

基礎檔案系統管理

Reading material

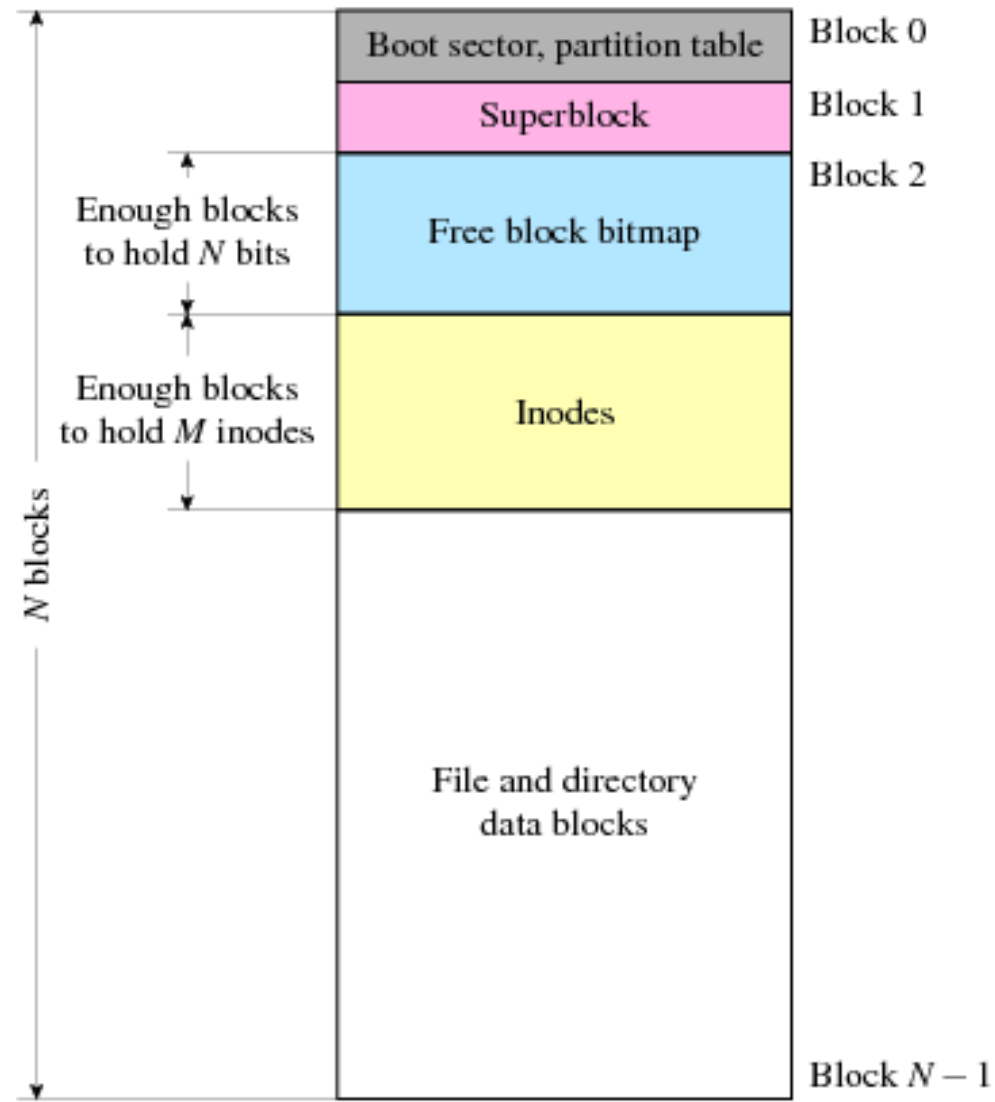
- 基礎檔案系統管理
 - http://linux.vbird.org/linux_basic_train/unit06.php#6.2
- linux 磁碟與檔案系統管理
 - http://linux.vbird.org/linux_basic/0230filesystem.php#link

認識 Linux 檔案系統

- 整顆磁碟必須要經過分割之後，Linux 才能夠讀取分割槽內的檔案系統。
- Linux 磁碟分割主要有兩種分割類型：
 - 早期的 MBR
 - 有 2TB 容量的限制，但虛擬機的硬碟不會大過2TB，所以上課用MBR說明
 - 現今的 GPT
 - 無 2TB 容量的限制

Linux 的 EXT2 檔案系統

- 分割完後，要經過格式化成不同檔案系統才能使用
 - 檔案系統的功用在於記錄檔案的屬性(如權限、時間、身份資料紀錄等)以及實際資料的紀錄。
- Linux的EXT2，包含：
 - Superblock
 - 記錄此 filesystem 的整體資訊，包括inode/block的總量、使用量、剩餘量，以及檔案系統的格式與相關資訊等；
 - Inode
 - 記錄檔案的屬性，一個檔案佔用一個inode，同時記錄此檔案的資料所在的 block 號碼；
 - Block
 - 實際記錄檔案的內容，若檔案太大時，會佔用多個 block 。



檔案系統讀寫檔案的流程

- 讀檔案
 - 讀到檔案的 **inode** 號碼
 - 由 **inode** 內的權限設定判定使用者能否存取此檔案
 - 若能讀取則開始讀取 **inode** 內所紀錄的資料放置於哪些 **block** 號碼內
 - 讀出 **block** 號碼內的資料，組合起來成為一個檔案的實際內容。
- 建立檔案
 - 有寫入檔案的需求時，先到 **metadata** 區找到沒有使用中的 **inode** 號碼
 - 到該 **inode** 號碼內，將所需要的權限與屬性相關資料寫入，然後在 **metadata** 區規範該 **inode** 為使用中，且更新 **superblock** 資訊
 - 到 **metadata** 區找到沒有使用中的 **block** 號碼，將所需要的實際資料寫入 **block** 當中，若資料量太大，則繼續到 **metadata** 當中找到更多的未使用中的 **block** 號碼，持續寫入，直到寫完資料為止。
 - 同步更新 **inode** 的紀錄與 **superblock** 的內容。

目錄與檔名

- 讀取檔案資料時，要讀到檔案的 **inode** 號碼
 - 實際操作系統時，是透過『檔名』來讀寫資料的
- 建立目錄時，會分配一個**inode**與一個**block**
 - Inode 記錄相關權限與屬性
 - block 會記錄檔名與 inode 號碼對應

Inode number	檔名
53735697	anaconda-ks.cfg
53745858	initial-setup-ks.cfg
...	...

例題：

- 使用 `ls -ali /etc/hosts*`，觀察出現在最前面的數值，該數值即為 `inode` 號碼。
- 建立 `/tmp/inodecheck/` 目錄，
 - 觀察 `/tmp/inodecheck/`, `/tmp/inodecheck/.` 這兩個檔名的 `inode` 號碼
 - 觀察 `/tmp/` , 與 `/tmp/inodecheck/..` 這二個檔名的 `inode` 號碼
 - 這說明了什麼事實

實體連結、hard link

- 每個檔案都會佔用一個 **inode**，檔案內容由 **inode** 的記錄來指向；
- 想要讀取該檔案，必須要經過目錄記錄的檔名來指向到正確的 **inode** 號碼才能讀取。
 - 多個檔名可以對應至一個 **inode**

```
[student@localhost ~]$ ls -ld /var/spool/mail
drwxrwxr-x. 2 root mail 4096 6月 29 03:29 /var/spool/mail
[   A   ][B][C ][D ][E ][   F   ][   G   ]
```

- B: 檔案連結數，這與檔案系統有關，讀者可暫時略過
 - 對應至相同 **inode** 的檔名數量

- 使用 **hard link** 設定連結檔時，磁碟的空間與 **inode** 的數目都不會改變
 - 只是在目錄裡加一筆**inode**與檔名的對應
- 將任何一個『檔名』刪除，其實 **inode** 與 **block** 都還是存在的
 - 直到**link count**為0
- 不論使用哪個『檔名』來編輯，最終的結果都會寫入到相同的 **inode** 與 **block**

建立實體連結

```
[root@study ~]# ll -i /etc/crontab
34474855 -rw-r--r--. 1 root root 451 Jun 10 2014 /etc/crontab

[root@study ~]# ln /etc/crontab . <==建立實體連結的指令
[root@study ~]# ll -i /etc/crontab crontab
34474855 -rw-r--r--. 2 root root 451 Jun 10 2014 crontab
34474855 -rw-r--r--. 2 root root 451 Jun 10 2014 /etc/crontab
```

例題：

- 建立 `/tmp/inodecheck-2/` 目錄，檢查此目錄的link數，這說明什麼現象。
- 建立 `/tmp/inodecheck-2/new` 目錄，再檢查此 `/tmp/inodecheck-2/` 目錄的link數，這說明什麼現象。
- 建立 `/tmp/inodecheck-3` 檔案，檢查此檔案的link數，這說明什麼現象。
- 建立一個hard link指到 `/tmp/inodecheck-3`，檢查此檔案的link數，這說明什麼現象。

符號連結、soft link、Symbolic Link

- Symbolic link 就是在建立一個獨立的檔案，這個獨立的檔案會指向目的檔案，當資料讀取時，會指向他 link 的那個檔案的檔名
- Symbolic link 所建立的檔案為一個獨立的新的檔案，會佔用掉 inode 與 block 喔！
 - Link 檔案大小為路徑名稱長度
- 當目的檔被刪除後，因為找不到原始檔名，link 會變成死link，而無法開啟檔案

建立符號連結

```
[root@study ~]# ln -s /etc/crontab crontab2  
[root@study ~]# ll -i /etc/crontab /root/crontab2  
34474855 -rw-r--r--. 2 root root 451 Jun 10 2014 /etc/crontab  
53745909 lrwxrwxrwx. 1 root root 12 Jun 23 22:31 /root/crontab2 -> /etc/crontab
```

- Hard link 限制較多
 - 不能跨 Filesystem ；
 - link 目錄會有問題。
- 一般常使用soft link

例題：

- 將 `/etc/hosts` 複製到家目錄下，同時觀察檔名連結數
- 使用『`ln hosts hosts.real`』建立 `hosts.real` 實體連結檔，同時觀察這兩個檔案的 `inode` 號碼、屬性權限等，是否完全相同？為什麼？
- 使用『`ln -s hosts hosts.symbo`』建立 `hosts.symbo` 符號連結，同時觀察這兩個檔案的 `inode` 號碼、屬性權限等，是否相同？
- 使用 `cat hosts; cat hosts.real; cat hosts.symbo`，查閱檔案內容是否相同？
- 請刪除 `hosts`，然後觀察 `hosts.real`, `hosts.symbo` 的 `inode` 號碼、連結數檔案屬性等資料，發現什麼情況？
- 使用 `cat hosts.real; cat hosts.symbo` 發生什麼狀況

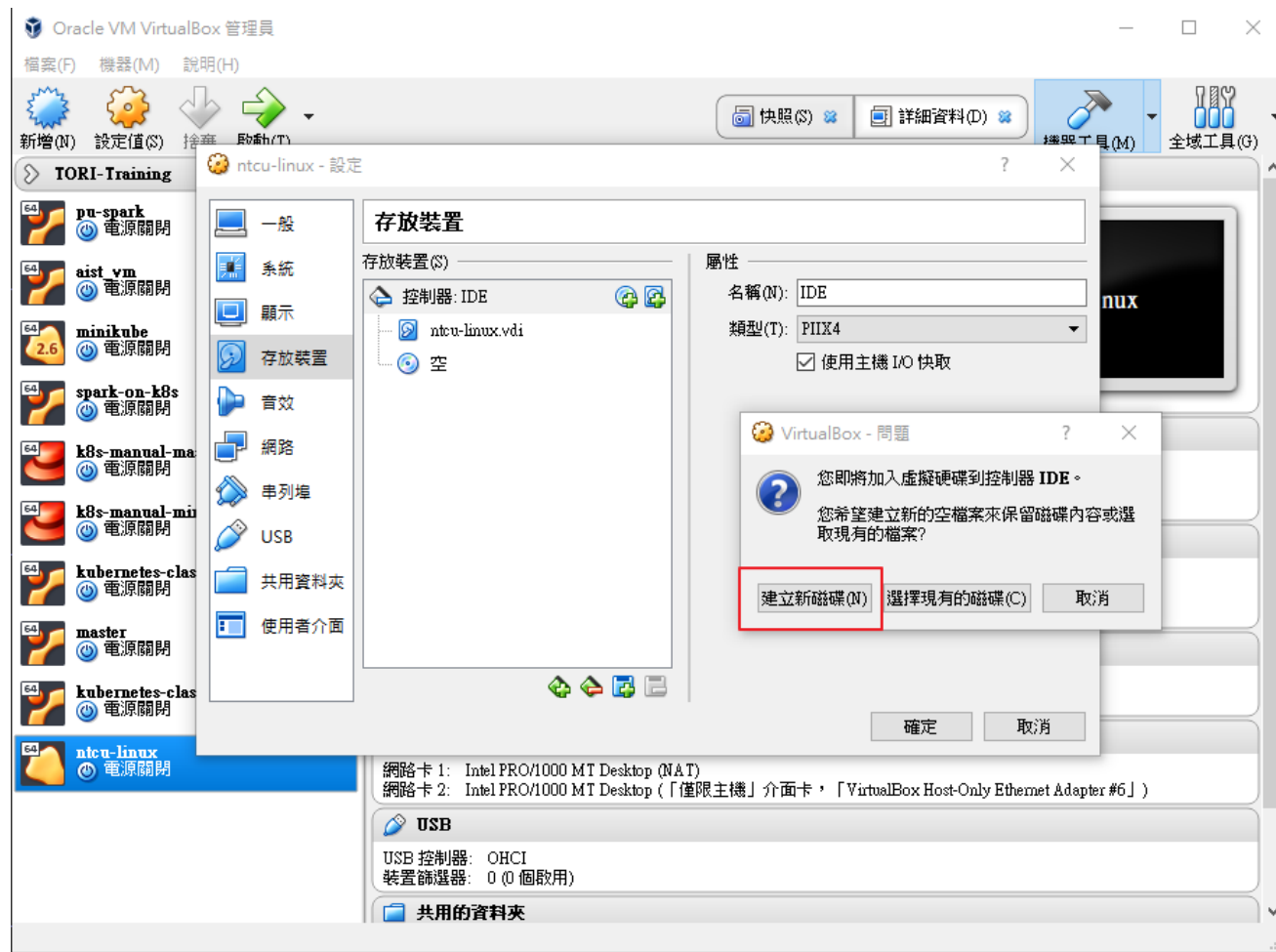
檔案系統的掛載

- 隨身碟放入 windows 作業系統後，需要取得一個 H:\> 才能夠被讀取一樣
- Linux 底下的目錄樹系統中，檔案系統裝置要能夠被讀取，就得要與目錄樹的某個目錄連結在一起，該目錄就被稱為掛載點。

檔案系統管理

- 1. 安裝新硬碟 (在virtualbox上做)
- 2. 建立分割
- 3. 建立檔案系統 (磁碟格式化)
- 4. 檔案系統的掛載/卸載
- 5. 開機自動掛載

新增硬碟



```
[root@localhost ~]# ls /dev/sd*  
/dev/sda /dev/sda1 /dev/sda2 /dev/sdb  
[root@localhost ~]#
```

例題：新增硬碟

- 在virtualbox裡新增一顆10GB的硬碟至虛擬機器
- 檢查/dev/目錄下是否有多一顆sdb，依每人的虛擬機器情況不同，也有可能是sdc，sdd...等等，但請確認自己的新硬碟編號

建立分割

- 以root執行lsblk，檢查目前硬碟與分割狀況

```
[root@localhost ~]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda          8:0    0   20G  0 disk
├─sda1       8:1    0   10G  0 part /
└─sda2       8:2    0   10G  0 part /home
sdb          8:16   0    8G  0 disk
sr0         11:0    1 1024M  0 rom
```

- 使用fdisk建立分割 (mbr模式)

```
[root@study ~]# fdisk /dev/sda
Command (m for help): m  <== 輸入 m 後，就會看到底下這些指令介紹
Command action
  a   toggle a bootable flag
  b   edit bsd disklabel
  c   toggle the dos compatibility flag
  d   delete a partition      <==刪除一個partition
  l   list known partition types
  m   print this menu
  n   add a new partition      <==新增一個partition
  o   create a new empty DOS partition table
  p   print the partition table  <==在螢幕上顯示分割表
  q   quit without saving changes  <==不儲存離開fdisk程式
  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  <==將剛剛的動作寫入分割表
  x   extra functionality (experts only)
```

建立第一個分割

```
Command (m for help): n
Partition type:
   p   primary (0 primary, 0 extended, 4 free)
   e   extended
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-16777215, default 2048):
Using default value 2048
Last sector, +sectors or +size{K,M,G} (2048-16777215, default 16777215): +4G
Partition 1 of type Linux and of size 4 GiB is set
```

```
Command (m for help): p

Disk /dev/sdb: 8589 MB, 8589934592 bytes, 16777216 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x47e6766c
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sdb1		2048	8390655	4194304	83	Linux

建立第二個分割

```
Command (m for help): n
Partition type:
   p   primary (1 primary, 0 extended, 3 free)
   e   extended
Select (default p): p
Partition number (2-4, default 2):
First sector (8390656-16777215, default 8390656):
Using default value 8390656
Last sector, +sectors or +size{K,M,G} (8390656-16777215, default 16777215):
Using default value 16777215
Partition 2 of type Linux and of size 4 GiB is set
```

```
Command (m for help): p

Disk /dev/sdb: 8589 MB, 8589934592 bytes, 16777216 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x47e6766c
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sdb1		2048	8390655	4194304	83	Linux
/dev/sdb2		8390656	16777215	4193280	83	Linux

寫入分割內容

```
Command (m for help): w  
The partition table has been altered!
```

例題：建立分割

- 建立三個分割區，請試著自行找出type的編碼
 - Sdb1: 4gb，做為linux使用，type為83
 - Sdb1: 4gb，做為windows/dos使用，type為b
 - Sdb3: 2gb，做為swap，type為82

建立檔案系統

- 檔案系統的建立使用 `mkfs` 即可處理。另外，記憶體置換應該要使用 `mkswap`
- 使用 `mkfs.xfs` 建立好 XFS 檔案系統
- 使用 `mkswap` 建立好 swap 記憶體置換空間

```
[root@localhost ~]# mkfs.xfs /dev/sdb1
meta-data=/dev/sdb1            isize=512    agcount=4, agsize=196608 blks
                               =               sectsz=512   attr=2, projid32bit=1
                               =               crc=1        finobt=0, sparse=0
data      =                     bsize=4096   blocks=786432, imaxpct=25
                               =               sunit=0      swidth=0 blks
naming    =version 2           bsize=4096   ascii-ci=0 ftype=1
log        =internal log       bsize=4096   blocks=2560, version=2
                               =               sectsz=512   sunit=0 blks, lazy-count=1
realtime  =none                extsz=4096   blocks=0, rtextents=0
```

```
[root@localhost ~]# mkswap /dev/sdb3
Setting up swapspace version 1, size = 2096124 KiB
no label, UUID=1240b811-3a85-4c11-b6f0-967a503ae8c2
```

例題：建立檔案系統

- 使用mkfs.xfs 建立好 XFS 檔案系統
- 使用mkfs.vfat建立好vfat檔案系統
 - 可能需要用 **yum install dosfstools**安裝 **mkfs.vfat**
- 使用mkswap 建立好 swap 記憶體置換空間

掛載

- 格式化後的硬碟，要掛載後才能使用
- 注意：
 - 單一檔案系統不應該被重複掛載在不同的掛載點(目錄)中；
 - 單一目錄不應該重複掛載多個檔案系統；
 - 要作為掛載點的目錄，理論上應該都是空目錄才是。

- 使用mount掛載

```
[root@localhost ~]# mkdir /srv/linux /srv/win
[root@localhost ~]# mount /dev/vda4 /srv/linux
[root@localhost ~]# mount /dev/vda5 /srv/win
[root@localhost ~]# df -T /srv/linux /srv/win
```

檔案系統	類型	1K-區段	已用	可用	已用%	掛載點
/dev/vda4	xfs	1038336	32928	1005408	4%	/srv/linux
/dev/vda5	vfat	1532988	4	1532984	1%	/srv/win

- 使用swapon掛載swap

```
[root@localhost ~]# swapon /dev/vda6
[root@localhost ~]# swapon -s
```

Filename	Type	Size	Used	Priority
/dev/dm-1	partition	2097148	3752	-1
/dev/vda6	partition	1048572	0	-2

例題：掛載

- 掛載分割區
 - 將sdb1掛載至/srv/linux
 - 將sdb2掛載至/srv/win
 - 掛載sdb3的swap
- 用umount與swapoff 解除掛載

自動掛載

- 使用mount掛載，重開機後並不會自動掛載，需將掛載的資訊寫入至/etc/fstab裡，開機後會依據些檔進行掛載
- 系統掛載限制
 - 根目錄 / 是必須掛載的，而且一定要先於其它 mount point 被掛載進來。
 - 其它 mount point 必須為已建立的目錄，
 - 所有 mount point 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。

/etc/fstab

```
[root@localhost ~]# cat /etc/fstab
/dev/mapper/centos-root / xfs defaults 0 0
UUID=a026bf1c-3028-4962-88e3-cd92c6a2a877 /boot xfs defaults 0 0
/dev/mapper/centos-home /home xfs defaults 0 0
/dev/mapper/centos-swap swap swap defaults 0 0
```

[裝置/UUID等]	[掛載點]	[檔案系統]	[檔案系統參數]	[dump]	[fsck]
------------	-------	--------	----------	--------	--------

例題：自動掛載

- 使用blkid找出新分割區的uuid，
- 使用 vim 在 /etc/fstab 最底下新增三行資料

```
[root@localhost ~]# vim /etc/fstab
UUID="2a409620-c888-41ca-89fa-2737cca74f19" /srv/linux xfs defaults 0 0
UUID="4AF7-0017" /srv/win vfat defaults 0 0
UUID="de7e7a05-7b54-40c3-b663-142e4d545265" swap swap defaults 0 0
```

- 使用mount -a /swapon -a對進行掛載

```
[root@localhost ~]# mount -a
[root@localhost ~]# swapon -a
[root@localhost ~]# df -T /dev/vda4 /dev/vda5
[root@localhost ~]# swapon -s
```

- 重開機後，觀查是否有全數掛載