Two hours

UNIVERSITY OF MANCHESTER
SCHOOL OF COMPUTER SCIENCE

Operating Systems

Date: Thursday 26th January 2012

Time: 14:00 - 16:00

Please answer Question ONE and any TWO Questions from the other THREE questions provided

For full marks your answers should be concise as well as accurate. Marks will be awarded for reasoning and method as well as being correct

This is a CLOSED book examination

The use of electronic calculators is NOT permitted

1. **Compulsory**

a) What is the difference between a process and a program?
What is the difference between a process and a thread? (2 marks)

b) What is the key difference between a "system call" and calling an ordinary method or function? Briefly explain why this difference is important. (2 marks)

c) Explain what is meant by "Deadlock", and how it might occur. (2 marks)

d) Briefly explain what is a "FAT" (File Allocation Table), and how it is used.
Briefly explain what is an "inode", and how it is used. (4 marks)

e) When an interrupt occurs the processor undertakes a number of steps to handle it. The summary of the six steps is listed below but in the wrong order. In your answer put the six steps A to F into the correct order.

A. Processor accepts interrupt after current instruction.
B. External line interrupts processor.
C. The processor stores the information necessary to restart the original program following the interrupt.
D. Stored information is reloaded into the processor; processor continues executing the original program as if nothing had happened.
E. Interrupt Service Routine (ISR) is run for interrupting device until return from interrupt instruction is reached.
F. Interrupt acknowledgement (IACK) cycle identifies the interrupting device.
(2 marks)

f) Two registers are memory mapped to the following [memory addresses]:
Status Register at address 0xFFFF0000; and
Data Register at address 0xFFFF0004.

These are used to read a character from a peripheral. One of the registers is interrogated and the other is used to transfer data. In your answer state briefly:

i) Which register is first interrogated; and how [explicitly] is this undertaken. Then, if a character is to be read where it is read from; and subsequently stored. (1 mark)

ii) Finally, after reading the character what happens next? (1 mark)

g) Differentiate between "Relocation" and "Relocation & Protection". (2 marks)

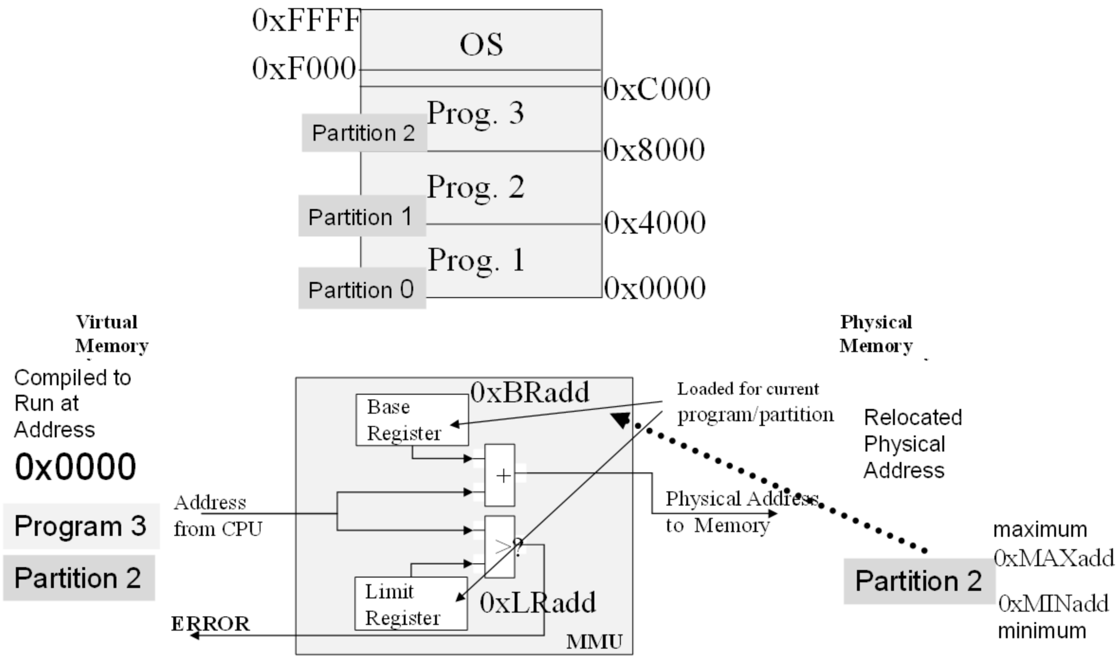
h) The diagram below shows three programs in three memory partitions, and base and limit registers. The base and limit hardware is to translate program 3, compiled to a base address 0x0000 in virtual memory, to run in partition 2 in physical memory. In the context of the depicted base and limits registers, state:

what addresses the base (BRadd) and limits (LRadd) registers will hold, and

(1 mark)

the relocated minimum (MINadd) and maximum (MAXadd) addresses in the physical memory.

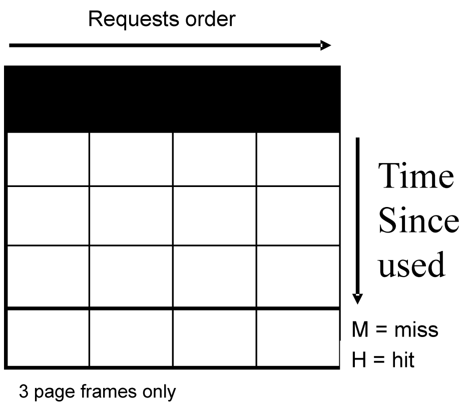
(1 mark)



i) The diagram overleaf depicts a table of a memory which has space for 3 page frames in each column. The diagram depicts space for four replacement operations; depicted by four columns. The page requested execution order is: 4 1 2 4; using the Least Recently Used policy.

Redraw the diagram of four time slots of a physical paged memory in your answer book. Then correctly fill in the 3 page frames; e.g. three rows under the "Requests order" row. Also, in the fifth row state if the page replaced is either a hit (H) or a miss (M).

(2 marks)


Requests order
Time
Since
used
M = miss
H = hit
3 page frames only

2. a) In the context of process scheduling, what are the three main states that processes can be in, and what are the four main transitions between these states caused by? (3 marks)

b) Explain the “First come first served” (FCFS) scheduling algorithm.
Explain what is meant by “preemption” and “time-slice”
Explain the “Round robin” (RR) scheduling algorithm.
In what way is RR better than FCFS scheduling?
Explain why new processes should be put at the back of the RR scheduling queue, rather than at the front. (5 marks)

c) Three processes A, B, and C all alternate between a CPU burst of 5 time units and an IO burst of 3 time units, where the length of a time-slice is 2 time units.

Draw a diagram to show the states of these processes as they are run by a RR scheduler for a total of 30 time-units, assuming that they all start ready to run at time-unit 0.

For how many of those 30 time-units is the CPU idle? (4 marks)

d) Modern process schedulers can have multiple queues, e.g. for processes with different priorities.

i) Explain what can go wrong if a high-priority queue has to be empty before processes on lower-priority queues are allowed to run, if priorities are static (externally defined). (1 mark)

ii) State and briefly explain an important reason for using dynamic (scheduler-defined) priorities, not directly related to your answer to part (i). (1 mark)

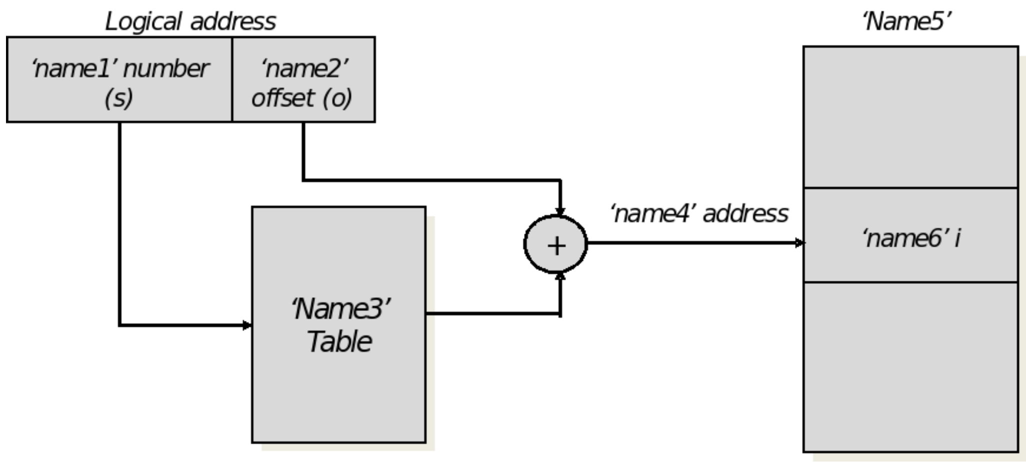
iii) A scheduler has 3 queues (Q1 to Q3). Processes on each queue are run round-robin for 1 time-unit each. Q1 gets 3 time-units, then Q2 gets 2 time-units, then Q3 gets 1 time-unit, and then Q1 gets 3 time-units again.
Assume that there are 6 processes (P1 to P6) waiting to run, each needing 5 time-units, and that P1 and P2 are in Q1, P3 and P4 are in Q2, and P5 and P6 are in Q3.
Draw a diagram to show the states of these processes as they are run by the scheduler for a total of 30 time-units (i.e. until they have all finished), assuming that they all start ready to run at time-unit 0.
What is the average turn-around time for the processes in each queue? (6 marks)

3. a) With respect to segmented virtual memory:

   i) State how segmented virtual memories "support the management of the execution of multiple processes in an operating system".
   (1 mark)

   ii) What does segmented virtual memory ensure in an operating system?
   (2 marks)

b) In the context of "segmented virtual address mapping" and given the diagram below; answer the following questions:



   i) Identify all the missing names of the components: "name1" to "name6".
   (3 marks)

   ii) Describe in detail how a "segment" is loaded. (5 marks)

   iii) External Fragmentation can occur, where memory space is wasted due to "holes" in the physical memory; when segmented virtual memory is used. State one way of avoiding external fragmentation; and state any drawback of this technique. (1 mark)

   iv) Name and briefly describe two alternative algorithms used to determine where to place the segment in a memory that has external fragmentation.
   (4 marks)

c) Given an 4G address spaces and associated 64K page sizes; calculate the number of pages that result in the virtual address space. (2 marks)

d) Given a physical address space size of 256MB and associated 64K block size below. Calculate the number of page frames in the physical address space. (2 marks)

4. a) One of the page replacement policies is the not recently used (NRU) algorithm; answer the following question with respect to this policy

   i) State the function of the "R" and "M" bits, which appear in a page table using the NRU algorithm. (2 marks)

   ii) State, in some detail, how the NRU algorithm works, explicitly saying how it utilises the "R" bit. (4 marks)

   b) When replacing a page, utilizing the NRU algorithm, the operating system divides the pages into four classes. To explain this concept draw up a table with three column headings "Class", "R bit", and "M bit". Then under this place four rows. Finally, use this table to state the settings of the "R" and "M" bits given these four classes, and state what classes "0" and "3" represent. (4 marks)

   c) Explain what is meant by a critical region of a multi-threaded program.
   Explain what is meant by mutual exclusion for critical regions.
   Explain what is meant by a semaphore, and the semaphore operations P() "procure" and V() "vacate".
   Explain the use of semaphores to enforce mutual exclusion for critical regions of a multi-threaded program. (5 marks)

   d) In a certain system, the execution of three threads A, B and C, is synchronised by semaphores S1 and S2, both initialised to 0, and only used as shown below. The threads share the variables x and y.

Thread A	Thread B	Thread C
P(S1)	P(S2)	x = 0
x = x + 1	y = 2	V(S1); V(S2)
x = x – 2	y = y – 1	y = 3

   What happens to the variable x, and why?

   There can be more than one outcome for variable y. What can happen, and why?

   Rewrite the code above, keeping the assignments to x and y unchanged, but changing how and/or where S1 and S2 are used so that, when all three threads have finished, x has the same final value as before and y has the value 1. Explain why your answer is correct.

   Rewrite the code above, keeping the assignments to x and y unchanged, but using any number of semaphores you need in any way so that, when all three threads have finished, x has the same final value as before and y has the value 3. Use the minimum number of extra sempahores to achieve this. Explain why your answer is correct. (5 marks)

END OF EXAMINATION