Two hours

Question ONE is COMPULSORY

UNIVERSITY OF MANCHESTER
SCHOOL OF COMPUTER SCIENCE

Operating Systems

Date: Wednesday 21st January 2015

Time: 14:00 - 16:00

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

This is a CLOSED book examination

The use of electronic calculators is NOT permitted

[PTO]

1. **Compulsory**

a) In the worst case, after making an access for data from a disk, the operating system has to wait for the disk head to move all the way across the disk (the maximum "seek time") and then for the disk to make a full rotation through 360 degrees (the maximum "rotation time") before the data starts arriving. Given a disk that makes 7200 revolutions per minute and has a maximum seek time of 20 milliseconds, what is the worst case wait time for data? How many clock cycles does this represent for a 3GHz CPU? What is the implication of this wait for the operating system. (2 marks)

b) In the context of process scheduling, explain briefly what a *CPU burst* is and what an *I/O burst* is. What is a *CPU bound* process, and what is an *I/O bound* process? Why is it a good scheduling strategy to give higher priority to I/O bound processes? (3 marks)

c) What does the term *deadlock* mean? How may deadlock occur? (2 marks)

d) Briefly explain the difference between a *process* and a *program*. (1 mark)

e) Why is the size of the time-slice in pre-emptive scheduling algorithms chosen to be significantly higher than the time taken for a context switch? (2 marks)

f) What is meant by the term *programmed I/O* in the context of operating systems? Give a brief answer – flowcharts are not required. (2 marks)

g) Briefly explain what is meant by the terms *fixed partitions* and *multiprogramming*. (2 marks)

h) A segmented memory, shown in Figure 1h, is to swap out segment 2 (of size 20kB) for segment 7 (of size 14kB). Draw a diagram depicting the segmented memory layout before and after the swap. Comment on any effect that swapping segment 2 for segment 7 has on the overall memory space. (2 marks)

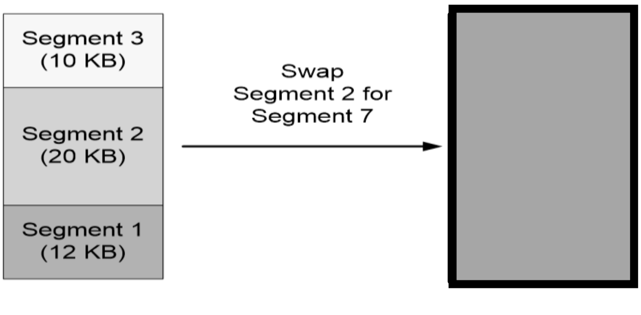

Figure 1h. A segmented memory, showing 3 segments.

i) In the context of relocation of memory addresses what are *base* and *limit* registers and what do they enable a virtual memory to undertake? (2 marks)

j) What is a *page fault*? Describe how a page fault is handled by the memory management unit and the operating system. (2 marks)

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

b) Three processes A, B and C all alternate between CPU bursts and I/O bursts of 5 time units each. Draw a diagram showing the states of these processes as they are run by a pre-emptive Round Robin scheduler for a total of 30 time-units, assuming that they all start ready at time-unit 0 and are queued in the order A (first), B, C (last), the time-slice adopted by the scheduler is 3 time-units, and the time for a context switch is negligible. For what fraction of the time is no process running because they are all blocked? How many context switches are performed during the 30 time-units? (4 marks)

c) Change the time-slice to 4 time-units and *either* repeat part b) for the new setup and, *or* otherwise, explain why a 3 time-unit time-slice produces a better schedule than a 4 time-unit time-slice for these processes. Are there any conditions under which the 4 time-unit time-slice might produce a schedule as good as that for the 3 time-unit time-slice (for these processes)? (5 marks)

d) Explain what a *semaphore* is and describe the operations that can be performed on it. (3 marks)

e) In a certain system, the execution of three threads is synchronised using three semaphores, S1, S2 and S3, as shown below. Semaphores S1 and S2 are initialised to zero, while semaphore S3 is initialised to 1. All three semaphores are used only in the sections of code shown below.

```
Thread A          Thread B          Thread C
…                 …                 …
P(S2)             P(S1)             P(S3)
P(S2)             P(S2)             V(S2)
x=3*x+4           x=x+7             x=x*5
V(S1)             V(S1)             V(S2)
 V(S2)            V(S2)             V(S3)
…                 …                 …
```

i) If the variable x is defined as an integer shared variable, initialised to 1, and is not assigned a value in any other sections of the code apart from those shown above, what will be its value when all three threads have finished executing? What will be the values of the three semaphores S1, S2 and S3? Justify your answers. (3 marks)

ii) Would there be any impact on the execution of the three threads if the two P operations in Thread B were interchanged (i.e. P(S2) is followed by P(S1))? Justify your answer. (2 marks)

PTO

3. a) To address the question: “What happens in an interrupt?”, it is important to align the appropriate set of steps to the proper interrupt sequence. Figure 3a shows a list of out-of-order steps relating to an interrupt sequence.

Put the steps into the correct order so they enable an interrupt to sequence through the proper interrupt steps. (3 marks)

a) return
b) code executed
c) interrupt acknowledgement
d) interrupt service routine
e) processor saves registers
f) interrupt line
g) interrupt vector

Figure 3a. Table showing a sequence of out-of-order steps for an interrupt.

b) Given that the now in-order steps enable an interrupt to follow the appropriate sequence, the table in Figure 3b shows a set of (out-of-order) steps for an interrupt sequence. Copy the re-ordered table from part a) into your answer book (plus a “detailed explanation of step” column) and then describe in full what happens at each step. Clearly put each step into context and describe exactly what function each performs. (5 marks)

Steps [please re-order]	**Detailed explanation of step**
a) return	
b) code executed	
c) interrupt acknowledgement	
d) interrupt service routine	
e) processor saves registers	
f) interrupt line	
g) interrupt vector	

Figure 3b. A keyword table, requiring re-ordering and adding a comprehensive explanation of each step.

c) With respect to relocation and swapping, state why the ability to relocate programs is useful in the context of swapping? (2 marks)

d) Given an 8GB virtual address space and associated 256KB page size, calculate the *number of pages* in this virtual address space. **NOTE**: To gain full marks you must show full working. (2 marks)

(Question 3 continues on the following page)

(Question 3 continues from the previous page)

e) Given a physical address size of 2GB and associated 128KB page size, calculate the *number of page frames* in the physical address space. **NOTE**: To gain full marks you must show full working. (2 marks)

f) One page replacement policy is the “not recently used” (NRU) algorithm; answer the following 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) Describe in detail how the NRU algorithm works, stating explicitly how it utilises the “R” bit. (4 marks)

PTO

4. a) On a paged machine with 3 physical page frames available, a particular process makes access to the following virtual pages in the order shown:

0, 3, 7, 1, 3, 2, 1, 3, 7

Show the contents of the 3 physical page frames and the cumulative total number of page faults (PFs) after each memory access, assuming that an LRU page replacement algorithm is in use and that the 3 physical page frames are initially empty. The kind of diagram you should draw is shown in Figure 4a. (4 marks)

Access:	0	3	7	1	3	2	1	3	7		
Most recent:	X	X	X	X	X	X	X	X	X	)	Contents of
Second most :	X	X	X	X	X	X	X	X	X	)	page frames
Third most:	X	X	X	X	X	X	X	X	X	)	in memory
Total PFs:	X	X	X	X	X	X	X	X	X		

Figure 4a. Diagram showing 3 page frames and the cumulative total number of page faults.

b) When a page fault occurs the CPU uses a DMA unit to move pages from/to disk (termed *disk I/O*); in this context:

i) Name and briefly describe the two registers in the DMA controller that are normally used during the disk I/O DMA process. (2 marks)

ii) Transfer of data to a disk can actually be undertaken by programmed input/output (programmed I/O) or by interrupt I/O. Draw two flow charts [diagrams] that depict the sequence of events needed to perform writing to disk using programmed I/O and using interrupt I/O. (4 marks)

c) Describe clearly the algorithm used to locate a file referred to by a full (absolute) pathname in a hierarchical file system. (4 marks)

d) In a file system using a File Allocation Table (FAT), the value obtained by the algorithm in part c) is the number of the first block of the file. Describe carefully how such a file system thereby provides access to the whole file. (3 marks)

e) In a file system using index nodes (i-nodes), the value obtained by the algorithm in part c) is the i-node number for the file. Describe carefully how such a file system thereby provides access to the whole file. (3 marks)

END OF EXAMINATION