

學號：404410030

姓名：鄭光宇

Email：jengku@gmail.com

Lab3: Bootloader, Linux kernel, Root filesystem

實驗目的：

實際編譯、安裝一次簡易的 Linux 作業系統。

將 Bootloader、Kernel Image、Root filesystem 編譯好安裝到 Raspberry Pi 3 上。

實驗步驟：

接續 Lab.1，將編譯好的 cross-compiler 準備好，之後會使用 Lab.1 的 cross-compiler 編譯 Linux 作業系統。確認 cross-compiler 在 PATH 中。

先下載 buildroot，這是一個讓我們很方便編譯、安裝 Embedding Linux 的工具。

```
git clone git://git.busybox.net/buildroot
```

設定 Kernel 設定檔，開始編譯：

```
cd buildroot
make raspberrypi3_defconfig
make source
make menuconfig
make -j # 如果記憶體不夠的話，就把 -j 參數拿掉
```

編譯好後，在 `buildroot/output/images` 路徑下，應該會有 `sdcard.img` 檔案，這是編譯、打包好的整個作業系統。

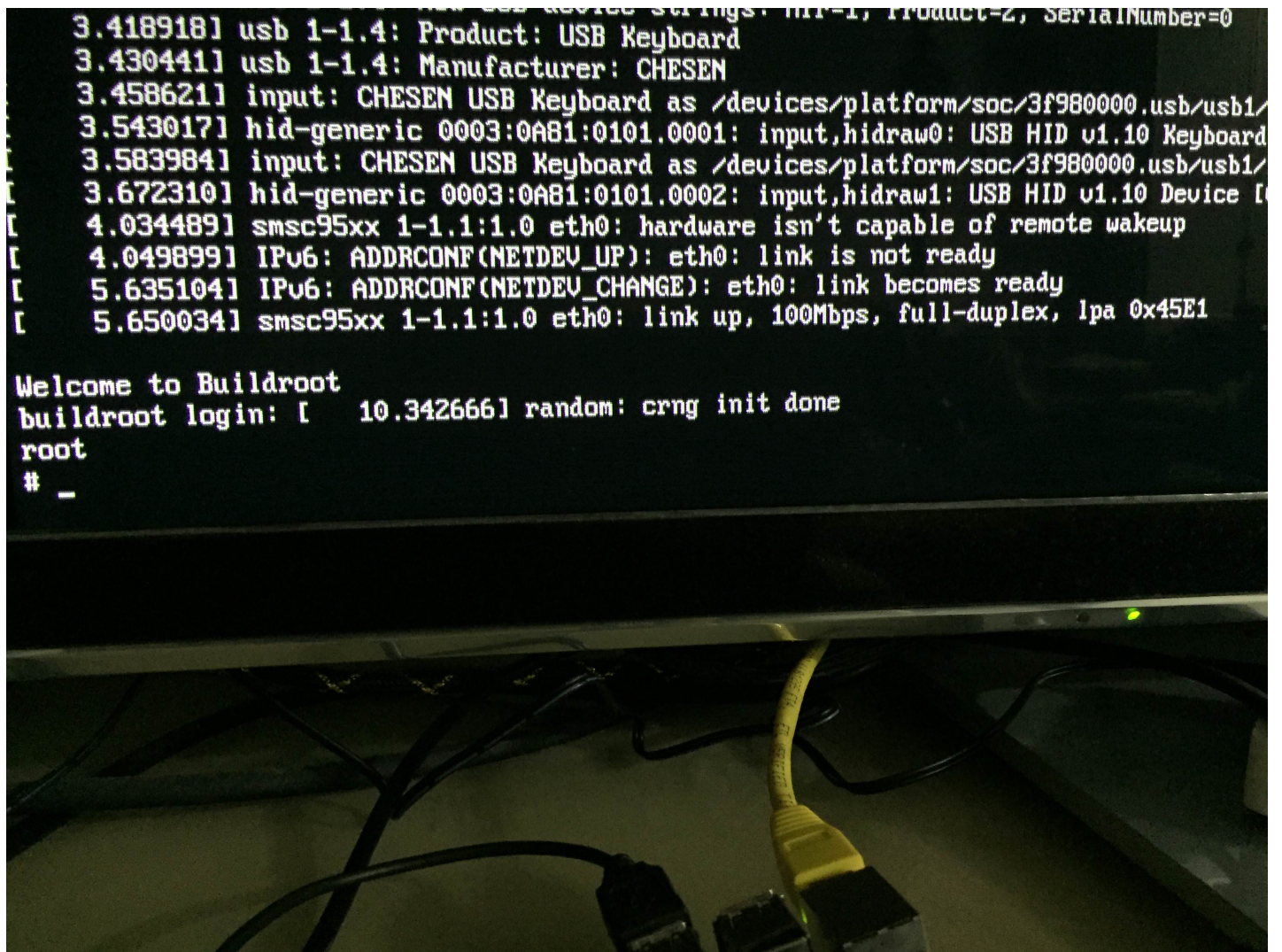
切換成 `root` 使用者，使用 `pv` 指令，一邊快速地將作業系統寫入 SD 卡，一邊監看寫入進度：

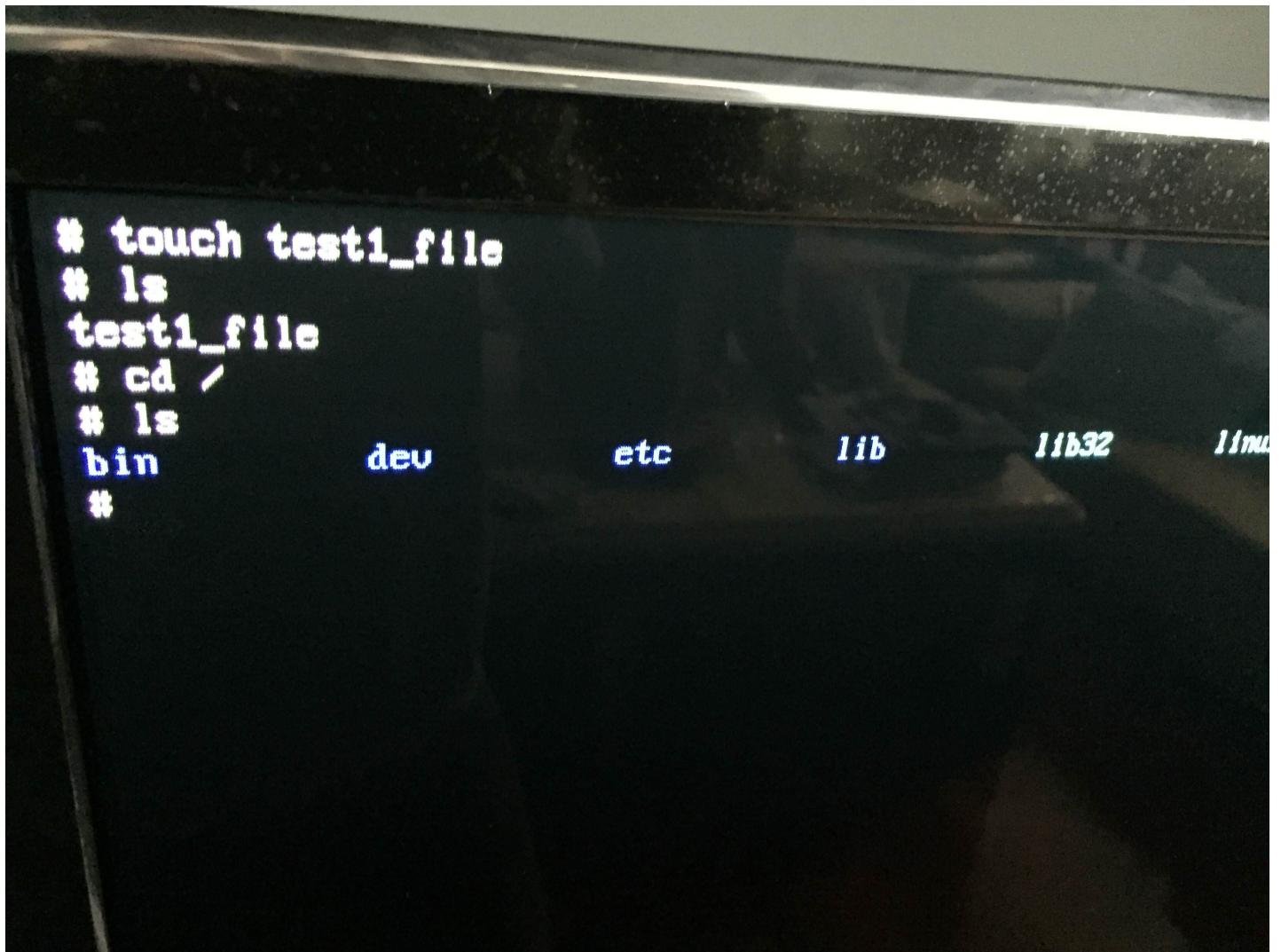
```
sudo su
pv < sdcard.img > /dev/your_sd_card
```

也可以使用 `dd` 指令，但是就看不到進度條：

```
sudo dd if=sdcard.img of=/dev/your_sd_card
```

寫入完成後，插入 SD 卡到 Raspberry Pi 上。開機測試。





Q&A

bootcode.bin:

Raspberry Pi 上的第二階段 bootloader，Raspberry Pi 上的 GPU 會去讀這個檔案並執行它。bootcode.bin 的主要工作是用來啟動 SDRAM，還有讓 GPU 之後可以去讀取、執行 start.elf 檔。

cmdline.txt:

Kernel 啟動時的參數。可以指定 root 位置、root 檔案系統種類 (ext4 之類的)、IP 之類的。

config.txt:

這是 Raspberry Pi 的系統設定參數，作用有點像 PC 上的 BIOS。裡面可以指定 CPU/GPU 記憶體分配、設定 HDMI 輸出、超頻等。

fixup.dat:

設定 GPU 與 CPU 之間，SDRAM 的分配。

start.elf:

用來讀取 kernel image (kernel.img) 、 config.txt 、 cmdline.txt ，並把 kernel 載入 memory ，最後啟動 ARM 處理器。最後讓 Linux 接管後面的工作。

cmdline.txt 的內容、參數：

內容：

root=/dev/mmcblk0p2 rootwait console=tty1 console=ttyAMA0,115200

參數：

root: 指定 root filesystem 的位置

rootwait: 要求 kernel 先等待所有的 module 載入後，再去 mount/read rootfs 。主要用途就是等待 MMC storage module 啟動、rootfs 出現後，kernel 才去 mount 它。

console: console 的位置，使用 tty 代表使用虛擬主控台。

可能會用得上的其他參數：

rootfstype: 指定 root filesystem 的種類 (例如: ext4)

dwc_otg.speed: 設為 1 指定 USB 使用 USB v1 速度

ip: 指定靜態 IP

smc95xx.macaddr: 指定 Pi 的網卡 MAC address

elevator: kernel 預設的 I/O scheduler 可能有 noop, deadline, cfq 等。