PAPER CODE	EXAMINER	DEPARTMENT	TEL
CPT 104	Gabriela Mogos Dongyao Jia	Department of Computing	1515

2nd SEMESTER *2023/24 FINAL* EXAMINATION

Undergraduate Stage 2

OPERATING SYSTEMS CONCEPTS

TIME ALLOWED: 2 Hours

INSTRUCTIONS TO CANDIDATES

1. This is an open-book examination.
2. Total marks available are 100, accounting for 80% of the overall module marks.
3. Answer all FOUR questions.
4. The number in the column on the right indicates the marks for each question.
5. Relevant and clear steps should be included in the answers.
6. The university approved calculator - Casio FS82ES/83ES can be used.
7. Only English solutions are accepted.
8. All materials must be returned to the exam invigilator upon completion of the exam. Failure to do so will be deemed academic misconduct and will be dealt with accordingly.

QUESTION I. Fundamentals (33 marks)

1. Does **Peterson's solution** to the mutual-exclusion problem work when process scheduling is preemptive? How about when it is non preemptive? Explain your answer.

(6 marks)

2. Describe the effects of a corrupted **data block** for a given file for: (a) contiguous, (b) linked, and (c) indexed allocation systems.

(9 marks)

3. In a **fixed-size partitioning** scheme, what are the **two** advantages of using **unequal-size partitions**?

(6 marks)

4. In a system with threads, is there one stack per thread or one stack per process when **user-level threads** are used? What about when **kernel-level threads** are used? Explain your answer.

(6 marks)

5. Consider a system that supports 1,000 users. Suppose that you want to allow 990 of these users to be able to access one file. Please indicate **two methods**. (6 marks)

QUESTION II. CPU scheduling, Memory management and Disk scheduling

(43 marks)

1. Consider the following scenario of processes. Their arrival time and burst time are as follows:

Process	Arrival time (ms)	Burst time (ms)
P1	0	10
P2	1	29
P3	2	3
P4	3	7

Draw the Gantt chart for the execution of the processes using the **Round Robin scheduling algorithm** (time quantum = 10 ms). (4 marks)

Calculate **the average waiting time** for the system. (4 marks)

Calculate **the average turnaround time** for the system. (4 marks)

2. Calculate the number of page faults for the following sequence of page references (each element in the sequence represents a page number) using the **First-In, First-Out (FIFO) replacement algorithm** with frame size of **4**.

2 5 7 3 1 1 2 3 0 7 6 5 4 3 7 5 2 0 0 5

(6 marks)

3. Consider a disk queue with I/O requests on the following cylinders in their arriving order:

70, 20, 30, 530, 330, 150, 450, 400, 570, 290

We assume a hard disk drive with 600 cylinders numbered 0 to 599 and the disk head is initially located at cylinder 60.

Write the sequence in which the requested tracks are serviced using the **Shortest Seek Time First (SSTF) algorithm** and calculate the **total head movement** (in number of cylinders) incurred while servicing these requests. (6 marks)

4. In a paging system with **Translation Lookaside Buffer** (TLB), it takes 50 ns to search the TLB and 100 ns to access the memory.

What is the effective access time (in ns) if the Translation Lookaside buffer (TLB) hit ratio is 60%?

(6 marks)

What should be the hit ratio to achieve the effective memory access time of 210 ns?

(4 marks)

5. Consider the following segment table:

Segment number	**Base address**	**Length**
0	219	600
1	2300	14
2	90	100

What are the physical addresses for the following logical addresses?

Segment number, s	**Offset, d**
0	430
1	10
2	500

(9 marks)

QUESTION III. Resource allocation (12 marks)

Consider the following snapshot of a system:

Available

R1	R2	R3	R4
3	3	2	1

Process	Max				Allocation			
	R1	R2	R3	R4	R1	R2	R3	R4
P0	4	2	1	2	2	0	0	1
P1	5	2	5	2	3	1	2	1
P2	2	3	1	6	2	1	0	3
P3	1	4	2	4	1	3	1	2
P4	3	6	6	5	1	4	3	2

Is this system in a safe state? If your answer is yes, please give a safe sequence and resources available after each process finished. If your answer is no, please specify the processes that might involve in a deadlock (unsafe). (6 marks)

If a request from P4 arrives for (0, 0, 2, 0), can that request be safely granted immediately? Explain your answer. (6 marks)

QUESTION IV. Operating System in C Language (12 marks)

Consider the following C language code implementing a bounded buffer using semaphores:

```
#include <stdio.h>
#include <pthread.h>
#include <semaphore.h>

#define BUFFER_SIZE 5

sem_t mutex, empty, full;
int buffer[BUFFER_SIZE];
```

int in = 0, out = 0;

producer process:	consumer process:

void *producer(void *arg) {
    int item = *(int *)arg;
    sem_wait(&empty);
    sem_wait(&mutex);
    buffer[in] = item;
    in = (in + 1) % BUFFER_SIZE;
    printf("Produced item: %d\n", item);
    sem_post(&mutex);
    sem_post(&full);
    pthread_exit(NULL);
}
	void *consumer(void *arg) {
	    int item;
	    sem_wait(&full);
	    sem_wait(&mutex);
	    item = buffer[out];
	    out = (out + 1) % BUFFER_SIZE;
	    printf("Consumed item: %d\n", item);
	    sem_post(&mutex);
	    sem_post(&empty);
	    pthread_exit(NULL);
	}

a) Explain the purpose of the *empty*, *full*, and *mutex* semaphores in the bounded buffer implementation. (4 marks)

b) Discuss a scenario where deadlock might occur in this implementation. (4 marks)

c) Propose an alternative synchronization mechanism that could replace the semaphore-based approach in the provided code and provide a brief description. (4 marks)

END OF EXAM PAPER