



---

第 31 章  
核 心

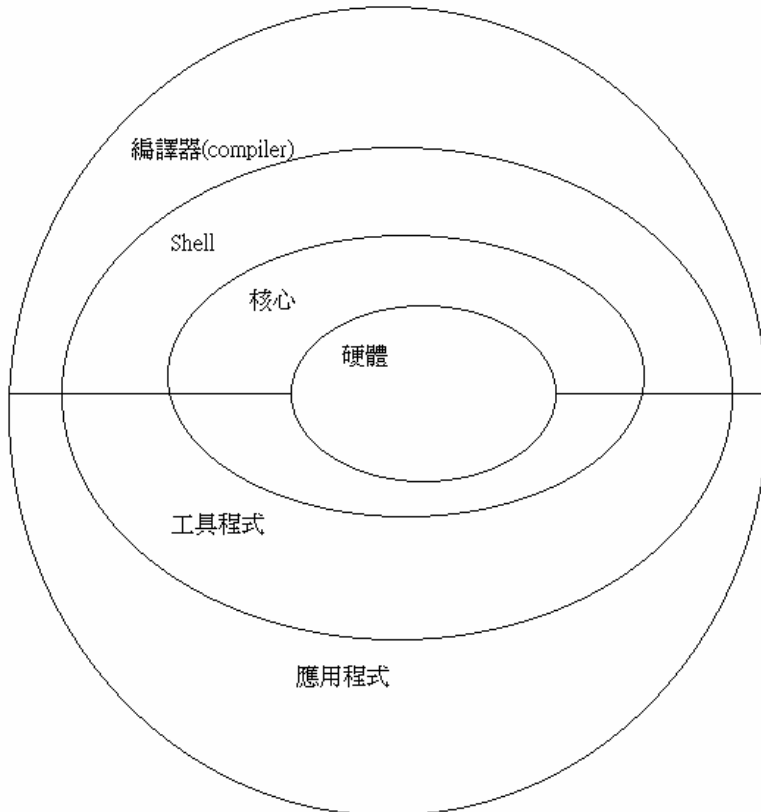
---

麒麟  
Linux

## 第 31 章 核 心

**核心(kernel)**：作業系統的核心，並在系統一啟動時即被載入，其功能為控制系統硬體及所有軟體的程式、管理電腦的所有資源，為硬體和使用者間的介面，可讓使用者使用系統所提供的服務。

核心是最底層的軟體，它是和硬體整合的介面，而且核心是唯一允許直接存取硬體的軟體。這是由硬體的驅動程式所提供，讓我們的核心能夠和硬體直接溝通。例如當我們的程式要連上網路，程式會呼叫核心 Kernel 而且告訴它要存取乙太網路卡的介面，而此時核心會轉換這些呼叫到指定的介面卡硬體。



下面是我們為何要編譯核心的理由：

1. **新的驅動程式**：當我們安裝新的硬體，但它的驅動程式並不包含在標準的核心時，我們就可能要重新設定核心的原始檔並且編譯它。

2. **更新和除錯**：當我們要更新 RedHat Linux Fedora 1 作業系統的核心，或者除錯 RedHat Linux Fedora 1 作業系統的核心來增加系統效能時，我們就需要重新設定核心的原始檔並且編譯它。

3. **減少記憶體消費**：標準的 RedHat Linux Fedora 1 作業系統核心包括許多我們不會使用到的驅動程式，我們可以去掉這些驅動程式來減少記憶體的消費。

4. **增加核心的特性和最佳化核心**：標準的 RedHat Linux Fedora 1 作業系統可能不支援多顆 CPU 作運算，我們可以編譯新的核心來支援多 CPU 的運算。

5. **硬體衝突**：我們可以重新編譯 RedHat Linux Fedora 1 作業系統的核心原始檔來減少硬體的衝突。

我們可以使用 `rpm -q kernel` 來查詢目前安裝在 RedHat Linux Fedora 1 作業系統的是什麼版本的核心。Linux 核心版本，第一個數字隨著主要核心設計的改變而改變。Linux 核心版本，第二個數字代表穩定性，偶數編號是代表穩定的版本。奇數是代表正在開發的版本，因此可能不穩定。例如核心 2.4.20 有第一個數字 2，第二個數字 4(這表示是穩定版本核心)，和修補編號 20。例如核心 2.5.75 有第一個數字 2，第二個數字 5(這表示是發展中版本核心)，和修補編號 75。

```
# rpm -q kernel
```

```
kernel-2.4.22-8
```

這是保存 Linux 核心的網址 [kernel.org](http://kernel.org)。當我們需要更新我們核心時，我們可以從下列的地方下載最新的核心並且更新安裝。我們可以從 [kernel.org](http://kernel.org) 學習更多有關 Linux 核心的知識。核心原始碼通常都放在 `/usr/src/linux-版本` 的目錄下。

這是 linux 的資訊中心。

[www.linuxhq.com](http://www.linuxhq.com)

這是中科院的 Linux 下載。

<ftp://linux.sinica.edu.tw>

中央大學的 Linux 下載中心。

<ftp://ftp.ncu.edu.tw>



### 31-1 核心微調

有一些核心的特色可以被打開或關閉而不用經過重新安裝或編譯新的核心或模組，例如 IP 轉向。這些微調參數是由 `/proc/sys` 檔案所控制。我們所設定的參數都儲存在 `/etc/sysctl.conf` 的組態檔中。例如當我們在網路 IP 面板中設定啟動 IP 轉向，將設定 `/proc/sys/net/ipv4/ip_forward` 到 1。我們也可以直接使用指令 `sysctl` 來設定。我們使用 `-p` 選項，這樣 `sysctl` 將從 `/etc/sysctl.conf` 檔讀取參數。我們可以使用 `-w` 選項來改變指定參數。

使用者可以使用 `kernel.domainname` 來改變 domain name 參數。

```
#/sbin/sysctl -w kernel.domainname="flash.aasir.com"
```

```
[root@flash src]# /sbin/sysctl -w kernel.domainname="flash.aasir.com"
kernel.domainname = flash.aasir.com
```

這是啟動 IP 轉向。

```
# /sbin/sysctl -w net.ipv4.ip_forward=1
```

```
[root@flash src]# /sbin/sysctl -w net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
```

我們也可以顯示目前參數的數值。

```
# /sbin/sysctl net.ipv4.ip_forward
```

```
[root@flash src]# /sbin/sysctl net.ipv4.ip_forward
net.ipv4.ip_forward = 1
```

### 31-2 修改核心前的設定

修改核心前我們應該保留我們目前的核心，否則我們目前正在使用的核心將會被覆寫。我們可以複製核心來保留目前核心的備份。這核心影像檔稱為 `vmlinuz`-版本。它是被放置在 `/boot` 目錄下。`/boot/vmlinuz` 是符號連接檔，他是連接真實的核心檔。`Vmlinuz-2.4.20-8.back` 為我們核心的備份檔案。

```
# cd /boot
```



```
[root@flash src]# cd /boot
[root@flash boot]# ls
boot.b          message          System.map-2.4.20-8
chain.b         message.java     vmlinuz-2.4.20-8
config-2.4.20-8 module-info      vmlinuz
grub            module-info-2.4.20-8 vmlinuz-2.4.20-8
initrd-2.4.20-8.img os2_d.b         vmlinuz-2.4.20-8.back
kernel.h        System.map
```

我們可以編輯 Makefile，我們可以改變到/usr/src/linux-2.4 的目錄下。我們可以將修改時間加到 Makefile 中，只需加入 EXTRAVERSION 為版本，後面我們可以指定時間。

```
# cd /usr/src/linux-2.4
#vi Makefile
# EXTRAVERSION=--september2003
```

我們可以複製核心的備份。

```
# cp /boot/vmlinuz-2.4.20-8 /boot/vmlinuz-2.4.20-8.back
```

我們也可以備份 System.map 檔。這個檔案 System.map 包含去啟動核心功能模組的核心符號。例如 2.4.20 的核心，其啟動核心功能模組的核心符號檔案為 System.map-2.4.20-8。我們也可以備份位於/lib/modules/版本的目錄，例如 2.4.20 的核心，其目錄為/lib/modues/2.4.20-8。/lib/modules/版本的目錄是建立在與我們核心一起運作的模組。

假如我們使用 boot 載入啟動核心，我們應該建立新的進入點來給舊的核心，然後我們可以在這組態中建立一個進入點給新的核心。例如在/etc/grub.conf 載入核心組態中。假如我們使用載入分割的標籤，然後這個核心的 root 選項看起來會像是啟動分割標籤"/。

```
#vi /etc/grub.conf
```

```
default=0
timeout=10
splashimage=(hd0,0)/boot/grub/splash.xpm.gz
title Red Hat Linux (2.4.20-8)
    root (hd0,0)
    kernel /boot/vmlinuz-2.4.20-8 ro root=LABEL=/ hdc=ide-scsi
    initrd /boot/initrd-2.4.20-8.img
```



### 31-3 安裝核心

為了安裝新核心，我們需要下載核心套件。我們可以先觀看我們 RedHat Linux Fedora 1 作業系統作業系統上裝上哪一些核心套件。

```
# rpm -qa|grep kernel
[root@flash chaiyen]# rpm -qa|grep kernel
kernel-2.4.22-1.2115.nptl
kernel-source-2.4.22-1.2115.nptl
kernel-doc-2.4.22-1.2115.nptl
kernel-utils-2.4-9.1.101.fedora
kernel-pcmcia-cs-3.1.31-13
```

假如我們將 RedHat Linux Fedora 1 作業系統安裝在筆記型電腦，或者是在有 PCMCIA 卡的系統上，我們將需要安裝 pcmcia-cs 套件。

```
kernel-pcmcia-cs-3.1.31-13
```

假如我們想要客製化我們的核心，我們應該下載 source 套件。假如我們需要 Linux 使用模式工具，我們需要 utils 套件，這 doc 套件提供更新文件。

```
kernel-source-2.4.20-8
```

```
kernel-utils-2.4-8.29
```

```
kernel-doc-2.4.20-8
```

假如我們只需要簡單的核心，我們只需要下面的套件。

```
kernel-2.4.20-8
```

我們使用 rpm -i 來安裝我們的新核心。

```
# rpm -ivh kernel-2.4.20-20.9.i586.rpm
```

```
[root@flash /]# rpm -ivh kernel-2.4.20-20.9.i586.rpm
warning: kernel-2.4.20-20.9.i586.rpm: V3 DSA signature: NOKEY, key ID db42a60e
Preparing...
  l:kernel
```

我們也可以使用 rpm -Uvh 來更新我們的核心。我們最新的核心都可以在 RedHat Linux Fedora 1 作業系統的網站上下載。

```
# rpm -Uvh kernel-2.4.20-20.9.i586.rpm
```

這時後我們核心已經由 kernel-2.4.20-8 更新為 kernel-2.4.20-20.9。



```
[root@flash boot]# rpm -qa|grep kernel
kernel-pcmcia-cs-3.1.31-13
kernel-source-2.4.20-8
kernel-2.4.20-20.9
kernel-utils-2.4-8.29
kernel-doc-2.4.20-8
kernel-2.4.20-8
```

在 RedHat Linux Fedora 1 作業系統中核心是安裝在/boot 目錄。所有新舊的核心都是安裝在/boot 目錄下，而 vmlinuz 則是連接到新的核心。假如我們使用 GRUB 載入器，我們則不需要改變/etc/grub.conf 的組態，因為/boot/vmlinuz 已經連接新的核心了。

```
[root@flash chaiyen]# cd /boot
[root@flash boot]# ls
boot.b                message               System.map-2.4.20-8
chain.b               message.ja            vmlinux-2.4.20-20.9
config-2.4.20-20.9    module-info           vmlinux-2.4.20-8
config-2.4.20-8      module-info-2.4.20-20.9 vmlinuz
grub                  module-info-2.4.20-8  vmlinuz-2.4.20-20.9
initrd-2.4.20-20.9.img os2_d.b               vmlinux-2.4.20-8
initrd-2.4.20-8.img  System.map            vmlinuz-2.4.20-8.back
kernel.h              System.map-2.4.20-20.9
```

我們在/etc/grub 新增 vmlinuz-2.4.20-20.9 的新核心，這樣在開機時就可以選取。

#vi /etc/grub

```
default=1
timeout=10
splashimage=(hd0,0)/boot/grub/splash.xpm.gz
title Red Hat Linux (2.4.20-20.9)
    root (hd0,0)
    kernel /boot/vmlinuz-2.4.20-20.9 ro root=LABEL=/ hdc=ide-scsi
    initrd /boot/initrd-2.4.20-20.9.img
title Red Hat Linux (2.4.20-8)
    root (hd0,0)
    kernel /boot/vmlinuz-2.4.20-8 ro root=LABEL=/ hdc=ide-scsi
    initrd /boot/initrd-2.4.20-8.img
```

### 31-3-1 修補核心

如果只是小部份更改核心，我們可以使用修補核心來解決問題，這樣就可以不用重新編譯核心。我們只需下載 patch 修補核心檔，然後解壓縮，然後再使用 patch 指



令來完成修補核心這個動作。

```
#cd /usr/src
#gzip -cd patchvnum.gz |patch -p0
這是修補核心的其它方法。
#gunzip patch-2.4.22gz
#patch -p0 < patch-2.4.22
```

### 31-4 從原始檔編譯核心

我們剛剛安裝已經編譯過的核心，現在我們要編譯原始檔的核心，然後自己建立這已編譯的核心。原始碼的核心是由 gcc 編譯器來編譯，就像是其它原始檔的 C 程式碼，經過 gcc 編譯成二進位碼。我們現在的核心是 linux-2.4.20，我們現在有 linux-2.4.22.tar.gz 的原始檔，現在我們要將核心安裝並且昇級。這原始檔是放在 /usr/src 目錄，當我們下載並且安裝 linux-2.4.22.tar.gz 的原始檔，它會在 /usr/src 下建立 /usr/src/linux-2.4.22 的目錄。我們的原始檔 linux.2.4.22.tar.gz 是在 [www.kernel.org](http://www.kernel.org) 中下載，我們可以從該網站下載最新的 Linux 核心。

```
# cd /usr/src/
# gzip -cd linux-2.4.22.tar.gz |tar xvf -
```

#### 31-4-1 組態核心

當我們從原始檔安裝核心後，我們一定要組態核心。組態核心包含我們希望我們的核心提供哪一些功能選項。這包含不同裝置的驅動程式，像是音效卡和 SCSI 卡。這些過程都需要組態核心。我們可以由 /usr/share/doc 來查看許多核心相關的文件。

```
# rpm -ql kernel-doc
```

我們可以使用 config 工具、menuconfig 工具和 xconfig 工具來組態核心，這些工具都可以執行組態核心的工作。我們工具所有的設定都在核心原始檔目錄中的 .config 檔案。假如我們要移除所有組態，我們可以使用 mrproper 的選項來移除 .config 檔。

```
# make mrproper
```

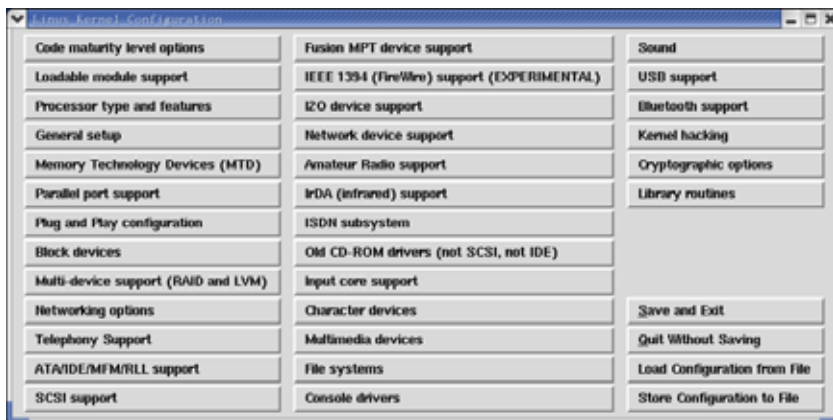




我們進入/usr/src/linux-2.4.22目錄，並且使用xconfig工具，這樣就可以打開Linux核心組態視窗。這要在X視窗模式下。

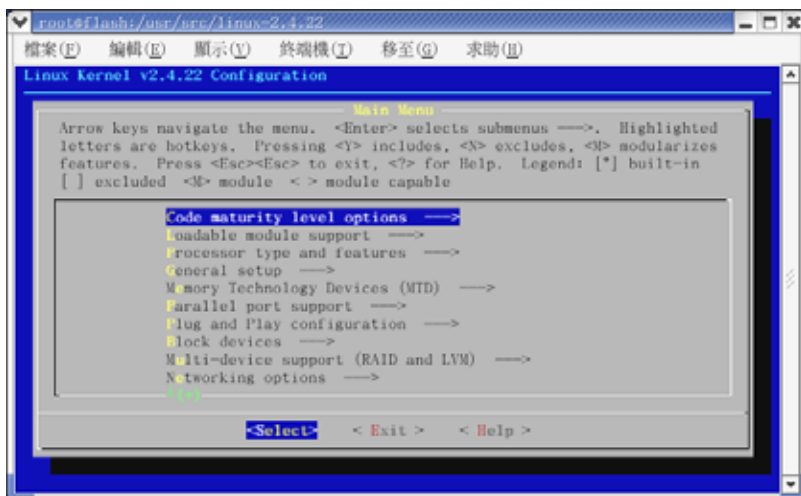
```
#cd /usr/src/linux-2.4.22
```

```
#make xconfig
```



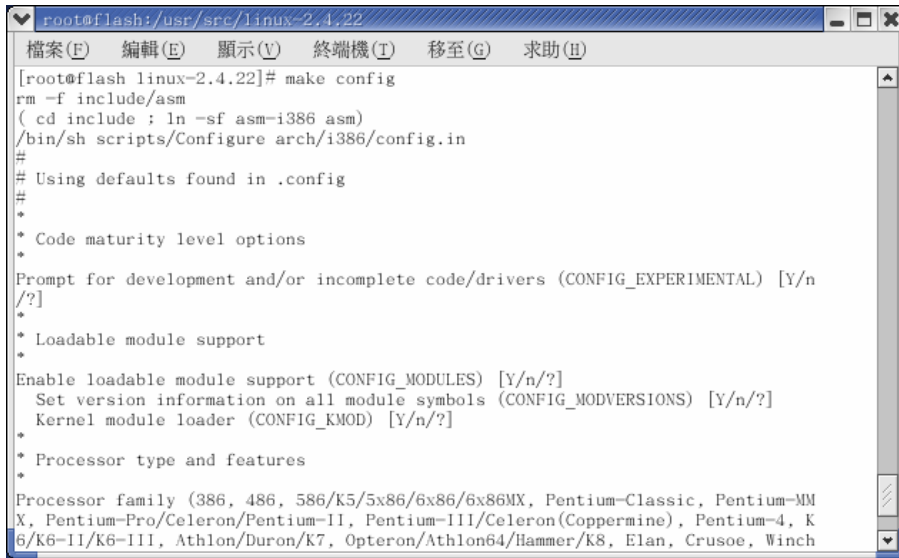
我們進入/usr/src/linux-2.4.22目錄，並且使用menuconfig工具，這樣就可以打開Linux核心組態選項。

```
#make menuconfig
```



我們也可以使用 make config 來組態核心選項，這是使用文字模式。

#make config



```
root@flash:/usr/src/linux-2.4.22
檔案(F) 編輯(E) 顯示(V) 終端機(T) 移至(G) 求助(H)
[root@flash linux-2.4.22]# make config
rm -f include/asm
( cd include ; ln -sf asm-i386 asm)
/bin/sh scripts/Configure arch/i386/config.in
#
# Using defaults found in .config
#
*
* Code maturity level options
*
Prompt for development and/or incomplete code/drivers (CONFIG_EXPERIMENTAL) [Y/n
/?]
*
* Loadable module support
*
Enable loadable module support (CONFIG_MODULES) [Y/n/?]
Set version information on all module symbols (CONFIG_MODVERSIONS) [Y/n/?]
Kernel module loader (CONFIG_KMOD) [Y/n/?]
*
* Processor type and features
*
Processor family (386, 486, 586/K5/5x86/6x86/6x86MX, Pentium-Classic, Pentium-MM
X, Pentium-Pro/Celeron/Pentium-III, Pentium-III/Celeron(Coppermine), Pentium-4, K
6/K6-II/K6-III, Athlon/Duron/K7, Opteron/Athlon64/Hammer/K8, Elan, Crusoe, Winch
```

### 31-4-2 使用 config 組態核心

我們在文字模式輸入 make config 來組態核心的設定。

#make config

這是實驗性的組態，是否用來發展核心，預設為 NO。

Prompt for development and/or incomplete code/drivers  
(CONFIG\_EXPERIMENTAL) [Y/n/?]

(1)這是支援載入模組的選項。

\* Loadable module support

我們可以啟動載入模組，所以選取 Yes。動態載入模組可以讓我們核心更有彈性，且可以容納較多的模組。

Enable loadable module support (CONFIG\_MODULES) [Y/n/?]



是否可以讓其它版本的核心使用模組。

Set version information on all module symbols (CONFIG\_MODVERSIONS)  
[Y/n/?]

是否支援核心模組載入器。

Kernel module loader (CONFIG\_KMOD) [Y/n/?]

## (2)CPU 的型態選項

這是我們 CPU 的選項。如果我們的中央處理器是 Pentium，我們可以輸入 Pentium-III 或 Pentium-4。如果我們的中央處理器是超微的 AMD，則可以輸入其 CPU 的型號。

Processor family (386, 486, 586/K5/5x86/6x86/6x86MX, Pentium-Classic, Pentium-MMX, Pentium-Pro/Celeron/Pentium-II, Pentium-III/Celeron(Coppermine), Pentium-4, K6/K6-II/K6-III, Athlon/Duron/K7, Opteron/Athlon64/Hammer/K8, Elan, Crusoe, Winchip-C6, Winchip-2, Winchip-2A/Winchip-3, CyrixIII/VIA-C3, VIA-C3-2)  
[Pentium-III/Celeron(Coppermine)]

這是是否支援例外處理。

Machine Check Exception (CONFIG\_X86\_MCE) [Y/n/?]

這是大容量記憶體의 支援。如果我們使用大容量記憶體 4GB 則可以選取 on。目前 2.4 的核心已經可以支援 64GB 的記憶體。

High Memory Support (off, 4GB, 64GB) [off]

這是讓核心支援多顆 CPU 運算處理。

Symmetric multi-processing support (CONFIG\_SMP) [Y/n/?]

## (3)一般設定

\* General setup

核心的一般設定包含網路支援、PCI 卡裝置支援、ISA 裝置支援和熱插拔裝置的支援選項。我們在這也可以啟動加速 SSL 的選項。

這是核心是否提供網路支援，我們選取 Yes。



Networking support (CONFIG\_NET) [Y/n/?]

這是提供 PCI 插槽支援，一般的網路卡或介面卡都是屬於 PCI 卡。

PCI support (CONFIG\_PCI) [Y/n/?]

這是 PCI 存取模式，有 BIOS 或 Direct，Any 是支援這兩種選項。

PCI access mode (BIOS, Direct, Any) [Any]

這是支援 ISA 插槽的支援。

ISA bus support (CONFIG\_ISA) [Y/n/?]

這是支援熱插拔裝置的選項。例如可任意拔除的 USB 界面隨身碟或 USB 界面無線網路裝置。

Support for hot-pluggable devices (CONFIG\_HOTPLUG) [Y/n/?]

#### (4) 平行埠裝置

\* Parallel port support

有一些老舊的印表機使用平行連接埠，這樣我們就可以啟動這個選項。

Parallel port support (CONFIG\_PARPORT) [N/y/m/?]

#### (5) 即插即用

\* Plug and Play configuration

即插即用選項可以讓我們的系統自動偵測新增或移除的硬體裝置。

Plug and Play support (CONFIG\_PNP) [Y/m/n/?]

ISA Plug and Play support (CONFIG\_ISAPNP) [Y/m/n/?]

#### (6) 儲存裝置

\* Block devices

這是儲存裝置的選項。



這是核心是否支援軟碟機。

Normal floppy disk support (CONFIG\_BLK\_DEV\_FD) [Y/m/n/?]

這是核心是否支援網路儲存裝置。

Network block device support (CONFIG\_BLK\_DEV\_NBD) [N/y/m/?]

### (7)磁碟陣列支援

Multi-device support (RAID and LVM)

支援多重裝置，如磁碟陣列的支援。

Multiple devices driver support (RAID and LVM) (CONFIG\_MD) [N/y/?]

### (8)網路選項

\* Networking options

這是設定各種網路功能和選項，也包含無線網路協定 802.1Q 的支援。

這是支援網路封包傳送。

Packet socket: mmaped IO (CONFIG\_PACKET\_MMAP) [N/y/?]

這是網路連接裝置模擬。

Netlink device emulation (CONFIG\_NETLINK\_DEV) [N/y/m/?]

是否支援網路封包過濾。

Network packet filtering (replaces ipchains) (CONFIG\_NETFILTER) [N/y/?]

是否讓核心支援網路封包的傳送協定。

Socket Filtering (CONFIG\_FILTER) [N/y/?]

是否讓核心支援 TCP/IP 網路協定。

TCP/IP networking (CONFIG\_INET) [Y/n/?]

是否讓核心支援多重封包傳播。



IP: multicasting (CONFIG\_IP\_MULTICAST) [Y/n/?]

是否讓核心支援 router 路由。

IP: advanced router (CONFIG\_IP\_ADVANCED\_ROUTER) [N/y/?]

這是讓核心支援無線網路 802.1Q。如果我們 Linux 主機有當作無線網路的基地台，則可以選取 Yes。

802.1Q VLAN Support (CONFIG\_VLAN\_8021Q) [N/y/m/?]

## (9)IDE、ATA 和 ATAPI 儲存裝置

\* IDE, ATA and ATAPI Block devices

我們要使用 IDE 的磁碟裝置，所以選取 Yes。

Enhanced IDE/MFM/RLL disk/cdrom/tape/floppy support  
(CONFIG\_BLK\_DEV\_IDE) [Y/m/n/?]

這是包含 IDE/ATA-2 磁碟支援選項。

Include IDE/ATA-2 DISK support (CONFIG\_BLK\_DEV\_IDEDISK) [Y/m/n/?]

這是支援選項 IDE/ATAPI 光碟機。

Include IDE/ATAPI CDROM support (CONFIG\_BLK\_DEV\_IDECD) [Y/m/n/?]

這是包含 IDE/ATAPI 磁帶支援選項。

Include IDE/ATAPI TAPE support (CONFIG\_BLK\_DEV\_IDETAPE) [N/y/m/?]

這是包含 IDE/ATAPI 軟碟支援選項。

Include IDE/ATAPI FLOPPY support (CONFIG\_BLK\_DEV\_IDEFLOPPY)  
[N/y/m/?]

這是 SCSI 模擬支援選項。

SCSI emulation support (CONFIG\_BLK\_DEV\_IDESCSI) [N/y/m/?]

## (10)SCSI 支援

\* SCSI support



這是使用 SCSI 的磁碟、磁帶和光碟機選項。假如我們有使用 SCSI 的裝置，請選取這個選項。

這是 SCSI 磁碟的支援選項。

SCSI disk support (CONFIG\_BLK\_DEV\_SD) [Y/m/n/?]

這是 SCSI 磁帶的支援選項。

SCSI tape support (CONFIG\_CHR\_DEV\_ST) [N/y/m/?]

這是 SCSI 光碟機的支援選項。

SCSI CD-ROM support (CONFIG\_BLK\_DEV\_SR) [N/y/m/?]

### (11)網路裝置支援

\* Network device support

這是核心支援網路裝置的選項。如果我們使用 ADSL、Cable Modem、或電話撥接上網，就應該選取 Yes。我們在了解我們的網卡晶片製造商後，就要在核心選項中選取。

Network device support (CONFIG\_NETDEVICES) [Y/n/?]

如果我們使用螃蟹卡來上網，則選取 RealTek 乙太網路選項 Yes。

RealTek RTL-8139 PCI Fast Ethernet Adapter support (CONFIG\_8139TOO) [Y/m/n/?]

這是威盛的網路卡晶片支援選項。

VIA Rhine support (CONFIG\_VIA\_RHINE) [N/y/m/?]

### (12)高速乙太網路卡

\*Ethernet (1000 Mbit)

這是友訊的高速乙太網路卡支援選項。

D-Link DL2000-based Gigabit Ethernet support (CONFIG\_DL2K) [N/y/m/?]



這是 Realtek 的高速乙太網路卡選項。

Realtek 8169 Gigabit Ethernet support (CONFIG\_R8169) [N/y/m/?]

這是點對點傳播協定的選項。如果我們使用 ADSL 或撥接 Modem，則要選取。

PPP (point-to-point protocol) support (CONFIG\_PPP) [N/y/m/?]

這是 serial line 的支援，如果我們使用撥接 Modem 則選取 Yes。

SLIP (serial line) support (CONFIG\_SLIP) [N/y/m/?]

### (13)環狀網路裝置支援

\* Token Ring devices

這是環狀網路驅動程式的支援選項。

Token Ring driver support (CONFIG\_TR) [N/y/?]

這是光籤驅動程式支援選項。

Fibre Channel driver support (CONFIG\_NET\_FC) [N/y/?]

### (14)PCMCIA 網路裝置

\* PCMCIA network device support

一般筆記型電腦都是使用 PCMCIA 網路介面卡，這是核心支援 PCMCIA 網路裝置的選項。

PCMCIA network device support (CONFIG\_NET\_PCMCIA) [Y/n/?]

這是 PCMCIA 無線網路卡的支援。

Pcmcia Wireless LAN (CONFIG\_NET\_PCMCIA\_RADIO) [Y/n/?]

### (15)字元裝置

\* Character devices

字元裝置有鍵盤、滑鼠、虛擬終端機、虛擬螢幕 Unix98 PTY 和序列連接埠。





這是核心是否支援虛擬終端機的選項。我們 Linux 是支援多個虛擬終端機。

Virtual terminal (CONFIG\_VT) [Y/n/?]

這是支援虛擬終端機螢幕的選項。

Support for console on virtual terminal (CONFIG\_VT\_CONSOLE) [Y/n/?]

這是支援標準序列連接埠。

Standard/generic (8250/16550 and compatible UARTs) serial support (CONFIG\_SERIAL) [Y/m/n/?]

這是支援序列連接埠的螢幕。

Support for console on serial port (CONFIG\_SERIAL\_CONSOLE) [N/y/?]

這是支援非標準序列連接埠。

Non-standard serial port support (CONFIG\_SERIAL\_NONSTANDARD) [N/y/?]

這是虛擬終端裝置的支援選項。

Unix98 PTY support (CONFIG\_UNIX98\_PTYS) [Y/n/?]

如果我們 Linux 要提供多個虛擬終端機給使用者，最多可達 2048 個虛擬終端機，預設為 256 個。

Maximum number of Unix98 PTYs in use (0-2048) (CONFIG\_UNIX98\_PTY\_COUNT) [256]

## (16)滑鼠

\* Mice

這是讓核心支援 Bus Mouse、PS/2 Mouse。

這是讓核心支援 Bus Mouse 的選項。

Bus Mouse Support (CONFIG\_BUSMOUSE) [N/y/m/?]

這是讓核心支援非序列埠或 Bus 的滑鼠。

Mouse Support (not serial and bus mice) (CONFIG\_MOUSE) [Y/m/n/?]

這是讓核心支援 PS/2 的滑鼠。

PS/2 mouse (aka "auxiliary device") support (CONFIG\_PSMOUSE) [Y/n/?]



## (17)多媒體裝置

\* Multimedia devices

這是核心是否支援影像的選項。

Video For Linux (CONFIG\_VIDEO\_DEV) [N/y/m/?]

## (18)檔案系統

\* File systems

這是 Linux 核心支援其它作業系統的檔案系統選項。

核心是否要支援磁碟分配的選項。

Quota support (CONFIG\_QUOTA) [N/y/?]

這是核心是否要支援自動掛載選項。

Kernel automounter support (CONFIG\_AUTOFS\_FS) [N/y/m/?]

這是核心是否支援 Ext3 的檔案系統選項。

Ext3 journalling file system support (CONFIG\_EXT3\_FS) [N/y/m/?]

這是核心是否支援微軟 FAT 檔案系統的選項。

DOS FAT fs support (CONFIG\_FAT\_FS) [N/y/m/?]

這是核心是否支援虛擬記憶體檔案系統的選項。

Virtual memory file system support (former shm fs) (CONFIG\_TMPFS) [Y/n/?]

這是核心是否支援光碟機的檔案系統 ISO9660。

ISO 9660 CDROM file system support (CONFIG\_ISO9660\_FS) [Y/m/n/?]

這是支援微軟 NTFS 的檔案系統選項。

NTFS file system support (read only) (CONFIG\_NTFS\_FS) [N/y/m/?]

這是 Unix98 虛擬終端機的檔案系統選項支援。

/dev/pts file system for Unix98 PTYs (CONFIG\_DEVPTS\_FS) [Y/n/?]



這是 Linux 的 ext2 檔案系統選項支援。

Second extended fs support (CONFIG\_EXT2\_FS) [Y/m/n/?]

這是支援其它 Unix 的檔案系統選項。

UFS file system support (read only) (CONFIG\_UFS\_FS) [N/y/m/?]

## (19)網路檔案系統

\* Network File Systems

這是核心支援網路檔案系統的選項。例如 NFS、SMB(Samba)、NCP(NetWare) 和 HFT(麥金塔)。

這是核心支援網路檔案系統 NFS 的選項。

NFS file system support (CONFIG\_NFS\_FS) [Y/m/n/?]

這是核心支援網路檔案系統使用者端的支援。

Provide NFSv3 client support (CONFIG\_NFS\_V3) [N/y/?]

這是核心允許透過網路檔案系統直接存取的支援選項。

Allow direct I/O on NFS files (EXPERIMENTAL) (CONFIG\_NFS\_DIRECTIO) [N/y/?]

這是網路檔案系統伺服器端的選項支援。

NFS server support (CONFIG\_NFSD) [Y/m/n/?]

## (20)分割類型

\* Partition Types

這是分割類型的選項。

Advanced partition selection (CONFIG\_PARTITION\_ADVANCED) [N/y/?]



## (21) 螢幕驅動程式

\* Console drivers

這是核心使用螢幕 VGA 文字模式。

VGA text console (CONFIG\_VGA\_CONSOLE) [Y/n/?]

這是影像選項支援。

Video mode selection support (CONFIG\_VIDEO\_SELECT) [N/y/?]

## (22) 音效卡

Sound

這是核心支援音效卡的選項。一般要在 Linux 上安裝音效卡是較為不容易，但在 RedHat Linux Fedora 1 作業系統中音效卡的設定較為簡單，如果我們有使用音效卡，則選取 Yes。

Sound card support (CONFIG\_SOUND) [Y/m/n/?]

這是麗臺的音效卡。

ALi5455 audio support (CONFIG\_SOUND\_ALI5455) [N/y/m/?]

這是創建的音效卡。

Creative SBLive! (EMU10K1) (CONFIG\_SOUND\_EMU10K1) [N/y/m/?]

這是各家廠牌的音效卡。

Trident 4DWave DX/NX, SiS 7018 or ALi 5451 PCI Audio Core (CONFIG\_SOUND\_TRIDENT) [N/y/m/?]

Support for Turtle Beach MultiSound Classic, Tahiti, Monterey (CONFIG\_SOUND\_MSNDCLAS) [N/y/m/?]

Support for Turtle Beach MultiSound Pinnacle, Fiji (CONFIG\_SOUND\_MSNDPIN) [N/y/m/?]

這是威盛的音效卡。

VIA 82C686 Audio Codec (CONFIG\_SOUND\_VIA82CXXX) [N/y/m/?]



### (23)USB 支援

\* USB support

這是核心支援 USB 的選項。

Support for USB (CONFIG\_USB) [Y/m/n/?]

### (24)核心除錯

\* Kernel hacking

這是核心支援偵錯模式的選項。這是提供給修改核心的程式設計師的功能，它包含了顯示核心除錯資訊。

Kernel debugging (CONFIG\_DEBUG\_KERNEL) [N/y/?]

### (25)加密選項

\* Cryptographic options

這是核心支援加密的選項。

Cryptographic API (CONFIG\_CRYPTO) [N/y/?]

\*\*\* End of Linux kernel configuration.

\*\*\* Check the top-level Makefile for additional configuration.

\*\*\* Next, you must run 'make dep'.

## 31-5 編譯和安裝核心

當用來進行配置的指令稿執行完成時，它會告訴我們去 `make dep` 以及 `clean`。所以要執行 `make dep`。這將確保所有的相依關係。然後做 `make clean`。這會清除核心編譯的所有目的檔以及其它東西。

這是產生核心的檔案相依。



```
#make dep
```

這是清除檔案。

```
#make clean
```

這是建立核心 bzImage。

```
#make bzImage
```

這是產生和安裝核心。

```
#make install
```

在完成 `make dep` 及 `make clean` 工作之後，我們可以執行 `make zImage`。`make zImage` 會編譯核心，並且在 `/usr/src/linux-2.4.22/arch/i386/boot` 留給我們一個 `zImage` 的檔案。這就是新的壓縮核心。

我們將新核心複製到/boot 目錄下的 vmlinuz-2.4.22

```
#cp /usr/src/linux-2.4.22/arch/i386/boot/bzImage /boot/vmlinuz-2.4.22
```

我們使用符號連接/boot/vmlinuz 到/boot/vmlinuz-2.4.22

```
#ln -s /boot/vmlinuz /boot/vmlinuz-2.4.22
```

可載入模組能夠節省記憶體，而且很容易設定。模組的範圍已經包含了檔案系統，乙太網路卡驅動程式，磁帶機，印表機驅動程式，還有更多。幾乎所有的東西都可以變成可載入模組。要使用可載入模組，首先確定我們沒有將其配置到正常的核心裡，也就是說，在 `make config` 時不是回答 Yes。編譯新核心並用它來開機。然後，再次 cd 進入到 /usr/src/linux-2.4.22，然後執行 `make modules`。這將編譯所有我們未指定在核心配置裡的模組，並將其連結到 /usr/src/linux-2.4.22/modules。我們可以直接由此目錄使用，或是執行 `make modules\_install` 將其安裝到 /lib/modules/核心版本。

```
#make modules
```

```
#make modules_install
```



### 31-6 載入啟動組態檔

我們可以使用 LILO (Linux Loader) 來載入新核心。`make zliilo` 將會安裝核心，然後對它執行 LILO，使我們完成所有開機的準備，但是這只有在 LILO 已經以下列這種方式配置好的時候才可以：核心為 `/vmlinuz`，`lilo` 在 `/sbin` 下，而且與我們的 LILO 配置(`/etc/lilo.conf`)一致。

```
#make zliilo
```

我們可能需要直接使用 LILO。這是一個相當容易安裝的軟體，`lilo`組態檔(`/etc/lilo.conf`)。`image` 設定為已經安裝的核心。我們將新核心設在 `/boot/vmlinuz-2.4.22`。`label` 則是給 `lilo` 用來決定現在要啟動的是那個核心或作業系統，而`root` 則是這個特別的作業系統的根目錄 `/`。`prompt`是開機時自動顯示出`lilo`開機。`timeout=50`是指定 50 秒後以指定的選項啟動電腦。`default`是指定的選項為`linux`。`boot=/dev/hda`是將`lilo`放置在第一顆硬碟。`initrd`是使用`ram`當作磁碟。`read-only`是指唯讀。

```
#vi /etc/lilo.conf
prompt
timeout=50
default=linux
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
message=/boot/message
linear

image=/boot/vmlinuz-2.4.22
    label=linux
    initrd=/boot/initrd-2.4.22.img
    read-only
    append="hdc=ide-scsi root=LABEL=/"
```



我們也可以使用 grub.conf 來載入新核心。

default=1 是載入新版的核心 linux2.4.20-20.9。timeout=10 是在 10 秒後載入核心。splashimage 是開機的載入畫面。root 是 root 分割的位置。kernel 是指定核心的位置。initrd 是指定使用虛擬記憶體當磁碟機的位置。

```
#vi /etc/grub.conf
# grub.conf generated by anaconda
#
# Note that you do not have to rerun grub after making changes to this file
# NOTICE: You do not have a /boot partition. This means that
#           all kernel and initrd paths are relative to /, eg.
#           root (hd0,0)
#           kernel /boot/vmlinuz-version ro root=/dev/hda1
#           initrd /boot/initrd-version.img
#boot=/dev/hda
default=1
timeout=10
splashimage=(hd0,0)/boot/grub/splash.xpm.gz
title Red Hat Linux (2.4.20-20.9)
    root (hd0,0)
    kernel /boot/vmlinuz-2.4.20-20.9 ro root=LABEL=/ hdc=ide-scsi
    initrd /boot/initrd-2.4.20-20.9.img
title Red Hat Linux (2.4.20-8)
    root (hd0,0)
    kernel /boot/vmlinuz-2.4.20-8 ro root=LABEL=/ hdc=ide-scsi
    initrd /boot/initrd-2.4.20-8.img
```





## 31-7 核心工具說明

這是我們在 make 編譯新核心時常用的工具和選項。

Configuration 組態工具	說明
config	文字模式的核心組態介面
menuconfig	視窗選項式的核心組態介面
xconfig	這是 X 視窗系統的核心組態介面

維護選項	說明
checkhelp	以選項來檢查組態。
checkconfig	檢查查詢遺失的標頭綁
clean	移除舊版的檔案
mrproper	執行完整的移除物件檔，包含核心組態檔.config，這通常在 patch 之前使用。

編譯選項	說明
zImage	在/usr/src/linux/arch 或 arch/i386/boot 目錄建立新核心 zImage。
install	建立和安裝新核心到我們作業系統。
zdisk	建立新核心和安裝到軟碟的磁片上。
bzImage	建立 bzImage 的核心檔。
bzdisk	建立新核心和安裝到軟碟的磁片上。

模組選項	說明
modules	建立核心模組。
modules_install	在/lib/modules 目錄下建立核心模組。



### 31-8 RAM Disks

當我們載入系統時需要特定模組，這時我們就可以使用 RAM Disk。假如我們使用光碟機開機時，這時就會使用到 RAM Disk。當我們開機時，就會從 RAM Disk 來讀取，並且載入作業系統。假如我們建立新的核心需要載入模組來啟動作業系統，我們則要新建立新的 RAM Disk 來載入這些模組。

我們可以使用 mkinitrd 的指令來建立 RAM Disk。當我們使用/sbin/mkinitrd 來建立時，就會在/boot 目錄下建立 initrd-2.4.20.img 的檔案。而 2.4.20-20.9 是在 /lib/modules 目錄下的核心模組 我們可以使用 man mkinitrd 來觀看 mkinitrd 的指令。

```
# /sbin/mkinitrd /boot/initrd-2.4.20.img 2.4.20-20.9  
[root@flash modules]# cd /lib/modules  
[root@flash modules]# ls  
2.4.20-20.9 2.4.20-8
```

這是在/etc/grub.conf 組態中設定的開機讀取檔。

```
initrd /boot/initrd-2.4.20-20.9.img
```

### 31-9 /lib/modules/kernel 版本/目錄結構

所有的核心模組都放在/lib/modules/核心版本/目錄。我們可以使用下列指令在 /usr/src/linux 目錄中來編譯模組和將它置入到相關的目錄。

```
#make modules  
#make modules_install
```

所有的模組都有.o 的副檔名，我們可以使用 lsmod 指令來觀看 Linux 載入的模組。



```
[root@flash chaiyen]# /sbin/lsmmod | more
Module          Size Used by      Not tainted
parport_pc      19076  1 (autoclean)
lp              8996   0 (autoclean)
parport         37056  1 (autoclean) [parport_pc lp]
nfsd            80176  8 (autoclean)
lockd           58704  1 (autoclean) [nfsd]
sunrpc          81564  1 (autoclean) [nfsd lockd]
autofs          13268  0 (autoclean) (unused)
8139too        18088  1
mii             3976   0 [8139too]
```

我們可以使用 `modprobe -t net` 指令來載入所有在 `/lib/modules/核心版本/net` 的模組，我們也可以使用 `modprobe -r` 來移除相關的模組。

```
#modprobe -t net
```



## 課後練習

---

1. 下列哪一個目錄通常會 soft-linked 連接到核心?

- (A). /usr/src
  - (B). /usr/src/linux
  - (C). /usr/sbin/linux
  - (D). /usr/src/linux-2.4
- 

2. 假如我們要更新核心，而我們應該使用下列何種指令?

- (A). `cd /usr/src; tar kzf linux-2.4.*.tar.gz`
  - (B). `cd /usr/src; tar zxvf linux-2.4.*.tar.gz`
  - (C). `cd /usr/src/linux; tar kzf linux-2.4.*.tar.gz`
  - (D). `cd /usr/src/linux; tar zxvf linux-2.4.*.tar.gz`
- 

3. 下列哪一些指令是建立模組核心所需要的指令? 複選

- (A). `make lilo`
  - (B). `make modules`
  - (C). `make modules_install`
  - (D). `make bzImage`
- 

4. 下列何者是正在發展 development 的核心?

- (A). 2.4.8
- (B). 2.18.5
- (C). 5.0.12
- (D). 2.3.8



5. 下列何者在系統一啟動時即被載入？其功能為控制系統硬體及所有軟體的程式、管理電腦的所有資源，為硬體和使用者間的介面，可讓使用者使用系統所提供的服務。

- (A). lilo
- (B). 作業系統的核心
- (C). grub
- (D). 記憶體

6. 有一些核心的特色可以被打開或關閉而不用經過重新安裝或編譯新的核心或模組，例如 IP 轉向。這些微調參數是由下列何者檔案所控制？我們所設定的參數都儲存在 `/etc/sysctl.conf` 的組態檔中。例如當我們在網路 IP 面版中設定啟動 IP 轉向，將設定 `/proc/sys/net/ipv4/ip_forward` 到 1。我們也可以直接使用指令 `sysctl` 來設定。我們使用 `-p` 選項，這樣 `sysctl` 將從 `/etc/sysctl.conf` 檔讀取參數。我們可以使用 `-w` 選項來改變指定參數。

- (A). `/etc/sys`
- (B). `make kernel`
- (C). `make install`
- (D). `/proc/sys`

7. 現在我們要編譯原始檔的核心，然後自己建立這已編譯的核心。原始碼的核心是由 `gcc` 編譯器來編譯，就像是其它原始檔的 C 程式碼，經過 `gcc` 編譯成二進位碼。我們現在的核心是 `linux-2.4.20`，我們現在有 `linux-2.4.22.tar.gz` 的原始檔，現在我們要將核心安裝並且昇級。這原始檔是放在 `/usr/src` 目錄，當我們下載並且安裝 `linux-2.4.22.tar.gz` 的原始檔，它會在 `/usr/src` 下建立下列哪一個目錄？我們的原始檔 `linux.2.4.22.tar.gz` 是在 `www.kernel.org` 中下載，我們可以從該網站下載最新的核心

- (A). `/usr/src/linux-2.4.22`
- (B). `B./usr/src/linux`
- (C). `C./usr/src/`
- (D). `D./usr/src/kernel`



8. 我們在文字模式輸入下列哪一個指令來組態核心的設定？

- (A). make config
- (B). make menuconfig
- (C). make module
- (D). make install

9. 我們可能需要直接使用 LILO . 這是一個相當容易安裝的軟體，lilo 組態檔 (/etc/lilo.conf)。`image =` 設定為已經安裝的核心。請問通常我們將新核心設在哪裏？

- (A). A./usr/src
- (B). B./usr/src/linux
- (C). C. /boot/vmlinuz-2.4.22
- (D). D./boot/linux

10. 當我們編譯核心時，請問下列哪一個指令我們不會用到？

- (A). make bzImage
- (B). make friend
- (C). make dep
- (D). make clean

11. 下列何者不是描述核心模組？

- (A). 核心模組可以讓核心的檔案更小.
- (B). 它們可以增加核心速度.
- (C). 我們可以動態的從正在執行的核心解除模組.
- (D). 模組的延伸檔是.dll.

12. 我們可以使用 lsmod 指令來看出 Linux 的模組支援，而在其中會出現 autoclean，



請問 autoclean 是什麼意義？

- (A). 核心將自動小心的從記憶體移除它
- (B). 這模組將自動的移除它自己的檔案
- (C). 這模組已經不會被用到了，應該被移除
- (D). 這核心將自動的增加這核心

13. 我們要用下列哪一個指令來最快速的載入我們網路卡裝置？

- (A). 從選單
- (B). `modprobe \*`
- (C). `loadmodule -t net`
- (D). `modprobe -t net`

14. 假如我們在 `/lib/modules/核心版本/目錄` 找不到相關的模組，我們可以使用下列哪一些指令在 `/usr/src/linux` 目錄中來編譯模組和將它置入到相關的目錄？

- (A). `make modules ; make modules_install`
- (B). `make dep ; make install`
- (C). `make clean ; make modules`
- (D). `make modules ; make install`

【答案】

1. D   2. B   3. B,C   4. D   5. B   6. D   7. A  
8. A   9. C   10. B   11. D   12. A   13. D   14. A



NOTE

Handwriting practice area with 18 horizontal dotted lines.

