CS-228
OPERATING SYSTEMS

Attempt 2 questions out of 3

Question 1

(a) With the aid of a state transition diagram, illustrate the transitions between the process states **running**, **ready**, **blocked**, **suspended-ready**, and **suspended-blocked**. The events causing the transitions should be clearly indicated.

Consider each of the following suggested transitions. If it is included in your state transition diagram, give an example event that would cause such a transition; otherwise, explain why such a transition is not possible.

(i) From “running” to “ready”;
(ii) from “blocked” to “suspended blocked”;
(iii) from “blocked” to “running”;
(iv) from “suspended-ready” to “blocked”;
(v) from “suspended-blocked” to “suspended-ready”.

[10 marks]

(b) With demand paging memory management and the **least recently used (LRU) page replacement strategy**, the current LRU matrix (6x6) is shown below, assuming that the memory has only six page frames. If the next three page reference numbers are 2, 4, and 6 respectively, illustrate the changes to the LRU matrix step by step.

$$\begin{array}{c|c|c|c|c|c|c|}
 & \mathbf{5} & \mathbf{3} & \mathbf{2} & \mathbf{7} & \mathbf{9} & \mathbf{8} \\ \hline
\mathbf{5} & 0 & 0 & 1 & 0 & 0 & 0 \\ \hline
\mathbf{3} & 1 & 0 & 1 & 0 & 1 & 0 \\ \hline
\mathbf{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline
\mathbf{7} & 1 & 1 & 1 & 0 & 1 & 0 \\ \hline
\mathbf{9} & 1 & 0 & 1 & 0 & 0 & 0 \\ \hline
\mathbf{8} & 1 & 1 & 1 & 1 & 1 & 0 \\ \hline
\end{array}$$

Does the LRU page replacement strategy suffer from Belady’s anomaly? Explain Belady’s anomaly and your answer.

[6 marks]

(c) (i) In the context of security threats, briefly explain the difference between a Trojan Horse and a Worm.
(ii) Describe the methods used by the Morris Internet Worm to identify potential target hosts for attack.
(iii) Describe the methods used by the Morris Internet Worm to attack target hosts once they had been identified.

[9 marks]

Question 2

(a) Consider the set of processes shown in the table below. We assume that the CPU ready queue is empty at time 0, and that the time needed for context switches is negligible.

	P_1	P_2	P_3	P_4
Arrival time (ms)	0	1	3	6
Burst time (ms)	5	5	2	3

With the aid of a Gantt chart or an equivalent, determine the average waiting time (over all four processes) for each of the following scheduling algorithms:

(i) non-preemptive FIFO (first in first out) scheduling;
(ii) non-preemptive SJF (shortest job first) scheduling;
(iii) preemptive RR (round-robin) scheduling (with quantum = 3ms).

What is the effect of increasing the time quantum to an arbitrarily large number for the round-robin scheduling?

[7 marks]

(b) In the context of demand paging, consider the following code for initialising an array with 10 rows and 32 columns. Suppose that:

- The rows are stored in contiguous blocks of virtual memory and the elements in each row are stored in contiguous words;
- the memory consists of page frames, each holding a row of 32 elements;
- a maximum of 9 page frames may be allocated to the array;
- initially no part of the array is in memory.

```
for col := 1 to 32 do
    for row := 1 to 10 do
        A[row,col] := 0;
```

(i) How many page faults would running the above code cause if the first-in-first-out (FIFO) page replacement algorithm was used?
(ii) How could you reduce page faults by modifying the program? How many page faults would the new code cause if the FIFO strategy is used?
(iii) Describe a page replacement strategy that would reduce page faults in this particular case (without modifying the original program).

[6 marks]

(c) For each of the following pairs of approaches to the design of file systems, discuss their relative merits from the viewpoint of users as well as that of the operating system designers:

(i) limited or arbitrary (> 0) length permitted for file names;
(ii) restricted or arbitrary character set permitted for file names;
(iii) tree structure or general graph structure for directories.

[6 marks]

(Question 2 continued overleaf.)

(d) (i) Describe in detail how events may be ordered in a distributed environment without the benefit of a common clock.

(ii) Consider three processes running concurrently on different computers, each with its own clock. During the execution, these processes exchange messages frequently, and generate reports regularly (e.g. every hour) which must be displayed on the screen at a similar time. However, as shown below, the speeds (in seconds) of the three clocks are quite inconsistent with each other. Using this example, explain how the 'time stamping' concept can be used to synchronise the activities of these three processes.

C1	**C2**	**C3**
0	0	0
1	4	5
2	8	10
3	12	15
4	16	20
...	...	...
16	64	80
17	68	85
18	72	90
...	...	...

[6 marks]

Question 3

(a) In the context of Dijkstra's deadlock avoidance algorithm (banker's algorithm) for single-type resource management, state whether each of the following states is safe or unsafe, briefly justifying your answer in each case.

We assume that there are **8** resources in total and only **2** are free at the moment. In the tables, for each process, *loan* is the number of resources currently held by it, *max* is the maximum number of resources needed, and *claim* is the number of resources to be claimed.

STATE I	*loan*	*max*	*claim*
P_1	3	6	3
P_2	2	7	5
P_3	1	3	2
P_4	0	7	7

STATE II	*loan*	*max*	*claim*
P_1	2	6	4
P_2	4	7	3
P_3	0	5	5
P_4	0	4	4

STATE III	*loan*	*max*	*claim*
P_1	0	6	6
P_2	2	3	1
P_3	1	3	2
P_4	2	8	6

STATE IV	*loan*	*max*	*claim*
P_1	2	6	4
P_2	1	7	6
P_3	3	5	2
P_4	0	4	4

[4 marks]

(b) In the context of memory management, what is meant by the following?

- (i) Fragmentation;
- (ii) page fault;
- (iii) anticipatory paging;
- (iv) storage hierarchy.

[8 marks]

(c) (i) A file containing 1500Kb of data is to be stored in a Unix file system with 1Kb data blocks. Each indirect block can hold 256 disk addresses (i.e. 32-bits per address). What is the total number of data blocks required to store this file (including all indirect blocks but not the inode itself)? Explain your answer in detail.

(ii) Describe, with the aid of a diagram if necessary, the concepts of Networked File System (NFS, designed by Sun Microsystems).

[7 marks]

(d) Consider a **variation of round-robin scheduling** in which a process that has used x% of its quantum is returned to the ready queue at a place x% of the distance away from the beginning of the queue. For example, a process that has used a full quantum is returned to the end of the ready queue, and one that has used half a quantum is returned to the middle of the queue. Discuss the merits and demerits of this algorithm in comparison with both **simple round-robin** and **multi-level feedback queues** algorithms. (*Hint: consider issues such as I/O bound processes, CPU-bound processes, fairness, waiting time, queue management, newly-arrived processes and quantum length.)*

[6 marks]