THE UNIVERSITY OF ADELAIDE

Examination for the Degrees of
Bachelor of Computer Science, Bachelor of Design Studies,
Bachelor of Engineering (Computer Systems Engineering),
Bachelor of Engineering (Information Technology and Telecommunications Engineering),
Bachelor of Information Science, Bachelor of Science,
Bachelor of Science (Mathematical and Computer Sciences),
Graduate Certificate in Computer Science,
Graduate Diploma in Computer Science,
Master of Information Technology,
Master of Engineering Science and
Master of Computer Science

Primary Examination – Semester Two, November, 2004

Comp Sci 3004/6014/7064: Operating Systems

Official Reading Time: 10 minutes.
Writing Time: 120 minutes.
Total Duration: 130 minutes.

Total Marks: 120

THIS IS A CLOSED BOOK EXAMINATION

ANSWER ALL SIX QUESTIONS, SHOWING ALL WORKING

CALCULATORS ARE NOT PERMITTED

EXAMINATION MATERIALS MUST NOT BE REMOVED FROM THE EXAMINATION ROOM

THERE ARE FIVE PAGES IN ADDITION TO THIS PAGE

DO NOT COMMENCE WRITING UNTIL INSTRUCTED TO DO SO

(1) Principles and Process Scheduling

(a) Using examples, explain the distinction between policy and mechanism in operating system design.

[8 Marks]

(b) Identify and define performance metrics appropriate for batch computing systems.

[3 Marks]

(c) Identify and define performance metrics appropriate for interactive computing systems.

[3 Marks]

(d) In the CPU burst model of process execution, processes alternate between bursts of computation and waiting for IO to complete. You are to trace the operation of a pre-emptive process scheduling algorithm, with a quantum of 10 milliseconds. Pre-emption causes the scheduler to make a scheduling decision and can occur both on expiration of the time quantum, and whenever IO completes. Note that a time quantum expires every 10ms, not 10ms since the last scheduling decision, so the completion of IO can cause a CPU bound process to receive less than 10ms of CPU time. Each time it makes a scheduling decision, the scheduler selects the process having the shortest time remaining in its current CPU burst as the process to be run.

There are three processes (P1, P2 and P3), each of which is ready to run at time zero. The processes alternate between computation and waiting for IO as follows:

P1: 32ms CPU; 11ms IO; 10ms CPU; 15ms IO; 12ms CPU

P2: 8ms CPU; 20ms IO; 6ms CPU; 20ms IO; 14ms CPU

P3: 16ms CPU; 100ms IO; 2 ms CPU; 100ms IO

Using a diagram, show when each process is running on the CPU, including any time when the CPU is idle. In addition, show when each process is waiting for IO. You may assume that the system has sufficient IO capacity that all three processes can have IO operations in progress at the same time.

[12 Marks]

[Total Marks for Question 1: 26]

(2) Filesystems, Disks and IO

(a) You are to evaluate the effectiveness of a disk scheduling algorithm called SSTF-R. This is a modified version of SSTF (shortest seek time first).

The base cost of a seek (moving the head) from track C to track T is given by the absolute value of the difference between the current track C and the target track T. However, at all times, the head is moving in the direction implied by the last operation, and if the head must reverse direction, the cost of the seek is increased by five units. You may assume that the head never reaches the inner or outer track, and hence ignore the cost of reversing the head movement in such cases. Given this, you can evaluate the cost of servicing each of the remaining operations in the disk scheduling queue.

The SSTF-R algorithm operates incrementally, and chooses the operation for which the cost described above is minimal, services that operation, then repeats this process for the remaining operations.

The head starts at track 11, moving outwards (towards higher numbered tracks)

Disk operations are required on the following tracks:

13, 35, 40, 10, 9, 2, 28, 27

Show the order in which tracks are accessed using SSTF-R, the cost associated with each access, and the total cost of the accesses.

[6 Marks]

(b) You are to evaluate a modified disk scheduling algorithm called SSTF-2-R. This has all the properties of SSTF-R as described above, except that, when there are three or more operations remaining, SSTF-2-R finds the set of two remaining operations for which the sum of the cost of seeks is minimised, then selects the lower cost of the two as the operation to perform. Having performed this operation, it repeats the entire process for the remaining operations (i.e at each step, it finds a set of two operations, but only actually performs one operation). Once there are two or less operations remaining, SSTF-2-R behaves the same as SSTF-R.

Given this, and the specific example described in (a), show the two-operation subsets selected by SSTF-2-R at each step, the order in which tracks are accessed, the cost associated with each access, and the total cost of the accesses.

[6 Marks]

(c) Traditional Unix filesystems divide disk information into data blocks (containing data) and inodes (containing file meta-data and index pointers enabling the location of the data blocks of the file). It is suggested that support be added to store data directly in inodes (replacing the index pointers), in addition to the existing approach (i.e. not replacing it). Write an evaluation of this suggestion.

[8 Marks]

[Total Marks for Question 2: 20]

(3) Memory Management

(a) You are given the following sequence of page accesses:

12, 14, 9, 11, 12, 9, 10, 11, 3, 14

For a machine with four physical memory frames available, evaluate four page replacement policies: FIFO, LRU, Optimal and Random. The operation of Random is that it selects the frame into which the newly required page is to be loaded according the following sequence:

0, 3, 2, 0, 2, 1, 3, 1, 1, 0

One element of this sequence is consumed each time a page is loaded. For all four policies, the four frames are initially empty.

Trace through the execution of all four policies, showing when page faults occur, where pages are loaded and the total count of page faults in each case.

[8 Marks]

(b) Modify the FIFO, LRU and Random policies to load the page adjacent to the required page, as well as the page itself. Pairs of adjacent pages are those for which integer division (which rounds down) by two gives the same result. So for example:

pages 2 and 3 are adjacent

pages 10 and 11 are adjacent

pages 9 and 10 are NOT adjacent

A consequence of the adjacent page modification is that Random will consume two frames from the random sequence each time a page fault occurs, in order to select the two frames into which to load two new pages.

Trace through the execution of the modified LRU, FIFO and Random policies, showing when page faults occur, where pages are loaded and the total count of page faults in each case.

[6 Marks]

(c) What property of memory access is the adjacent page modification in (b) attempting to exploit?

[2 Marks]

(d) We excluded Optimal from evaluation with the adjacent page modification in (b). Is it meaningful to re-evaluate Optimal in this context?

[4 Marks]

[Total Marks for Question 3: 20]

(4) Synchronisation

(a) Define the terms *race condition* and *critical section*?

[3 marks]

(b) Consider the following naïve implementation of two processes, *P0* and *P1*:

```
P_i:
do {
        while (turn != i) {}
        // critical section
        turn = j; // (where j == 1 - i)
        // remainder section
} while(true);
```

This implementation successfully ensures the critical-section problem requirement of *mutual exclusion*. Hence it will never be the case the the critical sections of *P0* and *P1* are executing simultaneously. However this implementation fails two other requirements of the critical-section problem. Describe these requirements, and how the above solution fails to meet them.

[5 Marks]

(c) Ignoring the unmet requirements you identified in (b) – what feature inherent to the implementation in (b) makes it inefficient when run on a single processor machine?

[2 Marks]

(d) Show how the above mutually exclusion implementation may be implemented using a hardware “test and set” atomic instruction. Under which circumstances will your solution still be correct in the case that *P0* and *P1* are run on different processors?

[5 Marks]

(e) Explain how counting semaphores may be used to provide process synchronisation while avoiding busy waiting on a spin lock similar to that in the implementation shown above. In general, are spin locks avoided entirely?

[2 Marks]

(f) In which circumstances are spin locks preferred to queue based implementaions of synchronisation primitives?

[2 Marks]

(g) Show how a counting semaphore may be employed to implement barrier synchronisation to enable two processes to synchronise. Extend your answer to the case of *n* processes.

[4 Marks]

(h) Consider two processes, a producer:

```
producer:
while(true) {
        if (top < N) {
            v = produce();
            store[top] = v;
            top = top + 1;
        }
}
```

and a consumer:

```
consumer:
while(true) {
        if (top > 0) {
            v = store[top-1];
            top = top - 1;
            consume(v)
        }
}
```

Run in parallel, these processes will exhibit both busy waiting and an unchecked race condition. Using multiple semaphores, rewrite the above processes eliminating all race conditions and busy waiting.

[6 Marks]

[Total Marks for Question 4: 29]

(5) Deadlock

(a) Deadlock can occur if all four of the following conditions hold:

- mutual exclusion,
- hold and wait,
- no preemption, and
- circular wait

Describe how each condition may be violated by an operating system (and hence ensure deadlock is avoided) and the ramifications of doing so.

[8 Marks]

(b) Many operating systems choose to do nothing at all about the problem of deadlock in the system. Describe two different strategies which may be adopted by an operating system to deal with deadlock.

[6 Marks]

[Total Marks for Question 5: 14]

(6) Security and Distributed Systems

(a) How can public key cryptography be used to authenticate users in a distributed system?

[2 Marks]

(b) Describe some strategies by which the threat of buffer overflow exploitations by external malicious entities may be reduced or eliminated.

[3 Marks]

(c) What issues must be considered when marshaling parameters in RPC?

[6 Marks]

[Total Marks for Question 6: 11]

End of Examination