PAPER CODE	EXAMINER	DEPARTMENT	TEL
CPT 104		Department of Computing	

2020/21 SEMESTER 2 – Final Exam

BACHELOR DEGREE – Level 2

Operating Systems Concepts

TIME ALLOWED: 2 hours

INSTRUCTIONS TO CANDIDATES

1. **Total marks available are 100, accounting for 80% of the overall module marks.**
2. **Answer all FOUR questions.**
3. **The number in the column on the right indicates the marks for each question.**
4. **Relevant and clear steps should be included in the answers.**
5. **The university approved calculator - Casio FS82ES/83ES can be used.**
6. **All the answers must be in English in the answer script provided.**

QUESTION I. Fundamentals (46 marks)

1. Compare and contrast **internal fragmentation** and **external fragmentation**. Explain the circumstances where one might be preferred over the other. (6 marks)

2. What is a **virtual machine**? Briefly discuss each of the two components that make up a virtual machine. (6 marks)

3. The **file system** resides permanently on secondary storage. Briefly discuss each of the three major methods of allocating disk space. (12 marks)

4. Describe in your own words how a parent and child are created from the **fork ()** system call and why you think the child process will or will not continue if the parent process is terminated. (6 marks)

5. Describe in your own words why **mutual exclusion** is necessary for multiprogramming systems. (4 marks)

6. System protection is multifaceted. What are the four levels of **security** measures that are necessary for system protection? Name and explain them. (12 marks)

QUESTION II. CPU scheduling, Memory management, Disk scheduling

(34 marks)

1. Consider the following scenario of processes:

Process	Arrival time	Burst time
P1	0	9
P2	1	5
P3	2	3
P4	3	4

Draw the Gantt chart for the execution of the processes, showing their start time and end time, using ***First-Come First-Served FCFS*** scheduling algorithm. (2 marks)

Calculate average turnaround time, and average waiting time for the system in ***First-Come First-Served FCFS*** scheduling algorithm. (8 marks)

2. There is a system with **64 pages** of **512 bytes** page size and a physical memory of **32 frames**. How many bits are required in the logical and physical address? (6 marks)

3. Calculate the number of page faults for the following reference string using **First-In, First-Out FIFO algorithm** with frame size as **4**.

5 0 2 1 0 3 0 2 4 3 0 3 2 1 3 0 1 5

(6 marks)

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

54, 97, 73, 128, 15, 44, 110, 34, 45

The disk head is assumed to be at cylinder number **23**.

Write the sequence in which requested tracks are serviced using **Shortest-Seek-Time-First (SSTF) algorithm** and calculate the total head movement (in number of cylinders) incurred while servicing these requests. (5 marks)

5. Consider a **real-time system** in which there are three processes. Their period and execution time are as follows:

Process	Execution time, t	Period, p
P1	35	100
P2	10	50
P3	30	150

Calculate the total utilization of CPU. (1 mark)

We assume that all three processes are released at time 0. Explain the **Rate Monotonic Scheduling Algorithm** of the processes. (4 marks)

Show the processes on timing diagram. (2 marks)

QUESTION III. Resource allocation (10 marks)

Consider a system with the following information and assume that the system is in safe state.

At this moment, if **P4** requests two more instances of **R1** and two instances of **R3**, will the system still be in safe state? Explain the answer.

Total resources

R1	R2	R3
15	8	8

Process	Max			Allocation		
	R1	R2	R3	R1	R2	R3
P1	5	6	3	2	1	0
P2	8	5	6	3	2	3
P3	4	9	2	3	0	2
P4	7	4	3	3	2	0
P5	4	3	3	1	0	1

QUESTION IV. Operating System in C Language (10 marks)

If the statement **while** *flag*[j] **and** *turn = j* in Peterson's algorithm is changed to **while** *flag[j]* **or** *turn = j*, which properties of **Critical Section** implementation are violated by the resulting system? Explain.

```
int turn;
boolean flag[2];
        do
        {
        flag[i] = true;
        turn = j;
        while (flag[j] && turn == j);
        // critical section
        flag[i] = false;
        // remainder section
        }
        while (true);
```

Xi'an Jiaotong-Liverpool University
西交利物浦大學

END OF EXAM PAPER