CS_228
OPERATING SYSTEMS
Attempt 2 questions out of 3

Question 1

(a) 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]

(b) In the context of demand paging, consider the following code for initialising an array with 5 rows and 5 columns, with the following suppositions:

- 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 5 elements;
- a maximum of 4 page frames may be allocated to the array;
- initially no part of the array is in memory.

```
for col := 1 to 5 do
    for row := 1 to 5 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) (i) From an operating system perspective, compare and contrast the following methods for destroying a Unix process:

    (A) Typing `Ctrl-C` at the terminal from which the process was activated.
    (B) Finding out the process identifer `pid` of the process, and then issuing the following command from another terminal: “`kill -9 <pid>`”

  (ii) Describe how a process may destroy another process using a system call, and what access permission is normally required.

[5 marks]

(d) 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. A preemptive round robin algorithm is used to schedule these processes, and the quantum is set to 2ms.

	P_1	P_2	P_3	P_4
Arrival time (ms)	0	1	3	7
Burst time (ms)	6	6	2	2

(i) With the aid of a Gantt chart or an equivalent, determine the average waiting time (over all four processes).

(ii) Consider a simplified multi-level feedback queue scheduling algorithm with only **two levels**. The quantum for processes in the first level is set to 2ms, and that for the second level is 4ms. Determine the average waiting time (over all four processes) with the same set of processes as above.

[6 marks]

Question 2

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

[10 marks]

(b) In the context of Dijkstra’s deadlock avoidance algorithm (banker’s algorithm) for single-type resource management, discuss whether each of the following states is safe or unsafe.

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	1	6	5
P_2	4	7	3
P_3	0	3	3
P_4	1	4	3

STATE II	*loan*	*max*	*claim*
P_1	3	6	3
P_2	1	7	6
P_3	1	3	2
P_4	1	4	3

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

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

[4 marks]

(c) Suppose you are managing a large multi-user system with a multi-level feedback queues scheme and a demand paging scheme. Suppose you have received complaints from many users about poor response and turnaround times, while a system operator informs you that the CPU is in use only 10% of the time, and that I/O devices (including the driver of the secondary storage) are active 99% of the time. Which of the following **could** be reasons for the poor responses? Explain your answers.

(i) The quantum used in the scheduling scheme is too short.
(ii) The quantum used in the scheduling scheme is too long.
(iii) The CPU is too slow.
(iv) There are too many small I/O bound processes.
(v) There are too many small CPU bound processes.
(vi) There are too many large I/O bound processes.
(vii) There are too many large CPU bound processes.
(viii) The page size is too small.
(ix) The page size is too large.

[3 marks]

(Question 2 continued overleaf.)

(d) Consider the *bounded buffer producer/consumer* problem: there is a buffer shared among a number of processes, some of which (producers) only add items to the buffer and the others (consumers) only take items out of the buffer. The pseudocode describing this problem is given below:

```
1   program Producer/Consumer;
2   const NumberOfProducers=10;
3         NumberOfConsumers=10;
4         BufferSize=100;           { maximum no. of items buffer can hold }
5   shared var num_items: integer;        { current no. of items in buffer }
6   var i: integer;
7   concurrent_procedure Producer;
8   begin
9       repeat
10          ProduceAnItem;                 { takes a random time to complete }
11          while num_items > BufferSize do;                  { busy waiting }
12          AddTheItemToBuffer;
13          num_items := num_items + 1;
14      forever
15  end;
16  concurrent_procedure Consumer;
17  begin
18      repeat
19          while num_items <= 0 do;                          { busy waiting }
20          RemoveAnItemFromBuffer;
21          num_items := num_items - 1;
22          ConsumeTheItem;                { takes a random time to complete }
23      forever
24  end;
25  begin                                                     { main program }
26      num_items := 0;
27      current_begin
28          for i := 1 to NumberOfProducers do
29              Producer;
30          for i := 1 to NumberOfConsumers do
31              Consumer;
32      current_end;
33  end.
```

(i) Indicate all critical sections in the above program by giving line numbers.

(ii) Provide the program with mutual exclusion by inserting necessary semaphore operations.

(iii) Assume that producer and consumer processes cannot share any global variables (such as `num_items`) except semaphores. Using pseudocode, outline a solution to the problem with appropriate semaphore operations.

[8 marks]

Question 3

(a) What is meant by the following?

- (i) A "pipe" in Unix;
- (ii) a networked file system;
- (iii) a polymorphic virus.

[6 marks]

(b) Describe in detail how a virus infects programs, and how an anti-virus program detects known viruses.

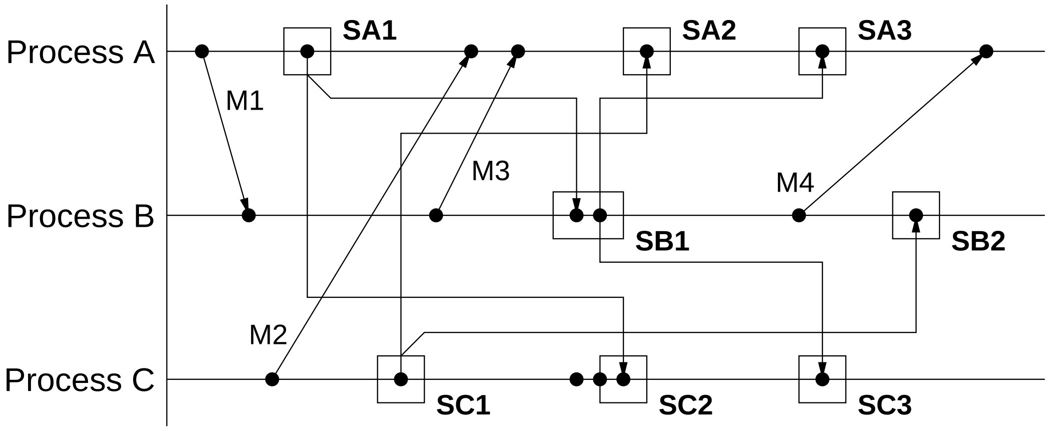
[5 marks]

(c) (i) What is an inode in a Unix file system? Show four examples of fields (other than addresses) contained in an inode.

(ii) A file containing 2050K bytes of data is to be stored in a Unix file sytem with 1K byte data blocks. Each indirect block can hold 256 disk addresses (i.e. addresses are 32 bits long). 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.

[7 marks]

(d) The following diagram illustrates the events and communications taking place prior to and during the execution of a distributed snapshot algorithm. In the diagram, M1, M2, . . . , are messages transferred between three processes; SA1, SA2, . . . , are the events in which processes receive and send markers. For instance, SA1 is the event where A initiates the algorithm and sends a marker to B and C respectively.



Describe the distributed snapshot algorithm with the aid of the above example, and indicate what are the states (i.e. incoming and outgoing messages) recorded upon each of the events.

[7 marks]