FIT2100
Semester 2 2019
Revision Questions

October 21, 2019

Revision Status:

`$Id: FIT2100-Sample-Questions.tex, Version 2.0 2018/10/25 11:45 Jojo $`

`Updated by Daniel Kos, Oct 2019`

Contents

1 Introduction

The final examination will contain a mixture of multiple choice questions and multi-part short answer questions.

The following review questions are more open-ended in nature with no given choices. They are to encourage you to consider a range of possible angles when preparing for the exam.

2 Part I: Review Questions

Question 1

Describe the two general roles of an operating system, and elaborate why these roles are important.

Question 2

Describe the *three-state process model*, describe what transitions are valid between the three states, and describe an event that might cause such a transition.

Question 3

What is a *process*? What are the attributes of a process?

Question 4

What is the function of the *ready queue*?

Question 5

What is the relationship (or differences) between threads and processes?

Question 6

Name the advantages and disadvantages of *user-level* and *kernel-level* threads.

Question 7

Describe the *process control block* and the details of information it maintains?

Question 8

What is a *race condition*? Give an example.

Question 9

What is a *critical section*? How do they relate to controlling access to shared resources?

Question 10

What is *deadlock*? What is *starvation*? How do they differ from each other?

Question 11

What are the four conditions required for deadlock to occur?

Question 12

Describe general strategies for dealing with deadlocks.

Question 13

For single unit resources, we can model resource allocation and requests as a *directed graph* connecting processes and resources. Given such a graph, what is involved in deadlock detection?

Question 14

Assuming the operating system detects the system is deadlocked, what can the operating system do to recover from deadlock?

Question 15

What must the *banker's algorithm* know a priori in order to prevent deadlock?

Question 16

Describe the general strategy behind *deadlock prevention*, and give an example of a practical deadlock prevention method.

Question 17

Explain what is meant by *short term* and *medium term* scheduling policy.

Question 18

Define *turnaround time* and *normalised turnaround time*. Why are these useful for measuring the performance of a scheduling algorithm? (Note: normalised t.t. not examinable)

Question 19

Describe the difference between *external* and *internal* fragmentation. Indicate which of the two are most likely to be an issue on: (a) a simple memory management machine using static partitioning; and (b) a similar machine using dynamic partitioning.

Question 20

What is *thrashing*? Provide an example of a situation in which thrashing could occur.

Question 21

Enumerate some pros and cons for increasing the page size with regard to memory management.

Question 22

Describe two virtual memory *page replacement* algorithms. Which is less common in practice? Why?

Question 23

What is the maximum file size supported by an inode file system with 16 direct blocks, 1 single, 1 double, and 1 triple indirection block? The block size is 512 bytes. Disk block numbers can be stored in 4 bytes.

Question 24

Explain the following basic algorithms that are used for the selection of a page as replacement algorithms: clock policy, first-in-first-out (FIFO), least recently used (LRU), and optimal policy? (Note: clock policy is not examinable.)

3 Part II: Problem-Solving Questions

3.1 Task 1

Consider the following table, which shows when each of the processes arrives to the system and the CPU time (in seconds) required for its execution. Assume that no I/O operations are involved in these processes.

Process	Arrival Time	Service Time
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2

Draw a chart (or sequence) of process execution under the following process scheduling:

(a) First-Come-First-Served (FCFS) or First-In-First-Out (FIFO)

(b) Round Robin with the CPU time slice quantum of 1 second

(c) Shortest Process Next (SPN)

(d) Shortest Remaining Time (SRT)

3.2 Task 2

Consider the following snapshot of a system:

R1	R2	R3	R4
3	4	5	5

Table 1: Available resources

Process	R1	R2	R3	R4
P1	1	1	2	2
P2	1	0	0	0
P3	0	0	1	2

Table 2: Current allocation matrix

Process	R1	R2	R3	R4
P1	2	4	5	2
P2	1	1	3	5
P3	1	0	2	2

Table 3: Maximum request matrix

(a) Prove that the system is in a *safe state* by showing a sequence of process completion even when all the processes eventually request for their maximum requirements.

(b) If a request from process `P2` arrives for `(0, 1, 0, 0)`, can the request be granted immediately? Justify your answer.