

# 2019，作業系統概論，期中考筆試部分

(2 pt) 1.你的學號

(2 pt) 2.你的姓名

(3 pt) 3.一個安全性的作業系統（如：Linux）最起碼需要二個執行模式（dual mode operation）。請問：（5 Points）

- 甲、Dual mode 中的 kernel mode 和 user mode 的差異（存取硬體？記憶體？）
- 乙、在記憶體方面需要什麼樣的配合？（hint：記憶體好像檔案一樣對不同人要有不同的？？）
- 丙、系統如何從 user mode 切換到 kernel mode（hint：特別的指令、單一進入點？）

答案:

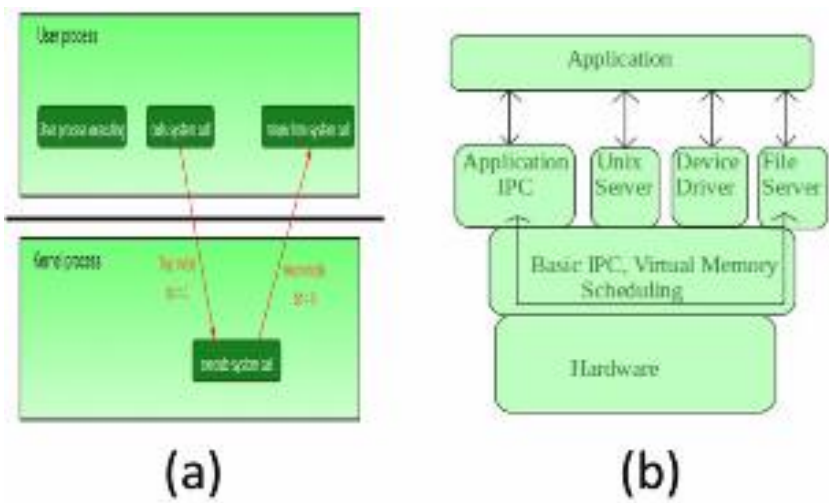
- 甲 kernel mode 可以存取硬碟及所有記憶體  
User mode 僅能存取 task 所分配到的記憶體
- 乙 kernel mode 有自己的 kernel space，僅 kernel mode 可存取  
User mode 有自己的 user space，kernel mode 和 user mode 都可存取
- 丙 使用 syscall，並將 rip 設為單一進入點

(3 pt) 4.Signal 對 system call 的影響為何？（hint：在 interruptable 和 uninteruptable 時，system call 收到 signal 會怎樣？restart？）(3 Points)

答案:

Interruptable 的會暫停當下的 syscall，處理完中斷後，再重新執行  
Uninteruptable 會執行完當下的 syscall，再執行中斷

(3 pt) 5.請從 context-switch 和 mode change 的角度，說明 monolithic kernel 的優點 (3 Points)



答案:

Monolithic kernel 會把眾多模組放 kernel mode，而微核心會將東西放在 user mode，模組間的溝通都需要用到 context-switch 和 mode change，因此在 monolithic kernel 下他只需要消耗 function call 的代價，可提升效能。

(3 pt) 76.電腦的主記憶體的三大主要用途？請問是否需要定期清除記憶體，說明你的觀點

答案:

Cache:存放硬體及儲存裝置的一些快取  
Buffer:先將 i/o 資料緩衝下來，等到一定量再一起傳  
Program:儲存程式執行所需運用的資料  
我認為應該不用，核心會自己分配該使用的記憶體，如果快取被清除，效能可能反倒會降低

(3 pt) 7.請設計一個系統，可以監控印表機的使用量（hint：印表機在 Linux 中為一個特殊檔案「/dev/usb/lp0」，假設可以使用「lpr」將文字檔案送到印表機，請問你要怎樣設定這些檔案的權限）（hint：/dev/usb/lp0 的權限要設定為何，lpr 又為何？）

答案:

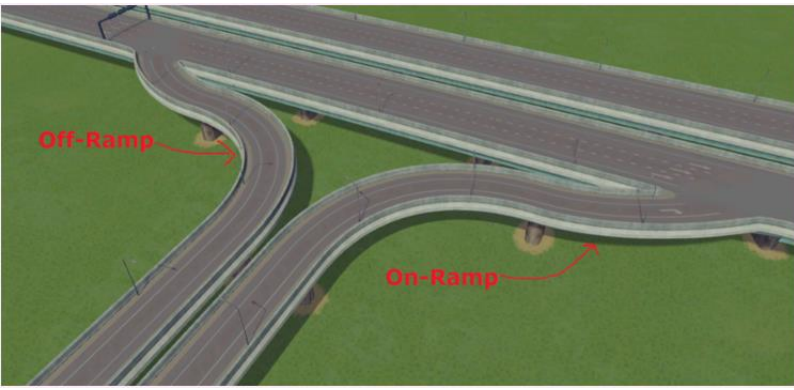
將/dev/usb/lp0 的權限設為僅有 root 可以存取，lpr 的權限僅有 root 可以執行，再寫一個 setuid 且 owner 是 root，但所有人都可以執行，因此就可以讓一般使用者可以執行，且達到監控的功能。

(3 pt) 8.請問何謂 port I/O 和 memory mapped I/O？請問 memory mapped I/O 的優勢。

答案:

Port i/o 會使用 port 空間進行定址，需搭配組語，像是 in, out。memory mapped i/o 為使用 memory 空間進行定址，搭配 mov 進行 i/o，相較起來容易使用，定址空間也較大。

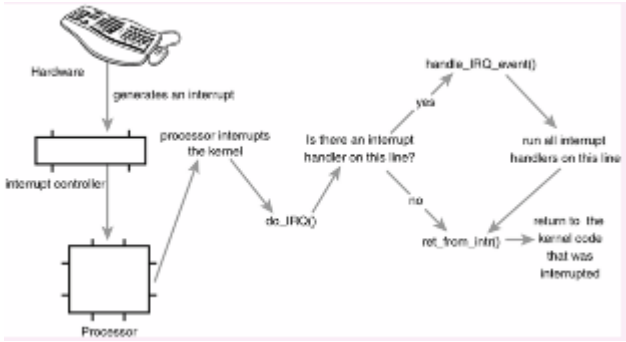
(3 pt) 9 目前的硬碟的控制器多半具有 DMA，請問 DMA 的功能為何？請問硬碟中為什麼要有 DRAM 充當 buffer？（注意：是「buffer」不是「cache」）



答案:

DMA 負責裝置搬移資料  
因為硬碟傳輸速度通常不快，如果直接傳，bus 頻寬大，硬碟的頻寬小，容易資源浪費。因此先傳到 buffer，足夠了再一起傳。

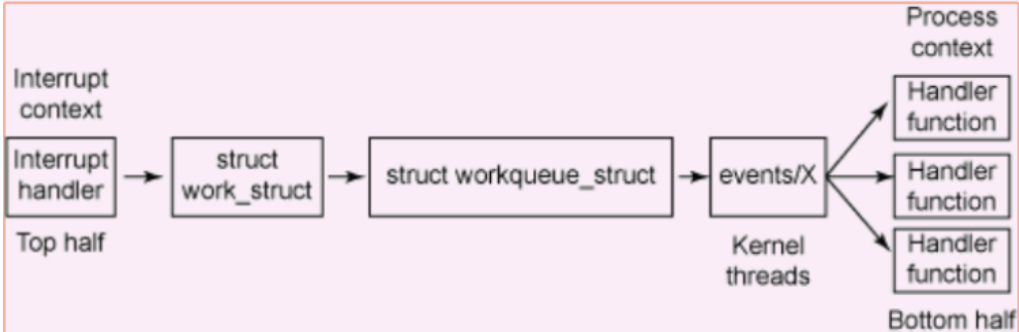
(3 pt) 10.請說明中斷的流程（hint：硬體、中斷訊息、CPU 怎樣暫停目前的執行、從哪邊找到這個中斷的「軟體處理流程」，中斷處理完以後要幹什麼事）



答案:

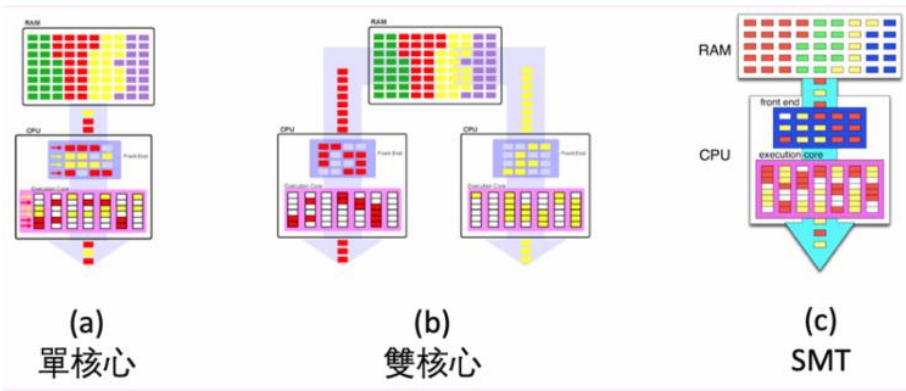
CPU 先呼叫中斷，會把當前的 task 的相關資料保存下來，透過中斷量表找到對應的 ISR，中斷結束後，再繼續執行 task。

(3 pt) 11.Linux 為什麼要把驅動程式分成 top half 和 bottom half ? (3 Points)



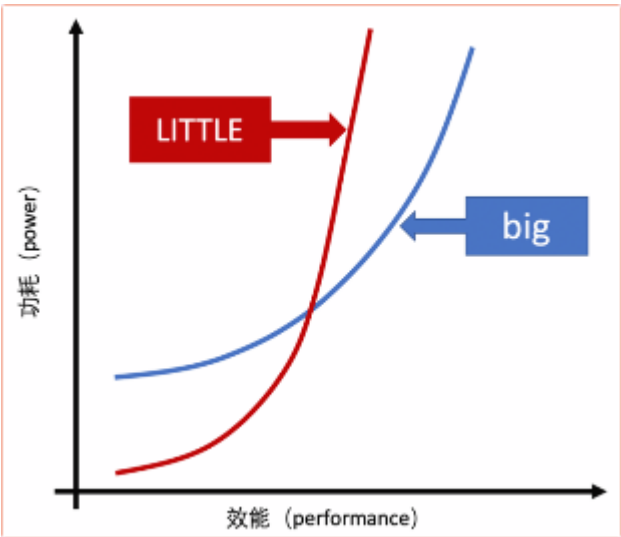
答案:  
當中斷時，盡快處理 top half 所該處理的事，可以提升 response time，不重要的等到 bottom half 再做。

(3 pt) 12.請解釋 SMT（同時多執行緒，Simultaneous multithreading）大致的運作原理。（hint：同時執行多個「行程」？一起執行的好處？）



答案:  
SMT 讓原本的處理器可以處理多個行程，如果照以往只能處理一個行程，會造成很多資源沒有被使用到。反之，則可以有效率的被運用。

(3 pt) 13.請解釋 bigLITTLE 的原理（hint：小核心和大核心，主要是速度增加還是省電，如何達成）



答案:  
小核心:處理比較不需要效能的工作，透過低時脈，達成省電。  
大核心:執行須需要效能的東西，使用高時脈的大核心。

(3 pt) 14.請問 32 條記憶體組成超寬通道（memory channel），和 32 條記憶體組成 8 條小通道的差異

答案:  
一般情況下不會使用到 32 條記憶體，如果要同時維護，會犧牲很多資源。因此分為好幾組通道，特定 CPU 使用不同通道，以此提升效能。

(3 pt) 15.為什麼 CPU 剛啟動時 cache 沒有打開（hint：I/O）

答案:  
因為需要將 i/o 通道先設定為 non cache

(3 pt) 16.請大略用組合語言，透過 system call 在螢幕上印出“hello”。（就是要跟附圖的方式意思一樣的程式碼）

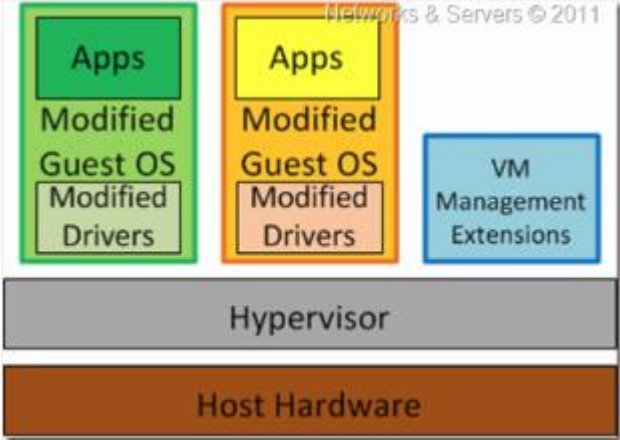
```
#include <stdio.h>
void main(){
    printf("Hello World");
}
```

```
#include <stdio.h>
#include <string.h>
Int main(){
    Char *text = "hello";
    Int len = strlen(text) + 1;
    __asm__ volatile(
        "mov $1, %%rax\n" // write 是 1
        "mov $1, %%rdi\n" // stdout 是 1
        "mov %1, %%rsi\n" // text
        "mov %2, %%rdx\n" // len
        "syscall\n"
        "mov %%rax, %0\n"
        : "=m"(ret)
        : "g"(text), "g"(len)
        : "rax", "rbx", "rcx", "rdx");
}
```

(3 pt) 17.什麼是 vdso（hint：核心將...跟使用者行程...，就是什麼樣的情況下會用 vdso，有什麼好處）

答案:  
Vdso 將 clock\_gettime 一些非機密的 function 從 kernel space 搬到 user space，使用者使用時，不用進行 mode change，以此提升效能。

(4 pt) 18.舉出一個例子說明 guest OS（就是被模擬機執行的那個 OS）最好要能與 vmm（virtual machine monitor，就是能執行 guest OS 的那個模擬機）溝通（例如：提供什麼樣的特殊指令，提供什麼樣的特殊驅動程式，為什麼要提供你所說的「額外機制」）



答案:  
特權指令，trap。因為 guest 如果使用權指令，可能不是得到想要的結果。



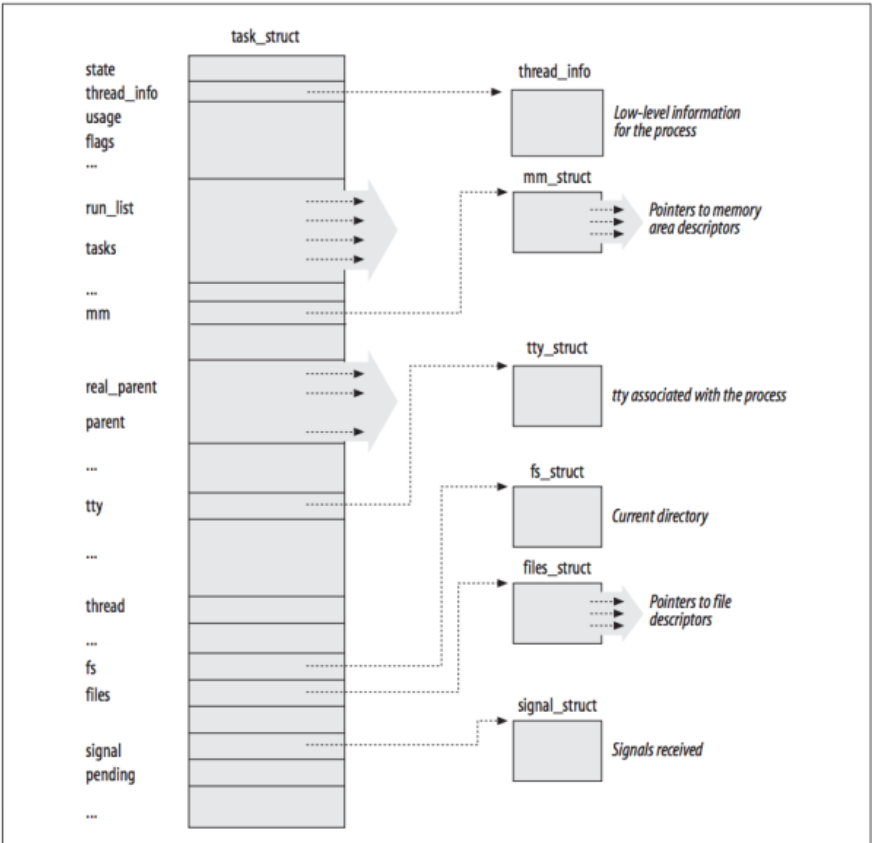
(3 pt) 19.請問如果取消 ASLR（位址空間組態隨機載入，Address space layout randomization）的話，可以進行怎樣的優化

答案:  
如果一開始就知道位置，就不用進行動態載入了。

(3 pt) 20.請問如果讓 CPU-bound 的優先權比較高會造成什麼樣的現象（hint：注意一下，是「CPU」，稍微敘述一下為什麼會有這樣的現象。可以使用 C 代表電腦正在處理 CPU-bound，用 I 代表電腦正在處理 I/O bound）

答案:  
如果 CPU bound 較高，C 比 I 先執行，I 要等 C 執行完，才能發出 i/o，然後等待 i/o 無法有效利用資源。如果能讓 I/O bound 先發出 i/o，再換 C，等待時就可以讓 C 先執行，提升效率。

(3 pt) 21.在 Linux kernel 中，隸屬於同一個 process 的 thread 至少共用哪個物件？

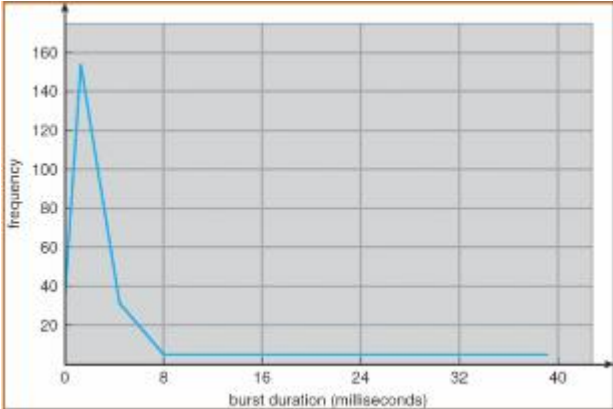


答案:  
mm\_struct

(3 pt) 22.舉一個例子說明，non-preemptive OS 在什麼樣的情境下效能會比 preemptive OS 好。大概說明原因。

答案:  
Netware。讓每個 task 執行完且發出 i/o，馬上接著下個 task，便能有效率的利用資源。

(3 pt) 23.如果要讓 I/O 的效能提高，又要能夠有好一點的 response time，請問應該把 time slice 設定為多少？（hint：一個數字，說明為何要設定成那個數字）



答案:  
8ms。因為這時間之前大部分都能發出 i/o。

(3 pt) 24.請問 Linux 2.4 是怎樣提高效能（hint：先定義 epoch 的轉換時機，然後累積了什麼，累積的東西跟優先權有何關係）

答案:  
Epoch 會在大部分 CPU bound 為 0 時轉換。I/O bound 剩餘時間會除以 2 後，再加上 time slice。因此執行時間永遠較高，I/O bound 永遠有優先權。

(3 pt) 25.請問在 CFS 中有辦法提高 task 的優先權，使得他的優先權「絕對」高過 I/O 嗎？請說明原因

答案:  
沒辦法，因為 i/o task 的時間，加上 queue 的時間，會是 vruntime - X。因此，I/O 會有最小值。

(3 pt) 26.什麼是 race condition（請寫清楚，不要英翻中。可以用舉例的方式說明）

答案:  
因為執行順序與預期不同，指令上會有時間差，導致出現非預期現象，像是兩邊 CPU 改到同個變數值。

(3 pt) 27.請解釋 critical section 一定要滿足 progress 的原因（hint：A 想進去，裡面沒人但...）

答案:  
Progress 的條件是：如果有人想進去 critical section，就讓他進去。如果沒有這個條件，就有可能會發生都不讓大家進去的情況。

(3 pt) 28.A 可以 preempt B，如果要製造 critical section，可以怎樣做最簡單（hint：不可以使用 spinlock、semaphore 等，disable...）

答案:  
B disable A

(3 pt) 29.請問如果設定 mutex「不支援」巢狀鎖定的好處（換句話說，在什麼樣的情況下，你會用「不支援巢狀」）

Mutex Type	Robustness	Relock	Unlock When Not Owner
NORMAL	non-robust	deadlock	undefined behavior
NORMAL	robust	deadlock	error returned
ERRORCHECK	either	error returned	error returned
RECURSIVE	either	recursive (see below)	error returned
DEFAULT	non-robust	undefined behavior†	undefined behavior†
DEFAULT	robust	undefined behavior†	error returned

答案:  
如果 relock 產生在 deadlock 下。我們如果能選擇不支援巢狀，比較能發現問題所在。

(3 pt) 30.DMA 和 CPU 都會更新 DRAM，請列出一種方法，可以確保 CPU 會看到最新的資料 (hint：有硬體法跟軟體法，硬體法要想辦法把資料寫到 X X X，軟體法要想辦法取消 X X X)

答案:

## 硬體法:想辦法將資料寫到 CPU

軟體法:取消 cache line

(3 pt) 31. 什麼是 readers-writers 問題？ (hint：一定要寫出在平行化方面的優勢)

答案:

一個東西可以很多人讀，但 reader writer 不能同時存取，也不能有兩個 writer。

平行話的話，一次有多 reader 讀資料，可以增加效能，因為不用管 race condition。

(3 pt) 32.新「櫃」結衣的注音 (hint: 可以寫在紙上)

㊦ 注音符號表：ㄅ ㄆ ㄇ ㄋ ㄌ ㄍ ㄎ ㄑ ㄒ ㄓ ㄔ ㄕ ㄖ ㄗ ㄘ ㄙ ㄥ ㄉ ㄊ ㄐ ㄑ ㄒ ㄓ ㄔ ㄕ ㄖ ㄗ ㄘ ㄙ ㄥ

ㄇ 音韻表：一聲、二聲、三聲、四聲、輕聲

㊦如：羅，ㄌㄨㄛˊ ㄖㄨㄛˊ 二聲

(3 pt) 33.How was your night? Did you sleep well? (3 Points)

☒ My knight is very brave.

☒ I heard my alarm clock bleeping this morning.

想著如何脫離宅宅

☒整晚想著如何表白

☒覺得老肥宅壓床（如授課老師）是很恐怖的事情

(3 pt) 34.今天天氣如何？

☒ fine, thank you

☒ you are welcome

❑ 以何問天？天之道，損有餘而補不足。人之道，則不然，損不足以奉有餘。孰能有餘以奉天下，唯有道者。

☑這問題並不完整，我不曉得指的是嘉義還是紐約

☒ 本宅一直在讀 OS 還沒看到陽光

☒ 反正明天的天氣不會比今天更糟

老師你會不會覺得「不問蒼生，問蒼天」是很幼稚的？

難道我不能不完全不回答這個問題嗎？

- ❑ 難道我不可以不回答這個問題？
- ❑ 生氣如何是不是應該問 Siri 呢？

☑天氣如何是不是應該問 Siri 呢？

☑今天的天氣如同我的心情，你知道的。  
老試中：無計胸中事，看我無計同筭。