

# 磁碟檔案系統

崑山科技大學資訊傳播系

蔡德明

(鳥哥, VBird)

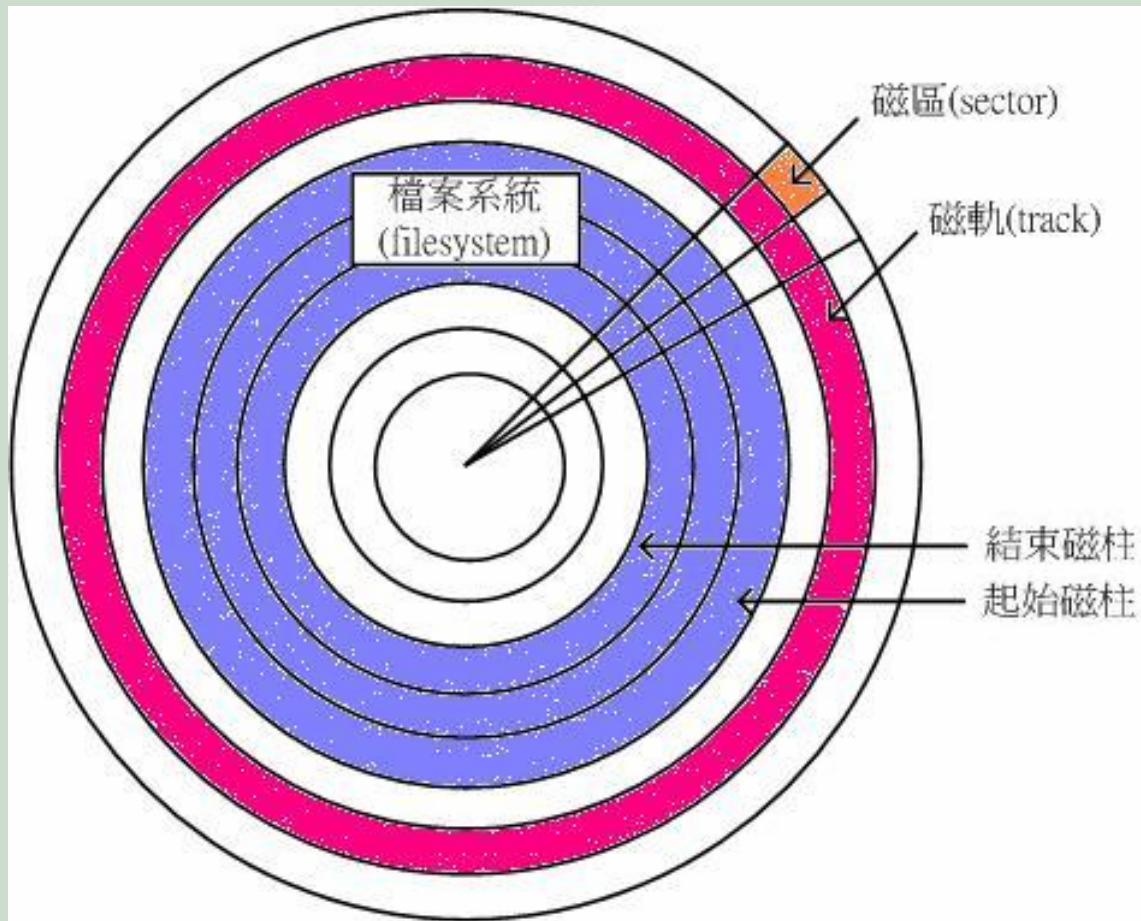
# 分享指引

- 磁碟分割(partition)
- 建立檔案系統(filesystem)
- 磁碟掛載 (mount)
- 檔案權限
- 檔案系統觀察與操作
- 壓縮指令



# 磁碟分割

# 磁碟組成



- 磁區：最小物理儲存量，為 512bytes
- 磁柱：partition 的單位

# 磁碟使用流程

- 新增一顆磁碟的方法
  - ☞ 硬體安裝、核心偵測
    - IDE裝置可直接被核心支援
    - SATA或USB磁碟，需有SCSI模組支援
  - ☞ partition(磁碟分割)
    - 透過 `fdisk` 指令的處理
  - ☞ format(格式化)
    - 透過 `mkfs` 或其他檔案系統工具 (`mke2fs`)
  - ☞ mount(掛載)
    - 與目錄樹結合在一起



# 硬體資訊

## ■ Linux硬體分類

☞ **character**：周邊設備，需一次讀取所有資訊

- 終端介面     /dev/tty[S]
- 印表機        /dev/lp0

☞ **block**：儲存設備，可隨機存取的裝置

- IDE            /dev/hd[a-d][1-63]
- SATA           /dev/sd[a-p][1-15]
- 光碟            /dev/cdrom
- 軟碟            /dev/fd0



# 分割表

## ■ 磁碟分割表

∞ 在第一個磁區(sector)僅佔有 64bytes

∞ 最多紀錄四筆主要紀錄，分割格式為

■ Primary 主要分割磁區，最多四筆

■ Extended 延伸分割區，最多一筆

∞ 可繼續延伸出邏輯分割區

∞ P+E 最多為四筆

■ Logical 邏輯分割區，可到 63 或 15 號

∞ 第一個邏輯分割區必定為 5 號。



# fdisk

## ■ fdisk /dev/hda

⌘ Command (m for help): p

⌘ Disk /dev/hda: 200.0 GB, 200049647616 bytes

⌘ 255 heads, 63 sectors/track, 24321 cylinders

⌘ Units = cylinders of 16065 \* 512 = 8225280 bytes

⌘	Device	Boot	Start	End	Blocks	Id	System
⌘	/dev/hda1	*	1	13	104391	83	Linux
⌘	/dev/hda2		14	2563	20482875	83	Linux
⌘	/dev/hda3		2564	3200	5116702+	83	Linux
⌘	/dev/hda4		3201	24321	169654432+	5	Extended
⌘	/dev/hda5		3201	4475	10241406	83	Linux
⌘	/dev/hda6		4476	5750	10241406	83	Linux
⌘	/dev/hda7		5751	6005	2048256	83	Linux
⌘	/dev/hda8		6006	6132	1020096	82	Linux swap



# fdisk (續)

## ■ fdisk /dev/hda

- ⌘ Command (m for help): m
- ⌘ Command action
- ⌘ a toggle a bootable flag
- ⌘ b edit bsd disklabel
- ⌘ c toggle the dos compatibility flag
- ⌘ d delete a partition
- ⌘ l list known partition types
- ⌘ m print this menu
- ⌘ n add a new partition
- ⌘ o create a new empty DOS partition table
- ⌘ p print the partition table
- ⌘ q quit without saving changes
- ⌘ s create a new empty Sun disklabel
- ⌘ t change a partition's system id
- ⌘ u change display/entry units
- ⌘ v verify the partition table
- ⌘ w write table to disk and exit
- ⌘ x extra functionality (experts only)



# System ID

## ■ Windows常見

- ☞ b W95 FAT32
- ☞ c W95 FAT32 (LBA)
- ☞ e W95 FAT16 (LBA)
- ☞ f W95 Ext'd (LBA)
- ☞ 7 HPFS/NTFS
- ☞ 86 NTFS volume set
- ☞ 87 NTFS volume set

## ■ Linux常見

- ☞ 80 Old Minix
- ☞ 81 Minix / old Lin
- ☞ 82 Linux swap
- ☞ 83 Linux
- ☞ 8e Linux LVM





# 建立檔案系統

# Linux 的檔案系統

## ■ Linux 檔案系統所記錄的資訊

### ☞ inode

- 記錄檔案屬性/權限，包括 **user, group, r, w, x**, 特殊權限, 三種時間參數, 檔案類型以及實際的檔案內容指向 (point)
- 每個inode為 128bytes
- 每個檔案需佔用一個 inode

### ☞ Block

- 實際記錄的檔案內容
- 傳統的EXT2/EXT3僅支援 1k, 2k, 4k大小



# Linux 的檔案系統(續)

## ■ 目錄樹的功能

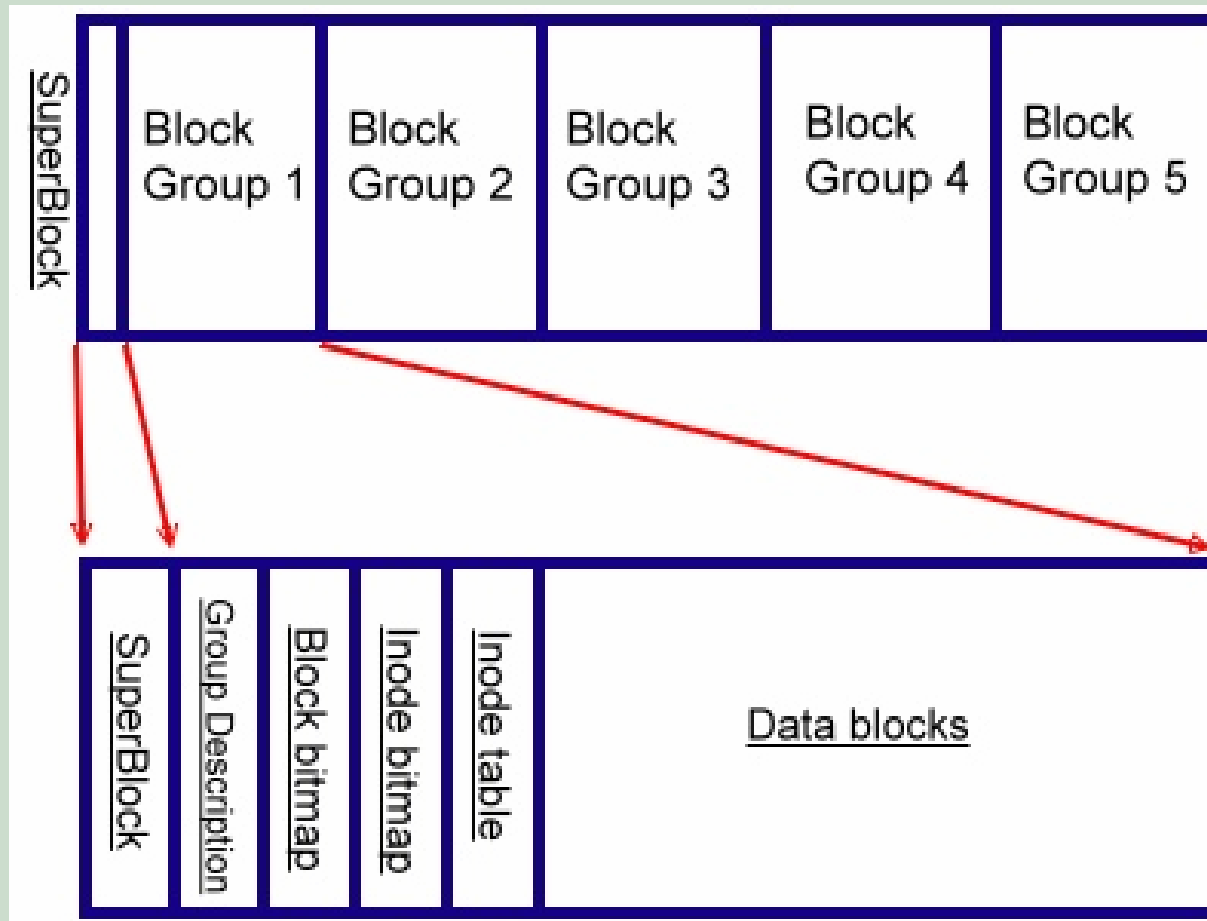
∞ 一般檔案：實際記錄資料者

∞ 目錄：記錄 **inode** 號碼與檔名的對應！

■ 所以具有目錄 **w** 權限者，在該目錄下就『無敵』了！



# 檔案系統格式



# 檔案系統的運作

- 新增一個檔案(目錄)的動作
  - ☞ 根據 inode bitmap / block bitmap 的資訊，找到尚未被使用的 inode 與 block，進而將檔案的屬性與資料分別記載進 inode 與 block；
  - ☞ 將剛剛被利用的 inode 與 block 的號碼 (number) 告知 superblock、inode bitmap、block bitmap 等，讓這些 metadata 更新資訊。
  - ☞ 問題：
    - 如果進行到一半，系統掛點呢？
    - fsck 會針對整個filesystem檢查，耗費時間
    - 日誌式檔案系統(journaling filesystem)的來源



# 日誌式檔案系統

- **Journaling filesystem 的作用：**
  - ☞ 當系統要寫入一個檔案的時候，會先在日誌記錄區塊中紀錄：某個檔案準備要寫入磁碟了；
  - ☞ 開始寫入檔案的權限與資料；
  - ☞ 開始更新 **metadata** 的資料；
  - ☞ 完成資料與 **metadata** 的更新後，在日誌記錄區塊當中完成該檔案的紀錄。
  - ☞ 好處：
    - 系統出問題時，可加速復原的動作
    - 將**filesystem**的活動做紀錄，可增加效率





# 格式化的指令

## ■ mkfs

☞ 這個指令為一個複合的指令，主要預設包括

- mkfs.cramfs mkfs.ext2 mkfs.ext3 mkfs.msdos mkfs.vfat
- ex> mkfs -t ext3 /dev/sda5

## ■ mke2fs

☞ ext2/ext3 的公用程式

- ex> mke2fs -j /dev/xxx → 建立 ext3 的檔案系統

## ■ mkswap

☞ 可建立虛擬記憶體的檔案格式



# 檔案系統支援

## ■ Linux支援的檔案系統

### ☞ 傳統檔案系統：

- ext2 / minix / MS-DOS / FAT (用 vfat 模組) / iso9660 (光碟)等等；

### ☞ 日誌式檔案系統：

- ext3 / ReiserFS / Windows' NTFS / IBM's JFS / SGI's XFS

### ☞ 網路檔案系統：

- NFS / SMBFS / CIFS



# 虛擬記憶體swap

## ■ swap的功能

- ☞ CPU處理的資料『一定是由實體記憶體RAM提供』
- ☞ 當實體記憶體不夠大時，則系統會將不常用的程式/資料移動到swap中，以空出更多的記憶體給程式使用

## ■ Linux swap的限制 (man mkswap)

- ☞ 一個swap理當不可超過2GB(i386架構)
- ☞ 最多可以支援的swap裝置可達 32 個
- ☞ 安裝時，建議最好要建置此swap裝置
- ☞ swap通常建議為實體記憶體的兩倍(但目前已沒需要)

# swap的使用

- swap裝置
  - ☞ 利用fdisk建立一個partition
  - ☞ 利用dd建立一個大檔案
    - `dd if=/dev/zero of=/some/file bs=1M count=1024`
- swap檔案格式
  - ☞ `mkswap /dev/sdaXX`
  - ☞ `mkswap /some/file`
- swap的使用
  - ☞ `swapon [-a] /dev/sdaXX`
- swap的移除
  - ☞ `swapoff /dev/sdaXX` → `cat /proc/swaps`





# 磁碟掛載

# 磁碟與目錄樹

## ■ Linux檔案系統的使用

- ☞ Linux的檔案使用目錄樹，任何資料都以檔案的型態存在
- ☞ 磁碟必須要跟目錄樹結合在一起後，才能夠使用磁碟內的資料
- ☞ 磁碟必須要『掛載』在目錄樹的『掛載點』上，掛載點為目錄！

## ■ 不能與根目錄分開的目錄

- ☞ 由於開機會使用到某些重要目錄，下列目錄不可與根目錄分離
  - /etc, /bin, /sbin, /lib, /dev



# mount語法

## ■ 磁碟掛載使用 mount

☞ `mount -t [fstype] [device] [mount point]`

- `mount -t iso9660 /dev/cdrom /media/cdrom`
- `mount -t ext3 /dev/sda5 /mnt/`
- `mount -t vfat /dev/sdb1 /mnt/windows`

## ■ 預設會直接掛載的檔案系統

☞ `/etc/filesystem`

☞ `/proc/filesystems`

- `mount /dev/sda5 /mnt`

## ■ 中文檔名的掛載

☞ `mount -t vfat -o iocharset=cp950 /dev/hda1 /mnt/win`



# 特殊掛載方式

## ■ 利用『-o 參數』特殊掛載

☞ mount -o remount,rw / → 不卸載重新掛載

☞ mount -o loop /path/file.img /mnt → 掛載映象檔

☞ mount -L 'label\_name' /mnt → 利用label名稱

☞ mount --bind /home /mnt/home → 掛載目錄

## ■ 卸載

☞ umount [掛載點|裝置]

☞ umount /home

☞ umount /dev/sda5





# 開機自動掛載

## ■ 掛載的一些限制

- ❧ 根目錄 / 是必須，且最早被掛載起來的
- ❧ 其它掛載點必須為已建立的目錄，可任意指定，但一定要遵守必須的系統目錄架構原則
- ❧ 所有掛載點在同一時間之內，只能掛載一次。
- ❧ 所有 **partition** 在同一時間之內，只能掛載一次。
- ❧ 如若進行卸載，您必須先將工作目錄移到 **mount point**(及其子目錄) 之外。



# 開機自動掛載(續)

## ■ 開機自動掛載的設定檔 `/etc/fstab` 語法

# Device	Mount point	filesystem	parameters	dump	fsck
LABEL=/ /dev/hda5	/ /home	ext3	defaults	1	1
/dev/hda3	swap	swap	defaults	0	0
/dev/hdc	/media/cdrom	auto	pamconsole,exec,noauto,managed		

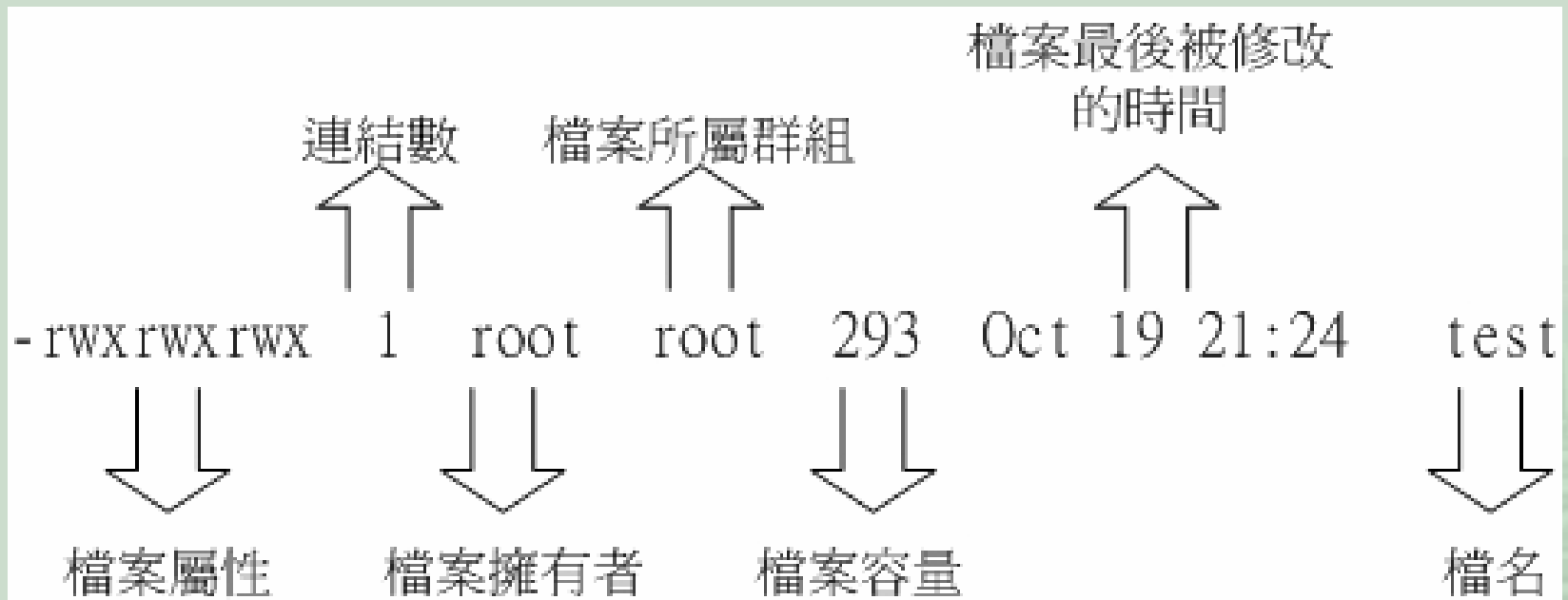
0 0





# 檔案權限

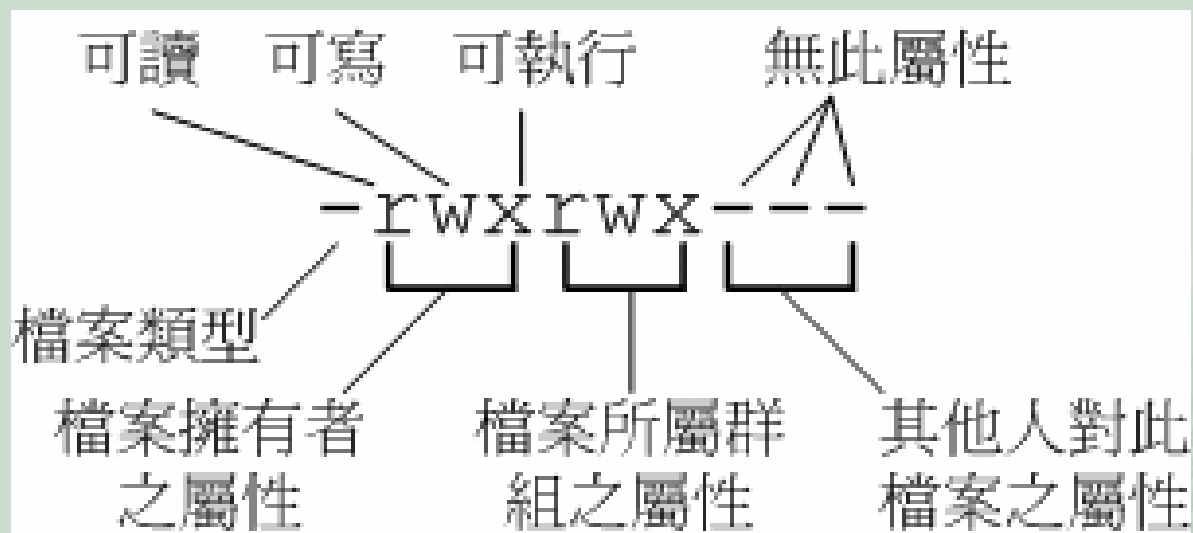
# 檔案的屬性



- 共分為七個欄位，其中第一個欄位為檔案的權限



# 檔案的權限



## ■ 檔案類型

☞ [ d ] 是目錄

☞ [ - ] 是檔案

☞ [ l ] 為連結檔(link file)；

☞ [ b ] 為裝置檔裡面的可供儲存的周邊設備；

☞ [ c ] 則表示為裝置檔裡面的序列埠設備



# 檔案/目錄權限的意義

## ■ 檔案與目錄

- ☞ 檔案可記錄實際的資料
- ☞ 目錄記錄的是『該目錄下的檔名』

## ■ 一般檔案

- ☞ r 可讀取該檔案的內容
- ☞ w 可修改/編輯檔案的內容
- ☞ x 可執行該檔案

## ■ 目錄檔案

- ☞ r 可讀取目錄內記錄的檔名
- ☞ w 可在該目錄內建立/刪除檔名
- ☞ x 可進入該目錄



# 權限的應用

- 進入某目錄成爲『可工作目錄』的基本權限
  - ☞ 使用者可進入該目錄需具備  $x$  權限
  - ☞ 使用者可在該目錄查閱檔名，需具備  $r$  權限
- 讀取一個檔案的基本權限
  - ☞ 使用者在該檔案所在的目錄至少要有  $x$  權限
  - ☞ 使用者對該檔案至少要有  $r$  權限
- 修改一個檔案的基本權限
  - ☞ 使用者在該檔案所在的目錄至少要有  $x$  權限
  - ☞ 使用者對該檔案至少要有  $r, w$  權限



# 權限的應用(續)

- 建立一個檔案的基本權限
  - ∞ 使用者在該目錄要具有 **w,x** 的權限
- 進入某目錄並執行該目錄下的某個指令
  - ∞ 使用者在該目錄至少要有 **x** 的權限
  - ∞ 使用者在該檔案至少需要 **x** 的權限
- 思考
  - ∞ 對於 **cp** 來說，來源/目標的權限各是如何？



# 檔案屬性的修改

- 修改檔案擁有者 (owner)
  - ☞ `chown [-R] user[:group] 檔案`
- 修改檔案擁有群組 (group)
  - ☞ `chgrp [-R] group 檔案`
- 修改檔案的日期/時間
  - ☞ 透過 `touch`



# 檔案權限的修改

chmod	u g o a	+(加入) -(除去) =(設定)	r w x	檔案或目錄
-------	------------------	-------------------------	-------------	-------

## ■ 符號法

☞ 讓所有人都可以執行 `run.sh` 程式

■ `chmod a+x run.sh`



# 檔案權限的修改(續)

## ■ 利用數字法

☞ 三種身份，每種身份有 r, w, x

■ r 4分

■ w 2分

■ x 1分

☞ 將檔案改成 `-rwxr-xr--` 的分數 → 754



# 預設權限umask

- 每個使用者新建檔案時，該檔案的預設權限
  - ☞ `umask` → 顯示的結果為『拿掉的權限』
  - ☞ 預設權限在目錄/檔案並不相同
    - 目錄：`drwxrwxrwx` (因為 `x` 對目錄很重要)
    - 檔案：`-rw-rw-rw-` (因為檔案預設並不可執行)
  - ☞ `ex> umask 002` 的新建資料
    - 目錄：`drwxrwxr-x`
    - 檔案：`-rw-rw-r--`





# 檔案系統觀察與操作

# 整體檔案系統觀察

## ■ 查詢partition的使用量與剩餘量

☞ `df [-hi] [目錄]`

- `df -h` → 用G,M,K表示partition使用情況
- `df /tmp` → 顯示/tmp還能夠用多少容量

## ■ 查詢某目錄的總容量

☞ `du [-s] [目錄]`

- `du -sm` → 以MB為單位，進行該目錄所有檔案的加總之容量



# 目錄的操作

- 特殊目錄，需要特別注意：

- ☞ . 代表此層目錄

- ☞ .. 代表上一層目錄

- ☞ - 代表前一個工作目錄

- ☞ ~ 代表『目前使用者身份』所在的家目錄

- ☞ ~account 代表 **account** 這個使用者的家目錄

- 常用指令：

- ☞ cd : 變換目錄

- ☞ pwd : 顯示目前的目錄

- ☞ mkdir : 建立一個新的目錄

- ☞ rmdir : 刪除一個空的目錄



# 檔案的複製

```
[root@linux ~]# cp [-adfilprsu] 來源檔(source) 目的檔(destination)
[root@linux ~]# cp [options] source1 source2 source3 .... directory
```

參數：

- a : 相當於 -pdr 的意思；
- d : 若來源檔為連結檔的屬性(link file)，則複製連結檔屬性而非檔案本身；
- f : 為強制 (force) 的意思，若有重複或其他疑問時，不會詢問使用者，而強制複製；
- i : 若目的檔(destination)已經存在時，在覆蓋時會先詢問是否真的動作！
- l : 進行硬式連結 (hard link) 的連結檔建立，而非複製檔案本身；
- p : 連同檔案的屬性一起複製過去，而非使用預設屬性；
- r : 遞迴持續複製，用於目錄的複製行為；
- s : 複製成為符號連結檔 (symbolic link)，亦即『捷徑』檔案；
- u : 若 destination 比 source 舊才更新 destination ！

最後需要注意的，如果來源檔有兩個以上，則最後一個目的檔一定要是『目錄』才行！

範例：

範例一：將家目錄下的 .bashrc 複製到 /tmp 下，並更名為 bashrc

```
[root@linux ~]# cd /tmp
[root@linux tmp]# cp ~/.bashrc bashrc
[root@linux tmp]# cp -i ~/.bashrc bashrc
cp: overwrite `bashrc'? n
```



# 檔案的刪除與移動

```
[root@linux ~]# rm [-fir] 檔案或目錄
```

參數：

- f : 就是 force 的意思，強制移除；
- i : 互動模式，在刪除前會詢問使用者是否動作
- r : 遞迴刪除啊！最常用在目錄的刪除了

```
[root@linux ~]# mv [-fiu] source destination
```

```
[root@linux ~]# mv [options] source1 source2 source3 .... directory
```

參數：

- f : force 強制的意思，強制直接移動而不詢問；
- i : 若目標檔案 (destination) 已經存在時，就會詢問是否覆蓋！
- u : 若目標檔案已經存在，且 source 比較新，才會更新 (update)

範例：

範例一：複製一檔案，建立一目錄，將檔案移動到目錄中

```
[root@linux ~]# cd /tmp
```

```
[root@linux tmp]# cp ~/.bashrc bashrc
```

```
[root@linux tmp]# mkdir mvtest
```

```
[root@linux tmp]# mv bashrc mvtest
```

# 將某個檔案移動到某個目錄去，就是這樣做！

範例二：將剛剛的目錄名稱更名為 mvtest2

```
[root@linux tmp]# mv mvtest mvtest2 <== 這樣就更名了！簡單～
```



# 壓縮指令

# 壓縮指令

## ■ 常見的壓縮指令

### ⌘ compress

- 最老牌的，幾乎所有的Unix Like 『以前』都有
- 副檔名為 \*.Z

### ⌘ gzip

- GNU出的，取代了 compress，支援度佳的壓縮指令
- 副檔名為 \*.gz，可使用 『gzip -d file.gz』 解壓縮

### ⌘ bzip2

- 比 gzip 壓縮比更好
- 副檔名為 \*.bz2，可使用 『bzip2 -d file.bz2』 解壓縮



# 打包指令

- 檔案的打包與壓縮：透過**tar**，其參數有：
  - ☞ 建立、解壓縮與查閱的參數：
    - **-c** : 建立一個壓縮檔案的參數指令(**create**的意思)
    - **-x** : 解開一個壓縮檔案的參數指令！
    - **-t** : 查看 **tarfile** 裡面的檔案
  - ☞ 是否進行壓縮的參數：
    - **-j** : 使用 **bzip2** 壓縮/解壓縮
    - **-z** : 使用 **gzip** 壓縮/解壓縮
  - ☞ 其他重要參數：
    - **-p** : 保持檔案原有的屬性/權限
    - **-v** : 查閱打包過程的資訊
    - **-f file** : 後續處理的檔名！



# 打包指令(續)

- 將 `/etc` 建立打包檔案到 `/root/` 底下
  - ☞ `tar -pzcvf /root/etc.tar.gz /etc`
  - ☞ `tar -pjcvf /root/etc.tar.bz2 /etc`
- 查閱剛剛建立的檔案資訊
  - ☞ `tar -ztvf /root/etc.tar.gz`
- 將該檔案解壓縮到 `/tmp` 底下
  - ☞ `tar -zxvf /root/etc.tar.gz -C /tmp`



# 副檔名與壓縮指令

## ■ 副檔名的意義

☞ Linux的副檔名意義不大，幾乎都是用來提醒使用者的，但有些軟體則與副檔名有關

☞ 壓縮檔案的副檔名與壓縮指令的對應

- \*.Z compress 壓縮的檔案；
- \*.bz2 bzip2 壓縮的檔案；
- \*.gz gzip 程式壓縮的檔案；
- \*.tar tar 打包的資料，並沒有壓縮過；
- \*.tar.gz tar 打包的檔案，並且經過 gzip 的壓縮
- \*.tar.bz2 tar 打包的檔案，並且經過 gzip 的壓縮