Linux File System

魏凡琮 (Jerry Wei)

Agenda

- Linux FileSystem
- Partition
- LVM
- RAID
- Q&A

FileSystem

什麼是 FileSystem?

- Disk File System. 。
- 儲存和組織電腦檔案和資料的方法。
- 可以存放檔案及取回檔案的系統。
- Everything is file, file is everything.

FileSystem

選擇FileSystem ：

- 穩定性。
- 容量。
- 擴充性。
- 效能。

FileSystem

FileSystem 種類：

- 傳統 (非日誌式) 檔案系統。
  - ext2 、 ms-dos 、 VFAT....etc.

- 日誌式檔案系統。
  - ext3 、 ReiserFS 、 XFS...etc.

FileSystem

常見的FileSystem ：

- ext2 ： Linux 早期使用的檔案系統，基於 inode 來管理檔案。
- ext3 ： ext2 的強化版 ，增加了日誌功能，目前是大部份 Linux 預設使用的檔案系統。
- ext3 ： ext2 的強化版 ，增加了日誌功能，目前是大部份 Linux 預設使用的檔案系統。

FileSystem

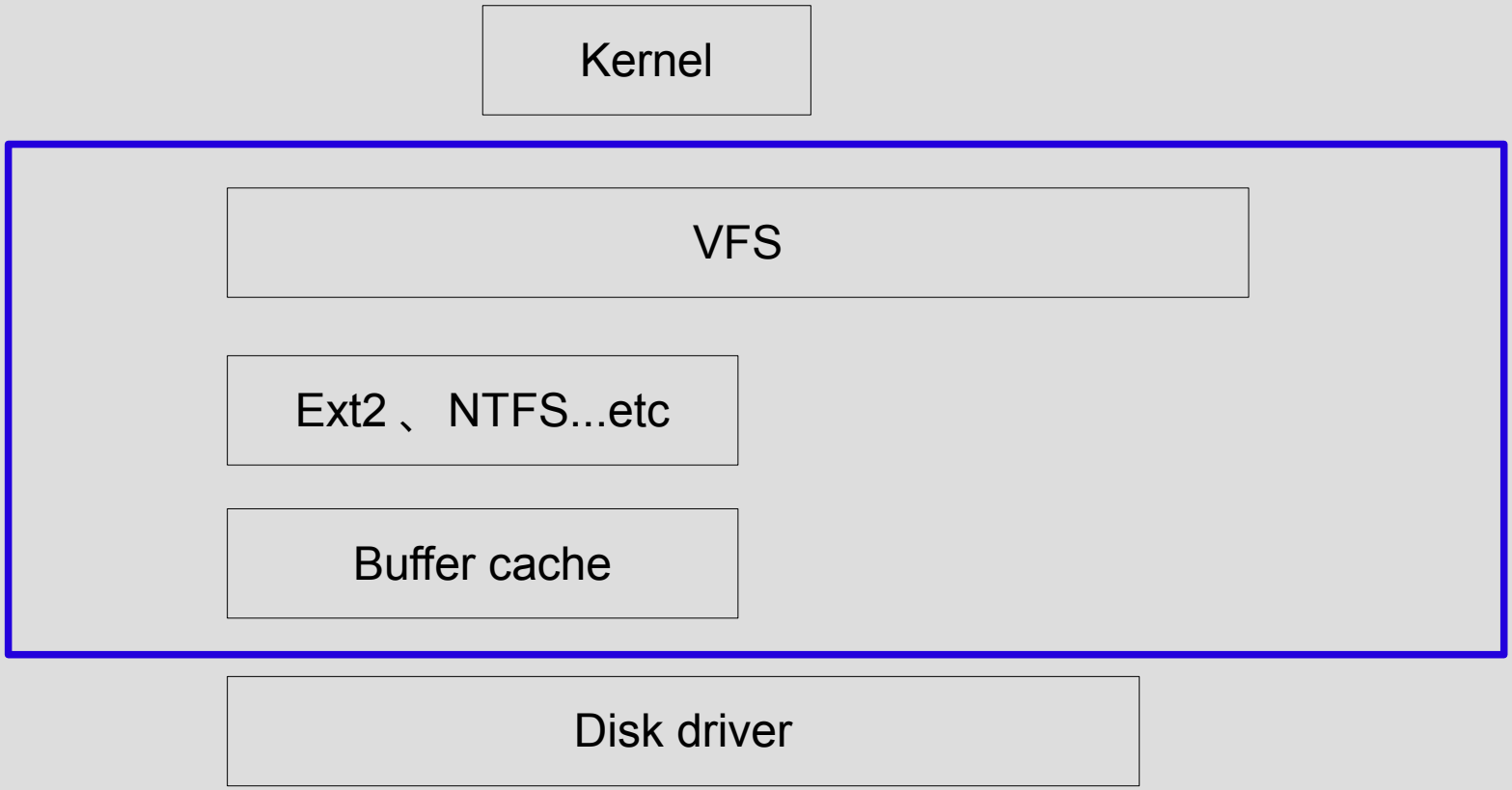
常見的FileSystem ：

- xfs ：原本是 SGI 系統所使用的檔案系統，特點是處理大檔案速度快。
- ReiserFS ： Hans Reiser 及其團隊所開發，使用了 B^+-tree ，特點是對小檔案的處理速度快。
- zfs ：由 Sun 所開發的檔案系統， 128 位元的檔案系統，幾乎可達到無限容量大小的支援。

FileSystem

VFS ：

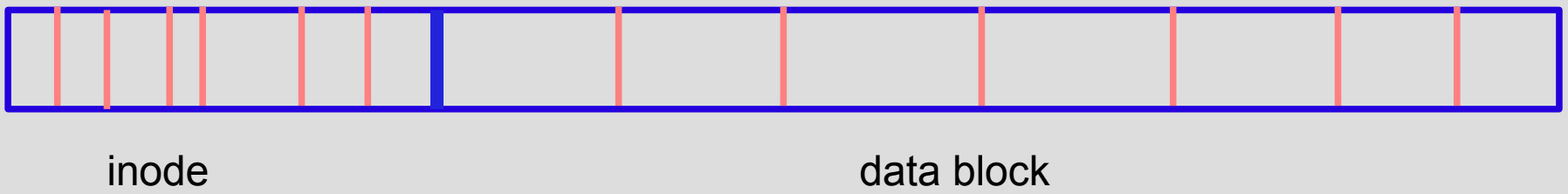
- Virtual FileSystem Switch.



FileSystem

檔案組成(ext2)：

- 檔案 (file) = 資料 (data) + 資訊 (metadata)
- 資料存放於 data block 。
- 資訊存放於 inode 。



FileSystem

inode ：

- 128 bytes
- 儲存檔案的各項屬性 (類別、權限、大小、修改時間、 data block 位置 ...etc)
- 每個檔案都有其獨立的 inode 。

FileSystem

block :

- Data Block
- inode Table
- inode Bit Map
- Block Bit Map
- Block Group Descriptor
- Super Block

FileSystem

block ：

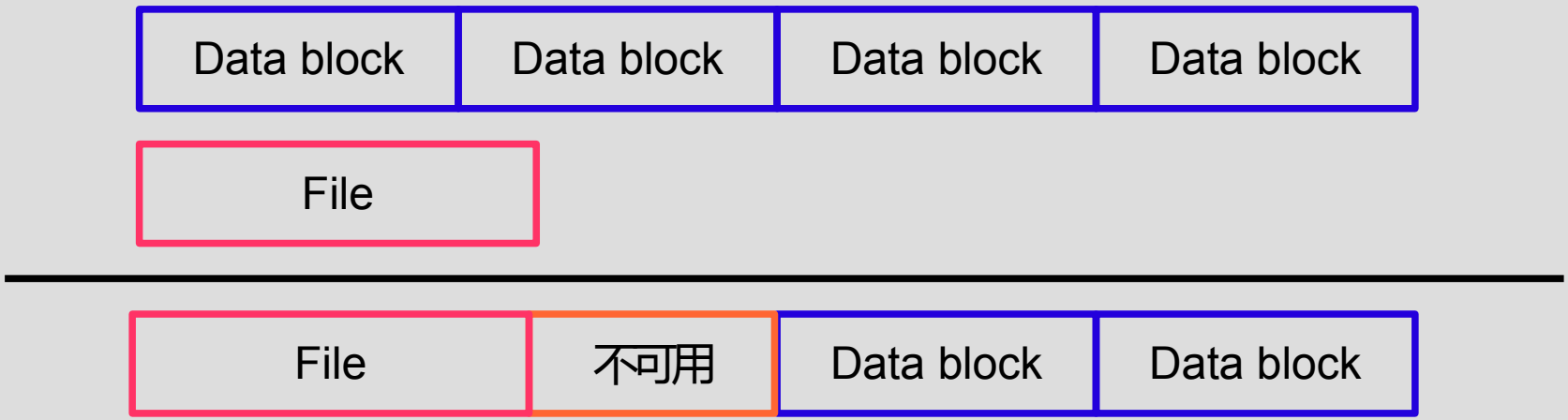
- 建立 block 時可指定不同大小 (format)
  - 建立數量不同。
  - 浪費程度不同。
  - 效能不同。

FileSystem

block ：

- block 佔用空間方式：

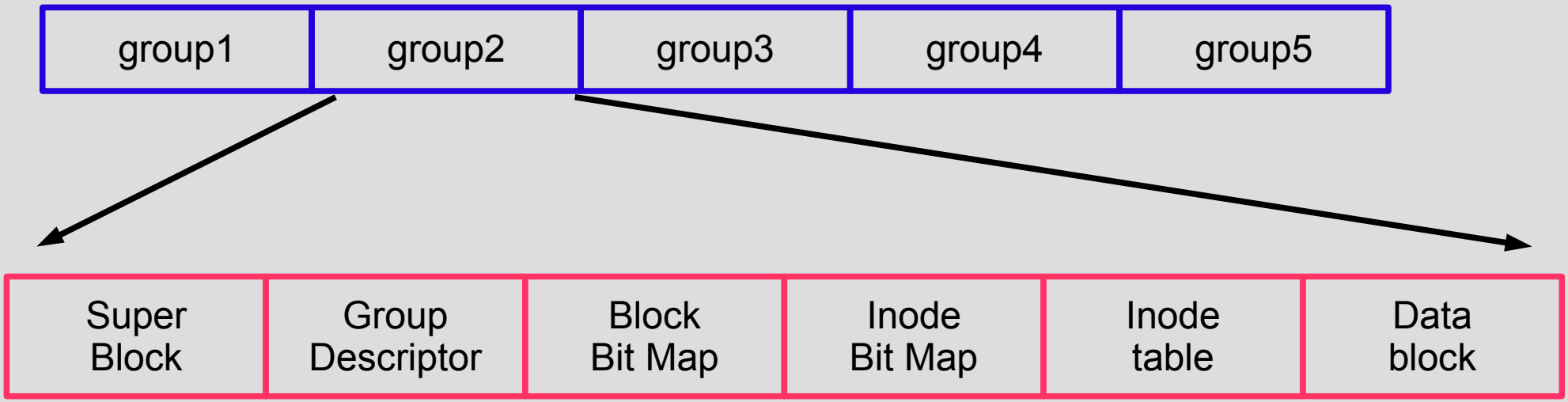
Ex ：File Size ： 5k ，Block Size ：4k



FileSystem

block ：

- Block group (index)



FileSystem

block ：

- Data block ：

用來存放檔案資料。佔用最多數量。

Q ：如何知道檔案資料放那？

FileSystem

block ：

- Inode table ：

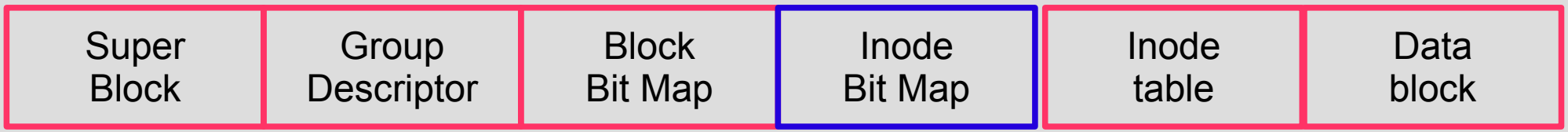
存放檔案的 inode 資訊， inode 一旦滿了也一樣會無法儲存新檔案， inode 會告知檔案所使用的 data block 位置。

Q ：如何知道那些 inode 和 block 用掉了？

FileSystem

block ：

- Inode bit map ：



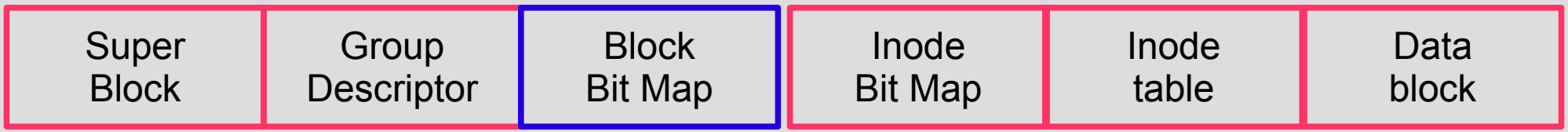
利用 0 或 1 的狀態來對照 inode map 的分配狀況。

Q ：如何知道 inode bit map 有多大？

FileSystem

block ：

- Block bit map ：



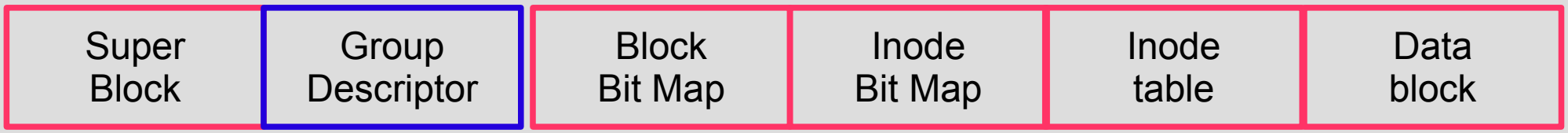
利用 0 或 1 的狀態來對照 data block 的分配狀況。

Q ：如何知道 block bit map 有多大？

FileSystem

block ：

- Group Descriptor ：



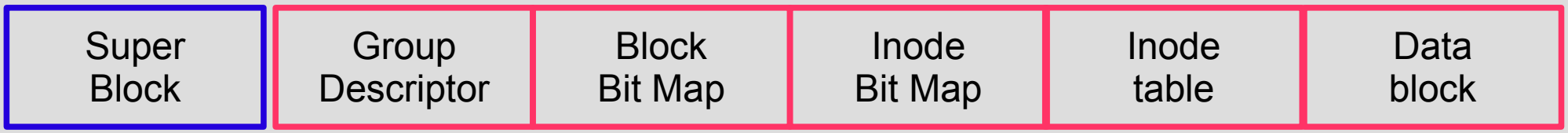
記錄及標示 bit map 、 inode table 、 data block 等區塊的範圍及指標。

Q ：如何知道 filesystem 的分配及使用情形？

FileSystem

block ：

- Super block ：



記錄整個 filesystem 分配及使用狀況。

Q ：如何從 filesystem 找到一份檔案？

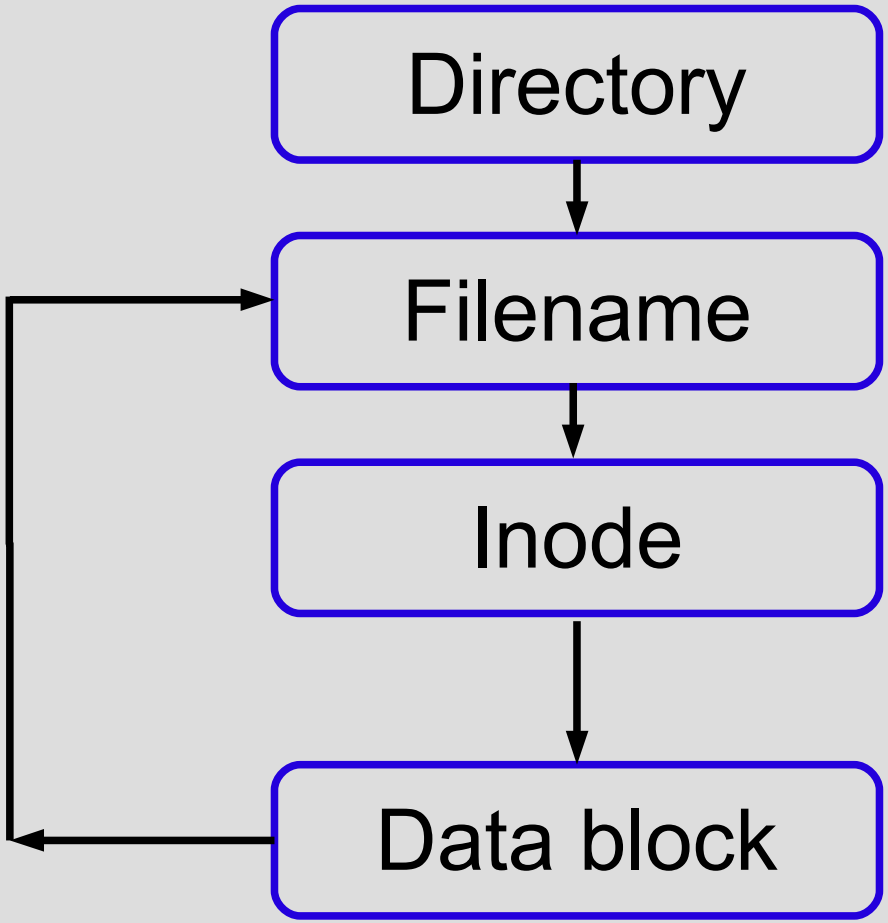
FileSystem

directory ：

- 也是一份檔案。
- 記錄其下所索引的每一份檔案名稱及 inode 。
- 絕對路徑：從根目錄 (/home/abc/def) 開始索引。
- 相對路徑：從當前目錄 (./abc/def) 開始索引。

FileSystem

directory ：



FileSystem

link ：

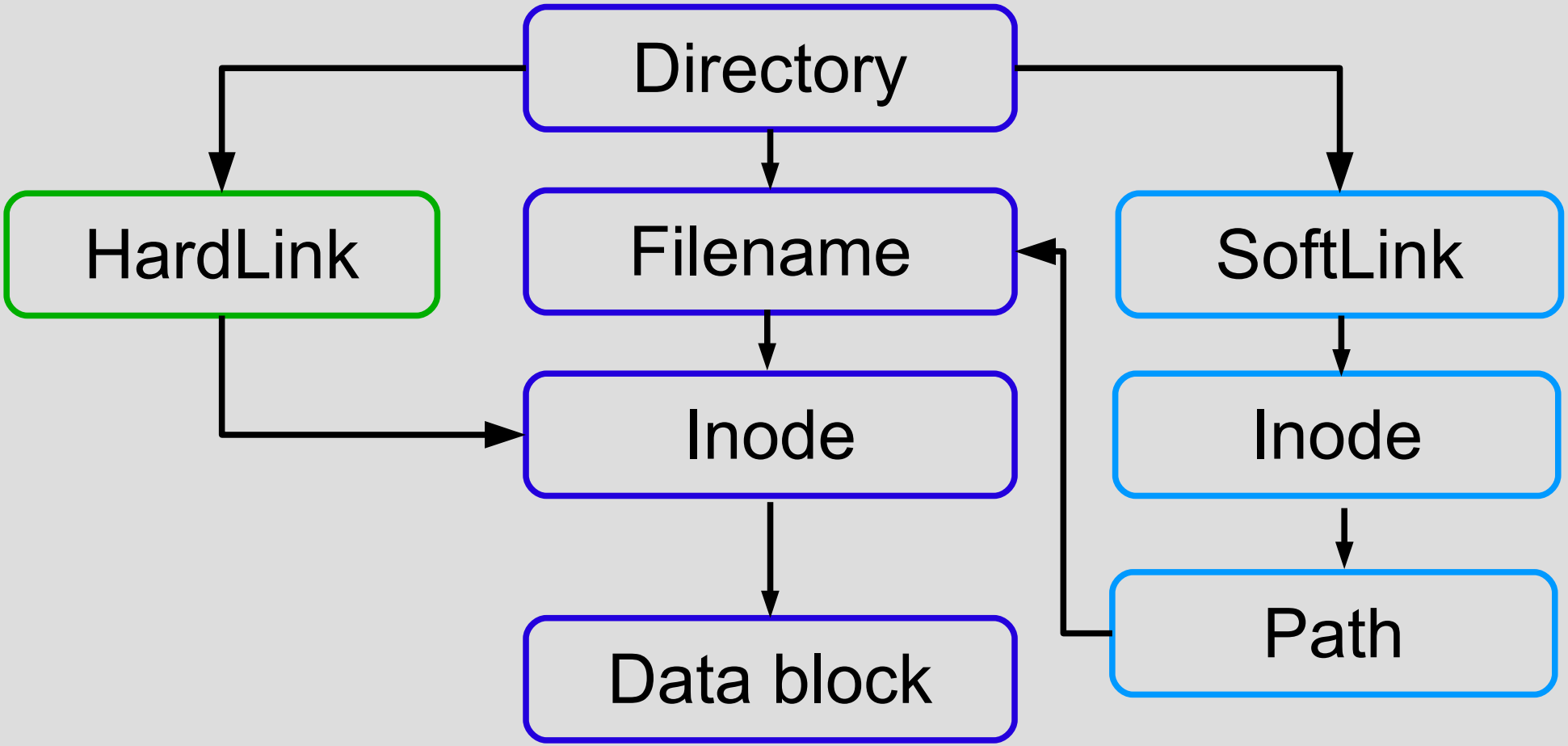
- Hard Link (不可跨 filesystem 、不可 link 目錄)
  - 使用與原檔案相同的 inode 。

    #ln [src-file] [link-file]
- Symbolic Link (Soft Link)
  - 另外新增的 inode ，但資料內容是原檔案的路徑。

    #ln -s [src-file] [linkfile]

FileSystem

link ：



FileSystem

開機磁碟檢查：

- 檢查 super block 來確認狀況是否需要修復。
- 非日誌式檔案系統會每個檔案都比對檢查。(花費大量時間)
- 日誌式檔案系統會對日誌有記錄的檔案進行檢查。(節省時間)

Q & A

休息一下

Partition

硬碟名稱：

- IDE HDD/ATAPI CDROM
  - hda
  - hdb
  - . . . . .
- SCSI HDD/ SATA HDD/ USB DISK
  - sda
  - sdb
  - . . . . .

Partition

分割區名稱：

- IDE HDD/ATAPI CDROM
  - hda1
  - hda2
  - . . . . .
- SCSI HDD/ SATA HDD/ USB DISK
  - sda1
  - sda2
  - . . . . .

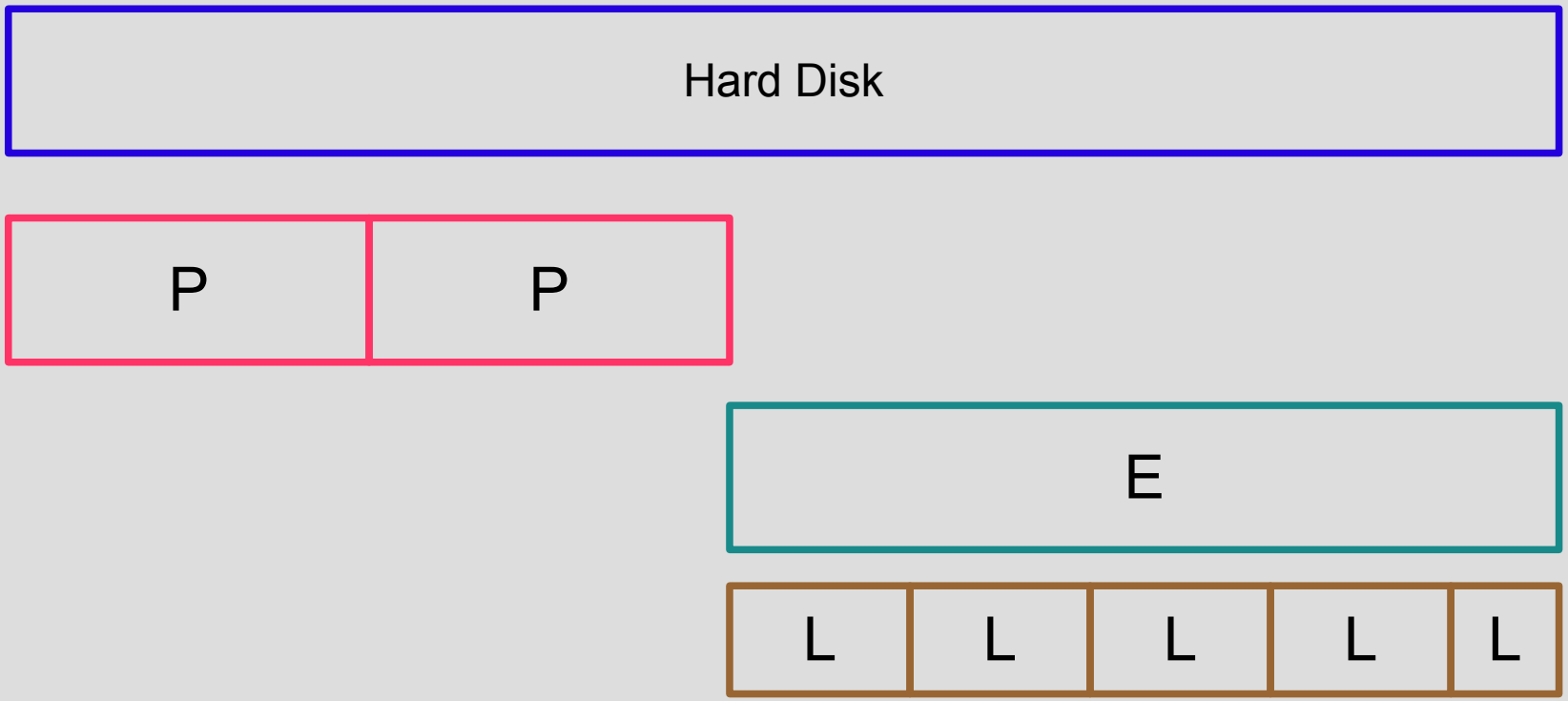
Partition

分割區類別：

- Primary Partition
  - 最多 4 個。
  - 1-4 ，不需連號，不用照順序。
- Extended Partition
  - 從 primary 轉換過來。
  - 最多一個。
- Logical Partition
  - 只能在 Extended Partition 內建立。
  - 需連號 (從 5 開始)
  - 可以不用照順序。

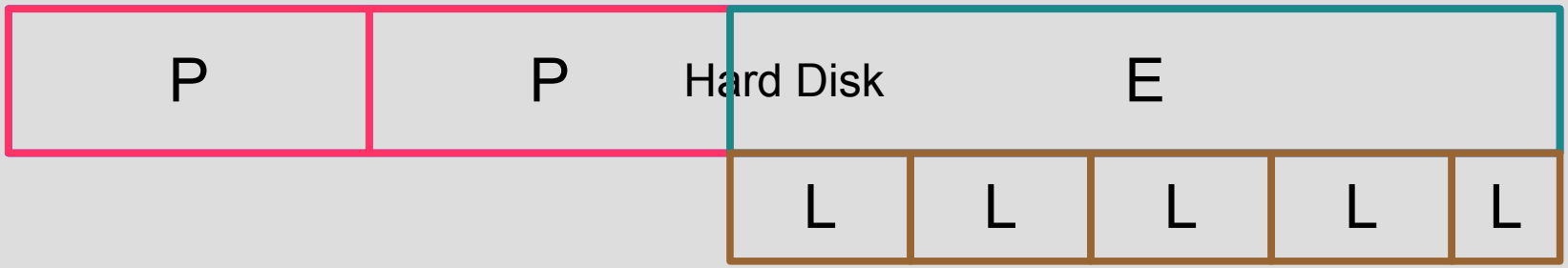
Partition

分割區類別：



Partition

分割區類別：



Partition

分割區考量因素：

- 內容分類：系統檔案、程式 / 原始碼、資料文件。
- 檔案性質： size 、存取行為。
- 安全性：掛載選項。
- 效能。
- 擴充性。

Partition

基本分割區：

- / (root)
- swap

Partition

常見分割區：

- /boot
- /home
- /usr
- /var
- /tmp

Partition

建立流程：

- fdisk
- partprobe
- mkfs
- mkdir
- mount
- /etc/fstab

Partition

fdisk ：

- 建立 / 修改 partition
- #fdisk /dev/sdb
  - m ：參數說明。
  - p ：列出 partition 。
  - n ：建立 partition 。
  - d ：刪除 partition 。
  - l ：列出 partition type 。
  - t ：修改 partition type 。
  - q ：不修改退出。
  - w ：儲存後退出。

Partition

partprobe ：

- 強制更新 partition 資訊。 (reboot or partprobe)
- 當 fdisk 完後， kernel 沒有 reload partition info 時使用
- #partprobe

Partition

mkfs ：

- 格式化。
- mkfs 、 mkfs.ext2 、 mkfs.ext3 都可以。
- #mkfs.ext3 /dev/sdb1
- #mkfs.ext2 -j /dev/sdb1
- #mkfs -t ext3 /dev/sdb1

Partition

mkfs ：

- -b ： block size.
- -c ： check block for bad block.
- -L ： volum label.

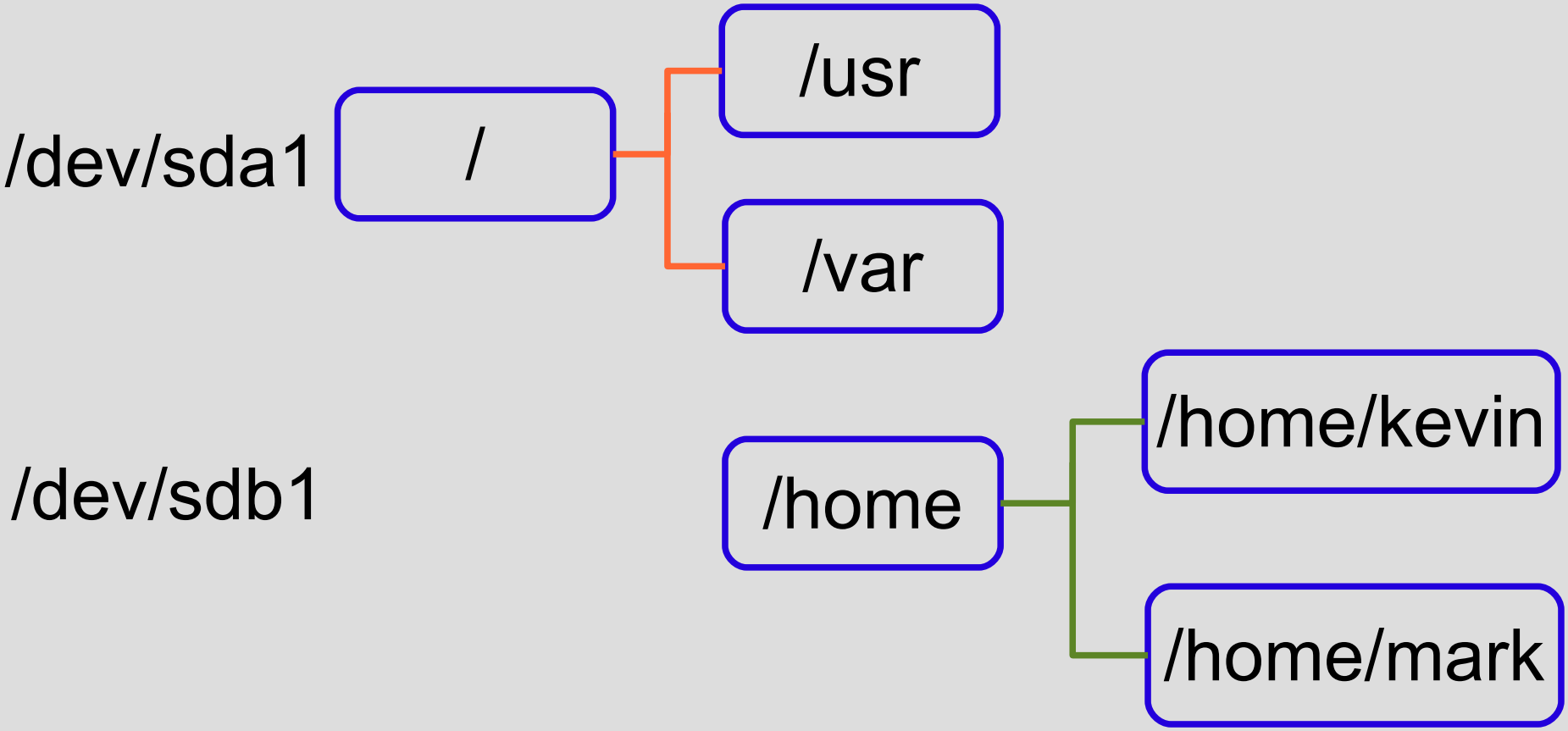
Partition

mount point ：

- linux 沒有磁碟代號 (c: 、 d:) 。
- 所有 device 都需要 mount 才可以使用。
- mount point 一定是目錄。
- 不使用時可以 umount 。

Partition

mount point ：



Partition

mount ：

- 掛載 device 至 mount point 。
- #mount /dev/cdrom /mnt/cdrom
- -t ： filesystem type
- -r ： ready only
- -w ： read / write
- -o ： async 、 atime 、 owner 、 rw 、 roex.....etc
- mount point 的目錄需要先被建立 (mkdir) 。

Partition

umount ：

- 卸載 mount point 。
- #umount /mnt/cdrom
- mount point 不可以在使用中。

Partition

fstab ：

- /etc/fstab (掛載設定檔) 。
- Device (device path 、 label)
- mount point
- Filesystem type
- mount option
- dump
- fsck

Partition

fstab ：

- mount option ：
  - auto / noauto ： mount -a 時是否掛載。
  - user / nouser ：是否允許非 root 掛載。
  - defaults ： rw 、 suid 、 dev 、 exec 、 auto 、 nouser async(no acl support) 。

Q & A

休息一下

LVM

LVM ：

- Logical Volume Manager (LVM2) 。
- 彈性管理硬碟大小。
- mapping mode : linear 、 stripe 、 mirror 。
- snapshot 。

LVM

LVM ：

- PV (physical volume) 。
- VG (volume group) 。
- PE (physical extent) 。
- LV (logical volume) 。

LVM

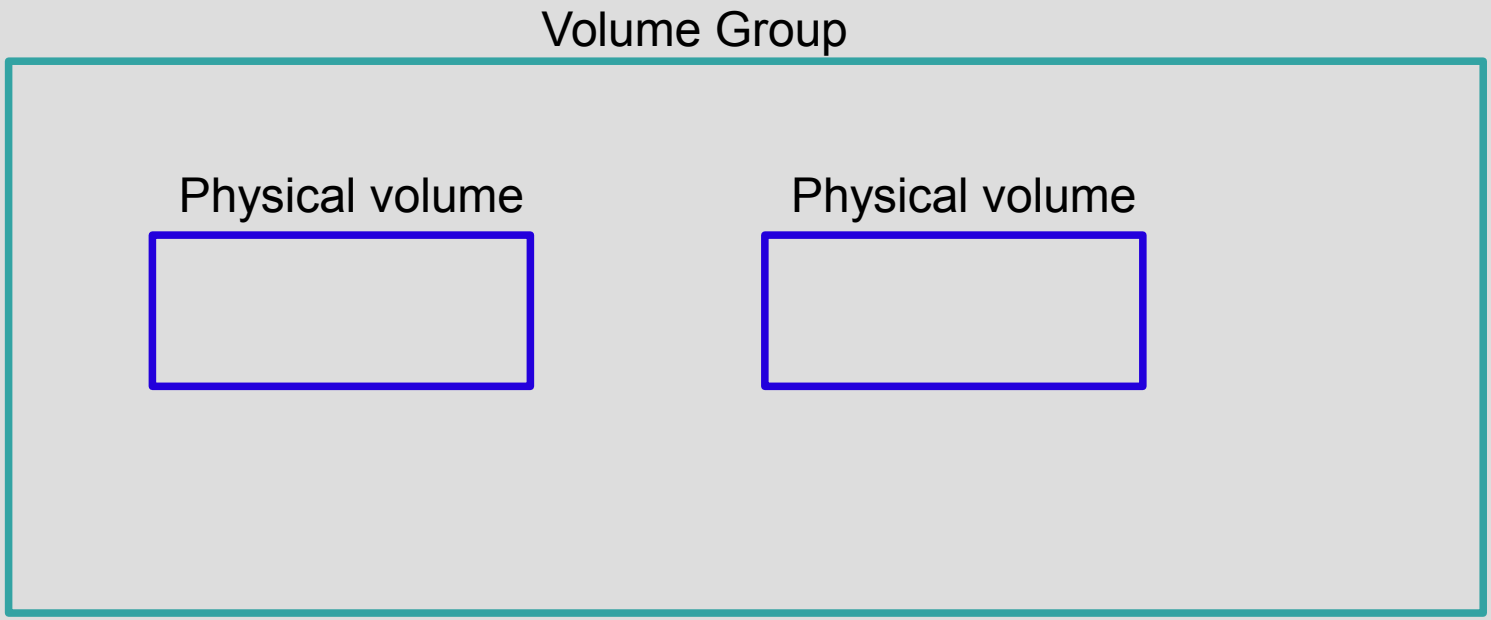
PV ：

- Partition (8e) or 整顆硬碟 。
- ex : /dev/sda1 、 /dev/sdb2 。

LVM

VG ：

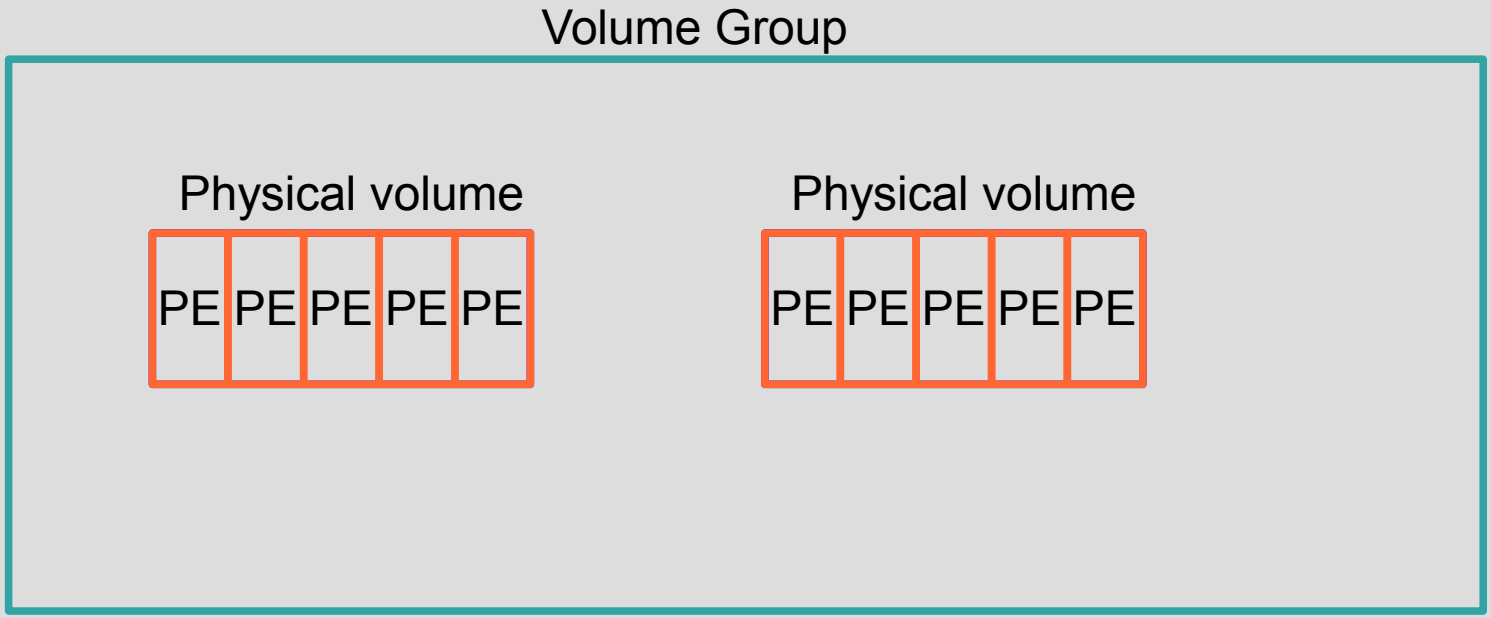
- 將一個或多個 PV 組成群組。



LVM

PE ：

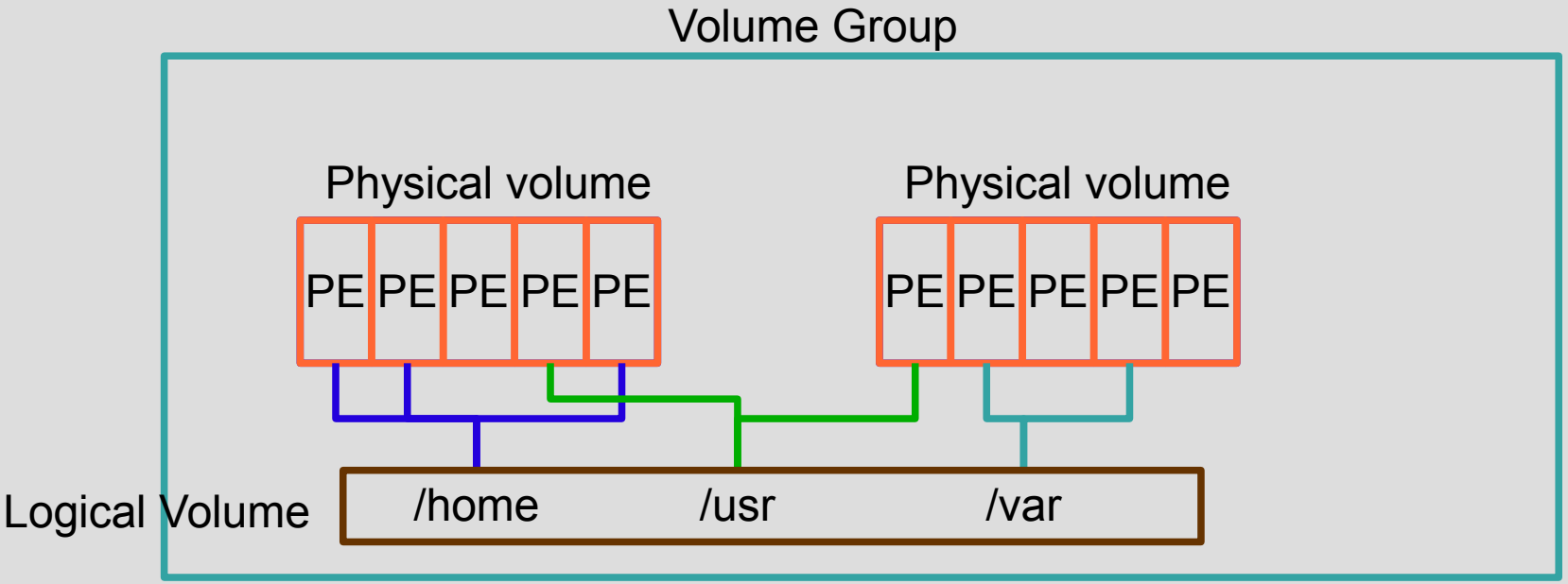
- 重新劃分 VG 空間的單位。(類似 block 的概念)
- Default : 4M



LVM

LV ：

- 由 VG 切出的 partition (PE 所組成的空間)。
- 可掛載。



LVM

lvreduce ：

- 釋放 LV 中未使用的 PE 。
- 減少 LV 的空間。

LVM

lvextend ：

- 將未使用的 PE 加入 LV 中。
- 擴大 LV 的空間。

LVM

vgextend ：

- 將 PV 加入到 VG 中 (PE 增加)。
- 擴大 VG 的空間。

LVM

vgreduce ：

- 將 PV 從 VG 中移除 (PE 減少)。
- 減少 VG 的空間。

LVM

LVM 基本建置：

- fdisk /dev/sdb (8e)
- pvcreate /dev/sdb1　　(pvs 、 pvscan 、 pvdisplay)
- vgcreate VG1 /dev/sdb1　　(vgs 、 vgscan 、 vgdisplay)
- lvcreate -L 300M -n LV1 VG1 (lvs 、 lvscan 、 lvdisplay)
- mkfs.ext3 /dev/VG1/LV1
- mkdir /mnt/myLV1
- mount /dev/VG1/LV1 /mnt/myLV1

LVM

LVM 基本建置：

- fdisk /dev/sdb (8e)
- pvcreate /dev/sdb1　　(pvs 、 pvscan 、 pvdisplay)
- vgcreate VG1 /dev/sdb1　　(vgs 、 vgscan 、 vgdisplay)
- lvcreate -L 300M -n LV1 VG1 (lvs 、 lvscan 、 lvdisplay)
- mkfs.ext3 /dev/VG1/LV1
- mkdir /mnt/myLV1
- mount /dev/VG1/LV1 /mnt/myLV1

LVM

LVM 擴充空間：

- fdisk /dev/sdc (8e)
- pvcreate /dev/sdc1
- vgextend VG1 /dev/sdc1
- lvextend -L 600M /dev/VG1/LV1
- e2fsck -f /dev/VG1/LV1
- resize2fs /dev/VG1/LV1

LVM

建立snapshot ：

- 建立一個時間點的快照。
- lvcreate -L 100M -s -n S1 /dev/VG1/LV1
- mkdir /mnt/LV1-snapshot
- mount /dev/VG1/S1 /mnt/LV1-snapshot
- umount /mnt/LV1-snapshot
- lvremove /dev/VG1/LV1

Q & A

休息一下

RAID

RAID ：

- 磁碟陣列。
- 將多顆硬碟組合成一顆。
- Hardware RAID 。
- Software RAID 。
- Host RAID 。

RAID

Hardware RAID ：

- 有獨立處理的硬體。 (raid card)
- 所有 RAID 運算由硬體完成。
- 對系統來說，直接視為一個 device (/dev/sda)
- 效能高，成本高。

RAID

Software RAID ：

- 在組成 RAID 前，各 device 為獨立的。 (sda 、 sdb)
- 組成後 RAID 後，系統以 md0 、 md1 辨視 (mdadm)
- 吃系統本身資源來進行 RAID 運算。
- 效能差、成本低。

RAID

Host RAID ：

- 介於 Hardware RAID 與 Software RAID 之間。
- 可以想成是主機內建的 RAID 功能。
- 功能較陽春。
- 效能比 Software RAID 來的稍好。

RAID

RAID0 ：

- striping 。
- 至少 2 顆硬碟。
- 存取速度最快。
- 沒有容錯能力。
- 總容量 = 所有硬碟總和。

RAID

RAID1 ：

- mirror 。
- 至少 2 顆硬碟。
- 效能與原本差不多。
- 具有容錯能力。
- 總容量 = 所有硬碟容量的一半 。

RAID

RAID5 ：

- 至少 3 顆硬碟。
- 可容錯 1 顆硬碟壞掉。
- 效能較 RAID1 高。
- 總容量 = (硬碟數量 -1) * 硬碟容量。

RAID

Hot spare ：

- 需至少 1 顆閒置硬碟。
- 當有硬碟壞掉時，可最快時間替代並開始 rebuild 。
- RAID rebuild 過程中，效能會大幅下降。

RAID

RAID1+0 :

- RAID10
  - RAID1 + RAID0
- RAID01
  - RAID0 + RAID1

RAID

mdadm ：

- multi device administers. (Linux)
- Software RAID or multipath.
  - RAID0 、 RAID1 、 RAID5 、 RAID6.....etc.
- /dev/md0 、 /dev/md1 、 /dev/md2 .....

RAID

mdadm ：

- --create　　--create /dev/md0
- --raid-device　　--raid-device=2
- --level　　--level=1 or --level=mirror
- --spare-device　　--spare-device=1 (raid0 不適用)
- --detail　　--deltail /dev/md0 (--scan)
- /etc/mdadm/mdadm.conf

RAID

mdadm ：

- --manage　　--stop /dev/md0
  --add /dev/sdd
  --fail /dev/sdb
  --remove /dev/sdb
- --assmeble　　--run /dev/md0
- /etc/mdadm/mdadm.conf

RAID

基本操作：

- 建立 /dev/md0 (raid1)

```
#mdadm  --create /dev/md0 --raid-device=2 \
> --level=1 --spare-devices=0 \
> /dev/sdb /dev/sdc
```

- mkfs.ext3 /dev/md0
- mkdir /mnt/raid-test
- mount /dev/md0 /mnt/raild-test
- mdadm --detail --scan >> /etc/mdadm/mdadm.conf

RAID

基本操作：

- 加入 device

#mdadm --manage /dev/md0 --add /dev/sdd

- 設定 failure device

#mdadm --manage /dev/md0 --fail /dev/sdc

- 移除 device

#mdadm --manage /dev/md0 --remove /dev/sdc

Q & A