

## HW1 M11315051 林子新

### Hardware :

1. CPU : i5-12400
2. GPU : RTX4070
3. RAM : 32GB (partition 512 MB for Virtual Machine)
4. SSDd : 1TB (partition 20GB for Virtual Machine)

### Software :

1. MobaXterm WSL(用來製作 Linux OS 的環境)
2. VMware(虛擬環境)
3. BusyBox(用來創建 rootfs)
4. GRUB(創建引導映像)

### 實作步驟：

1. MobaXterm 安裝 參考網站：[\[20211203\] OpenROAD-flow-scripts Installation \(youtube.com\)](https://www.youtube.com/watch?v=202112031)
2. 在 WSL 上面進行 Linux OS 的製作，製作 bzImage 作為 kernel file 在虛擬機中使用 (Ubuntu 環境)
  - (1) sudo apt-get update 更新包 apt-get

```
xup6y7vup@DESKTOP-JHSROAD:~/Linux$ sudo apt-get update
[sudo] password for xup6y7vup:
Hit:1 http://archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [129 kB]
```

- (2) sudo apt-get install build-essential 安裝 GCC 編譯器
- (3) sudo apt-get install libncurses-dev bison flex libssl-dev libelf-dev
- (4) git clone git://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux-2.6.git

```
xup6y7vup@DESKTOP-JHSROAD:~/Linux$ git clone git://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux-2.6.git
Cloning into 'linux-2.6'...
remote: Enumerating objects: 10352430, done.
remote: Counting objects: 100% (111/111), done.
remote: Compressing objects: 100% (87/87), done.
remote: Total 10352430 (delta 51), reused 47 (delta 24), pack-reused 10352319
```

- (5) cd linux-2.6
- (6) make menuconfig (配置 kernel，我是使用 default。)

```
xup6y7vup@DESKTOP-JHSROAD:~/Linux/linux-2.6$ make menuconfig
HOSTCC scripts/basic/fixdep
HOSTCC scripts/kconfig/mconf.o
HOSTCC scripts/kconfig/Lxdialog/checklist.o
```

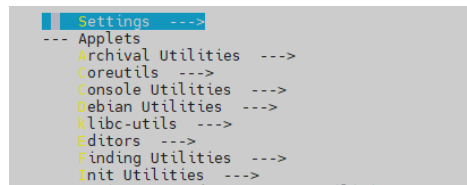
```
General setup --->
[*] 64-bit kernel (NEW)
Processor type and features --->
[*] Mitigations for CPU vulnerabilities (NEW) --->
Power management and ACPI options --->
Bus options (PCI etc.) --->
Binary Emulations --->
[*] Virtualization (NEW) --->
```

會看到像是這樣的畫面

- (7) make -> 遇到 error -> error 原因：系統中缺少 bc 工具
- (8) sudo apt install bc
- (9) make -> kernel 編譯完成
- (10) make modules\_install (將已編譯的內核模塊安裝到系統的 /lib/modules/)
- (11) sudo make install
- (12) cd arch/x86/boot/找到 bzImage (要在虛擬機中指定 bzImage 作為 kernel file -> 之後要將 bzImage 轉為 vmlinuz，如果沒有可以用 make bzImage 指令創建 bzImage )

### 使用 BusyBox 創建 rootfs 用來當作根文件系統 (root file system)

- (13) wget <https://busybox.net/downloads/busybox-1.32.1.tar.bz2>
- (14) tar xjf busybox-1.32.1.tar.bz2
- (15) cd busybox-1.32.1
- (16) make menuconfig (如右圖一)
- (17) make
- (18) make install CONFIG\_PREFIX=\$HOME/Linux/linux-2.6/initrd/rootfs(安裝 BusyBox，並指定輸出目錄)



圖一

- (19) cd initrd/rootfs ( 如右圖二，init 是後來創建的 )
- 創建 init 當作 Linux 系統啟動時第一個被內核執行的進程，負責初始化系統



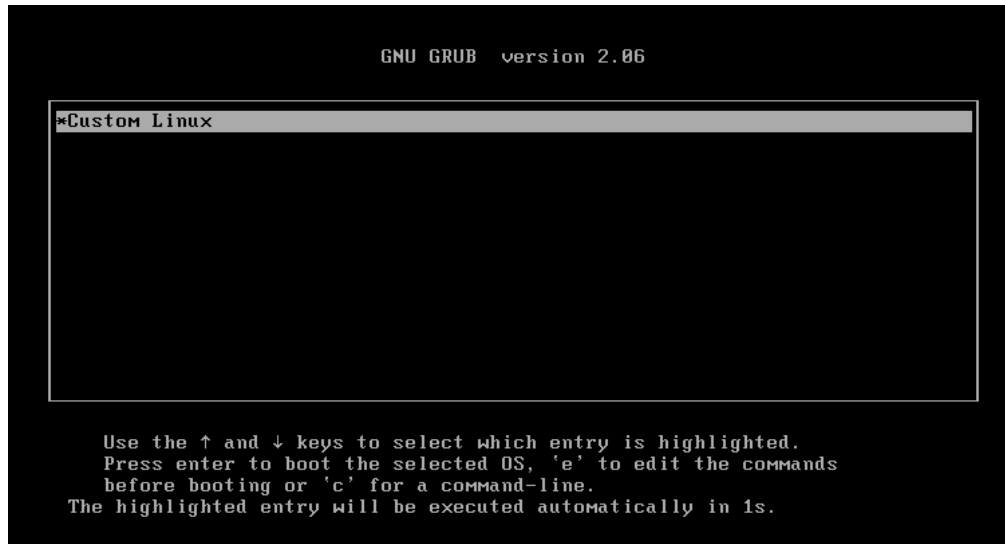
圖二

- (20) vi init ( 輸入 i 進入編輯模式, esc :wa 儲存 :qa 可以離開 )
- (21) #!/bin/sh
  - mount -t proc none /proc ( 將 proc 文件系統掛載到 /proc 目錄。proc 文件系統是用來存放系統進程和內核信息的虛擬文件系統。 )
  - mount -t sysfs none /sys ( 將 sysfs 文件系統掛載到 /sys 目錄。sysfs 文件系統提供了與硬體設備和內核信息相關的虛擬文件。 )
  - mdev -s (使用 mdev 來管理設備檔案，並掃描所有設備，建立相應的 /dev 目錄中的設備文件。)
  - /bin/sh (指向 BusyBox 提供的內建 Shell。)
- (22) chmod +x rootfs/init (確保 init 文件具有可執行的權限)
- (23) find . | cpio -o --format=newc | gzip > ../initrd.img (將根文件系統目錄打包為 initrd.img)

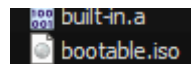
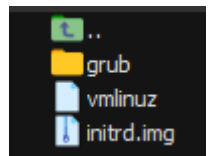
### 使用 GRUB 創建引導映像

- (24) sudo apt install grub-pc-bin grub-common
- (25) mkdir -p iso/boot/grub
- (26) nano grub.cfg ( 編輯 grub.cfg 文件 )
  - set timeout=10
  - set default=0


```
menuentry "Custom Linux" {
    linux /boot/vmlinuz root=/dev/ram0 init=/init (讓 GRUB 指向內核)
    initrd /root/initrd.img (讓 GRUB 指向根文件系統)
} ( 進入系統前會先出現下圖這個畫面 )
```



- (27) `cp arch/x86/boot/bzImage iso/boot/vmlinuz` ( 複製 bzImage 到你的 iso/boot/ 目錄中 )
- (28) `cp initrd.img iso/boot/` ( 將 initrd.img 複製到 iso/boot/ ，如右圖)
- (29) `grub-mkrescue -o bootable.iso iso` (生成 iso 檔，因為我們 VMware 要使用 iso 檔啟動虛擬器)
- (30) 將 bootable.iso 從 WSL 下載到本機



### 3. 在 VMware 上執行 Linux 系統

- (1) VMware 安裝 參考網站：[Windows/Mac 免費虛擬機 VMware 下載教學，雙版本差異與支援比較 - 瘋先生 \(mrmad.com.tw\)](http://mrmad.com.tw)
- (2) 開啟後點選 Create a New Virtual Machine -> typical -> I will install operating system later -> Linux -> Other Linux 2.6.x kernel
- (3) Disk size 我設定 20GB -> finish
- (4) 點選  Edit virtual machine settings -> 在 CD/DVD 選擇 use ISO image file -> 選取 bootable.iso
- (5) 執行 虛擬環境 如果沒有 error 就可以看到 Linux 系統

```
[ 1.384716] usb 2-1: New USB device strings: Mfr=1, Product=2, SerialNumber=0
[ 1.385576] usb 2-1: Product: VMware Virtual USB Mouse
[ 1.386286] usb 2-1: Manufacturer: VMware
[ 1.314220] input: VMware VMware Virtual USB Mouse as /devices/pci0000:00/000
0:00:11.0/0000:02:00.0/usb2/2-1/2-1:1.0/0003:0E0F:0003.0001/input/input4
[ 1.316489] hid-generic 0003:0E0F:0003.0001: input,hidraw0: USB HID v1.10 Mou
se [VMware VMware Virtual USB Mouse] on usb-0000:02:00.0-1/input0
[ 1.432963] usb 2-2: new full-speed USB device number 3 using uhci_hcd
[ 1.574166] usb 2-2: New USB device found, idVendor=0e0f, idProduct=0002, bcd
Device= 1.00
[ 1.574685] usb 2-2: New USB device strings: Mfr=1, Product=2, SerialNumber=0
[ 1.575189] usb 2-2: Product: VMware Virtual USB Hub
[ 1.575450] usb 2-2: Manufacturer: VMware, Inc.
[ 1.580384] hub 2-2:1.0: USB hub found
[ 1.582437] hub 2-2:1.0: 7 ports detected
/ # uname -r
6.11.0-rc7-00020-g8d8d276ba2fb
/ # ls
bin      etc      init     proc     sbin     usr
dev      hello.txt linuxrc  root     sys
/ # cat hellot
cat: can't open 'hellot': No such file or directory
/ # cat hello.txt
Hello world
/ # -
```

我們可以對上述系統進行簡單的操作，上圖先用 `uname -r` 檢視 kernel version，然後 `ls` 列出系統 file，再用 `cat hello.txt` 執行 `hello.txt`，那可以看到 Hello world

```
/ # cat /proc/cpuinfo
processor       : 0
vendor_id     : GenuineIntel
cpu family    : 6
model         : 151
model name    : 12th Gen Intel(R) Core(TM) i5-12400
stepping      : 5
microcode     : 0xffffffff
cpu MHz       : 2495.999
cache size    : 18432 KB
```

用 `cat /proc/cpuinfo` 可以列出現在 cpu 的資訊

```
/ # cat /proc/meminfo
MemTotal:      395696 kB
MemFree:       366288 kB
MemAvailable:  363384 kB
Buffers:        0 kB
Cached:        4124 kB
SwapCached:    0 kB
Active:        8 kB
```

`cat /proc/meminfo` 可以列出現在 mem 的資訊 (如果資訊太多，可以用 Shift 再按 Page Up 或 Page Down 可以滾動。)

這樣就有了最簡易的 Linux 系統了!!

## 實作過程問題 (由處理花費的時間排序):

### 1. Init 無法被 VMware 辨識

```
[ 1.104970] x86/mm: Checking user space page tables
[ 1.110027] x86/mm: Checked W+X mappings: passed, no W+X pages found.
[ 1.110568] Run /init as init process
[ 1.110834] Failed to execute /init (error -2)
[ 1.111092] Run /init as init process
[ 1.111327] Kernel panic - not syncing: Requested init /init failed (error -2)
).
[ 1.111744] CPU: 0 UID: 0 PID: 1 COMM: swapper/0 Not tainted 6.11.0-rc7-00020
-g8d8d276ba2fb #2
[ 1.112219] Hardware name: VMware, Inc. VMware Virtual Platform/440BX Desktop
Reference Platform, BIOS 6.00 11/12/2020
[ 1.112794] Call Trace:
[ 1.112952] <TASK>
[ 1.113092] panic+0x31d/0x350
[ 1.113406] ? __pfx_kernel_init+0x10/0x10
[ 1.113666] kernel_init+0xc8/0x1c0
[ 1.113811] ret_from_fork+0x2c/0x50
[ 1.114107] ? __pfx_kernel_init+0x10/0x10
[ 1.114372] ret_from_fork_asm+0x1a/0x30
[ 1.114609] </TASK>
[ 1.114815] Kernel Offset: 0x5600000 from 0xffffffff81000000 (relocation rang
e: 0xffffffff80000000-0xffffffffbfffffff)
[ 1.115457] ---[ end Kernel panic - not syncing: Requested init /init failed
(error -2). ]---
```

我一開始打包好 iso 檔之後，init 一直無法被 VMware 認到，然後就一直跑出上述 error，所以我就

- (1) 檢查 init 文件的可執行權限
- (2) 檢查 initrd 文件的打包過程 (解壓縮確認包含了 init 文件)
- (3) 檢查 GRUB 是否正確地指向了內核和 init 文件。
- (4) 檢查 iso 檔裡面有沒有 initrd 跟 vmlinuz 文件
- (5) 檢查完後重新打包 rootfs 然後重新包成 iso 檔
- (6) 重新啟動虛擬機
- (7) 出現同樣的 error (基本上在這個過程重複了 2、3 次)

解決方法：[embedded linux - Kernel Panic : No working init found. Try passing init= option to kernel - Stack Overflow](#)

解決描述：按照下面的描述，重新打包就辨識到 init 了，可喜可賀。

In my case, the error was caused because my BusyBox binary was **dynamically linked**. To resolve this, I simply made it statically linked by doing the following in `make menuconfig` when making BusyBox:

Navigate to "**Settings**" -> "**Build Options**" -> "**Build static binary (no shared libs)**" and press **Y**.

After `make` and repacking it into initramfs, there was no more kernel panic.

### 2. 打不開/dev/tty

```
can't open /dev/tty2: No such file or directory
can't open /dev/tty3: No such file or directory
can't open /dev/tty4: No such file or directory
```

這個原因是因為一開始 init 沒有用 `mdev -s` 這個指令，使用 `mdev` 來管理設備檔案，並掃描所有設備，建立相應的 `/dev` 目錄中的設備文件。所以虛擬終端設備 (tty) 不存在，理所當然找不到檔案，後來在 init 中加入 `mdev -s`，就解決了

## HW comment

這次從零做 Linux OS 的 HW，我學到很多，從一開始毫無頭緒，到後面開始了解 HW 到底想要我們做什麼也是花了一段時間，這次安裝的系統真的是最簡易的系統，有很多現在很方便的指令都不能使用，那過程中 Chat gpt 幫我釐清了很多的問題，雖然在 Init 無法被 VMware 辨識這個問題上一直鬼打牆，最後還是爬了 [stackoverflow](#) 的文章才找到解決的方法，除了這部分以外，大部分的問題 Chat gpt 都幫忙很多。