hw0D.make menuconfig

406410114 郭晏誠

1.繳交報告

2.字數：250~300（可以複製，但不可以超過一半，詳列引用來源）

(1)複製的部分用紅色字

(2)自己寫的部分，用黑色字

主題： 介紹 menuconfig 中 I/O Scheduler 中的優化

More accurate cgroup I/O control with blk-iocost

此功能為 Linux 5.4 新增的 blk-iocost I/O 控管功能，引用 Linux 官網原文如下：

One challenge of controlling I/O resources is the lack of reliability of trivial cost metrics. Bandwidth and iops can be off by orders of magnitude depending on the device type and I/O pattern. This is challenging for the I/O cgroup controllers: while io.latency provides the capability to comprehensively prioritize and protect IOs depending on the cgroups, its protection is binary - the lowest latency target cgroup is protected at the cost of all others.

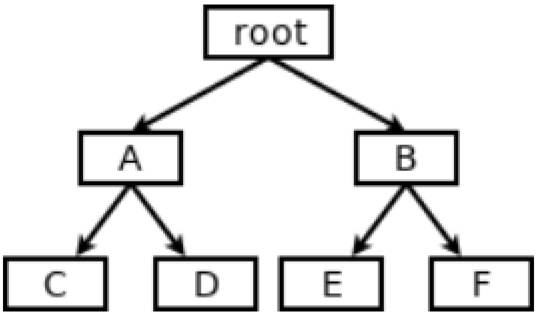
可以發現做 I/O 的挑戰在於要做到許多不同情境下的低延遲，像是 Linux 桌面延遲一直都是 Linux 一個重大的問題，這個版本利用了 blk-iocost，這是基於 I/O 成本節省工作模型的比例控制器，其中每個 I/O 分為順序或隨機並且相對應分配了基本的成本，再額外付加上比例成本，然後再根據 cgroup 的層次結構分配其 I/O 容量。

先來看一下 Cgroup

(1)分為 the core and controllers，core 負責組織流程，controllers 岩層是結構分配特定類型系統資源

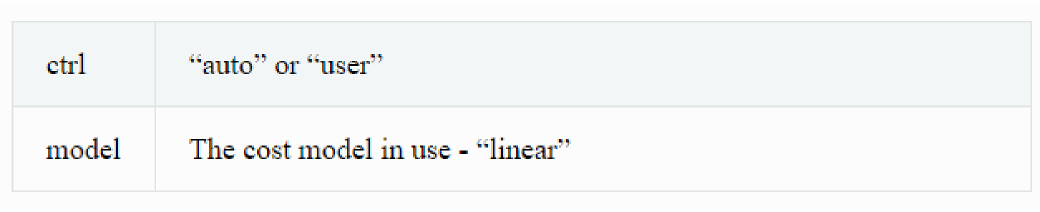
(2)每個 process 都屬於一個 cgroup，每個 process 下的 thread 都屬於同一個 cgroup

EX：若 A 權重為 100，B 為 300，那麼 B 可以拿到 75% I/O 頻寬



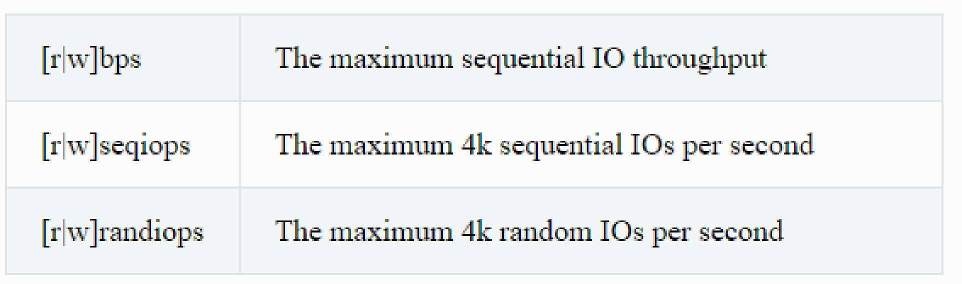
再來進到了 blk-iocost 的 io.cost.model 部分

具有讀寫能力的套件會存在於 root cgroup 部分，利用了 CONFIG_BLK_CGROUP_IOCOST (I/O 成本模型)，以下為定義一開始進入 cgroup 的參數



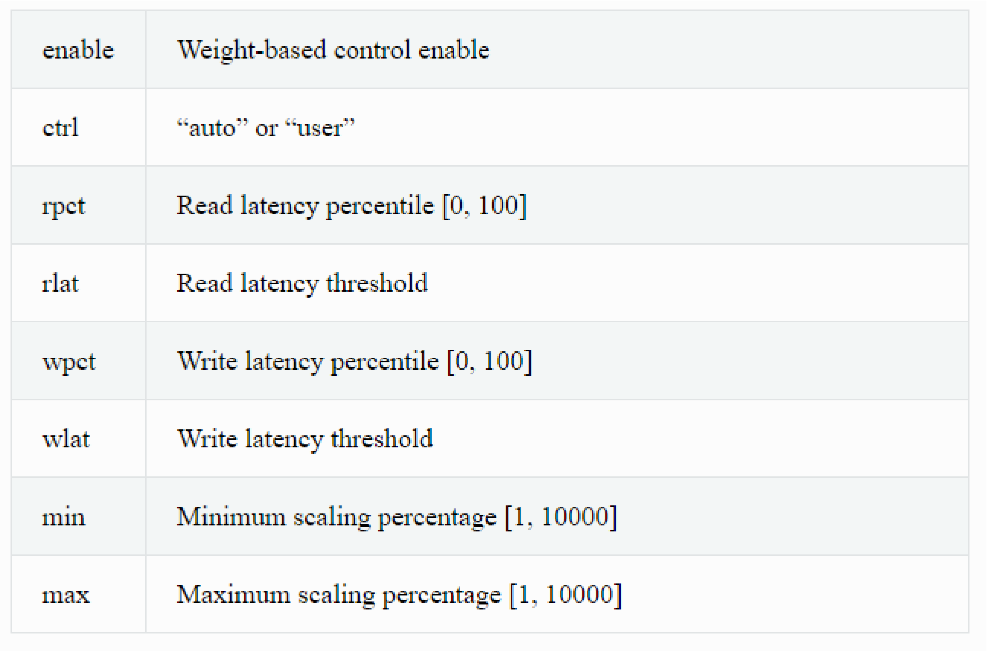
ctrl	“auto” or “user”
model	The cost model in use - “linear”

當 ctrl 為自動的時候，Kernel 可以動態修改任何參數，當為 user 時不能更改，參數定義如下：



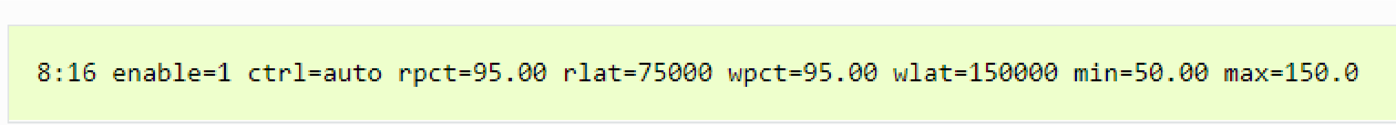
[r|w]bps	The maximum sequential IO throughput
[r|w]seqiops	The maximum 4k sequential IOs per second
[r|w]randiops	The maximum 4k random IOs per second

最後介紹 blk-iocost 的 io.cost.qos 部分



enable	Weight-based control enable
ctrl	“auto” or “user”
rpct	Read latency percentile [0, 100]
rlat	Read latency threshold
wpct	Write latency percentile [0, 100]
wlat	Write latency threshold
min	Minimum scaling percentage [1, 10000]
max	Maximum scaling percentage [1, 10000]

default 情況下 enable 為 1，wpct 為 zero，控制器利用內部飽和狀態去調整 min 和 max，當需要更好的控制的時候可以配置 Qos 參數，例如：



```
8:16 enable=1 ctrl=auto rpct=95.00 rlat=75000 wpct=95.00 wlat=150000 min=50.00 max=150.0
```

當 95%讀取完成時延遲超過 75ms 或 150ms 那麼就會把總體獲得資源量從 50%提升到 150%來進能效能的調整。

參考網址：

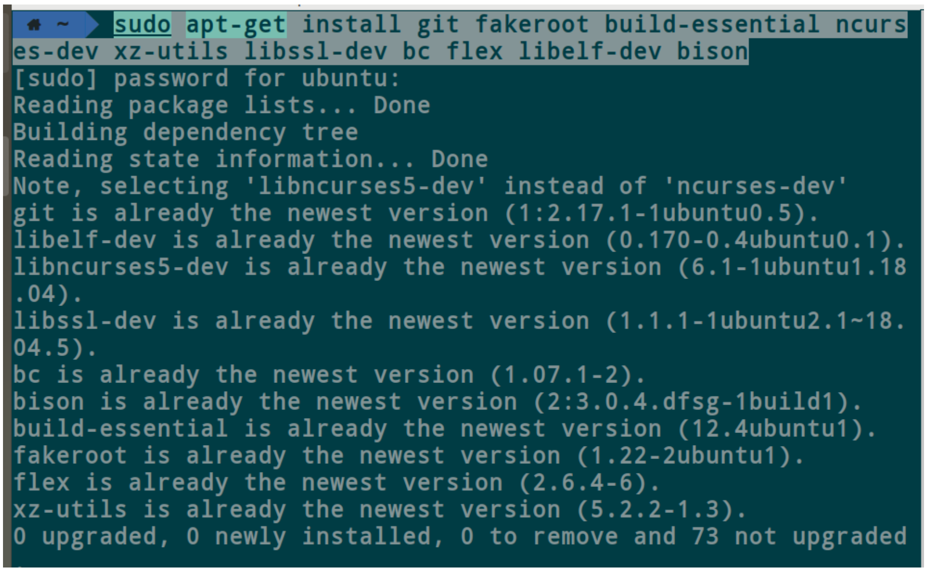
https://kernelnewbies.org/Linux_5.4#More_accurate_cgroup_I.2FO_control_with_blk-iocost
https://www.kernel.org/doc/html/latest/admin-guide/cgroup-v2.html#io
https://lwn.net/Articles/792256/

//以下為 youtube 老師大略介紹的筆記

3.安裝相關套件

sudo apt-get install git fakeroot build-essential ncurses-dev xz-utils libssl-dev bc flex libelf-dev bison

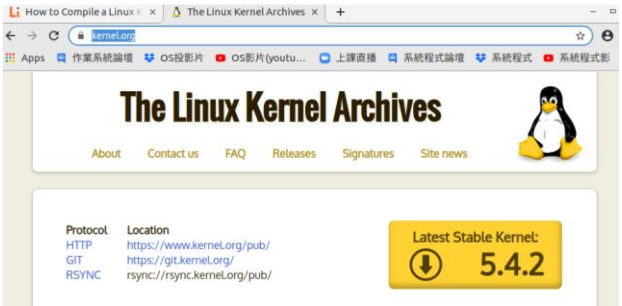
PS 如有 BUG 請見 https://www.linuxuprising.com/2018/07/how-to-fix-could-not-get-lock.html



```
 ~ sudo apt-get install git fakeroot build-essential ncurs
es-dev xz-utils libssl-dev bc flex libelf-dev bison
[sudo] password for ubuntu:
Reading package lists... Done
Building dependency tree
Reading state information... Done
Note, selecting 'libncurses5-dev' instead of 'ncurses-dev'
git is already the newest version (1:2.17.1-1ubuntu0.5).
libelf-dev is already the newest version (0.170-0.4ubuntu0.1).
libncurses5-dev is already the newest version (6.1-1ubuntu1.18
.04).
libssl-dev is already the newest version (1.1.1-1ubuntu2.1~18.
04.5).
bc is already the newest version (1.07.1-2).
bison is already the newest version (2:3.0.4.dfsg-1build1).
build-essential is already the newest version (12.4ubuntu1).
fakeroot is already the newest version (1.22-2ubuntu1).
flex is already the newest version (2.6.4-6).
xz-utils is already the newest version (5.2.2-1.3).
0 upgraded, 0 newly installed, 0 to remove and 73 not upgraded
```

4.下載 5.4.2 kernel

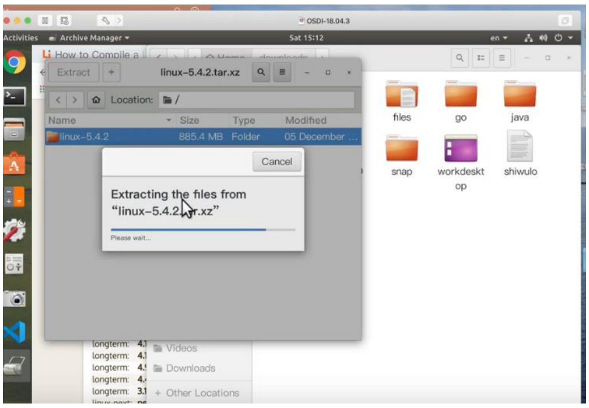
(1)https://www.kernel.org/ 選 5.4.2



The Linux Kernel Archives

Protocol	Location
HTTP	https://www.kernel.org/pub/
GIT	https://git.kernel.org/
RSYNC	rsync://rsync.kernel.org/pub/

(2)解壓縮到 home



Name	Size	Type	Modified
linux-5.4.2	885.4 MB	Folder	05 December ...

Extracting the files from "linux-5.4.2.tar.xz"

Please wait...

5.製造 config

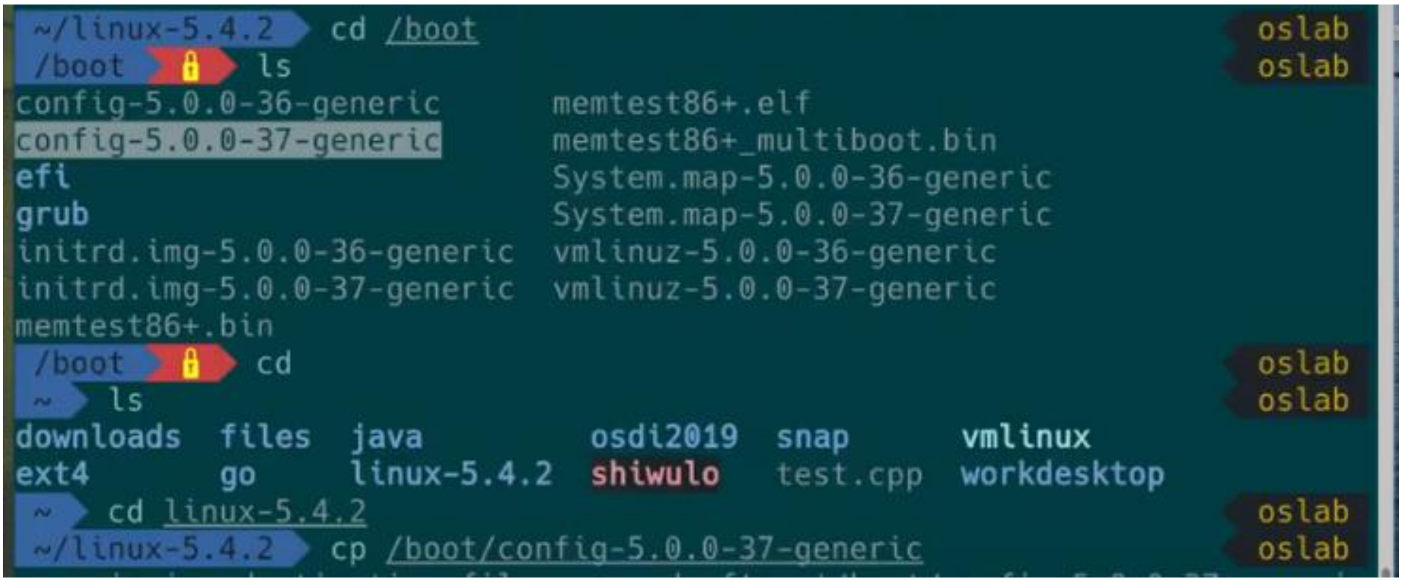
(1)保證視窗 80↑*25↑字元(此為最古老視窗，也是現代正規 20 萬以上伺服器常使用的，極其穩定好 Debug，常使用 D-sub 接頭)

(2)複製 config

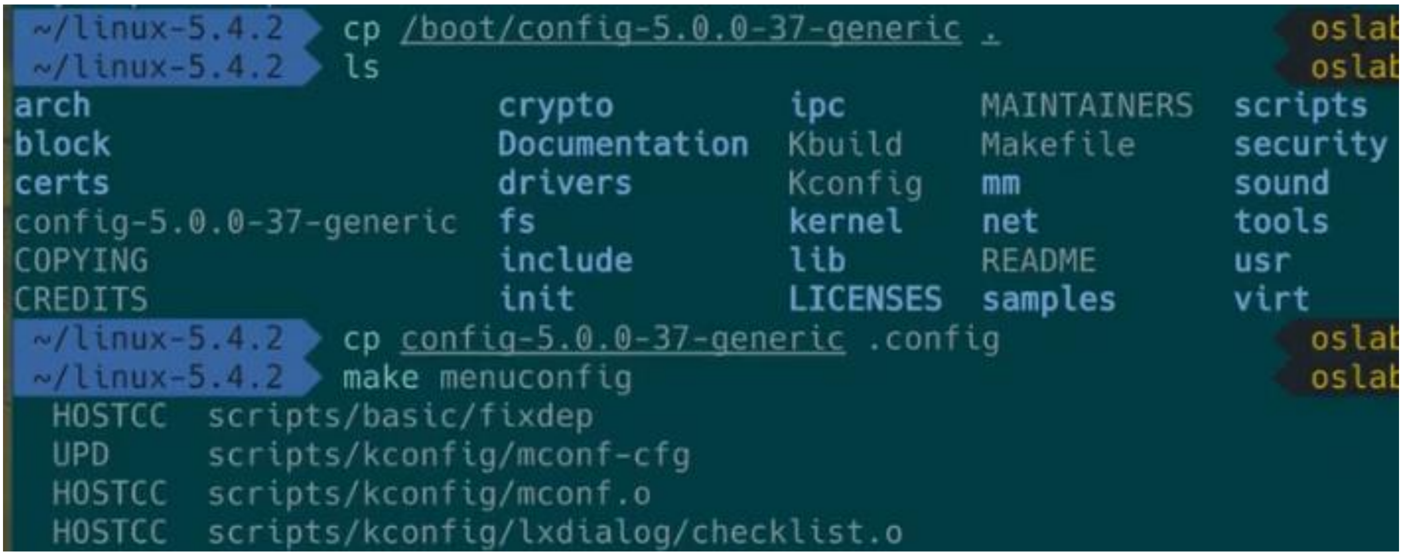
cd boot ##找到 config-5.0.0-36-generic

cp /boot/config-5.0.0-36-generic . #拷貝一份備份

cp /boot/config-5.0.0-36-generic .config #內部 make 使用



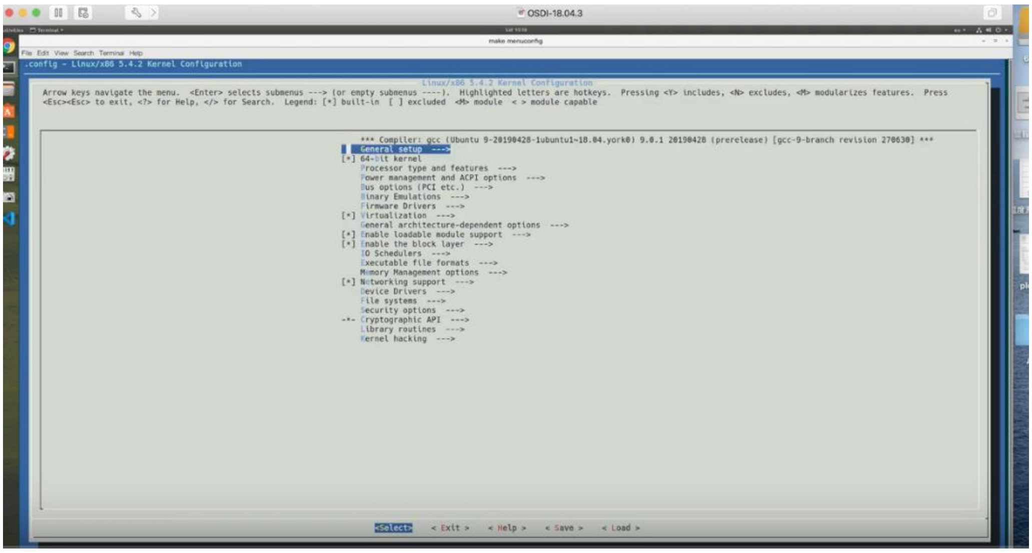
```
~/linux-5.4.2  cd /boot                                                          oslab
/boot 🔒  ls                                                                       oslab
config-5.0.0-36-generic       memtest86+.elf
config-5.0.0-37-generic       memtest86+_multiboot.bin
efi                           System.map-5.0.0-36-generic
grub                          System.map-5.0.0-37-generic
initrd.img-5.0.0-36-generic   vmlinuz-5.0.0-36-generic
initrd.img-5.0.0-37-generic   vmlinuz-5.0.0-37-generic
memtest86+.bin
/boot 🔒  cd                                                                       oslab
~  ls                                                                              oslab
downloads  files  java         osdi2019  snap      vmlinux
ext4       go     linux-5.4.2  shiwulo   test.cpp  workdesktop
~  cd linux-5.4.2                                                                  oslab
~/linux-5.4.2  cp /boot/config-5.0.0-37-generic                                    oslab
```



```
~/linux-5.4.2  cp /boot/config-5.0.0-37-generic .                                    oslab
~/linux-5.4.2  ls                                                                    oslab
arch                      crypto         ipc       MAINTAINERS  scripts
block                     Documentation  Kbuild    Makefile     security
certs                     drivers        Kconfig   mm           sound
config-5.0.0-37-generic   fs             kernel    net          tools
COPYING                   include        lib       README       usr
CREDITS                   init           LICENSES  samples      virt
~/linux-5.4.2  cp config-5.0.0-37-generic .config                                    oslab
~/linux-5.4.2  make menuconfig                                                       oslab
  HOSTCC  scripts/basic/fixdep
  UPD     scripts/kconfig/mconf-cfg
  HOSTCC  scripts/kconfig/mconf.o
  HOSTCC  scripts/kconfig/lxdialog/checklist.o
```

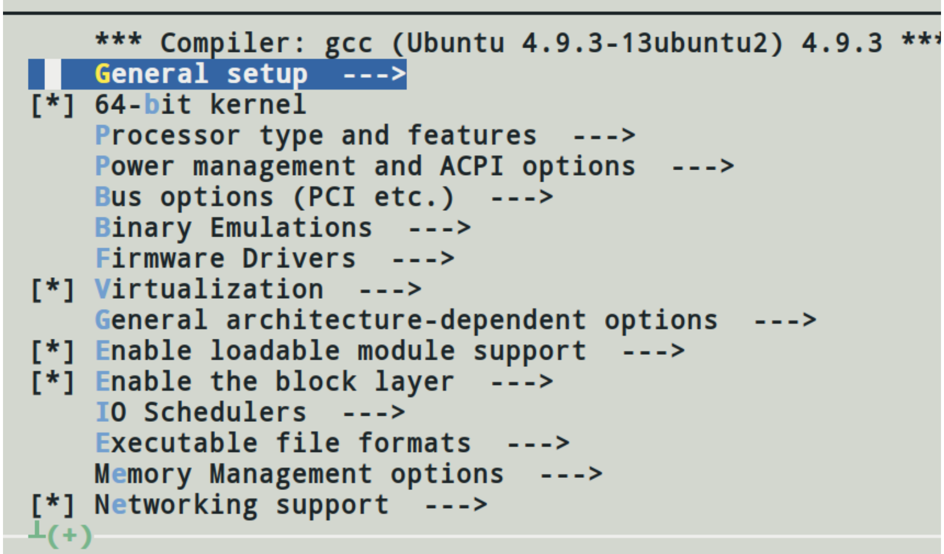
6. make menuconfig

（1）make menuconfig



```
.config - Linux/x86 5.4.2 Kernel Configuration
                         Linux/x86 5.4.2 Kernel Configuration
 Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus ----).  Highlighted letters are hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes features.  Press
 <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*] built-in  [ ] excluded  <M> module  < > module capable

        *** Compiler: gcc (Ubuntu 9-20190428-1ubuntu1~18.04.york0) 9.0.1 20190428 (prerelease) [gcc-9-branch revision 270630] ***
        General setup  --->
    [*] 64-bit kernel
        Processor type and features  --->
        Power management and ACPI options  --->
        Bus options (PCI etc.)  --->
        Binary Emulations  --->
        Firmware Drivers  --->
    [*] Virtualization  --->
        General architecture-dependent options  --->
    [*] Enable loadable module support  --->
    [*] Enable the block layer  --->
        IO Schedulers  --->
        Executable file formats  --->
        Memory Management options  --->
    [*] Networking support  --->
        Device Drivers  --->
        File systems  --->
        Security options  --->
    -*- Cryptographic API  --->
        Library routines  --->
        Kernel hacking  --->

            <Select>    < Exit >    < Help >    < Save >    < Load >
```

(2)General setup



```
    *** Compiler: gcc (Ubuntu 4.9.3-13ubuntu2) 4.9.3 ***
    General setup  --->
[*] 64-bit kernel
    Processor type and features  --->
    Power management and ACPI options  --->
    Bus options (PCI etc.)  --->
    Binary Emulations  --->
    Firmware Drivers  --->
[*] Virtualization  --->
    General architecture-dependent options  --->
[*] Enable loadable module support  --->
[*] Enable the block layer  --->
    IO Schedulers  --->
    Executable file formats  --->
    Memory Management options  --->
[*] Networking support  --->
┴(+)
```

(3)設定

(i)Support for paging of anonymous memory (swap) #swap 用

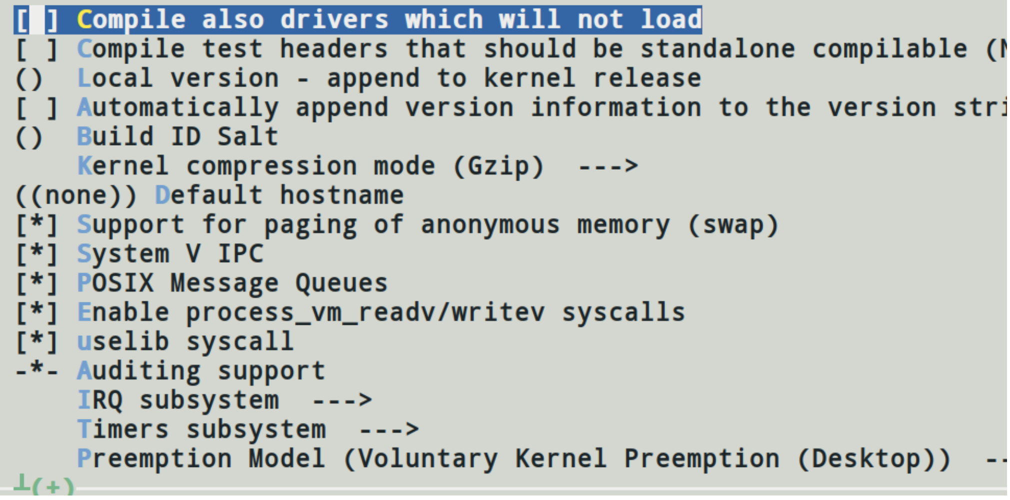
(ii)System V IPC # five IPC

(iii)Enable process_vm_readv/writev syscalls #對另一個 process 讀寫

(vi)uselib syscall #支援舊的部分

(v)Auditing support #系統做統計用

(vi)IRQ subsystem ---> #系統除錯的時候可以進去做一些選擇



```
[ ] Compile also drivers which will not load
[ ] Compile test headers that should be standalone compilable (N
()  Local version - append to kernel release
[ ] Automatically append version information to the version stri
()  Build ID Salt
    Kernel compression mode (Gzip)  --->
((none)) Default hostname
[*] Support for paging of anonymous memory (swap)
[*] System V IPC
[*] POSIX Message Queues
[*] Enable process_vm_readv/writev syscalls
[*] uselib syscall
-*- Auditing support
    IRQ subsystem  --->
    Timers subsystem  --->
    Preemption Model (Voluntary Kernel Preemption (Desktop))  --
⊥(+)
```

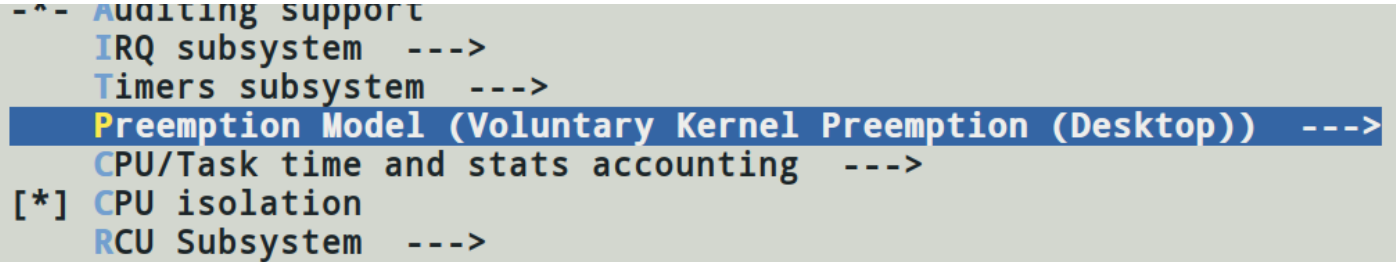
(vii)Preemption Model (Voluntary Kernel Preemption (Desktop)) --->

(a)No Forced Preemption (Server) #在 kernel space 遇到中斷部會馬上執行，先

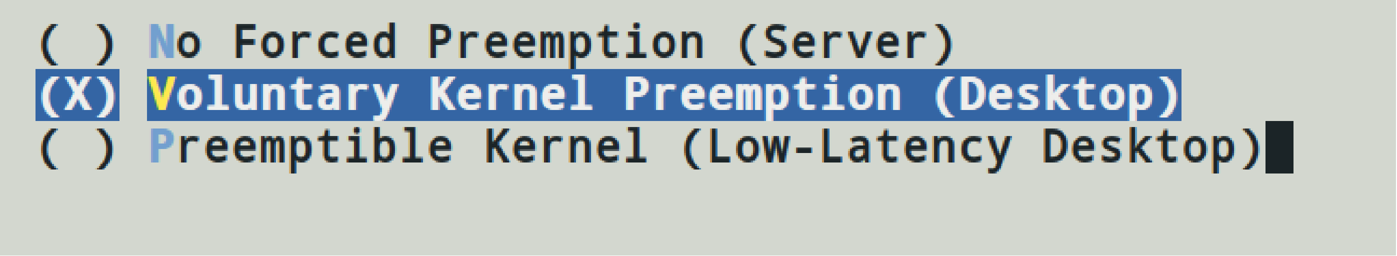
做 LOG 紀錄

(b)Voluntary Kernel Preemption (Desktop)

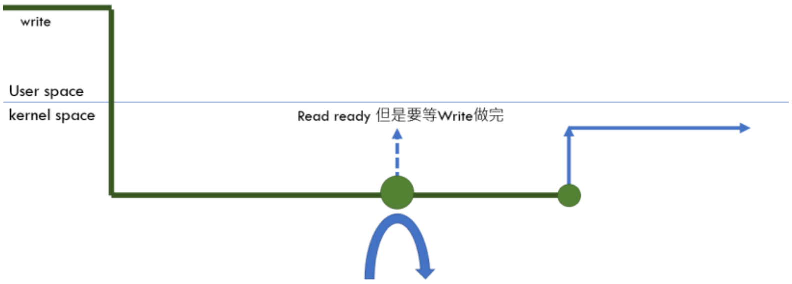
(c)Preemptible Kernel (Low-Latency Desktop)#只要能 Preempt 就 Preempt



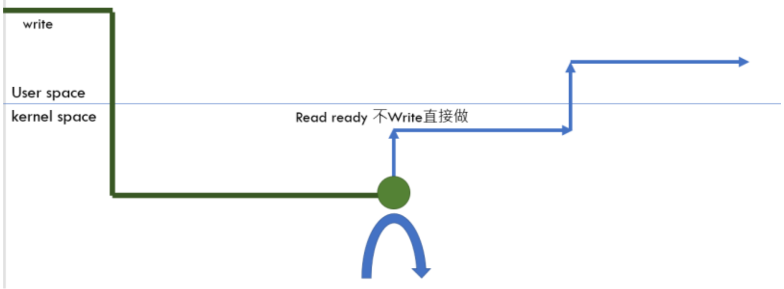
```
-*- Auditing support
    IRQ subsystem  --->
    Timers subsystem  --->
    Preemption Model (Voluntary Kernel Preemption (Desktop))  --->
    CPU/Task time and stats accounting  --->
[*] CPU isolation
    RCU Subsystem  --->
```



```
( ) No Forced Preemption (Server)
(X) Voluntary Kernel Preemption (Desktop)
( ) Preemptible Kernel (Low-Latency Desktop)
```

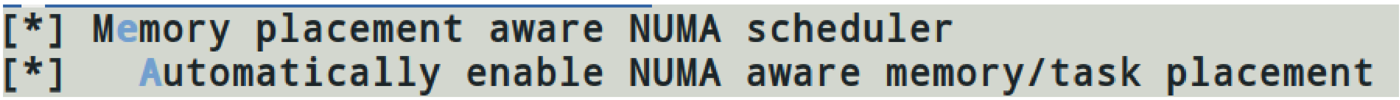


No Forced Preemption（會記錄不做中斷）



Preemptible Kernel (Low-Latency Desktop)

(viii) NUMA 支援

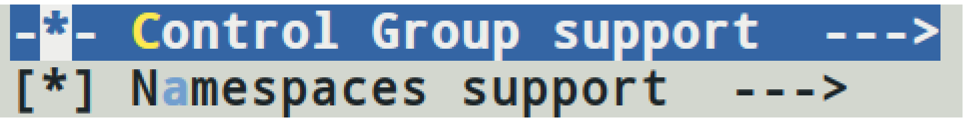


```
[*] Memory placement aware NUMA scheduler
[*]   Automatically enable NUMA aware memory/task placement
```

(ix)用來做容器 container

Control Group support#限定一群 process 只能用固定資源

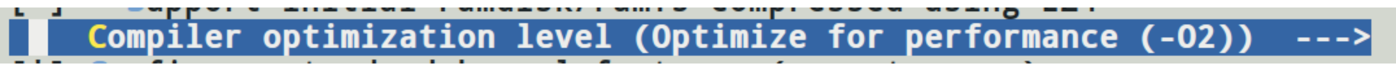
Namespaces support #在 container 中第一個 process id 是 1 非真的 1，只是用來在 container 中作為 priority 用



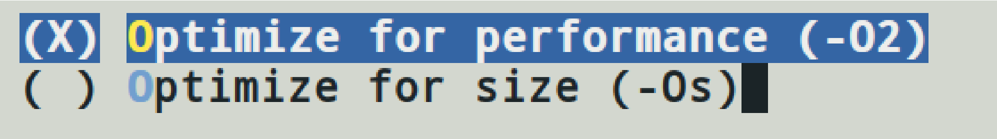
```
-*- Control Group support  --->
[*] Namespaces support  --->
```

(x) Compiler optimization level (Optimize for performance (-O2))

要選-02　因為到-03 會有向量指令集，在 kernel 中用不到

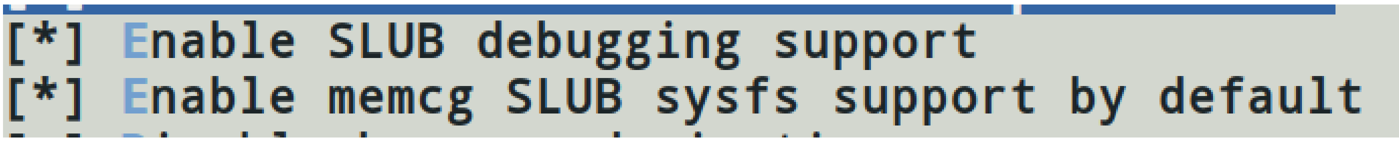


```
 Compiler optimization level (Optimize for performance (-O2))  --->
```

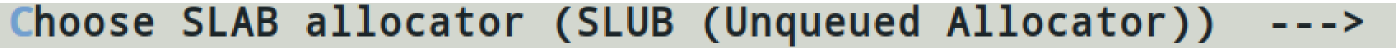


```
(X) Optimize for performance (-O2)
( ) Optimize for size (-Os)
```

(xi)SLUB　#在 kernel space 實現 malloc 類似於 SLAB　也用在 ASLR(地址布局隨機化)

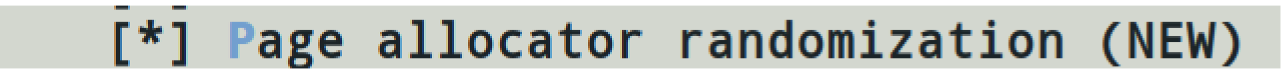


```
[*] Enable SLUB debugging support
[*] Enable memcg SLUB sysfs support by default
```



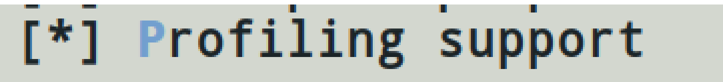
```
Choose SLAB allocator (SLUB (Unqueued Allocator))  --->
```

(xii)只要有 randomization 跟安全性有關



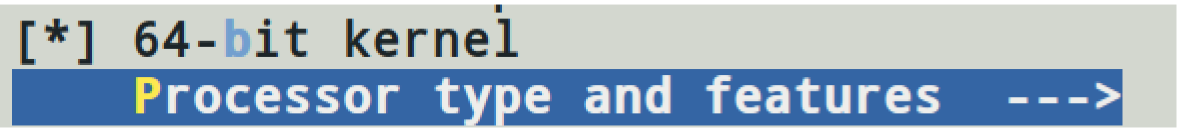
```
    [*] Page allocator randomization (NEW)
```

(xiii)Profiling support　#支援 OProfile 跟 herf 有點像



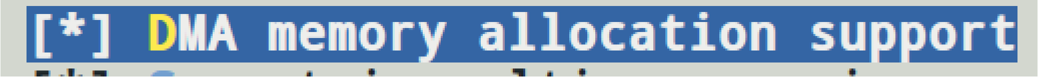
```
[*] Profiling support
```

(4)Processor type and features



```
[*] 64-bit kernel
    Processor type and features  --->
```

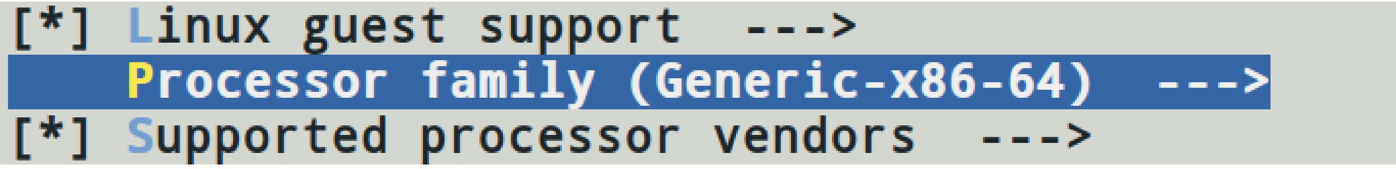
(i)DMA



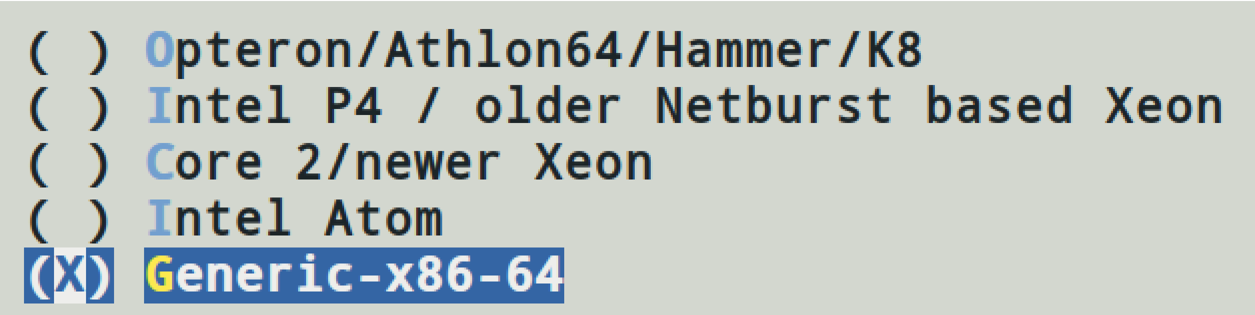
```
[*] DMA memory allocation support
```

(ii)Processor family (Generic-x86-64)

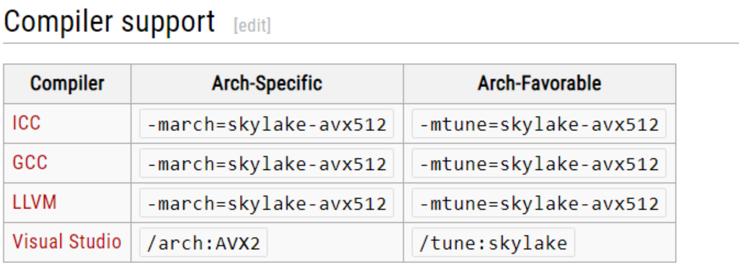
#可以選擇處理器 可參考 https://en.wikichip.org/wiki/WikiChip



```
[*] Linux guest support  --->
    Processor family (Generic-x86-64)  --->
[*] Supported processor vendors  --->
```



```
( ) Opteron/Athlon64/Hammer/K8
( ) Intel P4 / older Netburst based Xeon
( ) Core 2/newer Xeon
( ) Intel Atom
(X) Generic-x86-64
```



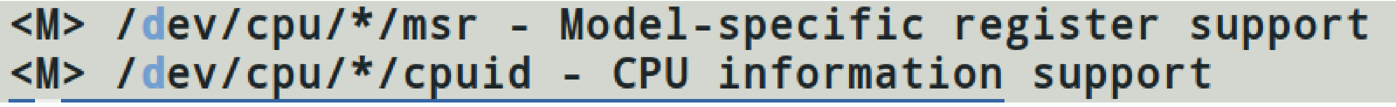
Compiler support [edit]

Compiler	Arch-Specific	Arch-Favorable
ICC	-march=skylake-avx512	-mtune=skylake-avx512
GCC	-march=skylake-avx512	-mtune=skylake-avx512
LLVM	-march=skylake-avx512	-mtune=skylake-avx512
Visual Studio	/arch:AVX2	/tune:skylake

Skylake (server) - Microarchitectures – Intel

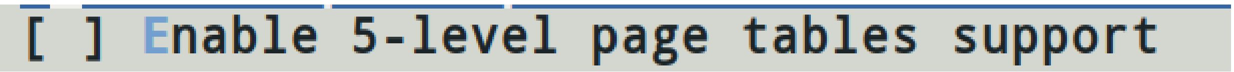
左邊為止能在此 CPU 運行的指令集，右邊為通用但是在此 CPU 特別快

(ii)msr 和 cpuid #可打開 msr 是跟型號有關的暫存器 可透過 cpuid 查詢



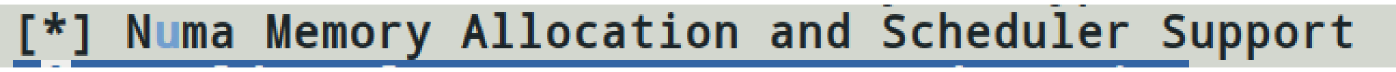
```
<M> /dev/cpu/*/msr - Model-specific register support
<M> /dev/cpu/*/cpuid - CPU information support
```

(iii)5-level page table # data center 要打開



```
[ ] Enable 5-level page tables support
```

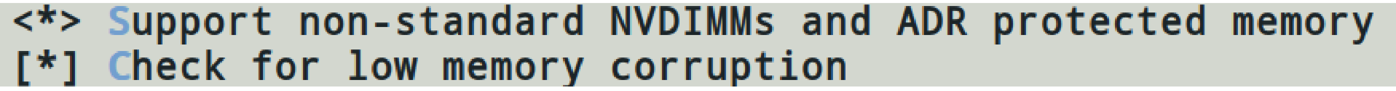
(iv) Numa Memory Allocation and Scheduler Support #不同核心會有不同的速度 Linux 針對此做的優化



```
[*] Numa Memory Allocation and Scheduler Support
```

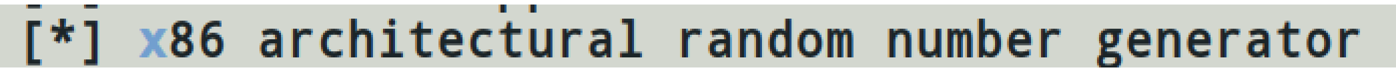
https://www.youtube.com/watch?v=ZBDuvrVckik

(v)NVIMMS #用在伺服器上 有多種形式 其中一種為停電以後 上面有一顆電池會把 DRAM 上資料寫到 DRAM 的 FALSH 中，下次正常開機後，主機板會發號司令叫 DRAM 把 FALSH 中資料還原到 DRAM



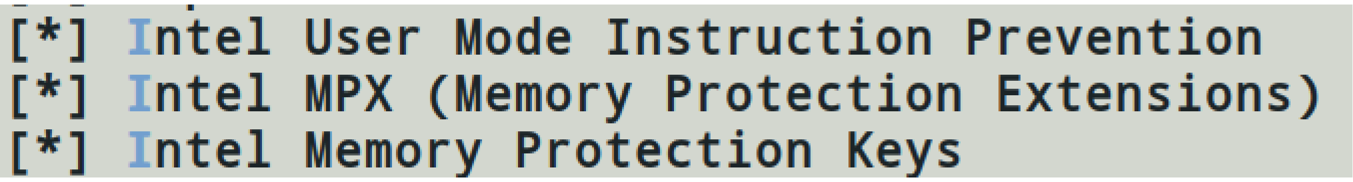
```
<*> Support non-standard NVDIMMs and ADR protected memory
[*] Check for low memory corruption
```

(vi) x86 architectural random number generator #用於加密



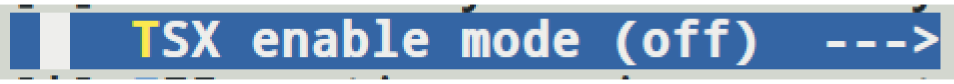
```
[*] x86 architectural random number generator
```

(vii)Intel 安全相關模組



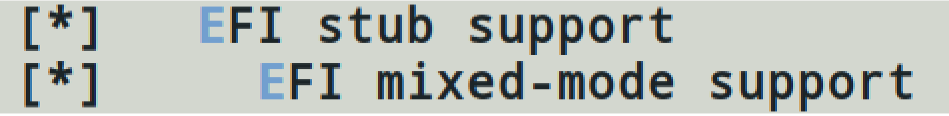
```
[*] Intel User Mode Instruction Prevention
[*] Intel MPX (Memory Protection Extensions)
[*] Intel Memory Protection Keys
```

(vii) TSX enable mode #為 Transaction no memory 可打開



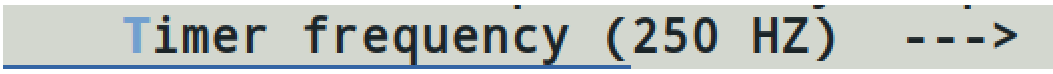
```
TSX enable mode (off)  --->
```

(viii)UEFI #

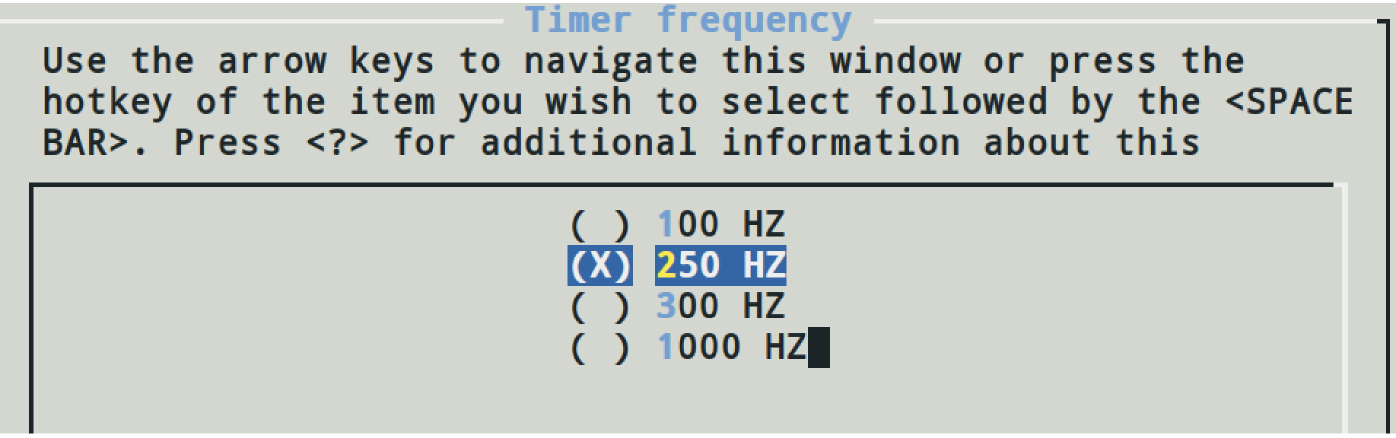


```
[*]   EFI stub support
[*]     EFI mixed-mode support
```

(ix) Timer frequency #100 or 250 or1000 較常用 傳統上每秒鐘發出 1000 次中斷 做出 Round R 但缺點是 CPU 不能跑去睡，但是現在沒有絕對要用 1000

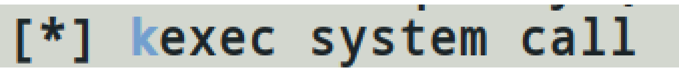


```
   Timer frequency (250 HZ)  --->
```



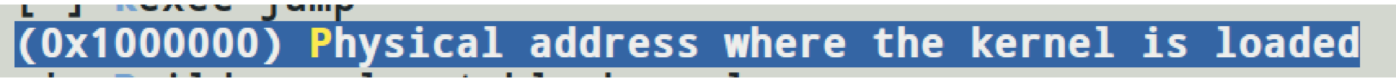
```
                      Timer frequency
 Use the arrow keys to navigate this window or press the
 hotkey of the item you wish to select followed by the <SPACE
 BAR>. Press <?> for additional information about this

                      ( ) 100 HZ
                      (X) 250 HZ
                      ( ) 300 HZ
                      ( ) 1000 HZ
```

(x) kexec system call #原本設計電腦伺服器不用停機 可以換掉 kernel 也支援 kernel dump(kernel 死掉 把資料透過 kexec 寫到 Disk，其實開機有 2 個 kernel，正的 kernel 掛掉，kexec 是副的 kernel 把資料倒到暫存器)



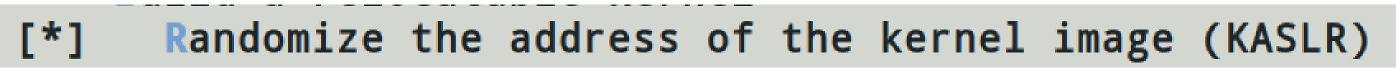
```
[*] kexec system call
```

(xi) Physical address where the kernel is loaded #kernel 載入 不要隨便改會無法開機



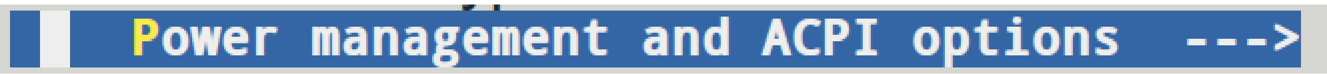
```
(0x1000000) Physical address where the kernel is loaded
```

(xii)KXSLR #保護 Kernel



```
[*]   Randomize the address of the kernel image (KASLR)
```

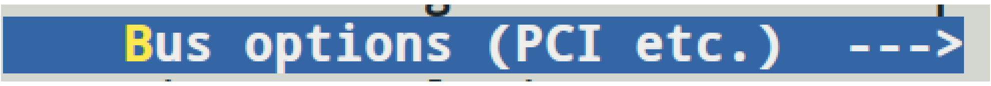
(5) Power management and ACPI options #跟省電和高效能有關再進去調整



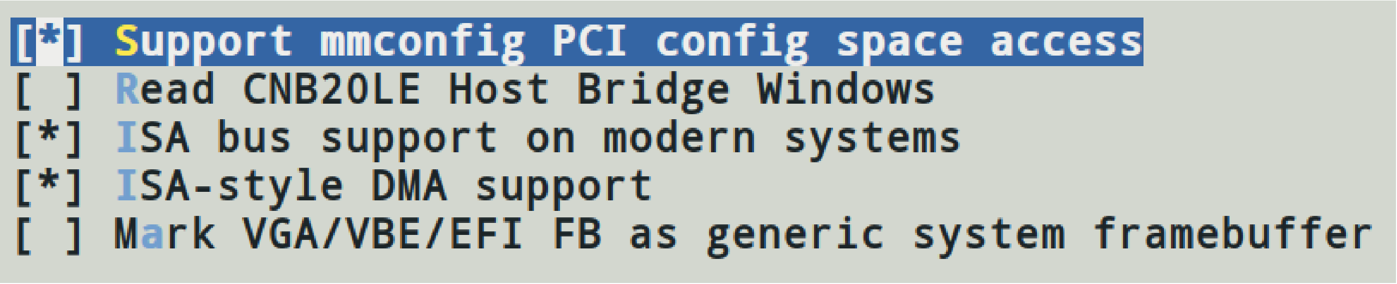
```
Power management and ACPI options  --->
```

(6) Bus options (PCI etc.) #不要動

#若會用到 framebuffer 可打開 它為舊型圖形介面驅動程式(伺服器 因每個 pixel 去化效能低)

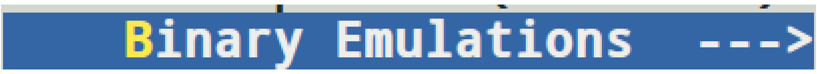


```
Bus options (PCI etc.)  --->
```

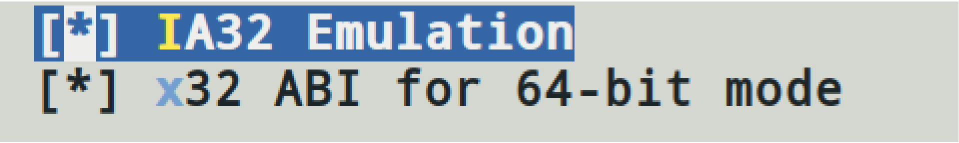


```
[*] Support mmconfig PCI config space access
[ ] Read CNB20LE Host Bridge Windows
[*] ISA bus support on modern systems
[*] ISA-style DMA support
[ ] Mark VGA/VBE/EFI FB as generic system framebuffer
```

(7) Binary Emulations #支援 32 位元

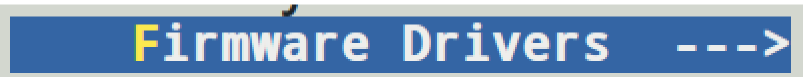


```
Binary Emulations  --->
```

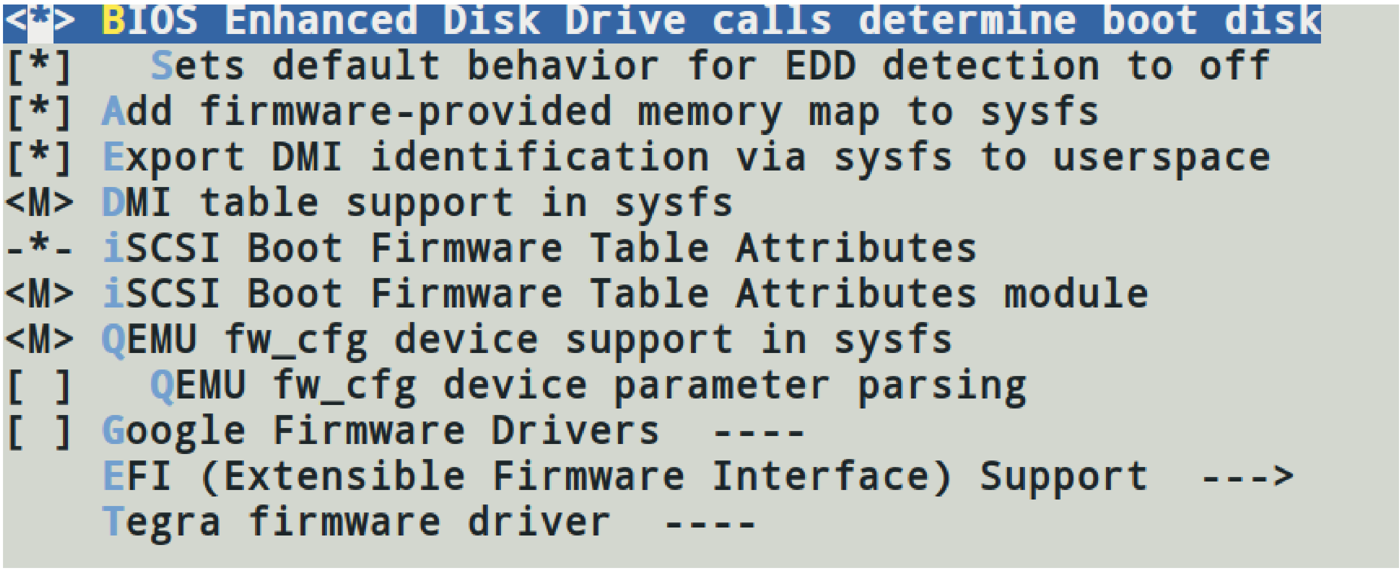


```
[*] IA32 Emulation
[*] x32 ABI for 64-bit mode
```

(8) Firmware Drivers #韌體 driver 盡量不要動

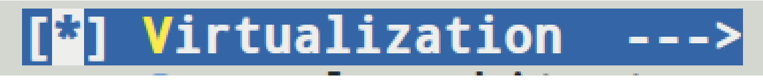


```
Firmware Drivers  --->
```

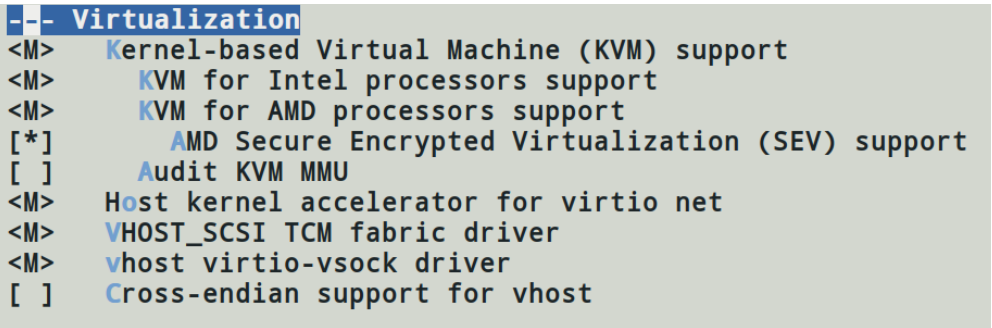


```
<*> BIOS Enhanced Disk Drive calls determine boot disk
[*]   Sets default behavior for EDD detection to off
[*] Add firmware-provided memory map to sysfs
[*] Export DMI identification via sysfs to userspace
<M> DMI table support in sysfs
-*- iSCSI Boot Firmware Table Attributes
<M> iSCSI Boot Firmware Table Attributes module
<M> QEMU fw_cfg device support in sysfs
[ ]   QEMU fw_cfg device parameter parsing
[ ] Google Firmware Drivers  ----
    EFI (Extensible Firmware Interface) Support  --->
    Tegra firmware driver  ----
```

(9) Virtualization #核心內建支援虛擬化技術 Linux 可當 Host 或 Guest



```
[*] Virtualization  --->
```



```
--- Virtualization
<M>   Kernel-based Virtual Machine (KVM) support
<M>     KVM for Intel processors support
<M>     KVM for AMD processors support
[*]       AMD Secure Encrypted Virtualization (SEV) support
[ ]     Audit KVM MMU
<M>   Host kernel accelerator for virtio net
<M>   VHOST_SCSI TCM fabric driver
<M>   vhost virtio-vsock driver
[ ]   Cross-endian support for vhost
```

Guest
(Linux)
Host
(Mac OS)

(10) General architecture-dependent options

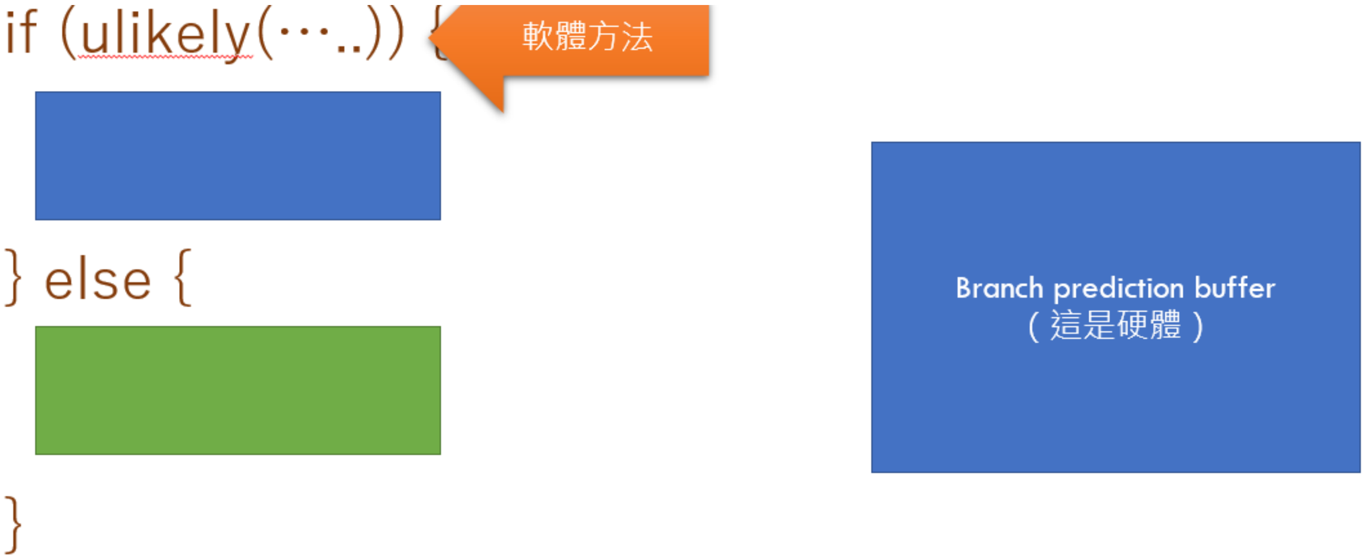
```
    General architecture-dependent options  --->
```

(i) OProfile system profiling #選 M 是編譯成 Module 會動態載入到 Kernel 類似瀏覽器插件，編成* 一開始就進入 kernel，如果是檔案系統盡量要編譯成*，怕一開始找不到

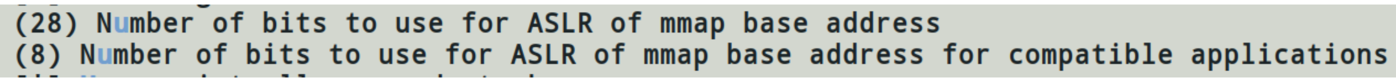
```
<M> OProfile system profiling
```

(ii)對 likely 和 unlikely 做優化 #likely 代表這個 case 執行機會高 unlikely 相反
#CPU 有 branch prediction buffer 通常程式行為會跟上次一樣例如 while 就是種 likely

```
[*] Optimize very unlikely/likely branches
```



(iii) Number of bits to use for ASLR of mmap base address #用多少為原來做隨機



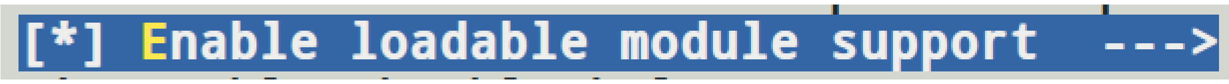
```
(28) Number of bits to use for ASLR of mmap base address
(8) Number of bits to use for ASLR of mmap base address for compatible applications
```

(iv) Locking event counts collection #有多隨機

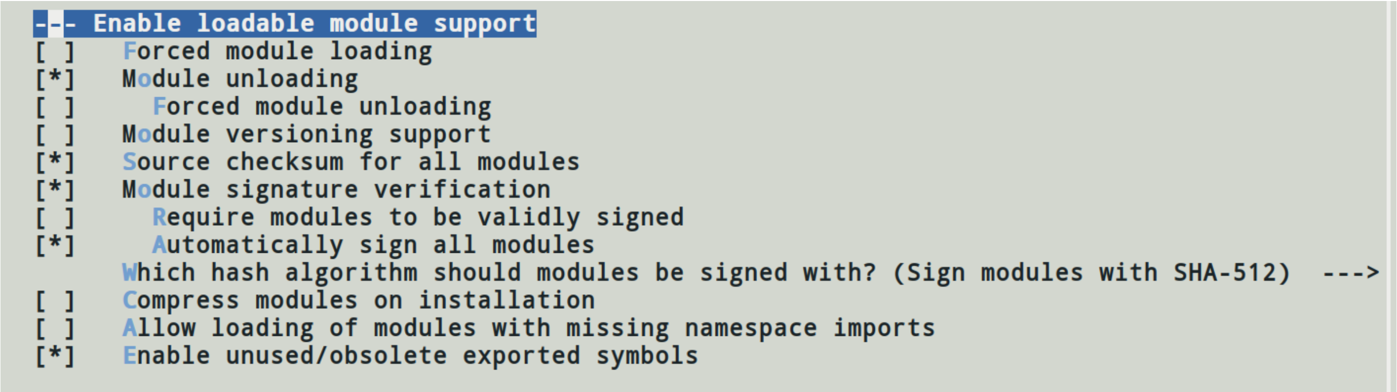
```
[ ] Locking event counts collection
```

(11) Enable loadable module support #一定要選 動態載入模組 USB 網卡等等 嵌入式系統

不用選

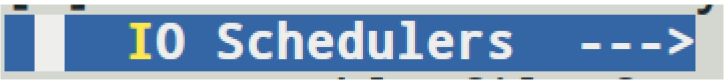


```
[*] Enable loadable module support  --->
```

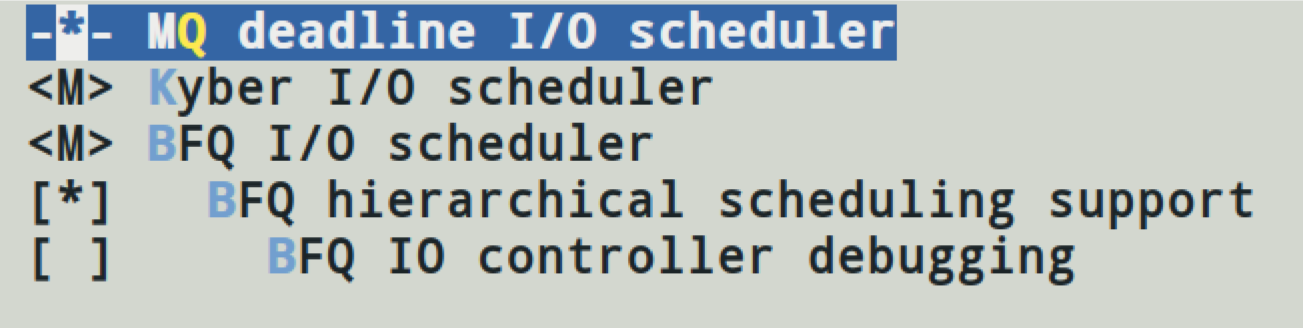


```
--- Enable loadable module support
[ ]   Forced module loading
[*]   Module unloading
[ ]     Forced module unloading
[ ]   Module versioning support
[*]   Source checksum for all modules
[*]   Module signature verification
[ ]     Require modules to be validly signed
[*]     Automatically sign all modules
        Which hash algorithm should modules be signed with? (Sign modules with SHA-512)  --->
[ ]   Compress modules on installation
[ ]   Allow loading of modules with missing namespace imports
[*]   Enable unused/obsolete exported symbols
```

(12) IO Schedulers # 比較不重要 ，因為像 NVME Queue 有 2048 個 排程完全交給控制器排程

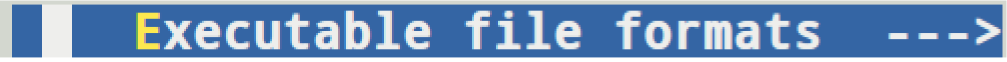


```
    IO Schedulers  --->
```

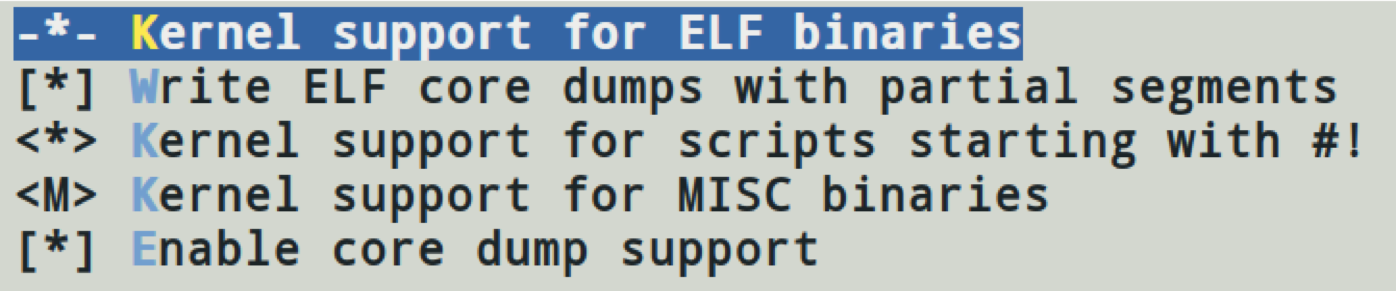


```
-*- MQ deadline I/O scheduler
<M> Kyber I/O scheduler
<M> BFQ I/O scheduler
[*]   BFQ hierarchical scheduling support
[ ]     BFQ IO controller debugging
```

(13) Executable file formats #支援執行檔



```
    Executable file formats  --->
```



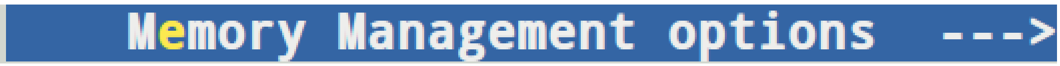
```
-*- Kernel support for ELF binaries
[*] Write ELF core dumps with partial segments
<*> Kernel support for scripts starting with #!
<M> Kernel support for MISC binaries
[*] Enable core dump support
```

(i) Write ELF core dumps with partial segments #支援執行檔

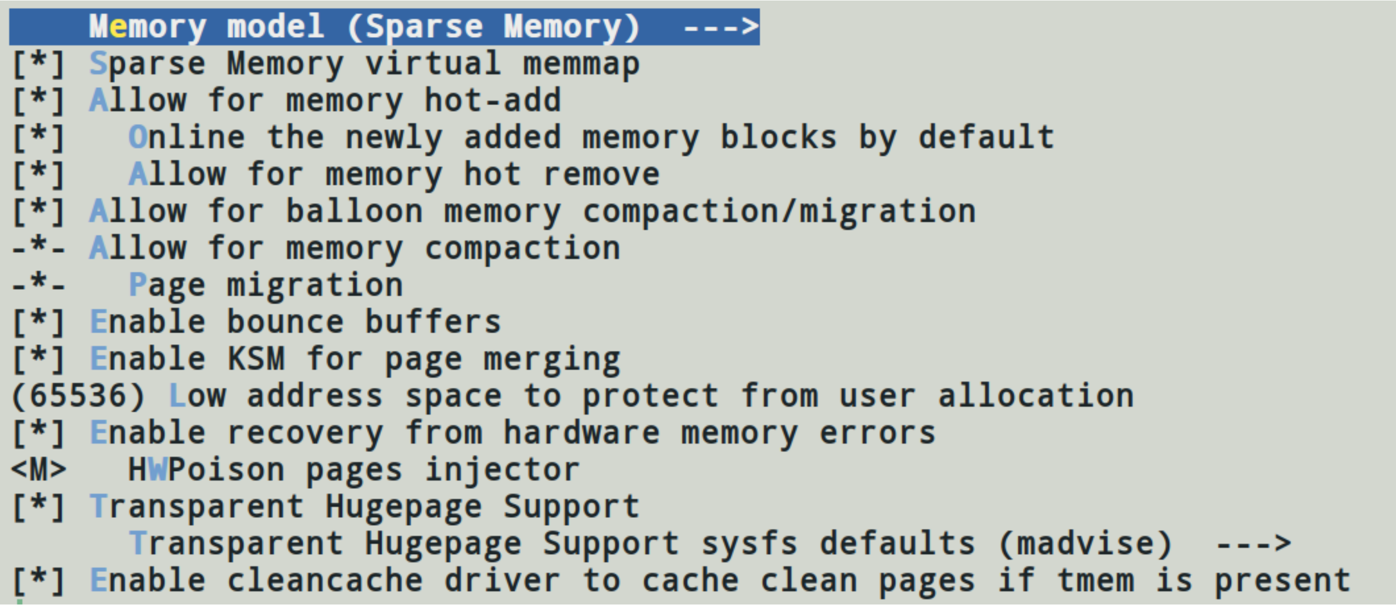
(ii) Kernel support for scripts starting with #! #支援 shell script

(iii) Enable core dump support #支援 core dump kernel 掛掉可以寫東西出去

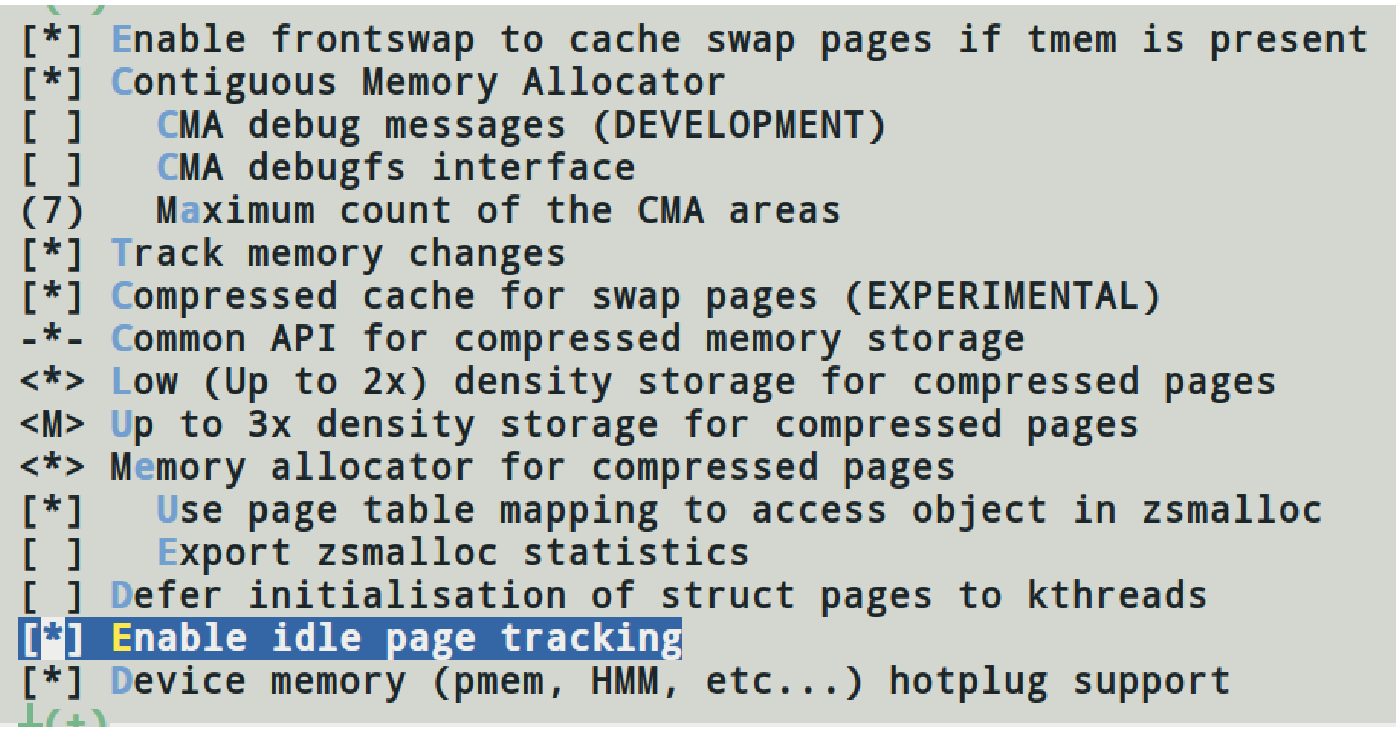
(14) Memory Management options #記憶體管理



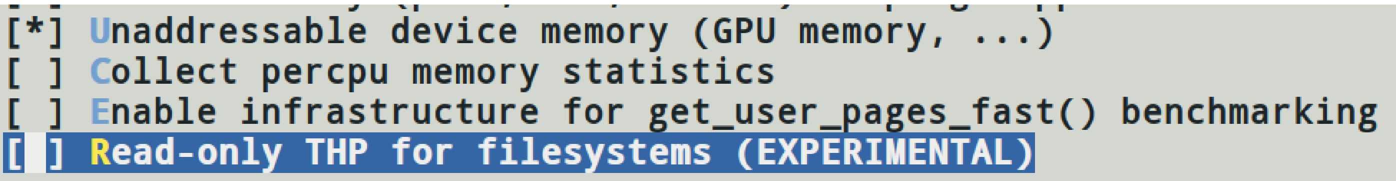
```
    Memory Management options  --->
```



```
    Memory model (Sparse Memory)  --->
[*] Sparse Memory virtual memmap
[*] Allow for memory hot-add
[*]   Online the newly added memory blocks by default
[*]   Allow for memory hot remove
[*] Allow for balloon memory compaction/migration
-*- Allow for memory compaction
-*-   Page migration
[*] Enable bounce buffers
[*] Enable KSM for page merging
(65536) Low address space to protect from user allocation
[*] Enable recovery from hardware memory errors
<M>   HWPoison pages injector
[*] Transparent Hugepage Support
      Transparent Hugepage Support sysfs defaults (madvise)  --->
[*] Enable cleancache driver to cache clean pages if tmem is present
```



```
[*] Enable frontswap to cache swap pages if tmem is present
[*] Contiguous Memory Allocator
[ ]   CMA debug messages (DEVELOPMENT)
[ ]   CMA debugfs interface
(7)   Maximum count of the CMA areas
[*] Track memory changes
[*] Compressed cache for swap pages (EXPERIMENTAL)
-*- Common API for compressed memory storage
<*> Low (Up to 2x) density storage for compressed pages
<M> Up to 3x density storage for compressed pages
<*> Memory allocator for compressed pages
[*]   Use page table mapping to access object in zsmalloc
[ ]   Export zsmalloc statistics
[ ] Defer initialisation of struct pages to kthreads
[*] Enable idle page tracking
[*] Device memory (pmem, HMM, etc...) hotplug support
```



```
[*] Unaddressable device memory (GPU memory, ...)
[ ] Collect percpu memory statistics
[ ] Enable infrastructure for get_user_pages_fast() benchmarking
[ ] Read-only THP for filesystems (EXPERIMENTAL)
```

(i) Allow for balloon memory compaction/migration #記憶體 migration 用氣球壓縮

(ii) Enable KSM for page merging #kernel samepage merging 會把一樣的記憶體內容合併 例如跑兩個 VM 會共用記憶體內容

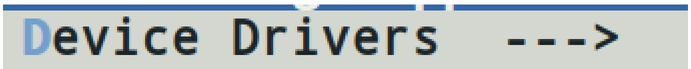
(iii) Enable cleancache driver to cache clean pages if tmem is present #cleancache driver

(iv) Contiguous Memory Allocator #連續記憶體分配

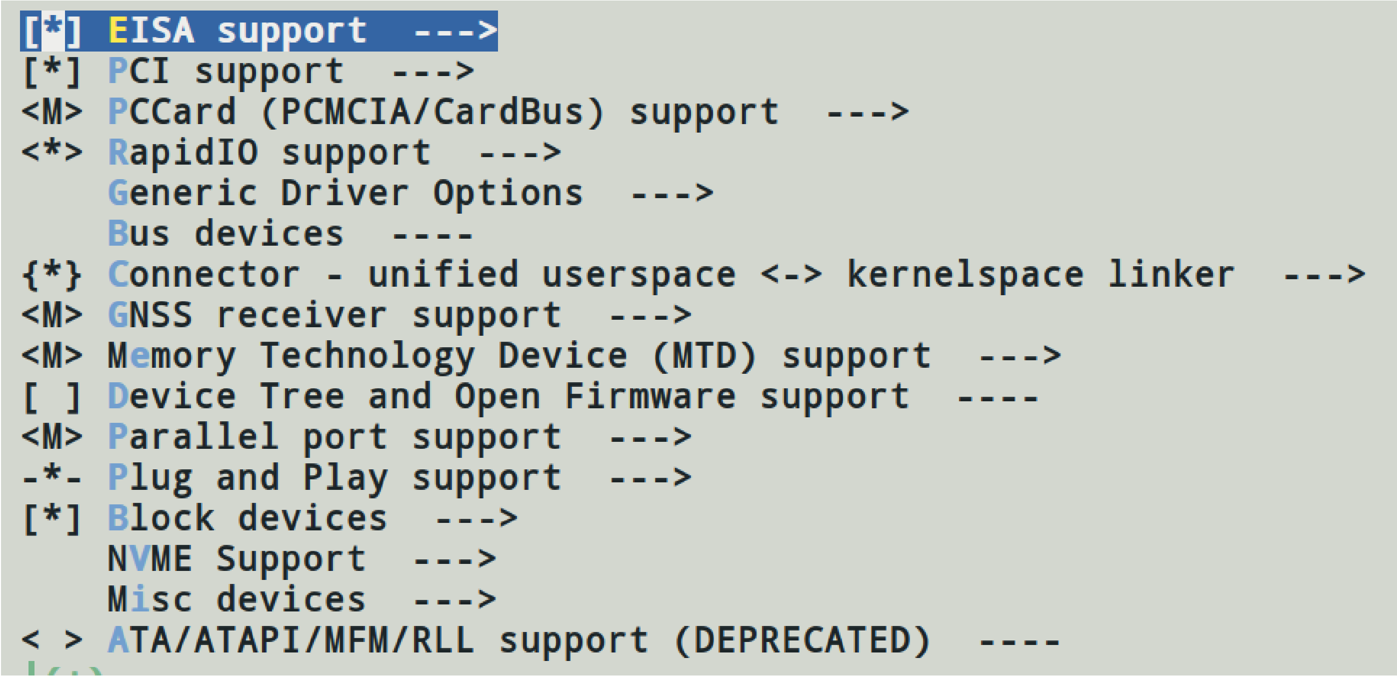
(v) Low (Up to 2x) density storage for compressed pages #兩倍記憶體壓縮
Up to 3x density storage for compressed pages #三倍記憶體壓縮
#記憶體壓縮要認真考，Zswap 也是一種記憶體方法，讀寫硬碟速度太慢

(15) Networking support #網路 盡量不要亂動

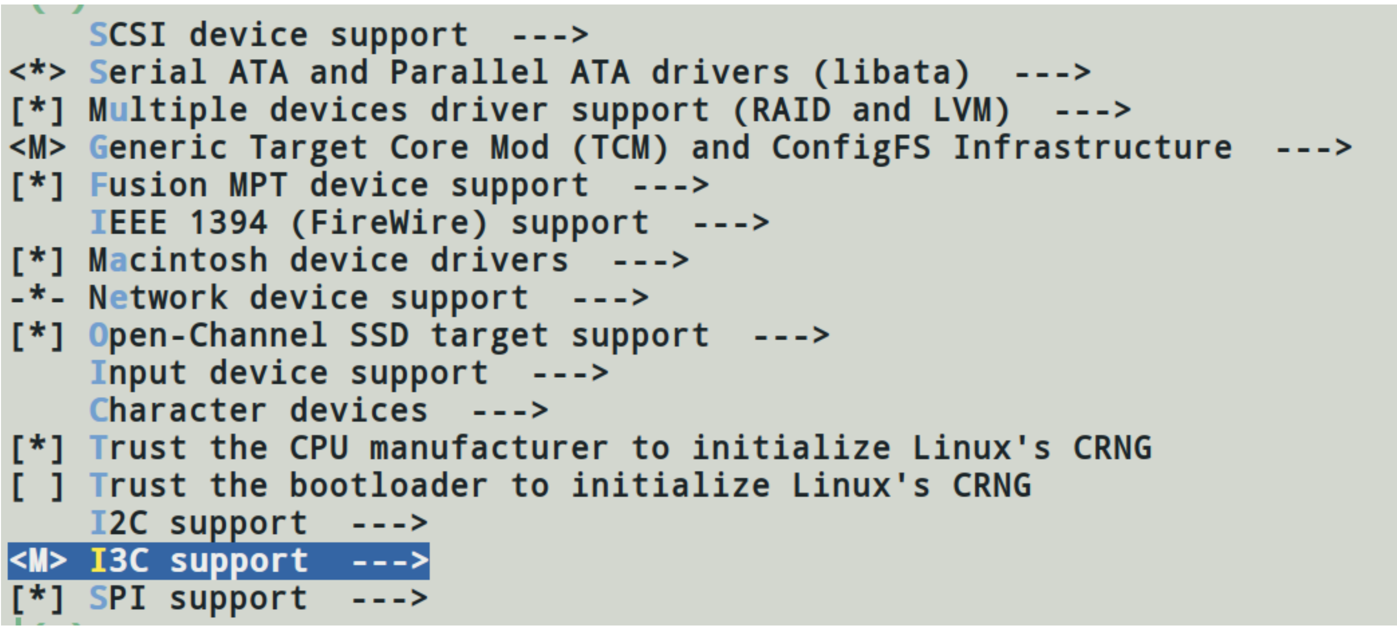
(16) Device Drivers #不要動 目前可以開機的東西已經複製上來 這裡對速度沒有太大影響 只有對編譯時間跟硬碟大小有影響(嵌入式常會改這裡) 可用 lsmod 列出用到的 module 去參考要選的部分



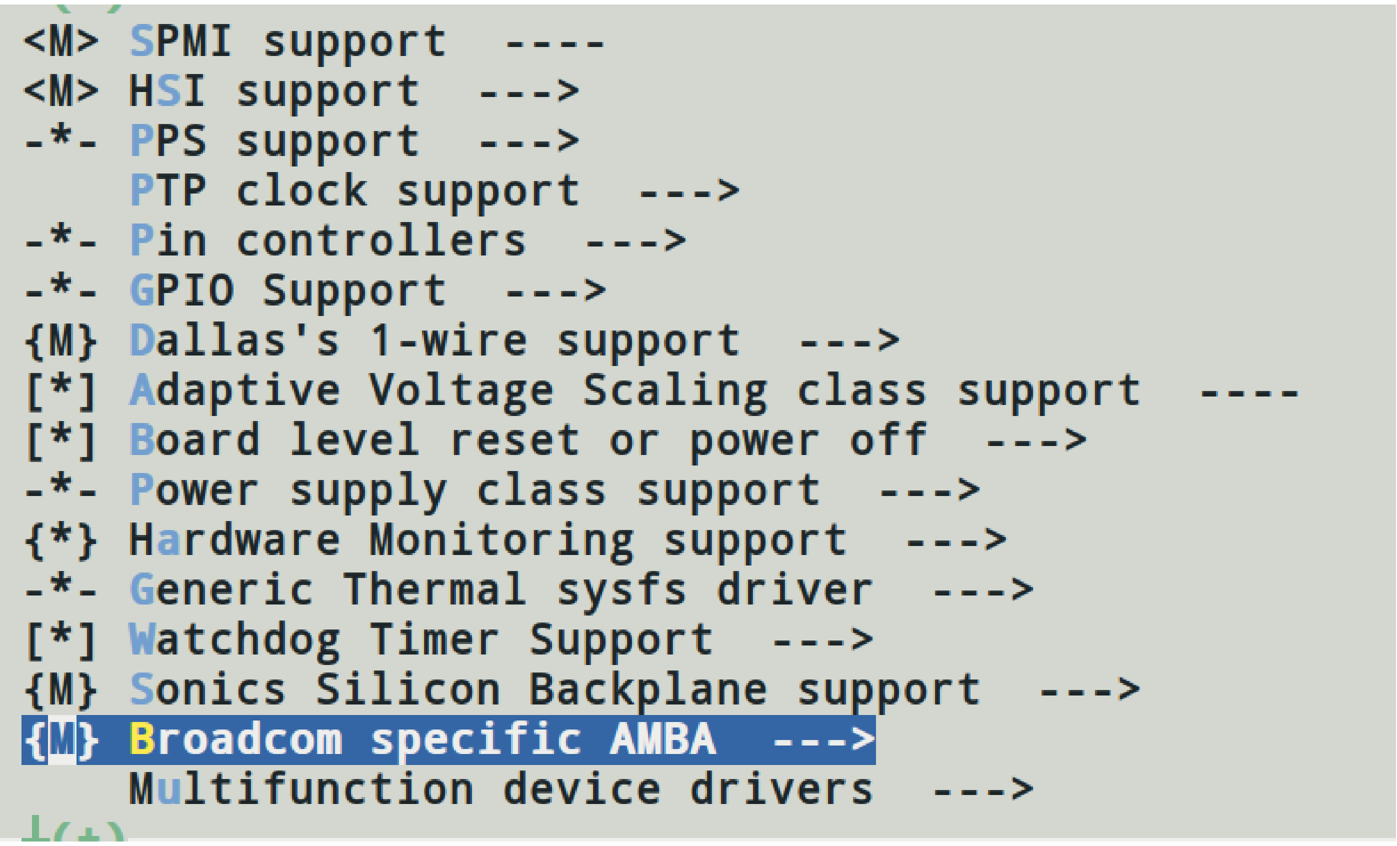
```
Device Drivers  --->
```



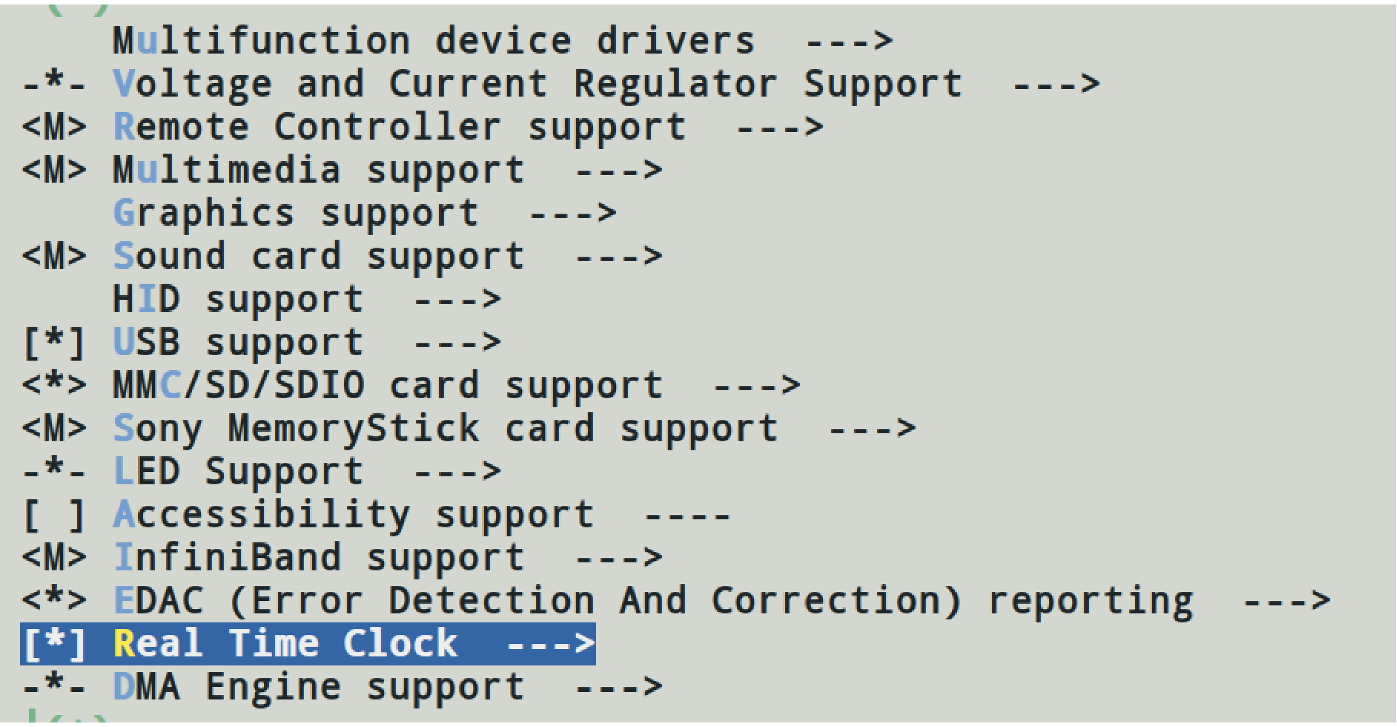
```
[*] EISA support  --->
[*] PCI support  --->
<M> PCCard (PCMCIA/CardBus) support  --->
<*> RapidIO support  --->
    Generic Driver Options  --->
    Bus devices  ----
{*} Connector - unified userspace <-> kernelspace linker  --->
<M> GNSS receiver support  --->
<M> Memory Technology Device (MTD) support  --->
[ ] Device Tree and Open Firmware support  ----
<M> Parallel port support  --->
-*- Plug and Play support  --->
[*] Block devices  --->
    NVME Support  --->
    Misc devices  --->
< > ATA/ATAPI/MFM/RLL support (DEPRECATED)  ----
```



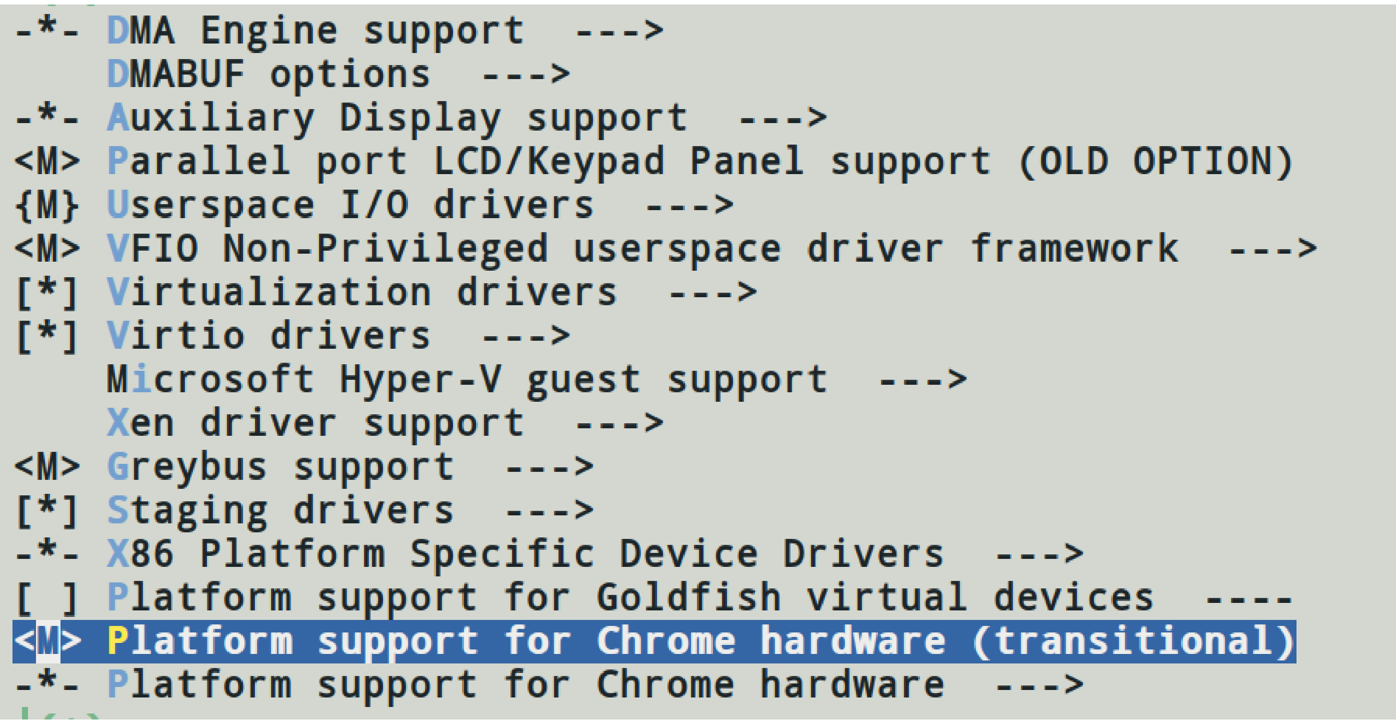
```
    SCSI device support  --->
<*> Serial ATA and Parallel ATA drivers (libata)  --->
[*] Multiple devices driver support (RAID and LVM)  --->
<M> Generic Target Core Mod (TCM) and ConfigFS Infrastructure  --->
[*] Fusion MPT device support  --->
    IEEE 1394 (FireWire) support  --->
[*] Macintosh device drivers  --->
-*- Network device support  --->
[*] Open-Channel SSD target support  --->
    Input device support  --->
    Character devices  --->
[*] Trust the CPU manufacturer to initialize Linux's CRNG
[ ] Trust the bootloader to initialize Linux's CRNG
    I2C support  --->
<M> I3C support  --->
[*] SPI support  --->
```



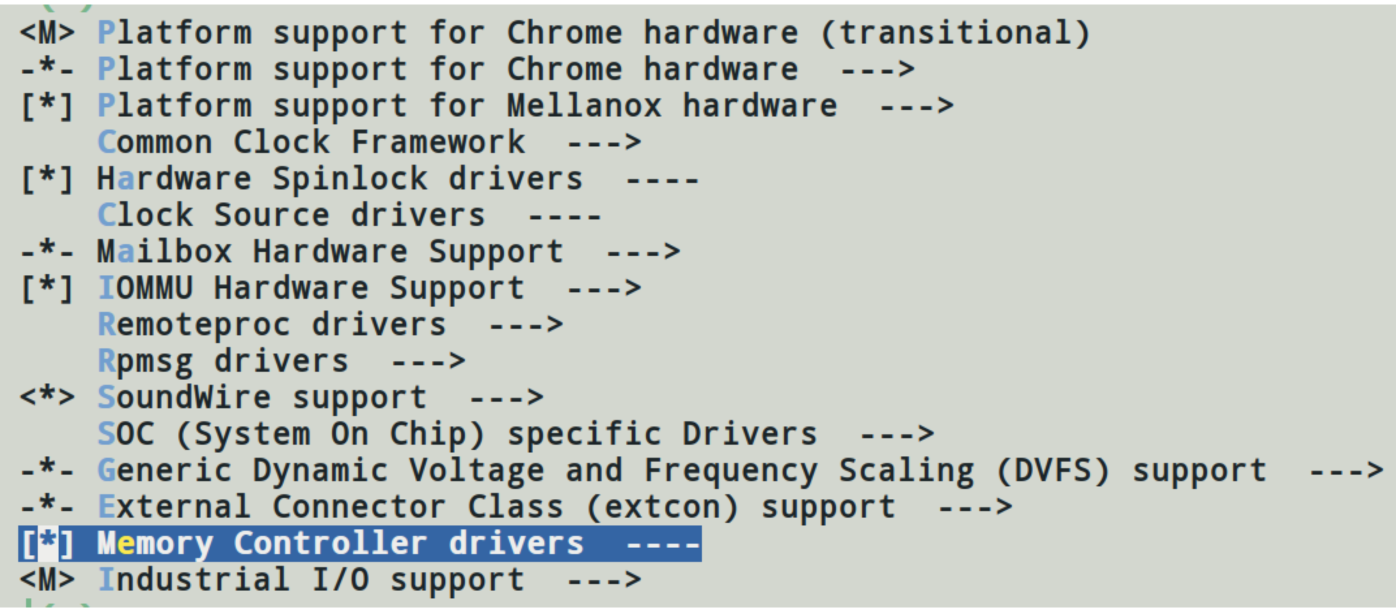
```
<M> SPMI support  ----
<M> HSI support  --->
-*- PPS support  --->
    PTP clock support  --->
-*- Pin controllers  --->
-*- GPIO Support  --->
{M} Dallas's 1-wire support  --->
[*] Adaptive Voltage Scaling class support  ----
[*] Board level reset or power off  --->
-*- Power supply class support  --->
{*} Hardware Monitoring support  --->
-*- Generic Thermal sysfs driver  --->
[*] Watchdog Timer Support  --->
{M} Sonics Silicon Backplane support  --->
{M} Broadcom specific AMBA  --->
    Multifunction device drivers  --->
↓(+)
```



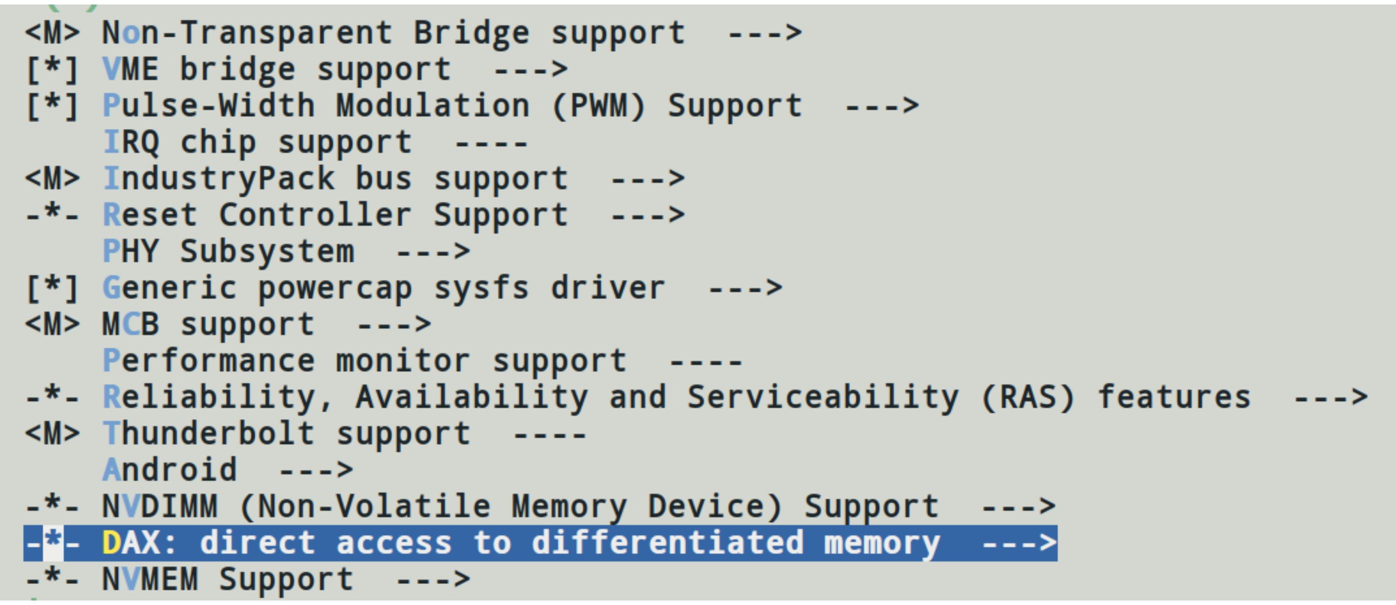
```
    Multifunction device drivers  --->
-*- Voltage and Current Regulator Support  --->
<M> Remote Controller support  --->
<M> Multimedia support  --->
    Graphics support  --->
<M> Sound card support  --->
    HID support  --->
[*] USB support  --->
<*> MMC/SD/SDIO card support  --->
<M> Sony MemoryStick card support  --->
-*- LED Support  --->
[ ] Accessibility support  ----
<M> InfiniBand support  --->
<*> EDAC (Error Detection And Correction) reporting  --->
[*] Real Time Clock  --->
-*- DMA Engine support  --->
↓(+)
```



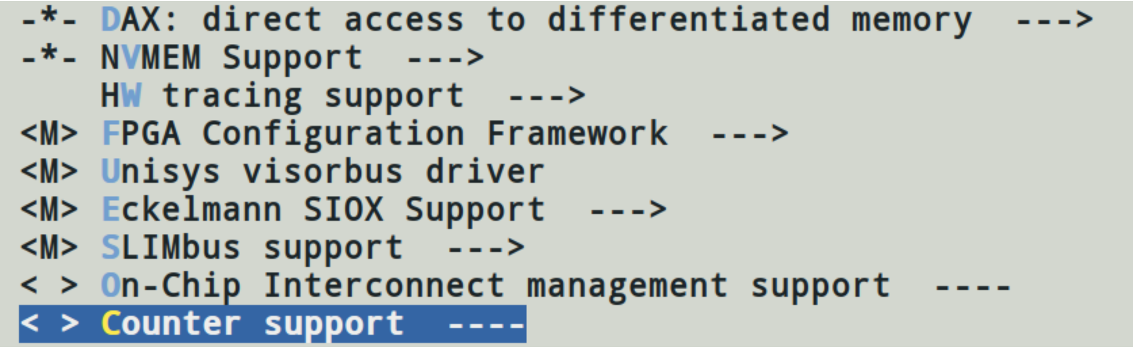
```
-*- DMA Engine support  --->
    DMABUF options  --->
-*- Auxiliary Display support  --->
<M> Parallel port LCD/Keypad Panel support (OLD OPTION)
{M} Userspace I/O drivers  --->
<M> VFIO Non-Privileged userspace driver framework  --->
[*] Virtualization drivers  --->
[*] Virtio drivers  --->
    Microsoft Hyper-V guest support  --->
    Xen driver support  --->
<M> Greybus support  --->
[*] Staging drivers  --->
-*- X86 Platform Specific Device Drivers  --->
[ ] Platform support for Goldfish virtual devices  ----
<M> Platform support for Chrome hardware (transitional)
-*- Platform support for Chrome hardware  --->
```



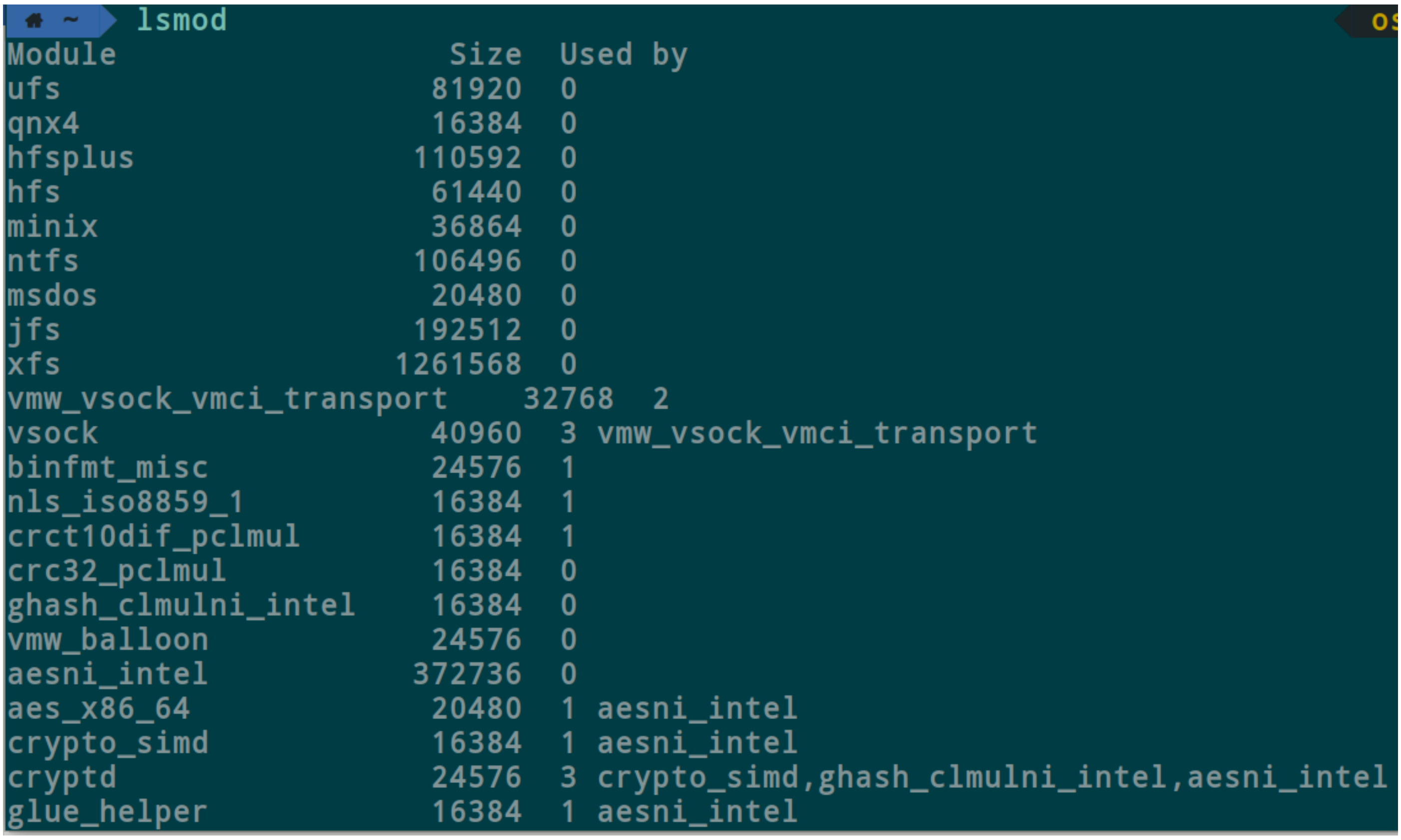
```
<M> Platform support for Chrome hardware (transitional)
-*- Platform support for Chrome hardware  --->
[*] Platform support for Mellanox hardware  --->
    Common Clock Framework  --->
[*] Hardware Spinlock drivers  ----
    Clock Source drivers  ----
-*- Mailbox Hardware Support  --->
[*] IOMMU Hardware Support  --->
    Remoteproc drivers  --->
    Rpmsg drivers  --->
<*> SoundWire support  --->
    SOC (System On Chip) specific Drivers  --->
-*- Generic Dynamic Voltage and Frequency Scaling (DVFS) support  --->
-*- External Connector Class (extcon) support  --->
[*] Memory Controller drivers  ----
<M> Industrial I/O support  --->
```



```
<M> Non-Transparent Bridge support  --->
[*] VME bridge support  --->
[*] Pulse-Width Modulation (PWM) Support  --->
    IRQ chip support  ----
<M> IndustryPack bus support  --->
-*- Reset Controller Support  --->
    PHY Subsystem  --->
[*] Generic powercap sysfs driver  --->
<M> MCB support  --->
    Performance monitor support  ----
-*- Reliability, Availability and Serviceability (RAS) features  --->
<M> Thunderbolt support  ----
    Android  --->
-*- NVDIMM (Non-Volatile Memory Device) Support  --->
-*- DAX: direct access to differentiated memory  --->
-*- NVMEM Support  --->
```



```
-*- DAX: direct access to differentiated memory  --->
-*- NVMEM Support  --->
    HW tracing support  --->
<M> FPGA Configuration Framework  --->
<M> Unisys visorbus driver
<M> Eckelmann SIOX Support  --->
<M> SLIMbus support  --->
< > On-Chip Interconnect management support  ----
< > Counter support  ----
```

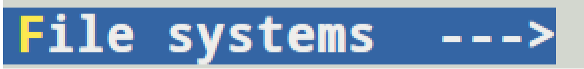


```
~  lsmod
Module                  Size  Used by
ufs                    81920  0
qnx4                   16384  0
hfsplus               110592  0
hfs                    61440  0
minix                  36864  0
ntfs                  106496  0
msdos                  20480  0
jfs                   192512  0
xfs                  1261568  0
vmw_vsock_vmci_transport    32768  2
vsock                  40960  3 vmw_vsock_vmci_transport
binfmt_misc            24576  1
nls_iso8859_1          16384  1
crct10dif_pclmul       16384  1
crc32_pclmul           16384  0
ghash_clmulni_intel    16384  0
vmw_balloon            24576  0
aesni_intel           372736  0
aes_x86_64             20480  1 aesni_intel
crypto_simd            16384  1 aesni_intel
cryptd                 24576  3 crypto_simd,ghash_clmulni_intel,aesni_intel
glue_helper            16384  1 aesni_intel
```

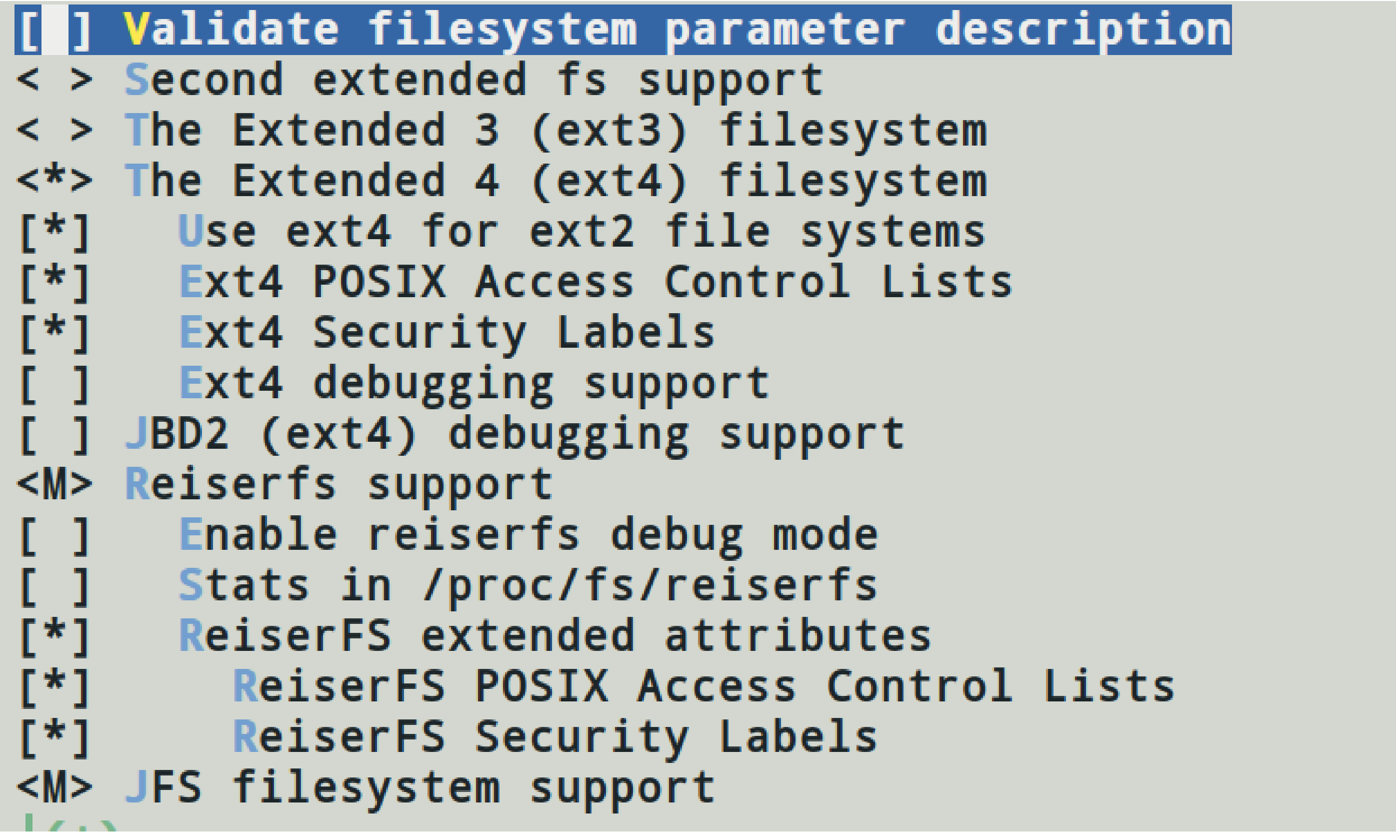
(17)File systems #有用到的選起來

(i) ext4

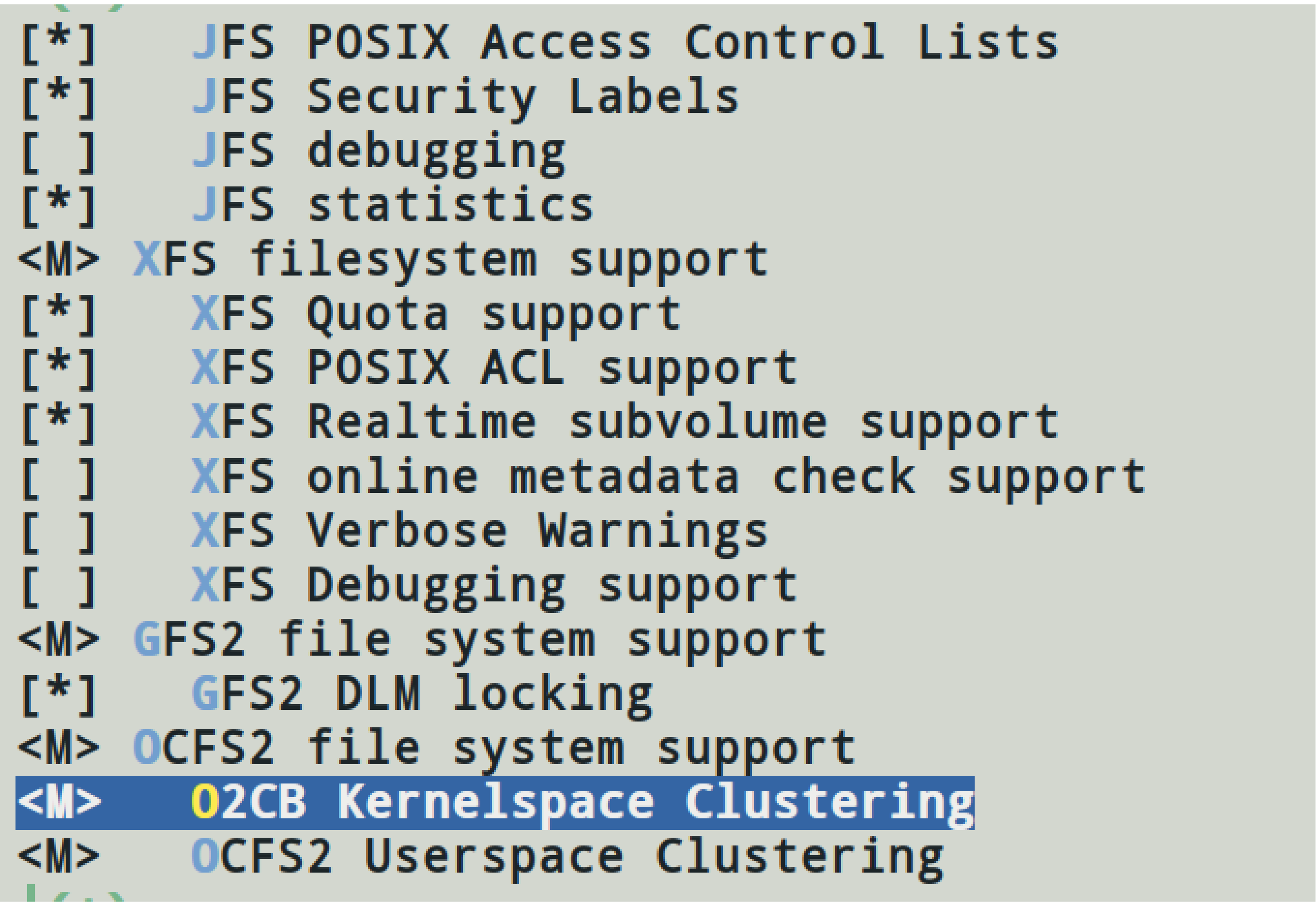
(ii)btrfs　#若主功能編譯成 M 子功能也會變成 M



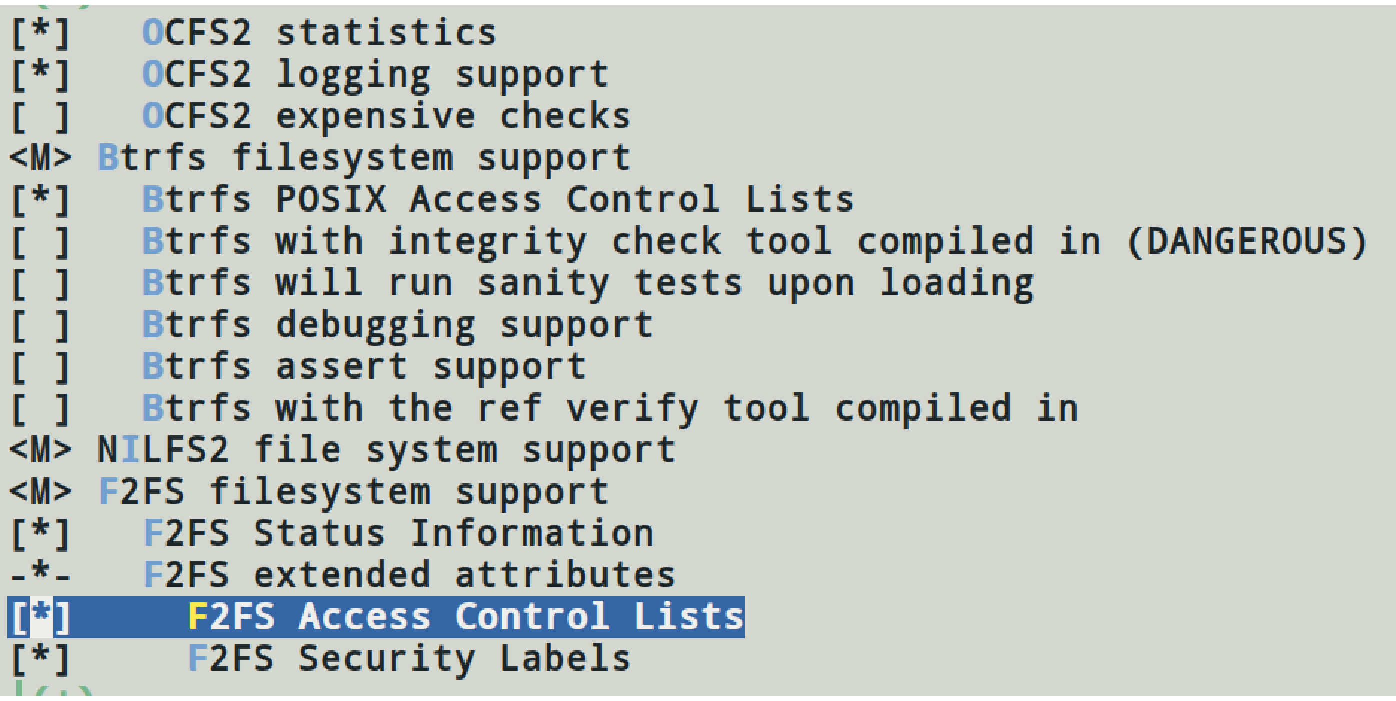
```
File systems  --->
```



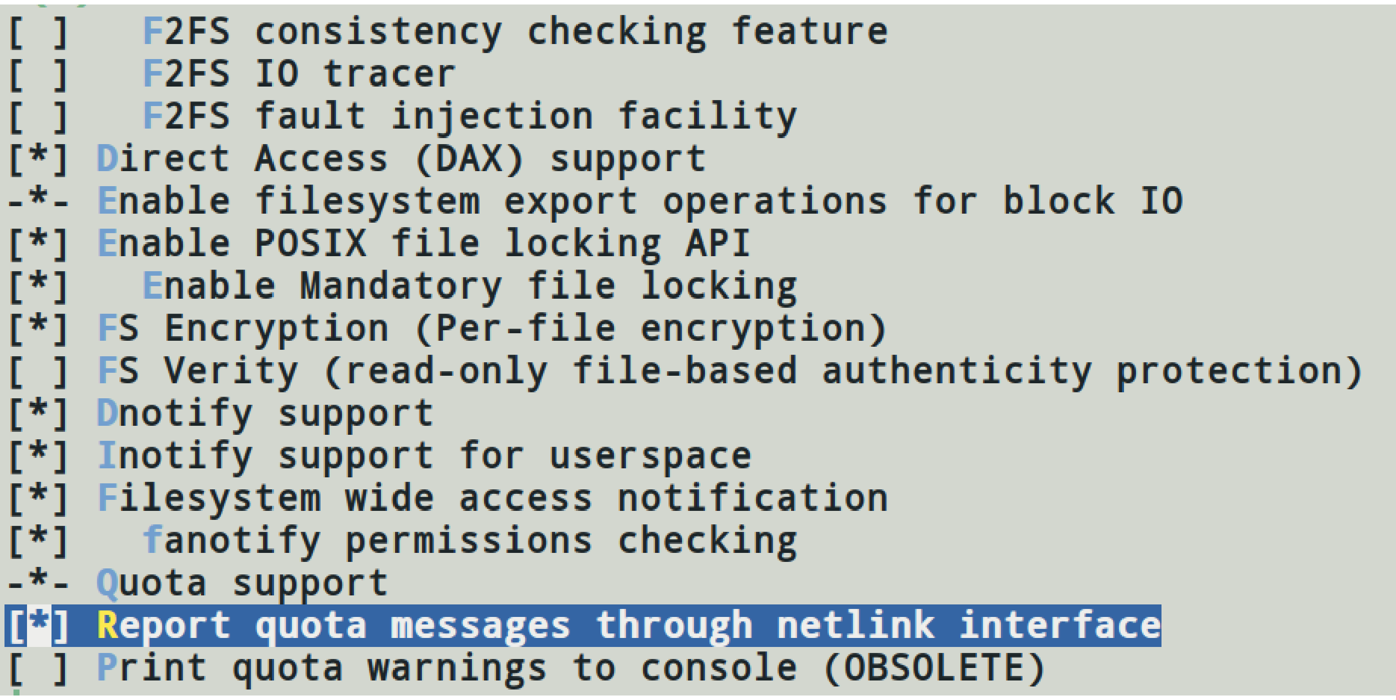
```
[ ] Validate filesystem parameter description
< > Second extended fs support
< > The Extended 3 (ext3) filesystem
<*> The Extended 4 (ext4) filesystem
[*]   Use ext4 for ext2 file systems
[*]   Ext4 POSIX Access Control Lists
[*]   Ext4 Security Labels
[ ]   Ext4 debugging support
[ ] JBD2 (ext4) debugging support
<M> Reiserfs support
[ ]   Enable reiserfs debug mode
[ ]   Stats in /proc/fs/reiserfs
[*]   ReiserFS extended attributes
[*]     ReiserFS POSIX Access Control Lists
[*]     ReiserFS Security Labels
<M> JFS filesystem support
```



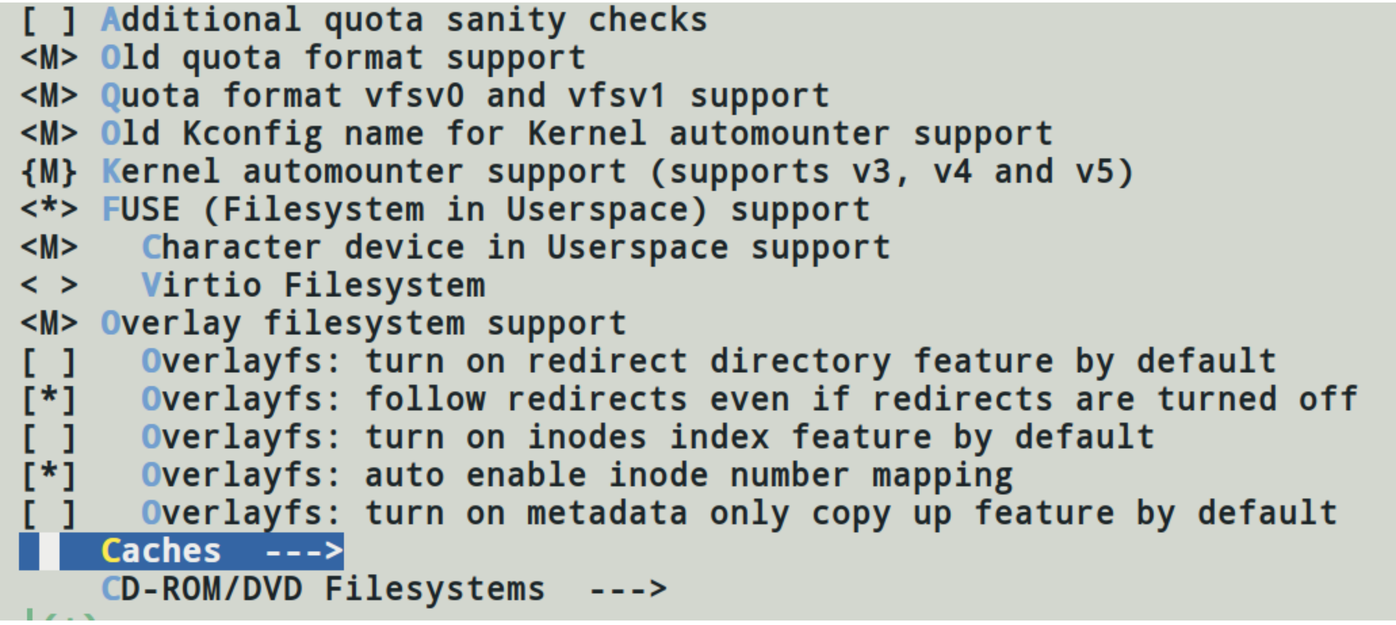
```
[*]   JFS POSIX Access Control Lists
[*]   JFS Security Labels
[ ]   JFS debugging
[*]   JFS statistics
<M> XFS filesystem support
[*]   XFS Quota support
[*]   XFS POSIX ACL support
[*]   XFS Realtime subvolume support
[ ]   XFS online metadata check support
[ ]   XFS Verbose Warnings
[ ]   XFS Debugging support
<M> GFS2 file system support
[*]   GFS2 DLM locking
<M> OCFS2 file system support
<M>   O2CB Kernelspace Clustering
<M>   OCFS2 Userspace Clustering
```



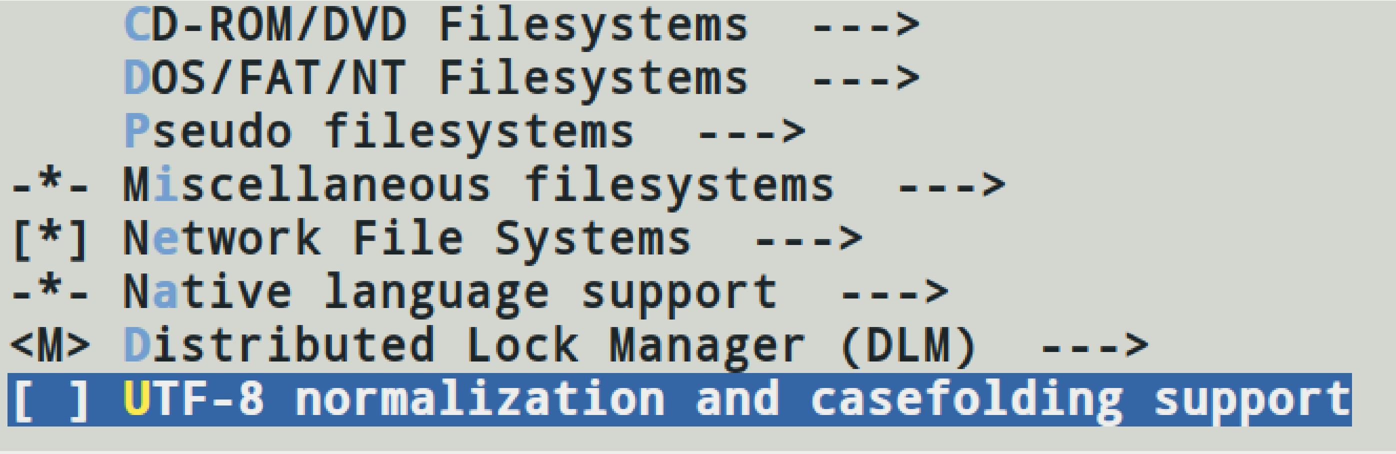
```
[*]   OCFS2 statistics
[*]   OCFS2 logging support
[ ]   OCFS2 expensive checks
<M> Btrfs filesystem support
[*]   Btrfs POSIX Access Control Lists
[ ]   Btrfs with integrity check tool compiled in (DANGEROUS)
[ ]   Btrfs will run sanity tests upon loading
[ ]   Btrfs debugging support
[ ]   Btrfs assert support
[ ]   Btrfs with the ref verify tool compiled in
<M> NILFS2 file system support
<M> F2FS filesystem support
[*]   F2FS Status Information
-*-   F2FS extended attributes
[*]     F2FS Access Control Lists
[*]     F2FS Security Labels
```



```
[ ]   F2FS consistency checking feature
[ ]   F2FS IO tracer
[ ]   F2FS fault injection facility
[*] Direct Access (DAX) support
-*- Enable filesystem export operations for block IO
[*] Enable POSIX file locking API
[*]   Enable Mandatory file locking
[*] FS Encryption (Per-file encryption)
[ ] FS Verity (read-only file-based authenticity protection)
[*] Dnotify support
[*] Inotify support for userspace
[*] Filesystem wide access notification
[*]   fanotify permissions checking
-*- Quota support
[*] Report quota messages through netlink interface
[ ] Print quota warnings to console (OBSOLETE)
```



```
[ ] Additional quota sanity checks
<M> Old quota format support
<M> Quota format vfsv0 and vfsv1 support
<M> Old Kconfig name for Kernel automounter support
{M} Kernel automounter support (supports v3, v4 and v5)
<*> FUSE (Filesystem in Userspace) support
<M>   Character device in Userspace support
< >   Virtio Filesystem
<M> Overlay filesystem support
[ ]   Overlayfs: turn on redirect directory feature by default
[*]   Overlayfs: follow redirects even if redirects are turned off
[ ]   Overlayfs: turn on inodes index feature by default
[*]   Overlayfs: auto enable inode number mapping
[ ]   Overlayfs: turn on metadata only copy up feature by default
    Caches  --->
    CD-ROM/DVD Filesystems  --->
```



```
    CD-ROM/DVD Filesystems  --->
    DOS/FAT/NT Filesystems  --->
    Pseudo filesystems  --->
-*- Miscellaneous filesystems  --->
[*] Network File Systems  --->
-*- Native language support  --->
<M> Distributed Lock Manager (DLM)  --->
[ ] UTF-8 normalization and casefolding support
```

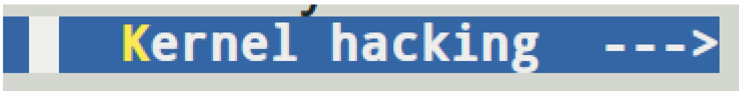
(18)Security options #不要動

(19)Cryptographic API #不要動

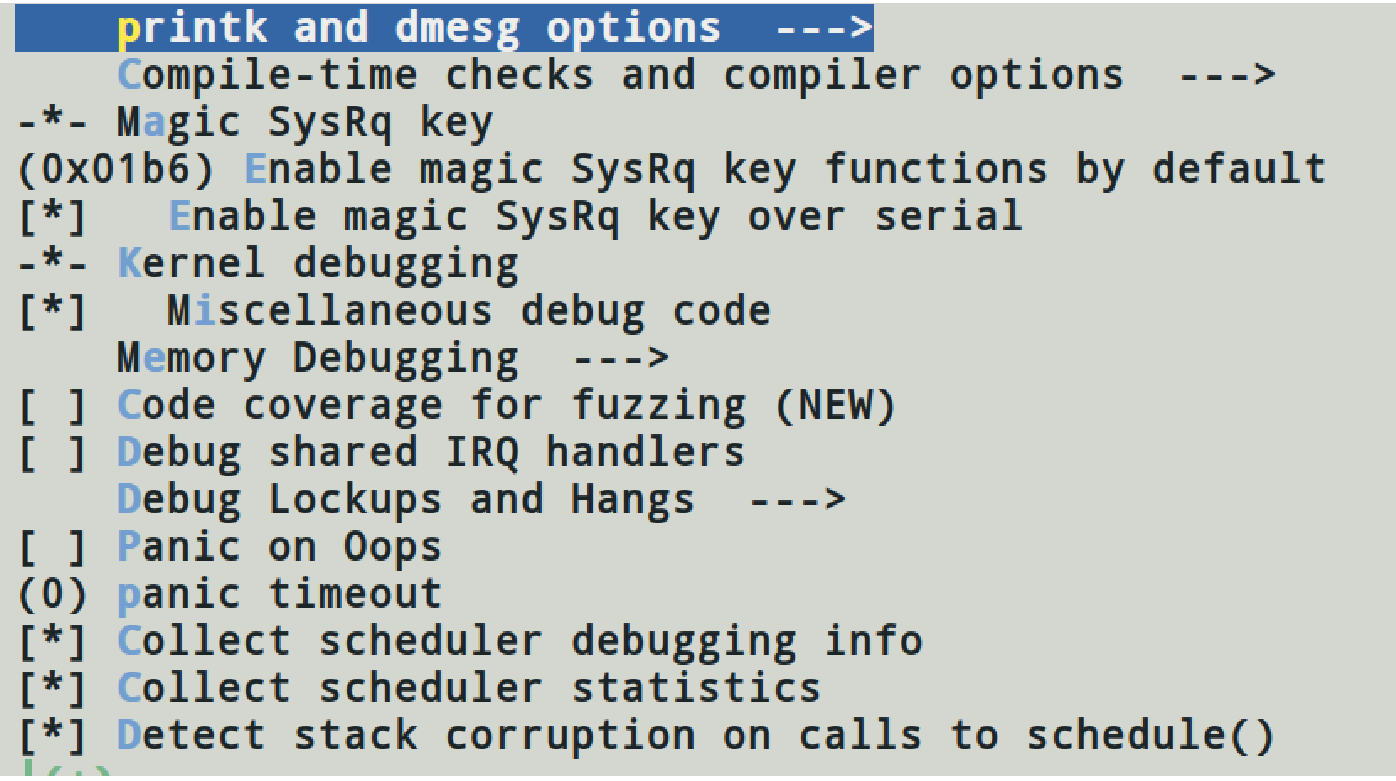
(20)Library routines #不要動

(21) Kernel hacking #不能亂選　速度會變慢

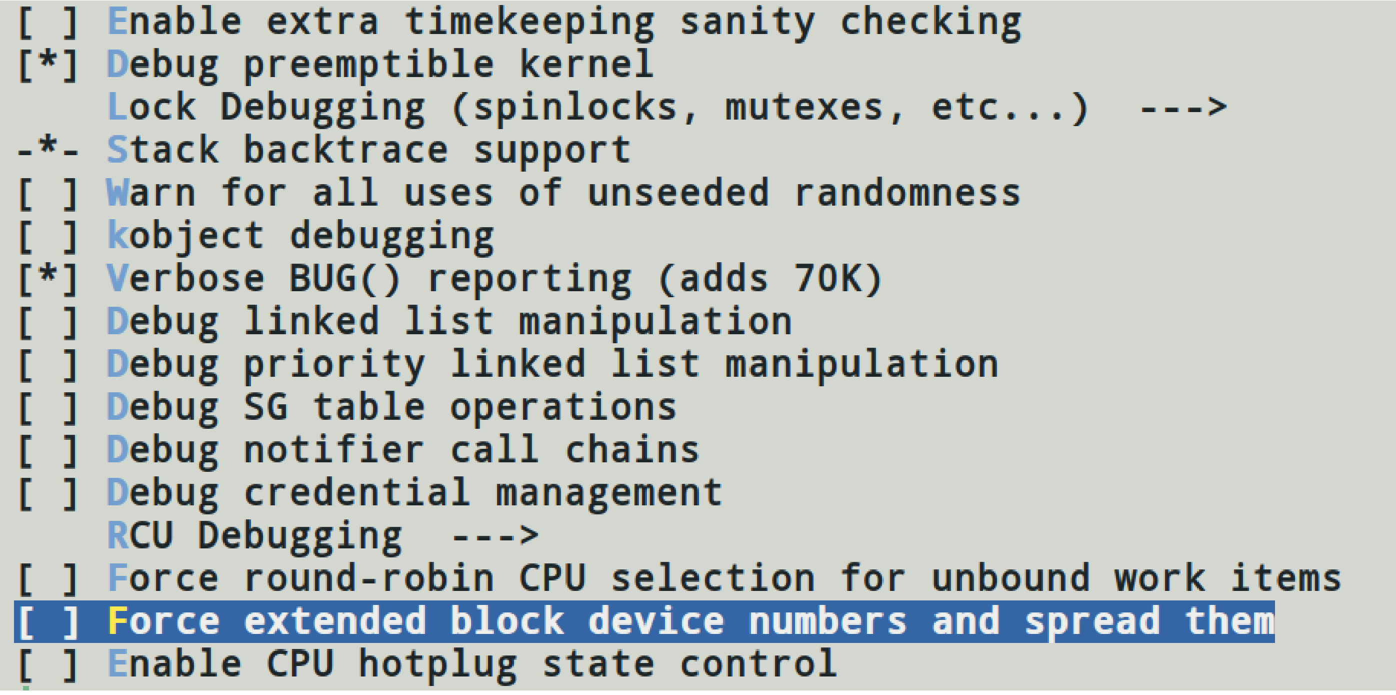
(i) Enable magic SysRq key over serial　#sysreq 一組特別的 key 可以觸動 kernel 緊急 shut down 用 類似 windos ctrl+alt+delete



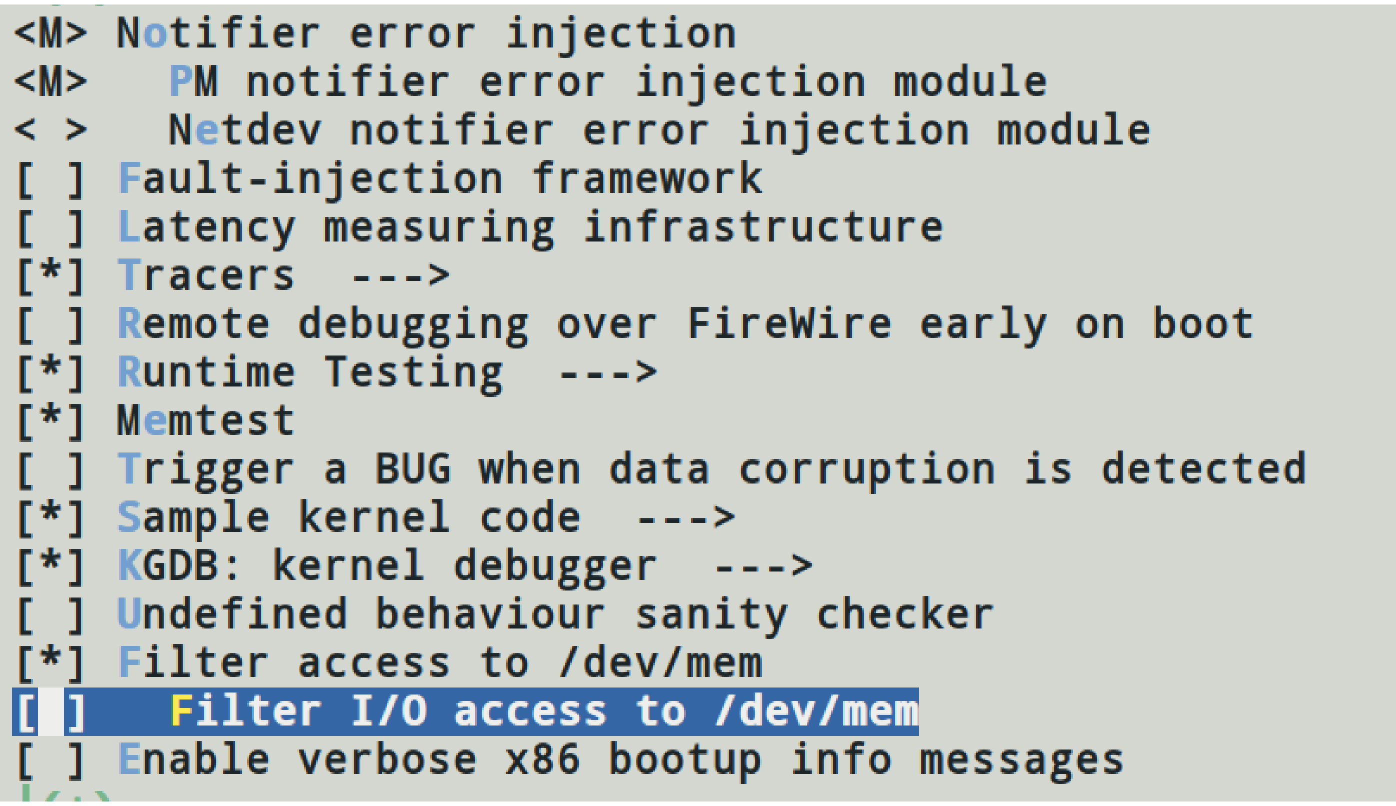
```
    Kernel hacking  --->
```



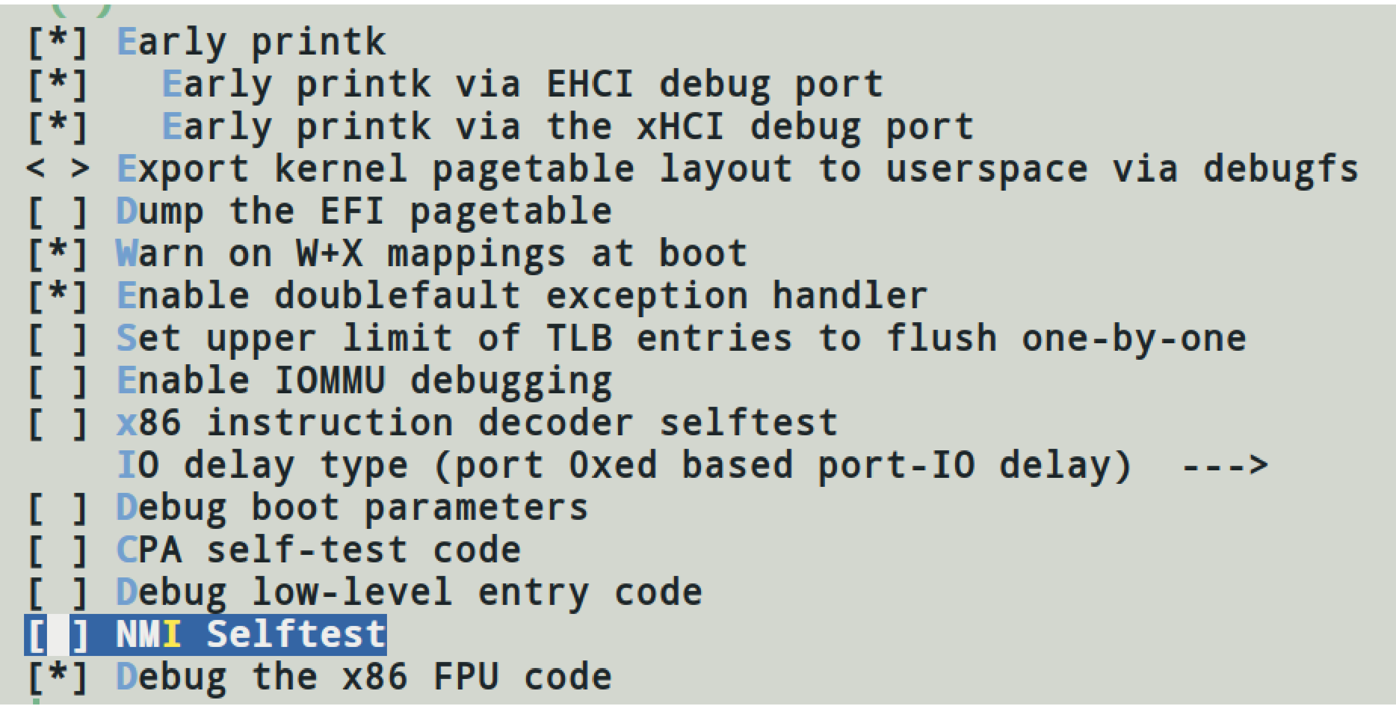
```
    printk and dmesg options  --->
    Compile-time checks and compiler options  --->
-*- Magic SysRq key
(0x01b6) Enable magic SysRq key functions by default
[*]   Enable magic SysRq key over serial
-*- Kernel debugging
[*]   Miscellaneous debug code
    Memory Debugging  --->
[ ] Code coverage for fuzzing (NEW)
[ ] Debug shared IRQ handlers
    Debug Lockups and Hangs  --->
[ ] Panic on Oops
(0) panic timeout
[*] Collect scheduler debugging info
[*] Collect scheduler statistics
[*] Detect stack corruption on calls to schedule()
```



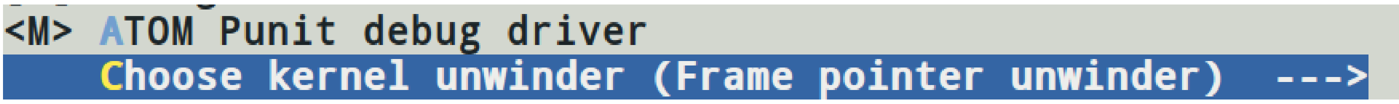
```
[ ] Enable extra timekeeping sanity checking
[*] Debug preemptible kernel
    Lock Debugging (spinlocks, mutexes, etc...)  --->
-*- Stack backtrace support
[ ] Warn for all uses of unseeded randomness
[ ] kobject debugging
[*] Verbose BUG() reporting (adds 70K)
[ ] Debug linked list manipulation
[ ] Debug priority linked list manipulation
[ ] Debug SG table operations
[ ] Debug notifier call chains
[ ] Debug credential management
    RCU Debugging  --->
[ ] Force round-robin CPU selection for unbound work items
[ ] Force extended block device numbers and spread them
[ ] Enable CPU hotplug state control
```



```
<M> Notifier error injection
<M>   PM notifier error injection module
< >   Netdev notifier error injection module
[ ] Fault-injection framework
[ ] Latency measuring infrastructure
[*] Tracers  --->
[ ] Remote debugging over FireWire early on boot
[*] Runtime Testing  --->
[*] Memtest
[ ] Trigger a BUG when data corruption is detected
[*] Sample kernel code  --->
[*] KGDB: kernel debugger  --->
[ ] Undefined behaviour sanity checker
[*] Filter access to /dev/mem
[ ]   Filter I/O access to /dev/mem
[ ] Enable verbose x86 bootup info messages
```



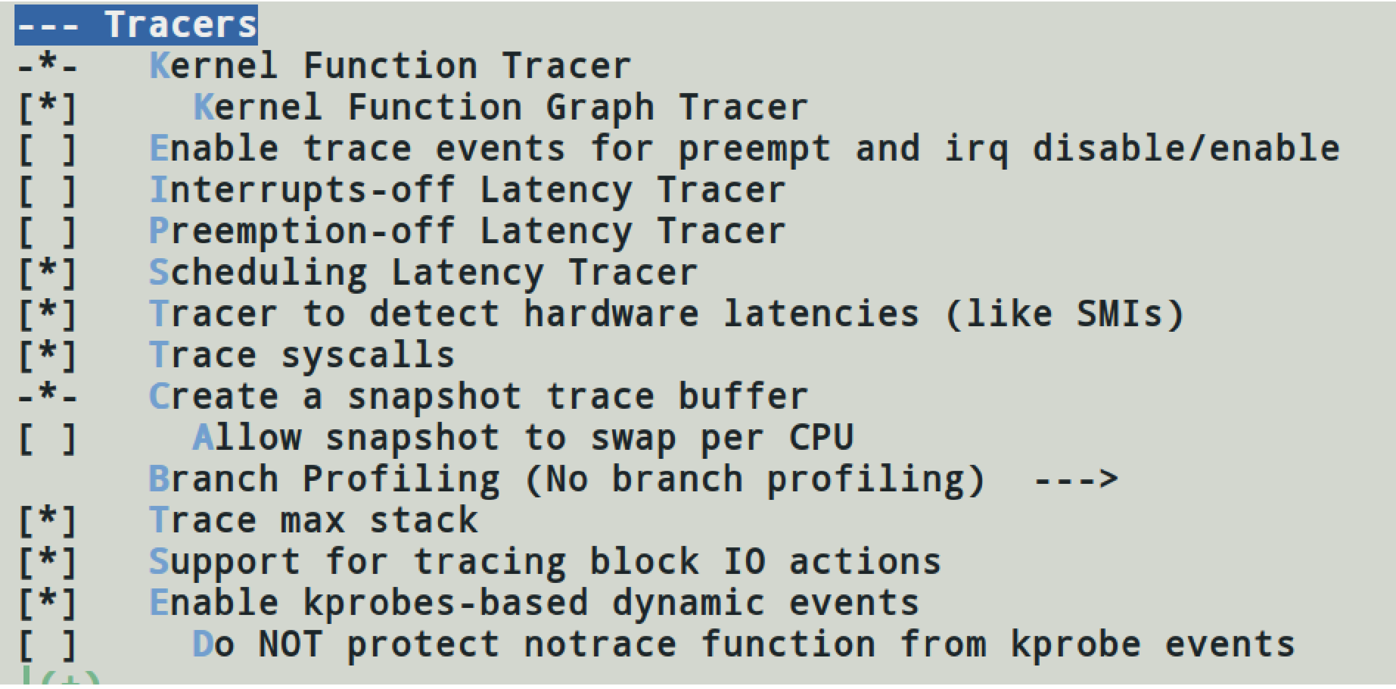
```
[*] Early printk
[*]   Early printk via EHCI debug port
[*]   Early printk via the xHCI debug port
< > Export kernel pagetable layout to userspace via debugfs
[ ] Dump the EFI pagetable
[*] Warn on W+X mappings at boot
[*] Enable doublefault exception handler
[ ] Set upper limit of TLB entries to flush one-by-one
[ ] Enable IOMMU debugging
[ ] x86 instruction decoder selftest
    IO delay type (port 0xed based port-IO delay)  --->
[ ] Debug boot parameters
[ ] CPA self-test code
[ ] Debug low-level entry code
[ ] NMI Selftest
[*] Debug the x86 FPU code
```



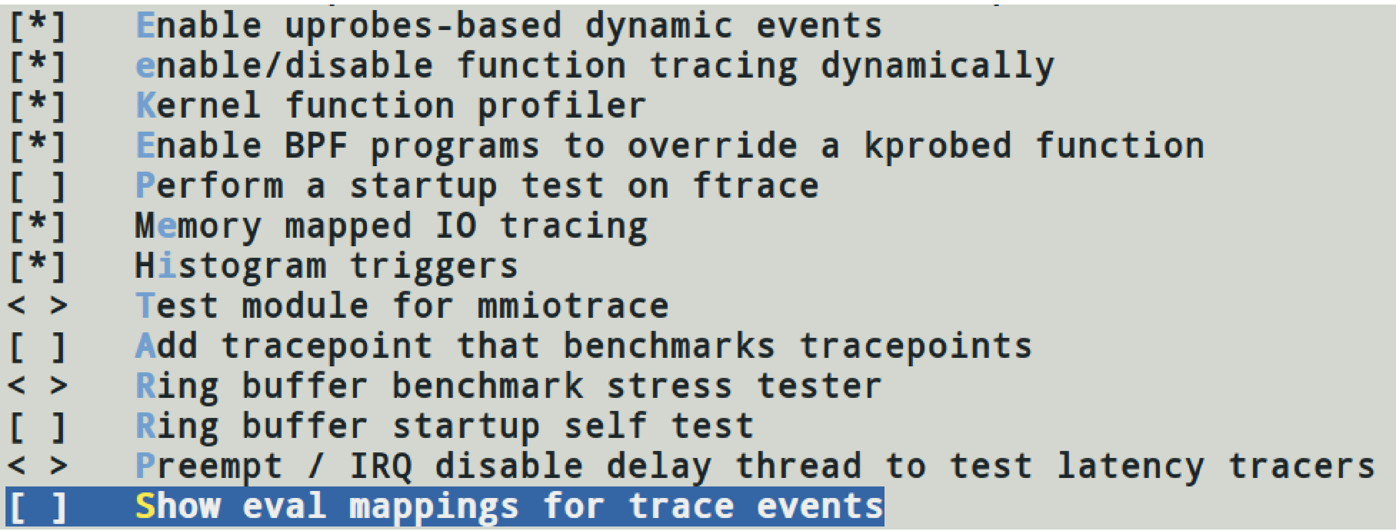
```
<M> ATOM Punit debug driver
    Choose kernel unwinder (Frame pointer unwinder)  --->
```

(ii)Tracers #有需要的再打開來看 預設外多半會影響效能 有 ftrace 功能

(a) Enable BPF programs to override a kprobed function #Berkeley Packet Filter 為動態載入器 允許把程式碼放到 kernel 裡面 只能往前執行 指標不能亂指 也稱作 EBPF



```
--- Tracers
-*-    Kernel Function Tracer
[*]      Kernel Function Graph Tracer
[ ]    Enable trace events for preempt and irq disable/enable
[ ]    Interrupts-off Latency Tracer
[ ]    Preemption-off Latency Tracer
[*]    Scheduling Latency Tracer
[*]    Tracer to detect hardware latencies (like SMIs)
[*]    Trace syscalls
-*-    Create a snapshot trace buffer
[ ]      Allow snapshot to swap per CPU
       Branch Profiling (No branch profiling)  --->
[*]    Trace max stack
[*]    Support for tracing block IO actions
[*]    Enable kprobes-based dynamic events
[ ]      Do NOT protect notrace function from kprobe events
```



```
[*]    Enable uprobes-based dynamic events
[*]    enable/disable function tracing dynamically
[*]    Kernel function profiler
[*]    Enable BPF programs to override a kprobed function
[ ]    Perform a startup test on ftrace
[*]    Memory mapped IO tracing
[*]    Histogram triggers
< >    Test module for mmiotrace
[ ]    Add tracepoint that benchmarks tracepoints
< >    Ring buffer benchmark stress tester
[ ]    Ring buffer startup self test
< >    Preempt / IRQ disable delay thread to test latency tracers
[ ]    Show eval mappings for trace events
```

6. 編譯

(1) make modules_install #比較古老方式 要刪除要一個子目錄慢慢刪很麻煩 推薦用安裝包要刪除只要用 dpkg uninstall

詳細步驟也可參考：

https://www.linux.com/tutorials/how-compile-linux-kernel-0/

https://www.youtube.com/watch?v=ZBDuvrVckik