CS-228
OPERATING SYSTEMS

Attempt 2 questions out of 3

Question 1

(a) With the aid of a state transition diagram, illustrate the state transitions between process states **Running**, **Ready**, **Suspended-Ready**, and **Suspended-Blocked**. The events causing the transitions should be clearly indicated.

[5 marks]

(b) 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	P_5
Arrival time (ms)	0	1	4	6	9
Burst time (ms)	4	7	3	5	5

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

(i) Pre-emptive round robin scheduling with quantum = 2ms.

(ii) Pre-emptive round robin scheduling with quantum = 3ms.

[6 marks]

(c) Consider a simplified not-recently-used (NRU) page replacement strategy for demand paging memory management. Each page frame is associated with only the referenced bit **R**, which is set to 1 when the page is either referenced or modified.

(i) Outline a one-handed clock algorithm for the simplified NRU strategy.

(ii) What is the number of page frames that the algorithm has to scan in the worst case?

Unix System V (release 4) uses a two-handed clock to implement the NRU strategy. The first hand (reset hand) is followed by the second hand (search hand) by some fixed distance. When a page fault causes the second hand to search for a page frame with a 0 reference bit, the first hand moves ahead of it as well to reset the encountered reference bits.

(iii) What is the number of page frames that the algorithm has to scan in the worst case?

(iv) Discuss the merits and demerits of this approach in comparison with the one-handed clock algorithm.

[6 marks]

(d) What is meant by the following?

(i) A “signal” in Unix;

(ii) a semaphore;

(iii) a networked file system;

(iv) a polymorphic virus.

[8 marks]

Question 2

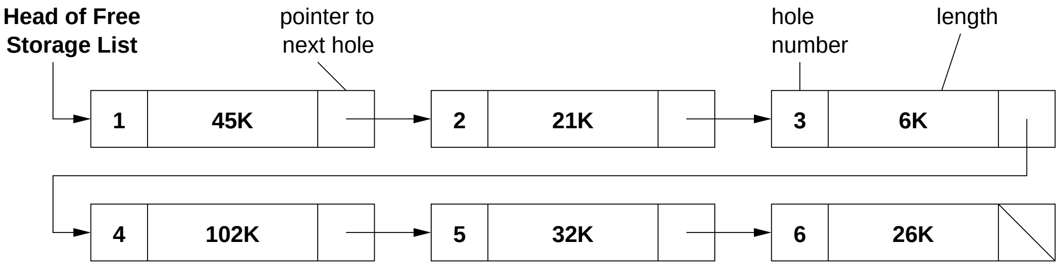
(a) (i) What is an inode in a Unix file system?

(ii) Give six examples of fields contained in an inode (other than pointers to disk blocks).

(iii) What is the difference between a hard link and a soft link in a Unix file system?

[7 marks]

(b) In a multi-tasking environment with a variable partition memory management scheme, the current free storage list (in storage address order) is shown below.



How would each of the following strategies place four incoming processes A, B, C and D, which arrive in that order and require 16K, 46K, 34K, and 11K memory respectively? Assume that no two holes are positioned next to each other.

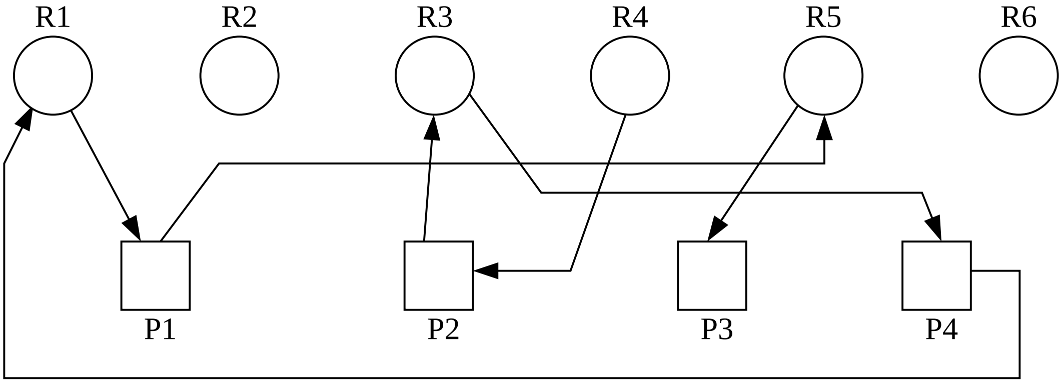
(i) first-fit;

(ii) next-fit;

(iii) best-fit.

[6 marks]

(c) Consider the following resource allocation graph.



(i) Give an example of a new request which would deadlock all four processes if it were granted. Briefly explain your answer.

(ii) Give an example of another request which would deadlock only two processes if it were granted. Briefly explain your answer.

[4 marks]

(Question 2 continued on next page.)

(d) A proposed solution, using two signal variables, to the problem of providing mututal exclusion to two concurrent processes, is given in pseudocode below.

Each process has associated with it a flag indicating its desire to enter the critical section. When a process wishes to enter the critical section, it first sets its flag, then checks the flag belonging to the other process. Provided the flag of other process is not set, it continues, otherwise it waits until the flag is cleared. When the process leaves its critical section, it clears its flag.

```
program Proposed Solution

shared var
    flags: array[0..1] of integer;

concurrent_procedure child(i: integer);
begin
    repeat                                       { child loops forever }
        do_non_critical_stuff();    { takes some random period of time }
        flags[i] = true;                   { set flag for this process }
        while flags[1-i] do             { check flag for other process }
            sleep(1);                      { if it’s set, wait a while }
        do_critical_section();          { perform the critical section }
        flags[i] = false;                  { when finished, clear flag }
    forever
end;

begin                                                 { main procedure }
    flags[0] = false;
    flags[1] = false;
    concurrent_begin                  { runs both children in parallel }
        child(0);
        child(1);
    concurrent_end
end.
```

(i) Show that the proposed solution provides mutual exclusion to the critical section.

(ii) Explain why the proposed solution is nonetheless unacceptable.

(iii) Given a function `set_if_true(a, b)`, which **atomically** performs

```
 if a then b := true;
```

rewrite the `child()` procedure to give a solution which meets all of the requirements of a mutual exclusion protocol.

[8 marks]

Question 3

(a) With demand paging memory management, a process references pages in the following order:

1 2 3 4 2 1 5 6 2 1 2 3 7 6 3 2 1 2 3 6

Assuming that all frames are initially empty, how many page faults would occur for the following page replacement strategies?

(i) First-in first-out, with a fixed allocation of 4 page frames.

(ii) Least-recently-used, with a fixed allocation of 4 page frames.

[6 marks]

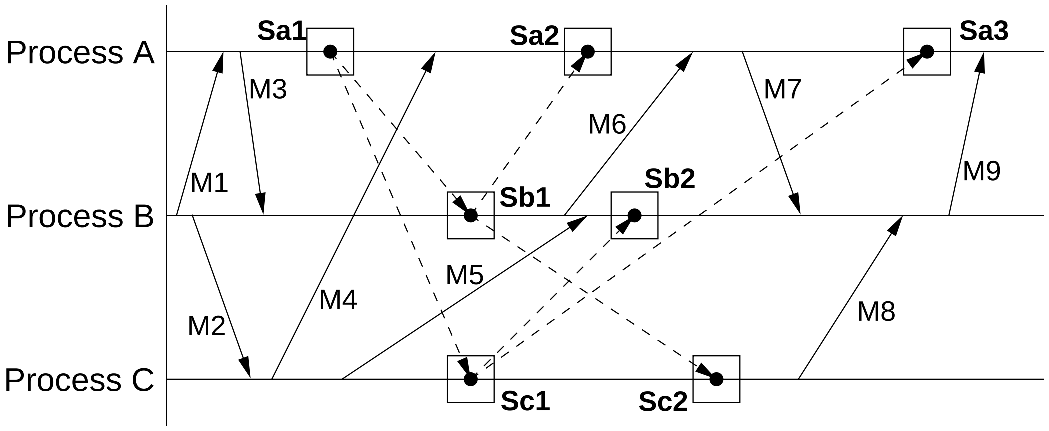
(b) Using a flowchart or pseudo-code, outline Dijkstra's deadlock avoidance algorithm (banker's algorithm) for resources of a single type.

[5 marks]

(c) In a multi-level secure system, what is the role of the reference monitor? What are the two rules which must be enforced during reading and writing operations? For each rule explain its importance with the aid of an example.

[6 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. The markers are represented with dotted lines.



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, and the final snapshot produced as a combination of those states.

[8 marks]