長庚大學114學年度第一學期作業系統期中測驗（滿分102）

系級: 姓名: 學號:

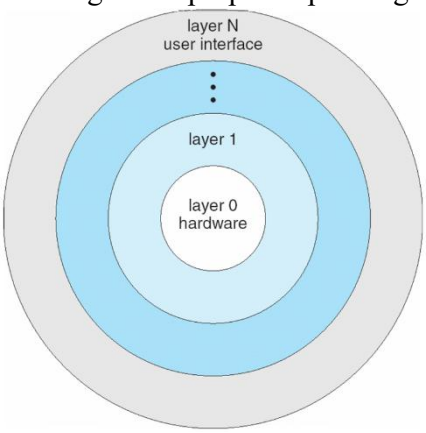
1. (8%) Please provide the definitions of (a) system calls and (b) API

Answer: (a) The system calls provide the routines for user applications to use the functions provided by the operating system (4%).
(b)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 the programmer by the API and are managed by the run-time support library.

2. (10%) Please provide the definitions of (a) multiprogramming (3%) and (b) time sharing (3%). (c) What is the problem if an OS has only multiprogramming technique without the time-sharing support? (4%)

Answer: (a) Multiprogramming: The operating system keeps several jobs in memory simultaneously.
(b) 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.
(c) When some processes take long execution time continuously, the other processes would suffer from long waiting time (thus, the response time could be long).

3. (8%) The layered approach is a typical operating system structure, but to use it for general-purpose operating systems, there could be some problem. Please illustrate one problem when we use the layered approach for a general-purpose operating system.



Answer: Let's have User 1 and User 2 in a system, where User 1 can access Resource 1 and can't access Resource 2, and User 2 can access Resource 2 and can't access Resource 1. It is difficult the properly assign the priorities for User 1 and 2, and the layer of Resource 1 could contradict the layer assignment of Resource 2.
For example, if we let Resource 1 be at layer n and Resource 2 be at layer n+1 for User 2 which can access Resource 2 and can't access Resource 1, then it is impossible to let User 1 can access Resource 1 and can't access Resource 2.

4. (10%) For inter-process communication (IPC), we have message passing IPC and shared memory IPC. (1) Why does the shared memory IPC provide better performance? (4%) (2) Please complete the following example of using shared memory IPC by providing the two lines of C codes.

Shared Memory- Producer

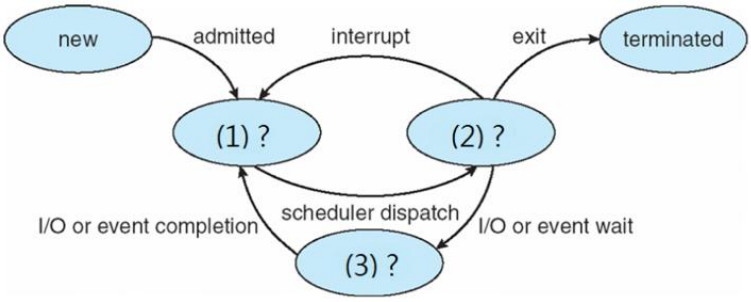
```
while (true)
{
  … /* produce a new item */
  while (((in + 1) % BUFFER_SIZE) == out);
  /* do nothing */
  buffer[in] = next_produced;
  in = (in + 1) % BUFFER SIZE;
}
```

Shared Memory- Consumer

```
while (true)
{
  while (   LINE 1 for Question 2    );
  /* do nothing and have to wait */
  next_consumed = buffer[out];
  out =  LINE 2 for Question 2 ;
  … /* use the consumed item */
}
```

Answer: (1) If we use message passing, a serious of system calls must 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. (4%)
(2) Line 1: in == out (3%)
Line 2: (out + 1) % BUFFER_SIZE (3%)

5. (9%) Please fill up the following figure for the life of a process by provide the contents for (1), (2), and (3).



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

6. (8%) For Inter-Process Communication (IPC), please explain the behaviors of a receiver for conducting (a) Synchronous Message Passing IPC and (b) Asynchronous Message Passing IPC.

Answer: Synchronous Message Passing IPC: The receiver has to wait until the message is received.
Asynchronous Message Passing IPC: If a message is ready, the receiver gets the message. If there is no available message, the receiver just returns with null.

7. (9%) For multithreading models please provide the definitions of (a) Many-to-One, (b) One-to-One and (c) Many-to-Many

Answer: Many-to-One Model: Many user threads are mapped to one kernel thread.
One-to-One Model: One user thread is mapped to one kernel thread.
Many-to-Many Model: Many user threads are mapped to many kernel threads.

8. (8%) Please define (a) the multi-core technique on a processor and the (b) the hyper-threading technique on a core.

Answer: (a) A physical chip with multiple processor cores which can share some resource on the chip.
(b) Hyper-Threading (HT) duplicates the register file of a core so as to allow more than one thread to be brought into the core. When there is some memory stall of a thread, another thread in the same core can immediately takeover the ALU for doing its computation. Thus, HT has the potential for improving parallelization of computations.

9. (8%) Thread Local Storage (TLS) is used to allow thread to have its own copy of data. (a) What is the difference between TLS and global variables? (b) What is the difference between TLS and local variables?

Answer: (a) Global variables are visible across all threads, but TLS is only available in a thread. (4%)
(b) Local variables are visible in only a function, but TLS is available everywhere within its thread. (4%)

10. (12%) We assume that all the fork functions are successfully executed. Please provide the output of the following program:

```
#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("Hi\n");
                }
                else
                {
                        printf("Hola\n");
                }
        }
        else
        {
                wait(NULL);
                printf("Bonjour\n");
        }
        printf("Guten tag\n");
        return 0;
}
```

Answer:
Hello
Hola
Guten tag
Hi
Guten tag
Bonjour
Guten tag

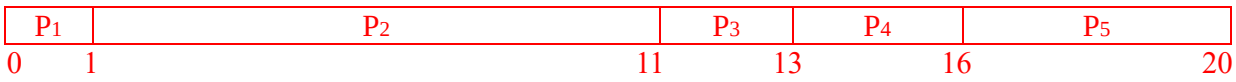
11. (12%) For five ready processes with the arriving order: P_1, P_2, P_3, P_4, P_5, let's use FCFS, SJF and RR for the process scheduling, where the time quantum of RR is 4 ms. (a) Please draw scheduling results of the three scheduling algorithms. (9%) (b) Provide the waiting time of each process for the three scheduling algorithms respectively. (3%)

Process	Burst Time
P_1	1 ms
P_2	10 ms
P_3	2 ms
P_4	3 ms
P_5	4 ms

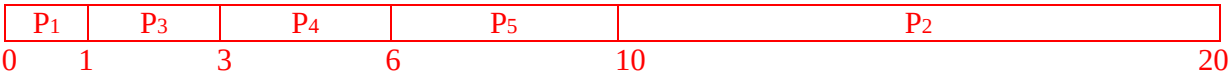
Answer:

(a)

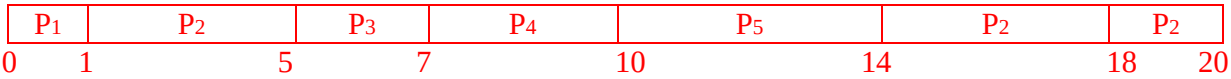
FCFS:



SJF:



RR:



(b)

FCFS: P_1: 1-1=0, P_2: 11-10=1, P_3: 13-2=11, P_4: 16-3=13, P_5: 20-4=16

SJF: P_1: 1-1=0, P_2: 20-10=10, P_3: 3-2=1, P_4: 6-3=3, P_5: 10-4=6

RR: P_1: 1-1=0, P_2: 20-10=10, P_3: 7-2=5, P_4: 10-3=7, P_5: 14-4=10