NATIONAL EXAMINATIONS May 2013

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 the following arrivals on a system.

| Process | Arrival Time (seconds) | Execution Time(seconds) |
|---|---|---|
| P1 | 0 | 3 |
| P2 | 12 | 6 |
| P3 | 14 | 4 |
| P4 | 16 | 5 |
| P5 | 18 | 2 |

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

(i) When the FCFS policy is used for CPU scheduling
(ii) When the optimal CPU scheduling policy is used for CPU scheduling
(iii) 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)** Using examples, distinguish between a hard and a soft real time system[be brief].

**2. [20 marks]**

Consider a system in which N processes P1 .. PN execute concurrently. The system contains a shared data record, which may be concurrently accessed by up to K processes. A monitor is used to control access to the shared data record. A requesting process is blocked if there are already K processes accessing the shared data record when the request is made. A blocked process is allowed to proceed when one of the processes exits the monitor. For example, consider a system with K=3. If P1, P3 and P5 are currently accessing the shared data record and P2 wants to concurrently access the shared data record, P2 is blocked. If P3 leaves the monitor after P2 is blocked, P2 is allowed to proceed after P3 has left the monitor.

The monitor contains two procedures: get_access and finish_access. Whenever a process wants to access the shared data record it calls the procedure get_access. If the number of processes accessing the shared data is less than K, it is allowed to proceed; it is blocked otherwise.

Whenever a process finishes its operation on the shared data record it calls the procedure finish_access. If one or more processes is/are blocked when finish_access is called, a blocked process is allowed to access the shared data record after the completion of the finish_access procedure.

The typical operations performed by a process Pj (j = 1 .. N) are given by the following algorithm.

Process Pj

```
do{
        1.    Perform computation.
        2.    Call procedure get_access in the monitor.
        {If the process is not blocked inside the monitor it means that the desired operation on the
        shared data record can be performed by the process.}
        3.    Perform the desired operation on the shared data record.
        4.    Call procedure finish_access in the monitor.

} while (TRUE)
```

(i) Write the algorithm (pseudo-code) for a monitor that will control access to the shared data record. The monitor must contain the two procedures get_access and finish_access (described above) that are called by the processes. You may incorporate other procedures/functions inside the monitor if necessary. You can pass any parameter to the procedures that you deem necessary.

(ii) Briefly outline the three requirements associated with the solution to the critical section problem. Discuss your algorithm in the context of these three requirements.

**3. [20 marks]**

(a) Different methods exist for storing information on the disk. Consider a file currently consisting of 120 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 contiguous allocation 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 90 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 linked allocation (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 90 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 an acyclic graph directory? Discuss how it can be used for file sharing. Include its merits as well as the overheads associated with file creating and deletion in your dicussion.

**4. [20 marks]**

(a) Consider the following page reference string on a demand paged virtual memory system:
51,52,53,54,55,53,54,51,56,57,58,57,58,59,57,58,59

(i) Determine the number of page faults that would occur with the LRU page replacement algorithm when 3 frames are allocated to the program.

(ii) What is the minimum number of page faults that can occur on the system when 3 frames are allocated to the program?

**(b).** Explain with the help of an example of how thrashing occurs on a virtual memory system. Discuss the mechanisms for detecting and controlling thrashing.

(c) What is external fragmentation? Briefly describe a method for controlling external fragmentation in the context of memory management.

**5. [20 marks]**

**(a)** Consider a **preemptive** short term scheduling strategy in which the priority of a process may change dynamically with time.(Larger priority numbers imply higher priority). At any point in time the highest priority process is run on the system. Ties are broken in favour of the process that entered the ready to run queue (ready queue) first. When a process is preempted it joins the ready to run queue.

The priority of a process waiting in the ready to run queue is computed as a function of **t**, **a** and **R**:

Priority of a process in the ready to run queue = $R + t^a$

where, t is the time elapsed after the process arrived at the system.

When a process is selected to run on the CPU its priority is computed as a function of **t**, **b** and **C**:

Priority of the process running on the CPU = $C\,t^b$

where, t is the time elapsed after the process arrived at the system.

The parameters R, C, a, and b can be set to give many different scheduling policies.

Determine the values of the parameters R, C, a, and b that are to be used for obtaining the Last Come First Served policy. Under this policy whenever a process arrives on the system it preemptively captures the CPU. That is, if the CPU is free the process is allocated the CPU; if the CPU is busy then the executing process is preempted and the CPU is allocated to the process that just entered the ready to run queue. Whenever a process completes its

CPU burst and the CPU is free, the process (in the ready to run queue) that was preempted most recently is allocated the CPU. **Justify your answer.**

**NOTE:** -- Assume that all processes have only a single CPU burst and they do not perform any I/O.
-- R, C, a, and b are constants and can not change with time.

**(b)** Using an example explain the optimal strategy for page replacement in a virtual memory system. Discuss why it is hard to implement the strategy on a real system.

**6. [20 marks]**

**(a)** Briefly discuss the motivations for file protection. Using an example, discuss the file protection strategy based on access lists. Include the merits and deficiencies of the strategy in your discussion.

**(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 151 and has just finished a request at track 140. The queue of pending requests in FIFO order is:

*96, 157, 101, 187, 104, 160, 112, 185, 140.*

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

(i) SCAN (ii) SSTF

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

**(c)** Briefly discuss a method for handling disk failures on the system. Include the merits and overheads associated with the technique.

**7. [20 marks]**

**(a)** Briefly discuss the roles of strategies used for free disk space management and for allocation of disk space to files in a computer system. Discuss any one free space management strategy OR any one allocation strategy.

**(b)** Compare the complexities of performing job scheduling on a multiprocessor system with that on a system deploying a single CPU.

**(c)** Discuss the difference between the following approaches to deadlock handling: *deadlock avoidance and* deadlock prevention. Include the merits and short comings (if any) of each of these techniques in your discussion.

**(d)** Briefly discuss the two components of the deadlock detection and recovery approach used for handling deadlocks.