

經濟部資訊專業人員鑑定—開放式系統類

Linux進階系統管理

核心編譯、模組管理與boot loader

崑山科技大學資訊傳播系


蔡德明

(鳥哥, VBird)

分享指引

- 核心功能與核心模組
- 開機管理程式
- 精選範例





核心功能與核心模組

核心功能

- **Linux**的核心功能主要為：
 - ❧ 開機偵測硬體，並載入適當的驅動程式
 - ❧ 驅動所有硬體
 - ❧ CPU排程
 - ❧ 記憶體管理
 - ❧ 檔案系統功能
 - ❧ 網路功能(含通訊協定與防火牆netfilter)
 - ❧ 電源管理與其他



核心版本

- 核心所在處
 - ❧ `/boot/vmlinuz-version`
- 核心版本：用 `uname -r` → 2.6.xx
 - ❧ 穩定版：2.6.xx → 偶數版
 - ❧ 發展中：2.5.xx → 奇數版
- 核心模組目錄：
 - ❧ `/lib/modules/$(uname -r)/kernel/*s`
- 核心原始碼：
 - ❧ `/usr/src/linux/`



核心模組

■ 核心模組：

☞ 可抽換，動態管理，使用上具有彈性

☞ `/lib/modules/$(uname -r)/kernel/*`

- `arch`：與硬體平台有關的項目，例如 CPU 的等級等等；
- `crypto`：核心所支援的加密的技術，例如 `md5` 或者是 `des` 等等
- `drivers`：一些硬體的驅動程式，例如顯示卡、網路卡、等等；
- `fs`：核心所支援的 `filesystems`，例如 `vfat`, `reiserfs`, `nfs` 等等
- `lib`：一些函式庫；
- `net`：與網路有關的各項協定資料，還有防火牆模組 (`net/ipv4/netfilter/*`) 等等；
- `sound`：與音效有關的各項模組；



模組相依性

- 模組相依性分析檔案：
 - ☞ `/lib/modules/`uname -r`/modules.dep`
- 更新該檔案的方法：
 - ☞ `depmod`
- 模組的載入與移除(透過上述檔案)
 - ☞ 載入模組：`modprobe module`
 - ☞ 移除模組：`modprobe -r module`



模組管理

- **lsmod**
 - ☞ 列出已經載入的核心模組
- **insmod module.ko**
 - ☞ 載入一個名為module.ko的核心模組
 - ☞ 已被 modprobe module取代
- **rmmod module**
 - ☞ 移除一個已經載入的名為module的模組
 - ☞ 已被 modprobe -r module取代
- **modinfo { module | module.ko }**
 - ☞ 查閱某個模組的資訊
 - ☞ 可查詢已載入的模組及為載入的模組檔案



USB模組

■ 兩種常見的USB模組

☞ OHCI (Open Host Controller Interface) :

- 主要由 Compaq 所發展，包括 Compaq, SiS, ALi 等等廠商發展的晶片都是用這個模組；

☞ UHCI (Universal Host Controller Interface) :

- 主要由 Intel 所發展，包括 Intel, VIA 等等廠商發展的晶片都是使用這個模組

■ 高速 USB (version 2.0)

☞ EHCI (Enhance Host Controller Interface)

- 可達到USB 2.0的速度。

核心編譯的需求

- 何時需要編譯核心：
 - ☞ 新功能的需求
 - ☞ 原本核心太過臃腫
 - ☞ 與硬體的搭配穩定性
 - ☞ 其他：如客製化核心、**bug**修復、**security**修復等
- 核心的來源
 - ☞ 原有的distributions提供的原始碼
 - ☞ <http://www.kernel.org>



核心編譯前所需套件

- 由於核心需要編譯，因此需要的套件有：

☞ make

☞ gcc

☞ automake

☞ autoconf

☞ ncurses

☞ bison



核心編譯的程序

- 取得核心原始碼
 - ☞ 放置到/usr/src/linux/中或/usr/src/kernels/中
- 進入上述目錄後，刪除不必要的資料
 - ☞ make mrproper
- 開始選擇核心功能
 - ☞ * 可以整合到核心中
 - ☞ M 可以選擇成爲核心模組
- 開始進行核心編譯、模組編譯
- 開始安裝模組
- 開機管理程式的修訂

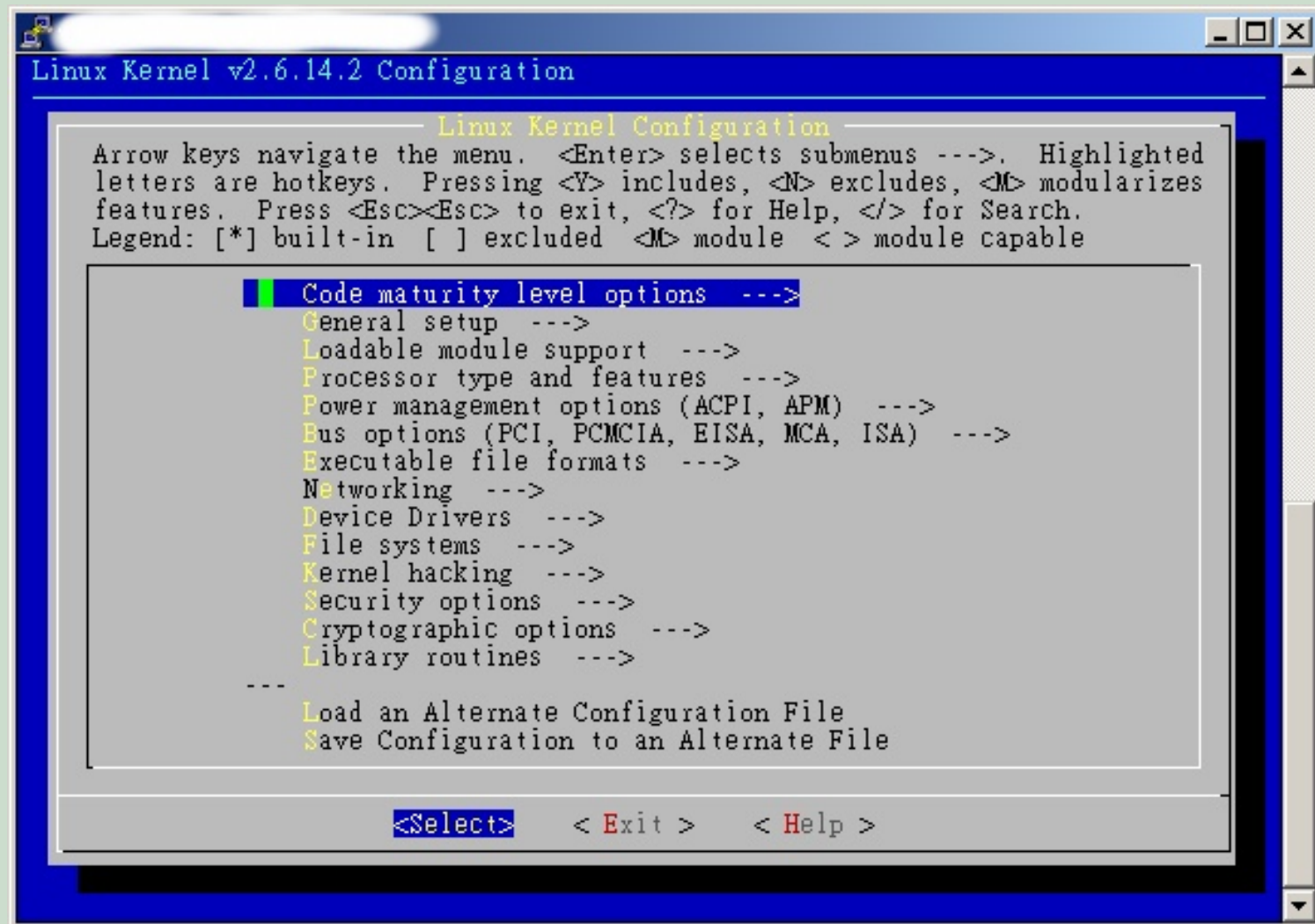


核心功能選擇的方式

- 目的在建立 `.config` 檔案，系統的核心提供的資料通常放置在：`/boot/config-version`
- 進入選擇畫面的方法：
 - ☞ `make menuconfig`
 - 在終端機進入類圖形介面的畫面
 - ☞ `make oldconfig`
 - 使用舊的預設值
 - ☞ `make {xconfig | gconfig}`
 - 使用圖形介面的方法



make menuconfig 示意圖



編譯的動作

- **make dep clean**
 - ☞ 建立函式庫的相關性，清除以前所留下來的object檔案
- **make bzImage**
 - ☞ 編譯核心，核心檔案放置於
 - /usr/src/linux/arch/i386/boot/
- **make modules**
 - ☞ 開始編譯每個可外掛的核心模組
- **make modules_install**
 - ☞ 將編譯完成的模組安裝到/lib/modules/(版本)/



核心安裝

- `make {install | bzdisk | bzlilo}`

- ☞ `install` 安裝核心到硬碟中

- ☞ `bzdisk` 安裝核心到軟碟中

- ☞ `bzlilo` 安裝核心後，並且執行 `lilo` 指令。



開機管理程式

Boot Loader

- Boot Loader 的任務
 - ☞ 呼叫核心，載入核心執行作業系統
 - ☞ 提供選單功能，可載入其他不同作業系統
 - ☞ Linux常用 grub, lilo
- 安裝處所
 - ☞ MBR
 - 但只有 446 bytes
 - ☞ Super block
- Boot Loader 的兩階段安裝
 - ☞ Stage 1：boot loader 的主程式，必須要安裝在開機區
 - ☞ Stage 2：boot loader 的所有設定檔(/boot下檔案)

Linux LOader(LILO)

■ LILO

- ❧ 老牌的Linux開機管理程式
- ❧ 可將開機資訊直接寫入開機磁區 (MBR/Superblock)
- ❧ 每次修改完設定檔，都需要重新安裝LILO才行
- ❧ 設定檔在 `/etc/lilo.conf`
- ❧ 安裝LILO的指令 `lilo -v`



/etc/lilo.conf

```
[root@linux ~]# vi /etc/lilo.conf
```

```
# 第一部份，整體的設定部分
```

```
prompt          <==強制出現 boot 的開機訊息囉！
Compact         <==可以整合一些讀取的磁區，可以保持 map 較小，適合軟碟開機時使用
timeout=50      <==如果有多重開機的話，可以設定這個延遲時間，單位0.1秒
default=linux-2.4.18 <==預設的開機項目，與底下的 label 對應！
boot=/dev/hda   <==Lilo 的開機資訊寫入到 /dev/hda 這顆硬碟的 MBR 當中。
map=/boot/map   <==用來說明 local 主機的地圖資訊囉！
install=/boot/boot.b <==關於開機區的訊息(boot sector)，不用理他沒關係！
Linear          <==在較大容量的硬碟使用時，可以加入這一個參數試試看！
lba32           <==這個東西也是在大容量的硬碟使用時候會需要的參數！
password=1234567 <==設定密碼！如果為了安全起見，可以設定您的 lilo 密碼哩！
message=/boot/message <==那個 LILO 的訊息就是在裡面出現的啦！
```

```
# 第二部分，個別的開機設定部分，一個 image 或 other 均代表一個開機設定！
```

```
image=/boot/vmlinuz-2.4.7-10 <==核心檔案啦！
    label=linux-2.4.7        <==請注意！label 前面以 [tab] 按鍵來作為分隔！
    initrd=/boot/initrd-2.4.7-10.img
    read-only                <==開機磁區掛載為唯讀！
    root=/dev/hda1           <==掛載成 / 這個 root 目錄的磁碟！
other=/dev/hdb1             <==如果是『非 Linux 核心』就以 other 來設定
    label=Windows2k         <==同樣的要有 label 來表示這個開機磁區的名稱！
```


LILO的問題

- 沒有提供互動介面，所以如果partition亂了，容易導致無法開機的問題
- 每次安裝新核心，都需要重新安裝lilo一次
- 早期的LILO在磁柱數量高於1024的磁碟中，安裝起來可能會發生問題(/boot/vmlinuz)
- 維護模式的進入：
 - ❧ linux -s
 - ❧ linux single



Grub的優點

- 只需要安裝一次
 - ☞ **stage 1** 與 **stage 2** 是分開的，只要改了設定檔就立刻生效。
 - ☞ 亦即是只要修改 `/boot/grub/menu.lst` 裡頭的設定，下次開機就生效
- 認識與支援較多的 **filesystem**：
 - ☞ 可以使用 **grub** 的主程式直接在 **filesystem** 當中搜尋核心
- 有互動介面：
 - ☞ 開機的時候，可以進入『自行編輯與修改開機設定項目』模式
 - ☞ 可進入維護模式



Grub的硬碟代號問題

- 硬碟代號以先偵測到為先設定
∞(hd0) → (hd0,0)

硬碟搜尋順序	在 Grub 當中的代號
第一顆	(hd0) (hd0,0) (hd0,1) (hd0,4)....
第二顆	(hd1) (hd1,0) (hd1,1) (hd1,4)....
第三顆	(hd2) (hd2,0) (hd2,1) (hd2,4)....



Grub 主設定檔

- 設定檔 `/boot/grub/menu.lst` 的語法
 - ∞ 重點在載入核心與虛擬磁碟(`initrd`)

```
[root@linux ~]# vi /boot/grub/menu.lst
default=0
timeout=5
splashimage=(hd0,0)/boot/grub/splash.xpm.gz
hiddenmenu
title Fedora Core (2.6.12-1.1456_FC4)
    root (hd0,0)
    kernel /boot/vmlinuz-2.6.12-1.1456_FC4 ro root=/dev/hda1 quiet vga=787
    initrd /boot/initrd-2.6.12-1.1456_FC4.img
title Fedora Core (2.6.11-1.1369_FC4)
    root (hd0,0)
    kernel /boot/vmlinuz-2.6.11-1.1369_FC4 ro root=/dev/hda1 quiet vga=787
    initrd /boot/initrd-2.6.11-1.1369_FC4.img
```

Grub 的密碼保護

- 支援 MD5 加密的密碼：
 - ☞ 在 grub shell 中進行
 - ☞ 利用 grub-md5-crypt 產生MD5密碼
- 設定檔中(menu.lst)的撰寫格式
 - ☞ password --md5 PASSWORD



initrd的任務

■ kernel file的功能

- ☞ 偵測硬體並載入適當的模組
- ☞ 模組所在處為 `/lib/modules/$(uname -r)/kernel`
- ☞ 故開機過程中『需要掛載根目錄』以載入模組
- ☞ 思考：如果核心無法載入磁碟呢？(ex>RAID)

■ 虛擬磁碟裝置 (Initial Ram Dist)

- ☞ 提供開機時所必須要的模組
- ☞ 建立虛擬檔案系統，讓核心在開機過程中可使用
- ☞ 常見模組：LVM, RAID, SCSI, USB...
- ☞ 可用 `mkinitrd` 來建立新的 `initrd` 檔案





精選範例

- 核心編譯的方法包括下列哪幾種方式(不考慮移除檔案)？
(複選) ABD
 - ☞ (A) make bzdisk
 - ☞ (B) make bzlilo
 - ☞ (C) make bzvmlinuz
 - ☞ (D) make mrproper

- 下列哪一個項目不是 **bootloader** 對系統所應負的責任？ A
 - ☞ (A) 負責決定執行哪一個 **runlevel**
 - ☞ (B) 負責載入其他非 **Linux** 的作業系統
 - ☞ (C) 負責載入系統核心映像檔
 - ☞ (D) 負責載入其他的 **Linux** 作業系統



- 在核心編譯前會先移除重製核心不必要的殘餘檔案，要執行那個指令呢？ C
 - ❧ (A) make oldconfig
 - ❧ (B) make config
 - ❧ (C) make mrproper
 - ❧ (D) make depmod

- 下列哪一個命令會自動的幫你參考舊有組態作基本設定？ D
 - ❧ (A) make menuconfig
 - ❧ (B) make config
 - ❧ (C) make xconfig
 - ❧ (D) make oldconfig



■ 編譯完成的核心放在那個位置？ D

- ☞ (A) /usr/src/linux-2.4/configs
- ☞ (B) /usr/src/linux-2.4/kernel
- ☞ (C) /usr/src/linux-2.4/arch/i386
- ☞ (D) /usr/src/linux-2.4/arch/i386/boot

■ 若要自行重新編譯核心(kernel)，以下那個套件『不必』安裝？ B

- ☞ (A) gcc
- ☞ (B) java
- ☞ (C) make
- ☞ (D) bison



- 如果我們嘗試自行修改系統核心原始碼，應從以下那個目錄開始進行？ D
 - ☞ (A) /linux
 - ☞ (B) /sys/linux
 - ☞ (C) /src/linux
 - ☞ (D) /usr/src/linux

- 以下哪種核心支援 Intel USB controller？ B
 - ☞ (A) OHCI
 - ☞ (B) UHCI
 - ☞ (C) IHCI
 - ☞ (D) VHCI



- 如下哪一種核心編譯操作順序是最為建議的？ B
 - ☞ (A) make configs; make; make install
 - ☞ (B) make menuconfig; make dep; make bzImage modules
 - ☞ (C) ./configure; make dep; make bzImage; make modules
 - ☞ (D) make oldconfig; make clean; make dep; make bzImage; make modules

- 請問 `depmod -a` 命令會更新哪一個檔案？ D
 - ☞ (A) /etc/modules.conf
 - ☞ (B) /etc/conf.modules
 - ☞ (C) /etc/modules.dep
 - ☞ (D) /lib/modules/\$(uname -r)/modules.dep



- 核心設定完成後下一個步驟是連結程式碼與函式庫，其指令為？ C
 - ☞ (A) make link
 - ☞ (B) make lib
 - ☞ (C) make dep
 - ☞ (D) make mrproper

- 模組 **foo** 需要模組 **bar** 才能使用，但模組 **bar** 還沒有被載入。請問如何把模組 **foo** 載入核心中？ D
 - ☞ (A) insmod foo
 - ☞ (B) depmod foo
 - ☞ (C) insmod -bar foo
 - ☞ (D) modprobe foo



- 當完成kernel功能的設定後，要如何重新建構新的kernel？A
 - ☞ (A) make dep clean bzImage modules modules_install
 - ☞ (B) make bzImage clean dep modules_install modules
 - ☞ (C) make mrproper bzImage modules modules_install
 - ☞ (D) make bzImage mrproper modules_install modules

- 請問通常 kernel 都放置在那個目錄下？A
 - ☞ (A) /boot
 - ☞ (B) /kernel
 - ☞ (C) /images
 - ☞ (D) /kernels



- 請問 `/proc/cmdline` 這個檔案所儲存的資訊為何？ C
 - ☞ (A) 目前終端機中所執行的指令與參數
 - ☞ (B) 目前微處理器所執行的程序的命令列參數
 - ☞ (C) 開機時傳遞給核心的參數
 - ☞ (D) 當初編譯核心時所下的所有參數

- ____指令是用來載入模組以及依附的所有模組，而模組相依性的資訊是存在____。請問這兩個空格的內容？ D
 - ☞ (A) `insmod; /etc/modprobe.conf`
 - ☞ (B) `depmod; /etc/sysconfig/kernel`
 - ☞ (C) `modprobe; /lib/modules/$(uname -r)/modules.alias`
 - ☞ (D) `modprobe; /lib/modules/$(uname -r)/modules.dep`



- 開機時如果使用grub開機管理程式，切至單人模式時，該按什麼鍵設定？ AD
 - ☞ (A) 按『A』，然後編輯設定進入單人模式
 - ☞ (B) 按[ctrl]-S，然後編輯設定進入單人模式
 - ☞ (C) 按『S』，然後編輯設定進入單人模式
 - ☞ (D) 按『E』，然後編輯設定進入單人模式

- 以下關於initrd的敘述，何者正確？ AC
 - ☞ (A) 開機時，由boot loader呼叫執行
 - ☞ (B) 該程式於kernel載入前即以執行
 - ☞ (C) 該程式會產生一塊 RAM disk
 - ☞ (D) 該程式會初始化 (initial) 定義在/etc/rc.d/裡面的程式



- 如果要以密碼來保護grub boot loader，必須使用下列哪一行在/boot/grub/grub.conf中？ A
 - ❧ (A) password --md5 <MD5加密字串>
 - ❧ (B) lock --md5 <MD5加密字串>
 - ❧ (C) passwd --md5 <MD5加密字串>
 - ❧ (D) setpassword --md5 <MD5加密字串>

