CITY UNIVERSITY OF HONG KONG

Course code & title	:	CS3103 Operating Systems
Session	:	Semester B 2021/22
Time allowed	:	Two hours

This paper has 9 pages (including this cover page).

1. This paper consists of 2 sections.
2. Answer ALL questions in ALL sections.

*This is a **closed-book** examination.*

Candidates are allowed to use the following materials/aids:
Approved calculators.

No other materials or aids are allowed during the whole examination. If any unauthorized materials or aids are found on a student during the examination, the student will be subject to disciplinary action.

Academic Honesty

I pledge that the answers in this examination are my own and that I will not seek or obtain an unfair advantage in producing these answers. Specifically,

- ❖ *I will not plagiarize (copy without citation) from any source;*
- ❖ *I will not communicate or attempt to communicate with any other person during the examination; neither will I give or attempt to give assistance to another student taking the examination; and*
- ❖ *I will use only approved devices (e.g., calculators) and/or approved device models.*
- ❖ *I understand that any act of academic dishonesty can lead to disciplinary action.*

I pledge to follow the Rules on Academic Honesty and understand that violations may lead to severe penalties.

Student ID:
Name:

CS Departmental Hotline (phone, whatsapp, wechat)
+852 6375 3293

Section A: Short Essay Questions (28 marks)

Question A1 (8 marks)

In modern operating systems, a process has a *ready* state, a *running* state, and a *created* state (also called *new* state).

(1) Please explain each of these three states. (6 marks)

(2) If you want to simply the state transition system of the process, you may merge two of the three above-mentioned states. Which two states of the three above-mentioned states would you choose to merge? Why? (2 marks)

Question A2 (8 marks)

(1) List and explain the reasons of using multi-threads in a process. (4 marks)

(2) What resources are used when a thread is created? How do they differ from those used when a process is created? (4 marks)

Question A3 (4 marks)

Please discuss the pros and cons of multi-level page tables, compared with linear page tables (single-level page tables).

Question A4 (4 marks)

What are the advantages and disadvantages of linked file allocation (also called chained file allocation) compared with contiguous file allocation?

Question A5 (4 marks)

What are long-term scheduling and short-term scheduling?

Section B: Analysis and Calculation Questions (72 marks)

Question B1 (9 marks) Suppose the system currently has two processes *P0* and *P1*.

(1) Let *S* and *Q* be two semaphores, which are both initialized to 1. The pseudo-codes of *P0* and *P1* are shown as follows. Is it possible for the two processes to have a deadlock? Why? (4 marks)

P0	P1

wait(S)
wait(Q)
// critical section
 	
signal(S)
signal(Q)
	wait(S)
	wait(Q)
	// critical section
	
	signal(Q)
	signal(S)

(2) Let *S, T and Q* be three semaphores, where *S* and *T* are both initialized to 1, and *Q* is initialized to 2. The pseudo-codes of *P0* and *P1* are shown as follows. Is it possible for the two processes to have a deadlock? Why? (5 marks)

P0	P1

wait(S)
wait(Q)
wait(T)
// critical section
 	
signal(T)
signal(Q)
signal(S)
	wait(S)
	wait(T)
	wait(Q)
	// critical section
	
	signal(Q)
	signal(T)
	signal(S)

Question B2 (9 marks)

Suppose a hard disk with 3000 tracks, numbered 0 to 2999, is currently serving a request at track 133 and has just finished a request at track 125, and will serve the following sequence of requests:

85, 1470, 913, 1764, 948, 1509, 1022, 1750, 131

Please state the order of processing the requests using the following disk scheduling algorithms and calculate the total movement (number of tracks) for each of them.
(1) SSTF
(2) SCAN
(3) C-SCAN

Hints:

SSTF: Selects the request with the minimum seek time from the current head position.

SCAN: The disk arm starts at one end of the disk, and moves toward the other end, servicing requests until it gets to the other end of the disk, where the head movement is reversed and servicing continues.

C-SCAN: The head moves from one end of the disk to the other, servicing requests as it goes. When it reaches the other end, however, it immediately returns to the beginning of the disk, without servicing any requests on the return trip. Treats the cylinders as a circular list that wraps around from the last cylinder to the first one.

Question B3 (16 marks) Consider the following processes A, B, C and D with arrival times and burst times (i.e., execution times) given in the table:

Process Name	Arrival Time	Burst Time
A	0	12
B	2	8
C	3	5
D	5	4

Compute the *finish time (T_{finish})*, *response time ($T_{response}$)*, *turnaround time* (*$T_{turnaround}$*) and *waiting time* (*$T_{waiting}$*) for each process for the following CPU scheduling algorithms:

a) Preemptive Shortest-Job-First (SJF), which is also called Shortest-Remaining-Time-First (SRTF)

b) Round-Robin (RR) with time quantum = 2

Hints:

$T_{turnaround} = T_{finish} - T_{arrival}$

$T_{response} = T_{firsturn} - T_{arrival}$

$T_{waiting} = T_{turnaround} - T_{burst}$

Please draw and complete the following table on the answer sheet to answer this question.

		A	B	C	D
SJF	Finish				
	Response				
	Turnaround				
	Waiting				
RR (quantum=2)	Finish				
	Response				
	Turnaround				
	Waiting				

Question B4 (14 marks)

Suppose a process has a linear page table with the following assumptions (B: byte; b: bit):

- The page size is 32B.
- The virtual address space for the process is 32KB.
- The physical memory has 256 frames (a frame has the same size as a page).
- Every entry in the page table has one extra VALID bit.
- Each page stores 16 page-table entries (PTEs).

(1) What is the size of each page-table entry? Why? (2 marks)

(2) In each page-table entry, how many bits are used for the page frame number (PFN)? Why? (2 marks)

(3) In the virtual address, how many bits are used for the in-page offset? Why? (2 marks)

(4) How many bits are used by the physical address? Why? (2 marks)

(5) How many bits are used by the virtual address? Why? (2 marks)

(6) What is the total size of the linear page table? Why? (2 marks)

(7) If the process actually uses only 10 pages, how would you design the page table to reduce the total memory overhead of the page table? (You only need to describe the high-level idea of your design, but not the format details.) (2 marks)

Question B5 (14 marks)

Suppose the following global variables, a Boolean array `flag[n]`, an integer variable `turn` and an integer variable `ans` are defined.

```
bool flag[2] = {false, false};
int turn;
int ans = 0;
```

Here is a lock/unlock function pair using `flag[n]` and `turn`.

```
void lock(int self)
{
    flag[self] = true;
    turn = 1-self;
    while (flag[1-self] == true && turn == 1-self);
}

void unlock(int self)
{
    flag[self] = false;
}
```

Suppose there are only two processes P0 and P1 executing concurrently in the system.

They both work on the global variable `ans`, more specifically:

Process 0 executes:

```
void * func_0()
{
    int i = 0;
    lock(0);
    for (i=0; i<10; i++){
        lock(0);
        ans++;  //critical section
        unlock(0);
    }
}
```

Process 1 executes:

```
void * func_1()
{
    int i = 0;
    for (i=0; i<10; i++){
        lock(1);
        ans++;  //critical section
        unlock(1);
    }
}
```

Please answer the following questions.

(1) Is it possible that both processes are trapped in `lock()` (which means the while loop condition is true and thus looping at the while loop) at the same time and thus none of them can enter the critical section? Why? (4 marks)

(2) When the two processes finish execution, is it possible that the value of `ans` is smaller than 20? Why? (4 marks)

(3) If P0 and P1 start execution at the same time, is it possible that P0 executes `ans++` for multiple times before P1 executes `ans++` for the first time? Why? (4 marks)

(4) If P0 and P1 are executing on the same CPU core, is there any disadvantage to implement the lock/unlock function pair as shown in the question? If yes, what are the approaches to address this kind of disadvantage? (2 marks)

Question B6 (10 marks)

(1) Consider a page reference sequence (i.e., reference string) and 3 frames.

0, 1, 0, 2, 3, 0, 4, 1

Please calculate the page hit rate of this reference sequence with FIFO and LRU page replacement policy, respectively. Please write down the calculation procedure.

(2) Please construct a page reference sequence (only using pages 0, 1, 2, 3, 4), with which FIFO outperforms LRU (i.e., FIFO has a higher hit rate than LRU) with 3 frames. Please write down the frame update procedures of your constructed reference sequence with FIFO and LRU, and compute their hit rates, respectively.

Hints: You can draw a table as follows to show the frame update procedure (the number of columns can be increased as needed).

Reference sequence							
FIFO							
LRU							

- END -