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

PAPER CODE	EXAMINER	DEPARTMENT	TEL
CPT 104	XXXX	XXXXX	XXX

2021/22 SEMESTER 2 – Open Book Final Exam

BACHELOR DEGREE – Year 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 (45 marks)

1. **Multi-threading** is implemented in modern operating systems for better efficiency and performance. Describe **3 key factors** to be considered while choosing multi-threading for an application. (12 marks)

2. Briefly describe the purpose of a **translation lookaside buffer TLB**. (5 marks)

3. There are many reasons why the system administrator would want to **restrict access** to areas of **memory**. Briefly discuss **three important reasons**. (9 marks)

4. The directories are used to maintain the structure of a file system. Describe **3 advantages** to be considered while using directories. (9 marks)

5. If a **process** terminates, will its threads also terminate, or will they continue to run? If all of its threads terminate, will the process also terminate, or will it continue to run? Explain your answers. (6 marks)

6. Describe why **Direct Memory Access (DMA)** is considered an efficient mechanism for performing I/O. (*NO answer should be longer than two or three sentences*) (4 marks)

QUESTION II. CPU scheduling, Memory management, Disk scheduling

(31 marks)

1. Consider a **Real-Time System** in which there are three processes. Their period and execution time are as follows:

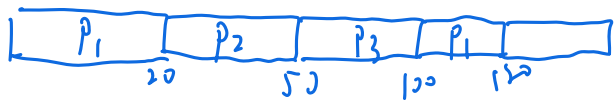
Process	Execution time, t	Period, p
P1	20	100
P2	30	145
P3	68	150

Calculate the total utilization of CPU. $\frac{20}{100}+\frac{30}{145}+\frac{68}{150}=0.86$ (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)

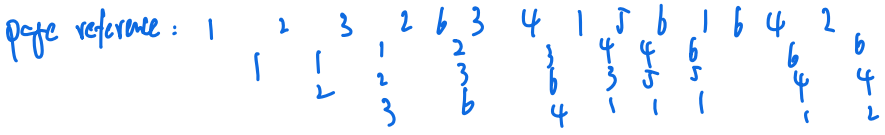
Since 150 > 145 > 100, so the priority is $P_1 > P_2 > P_3$.



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 **Least Recently Used LRU page-replacement algorithm** with frame size of **3**.

1 2 3 2 6 3 4 1 5 6 1 6 4 2

(6 marks)



page faults: 10

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

55, 58, 39, 18, 90, 160, 150, 38, 184

We assume a disk with 200 tracks and the disk head is initially located at track **100**.

Write the sequence in which requested tracks are serviced using the **SCAN algorithm** and calculate the **total head movement** (in number of cylinders) incurred while servicing these requests. 0 18 38 39 55 58 90 100 150 160 184 (6 marks)

100 + 184 = 284

1 2 3 4

4. Given the actual (measured) CPU bursts of the first four processes are **5, 2, 6, 4** (ms), and predicted burst time for the first process is **11** ms, show the fifth process **prediction** when the factor **a** takes the value of **0.3**. (5 marks)

$\tau_2 = \alpha t_1 + (1-\alpha)\tau_1 = 0.3\times5 + 0.7\times11 = 9.2$

$\tau_3 = \alpha t_2 + (1-\alpha)\tau_2 = 0.3\times2 + 0.7\times9.2 = 7.04$

$\tau_4 = \alpha t_3 + (1-\alpha)\tau_3 = 0.3\times6 + 0.7\times7.04 = 6.728$

$\tau_5 = \alpha t_4 + (1-\alpha)\tau_4 = 5.91$ ms

5. Three processes **P1**, **P2**, and **P3** of size 67000, 65000, and 60000 bytes, respectively, need space in the memory.

If partitions of equal size, for example, 70000 bytes, are allocated to P1, P2, and P3, will there be any **fragmentation** in this allocation? If, yes, then what is the size of the space left? (4 marks)

Can a process of 15000 bytes be accommodated? (3 marks)

P_1: 70000 - 67000 = 3000

P_2: 70000 - 65000 = 5000

P_3: 70000 - 60000 = 10000

total: 3000 + 5000 + 10000 = 18000 bytes

No. there are no partitions have sizes ≥ 18000 bytes.

QUESTION III. Resource allocation (12 marks)

Consider a system with the following information.

Total resources

R1	R2	R3	R4
6	4	4	2

Available

R_1	R_2	R_3	R_4
1	1	2	0

Process	Max				Allocation			
	R1	R2	R3	R4	R1	R2	R3	R4
P1	3	2	1	1	2	0	1	1
P2	1	2	0	2	1	1	0	0
P3	1	1	2	0	1	1	0	0
P4	3	2	1	0	1	1	1	0
P5	2	1	0	1	0	0	0	1

Need

R_1	R_2	R_3	R_4
1	2	0	0
0	1	0	2
0	0	2	0
2	1	1	0
2	1	0	0

Does this initial allocation lead to a safe state? If yes, find out the safe sequence of process execution. Explain with reason.

$P_3 \rightarrow P_1 \rightarrow P_4 \rightarrow P_5 \rightarrow P_2$

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

Consider **Readers–Writers Problem** that single writer and multiple readers share a memory area (critical section).

Suppose (1) only one single writer can access the shared data at the same time, any other writers or readers must be blocked, and (2) allow multiple readers to read at the same time, any writers must be blocked.

The corresponding writer process and reader process are programmed as follows, where *readcount* variable keeps track of how many processes are currently reading the object, semaphore *mutex* is used to ensure mutual exclusion when the variable *readcount* is updated, and semaphore *rw_mutex* is used to ensure mutual exclusion when accessing the critical section.

```
/* semaphore initialization*/
semaphore mutex = 1;
semaphore rw_mutex = 1;
int readcount = 0;
```

Writer process:	Reader process:

while (true) {
  wait(rw_mutex);
  . . .
  /* writing is performed */
  . . .
  signal(rw_mutex);
}
	1. while (true) {
	2. wait(mutex);
	3. readcount++;
	4. if (readcount == 1)
	5. wait(rw_mutex);
	6. signal(mutex);
	7. . . .
	8. /* reading is performed */
	9. . . .
	10. wait(mutex);
	11. readcount--;
	12. if (readcount == 0)
	13. signal(rw_mutex);
	14. signal(mutex);
	15. }

1. Please indicate limitation of the solution. Starvation. The writer process will be blocked. (4 marks)

2. What may happen if *mutex* related code are removed from reader process (i.e. line 2, 6, 10, 14 are deleted)? (4 marks)

There will be a race condition that readcount will be modified and destroy the mutual exclusion.

3. What may happen if the reader process is modified as follows (i.e., only *rw_mutex* is used)?

(4 marks)

Reader process:

while (true) {
  wait(rw_mutex);
  . . .
  /* reading is performed */
  . . .
  signal(rw_mutex);
}

Only one reader can read at the same time.
Which will against the function that multiple readers can read at the same time.

END OF EXAM PAPER