CONFIDENTIAL

UNIVERSITI TUN HUSSEIN ONN MALAYSIA

FINAL EXAMINATION
SEMESTER II
SESSION 2022/2023

COURSE NAME	:	OPERATING SYSEM
COURSE CODE	:	BNF32303
PROGRAMME CODE	:	BNF
EXAMINATION DATE	:	JULY / AUGUST 2023
DURATION	:	2 HOURS 30 MINUTES
INSTRUCTION	:	1. ANSWER **ALL** QUESTIONS 2. THIS FINAL EXAMINATION IS CONDUCTED VIA **CLOSE BOOK** 3. STUDENTS ARE **PROHIBITED** TO CONSULT THEIR OWN MATERIAL OR ANY EXTERNAL RESOURCES DURING THE EXAMINATION CONDUCTED VIA CLOSED BOOK

THIS QUESTION PAPER CONSISTS OF **SEVEN (7)** PAGES

CONFIDENTIAL

Q1 (a) List **SIX (6)** services provided by an operating system. Describe how each creates convenience for users and justify in which cases would it be impossible for user-level programs to provide these services?

(6 marks)

(b) Operating Systems employ dual mode operation in order to protect computer hardware from erroneous user program.

(i) Determine and state whether the following instructions can be done in user mode or kernel (privileged) mode.

- Set value of timer.
- Read the clock.
- Clear memory.
- Turn off interrupts.
- Switch from user to kernel mode.
- Access I/O device.

(3 marks)

(ii) Assume a user program intend to open a file using system-call 'open()'. Draw a diagram to demonstrate the implementation of this system-call. Include both modes operation and 'system-call interface' in your diagram.

(4 marks)

(c) In computer system structure, there are **TWO (2)** types of input and output access method namely synchronous and asynchronous. Differentiate these methods based on its process and aid your answer by drawing the process diagram.

(7 marks)

Q2 (a) Using the program shown in **Figure Q2(a)**, determine the output at LINE A. Explain your answer.

(3 marks)

(b) Including the initial parent process, determine how many processes are created by the program shown in **Figure Q2(b)**?

(2 marks)

(c) As a process executes, it changes state. Each process may be in one of the following states: new, running, waiting, ready and terminated.

(i) Explain each of the process state.

(5 marks)

(ii) Illustrate the process state diagram and explain the connections between each state.

(4 marks)

(d) Using Amdahl's Law, calculate the speedup gain of an application that has a 60 percent parallel component for:

(i) Two processing cores
(ii) Four processing cores

(6 marks)

Q3 (a) List **THREE (3)** examples of deadlocks that are not related to a computer system environment.

(3 marks)

(b) Consider the snapshot of a system as shown in **Figure Q3(b)**. Analyse the system and answer the following questions using the banker's algorithm:

(i) Determine what is the content of the matrix **Need.**

(2 marks)

(ii) Determine whether the system in a safe state.

(2 marks)

(iii) If a request from thread T_1 arrives for (0,4,2,0), can the request be granted immediately?

(3 marks)

(c) Given the system model:

Processes, $P = \{P_1, P_2, P_3, P_4\}$
Resources, $R = \{R_1, R_2, R_3\}$
Resource instances, $W_1 = 3$, $W_2 = 2$, $W_3 = 4$
Edges, $E = \{P_1 \rightarrow R_1, P_1 \rightarrow R_2, P_2 \rightarrow R_2, P_3 \rightarrow R_1, P_3 \rightarrow R_3, R_1 \rightarrow P_2, R_2 \rightarrow P_4, R_3 \rightarrow P_3\}$

(i) Draw the corresponding resource allocation graph based on the system model.

(ii) Determine whether the resource allocation graph in question **Q3(c)(i)** is in deadlock or not.

(iii) Determine the condition if P_4 request for R_3 is granted.

(10 marks)

Q4 (a) Consider the set of processes given in **Figure Q4(a)**, with the length of the CPU burst time given in milliseconds. The processes are assumed to have arrived in the order P_1, P_2, P_3, P_4, P_5, all at time 0.

(i) Draw four Gantt charts that illustrate the execution of these processes using the following scheduling algorithms: FCFS, SJF, non-preemptive priority (a larger priority number implies a higher priority), and RR (quantum = 2).

(8 marks)

(ii) Calculate the turnaround time of each process for each of the scheduling algorithms in **Q4(a)(i)**.

(5 marks)

(iii) Calculate the waiting time of each process for each of these scheduling algorithms.

(5 marks)

(iv) Determine which of the algorithms results in the minimum average waiting time (over all processes). Justify your answer.

(2 marks)

Q5 (a) Given process P_1, P_2, P_3, P_4 and P_5 are requesting memory of 115 kB, 500 kB, 358 kB, 200 kB, and 375 kB (in order) respectively. Original memory partitions are given in **Table Q5(a)**. Illustrate the new memory partition after inserting the process by using:

(i) First fit algorithm
(ii) Best fit algorithm

(10 marks)

(b) Assuming a 1-KB page size, determine the page numbers and offsets for the following address references (provided as decimal numbers):

(i) 3085
(ii) 42095
(iii) 215201
(iv) 650000
(v) 2000001

(5 marks)

(c) Consider a file currently consisting of 100 blocks. Assume that the file-control block (and the index block, in the case of indexed allocation) is already in memory. Calculate how many disk I/O operations are required for contiguous, linked, and indexed (single-level) allocation strategies, if, for one block, the following conditions hold. In the contiguous-allocation case, assume that there is no room to grow at the beginning but there is room to grow at the end. Also assume that the block information to be added is stored in memory.

(i) The block is added at the beginning.
(ii) The block is added in the middle.
(iii) The block is added at the end.
(iv) The block is removed from the beginning.
(v) The block is removed from the middle.

(5 marks)

- END OF QUESTIONS -

FINAL EXAMINATION

SEMESTER / SESSION : SEM II / 2022/2023
COURSE NAME : OPERATING SYSTEM

PROGRAMME CODE : BNF
COURSE CODE : BNF 32303

```
#include <sys/types.h>
#include <stdio.h>
#include <unistd.h>

int value = 5;

int main()
{
pid_t pid;

   pid = fork();

   if (pid == 0) { /* child process */
      value += 15;
      return 0;
   }
   else if (pid > 0) { /* parent process */
      wait(NULL);
      printf("PARENT: value = %d",value); /* LINE A */
      return 0;
   }
}
```

Figure Q2(a)

```
#include <stdio.h>
#include <unistd.h>

int main()
{
    /* fork a child process */
    fork();

    /* fork another child process */
    fork();

    /* and fork another */
    fork();

    return 0;
}
```

Figure Q2(b)

FINAL EXAMINATION

SEMESTER / SESSION : SEM II / 2022/2023
COURSE NAME : OPERATING SYSTEM

PROGRAMME CODE : BNF
COURSE CODE : BNF 32303

	Allocation	*Max*	*Available*
	A B C D	*A B C D*	*A B C D*
T_0	0 0 1 2	0 0 1 2	1 5 2 0
T_1	1 0 0 0	1 7 5 0	
T_2	1 3 5 4	2 3 5 6	
T_3	0 6 3 2	0 6 5 2	
T_4	0 0 1 4	0 6 5 6	

Figure Q3(b)

Process	Burst Time	Priority
P_1	2	2
P_2	1	1
P_3	8	4
P_4	4	2
P_5	5	3

Figure Q4(a)

Table Q5(a)

300 kB
600 kB
350 kB
200 kB
750 kB
125 kB