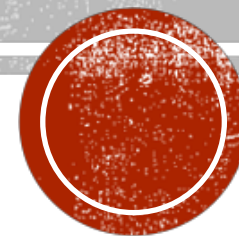
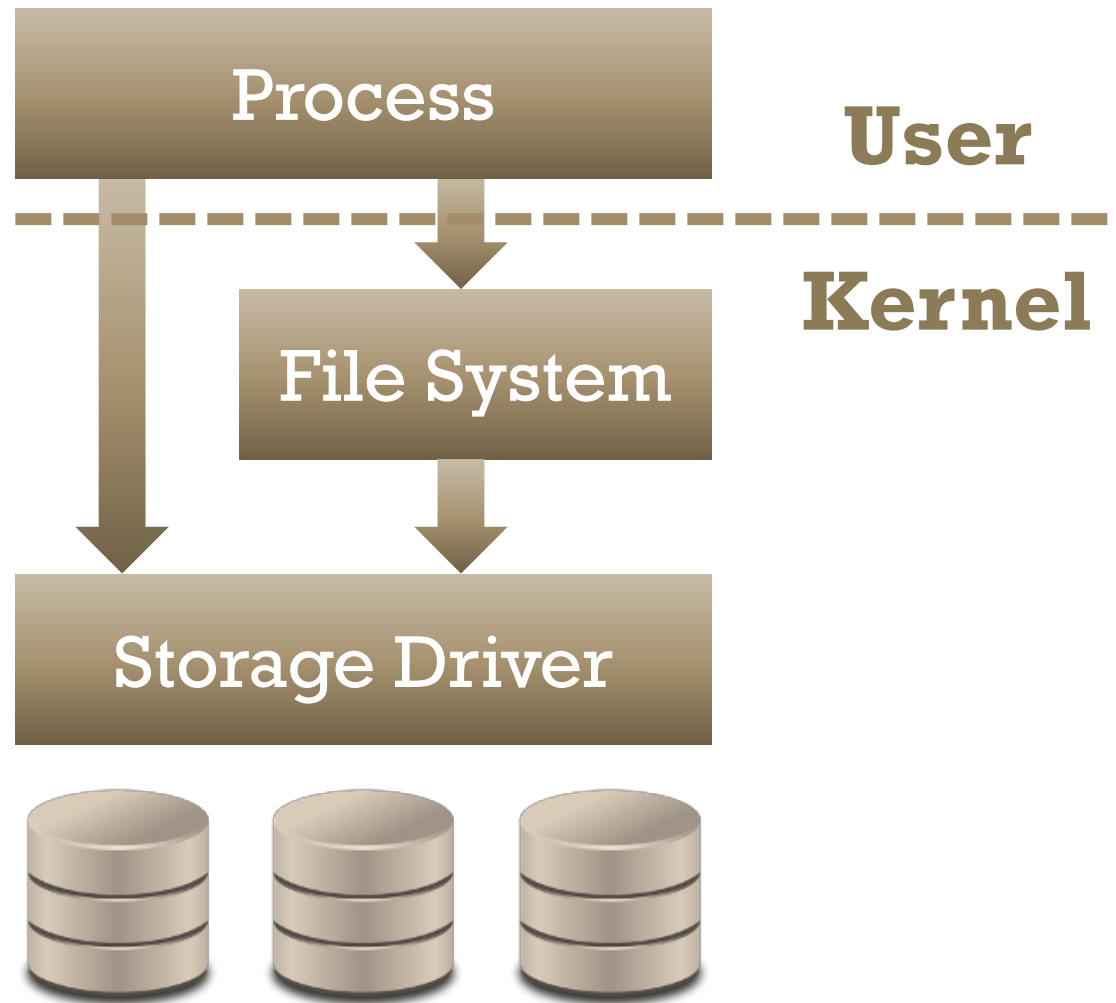


LINUX系統實務

儲存管理

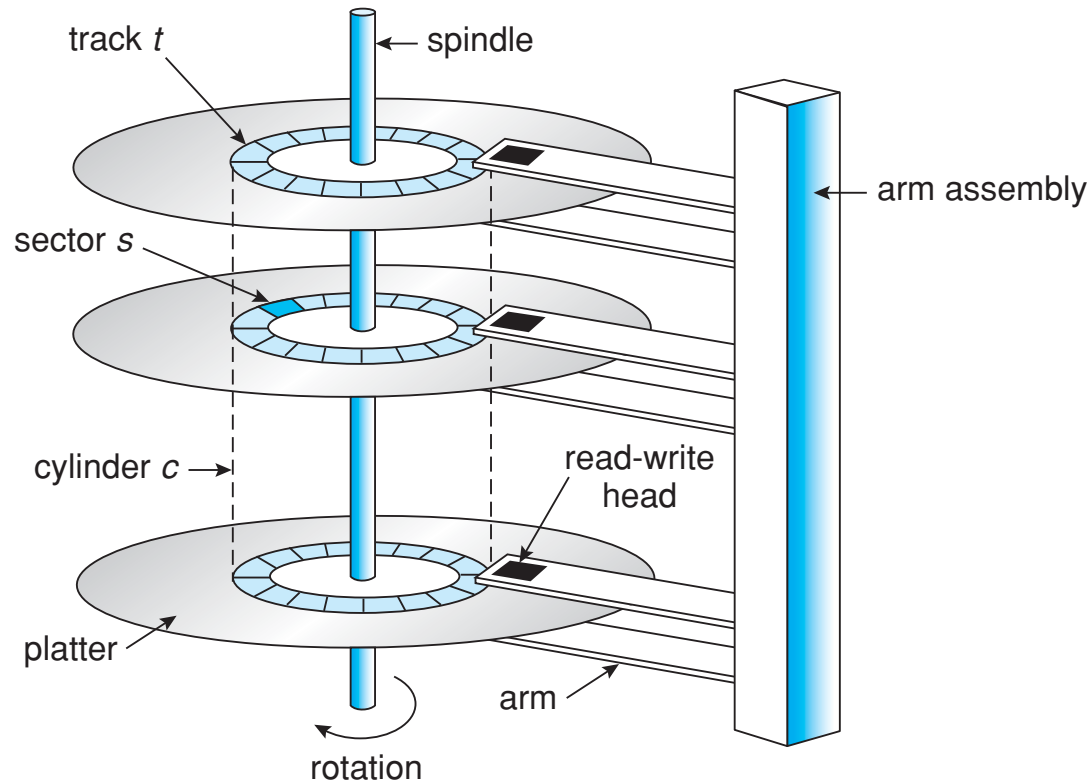


檔案系統 與 儲存裝置



- 一般而言，存取儲存裝置上的資料是透過檔案系統
- 如果沒有檔案系統，會怎樣？

DISK



- Linux中的設備，通常會以檔案的形式存放於/dev
- 硬碟設備檔的命名規則為sda, sdb...依此類推
- 當硬碟被分割後，Partition的檔名為所屬硬碟檔名後加上一數字
- 例如：
 - /dev/sda1
 - /dev/sdc3

PARTITION

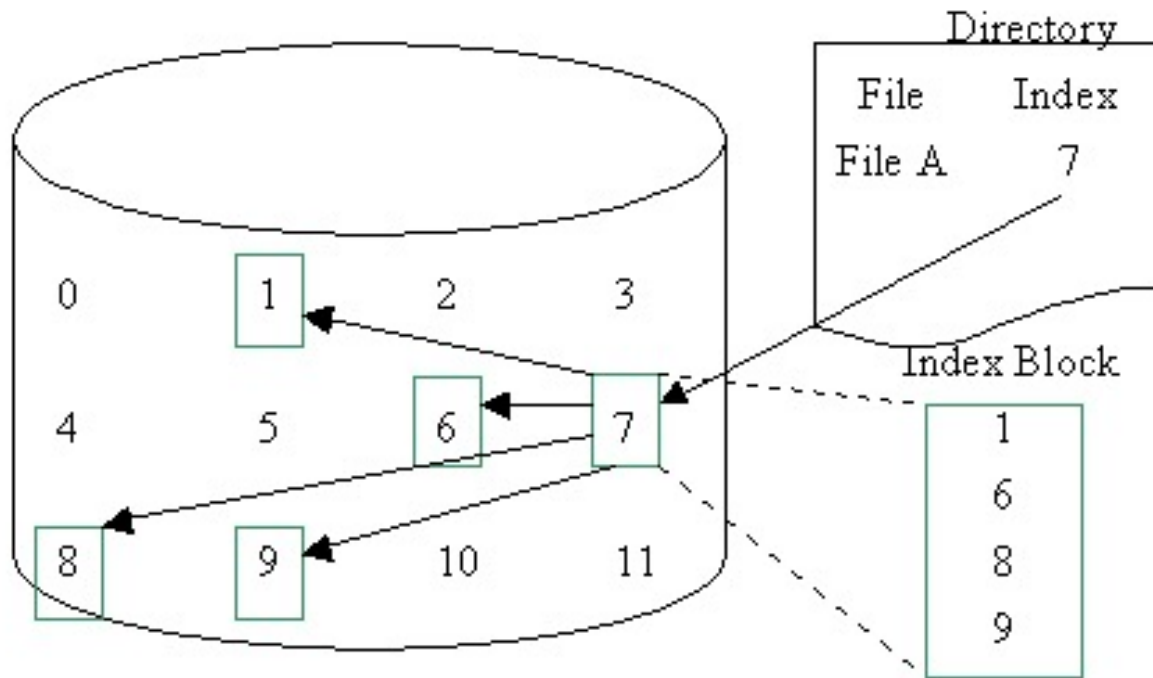
- sda, sdb, sdc 分別為虛擬機中的三顆硬碟
- sdb1 以及 sdb2 代表 sdb 被分割成兩個 partitions
- Partition 的大小各不相同，且也掛在在於不同地方

```
sda          8:0      0      10G   0 disk
├─sda1       8:1      0      10G   0 part
│   ├─ubuntu--vg-root 253:0    0       9G   0 lvm  /
│   └─ubuntu--vg-swap_1 253:1    0    976M  0 lvm  [SWAP]
sdb          8:16     0       1G   0 disk
├─sdb1       8:17     0    100M  0 part /home/wang/project1
└─sdb2       8:18     0    200M  0 part /mnt/sdb2
sdc          8:32     0       2G   0 disk
```

FILE SYSTEM

- 檔案一經生成之後，必須在磁碟內找到可用的配置空間，磁碟內配置的單位是區塊（**Block**）
 - 一個區塊可以對應一個或多個磁區（**sectors**）
- 建立檔案配置表（**File Allocation Table**），記錄那些磁碟空間是已被使用或未被使用
- 一個作業系統可支援多種檔案系統格式，且一般使用情況下，一個**Partition**使用一種檔案系統格式
 - **FAT32, NTFS, EXT4**
 - 一個**Partition**也可能格式化為多種檔案系統格式
 - **LVM, RAID**

索引配置法(INDEXED ALLOCATION)



- 當檔案生成時，檔案系統除了配置可用區塊存放資料之外，額外配置一個**索引區塊**，此索引區塊內均用來擺放指標
- UNIX系統中，任何一個檔案，均有一個i-node

```
root@vhw-VirtualBox:/home/yhw/test# ls -li a.cpp  
139030 -rw-r--r-- 2 root root 12 Aug  1 23:21 a.cpp
```

- **superblock**：記錄此 **file system** 的整體資訊，包括inode與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；
- **data block**：實際記錄檔案的內容，若檔案太大時，會佔用多個 block

格式化

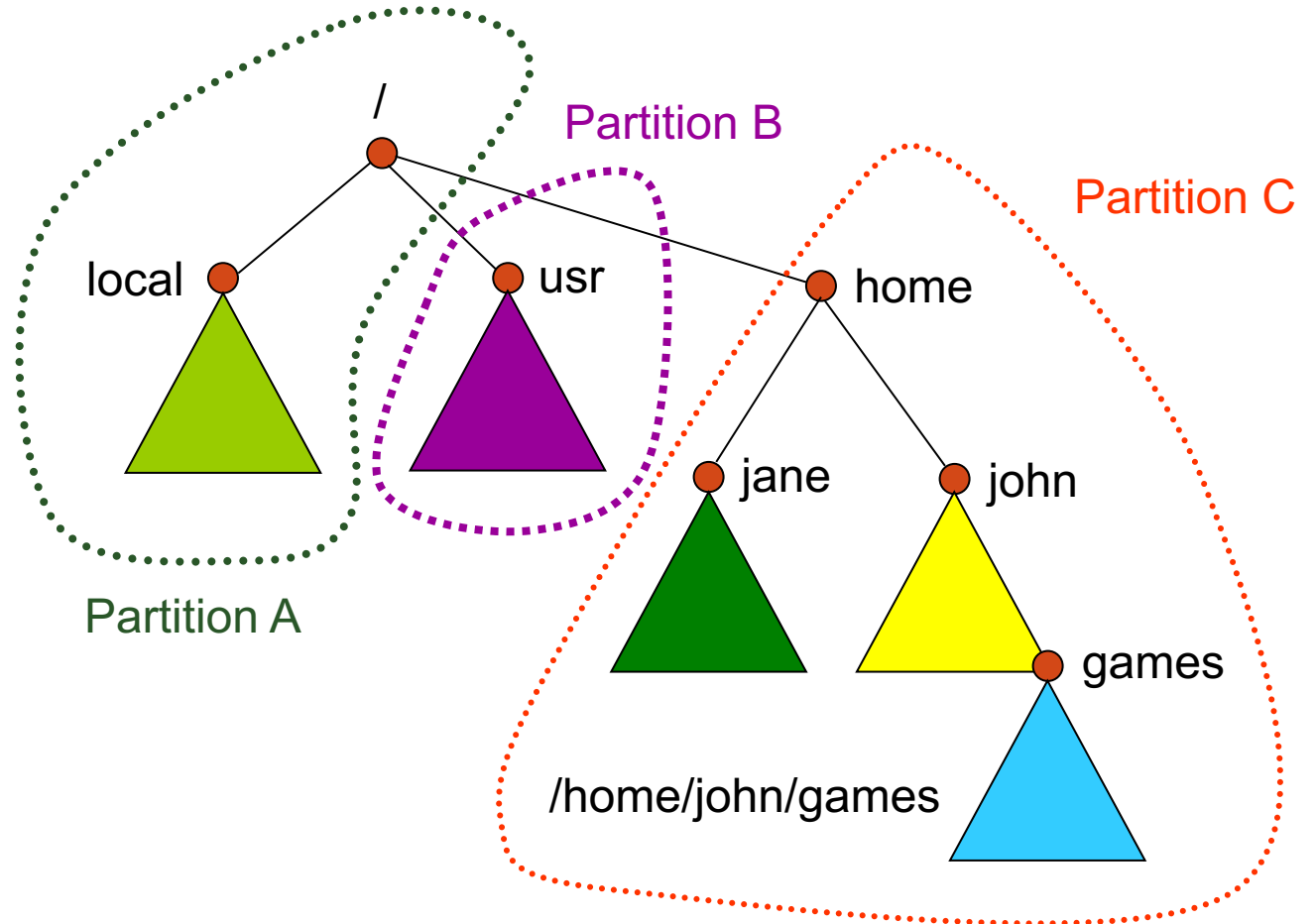
- 格式化可以理解為建立檔案系統，在Linux中常使用指令為：
 - `mkfs`
 - `-t` : 可指定想要使用的 `file system` 格式
- 觀察每個partition各自使用到的檔案系統的指令：
 - `parted -l`

```
Disk /dev/sdb: 1074MB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start   End     Size    Type    File system  Flags
  1      1049kB  106MB   105MB   primary ext4
  2      106MB   316MB   210MB   primary ext2
```

MOUNT POINTS

- A file system must **be mounted** before it can be available to processes on the system.
- **Mount point:** the root path that a file system will be mounted to
- **Mount:** 將檔案系統(格式化後的 **partition**)與目錄樹結合的動作



掛載點

```
sda      8:0    0    10G  0 disk
├─sda1    8:1    0    10G  0 part
│   └─ubuntu--vg-root 253:0    0     9G  0 lvm  /
│   └─ubuntu--vg-swap_1 253:1    0    976M  0 lvm  [SWAP]
└─sdb     8:16   0     1G  0 disk
    ├─sdb1    8:17   0    100M  0 part /home/wang/project1
    └─sdb2    8:18   0    200M  0 part /mnt/sdb2
```

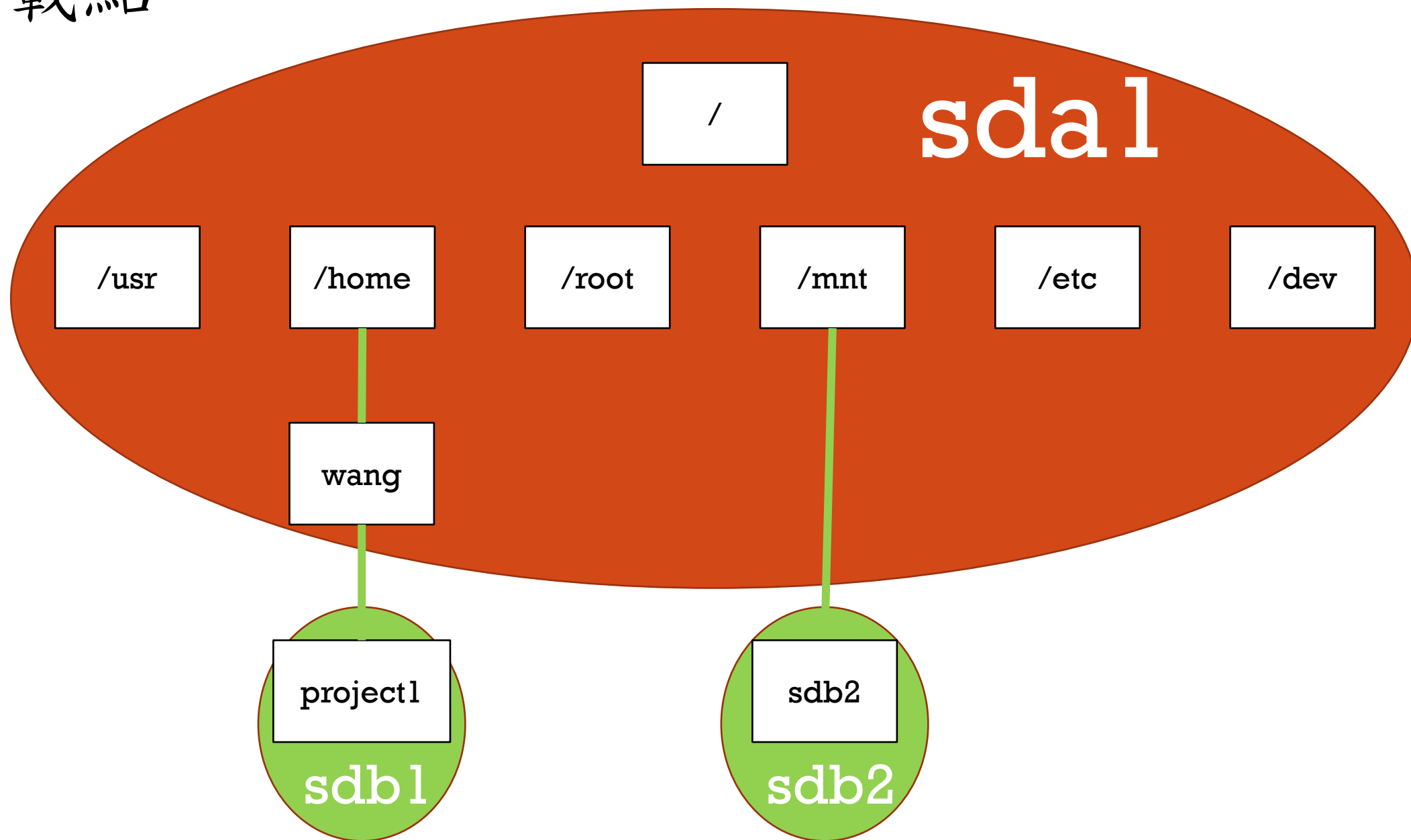
- 掛載點(**mount point**)一定是目錄，該目錄為進入該檔案系統的入口
- 透過此目錄可存取**partition**中的資料
- 指令
 - **lsblk** 可查看**block**設備的資訊

可用空間查詢

- 可以使用 **df -h** 查看硬碟空間大小、使用情形，甚至是掛載點
- 當空間不夠時，除了增加額外空間以外，我們也可以進入 **mount point** 刪除不必要的大型檔案

```
root@wang-VirtualBox:/home/wang# df -h
Filesystem                Size      Used Avail Use% Mounted on
udev                      1.9G         0   1.9G   0% /dev
tmpfs                     394M      1.7M   393M   1% /run
/dev/mapper/ubuntu--vg-root 8.9G     5.4G   3.0G  65% /
```

掛載點



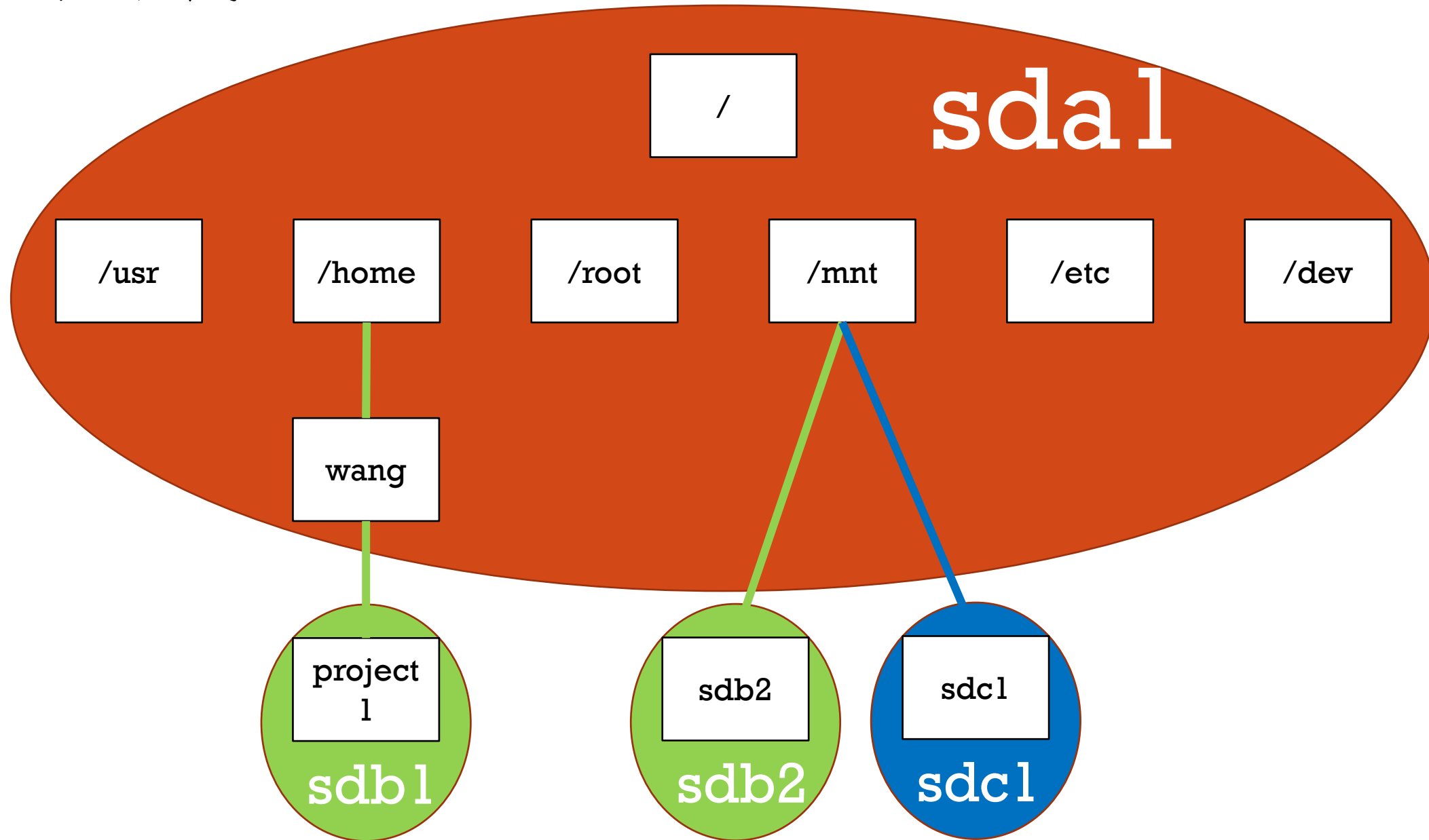
新增掛載

```
root@wang-VirtualBox:/home/wang# mkdir /mnt/sdc1
root@wang-VirtualBox:/home/wang# mount /dev/sdc1 /mnt/sdc1
root@wang-VirtualBox:/home/wang# lsblk
```

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINT
sda	8:0	0	10G	0	disk	
└─sda1	8:1	0	10G	0	part	
└─ubuntu--vg-root	253:0	0	9G	0	lvm	/
└─ubuntu--vg-swap_1	253:1	0	976M	0	lvm	[SWAP]
sdb	8:16	0	1G	0	disk	
└─sdb1	8:17	0	100M	0	part	/home/wang/project1
└─sdb2	8:18	0	200M	0	part	/mnt/sdb2
sdc	8:32	0	2G	0	disk	
└─sdc1	8:33	0	200M	0	part	/mnt/sdc1

- 建立準備掛載的目錄
 - **mkdir** 目錄路徑
- 執行掛載指令
 - **mount** 設備名稱 掛載路徑

新增掛載點

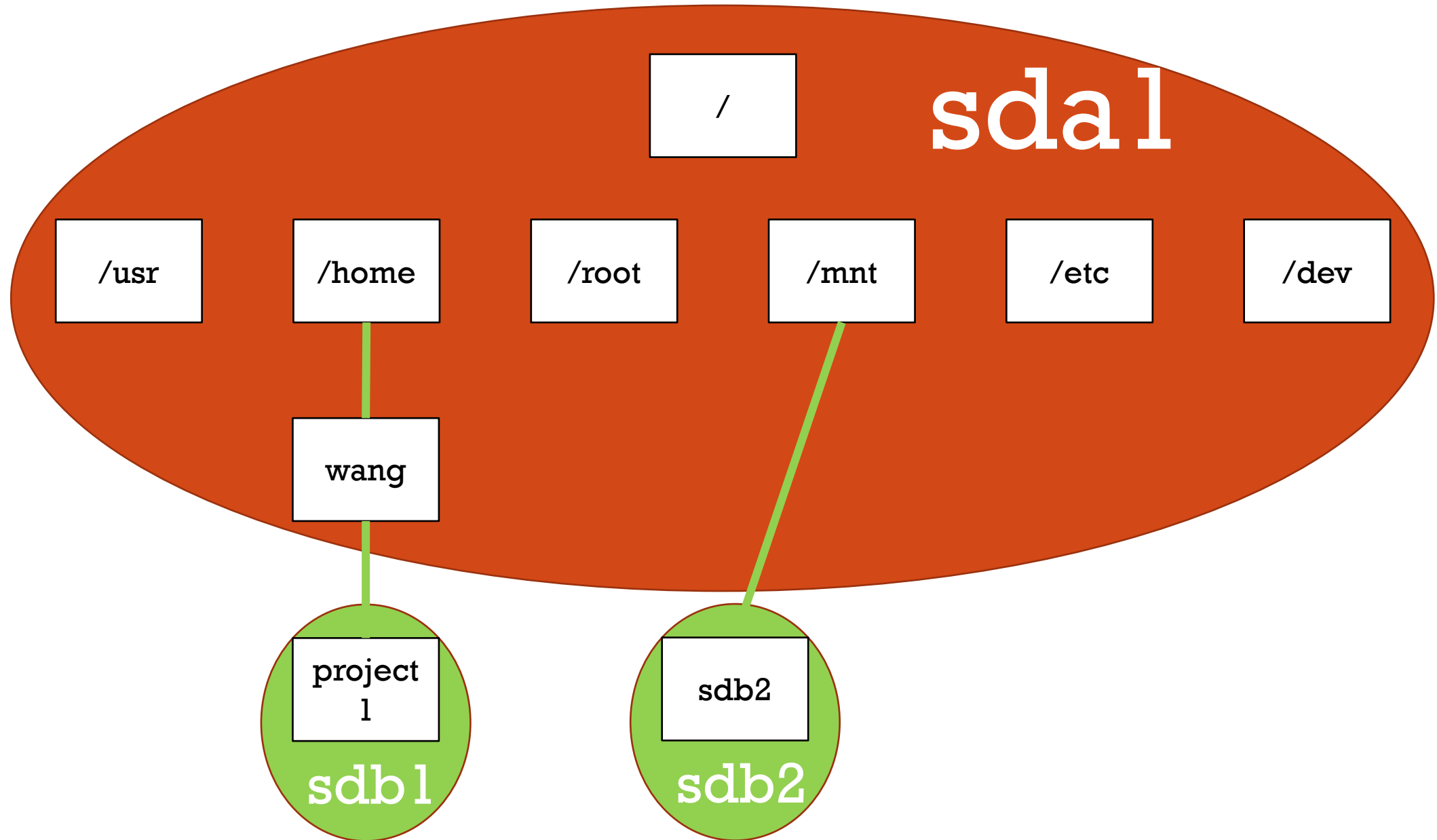


UMOUNT

- 當執行umount時，會將partition卸載，卸載以後若想重新存取partition，需要重新掛載

```
root@wang-VirtualBox:/home/wang# mount /dev/sdc1 /mnt/sdc1
root@wang-VirtualBox:/home/wang# umount /mnt/sdc1
root@wang-VirtualBox:/home/wang# lsblk
sda                8:0      0      10G   0 disk
└─sda1              8:1      0      10G   0 part
   └─ubuntu--vg-root 253:0     0       9G   0 lvm   /
      └─ubuntu--vg-swap_1 253:1     0      976M  0 lvm   [SWAP]
sdb                8:16     0       1G   0 disk
└─sdb1              8:17     0      100M  0 part /home/wang/project1
   └─sdb2            8:18     0      200M  0 part /mnt/sdb2
sdc                8:32     0       2G   0 disk
└─sdc1              8:33     0      200M  0 part
```

Umount



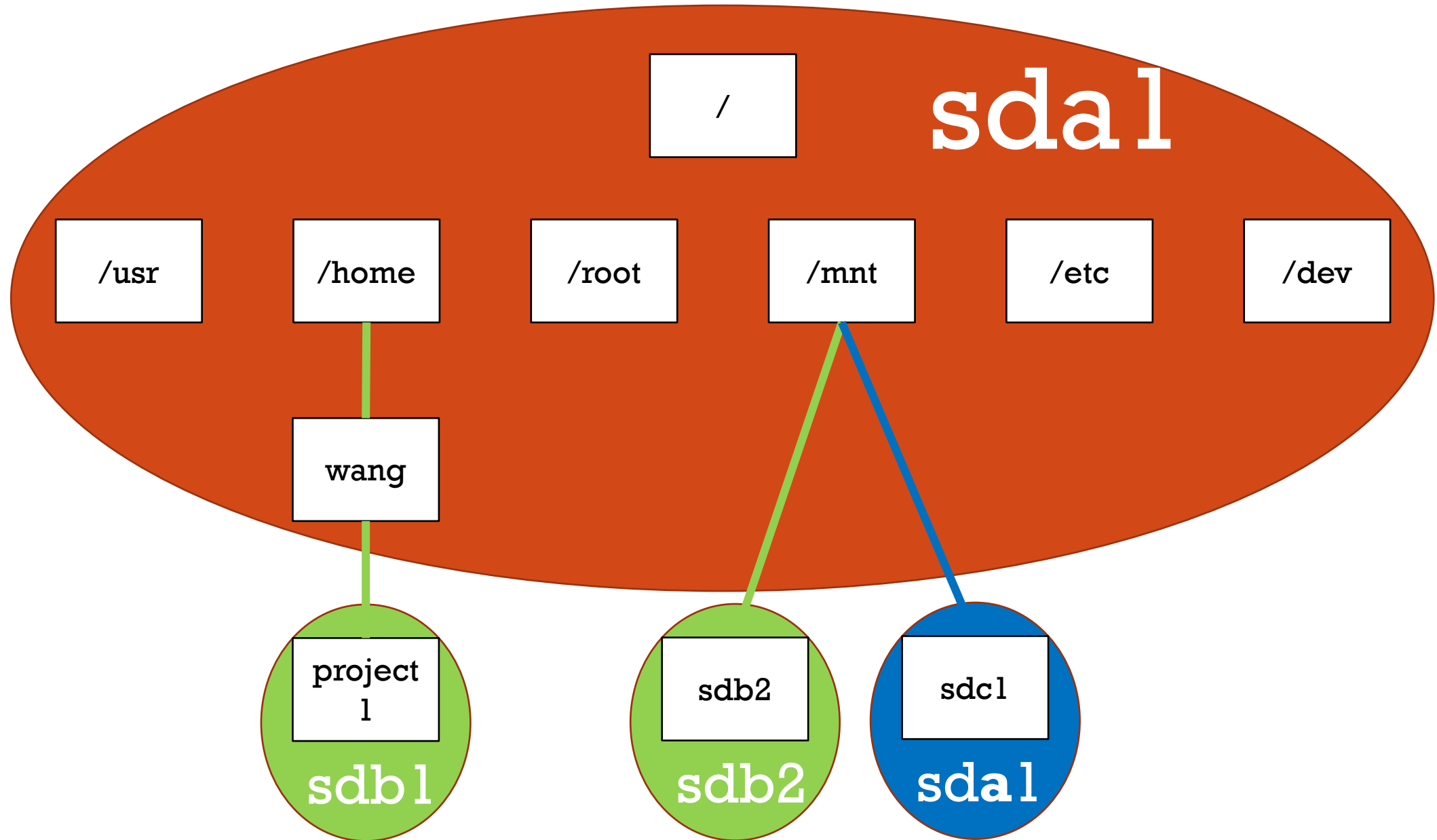
注意

- 無論是否掛載/**dev/sdc1**，該資料夾都會存在，但掛載與不掛載所使用到的空間不同
 - 沒掛載：
 - 從/**mnt/sdc1**開始向上找掛載點，最近的掛載點為 /
 - 此時/**mnt/sdc1**連同下方的資料都會儲存在/**dev/sda1**中
 - 有掛載：
 - 因為/**mnt/sdc1**本身就是掛載點，/**dev/sdc1**掛載於此
 - 此時/**mnt/sdc1**連同下方的資料都還會存在/**dev/sdc1**中
- 可是，同樣都是/**mnt/sdc1**，但**mount**內容不同，資料將被儲存在不同地方

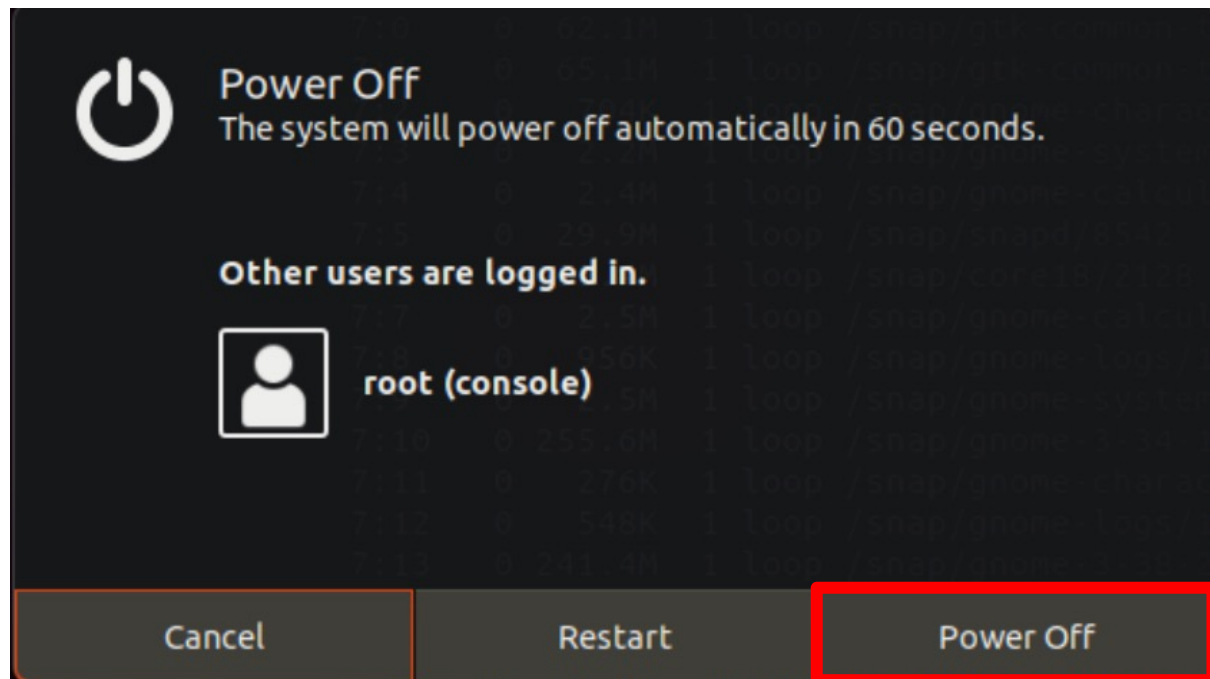
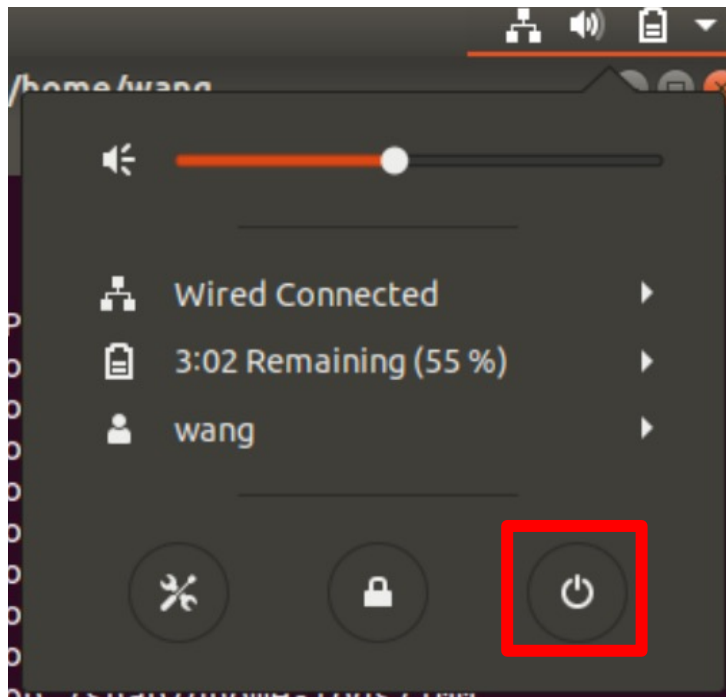
舉例

- 假設原先資料夾/**mnt/sdcl**中的資料儲存在/**dev/sdal**上
 - 往後存取資料夾/**mnt/sdcl**相當於存取設備/**dev/sdal**中的資料
 - 此時系統上並沒有掛載設備/**dev/sdcl**
- 後來將設備/**dev/sdcl**掛載到資料夾/**mnt/sdcl**
 - 往後存取資料夾/**mnt/sdcl**相當於存取設備/**dev/sdcl**中的資料
 - 此時掛載後會發現：原先在資料夾/**mnt/sdcl**的資料找不到
 - => 不是不見，只是暫時無法存取
 - => 因為那些資料存放在/**dev/sdal**上，如果要取得舊資料，則卸載設備/**dev/sdcl**即可

Umount



LAB: 新增硬碟並掛載



設定VIRTUALBOX



工具



新增(N)



設定(S)



捨棄



啟動(T)



Ubuntu18.04 - 1



已關閉電源



一般

名稱: Ubuntu18.04 - 1
作業系統: Ubuntu (64-bit)

- 一般
- 系統
- 顯示
- 存放裝置**
- 音訊
- 網路
- 序列埠
- USB
- 共用資料夾
- 使用者介面

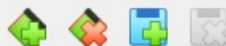
存放裝置

存放裝置(S)

- 控制器: IDE
 - 空的
- 控制器: SATA
 - Ubuntu18.04 - 1.vdi

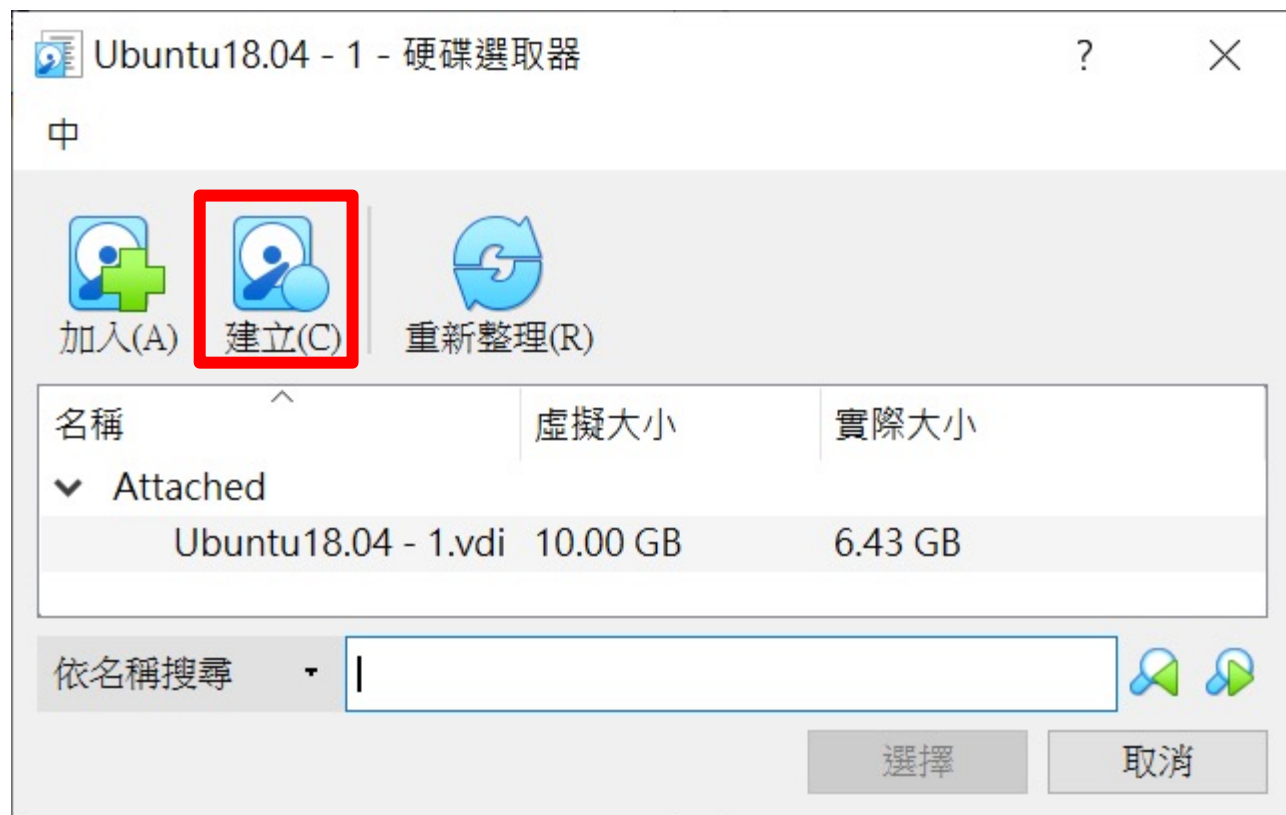
屬性

- 名稱(N): SATA
- 類型(T): AHCI
- 連接埠數(P): 1
- ☐ 使用主機 I/O 快取



確定

取消



← 建立虛擬硬碟

存放裝置在實體硬碟

請選擇新的虛擬硬碟檔是否應根據使用來成長 (動態分配)，或建立為其最大大小 (固定大小)。

動態分配的硬碟檔將只在填滿時使用實體硬碟的空間 (直到最大的**固定大小**)，儘管它的空間釋放時不會再次自動縮小。

固定大小硬碟檔在某些系統需要比較長的時間建立，但通常用起來比較快。

☒ 動態分配(D)

☐ 固定大小(F)

下一個(N)

取消

← 建立虛擬硬碟

檔案位置 and 大小

請在以下的方塊中輸入新虛擬硬碟檔的名稱，或按一下資料夾圖示以選擇建立檔案的其它資料夾。

C:\Users\L14\VirtualBox VMs\Ubuntu18.04 - 1\Ubuntu18.04 - 1_1



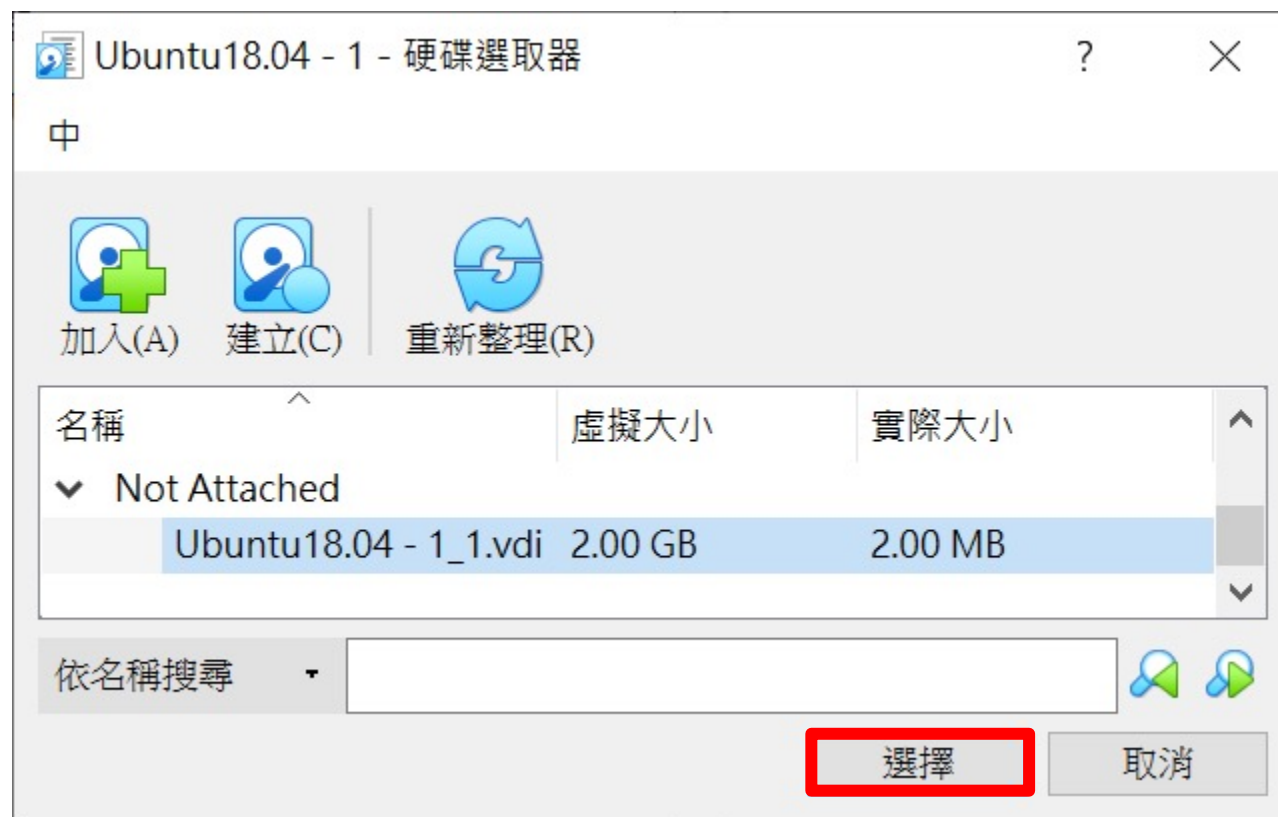
選擇虛擬硬碟的大小 (以 MB 位元組為單位)。這個大小是對虛擬機器將能夠存儲在硬碟上的檔案資料量的限制。



建立

取消

建立完成後重新開啟虛擬機



找到先前新增的硬碟

```
sda                8:0      0      10G    0 disk
└─sda1             8:1      0      10G    0 part
   └─ubuntu--vg-root 253:0    0        9G    0 lvm  /
      └─ubuntu--vg-swap 1 253:1    0      976M    0 lvm  [SWAP]
sdb                8:16     0       1G    0 disk
sdc                8:32     0       2G    0 disk
```

硬碟分割

```
root@wang-VirtualBox:/home/wang# fdisk /dev/sdb

Welcome to fdisk (util-linux 2.31.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0x5dbff80f.

Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-2097151, default 2048): 2048
Last sector, +sectors or +size{K,M,G,T,P} (2048-2097151, default 2097151): +100
M

Created a new partition 1 of type 'Linux' and of size 100 MiB.
```

硬碟分割

```
Command (m for help): p
Disk /dev/sdb: 1 GiB, 1073741824 bytes, 2097152 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
Disklabel type: dos
Disk identifier: 0x5dbff80f
```

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sdb1		2048	206847	204800	100M	83	Linux

```
Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.
```


格式化PARTITION

```
root@wang-VirtualBox:/home/wang# mkfs /dev/sdb1
mke2fs 1.44.1 (24-Mar-2018)
Creating filesystem with 102400 1k blocks and 25688 inodes
Filesystem UUID: 6ebe284a-02b2-4cbe-ba01-892b01c6875c
Superblock backups stored on blocks:
 8193, 24577, 40961, 57345, 73729

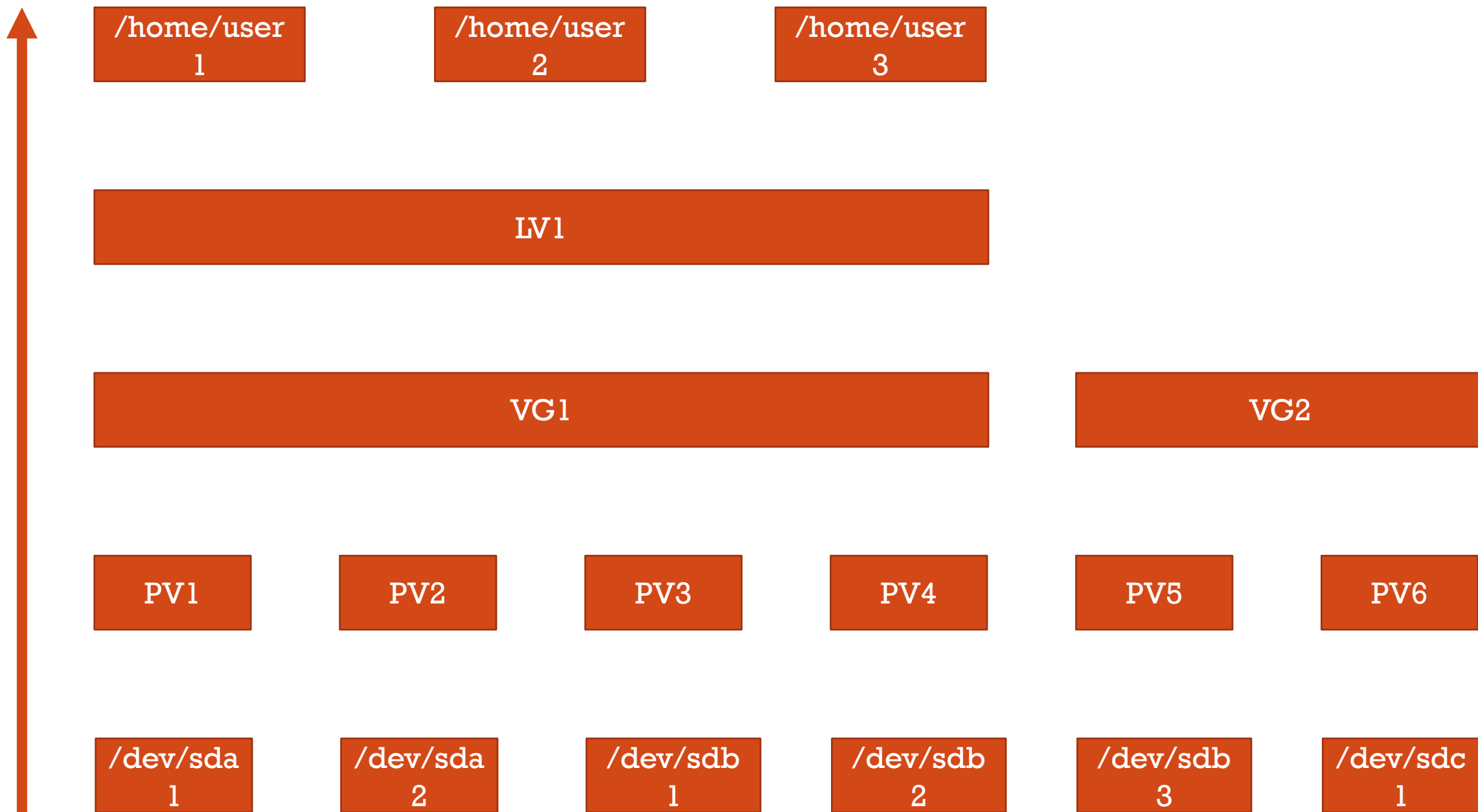
Allocating group tables: done
Writing inode tables: done
Writing superblocks and filesystem accounting information: done
```


掛載並檢查

```
root@wang-VirtualBox:/home/wang# mkdir /home/wang/project1
root@wang-VirtualBox:/home/wang# mount /dev/sdb1 /home/wang/project1/
root@wang-VirtualBox:/home/wang# df -h
/dev/sdb1                93M  1.6M   85M   2% /home/wang/project1
```

LVM

- **Logical Volume Manager**可以彈性調整硬碟大小，將多個 **partition**組成一個**volume group**
- 可以將**Volume Group**視為一顆大硬碟，再將大硬碟分割成 **Logical Volume**，而後將**LV**進行掛載與使用



LVM範例

```
root@wang-VirtualBox:/home/wang# fdisk /dev/sdb
```

```
Welcome to fdisk (util-linux 2.31.1).
```

```
Changes will remain in memory only, until you decide to write them.
```

```
Be careful before using the write command.
```

```
root@wang-VirtualBox:/home/wang# fdisk /dev/sdb
```

```
Welcome to fdisk (util-linux 2.31.1).
```

```
Changes will remain in memory only, until you decide to write them.
```

```
Be careful before using the write command.
```

```
Command (m for help): n
```

```
Partition type
```

```
Using default type: p primary (2 primary, 0 extended, 2 free)
```

```
Selected type: e extended (container for logical partitions)
```

```
First sector (616448-2097151, default 616448):
```

```
Last sector, +sectors or +size{K,M,G,T,P} (616448-2097151, default 2097151):
```

```
200M
```

```
First sector (616448-2097151, default 616448):
```

```
Last sector, +sectors or +size{K,M,G,T,P} (616448-2097151, default 2097151):
```

```
+200M
```

```
Created a new partition 3 of type 'Linux' and of size 200 MiB.
```

```
The partition table has been altered.
```

```
Calling ioctl() to re-read partition table.
```

```
Syncing disks.
```

新增partitions

- 找到partition名稱

```
└─sdb3          8:19    0   200M    0 part
└─sdb4          8:20    0   200M    0 part
```

- 使用pvcreate指令建立pv

```
root@wang-VirtualBox:/home/wang# pvcreate /dev/sdb3 /dev/sdb4
Physical volume "/dev/sdb3" successfully created.
Physical volume "/dev/sdb4" successfully created.
```

- 檢查pv是否建立

```
root@wang-VirtualBox:/home/wang# pvdisplay /dev/sdb3
```

```
-- root@wang-VirtualBox:/home/wang# pvdisplay /dev/sdb4
PV      --- Physical volume ---
VG      PV Name                    /dev/sdb4
PV      VG Name                  myvg1
AL      PV Size                  200.00 MiB / not usable 4.00 MiB
PE      Allocatable             yes
To      PE Size                  4.00 MiB
Fr      Total PE                 49
AL      Free PE                  49
PV      Allocated PE             0
        PV UUID                  e6YKa9-e3J9-jFNQ-Zpof-8Lr6-3Nza-pcaI4G
```


使用vgcreate建立vg

```
root@wang-VirtualBox:/home/wang# vgcreate myvg1 /dev/sdb3 /dev/sdb4
Volume group "myvg1" successfully created
```

檢查vg狀態

```
root@wang-VirtualBox:/home/wang# vdisplay myvg1
--- Volume group ---
VG Name                myvg1
System ID
Format                 lvm2
Metadata Areas         2
Files
Metadata Sequence No  2
VG Access              read/write
VG Status              resizable
MAX LV                 0
Cur LV                1
Open LV                0
Max PV                 0
Cur PV                2
Act PV                 2
VG Size                392.00 MiB
PE Size                4.00 MiB
Total PE               98
Alloc PE / Size        50 / 200.00 MiB
Free PE / Size          48 / 192.00 MiB
VG UUID                UffIlU-cOFt-9g2i-qo10-LsHa-Tbdi-lKzauE
```


- 使用 `lvcreate` 建立 `lv`

```
root@wang-VirtualBox:/home/wang# lvcreate -L 200m myvg1 -n mylv1
Logical volume "mylv1" created.
```

- 使用 `lvdisplay`，並找到 `lv`

```
--- Logical volume ---
LV Path                /dev/myvg1/mylv1
LV Name                 mylv1
VG Name                myvg1
LV UUID                nbXfRN-Z08i-wTae-rt0q-Hmzb-e8M6-00Rmpj
LV Write Access        read/write
LV Creation host, time wang-VirtualBox, 2021-09-10 13:30:26 +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    256
Block device           253:2
```

找到路徑

- 將前頁找到的路徑當一般partition使用
- 格式化成ext4

```
root@wang-VirtualBox:/home/wang# mkfs -t ext4 /dev/myvg1/mylv1
mke2fs 1.44.1 (24-Mar-2018)
/dev/myvg1/mylv1 contains a ext2 file system
        created on Fri Sep 10 13:34:47 2021
Proceed anyway? (y,N) y
Creating filesystem with 204800 1k blocks and 51200 inodes
Filesystem UUID: 9db4f6a1-cf15-428a-91ca-11ee206c45d2
Superblock backups stored on blocks:
        8193, 24577, 40961, 57345, 73729

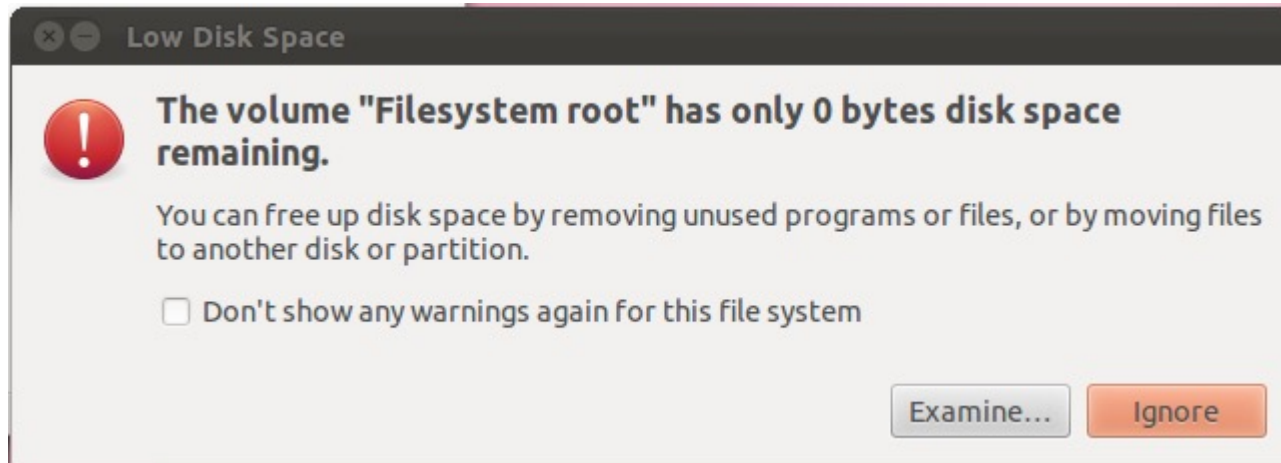
Allocating group tables: done
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done
```

■ 掛載後即可使用

```
root@wang-VirtualBox:/home/wang# mkdir /mnt/mvly1
root@wang-VirtualBox:/home/wang# mount /dev/myvg1/mylv1 /mnt/mvly1/
root@wang-VirtualBox:/home/wang# lsblk
```

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINT
sdb	8:16	0	1G	0	disk	
├─sdb1	8:17	0	100M	0	part	
├─sdb2	8:18	0	200M	0	part	
├─sdb3	8:19	0	200M	0	part	
│ └─myvg1-mylv1	253:2	0	200M	0	lvm	/mnt/mvly1
└─sdb4	8:20	0	200M	0	part	
└─myvg1-mylv1	253:2	0	200M	0	lvm	/mnt/mvly1

HOMEWORK



From :

<https://itectec.com/ubuntu-the-volume-filesystem-root-has-only-0-bytes-disk-space-remaining/>

- 情境：儲存空間不足造成錯誤
- 目的：增加儲存空間500M，且已存在的資料不能遺失
- 工具：df、fdisk、pv、vg、lv、lvresize、resize2fs
- 提示：新增硬碟/分割區、增加指定lv容量、放大檔案系統空間
- 驗收：請撰寫script完成上述功能
 - ./your-script [-n name-of-partition] [-s size]