

長庚大學105學年度第一學期作業系統期中測驗（滿分111）

系級:

姓名:

學號:

1. (8%) Real-Time Embedded Systems有許多常見的應用，譬如行車安全系統、手持式多媒體系統、居家或工廠自動控制系統。請說明Real-Time所指的意思為何？

Answer: Real-time means on-time instead of fast. Real-time systems have to get the correct (or qualified) computing results and have to make sure that the results can be derived on time.

2. (8%) 請描述Application Programming Interface (API)、System-Call、Operating System三者之間的關係。

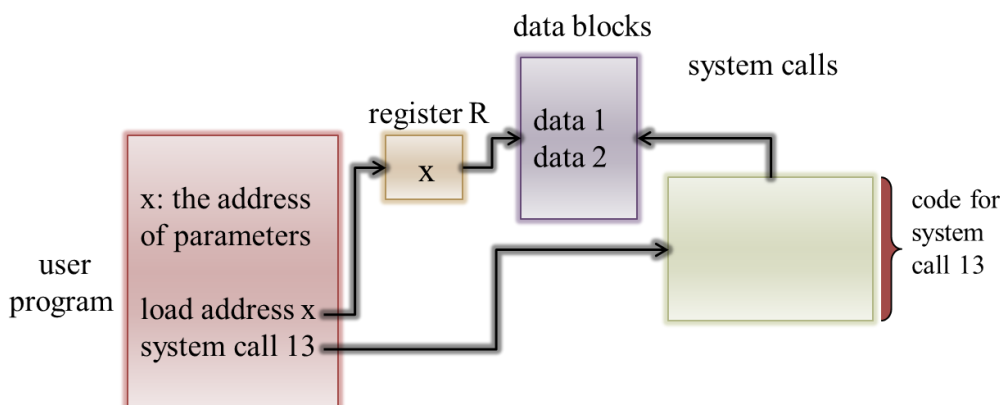
Answer: System calls provide the routines for user applications to use the functions provided by operating systems (4%). The API of a programming language serves as a user-friendly link to system calls made available by the operating system (4%). Thus, most of the details of the operating-system interface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

3. (9%) 在作業系統中請說明 (a) Microkernel System Architecture的設計理念為何？使用Microkernel的概念來設計作業系統 (b)有何優點？ (c)有何缺點？

Answer: Concept: moving all nonessential components from the kernel to user or system programs
Advantage: portability, reliability, security
Disadvantage: the overheads of extra system calls
(3% for each answer)

4. (8%) 應用程式在呼叫system call的時候會需要傳參數給作業系統，傳參數的方法有三種。第一種是用registers：應用程式把參數存在registers裡後再呼叫system call，作業系統便可直接從registers裡讀出參數。第二種做法是用stack：應用程式把參數push進stack裡後再呼叫system call，作業系統便可從stacks裡pop出所需的參數。第三種做法是用register pointing to block，請解釋如何使用register pointing to block來傳遞參數。

Answer: The pointer to the data blocks is stored in register R before the system call is invoked. The system call then gets the pointer from register R to access the data blocks.



5. (8%) 請定義 (a) I/O-bound process 與 (b) CPU-bound process 。

Answer: I/O-bound process – spends more time doing I/O than computations; many short CPU bursts. (4%)

CPU-bound process – spends more time doing computations; few very long CPU bursts. (4%)

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，請說明 (a) 使用Message Passing的優點為何？ (b) 使用Shared Memory的優點為何？

Answer: Shared Memory: better performance (4%)

Message Passing: easy to use (4%)

7. (8%) 在作業系統中請簡述Remote Procedure Call (RPC)主要功能為何？(hint: 可以說明RPC的使用情境，或是簡述使用RPC的流程。)

Answer: RPC is a way to abstract the procedure-call mechanism for use between systems with network connection. To use a RPC, a system has to access the matchmaker in the RPC server to get the port for the RPC service. Input data are then sent to the port of the RPC server, and the result is returned to the system.

8. (8%) 當我們在伺服器上設計網服務程式(如：網頁伺服器、FTP伺服器)，一般來說我們會用multiple threads而不是multiple processes來服務多位使用者。請問，相較之下使用multiple threads的優點為何？

Answer: Threads can share resources of a process, e.g., global data, binary code and opened files. Thus, it is much more efficient in terms of resource saving. Commutation among the threads of a process is easier than that among processes.

9. (10%) 使用Round Robin (RR) Scheduling 技術時，設定time quantum的大小是一件很重要的事，請說明 (a)當time quantum太大時會有什麼缺點？ (b)當time quantum太小時會有什麼缺點？

Answer: Too long: The RR scheduling will become the FCFS scheduling. Thus, the average waiting time could be long. (5%)

Too short: The overhead of context switches will be high. (5%)

10. (10%) 假設每次呼叫fork()都是成功的，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)
    {
        wait(NULL);
```

```

printf("AAA\n");
pid2 = fork();
if (pid2 != 0)
{
    wait(NULL);
    printf("BBB\n");
}
else
{
    printf("CCC\n");
}
}
else
{
    printf("DDD\n");
}
printf("EEE\n");
return 0;
}

```

Answer:

DDD

EEE

AAA

CCC

EEE

BBB

EEE

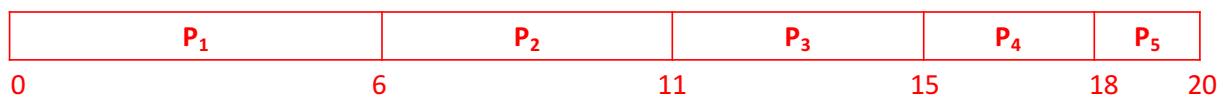
11. (18%) 考慮已經就緒的五個工作，依序為P₁, P₂, P₃, P₄, P₅。使用三個排程演算法FCFS (First-Come, First-Served)、SJF (Shortest-Job-First)以及RR (Round Robin)來排程，而RR所使用的time quantum為2ms。(1)請畫下三個排程演算法的排程圖，(2)請分別算出三個排程演算法中每個工作的等待時間，若無算式一率不給分(算式可以只是簡單的加減法運算)，(3)請分別算出三個排程演算法的平均等待時間，若無算式一率不給分。

Process	Burst Time
P ₁	6 ms
P ₂	5 ms
P ₃	4 ms
P ₄	3 ms
P ₅	2 ms

Answer:

(1) (6%)

FCFS:



SJF:

P₅	P₄	P₃	P₂	P₁	
0	2	5	9	14	20

RR

P₁	P₂	P₃	P₄	P₅	P₁	P₂	P₃	P₄	P₁	P₂	
0	2	4	6	8	10	12	14	16	17	19	20

(2) (6%) (一定要有算式才給分)

FCFS: P₁: 6-6=0, P₂: 11-5=6, P₃: 15-4=11, P₄: 18-3=15, P₅: 20-2=18

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

RR: P₁: 19-6=13, P₂: 20-5=15, P₃: 16-4=12, P₄: 17-3=14, P₅: 10-2=8

(3) (6%) (一定要有算式才給分)

FCFS: $(0+6+11+15+18)/5 = 10 \text{ ms}$ SJF: $(14+9+5+2+0)/5 = 6 \text{ ms}$ RR: $(13+15+12+14+8)/5 = 12.4 \text{ ms}$

12. (8%) 請簡短說明Preemptive SJF以及Non-preemptive SJF有何不同？

Answer: When a task is running, non-preemptive SJF does not allow any preemption, but preemptive SJF will grant the preemption if a new task is ready, and its CPU burst time is shorter then the remaining CPU burst time of the running task.