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：

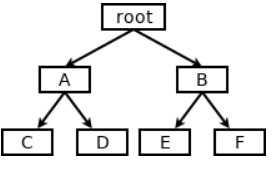
One challenge of controlling I/O resources is the lack of reliability of trivial cost metrics. Bandwidth and iops can be off by orders of magnitude depending on the device type and I/O pattern. This is challenging for the I/O cgroup controllers: while io.latency provides the capability to comprehensively 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

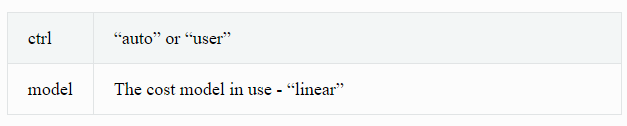
EX：若A權重為100，B為300，那麼B可以拿到75% I/O頻寬



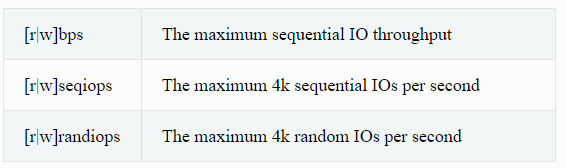
再來進到了blk-iocost的io.cost.model部分

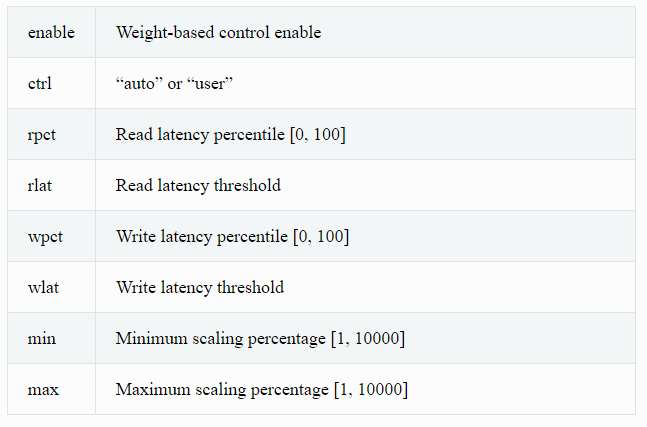
具有讀寫能力的套件會存在於root cgroup部分，利用了CONFIG\_BLK\_CGROUP\_IOCOST

(I/O成本模型)，以下為定義一開始進入cgroup的參數

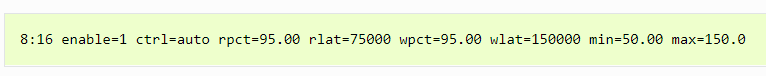


當ctrl為自動的時候，Kernel可以動態修改任何參數，當為user時不能更改，參數定義如下：



最後介紹blk-iocost的io.cost.qos部分

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：



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>

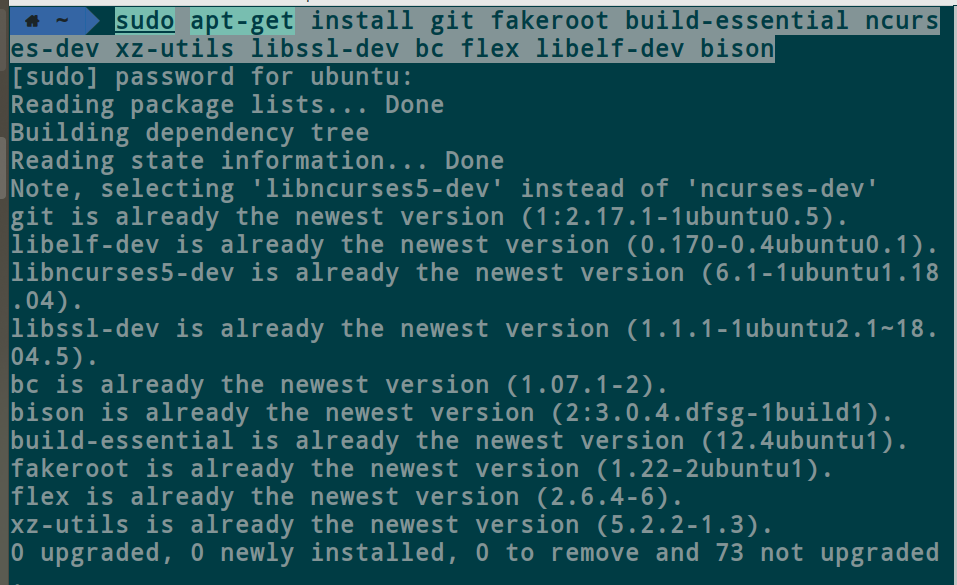
<https://lwn.net/Articles/792256/>

//以下為youtube老師大略介紹的筆記

3.安裝相關套件

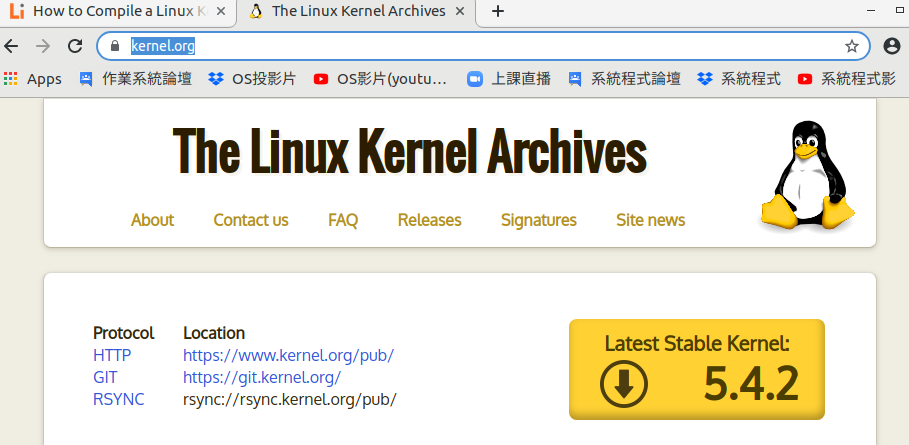
sudo apt-get install git fakeroot build-essential ncurses-dev xz-utils libssl-dev bc flex libelf-dev bison

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

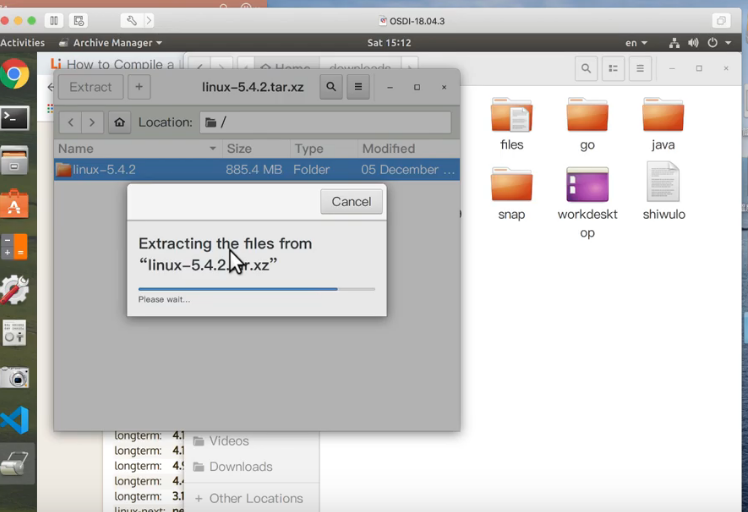


4.下載5.4.2 kernel

(1)<https://www.kernel.org/> 選5.4.2



(2)解壓縮到home



5.製造config

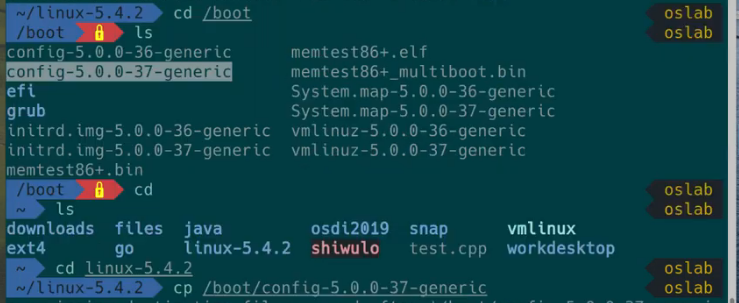
(1)保證視窗 80↑\*25↑字元(此為最古老視窗，也是現代正規20萬以上伺服器常使用的，極其穩定好Debug，常使用D-sub接頭)

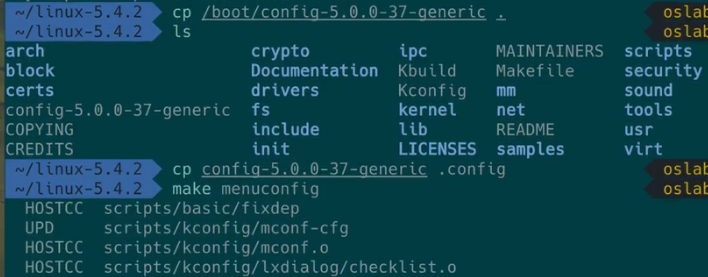
(2)複製config

cd boot ##找到config-5.0.0-36-generic

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

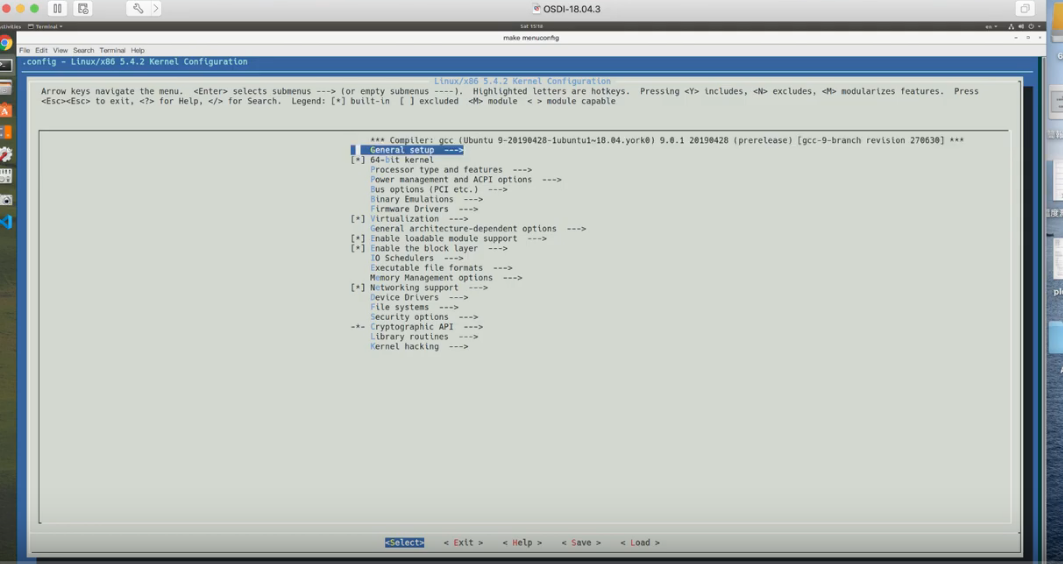
cp /boot/config-5.0.0-36-generic .config #內部make使用



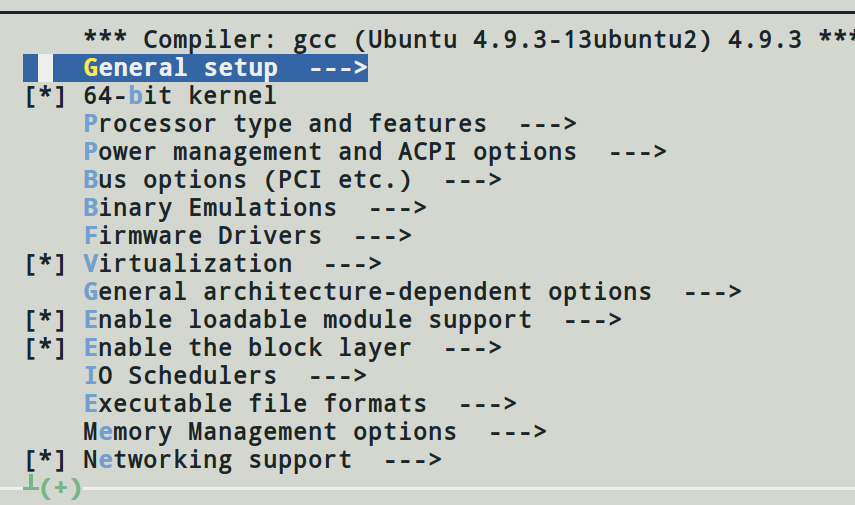


6. make menuconfig

(1) make menuconfig



(2)General setup



(3)設定

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

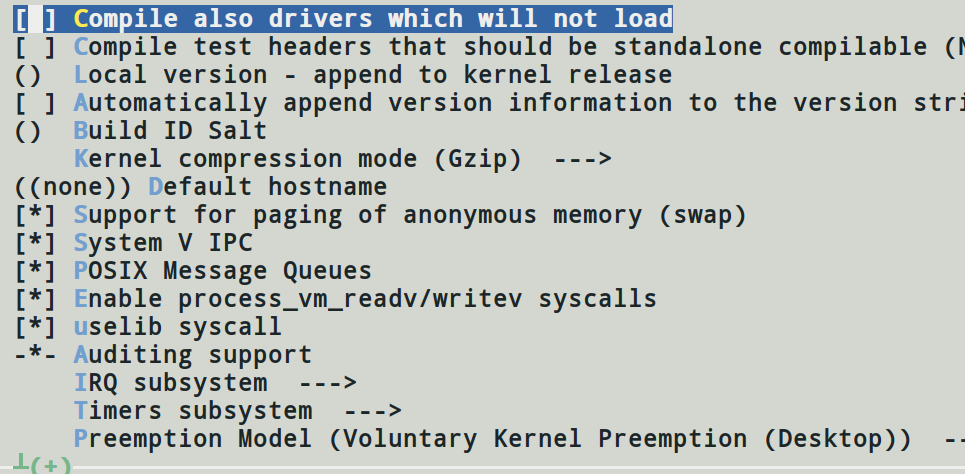
(ii)System V IPC # five IPC

(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

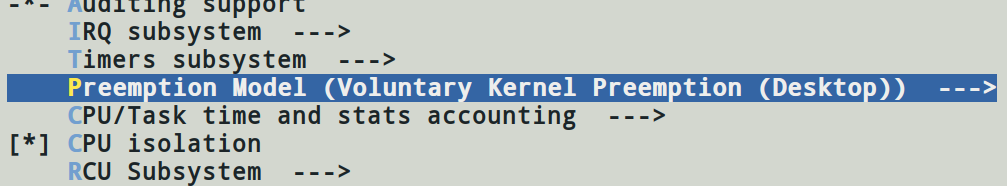


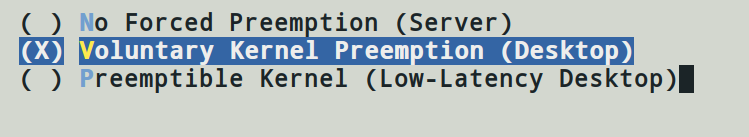
(vii)Preemption Model (Voluntary Kernel Preemption (Desktop)) --->

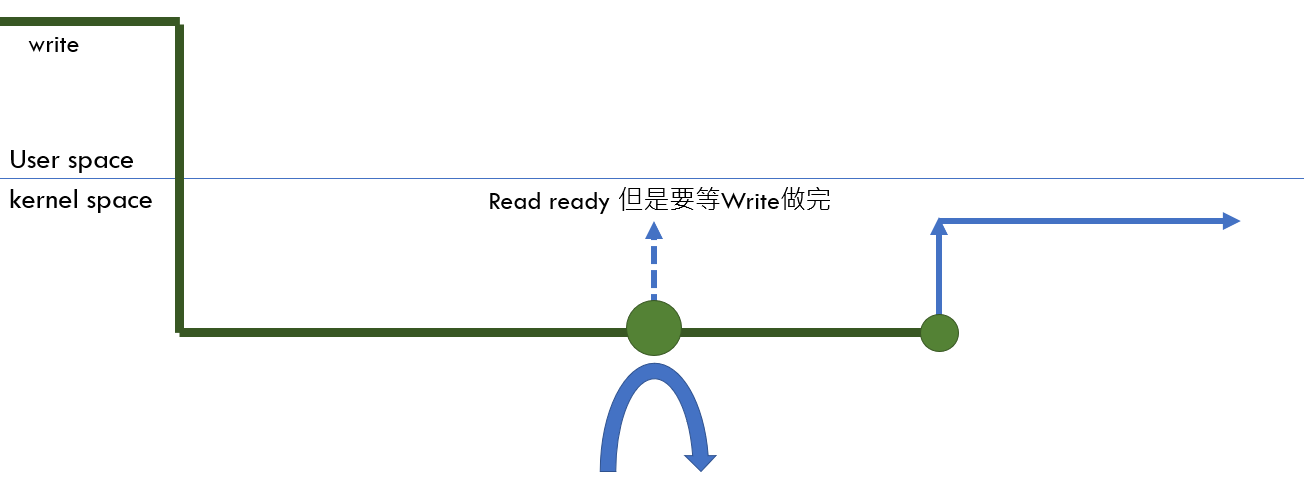
(a)No Forced Preemption (Server) #在kernel space遇到中斷部會馬上執行，先做LOG紀錄

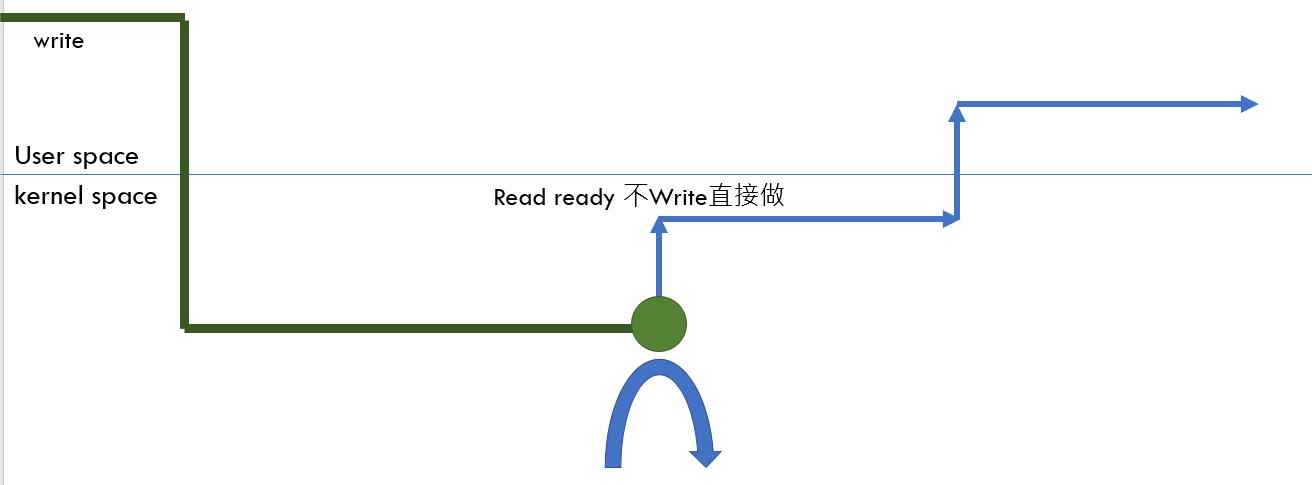
(b)Voluntary Kernel Preemption (Desktop)

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





 No Forced Preemption (會記錄不做中斷)

 Preemptible Kernel (Low-Latency

Desktop)

(viii) NUMA支援



(ix)用來做容器container

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

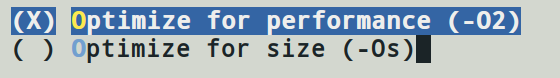
Namespaces support #在container中第一個process id 是1非真的1，只是用來在container 中作為priority用



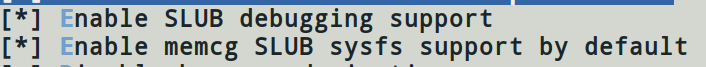
(x) Compiler optimization level (Optimize for performance (-O2))

要選-O2 因為到-O3會有向量指令集，在kernel中用不到





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





(xii)只要有randomization跟安全性有關



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



(4) Processor type and features

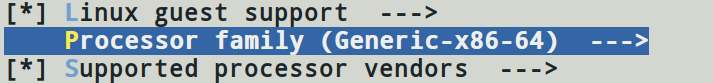


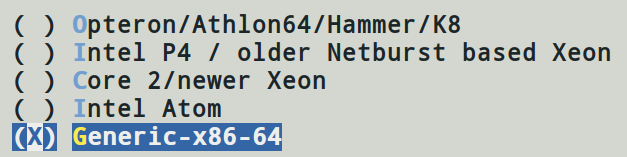
(i)DMA

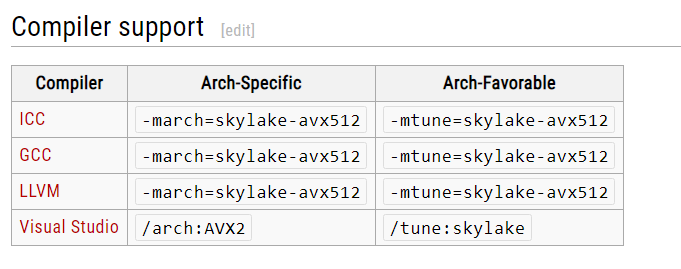


(ii) Processor family (Generic-x86-64)

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





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



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



(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



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

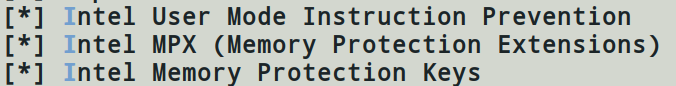
(v)NVIMMS #用在伺服器上 有多種形式 其中一種為停電以後 上面有一顆電池會把DRAM上資料寫到DRAM的FALSH中，下次正常開機後，主機板會發號司令叫DRAM把FALSH中資料還原到DRAM



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



(vii)Intel安全相關模組



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

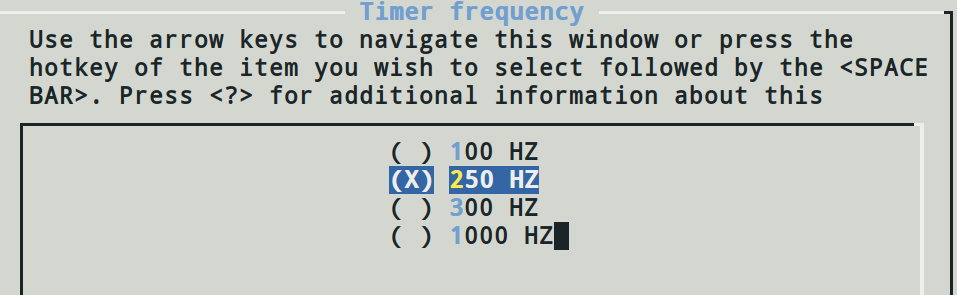


(viii)UEFI #



(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





(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)



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



(xii)KXSLR #保護Kernel



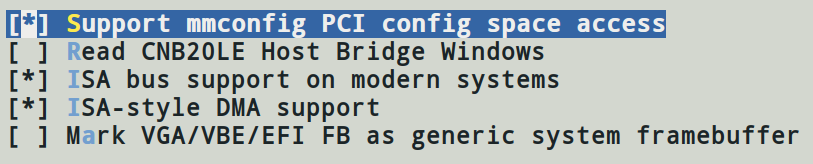
(5) Power management and ACPI options #跟省電和高效能有關再進去調整



(6) Bus options (PCI etc.) #不要動

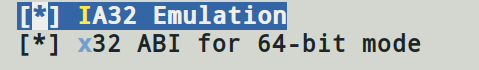
#若會用到framebuffer可打開 它為舊型圖形介面驅動程式(伺服器 因每個pixel去化效能低)





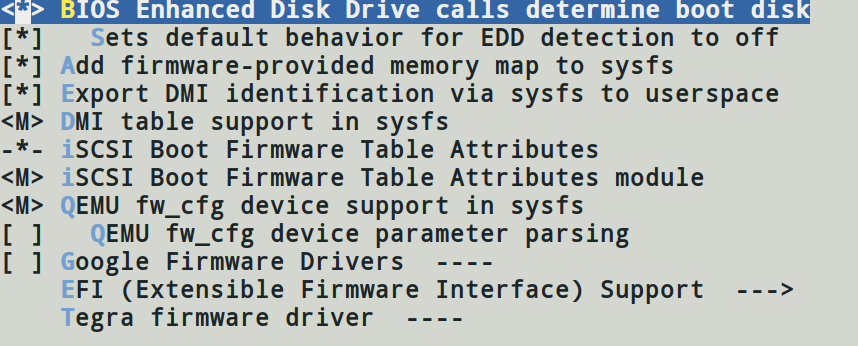
(7) Binary Emulations #支援32位元





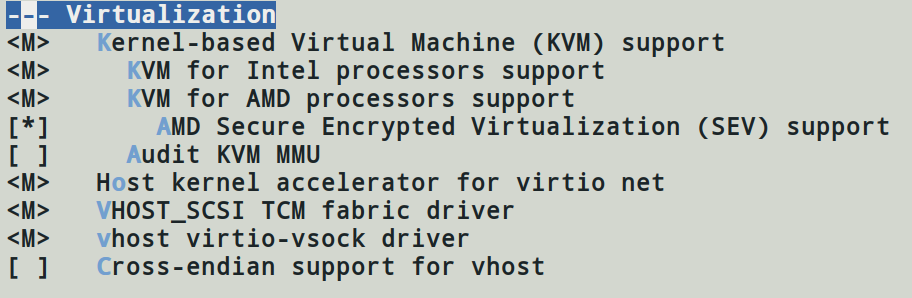
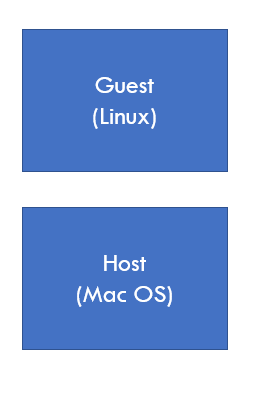
(8) Firmware Drivers #韌體driver 盡量不要動





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



(10) General architecture-dependent options



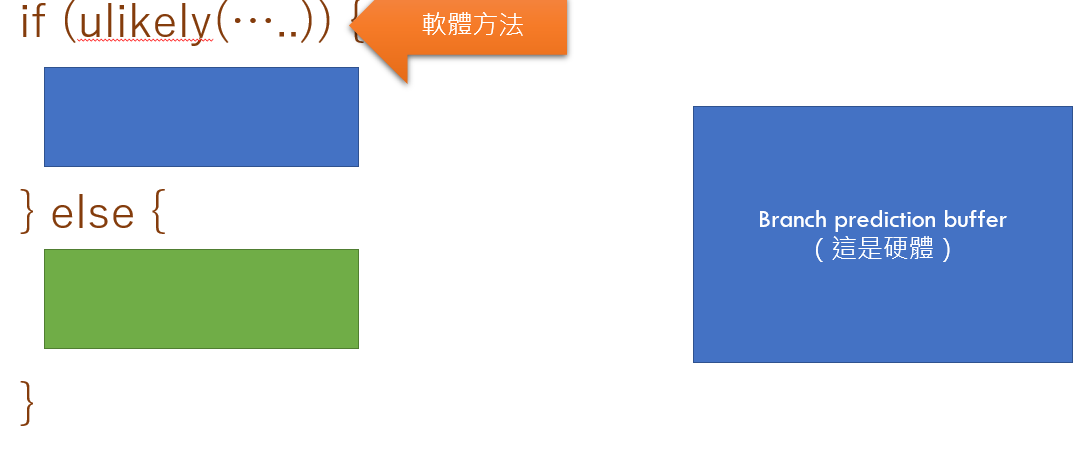
(i) OProfile system profiling #選M是編譯成Module 會動態載入到Kernel 類似瀏覽器插件，編成\* 一開始就進入kernel，如果是檔案系統盡量要編譯成\*，怕一開始找不到



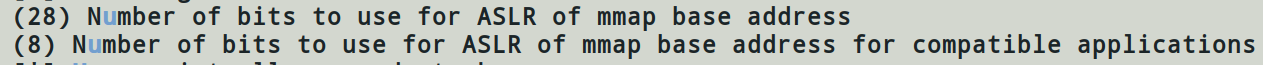
(ii)對likely和unlikely做優化 #likely代表這個case執行機會高 unlikely相反

#CPU 有branch prediction buffer 通常程式行為會跟上次一樣例如while就是種likely





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

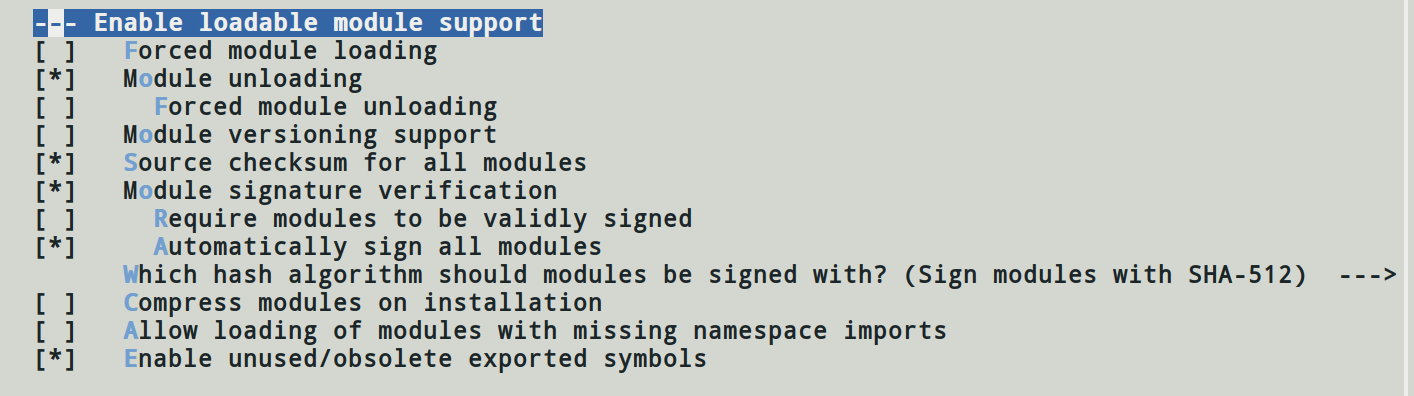


(iv) Locking event counts collection #有多隨機



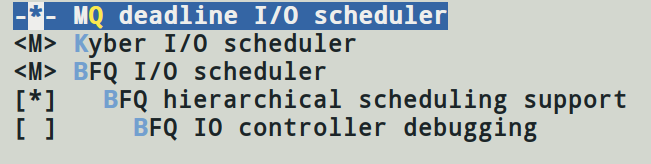
(11) Enable loadable module support #一定要選 動態載入模組 USB 網卡等等 嵌入式系統不用選





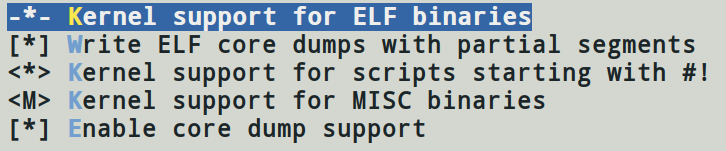
(12) IO Schedulers # 比較不重要，因為像NVME Queue 有2048個 排程完全交給控制器排程





(13) Executable file formats #支援執行檔





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

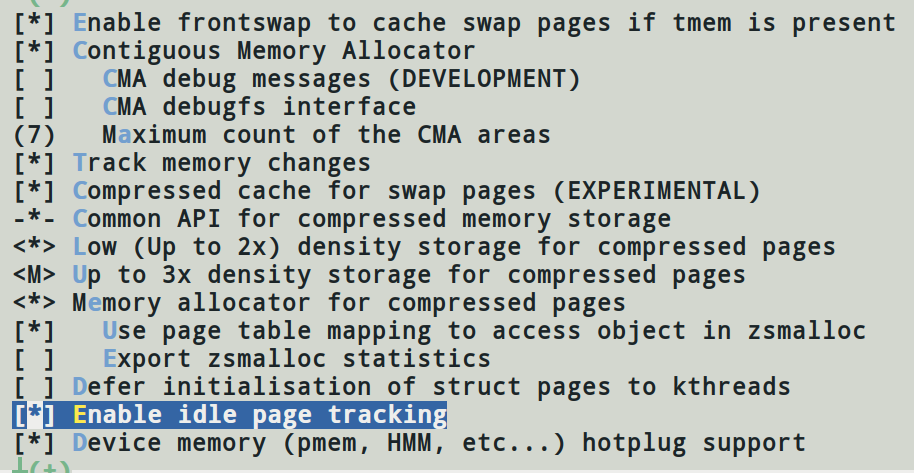
(ii) Kernel support for scripts starting with #! #支援shell script

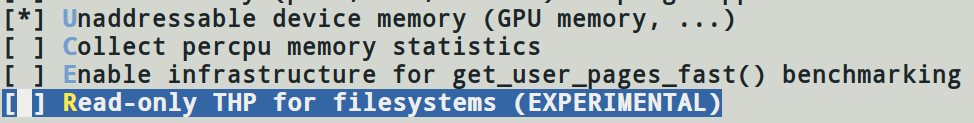
(iii) Enable core dump support #支援core dump kernel掛掉可以寫東西出去

(14) Memory Management options #記憶體管理









1. Allow for balloon memory compaction/migration #記憶體migration 用氣球壓縮
2. 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
3. Enable cleancache driver to cache clean pages if tmem is present #cleancache driver
4. Contiguous Memory Allocator #連續記憶體分配
5. Low (Up to 2x) density storage for compressed pages #兩倍記憶體壓縮

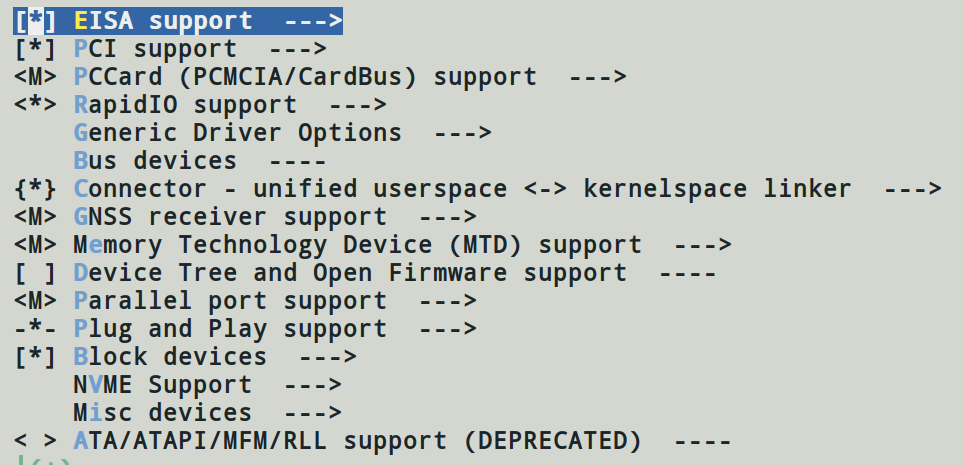
Up to 3x density storage for compressed pages #三倍記憶體壓縮

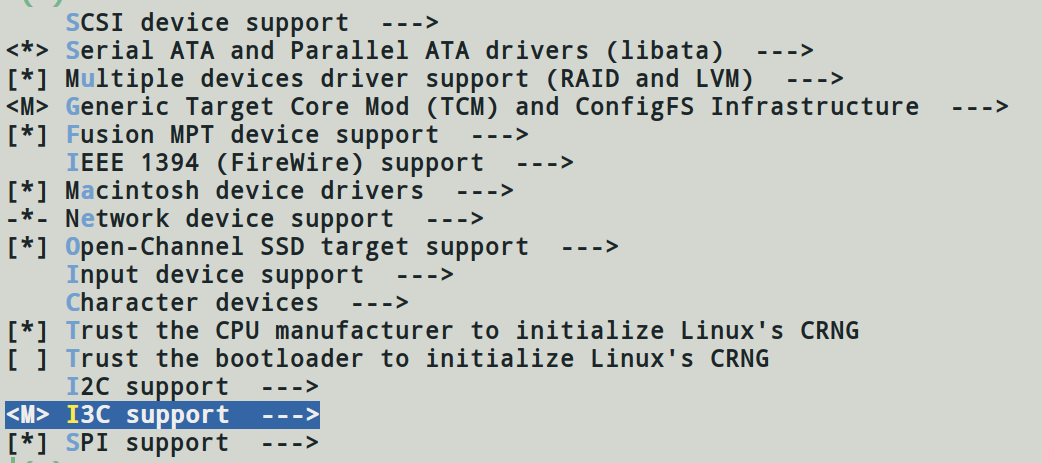
#記憶體壓縮要認真考，Zswap 也是一種記憶體方法，讀寫硬碟速度太慢

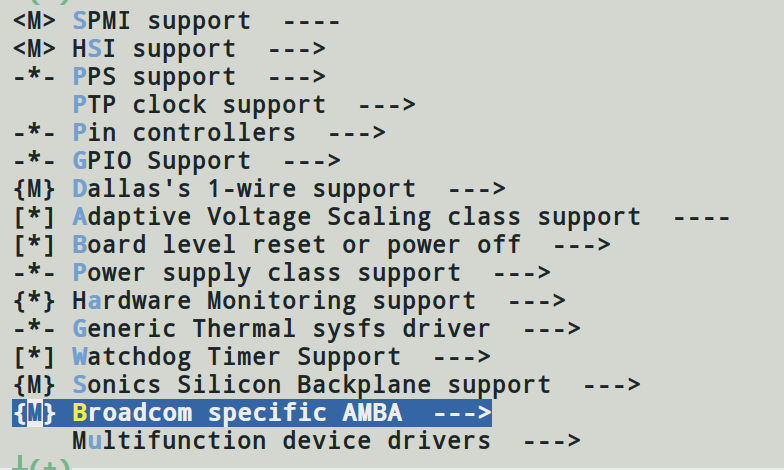
(15) Networking support #網路 盡量不要亂動

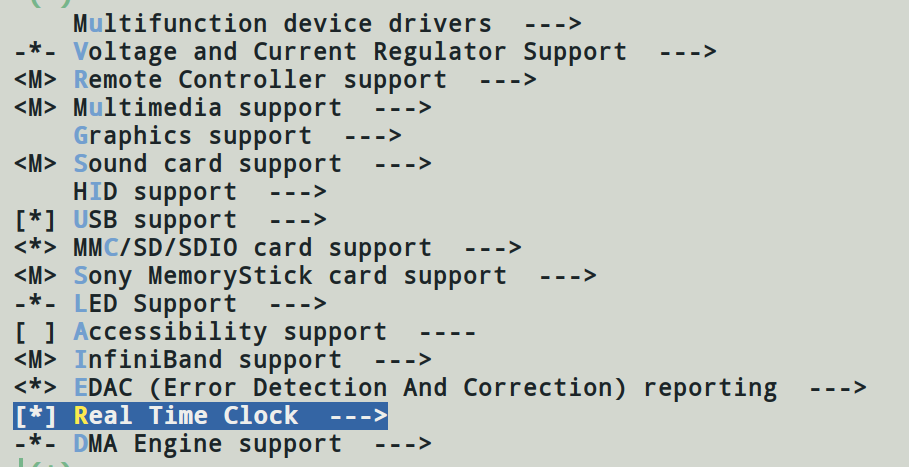
(16) Device Drivers #不要動 目前可以開機的東西已經複製上來 這裡對速度沒有太大影響 只有對編譯時間跟硬碟大小有影響(嵌入式常會改這裡) 可用lsmod 列出用到的module去參考要選的部分

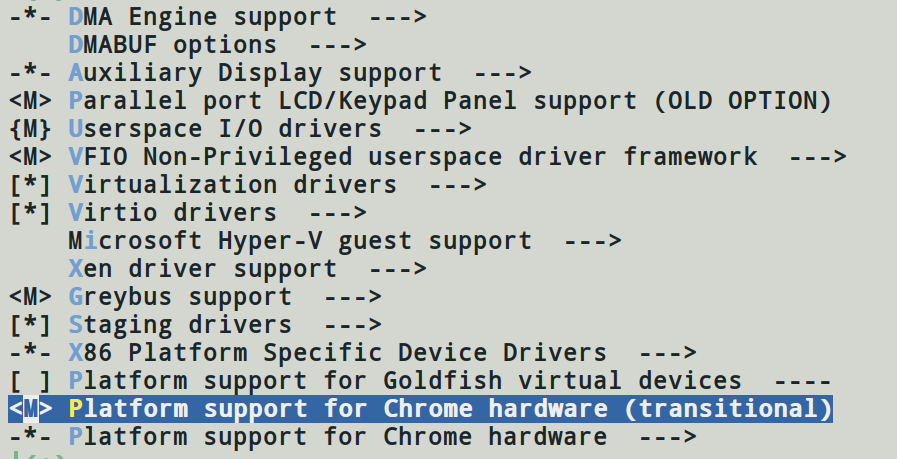


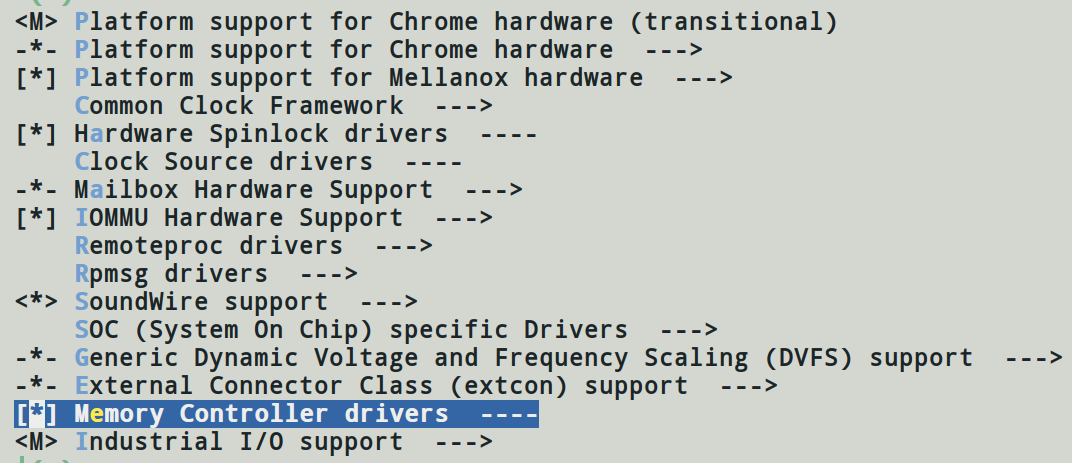


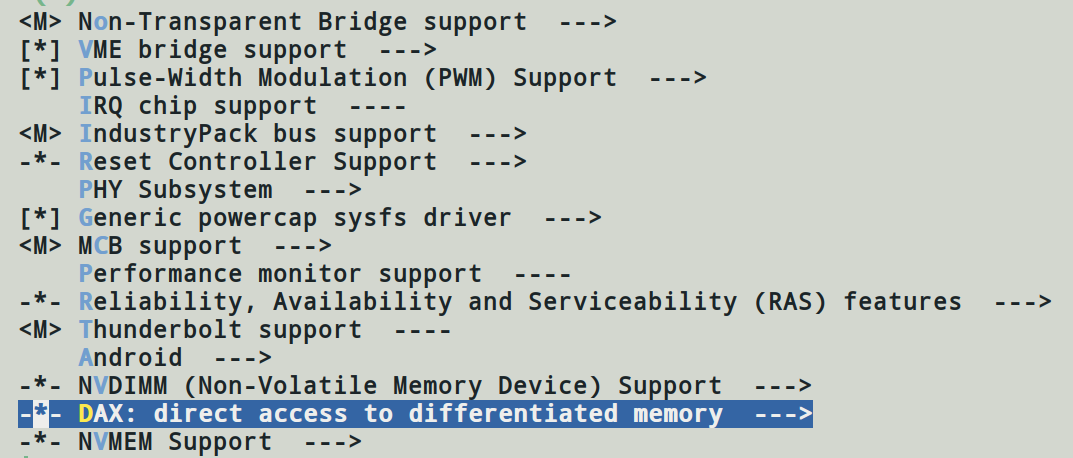


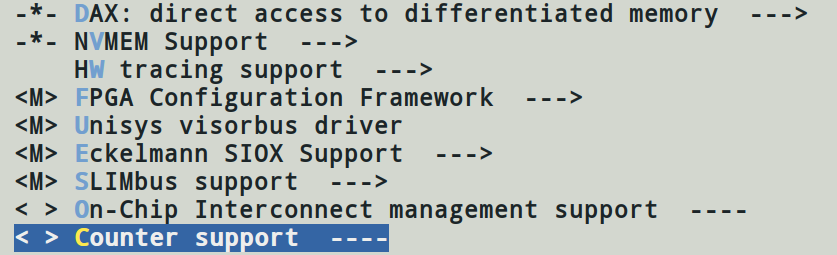


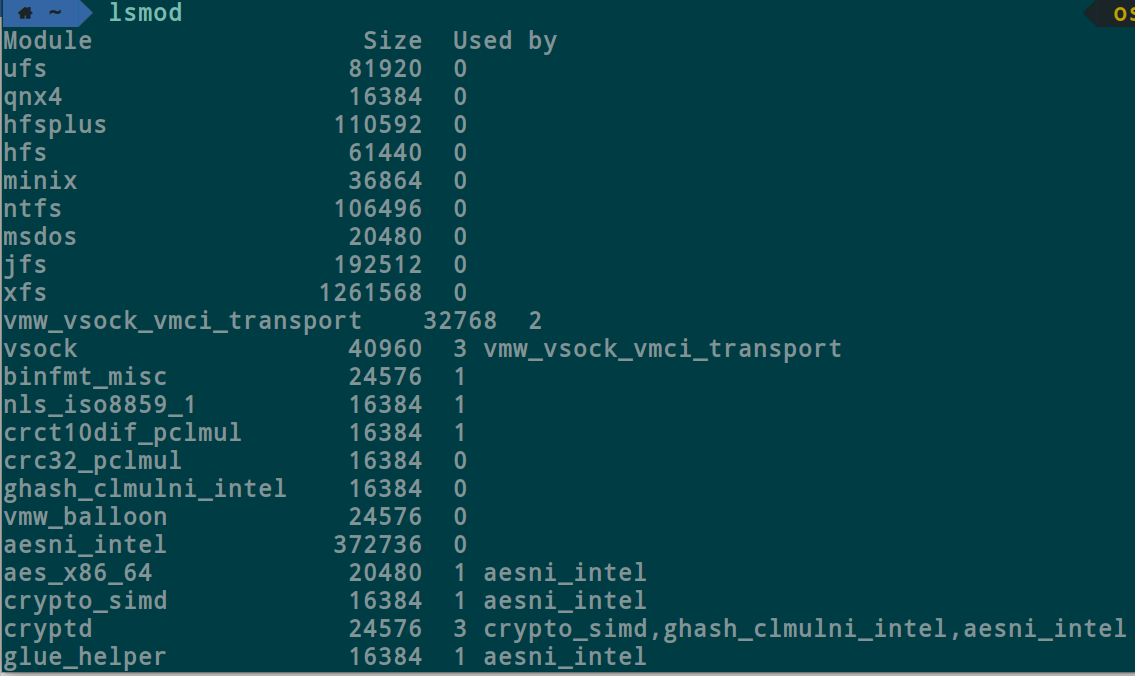










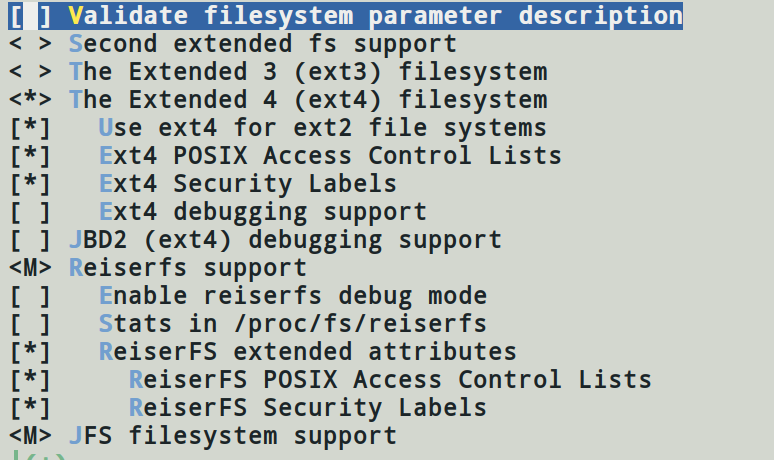


(17) File systems #有用到的選起來

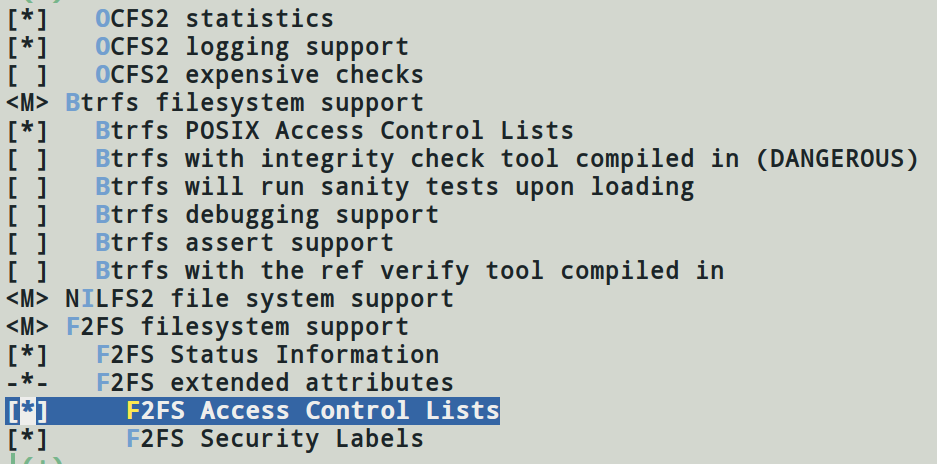
(i) ext4

(ii)btrfs #若主功能編譯成M 子功能也會變成M

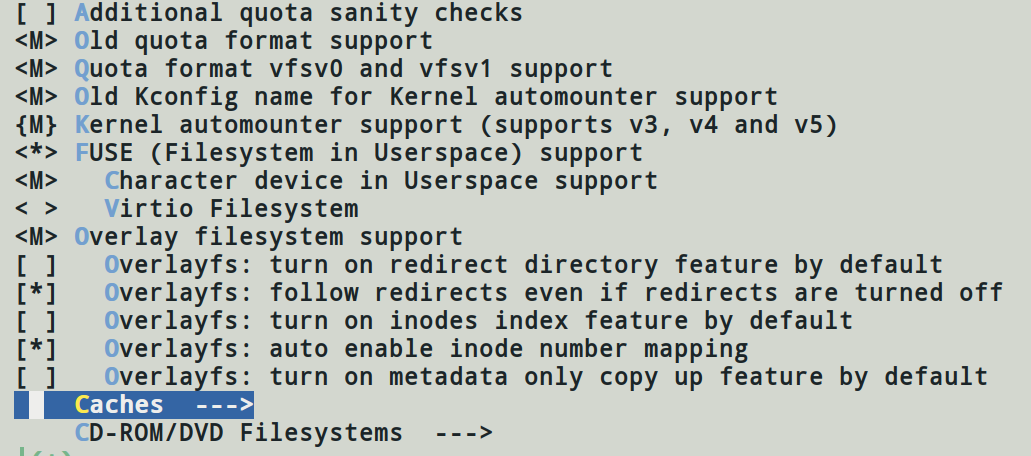


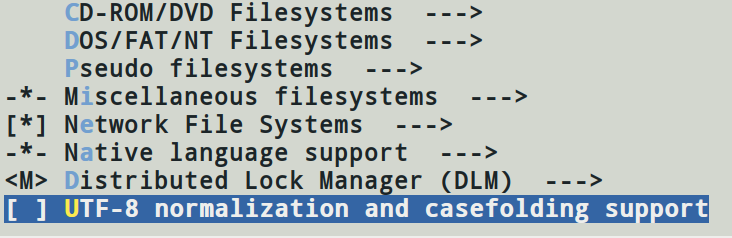












(18) Security options #不要動

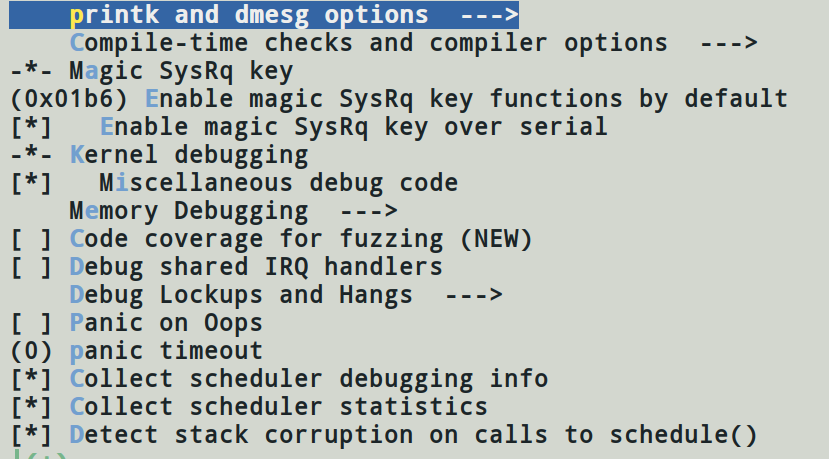
(19) Cryptographic API #不要動

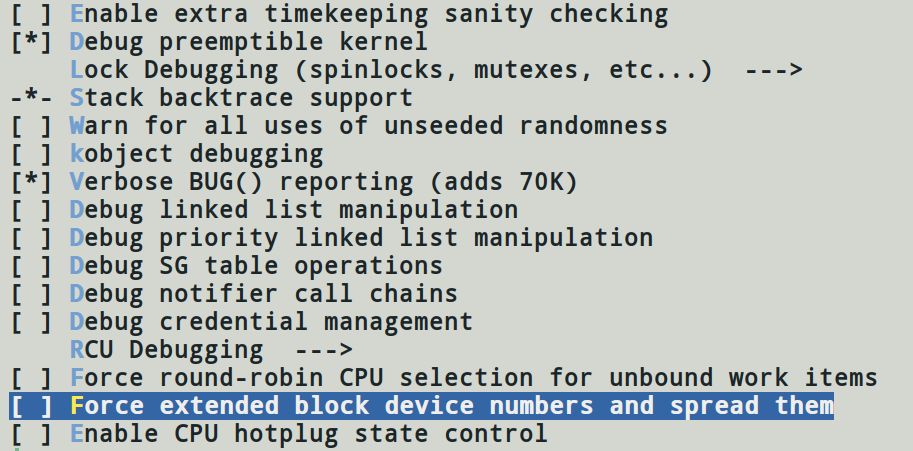
(20) Library routines #不要動

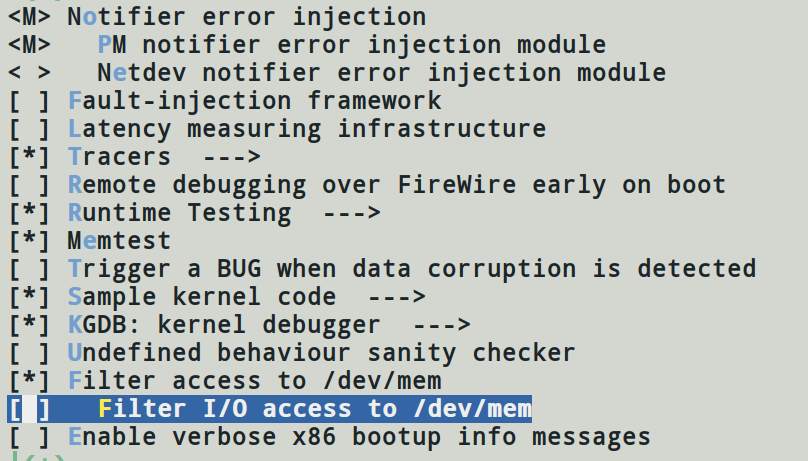
(21) Kernel hacking #不能亂選 速度會變慢

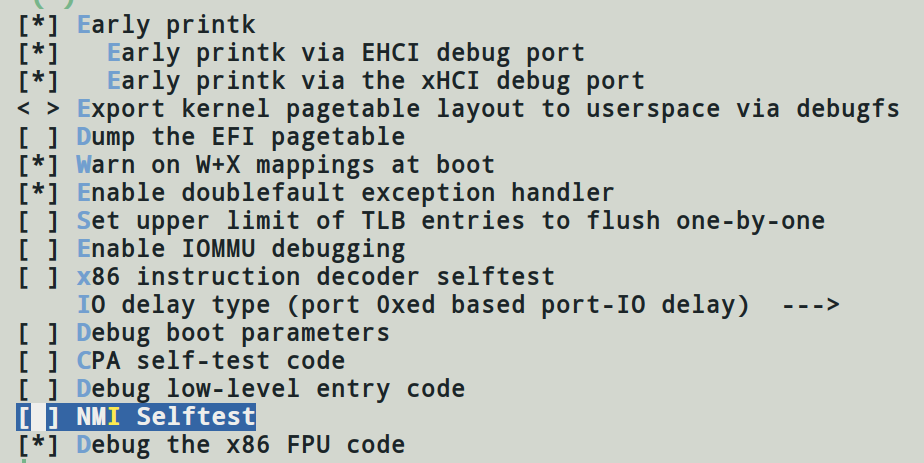
(i) Enable magic SysRq key over serial #sysreq 一組特別的key 可以觸動kernel 緊急shut down用 類似windos ctrl+alt+delete







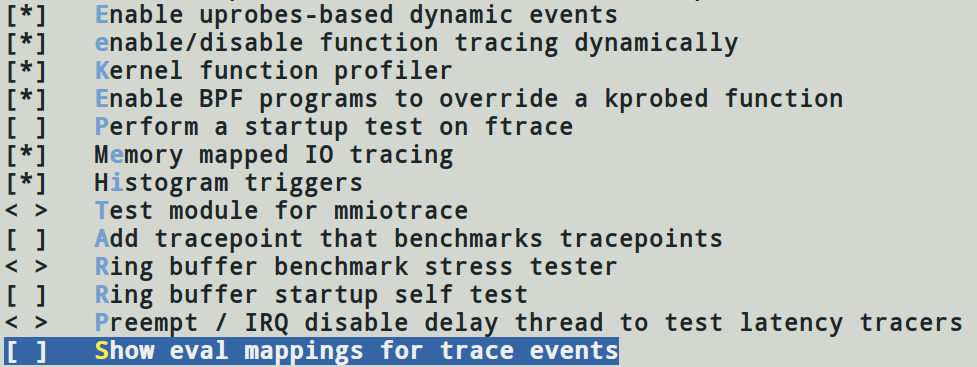
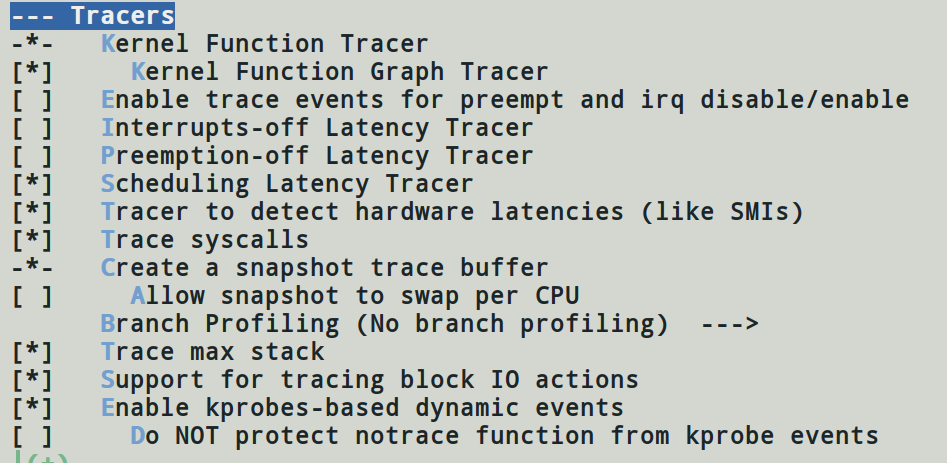






(ii)Tracers #有需要的再打開來看 預設外多半會影響效能 有ftrace功能

(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



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>