CURTIN UNIVERSITY
DEPARTMENT OF COMPUTING

Test – S1/2017

SUBJECT: OPERATING SYSTEMS Code COMP2006

TIME ALLOWED:

90 minutes test preceded by a 10-MINUTE READING PERIOD during which time only notes may be made. The supervisor will indicate when answering may commence.

AIDS ALLOWED:

To be supplied by the Candidate: Nil

To be supplied by the University: Nil

Calculators are NOT allowed.

GENERAL INSTRUCTIONS:

This paper consists of Four (4) questions with a total of 100 marks.

ATTEMPT ALL QUESTIONS

Name: _____

Student No: _____

Tutorial Time/Tutor: _____

QUESTION ONE (total: 26 marks): Operating Systems.

a) **(Total: 4 marks).** Explain the main advantage of a multitasking system, and explain how the system achieves the advantage.

Answer:

b) **(Total: 4 marks).** Consider the following storage types:

Cache, magnetic disk, magnetic tape, main memory, register, solid state disk.

(i) **(2 marks).** List the storage types in order of their increasing capacity/size.

(ii) **(2 marks).** List all of the storage types that belong to volatile storage.

Answer:

(i)

(ii)

c) **(6 marks).** A host communicates with an I/O controller using one of the following three methods: polling, interrupts, or direct memory access.

Explain how the **polling** method knows when the I/O is available, and describe its main advantage and main disadvantage.

Answer:

d) **(4 marks).** Describe two main advantages of a multiprocessor system.

Answer:

e) **(6 marks).** Briefly describe three examples of event that may trigger interrupts.

Answer:

f) **(2 marks).** List two operating system designs other than the microkernel.

Answer:

END OF QUESTION ONE

QUESTION TWO (total: 22 marks): Processes and threads

a) **(4 marks).** Explain the main differences between a process and a program.

Answer:

b) **(Total: 4 marks).**

(i) **(2 marks).** Explain how a zombie process is created.

(ii) **(2 marks).** Explain the role of the **init** process on UNIX and Linux systems in regards to zombie process.

Answer:

(i)

(ii)

c) **(Total: 6 marks).** Consider the following C program that runs on Linux.

```
int main(int argc, char** argv)
{
    int i;
    for (i = 0; i < 2; i++) {
        fork();
        printf("AAA\n");  // Line A
    }
    printf("BBB\n"); // Line B
}
```

(i) **(2 marks).** When a Linux program calls fork(), what possible values will be returned if there is no error?

(ii) **(2 marks).** How many times would this program print "AAA"? Justify your answer.

(iii) **(2 marks).** How many times would this program print "BBB"? Justify your answer.

Answer:

(i)

(ii)

(iii)

d) **(4 marks).** Among others, thread creation and deletion are more efficient as compared to those in processes. Describe two other main advantages of multiple threads as compared to multiple processes.

Answer:

e) **(4 marks).** Consider the following two components of a process control block.

- Register values
- Heap memory

Explain why or why not each component is shared across threads in a multithreaded process.

Answer:

- Register values

- Heap memory

END OF QUESTION TWO

QUESTION THREE (total: 24 marks): Process Synchronization

a) **(4 marks).** Describe the Critical-Section Problem.

Answer:

b) **(4 marks).** One solution to enforce mutual exclusion is by disabling and enabling interrupt. Is such solution good **in general**? If yes, explain its advantages. If not, describe its disadvantages or its possible problems.

Answer:

c) **(Total: 4 marks).**

(i) **(2 marks).** What is an *atomic instruction*?

(ii) **(2 marks).** Give two examples of atomic instruction.

Answer:

(i)

(ii)

d) **(4 marks).** Explain one main advantage and one main disadvantage of using spinlock.

Answer:

e) **(Total: 8 marks).** Consider the following solution to the readers-writers problem.

```
Shared data
var mutex, wrt: semaphore;
    readcount: integer;
    readcount := 0;
    mutex := 1;
    wrt := ?;
```

Reader process

```
repeat

    P(mutex);
            readcount = readcount + 1;
            if readcount = 1 then
                    P(wrt);
    V(mutex);
            ...
            reading is performed;
            …
    P(mutex);
            readcount = readcount - 1;
            if readcount = 0 then
                V(wrt);
    V(mutex);

until false;
```

Writer process

```
repeat

    P(wrt);
            …
            writing is performed
            …
    V(wrt);

until false;
```

(i) **(2 marks).** Explain the use of semaphore `mutex` in the solution.

(ii) **(2 marks).** Should semaphore `wrt` be initialized to 0 or 1? Why?

(iii) **(4 marks).** The code allows each arriving reader to read when there is at least one reader reading. Which part of the code enforces it?

Answer:

(i)

(ii)

(iii)

END OF QUESTION THREE

QUESTION FOUR (total: 28 marks): CPU Scheduling

a) **(4 marks).** Give two examples of the pre-emptive scheduler. For each example, explain why it is a pre-emptive scheduler.

Answer:

b) **(Total: 8 marks).**

(i) **(2 marks).** Some scheduling criteria include: CPU utilisation and response time. Describe each of the two criteria.

(ii) **(6 marks).** The performance of a round robin scheduling system depends on the size of time quantum q. Explain the reasons. In your explanation, discuss the performance of the system, in terms of **CPU utilization** and **response time**, when the size of q is small and when it is large with respect to the average size of processes running in the system.

Answer:

(i)

(ii)

c) **(Total: 8 marks).** Consider the following function, i.e., exponential average.

$$\tau_{n+1} = \alpha.t_n + (1-\alpha).\tau_n$$

(i) **(4 marks).** For CPU scheduler, when and why is the function used?

(ii) **(4 marks).** Explain how to set the value of α.

Answer:

(i)

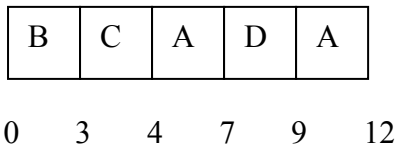
(ii)

d) **(4 marks).** Given the following set of processes with their arrival times and burst time, draw the Gantt chart for **non pre-emptive** shortest job first scheduling.

Process	Arrival Time	Burst Time
A	0	6
B	0	3
C	2	1
D	7	2

Answer:

e) **(Total: 4 marks).** For the given set of processes with their arrival times and burst time in question part d), consider a scheduling algorithm that has the following Gantt chart.



(i) **(2 marks).** What CPU scheduler produces the Gantt chart?

(ii) **(2 marks).** Calculate the average waiting time produced by the scheduler.

Answer:

(i)

(ii)

END OF TEST PAPER