直接作答.若空間不夠,請在背面作答.(注意:背面作答要標明題號)

1.(10%)Please write down the full names of the following abbreviations. Please do **not** make any explanations.

(1) FCFS: First-come-First-service

(2) FIFO: First-in-First-out

(3) RR (in scheduling): Round Robin

(4) NUMA (for memory access): Non-Uniform memory access

(5) CLI (for interface): Command line

(6) GUI (for interface): Graphics users interface

(7) API (for interface): Application program interface

(8) RPC: Remote Procedure call

(9) PCB: Process control block

(10) IPC: Interprocess communication

2. (30%)Please explain the following terminology briefly.

(1)Kernel:

OS的主架構,一種不斷執行的 program,

(2)firmware

在於 ROM, EPROM 的 軟體豐 (bootstrap program)

(3)system call

用於 user mode 的 命令 (command)

(4)trap

軟體中斷 (ex: 除以零)

(5)matchmaker (in RPC)

查表

(6) heap:

在執行期間 用來做動態記憶體的分配

(7)socket:

溝通 (communication) 的 終端點 (end point)

(8)Pthread:

POSIX thread, 用於 user mode, 在 kernel mode 常作於 UNIX OS 中

(9)Load balancing:

(10)mode bit:

3.(4%)Consider (i)FCFS, (ii)shortest job first, (iii)RR, (iv) priority, (v)multilevel feedback queue.

(1)(iv)Which of the above scheduling algorithm could result in starvation?

(2)(iii)Which of the above scheduling algorithm is(are) better of interactive processes?

4. (4%)(1) What is a privileged instruction?

擁有較高執行順位的程式

(2)(abc)(multiple choices) Which of the following instructions should be privileged? (a)set value of timer; (b)clear memory; (c)turn off interrupts, (d)issue a trap instruction

5. (4%)(1)Explain what is a (process) context switch

切換PCB，當process被切換時，其狀態(state)會被儲存
然後切換到新的process後，已儲存的狀態會被喚起

(2) What state information do you need to save/restore about threads when performing a context switch? (Note: only the thread, Not the process.)

waiting

6. (4%)Including the initial parent process, how may processes are created by the program show below:

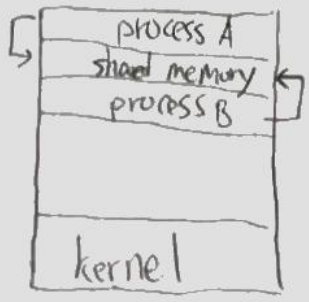
```
#include <stdio.h>
#include <unistd.h>
 int main()
{ /*for a child process */
   fork;
   /*fork another child process*/
   fork;
   /*and fork another*/
   fork;
   return 0;
}
```

2×2×2 = 8個 processes

7. (8%)Describe what is a shared memory model and what is a message passing model.

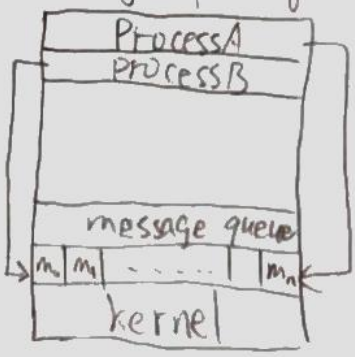
+8

shared memory



藉由將雙方的資訊傳入 shared memory 中來達成 IPC

message passing



ProcessA 將資訊傳入 message queue 然後 ProcessB 進入 message queue 接收 藉此達成 IPC

8. (4%)(1)Why so some operating systems store the operating system in firmware, while others store it on disk?

+4

好比說嵌入式系統中可能沒有 disk，所以藉由 firmware 來存放 OS

(2)How could a system be designed to allow a choice of operating systems from which to boot? What would the bootstrap program need to do?

可以藉由 啟動管理 (boot management) 來選擇要進入的 OS，通常啟動管理會被儲存在 disk 中，以便 bootstrap program 去找尋及使用

9. (8%)Define what is a policy and what is a mechanism and show an example for each of them.

+8

policy：做什麼

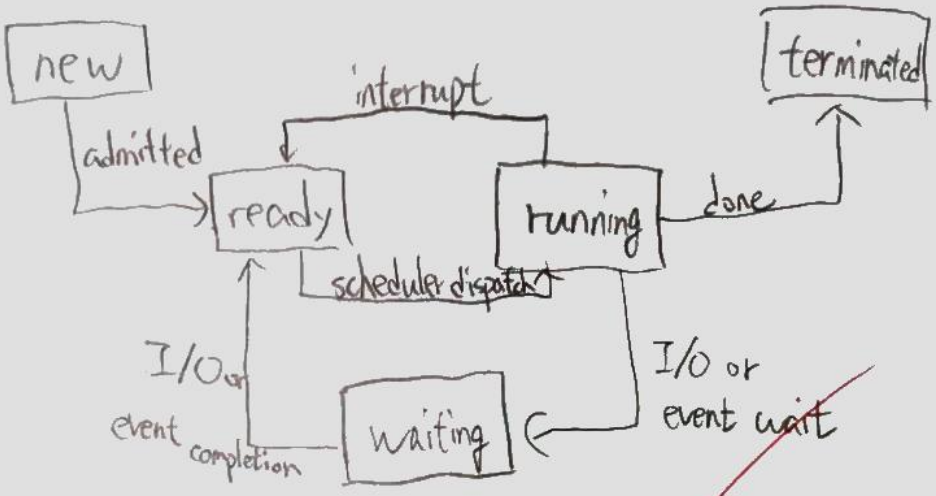
mechanism：怎麼做

以排程為例 policy：排程的方式（RR, SJF, FCFS）

mechanism：FCFS的演算法

10. (12%)(1)Draw a process state (transition) diagram. The states are new, ready(runnable), running, terminated, and waiting(blocked). Note that you should identify the events that trigger each state transition.

+11



(2) For each of the following process state transitions, say whether the transition is legal ***and*** how the transition occurs or why it cannot.

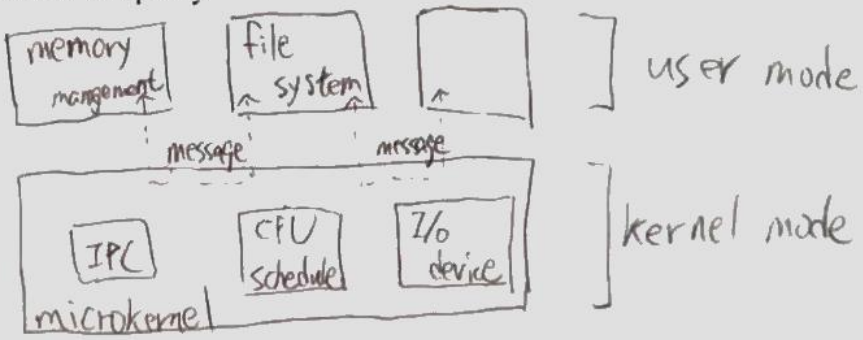
(a) Change from the state **BLOCKED** to thread state **RUNNING**.

illegal 要回到 ready，並由排程決定執行的 process 才能執行 (running)

(b) Change from the state **RUNNING** to state **BLOCKED**.

legal 由 Blocked 回到 ready 準備執行 (running) 下個 process

11. (8%) Draw a block diagram showing the structure of a microkernel. Note that you must show that how the kernel invokes a user level process which would serve as a system service or as a policy.



+6

12. (8%)(1)List two reasons why ***overuse*** of threads is bad (i.e., using too many threads for different tasks). Be explicit in your answers.

速度慢

+3

(2) List two reasons why threads are useful/important.

效能加
可變性加

13. (8%)You're hired by COOL Computers Inc. to improve the performance of their system.They point out that their applications only use 10 of the CPU's 32 registers; so to improve the performance of the applications, they suggest that you change the OS's context switch routine so it only saves the 10 registers used by the applications. Assume that you can correctly change the context switch routine. Is this a good or bad idea? Why?

+0

14. (20%)Assume the following processes are to be scheduled using a ***preemptive, round-robin*** scheduling algorithm. Each process is assigned a priority (value). A higher value indicates a higher priority. In addition, assume that the system has an idle process. The idle process, identified as P_{idle}, has priority value 0 and it is scheduled when there is no available process to run. The length of a time quantum is 8 units. If a process is preempted by a higher priority process, the preempted process is placed at the end of the queue.

thread	priority	burst	arrival
P1	40	20	0
P2	30	25	25
P3	30	25	30
P4	35	15	60
P5	5	10	100
P6	10	10	105

+8

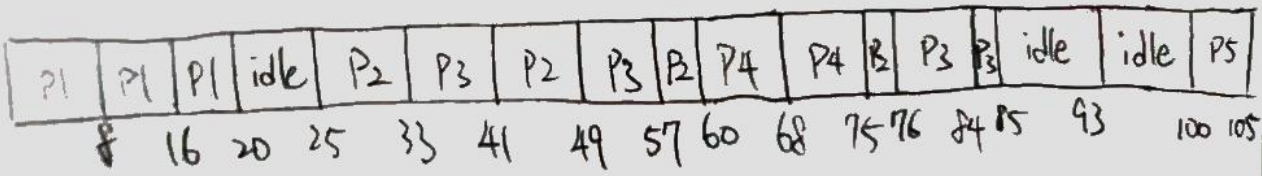
(1)Show the scheduling order of the processes using a Gantt chart.

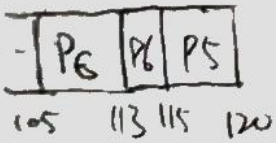
(2)What is turnaround time for each process?

(3)What is the waiting time for each process?

(4)What is the CPU utilization rate?

(Hint: you can make your own assumptions if you think the problem here is not clearly described.)





	turnaround time	waiting time
P1	20	0
P2	31 ✗	24 ✗
P3	35	35
P4	K ✗	0 ✗
P5	10 ✗	10
P6	10 ✓	0 ✗