

**COMP25111 OPERATING SYSTEMS 1 [R. NEVILLE'S AND CHRIS
KIRKHAM'S SECTIONS]
MAIN EXAMINATION 2010/11**

**MARKING SCHEME AND EXAM PERFORMANCE FEEDBACK SECOND
YEAR - SEMESTER 1**

You **must answer** Question 1, which is worth 2 marks for each question in Question 1 of the paper, and **any two** questions from Questions 2, 3 and 4, which are each worth 20 marks.

NOTE [WITH RESPECT TO RN'S QUESTIONS]: DUE TO THE FEEDBACK MECHANISM REQUIRED FOR (UG AND PGT) EXAMS – IN LINE WITH UNIVERSITY POLICY – EACH QUESTION [ANSWER] WILL REFERENCE: 1. LEARNING RESOURCES, 2. BACKGROUND READING [WHICH (1. & 2.) ARE BOTH SPECIFIED IN THE LECTURE], AND 3. LECTURES THEMSELVES IN ORDER TO FEEDBACK THIS INFORMATION TO THE STUDENT – AND HENCE PROVIDE PERTINENT [STUDENT] FEEDBACK.

QUESTION 1

- a) In the context of converting an address generated by a program [a compiler] to the actual address; state:
- i) The names of the two memories involved; &
 - ii) The unit that performs [undertakes] this translation process.

(2 marks)

Bookwork Application

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

In the context of converting an address generated by a program [a compiler] to the actual address:

- i) The names of the two memories involved are:
 - a. Virtual memory; and
 - b. Physical memory.
- ii) The unit that performs [undertakes] this translation process is the:

This process is undertaken by the MMU (Memory management unit).

2 marks for an answer that mentions all the salient facts in a sensible way (**2 marks** for a clear, correct answer, **1 mark** for a 'right lines' solution),

1 ½ marks for correct answer but not detailed [enough],

1 mark for a right-lines approach,

½ marks 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).

TOTAL marks (2 marks) [2]

Marker's feedback

The question assesses lecture 11 learning objective 7; discuss virtual addresses; as the terms 'virtual' and 'physical' are directly related to converting an address generated by a program [a compiler] to the actual address.

The question's answer should clearly evidence knowledge of required salient facts relating to the terms converting an address generated by a program [a compiler] to the actual address.

In the answer [some of] the following terminology (keywords and naming conventions) should be utilised in context, for example: with respect to the names of the two memories involved: virtual, physical, [to be explicit and gain full marks] virtual memory and physical memory; with respect to the unit that performs [undertakes] this translation process, for example: memory management unit.

Main differentiation that must be clearly evidenced in your answer is that: the FULL names of the two memories involved are explicitly stated. These must be stated plainly in the context of your answer to the unit that performs [undertakes] this translation process (for example the MMU) for full marks the full name should be explicitly stated as opposed to the short form (for example abbreviation, acronym, abbreviation, or initials [that] stands for the full naming convention) – which does not infer full comprehension of naming convention required.

Full marks were only awarded if the points above (and in the Example answer) evidence knowledge of required salient facts were explicitly in your answer.

b) Differentiate between multiprogramming and fixed partitions.

(2 marks)

Bookwork Application

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

Multiprogramming:

Multiprogramming has been used in the past; to differentiate from an operating system (OS) running a single program [or Uniprogramming] and one that runs a number of programs concurrently (or multiprogramming).

The OS must first load the multiple programs (into memory [primary {physical} memory]).

The OS will then switch between them [the different programs]; this may be due to the program requiring I/O, or at regular intervals the OS will switch to another of the other programs.

When one of the programs is finished the OS bring in a new one.

Fixed partitions:

Fixed partition divides memory into fixed size blocks.

Fixed partitioning: involved partitioning the available primary memory into a number of regions with each region having a fixed size. The sum of the sizes of all regions [plus that used by the OS itself] equals the size of the primary memory.

2 marks for an answer that mentions the salient facts in a sensible way (**2 marks** for describing two of the above issues, **one mark** if one value identified correctly.),

1 ½ marks for correct answer but not detailed [enough],

1 mark for a right-lines approach,

½ marks 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).

TOTAL marks (2 marks) [4]

Marker's feedback

The question assesses lecture 11 learning objective 4; explain multiprogramming [and fixed partitions]; as the terms 'multiprogramming' and 'fixed' partitions are directly related to the methodology, used in the past, to load multiple programs into memory.

These multiprogrammes are placed in memory; the naming convention used for the area of memory they were placed in is a 'partition.' It is essential in your answer that you made it explicit that these partitions are 'fixed partitions' as opposed to variable [in size]; [variable partitions are discussed further on in the lecture series]...

The question's answer should clearly evidence knowledge of required salient facts relating to the terms multiprogramming and fixed partitions.

In the answer [some of] the following terminology (keywords and naming conventions) should be utilised in context, for example, with respect to Multiprogramming: [different from] uniprogramming, [different from] single program, runs a number of programs, concurrently, programs are loaded [into memory], loaded in to primary [memory], loaded into physical [memory], [the OS] switches [between them], [switches at] regular intervals. With respect to fixed partitions: fixed size, memory divided, partitions fixed in size.

Main differentiation that must be clearly evidenced in your answer is that: the FULL names of the two items involved are explicitly stated. In the context of your answer to the multiprogramming it must be made clear that

it is a technique for regularly [context] switching between multiple programs. Whereas fixed partitions are memory locations that are of fixed size.

Full marks were only awarded if the points above (and in the Example answer) evidence knowledge of required salient facts were explicitly in your answer.

- c) Differentiate between software interrupts and hardware interrupts; in your answer give a brief description of each type of interrupt.

(2 marks)

Application.

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

Basic description of two types of interrupt:

1. Software Interrupt (or Exception)

These occur when the processor meets a special interrupt instruction or when an error has occurred.

For example, executing a program that leads to an overflow in arithmetic calculation will lead to an overflow exception which will then cause some code to run – generally to inform the user of the overflow (a software interrupt is used).

2. Hardware Interrupt

The processor has a number of external connections called interrupt lines, these can be connected to external devices that can signal (by changing the logic level from, say, '0' to '1') the external device wants to interrupt the processor.

2 marks for an answer that gives a concise description of both (**2 marks** for a correct answer, 1 mark for a 'right lines' approach. Moderate marks will be awarded in the case of correct application for a wrongly calculated.),

1 ½ marks for correct answer but not detailed [enough],

1 mark for a right-lines approach,

½ marks for some basic understanding (or attempt).

Reference Learning Resources, Background Reading, and Lecture itself for detailed information; **Lecture(s)** No(s): 16; & Controlling Input and Output 1.

TOTAL marks (2 marks) [6]

Marker's feedback

The question assesses lecture 16 learning objective 1; Explain in simple terms how a processor interacts with an I/O device; as the term 'interrupt' is directly related to explaining how a processor interacts with an I/O device.

In the answer [some of] the following terminology (keywords and naming conventions) should be utilised in context, for example: with respect to interrupts: stop processing, execute another program, and switch to service I/O. In the context of software interrupts: these occur when the processor meets a special interrupt instruction; or an error has occurred. In the context of hardware interrupts: external interrupt lines, interrupt lines connect to processor, external device signal [on interrupt line], level change...

Main differentiation that must be clearly evidenced in your answer is that: the two types of interrupt are initiated by different sources. One is initiated by programs running on the processor, the software interrupt, such as a software exception [Software interrupt]. The other is initiated externally by an external input / output (I/O) device wanting the processors to undertake a task – such as transferring data [Hardware interrupt].

The question's answer should clearly evidence knowledge of required salient facts relating to the terms Software Interrupt (or Exception) and Hardware Interrupt; this was not done explicitly in some of the answers given. If the answers did not detail the differences plainly as stated in the above (and in the Example answer), for example answer gives evidence of knowledge of required salient facts, full marks were not awarded.

- d) In the context of shared memory, state three reasons why it is necessary for processes to share memory.

(2 marks)

Bookwork.

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

In some cases it is necessary for processes to share memory:

- 1) Shared user code;
- 2) Shared data space (for example Unix pipes); or
- 3) Shared Library Code (dll's, that is dynamic link libraries).

2 marks for an answer that mentions all the salient facts in a sensible way (**2 marks** for a reasonable coverage of the salient points, 1 mark for a 'right lines' answer),
1 ½ marks for correct answer but not detailed [enough],
1 mark for a right-lines approach,
½ marks for some basic understanding (or attempt).

Reference Learning Resources, Background Reading, and Lecture itself for detailed information; **Lecture(s)** No(s): 15; Virtual Memory (1).

TOTAL marks (2 marks) [4]

Marker's feedback

The question assesses lecture 15 learning objective 5; State why and how Shared Memory is used; as the term related directly to explaining how a memory is utilised.

In the answer [some of] the following terminology (keywords and naming conventions) should be utilised in context, for example: with respect to Shared Memory: user, code, data, library [code, for example dll's]; these are all explicitly shared.

Main differentiation that must be clearly evidenced in your answer is that: the three types of shared memory are explicitly different types (or sources of item that use memory); for example either code or data.

The question's answer should clearly evidence knowledge of required salient facts relating to the three terms; this was not done explicitly in some of the answers given. If the answers did not detail the three different reasons plainly as stated in the above (and in the Example answer), for example answer gives evidence of knowledge of required salient facts, full marks were not awarded.

- e) In the context of permission information, for example: read (R), write (W), and execute (X). What can the RWX permission information control access to; with respect to security and protection?

(2 marks)

Bookwork:

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

Permission information: like R, W, & X. can be used to control access, for example to:

- 1) Code can be marked 'read only';
- 2) Page tables can be marked 'read only'; or
- 3) Data can be 'read/write' but not 'execute'.

2 marks for an answer that mentions the salient facts in a sensible way (**2 marks** for correct answer, where answer is incorrect 1 mark may be awarded for use of a reasonable method if shown.),

1 ½ marks for correct answer but not detailed [enough],

1 mark for a right-lines approach,

½ marks for some basic understanding (or attempt).

Reference Learning Resources, Background Reading, and Lecture itself for detailed information; **Lecture(s)** No(s): Lectures 15: Virtual Memory (1).

TOTAL marks (2 marks) [4]

Marker's feedback

The question assesses lecture 15 learning objective 6; Assess Security & Protection issues related to memory; as the abbreviation, acronym 'RWX' is directly related to permission information.

In the answer [some of] the following terminology (keywords and naming conventions) should be utilised in context, for example: with respect to access control: code (associated to R {read only}), page table (associated to R {read only}), data (associated to RX {read and write only; not execute}).

Main differentiation that must be clearly evidenced in your answer is that: the three types of permission information are explicitly related to code or data (page tables). One (code) should never be modified; inferring never be written to. The other (data) is utilised by the program (c.f. code) and may be updated; inferring reading and writing is permissible.

The question's answer should clearly evidence knowledge of required salient facts relating to three types; this was not done explicitly in some of the answers given. If the answers did not detail the differences plainly as stated in the above (and in the Example answer); for example answer gives evidence of knowledge of required salient facts, full marks were not awarded.

- f) In a segmented virtual memory system each segment has attributes associated with it; name and briefly describe the attributes.

(2 marks)

Bookwork Application:

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

Each segment has attributes associated with it:

- 1) **Access rights** determine which programs can use the segment (particular users or the operating system); &
- 2) **Usage rights** determine what operations can be performed on the memory (for example read-only, read-execute).

2 marks for a reasonably correct description of both attributes and 1 mark for a 'right lines answer' that notes the processor checks status.

2 marks for an answer that mentions the salient facts in a sensible way.

1 ½ marks for correct answer but not detailed [enough],

1 mark for a right-lines approach,

½ marks for some basic understanding (or attempt).

Reference Learning Resources, Background Reading, and Lecture itself for detailed information; **Lecture(s)** No(s): Lecture 14 Segmented Virtual Memory.

TOTAL marks (2 marks) [10]

OVERALL TOTAL marks (XX marks)

Marker's feedback

The question assesses lecture 14 learning objective 1 & 3; Explain what is meant by segmented virtual memory; and describe how segmented and paged virtual memory systems can be used; as the term 'attributes' is directly related to explaining how a segments may be viewed as [separate] address space [in its own right].

In the answer [some of] the following terminology (keywords and naming conventions) should be utilised in context, for example: with respect to attributes of segments: access and usage.

Main differentiation that must be clearly evidenced in your answer is that: the two types of attributes of segments are: who uses it and how they use it.

The question's answer should clearly evidence knowledge of required salient facts relating to the terms access rights and usage rights; this was not done explicitly in a number of the answers given. If the answers did not detail the differences plainly as stated in the above (and in the Example answer); for example answer gives evidence of knowledge of required salient facts, full marks were not awarded.

UG Exam Performance Feedback Second Year - Semester 1 from CCK

RN: Comments on previous pages for Question 1 a to f and after CCK feedback for Question 4 CCK [Exam Performance Feedback Second Year]:

Q1:

g) No comment

h) Something which most answers missed is that making an instance method synchronized causes it, when called, to acquire the lock on the object which is the target of the call. So there is no lock on the method itself!

i) No comment

j) Too many gave the answer "I", rather than explaining how it was done.

Q2: nearly everyone answered this.

A) No comment

b) Kernel threads do not run in kernel mode while user-level threads run in user mode! Nor do they run faster!

C) No comment

d) (i) and (ii) no comment. (iii) Just saying that the behaviour would change was not enough. Work out what the possible different answers might be!

Q3: not many answered this one.

A) An explanation of tree search was not what this question wanted. Answers getting full marks said more about reading directories and checking permissions.

B) No comment

c) No comment

d) Some thought the 2 pointers to indirect blocks pointed to cnodes instead of block of pointers. This is clearly wrong.

E) A complete algorithm needs to actually read an indirect block from the disk. Many answers failed to express this!

1. g) Key difference is that a System Call changes from user mode to kernel mode. A library routine call does NOT involve a change in protection regime. (1 mark) This is important because there should be no other way for user written code to enter kernel mode, as is needed to do I/O, or change the virtual memory mapping, etc. (1 mark)

h) Synchronized methods have to acquire the lock on the object before they can execute, and release it when they finish. (1 mark). This has the effect of excluding all other synchronized methods (or blocks) operating on the object during execution of the method – so that instance variables can be safely modified for example. (1 mark)

i) Deadlock involves a set of processes. Each process in the set is unable to progress because it requires some non-shared resource – but that resource is currently held by another process in the set. (2 marks)

j) A shell implements a pipe between two commands by creating a process for each command, with the stdout of the first being to a pipe-object (a buffer), and the stdin of the second being from the same pipe-object. (2 marks)
2. a) A process is a program in execution. It has its own address space, and may consist of one or more threads. A thread is a flow-of-control using a virtual processor. (2 marks)

b) User-level threads are something the OS doesn't see – they are implemented by library code in the user process. Kernel-level threads are supported by the OS. (2 marks) So kernel-level threads are scheduled by the OS, and they can be written to block for I/O in the normal way – but they are more expensive to create. (1 mark) User-level threads are cheap to create (no OS interaction), but if one does blocking I/O the whole process is blocked. (1 mark).

c) A semaphore is a synchronisation primitive. It has a non-negative integer value (general semaphore), and the operations on it are P and V. (1 mark)

P(S) decrements S if it is +ve, and continues. Otherwise it waits for S to become +ve before decrementing it. This activity is atomic! (1 mark)

V(S) increments S atomically. (1 mark).

d) i) x is 12. (2 marks) Initially Thread-3 is the only thread able to complete its P operation. V(S2) doesn't allow either of the others to proceed, so x is set to -1. Then a second V(S2) allows Thread-2 to execute. It sets x to +4 before releasing Thread-1. The final V(S2) in Thread-3 ensures that Thread-1 can proceed, and x becomes +12. (2 marks)

ii) The semaphores return to their initial values, namely 0, 0 and 1 respectively. (3 marks)

e) All that would happen is that S2 could return to 0 before Thread-2 completed – but that doesn't matter. The values reported above would NOT change. Thread-1 would still be delayed until Thread-2 completed, but would then proceed as before. We would not get deadlock!
3. a) Start at the root directory. Take the first component of the filename (that follows the '/'); look it up in the directory. If this is not the last component in the pathname, check that it is a directory (and that the user has permission), and make it the current directory for this algorithm. Repeat with successive components of the pathname. When the last has been used to identify a file (which may be a directory), that is the file. (4 marks)

b) For each block in the filestore, there is an entry in the FAT giving the number of the next block in the file. If this is the last block in the file the FAT entry will instead indicate that – and possibly how many bytes in the block are in use. So the whole file can be accessed by following this chain within the FAT. (4 marks)

c) The size of the FAT is determined by the size of the disk partition and the number of bytes needed for a block number. This can be quite large. Unless all files are small, inodes will use less space than FATs. Can chose how much space to allocate to inodes. (1 mark) Lack of chaining makes it easier to do direct access with inodes. (1 mark) However small files cause wasted space in an inode. (1 mark) If an inode is corrupted, it affects that file. If a small part of the FAT becomes corrupted it can affect many! (1 mark)

d) A block can contain $b/4$ pointers, where b is the blocksize in bytes.

So a cnode file will have as maximum size $(6 + 2 \cdot b/4)$ blocks, i.e. $6b + b^2/2$ bytes. (4 marks)

e) Here assume $i \geq 1$, but an answer which starts at 0 should also gain full marks. (i.e. 4 marks)

```

    if (i <= 6) return cnode[i-1] ;
if (cnode[6] == 0) return 0 ;
    if (i < (6 + b/4) {
        read block at cnode[6] into int [] temp ;
        return temp[i-7] ;
    }
    if (cnode[7] == 0) return 0 ;
    if (i >= (6 + b/2)) return 0 ;
    read block at cnode[7] into int [] temp ;
    return temp[i-7-b/4] ;

```

Lose a mark for failing to test indirect pointers against zero.

Lose a mark for reading 1st indirect pointer block unnecessarily.

QUESTION 4 (20 MARKS IN TOTAL)

- a) Direct memory access (DMA) is interrupt driven. Given that a processor writes to a disk, utilizing DMA, describe the four-step DMA process for writing data.

(4 marks)

Bookwork (4 marks)

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

A processor writes to a disk; the process can be speeded up utilising direct memory access (DMA).

The 4-steps DMA process is:

- 1) Processor inform the disk DMA I/O - to write data by writing to a command register in the DMA I/O device;
- 2) DMA controller starts write process;
- 3) Process gets on with another work [process] until;
- 4) DMA device is finished; which is signalled by an interrupt.

4 marks for a concise description of all the four,

2 marks for two,

1 mark for one,

½ marks for some basic understanding (or attempt).

Reference Learning Resources, Background Reading, and Lecture itself for detailed information; **Lecture(s)** No(s): 17; Controlling Input and Output 2.

TOTAL marks (4 marks) [4]

Marker's feedback

The question assesses lecture 17 learning objective 3; Explain what is meant by direct memory access and how it is useful in I/O; as the terms 'direct memory access' are directly related to how a read or write access that does not use the processor is undertaken; using DMA.

In the answer [some of] the following terminology (keywords and naming conventions) should be utilised in context, for example: with respect to DMA: inform (DMA device to write), command register, (start) write, [processor does other work, finish [DMA], (signal with) interrupt...

Main differentiation that must be clearly evidenced in your answer is you list the four steps explicitly: the four steps must be clearly delineated; your description of each step must include terms like: inform [DMA (controller) to write data]; [DMA] starts [write process], [processor] gets on with other process; [DMA] finishes; [signal by] interrupting CPU.

The question's answer should clearly evidence knowledge of required salient facts relating to the steps involved; this was not done explicitly in some of the answers given – as the terminology (keywords and naming conventions) given above was not used. If the answers did not detail the differences [for each step] plainly as stated in the above (and in the Example answer); for example answer gives evidence of knowledge of required salient facts, full marks were not awarded.

- b) A CPU instructs a hard disk that utilises DMA to write data to disk.

- (i) Name and briefly describe the two registers that are normally used during the DMA process.

(1 mark)

Bookwork (2 marks).

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

- (i) The disk I/O has two registers:

1. Command register where commands such as: READ, WRITE, FORMAT can be written;
2. Status register that indicates whether the disk is ready for the data or not.

2 marks, 1 for names and 1 for descriptions.

Reference Learning Resources, Background Reading, and Lecture itself for detailed information; **Lecture(s)** No(s): 17; Controlling Input and Output 2.

TOTAL marks (1 marks) [5]

Marker's feedback

The question assesses lecture 17 learning objective 3; Explain what is meant by direct memory access and how it is useful in I/O; as the terms 'DMA' are directly related to explaining how a I/O device [the DMA controller] interacts with the processor – via the DMA's command and status registers – is directly aligned to explain what is meant by direct memory access – in the context of explain how the I/O device utilizes its registers.

In the answer [some of] the following terminology (keywords and naming conventions) should be utilised in context, for example: Command register, READ, WRITE, FORMAT; and Status register, device ready or not...

Main differentiation that must be clearly evidenced in your answer is that: the two registers perform different functions – these must be explicitly explained.

The question's answer should clearly evidence knowledge of required salient facts relating to the terms command register and status register; this was not done explicitly in some of the answers given. If the answers did not detail the differences plainly as stated in the above (and in the Example answer); e.g. answer gives evidence of knowledge of required salient facts, full marks were not awarded.

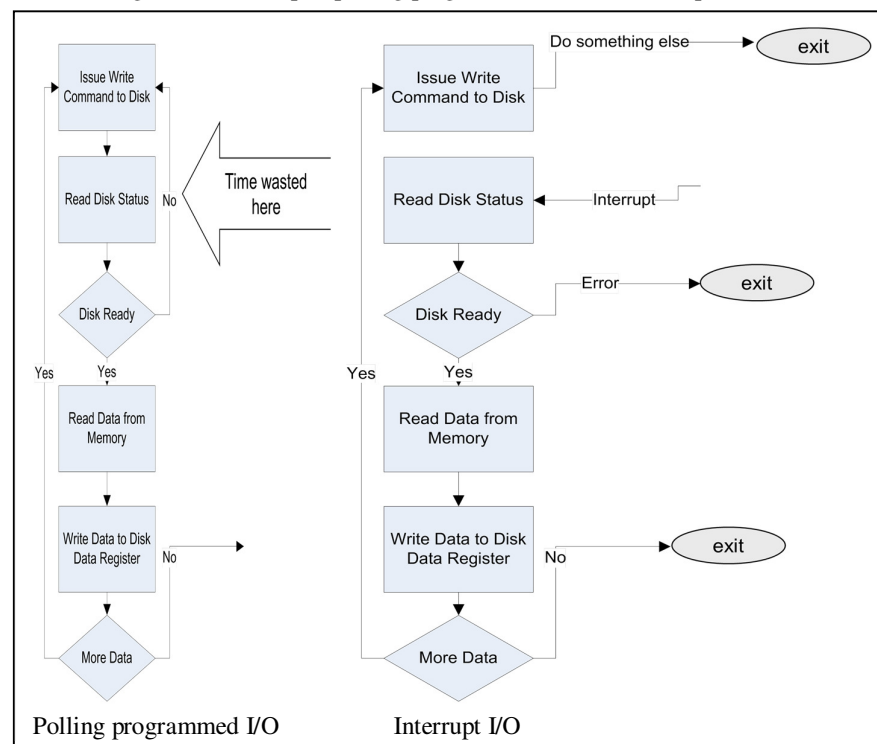
- (ii) Transfer of data to a disk can actually be undertaken by programmed input/output (I/O) or interrupt I/O. Draw two flow charts [diagrams] that depict the sequence of events needed to perform disk writing using polling [programmed] I/O and interrupt I/O.

(4 mark)

Bookwork (4 marks).

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

- (ii) The two diagrams below depict polling programmed I/O and interrupt I/O:



4 marks for an answer that depicts all steps in both diagrams.

2 marks for correct answer but not detailed [enough],

1 mark for a right-lines approach,

½ marks for some basic understanding (or attempt).

Reference Learning Resources, Background Reading, and Lecture itself for detailed information; **Lecture(s)** No(s): 17; Controlling Input and Output 2.

TOTAL marks (4 marks) [9]

Marker's feedback

The question assesses lecture 17 learning objective 2; Discuss what is meant by interrupt-driven I/O; as the terms 'interrupt' is directly related to explaining how a processor interacts with an I/O device; or [indeed] polling.

In the answer [some of] the following terminology (keywords and naming conventions) should be utilised in context, for example: with respect to polling [programmed I/O]: Issue write [command to disk], [read disk] status, [disk] ready, read data [from memory], write data [to disk data register], data register. In the context of interrupt I/O: issue write [command to disk], [read disk] status, [disk] ready, read data [from memory], write data [to disk], data register...

Main differentiation that must be clearly evidenced in your answer is that: the two types of I/O are that only the interrupt I/O exits from the flow chart [in three different places] and the processor is interrupted by the I/O device to tell the processor that it is ready to start the write process.

The question's answer should clearly evidence knowledge of required salient facts relating to the terms polling and interrupt I/O; this was not done explicitly in some of the answers given. If the answers did not detail the differences plainly [in two different flow charts] as stated in the above (and in the Example answer); e.g. answer gives evidence of knowledge of required salient facts, full marks were not awarded.

(iii) Give a description of DMA. In your explanation address the following issues:

- a) If DMA can transfer data in two directions; (1 mark)
- b) The extent to which the processor intervenes. (1 mark)

Bookwork (2 marks).

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

(iii) Description of DMA; explanation also address the following issues:

- a) If DMA can transfer data in two directions:
A DMA device is capable of reading from or writing directly to memory in the same way as a processor does. (1 mark)
- b) The extent to which the processor intervenes:
The DMA will handle the transfer of a whole block of data **without** processor intervention. (1 mark)

2 marks, 1 mark for each and ½ mark for basic understanding.

Reference Learning Resources, Background Reading, and Lecture itself for detailed information; **Lecture(s)** No(s): 17; Controlling Input and Output 2.

TOTAL marks (2 marks) [11]

Marker's feedback

The question assesses lecture 17 learning objective 1; Explain what is meant by direct memory access (DMA) and how it is useful in I/O; as the term 'DMA' is directly related to explaining how a processor interacts with an I/O device – in cases where the processor only initiates the read or write process.

In the answer [some of] the following terminology (keywords and naming conventions) should be utilised in context, for example: DMA can transfer data in two directions: read, write, directly [to memory]. In the context of the extent to which the processor intervenes: none, without [processor intervention, reading & writing, DMA handles transfer...

Main differentiation that must be clearly evidenced in your answer is that: the two issues are that: i) yes, it can transfer in both directions [read & write]. ii) The second issue is clearly highlighting the processor leaves the DMA to undertake the data transfer – without processor intervention – in the explicit transfer of data.

The question's answer should clearly evidence knowledge of required salient facts relating to the two issues; this was not done explicitly in some of the answers given. If the answers did not detail the differences plainly as stated in the above (and in the Example answer); e.g. answer gives evidence of knowledge of required salient facts, full marks were not awarded.

(iv) Briefly compare and contrast interrupt-driven I/O and programmed I/O. In your critique address the following issues:

- a. Which is more efficient? (1 marks)

- b. What they both rely on to transfer data? (1 marks)
- c. What limits the processor with respect to the transfer of data? (1 marks)
- d. What technique should be used for large volumes of data transfer? (1 marks)

Critique (4 marks).

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

- (iv) Briefly compare and contrast interrupt-driven I/O and programmed I/O.

Briefly compare and contrast interrupt-driven I/O and programmed I/O. In your critique address the following issues:

- a. Which is more efficient?
Interrupt-driven I/O is more efficient than programmed I/O, less time is wasted. (1 marks)
- b. What they both rely on to transfer data?
They both schemes rely on the processor to transfer the data. (1 marks)
- c. What limits the processor with respect to the transfer of data?
Data rate is limited by the processor's ability to read/write to/from the I/O device. (1 marks)
- d. What technique should be used for large volumes of data transfer?
For large volumes of data (e.g. disks) Direct Memory Access is performed. (1 marks)

4 marks [in total] *for an answer that cover all four issues; in correct context.*

2 marks *for an answer that a moderate description of each issue.*

Reference Learning Resources, Background Reading, and Lecture itself for detailed information; **Lecture(s)** No(s).: 17; Controlling Input and Output 2.

TOTAL marks (4 marks) [15]

Marker's feedback

The question assesses lecture 17 learning objective 2; Discuss what is meant by interrupt-driven I/O; as the terms 'interrupt driven I/O' and 'polled I/O' or programmed I/O are directly related to explaining how a processor interacts with an I/O device.

In the answer [some of] the following terminology (keywords and naming conventions) should be utilised in context, for example: a. [Interrupt-driven I/O] more efficient; b. both [rely on the processor], [to] transfer data; c. [Data rate] limited (by the processor's ability to read/write to/from the I/O device) – e.g. bus speed; & d. [large volumes of data] Direct Memory Access is performed....

Main differentiation that must be clearly evidenced in your answers is that: the four issues are addressed concisely with the appropriate [correct] supporting facts.

The question's answer should clearly evidence knowledge of required salient facts relating to the four issues; this was not done explicitly in some of the answers given. If the answers did not detail the differences plainly as stated in the above (and in the Example answer); e.g. answer gives evidence of knowledge of required salient facts, full marks were not awarded.

- c) Given an 8G address spaces and associated 256K page sizes; calculate the number of pages that result in the *virtual address space*. (1 marks)

Application (1 marks).

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

Virtual address space 8GB and page size 256KB.

If the virtual address space is 8 GB and the page size is 256 KB there are:

$$\begin{aligned}\text{Number of Pages} &= \frac{\text{Address space}}{\text{Page size}} \\ &= \frac{8 \text{ GB}}{256 \text{ KB}} = \frac{2^{33}}{2^{18}} = \frac{8,589,934,592.00}{262,144} = 32k (256K) \text{ pages.} \\ &[\text{subtract powers } 33-18=15; 2^{15} = 32,768 \approx 32k]\end{aligned}$$

1 marks for an answer that calculates the correct answer and is laid out correctly (**2 marks** for a correct answer, 1 mark for a 'right lines' approach. Moderate marks will be awarded in the case of correct application for a wrongly calculated answer.),
½ marks for some basic understanding (or attempt).

Reference Learning Resources, Background Reading, and Lecture itself for detailed information; **Lecture(s)** No(s): 13; Virtual Memory (1).

TOTAL marks (1 marks) [16]

Marker's feedback

The question assesses lecture 13 learning objective 1; Explain what is meant by a paged virtual memory system; as the calculation of 'number of pages' is directly related to comprehension of what is meant by a paged virtual memory system – with respect to address space [size] and page size.

In the answer [some of] the following terminology (keywords and naming conventions) should be utilised in context, for example: with respect to number of pages: address space, page size, and pages [as the units for the answer].

Main differentiation that must be clearly evidenced in your answer is that: the sequence of mathematical steps is clearly (and explicitly) delineated in the answer. There are at least four steps that should be clearly enumerated: 1) Write down the full equation (in English – no abbreviations); 2) Substitute the correct values in the equation (nominally in GBs and KB); 3) Translate the GB & KB to powers of 2 [NOTE 2) and 3) any be in any order 3) then 2) or 2) then 3)]; 4) Finally, calculate the number of pages. Nominally, the answer is written as "Number of" (size of actual pages) "pages"; e.g. *32k (256K) pages*.

The question's answer should clearly evidence knowledge [in this case applied mathematical process] of required salient facts relating to the terms the explicit calculation process used to sequence through the steps of calculating the number of pages. If the answers did not detail the explicit sequence of calculations plainly as stated in the above (and in the Example answer); e.g. answer gives evidence of knowledge of required salient facts, full marks were not awarded. NOTE: The alternative rationale for writing down a full and concise description of the mathematical process is that if any mistakes were made the marker can still award some marks – however if no working is shown [just the answer] this leaves the marker no leeway to allocate any marks.

- d) Given a physical address size of 2G and associated 128K block size below. Calculate the number of page frames in the physical address space.

(1 marks)

Application (1 marks).

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

Given a block [page] size of 128 KB and a physical address space of 2 GB.

If the virtual address space is 2 GB and the block size is 128 KB there are:

$$\begin{aligned}\text{Number of Page frames} &= \frac{\text{Address space}}{\text{Block size}} \\ &= \frac{2 \text{ GB}}{128 \text{ KB}} = \frac{2^{31}}{2^{17}} = \frac{2,147,483,648}{131,072} = 16k (128K) \text{ page frames.} \\ &[\text{Subtract powers } 31-17=14; 2^{14} = 16,384 \approx 16k]\end{aligned}$$

1 marks for an answer that calculates the correct answer and is laid out correctly (**2 marks** for a correct answer, 1 mark for a 'right lines' approach. Moderate marks will be awarded in the case of correct application for a wrongly calculated.),
½ marks for some basic understanding (or attempt).

Reference Learning Resources, Background Reading, and Lecture itself for detailed information; **Lecture(s)** No(s).: 13; Virtual Memory (1).

TOTAL marks (1 marks) [17]

Marker's feedback

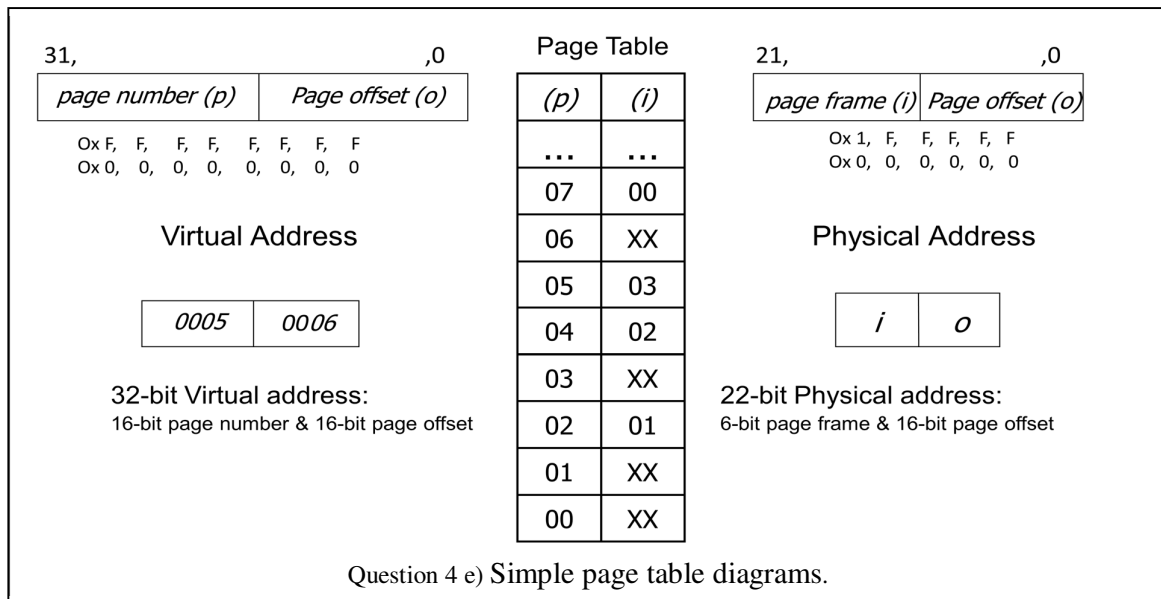
The question assesses lecture 13 learning objective 1; Explain what is meant by a paged virtual memory system; as the calculation of 'number of page frames' is directly related to comprehension of what is meant by a paged virtual memory system – with respect to address space [size] and block size.

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

Main differentiation that must be clearly evidenced in your answer is that: the sequence of mathematical steps is clearly (and explicitly) delineated in the answer. There are at least four steps that should be clearly enumerated: 1) Write down the full equation (in English – no abbreviations); 2) Substitute the correct values in the equation (nominally in GBs and KB); 3) Translate the GB & KB to powers of 2 [NOTE 2) and 3) any be in any order 3) then 2) or 2) then 3)]; 4) Finally, calculate the number of pages. Nominally, the answer is written as "Number of" (size of page frames) "page frames"; e.g. *32k (256K) page frames*.

The question's answer should clearly evidence knowledge [in this case applied mathematical process] of required salient facts relating to the terms the explicit calculation process used to sequence through the steps of calculating the number of pages. If the answers did not detail the explicit sequence of calculations plainly as stated in the above (and in the Example answer); e.g. answer gives evidence of knowledge of required salient facts, full marks were not awarded. NOTE: The alternative rationale for writing down a full and concise description of the mathematical process is that if any mistakes were made the marker can still award some marks – however if no working is shown [just the answer] this leaves the marker no leeway to allocate any marks.

- e) Given the simple page table diagram below, and the virtual address 0x00050006.
 - i) State the physical address given the data in the page table; (1 mark)
 - ii) Then give a brief description of the procedure the paged virtual memory utilises to generate a physical memory address from a virtual memory address. (1 mark)



Application (1 marks) & Bookwork (1 marks).

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

- The physical address (Ox i o) produced from the page table is:
Ox i o = Ox 03 0006;
(1 mark)
- Then give a brief description of the procedure the paged virtual memory utilises to generate a physical memory address from a virtual memory address is [These could even be viewed as steps]:
 - [Step 1] The processor generates a logical address.
 - [Step 2] The page number field is used by the MMU to look to see whether the page is in memory or not.
 - [Step 3] If it is in memory, a physical address is computed by replacing the page number with the page frame number of where the page can be found. Together with the offset this is used as a physical address to memory.
 - [Step 4] If it is not in memory, the transfer is aborted (page fault) and the operating system will load the page from disk to memory.
 (2 mark)

3 marks for an answer that mentions all the salient facts in a sensible way (**2 marks** for correct answer (not all salient facts mentioned), where answer is incorrect 1 mark may be awarded for use of a reasonable method if shown.),

1 ½ marks for correct answer but not detailed [enough],

1 mark for a right-lines approach,

½ marks for some basic understanding (or attempt).

Reference Learning Resources, Background Reading, and Lecture itself for detailed information; **Lecture(s)** No(s).: 13; Virtual Memory (1).

TOTAL marks (3 marks) [20]

OVERALL TOTAL marks (20 marks)

Marker's feedback

The question assesses lecture 13 learning objectives 1, 2, & 3 [each to some degree]; Explain what is meant by a paged virtual memory system; Determine the structure of an address in a paged virtual memory system; & Establish the outcome of memory references (specified by address) for a paged virtual memory system with a specific page table; as the terms 'physical address,' & 'generating a physical memory address,' are directly related to explaining how to calculate the physical page address and explicitly explain the process.

In the answer [some of] the following terminology (keywords and naming conventions) should be utilised in context, for example: with respect to calculating [looking up] the page frame number in the page table: page number, logical address, page table, page frame number, relative address [within the frame]. In the context of the process of generating the page frame [given the page number and relative address]: [processor] generate logical address, [check] page number [in memory], [if so] compute page frame [number], [if not] throw page fault, [then] load page [from memory].

Main differentiation that must be clearly evidenced in your answer is that: i) the diagram and text [diagrammatic and textual answer] states the correct page frame – meta data that would be helpful to you is the use of symbols: 'p' (page number), 'o' (page offset), 'i' (page iframe). The other [in respect to a full, concise description of the process] is to state all four steps utilising all [in the correct context] the correct terminology (keywords and naming conventions).

The question's answer should clearly evidence knowledge of required salient facts relating to the calculation of the page frame (and page offset) and the process; this was not done explicitly in some of the answers given. If the answers did not detail the differences plainly as stated in the above (and in the Example answer); e.g. answer gives evidence of knowledge of required salient facts, full marks were not awarded.

END OF EXAMINATION