NATIONAL UNIVERSITY OF SINGAPORE

CS2106 —Operating Systems

Final Assessment

Semester 2 AY2023/24

Time Allowed: 2 Hours

INSTRUCTIONS TO STUDENTS

1. Please WRITE AND SHADE your Student Number. If you do not WRITE AND SHADE your student number properly, you will get zero (0) for this assessment.
2. Do not write your name.
3. This assessment paper contains **FOURTEEN (14)** questions and comprises **NINE (9)** printed pages including this page. The maximum attainable mark is **80**.
4. Students are required to answer **ALL** questions.
5. Write your answers in the spaces provided. Answers written outside of the spaces will not be graded.
6. This is a cheat-sheet assessment. You are allowed a single A4 sheet printed on both sides.
7. Only standalone calculators (normal, scientific, business, graphing) are allowed. No other devices are allowed.

STUDENT NUMBER

A

U
A
HT
NT

For examiner's use only		
Qns	**Marks**	**Remarks**
Q1 to 10		
Q11		
Q12		
Q13		
Q14		
Total		

Part 1. MCQ (20 MARKS)

Choose the best option for each of the following questions. Each question has exactly one answer and is worth 2 marks.

1. A logical memory address
   - ◯ Allows direct access to the physical RAM location.
   - ◯ Is visible to the programmer.
   - ◯ Is the address after translation by the hardware.
   - ◯ Is generated by the operating system.
   - ◯ None of the above A. to D.

2. Memory fragmentation is
   - ◯ The existence of usable areas in the memory of a computer system.
   - ◯ The existence of usable areas within a memory partition of a computer system.
   - ◯ The existence of unreachable area in the memory of computer system.
   - ◯ The existence of unusable areas in the memory of a computer system.
   - ◯ None of the above.

3. Consider a buddy allocation system. Which of the following pairs of addresses belongs to buddy blocks of size 1KiB?
   - ◯ 0xF123, 0xF823
   - ◯ 0xF423, 0xFC23
   - ◯ 0xF034, 0xF234
   - ◯ 0xF234, 0xD234
   - ◯ None of the above

4. Following up from the previous question (Q3), what are the starting addresses of the buddy blocks to which your selected pairs of addresses belong?
   - ◯ 0xF000, 0xF800
   - ◯ 0xF200, 0xF800
   - ◯ 0xF000, 0xFC00
   - ◯ 0xF000, 0xF200
   - ◯ None of the above

5. Consider a machine with 64 MiB physical memory and 32-bit virtual address space. If the page size is 4 KiB, what is the number of page table entries in the direct page table and inverted page table?
   - ◯ Direct: 2^{20}, Inverted: 2^{20}
   - ◯ Direct: 2^{20}, Inverted: 2^{14}
   - ◯ Direct: 2^{14}, Inverted: 2^{20}
   - ◯ Direct: 2^{14}, Inverted: 2^{14}
   - ◯ Direct: 2^{20}, Inverted: 2^{13}

6. Consider the following scenario that happens in order: (1) Process A opens two files File1.txt and File2.txt, (2) Process A then forks a child process B, (3) Process B opens File3.txt, (4) Process B forks a child process C, and (5) Process C close File1.txt. What are the values of the reference count for the files, File1.txt, File2.txt, and File3.txt in the system-wide open file table?
   - ◯ File1.txt: 3 ; File2.txt: 3 ; File3.txt: 3
   - ◯ File1.txt: 3 ; File2.txt: 3 ; File3.txt: 2
   - ◯ File1.txt: 2 ; File2.txt: 3 ; File3.txt: 2
   - ◯ File1.txt: 3 ; File2.txt: 2 ; File3.txt: 3
   - ◯ File1.txt: 2 ; File2.txt: 3 ; File3.txt: 3

7. Consider a combined index allocation scheme implemented for a file system. For simplicity, consider a single disk block can have only 2 entries. Let's assume the following inode structure that contains 5 disk block pointers: 2 direct block pointers, 1 single indirect block pointer, 1 double indirect pointer, and 1 triple indirect pointer. What is the maximum size of a file, in terms of the number of blocks that can be stored in this inode structure?
   - ◯ 2 blocks
   - ◯ 4 blocks
   - ◯ 8 blocks
   - ◯ 16 blocks
   - ◯ None of the above

8. Which of the following statements is false regarding the directory structure in a file system?
   - ◯ Represents the smallest unit of storage of data.
   - ◯ Organize files into a hierarchical structure for easier management.
   - ◯ Provide a way to navigate and access files and directories.
   - ◯ Allow for efficient storage and retrieval of file metadata.
   - ◯ All statements are true.

9. Let the time taken to switch between user and kernel modes of execution be t_1 while the time taken to switch between two processes be t_2. Which of the following is true?
   - ◯ $t_1 = t_2$
   - ◯ $t_1 > t_2$
   - ◯ $t_1 < t_2$
   - ◯ $t_1 >= t_2$
   - ◯ Cannot say

10. Consider a memory snapshot in a point of time contains five memory partitions of size 75KiB (allocated to P1), 60 KiB (free), 100 KiB (allocated to P2), 200 KiB (free), and 50 KiB (allocated to P3). Process requests come in the order: 25KiB, 75KiB, free P1, 50KiB, 75KiB. Which of the following algorithms results in the *largest* free contiguous memory space, after satisfying all the process requests? Assume that the adjacent free locations can be merged.
    - ◯ First Fit
    - ◯ Worst Fit
    - ◯ Best Fit
    - ◯ All three algorithms result in the same *largest* contiguous free space.
    - ◯ No contiguous memory space is free after allocation.

Part 2. Short Questions (60 MARKS)

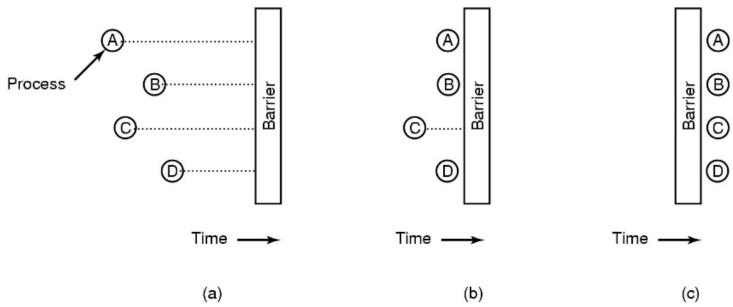
Question 11. Synchronization (10 MARKS)

a. (4 marks) We have the following two processes updating a variable x, whose starting value is 2.

Process A	Process B
x = x + 1	x = x – 3
x = x / 2	

How many possible values of x IN TOTAL are there?

b. (6 marks) A barrier is a synchronization mechanism for *n* processes. Each process will call the barrier. For as long as fewer than *n* processes have called the barrier, all processes will block. When the *nth* process calls the barrier, all processes are unblocked simultaneously. The diagram below shows this idea with a barrier for 4 processes.



In figures a. and b., processes call the barrier and are blocked, while in figure c., the 4th process finally calls the barrier and all are unblocked simultaneously.

We are given the following functions that set the number of processes, and clear the number of processes that have called the barrier:

```
// numProcesses = # of processes to call the barrier before
// it unblocks. processesCalled = # of processes that have
// called the barrier.
int numProcesses = 0, processesCalled = 0;

void setBarrierN(int n) {
      numProcesses = n;
}

void clearBarrierCount() {
      processesCalled = 0;
}
```

You are given the following semaphore operations:

```
// Set semaphore s to initial value v
void initializeSema(TSema *s, int v);

void downSema(TSema *s); // Down the semaphore
void upSema(TSema *s); // Up the semaphore
```

Complete the following code:

```
// Declare your semaphores here.
// Semaphores are of type TSema. For example,
// declaring TSema s; creates a semaphore called s,
// which you can then call initializeSema to
// initialize it, upSema and downSema to use it.









// Initialize you semaphores
void initSema() {






}

// Implement the barrier here
void barrier() {


















}
```

Question 12. Segmented and Disjoint Memory Models (20 MARKS)

In a particular operating system, the text, data, stack and heap segments are laid out next to each other in the process memory space as shown below. Here the code segment starts at address V and is W bytes long. The data segment is X bytes long, the stack segment is Y bytes long, and the heap segment is Z bytes long.

Address Rage	Segment
V to V + (W - 1)	Code (C)
V + W to V + W + (X – 1)	Data (D)
V + W + X to V + W + X + (Y – 1)	Stack (S)
V + W + X + Y to V + W + X + Y + (Z – 1)	Heap (H)

a. (1 marks) We are given the following limits for the segments (as well as the base for the code segment), where the limits are the length of the respective segment in bytes, and base is the starting address of the segment. Given again that the segments are adjacent to each other in the memory, calculate the base addresses of each segment (the base address of the code segment is already given).

Segment	Base	Limit
C	1000	3842
D		286
S		512
H		1024

b. (2 marks) An array A containing 32-bit integers has its first element at (D, 100). Compute the addresses of the following elements, indicating N/A if it results in an access violation:

Array Element	Address (or NA if access violation)
A[4]	
A[20]	
A[26]	
A[31]	

Our segmented memory system is now implemented on a virtual memory system. Each address that you calculated earlier is now a virtual address that must be translated to a physical address. Our system has 512-byte pages and frames.

c. (1 marks) In general, operating systems would not place two different segments in the same page. Explain briefly why (hint: This is particularly true with the code and data segments).

d. (8 marks) Given that two segments cannot be in the same page, we now move each segment to the next adjacent page if they are currently occupying the same page as the previous segment. For example, if both the code and data segments are in page 3, then the data segment will be moved to the start of page 4.

Complete the following table, calculating the base address, starting page and ending page of each segment.

Note: Unlike what is shown in the lecture notes, there is only one page table used for all four segments. Virtual address 0 starts at page 0.

Segment	Base	Limit	Starting Page	Ending Page
C	1000	3842		
D		286		
S		512		
H		1024		

e. (8 marks) Compute the total internal fragmentation of your layout in part d.

Question 13. Virtual Memory (20 MARKS)

Consider a byte-addressed virtual memory system with the following set up: (i) virtual addresses are 64 bits long, (ii) physical addresses are 48 bits long, (iii) a page is 8KiB, (iv) each page table is a page in length, (v) page table entries are 8 bytes, and (vi) it uses 4 levels of page tables.

a. (3 marks) How many logical pages and physical frames are there?

b. (3 marks) How many page entries can a single page table chunk (or tablets) have?

c. (4 marks) How many bits in the logical address remain unused? [2 marks]

d. (4 marks) Suppose a process has 1 frame allocated to it. What is minimum memory space required by the page tables. Do not include the space required by actual frames. Provide your answer in KiB (1 KiB =1024 bytes).

e. (6 marks) Suppose a process has 2097152 (2^{21}) frames allocated to it. What is the minimum memory space required by the page tables. Do not include the space required by actual frames. Provide your answer in KiB (1 KiB =1024 bytes).

Question 14. File Systems (10 MARKS)

We consider an inode file system with 8 direct pointers, 4 single-indirect pointers, 2 double-indirect pointers, and 1 triple-indirect pointer. Each data block is 1024 bytes, and each block pointer is 8 bytes.

a. (2 marks) What is the maximum file size possible on this file system? Express your answer in GiB (1 GiB = 1024 MiB = 1024 x 1024 KiB = 1024 x 1024 x 1024 bytes) correct to 2 decimal places. Assume that there is more than sufficient disk space available to hold the largest possible file.

b. (8 marks) Assuming that only the inode has been loaded into memory, and none of the lower level index blocks have been loaded yet, what is the TOTAL number of blocks that must be loaded from disk when reading data from the following byte offsets (relative to the start of the file):

Offset (bytes from start)	TOTAL # of block read
67	
8100	
534550	
34082815	

~~ END OF ASSESSMENT ~~