

學號：405410117

姓名：郭紘安

Email：andy8766kuo@gmail.com

實驗名稱：Bootloader, Linux kernel, Root filesystem

實驗目的：

於Raspberry Pi3上安裝Linux作業系統。編譯Bootloader、Kernel Image與Root file system，將其存入儲存裝置中，將Raspberry Pi 3開機。

實驗步驟：

Part1: 取得 Buildroot source code

andy@ubuntu:~/disk\$ git clone git://git.busybox.net/buildroot

Part2: make config (Kernel image & Bootloader)

Step1:進入 buildroot 資料夾內

Step2:確認該版本 buildroot 是否支援 Raspberry Pi 3

```
andy@ubuntu:~/disk$ cd buildroot/  
andy@ubuntu:~/disk/buildroot$ ls configs/raspberry*  
configs/raspberrypi0_defconfig      configs/raspberrypi3_qt5we_defconfig  
configs/raspberrypi0w_defconfig     configs/raspberrypi4_64_defconfig  
configs/raspberrypi2_defconfig      configs/raspberrypi4_defconfig  
configs/raspberrypi3_64_defconfig   configs/raspberrypi_defconfig  
configs/raspberrypi3_defconfig
```

找到 configs/raspberrypi3_defconfig

```
andy@ubuntu:~/disk$ cd buildroot/  
andy@ubuntu:~/disk/buildroot$ ls configs/raspberry*  
configs/raspberrypi0_defconfig      configs/raspberrypi3_qt5we_defconfig  
configs/raspberrypi0w_defconfig     configs/raspberrypi4_64_defconfig  
configs/raspberrypi2_defconfig      configs/raspberrypi4_defconfig  
configs/raspberrypi3_64_defconfig   configs/raspberrypi_defconfig  
configs/raspberrypi3_defconfig
```

Step3:make config

andy@ubuntu:~/disk/buildroot\$ make raspberrypi3_defconfig

Step4: 藉由 buildroot 提供的 makefile 下載所有需要的套件

andy@ubuntu:~/disk/buildroot\$ make source

Part3: make Kconfig Root filesystem

buildroot 有支援 Kconfig 可供操作：

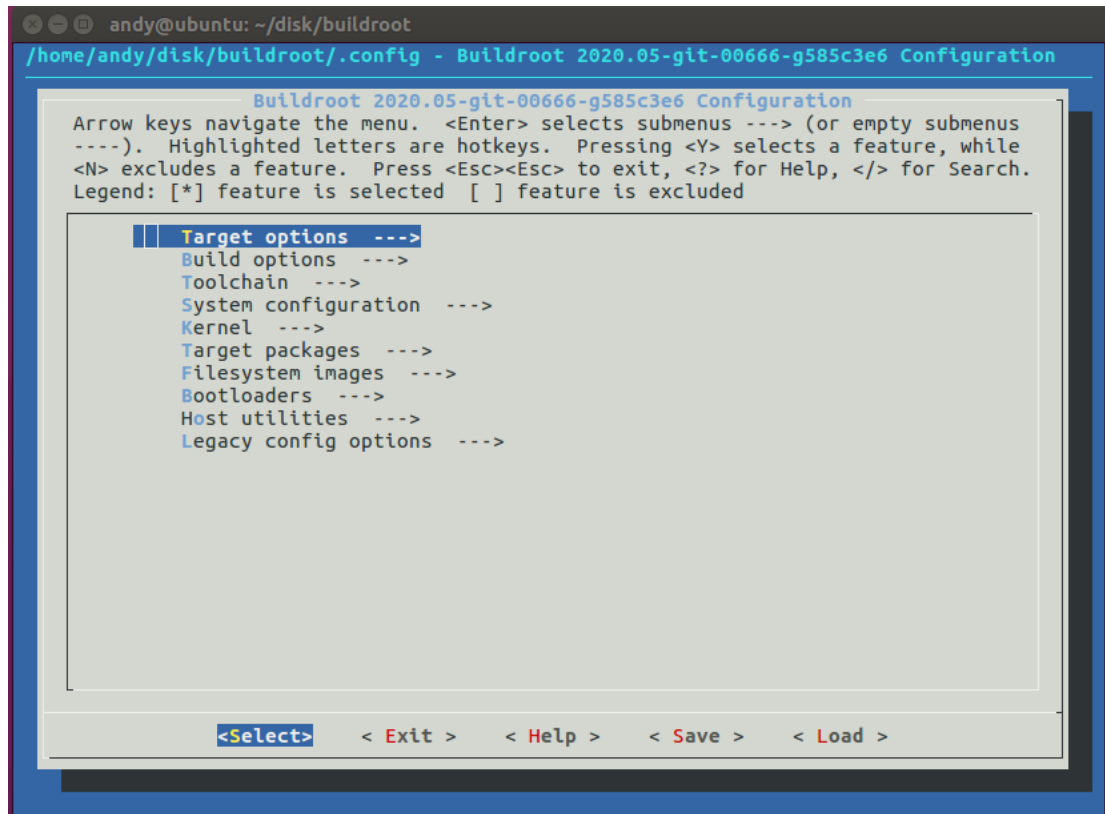
學號：405410117

姓名：郭紘安

Email：andy8766kuo@gmail.com

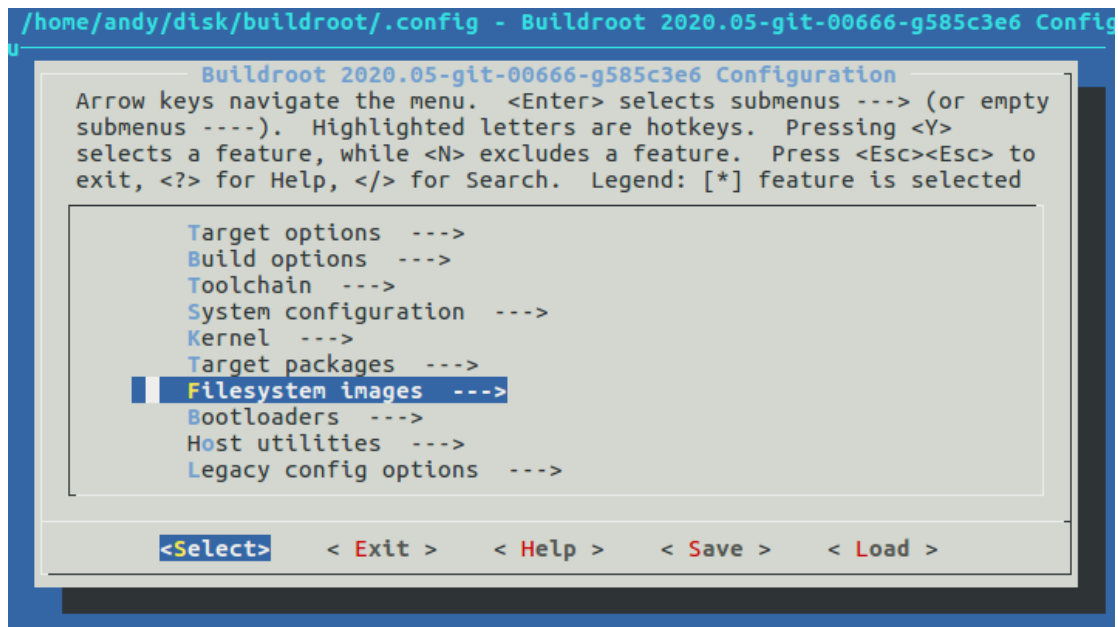
Step1: make Kconfig

andy@ubuntu:~/disk/buildroot\$ make menuconfig



Step2: 設定 menuconfig 以編譯並打包出 root filesystem

1. Filesystem images



學號：405410117

姓名：郭紘安

Email：andy8766kuo@gmail.com

2.tar the root filesystem

```
/home/andy/disk/buildroot/.config - Buildroot 2020.05-git-00666-g585c3e6 Config
u> Filesystem images

Filesystem images
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty
submenus ----). Highlighted letters are hotkeys. Pressing <Y>
selects a feature, while <N> excludes a feature. Press <Esc><Esc> to
exit, <?> for Help, </> for Search. Legend: [*] feature is selected
^(-)
(5) reserved blocks percentage
(-0 ^64bit) additional mke2fs options
    Compression method (no compression) --->
[ ] f2fs root filesystem
[ ] initial RAM filesystem linked into linux kernel
[ ] jffs2 root filesystem
[ ] romfs root filesystem
[ ] squashfs root filesystem
[*] tar the root filesystem
    Compression method (gzip) --->
+ (+)

<Select>  < Exit >  < Help >  < Save >  < Load >
```

3.Compression method() (gzip 或 bzip2 皆可)

```
/home/andy/disk/buildroot/.config - Buildroot 2020.05-git-00666-g585c3e6 Config
u> Filesystem images

Filesystem images
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty
submenus ----). Highlighted letters are hotkeys. Pressing <Y>
selects a feature, while <N> excludes a feature. Press <Esc><Esc> to
exit, <?> for Help, </> for Search. Legend: [*] feature is selected
^(-)
(-0 ^64bit) additional mke2fs options
    Compression method (no compression) --->
[ ] f2fs root filesystem
[ ] initial RAM filesystem linked into linux kernel
[ ] jffs2 root filesystem
[ ] romfs root filesystem
[ ] squashfs root filesystem
[*] tar the root filesystem
    Compression method (gzip) --->
    () other random options to pass to tar
+ (+)

<Select>  < Exit >  < Help >  < Save >  < Load >
```

Step3: Save & Exit 儲存 Kconfig 之設定

Part4: 編譯 Buildroot

設定完畢所有設定值後，使用make指令編譯出所有目標系統檔案

andy@ubuntu:~/disk/buildroot\$ make -j4

PS: make -j4 (run with 4 cores)

學號：405410117

姓名：郭紘安

Email：andy8766kuo@gmail.com

Part5: 切割SD card

Step1: 進入output/images路徑

```
andy@ubuntu:~/disk$ cd buildroot/output/images
```

Step2: 掛載由 buildroot 產生的 sdcard.img 來切割 sdcard

PS: 可由下列指令觀察sdcard編號:

df -a 若sdcard為/dev/sdd，則sdx為sdd

Part6: 將建置好的系統檔案放入切割過的SD card內

```
andy@ubuntu:~/disk/buildroot/output/images$ sudo dd if=sdcard.img of=/dev/sdc
311297+0 records in
311297+0 records out
159384064 bytes (159 MB, 152 MiB) copied, 131.931 s, 1.2 MB/s
```

Step1: 建立資料夾以便掛載使用

```
andy@ubuntu:~/disk$ mkdir mmc1 mmc2
```

```
/dev/sdd2    114901    60449    45851    57% /media/andy/db1fc63f-5666-4e5e-8163-3a8c29014a20
/dev/sdd1    32686     8834    23852    28% /media/andy/0A85-C5A8
```

Step2: 分別掛載 mmc1 與 mmc2 至 BOOT 與 FILESYSTEM

Step3: 將系統檔案放入 BOOT

```
andy@ubuntu:~/disk$ cd buildroot/output/images/
andy@ubuntu:~/disk/buildroot/output/images$ sudo cp -rf
bcm2710-rpi-3-b.dtb zImage /home/andy/disk/mmc1/
andy@ubuntu:~/disk/buildroot/output/images$ sudo cp -rf
rpi-firmware/* /home/andy/disk/mmc1/
```

```
andy@ubuntu:~/disk$ sudo mount /dev/sdd1 ~/disk/mmc1
andy@ubuntu:~/disk$ sudo mount /dev/sdd2 ~/disk/mmc2
```

Step4: 將 root filesystem.gz/bz2 放入 FILESYSTEM 並解壓

縮

```
andy@ubuntu:~/disk/buildroot/output/images$ sudo cp -rf
/home/andy/disk/buildroot/output/images/rootfs.tar.gz
/home/andy/disk/mmc2/
andy@ubuntu:~/disk$ cd mmc2
andy@ubuntu:~/disk/mmc2$ sudo tar -zxvf rootfs.tar.gz
andy@ubuntu:~/disk/mmc2$ sudo rm rootfs.tar.gz
```

學號：405410117

姓名：郭紘安

Email：andy8766kuo@gmail.com

Part7：開機測試

此最簡易Linux系統只有root，login處鍵入root即可取得root權限
即可看到整個 root filesystem

```
Welcome to Buildroot
buildroot login: root
# [ 32.738297] random: crng init done
cd ..
# ls
bin          lib          lost+found  opt          run          tmp
dev          lib32        media       proc          sbin         usr
etc          linuxrc      mnt         root         sys          var
#
```

問題與討論：

1. 請問下面firmware目錄下的各個檔案的用途分別是什麼？

output/rpi-firmware/

● bootcode.bin

- GPU 載入 SD 卡中 boot 目錄中的第二階段開機程序 bootcode.bin 到 L2 Cache 中，並執行 bootcode.bin.
- bootcode.bin 啟動了 SDRAM，並載入第三階段開機程序 loader.bin 到 RAM 中，並執行 loader.bin.

● cmdline.txt

- 開機透過 bootloader傳送 載入 cmdline.txt 給 kernel.
- 啟動時參照 提供給kernel啟動參數.
- 通常依版本，CPU等，內容不同.

● cofing.txt

- 由於樹莓派並沒有傳統意義上的 BIOS，所以現在各種系統配置參數通常被存在 config.txt 這個文本文件中.
- 樹莓派的 config.txt 文件會在 ARM 內核初始化之前被 GPU 讀取.
- 這個文件存在引導分區上的.對於 Linux，路徑通常是 /boot/config.txt

● fixup.dat

- Is used to configure the SDRAM partition between the GPU

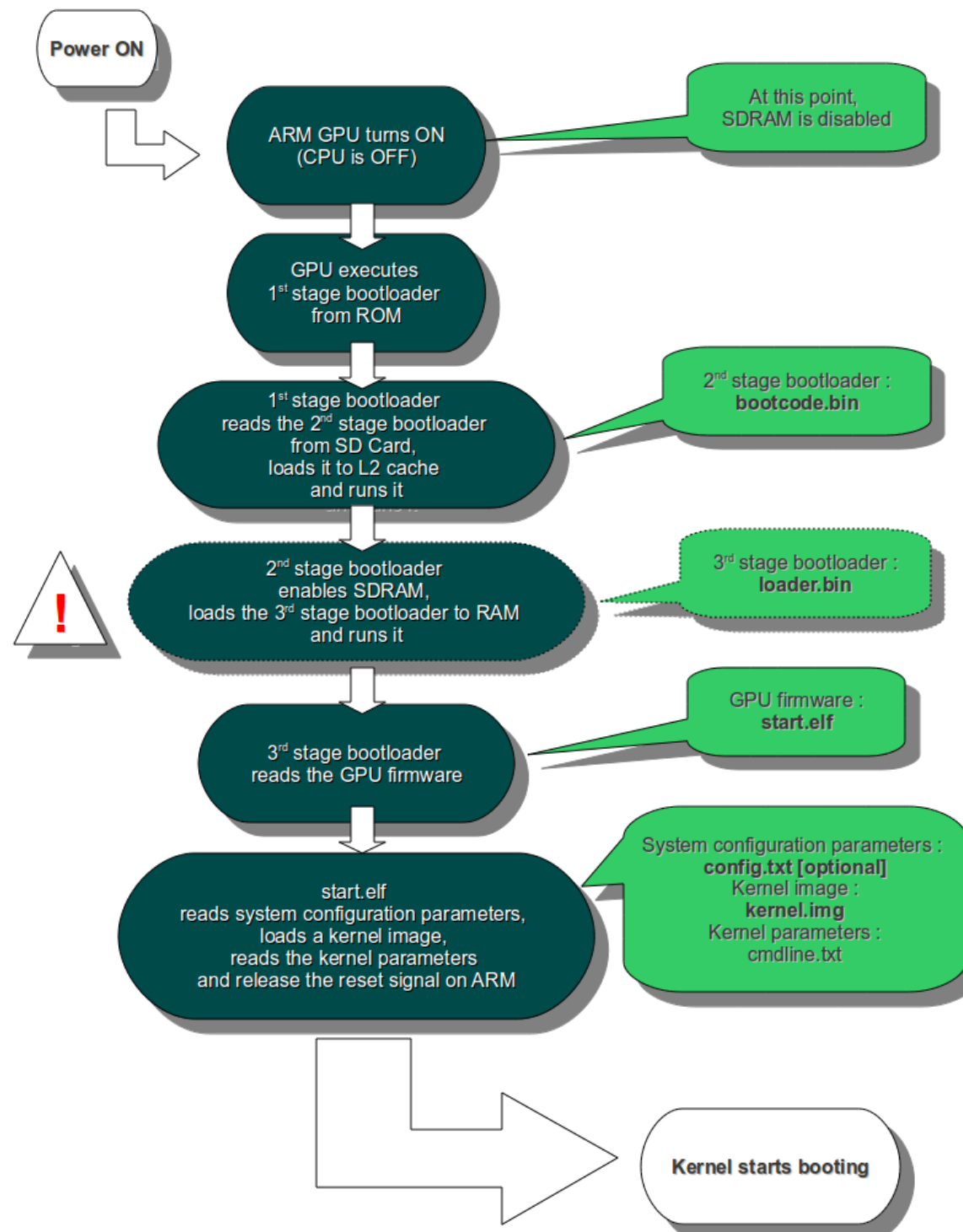
學號：405410117

姓名：郭紘安

Email：andy8766kuo@gmail.com

● start.elf

- loader.bin 載入 GPU 的韌體 start.elf
- start.elf 讀取設定檔 config.txt 及 cmdline.txt，並載入最重要的 Linux 核心 kernel.img。
- start.elf 在載入 kernel.img 後啟動了 CPU。

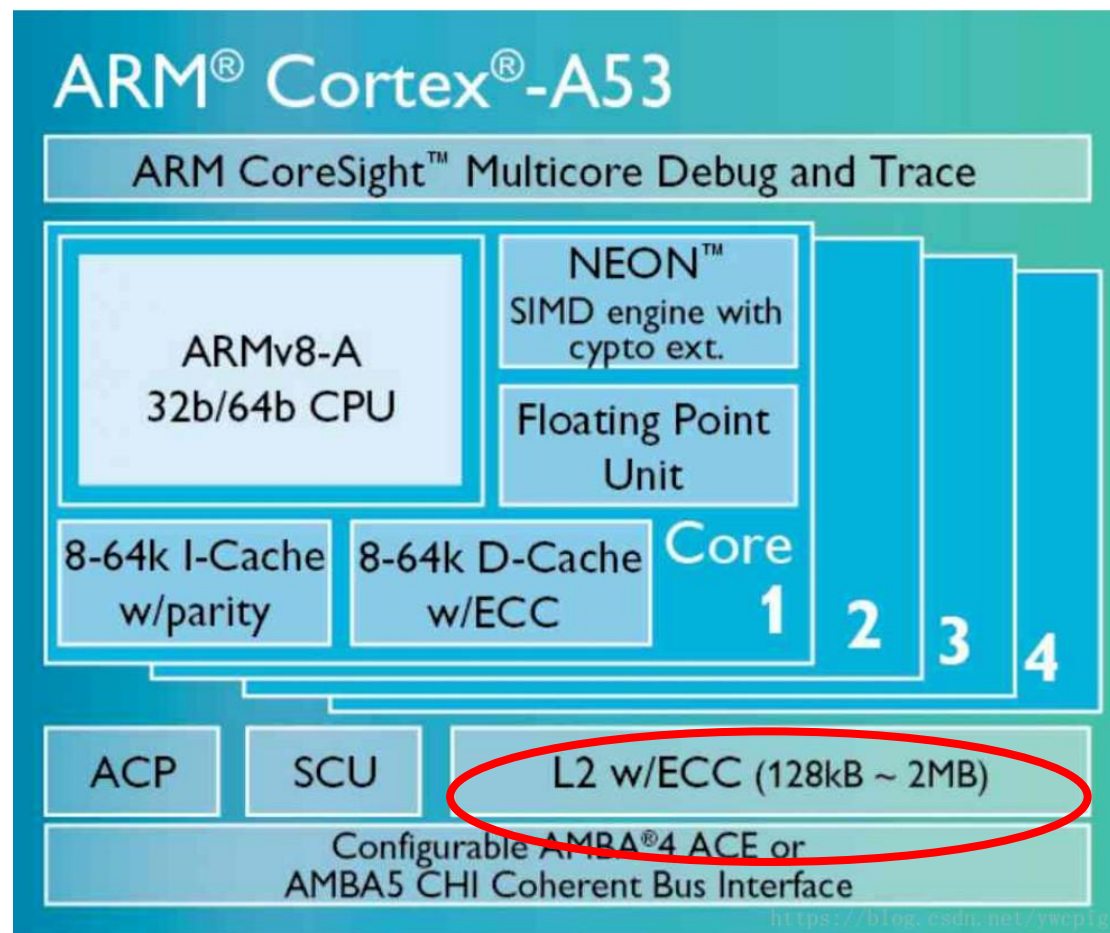


學號：405410117

姓名：郭紘安

Email：andy8766kuo@gmail.com

L2 Cache：



2. "cmdline.txt"的內容是什麼，相關參數的意義是什麼？

- **dwc_otg.lpm_enable=0**

dwc_otg 指的是 HS OTG USB Controller driver 相關參數。例如 lpm_enable=0 即預設 USB HS OTG device 的 Link Power management 為關閉。

- **console=ttyAMA0,115200**

RPi 用 ttyAMA0 做為 UART port (GPIO 8 and 10) 的終端定義，所以這裡表示在 UART port 上開 console。如果後續需要使用 UART 為其他用途，則需要把這部分拿掉，以免干擾。

- **kgdboc=ttyAMA0,115200**

在 UART port 上開 KGDB debug port。如果後續需要使用 UART 為其他用途，則需要把這部分拿掉，以免干擾。

- **console=tty1**

在 tty1 上開 console。

學號：405410117

姓名：郭紘安

Email：andy8766kuo@gmail.com

- **root=/dev/mmcblk0p2**

定義 rootfs path. 如果你希望把 rootfs 改放到 USB disk 或者 NFS 上,這裡可以修改.

- **rootfstype=ext4**

rootfs 格式.

- **elevator=deadline**

kernel 預設的 I/O scheduling 的方法(例如 noop, deadline ,)這個比較明顯的影響應該是對於 storage IO 的效能.

- **rootwait**

kernel 要先初始化 MMC storage module 後才能完成 mount MMC,而預設的 rootfs 又是放在 MMC 上,所以利用此參數要求 kernel 在載入全部的 module 後再開始 mount and read rootfs.

心得：

這次的實驗讓我了解樹梅派整個開機的流程,以及如何創建一個 SD 卡來讓一個樹梅派能夠順利地啟動運作.