CITY UNIVERSITY OF HONG KONG

Course code & title	:	CS3103 Operating Systems
Session	:	Semester B 2020/21
Time allowed	:	Two hours

---

1. This paper consists of 2 sections.
2. Answer ALL questions in both sections.
3. Section A (40%): multiple choices, fill-in-the-blank, short answers
4. Section B (60%): 5 long questions, 12 marks each.
5. Specify the Section and Question number clearly for EACH answer in the answer script.
6. Submit ONE pdf file of the answer script to Canvas.
7. Use your Student ID to name the pdf file.

---

*This is an **open-book** examination.*

***Only course materials** can be accessed.*

***No** access to the Internet and other materials.*

Candidates are allowed to use the following materials/aids:

Approved Calculator

Copy-and-paste the following honesty pledge in the beginning of the answer script and sign your student ID and name to reaffirm your academic honesty pledge.

"I pledge that the answers in this exam/quiz 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 exam/quiz; neither will I give or attempt to give assistance to another student taking the exam/quiz; 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."

Student ID: _____

Student Name: _____

Section A (40 marks), multiple choices, fill-in-the-blank, short answers
Attempt ALL questions from this Section.

1. [1 mark]
Comparing with uniprogramming, multiprogramming generally achieves _____ throughput and _____ resource utilization.
a) higher, higher
b) higher, lower
c) lower, higher
d) lower, lower

2. [1 mark]
Which of the following event(s) may take a process to a ready state?
(i) A process is newly created
(ii) Timeout of a running process
(iii) An I/O event a process waiting for occurs
(iv) Swap a process from disk into main memory
a) (ii) and (iii)
b) (i), (ii) and (iii)
c) (ii) only
d) All of the above

3. [1 mark]
Which of the following about a suspended process must be wrong?
a) Not immediately available for execution
b) Not waiting on an event.
c) Waiting on an event in main memory.
d) Suspended by its parent process.

4. [1 mark]
Which of the following is a possible description of the program output?

```
main()
{     int pid;
      pid = fork();
      printf("%d ", pid);
}
```

a) Two positive integers
b) Two '0's
c) One positive integer
d) A '0' followed by a positive integer

5. [3 marks]
How many times does the following program print CS3103?

```
main( )
{     int i;
      for (i=0; i<3; i++)
      fork( );
      printf("CS3103\n");
}
```

6. [1 mark]
What characteristics are observed while going up the memory hierarchy?
(i) increasing cost per bit
(ii) increasing capacity
(iii) increasing access time
(iv) increasing frequency of access to the memory by the processor
a) (i), (iii) and (iv)
b) (i) and (iv)
c) (ii) and (iii)
d) All of the above

7. [1 mark]
_____, which refers to the tendency of execution to involve a number of memory locations that are clustered, can be exploited by using _____ cache blocks.
a) Temporal locality, recently used
b) Temporal locality, frequently used
c) Spatial locality, small
d) Spatial locality, large

8. [1 mark]
Which of the following is/are disadvantages of kernel-level threads (KLTs)?
(i) The kernel cannot simultaneously schedule multiple threads from the same process on multiple processors.
(~~ii~~) If a thread of a process gets blocked, the whole process blocks.
(iii) The transfer of control from one thread to another within the same process requires a mode switch to the kernel and so is time consuming.
(iv) Creation and destruction of threads in the kernel is costlier.
a) (iii) and (iv)
b) (iii) only
c) (i) and (ii)
d) (i) and (iii)

9. [1 mark]
Which of the following statements about multithreading is false?
a) Threads of the same process can share the same memory.
b) Thread creation is much faster than process creation.
c) Inter-process communication is more efficient than communication between threads.
d) On multiprocessor systems, multithreading takes advantage of the additional hardware, thus resulting in better overall performance.

10. [2 marks]
Consider a memory system with cache access time of 0.1 µs and memory access time (time needed to load a word into the cache) of 1 µs. If the hit ratio is 0.9, what is the average time to access a word?
a) 0.15 µs
b) 0.19 µs
c) 0.20 µs
d) 1.01 µs

11. [2 marks]
Three processes, P0, P1 and P2 are competing for three resources, X, Y, Z. Which of the following request sequence can prevent deadlock?

a) P0: request(X); P1: request(Z); P2: request(X); P0: request(Z); P1: request(Y); P2: request(Y)
b) P0: request(X); P1: request(Z); P2: request(Y); P0: request(Z); P1: request(Y); P2: request(X)
c) P0: request(Z); P1: request(Y); P2: request(X); P0: request(X); P1: request(Z); P2: request(Y)
d) None of the above

12. [2 marks]
If there are three processes and ten identical resources in the system, to ensure that deadlock will never occur, what is the maximum number of resources each process could request, if all processes must have the same maximum number of resources?

13. [1 mark]
When several threads access shared information in main memory, mutual exclusion must be enforced.

a) True
b) False

14. [1 mark]
Which of the following statements about achieving mutual exclusion using semaphore is false?

a) A thread performs the semSignal operation when it exits its critical section.
b) A thread, which performs the semWait operation and sets the semaphore value to 0, can enter its critical section.
c) The semaphore is initialized to 1.
d) None of the above.

15. [1 mark]
Which of the following statements about binary semaphore and mutex is true?

a) A binary semaphore may be initialized to any non-negative integer value.
b) A mutex can be locked and unlocked by different processes.
c) A binary semaphore can always be replaced by a mutex.
d) None of the above

16. [1 mark]
The solution to the Dining Philosophers problem must _____.

a) prevent deadlock
b) prevent starvation
c) ensure mutual exclusion
d) all of the above

17. [1 mark]
Long-term scheduling is used to determine _____.

a) when and which new process is admitted to the ready queue.
b) whether a process is swapped to main memory.
c) which ready process to execute next
d) None of the above

18. [1 mark]
Among the following scheduling algorithms, _____ is preemptive.
a) FCFS
b) SPN
c) HRRN
d) None of the above

19. [2 marks]
Select all scheduling policy(ies) that favour CPU bound jobs over I/O bound jobs.
a) Round Robin
b) FCFS
c) SPN
d) Feedback

20. [1 mark]
Among the following processor scheduling algorithms, _____ requires the estimation of the expected service time.
a) RR
b) Feedback
c) FCFS
d) HRRN

21. [1 mark]
For Round Robin scheduling, when the time slice is too large, Round Robin is equivalent to _____ scheduling algorithm.

22. [2 marks]
Suppose the ready queue is empty while 4 jobs arrive at the same time. Each job's service time is 2 minutes. With the FCFS scheduling policy, what is the average turnaround time?

23. [3 marks]
Consider the following five processes in the table that need to be scheduled. If SPN is used, what is the average turnaround time of the five processes?

Process	arrival time	service time
P1	0.0	9
P2	0.4	4
P3	1.0	1
P4	5.5	4
P5	7	2

13524
9 10 12 16 20
0 1 7 .4 5.5

24. [1 mark]
The correct description about 'memory protection' is _____
a) Prevent hardware memory from damaging
b) Prevent our program from losing data
c) Ensure processes should not be able to reference memory locations in another process without permission.
d) Prevent the program from being swapped

25. [1 mark]
For the following descriptions about page replacement algorithms, _____ is false.

a) A good replacement algorithm should try to minimize the number of page faults
b) FIFO replaces the page with the longest age
c) LRU replaces the page that has not been used for the longest duration
d) FIFO requires significant overhead.

26. [3 marks]
Suppose we have the following page access stream: 5 0 1 2 0 3 0 4 2 3 0 for a system with four frames which are empty at the beginning. Using the least recently used (LRU) page replacement policy, find the number of page faults for the given reference string, assuming that before the frame allocation is initially filled, all first unique pages are also counted as page faults.

27. [2 marks]
Assuming that the rotational speed of the disk is 3000 rpm and the disk surface is divided into 10 sectors, what is the transfer time of one sector?

28. [1 mark]
Thrashing occurs when _____

a) Inadequate resident set leads to frequent process swapping
b) The scheduler flip-flops between two processes, leading to the starvation of others.
c) Two or more processes compete for the same region of shared memory and wait on mutex locks.
d) Multiple processes execute in the same address space.

Section B (60 marks), long questions
Attempt ALL questions from this Section. Each question is worth 12 marks.

Q1 [12 marks]
Mutual Exclusion and Synchronization

Consider the following solution to the ***readers/writers*** problem using ***message passing***.

Suppose the variable `count` is initialized to 100 and all the mailboxes in the system are initially empty. Consider that the following three processes come to the system to read/write the shared file in the sequence described in part (i) to (iii) below.

Process name	Function call
Reader1	`reader(1)`
Reader2	`reader(2)`
Writer3	`writer(3)`

```
void reader(int i)
{
   message rmsg;
      while (true) {
         rmsg = i;
         send (readrequest, rmsg);
         receive (mbox[i], rmsg);
         READUNIT ();
         rmsg = i;
         send (finished, rmsg);
      }
 }
void writer(int j)
{
   message rmsg;
   while(true) {
      rmsg = j;
      send (writerequest, rmsg);
      receive (mbox[j], rmsg);
      WRITEUNIT ();
      rmsg = j;
      send (finished, rmsg);
   }
}
```

```
void  controller()
{
      while (true)
      {
         if (count > 0) {
            if (!empty (finished)) {
               receive (finished, msg);
               count++;
            }
            else if (!empty (writerequest)) {
               receive (writerequest, msg);
               writer_id = msg.id;
               count = count – 100;
            }
            else if (!empty (readrequest)) {
               receive (readrequest, msg);
               count--;
               send (msg.id, "OK");
            }
         }
         if (count == 0) {
            send (writer id, "OK");
            receive (finished, msg);
            count = 100;
         }
         while (count < 0) {
            receive (finished, msg);
            count++;
         }
      }
}
```

(i) [4 marks]

Firstly, Reader1 wants to read the file. List all the messages passing through the mailboxes, which contains *process name*, *action*, *message content* and *mailbox name*, to illustrate how Reader1 can read the file. Also show any change of the value of the variable `count`. The first two steps are shown below as examples.

`count` = 100
Reader1 sends rmsg=1 to readrequest
Reader1 blocks on receive message from mbox[1]

(ii) [2 marks]

Secondly, Reader2 wants to read the file while Reader1 is still reading the file. List all the messages passing through the mailboxes and the values of `count` to illustrate how both readers can read the file simultaneously.

(iii) [6 marks]

Lastly, Writer3 wants to write while the two readers are still reading the file. List all the messages passing through the mailboxes and the values of `count` until all the three processes are done to illustrate how mutual exclusion is achieved.

Q2 [12 marks]
Deadlock

A) [8 marks]

A system has **four** processes and **five** resources. The *Claim* matrix, current *Allocation* matrix and the current *Available* vector are as follows.

Process	Claim	Allocation
A	1 1 2 1 3	1 0 2 1 1
B	1 1 2 2 1	1 1 1 1 0
C	2 1 3 1 0	1 1 0 1 0
D	2 2 2 1 1	2 0 1 1 1

Available
0 0 ***x*** 1 1

What is the **smallest** value of ***x*** for which this is a ***safe state***? Explain and show the steps clearly.

B) [4 marks]

Consider a system with **five** processes P_0 through P_4 and **three** resource types A, B, and C. Resource type A has **5 instances**, resource type B has **3 instances** and resource type C has **5 instances**. Given the following system state, apply the ***deadlock detection algorithm*** to check whether deadlock has occurred or not. If yes, which processes are deadlocked? Show the steps clearly.

	Allocation			Request		
	A	*B*	*C*	*A*	*B*	*C*
P_0	0	2	0	3	0	0
P_1	0	0	0	2	0	2
P_2	0	0	3	0	2	0
P_3	0	1	0	1	0	0
P_4	3	0	2	0	0	1

Q3 [12 marks]
Processor scheduling

Consider the following set of processes in a uniprocessor system. Suppose we use HRRN (*highest response ratio next*) scheduling algorithm with the *response ratio* defined as follows,

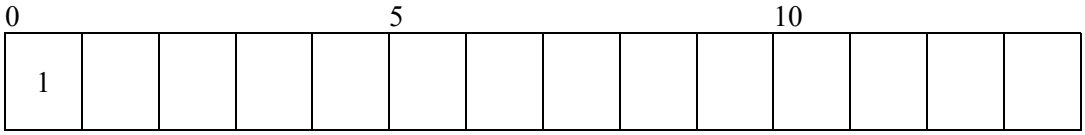
$$R = (w + s) / s,$$

where w is the time spent waiting for the processor and s is the expected service time.

Process ID	Service time	Arrival time
1	4	0
2	6	2
3	2	3
4	2	5

(i) [8 marks]

Each space in the following diagram represents one time unit. Fill in the running process in each space below to show the schedule using HRRN. You need to show the values of the *response ratio* of processes to explain why a process is selected.



(ii) [4 marks]

According to the previous scheduling results, discuss the characteristics of HRRN in terms of how it selects processes.

Q4 [12 marks]
Disk Scheduling

(i) [8 marks]

Given a hard disk of 1000 tracks (Track 0-999). Write down the track numbers that the disk head will travel for the following 4 disk scheduling algorithms, FIFO, SSTF, SCAN, and C-SCAN, with the following sequence of disk track requests: **123, 874, 692, 475, 105, 376**. The disk head is starting at track 345, in the direction of moving towards track 0.

FIFO						
SSTF						
SCAN						
C-SCAN						

(ii) [4 marks]

Select the most effective disk scheduling algorithm(s) (FIFO, SSTF, SCAN, C-SCAN) for the request sequence in part (i) in terms of the total number of tracks travelled. Explain your answer by comparing the total number of tracks travelled for each algorithm.

Q5 [12 marks]
Memory

A) [6 marks]

The following paging system has 64KB main memory, which is divided into 16 frames (frame numbers are from 0 to 15). Suppose the process A has 4 pages. The page numbers are 0, 1, 2, and 3, which are loaded to frame 9, 0, 1, and 14, respectively.

(i) [2 marks]

What is the length of process A?

(ii) [4 marks]

For logical addresses 0001 0000 0100 1000 and 0011 0000 0110 0011 (all logical addresses are binary values), convert them to the physical address in the main memory.

B) [6 marks]

Suppose **three** page frames are assigned to a process and they are initially empty. Before the frame allocation is initially filled, all first unique pages are also counted as page faults.
Complete the table below if we use the ***CLOCK*** replacement algorithm. The first row of the table is the request sequence. Your table should show the *page number* in each frame after each reference. Use a "-" to indicate that a frame is empty. In the last row of the table, mark an "F" for each page reference that results in a *page fault* (if there is no page fault, leave it blank). Use a "*" right next to the page number to indicate that the corresponding *use bit* is equal to 1. Use an arrow '→' in the corresponding frame to indicate the current position of the *next frame pointer*.

time →

Request	1	4	3	1	2	3	1	4	3
Frame 1	1*								
Frame 2	→ -								
Frame 3	-								
Fault	F								

- END -