

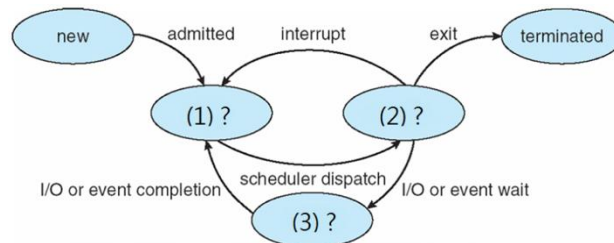
長庚大學110學年度第一學期作業系統期中測驗（滿分107）

系級:

姓名:

學號:

1. (6%) 下面這張圖簡單地陳述了Process的一生，請填上圖中(1)、(2)、(3)的內容



Answer: (1) Ready, (2) Running, (3) Waiting.

2. (8%) 在作業系統中，Interrupts是用來通知處理器來處理特殊事項的一個硬體機制與軟體技術，通常收到Interrupts後，作業系統會透過一連串的操作達成需要的結果。Interrupts又可以分为Software Interrupts與Hardware Interrupts，請分別舉出一項Software Interrupts與一項Hardware Interrupts？

Answer: Software Interrupts: signals, invalid memory access, division by zero, system calls, etc.

(a correct answer +4%, a wrong answer -4%)

Hardware Interrupts: services requests of I/O devices, e.g., keyboards, Ethernet adapters, touch panels, etc. (a correct answer +4%, a wrong answer -4%)

3. (8%) 在作業系統中請說明Multiprogramming及Time Sharing的定義。

Answer: Multiprogramming: The operating system keeps several jobs in memory simultaneously (4%).

Time Sharing: Time sharing is a logical extension of multiprogramming, in which CPU switches jobs frequently so that users can interact with each job while it is running (4%).

4. (8%) 在記憶體與儲存裝置架構中，課本說明到DRAM是Volatile的，而像是傳統硬碟這種儲存裝置是Nonvolatile的，請說明在記憶體與儲存裝置架構中何謂Volatile？

Answer: Volatile memory is computer memory that requires power to maintain the stored information, i.e., the stored information will be lost if the power supply of volatile memory is cut off.

5. (10%) (a)請定義何謂System Call。(b)Application Programming Interface (API) 與System Call有何不同？(何者可以跨平台？何者會多次呼叫另一者來達成較複雜的功能？)

Answer: System calls provide the routines for user applications to use the functions provided by operating systems (4%). The API of a programming language serves as a user-friendly link to system calls made available by the operating system (4%). Thus, most of the details of the operating-system interface are hidden from programmers by the API and are managed by run-time support libraries.

6. (8%) 作業系統中在做Inter Process Communication (IPC)時有兩種方法：Message Passing and Shared Memory。兩相比較下，Message Passing對應用程式的開發者而言較容易使用；而需要大量資

料傳輸與頻繁溝通時，妥善地使用Shared Memory將可以得到較好的效能。請說明為何Shared Memory效能較好。

Answer: If we use message passing, a series of system calls has to be invoked for sending each message. By using shared memory, multiple processes can directly access a shared memory area multiple times without invoking many system calls.

7. (8%) 一次的IPC中，會有資料傳送者(Sender)與資料接收者(Receiver)，請問對於資料傳送者而言，使用 Synchronous Message Passing IPC 與使用 Asynchronous Message Passing IPC 有何不同？

Answer: Synchronous Message Passing IPC: The sender has to wait until the message is received.
Asynchronous Message Passing IPC: The sender sends the message and continues.

8. (9%) 請說明 (a) User Thread與Kernel Thread有何不同？Kernel Thread與User Thread有許多種應對關係中請說明何謂 (b) Many-to-One Model 、 (c) One-to-One Model 。

Answer: (a) User threads are managed by user-level threads library, and kernel threads are supported by the kernel and are the unit for CPU scheduling.
(b) Many user threads of a process are mapped to one kernel thread.
(c) Each user thread is mapped to a kernel thread.

9. (8%) 請說明Thread Local Storage (TLS)的用途，並說明TLS與local variable有何不同。

Answer: Purpose: TLS allows each thread to have its own copy of data. (4%)
Difference: Local variables are visible only during single function invocation, but TLS visible across function invocations in a thread. (4%)

10. (10%) 在解釋 Race Condition 的時候，我們用 Consumer-Producer 程式中 counter++與 counter--來實際舉例。已知 counter++與 counter--可以拆解成以下共六行的執行行為，若 counter 初始值為 5，且 counter++與 counter--各會執行一次，請問：(a)在怎樣的執行順序下 counter 的最終值會是 4？(b)在怎樣的執行順序下 counter 的最終值會是 5？

counter++:	counter--:
r1 = counter	r2 = counter
r1 = r1 + 1	r2 = r2 - 1
counter = r1	counter = r2

Answer: (a) P: r1 = counter
P: r1 = r1 + 1
C: r2 = counter
C: r2 = r2 - 1
P: counter = r1
C: counter = r2 = 4
(b) P: r1 = counter
P: r1 = r1 + 1
P: counter = r1
C: r2 = counter
C: r2 = r2 - 1
C: counter = r2 = 5

11. (12%) 考慮已經就緒的五個工作，依序為P1, P2, P3, P4, P5。使用三個排程演算法FCFS (First-Come, First-Served)、SJF (Shortest-Job-First)以及RR (Round Robin)來排程，而RR所使用的time quantum為 4ms。 (a) (6%) 請畫下三個排程演算法的排程圖， (b) (3%) 請分別算出三個排程演算法中每個工作的Waiting Time， (c) (3%)請分別算出三個排程演算法中每個工作的Response Time。

Process	Burst Time
P1	6 ms
P2	4 ms
P3	3 ms
P4	5 ms
P5	2 ms

Answer:

(a)

FCFS:

P1		P2		P3		P4		P5	
0	6	10	13	18	20				

SJF:

P5	P3	P2	P4	P1	
0	2	5	9	14	20

RR:

P ₁		P ₂		P ₃		P ₄		P ₅		P ₁		P ₄	
0	4	8	11	15	17	19	20						

(b)

FCFS: P1: 6-6=0, P2: 10-4=6, P3: 13-3=10, P4: 18-5=13, P5: 20-2=18

SJF: P1: 20-6=14, P2: 9-4=5, P3: 5-3=2, P4: 14-5=9, P5: 2-2=0

RR: P1: 19-6=13, P2: 8-4=4, P3: 11-3=8, P4: 20-5=15, P5: 17-2=15

(c)

FCFS: P1: 0, P2: 6, P3: 10, P4: 13, P5: 18

SJF: P1: 14, P2: 5, P3: 2, P4: 9, P5: 0

RR: P1: 0, P2: 4, P3: 8, P4: 11, P5: 15

12. (12%)請寫出以下程式在POSIX環境下執行後的輸出結果。

```
#include<sys/types.h>
#include<stdio.h>
#include<unistd.h>
int main()
{
    pid_t pid, pid2;
    pid = fork();
    if (pid == 0)
    {
        printf("Hello\n");
        pid2 = fork();
        if (pid2 != 0)
        {
            wait(NULL);
            printf("The OS midterm\n");
        }
        else
        {

```

```
        printf("I would say that\n");
    }
}
else
{
    wait(NULL);
    printf("Is quite easy\n");
}
return 0;
}
```

Answer:

Hello

I would say that

The OS midterm

Is quite easy