National Examinations December 2010

98-Comp-A5, Operating Systems

3 Hours Duration

NOTES:

1. If doubts exist as to the interpretation of any question, the candidate is urged to submit with the answer paper a clear statement of any assumption made.

2. Provide justifications for your answers. Show all your work.

3. CLOSED BOOK. Candidates may use one of the two pocket calculators, the Casio approved model or Sharp approved model. No other aids.

4. The candidate has to answer **any five questions** (each question has multiple parts).

5. Total Marks = 100.

6. This exam has got 5 pages (including this page).

1 [20 marks]

(a). Consider the following arrivals on a system. Each process has a single CPU burst and does not perform any I/O.

Process	Arrival Time (seconds)	Execution Time (seconds)
Proc1	0	25
Proc2	3	7
Proc3	5	19
Proc4	11	2

What is the minimum mean process turnaround time that can be achieved when non-preemptive CPU scheduling is used?

(b) Why is the optimal CPU scheduling strategy difficult to implement in practice? Discuss with the help of an example, how a heuristic strategy based on multiple levels of queues can be used to approximate the behaviour of the optimal strategy.

(c). Explain why it is harder to perform real time scheduling on a multiple processor-based-system in comparison to a single CPU system.

2 [20 marks]

a). Discuss the optimal strategy for page replacement on a virtual memory system. Why is it difficult to implement it on a real system? Discuss with the help of an example, how a heuristic strategy that uses the past memory referencing behaviour of the program can be used to approximate the behaviour of the optimal strategy.

(b) Consider the following page reference string on a demand paged virtual memory system:
21, 22, 23, 24, 25, 23, 24, 21, 26, 27, 28, 27, 28, 29, 27, 28, 29

Determine the number of page faults that would occur with the FIFO page replacement algorithm when 2 frames are allocated to the program.

(c) Discuss why it is important for a virtual memory management strategy to be able to dynamically control the degree of multiprogramming on the system

3 [20 marks]

(a). Different methods exist for storing information on the disk. Consider a file currently consisting of 90 blocks. Assume that the directory is available in main memory.

(i) For each of the following cases (A-C) compute the minimum number of disk operations that are required when <u>contiguous allocation</u> is used. Assume that there is no room for the file to grow in the beginning but there is room to grow in the end.

(A) a block is removed from the beginning of the file.
(B) a block is added after the 80 th block in the file.
(C) The last block of the file is removed.

(ii) For each of the following cases (D-F) compute the minimum number of disk operations that are required when <u>linked allocation</u> (based on a singly linked list) is used.

(D) a block is added at the beginning of the file.
(E) a block is added after the 80 th block in the file.
(F) The last block in the file is removed.

Consider each case (A-F) separately. For (B), (D), and (E) assume that the information to be inserted into the file is available in main memory. Note that each disk operation corresponds to the reading of a block from the disk or the writing of a block to the disk. While computing the number of disk operations, ignore the disk operations that may be required for the location and maintenance of free space. Since the directory is in main memory any operation on the directory is not counted as a disk operation.

ASSUME: The length of the file is known to the system

(b). What is *fragmentation*? Discuss with the help of examples how fragmentation can occur on memory as well as on disks. Describe a method for controlling fragmentation on memory and a method for controlling fragmentation on disks.

4 [20 marks]

Consider a system in which multiple processes P1 .. Pn run concurrently. The system has got a plotter which can be directly used by the processes for plotting graphs. The plotter however can be used by a single process at a time. Access to the plotter is to be controlled with the help of a monitor. There are two procedures inside the monitor: *acquire* and *release*. There are three different sizes of graphs that can be plotted by a process on the plotter: *large*, *medium* and *small*. When a process wants to use the plotter it calls procedure acquire. The size of the data block ('S' for small, 'M' for medium and 'L' for large) is passed as an argument. If no other process is using the plotter the process gets control of the plotter; otherwise it is blocked.

After acquiring the plotter the process comes out of the monitor and uses the plotter. When the process has completed using the plotter it calls procedure release. The size of the graph that was plotted is passed as an argument to release. If a single process is waiting for the plotter when the plotter is freed, this process is allocated the plotter and is allowed to proceed. If multiple processes are waiting then the process that will be allocated the plotter is based on the number of processes waiting to plot graphs of a specific size. The number of processes waiting to plot large graphs (Nl), the number of processes waiting to plot medium graphs (Nm) and the number of processes waiting to plot small graphs (Ns) are compared. The plotter is allocated to a process waiting to plot a large graph if (Nl > Nm > Ns), to a process waiting to plot a medium graph if Nm ≥ Nl and Nm > Ns ; otherwise the plotter is allocated to a process waiting to plot a small graph.

The typical operations performed by a process Pi (i = 1 .. n) is given by the following algorithm.

Process Pi

repeat

1. Perform computation.

2. Call procedure acquire in the monitor. [Indicate the size of the graph to be plotted through the argument of the procedure.]

{process not blocked inside the monitor means that it can use the plotter.}

3. Use the plotter.

4. Call procedure release in the monitor. [Indicate the size of the graph plotted through the argument of the procedure.]

until false

Your job is to write the algorithm (pseudo code) for the monitor that will control access to the plotter. The monitor must contain the two procedures acquire and release (described above) that are called by the processes. [You may incorporate other procedures and functions in the monitor if you wish.]

Provide necessary documentation to explain your solution.

Declare the variables required by your solution in the variable declaration part of the monitor. However this variable declaration CAN NOT include a variable of type semaphore.

5 [20 marks]

(a). What is starvation? Discuss with the help of examples how starvation can occur in each of the following situations. Include a discussion of how it can be controlled in each case.

(i) when multiple processes in a single application are competing to enter the critical section

(ii) when multiple jobs are competing for the CPU on a multiprogrammed system

(iii) when multiple requests are competing for the disk in a multiprogrammed system

[Be as specific as you can when you describe the situations in which starvation occurs]

(b). Discus the role of paging in controlling fragmentation.

(c). Page size has an important impact on the performance of a page-based memory management system. Discuss the pros and cons of using a large page size.

6 [20 marks]

(a). Briefly discuss the roles of security and protection in the context of a multi-user computing system. Discuss the need for security and protection and how they can be achieved in the context of two important resources in a computing system: (i) the main memory and (ii) file system

(b). Discuss the advantages and the overheads associated with the use of (i) multiple CPUs and (ii) multiple disks on a computing system

(c). Consider the design of a parallel application. When will you use multiple processes and when will you use multiple threads in implementing the system? Justify your answer.

7 [**20 marks**]

(a). Consider a multiprogrammed system consisting of 10 resources of the same type. No deadlock handling technique is employed by the system. That is, if a resource is requested by a process and one is available, a resource is allocated to the requesting process; otherwise the requesting process is blocked. P processes are run concurrently on the system. Each process can simultaneously hold up to two resources at any given point in time.

Once a resource is acquired by a process, it must be released by the process before it can be assigned to another process. Assume that each process requests and releases one resource at a time.

(i) For P = 2, can a deadlock occur on the system?
(ii) For P = 15 can a deadlock occur on the system?
(iii) Determine the maximum value of P such that a deadlock can never occur on the system.

(b). Consider a moving head hard disk which consists of a single platter (surface) with 100 tracks on it. The tracks are numbered 0 to 99. The disk is currently serving a request at track 53. The queue of pending requests in FIFO order is:

37, 65, 92.

(i) What is the total head movement (in number of tracks) needed to satisfy all these requests when an SSTF algorithm is used for disk scheduling?

(ii) In what order should the requests be served to minimize the total head movement?

[Assume that no further requests arrive on the system during the service of the above requests.]

(c) Briefly explain what is meant by execution time address binding in the context of memory management.

Approximate Marking Scheme

1 (a) 8 marks
1 (b) 8 marks
1 (c) 4 marks

2 (a) 9 marks
2 (b) 7 marks
2 (c) 4 marks

3 (a) 12 marks
3 (b) 8 marks

4 20 marks

5 (a) 12 marks
5 (b) 4 marks
5 (c) 4 marks

6 (a) 10 marks
6 (b) 6 marks
6 (c) 4 marks

7 (a) 10 marks
7 (b) 8 marks
7 (b) 2 marks