

NATIONAL EXAMINATIONS **May 2002**

98-SOFT-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 five questions (each question has multiple parts).
- Answer ANY ONE of questions 1 and 2, ANY ONE of questions 3 and 4, ANY THREE of questions 5, 6, 7, and 8.
5. Total Marks = 100.
 6. This exam has got 6 pages (including this page).

Answer question 1 OR question 2

1 [20 marks].

(a). Consider the following arrivals on a system.

<u>Process</u>	<u>Arrival Time (seconds)</u>	<u>Execution Time(seconds)</u>
P1	0	3
P2	2	6
P3	4	4
P4	6	5
P5	8	2

Find the mean process turn around time in each of the following cases.

- (a) When the FCFS policy is used for CPU scheduling
- (b) When the optimal CPU scheduling policy is used for CPU scheduling
- (c) When the Highest Response Ratio Next (HRRN) policy is used for CPU scheduling [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:

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

(b). What is a real time system? Explain with the help of an example how "priority inversion" can take place in a real time system [be brief].

2 [20 marks].

(a). Consider an operating system which uses a Process Table for short term scheduling (CPU scheduling). The Process Table contains a set of records each of which represents a Process Control Block (PCB). The PCB contains the current state of the process (ready, blocked or running). Another record in the PCB is tickcount that is incremented periodically by the operating system. The initial value of tickcount is set to 0 when the job first enters the multiprogramming mix.

The short term scheduler used by this operating system is fairly simple: it performs a new scheduling decision only when the running process blocks for I/O or it completes. The short term scheduler scans the Process Table starting from the top and examines each PCB. The ready process that has the smallest tickcount is selected to run on the CPU.

Identify as many defects as you can in the design of the short term scheduler.

(b). In this question you are to consider *non-preemptive* scheduling policies only.

A scheduling policy is said to be work-conserving if the CPU is never kept idle when the ready to run queue is non-empty. Shortest Job First (SJF) is optimal among work-

98-SOFT-A5 OPERATING SYSTEMS

conserving policies.

Can a *non-work-conserving* policy produce a smaller mean turn around time on a system in comparison to SJF?

If your answer is NO then provide arguments for proving that a non-work-conserving policy can never produce a better performance than SJF.

If your answer is YES then provide an example that demonstrates that a mean job turn around time smaller than SJF can actually be produced by a non-work-conserving scheduling policy.

Answer question 3 OR question 4

3 [20 marks]

(a). What is thrashing ? Describe briefly a memory management strategy that can prevent thrashing and maintain a good degree of multiprogramming on a virtual memory system.

(b). 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 10 milliseconds to service a page fault if an empty frame is available or the replaced page is not modified, and 20 milliseconds if the replaced page is modified. Memory access time is 100 nanoseconds.

Assume that for 60% 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 200 nanoseconds?

4 [20 marks].

(a). Consider the following page reference string on a demand paged virtual memory system:

11,12,13,14,12,15,16,12,11,12,13,17,16,13,12,11,12,13,16

Determine the number of page faults that would occur with the following page replacement algorithms when 4 frames are allocated to the program. [Remember that the frames are initially empty and the first reference to a page will cause a page fault].

- (i) First In First Out (FIFO)
- (ii) Optimal replacement

(b). Consider a multiprogrammed system that uses multiple partitions (of variable size) for memory management. A linked list of holes called the free list is maintained by the operating system to keep track of the available memory in the system. At a given point in time the free list consists of holes with sizes:

10K, 4K, 20K, 18K, 7K, 9K, 12K, and 15K

The free list is also ordered in the sequence given above: the first hole in the list is of size 10K words which is followed by a hole of size 4K words and so on. Jobs with different memory requirements arrive on the system in the following order:

98-SOFT-A5 OPERATING SYSTEMS

	<u>Arrival Time</u>	<u>Memory Requirement</u>
Job 1	t1	12K
Job 2	t2	10K
Job 3	t3	9K
Job 4	t4	20K

[Given $t_1 < t_2 < t_3 < t_4$]

Explain how memory allocation would be performed in the given situation for (i) the best fit and (ii) the worst fit policy. [For each policy determine which hole is allocated to each job after it arrives on the system]

Answer any THREE of questions 5, 6, 7, and 8.

5 [20 marks]

Consider a system in which multiple processes $P_1 .. P_n$ 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: grab_plotter and free_plotter. There are two different sizes of graphs that can be plotted: large and small. When a process wants to acquire the plotter it calls procedure grab_plotter. The size of the graph ('S' for small 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 use of the plotter it calls procedure free_plotter. The size of the graph that was plotted is passed as an argument to free_plotter. 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 selected in the following way. If the number of processes waiting for plotting large graphs is 2 times or more than the number of processes waiting to plot small graphs then a process that is waiting to plot the large graph is selected; otherwise the plotter is allocated to a process waiting for plotting a small graph.

The typical operation performed by a process P_i ($i = 1 .. n$) is given by the following algorithm.

Process P_i
repeat

1. Perform computation.
2. Call procedure grab_plotter in the monitor. [Indicate the size of the graph through the argument of the procedure.]

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

3. Use the plotter.
4. Call free_plotter in the monitor. [Indicate the size of the graph 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 `grab_plotter` and `free_plotter` (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.

6 [20 marks].

(a). Briefly discuss the following disk space allocation strategies: contiguous allocation, and linked allocation. Discuss also the merits and deficiencies (if any) of these strategies.

(b). Consider a moving head hard disk which consists of a single platter (surface) with 200 tracks on it. The tracks are numbered 0 to 199. The disk is currently serving a request at track 140 and has just finished a request at track 138. The queue of pending requests in FIFO order is:

86, 147, 91, 77, 94, 150, 102, 175, 130.

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

(i) First Come First Served (FCFS) (ii) SCAN

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

(c). What are inodes (that are used in the Unix operating system)?

7 [20 marks].

(a) Consider a multiprogrammed system with D devices of the same type. Processes P_1 , P_2 , and P_3 run concurrently on the system and use these devices. If any process requests for a device then any of the unused devices can be given to the process to satisfy the request.

Show that the system is deadlock free when the sum of the maximum needs (for these devices) of all these processes is less than $D+3$.

Note: "Maximum need of a process is M " means that a maximum of M devices can be held simultaneously by this process.

Once a device 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 device at a time and the maximum need for each process is between 1 and D .

[A system is said to be deadlock free if there are enough resources in the system and we do not need to employ any deadlock handling technique. If a device is free, it can be immediately allocated to a requesting process, yet no deadlock can occur.]

(b) Three general approaches exist for handling deadlocks on general purpose multiprogrammed systems: deadlock prevention, deadlock avoidance, and deadlock

detection with recovery.

Briefly discuss the deadlock prevention approach.

8 [20 marks].

(a) Explain how Redundant Arrays of Inexpensive Disks (RAID) can improve system performance and reliability.

(b) The binding of instructions and data to memory addresses can be done at different times. Differentiate between compile time and load time address binding. What is dynamic loading?

(c) Briefly describe the direct and sequential access techniques used in the storage and retrieval of data. Can these techniques be used with (i) a disk (ii) a tape?

(d) Discuss why symmetric multiprocessing is considered to be harder in comparison to asymmetric multiprocessing.

(e) Describe with the help of an example any one method for file protection employed by an operating system.