinux Contexts，file1 及 file2 都是 httpd_sys_contect。

```
[root@kvm8 ~]# ls -Z /var/www/html/
system_u:object_r:httpd_sys_content_t:s0 file1
system_u:object_r:httpd_sys_content_t:s0 file2
```

14. 使用 curl 訪問網頁 file1 及 file2，都成功回應檔案內容。

```
[root@kvm8 ~]# curl http://127.0.0.1:82/file1
web test1
[root@kvm8 ~]# curl http://127.0.0.1:82/file2
web test2
```

7.7 實機練習

1. 網頁服務

   (a) 查看網頁服務是否啓動？若無法啓動查看爲何無法啓動？

   (b) 依錯誤訊息解決

   (c) 重啓網頁 httpd 服務必須成功啓動。

2. 網頁存取

   (a) 試著連線網頁根目錄下的所有檔案，看看是否都可連線存取？

   (b) 若無法連線存取，請解決問題。

Chapter 8

* 正規表示法

8.1 前言

- 什麼是正規表示法
  1. 處理字串的方法，以行爲單位進行字串的處理行爲。
  2. 透過一些特殊符號的輔助，讓使用者輕易的達到搜尋、取代，限定某特定字串的處理程序。
  3. 延伸的正規表示法：可以作群組的字串『或 (or)』及『且 (and)』的處理。
  4. 正規表示法與萬用字元不一樣，不可混淆。
- 正規表示法用途
  1. 系統管理員的可以透過『正規表示法』的功能，將登錄的資訊進行處理，僅取出『有問題』的資訊來進行分析。
  2. 一般支援正規表示法的軟體可利用其進行字串的處理，例如：『郵件伺服器』以『字串』的比對來過濾郵件。
  3. 軟體中輸入欄位的格式限制，例如：只能爲數字。
- vi, grep, awk ,sed 等工具皆支援正規表示法。

8.2 基礎正規表示法

- 重要特殊字元 (characters)

```
RE字符          意義與範例
^word    待搜尋的字串(word)在行首。
         #範例：grep -n '^#' re.txt
```

```
         搜尋行首爲 # 開始的那一行。
word$    待搜尋的字串(word)在行尾。
         #範例：grep -n '!$' re.txt
         將行尾爲 ! 的那一行列印出來。
  .      代表『任意一個』字符，一定是一個任意字符。
         #範例：grep -n 'e.e' re.txt
         搜尋的字串可以是 (eve) (eae) (eee) (e e)， 但不能僅有 (ee)。
         亦即 e 與 e 中間『一定』僅有一個字元，而空白字元也是字元。
  \      跳脫字符，將特殊符號的特殊意義去除。
         #範例：grep -n \' re.txt
         搜尋含有單引號 ' 的那一行。
  *      重複零個或多個的前一個 RE 字符
         #範例：grep -n 'ess*' re.txt
         找出含有 (es) (ess) (esss) 等等的字串。
\{n,m\}  連續 n 到 m 個的『前一個 RE 字符』
         若爲 \{n\} 則是連續 n 個的前一個 RE 字符，
         若是 \{n,\} 則是連續 n 個以上的前一個 RE 字符。
         #範例：grep -n 'go\{2,3\}g' re.txt
         在 g 與 g 之間有 2 個到 3 個的 o 存在的字串，亦即 (goog)(gooog)
  [ ]    在 [ ] 當中『謹代表一個待搜尋的字元』
         #範例：grep -n 'g[ld]' re.txt
         搜尋含有 (gl) 或 (gd) 的那一行
         #範例：grep -n '[0-9]' re.txt
         搜尋含有任意數字的那一行。
         在字元集合 [ ] 中的減號 - 是代表兩個字元之間的所有連續字元。
         [^]：^ 在 [ ] 內時， 代表的意義是『反向選擇』
         #範例：grep -n 'oo[^t]' re.txt
         搜尋的字串可以是 (oog) (ood) 但不能是 (oot)。
```

- 以 grep 擷取字串

1. 編輯 re.txt 純文字檔

```
[root@test root]# vi re.txt
"Open Source" is a good mechanism to develop programs.
apple is my favorite food.
Football game is not use feet only.
this dress doesn't fit me.
However, this dress is about $ 3183 dollars.
GNU is free air not free beer.
Her hair is very beauty.
I can't finish the test.
Oh! The soup taste good.
motorcycle is cheap than car.
This window is clear.
the symbol '*' is represented as start.
Oh! My god!
The gd software is a library for drafting programs.
You are the best is mean you are the no. 1.
The world  is the same with "glad".
I like dog.
```

```
19 google is the best tools for search keyword.
   goooooogle yes!
21 go! go! Let's go.
   # I am csie
```

2. 搜尋特定字串：從檔案 re.txt 中取得特定字串 the

```
[root@test root]# grep -n 'the' re.txt
2 8:I can't finish the test.
  12:the symbol '*' is represented as start.
4 15:You are the best is mean you are the no. 1.
  16:The world  is the same with "glad".
6 18:google is the best tools for search keyword.
```

3. 該行沒有字串 the 時，才顯示在螢幕上：

```
[root@test root]# grep -vn 'the' re.txt
2 # 螢幕上出現的行列爲除了 8,12,15,16,18 五行之外的其他行列。
```

4. 取得不論大小寫的字串 the：

```
[root@test root]# grep -in 'the' re.txt
2 8:I can't finish the test.
  9:Oh! The soup taste good.
4 12:the symbol '*' is represented as start.
  14:The gd software is a library for drafting programs.
6 15:You are the best is mean you are the no. 1.
  16:The world  is the same with "glad".
8 18:google is the best tools for search keyword.
```

5. 利用 [ ] 來搜尋集合字元

```
[root@test root]# grep -n 't[ae]st' re.txt
2 8:I can't finish the test.
  9:Oh! The soup taste good.
4 # [ ] 裡面不論有幾個字元，都謹代表某『一個』字元。
```

6. 搜尋有 oo 的字元：

```
[root@test root]# grep -n 'oo' re.txt
2 1:"Open Source" is a good mechanism to develop programs.
  2:apple is my favorite food.
4 3:Football game is not use feet only.
```

```
9:Oh! The soup taste good.
18:google is the best tools for search keyword.
19:gooooooogle yes!
```

7. 不想要 oo 前面有 g ，可以利用在集合字元的反向選擇 [^]

```
[root@test root]# grep -n '[^g]oo' re.txt
2:apple is my favorite food.
3:Football game is not use feet only.
18:google is the best tools for search keyword.
19:gooooooogle yes!
```

8. oo 前面不想要有小寫字元：

```
[root@test root]# grep -n '[^a-z]oo' re.txt
3:Football game is not use feet only.
```

9. 要求字串是數字與英文：[a-zA-Z0-9]

```
[root@test root]# grep -n '[0-9]' re.txt
5:However, this dress is about $ 3183 dollars.
15:You are the best is mean you are the no. 1.
```

10. 行首與行尾字元 ^ $：

```
[root@test root]# grep -n '^the' re.txt
12:the symbol '*' is represented as start.

[root@test root]# grep -n '^[a-z]' re.txt
2:apple is my favorite food.
4:this dress doesn't fit me.
10:motorcycle is cheap than car.
12:the symbol '*' is represented as start.
18:google is the best tools for search keyword.
19:gooooooogle yes!
```

11. 不要開頭是英文字母：

```
[root@test root]# grep -n '^[^a-zA-Z]' re.txt
1:"Open Source" is a good mechanism to develop programs.
20:# I am csie
```

12. 行尾結束爲小數點 (.) 的那一行：

```
[root@test root]# grep -n '\.$' re.txt
1:"Open Source" is a good mechanism to develop programs.
2:apple is my favorite food.
3:Football game is not use feet only.
4:this dress doesn't fit me.
10:motorcycle is cheap than car.
11:This window is clear.
12:the symbol '*' is represented as start.
15:You are the best is mean you are the no. 1.
16:The world  is the same with "glad".
17:I like dog.
18:google is the best tools for search keyword.
# 小數點具有其他意義，必須要使用跳脫字元(\)來加以解除其特殊意義。
# 第 5~9 行最後面也是 .？
```

13. Windows 的 notepad 會主動加上 ^M 作爲斷行的判斷。

```
[root@test root]# cat -A re.txt
However, this dress is about $ 3183 dollars.^M$
```

14. 找出哪一行是『空白行』?

```
[root@test root]# grep -n '^$' re.txt
21:

[root@test root]# cat /etc/syslog.conf
[root@test root]# grep -v '^$' /etc/syslog.conf | grep -v '^#'
```

15. 任意一個字元. 與重複字元 *

```
#『.』代表『絕對有一個任意字元』
[root@test root]# grep -n 'g..d' re.txt
1:"Open Source" is a good mechanism to develop programs.
9:Oh! The soup taste good.
16:The world  is the same with "glad".
#  『*』代表的是『重複 0 個或多個前面的 RE 字符』
[root@test root]# grep -n 'ooo*' re.txt
1:"Open Source" is a good mechanism to develop programs.
2:apple is my favorite food.
3:Football game is not use feet only.
9:Oh! The soup taste good.
18:google is the best tools for search keyword.
19:gooooooogle yes!
```

16. 字串開頭與結尾都是 g，但是兩個 g 之間僅能存在至少一個 o

```
# 亦即 gog, goog, gooog.... 等。
[root@test root]# grep -n 'goo*g' re.txt
18:google is the best tools for search keyword.
19:goooooogle yes!
```

17. 找出 g 開頭與 g 結尾的字串，當中的字元可有可無。(錯誤找法)

```
[root@test root]# grep -n 'g*g' re.txt
1:"Open Source" is a good mechanism to develop programs.
3:Football game is not use feet only.
9:Oh! The soup taste good.
13:Oh!  My god!
14:The gd software is a library for drafting programs.
16:The world  is the same with "glad".
17:I like dog.
18:google is the best tools for search keyword.
19:goooooogle yes!
# g*g 的內容是 g, gg, ggg, gggg。
```

18. 找出 g 開頭與 g 結尾的字串，當中的字元可有可無。

```
[root@test root]# grep -n 'g.*g' re.txt
1:"Open Source" is a good mechanism to develop programs.
14:The gd software is a library for drafting programs.
18:google is the best tools for search keyword.
19:goooooogle yes!
```

19. 要找出『任意數字』的行列：

```
[root@test root]# grep -n '[0-9][0-9]*' re.txt
5:However, this dress is about $ 3183 dollars.
15:You are the best is mean you are the no. 1.
# 使用 grep -n '[0-9]' re.txt 也可以得到相同的結果。
```

20. 限定連續 RE 字符範圍 {}：找到兩個 o 的字串。

```
[root@test root]# grep -n 'o\{2\}' re.txt
1:"Open Source" is a good mechanism to develop programs.
2:apple is my favorite food.
3:Football game is not use feet only.
9:Oh! The soup taste good.
18:google is the best tools for search keyword.
19:goooooogle yes!
```

21. 找出 g 後面接 2 到 5 個 o ，然後再接一個 g 的字串：

```
[root@test root]# grep -n 'go\{2,5\}g' re.txt
18:google is the best tools for search keyword.
```

22. 找出 2 個 o 以上的 goooo....g？除了可以是 gooo*g ，也可以是：

```
[root@test root]# grep -n 'go\{2,\}g' re.txt
18:google is the best tools for search keyword.
19:gooooooogle yes!
```

23. 找出 2 個 o 以上的 goooo....g，當找到一筆即停止。

```
[root@test root]# grep -m 1 'go\{2,\}g' re.txt
18:google is the best tools for search keyword.
19:gooooooogle yes!
```

- 『正規表示法的特殊字元』與『萬用字元』之差異：

字元或代表意義	萬用字元	正規表示法的特殊字元
*	0 至無限多個字元	重複 0 到多個的前一個 RE 字符
反向選擇	[!range]	[^range]

```
例題：不支援正規表示法的 ls 工具中
『ls -l * 』：代表列出任意檔名的檔案；
『ls -l a* 』：代表列出以 a 爲開頭的任何檔名的檔案。
在正規表示法中要找到含有以 a 爲開頭的檔案，必須搭配支援正規表示法的工具
   grep：
ls | grep -n '^a.*'
```

- 例題：以正規表示法顯示行號方式找出目錄 /etc 以下符合下列條件之檔案。

1. 包含 boot 或 root 字串的檔案。
2. 包含 b 開頭 t 結尾之字串的檔案。
3. b 開頭 t 結尾且中間 1 個 o 以上之字串的檔案。
4. 包含 b 開頭 t 結尾且中間 2 至 4 個 o 之字串的檔案。
5. 包含行首爲 root 之字串的檔案。

6. 包含行尾爲 root 之字串的檔案。

- 例題：想要查出來檔案中含有! 與 > 的字行 (! 在正規表示法中並不是特殊字元)：

```
grep -n '[!>]' re.txt
```

8.3 延伸正規表示法

- grep 支援的是基礎型的正規表示法，延伸正規表示法 egrep 是 grep -E 的命令別名。

- 延伸型正規表示法之特殊符號

```
RE 字符          意義與範例
  +       重複『一個或一個以上』的前一個 RE 字符
          #範例：egrep -n 'go+d' re.txt
          搜尋 (god) (good) (goood)... 等等的字串。
  ?       『零個或一個』的前一個 RE 字符
          #範例：egrep -n 'go?d' re.txt
          搜尋 (gd) (god) 這兩個字串。
  |       用或( or )的方式找出數個字串
          #範例：egrep -n 'gd|good|dog' re.txt
          搜尋 gd 、 good 或 dog 這三個字串。
 ( )      找出『群組』字串或作爲『多個重複群組』的判別
          #範例：egrep -n 'g(la|oo)d'
          搜尋 (glad) 或 (good) 這兩個字串。
          #範例：egrep -n 'A(xyz)+C' re.txt
          找開頭是 A 結尾是 C ，中間有一個以上的 "xyz" 字串。
```

- 例題：去除空白行與行首爲 # 的行列

```
#基礎型的正規表示法需要使用到管線命令來搜尋兩次
grep -v '^$' re.txt | grep -v '^#'
#支援延伸型正規表示法可以簡化爲：
egrep -v '^$|^#' re.txt
```

- 例題：以延伸型正規表示法顯示行號方式，找出目錄 /etc 以下符合下列條件之檔案。

1. 包含 boot 或 root 字串的檔案。

2. 包含 b 開頭 t 結尾且中間 0 個或 1 個 o 之字串的檔案。
3. b 開頭 t 結尾且中間 1 個 o 以上之字串的檔案。

8.4 實機練習題

8.4.1 練習一

1. 在自己的家目錄建立一個新的目錄 zzz。
2. 進入目錄 zzz。
3. 下載檔案 re1.txt
4. 使用 grep 對 re1.txt 執行以下搜尋動作，不需要列印行號，結果分別導向到檔案 result1.txt 至 result7.txt，不要做任何的更動。
   (a) you 大小寫不分。
   (b) tast 或 test。
   (c) oo 前面不是 g，也不是 t。
   (d) 剛好四個阿拉伯數字。
   (e) 兩個以上阿拉伯數字。
   (f) 行首是大寫英文字母。
   (g) 行尾不是'.' 句點。

8.4.2 練習二

1. 在自己的家目錄建立一個新的目錄 zzz。
2. 進入目錄 zzz。
3. 下載檔案 re2.txt
4. 使用 grep 對 re2.txt 執行以下搜尋動作，不需要列印行號，所有結果依序累加到檔案 result2.txt，不要做任何的更動。
   (a) 開頭 b 結尾是 t，中間 1 個以上 a 的字串。
   (b) 開頭 b 結尾是 t，中間 0 個或 1 個 a 的字串。
   (c) dog, feet, good 三個字串。
   (d) boot 或 babt。
   (e) 開頭 b 結尾是 t，中間 pqr 重複 1 次以上。

8.4.3 練習三

1. 在 kvm8 執行 nmcli connection show 目前的乙太網路連線設定，利用管線命令，以 egrep 過濾 ipv?4' 大小寫不分，結果導向到 /root/myipv4。

Chapter 9

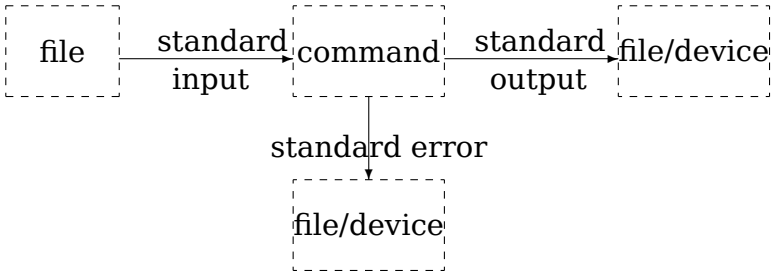
資料導向與管線處理

9.1 爲何要使用命令輸出重導向？

1. 儲存螢幕輸出的資訊；
2. 背景執行中的程式，不希望他干擾螢幕正常的輸出結果時；
3. 儲存系統例行命令（例如寫在 /etc/crontab 中的檔案）的執行結果；
4. 將已知的可能錯誤訊息丟掉『2> /dev/null 』；
5. 錯誤訊息與正確訊息需要分別輸出時。

9.2 資料流重導向

1. 指令執行過程之資料傳輸



2. 執行 cat /etc/crontab /etc/dywang，standard input 與 standard output 同時出現。

```
[dywang@dywHome ~]$ cat /etc/crontab /etc/dywang
# /etc/crontab 存在，螢幕出現以下 standard output
SHELL=/bin/bash

```

```
MAILTO=root
HOME=/

# run-parts
01 * * * * root nice -n 19 run-parts --report /etc/cron.hourly
02 4 * * * root nice -n 19 run-parts --report /etc/cron.daily
22 4 * * 0 root nice -n 19 run-parts --report /etc/cron.weekly
42 4 1 * * root nice -n 19 run-parts --report /etc/cron.monthly
# /etc/dywang 不存在，螢幕出現以下 standard error
cat: /etc/dywang: No such file or directory
```

3. 傳送指令

(a) 標準輸入 (stdin) ：代碼爲 0 ，使用 < 或 << ；

(b) 標準輸出 (stdout)：代碼爲 1 ，使用 > 或 >> ；

(c) 標準錯誤輸出 (stderr)：代碼爲 2 ，使用 2> 或 2>> ；

command	>, 1>	裝置或檔案	標準輸出至檔案，該檔案被覆蓋或建立。
	>>, 1>>		標準輸出至檔案，該檔案被建立或累加。
	2>		錯誤輸出至檔案，該檔案被覆蓋或建立。
	2>>		錯誤輸出至檔案，該檔案被建立或累加。
	<		輸入
	<<		結束的輸入字元

4. 檔案的建立方式

```
[root@linux ~]# ls -l /  >  ~/rootfile
# 將 ls -l / 要列出到螢幕上的資料『重新導向』到 ~/rootfile 檔案。
```

(a) 該檔案 (本例中是 ~/rootfile) 若不存在，系統會自動建立；

(b) 檔案存在時，系統會先將檔案內容清空再將資料寫入；

(c) 以 > 輸出到一個既存檔案中，該檔案會被覆蓋。

5. 下達 > 與 >>

```
# 將目前目錄下的檔案資訊儲存到 list.txt
[root@linux ~]# ls -al > list.txt

# 將根目錄下的資料累加到 list.txt
[root@linux ~]# ls -al / >> list.txt
```

6. 下達 2>

```
[csie@linux ~]$ find /home -name testing
find: /home/test1: Permission denied   <== Starndard error
find: /home/root: Permission denied    <== Starndard error
find: /home/masda: Permission denied   <== Starndard error
/home/csie/testing                   <== Starndard output

[csie@linux ~]$ find /home -name testing > list_right 2> list_error
[csie@linux ~]$ find /home -name testing > list_right 2> /dev/null
```

7. 下達 2>&1：標準輸出與標準錯誤輸出同時寫入同一個檔案

```
[csie@linux ~]$ find /home -name testing > list 2> list  <== 錯誤寫法
## 習題：請問上面命令的執行結果爲何？
[csie@linux ~]$ find /home -name testing > list 2>&1     <== 正確寫法
```

8. 使用 grep 命令過濾檔案 /usr/share/doc/autofs-5.0.7/README 中包含 configure 字串的所有行，共有 3 行。

```
[root@kvm7 ~]# grep configure /usr/share/doc/autofs-5.0.7/README
    ./configure
to configure the system.  See README.options for options that you can
give configure.
```

9. grep 命令過濾檔案 /usr/share/doc/autofs-5.0.7/README 中包含 configure 字串的 3 行原先顯示在螢幕上，現導向到檔案 /root/autofs.txt。

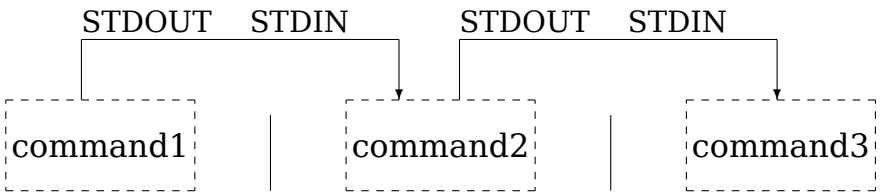
```
[root@kvm7 ~]# grep configure /usr/share/doc/autofs-5.0.7/README > /root
   /autofs.txt
```

10. 查看檔案 /root/autofs.txt 內容，與 grep 命令過濾檔案 /usr/share/doc/autofs-5.0.7/README 中包含 configure 字串的 3 行完全一樣。

```
[root@kvm7 ~]# cat /root/autofs.txt
    ./configure
to configure the system.  See README.options for options that you can
give configure.
```

9.3 管線命令 (pipe)

1. 管線命令的處理



2. 例題：使用 ls 指令輸出後的內容，被 less 讀取，並且利用 less 的功能前後翻動相關的資訊。

```
[root@linux ~]# ls -al /etc | less
```

9.4 實機練習

1. 資料導向

   (a) ls /etc/audit 導向到 /root/redirect1

   (b) ls /etcdy 錯誤導向到 /root/redirect2

   (c) ls /usr/local 導向到 /root/redirect3

   (d) ls /usr/etcdy 錯誤累加導向到 /root/redirect3

   (e) ls /bin/a* /lib64/autofs1 的 stdout 及 stderr 同時導向到 /root/redirect4

2. 管線命令過濾字串

   (a) 從檔案 /usr/share/doc/xz/README 中找出所有包含字串 info 的行列，導向到檔案 /root/info。

   (b) 從檔案 txtfile (例如 /usr/share/doc/systemd/LICENSES/OFL-1.1.txt) 中找出所有包含字串 str 的行列，導向到檔案 wlist。

   (c) 使用 nmcli 命令 show 目前的網路連線設定，經由管線命令找出包含 ipv4 IPV4 ip4 IP4 等字串的行，導向到檔案 myipv4。

Chapter 10

群組管理

10.1 簡介

1. 群組的 ID(Group ID, GID)，/etc/group 存所有群組，每一行以『:』分隔，共分爲四項：

```
[root@kvm7 ~]# head -4 /etc/group
root:x:0:
bin:x:1:
daemon:x:2:
sys:x:3:
```

   (a) 群組名稱。

   (b) 群組密碼：密碼記錄在 /etc/gshadow，很少使用群組登入，故通常不需要設定。

   (c) GID。

   (d) 支援的帳號名稱：即群組裡的所有帳號。

2. 主要群組 (primary group)：

   (a) 使用者一登入系統就擁有的群組權限，其記錄在 /etc/passwd 的第四欄位。

   (b) 不需要在 /etc/group 的第四個欄位寫入該帳號。

3. 附屬群組 (supplementary group)：

   (a) 該帳號在 /etc/group 中所屬的群組（第四個欄位中包含該帳號）。

   (b) 一個帳號可同時支援主要群組及附屬群組，只要這兩群組擁有的功能該帳號皆可使用。

(c) 新建的檔案或目錄依附屬群組設定權限。

10.2 增加群組

1. 查看 group 開頭的命令。

```
[root@kvm7 ~]# group [TAB][TAB]
groupadd   groupdel   groupmems  groupmod   groups
```

2. 增加 sharegrp 群組。

```
[root@kvm7 ~]# groupadd sharegrp
```

3. 查詢 sharegrp 群組是否存在？

```
[root@kvm7 ~]# getent group sharegrp
sharegrp:x:1000:
```

10.3 刪除群組

1. 刪除 sharegrp 群組。

```
[root@kvm7 ~]# groupdel sharegrp
```

2. 再查詢 sharegrp 群組，已不存在。

```
[root@kvm7 ~]# getent group sharegrp
```

10.4 群組更名

1. 增加 deyugrp 群組。

```
[root@kvm7 ~]# groupadd deyugrp
```

2. 查詢 deyugrp 群組。

```
[root@kvm7 ~]# getent group deyugrp
deyugrp:x:1000:
```

3. 使用 groupmod 變更群組名稱，先查看輔助說明，選項 -n 可以變更群組名稱。

```
[root@kvm7 ~]# groupmod --help
Usage: groupmod [options] GROUP

Options:
  -g, --gid GID                 change the group ID to GID
  -h, --help                    display this help message and exit
  -n, --new-name NEW_GROUP      change the name to NEW_GROUP
  -o, --non-unique              allow to use a duplicate (non-unique)
      GID
  -p, --password PASSWORD       change the password to this (encrypted)
                                PASSWORD
  -R, --root CHROOT_DIR         directory to chroot into
```

4. 使用 groupmod 將群組 deyugrp 變更爲 sharegrp。

```
[root@kvm7 ~]# groupmod -n sharegrp deyugrp
```

5. 查詢 deyugrp 群組。

```
[root@kvm7 ~]# getent group sharegrp
sharegrp:x:1000:
```

10.5 實機練習

1. 群組管理

   (a) 新增群組 shareg

   (b) 變更群組名稱：shareg 改成 sharegrp

   (c) 新增群組 sharey，再刪除，看看是否成功？

Chapter 11

帳號與身份管理

11.1 簡介

1. 帳號與 ID
   (a) 帳號 ID：使用者的 ID(User ID, UID) 與群組的 ID(Group ID, GID)
   (b) Linux 主機只認識 UID 及 GID，不認識我們輸入的帳號名稱。
   (c) /etc/passwd 及 /etc/group 分別記錄 UID 及 GID 的對應。
2. 如何登入 Linux 主機？(系統必須讀取 /etc/passwd 與 /etc/shadow)
   (a) 先找尋 /etc/passwd 裡面是否有這個帳號？
      i. 如果沒有則跳出，
      ii. 如果有則將該帳號對應的 UID 與 GID 讀出，該帳號的家目錄與 shell 設定也一併讀出；
   (b) 核對密碼表：核對輸入的密碼與 /etc/shadow 對應的帳號與 UID 之密碼是否相符？
   (c) 如果一切正確，就進入 Shell 控管的階段。
3. /etc/passwd 每一行以『:』分隔，共分爲七項

```
[root@dywOffice dywang]# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/bin/sh
daemon:x:2:2:daemon:/sbin:/bin/sh
adm:x:3:4:adm:/var/adm:/bin/sh
test:x:500:100:test:/home/test:/bin/bash
```

   (a) 帳號名稱 (root)

(b) 密碼 (x)：不顯示

(c) UID(0)

ID 範圍	該 ID 使用者特性
0	系統管理員，只要將 UID 改爲 0 即可成爲系統管理員。
1 ~ 499	保留給系統使用的 ID，除了 0 之外，其他的 UID 並無不同， 預設 1 ~ 99 保留給系統預設的帳號， 100 ~ 499 保留給一些服務使用。
500 ~ 65535	一般使用者的 UID 從 500 編起。

(d) GID：與 /etc/group 有關。

(e) 備註說明 (root)

(f) 家目錄 (/root)

(g) Shell(/bin/bash)

4. 一般帳號 id 編號

(a) 在 rhel/CentOS 6 ，一般帳號 UID 從 500 編起。

(b) 在 rhel/CentOS 7 及 8，一般帳號 UID 從 1000 編起。

11.2 新增帳號

1. useradd：增加使用者。

```
[root@kvm8 ~]# useradd [-ungGmMd] user
參數：
-u:  指定帳號的 UID。
-n:  群組爲 users。
-g:  後接之群組爲「主要群組(primary group)」，變動 /etc/passwd。
-G:  後接之群組爲「附屬群組(supplementary group)」，變動 /etc/group。
-M:  不要建立使用者家目錄。
-m:  建立使用者家目錄。
-d:  指定某個目錄爲家目錄，而不使用預設目錄。
-s:  指定使用的 shell。
```

2. 新增帳號 deyu1，其附屬群組爲 sharegrp。

```
[root@kvm8 ~]# useradd -G sharegrp deyu1
```

3. 新增帳號 deyu2，其附屬群組爲 sharegrp。

```
[root@kvm8 ~]# useradd -G sharegrp deyu2
```

4. 新增帳號 deyu3，此帳號不能使用互動式 shell 登入。

```
[root@kvm8 ~]# useradd -s /sbin/nologin deyu3
```

5. 檢查帳號 deyu1 及 deyu2，附屬群組爲 sharegrp，帳號 deyu3 沒有附屬群組。

```
[root@kvm8 ~]# id deyu1
uid=1000(deyu1) gid=1001(deyu1) groups=1001(deyu1),1000(sharegrp)
[root@kvm8 ~]# id deyu2
uid=1001(deyu2) gid=1002(deyu2) groups=1002(deyu2),1000(sharegrp)
[root@kvm8 ~]# id deyu3
uid=1002(deyu3) gid=1003(deyu3) groups=1003(deyu3)
```

6. 檢查帳號 deyu3，使用的 shell 爲不能登入的 /sbin/nologin。

```
[root@kvm8 ~]# getent passwd deyu3
deyu3:x:1002:1003::/home/deyu3:/sbin/nologin
```

7. 新增帳號 deyu4，並指定其 uid 爲 3584。

```
[root@kvm8 ~]# useradd -u 3584 deyu4
```

8. 檢查帳號 deyu4，uid 爲 3584。

```
[root@kvm8 ~]# id deyu4
uid=3584(deyu4) gid=3584(deyu4) groups=3584(deyu4)
```

11.3 刪除帳號

1. userdel 刪除帳號，先查輔助說明，選項 -r 可以一併刪除帳號的家目錄及信箱。

```
[root@kvm8 ~]# userdel --help
Usage: userdel [options] LOGIN

Options:
  -f, --force                   force some actions that would fail
      otherwise
                                e.g. removal of user still logged in
                                or files, even if not owned by the user
  -h, --help                    display this help message and exit
  -r, --remove                  remove home directory and mail spool
  -R, --root CHROOT_DIR         directory to chroot into
  -Z, --selinux-user            remove any SELinux user mapping for the
      user
```

2. userdel 刪除帳號 deyu1。

```
[root@kvm8 ~]# userdel deyu1
```

3. userdel 刪除帳號 deyu1 時，沒有使用選項 -r 一併刪除帳號的家目錄及信箱，所有 /home/deyu1 還存在，所屬帳號及群組都以 ID 顯示，因該 ID 的帳號及群組已不存在。

```
[root@kvm8 ~]# ll /home/
total 21
drwx------. 2  1000  1001  1024 Nov 23 11:26 deyu1
drwx------. 2 deyu2 deyu2  1024 Nov 23 11:26 deyu2
drwx------. 2 deyu3 deyu3  1024 Nov 23 11:26 deyu3
drwx------. 2 deyu4 deyu4  1024 Nov 23 11:27 deyu4
drwx------. 2 root  root  12288 Jul 10 11:53 lost+found
```

4. useradd 再新增帳號 deyu1 時，警告家目錄 /home/deyu1 存在，所以沒有複製任何檔案到家目錄。

```
[root@kvm8 ~]# useradd -G sharegrp deyu1
useradd: warning: the home directory already exists.
Not copying any file from skel directory into it.
Creating mailbox file: File exists
```

5. 但查看 /home 目錄中的 deyu1 目錄，並不屬於帳號 deyu1，原因爲新增的帳號，其 UID 並不是 1001。

```
[root@kvm8 ~]# ll /home/
total 21
drwx------. 2  1000  1001  1024 Nov 23 11:26 deyu1
drwx------. 2 deyu2 deyu2  1024 Nov 23 11:26 deyu2
drwx------. 2 deyu3 deyu3  1024 Nov 23 11:26 deyu3
drwx------. 2 deyu4 deyu4  1024 Nov 23 11:27 deyu4
drwx------. 2 root  root  12288 Jul 10 11:53 lost+found
```

6. 若再使用 userdel 加選項 -r 刪除帳號及其家目錄，出現信箱 /var/spool/mail/deyu1 及 /home/deyu1 不屬於帳號 deyu1，無法刪除。

```
[root@kvm8 ~]# userdel -r deyu1
userdel: /var/spool/mail/deyu1 not owned by deyu1, not removing
userdel: /home/deyu1 not owned by deyu1, not removing
[root@kvm8 ~]# ll /home/
total 21
drwx------. 2  1000  1001  1024 Nov 23 11:26 deyu1
drwx------. 2 deyu2 deyu2  1024 Nov 23 11:26 deyu2
drwx------. 2 deyu3 deyu3  1024 Nov 23 11:26 deyu3
drwx------. 2 deyu4 deyu4  1024 Nov 23 11:27 deyu4
drwx------. 2 root  root  12288 Jul 10 11:53 lost+found
```

7. 使用 rm 加選項 -rf 強制刪除 /home/deyu1 目錄及其次目錄，刪除信箱 /var/spool/mail/deyu1。

```
[root@kvm8 ~]# rm -rf /home/deyu1
[root@kvm8 ~]# rm -f /var/spool/mail/deyu1
```

8. 再重新新增帳號 deyu1，正常產生家目錄 /home/deyu1 及信箱 /var/spool/mail/deyu1。

```
[root@kvm8 ~]# useradd -G sharegrp deyu1
[root@kvm8 ~]# ll /home/
total 21
drwx------. 2 deyu1 deyu1  1024 Nov 23 11:34 deyu1
drwx------. 2 deyu2 deyu2  1024 Nov 23 11:26 deyu2
drwx------. 2 deyu3 deyu3  1024 Nov 23 11:26 deyu3
drwx------. 2 deyu4 deyu4  1024 Nov 23 11:27 deyu4
drwx------. 2 root  root  12288 Jul 10 11:53 lost+found
[root@kvm8 ~]# ll /var/spool/mail/
total 0
-rw-rw----. 1 deyu1 mail 0 Nov 23 11:34 deyu1
-rw-rw----. 1 deyu2 mail 0 Nov 23 11:26 deyu2
-rw-rw----. 1 deyu3 mail 0 Nov 23 11:26 deyu3
```

```
-rw-rw----. 1 deyu4 mail 0 Nov 23 11:27 deyu4
-rw-rw----. 1 rpc   mail 0 Jul 10 11:53 rpc
```

11.4 變更帳號

1. usermod 可以變更帳號，先查輔助說明。

   (a) -G：變更附屬群組。

   (b) -s：變更帳號登入時使用的 shell。

   (c) -u：變更帳號的 uid。

   (d) -l：變更帳號登入的名稱。

```
[root@kvm8 ~]# usermod --help
Usage: usermod [options] LOGIN

Options:
  -c, --comment COMMENT         new value of the GECOS field
  -d, --home HOME_DIR           new home directory for the user account
  -e, --expiredate EXPIRE_DATE  set account expiration date to
      EXPIRE_DATE
  -f, --inactive INACTIVE       set password inactive after expiration
                                to INACTIVE
  -g, --gid GROUP               force use GROUP as new primary group
  -G, --groups GROUPS           new list of supplementary GROUPS
  -a, --append                  append the user to the supplemental
      GROUPS
                                mentioned by the -G option without
                                    removing
                                him/her from other groups
  -h, --help                    display this help message and exit
  -l, --login NEW_LOGIN         new value of the login name
  -L, --lock                    lock the user account
  -m, --move-home               move contents of the home directory to
      the
                                new location (use only with -d)
  -o, --non-unique              allow using duplicate (non-unique) UID
  -p, --password PASSWORD       use encrypted password for the new
      password
  -R, --root CHROOT_DIR         directory to chroot into
  -s, --shell SHELL             new login shell for the user account
  -u, --uid UID                 new UID for the user account
  -U, --unlock                  unlock the user account
  -Z, --selinux-user SEUSER     new SELinux user mapping for the user
      account
```

2. 變更帳號 deyu1 爲 linda。

```
[root@kvm8 ~]# usermod -l linda deyu1
[root@kvm8 ~]# id linda
uid=3585(linda) gid=3585(deyu1) groups=3585(deyu1),1000(sharegrp)
```

3. 變更帳號 linda 的 uid 爲 4000。

```
[root@kvm8 ~]# usermod -u 4000 linda
[root@kvm8 ~]# id linda
uid=4000(linda) gid=3585(deyu1) groups=3585(deyu1),1000(sharegrp)
```

4. 變更帳號 linda 的登入 shell 爲 /sbin/nologin。

```
[root@kvm8 ~]# usermod -s /sbin/nologin linda
[root@kvm8 ~]# getent passwd linda
linda:x:4000:3585::/home/deyu1:/sbin/nologin
```

5. 變更帳號 linda 的附屬群組爲 apache。

```
[root@kvm8 ~]# usermod -G apache linda
[root@kvm8 ~]# id linda
uid=4000(linda) gid=3585(deyu1) groups=3585(deyu1),48(apache)
```

11.5 設定使用者密碼

1. passwd 命令設定或變更密碼

```
[root@kvm8 ~]# passwd [-lunxwS] username
參數：
--stdin  ：經由標準輸入取得密碼，配合管線處理，可大量更改帳號密碼。
```

2. 使用 --stdin 選項變更 deyu1, deyu2, deyu3 的登入密碼。

```
[root@kvm8 ~]# echo 'qweqwe' | passwd --stdin deyu1
Changing password for user deyu1.
passwd: all authentication tokens updated successfully.
[root@kvm8 ~]# echo 'qweqwe' | passwd --stdin deyu2
```

```
Changing password for user deyu2.
passwd: all authentication tokens updated successfully.
[root@kvm8 ~]# echo 'qweqwe' | passwd --stdin deyu3
Changing password for user deyu3.
passwd: all authentication tokens updated successfully.
[root@kvm8 ~]# echo 'qweqwe' | passwd --stdin deyu4
Changing password for user deyu4.
passwd: all authentication tokens updated successfully.
```

11.6 帳號家目錄問題

1. 刪除帳號 deyu1，但沒有下 -r 選項，原帳號 deyu1 的家目錄 /home/deyu1 還是存在，但擁有者不存在，所以只顯示原帳號 deyu1 的 UID 1003。

```
[root@kvm8 ~]# userdel deyu1
[root@kvm8 ~]# ll /home/
total 20
drwx------. 2  1003 deyu1    4096 Nov 14 05:48 deyu1
drwx------. 2 deyu2 deyu2    4096 Nov 14 05:50 deyu2
drwx------. 2 deyu3 deyu3    4096 Nov 13 14:33 deyu3
drwx------. 2 deyu4 deyu4    4096 Nov 14 05:50 deyu4
```

2. 刪除群組 deyu1，目錄 /home/deyu1 所屬的群組不存在，所以也只顯示原群組 deyu1 的 GID 1004。

11.7 帳號家目錄問題

1. 刪除帳號 deyu1，但沒有下 -r 選項，原帳號 deyu1 的家目錄 /home/deyu1 還是存在，但擁有者不存在，所以只顯示原帳號 deyu1 的 UID 1003。

```
[root@kvm8 ~]# userdel deyu1
[root@kvm8 ~]# ll /home/
total 20
drwx------. 2  1003 deyu1    4096 Nov 14 05:48 deyu1
drwx------. 2 deyu2 deyu2    4096 Nov 14 05:50 deyu2
drwx------. 2 deyu3 deyu3    4096 Nov 13 14:33 deyu3
drwx------. 2 deyu4 deyu4    4096 Nov 14 05:50 deyu4
```

2. 刪除群組 deyu1，目錄 /home/deyu1 所屬的群組不存在，所以也只顯示原群組 deyu1 的 GID 1004。

```
[root@kvm8 ~]# groupdel deyu1
[root@kvm8 ~]# ll /home/
total 20
drwx------. 2  1003     1004 4096 Nov 14 05:48 deyu1
drwx------. 2 deyu2 deyu2    4096 Nov 14 05:50 deyu2
drwx------. 2 deyu3 deyu3    4096 Nov 13 14:33 deyu3
drwx------. 2 deyu4 deyu4    4096 Nov 14 05:50 deyu4
```

3. 重新產生帳號 deyu1，出現家目錄已存在，不把 skel 目錄的設定複製過去的警告訊息。

```
[root@kvm8 ~]# useradd -G sharegrp deyu1
useradd: warning: the home directory already exists.
Not copying any file from skel directory into it.
```

4. 查看 /home/deyu1 的擁有者及所屬群組都不對。

```
[root@kvm8 ~]# ll /home/
total 20
drwx------. 2  1003     1004 4096 Nov 14 05:48 deyu1
drwx------. 2 deyu2 deyu2    4096 Nov 14 05:50 deyu2
drwx------. 2 deyu3 deyu3    4096 Nov 13 14:33 deyu3
drwx------. 2 deyu4 deyu4    4096 Nov 14 05:50 deyu4
```

5. chwon 變更 /home/deyu1 的擁有者及所屬群組爲 deyu1。

```
[root@kvm8 ~]# chown deyu1.deyu1 /home/deyu1/
```

6. 再查看 /home/deyu1 的擁有者及所屬群組都是 deyu1。

```
[root@kvm8 ~]# ll /home/
total 20
drwx------. 2 deyu1 deyu1    4096 Nov 14 05:48 deyu1
drwx------. 2 deyu2 deyu2    4096 Nov 14 05:50 deyu2
drwx------. 2 deyu3 deyu3    4096 Nov 13 14:33 deyu3
drwx------. 2 deyu4 deyu4    4096 Nov 14 05:50 deyu4
```

7. su 切換到帳號 deyu1，正常存取家目錄的環境變數。

```
[root@kvm8 ~]# su - deyu1
[deyu1@kvm8 ~]$ exit
logout
```

11.8 帳號整體參數變更

1. 變更用户名、群組名及其家目錄，查看家目錄，用户 deyu1 主要群組是 deyu1，家目錄是 /home/deyu1。

```
[root@kvm8 ~]# ll /home/
total 23
drwx------. 2 deyu1 deyu1     1024 Nov 22 20:23 deyu1
drwx------. 2 deyu2 deyu2     1024 Nov 23 07:43 deyu2
drwx------. 2 deyu3 deyu3     1024 Nov 22 19:21 deyu3
drwx------. 2 deyu4 deyu4     1024 Nov 23 07:45 deyu4
drwx------. 2 root  root     12288 Jul 10 13:46 lost+found
```

2. 變更群組名 deyu1 爲 linda，/home/deyu1 的群組由 deyu1 變更成 linda 了。

```
[root@kvm8 ~]# groupmod -n linda deyu1
[root@kvm8 ~]# ll /home/
total 23
drwx------. 2 deyu1 linda     1024 Nov 22 20:23 deyu1
drwx------. 2 deyu2 deyu2     1024 Nov 23 07:43 deyu2
drwx------. 2 deyu3 deyu3     1024 Nov 22 19:21 deyu3
drwx------. 2 deyu4 deyu4     1024 Nov 23 07:45 deyu4
drwx------. 2 root  root     12288 Jul 10 13:46 lost+found
```

3. 變更用户名 deyu1 爲 linda，/home/deyu1 的用户由 deyu1 變更成 linda 了。

```
[root@kvm8 ~]# usermod -l linda deyu1
[root@kvm8 ~]# ll /home/
total 23
drwx------. 2 linda linda     1024 Nov 22 20:23 deyu1
drwx------. 2 deyu2 deyu2     1024 Nov 23 07:43 deyu2
drwx------. 2 deyu3 deyu3     1024 Nov 22 19:21 deyu3
drwx------. 2 deyu4 deyu4     1024 Nov 23 07:45 deyu4
drwx------. 2 root  root     12288 Jul 10 13:46 lost+found
```

4. 變更用户名 linda 的家目錄名，由 deyu1 變爲 linda，/home/deyu1 變成 /home/linda。

```
[root@kvm8 ~]# mv /home/deyu1 /home/linda
[root@kvm8 ~]# ll /home/
total 23
drwx------. 2 deyu2 deyu2     1024 Nov 23 07:43 deyu2
drwx------. 2 deyu3 deyu3     1024 Nov 22 19:21 deyu3
drwx------. 2 deyu4 deyu4     1024 Nov 23 07:45 deyu4
drwx------. 2 linda linda     1024 Nov 22 20:23 linda
drwx------. 2 root  root     12288 Jul 10 13:46 lost+found
```

5. 變更 /home/deyu1 目錄名爲 /home/linda 後，切換身份爲 linda 卻找不到家目錄，原因爲 linda 的，/home/deyu1 變成 /home/linda。

```
[root@kvm8 ~]# su - linda
su: warning: cannot change directory to /home/deyu1: No such file or
    directory
-bash-4.2$ exit
logout
```

6. linda 找不到家目錄原因爲 linda 的家目錄，還是設定爲 /home/deyu1，必須變更爲 /home/linda。

```
[root@kvm8 ~]# usermod -d /home/linda linda
```

7. 再切換身份爲 linda，正常進入家目錄 /home/linda。

```
[root@kvm8 ~]# su - linda
Last login: Fri Nov 30 08:31:56 CST 2018 on pts/0
[linda@kvm8 ~]$ exit
logout
```

8. 其他用户 deyu2 deyu3 deyu4 也用同樣的方式改名。

11.9 實機練習

1. 建立用户帳號
   (a) deyu1 附屬群組爲 sharegrp
   (b) deyu2 附屬群組爲 sharegrp
   (c) deyu3 無法使用 shell 登入
   (d) deyu4 指定 uid 爲 3854
   (e) 四個帳號的密碼皆設定爲 123qwe
2. 帳號名稱、UID、附屬群組設定錯誤皆可使用 usermod 修改。
3. 若刪除帳號重新新增，刪除時必須連「家目錄」一併刪除，否則新增的帳號可能無法存取家目錄。

Chapter 12

su 與 sudo

12.1 su 切換使用者

1. su：轉換帳號身份。

```
[root@kvm8 ~]# su [-lcm] [username]
參數：
-, -l, --login username ：變換身份爲 username，沒指定 username 則爲 root
    ，
       取得一個新的 shell，且使用 username 的所有相關環境設定檔。
-m, -p, --preserve-environment ：使用目前的環境設定，而不重新讀取新使用
   者的設定檔。
-c, --command <command>  ：僅進行一次指令 command。
```

2. su - 變換身份會使用新帳號 deyu1 的環境變數，所以所在目錄爲 deyu1 的家目錄。

```
[root@kvm8 ~]# su - deyu1
[deyu1@kvm8 ~]$ pwd
/home/deyu1
```

3. 由帳號 deyu1 使用 su 變換身份爲 root，使用原帳號 deyu1 的環境變數，所以所在目錄爲 deyu1 的家目錄。

```
[deyu1@kvm8 ~]$ su
Password:
[root@kvm8 deyu1]# pwd
/home/deyu1
[root@kvm8 deyu1]# exit
exit
```

4. 由帳號 deyu1 使用 su - 變換身份爲 root，使用帳號 root 的環境變數，所以所在目錄爲 root 的家目錄。

```
[deyu1@kvm8 ~]$ su -
Password:
Last login: Sat Oct  3 06:53:15 CST 2015 on pts/0
[root@kvm8 ~]# pwd
/root
[root@kvm8 ~]# exit
logout
[deyu1@kvm8 ~]$ exit
logout
```

5. 使用 su 變換身份爲 deyu1，使用帳號 root 的環境變數，所以所在目錄爲 root 的家目錄。

```
[root@kvm8 ~]# su deyu1
[deyu1@kvm8 root]$ pwd
/root
[deyu1@kvm8 root]$ exit
exit
[root@kvm8 ~]#
```

12.2 sudo 以其他身份執行命令

1. sudo 命令可以其他用户的權限執行命令，不指定用户就是以 root 權限，執行時是輸入自己的密碼，不是 root 的密碼。

2. 可以使用 sudo 用 root 權限執行命令的用户設定在 /etc/sudoers 檔案中，一般不會直接編輯 /etc/sudoers，而是執行 visudo 設定，格式如下：

```
帳號名稱     來源主機=(可切換帳號) 可執行的指令
%群組名稱     來源主機=(可切換帳號) 可執行的指令
```

3. 執行 sudo 時不用輸入密碼，可在「可執行的指令」前加上 NOPASSWD:。

```
帳號名稱     來源主機=(可切換帳號) NOPASSWD:可執行的指令
%群組名稱     來源主機=(可切換帳號) NOPASSWD:可執行的指令
```

12.3 sudo 用户测試

1. 範例：visudo 編輯 /etc/sudoers，增加用户 deyu9 可免密碼以 root 權限從任何主機執行任何命令。

```
[root@kvm8 ~]# visudo
[root@kvm8 ~]# grep deyu /etc/sudoers
deyu9   ALL=(ALL)   NOPASSWD:ALL
```

2. 新增用户 deyu9

```
[root@kvm8 ~]# useradd deyu9
```

3. 切換身份爲 deyu9

```
[root@kvm8 ~]# su - deyu9
```

4. 一般用户 deyu9 無法使用 hwclock 命令查詢硬體時間

```
[deyu9@kvm8 ~]$ hwclock
hwclock: Cannot access the Hardware Clock via any known method.
hwclock: Use the --verbose option to see the details of our search for
    an access method.
```

5. 一般用户 deyu9 可以使用 sudo hwclock 查詢硬體時間

```
[deyu9@kvm8 ~]$ sudo hwclock
2022-08-30 15:00:09.669163+08:00
```

12.4 sudo 群組測試

1. 範例：visudo 編輯 /etc/sudoers，增加群組 dygrp 可以 root 權限從任何主機執行命令 /sbin/hwclock。

```
[root@kvm8 ~]# visudo
[root@kvm8 ~]# grep dygrp /etc/sudoers
%dygrp  ALL=(ALL)   /sbin/hwclock
```

2. 新增群組 dygrp，並新增用户 deyu21，其附屬群組爲 dygrp。

```
[root@kvm8 ~]# groupadd dygrp
[root@kvm8 ~]# useradd deyu21 -G dygrp
```

3. 設定 deyu21 的密碼

```
[root@kvm8 ~]# echo qweqwe | passwd --stdin deyu21
Changing password for user deyu21.
passwd: all authentication tokens updated successfully.
```

4. 切換身份爲 deyu21

```
[root@kvm8 ~]# su - deyu21
```

5. 一般用户 deyu21 無法使用 hwclock 命令查詢硬體時間

```
[deyu21@kvm8 ~]$ hwclock
hwclock: Cannot access the Hardware Clock via any known method.
hwclock: Use the --verbose option to see the details of our search for
    an access method.
```

6. 一般用户 deyu21 附屬群組 dygrp，可以使用 sudo hwclock 查詢硬體時間。

```
[deyu21@kvm8 ~]$ sudo hwclock

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for deyu21:
2022-08-30 15:26:52.203540+08:00
```

12.5 實機練習

1. visudo 設定：deyu9, deyu21, dygrp, vgs 請依變數取代。
   (a) 增加用户 deyu9 可免密碼，從任何主機，切換任何帳號，執行任何命令。
   (b) 增加群組 dygrp 可以從任何主機，切換任何帳號，需要密碼可執行 /sbin/vgs。
2. 建立測試用户及群組
   (a) 群組 dygrp
   (b) 用户 deyu9，不屬於 dygrp 群組。
   (c) 用户 deyu21，附屬群組 dygrp 群組。
3. 測試
   (a) 切換用户 deyu9
      i. 執行 vgs，標準錯誤 (stderr) 導向到 ~/vgs.9.err。
      ii. 執行 sudo vgs，標準輸出 (stdout) 導向到 ~/vgs.9.out。
   (b) 切換用户 deyu21。
      i. 執行 vgs，標準錯誤 (stderr) 導向到 ~/vgs.21.err。
      ii. 執行 sudo vgs，標準輸出 (stdout) 導向到 ~/vgs.21.out。

Chapter 13

帳號密碼策略

13.1 /etc/shadow

1. Linux 帳號表在 /etc/passwd，帳號密碼雜湊後存在 /etc/shadow，每行密碼有九個欄位，以冒號「:」分隔如下：

```
[root@kvm8 ~]# head /etc/shadow
root:$6$6RvHznZZPKW3g82k$g5CJl8oIjczadx0ije46B1::0:99999:7:::
bin:*:18358:0:99999:7:::
daemon:*:18358:0:99999:7:::
adm:*:18358:0:99999:7:::
lp:*:18358:0:99999:7:::
sync:*:18358:0:99999:7:::
shutdown:*:18358:0:99999:7:::
halt:*:18358:0:99999:7:::
mail:*:18358:0:99999:7:::
operator:*:18358:0:99999:7:::
```

2. 密碼檔九個欄位：

   (a) 帳號名稱，要與 /etc/passwd 相同才可。

   (b) 密碼：眞正的密碼但已加密，密碼欄第一個字爲「*」或者是「!」，表示這個帳號不會被用來登入。

   (c) 上次更動密碼的日期，計算 Linux 日期的時間是以 1970 年 1 月 1 日爲 0，而 1971 年 1 月 1 日則爲 365。

   (d) 密碼不可被更動的天數，0 表示密碼隨時可以更動，如果設定爲 20，則 20 天之內皆無法改變密碼。

   (e) 密碼需要重新變更的天數，99999 表示密碼不需要重新設定。

   (f) 密碼需要變更前的警告：上例 7 表示 7 天內系統會警告帳號。

(g) 密碼過期的恕限時間：當密碼失效後，n 天內還可以登入。

(h) 帳號失效日期：計算方式同欄位 3。

(i) 保留：看以後有無新功能加入。

13.2 預設密碼過期

1. Linux 系統預設的密碼過期策略在 /etc/login.defs。

```
[root@kvm8 ~]# grep ^PASS_ -B7 /etc/login.defs
# Password aging controls:
#
# 密碼需要重新變更的天數，99999 表示密碼不需要重新設定。
#   PASS_MAX_DAYS   Maximum number of days a password may be used.
# 密碼不可被更動的天數，0 表示密碼隨時可以更動。
#   PASS_MIN_DAYS   Minimum number of days allowed between password
    changes.
# 密碼最短長度，5 表示密碼不能少於5個字元。
#   PASS_MIN_LEN    Minimum acceptable password length.
# 密碼需要變更前的警告，7 表示7天之內系統會警告帳號。
#   PASS_WARN_AGE   Number of days warning given before a password
    expires.
#
PASS_MAX_DAYS   99999
PASS_MIN_DAYS   0
PASS_MIN_LEN    5
PASS_WARN_AGE   7
```

2. 新增帳號 deyu11

```
[root@kvm8 ~]# useradd deyu11
```

3. chage 命令查詢變醱帳號密碼過期狀態。

```
[root@kvm8 ~]# chage --help
Usage: chage [options] LOGIN

Options:
  -d, --lastday LAST_DAY        set date of last password change to
      LAST_DAY
  -E, --expiredate EXPIRE_DATE  set account expiration date to
      EXPIRE_DATE
  -h, --help                    display this help message and exit
  -I, --inactive INACTIVE       set password inactive after expiration
                                to INACTIVE
  -l, --list                    show account aging information
```

```
  -m, --mindays MIN_DAYS        set minimum number of days before
      password
                                change to MIN_DAYS
  -M, --maxdays MAX_DAYS        set maximum number of days before
      password
                                change to MAX_DAYS
  -R, --root CHROOT_DIR         directory to chroot into
  -W, --warndays WARN_DAYS      set expiration warning days to WARN_DAYS
```

4. 查詢帳號 deyu11 的密碼過期狀態，如預設策略。

```
[root@kvm8 ~]# chage -l deyu11
Last password change                    : Dec 17, 2021
Password expires                        : never
Password inactive                       : never
Account expires                         : never
Minimum number of days between password change      : 0
Maximum number of days between password change      : 99999
Number of days of warning before password expires   : 7
```

5. 查詢帳號 deyu11 在 /etc/shadow 的各欄位值，與 chage 查詢結果一樣。

```
[root@kvm8 ~]# getent shadow deyu11
deyu11:!!:18978:0:99999:7:::
```

13.3 變更預設密碼過期策略

1. 修改預設密碼過期策略，密碼最長使用天數 30 天。

```
[root@kvm8 ~]# vim /etc/login.defs
[root@kvm8 ~]# grep ^PASS_  /etc/login.defs
PASS_MAX_DAYS   30
PASS_MIN_DAYS   0
PASS_MIN_LEN    5
PASS_WARN_AGE   7
```

2. 新增帳號 deyu12

```
[root@kvm8 ~]# useradd deyu12
```

3. 查詢帳號 deyu12 的密碼過期狀態，如預設策略，最長使用天數 30 天。

```
[root@kvm8 ~]# chage -l deyu12
Last password change                                    : Dec 17, 2021
Password expires                                        : Jan 16, 2022
Password inactive                                       : never
Account expires                                         : never
Minimum number of days between password change          : 0
Maximum number of days between password change          : 30
Number of days of warning before password expires       : 7
```

4. 查詢帳號 deyu12 在 /etc/shadow 的各欄位值，與 chage 查詢結果一樣。

```
[root@kvm8 ~]# getent shadow deyu12
deyu12:!!:18978:0:30:7:::
```

13.4 設定已存在帳號密碼過期狀態

1. 設定帳號 deyu11 密碼 20 天後必須變更。

```
[root@kvm8 ~]# chage -M 20 deyu11
[root@kvm8 ~]# chage -l deyu11
Last password change                                    : Dec 17, 2021
Password expires                                        : Jan 06, 2022
Password inactive                                       : never
Account expires                                         : never
Minimum number of days between password change          : 0
Maximum number of days between password change          : 20
Number of days of warning before password expires       : 7
```

2. 設定帳號 deyu11 密碼 5 天內不能變更。

```
[root@kvm8 ~]# chage -m 5 deyu11
[root@kvm8 ~]# chage -l deyu11
Last password change                                    : Dec 17, 2021
Password expires                                        : Jan 06, 2022
Password inactive                                       : never
Account expires                                         : never
Minimum number of days between password change          : 5
Maximum number of days between password change          : 20
Number of days of warning before password expires       : 7
```

3. 設定帳號 deyu11 密碼過期前 6 天警告。

```
[root@kvm8 ~]# chage -W 6 deyu11
[root@kvm8 ~]# chage -l deyu11
Last password change                                    : Dec 17, 2021
Password expires                                        : Jan 06, 2022
Password inactive                                       : never
Account expires                                         : never
Minimum number of days between password change          : 5
Maximum number of days between password change          : 20
Number of days of warning before password expires       : 6
```

13.5 實機練習

1. 設定預設密碼過期略策，也就是新增的帳號，其密碼會有以下策略。
   (a) 最長不用變更天數 30
   (b) 可以變更的最短天數 4
   (c) 密碼過期前 6 天警告
   (d) 密碼長度至少 8
2. 新增帳號 deyu11，chage 查看是否如預設，以管線命令配合 tee 命令導向到 /root/chage1。
3. 新增帳號 deyu12，chage 設定
   (a) 最長不用變更天數 20
   (b) 可以變更的最短天數 7
   (c) 密碼過期前 4 天警告
4. chage 查看 deyu12 帳號，是否如變更，並導向 /root/chage2。

Chapter 14

Linux 檔案權限與屬性

14.1 檔案權限

- 檔案權限與屬性爲學習 Linux 之重要關卡。
- 在根目錄"/" 下，輸入指令 `ls -al`

```
[dywang@mdk-dyw /]$ ls -al
total 72
drwxr-xr-x   18 root adm   4096 Sep 16 11:11 ./
drwxr-xr-x   18 root adm   4096 Sep 16 11:11 ../
-rw-r--r--    1 root root     0 Sep 16 11:11 .autofsck
drwxr-xr-x    2 root root  4096 Sep 28 12:50 bin/
drwxr-xr-x    3 root root  4096 Sep 16 11:11 boot/
drwxr-xr-x   26 root root 14420 Oct  1 12:30 dev/
drwxr-xr-x   85 root root  8192 Sep 29 12:59 etc/
drwxr-xr-x   18 root root  4096 Sep 26 12:45 home/
drwxr-xr-x    2 root root  4096 Sep  9 16:14 initrd/
drwxr-xr-x   11 root root  4096 Sep 28 12:49 lib/
drwxr-xr-x    5 root root  4096 Sep 29 12:59 mnt/
drwxr-xr-x    2 root root  4096 Jan  5  2004 opt/
dr-xr-xr-x  179 root root     0 Sep 16 11:10 proc/
-rw-------    1 root root  1024 Sep  9 13:06 .rnd
drwx------   16 root root  4096 Oct  1 13:13 root/
drwxr-xr-x    2 root root  8192 Sep 28 12:50 sbin/
drwxr-xr-x   10 root root     0 Sep 16 11:10 sys/
drwxrwxrwt   23 root root  4096 Oct  1 13:13 tmp/
drwxr-xr-x   12 root root  4096 Sep  9 13:15 usr/
drwxr-xr-x   23 root root  4096 Sep  9 17:31 var/
```

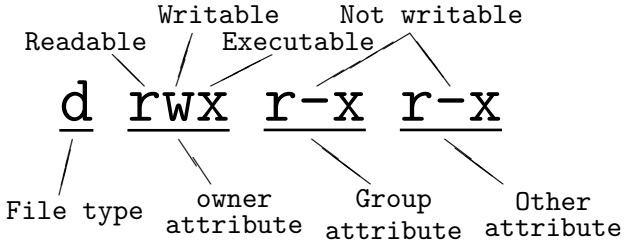
說明：

- total 72：檔案共 72 Blocks。

- 各欄位說明：

欄位一	欄位二	欄位三	欄位四	欄位五	欄位六	欄位七
drwxr-xr-x	3	root	root	4096	Sep 16 11:11	boot
檔案屬性	硬連結數目	擁有者	擁有者群組	大小	建檔日期	檔名

1. 第一欄位 drwxr-xr-x，共有檔案的 10 個屬性：



14.2 改變檔案權限

1. chmod：改變權限。

```
chmod [ -R ] xyz 檔案或目錄
chmod [ -R ] u = r 檔案或目錄
             g + w 檔案或目錄
             o - x 檔案或目錄
             a     檔案或目錄
chmod u=rwx,go=rx 檔案或目錄
chmod a-x 檔案或目錄
```

- xyz 爲三組 rwx 屬性數值之加總。

  r : 4

  w : 2

  x : 1

- -rwxr-xr-- 之屬性數值爲 754。

- 符號類型

  u : user

  g : group

  o : other

  a : all

  + : 加入

  - : 除去

  = : 設定

2. 例題：若從光碟複製某一目錄 abc （包含次目錄之所有檔案）至硬碟，則目前使用者雖爲檔案擁有者，但卻無法修改該檔案，請問如何以簡單的方式，讓使用者可以修改所有檔案？

```
chmod u+w -R abc
```

14.3 檔案的預設權限

1. umask：新增檔案或目錄預設權限，不指定 mode 列出目前預設權限，指定 mode 則設定預設權限。

```
[root@dywHome2 test]# umask [-S] [mode]
參數說明：
-S(Symbolic): 以符號類型顯示權限
```

2. 切換身份爲一般用户 deyu6。

```
[root@kvm8 ~]# su - deyu6
```

3. 執行 umask 不加 mode 參數，輸出 0022。

```
[deyu6@kvm8 ~]$ umask
0022
```

4. 新增檔案 file1 及目錄 dir1。

```
[deyu6@kvm8 ~]$ touch file1
[deyu6@kvm8 ~]$ mkdir dir1
```

5. umask 0022 表示擁有者不扣除任何權限，同群組及其他使用者皆須扣除『寫』的權限，所以建立新檔案或目錄的權限如下：

```
[deyu6@kvm8 ~]$ ll
total 4
drwxr-xr-x. 2 deyu6 deyu6 4096 Apr 29 16:29 dir1
```

```
4 -rw-r--r--. 1 deyu6 deyu6    0 Apr 29 16:29 file1
```

6. 執行 umask 指定 mode 0022。

```
[deyu6@kvm8 ~]$ umask 0002
```

7. 執行 umask 以符號方式查詢。

```
1 [deyu6@kvm8 ~]$ umask -S
u=rwx,g=rwx,o=rx
```

8. 新增檔案 file2 及目錄 dir2。

```
[deyu6@kvm8 ~]$ touch file2
2 [deyu6@kvm8 ~]$ mkdir dir2
```

9. umask 0002 表示擁有者及同群組不扣除任何權限，其他使用者皆須扣除『寫』的權限，所以建立新檔案或目錄的權限如下：

```
[deyu6@kvm8 ~]$ ll
2 total 8
drwxr-xr-x. 2 deyu6 deyu6 4096 Apr 29 16:29 dir1
4 drwxrwxr-x. 2 deyu6 deyu6 4096 Apr 29 16:42 dir2
-rw-r--r--. 1 deyu6 deyu6    0 Apr 29 16:29 file1
6 -rw-rw-r--. 1 deyu6 deyu6    0 Apr 29 16:42 file2
```

10. 登出 deyu6，再登入後 umask 查詢又回到 0222，表示 umask 設定只是目前的 shell 生效。

```
[deyu6@kvm4 ~]$ exit
2 logout
[root@kvm4 ~]# su - deyu6
4 Last login: Thu Oct 27 12:17:24 CST 2022 on pts/0
[deyu6@kvm4 ~]$ umask
6 0022
```

11. 要讓 deyu6 的 umask 維持 0002，必須將其寫入.bashrc。

```
[deyu6@kvm4 ~]$ vim .bashrc
[deyu6@kvm4 ~]$ tail -1 .bashrc
umask 0002
```

12. 登出 deyu6，再登入 deyu6，取得 shell 時會執行.bashrc，因此 umask 0002 會生效。

```
[deyu6@kvm4 ~]$ exit
logout
[root@kvm4 ~]# su - deyu6
Last login: Thu Oct 27 12:17:51 CST 2022 on pts/0
[deyu6@kvm4 ~]$ umask
0002
```

14.4 檔案的特殊屬性

1. 檔案的特殊屬性 SUID/SGID/SBIT
   (a) Set UID (SUID)；設置使文件在執行階段具有文件擁有者的權限。
      i. SUID 典型的的例子爲 /usr/bin/passwd。
         A. 帳號與密碼存放於 /etc/passwd 與 /etc/shadow。而 /etc/shadow 的權限爲 『-r--------』，且擁有者爲 root。
         B. 一般身份使用者要更新自己的密碼，執行 /usr/bin/passwd 卻可以存取密碼檔 /etc/shadow。
      ii. SUID 僅可用在『二進位制檔案 (binary file) 』上。
   (b) Set GID (SGID)：只對目錄有效。
      i. 目錄被設置後，任何用户在此目錄下建立的文件，所屬群組皆與該目錄所屬的群組相同。
      ii. 可使用在特定的多人團隊的專案開發上。
   (c) Sticky Bit (SBIT)：防刪除屬性，只對目錄有效。
      i. 在具有 SBIT 的目錄下，使用者若在該目錄下具有 w 及 x 的權限，則當使用者在該目錄下建立檔案或目錄時，只有檔案擁有者與 root 才有權力刪除。
      ii. 典型的例子如 /tmp 的權限是 『drwxrwxrwt』，任何用户都可在 /tmp 內新增、修改檔案，但僅有該檔案/目錄建立者與 root 能夠刪除自己的目錄或檔案。

2. SUID/SGID/SBIT 權限的顯示與設定

(a) 使用 ls -l 列出檔案屬性，如：

rwsrw-r--	表示設置 SUID 屬性
rwxrwsrw-	表示設置 SGID 屬性
rwxrw-rwt	表示設置 SBIT 屬性

(b) 如果本來在該位上有 x 屬性，則這些特殊屬性顯示爲小寫字母 (s, s, t)，否則顯示爲大寫字母 (S, S, T)。

(c) SUID/SGID/SBIT 權限設定與一般 rwx 屬性之設定類似：
4 爲 SUID
2 爲 SGID
1 爲 SBIT

(d) 例題：更改並顯示檔案 foo 的屬性。

```
[root@dywHome2 test]# chmod 4755 foo; ll foo
-rwsr-xr-x 1 root root 7 Sep  4 13:25 foo*
[root@dywHome2 test]# chmod 6755 foo; ll foo
-rwsr-sr-x 1 root root 7 Sep  4 13:25 foo*
[root@dywHome2 test]# chmod 1755 foo; ll foo
-rwxr-xr-t 1 root root 7 Sep  4 13:25 foo*
[root@dywHome2 test]# chmod 7666 foo; ll foo
-rwSrwSrwT 1 root root 7 Sep  4 13:25 foo
```

(e) 查 SUID 檔案 /usr/bin/passwd 的屬性爲 -rwsr-xr-x，其中 s 就是 SUID。

```
[root@kvm8 ~]# ll /usr/bin/passwd
-rwsr-xr-x. 1 root root 27832 Jun 10  2014 /usr/bin/passw
```

(f) 查 SGID 目錄 /usr/log/joural 的屬性爲 drwxr-sr-x，其中 s 就是 SGID。

```
[root@kvm8 ~]# ll -d /run/log/journal
drwxr-sr-x. 3 root systemd-journal 60 Dec  1 10:08 /run/log/journal
```

(g) 查 SBIT 目錄 /tmp 的屬性爲 drwxrwxrwt，其中 t 就是 SBIT。

```
[root@kvm8 ~]# ll -d /tmp
drwxrwxrwt. 7 root root 4096 Dec  2 09:27 /tmp
```

14.5 特殊屬性 SGID 練習

1. 先建立練習用目錄 /home/shared

```
[root@kvm8 ~]# mkdir /home/shared
```

2. 查看目錄 /home/shared，目前擁有者及群組皆爲 root，擁有者可讀、可寫、可訪問此目錄，同群組及其他人只能訪問及讀取。

```
[root@kvm8 ~]# ll -d /home/shared
drwxr-xr-x. 2 root root 1024 Nov 17 19:34 /home/shared
```

3. 改變目錄 /home/shared 群組爲 sharegrp，如果 sharegrp 群組不存在，必須先新增。

```
[root@kvm8 ~]# chgrp sharegrp /home/shared
```

4. 查看目錄 /home/shared 群組爲 sharegrp。

```
[root@kvm8 ~]# ll -d /home/shared
drwxr-xr-x. 2 root sharegrp 1024 Nov 17 19:34 /home/shared
```

5. 改變目錄 /home/shared，群組可讀、可寫、可訪問此目錄，其他人沒有任何權限，且設定 SGID，讓用户在此目錄內產生的檔案，所屬群組皆爲 sharegrp。

```
[root@kvm8 ~]# chmod 2770 /home/shared
```

6. 再查看目錄 /home/shared，其屬性出現 drwxrws---，符合要求。

```
[root@kvm8 ~]# ll -d /home/shared
drwxrws---. 2 root sharegrp 1024 Nov 17 19:34 /home/shared
```

14.6 特殊屬性 SGID 測試

1. 查詢 deyu1 及 deyu2 的附屬群組是 sharegrp。

```
[root@kvm8 ~]# getent group sharegrp
sharegrp:x:2206:deyu1,deyu2
```

2. 切換身份成 deyu1。

```
[root@kvm8 ~]# su - deyu1
Last login: Sat Nov  3 02:50:16 CST 2018 on pts/0
```

3. deyu1 的附屬群組是 sharegrp，所以可以進到 SGID 的目錄 /home/shared。

```
[deyu1@kvm8 ~]$ cd /home/shared/
```

4. 查看目錄 /home/shared 中沒有任何檔案。

```
[deyu1@kvm8 shared]$ ll
total 0
```

5. 新增空檔案 deyu1.txt。

```
[deyu1@kvm8 shared]$ touch deyu1.txt
```

6. 查看新增的空檔案 deyu1.txt，擁有者是 deyu1，因爲目錄 /home/shared SGID，所以在此目錄中新增的檔案群組是 sharegrp。

```
[deyu1@kvm8 shared]$ ll
total 1
-rw-rw-r--. 1 deyu1 sharegrp 0 Nov 23 14:11 deyu1.txt
```

7. 退出帳號 deyu1，再切換成 deyu2。

```
[deyu1@kvm8 shared]$ exit
logout
[root@kvm8 ~]# su - deyu2
```

8. deyu2 的附屬群組也是 sharegrp，所以也可以進到 SGID 的目錄 /home/shared。

```
[deyu2@kvm8 ~]$ cd /home/shared/
```

9. 目錄 /home/shared 中同群組的帳號，都可以讀取、寫入，所以 deyu2 也可以變更 deyu1 產生的檔案 deyu1.txt。

```
[deyu2@kvm8 shared]$ echo 'abcde' >> deyu1.txt
[deyu2@kvm8 shared]$ cat deyu1.txt
abcde
```

10. 退出帳號 deyu2，再切換成 deyu4。

```
[deyu2@kvm8 shared]$ exit
logout
[root@kvm8 ~]# su - deyu4
```

11. deyu4 不是 sharegrp 群組成員，所以無法進入目錄 /home/shared。

```
[deyu4@kvm8 ~]$ cd /home/shared/
-bash: cd: /home/shared/: Permission denied
```

12. 退出 deyu4 帳號。

```
[deyu4@kvm8 ~]$ exit
logout
```

14.7 實機練習

1. 建立用户的預設權限。

   (a) 設定帳號 deyu6 預設權限爲 0002，且登入後必須立即生效。

   (b) deyu6 登入後，分別建立 maskf 檔案及 maskd 目錄。

   (c) 觀察檔案 maskf 的屬性是否爲 664？目錄 maskd 的屬性是否爲 775？

2. 建立 sharegrp 團隊工作目錄，deyu1.f, deyu2.f 中的 deyu1, deyu2 請依參數檔變更。

   (a) 建立 /home/shared 目錄，並設定其爲群組 sharegrp 團隊工作目錄，擁有者及群組有 rwx 權限，其他沒有任何權限，且 SETGID。

   (b) deyu1 與 deyu2 都屬於 sharegrp 群組，分別用 deyu1 及 deyu2 帳號在 /home/shared 目錄中建立檔案 deyu1.f 及 deyu2.f。

   (c) 觀察 deyu1.f 及 deyu2.f 所屬的群組，看看 deyu1 能不能更改 deyu2.f 內容？deyu2 是否可以變更 deyu1.f 內容？

   (d) deyu4 不屬於 sharegrp 群組，測試 deyu4 是否可以進入 /home/shared 目錄？將標準錯誤 (stderr) 導向到 /home/deyu4/cd.shared。

Chapter 15

Access Control Lists

15.1 簡介

1. 什麼是 ACL (Access Control List)？
   (a) 提供傳統的 owner,group,others 的 read,write,execute 權限之外的細部權限設定。
   (b) ACL 可以針對單一使用者，單一檔案或目錄來進行 r, w, x 的權限規範。
   (c) ACL 主要可以針對以下項目來控制權限：
      i. 使用者 (user)：可以針對使用者來設定權限；
      ii. 群組 (group)：針對群組爲對象來設定其權限；
      iii. 預設屬性 (mask)：針對在該目錄下在建立新檔案/目錄時，規範新資料的預設權限。
2. `setfacl`：設定 ACL 的狀態。

```
[root@demo ~]# setfacl [-bkndRLPvh] [{-m|-x} acl_spec] [{-M|-X} acl_file
   ] file ...
選項：
-m ：設定後續的 acl 參數給檔案使用，不可與 -x 合用；
-x ：刪除後續的 acl 參數，不可與 -m 合用；
-b ：移除所有的 ACL 設定參數；
-k ：移除預設的 ACL 參數；
-R ：遞迴設定 acl，即包括次目錄都會被設定；
-d ：設定『預設 acl 參數』，只對目錄有效，在該目錄新建的資料會引用此預設
   值
acl_spec:
[d[efault]:] [u[ser]:]uid [:perms]
[d[efault]:] g[roup]:gid [:perms]
[d[efault]:] m[ask][:] [:perms]
[d[efault]:] o[ther][:] [:perms]
```

```
14 # d 選項，表示該目錄預設有設定的 acl 規範，類似 umask 的功能。
```

15.2 ACL 練習

1. 先複製測試檔案

```
[root@kvm8 ~]# cp /etc/fstab /var/tmp/
```

2. 查看複製的檔案擁有者，群組皆爲 root，目前權限爲 644，root 有讀取及寫入的權限。

```
1 [root@kvm8 ~]# ll /var/tmp/fstab
-rw-r--r--. 1 root root 694 Oct 15 06:29 /var/tmp/fstab
```

3. 使用 getfacl 查看 /var/tmp/fstab 目前的 ACL，沒有其他用户有特殊的權限。

```
  [root@kvm8 ~]# getfacl /var/tmp/fstab
2 getfacl: Removing leading '/' from absolute path names
  # file: var/tmp/fstab
4 # owner: root
  # group: root
6 user::rw-
  group::r--
8 other::r--
```

4. 改變 /var/tmp/fstab 的 ACL，用户 deyu1 有讀取及寫入的權限。

```
[root@kvm8 ~]# setfacl -m u:deyu1:rw /var/tmp/fstab
```

5. 改變 /var/tmp/fstab 的 ACL，用户 deyu2 沒有任何權限。

```
[root@kvm8 ~]# setfacl -m u:deyu2:- /var/tmp/fstab
```

6. 再使用 getfacl 查看 /var/tmp/fstab 目前的 ACL，其他用户 deyu1 及 deyu2 有特殊的權限。

```
[root@kvm8 ~]# getfacl /var/tmp/fstab
getfacl: Removing leading '/' from absolute path names
# file: var/tmp/fstab
# owner: root
# group: root
user::rw-
user:deyu1:rw-
user:deyu2:---
group::r--
mask::rw-
other::r--
```

7. 列出 /var/tmp/fstab 可以看到權限部分多了個 +，且權限不是原本的“644”？

```
[root@kvm8 ~]# ll /var/tmp/fstab
-rw-rw-r--+ 1 root root 694 Oct 15 06:29 /var/tmp/fstab
```

15.3 ACL 變更

1. 測試檔案 /var/tmp/fstab 目前的 ACL。

```
[root@kvm8 ~]# getfacl /var/tmp/fstab
getfacl: Removing leading '/' from absolute path names
# file: var/tmp/fstab
# owner: root
# group: root
user::rw-
user:deyu2:---
user:deyu1:rw-
group::r--
mask::rw-
other::r--
```

2. /var/tmp/fstab 新增 deyu4 可讀寫。

```
[root@kvm8 ~]# setfacl -m u:deyu4:rw /var/tmp/fstab
```

3. 查詢 /var/tmp/fstab 的 ACL 有 deyu4 可讀寫。

```
[root@kvm8 ~]# getfacl /var/tmp/fstab
getfacl: Removing leading '/' from absolute path names
# file: var/tmp/fstab
# owner: root
# group: root
user::rw-
user:deyu2:---
user:deyu4:rw-
user:deyu1:rw-
group::r--
mask::rw-
other::r--
```

4. 刪除 /var/tmp/fstab ACL 中的 deyu4 可讀寫。

```
[root@kvm8 ~]# setfacl -x u:deyu4 /var/tmp/fstab
```

5. 查詢 /var/tmp/fstab 的 ACL 沒有 deyu4 設定了。

```
[root@kvm8 ~]# getfacl /var/tmp/fstab
getfacl: Removing leading '/' from absolute path names
# file: var/tmp/fstab
# owner: root
# group: root
user::rw-
user:deyu2:---
user:deyu1:rw-
group::r--
mask::rw-
other::r--
```

15.4 實機練習

1. 建立包含 ACL 的檔案
   (a) 複製 /etc/fstab 到 /var/tmp 目錄下
   (b) 查看 /var/tmp 目錄下的 fstab 檔案，其 acl 有無特殊的用户權限設定？
   (c) 設定 deyu1 可讀寫 fstab
   (d) 設定 deyu2 不可讀寫 fstab
   (e) 不要變動其他設定
2. 變更 ACL 的檔案
   (a) 查看 /var/tmp 目錄下的 fstab 檔案，其 acl 有無特殊的用户權限設定？
   (b) 設定 deyu4 可讀寫 fstab
   (c) 刪除 deyu4 可讀寫 fstab
   (d) 不要變動其他設定

Chapter 16

例行性命令

16.1 Linux 的例行性命令

1. Linux 工作排程的種類：at, cron

   (a) 例行性：每隔一定的週期要執行的工作；
   cron：工作持續例行性的工作；

   (b) 突發性：只做一次；
   at：工作僅執行一次就從 Linux 系統中的排程中取消。

2. 常見的例行性工作：

   (a) 進行登錄檔的資料輪替：適時適量將登錄檔備份，並以新開的檔案進行記錄。

   (b) rpm 資料庫的建立。

   (c) 建立 locate 的資料庫。

   (d) 進行程序的分析：刪除僵屍程序，以保持記憶體的工作能力。

   (e) 登錄檔視察：解析有問題的紀錄檔，以維護主機的安全性。

16.2 僅執行一次的工作排程：at

1. at 的使用限制如下：

   (a) 先在 /etc/at.allow 檔案中尋找可以使用 at 指令的使用者。

   (b) 若無 /etc/at.allow 檔案，則尋找 /etc/at.deny 檔案不能使用 at 指令的使用者。

   (c) 若上述兩檔案皆不存在，則只有 root 可以使用 at 指令。

(d) 預設只有空白的 /etc/at.deny 檔，即所有人皆可使用 at 指令。

```
[root@localhost csie]# cat /etc/at.allow
cat: /etc/at.allow: No such file or directory
[root@localhost csie]# cat /etc/at.deny

[root@localhost csie]#
```

2. at：僅執行一次的工作排程。

```
[root@linux ~]# at [-m] TIME
參數：
-m   ：當 at 的工作完成後，以 email 的方式通知使用者該工作已完成。
TIME：時間格式，定義『什麼時候要進行 at 這項工作』的時間，格式有：
  HH:MM                          ex> 04:00
        在今日的 HH:MM 時刻進行，若該時刻已超過，則明天的 HH:MM 進行此工
            作。
  HH:MM YYYY-MM-DD               ex> 04:00 2005-12-03
        強制規定在某年某月的某一天的特殊時刻進行該工作。
  HH:MM[am|pm] [Month] [Date]    ex> 04pm December 3
        強制在某年某月某日的某時刻進行。
  HH:MM[am|pm] + number [minutes|hours|days|weeks]
        ex> now + 5 minutes      ex> 04pm + 3 days
        某個時間點『再加幾個時間後』才進行。
```

3. 下達工作排程

```
[dywang@dywOffice ~]$ at 17:00 063007
at> echo Hello!
at> <EOT>  <==輸入[Ctrl]+d，出現結束字樣<EOT>
job 1 at 2007-06-30 17:00
```

4. 查看目前的工作排程

```
[dywang@dywOffice ~]$ atq
1       2007-06-30 17:00 a dywang
```

5. 刪除工作排程

```
[dywang@dywOffice ~]$ atrm 1
[dywang@dywOffice ~]$ atq
[dywang@dywOffice ~]$
```

16.3 循環執行的工作排程

1. crontab 的使用限制如下：

   (a) 先在 /etc/cron.allow 檔案中尋找可以使用 crontab 指令的使用者。

   (b) 若無 /etc/cron.allow 檔案，則尋找 /etc/cron.deny 檔案不能使用 crontab 指令的使用者。

   (c) 若上述兩檔案皆不存在，則只有 root 可以使用 crontab 指令。

   (d) 預設只有空白的/etc/cron.deny 檔，即所有人皆可使用 crontab 指令。

```
[root@kvm8 ~]# ll /etc/cron [TAB][TAB]
cron.d/       cron.deny     cron.monthly/ cron.weekly/
cron.daily/   cron.hourly/  crontab
[root@kvm8 ~]# cat /etc/cron.deny
```

2. crontab：循環執行的工作排程。

```
[root@kvm8 ~]# crontab [-u user] [-l | -e | -r]
參數說明：
-u user ：只有 root 能下達的參數，視察或編譯其他使用者的 crontab 內容
-l      ：列出 crontab 的內容
-e      ：編輯 crontab 的內容
-r      ：刪除 crontab 的內容
```

   (a) 五個時間數字參數分別代表：

數字代表的意義	分鐘	小時	日期	月份	週
範圍	0-59	0-23	1-31	1-12	0-6 (0 或 7 爲星期天)

   (b) 輔助字符：

特殊字符	代表意義
*	代表任何時刻都接受。
,	代表分隔時段。 例如：在 0300 及 0600 時工作，則 `0 3,6 * * * command`
-	代表一段時間範圍內。 例如：在 0800 到 1200 之間每小時的 20 分都進行工作，則 `20 8-12 * * * command`
/n	每隔 n 單位間隔。 例如每 5 分鐘進行一次，則 `*/5 * * * * command`

3. 編輯排程：進入 vi 環境編輯排程。

```
[root@kvm8 ~]# crontab -e
crontab: installing new crontab
```

4. 查看排程：每星期一 14:00 touch 檔案 /tmp/crontab_test1。

```
[root@kvm8 ~]# crontab -l
0 14 * * 1 /usr/bin/touch /tmp/crontab_test1
```

5. 在星期一 14:00 後查看 /tmp/crontab_test1，存在且修改時間 14:00。

```
[root@kvm8 ~]# ll /tmp/crontab_test1
-rw-r--r--. 1 root root 0 Nov 11 14:00 /tmp/crontab_test1
```

6. 刪除排除

```
[root@kvm8 ~]# crontab -r
[root@kvm8 ~]# crontab -l
no crontab for root
```

7. 例題：如果您 (帳號 deyu2) 要建立一個自動執行的工作，在每天 17:00 執行 /bin/echo "closing time"，該如何做？

   (a) 若指定用户 deyu2 編輯工作排程，可以先切換到 deyu2，再執行 crontab -e 進行編輯，完成編輯後退出回到 root 身份。

```
[root@kvm8 ~]# su - deyu2
[deyu2@kvm8 ~]$ crontab -e
no crontab for deyu2 - using an empty one
crontab: no changes made to crontab
[deyu2@kvm8 ~]$ exit
logout
```

   (b) 建議直接以 root 身份執行 crontab -e 加上 -u deyu2 參數指定編輯用户 deyu2 的工作排程，就可以不用切換身份。

```
[root@kvm8 ~]# crontab -e -u deyu2
no crontab for deyu2 - using an empty one
crontab: installing new crontab
```

(c) 編輯完用户 deyu2 的工作排程後，以 -l 選項查看排程內容是否正確。

```
[root@kvm8 ~]# crontab -l -u deyu2
0 17 * * * /bin/echo "closing time"
```

16.4 實機練習

1. 用户 cronuser 依下列時間執行例行性命令"/bin/echo hiya"。

   (a) 每 2 小時

   (b) 0-6 點的 12 分

   (c) 每月 1 及 15 日的 10 點 15 分

   (d) 每週日的 13 點每 3 分鐘

   (e) 每天 18 點 25 分

Chapter 17

Chronyd vs. ntpd 校時

17.1 前言

1. RHEL/CentOS 7 校時新套件 chrony 取代 ntp。
   (a) 更快速以及更精確的同步。
   (b) 頻率校正的範圍較大。
   (c) 擁有對於時鐘頻率快速變更的反應速度。
   (d) 在初始同步後不會發生 clock stepping。Stepping 意思是時間突然從不正確的時間一步調到正確時間的合理誤差時間，如果誤差小於 128ms 則以 slewing 擺動修正。
   (e) 適用於間歇性的網路連線。
2. chrony 不完全支援 ntp 中的所有可用功能，因此基於相容性上的原因，RHEL/CentOS 依然提供 ntp 套件，使用時必須明確地移除 chrony 並安裝 ntp 來代替。
3. Chrony 的時間設定演算法擁有幾項 ntp 實作所沒有的優點。

17.2 chronyd 使用

1. chrony 是 RHEL/CentOS 8 預設的校時 dameon，設定檔在 /etc/chrony.conf，新增校時伺服器 server.deyu.wang，其中 iburst 選項是用來加速初始同步。

```
[root@kvm8 ~]# vim /etc/chrony.conf
[root@kvm8 ~]# grep wang /etc/chrony.conf
pool server.deyu.wang iburst
```

2. 重新啓動 chronyd。

```
[root@kvm8 ~]# systemctl restart chronyd.service
```

3. chronyd 預設開機啓動，不放心可再設定一次。

```
[root@kvm8 ~]# systemctl enable chronyd.service
```

4. chronyd 使用 tracking 命令追踪校時來源伺服器，tracking 命令可以用 TAB 提示補齊，等一段時間，系統校正好時間後，Leap status : Normal。

```
[root@kvm8 ~]# chronyc tracking
Reference ID    : C0A87A01 (deyu.wang)
Stratum         : 5
Ref time (UTC)  : Mon Dec 16 01:59:45 2019
System time     : 0.000014439 seconds slow of NTP time
Last offset     : +0.015839783 seconds
RMS offset      : 0.015839783 seconds
Frequency       : 17.482 ppm fast
Residual freq   : +250.266 ppm
Skew            : 19.178 ppm
Root delay      : 0.005921864 seconds
Root dispersion : 1.105486751 seconds
Update interval : 64.1 seconds
Leap status     : Normal
```

5. chronyc 使用 sources 命令檢查 chronyd 校時來源伺服器，sources 可用 TAB 提示補齊，deyu.wang (server.deyu.wang 與 deyu.wang 是同一台主機) 已在表列中。

```
[root@kvm8 ~]# chronyc sources
210 Number of sources = 2
MS Name/IP address         Stratum Poll Reach LastRx Last sample
===============================================================================

^? deyu.wang                     0   7     0     -     +0ns[   +0ns] +/-
       0ns
^? deyu.wang                     0   7     0     -     +0ns[   +0ns] +/-
       0ns
```

17.3 立即校時

1. 查詢目前時間

```
[root@kvm8 ~]# date
Wed Nov  2 02:05:08 PM CST 2022
```

2. 將時間設定成不是今天的日期 (2021-11-11)。

```
[root@kvm8 ~]# date -s 2021-11-11
Thu Nov 11 12:00:00 AM CST 2021
```

3. 不關閉 chronyd 服務，使用 -Q 選項，可以查看時間誤差，但不會眞的校時。

```
[root@kvm8 ~]# chronyd -Q "pool deyu.wang iburst"
2021-11-10T16:00:17Z chronyd version 4.1 starting (+CMDMON +NTP +
    REFCLOCK
+RTC +PRIVDROP +SCFILTER +SIGND +ASYNCDNS +NTS +SECHASH +IPV6 +DEBUG)
2021-11-10T16:00:17Z Disabled control of system clock
2021-11-10T16:00:21Z System clock wrong by 30809113.195522 seconds (
    ignored)
2021-11-10T16:00:21Z chronyd exiting
```

4. 要用 chronyd 立即校時，必須先關閉 chronyd 服務。

```
[root@kvm8 ~]# systemctl stop chronyd.service
```

5. chronyd 立即校時，校正 30809113.195512 秒。

```
[root@kvm8 ~]# chronyd -q "pool deyu.wang iburst"
2021-11-10T16:01:04Z chronyd version 4.1 starting (+CMDMON +NTP +
    REFCLOCK
+RTC +PRIVDROP +SCFILTER +SIGND +ASYNCDNS +NTS +SECHASH +IPV6 +DEBUG)
2021-11-10T16:01:04Z Initial frequency -11.769 ppm
2021-11-10T16:01:08Z System clock wrong by 30809113.195512 seconds (step
    )
2022-11-02T06:06:21Z chronyd exiting
```

6. 再查時間，校時成功。

```
[root@kvm8 ~]# date
Wed Nov  2 02:06:25 PM CST 2022
```

7. 恢復啓動 chronyd 服務

```
[root@kvm8 ~]# systemctl start chronyd.service
```

17.4 *makestep 校時

1. chronyd 服務的設定檔 /etc/chrony.conf 中有一行 makestep threshold limit，以下例子表示當前 3 次時間相差 1.0s，就跳躍式校時，以加速校時。

```
[root@kvm8 ~]# grep makestep /etc/chrony.conf
makestep 1.0 3
```

2. 將時間設定成不是今天的日期 (2019-11-30)。

```
[root@kvm8 ~]# date -s 2019-11-30
Sat Nov 30 00:00:00 CST 2019
```

3. makestep 立即校時。

```
[root@kvm8 ~]# chronyc makestep
200 OK
```

4. 再查時間，沒有更動 (問題待查)。

```
[root@kvm8 ~]# date
Sat Nov 30 00:00:07 CST 2019
```

5. 如果有安裝 ntpdate，使用 ntpdate 立即校時。

```
[root@kvm7 ~]# ntpdate server.deyu.wang
 9 Dec 13:47:38 ntpdate[1866]: step time server 192.168.122.1 offset
    827238.381751 sec
```

6. 校時成功。

```
[root@kvm8 ~]# date
Mon Dec  9 13:47:41 CST 2019
```

17.5 實機練習

1. 設定自動效時

   (a) 自動校時伺服器 server.deyu.wang

   (b) 確認 chronyd 服務是否開機啓動 (enable)？導向到 /root/chronyd-enabled。

   (c) 確認 chronyd 服務是否啓動 (active)？導向到 /root/chronyd-active

   (d) chronyc 查詢設定的伺服器是否出現？管線處理過濾 deyu，導向到 /root/chronyc-sources。

Chapter 18

Network File System, NFS

18.1 NFS 簡介

1. 讓遠端主機可以掛載使用本機的檔案系統。
2. 系統管理員可藉 NFS 整合網路上的服務器。例如：多台主機使用相同的 LDAP 提供的帳號，同一帳號當然使用相同的家目錄，若各主機上都各自建立每一用户的家目錄，則用户的資料將無法同步。此時使用 NFS 就可提供家目錄的整合。
3. RHEL 6 (CentOS 6) 支援 NFSv2, NFSv3, NFSv4。
4. NFS 需要 nfslock 及 rpcbind 服務。但 NFSv4 不再使用 rpcbind。

18.2 NFS Client 端設定

1. 如果沒有 nfs-utils，必須先安裝。

```
[root@kvm8 ~]# dnf install nfs-utils
```

2. showmount 查看 NFS 伺服器 deyu.wang 分享的目錄。

```
[root@kvm8 ~]# showmount -e deyu.wang
Export list for deyu.wang:
/rhome       192.168.122.0/24
/home/guests 192.168.122.0/24
```

3. 建立掛載 NFS 的掛載點 /home/guests，目錄已存在且目錄內沒有任何檔案或次目錄。

```
[root@kvm8 ~]# mkdir /home/guests
mkdir: cannot create directory ‘/home/guests’: File exists
[root@kvm8 ~]# ll /home/guests/
total 0
```

4. 手動掛載 NFS 於掛載點 /home/guests。

```
[root@kvm8 ~]# mount deyu.wang:/home/guests /home/guests
```

5. 查詢掛載狀況。

```
[root@kvm8 ~]# df -h /home/guests/
Filesystem              Size  Used Avail Use% Mounted on
deyu.wang:/home/guests   99G   61G   33G  65% /home/guests
```

6. /home/guests 目錄內出現 ldapuser1 及 ldapuser2 兩個目錄。

```
[root@kvm8 ~]# ll /home/guests
total 8
drwxr-xr-x. 2 2001 2001 4096 Nov 14  2016 ldapuser1
drwxr-xr-x. 2 2002 2002 4096 Aug 23  2016 ldapuser2
```

7. 將掛載的 NFS 目錄卸載

```
[root@kvm8 ~]# umount /home/guests
```

8. NFS 特殊的掛載參數

參數	預設	內容意義
fg/bg	fg	當執行掛載時，該掛載的行爲會在前景 (fg) 還背景 (bg) 執行。
soft/hard	hard	當兩者之間有任何一部主機離線，hard RPC 續的呼叫，直到對方恢復連線爲止；soft RPC time out 後『重複』呼叫，而非『持續』呼叫
intr	沒中斷	使用 hard 方式掛載若加上 intr，則當 RPC 呼叫中，該次呼叫可以被中斷。
rsize/wrize	1024	讀出 (rsize) 與寫入 (wsize) 的區塊大小 (b size)。

18.3 autofs 自動掛載

1. 爲何需要自動掛載 autofs？
   (a) 讓用户端在有使用到 NFS 檔案系統時才掛載；
   (b) 不需要事先建立掛載點；
   (c) 當 NFS 檔案系統使用完畢後，讓 NFS 自動卸載，以避免可能的 RPC 錯誤。
2. 如果沒安裝 autofs 套件，必須先安裝。

```
[root@kvm8 ~]# dnf install autofs
```

3. 掃瞄 NFS 伺服器分享的目錄，並瞭解是否可以使用 (showmount)；

```
[root@kvm8 ~]# showmount -e deyu.wang
Export list for deyu.wang:
/rhome       192.168.122.0/24
/home/guests 192.168.122.0/24
```

4. 確認先卸載 /home/guests。

```
[root@kvm8 ~]# #mount deyu.wang:/home/guests /home/guests
[root@kvm8 ~]# umount /home/guests/
```

5. 建立主要設定檔 /etc/auto.master
   (a) 格式：< 預設目錄 > < 資料對應檔 >
   (b) 預設目錄：用户端要使用 /home/guests/ldapuser1，會到資料對應檔中找次目錄 ldapuser1 的對應。
   (c) 資料對應檔的檔名是可以自行設定的，此例中使用 /etc/auto.dyw。

```
[root@kvm8 ~]# vim /etc/auto.master
/home/guests    /etc/auto.dyw
```

6. 資料對應檔格式：

```
[本地端目錄]  [-掛載參數]  [伺服器所提供的目錄]
```

本地端目錄 ：指的是在 /etc/auto.master 內指定的目錄之次目錄，本地端目錄 "*" 表示 /home/guests 下的任何次目錄。

-掛載參數 ：就是 rw,bg,soft 等參數，可有可無；

伺服器所提供的目錄 ：例如 deyu.wang:/home/guests/& 中 "&" 表示對應到 "*"，當使用到目錄 /home/guests/ldapuser1，則 "*" 及 "&" 即爲 ldapuser1。

7. 建立資料對應檔內的掛載資訊，若 NFS server 有限制版本爲 v3，則必須加入參數 -vers=3。

```
[root@kvm8 ~]# vim /etc/auto.dyw
[root@kvm8 ~]# cat /etc/auto.dyw
*  -vers=3  deyu.wang:/home/guests/&
```

8. 查看開機是否啓動 autofs？disabled 未啓動。

```
[root@kvm8 ~]# systemctl is-enabled autofs.service
disabled
```

9. 設定開機啓動 autofs，且立即啓動。

```
[root@kvm8 ~]# systemctl enable --now autofs.service
Created symlink /etc/systemd/system/multi-user.target.wants/autofs.
    service →
/usr/lib/systemd/system/autofs.service.
```

18.4 autofs 自動掛載測試一

1. 確認目前並沒有掛載任何來自 deyu.wang NFS 伺服器的目錄

```
[root@kvm8 ~]# df -h
Filesystem                Size  Used Avail Use% Mounted on
devtmpfs                  901M     0  901M   0% /dev
tmpfs                     915M     0  915M   0% /dev/shm
tmpfs                     915M  8.5M  907M   1% /run
tmpfs                     915M     0  915M   0% /sys/fs/cgroup
```

```
/dev/mapper/vg_kvm8-root  6.9G  1.5G  5.1G  22% /
/dev/vda1                 190M  126M   50M  72% /boot
tmpfs                     183M     0  183M   0% /run/user/0
```

2. 切換工作目錄到 /home/guests/ldapuser1，此時檔案系統自動掛載。

```
[root@kvm8 ~]# cd /home/guests/ldapuser1
[root@kvm8 ldapuser1]# pwd
/home/guests/ldapuser1
```

3. 查看 /home/guests/ldapuser1 是掛載在 912.168.122.1 的 /home/guests/ldapuser1。

```
[root@kvm8 ldapuser1]# df -h /home/guests/ldapuser1
Filesystem                            Size  Used Avail Use% Mounted on
192.168.122.1:/home/guests/ldapuser1   99G   61G   33G  65% /home/guests
    /ldapuser1
```

4. 查看 /home/guests/ldapuser1 掛載版本 3。

```
[root@kvm8 ~]# mount | grep ldap
deyu.wang:/home/guests/ldapuser1 on /home/guests/ldapuser1 type nfs
(rw,relatime,vers=3,rsize=1048576,wsize=1048576,namlen=255,hard,proto=
    tcp,
timeo=600,retrans=2,sec=sys,mountaddr=192.168.122.1,mountvers=3,
mountport=20048,mountproto=udp,local_lock=none,addr=192.168.122.1)
```

5. 退回家目錄。

```
[root@kvm8 ldapuser1]# cd
```

6. 卸載 /home/guests/ldapuser1。

```
[root@kvm8 ~]# umount /home/guests/ldapuser1
```

18.5 autofs 自動掛載測試二

1. 系統有一用户 ldapuser1，其家目錄爲 /home/guests/ldapuser1。

```
[root@kvm8 ~]# getent passwd ldapuser1
ldapuser1:x:1002:1002::/home/guests/ldapuser1:/bin/bash
```

2. 登入 ldapuser1，使用到家目錄 /home/guests/ldapuser1，此時檔案系統自動掛載。

```
[root@kvm8 ~]# su - ldapuser1
[ldapuser1@kvm8 ~]$ pwd
/home/guests/ldapuser1
```

3. 查看 /home/guests/ldapuser1 是掛載在 912.168.122.1 的 /home/guests/ldapuser1。

```
[ldapuser1@kvm8 ~]$ df -h /home/guests/ldapuser1/
Filesystem                              Size  Used Avail Use% Mounted on
192.168.122.1:/home/guests/ldapuser1    99G   61G   33G  65% /home/guests
    /ldapuser1
```

4. 登出 ldapuser1。

```
[ldapuser1@kvm8 ~]$ exit
logout
```

18.6 *NFS 掛載非常慢

1. 掛載很慢，但最後是掛載成功，往往是 NFS server 限制掛載版本，而 client 預設掛載版本掛載版本不成功，進而再嘗試其他版本的掛載。

2. 解決方式：先查 server 端的的設定，以下例子是限制不能使用 v2 及 v4。

```
[root@deyu ~]# grep RPCNFS /etc/sysconfig/nfs
RPCNFSDARGS="-N 2 -N 4"
```

3. Client 端查詢掛載的版本，確實爲 v3。

```
[root@kvm7 ~]# nfsstat -m
/home/guests/ldapuser1 from deyu.wang:/home/guests/ldapuser1
 Flags: rw,relatime,vers=3,rsize=1048576,wsize=1048576,\
namlen=255,hard,proto=tcp,timeo=600,retrans=2,sec=sys,\
mountaddr=192.168.122.1,mountvers=3,mountport=44465,\
mountproto=udp,local_lock=none,addr=192.168.122.1
```

4. Client 預設掛載版本爲 v4，如果 server 端不能更動，client 端自動掛載就必須指定掛載版本 -vers=3，這樣掛載就會快速。

5. 由於 NFS v4 已不使用 rpcbind 服務，所以若以 v4 版本掛載，就可以不管 rpcbind 服務是否啓動；反之，若不是以 v4 版本掛載，則要確定 client 端 rpcbind 服務是否啓動，否則會掛載不起來。

18.7 實機練習

1. 自動掛載設定

   (a) 查看 deyu.wang 分享的 NFS 目錄，看是否有 /home/guests？查詢結果導向到 /root/smount。

   (b) getent 查看系統中是否有 ldapuser1 帳號，導向到 /root/ldapuser，觀察其家目錄在哪？

   (c) 設定自掛載伺服器 deyu.wang:/home/guests 到本機的 /home/guests。

   (d) 掛載 NFS 版本爲 3。

   (e) 切換用户爲 ldapuser1，看是否有家目錄？

   (f) 確認 autofs 服務是否開機啓動 (enable)？導向到 /root/autofs-enabled。

   (g) 確認 autofs 服務是否啓動 (active)？導向到 /root/autofs-active。

Chapter 19

檔案搜尋

19.1 find 命令

1. find：功能強大的檔案搜尋指令。

```
[root@kvm7 ~]# find [PATH] [option] [action]
參數：
1. 與時間有關的參數：
   -atime n ：在 n 天之前的『一天之內』被 access 過的檔案；
   -ctime n ：在 n 天之前的『一天之內』被 change 過狀態的檔案；
   -mtime n ：在 n 天之前的『一天之內』被 modification 過的檔案；
   -newer file ：比檔案 file 還新的檔案。
2. 與使用者或群組名稱有關的參數：
   -uid n ：使用者的帳號 ID，UID=n 的檔案
   -gid n ：群組名稱的 ID，GID=n 的檔案
   -user name ：使用者帳號名稱爲 name 的檔案
   -group name：群組名稱爲 name 的檔案
   -nouser    ：檔案的擁有者不存在的檔案
   -nogroup   ：檔案的擁有群組不存在的檔案
3. 與檔案權限及名稱有關的參數：
   -name filename：檔案名稱爲 filename 的檔案；
   -size [+-]SIZE：比 SIZE 還要大(+)或小(-)的檔案。SIZE 的規格有：
                   c:byte， k:1024bytes M:1048576bytes。例如：-size +50k
                   。
   -type TYPE     ：檔案的類型爲 TYPE 的檔案，類型主要有：一般正規檔案 (f
      ),
                   裝置檔案 (b, c), 目錄 (d), 連結檔 (l), socket (s),
                   及 FIFO (p) 等屬性。
   -perm mode  ：檔案屬性『剛好等於』 mode 的檔案
   -perm -mode ：檔案屬性『必須包含所有 mode 的屬性』的檔案，權限數值大
      於等於 mode。
   -perm +mode ：已不支援。
4. 額外可進行的動作：
   -exec command {} \;  ：-exec 後再接指令 command 來處理搜尋的結果，
                          {} 爲找到的內容，
                          分號 ; 爲命令的結束符號，但要用反斜線 \ 跳脫。
   -print          ：將結果列印到螢幕上（預設動作）
```

2. 參數代換：xargs 可以讀入 stdin 的資料，並且以空白字元或斷行字元作爲分辨，將 stdin 的資料分隔成爲 arguments。

```
[root@linux ~]# xargs [-0epn] command
選項：
-0  ：如果輸入的 stdin 含有特殊字元，例如 `, \, 空白鍵等等字元時，這個 -0 參數
    可以將他還原成一般字元。
-e  ：EOF (end of file) 。當 xargs 分析到其後接的字串時，會停止繼續工作。
-p  ：在執行每個指令的 argument 時，都會詢問使用者；
-n  ：後面接次數，每次 command 指令執行時，要使用幾個參數。看範例三。
當 xargs 後面沒有接任何的指令時，預設是以 echo 來進行輸出。
```

19.2 找屬於某用户的檔案

1. 先找出系統中屬於用户 deyu8 的檔案，輸出除了標準輸出 (stdout) 及目錄外，還有標準錯誤 (stderr)。

```
[root@kvm8 ~]# find / -user deyu8
/var/spool/mail/deyu8
/usr/local/bin/garmin
/usr/lib64/libbryton.so.1
find: ??/proc/7636/task/7636/fd/7??: No such file or directory
find: ??/proc/7636/task/7636/fdinfo/7??: No such file or directory
find: ??/proc/7636/fd/8??: No such file or directory
find: ??/proc/7636/fdinfo/8??: No such file or directory
```

2. 建立要將檔案複製過去的目標目錄 /root/findresults。

```
[root@kvm8 ~]# mkdir /root/findresults
```

3. 使用管線命令及 xargs 將 find 命令的標準輸出一個一個送給 cp 命令做爲標準輸入，複製到目標目錄 /root/findresults，標準錯誤 (stderr) 會留在螢幕上，不會送給後面命令當 stdin。

```
[root@kvm8 ~]# find / -user deyu8 | xargs cp -t /root/findresults/
```

```
find: ??/proc/7654/task/7654/fd/6??: No such file or directory
find: ??/proc/7654/task/7654/fdinfo/6??: No such file or directory
find: ??/proc/7654/fd/7??: No such file or directory
find: ??/proc/7654/fdinfo/7??: No such file or directory
```

4. 查詢目標目錄 /root/findresults 內是不是有之前找到的那些檔案？

```
[root@kvm8 ~]# ll /root/findresults/
total 388
-rw-r-----. 1 root root      0 Dec  9 10:16 deyu8
-rw-r--r--. 1 root root     11 Dec  9 10:16 garmin
-rwxr-xr-x. 1 root root 391104 Dec  9 10:16 libbryton.so.1
```

5. 如果不想看到標準錯誤 (stderr)，可將其送到 /dev/null，再管線處理。

```
[root@kvm8 ~]# find / -user deyu8 2> /dev/null | xargs cp -t /root/
    findresults/
```

6. 使用 find 內建的 -exec 選項執行 cp 複製命令，將找到的內容複製製到 findre 目錄。

```
[root@kvm8 ~]# mkdir findre
[root@kvm8 ~]# find / -user deyu8 -exec cp {} findre \;
find: ‘/proc/105595/task/105595/fd/7’ : No such file or directory
find: ‘/proc/105595/task/105595/fdinfo/7’ : No such file or directory
find: ‘/proc/105595/fd/8’ : No such file or directory
find: ‘/proc/105595/fdinfo/8’ : No such file or directory
[root@kvm8 ~]# ll findre
total 336
-rw-r-----. 1 root root      0 Dec  8 19:35 deyu8
-rw-r--r--. 1 root root     11 Dec  8 19:35 garmin
-rwxr-xr-x. 1 root root 339328 Dec  8 19:35 libbryton.so.1
```

19.3 找特定大小的檔案

1. 找目錄 /usr 中大於 10M 的檔案。

```
[root@kvm8 ~]# find /usr -size +10M
/usr/lib64/libicudata.so.60.3
/usr/lib64/ceph/libceph-common.so.0
/usr/lib64/libmozjs-60.so.0.0.0
```

```
/usr/bin/podman
/usr/libexec/cni/dhcp
/usr/share/GeoIP/GeoLite2-City.mmdb
```

2. 找目錄 /usr/local 中小於 2k 的檔案。

```
[root@kvm8 ~]# find /usr/local/ -size -2k
/usr/local/bin/garmin
```

3. 找目錄 /usr 中介於 290k 到 300k 的檔案。

```
[root@kvm8 ~]# find /usr/ -size -300k -size +290k
/usr/lib64/libgfrpc.so.0.0.1
/usr/lib64/libnettle.so.6.5
/usr/lib64/ld-2.28.so
/usr/lib64/girepository-1.0/NM-1.0.typelib
/usr/lib64/libncursesw.so.6.1
/usr/lib64/libselinux.so.1
/usr/bin/qemu-ga
/usr/libexec/libvirt_leaseshelper
/usr/share/locale/ru/LC_MESSAGES/gnupg2.mo
/usr/share/locale/sv/LC_MESSAGES/NetworkManager.mo
/usr/share/doc/libxslt/ChangeLog
/usr/lib/locale/zh_SG.gbk/LC_CTYPE
/usr/lib/locale/zh_CN.gbk/LC_CTYPE
/usr/lib/firmware/ath10k/QCA9377/hw1.0/board-2.bin
/usr/lib/firmware/bnx2x/bnx2x-e2-7.8.2.0.fw
```

19.4 找特定屬性的檔案

1. 找目錄 /usr/local 中屬性是 644 的檔案。

```
[root@kvm8 ~]# find /usr/local/ -perm 644
/usr/local/bin/garmin
[root@kvm8 ~]# ll /usr/local/bin/garmin
-rw-r--r--. 1 deyu8 deyu8 11 Dec 25  2020 /usr/local/bin/garmin
```

2. 找目錄 /usr/local 中屬性包含 644 的檔案。

```
[root@kvm8 ~]# find /usr/local/ -perm -644
/usr/local/
```

```
/usr/local/lib64
/usr/local/bin
/usr/local/bin/garmin
/usr/local/src
/usr/local/etc
/usr/local/games
/usr/local/libexec
/usr/local/sbin
/usr/local/include
/usr/local/share
/usr/local/share/info
/usr/local/share/applications
/usr/local/share/man
/usr/local/share/man/man5
/usr/local/share/man/man3
/usr/local/share/man/man9
/usr/local/share/man/man6x
/usr/local/share/man/man3x
/usr/local/share/man/man1x
/usr/local/share/man/mann
/usr/local/share/man/man2x
/usr/local/share/man/man4x
/usr/local/share/man/man9x
/usr/local/share/man/man5x
/usr/local/share/man/man6
/usr/local/share/man/man8x
/usr/local/share/man/man8
/usr/local/share/man/man7x
/usr/local/share/man/man2
/usr/local/share/man/man4
/usr/local/share/man/man7
/usr/local/share/man/man1
/usr/local/lib
[root@kvm8 ~]# ll -d /usr/local/lib
drwxr-xr-x. 2 root root 4096 May 11  2019 /usr/local/lib
```

3. 找目錄 /usr 中屬性包含 SETUID 的檔案。

```
[root@kvm8 ~]# find /usr -perm -4000
/usr/bin/passwd
/usr/bin/sudo
/usr/bin/pkexec
/usr/bin/su
/usr/bin/umount
/usr/bin/newgrp
/usr/bin/chage
/usr/bin/mount
/usr/bin/crontab
/usr/bin/gpasswd
/usr/libexec/dbus-1/dbus-daemon-launch-helper
/usr/sbin/unix_chkpwd
/usr/sbin/mount.nfs
```

```
15 /usr/sbin/userhelper
   /usr/sbin/grub2-set-bootflag
17 /usr/sbin/pam_timestamp_check
   /usr/lib/polkit-1/polkit-agent-helper-1
19 [root@kvm8 ~]# ll /usr/bin/passwd
   -rwsr-xr-x. 1 root root 33600 Apr  7  2020 /usr/bin/passwd
```

4. 找目錄 /usr 中屬性包含 SETGID 的檔案。

```
   [root@kvm8 ~]# find /usr -perm -2000
2  /usr/bin/write
   /usr/libexec/openssh/ssh-keysign
4  /usr/libexec/utempter/utempter
   [root@kvm8 ~]# ll /usr/bin/write
6  -rwxr-sr-x. 1 root tty 21344 Apr 24  2020 /usr/bin/write
```

5. 搜尋 Linux 系統中 SGID 目錄：

```
   [root@kvm8 ~]# find / -perm -2000 2>/dev/null
2  /run/log/journal
   /run/log/journal/23dc31d09bc35ea4f7b3e127f22919fc
4  /usr/bin/write
   /usr/libexec/openssh/ssh-keysign
6  /usr/libexec/utempter/utempter
   /home/shared
```

6. 搜尋 Linux 系統中 SBIT 目錄：

```
1  [root@kvm8 ~]# find / -perm -1000 2>/dev/null
   /sys/fs/bpf
3  /dev/mqueue
   /dev/shm
5  /tmp
   /tmp/systemd-private-922af849fd5e4d4285a98af342213321-httpd.service-
       CiR6sP/tmp
7  /tmp/systemd-private-922af849fd5e4d4285a98af342213321-chronyd.service-
       EpvurN/tmp
   /var/tmp
9  /var/tmp/systemd-private-922af849fd5e4d4285a98af342213321-chronyd.
       service-QwbDlW/tmp
   /var/tmp/systemd-private-922af849fd5e4d4285a98af342213321-httpd.service-
       Fvgsvc/tmp
```

19.5 實機練習

1. 找用户檔案
   (a) 找系統中所有屬於 deyu8 的檔案
   (b) 建立目錄 /root/findresults
   (c) 將找到的檔案複製到 /root/findresults 目錄中
2. 找特定大小檔案
   (a) 找目錄 /usr/share 中大於 1M 的檔案，導向到 /root/find1
   (b) 找目錄 /usr/share 中小於 5M 的檔案，導向到 /root/find2
   (c) 找目錄 /usr/share 中小於 300k 且大於 250k 的檔案，導向到 /root/find3
3. 找特定屬性檔案
   (a) 找目錄 /etc/systemd 屬性包含 664 的檔案，導向到 /root/find4
   (b) 找系統中屬性包含 SUID 的檔案，導向到 /root/find5
   (c) 找系統中屬性包含 SGID 的檔案，導向到 /root/find6
   (d) 找系統中屬性包含 SBIT 的檔案，導向到 /root/find7

Chapter 20

Shell Script

20.1 前言

1. 何謂 Shell Script

   (a) 文字介面下讓使用者與系統溝通的一個工具介面。

   (b) 利用 shell 的功能所寫的一支『程式 (program)』。

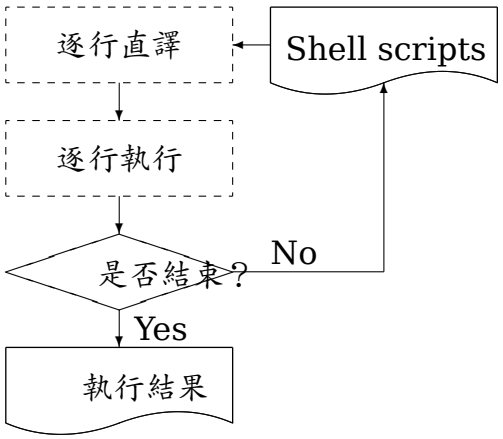
   (c) 將一些 shell 的語法與指令寫成純文字檔；

   (d) 可搭配正規表示法、管線命令與資料流重導向等功能；

   (e) shell script 更提供陣列、迴圈、條件與邏輯判斷等重要功能；

   (f) 使用者也可以直接以 shell 來撰寫程式。

2. Shell scripts 執行流程：



3. 爲什麼要學習 shell scripts?

   (a) 自動化管理的重要依據：自動處理分析主機狀態，若有問題才通知。

(b) 追蹤與管理系統的重要工作：Linux 系統的服務 (services) 啓動的介面，在目錄 /etc/init.d/ 下，所有的檔案都是 scripts。

(c) 簡單入侵偵測功能：主動分析系統登錄檔。

(d) 連續指令單一化：彙整在 command line 下達的連續指令。例如：/etc/rc.d/rc.local 裡的資料。

(e) 簡易的資料處理：處理數據資料的比對，文字資料的處理等。

(f) 跨平台支援與學習歷程較短：幾乎所有的 Unix Like 上面都可以跑 shell script 。

(g) shell script 是一項很好的系統管理工具，但數值運算的速度較慢，且使用的 CPU 資源較多。

4. Scripts 撰寫注意事項：

(a) 指令與參數間的多個空白會被忽略掉；

(b) 空白行及 [tab] 不會被理會；

(c) 讀到 [Enter] 符號，就開始執行該行命令；

(d) 一行的內容太多，可以使用 \[Enter] 來延伸至下一行；

(e) 加在 # 後面的字，全部被視爲註解。

20.2 執行 shell script

1. 寫好 shell script，例如：newfind.sh，內容只有一行 find 指令，找目錄 /usr 內大於 10M 的檔案。

```
[root@kvm8 ~]# vim newfind.sh
[root@kvm8 ~]# cat newfind.sh
find /usr/ -size +10M
```

2. 直接使用 sh 或 bash 執行腳本 newfind.sh。

```
[root@kvm8 ~]# bash newfind.sh
/usr/lib64/libicudata.so.60.3
/usr/lib64/ceph/libceph-common.so.0
/usr/lib64/libmozjs-60.so.0.0.0
/usr/bin/podman
/usr/libexec/cni/dhcp
/usr/share/GeoIP/GeoLite2-City.mmdb
```

3. 要獨立執行 newfind.sh 腳本，必須先增加其可執行屬性再執行。

```
[root@kvm8 ~]# ./newfind.sh
-bash: ./newfind.sh: Permission denied
[root@kvm8 ~]# chmod +x ./newfind.sh
[root@kvm8 ~]# ./newfind.sh
/usr/lib64/libicudata.so.60.3
/usr/lib64/ceph/libceph-common.so.0
/usr/lib64/libmozjs-60.so.0.0.0
/usr/bin/podman
/usr/libexec/cni/dhcp
/usr/share/GeoIP/GeoLite2-City.mmdb
```

4. newfind.sh 增加可執行的屬性後，執行結果沒問題，是因在 linux 的文字界面預設是使用 /bin/bash 執行腳本，如果是其他直譯程式，例如：python, curl 等腳本，也直接執行一定會出錯。

5. 腳本的第一行井號及驚嘆號#! 組成的字元組爲 shebang (hashbang)，其後面接的內容是直譯器指令。可以宣告此腳本使用的直譯器指令，就可以正確的直譯器指令執行腳本，

```
[root@kvm8 ~]# vim newfind.sh
[root@kvm8 ~]# cat newfind.sh
#!/bin/bash
find /usr/ -size +10M
```

20.3 可執行檔

1. 可執行檔放置目錄，使用文字界面執行檔案時，直接輸入指令名稱就可執行，完全不必輸入絕對路徑或相對路徑指明執行檔所在位置，是因爲環境變數 PATH 設定好執行檔可能的所在位置，系統會自動在這些目錄中搜尋。

```
[root@kvm8 ~]# echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin
[root@kvm8 ~]# echo ${PATH}
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin
```

2. 目前的目錄爲 /root，目錄內有 newfind.sh 可執行檔，但執行 newfind.sh 卻回應沒有此一命令，原因爲執行命令時，linux 系統只會在 PATH 變數包含的目錄中尋找指令。

```
[root@kvm8 ~]# pwd
/root
[root@kvm8 ~]# ll newfind.sh
-rwxr-xr-x. 1 root root 34 Oct 29 15:15 newfind.sh
[root@kvm8 ~]# newfind.sh
-bash: newfind.sh: command not found
```

3. 若要執行不是存在 PATH 變數指定的目錄的執行檔，必須明確指定路徑才可執行，以絕對路徑 /root/newfind.sh 或相對路徑./newfind.sh 都可以，其中. 表示目錄的位置。

```
[root@kvm8 ~]# /root/newfind.sh
/usr/lib64/libicudata.so.60.3
/usr/lib64/ceph/libceph-common.so.0
/usr/lib64/libmozjs-60.so.0.0.0
/usr/bin/podman
/usr/libexec/cni/dhcp
/usr/share/GeoIP/GeoLite2-City.mmdb
[root@kvm8 ~]# ./newfind.sh
/usr/lib64/libicudata.so.60.3
/usr/lib64/ceph/libceph-common.so.0
/usr/lib64/libmozjs-60.so.0.0.0
/usr/bin/podman
/usr/libexec/cni/dhcp
/usr/share/GeoIP/GeoLite2-City.mmdb
```

4. 若要讓所有用户都可執行 newfind.sh，必須將其移至環境變數 PATH 指定的目錄中，此例，將 newfind.sh 移至 /usr/loca/bin 目錄，就可以不指定執行檔位置直接執行。

```
[root@kvm8 ~]# mv newfind.sh /usr/local/bin/
[root@kvm8 ~]# newfind.sh
/usr/lib64/libicudata.so.60.3
/usr/lib64/ceph/libceph-common.so.0
/usr/lib64/libmozjs-60.so.0.0.0
/usr/bin/podman
/usr/libexec/cni/dhcp
/usr/share/GeoIP/GeoLite2-City.mmdb
[root@kvm8 ~]#
```

20.4 * 可執行檔別名

1. 如果執行檔已存在，只是執行時要加太長的參數很不方便，也可在用户登入時必須讀取的設定檔 /etc/bashrc 中加入別名。alias 設定在 /etc/bashrc 或 /etc/profile 兩個檔案 ssh 登入都會生效，但圖形界面開啓 gnome-terminal 不會讀取 /etc/profile，所以不建議放在 /etc/profile。

```
[root@kvm8 ~]# vim /etc/bashrc
[root@kvm8 ~]# tail -1 /etc/bashrc
alias qstat='/bin/ps -Ao pid,tt,user,fname,rsz'
```

2. 重新讀取 /etc/bashrc，別名 qstat 已生效。

```
[root@kvm8 ~]# source /etc/bashrc
[root@kvm8 ~]# alias
alias cp='cp -i'
alias egrep='egrep --color=auto'
alias fgrep='fgrep --color=auto'
alias grep='grep --color=auto'
alias l.='ls -d .* --color=auto'
alias ll='ls -l --color=auto'
alias ls='ls --color=auto'
alias mv='mv -i'
alias qstat='/bin/ps -Ao pid,tt,user,fname,rsz'
alias rm='rm -i'
alias which='alias | /usr/bin/which --tty-only --read-alias --show-dot
    --show-tilde'
```

3. 查看時過濾 qstat 更清楚。

```
[root@kvm8 ~]# alias | grep qstat
alias qstat='/bin/ps -Ao pid,tt,user,fname,rsz'
```

4. 直接執行看看。

```
[root@kvm8 ~]# qstat
  PID TT       USER     COMMAND    RSZ
    1 ?        root     systemd   7452
    2 ?        root     kthreadd     0
    3 ?        root     ksoftirq     0
    5 ?        root     kworker/     0
    6 ?        root     kworker/     0
    7 ?        root     migratio     0
    8 ?        root     rcu_bh       0
```

```
10     9 ?        root     rcuob/0      0
.....................................
12 12944 pts/0    root     ps         1276
```

5. 切換一般用户 deyu1，一樣可以執行 qstat，出現的結果不會一樣。

```
[root@kvm8 ~]# su - deyu1
2 Last login: Sun Aug  9 07:42:07 CST 2015 on pts/0
[deyu1@kvm8 ~]$ qstat
4   PID TT       USER     COMMAND    RSZ
    1 ?        root     systemd   7452
6   2 ?        root     kthreadd     0
    3 ?        root     ksoftirq     0
8   5 ?        root     kworker/     0
    6 ?        root     kworker/     0
10  7 ?        root     migratio     0
    8 ?        root     rcu_bh       0
12  9 ?        root     rcuob/0      0
.....................................
14 12973 pts/0    deyu1    ps         1280
```

6. 退出 deyu1。

```
[deyu1@kvm8 ~]$ exit
2 logout
```

20.5 實機練習

1. 寫腳本 newfind.sh
   (a) 第一行 shebang 宣告使用的直譯器指令
   (b) newfind.sh 所有人可執行
   (c) 找目錄 /usr 中小於 300k 且大於 250k 的檔案，導向到 /root/newfind.txt
   (d) 把 newfind.sh 放到 PATH 路徑可以找到的 /usr/local/bin 目錄，讓所有人可以不用指定路徑就可執行。
2. 寫腳本 newfind1.sh 找特定屬性檔案
   (a) 第一行 shebang 宣告使用的直譯器指令
   (b) 腳本包含以下三行命令

i. 找系統中屬性包含 SUID 的檔案

ii. 找系統中屬性包含 SGID 的檔案

iii. 找系統中屬性包含 SBIT 的檔案

(c) 把 newfind1.sh 放到 PATH 路徑可以找到的 /usr/bin 目錄，讓所有人可以不用指定路徑就可執行。

Chapter 21

檔案打包壓縮

21.1 檔案的壓縮

1. 壓縮檔案的附檔名

.bz2	bzip2 程式壓縮的檔案；
.gz	gzip 程式壓縮的檔案；
.xz	xz 程式壓縮的檔案；
.tar	tar 程式只打包的資料，沒有壓縮；
.tar.bz2	tar 程式打包的檔案，且經過 bzip2 壓縮。
.tar.gz (.tgz)	tar 程式打包的檔案，且經過 gzip 壓縮。
.tar.xz	tar 程式打包的檔案，且經過 xz 壓縮。

2. bzip2,bzcat：以 bzcat 直接讀取.bz2 壓縮檔案內容。

```
[root@dywHome2 test]# bzip2 [-d] filename
參數：
-d    :解壓縮
[root@dywHome2 test]# bzcat filename.bz2

#範例：
[dywang@dywOffice tmp]$ bzip2 -z test.txt
[dywang@dywOffice tmp]$ bzcat test.txt.bz2
echo Hello Latex!
[dywang@dywOffice tmp]$ bzip2 -d test.txt.bz2
```

3. gzip,zcat：以 zcat 直接讀取.gz 壓縮檔案內容。

```
[root@dywHome2 test]# gzip [-d] filename
參數：
-d    :解壓縮
```

```
-# --fast --best:# 爲數字，--fast 是 1 壓縮速度最快；--best 是 9 最佳壓縮比，預設是 6。
[root@dywHome2 test]# zcat filename.gz

#範例：
[dywang@dywOffice tmp]$ gzip test.txt
[dywang@dywOffice tmp]$ zcat test.txt.gz
echo Hello Latex!
[dywang@dywOffice tmp]$ gzip -d test.txt.gz
[dywang@dywOffice tmp]$ gzip -9 test.txt
```

21.2 檔案的壓縮與打包

1. tar 命令選項

```
[root@linux ~]# tar [-cxtzjvfpPN] 檔案與目錄 ....
參數：
-c  ：建立一個打包檔案(create)；
-x  ：解開一個打包檔案；
-t  ：查看 tarfile 內的檔案；(c/x/t 不可同時存在！)
-z  ：同時用 gzip 壓縮；
-j  ：同時用 bzip2 壓縮；
-J  ：同時用 xz 壓縮；
-v  ：壓縮的過程中顯示檔案；
-f  ：使用檔名，在 f 之後要立即接檔名！
-p  ：保留檔案的原來屬性（屬性不會依據使用者而變）
-P  ：保留絕對路徑；
-N  ：比後面接的日期(yyyy/mm/dd)還要新的才會被打包；
--exclude FILE：排除 FILE 。
```

2. 將檔案 txt 打包爲 txt.tar

```
[root@dywOffice tmp]# tar -cvf txt.tar txt
```

3. 將整個 /etc 目錄下的檔案全部打包成爲 /tmp/etc.tar

```
[root@dywOffice tmp]# tar -cvf /tmp/etc.tar /etc <== 僅打包，不壓縮
[root@dywOffice tmp]# tar -zcvf /tmp/etc.tar.gz /etc <== 打包後，以 gzip 壓縮
[root@dywOffice tmp]# tar -jcvf /tmp/etc.tar.bz2 /etc <== 打包後，以 bzip2 壓縮
[root@dywOffice tmp]# tar -Jcvf /tmp/etc.tar.xz /etc <== 打包後，以 xz 壓縮
```

4. 查閱 /tmp/etc.tar.gz 檔案內有那些檔案？

```
[root@dywOffice tmp]# tar -ztvf /tmp/etc.tar.gz
```

5. 將 /tmp/etc.tar.gz 檔案解壓縮

```
1 [root@dywOffice tmp]# tar -zxvf /tmp/etc.tar.gz
```

6. 只將 /tmp/etc.tar.gz 內的 etc/passwd 解開

```
1 [root@dywOffice tmp]# tar -zxvf /tmp/etc.tar.gz etc/passwd
```

7. 將 /home/csie/testdir 內的所有檔案備份 (打包壓縮)，並保存其絕對路徑

```
1 [root@dywOffice tmp]# tar -zcvPf testdir.tar.gz /home/csie/testdir
```

8. 將 /etc/ 內的所有檔案備份，並保存其權限

```
1 [root@dywOffice tmp]# tar -zcvpf /tmp/etc.tar.gz /etc
```

9. 在 /home 當中，比 2007/11/30 新的檔案才備份

```
1 [root@dywOffice tmp]# tar -N '2007/11/30' -zcvf home.tar.gz /home
```

10. 備份 /home ，但不要 /home/csie/tmp

```
1 [root@dywOffice tmp]# tar -zcvf home.tar.gz /home  --exclude /home/csie/
    tmp
```

21.3 實機練習

1. 將整個 /usr/local (由變數 tzdir 決定) 目錄整個打包壓縮成以下檔案：

(a) 壓縮方式 gzip，存成檔案 /root/tzfile.tgz。

(b) 壓縮方式 bzip2，存成檔案 /root/tzfile.tar.bz2。

(c) 壓縮方式 xz，存成檔案 /root/tzfile.tar.xz。

2. 不解開打包壓縮檔的情況下，查看以下檔案的內容，看是否與原始資料相同？

(a) /root/tzfile.tgz 查看結果導向到 /root/tzfile.tgz.t。

(b) /root/tzfile.tar.bz2 查看結果導向到 /root/tzfile.tar.bz2.t。

(c) /root/tzfile.tar.xz 查看結果導向到 /root/tzfile.tar.xz.t。

3. 解壓縮打包，查看解開的檔案是否與原始資料相同？

(a) 建立目錄 /root/mytar.gz

(b) 切換工作目錄到 /root/mytar.gz

(c) 解打包壓縮 /root/tzfile.tgz

(d) 建立目錄 /root/mytar.bz

(e) 切換工作目錄到 /root/mytar.bz

(f) 解打包壓縮 /root/tzfile.tar.bz2

(g) 建立目錄 /root/mytar.xz

(h) 切換工作目錄到 /root/mytar.xz

(i) 解打包壓縮 /root/tzfile.tar.xz

Chapter 22

Podman 簡介

22.1 前言

1. Pod Manager tool (podman) 是一個開源的容器運作工具，RHEL/CentOS 8 起，已用 podman 取代 docker daemon。
2. Podman 提供與 Docker 非常相似的功能，但 podman 不需要啓用任何 daemon 程序，而且可以在沒有 root 權限的情況下運作。
3. Podman 可以管理和運行任何符合 OCI（Open Container Initiative）規範的容器和容器鏡像。
4. Podman 提供了一個與 Docker 兼容的命令行前端來管理 Docker 鏡像。

22.2 Podman 安裝

1. CentOS 8 repository 內建 podman 套件，可直接以 dnf 安裝。

```
1 [root@kvm8 ~]# dnf install -y podman
```

22.3 Podman 命令

1. podman –help 查詢輔助說明

```
1 [root@kvm8 ~]# podman --help
```

2. podman 按兩次 TAB 鍵，提示 podman 可用的命令。

```
[root@kvm8 ~]# podman [TAB][TAB]
attach       exec         info         network      rm           top
auto-update  export       inspect      pause        rmi          umount
build        generate     kill         play         run          unmount
commit       healthcheck  load         pod          save         unpause
container    help         login        port         search       untag
cp           history      logout       ps           start        varlink
create       image        logs         pull         stats        version
diff         images       manifest     push         stop         volume
events       import       mount        restart      tag          wait
```

3. podman 每個命令也都可以使用 -help 查詢輔助說明，例如 pull 命令。

```
[root@kvm8 ~]# podman pull --help
Pull an image from a registry

Description:
  Pulls an image from a registry and stores it locally.

  An image can be pulled by tag or digest. If a tag is not specified,
  the image with the 'latest' tag is pulled.

Usage:
  podman pull [flags] IMAGE

Examples:
  podman pull imageName
  podman pull fedora:latest

Flags:
      --all-tags                All tagged images in the repository will
          be pulled
      --authfile string         Path of the authentication file.
                                Use REGISTRY_AUTH_FILE environment
                                    variable to override
      --cert-dir Pathname       Pathname of a directory containing TLS
          certificates and keys
      --creds Credentials       Credentials (USERNAME:PASSWORD) to
                                use for authenticating to a registry
      --disable-content-trust   This is a Docker specific option and is
          a NOOP
      --override-arch ARCH      Use ARCH instead of the architecture
                                of the machine for choosing images
      --override-os OS          Use OS instead of the running OS for
          choosing images
  -q, --quiet                   Suppress output information when pulling
       images
      --tls-verify              Require HTTPS and verify certificates
                                when contacting registries (default true
                                    )
```

22.4 Registry 設定

1. 安裝好 podman 後，podman 可以搜尋需要的 images 來啓動容器，提供 images 的 server 爲 registry。

2. podman 由 /etc/containers/registries.conf 設定檔決定搜尋的 registries 及方式。

   (a) [registries.search] 設定查詢的 registries：預設有 registry.access.redhat.com, registry.redhat.io, docker.io 三個伺服器。

   (b) [registries.insecure] 設定是否使用 http 不安全連線：若需要，則將 server 列到中括號中，例如 registries = ['registry.server']。

```
[deyu5@kvm8 ~]$ egrep -v '^(#| |$)' /etc/containers/registries.conf
[registries.search]
registries = ['registry.access.redhat.com', 'registry.redhat.io', '
    docker.io']
[registries.insecure]
registries = []
[registries.block]
registries = []
```

3. 爲提供 dyw linux 練習系統客製化 image，架設 registry server，主機名稱 registry.csie.cyut.edu.tw，只提供 https 連線，且需要帳號密碼登入。

4. 以 deyu5 帳號登入 kvm8，不能使用 su 或 su - 切換身份，否則無法執行 systemctl --user。

```
[root@kvm8 ~]# ssh deyu5@kvm8
deyu5@kvm8's password:
Last login: Thu Dec 10 21:42:54 2020 from 192.168.122.1
```

5. 一般用户若不使用系統預設的 registry 設定，可獨立在自己家目錄下建立設定檔 ~/.config/containers/registries.conf，以下 unqualified-search-registries 指定使用內網的 registry，並設定可以使用不安全的 http 連線。

```
[deyu5@kvm8 ~]$ cat ~/.config/containers/registries.conf
unqualified-search-registries = ['registry.csie.cyut.edu.tw']

[[registry]]
location = "registry.csie.cyut.edu.tw"
insecure = true
blocked = false
```

Chapter 23

*Podman Registry 架設

23.1 建置 Registry

1. podman 可搜尋下載可用的 images 做爲容器，本節爲建置一個 image registry server，提供 podman 下載 image 用，不是考試的範圍。

2. registry 儲存目錄 /opt/registry，先在此目錄下建立 auth, certs, data 三個次目錄，auth 儲存認證用 htpasswd 檔，certs 儲存認證憑證，data 儲存 images。

```
[root@registry ~]# mkdir -p /opt/registry/{auth,certs,data}
```

3. 使用 htpasswd 工具產生存取 registry 所需，含用户帳密的證書，本範例帳號爲 admin，密碼爲 xxxxxx。

   (a) 選項 b：由命令列提供密碼。

   (b) 選項 B：使用 bcrypt 加密密碼。

   (c) 選項 c：產生檔案。

```
[root@registry ~]# htpasswd -bBc /opt/registry/auth/htpasswd admin
    xxxxxx
Adding password for user admin
```

4. 產生 TLS 金鑰對，其中 CN 要設定爲 registry host 的主機名稱。

```
[root@registry ~]# openssl req -newkey rsa:4096 -nodes -sha256 \
-keyout /opt/registry/certs/domain.key -x509 -days 100 \
-out /opt/registry/certs/domain.crt  \
```

```
-subj '/C=TW/ST=Taiwan/L=CYUT/O=CSIE/CN=registry.csie.cyut.edu.tw'
Generating a RSA private key
....++++
.....++++
writing new private key to '/opt/registry/certs/domain.key'
-----
```

5. 因爲是使用 https 連線，client 必須有證書才能連線，本例 registry host 也當 client 做測試，所以將證書複製到證書存放目錄。

```
[root@registry ~]# cp /opt/registry/certs/domain.crt /etc/pki/ca-trust/
    source/anchors/
```

6. 更新信任憑證清單。

```
[root@registry ~]# update-ca-trust
```

7. 查詢信任憑證清單，registry.csie.cyut.edu.tw 已出現在清單中。

```
[root@registry ~]# trust list | grep csie.cyut
    label: registry.csie.cyut.edu.tw
```

23.2 啓用 Registry

1. docker.io/library/registry 是一個可以儲存 images 的 registry，下載並啓動成名爲 myregistry 的容器

2. 使用 host 443 port 導向容器的 5000 port，且需要帳密認證。

```
[root@registry ~]# podman run --name registry0 \
-p 443:5000 \
-v /opt/registry/data:/var/lib/registry:z \
-v /opt/registry/auth:/auth:z \
-e "REGISTRY_AUTH=htpasswd" \
-e "REGISTRY_AUTH_HTPASSWD_REALM=Registry Realm" \
-e REGISTRY_AUTH_HTPASSWD_PATH=/auth/htpasswd \
-v /opt/registry/certs:/certs:z \
-e "REGISTRY_HTTP_TLS_CERTIFICATE=/certs/domain.crt" \
-e "REGISTRY_HTTP_TLS_KEY=/certs/domain.key" \
-e REGISTRY_COMPATIBILITY_SCHEMA1_ENABLED=true \
```

```
12 -d docker.io/library/registry:latest
```

3. 使用 host 443 port 導向容器的 5000 port，但不需要帳密認證。

```
[root@registry ~]# podman run --name registry1 \
2 -p 443:5000 \
-v /opt/registry/data:/var/lib/registry:z \
4 -v /opt/registry/auth:/auth:z \
-v /opt/registry/certs:/certs:z \
6 -e "REGISTRY_HTTP_TLS_CERTIFICATE=/certs/domain.crt" \
-e "REGISTRY_HTTP_TLS_KEY=/certs/domain.key" \
8 -e REGISTRY_COMPATIBILITY_SCHEMA1_ENABLED=true \
-d docker.io/library/registry:latest
```

4. 啓用不用帳密且不加密的 registry，host 80 port 導向容器的 5000 port。

```
1 [root@registry ~]# podman run --name registry2 \
-p 80:5000 \
3 -v /opt/registry/data:/var/lib/registry:z \
-e REGISTRY_COMPATIBILITY_SCHEMA1_ENABLED=true \
5 -d docker.io/library/registry:latest
```

5. 防火牆開放 http 及 https 服務。

```
1 [root@registry ~]# firewall-cmd --permanent --add-service=http --add-
    service=https
success
3 [root@registry ~]# firewall-cmd --reload
success
```

23.3 上傳 Image 到 Registry

1. 連線 registry，查看 images，回應不允許無認證存取。

```
[root@registry ~]# curl https://registry.csie.cyut.edu.tw/v2/_catalog
2 {"errors":[{"code":"UNAUTHORIZED","message":"authentication required","
    detail":
[{"Type":"registry","Class":"","Name":"catalog","Action":"*"}]}]}
4 podman login -uadmin -pxxxxxx https://registry.csie.cyut.edu.tw
```

2. 使用帳號 admin 查詢 registry.csie.cyut.edu.tw，repositories 中沒有任何的 image。

```
[root@registry ~]# curl -u admin:xxxxxx https://registry.csie.cyut.edu.tw/v2/_catalog
{"repositories":[]}
```

3. 下載 docker.io/library/hello-world。

```
[root@registry ~]# podman pull docker.io/library/hello-world
Trying to pull docker.io/library/hello-world...
Getting image source signatures
Copying blob 0e03bdcc26d7 [--------------------------------------] 0.0b / 0.0b
Copying config bf756fb1ae done
Writing manifest to image destination
Storing signatures
bf756fb1ae65adf866bd8c456593cd24beb6a0a061dedf42b26a993176745f6b
```

4. 查詢本地端 images，出現 docker.io/library/hello-world。

```
[root@registry ~]# podman images
REPOSITORY                      TAG     IMAGE ID      CREATED        SIZE
docker.io/library/registry      latest  2d4f4b5309b1  5 months ago   26.8 MB
docker.io/library/hello-world   latest  bf756fb1ae65  11 months ago  20 kB
```

5. 登入 https://registry.csie.cyut.edu.tw

```
[root@registry ~]# podman login -uadmin -pxxxxxx https://registry.csie.cyut.edu.tw
Login Succeeded!
```

6. 上傳本地端的 docker.io/library/hello-world 到 https://registry.csie.cyut.edu.tw，且命名爲 hello。

```
[root@registry ~]# podman push hello-world registry.csie.cyut.edu.tw/hello
Getting image source signatures
Copying blob 9c27e219663c done
Copying config bf756fb1ae done
```

```
Writing manifest to image destination
Storing signatures
```

7. 再查詢 registry.csie.cyut.edu.tw，repositories 中出現 hello 的 image。

```
[root@registry ~]# curl -u admin:xxxxxx https://registry.csie.cyut.edu.
    tw/v2/_catalog
{"repositories":["hello"]}
```

8. 下載 docker.io/voxxit/rsyslog。

```
[root@registry ~]# podman pull docker.io/voxxit/rsyslog
Trying to pull docker.io/voxxit/rsyslog...
Getting image source signatures
Copying blob 1eae7a7426b0 done
Copying blob e6ff9ca2051f done
Copying blob 0db111ab9ef6 done
Copying blob d97f592f5e1e done
Copying config 8411a1edd4 done
Writing manifest to image destination
Storing signatures
8411a1edd4bb97aeae6bf9124cb00c66ff577ae68848e50704e9157263127aeb
```

9. 查詢本地端 images，出現 docker.io/voxxit/rsyslog。

```
[root@registry ~]# podman images
REPOSITORY                      TAG     IMAGE ID      CREATED         SIZE
docker.io/library/registry      latest  2d4f4b5309b1  5 months ago    26.8
    MB
docker.io/library/hello-world   latest  bf756fb1ae65  11 months ago   20
    kB
docker.io/voxxit/rsyslog        latest  8411a1edd4bb  2 years ago     20
    MB
```

10. 上傳本地端的 docker.io/voxxit/rsyslog 到 https://registry.csie.cyut.edu.tw，且命名爲 rsyslog。

```
[root@registry ~]# podman push rsyslog registry.csie.cyut.edu.tw/rsyslog
Getting image source signatures
Copying blob 22ff8195a9d5 done
Copying blob c9e8b5c053a2 done
Copying blob 36da72a6f23e done
Copying blob 9bc4ee67fdda done
```

```
7 Copying config 8411a1edd4 done
  Writing manifest to image destination
9 Storing signatures
```

11. 再查詢 registry.csie.cyut.edu.tw，repositories 中除了 hello，還有 rsyslog 。

```
1 [root@registry ~]# curl -u admin:xxxxxx https://registry.csie.cyut.edu.
    tw/v2/_catalog
{"repositories":["hello","rsyslog"]}
```

12. 上傳本地端的 localhost/dywrsyslog 到 https://registry.csie.cyut.edu.tw，且命名爲 dywrsyslog。

```
  [root@registry ~]# podman push localhost/dywrsyslog registry.csie.cyut.
    edu.tw/dywrsyslog
2 Getting image source signatures
  Copying blob 16759751ab17 done
4 Copying blob 22ff8195a9d5 skipped: already exists
  Copying blob 9bc4ee67fdda skipped: already exists
6 Copying blob c9e8b5c053a2 skipped: already exists
  Copying blob 36da72a6f23e skipped: already exists
8 Copying config ce4e2bdc8a done
  Writing manifest to image destination
10 Storing signatures
```

23.4 * 修改 Image

1. 使用 image docker.io/voxxit/rsyslog，啓動容器 myrsyslog。

```
  [root@registry ~]# podman run -d --name myrsyslog docker.io/voxxit/
    rsyslog
2 73a6e9dc4b45a826d7c45eb84a01a2a7929b6656a70126ad40abaec4d4d00773
```

2. 查看容器，myrsyslog 啓動正常。

```
  [root@registry ~]# podman ps
2 CONTAINER ID IMAGE       COMMAND        CREATED STATUS    PORTS
                  NAMES
  52020091237a ../registry /etc/.regis... 4 m...  Up 21 h..
    0.0.0.0:80->5000/tcp  registry80
```

```
4 73a6e9dc4b45 ../rsyslog                17 s... Up 17 s..
                          myrsyslog
  bd408d2ca881 ../registry /etc/.regis... 4 m...  Up 21 h..
     0.0.0.0:443->5000/tcp registry
```

3. 以 podman exec 使用容器 myrsyslog，執行 /bin/bash 取得互動式 shell。

```
1 [root@registry ~]# podman exec -it myrsyslog /bin/bash
  bash-4.4#
```

4. 進入容器 myrsyslog 後，進行以下修改：

   (a) 在 /root/.bashrc 加入 shell 提示變數 PS1 設定。

```
  bash-4.4# vi /root/.bashrc
2 bash-4.4# cat /root/.bashrc
  PS1="[\u@\h \W]\\$ "
```

   (b) 建立目錄 /var/log/journal/csie.cyut.log，並在 /etc/rsyslog.conf，增加 local3 任何層級的訊息，都記錄到 /var/log/journal/csie.cyut.log。

```
1 bash-4.4# mkdir /var/log/journal
  bash-4.4# vi /etc/rsyslog.conf
3 bash-4.4# grep local3 /etc/rsyslog.conf
  local3.*                        /var/log/journal/csie.cyut.log
```

   (c) 退出容器 myrsyslog。

```
  bash-4.4# exit
2 exit
```

5. 因爲修改容器 myrsyslog 的 rsyslog 設定檔 /etc/rsyslog.conf，必須重啓容器才會生效。

```
[root@registry ~]# podman restart myrsyslog
2 73a6e9dc4b45a826d7c45eb84a01a2a7929b6656a70126ad40abaec4d4d00773
```

23.5 *commit 建立 Image

1. 以修改過的容器 myrsyslog，建立作者 dywang，名稱 dywrsyslog 的新 image。

```
[root@registry ~]# podman commit -q --author "dywang" 73a6e9dc4b45
    dywrsyslog
ce4e2bdc8a208d26e9e3a9794bf9359e2bfe1c476b2d5dcb8f57ee5594738694
```

2. 查詢新建的 image dywrsyslog。

```
[root@registry ~]# podman images | grep dyw
localhost/dywrsyslog          latest  ce4e2bdc8a20  11 seconds ago  20
    MB
```

23.6 登入 Registry

1. 架設的 Regitry，也提供不需要帳密的 http server，本節要從此 registry 下載 rsyslog image 建立容器。要求：

2. 從 registry 下載 dywrsyslog image 建立容器。要求：

   (a) 使用非 root 帳號 deyu5

   (b) 容器名稱 mylog

   (c) 設定名爲 container-mylog 的 systemd 服務，開機自動啓動。

Chapter 24

Images 下載建立

24.1 登入搜尋

1. 使用 ssh 登入 deyu5@kvm8，ssh 與 su 認證方式不同，不能使用 su 或 su - 切換身份，否則無法執行 systemctl --user。

```
[root@kvm8 ~]# ssh deyu5@kvm8
deyu5@kvm8's password:
Last login: Thu Dec 10 21:42:54 2020 from 192.168.122.1
```

2. *podman search 查詢 registry.csie.cyut.edu.tw 所有 images, 因爲 registry 設定需要帳密認證，所以回應錯誤。

```
[deyu5@kvm8 ~]$ podman search registry.csie.cyut.edu.tw/
ERRO[0000] error getting search results from v2 endpoint
"registry.csie.cyut.edu.tw":
unable to retrieve auth token: invalid username/password:
unauthorized: authentication required
```

3. *podman 以 admin 登入 registry.csie.cyut.edu.tw。

```
[deyu5@kvm8 ~]$ podman login -uadmin -pxxxxxx https://registry.csie.cyut
    .edu.tw
Login Succeeded!
```

4. podman search 再查詢 rsyslog，找到 registry.csie.cyut.edu.tw 有兩個 images。

```
[deyu5@kvm8 ~]$ podman search rsyslog
INDEX  NAME                       DESCRIPTION STARS OFFICIAL AUTOMATED
edu.tw re...edu.tw/rsyslog        0
edu.tw re...edu.tw/dywrsyslog     0
```

24.2 下載 Images

1. 從 registry.csie.cyut.edu.tw 下載 rsyslog 及 dywrsyslog 兩個 images。

```
[deyu5@kvm8 ~]$ podman pull registry.csie.cyut.edu.tw/rsyslog
Trying to pull registry.csie.cyut.edu.tw/rsyslog...Getting image source
    signatures
Copying blob e36a18df25d4: 2.04 MiB / 2.04 MiB
    [============================] 5s
Copying blob 5bbc26867c5f: 1.98 MiB / 1.98 MiB
    [============================] 5s
Copying blob 864ad45e3300: 2.97 MiB / 2.97 MiB
    [============================] 5s
Copying blob 68a85f8ea16b: 815 B / 815 B
    [==================================] 5s
Copying config 8411a1edd4bb: 3.89 KiB / 3.89 KiB
    [============================] 0s
Writing manifest to image destination
Storing signatures

8411a1edd4bb97aeae6bf9124cb00c66ff577ae68848e50704e9157263127aeb

[deyu5@kvm8 ~]$ podman pull dywrsyslog
Trying to pull registry.csie.cyut.edu.tw/dywrsyslog...
Getting image source signatures
Copying blob 5d3ef8f14a83 done
Copying blob 68a85f8ea16b done
Copying blob e36a18df25d4 done
Copying blob 5bbc26867c5f done
Copying blob 864ad45e3300 done
Copying config ce4e2bdc8a done
Writing manifest to image destination
Storing signatures
ce4e2bdc8a208d26e9e3a9794bf9359e2bfe1c476b2d5dcb8f57ee5594738694
```

2. 查詢 images。

```
[deyu5@kvm8 ~]$ podman images
REPOSITORY          TAG     IMAGE ID      CREATED                 SIZE
re...tw/dywrsyslog  latest  ce4e2bdc8a20  Less than a second ago  20 MB
re...tw/rsyslog     latest  8411a1edd4bb  2 years ago             20 MB
```

24.3 build 建立 Image

1. * 把使用的原始 image 及要修改的內容依 Dockerfile 語法寫入檔案，以下例子使用 registry.csie.cyut.edu.tw/rsyslog image 在其 /etc/rsyslog.conf 增加一行，將 local4.* 的訊息記錄到 /var/log/journal/podmanfile.log。

```
[deyu5@kvm4 ~]# vim Podmanfile
[deyu5@kvm4 ~]# cat Podmanfile
FROM registry.csie.cyut.edu.tw/rsyslog
MAINTAINER dywang@csie.cyut.edu.tw

RUN echo 'local4.*     /var/log/journal/podmanfile.log' >> /etc/rsyslog.
    conf
```

2. * 使用本地端的 Podmanfile 建立名爲 pmrsyslog 的 image。

```
[deyu5@kvm4 ~]$ podman build --tag=pmrsyslog --file Podmanfile
STEP 1/3: FROM registry.csie.cyut.edu.tw/rsyslog
STEP 2/3: MAINTAINER dywang@csie.cyut.edu.tw
--> Using cache 54
    f5812ec2a59f41288c5f32d648b938556489f89da12ff4af2541947e3cb44c
--> 54f5812ec2a
STEP 3/3: RUN echo 'local4.*     /var/log/journal/podmanfile.log' >> /
    etc/rsyslog.conf
--> Using cache 9
    fe0c3653d3f6b0e7d83ab95939c3abb9885b5068d03097beb2f443857982d9b
COMMIT pmrsyslog
--> 9fe0c3653d3
Successfully tagged localhost/pmrsyslog:latest
Successfully tagged localhost/myrsyslog:latest
9fe0c3653d3f6b0e7d83ab95939c3abb9885b5068d03097beb2f443857982d9b
```

3. 使用遠端 URL 的 Podmanfile 建立名爲 myrsyslog 的 image。

```
[deyu5@kvm4 ~]$ podman build --tag=myrsyslog \
http://dywang.csie.cyut.edu.tw/materials/Podmanfile
STEP 1/3: FROM registry.csie.cyut.edu.tw/rsyslog
Trying to pull registry.csie.cyut.edu.tw/rsyslog:latest...
Getting image source signatures
Copying blob 8104e5e36684 done
Copying blob bc8ac9851443 done
```

```
Copying blob 745ce1c08bfe done
Copying blob 718cd6a65a88 done
Copying config 8411a1edd4 done
Writing manifest to image destination
Storing signatures
STEP 2/3: MAINTAINER dywang@csie.cyut.edu.tw
--> 54f5812ec2a
STEP 3/3: RUN echo 'local4.*     /var/log/journal/podmanfile.log' >> /
    etc/rsyslog.conf
COMMIT myrsyslog
--> 9fe0c3653d3
Successfully tagged localhost/myrsyslog:latest
9fe0c3653d3f6b0e7d83ab95939c3abb9885b5068d03097beb2f443857982d9b
```

4. 查詢 images，本地端已建立了 pmrsyslog 及 myrsyslog 兩個 images。

```
[deyu5@kvm4 ~]$ podman images
REPOSITORY                  TAG         IMAGE ID      CREATED
    SIZE
localhost/pmrsyslog         latest      9fe0c3653d3f  21 minutes ago  20
    MB
localhost/myrsyslog         latest      9fe0c3653d3f  21 minutes ago  20
    MB
```

24.4 實機練習

1. 以 deyu5 帳號在本地端建立以下 podmaan 容器 images。

   (a) 下載 registry.csie.cyut.edu.tw 的 dywrsyslog。

   (b) 使用遠端的 Podmanfile，不要做任何修改，建立名爲 prsyslog (查看變數) 的 image。

   (c) 查詢 images，確認 dywrsyslog 及 prsyslog 是否建立？

Chapter 25

開機啓用 Podman 容器

25.1 *Systemd 啓動容器

1. podman 使用 image dywrsyslog，建立一個名 mylog 且背景執行的容器。

```
[deyu5@kvm8 ~]$ podman create --name mylog dywrsyslog
839bc31c9b83361328c7c0cd71283552d45a6e275d7b83bf09b37c1de4c7ccc1
```

2. 查詢容器 mylog 已建立，狀態爲 Created，不是執行中。

```
[deyu5@kvm8 ~]$ podman ps -a --format "{{.ID}} {{.Image}} {{.Status}}
    {{.Names}}"
85b5a03a3f13 docker.io/eboraas/apache-php:latest Up 5 hours ago web
384ce2ad0d31 docker.io/library/mariadb:latest Up 5 hours ago mdb
839bc31c9b83 registry.csie.cyut.edu.tw/dywrsyslog:latest Created mylog
```

3. 建立服務設定檔儲存目錄.config/systemd/user/，並切換工作目錄到該目錄。

```
[deyu5@kvm8 ~]$ mkdir -p .config/systemd/user/
[deyu5@kvm8 ~]$ cd .config/systemd/user/
```

4. podman 爲容器 mylog 產生服務的設定檔，--files 選項自動儲存成 container-mylog.service。

```
[deyu5@kvm8 user]$ podman generate systemd --name mylog --files
```

5. 查看產生的 container-mylog.service。

```
[deyu5@kvm8 user]$ cat container-mylog.service
# container-mylog.service
# autogenerated by Podman 3.2.3
# Fri Nov 12 15:31:40 CST 2021

[Unit]
Description=Podman container-mylog.service
Documentation=man:podman-generate-systemd(1)
Wants=network.target
After=network-online.target
RequiresMountsFor=/run/user/1000/containers

[Service]
Environment=PODMAN_SYSTEMD_UNIT=%n
Restart=on-failure
TimeoutStopSec=70
ExecStart=/usr/bin/podman start mylog
ExecStop=/usr/bin/podman stop -t 10 mylog
ExecStopPost=/usr/bin/podman stop -t 10 mylog
PIDFile=/run/user/1000/containers/overlay-containers/..../userdata/
    conmon.pid
Type=forking

[Install]
WantedBy=multi-user.target default.target
```

6. 重新載入服務。

```
[deyu5@kvm8 ~]$ systemctl --user daemon-reload
```

7. 設定開機啓動 container-mylog 服務，且現在啓動。

```
[deyu5@kvm8 ~]$ systemctl --user enable --now container-mylog.service
Created symlink /home/deyu5/.config/systemd/user/multi-user.target.wants
    /container-mylog.service
→ /home/deyu5/.config/systemd/user/container-mylog.service.
Created symlink /home/deyu5/.config/systemd/user/default.target.wants/
    container-mylog.service
→ /home/deyu5/.config/systemd/user/container-mylog.service.
```

8. 查詢 container-mylog 服務狀態，成功啓動。

```
[deyu5@kvm8 user]$ systemctl --user status container-mylog.service
  container-mylog.service - Podman container-mylog.service
   Loaded: loaded (/home/deyu5/.config/systemd/user/container-mylog.
       service; en>
```

```
   Active: active (running) since Fri 2021-11-12 15:33:31 CST; 38s ago
     Docs: man:podman-generate-systemd(1)
  Process: 15102 ExecStart=/usr/bin/podman start mylog (code=exited,
      status=0/S>
 Main PID: 15119 (conmon)
   CGroup: /user.slice/user-1000.slice/user@1000.service/container-mylog
       .service
           15113 /usr/bin/fuse-overlayfs -o ,lowerdir=/home/deyu5/.
             local/shar>
           15115 /usr/bin/slirp4netns --disable-host-loopback --mtu
             =65520 --e>
           15119 /usr/bin/conmon --api-version 1 -c 839
             bc31c9b83361328c7c0cd7>
           839
             bc31c9b83361328c7c0cd71283552d45a6e275d7b83bf09b37c1de4c7ccc1

             15129 rsyslogd -n
```

9. podman 查詢容器狀態，mylog 成功啓動。

```
[deyu5@kvm8 user]$ podman ps --format "{{.Names}} {{.Image}} {{.Status}}
    "
web docker.io/eboraas/apache-php:latest Up 5 hours ago
mdb docker.io/library/mariadb:latest Up 5 hours ago
mylog registry.csie.cyut.edu.tw/dywrsyslog:latest Up About a minute ago
```

10. 再確認 container-mylog 服務是否設定開機啓動。

```
[deyu5@kvm8 ~]$ systemctl --user is-enabled container-mylog.service
enabled
```

11. 以上的 systemd 設定是在登入 deyu5 後，才會啓動 container-mylog 服務，若要重開機後不管有沒有登入 deyu5，都啓動 container-mylog，則必須使用控制 systemd 登入管理命令 loginctl，啓用「用户保留登入狀態」，系統會在啓動後自動爲用户生出用户管理器，並且在用户登出後保持運作，deyu5 啓用保持登入狀態。

```
[deyu5@kvm8 user]$ loginctl enable-linger deyu5
```

12. deyu5 啓用保持登入狀態後，會建立 /var/lib/systemd/linger/deyu5 檔。

```
[deyu5@kvm8 user]$ ll /var/lib/systemd/linger
total 0
-rw-r--r--. 1 root root 0 Nov 12 15:36 deyu5
```

25.2 * 刪除容器

1. 之前啓動的 container-mylog.service 服務，沒有掛載 host 上的目錄。若現要實現掛載 host 上的目錄到 container mylog，要先停止 container-mylog 服務，並刪除容器，重新啓用容器。

2. 取消開機啓用 container-mylog 服務，且馬上停用。

```
[deyu5@kvm8 user]$ systemctl --user disable --now container-mylog.
    service
Removed /home/deyu5/.config/systemd/user/multi-user.target.wants/
    container-mylog.service.
Removed /home/deyu5/.config/systemd/user/default.target.wants/container-
    mylog.service.
```

3. 強制移除容器 mylog。

```
[deyu5@kvm8 user]$ podman rm mylog -f
839bc31c9b83361328c7c0cd71283552d45a6e275d7b83bf09b37c1de4c7ccc1
```

25.3 容器掛載儲存空間

1. 之前啓動的 container-mylog.service 服務，沒有掛載 host 上的目錄。

2. 實現掛載 host 上的目錄到容器，先建立要被掛載的目錄 container_journal。

```
[deyu5@kvm8 ~]$ mkdir container_journal
```

3. podman 使用 image dywrsyslog，建立一個名 mylog。

```
--privileged 給 container 權限，否則無法存取掛載的主機目錄。
-v host.dir:container.dir:Z 掛載主機的目錄到容器的目錄，
```

```
3    掛載的 :Z 選項，要 podman 重新 relabel 掛載目錄，以符合容器內 SELinux
         label。

5 [deyu5@kvm8 ~]$ podman create \
--name mylog \
7 --privileged \
--volume /home/deyu5/container_journal/:/var/log/journal/:Z \
9 dywrsyslog
be2df97e15bef589f73d5ae329134825dba8ff293a8603709ddb1fe4ac68c56e
```

4. podman ps 查看容器，沒有出現任何正在運作的容器。

```
[deyu5@kvm8 ~]$ podman ps
2 CONTAINER ID  IMAGE       COMMAND     CREATED     STATUS      PORTS
          NAMES
```

5. podman ps -all 查看容器，可以看到產生的 mylog 容器。

```
[deyu5@kvm8 ~]$ podman ps --all
2 CONTAINER ID  IMAGE                  COMMAND CREATED     STATUS PORTS
    NAMES
56c9eb2355a3  .../dywrsyslog:latest          48s Created        mylog
```

25.4 開機啓動容器

1. 建立服務設定檔儲存目錄.config/systemd/user/，並切換工作目錄到該目錄。

```
1 [deyu5@kvm8 ~]$ mkdir -p .config/systemd/user/
[deyu5@kvm8 ~]$ cd .config/systemd/user/
```

2. podman 爲容器 mylog 產生服務的設定檔，--files 選項自動儲存成 container-mylog.service。

```
[deyu5@kvm8 user]$ podman generate systemd --name mylog --files
2 /home/deyu5/.config/systemd/user/container-mylog.service
```

3. 查看產生的 container-mylog.service。

```
[deyu5@kvm8 user]$ ll
total 12
-rw-r--r--. 1 deyu5 deyu5  691 Nov 12 15:53 container-mylog.service
drwxr-xr-x. 2 deyu5 deyu5 4096 Nov 12 15:33 default.target.wants
drwxr-xr-x. 2 deyu5 deyu5 4096 Nov 12 15:33 multi-user.target.wants
```

4. 重新載入服務。

```
[deyu5@kvm8 user]$ systemctl --user daemon-reload
```

5. 設定開機啓動 container-mylog 服務，且現在啓動。

```
[deyu5@kvm8 user]$ systemctl --user enable --now container-mylog.service
Created symlink /home/deyu5/.config/systemd/user/multi-user.target.wants
    /container-mylog.service
→ /home/deyu5/.config/systemd/user/container-mylog.service.
Created symlink /home/deyu5/.config/systemd/user/default.target.wants/
    container-mylog.service
→ /home/deyu5/.config/systemd/user/container-mylog.service.
```

6. 查詢 container-mylog 服務狀態。

```
[deyu5@kvm8 user]$ systemctl --user status container-mylog.service
  container-mylog.service - Podman container-mylog.service
   Loaded: loaded (/home/deyu5/.config/systemd/user/container-mylog.
       service; en>
   Active: active (running) since Fri 2021-11-12 15:57:38 CST; 27s ago
     Docs: man:podman-generate-systemd(1)
  Process: 15708 ExecStart=/usr/bin/podman start mylog (code=exited,
      status=0/S>
 Main PID: 15725 (conmon)
   CGroup: /user.slice/user-1000.slice/user@1000.service/container-mylog
       .service
           15716 /usr/bin/fuse-overlayfs -o ,lowerdir=/home/deyu5/.
             local/shar>
           15720 /usr/bin/slirp4netns --disable-host-loopback --mtu
             =65520 --e>
           15725 /usr/bin/conmon --api-version 1 -c
             be2df97e15bef589f73d5ae32>

             be2df97e15bef589f73d5ae329134825dba8ff293a8603709ddb1fe4ac68c56e

             15735 rsyslogd -n
```

7. 以上的 systemd 設定是在登入 deyu5 後，才會啓動 container-mylog 服務，若要重開機後不管有沒有登入 deyu5，都啓動 container-mylog，則必須使用控制 systemd 登入管理命令 loginctl，啓用「用户保留登入狀態」，系統會在啓動後自動爲用户生出用户管理器，並且在用户登出後保持運作，deyu5 啓用保持登入狀態。

8. RHELv9 預設一般用户無法啓用 linger。

```
[deyu5@kvm8 user]$ loginctl enable-linger deyu5
Could not enable linger: Access denied
```

9. 使用 root 啓用 linger deyu5。

```
[deyu5@kvm8 ~]$ su -
Password:
Last login: Thu Nov 24 10:04:13 CST 2022 from 192.168.122.1 on pts/0
[root@kvm8 ~]# loginctl enable-linger deyu5
```

10. deyu5 啓用保持登入狀態後，會建立 /var/lib/systemd/linger/deyu5 檔。

```
[root@kvm8 ~]# ll /var/lib/systemd/linger/
total 0
-rw-r--r--. 1 root root 0 Nov 24 11:04 deyu5
```

11. 退出 root，回到 deyu5。

```
[root@kvm8 ~]# exit
logout
[deyu5@kvm8 ~]$
```

25.5 空間掛載測試

1. 使用 dywmylog 容器，並執行 /bin/bash 命令，取得互動命令界面，查看 /var/log/journal 目錄下的 2 個.journal 檔。

```
[deyu5@kvm8 ~]$ podman exec -it mylog /bin/bash
[root@1f73cadc2a3a /]# ls -l /var/log/journal/
total 16392
```

```
-rw-r-----    1 root     root       8388608 Jan  9 03:08 system.journal
-rw-r-----    1 root     root       8388608 Jan  9 03:07 user-2204.
   journal
```

2. dywrsyslog 容器 image，已設定 local3 的任何訊息，會寫到 /var/log/journal 目錄中的 csie.cyut.log，所以執行 looger -p local3.info 測試看看？

```
[root@1f73cadc2a3a /]# logger -p local3.info 'dywmylog test'
```

3. /var/log/journal 目錄中新增 csie.cyut.log，且檔案內容是 looger 寫入的'dywmylog test'。

```
[root@1f73cadc2a3a /]# ls -l /var/log/journal/
total 16396
-rw-r--r--    1 root     root            66 Apr 30 00:12 csie.cyut.log
-rw-r-----    1 root     root       8388608 Jan  9 03:08 system.journal
-rw-r-----    1 root     root       8388608 Jan  9 03:07 user-2204.
   journal
[root@1f73cadc2a3a /]# cat /var/log/journal/csie.cyut.log
2021-04-30T00:12:36.338653+00:00 1f73cadc2a3a root: dywmylog test
```

4. 退出 dywmylog 容器。

```
[root@1f73cadc2a3a /]# exit
exit
```

5. 查看 dywmylog 容器掛載的外部儲存空間 container_journal，一樣新增了 csie.cyut.log 檔案，內容也是'dywmylog test'。

```
[deyu5@kvm8 ~]$ cat container_journal/csie.cyut.log
2021-04-30T00:12:36.338653+00:00 1f73cadc2a3a root: dywmylog test
```

25.6 開機啓動容器檢查

1. kvm8 重開機

```
[root@kvm8 ~]# #reboot
```

2. 以 root 身份重新登入 kvm8。

```
[root@deyu ~]# ssh kvm8
Last login: Thu Dec 17 19:51:34 2020 from 192.168.122.1
```

3. deyu5 還沒登入前，查看是否有以 deyu5 啓動的 mylog 程序？

```
[root@kvm8 ~]# ps -u deyu5
  PID TTY          TIME CMD
  786 ?        00:00:00 systemd
 1104 ?        00:00:00 (sd-pam)
 1514 ?        00:00:00 podman pause
 1608 ?        00:00:00 slirp4netns
 1644 ?        00:00:00 fuse-overlayfs
 1666 ?        00:00:00 conmon
 1727 ?        00:00:00 rsyslogd
```

4. 以 deyu5 身份登入

```
[root@kvm8 ~]# ssh deyu5@localhost
deyu5@localhost's password:
Last login: Thu Dec 17 19:52:58 2020 from 192.168.122.8
```

25.7 *podman 版本問題

1. 以 deyu5 身份重新登入 kvm8。

```
[root@kvm8 ~]# ssh deyu5@kvm8
Last login: Fri Dec 25 18:43:39 2020 from 192.168.122.8
```

2. 查看 container-mylog.service 的 [Install] 設定，安裝在 muti-user.target。

```
[deyu5@kvm8 ~]$ tail -2 .config/systemd/user/container-mylog.service
[Install]
WantedBy=multi-user.target
```

3. 設定 container-mylog 服務開機啓動，且馬上啓動，安裝產生連結 /home/deyu5/.config/systemd/user/multi-user.target.wants/container-mylog.service 到 /home/deyu5/.config/systemd/user/container-mylog.service。

```
[deyu5@kvm8 ~]$ systemctl --user enable --now container-mylog.service
Created symlink /home/deyu5/.config/systemd/user/
multi-user.target.wants/container-mylog.service
 → /home/deyu5/.config/systemd/user/container-mylog.service.
```

4. 查看容器 mylog 啓動正常。

```
[deyu5@kvm8 ~]$ podman ps
CONTAINER ID  IMAGE                 COMMAND  CREATED     STATUS
    PORTS  NAMES
a615b1ee0585  registry..../rsyslog           15 hrs ago  Up 2 m ago
          mylog
```

5. 登出 deyu5，重開機 kvm8。

```
[deyu5@kvm8 ~]$ exit
logout
Connection to kvm8 closed.
[root@kvm8 ~]# reboot
Connection to kvm8 closed by remote host.
Connection to kvm8 closed.
```

6. 再以 root 登入 kvm8，查看 deyu5 程序，有 systemd，但沒有啓動的容器。

```
[root@deyu ~]# ssh kvm8
Last login: Sat Dec 26 09:51:13 2020 from 192.168.122.1
[root@kvm8 ~]# ps -u deyu5
    PID TTY          TIME CMD
   1004 ?        00:00:00 systemd
   1044 ?        00:00:00 (sd-pam)
```

7. 再以 deyu5 登入 kvm8，查看 container-mylog 服務狀態，並沒啓動。

```
[root@kvm8 ~]# ssh deyu5@kvm8
Last login: Sat Dec 26 09:38:47 2020 from 192.168.122.8
[deyu5@kvm8 ~]$ systemctl --user status container-mylog.service
  container-mylog.service - Podman container-mylog.service
```

```
   Loaded: loaded (/home/deyu5/.config/systemd/user/container-mylog.
       service; en>
   Active: inactive (dead)
     Docs: man:podman-generate-systemd(1)
```

8. 解決方法是在 container-mylog.service 的 [Install] 設定，增加 default.target。

```
[deyu5@kvm8 ~]$ tail -2 .config/systemd/user/container-mylog.service
[Install]
WantedBy=multi-user.target default.target
```

9. 重新載入服務設定。

```
[deyu5@kvm8 ~]$ systemctl --user daemon-reload
```

10. 再設定開機啓動且馬上啓動，此時又新增了連結 /home/deyu5/.config/systemd/user/default.target.wants/container-mylog.service 到 /home/deyu5/.config/systemd/user/container-mylog.service。

```
[deyu5@kvm8 ~]$ systemctl --user enable --now container-mylog.service
Created symlink /home/deyu5/.config/systemd/user/
default.target.wants/container-mylog.service
 → /home/deyu5/.config/systemd/user/container-mylog.service.
```

11. 登出 deyu5，重開機 kvm8。

```
[deyu5@kvm8 ~]$ exit
logout
Connection to kvm8 closed.
[root@kvm8 ~]# reboot
Connection to kvm8 closed by remote host.
Connection to kvm8 closed.
```

12. 再以 root 登入 kvm8，查看 deyu5 程序，有 systemd，也有啓動的容器程序 rsyslogd。

```
[root@deyu ~]# ssh kvm8
Last login: Sat Dec 26 09:53:36 2020 from 192.168.122.1
```

```
[root@kvm8 ~]# ps -u deyu5
    PID TTY          TIME CMD
    989 ?        00:00:00 systemd
   1034 ?        00:00:00 (sd-pam)
   1133 ?        00:00:00 podman pause
   1237 ?        00:00:00 slirp4netns
   1263 ?        00:00:00 fuse-overlayfs
   1280 ?        00:00:00 conmon
   1325 ?        00:00:00 rsyslogd
```

13. 目前版本是 1.6.4，會有此問題，podman 1.9.3 以上版本已修正。

```
[root@kvm8 ~]# podman --version
podman version 1.6.4
```

25.8 實機練習

1. 使用 dywrsyslog image 新增設定容器
   (a) 基本要求
      i. 名稱 mylog
      ii. 設定只有 deyu5 帳號能以 systemd 服務啓動容器
      iii. 容器啓動的服務名稱爲 container-mylog
      iv. 開機但未登入 deyu5 也要自動啓動 container-mylog 服務
   (b) 容器掛載儲存空間
      i. 在 deyu5 的家目錄新增名爲 cjournal 的次目錄
      ii. cjournal 必須掛載於 mylog 容器的 /var/log/journal
   (c) 容器測試
      i. 在 mylog 容器執行 logger -p local3.info 'dywmylog test'
      ii. 在 mylog 容器的 /var/log/journal 目錄中是否新增 csie.cyut.log，且檔案內容是 loger 寫入的'dywmylog test'？
      iii. 退出 mylog 容器，查看 deyu5 家目錄中的 cjournal 次目錄中是否一樣新增了 csie.cyut.log 檔案，內容也是'dywmylog test'。
2. 使用上一章以 podmanfile 產生的 image，localhost/prsyslog，新增設定容器。

(a) 基本要求

i. 名稱 podlog

ii. 設定只有 deyu5 帳號能以 systemd 服務啓動容器

iii. 容器啓動的服務名稱爲 container-podlog

iv. 開機但未登入 deyu5 也要自動啓動 container-podlog 服務

(b) 容器掛載儲存空間

i. 在 deyu5 的家目錄新增名爲 pjournal 的次目錄

ii. pjournal 必須掛載於 podlog 容器的 /var/log/journal

(c) 容器測試

i. 在 podlog 容器執行 logger -p local4.info 'podlog test'

ii. 在 podlog 容器的 /var/log/journal 目錄中是否新增 podmanfile.log，且檔案內容是 looger 寫入的'podlog test'？

iii. 退出 podlog 容器，查看 deyu5 家目錄中的 pjournal 次目錄中是否一樣新增了 podmanfile.log 檔案，內容也是'podlog test'。

Part III

儲存空間管理 KVM10

Chapter 26

開機

26.1 前言

1. Linux 出現開機問題或忘記 root 密碼時，必須進入維護模式。
2. RHEL 6 開關機層級分爲 0,1,2,3,4,5,6 等層級，1 爲單人模式，也就維修模式。
3. RHEL 6 開機時在 grub 的 kernel 參數中指定開機進入單人模式，也就是層級 1，就可免 root 密碼進入系統互動式命令模式。
4. RHEL 7,8,9 開機已不分層級，而是以目標 (targets) 來分。
5. RHEL 7,8,9 無論使用哪種目標開機，都必須輸入帳號及密碼才能進入互動式命令模式。
6. 因此 RHEL 7,8,9 變更 root 密碼，都必須先中斷開機流程，不要讓流程完成 initramfs 載入並進入系統。

26.2 開機 targets

1. 開機分成 4 個目標 targets，其開機結果如下表：

Target	Purpose
graphical.target	多人模式，支援圖形及文字兩種方式登入，等效 v6 的層級 3 及 5。
multi-user.target	多人模式，只支援文字登入方式，等效 v6 的層級 3。
rescue.target	sulogin (Single-user login)，單人模式，等效 v6 的層級 1。
emergency.target	sulogin ，單人模式，完成 initramfs 載入且系統根目錄 / 掛載成唯讀，等效 v6 開機掛載不成功時的維修模式。

2. 指定 target 開機：

   (a) 出現 grub boot loader 開機選單時，立即碰觸鍵盤任意鍵，boot loader 會暫停。

   (b) 按下'e'，編輯選單

   (c) 移動上下鍵至 linux 核心參數行

```
linux ($root)/vmlinuz-4.18.0-80.el8.x86_64 \
root=/dev/mapper/vg_kvm10-root ro crashkernel=auto \
resume=/dev/mapper/vg_kvm10-swap rhgb quiet
linux16 /vmlinuz-3.10.0-123.el7.x86_64 \
```

   (d) 在核心參數行最後加入 systemd.unit=desired.target (desired 爲上表中四種 target 之一)。

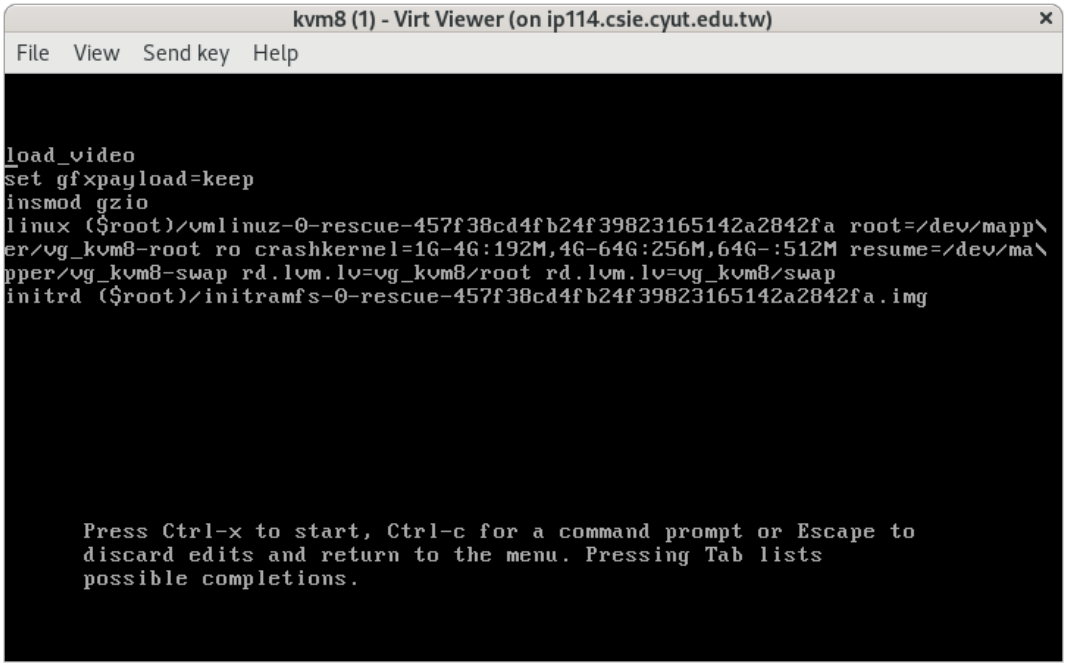
   (e) 再按下 Ctrl+x 重新以這個設定開機。

26.3 變更密碼

1. 出現 grub boot loader 開機選單時，立即碰觸鍵盤任意鍵，boot loader 會暫停。

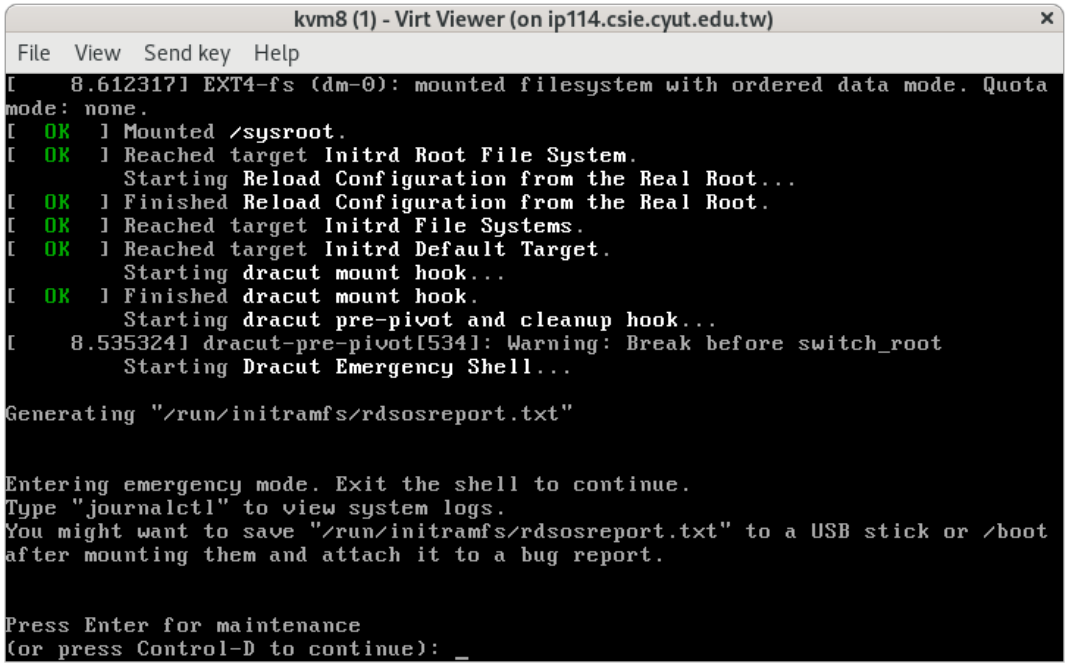
2. 按下'e'，編輯選單

3. 移動上下鍵至 linux 核心參數行




```
linux ($root)/vmlinuz-0-rescue-457f38cd4fb24f39823165142a2842fa \
root=/dev/mapper/vg_kvm8-root ro crashkernel=1G-4G:192M,4G-64G:256M,64G
    -:512M \
resume=/dev/mapper/vg_kvm8-swap rd.lvm.lv=vg_kvm8/root rd.lvm.lv=vg_kvm8
    /swap
```

4. 在核心參數行最後加入 rd.break 。
5. 再按下 Ctrl+x 重新以這個設定開機。
6. 開機後的互動式命令環境，並不是正常開機的系統，正常開機系統掛載在 /sysroot，且掛載成唯讀，必須重新掛載成可寫入，才能變更密碼，步驟如下：
   (a) 要求按 Enter 進入維修，或 Contol-D 繼續，按 Enter 進入維修。



(b) 重新掛載 /sysroot 成可讀可寫。

```
sh-5.1# mount -oremount,rw /sysroot
```

(c) chroot 工作目錄到 /sysroot。

```
sh-5.1# chroot /sysroot
```

(d) 設定新的 root 密碼。

```
sh-5.1# passwd
```

(e) 因爲在此情況下，SELinux 並沒有啓動，對所有檔案的變更，可能會造成檔案的 context 不正確，爲確保開機時重新設定 SELinux context，必須在根目錄下產生隱藏檔.autorelabel。

```
sh-5.1# touch /.autorelabel
```

(f) 退出 chroot

```
sh-5.1# exit
exit
sh-5.1#
```

(g) 退出系統，即重新開機。

```
sh-5.1# exit
```

26.4 實機練習

1. kvm10 root 密碼忘記了，重新設定 root 密碼爲 123qwe。

Chapter 27

YUM 套件管理

27.1 前言

1. 何謂 RPM？
   (a) RPM 全名是『RedHat Package Manager』簡稱爲 RPM。
   (b) 由 Red Hat 公司發展出來，由於 RPM 使用方便，所以成了目前最熱門的套件管理程式。
   (c) RPM 是以資料庫記錄的方式來將所需要的套件安裝到 Linux 主機的一套管理程式。
   (d) RPM 將要安裝的套件先編譯過，並且打包好，安裝時 RPM 會先依照套件裡的紀錄資料查詢相依屬性套件是否滿足，若滿足則予以安裝，若不滿足則不予安裝。
2. RPM 套件的屬性相依
   (a) RPM 打包套件檔案時，會同時加入套件的訊息。例如：
      i. 版本、
      ii. 打包套件者、
      iii. 相依屬性的套件、
      iv. 套件的功能說明、
      v. 套件的所有檔案與目錄紀錄。
   (b) 當要安裝某個以 RPM 型態提供的套件時，如果資料庫顯示其相依套件不存在，則會顯示錯誤訊息。
   (c) 屬性相依的克服方式
      i. 手動下載並安裝好所有相依套件。

ii. 利用 yum 安裝 rpm 套件。

3. YUM

(a) RHEL/AlmaLinux 都提供套件庫 (repository) 供 yum 下載安裝、升級、移除。

(b) 在安裝好 RHEL/AlmaLinux 系統後，都會將 YUM 的套件庫設定到同版本的下載連結點。

(c) 利用 yum 安裝 rpm 套件時，yum 會在設定的套件庫尋找未安裝的相依套件加以安裝。

(d) 也可自行建立套件庫，設定爲 yum 的套件庫，yum 就可以升級，安裝，移除套件庫中的套件。

27.2 yum 套件庫設定

1. 設定 yum 套件庫 (repository) 設定檔，必須在 /etc/yum.repo.d/ 目錄下，建立.repo 檔案，檔案設定方式可查詢 dnf.conf 手冊。

```
[root@kvm8 ~]# man dnf.conf
```

2. 撰寫 /etc/yum.repos.d/dywang.repo，建立 AlmaLinux 9 BaseOS 及 AppStream 兩類套件。

```
[root@kvm8 ~]# vi /etc/yum.repos.d/dywang.repo
[root@kvm8 ~]# cat /etc/yum.repos.d/dywang.repo
[AppStream]
name=App Stream
baseurl=http://dywang.csie.cyut.edu.tw/alma9/AppStream
gpgcheck=0

[BaseOS]
name=Base OS
baseurl=http://dywang.csie.cyut.edu.tw/alma9/BaseOS
gpgcheck=0
```

3. .repo 檔參數說明：

(a) [base]：套件庫的名稱，中刮號一定要存在，名稱可以自訂但不可重複。

(b) name：說明這個套件庫的意義。

(c) mirrorlist=：列出這個套件庫可以使用的映射站台，如果不想使用，可以註解到這行；

(d) baseurl=：指定固定套件庫的實際網址。

(e) enable=1：預設啓動套件庫，可以不設定。如果不想啓動可以使用 enable=0。

(f) gpgcheck=1：指定是否需要查閱 RPM 檔案內的數位簽章。

(g) gpgkey=：數位簽章的公鑰檔所在位置。

4. 列出 yum 使用的所有套件庫

```
[root@kvm8 ~]# dnf repolist
repo id                                     repo name
AppStream                                   App Stream
BaseOs                                      Base OS
```

5. 試安裝 vsftpd？如果出現 Is this ok [y//N]: 表示設定無誤可以取得套件進行安裝。y 安裝、N 不安裝，輸入 n 進行安裝。

```
[root@kvm8 ~]# dnf install vsftpd
Last metadata expiration check: 0:00:41 ago on Tue 04 Oct 2022 11:13:08
    AM CST.
Dependencies resolved.
================================================================================

 Package         Architecture    Version               Repository
             Size
================================================================================

Installing:
 vsftpd          x86_64          3.0.3-49.el9          AppStream
             158 k

Transaction Summary
================================================================================

Install  1 Package

Total download size: 158 k
Installed size: 348 k
Is this ok [y/N]: n
```

6. cache 套件庫資料刪除

```
[root@kvm8 ~]# dnf clean [packages|headers|all]
```

```
選項與參數：
 packages：將已下載的軟體檔案刪除
 headers ：將下載的軟體檔頭刪除
 all     ：將所有套件庫資料都刪除

#刪除已下載過的所有套件庫的相關資料 (含軟體本身與清單)
[root@kvm8 ~]# dnf clean all
```

27.3 *yum 查詢功能

1. 查詢選項：

```
[root@kvm8 ~]# dnf [options] [command] [package ...]
[option]：主要的搜尋選項有：
[查詢工作項目] [相關參數]：這方面的參數有：
  search  ：搜尋某個套件名稱或是描述 (description) 的重要關鍵字；
  list    ：列出目前 yum 所管理的所有的套件名稱與版本，有點類似 rpm -qa
      ;
  info    ：列出套件的相關訊息，類似 rpm -qai 的執行結果；
  provides or whatprovides：搜尋包含某檔案的套件，類似 rpm -qf 的功能。
```

2. 查詢名稱或是描述中包含關鍵字 cjk 的套件

```
[root@kvm8 ~]# dnf search cjk
Last metadata expiration check: 0:16:27 ago on Tue 04 Oct 2022 11:13:08
    AM CST.
========================== Name & Summary Matched: cjk
    ==========================
google-noto-cjk-fonts-common.noarch : Common files for Noto CJK fonts
google-noto-sans-cjk-ttc-fonts.noarch : Sans OTC font files for
                                      : google-noto-cjk-fonts
google-noto-serif-cjk-ttc-fonts.noarch : Serif OTC font files for
                                       : google-noto-cjk-fonts
texlive-cjk.noarch : CJK language support
texlive-xecjk.noarch : Support for CJK documents in XeLaTeX
============================== Summary Matched: cjk
    ==============================
bitmap-fangsongti-fonts.noarch : Selected CJK bitmap fonts for Anaconda
texlive-garuda-c90.noarch : TeX support (from CJK) for the garuda font
texlive-norasi-c90.noarch : TeX support (from CJK) for the norasi font
    in
                          : thailatex
```

3. 查看套件 google-noto-cjk-fonts-common 的相關訊息

```
[root@kvm8 ~]# dnf info google-noto-cjk-fonts-common.noarch
Last metadata expiration check: 0:18:58 ago on Tue 04 Oct 2022 11:13:08
    AM CST.
Installed Packages
Name         : google-noto-cjk-fonts-common
Version      : 20201206
Release      : 4.el9
Architecture : noarch
Size         : 32 k
Source       : google-noto-cjk-fonts-20201206-4.el9.src.rpm
Repository   : @System
From repo    : AppStream
Summary      : Common files for Noto CJK fonts
URL          : https://github.com/googlei18n/noto-cjk
License      : OFL
Description  :
             : Noto CJK fonts, supporting Simplified Chinese,
               Traditional
             : Chinese, Japanese, and Korean. The supported scripts are
               Han,
             : Hiragana, Katakana, Hangul, and Bopomofo. Latin, Greek,
               Cyrllic,
             : and various symbols are also supported for compatibility
               with CJK
             : standards.
             :
             :
             : Common files for Google Noto CJK fonts.
```

4. 列出所有 evince 開頭的套件

```
[root@cs ~]# dnf list evince*
Last metadata expiration check: 21:06:59 ago on Mon 03 Oct 2022 02:25:56
     PM CST.
Installed Packages
evince.x86_64                              40.4-4.el9
    @AppStream
evince-libs.x86_64                         40.4-4.el9
    @AppStream
evince-nautilus.x86_64                     40.4-4.el9
    @AppStream
evince-previewer.x86_64                    40.4-4.el9
    @AppStream
evince-thumbnailer.x86_64                  40.4-4.el9
    @AppStream
Available Packages
evince-libs.i686                           40.4-4.el9
    AppStream
```

5. 列出包含檔案 bin/evince 的所有套件，例如/usr/bin/evince。

```
[root@kvm8 ~]# dnf provides *bin/evince
Last metadata expiration check: 0:21:21 ago on Tue 04 Oct 2022 11:13:08
    AM CST.
evince-40.4-4.el9.x86_64 : Document viewer
Repo        : AppStream
Matched from:
Other       : *bin/evince
```

27.4 *yum 安裝、升級與移除

1. 安裝/升級選項

```
[root@www ~]# yum command [packages]
command:
  install  :安裝
  reinstall:重新安裝
  update   :升級，若要整個系統都升級，後不接套件
  remove   :移除
```

2. 安裝

```
[root@kvm8 ~]# dnf install bzip2
Last metadata expiration check: 0:01:07 ago on Tue 04 Oct 2022 11:53:21
    AM CST.
Dependencies resolved.
================================================================================
 Package         Architecture     Version                 Repository
            Size
================================================================================

Installing:
 bzip2           x86_64           1.0.8-8.el9             BaseOS
                52 k

Transaction Summary
================================================================================

Install  1 Package

Total download size: 52 k
Installed size: 94 k
Is this ok [y/N]: y
Downloading Packages:
```

```
bzip2-1.0.8-8.el9.x86_64.rpm                    721 kB/s |  52 kB
    00:00
--------------------------------------------------------------------------------

Total                                           379 kB/s |  52 kB
    00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                        1/1
  Installing       : bzip2-1.0.8-8.el9.x86_64                               1/1
  Running scriptlet: bzip2-1.0.8-8.el9.x86_64                               1/1
  Verifying        : bzip2-1.0.8-8.el9.x86_64                               1/1

Installed:
  bzip2-1.0.8-8.el9.x86_64

Complete!
```

3. 移除功能：yum [remove] 軟體

```
[root@kvm8 ~]# dnf remove bzip2
Dependencies resolved.
================================================================================
 Package         Architecture     Version                Repository
             Size
================================================================================

Removing:
 bzip2           x86_64           1.0.8-8.el9            @BaseOS
                94 k

Transaction Summary
================================================================================

Remove  1 Package

Freed space: 94 k
Is this ok [y/N]: y
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
```

```
  Preparing        :                                                1/1
  Erasing          : bzip2-1.0.8-8.el9.x86_64                       1/1
  Running scriptlet: bzip2-1.0.8-8.el9.x86_64                       1/1
  Verifying        : bzip2-1.0.8-8.el9.x86_64                       1/1

Removed:
  bzip2-1.0.8-8.el9.x86_64

Complete!
```

27.5 實機練習

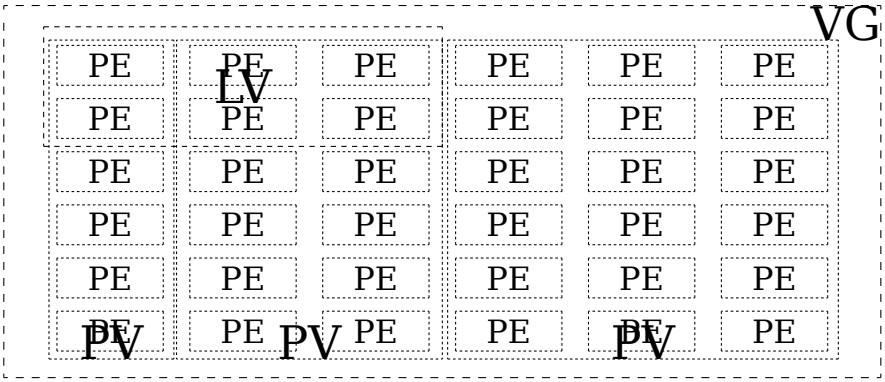
1. 建立 yum 安裝套件倉庫來源
   (a) http://dywang.csie.cyut.edu.tw/alma9/BaseOS
   (b) http://dywang.csie.cyut.edu.tw/alma9/AppStream
2. 取消 RPM 檔案數位簽章檢查
3. 試著安裝套件 vsftpd

Chapter 28

邏輯卷冊放大縮小

28.1 簡介

1. 何謂 LVM？
    - LVM (Logical Volume Manager)：邏輯卷冊管理。
    - LVM 目的爲彈性進行 filesystem 的擴充或縮小。
2. PV, PE, VG, LV 的意義
    - Physical Volume, PV, 實體捲軸：partition 時需要調整系統識別碼 (system ID) 成爲 8e (LVM 的識別碼)。
    - Volume Group, VG, 捲軸群組：將許多 PV 整合而成。
    - Physical Extend, PE, 實體延伸區塊：LVM 最小的儲存區塊。
    - Logical Volume, LV, 邏輯卷冊：從 VG 切割而成，爲最後可被格式化使用之格式，LV 的裝置檔名通常指定爲『/dev/vgname/lvname 』的樣式。



28.2 縮小 LV 標準作法

1. 查看要縮小 LV vg_kvm7home-vo 掛載在 /home，大小爲 74M (80M)。

```
[root@kvm7 ~]# df -h
Filesystem                      Size  Used Avail Use% Mounted on
/dev/mapper/vg_kvm7usb-root     3.1G  1.1G  1.8G  37% /
devtmpfs                        487M     0  487M   0% /dev
tmpfs                           497M     0  497M   0% /dev/shm
tmpfs                           497M  6.6M  491M   2% /run
tmpfs                           497M     0  497M   0% /sys/fs/cgroup
/dev/vda1                       197M  107M   90M  55% /boot
/dev/mapper/vg_kvm7home-vo       74M  1.6M   68M   3% /home
/dev/mapper/vg_mnt-pub           24M     0   24M   0% /mnt/pub
```

2. 卸載 LV vg_kvm7home-vo。

```
[root@kvm7 ~]# umount /dev/vg_kvm7home/vo
```

3. 先 resize LV vg_kvm7home-vo 檔案系統爲 30M。

```
[root@kvm7 ~]# resize2fs -f /dev/vg_kvm7home/vo 30M
resize2fs 1.42.9 (28-Dec-2013)
Resizing the filesystem on /dev/vg_kvm7home/vo to 30720 (1k) blocks.
The filesystem on /dev/vg_kvm7home/vo is now 30720 blocks long.
```

4. 再 resize LV vg_kvm7home-vo 爲 30M，縮小時此命令一定要在 resize2fs 命令之後執行，放大時則順序相反。

```
[root@kvm7 ~]# lvresize -L 30M /dev/vg_kvm7home/vo
  WARNING: Reducing active logical volume to 30.00 MiB
  THIS MAY DESTROY YOUR DATA (filesystem etc.)
Do you really want to reduce vo? [y/n]: y
  Size of logical volume vg_kvm7home/vo changed from 80.00 MiB (40
      extents) to 30.00 MiB (15 extents).
  Logical volume vo successfully resized
```

5. 再檢查 LV vg_kvm7home-vo 大小爲 24M (30M)。

```
[root@kvm7 ~]# df -h
Filesystem                      Size  Used Avail Use% Mounted on
/dev/mapper/vg_kvm7usb-root     3.1G  1.1G  1.8G  37% /
```

```
devtmpfs                    487M     0  487M   0% /dev
tmpfs                       497M     0  497M   0% /dev/shm
tmpfs                       497M  6.6M  491M   2% /run
tmpfs                       497M     0  497M   0% /sys/fs/cgroup
/dev/vda1                   197M  107M   90M  55% /boot
/dev/mapper/vg_mnt-pub       24M     0   24M   0% /mnt/pub
/dev/mapper/vg_kvm7home-vo   25M  784K   23M   4% /home
```

28.3 放大 LV 標準作法

1. 查看要放大 LV vg_kvm7home-vo 掛載在 /home，大小爲 24M (30M)。

```
[root@kvm7 ~]# df -h
Filesystem                  Size  Used Avail Use% Mounted on
/dev/mapper/vg_kvm7usb-root 3.1G  1.1G  1.8G  37% /
devtmpfs                    487M     0  487M   0% /dev
tmpfs                       497M     0  497M   0% /dev/shm
tmpfs                       497M  6.6M  491M   2% /run
tmpfs                       497M     0  497M   0% /sys/fs/cgroup
/dev/vda1                   197M  107M   90M  55% /boot
/dev/mapper/vg_mnt-pub       24M     0   24M   0% /mnt/pub
/dev/mapper/vg_kvm7home-vo   25M  784K   23M   4% /home
```

2. 先 resize LV vg_kvm7home-vo 爲 100M，放大時此命令一定要在 resize2fs 命令之前執行，縮小時則順序相反。

```
[root@kvm7 ~]# lvextend -L 100M /dev/vg_kvm7home/vo
  Size of logical volume vg_kvm7home/vo changed from 30.00 MiB (15
      extents) to 100.00 MiB (50 extents).
  Logical volume vo successfully resized
```

3. 再 resize LV vg_kvm7home-vo 檔案系統，不必輸入大小，檔案系統會依已 extend 的 LV resize。

```
[root@kvm7 ~]# resize2fs -f /dev/vg_kvm7home/vo
resize2fs 1.42.9 (28-Dec-2013)
Filesystem at /dev/vg_kvm7home/vo is mounted on /home; on-line resizing
    required
old_desc_blocks = 1, new_desc_blocks = 1
The filesystem on /dev/vg_kvm7home/vo is now 102400 blocks long.
```

4. 再檢查 LV vg_kvm7home-vo 大小爲 93M (100M)。

```
[root@kvm7 ~]# df -h
Filesystem                      Size  Used Avail Use% Mounted on
/dev/mapper/vg_kvm7usb-root     3.1G  1.1G  1.8G  37% /
devtmpfs                        487M     0  487M   0% /dev
tmpfs                           497M     0  497M   0% /dev/shm
tmpfs                           497M  6.6M  491M   2% /run
tmpfs                           497M     0  497M   0% /sys/fs/cgroup
/dev/vda1                       197M  107M   90M  55% /boot
/dev/mapper/vg_mnt-pub           24M     0   24M   0% /mnt/pub
/dev/mapper/vg_kvm7home-vo       93M  1.6M   87M   2% /home
```

28.4 單一指令放大縮小 LV

1. 以 lvresize 指令一次完成放大或縮小。特別注意！lv 縮小應先卸載，無法卸載的分割區，盡量不要使用此方法做。例如根目錄無法卸載，且 lvm 相關命令在 /sbin 目錄下，直接將掛載根目錄 / 的分割區，做放大縮小，風險非常高，可能造成資料的損壞。

2. 先檢查 LV vo 在 VG vg_kvm7home 內，大小爲 93M (100M)。

```
[root@kvm7 ~]# df -h
Filesystem                      Size  Used Avail Use% Mounted on
/dev/mapper/vg_kvm7usb-root     3.1G  1.1G  1.8G  37% /
devtmpfs                        487M     0  487M   0% /dev
tmpfs                           497M     0  497M   0% /dev/shm
tmpfs                           497M  6.6M  491M   2% /run
tmpfs                           497M     0  497M   0% /sys/fs/cgroup
/dev/vda1                       197M  107M   90M  55% /boot
/dev/mapper/vg_mnt-pub           24M     0   24M   0% /mnt/pub
/dev/mapper/vg_kvm7home-vo       93M  1.6M   87M   2% /home
```

3. lv 檔案系統的檔案有兩種。

```
[root@kvm5 ~]# ll /dev/vg_kvm5home/vo
lrwxrwxrwx. 1 root root 7 Oct 21 10:13 /dev/vg_kvm5home/vo -> ../dm-2
[root@kvm5 ~]# ll /dev/mapper/vg_kvm5home-vo
lrwxrwxrwx. 1 root root 7 Oct 21 10:13 /dev/mapper/vg_kvm5home-vo -> ../
    dm-2
```

4. 以單一指令 lvresize 將 LV vo 大小改爲 40M，不用理會是放大還是縮小，其中 -r 選項是一併執行 resize2fs 的命令，不知道 vo 所屬的群組，可以使用 df -h 查看，或下命令時使用 TAB 參數提示。

```
[root@kvm7 ~]# lvresize -L 40M -r /dev/vg_kvm7home/vo
Do you want to unmount "/home"? [Y|n] y
fsck from util-linux 2.23.2
/dev/mapper/vg_kvm7home-vo: 11/26624 files (9.1% non-contiguous),
    9015/102400 blocks
resize2fs 1.42.9 (28-Dec-2013)
Resizing the filesystem on /dev/mapper/vg_kvm7home-vo to 40960 (1k)
    blocks.
The filesystem on /dev/mapper/vg_kvm7home-vo is now 40960 blocks long.

  Size of logical volume vg_kvm7home/vo changed from 100.00 MiB (50
      extents) to 40.00 MiB (20 extents).
  Logical volume vo successfully resized
```

5. 再檢查 LV vg_kvm7home-vo 大小爲 35M (40M)。

```
[root@kvm7 ~]# df -h
Filesystem                  Size  Used Avail Use% Mounted on
/dev/mapper/vg_kvm7usb-root  3.1G  1.1G  1.8G  37% /
devtmpfs                    487M     0  487M   0% /dev
tmpfs                       497M     0  497M   0% /dev/shm
tmpfs                       497M  6.6M  491M   2% /run
tmpfs                       497M     0  497M   0% /sys/fs/cgroup
/dev/vda1                   197M  107M   90M  55% /boot
/dev/mapper/vg_mnt-pub       24M     0   24M   0% /mnt/pub
/dev/mapper/vg_kvm7home-vo   35M  784K   32M   3% /home
```

6. 以單一指令 lvresize 將 LV vg_kvm7home-vo 大小改爲 80M，不用理會是放大還是縮小，其中 -r 選項是一併執行 resize2fs 的命令。

```
[root@kvm7 ~]# lvresize -L 80M -r /dev/vg_kvm7home/vo
  Size of logical volume vg_kvm7home/vo changed from 40.00 MiB (20
      extents) to 80.00 MiB (40 extents).
  Logical volume vo successfully resized
resize2fs 1.42.9 (28-Dec-2013)
Filesystem at /dev/mapper/vg_kvm7home-vo is mounted on /home; on-line
    resizing required
old_desc_blocks = 1, new_desc_blocks = 1
The filesystem on /dev/mapper/vg_kvm7home-vo is now 81920 blocks long.
```

7. 再檢查 LV vg_kvm7home-vo 大小爲 74M (80M)。

```
[root@kvm7 ~]# df -h
Filesystem                      Size  Used Avail Use% Mounted on
/dev/mapper/vg_kvm7usb-root     3.1G  1.1G  1.8G  37% /
devtmpfs                        487M     0  487M   0% /dev
tmpfs                           497M     0  497M   0% /dev/shm
tmpfs                           497M  6.6M  491M   2% /run
tmpfs                           497M     0  497M   0% /sys/fs/cgroup
/dev/vda1                       197M  107M   90M  55% /boot
/dev/mapper/vg_mnt-pub           24M     0   24M   0% /mnt/pub
/dev/mapper/vg_kvm7home-vo       74M  1.6M   68M   3% /home
```

28.5 實機練習

1. df -h 查詢目前掛載的檔案系統，其中名爲 vo 的 LV 大小，將其寫到 /root/lvsize0，單位 M 不用寫。

2. 比較 vosize 與查詢到的 vo 大小，是放大還是縮小？將結果寫到 /root/lvmod，放大寫入"extend"，縮小寫入"reduce"。

3. 決定使用標準的放大縮小作法，或使用單一指令，完成 lv 的大小調整。將 vo 的大小調整爲 vosize。

4. df -h 再查詢掛載的檔案系統，觀察名爲 vo 的 LV 大小是否已經調整爲 vosize？將其大小寫到 /root/lvsize1，單位 M 不用寫，查到的大小與 vosize 差十幾 M 都是正確。

Chapter 29

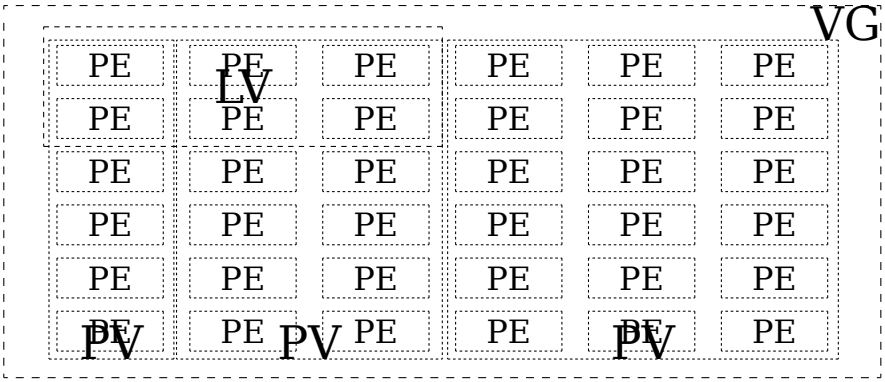
邏輯卷冊管理

29.1 前言

1. LVM (Logical Volume Manager)：邏輯卷冊管理，目的爲彈性進行 filesystem 的擴充或縮小。

2. PV, PE, VG, LV 的意義

    - Physical Volume, PV, 實體捲軸：partition 時需要調整系統識別碼 (system ID) 成爲 8e (LVM 的識別碼)。
    - Volume Group, VG, 捲軸群組：將許多 PV 整合而成。
    - Physical Extend, PE, 實體延伸區塊：LVM 最小的儲存區塊。
    - Logical Volume, LV, 邏輯卷冊：從 VG 切割而成，爲最後可被格式化使用之格式，LV 的裝置檔名通常指定爲『/dev/vgname/lvname 』的樣式。



29.2 新增分割區

1. 使用 gdisk 分割硬碟 /dev/vda。

```
[root@kvm10 ~]# gdisk /dev/vda
GPT fdisk (gdisk) version 1.0.7

Partition table scan:
  MBR: protective
  BSD: not present
  APM: not present
  GPT: present

Found valid GPT with protective MBR; using GPT.

Command (? for help):
```

2. 命令 p：列出分割表，目前有三個分割區。

```
Command (? for help): p
Disk /dev/vda: 12582912 sectors, 6.0 GiB
Sector size (logical/physical): 512/512 bytes
Disk identifier (GUID): 85954A41-1113-4533-A9A4-FFEB73552195
Partition table holds up to 128 entries
Main partition table begins at sector 2 and ends at sector 33
First usable sector is 34, last usable sector is 12582878
Partitions will be aligned on 2048-sector boundaries
Total free space is 2340797 sectors (1.1 GiB)

Number  Start (sector)    End (sector)  Size       Code  Name
   1            2048            4095   1024.0 KiB  EF02
   2            4096         1028095   500.0 MiB   8300
   3         1028096        10244095   4.4 GiB     8E00
```

3. 命令 n：新增分割區，如果沒有要再新增其他分割區，可以直接將硬碟剩餘空間「全部」分割爲 LVM 的分割區，分割區的 type 輸入 8e00 (Linux LVM) ，不知道 type 可輸入 L 查詢，其餘分割編號，分割起迄位置都按「Enter」接受系統建議值。

```
Command (? for help): n
Partition number (4-128, default 4):
First sector (34-12582878, default = 10244096) or {+-}size{KMGTP}:
Last sector (10244096-12582878, default = 12582878) or {+-}size{KMGTP}:
Current type is 8300 (Linux filesystem)
Hex code or GUID (L to show codes, Enter = 8300): 8e00
Changed type of partition to 'Linux LVM'
```

4. 命令 p：再次列出分割表，新增的第 4 個分割區 1.1 GiB，type 是 Linux LVM。

```
Command (? for help): p
Disk /dev/vda: 12582912 sectors, 6.0 GiB
Sector size (logical/physical): 512/512 bytes
Disk identifier (GUID): 85954A41-1113-4533-A9A4-FFEB73552195
Partition table holds up to 128 entries
Main partition table begins at sector 2 and ends at sector 33
First usable sector is 34, last usable sector is 12582878
Partitions will be aligned on 2048-sector boundaries
Total free space is 2014 sectors (1007.0 KiB)

Number  Start (sector)    End (sector)  Size       Code  Name
   1            2048            4095   1024.0 KiB  EF02
   2            4096         1028095   500.0 MiB   8300
   3         1028096        10244095   4.4 GiB     8E00
   4        10244096        12582878   1.1 GiB     8E00  Linux LVM
```

5. 命令 w：寫入 GTP 分割表，回答 y 寫入並退出。

```
Command (? for help): w

Final checks complete. About to write GPT data. THIS WILL OVERWRITE
    EXISTING
PARTITIONS!!

Do you want to proceed? (Y/N): y
OK; writing new GUID partition table (GPT) to /dev/vda.
Warning: The kernel is still using the old partition table.
The new table will be used at the next reboot or after you
run partprobe(8) or kpartx(8)
The operation has completed successfully.
```

6. partprobe 更新 /dev/vda 的分割表訊息。

```
[root@kvm10 ~]# partprobe /dev/vda
```

7. lsblk 查詢 /dev/vda，已出現 vda4。

```
[root@kvm10 ~]# lsblk /dev/vda
NAME              MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
vda               252:0    0    6G  0 disk
  vda1            252:1    0    1M  0 part
  vda2            252:2    0  500M  0 part /boot
  vda3            252:3    0  4.4G  0 part
    vg_kvm10-root 253:0    0  4.3G  0 lvm  /
    vg_kvm10-swap 253:1    0  124M  0 lvm  [SWAP]
  vda4            252:4    0  1.1G  0 part
```

29.3 產生 PV

1. 查看 pv 可用指令。

```
[root@kvm10 ~]# pv [TAB}{TAB]
pvchange   pvcreate   pvmove     pvresize   pvscan
pvck       pvdisplay  pvremove   pvs
```

2. 查看目前只有兩個 PV。

```
[root@kvm10 ~]# pvs
  PV         VG       Fmt  Attr PSize   PFree
  /dev/vda3  vg_kvm10 lvm2 a--    4.39g      0
  /dev/vdb1  myvdb    lvm2 a--  396.00m 296.00m
```

3. 將 /dev/vda4 建立爲 PV。

```
[root@kvm10 ~]# pvcreate /dev/vda4
  Physical volume "/dev/vda4" successfully created.
```

4. 再查看 /dev/vda4 是新建立的 PV，但還沒有 VG。

```
[root@kvm10 ~]# pvs
  PV         VG       Fmt  Attr PSize   PFree
  /dev/vda3  vg_kvm10 lvm2 a--    4.39g      0
  /dev/vda4           lvm2 ---   <1.12g  <1.12g
  /dev/vdb1  myvdb    lvm2 a--  396.00m 296.00m
```

29.4 產生 VG

1. 查看 vg 可用指令。

```
[root@kvm10 ~]# vg
vgcfgbackup    vgconvert      vgextend       vgmknodes      vgs
vgcfgrestore   vgcreate       vgimport       vgreduce       vgscan
```

```
vgchange        vgdisplay       vgimportclone  vgremove        vgsplit
vgck            vgexport        vgmerge        vgrename
```

2. vgs 查詢 VG，目前有兩個 VG。

```
[root@kvm10 ~]# vgs
  VG        #PV #LV #SN Attr   VSize   VFree
  myvdb       1   1   0 wz--n- 396.00m 296.00m
  vg_kvm10    1   2   0 wz--n-   4.39g      0
```

3. 查看 vgcreate 輔助說明。

```
[root@kvm10 ~]# vgcreate --help
  vgcreate - Create a volume group

  vgcreate VG_new PV ...
    [ -s|--physicalextentsize Size[m|UNIT] ]

  Use --longhelp to show all options and advanced commands.
```

4. 在 PV /dev/vda4 上建立名爲 vg_mnt PE size 2M 的 VG。

```
[root@kvm10 ~]# vgcreate -s 2M vg_mnt /dev/vda4
  Volume group "vg_mnt" successfully created
```

5. 再查看 VG，出現 vg_mnt。

```
[root@kvm10 ~]# vgs
  VG        #PV #LV #SN Attr   VSize   VFree
  myvdb       1   1   0 wz--n- 396.00m 296.00m
  vg_kvm10    1   2   0 wz--n-   4.39g      0
  vg_mnt      1   0   0 wz--n-   1.11g   1.11g
```

29.5 產生 LV

1. 查看 lv 可用指令。

```
[root@kvm10 ~]# lv [TAB][TAB]
lvchange        lvextend        lvmdiskscan     lvmsadc         lvrename
lvconvert       lvm             lvmdump         lvmsar          lvresize
lvcreate        lvmconfig       lvm_import_vdo  lvreduce        lvs
lvdisplay       lvmdevices      lvmpolld        lvremove        lvscan
```

2. lvs 查詢 LV，目前有三個 LV。

```
[root@kvm10 ~]# lvs
 LV   VG        Attr       LSize   Pool Origin Data% Meta% Move Log Cpy%
    Sync Convert
 vo   myvdb     -wi-ao---- 100.00m
 root vg_kvm10 -wi-ao----  <4.27g
 swap vg_kvm10 -wi-ao---- 124.00m
```

3. 查看 lvcreate 輔助說明。

```
[root@kvm10 ~]# lvcreate --help
  lvcreate - Create a logical volume

  Create a linear LV.
  lvcreate -L|--size Size[m|UNIT] VG
    [ -l|--extents Number[PERCENT] ]
    [ -n|--name String ]
```

4. 在 vg_mnt 上建立名爲 pub，大小爲 12 個 PE 的 LV。

```
[root@kvm10 ~]# lvcreate -l 12 -n pub vg_mnt
  Logical volume "pub" created.
```

5. 查看 LV 出現 pub 在 vg_mnt VG 內。

```
[root@kvm10 ~]# lvs
 LV   VG        Attr       LSize   Pool Origin Data% Meta% Move Log Cpy%
    Sync Convert
 vo   myvdb     -wi-ao---- 100.00m
 root vg_kvm10 -wi-ao----  <4.27g
 swap vg_kvm10 -wi-ao---- 124.00m
 pub  vg_mnt   -wi-a-----  24.00m
```

6. 在 vg_mnt 再新建立名爲 swap，大小爲 48M 的 LV，做爲 swap。

```
[root@kvm10 ~]# lvcreate -L 48M -n swap vg_mnt
  Logical volume "swap" created.
```

7. 再查看 LV 出現 swap 在 vg_mnt VG 內。

```
[root@kvm10 ~]# lvs
 LV   VG       Attr       LSize   Pool Origin Data% Meta% Move Log Cpy%
    Sync Convert
 vo   myvdb    -wi-ao---- 100.00m
 root vg_kvm10 -wi-ao----  <4.27g
 swap vg_kvm10 -wi-ao---- 124.00m
 pub  vg_mnt   -wi-a-----  24.00m
 swap vg_mnt   -wi-a-----  48.00m
```

29.6 格式化

1. 依照要求 (此例爲 vfat) 格式化 vg_mnt-pub。

```
[root@kvm10 ~]# mkfs.vfat /dev/vg_mnt/pub
mkfs.fat 4.2 (2021-01-31)
```

2. 格式化 vg_mnt-swap爲 swap 。

```
[root@kvm10 ~]# mkswap /dev/vg_mnt/swap
Setting up swapspace version 1, size = 48 MiB (50327552 bytes)
no label, UUID=0fe6d256-dcab-4382-9b1b-9502f76d5c2d
```

29.7 開機掛載 FS

1. 系統掛載限制：

   (a) 根目錄 / 一定要掛載且最先掛載。

   (b) 其它掛載點 (mount points) 必須爲自已建立的目錄。

   (c) 所有 mount points 在同一時間之內，只能掛載一次。

   (d) 所有 partitions 在同一時間之內，只能掛載一次。

(e) 若要卸載，必須先將工作目錄移到 mount point (及其子目錄) 之外。

2. 建立 vg_mnt-pub 掛載目錄 (此例爲 /mnt/pub)。

```
[root@kvm10 ~]# mkdir /mnt/pub
```

3. /etc/fstab 共有六個欄位，細節請參考手冊。

```
[root@kvm10 ~]# man 5 /etc/fstab
```

(a) 磁碟裝置代號或該裝置的 Label

(b) 掛載點 (mount point)

(c) 檔案系統格式

(d) 檔案系統掛載參數

(e) 能否被 dump 備份指令作用：0 表示不做 dump 備份，1, 2 表示要進行 dump 的動作，1 優先於 2。

(f) 是否以 fsck 檢驗磁區：0 是不要檢驗，1, 2 是要檢驗，1 優先於 2。

4. 編輯 /etc/fstab 設定開機自動掛載 vg_mnt-pub 於 /mnt/pub，vg_mnt-swap 爲 swap，建議選擇 /etc/fstab 類似的行，yy 複製，p 貼上，再來修改。

```
[root@kvm10 ~]# vim /etc/fstab
[root@kvm10 ~]# tail -2 /etc/fstab
/dev/vg_mnt/pub    /mnt/pub     vfat     defaults    1 2
/dev/vg_mnt/swap    swap        swap     defaults    0 0
```

5. 重新依 /etc/fstab 掛載表掛載檔案系統，如果沒有錯誤訊息表示正常。

```
[root@kvm10 ~]# mount -a
```

6. 查看 vg_mnt-pub 已成功掛載於 /mnt/pub，且 type 爲 vfat。

```
[root@kvm10 ~]# df -Th /mnt/pub
Filesystem             Type  Size  Used Avail Use% Mounted on
/dev/mapper/vg_mnt-pub vfat   24M     0   24M   0% /mnt/pub
```

7. 檢查 VG vg_mnt 的 PE size 是否如要求 (此例爲 2M)。

```
[root@kvm10 ~]# vgdisplay vg_mnt | grep 'PE Size'
  PE Size               2.00 MiB
```

8. 檢查 LV vg_mnt-pub 的 LE 是否如要求 (此例爲 12 個 PE)。

```
[root@kvm10 ~]# lvdisplay /dev/vg_mnt/pub| grep LE
  Current LE             12
```

29.8 開機啓動 Swap

1. 查詢啓動的 swap，只有 /dev/dm-1。

```
[root@kvm10 ~]# swapon -s
Filename        Type        Size        Used    Priority
/dev/dm-1       partition   126972      0       -2
```

2. 啓動掛載表 /etc/fstab 中設定的所有 swap。

```
[root@kvm10 ~]# swapon -a
```

3. 再查詢啓動的 swap，其中 /dev/dm-4 就是 LV vg_mnt-swap，大小是 48M。

```
[root@kvm10 ~]# swapon -s
Filename        Type        Size        Used    Priority
/dev/dm-1       partition   126972      0       -2
/dev/dm-4       partition   49148       0       -3
```

29.9 *SWAP LV 放大縮小

1. 要放大或縮小 swap，必須先關閉 swap。

```
[root@kvm10 ~]# swapoff /dev/vg_mntx/swap
```

2. 使用 lvresize 放大或縮小 swap，加 -r|--resizefs 選項會出現不支援的訊息，因爲 -r 選項是做 linux 檔案系統的 resize。

```
[root@kvm10 ~]# lvresize --size 20M -r /dev/vg_mntx/swap
fsck from util-linux 2.23.2
fsadm: Filesystem "swap" on device "/dev/mapper/vg_mntx-swap" is not
    supported by this tool.
  /usr/sbin/fsadm failed: 1
  Filesystem resize failed.
```

3. 拿掉 -r|--resizefs 選項後，lvresize 將 LV vg_mntx-pub 放大或縮小。

```
[root@kvm10 ~]# lvresize --size 20M /dev/vg_mntx/swap
  WARNING: Reducing active logical volume to 20.00 MiB.
  THIS MAY DESTROY YOUR DATA (filesystem etc.)
Do you really want to reduce vg_mntx/swap? [y/n]: y
  Size of logical volume vg_mntx/swap changed from 32.00 MiB (8 extents)
      to 20.00 MiB (5 extents).
  Logical volume vg_mntx/swap successfully resized.
```

4. 重新格式化 LV vg_mntx-swap。

```
[root@kvm10 ~]# mkswap /dev/vg_mntx/swap
mkswap: /dev/vg_mntx/swap: warning: wiping old swap signature.
Setting up swapspace version 1, size = 20476 KiB
no label, UUID=ad32041f-6822-464a-b58c-836f41f4018f
```

5. 將 /etc/fstab 中設定的 swap 啓動。

```
[root@kvm10 ~]# swapon -a
```

6. 再查詢 swap，/dev/dm-4 (/dev/cg_mntx-swap) 變更爲 20476 bytes (20M)。

```
[root@kvm10 ~]# swapon -s
Filename        Type        Size    Used    Priority
/dev/dm-0       partition   126972  0       -1
/dev/dm-4       partition   20476   0       -2
```

29.10 實機練習

1. 找到 kvm10 開機根目錄掛載的硬碟，將硬碟代號，例如 /dev/sda，寫到 /root/kvm10d。
2. gdisk 在 kvm10 開機硬碟新增一分割區，大小包含硬碟未使用的全部空間，type 是 Linux LVM。
3. pvcreate 初始化新分割的 LVM 空間。
4. 在新初始化的 LVM 實體空間，建立名爲vg_mnt，PE 大小爲 2M 的 VG。
5. 在新建立的vg_mnt，新增名爲 pub，大小爲 12 個 PE 的 LV。
6. 在新建立的vg_mnt，新增名爲 swap，大小爲 48M 的 LV。
7. 將名爲 pub LV 格式化爲 vfat。
8. 將名爲 swap LV 格式化爲 swap。
9. 建立目錄 /mnt/pub。
10. 設定開機自動掛載名爲 pub 的 LV 於目錄 /mnt/pub。
11. 設定開機自動啓動名爲 swap 的 LV 爲 swap。

Chapter 30

邏輯卷冊 - 可獨立修改項目

30.1 修改 LV 名稱

1. 查看目前掛載在 /mnt/pub 的 LV 爲 VG vg_mnt 下的 pub。

```
[root@kvm10 ~]# df -Th /mnt/pub
Filesystem             Type  Size  Used Avail Use% Mounted on
/dev/mapper/vg_mnt-pub vfat   24M     0   24M   0% /mnt/pub
```

2. 查看修改 LV 名稱指令輔助說明。

```
[root@kvm10 ~]# lvrename --help
  lvrename - Rename a logical volume

  lvrename VG LV LV_new
    [ COMMON_OPTIONS ]

  lvrename LV LV_new
    [ COMMON_OPTIONS ]
```

3. 將 VG vg_mnt 下名爲 pub 的 LV 改名爲 pubx。

```
[root@kvm10 ~]# lvrename vg_mnt pub pubx
  Renamed "pub" to "pubx" in volume group "vg_mnt"
```

4. 將 /mnt/pub 卸解再重新掛載會出現裝置不存在的訊息。

```
[root@kvm10 ~]# umount /mnt/pub
[root@kvm10 ~]# mount -a
mount: /mnt/pub: special device /dev/mapper/vg_mnt-pub does not exist.
```

```
[root@kvm10 ~]# df -Th /mnt/pub
Filesystem                Type  Size  Used Avail Use% Mounted on
```

5. 編輯 /etc/fstab 將裝置名稱由 vg_mnt-pub vg_mnt-pubx。

```
[root@kvm10 ~]# vim /etc/fstab
[root@kvm10 ~]# grep /mnt/pub /etc/fstab
/dev/mapper/vg_mnt-pubx    /mnt/pub              vfat    defaults
          1 2
```

6. 重新依 /etc/fstab 掛載。

```
[root@kvm10 ~]# mount -a
```

7. 成功掛載 vg_mnt-pubx。

```
[root@kvm10 ~]# df -Th /mnt/pub
Filesystem              Type  Size  Used Avail Use% Mounted on
/dev/mapper/vg_mnt-pubx vfat   24M     0   24M   0% /mnt/pub
```

30.2 修改 VG 名稱

1. 查看目前掛載在 /mnt/pub 的 LV 爲 VG vg_mnt 下的 pubx。

```
[root@kvm10 ~]# df -Th /mnt/pub
Filesystem              Type  Size  Used Avail Use% Mounted on
/dev/mapper/vg_mnt-pubx vfat   24M     0   24M   0% /mnt/pub
```

2. 查看修改 VG 名稱指令輔助說明。

```
[root@kvm10 ~]# vgrename --help
  vgrename - Rename a volume group

  Rename a VG.
  vgrename VG VG_new
    [ COMMON_OPTIONS ]
```

3. 將 VG vg_mnt 改名爲 vg_mntx。

```
[root@kvm10 ~]# vgrename vg_mnt vg_mntx
  Volume group "vg_mnt" successfully renamed to "vg_mntx"
```

4. 編輯 /etc/fstab 將裝置名稱由 vg_mnt-pubx vg_mntx-pubx。

```
[root@kvm10 ~]# vim /etc/fstab
[root@kvm10 ~]# grep /mnt/pub /etc/fstab
/dev/mapper/vg_mntx-pubx    /mnt/pub              vfat    defaults
          1 2
```

5. 重新依 /etc/fstab 掛載。

```
[root@kvm10 ~]# mount -a
```

6. 成功掛載 vg_mntx-pubx。

```
[root@kvm10 ~]# df -Th /mnt/pub
Filesystem              Type  Size  Used Avail Use% Mounted on
/dev/mapper/vg_mnt-pubx vfat   24M     0   24M   0% /mnt/pub
```

30.3 修改檔案系統格式

1. 查看目前掛載在 /mnt/pub 的 LV 格式爲 vfat。

```
[root@kvm10 ~]# df -Th /mnt/pub
Filesystem              Type  Size  Used Avail Use% Mounted on
/dev/mapper/vg_mnt-pubx vfat   24M     0   24M   0% /mnt/pub
```

2. 先卸載 /mnt/pub。

```
[root@kvm10 ~]# umount /mnt/pub
```

3. 重新格式化爲 ext3，格式化後原來存的資料當然也被清除。

```
[root@kvm10 ~]# mkfs.ext3 /dev/vg_mntx/pubx
mke2fs 1.44.3 (10-July-2018)
/dev/vg_mntx/pubx contains a vfat file system
Proceed anyway? (y,N) y
Creating filesystem with 24576 1k blocks and 6144 inodes
Filesystem UUID: 3ae13f81-2156-4680-9848-af37f61aa666
Superblock backups stored on blocks:
    8193

Allocating group tables: done
Writing inode tables: done
Creating journal (1024 blocks): done
Writing superblocks and filesystem accounting information: done
```

4. 編輯 /etc/fstab 將要掛載的裝置 vg_mntx-pubx，type 改成 ext3。

```
[root@kvm10 ~]# vim /etc/fstab
[root@kvm10 ~]# grep /mnt/pub /etc/fstab
/dev/mapper/vg_mntx-pubx    /mnt/pub         ext3    defaults        1 2
```

5. 重新依 /etc/fstab 掛載。

```
[root@kvm10 ~]# mount -a
```

6. 成功掛載 vg_mntx-pubx。

```
[root@kvm10 ~]# df -Th /mnt/pub
Filesystem               Type  Size  Used Avail Use% Mounted on
/dev/mapper/vg_mntx-pubx ext3   23M  209K   21M   1% /mnt/pub
```

30.4 修改掛載目錄

1. 先查詢要掛載狀況，假設要修改掛載目錄由 /mnt/pub 改成 /mnt/pubx。

```
[root@kvm7 ~]# df -h
Filesystem                  Size  Used Avail Use% Mounted on
devtmpfs                    482M     0  482M   0% /dev
tmpfs                       496M     0  496M   0% /dev/shm
tmpfs                       496M   20M  476M   4% /run
tmpfs                       496M     0  496M   0% /sys/fs/cgroup
/dev/mapper/vg_kvm7usb-root  3.1G  2.7G  188M  94% /
```

```
/dev/vda1                197M  179M   19M  91% /boot
/dev/mapper/vg_kvm7home-vo  74M  1.6M   67M   3% /home
tmpfs                    100M     0  100M   0% /run/user/0
/dev/mapper/vg_mntx-pub   23M  209K   21M   1% /mnt/pub
```

2. 卸載 /mnt/pub。

```
[root@kvm7 ~]# umount /mnt/pub/
```

3. 修改目錄名稱由 /mnt/pub 改成 /mnt/pubx。

```
[root@kvm7 ~]# mv /mnt/pub/ /mnt/pubx
```

4. 編輯開機掛載表 /etc/fstab，將掛載目錄由 /mnt/pub 改成 /mnt/pubx。

```
[root@kvm7 ~]# vim /etc/fstab
[root@kvm7 ~]# grep pubx /etc/fstab
/dev/mapper/vg_mntx-pub    /mnt/pubx                ext3    defaults
          1 2
```

5. 重新掛載。

```
[root@kvm7 ~]# mount -a
```

6. 查看掛載狀況，LV vg_mntx-pub 掛載在 /mnt/pubx 目錄。

```
[root@kvm7 ~]# df -Th
Filesystem                Type      Size  Used Avail Use% Mounted on
devtmpfs                  devtmpfs  482M     0  482M   0% /dev
tmpfs                     tmpfs     496M     0  496M   0% /dev/shm
tmpfs                     tmpfs     496M   20M  476M   4% /run
tmpfs                     tmpfs     496M     0  496M   0% /sys/fs/
    cgroup
/dev/mapper/vg_kvm7usb-root ext4      3.1G  2.7G  188M  94% /
/dev/vda1                 xfs       197M  179M   19M  91% /boot
/dev/mapper/vg_kvm7home-vo  ext4     74M  1.6M   67M   3% /home
tmpfs                     tmpfs     100M     0  100M   0% /run/user/0
/dev/mapper/vg_mntx-pub   ext3      23M  209K   21M   1% /mnt/pubx
```

30.5 VG 刪除重建

1. 重建 VG 必須先將現有的 VG 刪除，現有的所有資料都將被刪除，若資料必須被保留下來，就要先備份，否則還是以修改方式較好。要刪除 VG 前必須先將該 VG 內的 LV FS 卸載，SWAP 關閉。

```
[root@kvm10 ~]# umount /dev/vg_mntx/pubx
[root@kvm10 ~]# swapoff /dev/vg_mntx/swap
```

2. 移除 VG 時會同時移除所有 LV，回答 y 即可一併刪除 LV。

```
[root@kvm10 ~]# vgremove vg_mntx
Do you really want to remove volume group "vg_mntx" containing 2 logical
    volumes? [y/n]: y
Do you really want to remove active logical volume vg_mntx/pubx? [y/n]:
    y
  Logical volume "pubx" successfully removed
Do you really want to remove active logical volume vg_mntx/swap? [y/n]:
    y
  Logical volume "swap" successfully removed
  Volume group "vg_mntx" successfully removed
[root@kvm7 ~]# umount /mnt/pub/
```

3. 再重新建立 VG, LV 即可。

30.6 sulogin 解決

1. 當開機自動掛載不成功時會出現以下訊息，輸入 root 密碼可進入單人模式維修，或按 Ctrl+d 繼續。

```
Welcome to emergency mode! After logging in, type "journalctl -xb" to
    view
system logs, "systemctl reboot" to reboot, "systemctl default" to try
    again
to boot into default mode.
Give root password for maintenance
(or type Control-D to continue):
```

2. 進入單人模式後執行 mount -a 依 /etc/fstab 掛載表重新掛載，發現 device vg_kvm7home-voo 不存在。

```
1 [root@kvm7 ~]# mount -a
  mount: special device /dev/mapper/vg_kvm7home-voo does not exist
```

3. 編輯 /etc/fstab 將 device vg_kvm7home-voo 改成正確的 vg_kvm7home-vo 後存檔。此例只是 device 打錯，實際狀況若是 device 無法掛載，且一下子無法解決，建議先將不能掛載的那行註解不掛載，也就是該行最前面加井字號 "#" 。

```
2 [root@kvm7 ~]# vim /etc/fstab
  [root@kvm7 ~]# grep vo /etc/fstab
  /dev/mapper/vg_kvm7home-vo /home                    ext4    defaults
            1 2
```

4. 再執行 mount -a 依 /etc/fstab 掛載表重新掛載，沒有任何訊息表示掛載成功。

```
1 [root@kvm7 ~]# mount -a
```

5. 強調再強調！！mount -a 不能出現任何錯誤訊息才能重新開機，否則一定變成 sulogin 無法正常開機。

```
1 [root@kvm7 ~]# reboot
```

Chapter 31

邏輯卷冊 – PE Size 調整

31.1 前言

1. PE 的大小是 LVM 中的最小儲存區塊，VG, LV 的大小都以幾個 PE 來計算。
2. PE 的大小只能是 2 的次方，例如：1M, 2M, 4M, 8M, 16M 等。
3. 當 PE 由大變小，VG 或 LV 的 PE 個數加倍。例如：VG 的 PE 大小 8M，有 100 個 PE，總大小爲 800M，若將 PE 大小修改爲 4M，則會有 200 個 PE，總大小還是 800M。
4. 但當 PE 由小變大，則可能出現無法剛好換成整數個 PE。
   (a) VG 大小不匹配：例如：VG 的 PE 大小 4M，有 201 個 PE，總大小爲 804M，若將 PE 大小修改爲 16M，則 VG 會有 50 個 PE 及 4M 的剩餘空間，這時就不能直接修改，必須先將 PV 的大小修改到整數個 PE 大小才可以，在此例子中就是要捨棄多出來的 4M。
   (b) LV 大小不匹配：例如：VG 的 PE 大小 4M，有 200 個 PE，總大小爲 800M，若將 PE 大小修改爲 16M，則 VG 會有 50 個 PE，不會有問題。但假設 VG 中有一 LV，原有 22 個 PE，也就是 88M，當 PE 調整成 16M 時，88M/16M=5.5 個，不是整數。這時必須先將 LV 擴展成 16M 的整數倍，再來修改 PE 大小。
5. PE size 由大變小不會有問題，由小變大依下列步驟檢查並修改：
   (a) New extent size is not a perfect fit : pvresize 來符合新 PE size 的倍數。
   (b) New size xxxxx for vg... free space not an exact number of new extents. : 先調整 VG 內的所有 LV (fs 及 swap) 大小來符合新 PE size 的倍數。

31.2 改變 PE 大小 - 必須改變 PV 大小

1. 先卸載 /mnt/pub。

```
[root@kvm10 ~]# umount /mnt/pub
```

2. PE 改變大小必須看「新的 PE 大小是否可整除 VG 大小」，以下列 VG vg_mntx 爲例，PE 大小是 2MiB，總數有 1497 個，VG 總大小爲 2MiB*1497 = 2994MiB。

```
[root@kvm10 ~]# vgdisplay vg_mntx | egrep '(Size|Total)'
  VG Size               2.92 GiB
  PE Size               2.00 MiB
  Total PE              1497
  Alloc PE / Size       36 / 72.00 MiB
  Free  PE / Size       1461 / 2.85 GiB
```

3. vg_mntx 的 PE size 由 2M 改成 4M，但因 1497*2M 不是 4M 的倍數，所以出現以下訊息：

```
[root@kvm10 ~]# vgchange -s 4M vg_mntx
  New extent size is not a perfect fit
```

4. 必須先將 Total PE 改成偶數，例如 1496，則 1496*2M 才能是 4M 的倍數。要修改 PE 的個數要使用 pvresize，先查看其輔助說明：

```
[root@kvm10 ~]# pvresize --help
  pvresize - Resize physical volume(s)

  pvresize PV ...
    [    --setphysicalvolumesize Size[m|UNIT] ]
    [    --reportformat basic|json ]
    [ COMMON_OPTIONS ]
```

5. pvresize 設定 phusical volume size 爲 2993M。(大小 1K 爲 1024，如果設定 2992m 會少一個)。

```
[root@kvm10 ~]# pvresize /dev/vda3 --setphysicalvolumesize 2993m
/dev/vda3: Requested size 2.92 GiB is less than real size <2.93 GiB.
    Proceed?  [y/n]: y
  WARNING: /dev/vda3: Pretending size is 6129664 not 6135569 sectors.
  Physical volume "/dev/vda3" changed
```

```
5   1 physical volume(s) resized or updated / 0 physical volume(s) not
        resized
```

6. 查看 vg_mntx 的 PE 總數改成 1496。

```
1 [root@kvm10 ~]# vgdisplay vg_mntx | egrep '(Size|Total)'
    VG Size               2.92 GiB
3   PE Size               2.00 MiB
    Total PE              1496
5   Alloc PE / Size       36 / 72.00 MiB
    Free  PE / Size       1460 / 2.85 GiB
```

7. 查看 vgchange 的輔助說明，-s 改變 PE 的大小。

```
  [root@kvm10 ~]# vgchange --help
2   vgchange - Change volume group attributes

4   Change a general VG attribute.
    For options listed in parentheses, any one is
6   required, after which the others are optional.
    vgchange
8     ( -l|--logicalvolume Number,
        -p|--maxphysicalvolumes Number,
10      -u|--uuid,
        -s|--physicalextentsize Size[m|UNIT],
```

8. vgchange 改變 PE 的大小爲 4M。

```
1 [root@kvm10 ~]# vgchange -s 4M vg_mntx
    Volume group "vg_mntx" successfully changed
```

9. 查看 vg_mntx 的 PE Size 已改成 4M。

```
  [root@kvm10 ~]# vgdisplay vg_mntx | egrep '(Size|Total)'
2   VG Size               2.92 GiB
    PE Size               4.00 MiB
4   Total PE              748
    Alloc PE / Size       18 / 72.00 MiB
6   Free  PE / Size       730 / 2.85 GiB
```

31.3 改變 PE 大小 - LV 的 PE 數量

1. VG vg_mntx 的 PE 大小由 2M 改成 4M 後，LV pubx 的 LE 個數由原來的 12 個自動改成 6 個，總大小不變，一樣是 24M。

```
[root@kvm10 ~]# lvdisplay /dev/vg_mntx/pubx
  --- Logical volume ---
  LV Path                /dev/vg_mntx/pubx
  LV Name                pubx
  VG Name                vg_mntx
  LV UUID                FUbp9Q-jIOJ-rR6r-H2sv-mNqv-rlv5-3jYXEH
  LV Write Access        read/write
  LV Creation host, time kvm10.deyu.wang, 2019-11-11 14:16:55 +0800
  LV Status              available
  # open                 0
  LV Size                24.00 MiB
  Current LE             6
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     8192
  Block device           253:3
```

2. 修改 LV pubx 的 LE 個數由原來的 6 個改成 50 個。

```
[root@kvm10 ~]# lvresize -l 50 -r /dev/vg_mntx/pubx
fsck from util-linux 2.32.1
/dev/mapper/vg_mntx-pubx: 11/6144 files (0.0% non-contiguous),
    2012/24576 blocks
  Size of logical volume vg_mntx/pubx changed from 24.00 MiB (6 extents)
      to 200.00 MiB (50 extents).
  Logical volume vg_mntx/pubx successfully resized.
resize2fs 1.44.3 (10-July-2018)
Resizing the filesystem on /dev/mapper/vg_mntx-pubx to 204800 (1k)
    blocks.
The filesystem on /dev/mapper/vg_mntx-pubx is now 204800 (1k) blocks
    long.
```

3. 再查看 LV pubx 的 LE 個數已改成 50 個。

```
[root@kvm10 ~]# lvdisplay /dev/vg_mntx/pubx | grep LE
  Current LE             50
```

31.4 改變 PE 大小 - 必須調整 LV

1. 先卸載 /mnt/pub。

```
[root@kvm10 ~]# umount /mnt/pub
```

2. PE 改變大小必須看「新的 PE 大小是否可整除 VG 大小」，以下列 VG vg_mntx 爲例，PE 大小是 4MiB，總數有 748 個，VG 總大小爲 4MiB*748 = 2992MiB。

```
[root@kvm10 ~]# vgdisplay vg_mntx | egrep '(Size|Total)'
  VG Size               2.92 GiB
  PE Size               4.00 MiB
  Total PE              748
  Alloc PE / Size       62 / 248.00 MiB
  Free  PE / Size       686 / <2.68 GiB
```

3. vg_mntx 的 PE size 由 4M 改成 16M (放大 4 倍)，2992MiB/16MiB=187 個，應該可以順利改變 PE 大小，但卻出現剩餘的空間不是剛好整數個 PE 的問題。

```
[root@kvm10 ~]# vgchange -s 16M vg_mntx
  New size 5619712 for vg_mntx free space not an exact number of new
      extents.
```

4. 原因是 LV pubx 用了 50 個 PE，swap 用了 12 個 PE，共用了 62 個，剩下 686 個，不是 4 的倍數，必須先修改 LV 的大小，先符合 16M 的倍數。

```
[root@kvm10 ~]# lvdisplay /dev/vg_mntx/pubx
  --- Logical volume ---
  LV Path                /dev/vg_mntx/pubx
  LV Name                pubx
  VG Name                vg_mntx
  LV UUID                FUbp9Q-jIOJ-rR6r-H2sv-mNqv-rlv5-3jYXEH
  LV Write Access        read/write
  LV Creation host, time kvm10.deyu.wang, 2019-11-11 14:16:55 +0800
  LV Status              available
  # open                 0
  LV Size                200.00 MiB
  Current LE             50
  Segments               2
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     8192
  Block device           253:3
```

5. 假設最後要調整成 PE 16M, LV 30 個 PE，也就是 480M，可以在 PE 還是 4M 時，先將 LV 的 PE 個數調整成 120 個，接著去調整 PE 大小後就不用再修改 LV 的 PE 個數。

```
[root@kvm10 ~]# lvresize -l 120 -r /dev/vg_mntx/pubx
fsck from util-linux 2.32.1
/dev/mapper/vg_mntx-pubx: clean, 11/51200 files, 8076/204800 blocks
  Size of logical volume vg_mntx/pubx changed from 200.00 MiB (50
      extents) to 480.00 MiB (120 extents).
  Logical volume vg_mntx/pubx successfully resized.
resize2fs 1.44.3 (10-July-2018)
Resizing the filesystem on /dev/mapper/vg_mntx-pubx to 491520 (1k)
    blocks.
The filesystem on /dev/mapper/vg_mntx-pubx is now 491520 (1k) blocks
    long.
```

6. 再查看 VG 剩餘 PE 數量是 616，616/4=154。

```
[root@kvm10 ~]# vgdisplay vg_mntx | egrep '(Size|Total)'
  VG Size               2.92 GiB
  PE Size               4.00 MiB
  Total PE              748
  Alloc PE / Size       132 / 528.00 MiB
  Free  PE / Size       616 / <2.41 GiB
```

7. vg_mntx 的 PE size 由 4M 改成 16M (放大 4 倍)。

```
[root@kvm10 ~]# vgchange -s 16M vg_mntx
  Volume group "vg_mntx" successfully changed
```

8. 查看 VG 的 PE Size 是 16M，總個數是 187。

```
[root@kvm10 ~]# vgdisplay vg_mntx | egrep '(Size|Total)'
  VG Size               2.92 GiB
  PE Size               16.00 MiB
  Total PE              187
  Alloc PE / Size       33 / 528.00 MiB
  Free  PE / Size       154 / <2.41 GiB
```

9. 查看 LV 的 PE 個數由 120 個變成 30 個，總大小還是 480M。

```
[root@kvm10 ~]# lvdisplay /dev/vg_mntx/pubx | grep LE
```

```
2   Current LE             30
```

31.5 改變 PE 大小 - 必須調整 SWAP

1. PE size 由大改小不會有問題，先將 PE size 由 16M 改回 4M。

```
[root@kvm10 ~]# vgchange -s 4M vg_mntx
2   Volume group "vg_mntx" successfully changed
```

2. 原有的 swap 48M 整除 4，也整除 16，所以 PE size 變更時不會有問題，先關閉 swap，將其大小改成 460M 並格式化。

```
[root@kvm10 ~]# swapoff /dev/vg_mntx/swap
2   [root@kvm10 ~]# lvresize -L 460M /dev/vg_mntx/swap
      Size of logical volume vg_mntx/swap changed from 48.00 MiB (12 extents
          ) to 460.00 MiB (115 extents).
4     Logical volume vg_mntx/swap successfully resized.
    [root@kvm10 ~]# mkswap /dev/vg_mntx/swap
6   Setting up swapspace version 1, size = 460 MiB (482340864 bytes)
    no label, UUID=33107790-884e-4a99-873d-6f384ba7d6bb
```

3. 再查看 VG 剩下的 PE 個數是 513 個，不是 4 的倍數。

```
1   [root@kvm10 ~]# vgdisplay vg_mntx | egrep '(Size|Total)'
      VG Size               2.92 GiB
3     PE Size               4.00 MiB
      Total PE              748
5     Alloc PE / Size       235 / 940.00 MiB
      Free  PE / Size       513 / 2.00 GiB
```

4. 這時 PE size 由 4M 改成 16M，一樣會出現大小不匹配的問題。

```
[root@kvm10 ~]# vgchange -s 16M vg_mntx
2   New size 4202496 for vg_mntx free space not an exact number of new
        extents.
```

5. 要先修改 SWAP 的大小符合 16M 的倍數，假設 SWAP 要求的大小是 460M，最接近 16M 整數倍的大小是 464M，所以先將 swap 改成 464M。

```
[root@kvm10 ~]# lvresize -L 464M /dev/vg_mntx/swap
  Size of logical volume vg_mntx/swap changed from 460.00 MiB (115
      extents) to 464.00 MiB (116 extents).
  Logical volume vg_mntx/swap successfully resized.
[root@kvm10 ~]# mkswap /dev/vg_mntx/swap
mkswap: /dev/vg_mntx/swap: warning: wiping old swap signature.
Setting up swapspace version 1, size = 464 MiB (486535168 bytes)
no label, UUID=6c96490f-167c-4e5e-9899-0d1cf40a3932
```

6. 再將 PE size 由 4M 改成 16M，成功變更大小。

```
[root@kvm10 ~]# vgchange -s 16M vg_mntx
  Volume group "vg_mntx" successfully changed
```

7. 再查看 VG 的 PE size 是 16M。

```
[root@kvm10 ~]# vgdisplay vg_mntx | egrep '(Size|Total)'
  VG Size               2.92 GiB
  PE Size               16.00 MiB
  Total PE              187
  Alloc PE / Size       59 / 944.00 MiB
  Free  PE / Size       128 / 2.00 GiB
```

Chapter 32

Tuned 性能調整

32.1 前言

1. Tuned 是一個強大的服務程序，可根據系統使用情況信息，自動動態調整 Linux 的設定，以提昇性能。
2. 自動動態調整設定取決於系統活動及使用的調整配置文件 (tuning profile)。
3. 預設情況下，tuned 不會動態調整系統設置，但可經由修改 tuned 服務程序的運行方式，允許它根據系統使用情況動態更改設置。
4. 可使用 tuned-adm 命令工具管理服務程序的運行。

32.2 Tuned 配置

1. /etc/tuned 目錄儲存配置檔，其中 tuned-main.conf 是主配置檔。

```
[root@kvm10 ~]# ll /etc/tuned
total 12
-rw-r--r--. 1 root root    0 Dec  1 19:14 active_profile
-rw-r--r--. 1 root root 1111 Feb  9  2022 bootcmdline
-rw-r--r--. 1 root root    0 Apr  1  2022 post_loaded_profile
-rw-r--r--. 1 root root    0 Apr  1  2022 profile_mode
drwxr-xr-x. 2 root root 4096 Apr  1  2022 recommend.d
-rw-r--r--. 1 root root 1730 Apr  1  2022 tuned-main.conf
```

2. /usr/lib/tuned 目錄儲存所有配置文件 (profiles) 的目錄。

```
[root@kvm10 ~]# ll /usr/lib/tuned
total 72
drwxr-xr-x. 2 root root  4096 Dec  1 19:14 accelerator-performance
```

```
drwxr-xr-x. 2 root root  4096 Dec  1 19:14 balanced
drwxr-xr-x. 2 root root  4096 Dec  1 19:14 desktop
-rw-r--r--. 1 root root 15373 Feb  9  2022 functions
drwxr-xr-x. 2 root root  4096 Dec  1 19:14 hpc-compute
drwxr-xr-x. 2 root root  4096 Dec  1 19:14 intel-sst
drwxr-xr-x. 2 root root  4096 Dec  1 19:14 latency-performance
drwxr-xr-x. 2 root root  4096 Dec  1 19:14 network-latency
drwxr-xr-x. 2 root root  4096 Dec  1 19:14 network-throughput
drwxr-xr-x. 2 root root  4096 Dec  1 19:14 optimize-serial-console
drwxr-xr-x. 2 root root  4096 Dec  1 19:14 powersave
drwxr-xr-x. 2 root root  4096 Dec  1 19:14 recommend.d
drwxr-xr-x. 2 root root  4096 Dec  1 19:14 throughput-performance
drwxr-xr-x. 2 root root  4096 Dec  1 19:14 virtual-guest
drwxr-xr-x. 2 root root  4096 Dec  1 19:14 virtual-host
```

32.3 啓動 Tuned

1. 查看 tuned 服務狀態，開機預設是啓動，且目前是正常運作中。

```
[root@kvm10 ~]# systemctl status tuned.service
  tuned.service - Dynamic System Tuning Daemon
   Loaded: loaded (/usr/lib/systemd/system/tuned.service; enabled;
       vendor prese>
   Active: active (running) since Wed 2019-11-06 14:40:00 CST; 3h 18min
       ago
     Docs: man:tuned(8)
           man:tuned.conf(5)
           man:tuned-adm(8)
 Main PID: 779 (tuned)
    Tasks: 4 (limit: 11532)
   Memory: 19.4M
   CGroup: /system.slice/tuned.service
             779 /usr/libexec/platform-python -Es /usr/sbin/tuned -l -P

Nov 06 14:39:52 kvm10.deyu.wang systemd[1]: Starting Dynamic System
    Tuning Daem>
Nov 06 14:40:00 kvm10.deyu.wang systemd[1]: Started Dynamic System
    Tuning Daemo>
```

32.4 tuned-adm

1. 查詢目前使用的 profile。

```
[root@kvm10 ~]# tuned-adm active
```

```
Current active profile: throughput-performance
```

2. 列出可用的 profile。

```
[root@kvm10 ~]# tuned-adm list
Available profiles:
- balanced                    - General non-specialized tuned profile
- desktop                     - Optimize for the desktop use-case
- latency-performance         - Optimize for deterministic performance
    at the
 cost of increased power consumption
- network-latency             - Optimize for deterministic performance
    at the
 cost of increased power consumption, focused on low latency network
     performance
- network-throughput          - Optimize for streaming network
    throughput,
 generally only necessary on older CPUs or 40G+ networks
- powersave                   - Optimize for low power consumption
- throughput-performance      - Broadly applicable tuning that provides
 excellent performance across a variety of common server workloads
- virtual-guest               - Optimize for running inside a virtual
    guest
- virtual-host                - Optimize for running KVM guests
Current active profile: throughput-performance
```

3. 查看建議使用的 profile。

```
[root@kvm10 ~]# tuned-adm recommend
atomic-guest
```

4. 將預設的 profile 切換到系統建議的 atomic-guest，但 atomic-guest profile 不存在。

```
[root@kvm10 ~]# tuned-adm profile atomic-guest
Requested profile 'atomic-guest' doesn't exist.
```

5. 安裝 tuned-profiles-atomic。

```
[root@kvm10 ~]# dnf install tuned-profiles-atomic-2.18.0-1.el9.noarch
```

6. 將預設的 profile 切換到系統建議的 atomic-guest。

```
[root@kvm10 ~]# tuned-adm profile atomic-guest
```

7. 再查看目前使用的 profile 已改成 atomic-guest。

```
[root@kvm10 ~]# tuned-adm active
Current active profile: atomic-guest
```

32.5 實機練習

1. ll 查詢 /usr/lib/tuned 目錄下的配置文件，導向到 /root/tunedprofiles。
2. 查詢 tuned 服務是否設定開機啓動，導向到 /root/tuned-enabled。
3. 查詢 tuned 服務現在是否啓動，導向到 /root/tuned-active。
4. 查詢 tuned 服務目前啓用的 profile，並將查詢命令寫成腳本 /root/tuned.active.sh。
5. 查詢 tuned 服務建議的 profile，並將命令寫成腳本 /root/tuned.recommend.sh。
6. 將預設的 profile 設定成到系統建議的 profile。