**編譯前的任務：認識核心與取得核心原始碼**

Linux的『核心(kernel)』是整個作業系統的最底層，他負責了整個硬體的驅動，以及提供各種系統所需的核心功能，如果核心不認識某個最新的硬體，那麼該硬體也就無法被驅動，你當然也就無法使用該硬體囉！

* Kernel

當系統讀完BIOS並載入MBR內的開機管理程式後，就能夠載入核心到記憶體當中。然後核心開始偵測硬體，掛載根目錄並取得核心模組來驅動所有的硬體，之後呼叫/sbin/init就能夠依序啟動所有系統所需要的服務了！

* 核心模組 (kernel module) 的用途

核心檔案包含了硬體偵測與驅動模組，亦即是將一些不常用的類似驅動程式獨立出核心，編譯成為模組，核心可在系統正常運作的過程中載入模組到核心的支援。如此在不需要更動核心的前提下，只要編譯出適當的核心模組，並且載入他！模組放在/lib/modules/$(uname -r)/kernel/當中！

* 核心原始碼的解壓縮與放置目錄

假設你也是下載上述的連結內的檔案，然後該檔案放置到/root底下。2.6.x核心原始碼一般建議放置於/usr/src/kernels/目錄底下，因此你可以這樣處理：

|  |
| --- |
| [root@www ~]# tar -jxvf linux-2.6.30.3.tar.bz2 -C /usr/src/kernels/ |

此時會在/usr/src/kernels底下產生一個新的目錄，那就是 linux-2.6.30.3這個目錄囉！

* 核心原始碼下的次目錄

在上述核心目錄下含有哪些重要資料呢？基本上有底下這些東西：

* arch：與硬體平台有關的項目，大部分指的是CPU的類別，例如 x86, x86\_64, Xen 虛擬支援等；
* block：與區塊裝置較相關的設定資料，區塊資料通常指大量儲存媒體！包括類似ext3等檔案系統的支援是否允許等。
* crypto：核心所支援的加密的技術，例如md5或者是des等等；
* Documentation ：與核心有關的一堆說明文件，若對核心有極大的興趣，要瞧瞧這裡！
* drivers：一些硬體的驅動程式，例如顯示卡、網路卡、PCI 相關硬體等等；
* firmware：一些舊式硬體的微指令碼(韌體)資料；
* fs：核心所支援的filesystems，例如vfat, reiserfs, nfs等等；
* include ：一些可讓其他程序呼叫的標頭(header)定義資料；
* init：一些核心初始化的定義功能，包括掛載與init程式的呼叫等；
* ipc：定義 Linux 作業系統內各程序的溝通；
* kernel：定義核心的程序、核心狀態、執行緒、程序的排程(schedule)、程序的訊號(signal)等
* lib：一些函式庫；
* mm：與記憶體單元有關的各項資料，包括swap與虛擬記憶體等；
* 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；
* security：包括selinux等在內的安全性設定；
* sound：與音效有關的各項模組；
* virt：與虛擬化機器有關的資訊，目前核心支援的是KVM (Kernel base Virtual Machine)

小標題的圖示保持乾淨原始碼： make mrproper

第一次編譯，不清楚原始碼中有無保留目標檔案(\*.o)及相關設定檔存在，可透過指令清除編譯過程的目標檔案及設定檔：

|  |
| --- |
| [root@www linux-2.6.30.3]# make mrproper |

這會將以前的核心功能選擇檔案刪除，幾乎只有第一次執行核心編譯前才進行，其餘要刪除前一次編譯殘留的資料，只要下達：

|  |
| --- |
| [root@www linux-2.6.30.3]# make clean |

make clean僅會刪除類似目標檔之類的編譯過程產生的中間檔案，而不會刪除設定檔！

小標題的圖示開始挑選核心功能： make XXconfig

核心功能的挑選，最後會在/usr/src/kernels/linux-2.6.30.3/底下產生一個名為 .config 的隱藏檔，這個檔案就是 /boot/config-xxx 的檔案啦！那麼這個檔案如何建立呢？常見的方法有：([註1](http://linux.vbird.org/linux_basic/0540kernel.php#ps1))

* make menuconfig: 最常使用，是文字模式底下可以顯示類似圖形介面的方式，不需要啟動X Window就能夠挑選核心功能選單！
* make oldconfig: 透過使用已存在的./.config檔案內容，使用該檔案內的設定值為預設值，只將新版本核心內的新功能選項列出讓使用者選擇，可簡化核心功能的挑選過程！作為升級核心原始碼後的功能挑選，是非常好用的項目！
* make xconfig: 透過以Qt為圖形介面基礎功能的圖形化介面顯示，需要具有X window的支援。例如KDE就是透過Qt來設計的X Window，因此你如果在KDE畫面中，可以使用此一項目。
* make gconfig: 透過以Gtk為圖形介面基礎功能的圖形化介面顯示，需要具有X window的支援。例如 GNOME 就是透過Gtk來設計的X Window，因此你如果在 GNOME 畫面中，可以使用此一項目。
* make config: 最舊式的功能挑選方法，每個項目都以條列式的列出讓你選擇，設定錯誤只能再次選擇，不人性化！

以make menuconfig來說，出現的畫面會有點像這樣：

  
圖 2.3.1 、 make menuconfig 核心功能挑選選單示意圖

畫面主要分為兩大部分，一個是大框框內的反白光柱，另一個則是底下的小框框，裡面有 select, exit與help三個選項的內容。這幾個元件的大致用法如下：

* 『左右方向鍵』：可以移動最底下的<Select>, <Exit>, <Help>項目；
* 『上下方向鍵』：可以移動上面大框框部分的反白光柱，若該行有箭頭(--->)則表示該行內部還有其他細項需要設定；
* 選定項目：以『上下鍵』選擇好想要設定的項目之後，並以『左右鍵』選擇<Select>之後， 按下『Enter』就可以進入該項目去作更進一步的細部設定囉；
* 可挑選之功能：在細部項目的設定中，如果前有[ ]或<>符號時，該項目才可選擇，可以使用『空白鍵』來選擇；
* 若為[\*] <\*>則表示編譯進核心；若為<M>則表示編譯成模組！在不知道該項目為何，且有模組可以選，那麼就可以直接選擇為模組囉！
* 當在細項目選擇<Exit>後，並按下Enter，那麼就可以離開該細部項目囉！

小標題的圖示核心功能細項選擇

* General setup

Linux最相關的程序互動、核心版本說明、是否使用發展中程式碼等資訊都在這設定。這裡的項目主要針對核心與程式間的相關性來設計的，基本上保留預設值即可！不要隨便取消任何一個項目，可能會造成某些程式無法被同時執行的困境！

|  |
| --- |
| [ ] Prompt for development and/or incomplete code/drivers  # 這個建議不要選擇，因為我們不是核心專家，不需要使用發展中或不完整的程式碼！  (vbird) Local version - append to kernel release  [\*] Automatically append version information to the version string  # 我希望我的核心版本成為 2.6.30.3.vbird ，那這裡可以就這樣設定！  Kernel compression mode (Bzip2) --->  # 建議選擇成為 Bzip2 即可，因為壓縮比較佳！  [\*] Support for paging of anonymous memory (swap)  # 任何人均可存取 swap 是合理的！所以這裡務必要勾選！  [\*] System V IPC  # IPC 是 Inter Process Communication (程序通訊) 縮寫，與程序溝通有關，要選！  [\*] BSD Process Accounting  [ ] BSD Process Accounting version 3 file format  # 與標準 Unix (BSD) 的程序支援有關，但不要支援 version 3 ，可能有相容性問題  [ ] Export task/process statistics through netlink (EXPERIMENTAL)  # 這個額外的進階選項可以將他取消的！  [\*] Auditing support  [\*] Enable system-call auditing support  # 上面這兩個是額外核心功能 (如 SELinux) 載入時所需要的設定！務必選擇  RCU Subsystem --->  RCU Implementation (Classic RCU) --->  # 選擇標準 RCU 即可，不需要使用大量 CPU 的整合功能。  <M> Kernel .config support  [ ] Enable access to .config through /proc/config.gz (NEW)  # 讓 .config 這個核心功能列表可以寫入實際的核心檔案中！  (17) Kernel log buffer size (16 => 64KB, 17 => 128KB)  [ ] Control Group support (NEW) --->  # 整合 CPU 或分離裝置的功能，屬於進階設定，我們先不要使用這功能。  [\*] Create deprecated sysfs layout for older userspace tools (NEW)  # 如果使用支援舊式裝置，如 /sys/devices 者，這裡要勾選！但如果是 2008  # 年後的 distribution ，這裡可能需要取消喔！ CentOS 5.x 要選的！  -\*- Kernel->user space relay support (formerly relayfs)  -\*- Namespaces support  [\*] UTS namespace (NEW)  [\*] IPC namespace (NEW)  # 使用 uname 時，會輸出較多的資訊，所以可以嘗試選擇看看。  [\*] Initial RAM filesystem and RAM disk (initramfs/initrd) support  () Initramfs source file(s)  # 這是一定要的！因為要支援開機時載入 initail RAM disk 嘛！  [\*] Optimize for size  # 可以減低核心的檔案大小，其實是 gcc 參數使用 -Os 而不是 -O2  [ ] Configure standard kernel features (for small systems) --->  # 給嵌入式系統使用的，我們用 PC ，所以這裡不選。  [ ] Strip assembler-generated symbols during link (NEW)  [ ] Disable heap randomization (NEW)  # 2000 年後推出的版本，可以取消這個項目！  Choose SLAB allocator (SLAB) --->  [\*] Profiling support (EXPERIMENTAL)  [ ] Activate markers (NEW)  <M> OProfile system profiling (EXPERIMENTAL)  [ ] OProfile AMD IBS support (EXPERIMENTAL) (NEW)  [\*] Kprobes |

* loadable module + block layer

要讓你的核心能夠支援動態的核心模組，那麼底下的第一個設定就得要啟動才行！至於第二個block layer則預設是啟動的， 你也可以進入該項目的細項設定，選擇其中你認為需要的功能即可！

|  |
| --- |
| [\*] Enable loadable module support ---> <==底下為細項  --- Enable loadable module support  [ ] Forced module loading <==大概就是這個不要選，其他的都選起來！  [\*] Module unloading  [\*] Module versioning support  [\*] Source checksum for all module  ===============================================  -\*- Enable the block layer ---> <==看吧！預設就是已經選擇了！底下為細項  [ ] Block layer data integrity support <==特殊儲存裝置支援，可以不選  IO Schedulers --->  <\*> Anticipatory I/O scheduler <==較複雜的一種I/O排程  <\*> Deadline I/O scheduler <==較適用於 database 的載入  <\*> CFQ I/O scheduler <==較適用於 desktop 的環境  Default I/O scheduler (Deadline) ---> <==適用於鳥哥環境 |

* CPU 的類型與功能選擇

進入『Processor type and features』後，挑選主機的實際CPU形式。這裡使用的是Athlon 64的CPU，且主機還有啟動 Xen虛擬化的服務(在一部主機上面同時啟動多個作業系統)，因此底下的選擇是這樣的：

|  |
| --- |
| [\*] Tickless System (Dynamic Ticks) <==可增加些許省電功能  [ ] High Resolution Timer Support  [\*] Symmetric multi-processing support <==多核心 CPU 環境必選  [ ] Support sparse irq numbering  [\*] Enable MPS table <==讓多 CPU 支援 ACPI  [ ] Support for extended (non-PC) x86 platforms  [\*] Single-depth WCHAN output  [\*] Paravirtualized guest support ---> <==支援半虛擬化功能  --- Paravirtualized guest support <==底下為 Xen 與 KVM 兩種虛擬機器支援！  [\*] Xen guest support  (32) Maximum allowed size of a domain in gigabytes  [\*] Enable Xen debug and tuning parameters in debugfs  [\*] KVM paravirtualized clock  [\*] KVM Guest support  -\*- Enable paravirtualization code  ==================================================  [ ] paravirt-ops debugging (NEW) <==不需要具有 debug 的功能  [ ] Memtest  Processor family (Opteron/Athlon64/Hammer/K8) ---> <==要選對啊！  [\*] AMD IOMMU support <==啟動 AMD 的 IOMMU 功能！  (8) Maximum number of CPUs  [ ] SMT (Hyperthreading) scheduler support <==Intel CPU 的超執行緒功能  [\*] Multi-core scheduler support <==多核心功能的支援  Preemption Model (No Forced Preemption (Server)) --->  # 這是與程序有關的設定項目，鳥哥這裡建立 Server 主機，因此選這項！  # 如果是桌上型電腦的使用，建議進入選擇 desktop 項目。  [ ] Reroute for broken boot IRQs  [\*] Machine Check Exception <==可將核心偵測之錯誤回報到終端機顯示！  [\*] Intel MCE features (NEW)  [\*] AMD MCE features (NEW)  < > Dell laptop support  <M> /dev/cpu/microcode - microcode support  [ ] Intel microcode patch loading support  [\*] AMD microcode patch loading support  <M> /dev/cpu/\*/msr - Model-specific register support  <\*> /dev/cpu/\*/cpuid - CPU information support  < > /sys/kernel/debug/x86/cpu/\* - CPU Debug support  [ ] Numa Memory Allocation and Scheduler Support  Memory model (Sparse Memory) --->  [\*] Sparse Memory virtual memmap <==可強化些許核心效能  [ ] Allow for memory hot-add  [\*] Add LRU list to track non-evictable pages  (65536) Low address space to protect from user allocation  [ ] Check for low memory corruption  [\*] Reserve low 64K of RAM on AMI/Phoenix BIOSen <==重新偵測 BIOS 資訊  [\*] MTRR (Memory Type Range Register) support  # 可以讓 CPU 具有讀取記憶體特殊區塊的能力，尤其在高效能的顯示卡方面，  # 可以增進不少效能。這個項目會產生 /proc/mtrr ，X 會讀取這個咚咚喔。  [\*] MTRR cleanup support  (0) MTRR cleanup enable value (0-1)  (1) MTRR cleanup spare reg num (0-7)  [ ] x86 PAT support  [ ] EFI runtime service support  [\*] Enable seccomp to safely compute untrusted bytecode  Timer frequency (300 HZ) --->  # 這個項目則與核心針對某個事件立即回應的速度有關。Server 用途可以調整到  # 300Hz 即可，如果是桌上型電腦使用，需要調整高一點，例如 1000Hz 較佳！  [\*] kexec system call  [ ] kernel crash dumps  -\*- Support for hot-pluggable CPUs  [ ] Compat VDSO support <==舊式功能，可以不要選擇  [ ] Built-in kernel command line <==正常開機選單 (grub) 環境，不需要此項功能 |

* 電源管理功能

選擇『Power management and ACPI options』會進入系統的電源管理機制中。電源管理機制還要搭配主機板以及CPU的相關省電功能，才能夠達到省電的效率！

|  |
| --- |
| [\*] Power Management support  [ ] Power Management Debug Support  [\*] Suspend to RAM and standby  [ ] Hibernation (aka 'suspend to disk')  [\*] ACPI (Advanced Configuration and Power Interface) Support --->  # 這是個較新的電源管理模組，由於選擇後會增加核心約 70K ，所以  # 對嵌入式系統來說，可能需要考慮考慮。至於 desktop/server 當然就選擇啊  --- ACPI (Advanced Configuration and Power Interface) Support  [ ] Deprecated /proc/acpi files  [\*] Deprecated power /proc/acpi directories  [\*] Future power /sys interface  [\*] Deprecated /proc/acpi/event support  <M> AC Adapter  <M> Battery  <M> Button  -M- Video  <\*> Fan  <\*> Processor  <\*> Thermal Zone  [ ] Debug Statements  <M> PCI slot detection driver  <M> Smart Battery System  =======================================================  CPU Frequency scaling --->  # 可以經過核心修改 CPU 的運作頻率，在說明檔當中也提及，還需要啟動底下的  # dynamic cpufreq governor 才可以順利的啟動這個項目。  [\*] CPU Frequency scaling  [\*] Enable CPUfreq debugging  <M> CPU frequency translation statistics  [\*] CPU frequency translation statistics details  Default CPUFreq governor (userspace) --->  -\*- 'performance' governor  <M> 'powersave' governor  <M> 'userspace' governor for userspace frequency scaling  <M> 'ondemand' cpufreq policy governor  -\*- 'conservative' cpufreq governor  \*\*\* CPUFreq processor drivers \*\*\*  <M> ACPI Processor P-States driver  <\*> AMD Opteron/Athlon64 PowerNow! <==因為我們是 AMD 的 CPU 啊！  <M> Intel Enhanced SpeedStep (deprecated)  < > Intel Pentium 4 clock modulation  ==========================================================  -\*- CPU idle PM support  Memory power savings ---> |

* 一些匯流排(bus)的選項

這個項目則與匯流排有關！分為最常見的PCI與PCI-express的支援，還有筆記型電腦常見的PCMCIA插卡！要記住的是，那個PCI-E的介面務必要選取！不然新顯示卡可能會捉不到！

|  |
| --- |
| [\*] PCI support  [\*] Support mmconfig PCI config space access  [\*] PCI Express support  <M> PCI Express Hotplug driver  [\*] Root Port Advanced Error Reporting support  -\*- Message Signaled Interrupts (MSI and MSI-X)  [\*] Enable deprecated pci\_find\_\* API  [ ] PCI Debugging  <M> PCI Stub driver  [\*] Interrupts on hypertransport devices  [\*] PCI IOV support <==與虛擬化有關！請加選此項！  < > PCCard (PCMCIA/CardBus) support ---> <==鳥哥的主機不是 notebook，所以不選。  <\*> Support for PCI Hotplug ---> <==不關機情況下，熱拔插 PCI 裝置  --- Support for PCI Hotplug  <M> Fake PCI Hotplug driver  <M> ACPI PCI Hotplug driver  <M> ACPI PCI Hotplug driver IBM extensions  [ ] CompactPCI Hotplug driver  <M> SHPC PCI Hotplug driver |

* 編譯後執行檔的格式

底下『Executable file formats / Emulations』的選項必須要勾選才行！因為是給Linux核心運作執行檔之用的資料。通常是與編譯行為有關啦！

|  |
| --- |
| [\*] Kernel support for ELF binaries  [ ] Write ELF core dumps with partial segments  <\*> Kernel support for MISC binaries  [\*] IA32 Emulation <==因為我們這裡是64位元，因此32位元為模擬結果  <M> IA32 a.out support |

* 核心的網路功能

『Networking support』項目是重要的選項，因為還包含了防火牆相關的項目！大部分的參數都與網路、防火牆有關！由於防火牆是在啟動網路之後再設定即可，所以大部分的內容都可被編譯成為模組！有用到再載入到核心即可啊！

|  |
| --- |
| --- Networking support  Networking options --->  # 裡面的資料全都是重要的防火牆項目！盡量編成模組！不曉得功能的部分，就盡量保留預設值即可！  <\*> Packet socket <==網路封包，當然要選擇啊！  [\*] Packet socket: mmapped IO  <\*> Unix domain sockets <==Unix 插槽檔，也一定要選擇啊！  <\*> Transformation user configuration interface  <M> PF\_KEY sockets  [\*] TCP/IP networking <==能不選擇 TCP/IP 嗎？  [\*] IP: multicasting  [\*] IP: advanced router  Choose IP: FIB lookup algorithm (FIB\_HASH) --->  [\*] IP: policy routing  [\*] IP: equal cost multipath  [\*] IP: verbose route monitoring  [ ] IP: kernel level autoconfiguration  <M> IP: tunneling  <M> IP: GRE tunnels over IP  [\*] IP: broadcast GRE over IP  [\*] IP: multicast routing  [\*] IP: PIM-SM version 1 support  [\*] IP: PIM-SM version 2 support  [\*] IP: TCP syncookie support (disabled per default)  <M> IP: AH transformation  <M> IP: ESP transformation  <M> IP: IPComp transformation  <M> IP: IPsec transport mode  <M> IP: IPsec tunnel mode  <\*> IP: IPsec BEET mode  -\*- Large Receive Offload (ipv4/tcp)  <M> INET: socket monitoring interface  [\*] TCP: advanced congestion control ---> <==內部細項全為模組  <M> The IPv6 protocol ---> <==除必選外，內部細項全為模組  [\*] NetLabel subsystem support  -\*- Security Marking  [\*] Network packet filtering framework (Netfilter) --->  # 這個就是我們一直講的防火牆部分！裡面細項幾乎全選擇成為模組！  --- Network packet filtering framework (Netfilter)  [ ] Network packet filtering debugging <==debug 部分不選！  [\*] Advanced netfilter configuration  [\*] Bridged IP/ARP packets filtering  Core Netfilter Configuration --->  <M> IP virtual server support --->  IP: Netfilter Configuration --->  IPv6: Netfilter Configuration --->  <M> Ethernet Bridge tables (ebtables) support --->  # 上面的細項，除了必選外其他的都編成模組喔！原始沒選的也請選為模組  ============================================================  <M> Asynchronous Transfer Mode (ATM)  <M> Classical IP over ATM  [ ] Do NOT send ICMP if no neighbour  <M> LAN Emulation (LANE) support  < > Multi-Protocol Over ATM (MPOA) support  <M> RFC1483/2684 Bridged protocols  [ ] Per-VC IP filter kludge  <M> 802.1d Ethernet Bridging  <M> 802.1Q VLAN Support  [ ] GVRP (GARP VLAN Registration Protocol) support  <M> DECnet Support  <M> ANSI/IEEE 802.2 LLC type 2 Support  [ ] IPX: Full internal IPX network (NEW)  <M> Appletalk protocol support  < > Appletalk interfaces support  <M> Phonet protocols family  [\*] QoS and/or fair queueing ---> <==內容同樣全為模組！  [ ] Data Center Bridging support  Network testing ---> <==保留成模組預設值  =========================================================  # 底下的則是一些特殊的網路設備，例如紅外線啊、藍芽啊！ 如果不清楚就使用模組！除非你真的知道不要該項目！  [ ] Amateur Radio support --->  < > CAN bus subsystem support --->  < > IrDA (infrared) subsystem support --->  <M> Bluetooth subsystem support --->  # 這個是藍芽支援，同樣的，裡面除了必選之外，其他通通挑選成為模組！  [\*] Wireless --->  # 這個則是無線網路設備，裡面保留預設值，但可編成模組的就選模組  <M> WiMAX Wireless Broadband support --->  # 新一代的無線網路，也請勾選成為模組！  {M} RF switch subsystem support ---> |

* 各項裝置的驅動程式

進入『Device Drivers』是所有硬體裝置的驅動程式庫！還是得要一個項目一個項目的去挑選挑選才行～很多資料都與你的硬體有關。核心推出時的預設值是比較符合一般狀態的，所以很多資料其實保留預設值就可以編的很不錯了！

|  |
| --- |
| Generic Driver Options ---> <==與韌體有關，保留預設值即可  <\*> Connector - unified userspace <-> kernelspace linker --->  # 與使用者/核心層級的資訊溝通有關，務必要選擇啊！  <M> Memory Technology Device (MTD) support --->  # 例如快閃記憶體(拇指碟之類)之支援，通常與嵌入式系統有關！  # 但由於我們也會用到隨身碟，所以裡面的資料全編為模組！  <M> Parallel port support --->  # 平行序列埠的支援，例如早期的 25 針印表機與 9 針滑鼠等，細項全編為模組！  -\*- Plug and Play support ---> <==不囉唆！當然要選擇這個項目！  [\*] Block devices ---> <==區塊裝置，就是一些儲存媒體！細項內容請全編為模組  [\*] Misc devices ---> <==一些較冷門的設備，建議還是全部編為模組！  <\*> ATA/ATAPI/MFM/RLL support ---> <==IDE 介面相關的晶片組！  # 這個其實與主機板的南橋晶片有關！由於鳥哥的主機為 ALi 的板子，所以：  <\*> ALI M15x3 chipset support  # 除了可以保留預設值之外，你也可以將沒用到的驅動程式取消選擇。較重要的還有：  [ ] Support for SATA (deprecated; conflicts with libata SATA driver)  # 這個一定不能選！因為 SATA 的模組是在 SCSI 中！  <\*> Include IDE/ATAPI CDROM support  # IDE 的 CDROM 最好直接編譯進核心！  # 其餘的驅動程式鳥哥幾乎都選擇成為模組了！沒用到的晶片也將 \* 也改成 M 哩！  ==============================================================  SCSI device support --->  # 這部份是 SCSI 儲存媒體的驅動程式！請一定要選擇！因為：  # 1. 因為 USB 裝置用的就是模擬 SCSI 啊！ 2. 因為 SATA 的設定項目就在這裡面！  <M> RAID Transport Class  {M} SCSI device support  [\*] legacy /proc/scsi/ support  \*\*\* SCSI support type (disk, tape, CD-ROM) \*\*\*  <M> SCSI disk support <==幾乎全編為模組即可！  <M> SCSI tape support  <M> SCSI OnStream SC-x0 tape support  <M> SCSI CDROM support  [\*] Enable vendor-specific extensions (for SCSI CDROM)  <M> SCSI generic support  <M> SCSI media changer support  <M> SCSI Enclosure Support  \*\*\* Some SCSI devices (e.g. CD jukebox) support multiple LUNs \*\*\*  [\*] Probe all LUNs on each SCSI device  [\*] Verbose SCSI error reporting (kernel size +=12K)  [\*] SCSI logging facility  [\*] Asynchronous SCSI scanning  SCSI Transports ---> <==細項保留預設值  [\*] SCSI low-level drivers ---> <==主要是磁碟陣列卡，細項可全選為模組  <M> SCSI Device Handlers ---> <==細項全選為模組  < > OSD-Initiator library  ======================================================  <M> Serial ATA (prod) and Parallel ATA (experimental) drivers --->  # SATA之類的磁碟驅動程式！這裡的模組與SCSI模組是有相依屬性的關係！  # 底下的細項全部選擇模組，尤其是ALi的這個項目，對鳥哥來說，是一定要勾選的  <M> ALi PATA support  [\*] Multiple devices driver support (RAID and LVM) --->  # RAID 與 LVM 怎可不選！我們[第十五章](http://linux.vbird.org/linux_basic/0420quota.php)才講過這東西！細項均保留預設值即可  [ ] Fusion MPT device support --->  # 一種高階的 SCSI 控制器，可選可不選！因為鳥哥這裡不會用到，所以不選！  IEEE 1394 (FireWire) support --->  # 這個俗稱『火線』，許多外接式設備可能會用這個介面，因此，在此部分內的細項部分，請務必設定為模組！  <M> I2O device support ---> <==細項亦全選為模組！  [ ] Macintosh device drivers ---> <==我們是 PC ，所以不需支援麥金塔周邊  [\*] Network device support ---> <==網路設備的支援是必選！  --- Network device support  [\*] Enable older network device API compatibility  <M> Intermediate Functional Block support  <M> Dummy net driver support  <M> Bonding driver support  <M> EQL (serial line load balancing) support  <M> Universal TUN/TAP device driver support  <M> Virtual ethernet pair device  <M> General Instruments Surfboard 1000  < > ARCnet support ---> <==較早期的網卡規格，可不選擇！  {M} PHY Device support and infrastructure ---> <==細項全為模組  [\*] Ethernet (10 or 100Mbit) --->  [\*] Ethernet (1000 Mbit) --->  [\*] Ethernet (10000 Mbit) --->  # 上面三個乙太網路網卡支援，不論是否用的到，細項請全編為模組來待命吧！  < > Token Ring driver support ---> <==IBM 的 LAN ，可不選！  Wireless LAN --->  WiMAX Wireless Broadband devices --->  USB Network Adapters --->  # 上面三個為現階段很熱門的無線網路設備，所以全部內容的細項全選擇  # 為模組！免得未來你的主機加上新的無線設備時會找不到驅動程式！  [ ] Wan interfaces support ---> <==WAN 的廣域網路設備應該就不用選擇了！  [ ] ATM drivers ---> <==高階的 ATM 設備也不用選吧！  <\*> Xen network device frontend driver  <\*> FDDI driver support  <M> Digital DEFTA/DEFEA/DEFPA adapter support  [ ] Use MMIO instead of PIO (NEW)  <M> SysKonnect FDDI PCI support  <M> PLIP (parallel port) support  <M> PPP (point-to-point protocol) support  [\*] PPP filtering  <M> PPP support for async serial ports  <M> PPP support for sync tty ports  <M> PPP Deflate compression  <M> PPP BSD-Compress compression  <M> PPP over ATM  # 如果你有 ADSL 撥接的話，呵呵！PPP 的裝置也要選擇上喔！  <M> SLIP (serial line) support  [\*] CSLIP compressed headers  [\*] Keepalive and linefill  [ ] Six bit SLIP encapsulation  [\*] Fibre Channel driver support  ============================================================  [ ] ISDN support --->  < > Telephony support --->  # 這兩個設備沒用到，所以也可以不要選擇！  Input device support --->  # 這裡面含有滑鼠、鍵盤、搖桿、觸控版等輸入裝置，盡量全選為模組吧！  Character devices --->  # 周邊元件設備部分，也全選為模組吧！  {M} I2C support --->  # 測主機板的溫度與壓力是透過核心的I2C的模組功能！ALi 預設沒有被編入核心，所以請進入選擇成模組！  [ ] SPI support --->  [ ] GPIO Support --->  < > Dallas's 1-wire support --->  -\*- Power supply class support --->  # 絕大部分都沒有用到的咚咚，所以保留預設值，不選擇！  <M> Hardware Monitoring support --->  # 硬體偵測器的支援，記得也要挑選，然後內容全為模組！  -\*- Generic Thermal sysfs driver --->  [\*] Watchdog Timer Support ---> <==需搭配 watchdog 服務  # 若搭配 watchdog 服務，可以設定在某些特定狀況下重新啟動主機！  Sonics Silicon Backplane --->  Multifunction device drivers --->  # 鳥哥沒有這樣的設備，所以也沒有選擇！  [ ] Voltage and Current Regulator Support --->  Multimedia devices --->  # 一堆多媒體裝置如影像擷取卡、FM 廣播音效卡。但如果你的 Linux 是桌上型電腦，  # 裡面需要挑選成模組較佳！因為一大堆多媒體介面卡！  Graphics support ---> <==這就重要了！顯示卡選擇！  # 顯示卡的晶片組～前面提到的是主機板的對顯示卡的匯流排支援(PCI-E與AGP)，這裡是針對顯示卡晶片！  # 鳥哥的顯示卡是 NVidia的，所以將他選擇即可！其他的可以編成模組！  <M> Sound card support --->  # 音效卡部分，也全部選擇成為模組啦！  [\*] HID Devices ---> <==人機介面裝置，保留預設值即可(也可不選)  [\*] USB support --->  # 不能不選的 USB ，內容也全部是模組即可！尤其底下這三個：  <M> EHCI HCD (USB 2.0) support  <M> OHCI HCD support  <M> UHCI HCD (most Intel and VIA) support  <M> MMC/SD/SDIO card support ---> <==多媒體介面卡，保留預設值  < > Sony MemoryStick card support (EXPERIMENTAL) --->  -\*- LED Support --->  [ ] Accessibility support --->  <M> InfiniBand support ---> <==高階網路設備  [\*] EDAC - error detection and reporting --->  <M> Real Time Clock ---> <==內容選為模組吧！  [ ] DMA Engine support --->  [ ] Auxiliary Display support --->  < > Userspace I/O drivers --->  [\*] Xen memory balloon driver  [\*] Scrub pages before returning them to system  <\*> Xen filesystem  [\*] Create compatibility mount point /proc/xen  [ ] Staging drivers --->  [ ] X86 Platform Specific Device Drivers --->  # 一堆筆記型電腦的驅動，可以不選啦！ |

底下則與 Firmware Drivers 有關喔！基本上，都保留預設值就好了！

|  |
| --- |
| <M> BIOS Enhanced Disk Drive calls determine boot disk  [ ] Sets default behavior for EDD detection to off (NEW)  <M> BIOS update support for DELL systems via sysfs  <M> Dell Systems Management Base Driver  [\*] Export DMI identification via sysfs to userspace  [\*] iSCSI Boot Firmware Table Attributes  <M> iSCSI Boot Firmware Table Attributes module |

* 檔案系統的支援

如果不支援某個檔案系統，那麼Linux kernel就無法認識，也無法使用！例如Quota, NTFS等特殊的filesystem。因為涉及核心是否能夠支援某些檔案系統，以及某些作業系統支援的partition table項目。在選擇時，也務必要特別的小心在意喔！

|  |
| --- |
| <\*> Second extended fs support  [\*] Ext2 extended attributes  [\*] Ext2 POSIX Access Control Lists  [\*] Ext2 Security Labels  [\*] Ext2 execute in place support  <\*> Ext3 journalling file system support <==建議這裡直接編進核心  [ ] Default to 'data=ordered' in ext3 (legacy option)  [\*] Ext3 extended attributes  [\*] Ext3 POSIX Access Control Lists  [\*] Ext3 Security Labels  <M> The Extended 4 (ext4) filesystem  [\*] Enable ext4dev compatibility  [\*] Ext4 extended attributes (NEW)  [\*] Ext4 POSIX Access Control Lists  [\*] Ext4 Security Labels  # 上面是傳統的 EXT2/EXT3 及進階的 EXT4 支援！除了 EXT4 外，其他編入核心吧！  ==========================================================  [ ] JBD (ext3) debugging support  [ ] JBD2 (ext4) debugging support (NEW)  <M> Reiserfs support  [ ] Enable reiserfs debug mode (NEW)  [ ] Stats in /proc/fs/reiserfs (NEW)  [ ] ReiserFS extended attributes (NEW)  < > JFS filesystem support  <M> XFS filesystem support  [\*] XFS Quota support  [\*] XFS POSIX ACL support  [\*] XFS Realtime subvolume support  < > OCFS2 file system support  [\*] Dnotify support  [\*] Inotify file change notification support  [\*] Inotify support for userspace  [\*] Quota support  [ ] Report quota messages through netlink interface  [\*] Print quota warnings to console (OBSOLETE)  < > Old quota format support  <\*> Quota format v2 support  <M> Kernel automounter support  <M> Kernel automounter version 4 support (also supports v3)  < > FUSE (Filesystem in Userspace) support  # XFS 以及 Reiserfs 與 Quota 建議也是選擇起來放啦！  ============================================================  Caches --->  CD-ROM/DVD Filesystems ---> <==CD內的檔案格式，預設值即可  DOS/FAT/NT Filesystems ---> <==有支援 NTFS ，要進入挑挑！  <M> MSDOS fs support  <M> VFAT (Windows-95) fs support  (950) Default codepage for FAT <==支援繁體中文  (utf8) Default iocharset for FAT <==支援萬國碼  <M> NTFS file system support  [ ] NTFS debugging support (NEW)  [\*] NTFS write support  ======================================================  Pseudo filesystems ---> <==類似 /proc ，保留預設值  [\*] Miscellaneous filesystems ---> <==其他檔案系統的支援，保留預設值  [\*] Network File Systems ---> <==網路檔案系統！很重要！也要挑挑！  --- Network File Systems  <M> NFS client support  [\*] NFS client support for NFS version 3  [\*] NFS client support for the NFSv3 ACL protocol extension  <M> NFS server support  [\*] NFS server support for NFS version 3  [\*] NFS server support for the NFSv3 ACL protocol extension  < > SMB file system support (OBSOLETE, please use CIFS)  <M> CIFS support (advanced network filesystem, SMBFS successor)  # 最重要就這幾項，其他保留預設值即可！  =========================================================  Partition Types ---> <==分割類型，也是保持預設值即可！  -\*- Native language support ---> <==選擇預設的語系  --- Native language support  (utf8) Default NLS Option  <\*> Traditional Chinese charset (Big5)  # 除了上述這兩個之外，其他的請選擇成為模組即可！ |

* 核心駭客、資訊安全、密碼應用

『Kernel hacking』與核心開發者比較有關，建議保留預設值即可！『Security Options』是屬於資訊安全方面，包括SELinux細部權限強化模組也在這裡編入核心的。另外還有『Cryptographic API』密碼應用程式介面工具選項，也是可以保留預設值！

|  |
| --- |
| Security options --->  [\*] Enable access key retention support  [\*] Enable the /proc/keys file by which keys may be viewed  [\*] Enable different security models  [ ] Enable the securityfs filesystem  [\*] Socket and Networking Security Hooks  [\*] XFRM (IPSec) Networking Security Hooks  [ ] Security hooks for pathname based access control  [ ] File POSIX Capabilities  [ ] Root Plug Support  [\*] NSA SELinux Support  [\*] NSA SELinux boot parameter  (1) NSA SELinux boot parameter default value  [\*] NSA SELinux runtime disable  [\*] NSA SELinux Development Support  [\*] NSA SELinux AVC Statistics  (1) NSA SELinux checkreqprot default value  [ ] NSA SELinux maximum supported policy format version  [ ] Simplified Mandatory Access Control Kernel Support  [ ] TOMOYO Linux Support  [ ] Integrity Measurement Architecture(IMA)  # 基本上，這部分保留預設值就對了！你也會發現 NSA 的資料都是直接編進核心！  ================================================================  Cryptographic API --->  # 基本上，除了底下這兩個編譯進核心之外，其他的通通選擇成為模組吧！  {\*} MD5 digest algorithm  {\*} SHA1 digest algorithm |

在密碼應用程式介面方面，一般使用的帳號密碼登入就是MD5加密機制，核心有支援才行！幾乎所有的項目都給他做成模組即可！不過MD5與SHA1必須要直接由核心支援比較好！

* 虛擬化與函式庫

電腦的能力太強常閒置，可透過虛擬化技術在一部主機上面同時啟動多個作業系統來運作，這就是虛擬化。Linux核心已納入虛擬化功能！而Linux認可的虛擬化機制為KVM(Kernel base Virtual Machine)。常用的核心函式庫也可全部編為模組！

|  |
| --- |
| [\*] Virtualization --->  --- Virtualization  <M> Kernel-based Virtual Machine (KVM) support  <M> KVM for Intel processors support  <M> KVM for AMD processors support  [ ] KVM trace support (NEW)  <M> Virtio balloon driver (EXPERIMENTAL)  ==========================================================  Library routines --->  {M} CRC-CCITT functions  {M} CRC16 functions  {M} CRC calculation for the T10 Data Integrity Field  {M} CRC ITU-T V.41 functions  -\*- CRC32 functions  <M> CRC7 functions  {\*} CRC32c (Castagnoli, et al) Cyclic Redundancy-Check |

剛剛所做的設定只要在離開時選擇SAVE ，這些項目通通會記錄到目前這個目錄下的 .config檔案內。

大標題的圖示核心的編譯與安裝

將最複雜的核心功能選擇完畢後，接下來就是進行這些核心、核心模組的編譯了！底下我們就來處理處理：

小標題的圖示編譯核心與核心模組

核心與核心模組需要先編譯起來，可以先使用『make help』去查閱一下所有可用編譯參數，就會知道有底下這些基本功能：

|  |
| --- |
| [root@www linux-2.6.30.3]# make vmlinux <==未經壓縮的核心  [root@www linux-2.6.30.3]# make modules <==僅核心模組  [root@www linux-2.6.30.3]# make bzImage <==經壓縮過的核心(預設)  [root@www linux-2.6.30.3]# make all <==進行上述的三個動作 |

在/boot/下的核心檔案是壓縮過的，其中bzImage第三個字母是英文大寫的I！bzImage可以製作出壓縮過後的核心，也就是一般我們拿來進行系統開機的資訊！所以，基本上我們會進行的動作是：

|  |
| --- |
| [root@www linux-2.6.30.3]# make clean <==先清除暫存檔  [root@www linux-2.6.30.3]# make bzImage <==先編譯核心  [root@www linux-2.6.30.3]# make modules <==再編譯模組 |

最後製作出來的資料是被放置在 /usr/src/kernels/linux-2.6.30.3/這個目錄下，還沒有被放到系統的相關路徑中喔！最後注意到，下達了make bzImage 後，最終的結果應該會像這樣：

|  |
| --- |
| Root device is (8, 1)  Setup is 12696 bytes (padded to 12800 bytes).  System is 2207 kB  CRC 7701ab0e  Kernel: arch/x86/boot/bzImage is ready (#1)  [root@www linux-2.6.30.3]# ll arch/x86/boot/bzImage  -rw-r--r-- 1 root root 2272432 7月 30 13:35 arch/x86/boot/bzImage |

編譯好的核心在/usr/src/kernels/linux-2.6.30.3/arch/x86/boot/bzImage！然後是編譯模組的部分～make modules完畢後，就等著安裝！

小標題的圖示實際安裝模組

那麼如何安裝模組到正確的目標目錄呢？很簡單，同樣使用 make 的功能即可：

|  |
| --- |
| [root@www linux-2.6.30.3]# make modules\_install  [root@www linux-2.6.30.3]# ll /lib/modules/  drwxr-xr-x 3 root root 4096 7月 30 14:31 2.6.30.3vbird |

最終在/lib/modules下建立起核心的相關模組！模組已經處理妥當～接下來是準備核心的安裝了！又跟grub有關～

大標題的圖示額外(單一)核心模組編譯

核心支援的功能中，有直接編譯到核心內部的，也有外掛模組(驅動程式)！核心模組依據不同的版本，被分別放置到/lib/modules/$(uname -r)/kernel/目錄中，硬體的驅動程式則放在/lib/modules/$(uname -r)/kernel/drivers/中！

小標題的圖示編譯前注意事項

核心原始碼是放置在/usr/src/底下，如果有多個核心在一個Linux系統中， 2.6版以後，核心設計原始碼放置目錄，是以/lib/modules/$(uname -r)/build及/lib/modules/$(uname -r)/source這兩個連結檔來指向正確的核心原始碼放置目錄。以kernel 2.6.30.3 建立的核心模組來說:

|  |
| --- |
| [root@www ~]# ll -h /lib/modules/2.6.30.3vbird/  lrwxrwxrwx 1 root root 31 7月 30 14:29 build -> /usr/src/kernels/linux-2.6.30.3  drwxr-xr-x 10 root root 4.0K 7月 30 14:30 kernel  -rw-r--r-- 1 root root 337K 7月 30 14:31 modules.alias  -rw-r--r-- 1 root root 69 7月 30 14:31 modules.ccwmap  -rw-r--r-- 1 root root 224K 7月 30 14:31 modules.dep  ....(中間省略)....  lrwxrwxrwx 1 root root 31 7月 30 14:29 source -> /usr/src/kernels/linux-2.6.30.3 |

* 硬體開發商提供的額外模組

如果硬體開發商有提供驅動程式的話，下載該原始碼重新編譯，放置到核心模組該放置的地方後就能夠使用了：

|  |
| --- |
| # 1. 將檔案解壓縮：  [root@www ~]# cd /usr/local/src  [root@www src]# tar -jxvf /root/r8168-8.013.00.tar.bz2  [root@www src]# cd r8168-8.013.00/  # 2. 開始進行編譯與安裝：  [root@www r8168-8.013.00]# vi readme <==注意查一下該檔案內容  [root@www r8168-8.013.00]# make clean modules  [root@www r8168-8.013.00]# ll src/\*.ko <==建立底下的模組檔！  -rw-r--r-- 1 root root 112216 7月 31 01:11 src/r8168.ko  [root@www r8168-8.013.00]# make install  install -m 744 -c r8168.ko /lib/modules/2.6.30.3vbird/kernel/drivers/net/  # 重點在上面這行！會發現模組已經被移動到核心模組目錄！  4. 更新模組相依屬性！  [root@www r8168-8.013.00]# depmod –a |

透過這樣的動作，可以將模組編譯起來，並放置到核心模組目錄中，同時以depmod建立模組相關性，未來就能夠用modprobe直接取用！當自行編譯模組時，若核心有更新，則必須要重新編譯該模組一次才行！因為模組僅針對目前的核心來編譯！