NATIONAL EXAMINATIONS December 2014

98-COMP A-5 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 a priority-based CPU scheduling system in which the priority of a process is set as soon as the process arrives on the system. The assigned priority is an integer value and a lower value implies a higher priority. How should the priority value be set in order to achieve the following scheduling policies?

(i) First Come First Served (FCFS) and (ii) Shortest Job First

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

Process	Arrival Time (seconds)	Execution Time(seconds)
Proc1	0	21
Proc2	3	8
Proc3	4	14
Proc4	7	2
Proc 5	8	2

Compute the fairness index (F) and the mean job turnaround times for the following CPU scheduling policies. NOTE that the fairness index for a scheduling policy is defined as the ratio between the mean process waiting time and the mean process execution time.

(i) FCFS
(ii)the Highest Response Ratio Next (HRRN). HRRN is a non-preemptive policy. Under this policy, whenever the CPU is free the process with the highest response ratio is run. The response ratio for a process is defined as:

$$\text{response ratio} = \frac{\text{(waiting time for process + process execution time)}}{\text{process execution time}}$$

(c) Explain with the help of examples how the execution of a lower priority process can proceed when there is a higher priority process waiting on a real time system.

2 [20 marks].

(a) Consider the following sequence of memory addresses generated during a program execution:

100, 1001, 1250, 8800, 9990, 2780

Derive the page reference string for a system with a page size of 200.

(b) Consider a demand paged virtual memory system and the following page reference string:

71, 102, 103, 104, 105, 103, 104, 101, 106, 107, 108, 107, 108, 108, 107, 108, 101

Determine the number of pages faults for the Least Frequently Used page replacement policy when 4 frames are allocated to the program

(c) Consider a priority-based page replacement strategy for a demand paged virtual memory system on which the priority of a page has a numeric value and can be changed dynamically with time. Discuss with the help of examples how the priority of a page is to be set for achieving the following page replacement policies:

(i) First in First Out (ii) Least Recently Used (iii) Most Frequently Used (iv) the optimal page replacement policy

3 [20 marks].

(a) Distinguish between security and protection. Discuss the role of access control in achieving protection on a file system that is used by multiple users.

(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 61 and has just finished a request at track 40. The queue of pending requests in FIFO order is:

98, 58, 11, 87, 99, 62, 18, 40.

What is the total head movement (in number of tracks) needed to satisfy all these requests for the following disk scheduling algorithms?

(i) SSTF (ii) SCAN

[Note that no other requests arrive on the system during the service of the above requests.]

(c) Briefly discuss two different ways of using multiple disks to handle disk failures on a system.

4 [20 marks].

(a) Consider a multiprogrammed system consisting of seven 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. Three processes are run concurrently on the system. Each process can simultaneously hold up to three 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.

Can a deadlock occur on the system? Justify your answer.

(b) Briefly distinguish between deadlock and starvation.

(c) Briefly distinguish between sequential and random access in the context of disk systems.

(d) Can a deadlock occur from file sharing on a multiprogrammed system in which there are only read only files?

5 [20 marks].

(a) Given below is a solution to the critical section problem involving two concurrent processes Proc1 and Proc2. Identify as many **distinct problems** as you can in the design. If similar problems occur at multiple places identify them each time but explain it only once. Your list of errors should include defects (if any) that may not necessarily give rise to incorrect results but do indicate flaws in design. Justify your answer with the help of examples. Be as specific as you can when you describe the situations in which problems occur.

Algorithm

Proc 1

```
do{
        get_entry[0] = true;
        if get_entry[1] {
                get_entry[0] = false;
                while get_entry[1]  no-op;
                get_entry[0] = true;
        };

        Code for CS

        get_entry[0] = false;
    } while (TRUE) ;
```

Proc 2

```
do{
        get_entry[1]= true;
        if get_entry[0] {
                get_entry[1] = false;
                while get_entry[0]  no-op;
                 get_entry[1]= true;
        };

        Code for CS

        get_entry[1] = false;
    } while (TRUE);
```

Note: CS: Critical Section. no-op: no operation.

(b) With the help of examples discuss how mutual execution and synchronization for concurrent processes is achieved with a monitor.

(c) Discuss with the help of an example how process starvation is prevented by a semaphore that is guarding access to a critical section for concurrent processes.

6 [20 marks].

(a) Consider a demand paged virtual memory system in which a single program is currently running. The page map table is held in associative registers (associative memory). It takes S1 milliseconds to service a page fault if an empty frame is available or the replaced page is not modified, and S2 milliseconds if the replaced page is modified. Memory access time is A nanoseconds.

Assume that for F% of the page faults a page replacement is necessary and the page to be replaced is modified. What is the maximum acceptable page fault rate such that the effective memory access time for the program is not greater than 1.3A nanoseconds?

(b) What is virtual memory? Outline two advantages of using virtual memory.

(c) Discuss the role of "locality" of programs in the context of virtual memory management. Identify two virtual memory management techniques that exploit the locality of programs. Discuss how system performance is enhanced by these strategies by such exploitation of program locality.

7 [20 marks]

(a) Discuss the role of free space management on disks in supporting a file system. Include the description of any one free space management technique in your discussion. Why is it important that information regarding free space be saved on the disk?

(b) Discuss why it is hard to implement optimal strategies for managing resources such as CPU and memory on a real system.

(c) What is a shared file? How are shared files supported on a multi-user system? Discuss the problems that may exist with the deletion of shared files

(d) Consider a telephone switch and a process control system. On the telephone switch call switching must be performed within 500 ms for at least 98% of the calls. In the process control system a furnace needs to be switched off within 150 ms of observing that its temperature has gone over 370^0 C.

For each system discuss whether it should be considered as a hard real time or a soft real time system.