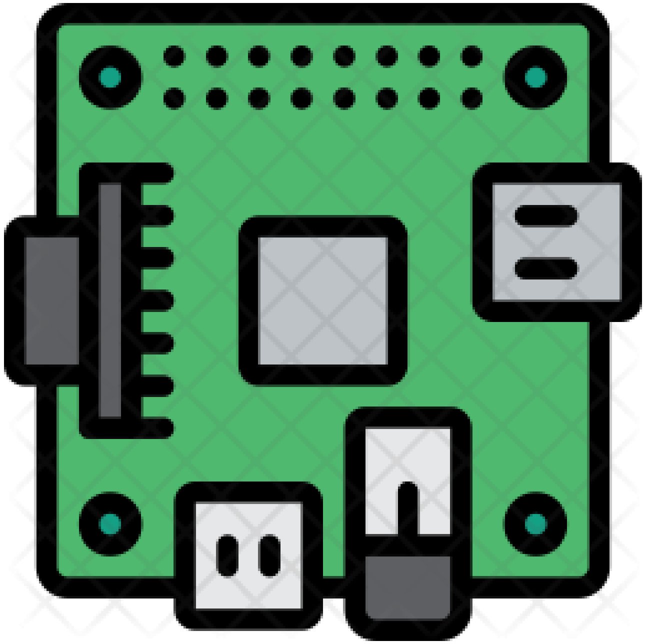


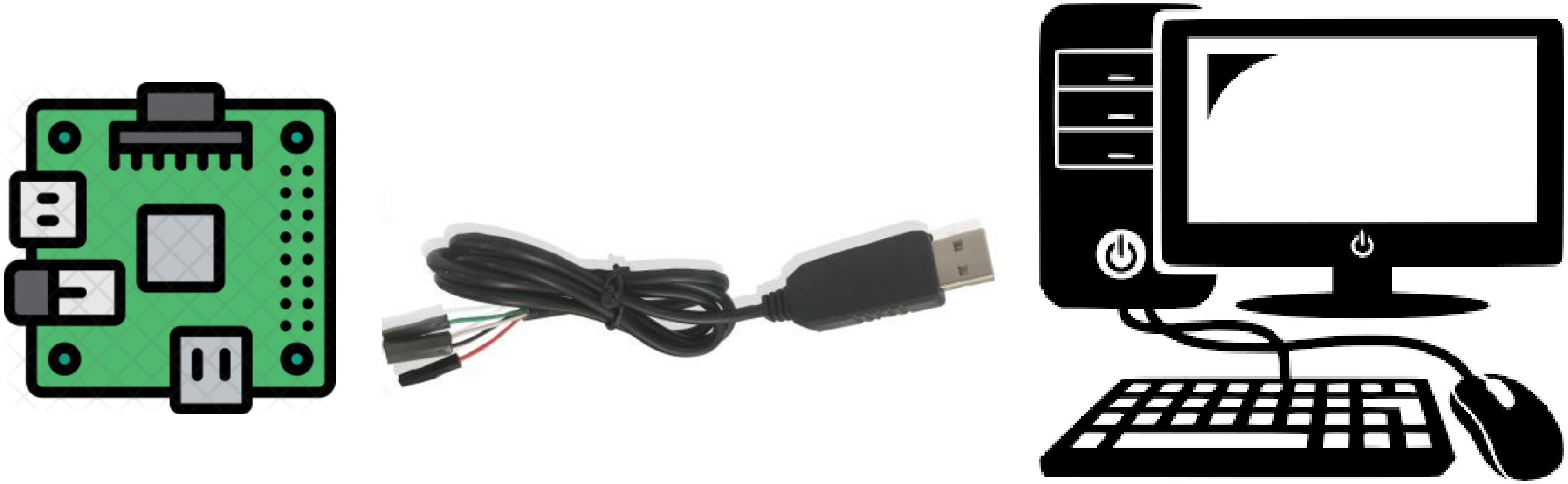
System Backup and Configuration

Peng-Sheng Chen

What can You Learn?

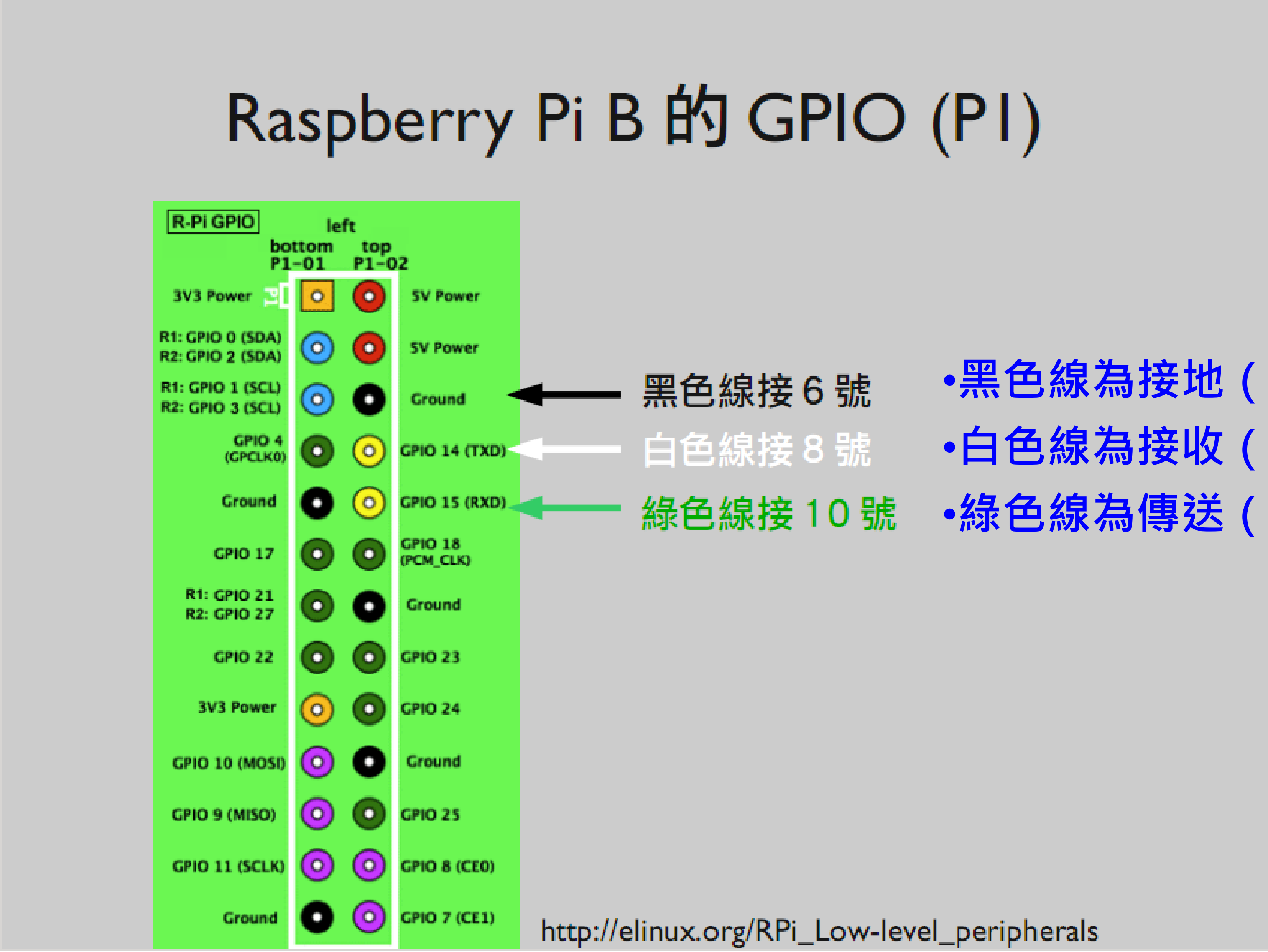
- 從序列埠登入 Raspberry Pi
- 擷取image file裡的檔案
- 備份系統: dd

從序列埠登入 Raspberry Pi (1)



USB-to-TTL (Transistor-to-Transistor Logic) serial port轉接線

從序列埠登入 Raspberry Pi (2)



Raspberry Pi B 的 GPIO (P1)

R-Pi GPIO

left

bottom P1-01	top P1-02
3V3 Power	5V Power
R1: GPIO 0 (SDA) R2: GPIO 2 (SDA)	5V Power
R1: GPIO 1 (SCL) R2: GPIO 3 (SCL)	Ground
GPIO 4 (GPCLK0)	GPIO 14 (TXD)
Ground	GPIO 15 (RXD)
GPIO 17	GPIO 18 (PCM_CLK)
R1: GPIO 21 R2: GPIO 27	Ground
GPIO 22	GPIO 23
3V3 Power	GPIO 24
GPIO 10 (MOSI)	Ground
GPIO 9 (MISO)	GPIO 25
GPIO 11 (SCLK)	GPIO 8 (CE0)
Ground	GPIO 7 (CE1)

黑色線接 6 號

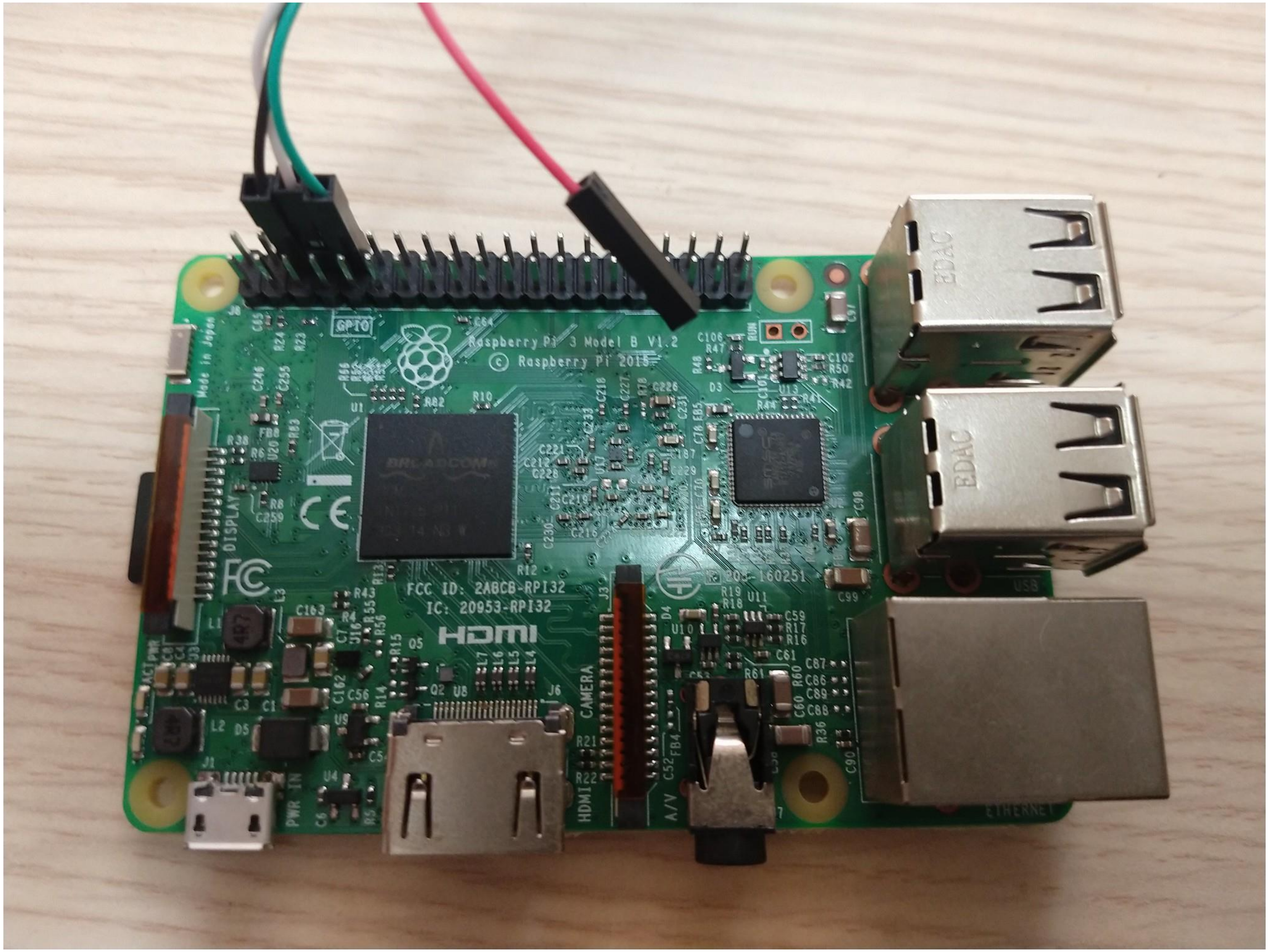
白色線接 8 號

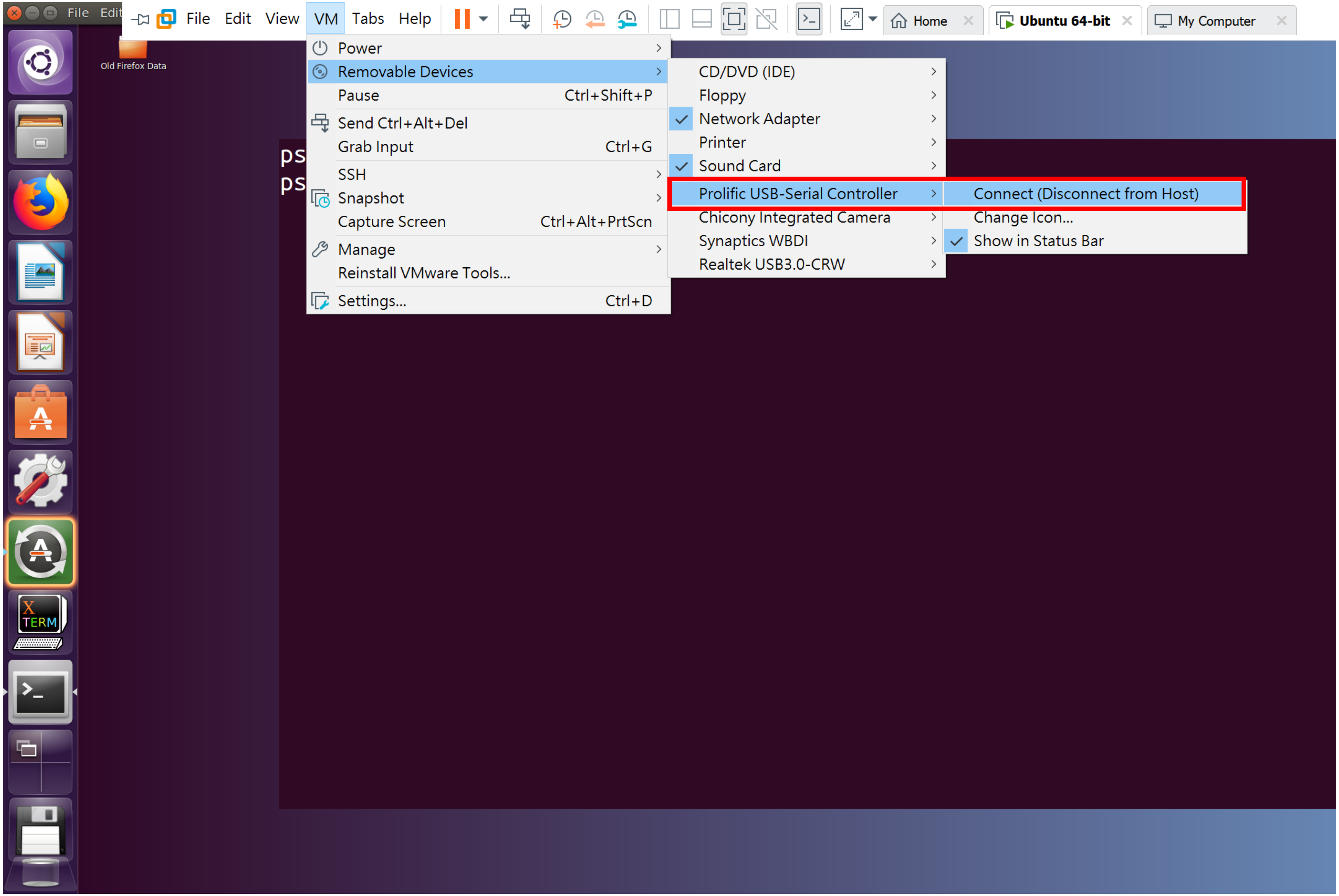
綠色線接 10 號

http://elinux.org/RPi_Low-level_peripherals

- 黑色線為接地 (GND)
- 白色線為接收 (RXD)
- 綠色線為傳送 (TXD)

從序列埠登入 Raspberry Pi (3)

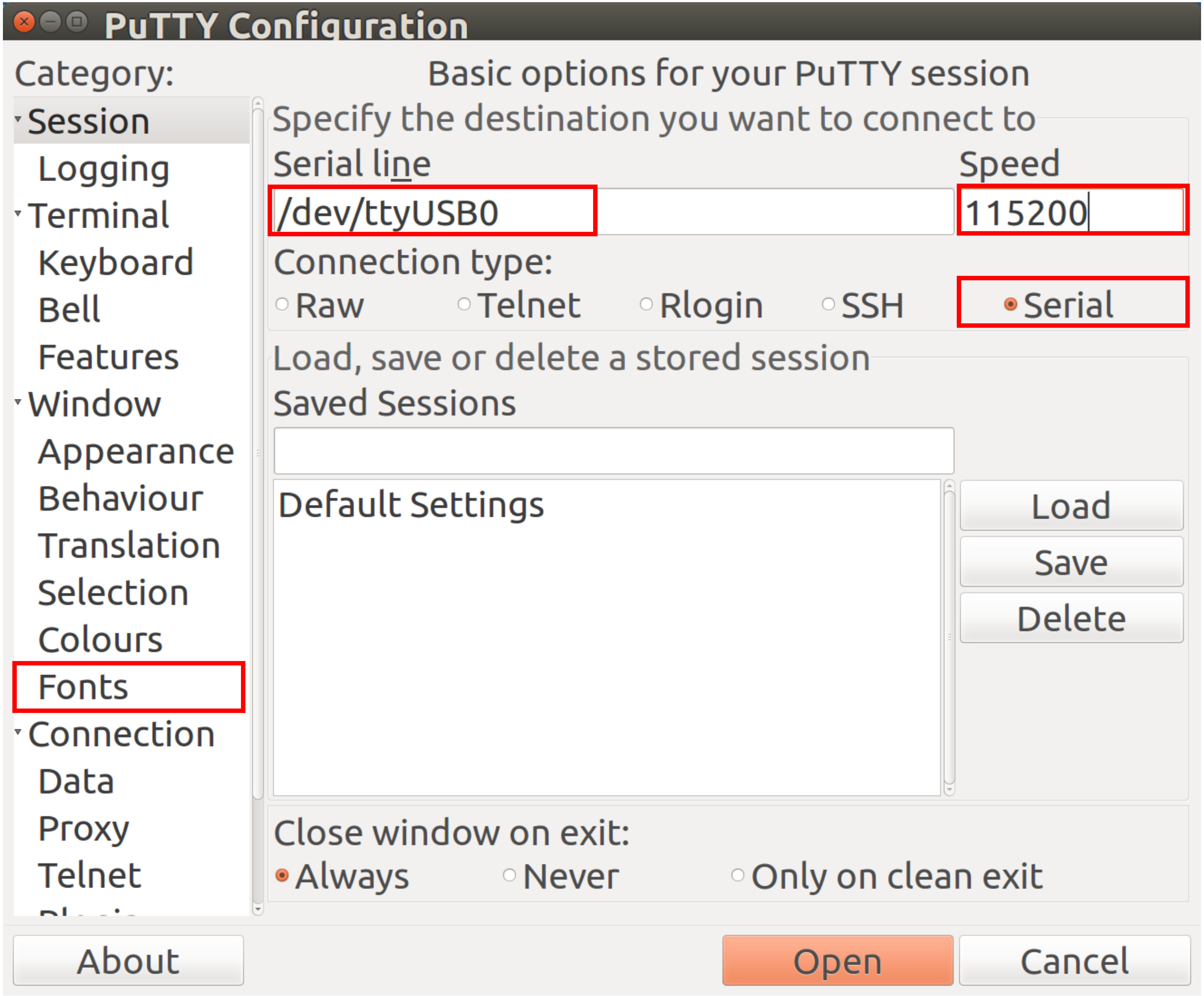



File
Edit
File
Edit
View
VM
Tabs
Help
Home
Ubuntu 64-bit
My Computer
Old Firefox Data
ps
ps
Power
Removable Devices
Pause Ctrl+Shift+P
Send Ctrl+Alt+Del
Grab Input Ctrl+G
SSH
Snapshot
Capture Screen Ctrl+Alt+PrtScn
Manage
Reinstall VMware Tools...
Settings... Ctrl+D
CD/DVD (IDE)
Floppy
Network Adapter
Printer
Sound Card
Prolific USB-Serial Controller
Chicony Integrated Camera
Synaptics WBDI
Realtek USB3.0-CRW
Connect (Disconnect from Host)
Change Icon...
Show in Status Bar
X TERM

從序列埠登入 Raspberry Pi (4)

- Change to root

```
$ sudo -s
$ screen /dev/ttyUSB0 115200



$ apt-get install putty
$ putty
```



Basic options for your PuTTY session

Specify the destination you want to connect to

Serial line: /dev/ttyUSB0

Speed: 115200

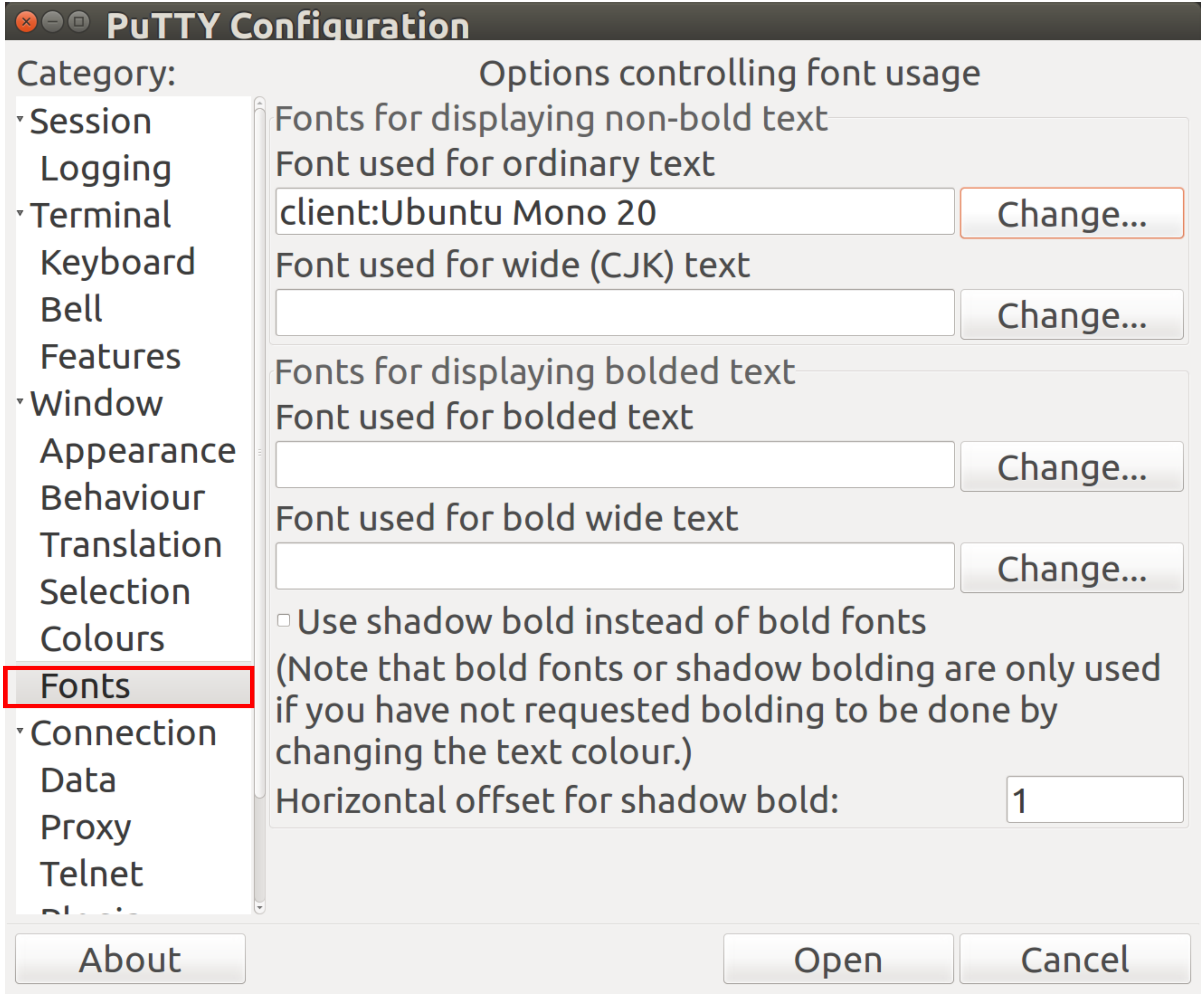
Connection type:
○ Raw ○ Telnet ○ Rlogin ○ SSH ● Serial

Load, save or delete a stored session

Saved Sessions

Default Settings

Close window on exit:
● Always ○ Never ○ Only on clean exit



Options controlling font usage

Fonts for displaying non-bold text

Font used for ordinary text: client:Ubuntu Mono 20

Font used for wide (CJK) text:

Fonts for displaying bolded text

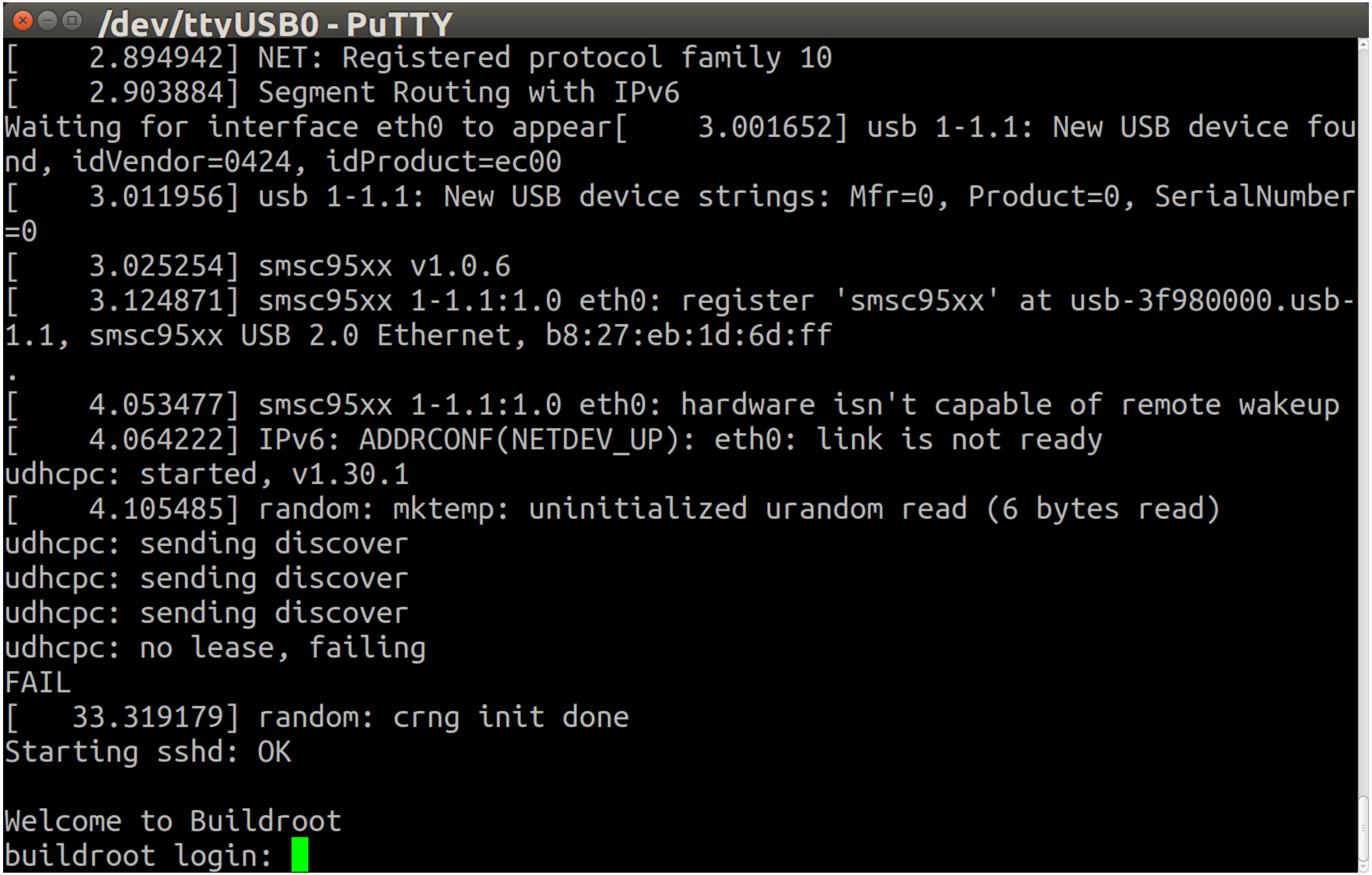
Font used for bolded text:

Font used for bold wide text:

□ Use shadow bold instead of bold fonts

(Note that bold fonts or shadow bolding are only used if you have not requested bolding to be done by changing the text colour.)

Horizontal offset for shadow bold: 1



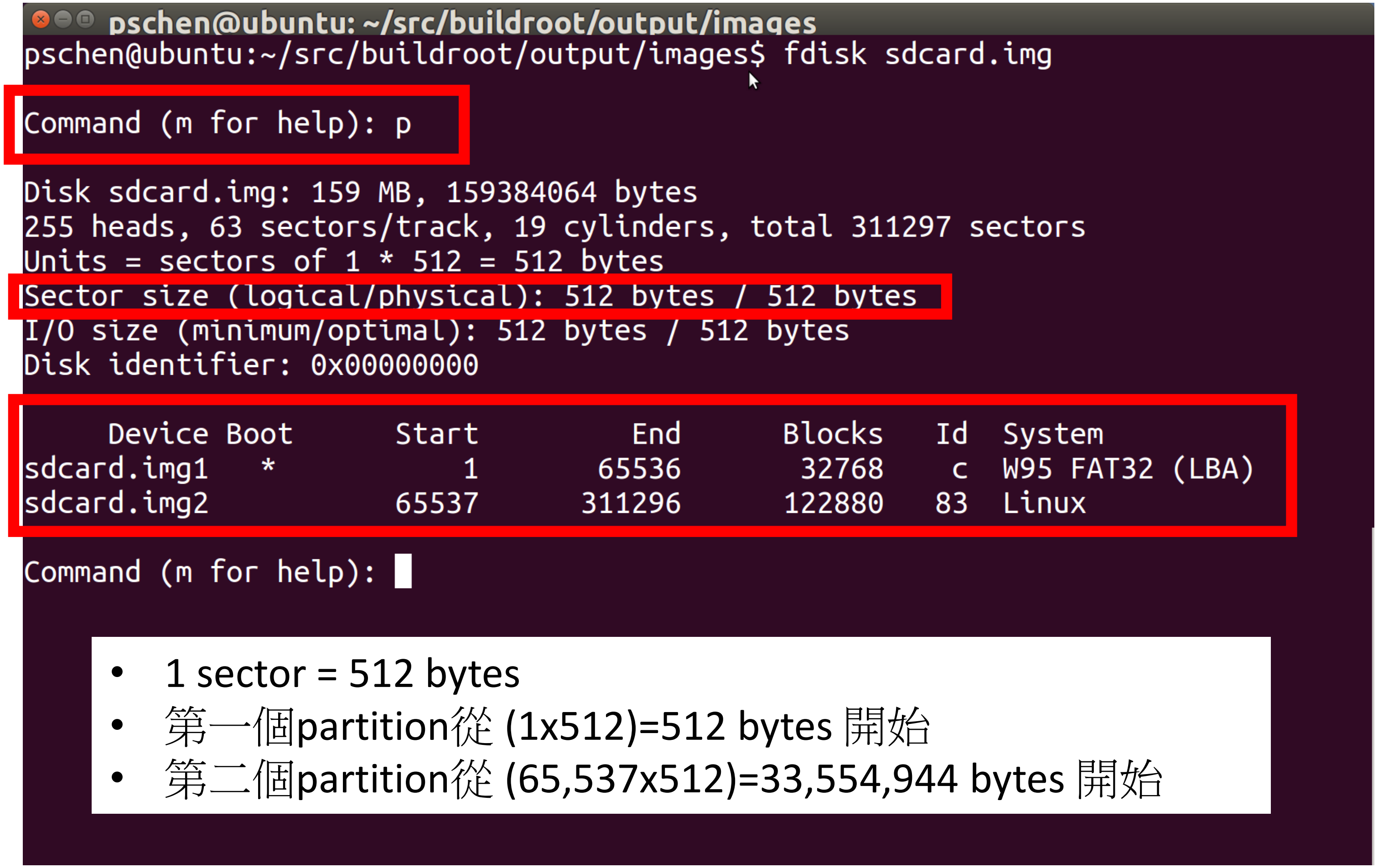
```
[    2.894942] NET: Registered protocol family 10
[    2.903884] Segment Routing with IPv6
Waiting for interface eth0 to appear[    3.001652] usb 1-1.1: New USB device fou
nd, idVendor=0424, idProduct=ec00
[    3.011956] usb 1-1.1: New USB device strings: Mfr=0, Product=0, SerialNumber
=0
[    3.025254] smsc95xx v1.0.6
[    3.124871] smsc95xx 1-1.1:1.0 eth0: register 'smsc95xx' at usb-3f980000.usb-
1.1, smsc95xx USB 2.0 Ethernet, b8:27:eb:1d:6d:ff
.
[    4.053477] smsc95xx 1-1.1:1.0 eth0: hardware isn't capable of remote wakeup
[    4.064222] IPv6: ADDRCONF(NETDEV_UP): eth0: link is not ready
udhcpc: started, v1.30.1
[    4.105485] random: mktemp: uninitialized urandom read (6 bytes read)
udhcpc: sending discover
udhcpc: sending discover
udhcpc: sending discover
udhcpc: no lease, failing
FAIL
[   33.319179] random: crng init done
Starting sshd: OK

Welcome to Buildroot
buildroot login: 
```

擷取image file裡的檔案

- buildroot最後會產生image file: **sdcard.img**
- 透過 mount，不需要把 **sdcard.img** 燒錄到sdcard，就可以擷取裡面的檔案

Mount Image file (1)

- `$ cd <buildroot目錄>/output/images/`
- `$ fdisk ./sdcard.img`   (**獲取分割區資訊**)



```
pschen@ubuntu:~/src/buildroot/output/images$ fdisk sdcard.img

Command (m for help): p

Disk sdcard.img: 159 MB, 159384064 bytes
255 heads, 63 sectors/track, 19 cylinders, total 311297 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 identifier: 0x00000000

     Device Boot      Start         End      Blocks   Id  System
sdcard.img1   *           1       65536       32768    c  W95 FAT32 (LBA)
sdcard.img2           65537      311296      122880   83  Linux

Command (m for help): 
```

- 1 sector = 512 bytes
- 第一個partition從 (1x512)=512 bytes 開始
- 第二個partition從 (65,537x512)=33,554,944 bytes 開始

Mount Image file (2)

- 第一個partition從 (1x512)=512 bytes 開始
- 第二個partition從 (65,537x512)=33,554,944 bytes 開始

```
$ mkdir P1_BOOT
$ mkdir P2_ROOTFS
```

需change to root執行mount指令

//Mount第一個partition到BOOT

```
$ sudo mount -o loop,offset=$((1*512)) sdcard.img
P1_BOOT
```

//Mount第二個partition到ROOTFS

```
$ sudo mount -o loop,offset=$((65537*512))
sdcard.img P2_ROOTFS
```

備份系統: 整個SD card備份

- 將SD card放入讀卡機，並與電腦連線
- Mount SD card，假設device file是/dev/sdd

```
// Change to root
$ sudo -s
$ dd if=/dev/sdd of=backup.img
$
```

- (SD card有多大，backup.img就有多大)
- 16GB SD card會花約25 min (I/O速度10MB/s)

備份系統: 分別備份SD card Partitions

- 分別備份BOOT partition與filesystem partition

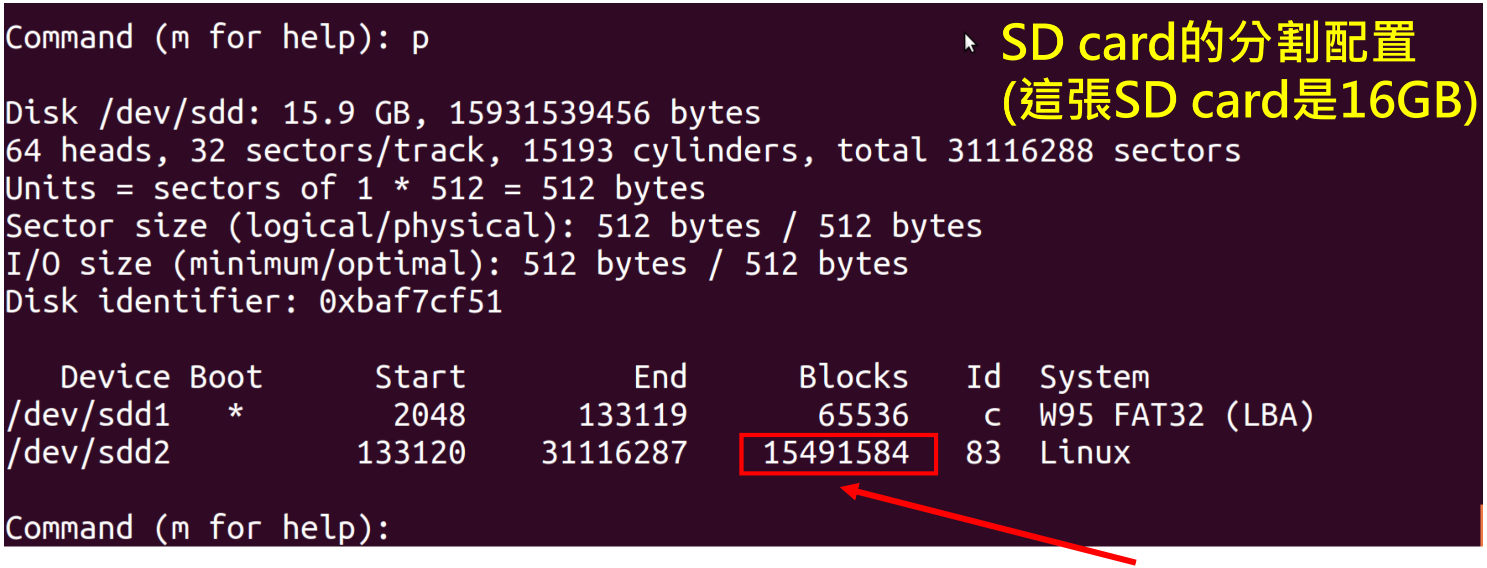
```
$ dd if=/dev/sdd1 of=boot_part.img
```

```
$ dd if=/dev/sdd2 of=filesys_part.img
```

備份系統: 使用 rsync (1)

- 利用dd建立一個image file，透過rsync將SD card資料備份在image file裡 (離線備份)
  - 利用dd建立image file
  - 透過fdisk分割partition
  - 透過losetup設定好pseudo device
  - 將SD card插入讀卡機，與電腦連接
  - 透過rsync，將SD card的資料複製到image file

備份系統: 使用 rsync (1)



```
Command (m for help): p

Disk /dev/sdd: 15.9 GB, 15931539456 bytes
64 heads, 32 sectors/track, 15193 cylinders, total 31116288 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 identifier: 0xbaf7cf51

   Device Boot      Start         End      Blocks   Id  System
/dev/sdd1   *        2048      133119       65536    c  W95 FAT32 (LBA)
/dev/sdd2          133120    31116287    15491584   83  Linux

Command (m for help):
```

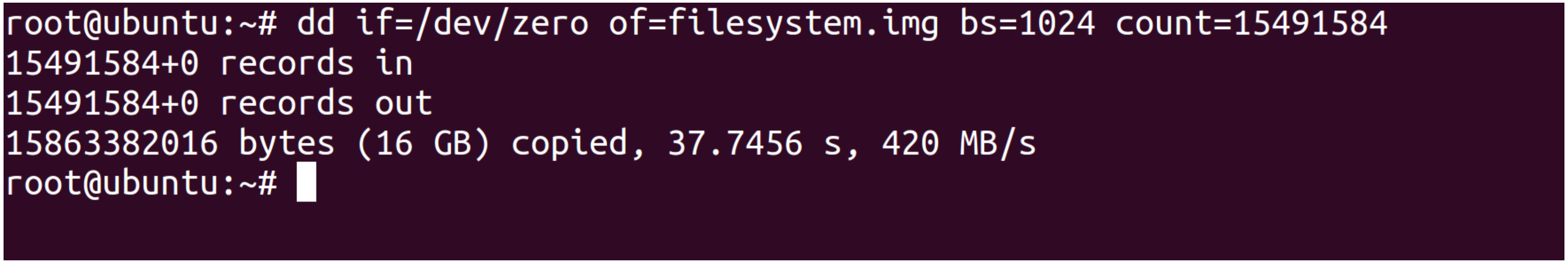
SD card的分割配置
(這張SD card是16GB)

```
// Change to root
$ sudo -s
$ mount /dev/sdd2 /mnt/mmc2
$
// 先產生一個與filesystem一樣大的image file
$ dd if=/dev/zero of=filesystem.img bs=1024
count=15491584
```

(這裡的值，會與你得SD card大小而有所不同，填入自己的值)

fdisk中，每個block為1024 bytes

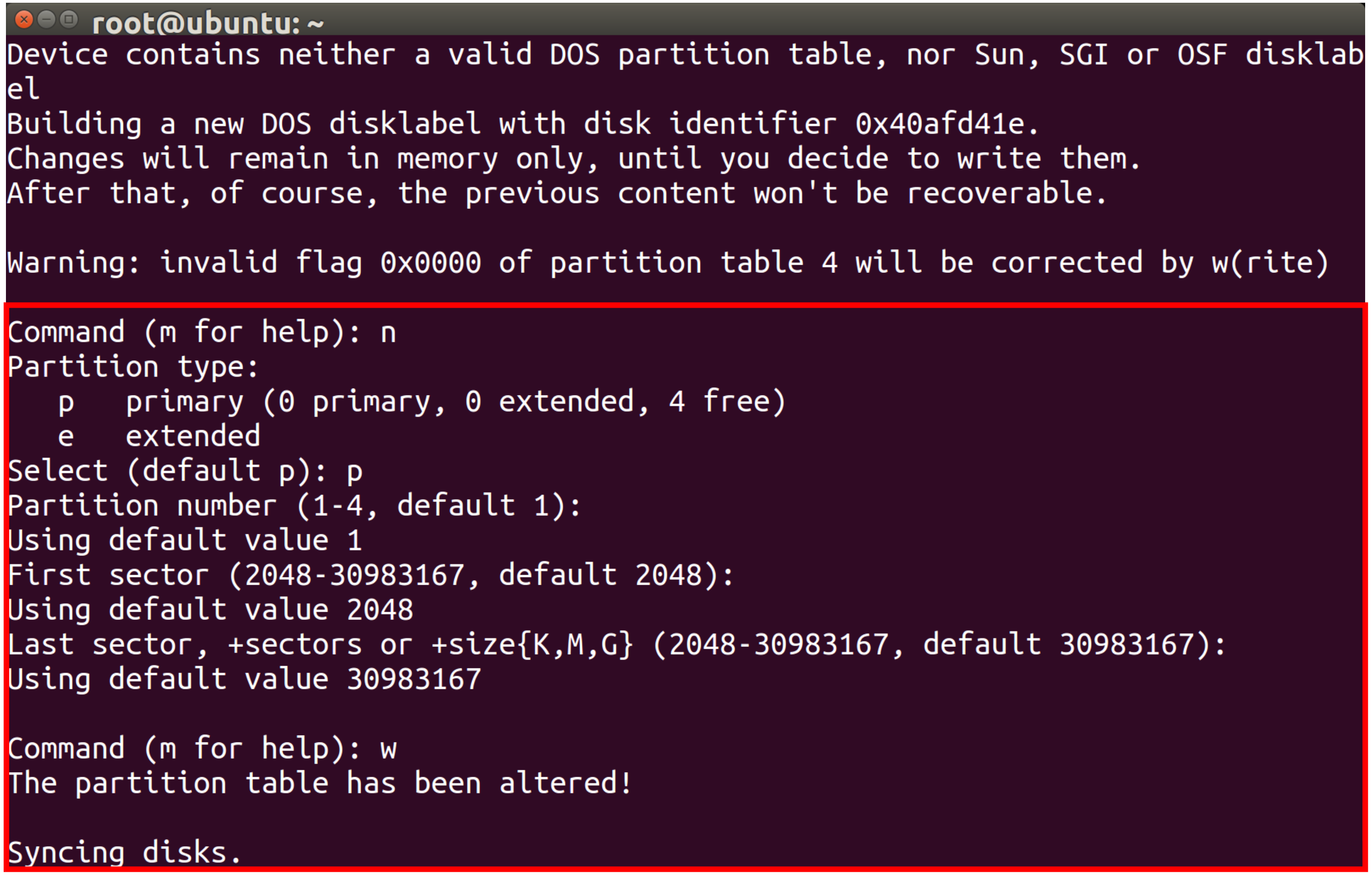
螢幕截圖: dd產生image file



```
root@ubuntu:~# dd if=/dev/zero of=filesystem.img bs=1024 count=15491584
15491584+0 records in
15491584+0 records out
15863382016 bytes (16 GB) copied, 37.7456 s, 420 MB/s
root@ubuntu:~# 
```

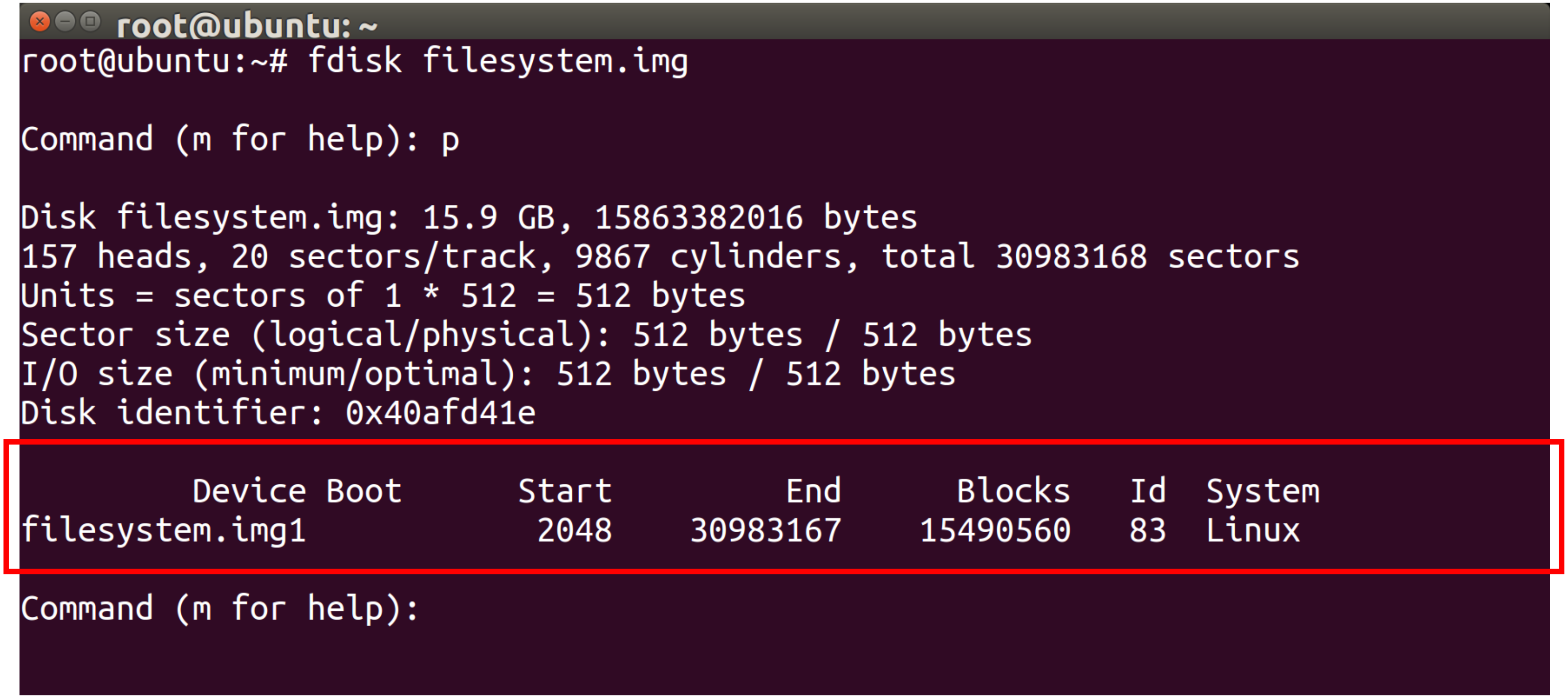
備份系統: rsync (2)

```
// 在image file裡create partition.
$ fdisk filesystem.img
```



```
Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklab
el
Building a new DOS disklabel with disk identifier 0x40afd41e.
Changes will remain in memory only, until you decide to write them.
After that, of course, the previous content won't be recoverable.

Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)

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):
Using default value 1
First sector (2048-30983167, default 2048):
Using default value 2048
Last sector, +sectors or +size{K,M,G} (2048-30983167, default 30983167):
Using default value 30983167

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

Syncing disks.
```

image file裡的partition狀態



```
root@ubuntu:~# fdisk filesystem.img

Command (m for help): p

Disk filesystem.img: 15.9 GB, 15863382016 bytes
157 heads, 20 sectors/track, 9867 cylinders, total 30983168 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 identifier: 0x40afd41e

         Device Boot      Start         End      Blocks   Id  System
filesystem.img1            2048    30983167    15490560   83  Linux

Command (m for help):
```

備份系統: rsync (3)

```
// 透過losetup連結image file中的partition。
// 建置pseudo device
$ losetup --offset $((2048*512)) \
          --sizelimit=$((30981120*512)) \
          /dev/loop0 filesystem.img
```

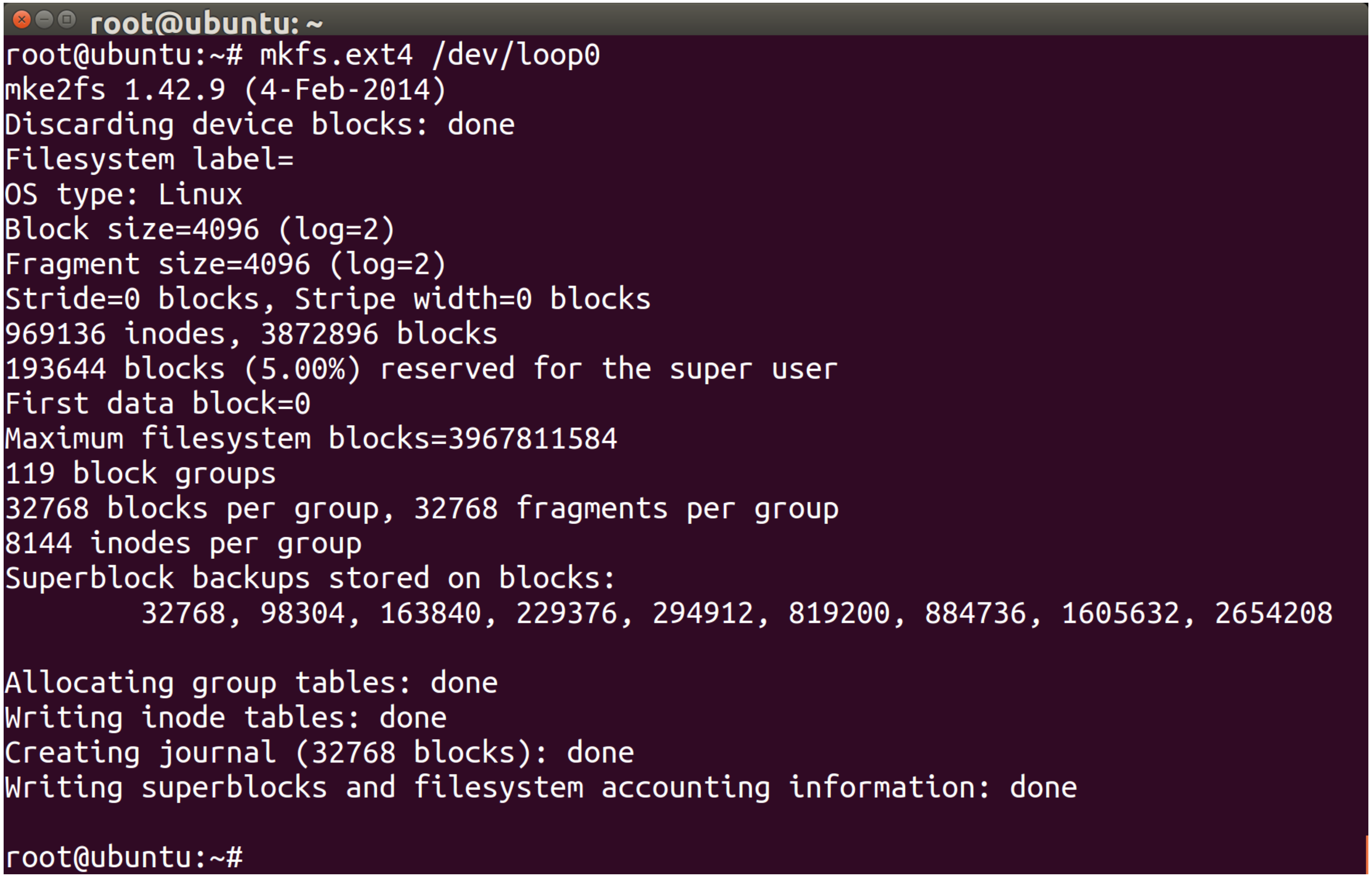
- --sizelimit是partition size
- 計算方式: **(End – Start + 1)*sector_size**

(30983167–2048+1)*512=30981120*512

(請參閱前一頁的資訊)

備份系統: rsync (4)

```
// format image file中的partition。
$ mkfs.ext4 /dev/loop0
```



root@ubuntu: ~

```
root@ubuntu:~# mkfs.ext4 /dev/loop0
mke2fs 1.42.9 (4-Feb-2014)
Discarding device blocks: done
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
969136 inodes, 3872896 blocks
193644 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=3967811584
119 block groups
32768 blocks per group, 32768 fragments per group
8144 inodes per group
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

root@ubuntu:~#
```

備份系統: rsync (5)

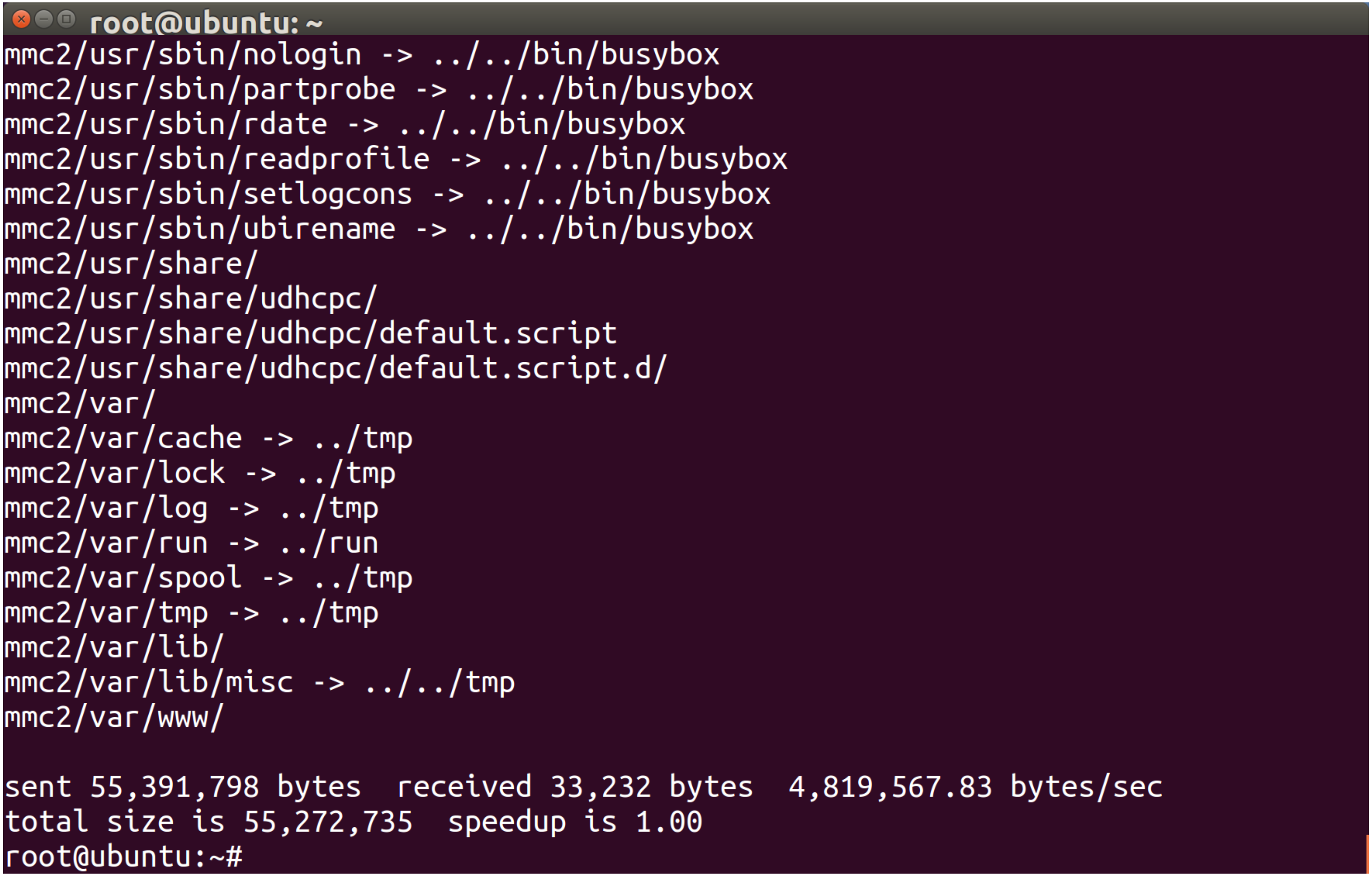
```
// Mount the partition
$ mkdir /mnt/sys_backup
$ mount /dev/loop0 /mnt/sys_backup
```

備份系統: rsync (6)

```
// Change to root
$ sudo -s
$ rsync  -axvH --delete /mnt/mmc2 \
                        /mnt/sys_backup/
```

- The option –x: “**Do not cross filesystem boundaries**” means “do not look inside mount points”.
- A boundary between filesystems is a mount point.
- Effectively, this means “**only act on the specified partition**”, except that not all filesystems are on a partition.

電腦截圖: rsync



```
mmc2/usr/sbin/nologin -> ../../bin/busybox
mmc2/usr/sbin/partprobe -> ../../bin/busybox
mmc2/usr/sbin/rdate -> ../../bin/busybox
mmc2/usr/sbin/readprofile -> ../../bin/busybox
mmc2/usr/sbin/setlogcons -> ../../bin/busybox
mmc2/usr/sbin/ubirename -> ../../bin/busybox
mmc2/usr/share/
mmc2/usr/share/udhcpc/
mmc2/usr/share/udhcpc/default.script
mmc2/usr/share/udhcpc/default.script.d/
mmc2/var/
mmc2/var/cache -> ../tmp
mmc2/var/lock -> ../tmp
mmc2/var/log -> ../tmp
mmc2/var/run -> ../run
mmc2/var/spool -> ../tmp
mmc2/var/tmp -> ../tmp
mmc2/var/lib/
mmc2/var/lib/misc -> ../../tmp
mmc2/var/www/

sent 55,391,798 bytes  received 33,232 bytes  4,819,567.83 bytes/sec
total size is 55,272,735  speedup is 1.00
root@ubuntu:~#
```

備份系統: rsync (7)

```
// 移除pseudo device
$ losetup -d /dev/loop0
```

Q&A (請於實驗報告裡回覆)

- What is loop device?

- 請說明下面rsync使用的各個參數的意義:

```
$ rsync  -axvH --delete /mnt/mmc2 \
/mnt/sys_backup/
```