Two hours

Closed Book Examination

UNIVERSITY OF MANCHESTER
SCHOOL OF COMPUTER SCIENCE

Operating Systems

? January

Time: ? – ? + 2.00

Marker's feedback version

Marking Scheme Included

Please answer Question ONE and TWO other questions.

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)

Model answer and marking scheme
1 mark: process = as it is running, program = e.g. code in a file
only .5 mark for: process = running; program = not running
1 mark: processes don't normally share resources (e.g. variables or address space), threads do.
only .5 mark for: process = several threads OR thread = unit of CPU utilisation (i.e. not mentioning shared resources)

Marker's feedback
process = threads **+ resources**
a program is not the same as a non-running process (see Q2a below)

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

Model answer and marking scheme
1 mark: system call used to change mode from user to system (or supervisor or kernel), to enable protected operations or to access restricted resources etc. (or else just give examples of protected operations)
1 mark: so that ordinary user code can't damage OS or other processes

Marker's feedback
Many students failed to answer the second sentence.
A system call is made **to** the system, not **by** it.
This wasn't a question about context switches (in fact linux & MSwindows **don't** context-switch for a system call, which is why user programs can only use part of the address space, as the system uses the rest within the user's process)

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

Model answer and marking scheme
2 marks: Set of processes or threads, all waiting for some resource(s) which others in the set already have, but can't release because they are also waiting.
only 1 mark if don't mention resources or event or lock or semaphore etc.
only 1 mark for: 2 processes waiting for same resource

Marker's feedback
Need to be clear **why** the processes are waiting – for resources or semaphores or ...
Not just process A waiting for process B – process B is also waiting for process A
Must be more than 1 resource
You can use an example to make this clear, if your english isn't clear enough by itself.
Several students talked about "non-shared" resources, whereas it should be shared resources with mutual exclusion. (see Q4c)

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.
(should I have asked for an example of each in use?) (4 marks)

Model answer and marking scheme
2 marks: e.g. FAT - used in old MS file-systems; a list of block indexes, used to indicate the next block of a disk used to store a file.
2 marks: e.g. inode - used in Linux file-systems - contains info about file + list of 15 blocks used, last 3 of which may be indirected to increase size.
for each:
1 mark for a simple description e.g. "list of clusters containing a file"
1 mark for more details or an example

Marker's feedback
A lot of students omitted this part or answered it very badly.
"FAT = table for allocating files" gets nothing.
Allocating **disk** space, not memory

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)

Model answer and marking scheme
Bookwork: correct order is B, F, A, C, E, D (or B A F C E D ?)
Distribution of Marks: 2 marks for a completely correct answer
1 mark for some basic understanding (or attempt).
Reference Learning Resources, Background Reading, and Lecture itself for detailed information; Lecture(s) No(s).: Lecture 16: Input and Output 2.

Marker's feedback
Pedagogic assessment [criterion]:
The question assesses lecture 15 learning objective [LO]: Describe what a processor does in response to an interrupt.
The answer should clearly evidence knowledge of required salient facts, the correct sequence.
Well done, most people were able to re-sort the list (of steps to handle an interrupt) into the correct order; in that case full marks were awarded. However, some [students] did not get the order explicitly correct and then marks were deducted.
Probably two ways that student's answers could have answered:
1) Re-write out the A, ... ,F capital letters in the correct order; or
2) re-write them [in full] in the correct order;
though this was totally up to the student - and may have aligned to how each student revised and remembered this information [list] associatively; sequentially, or used keywords [sequence].

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)

Model answer and marking scheme

Bookwork: The following points should be covered to some degree in the answer: Processor reads from address 0xFFFF0000 (status register), and has two possible consequences:

i.a. The status register is the first register to be interrogated;

i.b. The is undertaken by checking if one specific bit in the status register is set [equal “1”];

i.c. If, status bit cleared, bit is 0, there is no character to read.

i.d. If value is 1, there is a character to read, then:

i.e. Read character from address 0xFFFF0004 (data register); and

i.f. Stored in main memory;

ii. The I/O module will now clear the status register’s bit 0 to 0, it will remain at this value until another character is typed.

Distribution of Marks:

2 marks for an answer that depicts all the salient facts in a sensible way; all i.a to i.d and ii correctly delineated and briefly described;

1 mark for some basic understanding (or attempt).

Reference Learning Resources, Background Reading, and Lecture itself for detailed information; Lecture(s) No(s).: Lecture 15: Controlling Input and Output 1.

Marker's feedback
Pedagogic assessment [criterion]:
The question assesses lecture 14 learning objective [LO]: Explain in simple terms how a processor interacts with an I/O device; and State what is meant by programmed I/O; Simple I/O devices utilise the status and data register for 'processor interacts.'
The answer should clearly evidence knowledge of required salient facts. In the answer [some of] the following terminology (keywords and naming conventions) should be utilised in context; for example: status register [interrogated]; checking if one set; bit 0 is 0, there is no character to read; bit = 1, there is a character to read; Read character from data register; Stored in main memory; clear the status register's bit 0 to 0.
Well done most of you covered the majority of the seven pints in the template answer; and as such were awarded the appropriate marks.
Where marks were NOT awarded it was due to [not mentioning or stating or not utilising]:
1/ not utilising the correct [appropriate] technical terminology or naming conventions;
2/ not specifying [explicitly] stating the state [0 or 1] of the bits in the status register, when different evens happen (and what each means);
3/ stated addresses were not explicitly stated [utilises] in the answer, reference template answer.
4/ what finally happens to the status register bit, once the data register is read; and
5/ really, the marks were explicitly given [awarded] for an explicit [technically correct] answer; as it states [explicitly] in the question; this was a real hint to be verbose and not brief.

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

Model answer and marking scheme

Bookwork: The following points should be covered to some degree in the answer:

Relocation: "map each process' address space onto different parts of physical memory." It does NOT [per se] undertake to provide any protection.

[One way to solve the addresses problem is to have a loader which "fixes up" addresses, i.e. an instruction of the form BL 100 can be detected (its binary form) as it is being loaded and have the address modified.]

Relocation & Protection: Previous scheme [relocation only] does not solve any protection problem - programs can write to any memory [address].

An example is a simple (once used) solution is to introduce "base" & "limit" registers loaded before a program executes.

Address added to the base but must be less than the limit.

Distribution of Marks:

2 marks for an answer that depicts all the salient facts in a sensible way; provides a well-defined differentiation;

1 mark for some basic understanding (or attempt).

Reference Learning Resources, Background Reading, and Lecture itself for detailed information; Lecture(s) No(s).: Lecture 11: Memory Management (1).

Marker's feedback

Pedagogic assessment [criterion]:

The question assesses lecture 10 learning objective [LOs]; Differentiate between 'Relocation' and 'Relocation + Protection'.

The answer should clearly evidence knowledge of required salient facts.

In the answer [some of] the following terminology (keywords and naming conventions) should be utilised in context; for example: map each process, address space, different parts, physical memory, loader 'fixes up' addresses, base, limit, greater then [base], less [or equal to] limit.

Well done, most of you did correctly assign the six names to the Name1, ... Name6 - in the diagram - and were awarded full marks.

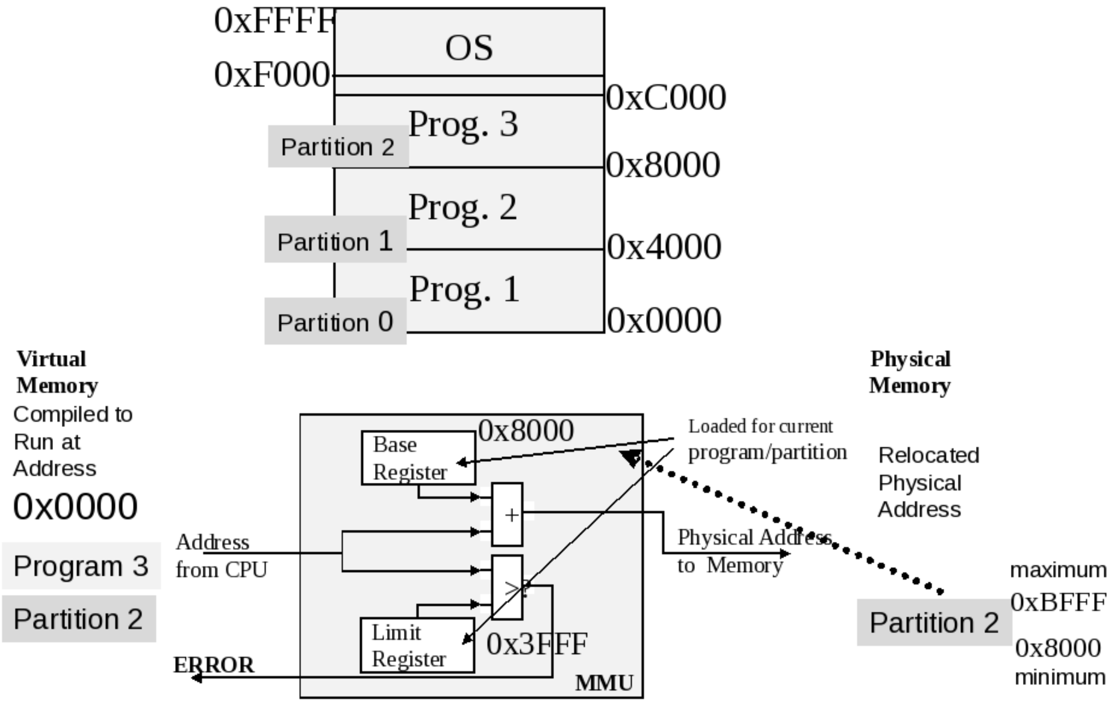
If any of the names were incorrect marks were not awarded.

NOTE: this diagram and associated theory in in a number of text book and in your notes.

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)



Model answer and marking scheme

Application

Example answer: The following points should be covered to some degree in the answer:

i) State what addresses the base:

0xBRadd = 0x8000 and limits:

0xLRadd = 0x3FFF registers will hold; (1 mark)

ii) Then state the relocated; minimum:

0xMINadd = 0x8000 and maximum:

0xMAXadd = 0xBFFF addresses in the physical memory. (1 mark)

Distribution of Marks:

2 marks for a totally correct four numbers; 0xBRadd, 0xLRadd, 0xMINadd & 0xMAXadd.

1 mark for some basic table (or attempt).

Reference Learning Resources, Background Reading, and Lecture itself for detailed information; Lecture(s) No(s).: 11; Memory Management (1).

Marker’s feedback

Pedagogic assessment [criterion]:

The question assesses lecture 10 learning objective(s): Demonstrate an understanding of Relocation; Differentiate between ’Relocation’ and ’Relocation + Protection’.

The answer should clearly evidence knowledge of required salient facts.

In the answer [some of] the following terminology (keywords and naming conventions) should be utilised in context; for example: base , 0xBRadd, limits, 0xLRadd, minimum [address], 0xMINadd, maximum [address], 0xMAXadd.

Well done, most of you were able to differentiate; given the real differentiator was to utilise the key words or technical terminology: ’base’ and ’limits’ registers.

If a good description - that differentiated well - was written [in your answer] appropriate marks were awarded.

However, a clear distinction had to be made - and if the appropriate keywords [or technical terminology was not utilised - marks were not awarded.

May be the terminology you should have used - if you did not - was [even] hinted at in the next question!

One of the issues that students did get confused was:

1/ process; and

2/ process address space.

It is not the process that is moved; rather it is the relocation of the mapped process address space e.g. each process is composed of a set [or sequence] of instructions; starting a a specific address - or base address. To move the set of instructions to the correct space [location] in primary [physical] memory from secondary [virtual] memory - the secondary [virtual] memory base address of the instructions [nominally compiled to 0x0...0] is relocated to a new base address in in primary [physical] memory e.g. 0x8000 [in this case].

[AGAIN] Well done, most of you did calculate the set of four addresses correctly; and were awarded full marks.

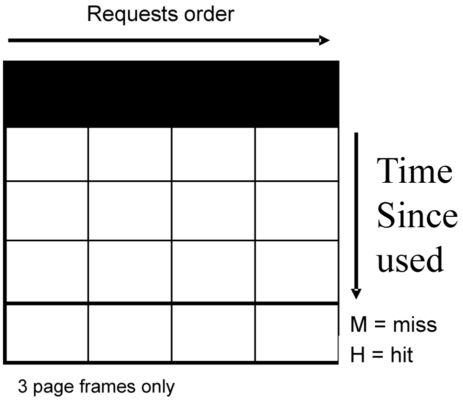
Some did not remember [or work out] that the ’Limit’ is always below [in numerical value] the next base address up; e.g. for a BRadd & LRadd for partition 0: base of parttion 0 is 0x000 - so remembering the limit LRadd for partition 0 is 0x3FFF not 0x4000 [which is the BRadd for partition 1].

The main issue here is that a program with an overall size of 0x4000, has a memory mapped virtual address min of 0x000 and max of 0x3FFF hence:

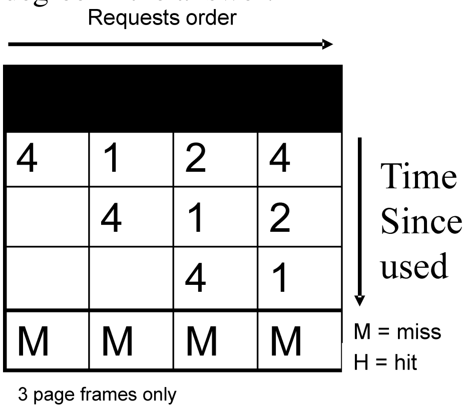
moving it to partition 0; sets the base to 0x0000 and Limit to 0x3FFF;
moving it to partition 1; sets the base to 0x4000 and Limit to 0x3FFF;
moving it to partition 2; sets the base to 0x8000 and Limit to 0x3FFF;
As you can see the Limit register is the same for each partition.

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)



Model answer and marking scheme
Application
Example answer: The following points should be covered to some degree in the answer:



4	1	2	4
	4	1	2
		4	1
M	M	M	M

Distribution of Marks: 2 marks for a totally correct table,
1 mark for some basic table (or attempt).
Reference Learning Resources, Background Reading, and Lecture itself for detailed information; Lecture(s) No(s).: 14; Virtual Memory (3)

Marker’s feedback

Pedagogic assessment [criterion]:

The question assesses lecture 13 learning objective [Los]: Discuss the concept of ’Page Replacement’ and Differentiate between First in First Out (FIFO) and Least Recently Used (LRU).

The answer should clearly evidence knowledge of required salient facts.

In the answer [some of] the following terminology (keywords and naming conventions) should be utilised in context; for example: request(s) order, 3 page frame, time since used, H, M, ... most of these should [really] be on your diagrams.

Well done, most were happy with the concept of:

a) a table depicting pages in a page frame table;
b) 3 page frames in each;
c) page request execution order; and
d) the least frequently used (LRU) policy; plus
e) the concept of hit [H] and miss [M].

and with this knowledge you depicted the correct answer in the diagram you draw up for your answer; and were awarded full marks.

Some, either did not write down the page numbers in the correct order [in the rows and columns]; or missed out H & M. Others were not sure at all and did not fill in the table correctly; hence marks could not be awarded.

Marker's feedback
Please start your answer to each new question at the start of a fresh page in your answer book!
Q2 was answered by 142 students, with an average mark of 13/20

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)

Model answer and marking scheme
1 mark: States = Running, Ready, Blocked (names can be approximate as long as concepts correct)
1 mark: Running to Blocked - waiting for e.g. I/O or semaphore
& Blocked to Ready - wait ended
1 mark: Running to Ready - time-slice over, preempted by scheduler
& Ready to Running - given time-slice by scheduler

Marker's feedback
This part was mostly answered very well

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)

Model answer and marking scheme
1 mark: FCFS = run each process in turn, in order of arrival
1 mark: preemption = scheduler stops process after some fixed maximum time (the time-slice), even if it hasn't finished or had to wait yet.
1 mark: RR = FCFS using time-slicing, putting each prempted process on the back of the queue
1 mark: "fairer" as each process gets a turn (or prevents "hogging" etc.) but also average turnaround time improves as shorter jobs finish earlier on average
1 mark: to avoid starvation (again) or equivalent explanation
simply saying "fairer" only gets .5 mark

Marker's feedback
I wanted to know **why** RR is better than FCFS, not just an assertion that it was.
Pre-emption is not about **predicting** the amount of CPU time that a process will need.
For the last question, I wanted to know about **strategies**, not about how hard it might be to get the code right, or how strict the definition of a queue is.

c) Three processes A, B, and C all [repeatedly] 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)

(should I have asked for an explanation of how you got the answer from your diagram?)

Model answer and marking scheme
3 marks for perfect diagram:

```
  123456789012345678901234567890
A:RR----RR----RIIIRR----RR----RI
B:--RR----RR---RIII-RR----RR---R
C:----RR----RR--RIII--RR----RR--

R=running, I=I/O, -=ready;
```

1 mark: CPU idle for 1 time-unit (at time 16, when the Is align)
0 marks for correct reading of incorrect diagram
I allowed for misreading the question e.g. not repeating the CPU burst after the I/O burst.
Only 2 marks out of 4 for a major mistake but otherwise plausible diagram e.g. no I/O burst so just timeslicing CPU bursts etc.
Only 1 mark out of 4 for 2 serious mistakes

Marker's feedback
I/O doesn't take CPU time, so I/Os can overlap (unless all process are blocked at once)
A timeslice can end early (e.g. 5th unit of CPU burst)
CPU idle is not the same as some processes waiting; it is only if **all** processes are waiting.

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)

Model answer and marking scheme
if new high-priority processes keep arriving, the low-priority processes may have to wait for ever (starvation).

Marker's feedback
Most students got this.

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)

Model answer and marking scheme
Used by the scheduler to try to optimise some extra criterion e.g. give I/O-bound processes higher priority, to improve turnaround.
0 marks unless mention "extra criterion" or equivalent

Marker's feedback
Many students missed "not directly related to your answer to part (i)"
I don't think anyone got the mark, and just a handful of students got .5 mark. Maybe I didn't explain this part very well in the lectures :-(
Hardly anyone mentioned that this is a way to try to optimes some (extra) criteria in some particular situation (e.g. real-time) and what exactly you do depends on what you are trying to optimise. Many just said e.g. "this is to improve turnaround for small processes" or "for I/O bound processes" etc.

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.
(should I have said "and then repeat giving each Q its time-slices in turn"?)

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)

Model answer and marking scheme
5 marks for a perfect diagram (as below) (allowing for misreading of question)
–1 leaving gaps for empty queues
–1 give empty Q1's time to Q2
–1 give all Q's time-slices to 1 process

```
   123456789012345678901234567890
P1:R-R----R----R-R               = 15
P2:-R----R-R----R----R           = 19
P3:---R-----R-----R---R--R       = 23
P4:----R-----R-----R---R--R      = 24
P5:-----R-----------R------R-R-R = 29
P6:-----------R---------R---R-R-R= 30

R=running, -=ready;
```

Some students gave a better diagram, concentrating on the Qs and showing which P is running within each time-slice:

```
   123456789012345678901234567890
Q1:121---212---121---2
Q2:---34----34----34--34-34
Q3:-----5-----6-----5---5--565656
```

1 mark: Average turnaround time correctly calculated from diagram (even if diagram is wrong) e.g. Q1=17, Q2=23.5, Q3=29.5
only 0.5 if turnarounds all 1 too small, or correct but not averaged properly

Marker's feedback
Empty queues don't get time allocated.
e.g. P1 and P2 round-robin within Q1, not P1 runs for 3 (or 5) and then P2, nor P1 always gets 1st & 3rd time-slices of each set of 3.
I didn't intend Q1 to get extra goes (Q1 Q2 Q3, Q1 Q2 Q3 not Q1 Q2 Q3 Q1, Q1 Q2 Q3 Q1) but I didn't knock marks off for anyone who misread the question

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)

Model answer and marking scheme
Bookwork (2 marks): The following points should be covered to some degree in the answer:
"Segments are a way of dividing [up] the virtual address space" to support the management of the execution of multiple processes in an operating system.
Distribution of Marks:
1 mark for an answer that depicts all the salient facts in a sensible way; and the use of keywords in context: "dividing", and "virtual address space";
.5 mark for some basic understanding (or attempt).
Reference Learning Resources, Background Reading, and Lecture itself for detailed information; Lecture(s) No(s).: Lecture 13: Controlling Input and Output 1.

Marker's feedback
Pedagogic assessment [criterion]:
The question assesses lecture 12 learning objective [LOs]: Explain what is meant by segmented virtual memory.
The answer should clearly evidence knowledge of required salient facts.
In the answer [some of] the following terminology (keywords and naming conventions) should be utilised in context; for example: Segments, dividing [up], virtual address space, support, management, execution, multiple processes, operating system.
Well done, a number of you were correct and delineated correctly the main rational [in many books and the lectures] that stated "Segments are a way of dividing [up] the virtual address space to support the management of the execution of multiple processes in an operating system;" if you put this in a concise sentence, and used all the correct technical terminology, you were awarded full marks.
Those whom did not write down the full set of salient facts were marked accordingly; see template answer.
Some [answers] were not detailed enough - and did not use the correct naming conventions [or terminology] in context to state how segmented virtual memory support the management of the execution of multiple processes in an operating system; and hence marks were not awarded.

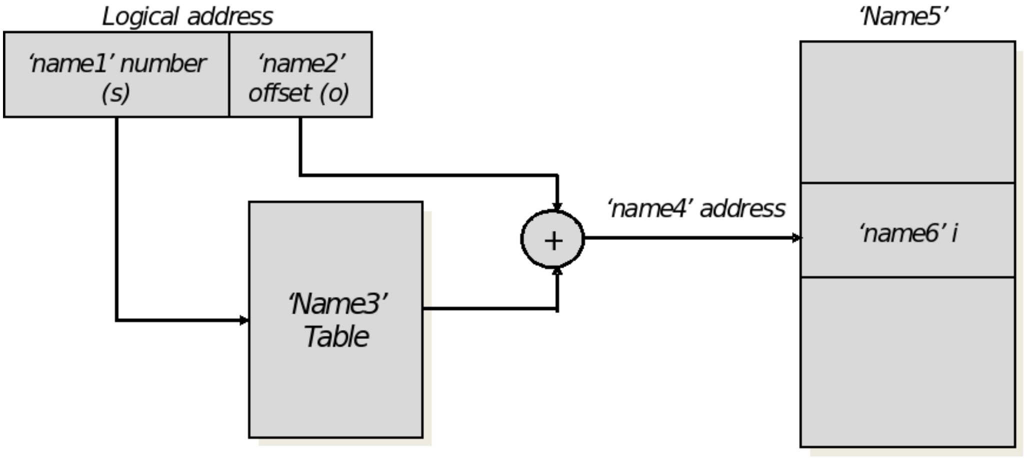
ii) What does segmented virtual memory ensure in an operating system?

(2 marks)

Model answer and marking scheme
Bookwork
The following points should be covered to some degree in the answer:
Dividing [up] the virtual address space to:
1) ensure that processes do not interfere with one another;
2) ensure that operating system has control of the computer and will prevent
3) programs written by users taking over the computer.
Distribution of Marks:
2 marks for an answer that depicts all the salient facts in a sensible way; and the use of keywords in context: “interfere”, “[OS has] control”, “prevent [others]”, “[taking over] control”, explicitly ensures two different issues;
1 mark for some basic understanding (or attempt).
Reference Learning Resources, Background Reading, and Lecture itself for detailed information; Lecture(s) No(s).: Lecture 15: Controlling Input and Output 1.

Marker’s feedback
Pedagogic assessment [criterion]:
The question assesses lecture 12 learning objective [LOs]: Explain what is meant by segmented virtual memory; and Describe how segmented and paged virtual memory systems can be used; in this case for the 3 issues mentioned in the template answer....
The answer should clearly evidence knowledge of required salient facts. In the answer [some of] the following terminology (keywords and naming conventions) should be utilised in context; for example: do not interfere, [operating] system has control, [another program] taking over.
Well done, most of you, correctly surmised that there were [in fact] three issues that needed to be covered [in your answer] - each had its own set of key words [terminology] that you should have utilised in context; if this was achieved full marks were awarded.
However, if:
1/ the three different issues and associated key words [terminology] wer not used in [the correct][context; or
2/ the different issues were not [all] covered;
marks were not awarded.

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)

Model answer and marking scheme
name1 = segment
name2 = segment
name3 = Segment
name4 = Physical
name5 = Memory
name6 = segment
Distribution of Marks:
3 marks for a concise naming of all the six,
2 marks for three,
1 mark for one,

Marker's feedback
Pedagogic assessment [criterion]:
The question assesses lecture 12 learning objective [LOs]; Explain what is meant by segmented virtual memory; and Describe how segmented and paged virtual memory systems can be used.
The answer should clearly evidence knowledge of required salient facts.
In the answer [some of] the following terminology (keywords and naming conventions) should be utilised in context; for example: segment, Physical, & Memory.
Well done, most of you did correctly assign six names to the Name1, ... ,Name6 to the unnamed components in the diagram; as such [and in that case] full marks were awarded.
Note this diagram or diagrams very similar is [actually] in a number of text books [that this course specifies and a number of courses round the world also utilise] and in the lecture notes.

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

Model answer and marking scheme

Bookwork

Example answer: The following points should be covered to some degree in the answer:

Description of the segmentation process; or how a "segment" is loaded; this works in almost exactly the same way as paged virtual memory:

1.The processor generates a logical address.

2.The segment number field is used by the MMU (memory management unit) to look to see whether the segment is in memory or not, then:

3.If it is in memory, a physical address is computed by adding the base address of the segment to the offset; this is used as a physical address to memory; or

4.If it is not in memory, the transfer is aborted (segment fault); and

5.then the operating system will load the segment from disk to memory.

Distribution of Marks:

5 marks for a concise description of complete segmentation loading process,

2.5 marks for half issues covered,

1 mark for some basic understanding (or attempt).

Marker's feedback

Pedagogic assessment [criterion]:

The question assesses lecture 12 learning objective [LOs]: Explain what is meant by segmented virtual memory.

The answer should clearly evidence knowledge of required salient facts.

In the answer [some of] the following terminology (keywords and naming conventions) should be utilised in context; for example: processor, generates, logical, address, segment, number, field, MMU (memory management unit), physical address, computed, adding, base address, segment offset, physical address, load, disk to memory.

Well done most students were able to list the five points [covering this issue] and utilised the correct technical naming conventions; hence were awarded full marks.

Where the process was not explicitly explained (in the correct sequence and using the correct terminology) as a sequence of concise steps; or is steps (and terminology) were missed [out] marks were reduced.

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)

Model answer and marking scheme
Bookwork
Example answer: The following points should be covered to some degree in the answer
One way of avoiding external fragmentation is: Compact the memory by “shuffling” segments in memory to fill the holes.
Drawback: But this requires extensive copying of data and is excessively time consuming.
Distribution of Marks:
1 marks for a concise description of method and drawback; and the use of keywords in context: “compact” and “shuffling”,
.5 mark for some basic understanding (or attempt).
Reference Learning Resources, Background Reading, and Lecture itself for detailed information; Lecture(s) No(s).: 13; Controlling Input and Output 2.

Marker’s feedback
Pedagogic assessment [criterion]:
The question assesses lecture 12 learning objective [LOs]; General information related to Virtual Memory (2); Segmented Virtual Memory.
The answer should clearly evidence knowledge of required salient facts
In the answer [some of] the following terminology (keywords and naming conventions) should be utilised in context; for example: Compact, memory, ’shuffling’, segments, fill holes.
Well done most of you were correct and did [correctly] ascertain that compaction [using shuffling] was the method utilised; if this was explained well [with correct terminology] full marks were awarded.
However, if the answer was fuzzy [either grammatically or in the technical sense], or not concise, and did not utilise the correct technical terminology marks would not have been awarded.

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)

Model answer and marking scheme
Bookwork (4 marks),
Example answer: The following points should be covered to some degree in the answer:
Alternatively, an algorithm can be used to determine where to place the segment in a memory that has external fragmentation.
The operating system maintains a list of the addresses and sizes of all the holes and can use algorithms like, two alternative algorithms are:
i) Best Fit - scan the complete "list of holes" and determine which best fits the segment; tends to produce a lot of small holes.
ii) First fit - scan the list until a [the first] hole is found that fits the segment.
Distribution of Marks:
4 marks for a concise description of complete segmentation loading process; names [correctly] both algorithms and concise descriptions,
2 marks for half issues covered,
1 mark for some basic understanding (or attempt).
Reference Learning Resources, Background Reading, and Lecture itself for detailed information; Lecture(s) No(s).: 13; Controlling Input and Output 2.

Marker's feedback
Pedagogic assessment [criterion]:
The question assesses lecture 12 learning objective [LOs]: Apply first-fit and best-fit algorithms for loading segments into memory.
The answer should clearly evidence knowledge of required salient facts.
In the answer [some of] the following terminology (keywords and naming conventions) should be utilised in context; for example: Best Fit, scan, list, holes, segment; small holes, First fit, fits the segment.
Well done most specified 'best' and 'first' fit correctly and gave descriptions that utilised the correct terminology when explain their operation; and hence full marks wer3e awarded.
However some, did not know the names of both, or they did not describe them using the correct technical terminology; hence [their] marks were reduced. Some could explain the algorithmics but could [did] not name them explicitly; so marks were deducted.

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)

Model answer and marking scheme

Application Example answer: The following points should be covered to some degree in the answer:

The virtual address space will consist of a number of pages:

Number of pages = address space / page size = 4GB / 64KB = $2^{32}/2^{16} = 2^{16}$ = 65,536 or 64K pages.

NOTE: 2^{32} = 4,294,967,296 or 4G and 2^{16} = 65,536 or 64K

e.g. if the virtual address space is 4 GB and the page size is 64 KB there are: (64K) pages.

Distribution of Marks:

2 marks for a totally correct calculation,

1 mark for some basic calculations (or attempt).

Reference Learning Resources, Background Reading, and Lecture itself for detailed information; Lecture(s) No(s).: 12; Virtual Memory (1).

Marker's feedback

Pedagogic assessment [criterion]:

The question assesses lecture 11 learning objective [LOs]; Determine [calculate] the structure of an address in a paged virtual memory system. The answer should clearly evidence knowledge of required salient facts. In the answer [some of] the following terminology (keywords and naming conventions) should be utilised in context; for example: virtual, address, space, number of pages.

Well done the majority of you for detailing the correct calculation for number of pages; if all the steps [and they had to be all] were presented in your answer [with all the correct mathematics, calculation, and symbols (k, M,G, B)] then full marks were awarded; if each and every step was completely correct; then full marks were awarded; if your answer did not meet these requirements marks were not awarded.

The question itself has been covered in:

1/ the lecture [series];

2/ Tutorials on Blackboard 9; 'NumberOfPagesTutorial' (*.pdf);

3/ Real time video tutorial [again on Blackboard 9] aligned to COMP25111 lecture 11 calculation of pages; and

4/ plus text books.

So lots of help has been provided for this type of exercise.

For those whom either:

1/ Did not write down the mathematical [symbolic] equation [in full]; or

2/ Did not show full working, on every step of the calculation; or

3/ Did not perform the calculations correctly

Full marks were not awarded.

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)

Model answer and marking scheme
Application
Example answer: The following points should be covered to some degree in the answer:
The physical address space will consist of a smaller number of page frames: Number of page frames = address space / block size = 256MB / 64KB = $2^{28}/2^{16} = 2^{12}$ = 4096 or 4K pages.
NOTE: 2^{28} = 268,435,456 or 256M; 2^{16} = 65,536 or 64K; 2^{12} = 4096 or 4K e.g. if the physical address space is 256 MB and the block size is 64 KB there are: (4K) page frames.
Distribution of Marks:
2 marks for a totally correct calculation,
1 mark for some basic calculations (or attempt).
Reference Learning Resources, Background Reading, and Lecture itself for detailed information; Lecture(s) No(s).: 12; Virtual Memory (1).

Marker’s feedback

Pedagogic assessment [criterion]:

The question assesses lecture 11 learning objective [LOs]; Determine [calculate] the structure of an address in a paged virtual memory system.

The answer should clearly evidence knowledge of required salient facts.

In the answer [some of] the following terminology (keywords and naming conventions) should be utilised in context; for example: virtual, address, space, number of page frames, block size.

Most, correctly ascertained the majority of you for detailing the correct calculation for number of pages; if all the steps [and they had to be all] were presented in your answer [with all the correct mathematics, calculation, and symbols (k, M,G, B)] then full marks were awarded; if each and every step was completely correct; then full marks were awarded; if your answer did not meet these requirements marks were not awarded.

The question itself has been covered in:

1/ the lecture [series];

2/ Tutorials on Blackboard 9; ’NumberOfPagesTutorial’ (*.pdf);

3/ Real time video tutorial [again on Blackboard 9] aligned to COMP25111 lecture 11 calculation of page frames; and

4/ plus text books.

So lots of help has been provided for this type of exercise.

For those whom either:

1/ Did not write down the mathematical [symbolic] equation [in full]; or

2/ Did not show full working, on every step of the calculation; or

3/ Did not perform the calculations correctly

Full marks were not awarded.

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)

Model answer and marking scheme
Bookwork
The following points should be covered to some degree in the answer:
The R and M bits in a page table are used [in the following way]:
The R bit is set when the page is referenced; while
The M bit is set when the page is modified (or written to).
Distribution of Marks:
2 marks for an answer that depicts all the salient facts in a sensible way; and the use of keywords in context: "referenced", and "modified";
1 mark for some basic understanding (or attempt).
Reference Learning Resources, Background Reading, and Lecture itself for detailed information; Lecture(s) No(s).: Lectures 14: Virtual Memory (3).

Marker's feedback
Pedagogic assessment [criterion]:
The question assesses lecture 13 learning objective [LOs]; Explain Not Recently Used (NRU).
The answer should clearly evidence knowledge of required salient facts.
In the answer [some of] the following terminology (keywords and naming conventions) should be utilised in context; for example: page table, referenced, modified.
Most [students], correctly ascertained that R = referenced and M = modified - well done - and if you gave a very brief description as well - no reason full marks were not awarded.
Marks were not awarded if:
1/ only one was correctly named;
2/ Both were named but the functions they performed were not explicitly presented to meet the questions "State the function" requirement;
3/ fuzzy grammar or technical description was given;
4/ correct [technical] terminology was not used in your description of these functions.
then marks were not awarded; or if the English description was not constructed well and put in plain English and good grammar was not used.

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

Model answer and marking scheme
Bookwork:
The following points should be covered to some degree in the answer:
How it works:
1. At fixed intervals, the clock interrupt triggers and clears the referenced bit (R = 0) of all the pages.
2. Referenced bit marks pages referenced in interval.
3. So during interval if page referenced R=1 then it is used.
4. If not R=0 then it is NOT used; and at the end of interval is a candidate for replacement.
Distribution of Marks:
4 marks for an answer that depicts all the salient facts in a sensible way; and the use of keywords in context;
2 marks for an average answer;
1 mark for some basic understanding (or attempt).
Reference Learning Resources, Background Reading, and Lecture itself for detailed information; Lecture(s) No(s).: Lectures 14: Virtual Memory (3).

Marker's feedback
Pedagogic assessment [criterion]:
The question assesses lecture 13 learning objective [LOs]; Explain Not Recently Used (NRU).
The answer should clearly evidence knowledge of required salient facts.
In the answer [some of] the following terminology (keywords and naming conventions) should be utilised in context; for example: fixed intervals, clock, interrupt, triggers, clears, referenced, bit, (R = 0) of all the pages, marks, pages, interval, page referenced, R=1, R=0 then it is NOT used; and end, interval, candidate, replacement.
Those that did explicitly explain the working [process behind] the M and R bits in the NRU page replacement algorithm full - utilising the correct terminology - and sequence of steps - were awarded full marks.
If however:
1/ the full sequence (up to 4 steps) was not fully explained; or
2/ the correct technical terminology was not utilised in the explanation; or
3/ the setting of the bits possible: 00, 01, 10, 11 combinations were not depicted and explained in your answer; or
4/ the preferred [candidate] for replacement was not stated explicitly 00-first etc
then marks were not awarded.

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)

Model answer and marking scheme
Bookwork The following points should be covered to some degree in the answer:

Class	R	M
0	0	0
1	0	1
2	1	0
3	1	1

Class 0: is the “Not recently used” page.
Class 3: is the “Recently used” page.
Distribution of Marks:
4 marks for an answer that depicts all the rows correctly, plus the meanings of classes 0 and 3;
2 marks for half the rows, attempt at meanings for classes 0 and 3;
1 mark for some basic understanding (or attempt).
Reference Learning Resources, Background Reading, and Lecture itself for detailed information; Lecture(s) No(s).: Lectures 14: Virtual Memory (3).

Marker’s feedback
Pedagogic assessment :
The question assesses lecture 13 learning objective [LOs]; Explain Not Recently Used (NRU).
The answer should clearly evidence knowledge of required salient facts.
In the answer [some of] the following terminology (keywords and naming conventions) should be utilised in context; for example: Class, R bit (referenced), M bit (modified), ’Not recently used’ page, & ’Recently used’ page.
Well done those that drew up the table correctly and named: class 0 as the ’not recently used’ page; plus named class 3 as the ’recently used’ page; in this case full marks were awarded.
However, if:
1/ The bit pattern in the columns R & M were wrong; or
2/ The naming conventions [technical terminology] for class 0 and class 3 were not explicitly [correctly] stated - using the correct terminology then marks were not awarded.

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 how semaphores can be used to enforce mutual exclusion for critical regions of a multi-threaded program.

(5 marks)

Model answer and marking scheme
1 mark: a critical region uses variables that are shared [modified?] by 2 or more threads
(only 0.5 mark for sharing **code**)
1 mark: mutual exclusion means that at most 1 thread can be using a particular shared variable (or in critical region) at once.
2 marks: a semaphore is an integer variable; P() waits until the semaphore is > 0 and then decrements it; V() increments it, and if there is a thread waiting in a P() operation on that semaphore, wakes it (or equivalent description)
(1 mark for increment/decrement etc.; 1 mark for wait/wake etc.)
1 mark: use a semaphore (initialised to 1) for each set of mutually exclusive regions; do a P() to enter a critical region and a V() to leave.

Marker’s feedback
mutex: at **most** 1, not at **least**

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)

Model answer and marking scheme
1 mark: order must be: x=0 V(S1) P(S1) x=x+1 x=x-2 so x ends up as -1
1 mark: after V(S2), can then do y=3 P(S2) y=2 y=y-1 i.e. y=1 or P(S2) y=2 y=y-1 y=3 i.e. y=3 (or “y=3 can happen anywhere”) (or, with pre-emption, even y=2 y=3 y=y-1 i.e. y=2)
I accepted any 2 or 3 alternatives
1 mark: in thread C, move V(S1); V(S2) below y=3
2 marks: no extra semaphore
B: y=2; y=y-1; V(S2);
C: x=0; V(S1); P(S2); y=3; (P can be earlier, V can be later)
also accepted extra S3=0, thread C does P(S3) before y=3, thread B does V(S3) after y=y-1

Marker’s feedback
A lot of students forgot to answer the “why” part of each question.
For the last 2 marks, several students came up with a scheme that didn’t guarantee success e.g.:
B: P(S2); y=2; y=y-1; V(S2);
C: x=0; V(S1); V(S2); P(S2); y=3;
only works if V(S2) in C is followed by P(S2) in B, but could continue with P(S2) in C instead.
I gave this kind of thing 1 mark out of 2

END OF EXAMINATION