THE UNIVERSITY of ADELAIDE

# Primary Examination

## School of Computer Science
## The University of Adelaide

**Semester 2, 2014**

**OPERATING SYSTEMS**
**COMP SCI 3004, 7064**

| | |
|---|---|
| Official Reading Time: | 10 mins |
| Writing Time: | 120 mins |
| Total Duration: | 130 mins |

| Questions | Time | Marks |
|---|---|---|
| Answer all **5** questions | 120 mins | 100 marks |
| | | 100 Total |

Instructions

- Begin each answer on a new page
- Examination material must not be removed from the examination room
- Simple, Non-programmable Calculators Allowed

Materials

- 1 Blue book

DO NOT COMMENCE WRITING UNTIL INSTRUCTED TO DO SO

**Question 1**

Answer the following questions:

(a) An operating system can be viewed as a resource manager. Explain how it manages the memory resource in paging.

[3 marks]

(b) Explain the purpose of demand paging. What are the pros and cons for admitting more processes to the memory?

[4 marks]

(c) What are the differences between a trap and an interrupt? Can traps be generated intentionally by a user program? If so, for what purpose?

[4 marks]

(d) Describe how to prevent deadocks for resource allocation in systems with each resource type containing multiple instances.

[4 marks]

**[Total for Question 1: 15 marks]**

**Question 2**

Consider the following sets of processes, with the length of the CPU-burst time given in milliseconds:

| Process | Arrival Time | Burst Time | Priority |
|---|---|---|---|
| $P_1$ | 0 | 6 | 1 |
| $P_2$ | 1 | 4 | 2 |
| $P_3$ | 4 | 6 | 4 |
| $P_4$ | 2 | 3 | 3 |

For each of the following scheduling algorithms, determine the *average turnaround time* and *average waiting time* respectively. Show working (using a Gantt chart).

(a) First-Come-First-Served scheduling.

[4 marks]

(b) Shortest-Job-First scheduling (preemptive).

[4 marks]

(c) Priority scheduling, where a larger priority number indicates a higher priority (preemptive).

[4 marks]

(d) Round-Robin scheduling (quantum=2).

[4 marks]

**[Total for Question 2: 16 marks]**

**Question 3**

Consider the following page reference string:

$$3, 1, 2, 3, 2, 1, 4, 3, 4, 5, 2, 5, 4, 6, 2, 3, 5, 2, 6, 1.$$

How many page faults would occur for each of the following replacement algorithms in the three cases of **two**, **three** and **four** frames available. Note that this means your answers to the questions below will consist of nine numbers in total. Also note that all frames are initially empty, so your first unique pages will all cost one fault each.

(a) FIFO replacement.

[6 marks]

(b) Optimal replacement.

[6 marks]

(c) LRU replacement.

[6 marks]

**[Total for Question 3: 18 marks]**

**Question 4**

(a) Consider a file system on a disk that has both logical and physical block sizes of 512 bytes. Assume that the information about each file is already in memory. For each of the three allocation strategies (**contiguous**, **linked**, and **indexed**), answer the following. You can assume the starting physical block number is **Z**, a disk address requires 4 bytes.

i. How is the logical-to-physical address mapping accomplished in this system? (For the indexed allocation, assume that a file is always less than 512 blocks long.)

[9 marks]

ii. If we are currently at logical block 10 (the last block accessed was block 10) and want to access logical block 6, how many physical blocks must be read from the disk?

[3 marks]

(b) Suppose that a disk drive has 5000 cylinders, numbered 0 to 4999. The drive is currently serving a request at cylinder 1143, and the previous request was at cylinder 1100. The queue of pending requests, in FIFO order, is 186, 2470, 1913, 774, 1948, 1022, 2750, 1130.

Starting from the current head position, what is the total distance (in cylinders) that the disk arm moves to satisfy all the pending requests, for the C-SCAN disk-scheduling algorithm?

[4 marks]

(c) Direct memory access is used for high-speed I/O devices in order to avoid increasing the CPU's execution load.

i. How does the CPU interface with the device to coordinate the transfer?

[2 marks]

ii. How does the CPU know when the memory operations are complete?

[2 marks]

(d) Discuss the strengths and weaknesses of implementing an access matrix using capabilities that are associated with domains.

[4 marks]

(e) Discuss a means by which managers of systems connected to the Internet could have designed their systems to limit or eliminate the damage done by a worm. What are the drawbacks of making the change that you suggest?

[3 marks]

(f) Compare symmetric and asymmetric encryption schemes, and discuss under what circumstances a distributed system would use one or the

other.

[4 marks]

**[Total for Question 4: 31 marks]**

**Question 5**

Another classical synchronization problem takes place in a barber shop. The barber shop has one barber, one barber chair, and N chairs for waiting customers, if any, to sit in. If there are no customers present, the barber sits down in the barber chair and falls asleep. When a customer arrives, he has to wake up the sleeping barber. If additional customers arrive while the barber is cutting a customers hair, they either sit down (if there are empty chairs) or leave the shop (if all chairs are full).

The problem is to program the barber and the customer without getting into race conditions. The following pseudo code fragment of a solution without the calls of the semaphore operations is provided for your convenience. Fill in the missing semaphore operations in the barber() and customer() functions to make the example work.

```
const int N;
semaphore customers = 0;
semaphore barbers = 0;
semaphore mutex = 1;
shared int waiting = 0;

void barber()
{
        while (true) {
            waiting– –;
            cut_hair();
        }
}

void customer()
{
        if (waiting i N) {
            waiting+ +;
            get_haircut();
        }
}
```

**[Total for Question 5: 20 marks]**

**End of exam**