



NASA_HW3

學號: B12902078、姓名: 蔡明龍

Reference:

分割硬碟FDISK設定教學

前言：一顆新買來的硬碟是必須經過FDISK的處理才可以使用的，如果您的觀念裡面，買回家的硬碟馬上可以使用那可是錯的。所以FDISK這個步驟相當重要。（也有其他的分割硬碟程式 ex:SPFdisk）
<http://www.slime.com.tw/teach/fdisk-jar.htm>

替 Linux 新增硬碟（磁碟分割、格式化與掛載） - G. T. Wang

最近要替我的 Linux Server 增加一顆硬碟，一般若是在安裝 Linux 時就將硬碟裝上去的話，就可以 [...]


 <https://blog.gtwang.org/linux/linux-add-format-mount-harddisk/>



How to know the suffix of size in lvcreate

I've to create a new volume group of 5G with lvcreate but the disk don't have enough space disk :


`lvcreate -n lv_new -L5G VGroup`

 <https://unix.stackexchange.com/questions/710696/how-to-know-the-suffix-of-size-in-lvcreate>



Create and mount an XFS file system on Linux

Create and mount an XFS file system on Linux? XFS is a high-performance file system which was design by SGI for their IRIX platform.

 <https://bobcares.com/blog/create-and-mount-an-xfs-file-system-on-linux/>

開機自動掛載

 <https://dywang.csie.cyut.edu.tw/dywang/rhcsaNote/node59.html>

ChatGPT

A conversational AI system that listens, learns, and challenges

 <https://chat.openai.com/share/73c5d570-ca86-42c6-a9bc-ac1d6a183b56>

ChatGPT ●

VM Setup

lsblk:

```
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINTS
fd0 2:0 1 4K 0 disk
loop0 7:0 0 748.3M 1 loop /run/archiso/airootfs
sr0 11:0 1 872.3M 0 rom /run/archiso/bootmnt
sr1 11:1 1 1024M 0 rom
vda 254:0 0 1G 0 disk
vdb 254:16 0 1G 0 disk
vdc 254:32 0 1G 0 disk
vdd 254:48 0 1G 0 disk
vde 254:64 0 1G 0 disk
vdf 254:80 0 1G 0 disk
```

1. Initial partition

指令: `fdisk /dev/vda`、進入 `fdisk` 後, 依照選單內容(), 依照題目劃分對應的磁區出來

fdisk:

```
Command (m for help): p
Disk /dev/vda: 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: gpt
Disk identifier: C5C1F985-688E-41B4-BEBA-12953D8ECAB8

Device      Start      End Sectors  Size Type
/dev/vda1    2048      4096      2048    1M BIOS boot
/dev/vda2    4096    413696   409600    200M Linux filesystem
/dev/vda3   413696 2095104 1681408    821M Linux filesystem

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

lsblk:

```
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINTS
fd0 2:0 1 4K 0 disk
loop0 7:0 0 748.3M 1 loop /run/archiso/airootfs
sr0 11:0 1 872.3M 0 rom /run/archiso/bootmnt
sr1 11:1 1 1024M 0 rom
vda 254:0 0 1G 0 disk
├─vda1 254:1 0 1M 0 part
├─vda2 254:2 0 200M 0 part
└─vda3 254:3 0 821M 0 part
```

2. RAID Setup

For creating RAID data, command:

```
mdadm --create --verbose --level=10 --metadata=1.2 --raid-devices=4 /dev/md/data /dev/vdc /dev/vdd /dev/vde /dev/vdf
```

For creating RAID linux, command:

```
mdadm --create --verbose --level=0 --metadata=1.2 --raid-devices=2 /dev/md/linux /dev/vda3 /dev/vdb
```

Result:

```
2 root@archiso ~ # cat /proc/mdstat
Personalities : [raid10] [raid0]
md126 : active raid0 vdb[1] vda3[0]
      1885184 blocks super 1.2 512k chunks

md127 : active raid10 vdf[3] vde[2] vdd[1] vdc[0]
      2093056 blocks super 1.2 512K chunks 2 near-copies [4/4] [UUUU]

unused devices: <none>
```


```
ARRAY /dev/md/data metadata=1.2 name=archiso:data UUID=cb7b2077:a7239651:ec2d708c:fc8f313
ARRAY /dev/md/linux metadata=1.2 name=archiso:linux UUID=e9c7047d:83811389:31201c65:9571f3bc
```

3. Disk encryption

Reference:

How To Linux Hard Disk Encryption With LUKS [cryptsetup encrypt command]

Explains how to use cryptsetup encryption command to encrypt partitions or hard disk on your Linux based Laptop/server/block storage/computer

 <https://www.cyberciti.biz/security/howto-linux-hard-disk-encryption-with-luks-cryptsetup-command/>

```
/dev/mapper/backup2 is active
type: luks
cipher: aes-xts-plaint64
keysize: 512 bits
key location: keyring
device: /dev/sdc
sector size: 512
offset: 32768 sectors
size: 167768832 sectors
mode: read/write
Command successful
root@archiso ~ # mkfs.ext4 /dev/mapper/backup2
mkfs: 145.5 (07-Jun-2020)
Creating filesystem with 209711104 4k blocks and 5242880 inodes
filesystem UUID: 727f8f5d-1797-4ae8-a2d0-58e683c7b3bf
superblock backups stored on blocks:
32768, 98304, 163840, 229376, 294912, 819200, 884736, 160
4096000, 7962624, 11239424, 20480000, 23887872, 71663616,
102400000
```

Linux Hard Disk Encryption With LUKS [cryptsetup command] - nixCraft (cyberciti.biz).

zh-tw/HowTos/EncryptedFilesystem

Windows 及 Mac OS X 各自提供了它們的標準加密檔案系統工具，但 Linux 當然提供了大量工具來完成這個工作。現時盛行的工具似乎是 dm-crypt。透過 cryptsetup 這個工具程式，dm-crypt 為 Linux 提供了一個頗為簡潔及易用的加密檔案系統工具。

[https://wiki.centos.org/zh\(2d\)tw\(2f\)HowTos\(2f\)EncryptedFilesystem.html](https://wiki.centos.org/zh(2d)tw(2f)HowTos(2f)EncryptedFilesystem.html)

dm-crypt/Device encryption - ArchWiki

This section covers how to manually utilize dm-crypt from the command line to encrypt a system.

 https://wiki.archlinux.org/title/Dm-crypt/Device_encryption

作法

/dev/vda2 以 luks1 加密，選擇一個好記的密碼，將加密之後的裝置路徑設在/dev/mapper/cryptboot。

指令: `sudo cryptsetup luksFormat --type luks1 /dev/vda2`

密碼: `nasa2024`

```
1 root@archiso ~ # sudo cryptsetup luksFormat --type luks1 /dev/vda2
WARNING!
=====
This will overwrite data on /dev/vda2 irrevocably.

Are you sure? (Type 'yes' in capital letters): YES
Enter passphrase for /dev/vda2:
Verify passphrase:
```

用 dd 指令以及 Linux 的隨機數產生裝置生成一個 256 字元的隨機檔案，並用它作為 /dev/vda2 的第二個密碼。

指令: `sudo dd if=/dev/urandom of=/root/key1 bs=1 count=256`

指令: `sudo cryptsetup luksAddKey /dev/md/linux /root/key1`

(第二個key)

scp:

```
$ scp -P 8668 root@140.112.30.186:/root/key1 ./key1
root@140.112.30.186's password:
key1
```

- /dev/md/linux 以 luks2 加密，使用先前生成的隨機檔案作為密碼，將加密之後的裝置路徑設在 /dev/mapper/cryptroot。

指令: `sudo cryptsetup luksFormat --type luks2 /dev/md/linux` 、 `sudo cryptsetup luksAddKey /dev/md/linux /root/key1`

(相同的key)

/dev/md/data 以 luks2 加密，使用先前生成的隨機檔案作為密碼，將加密之後的裝置路徑設在 /dev/mapper/cryptdata。

指令: `sudo cryptsetup luksFormat --type luks2 /dev/md/data` 、 `sudo cryptsetup luksAddKey /dev/md/data /root/key1`

分別: `cryptsetup open /dev/disk name (under /dev/mapper)` ，把他們開啟，並掛上去

```
vda                254:0    0    1G 0 disk
├─vda1             254:1    0    1M 0 part
├─vda2             254:2    0   200M 0 part
├─┬─cryptboot      253:0    0   198M 0 crypt
├─┬─vda3           254:3    0   821M 0 part
├─┬─┬─md126        9:126    0    1.8G 0 raid0
├─┬─┬─┬─cryptroot  253:1    0    1.8G 0 crypt
├─vdb              254:16   0    1G 0 disk
├─┬─md126          9:126    0    1.8G 0 raid0
├─┬─┬─cryptroot    253:1    0    1.8G 0 crypt
├─vdc              254:32   0    1G 0 disk
├─┬─md127          9:127    0    2G 0 raid10
├─┬─┬─cryptdata    253:2    0    2G 0 crypt
├─vdd              254:48   0    1G 0 disk
├─┬─md127          9:127    0    2G 0 raid10
├─┬─┬─cryptdata    253:2    0    2G 0 crypt
├─vde              254:64   0    1G 0 disk
├─┬─md127          9:127    0    2G 0 raid10
├─┬─┬─cryptdata    253:2    0    2G 0 crypt
├─vdf              254:80   0    1G 0 disk
├─┬─md127          9:127    0    2G 0 raid10
├─┬─┬─cryptdata    253:2    0    2G 0 crypt
```

- /etc/crypttab content

```
root@archiso ~ # cat /etc/crypttab
# Configuration for encrypted block devices.
# See crypttab(5) for details.

# NOTE: Do not list your root (/) partition here, it must be set up
#        beforehand by the initramfs (/etc/mkinitcpio.conf).

# <name>      <device>      <password>      <options>
# home        UUID=b8ad5c18-f445-495d-9095-c9ec4f9d2f37 /etc/mypassword1
# data1       /dev/sda3      /etc/mypassword2
# data2       /dev/sda5      /etc/cryptfs.key
# swap        /dev/sdx4      /dev/urandom     swap,cipher=aes-cbc-essiv:sha256,size=256
# vol         /dev/sdb7      none
cryptboot    /dev/vda2      /root/key1       luks
cryptroot    /dev/md/linux  /root/key1       luks
cryptdata    /dev/md/data   /root/key1       luks
```

LVM Setup

建立一個 Volume Group (VG)，稱為 linux，只包含 /dev/mapper/cryptroot 一個 Physical Volume (PV)。

指令: `pvcreate /dev/mapper/cryptroot` 、
`vgcreate linux /dev/mapper/cryptroot`

```
root@archiso ~ # pvs;vgs
PV          VG          Fmt  Attr  PSize  PFree
/dev/mapper/cryptroot linux lvm2 a--  1.78g  1.78g
VG          #PV  #LV  #SN Attr   VSize VFree
linux       1    0    0 wz--n-  1.78g  1.78g
```

```
root@archiso ~ # lvs
LV  VG      Attr      LSize   Pool Origin Data%  Meta%   Move Log Cpy%Sync Convert
home linux -wi-a----- 256.00m
root linux -wi-a-----  1.53g
```

在 VG linux 下，建立一個 Logical Volume (LV)，稱為 home，大小為 256M。
在 VG linux 下，建立一個 Logical Volume (LV)，稱為 root，佔據剩下的所有空間。

指令: `lvcreate -L +256M -n home linux` 、 `lvcreate -l 100%FREE -n root linux`

```
root@archiso ~ # lvs
LV VG Attr LSize Pool Origin Data% Meta% Move Log Cpy%Sync Convert
home linux -wi-a----- 256.00m
root linux -wi-a----- 1.53g
```

Formatting

使用指令: `mkfs.filesystem`

`/dev/disk`，按照題目的敘述去
設定對應的filesystem

```
vda
├─vda1
├─vda2 crypto_LUKS 1
├─┬─cryptboot ext2 1.0
├─vda3 linux_raid_member 1.2
├─┬─md126 crypto_LUKS 2
├─┬─cryptroot
├─┬─┬─LVM2_member LVM2 001
├─┬─┬─┬─linux-home
├─┬─┬─┬─┬─ext4 1.0
├─┬─┬─┬─┬─┬─linux-root
├─┬─┬─┬─┬─┬─┬─ext4 1.0
├─vdb linux_raid_member 1.2
├─┬─md126 crypto_LUKS 2
├─┬─┬─cryptroot LVM2_member LVM2 001
├─┬─┬─┬─linux-home
├─┬─┬─┬─┬─ext4 1.0
├─┬─┬─┬─┬─┬─linux-root
├─┬─┬─┬─┬─┬─┬─ext4 1.0
├─vdc linux_raid_member 1.2
├─┬─md127 crypto_LUKS 2
├─┬─┬─cryptdata xfs
├─vdd linux_raid_member 1.2
├─┬─md127 crypto_LUKS 2
├─┬─┬─cryptdata xfs
├─vde linux_raid_member 1.2
├─┬─md127 crypto_LUKS 2
├─┬─┬─cryptdata xfs
├─vdf linux_raid_member 1.2
├─┬─md127 crypto_LUKS 2
├─┬─┬─cryptdata xfs
```

Mounting

使用指令: `mount <device> <mounting_point>`

依照題目的敘述去掛載硬碟

```
/dev/mapper/linux-root 1.5G 428K 1.4G 1% /mnt
/dev/mapper/linux-home 230M 78K 214M 1% /mnt/home
/dev/mapper/cryptboot 185M 14K 175M 1% /mnt/boot
/dev/mapper/cryptdata 2.0G 71M 1.9G 4% /mnt/data
```

Trivia

a. 一個使用中的硬碟上除了已經切好的 partition 外，還會有什麼東西？


答: 通常會有磁碟的metadata存在，負責記錄一些磁碟訊息還有系統資訊等，如果該硬碟是開機碟，則會再包含啟動作業系統相關的程式與資料。

b. 在工作站上輸入 `df /` 指令時，會得到下圖的輸出，請問為什麼 Available + Used < 1Kblocks？

Reference:

df -h - Used space + Avail Free space is less than the Total size of /home

As the output of df -h shows here, something is eating up 5GB of free space. So, it's not available to use.

 <https://askubuntu.com/questions/249387/df-h-used-space-avail-free-space-is-less-than-the-total-size-of-home>



答: 依照上面這個文章, 如果是ext[2..4]的filesystem(該硬碟屬於ext4的形式), 會預留大概5%的空間專門給root來使用, 為了避免其他user把disk填滿, 導致系統在啟動時會出現問題。

```
b12902078@ws1 [~] df -hT | grep /dev/mapper
egrep: warning: egrep is obsolescent; using grep -E
/dev/mapper/ws-root                ext4      125G    49G    71G    41% /
```

- c. 什麼是 FUSE (Filesystem in Userspace) ? 請簡單介紹它的原理, 給出一個優點和缺點, 並舉出一個在 Linux 上用了這項技術才能支援的檔案系統。

Reference:

FUSE - ArchWiki

Filesystem in Userspace (FUSE) is a mechanism for Unix-like operating systems that lets non-privileged users create their own file systems without editing kernel code. This is achieved by running file system code in user space, while the FUSE kernel module provides only a "bridge" to the actual kernel interfaces.

 <https://wiki.archlinux.org/title/FUSE>

FUSE — The Linux Kernel documentation

A filesystem in which data and metadata are provided by an ordinary userspace process. The filesystem can be accessed normally through the kernel interface.

<https://www.kernel.org/doc/html/latest/filesystems/fuse.html>

答: FUSE是一個能夠在userspace層面實作出來的檔案系統, 跟傳統需要在kernel層面上實作有所不同, 所以他具備了很好的靈活性(優點), 不需要使用到最高的權限, 一般的使用者就可以去把FUSE掛載起來, 如果出現了BUG, 造成的影響也比傳統的filesystem小很多, 但缺點也明顯, 由於不是在kernel層面上實作出來, 所以在效率上會比較低。

- d. 你在幫新的硬碟建立分割區的時候可能看過 MBR 和 GPT 這兩個詞。請問他們分別代表了什麼, 並列出至少兩個差別。

Reference:

GPT 與 MBR 格式是什麼? 有什麼差異? 該選哪一種? | Ted聊科技

GPT 與 MBR 是兩種不同的硬碟分割方式, 它們各有優缺點, 適用於不同的情況。在這篇文章中, 你將學習到 GPT 與 MBR 的定義、優缺點、比較和選擇方法, 並且根據自己的需求做出適合的選擇。

 <https://adersaytech.com/tutorial/kb-article/gpt-mbr-compare.html>



答: MBR:(Master bott Record), 為早期的硬碟分割方式, GPT(GUID Partition Table)則為目前常用的磁碟分割標準

差別:

- MBR最多只支援四個分割區, 但GPT可以支援128個分割區

- MBR最大只能支援2.2TB的硬碟，GPT則可以支援更大的硬碟
 - GPT必須配合UEFI的主板來啟動，而MBR則是直接透過在ROM裡的啟動程式，因此MBR的相容性比GPT好
- e. 許多修 NASA 的同學都會有一台同時裝了 Windows 和 Linux 的電腦。為了避免浪費儲存空間，你可能會享有同時被這兩個作業系統支援的硬碟來存放你常用的大檔案 (e.g. Steam遊戲庫)。請找到滿足以下條件的檔案系統，並給出掛載這個檔案系統需要用到的指令 (例如：`mount -t ext4 /dev/sda /mnt`)
- 答:
- f. 根據 SI 及 IEC 的定義，GB 和 GiB 分別指的是多少 Bytes？在工作站上使用 `ls -lh` 時，預設使用的是哪種單位呢？

Reference:

記憶體容量換算 (GB vs. GiB) | zh - HackMD

```
<style> #doc.markdown-body, .ui-infobar, .container-thiner { max-width: 1080px; } .ui-content #d
https://man.twcc.ai/@twccdocs/concept-ccs-memory-conversion-zh
```

答: $1\text{GB} = 10^9 \text{ bytes}$ 、 $1\text{GiB} = 2^{30} \text{ bytes}$ ，透過 `ls -l`、`ls -lh` 的結果來做對照，可以發現上面的 $4.0\text{K} = 4 * 2^{10} \text{ bytes}$ ，所以可以推論為MiB

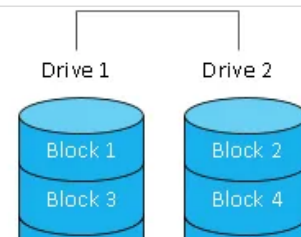
- g. 假設都是 4 個硬碟，請比較它們組成 RAID 0, RAID 10, RAID 5, RAID 6 的容錯能力、讀取速度、寫入速度。容錯能力指的是最多可以幾顆硬碟損壞而不造成資料遺失。而讀取和寫入速度則假設不需考慮計算、讀寫 parity 資料的額外時間。

Reference:

RAID Level Comparison: RAID 0, RAID 1, RAID 5, RAID 6 and RAID 10

In previous post, we have talked about the differences between hardware RAID and software RAID. We will compare different RAID levels this time. The RAID

<https://www.dataplugs.com/en/raid-level-comparison-raid-0-raid-1-raid-5-raid-6-raid-10/>



答:

- Raid 0
 - 容錯能力: 沒有，因為Raid 0就像是把很多硬碟組成一個大硬碟
 - 讀取速度: 最快
 - 寫入速度: 最快
- Raid 10
 - 容錯能力: 有，RAID 10是兩個Raid 1組在一起的結果，在本題的情況下，最多可以兩顆硬碟資料損壞(分別在不同的raid 1)
 - 讀取速度: 比Raid 0略慢一點
 - 寫入速度: 比Raid 0還慢(因為要把資料多複製出一份)
- Raid 5

- 容錯能力: 有, 最多可以一顆硬碟資料損壞
- 讀取速度: 跟Raid 0 一樣(不考慮計算、讀寫parity的話)
- 寫入速度: 跟Raid 0 一樣(不考慮計算、讀寫parity的話)

- Raid 6

- 容錯能力: 有, 最多可以二顆硬碟資料損壞
- 讀取速度: 跟Raid 0 一樣(不考慮計算、讀寫parity的話)
- 寫入速度: 跟Raid 0 一樣(不考慮計算、讀寫parity的話)

h. 當使用 VM 時, 如果發現將它的多個虛擬硬碟組成 RAID 0 時, 其寫入速度和沒有 RAID時幾乎沒有差別, 請問可能會是什麼原因呢?

Reference:

Raid 0 Performance Decreased

I was trying to understand the performance benefits of RAID. I created 4 virtual disks in Windows 10 (10 GB each). I then set up 2 of them as Stripped volumes and the other 2 as spanned volumes. No...

<https://superuser.com/questions/1148691/raid-0-performance-decreased>



Is RAID0 inside a virtual machine faster?

Has anyone experimented with running a virtual RAID0 inside a VM? For example, two virtual disks in a software RAID0 configuration, rather than a single virtual disk.

<https://serverfault.com/questions/224909/is-raid0-inside-a-virtual-machine-faster>



Which RAID Level Should We Run For Virtual Machine Storage

We just ordered a server which we are going to use as storage for our virtual machine environment. Basically it will work as a block level storage device running iSCSI to each hypervisor.

<https://serverfault.com/questions/301129/which-raid-level-should-we-run-for-virtual-machine-storage>



答: 虛擬硬碟, 在虛擬機上實作出來, 負責模擬實體硬碟的功能, 組成Raid 0寫入速度沒有太大提升可能有幾個原因, 第一個原因可能是出在虛擬層與主機層之間, 由於虛擬硬碟是模擬出來的, 不會像實體層這麼快, 這個效能的差距可能就讓寫入速度沒有太大的差別, 第二個原因可能是, 虛擬硬碟最後在存取資料時, 會一起把資料做一個統一的處理(最後的操作可能都一樣), 所以即便使用Raid 0, 但實際上操作的結果還是與原本相同。