SAMPLE EXAM PAPERS SOLUTIONS

Damian Gordon

Rough Work

Exam is two hours long

I need to answer three questions

So that's three questions in 120 minutes.

I'll give myself 10 minutes to read the paper at the start, and do calculations on how long things will take

I'll give myself 20 minutes to re-read my answers at the end

So that's 90 minutes to answer three questions

That's really 30 minutes per question.

Each question is 33 marks, so that's about 55 seconds per mark.

Question 1. Rough Work

Q1(a) = 10 marks

Q1(b) = 6 marks

Q1(c) = 6 marks

Q1(d) = 12 marks

So roughly

Q1(a) = 9 minutes

Q1(b) = 5 minutes

Q1(c) = 5 minutes

Q1(d) = 11 minutes

Timing

Exam starts at 16:00

Reading until 16:10

Q1(a) 16:19

Q1(b) 16:24

Q1(c) 16:29

Q1(d) 16:40

MUST finish Question 1 by 16:40

Question 1. Rough Work

Q. 1(a) Von Neumann architecture – programs and data in same memory, and mention Von Neumann bottleneck

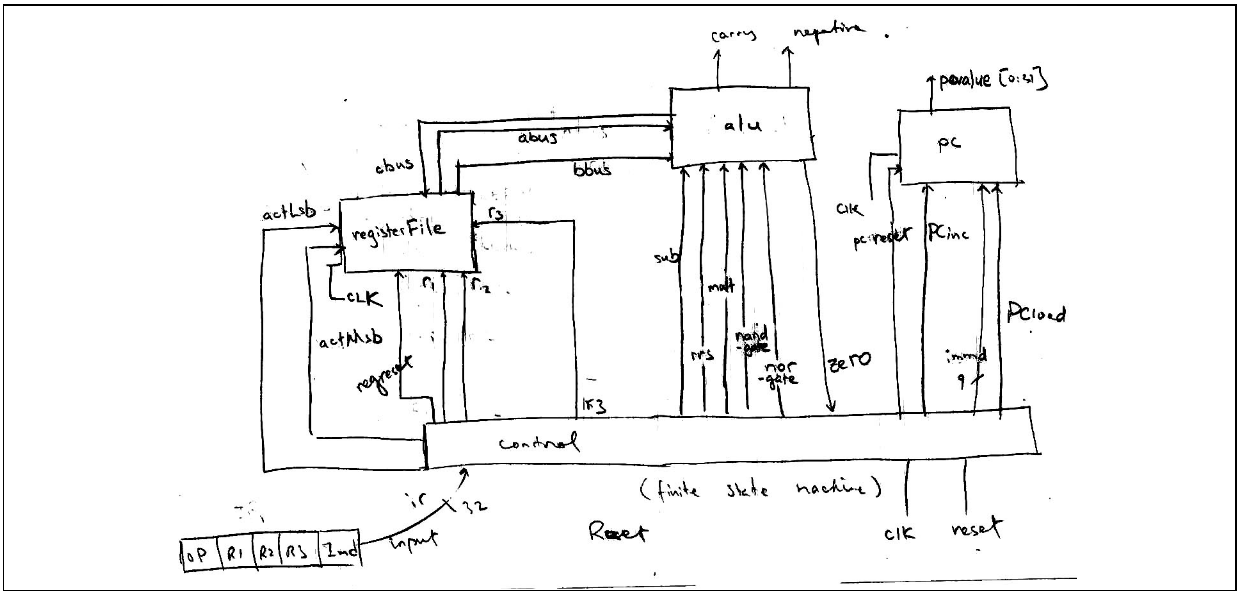
Q. 1(b) Instruction Cycle – Fetch Decode Execute - Fetch Decode INCREMENT Execute LOOP

Q1. (c) Assember Languages - low-level programming language, specific to a particular computer architecture, assembler. Advs. 1.structure 2.easier to understand 3.easier to correct errors 4.same efficiency of execution as machine level.

Q1. (d) LMC Instructions to add two numbers – get first, store, get second, add two together, output. Halt.

Q. 1(a) Von Neumann architecture

The Von Neumann architecture is a computer architecture model that describes a design architecture for an electronic digital computer with parts consisting of a processing unit containing an arithmetic logic unit and processor registers, a control unit containing an instruction register and program counter, a memory to store both data and instructions, external mass storage, and input and output mechanisms.



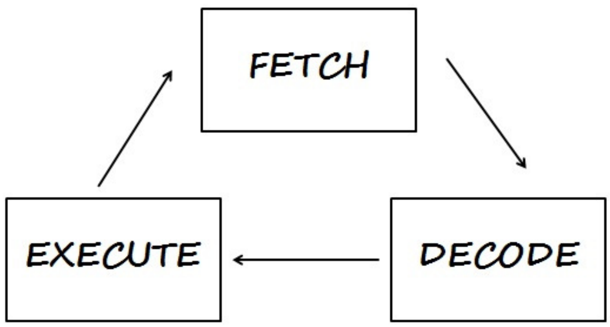
Von Neumann bottleneck

The meaning has evolved to be any stored-program computer in which an instruction fetch and a data operation cannot occur at the same time because they share a common bus. This is referred to as the Von Neumann bottleneck and often limits the performance of the system.

Q. 1(b) Instruction Cycle

The instruction cycle, or Fetch-Decode-Execute (FDE) cycle, is used in Stored-Program Computers since the separation of storage from the processing unit is implicit in this model.

1. **Fetch** the next instruction from memory at the address in the program counter
2. **Decode** the instruction using the control unit
3. Increment the Program Counter
4. The control unit commands the rest of the computer to **execute** the instruction
5. Go to step 1



Q1. (c) Assember Languages

An assember language is a low-level programming language for a computer, or other programmable device, in which there is a very strong (generally one-to-one) correspondence between the language and the architecture's machine code instructions. Each assembly language is specific to a particular computer architecture, in contrast to most high-level programming languages, which are generally portable across multiple architectures, but require interpreting or compiling. Assembly language is converted into executable machine code by a utility program referred to as an assembler; the conversion process is referred to as assembly, or assembling the code.

Advantages over Machine Code

1. It creates programming structure.
2. It's easier to understand and saves a lot of time and effort of the programmer.
3. It is easier to correct errors and modify program instructions.
4. Assembly Language has the same efficiency of execution as the machine level language. Because this is one-to-one translator between assembly language program and its corresponding machine language program.

Q1. (d) LMC Instructions to add two numbers

PseudoCode (I'm not asked for PseudoCode, but it helps me)

PROGRAM AddNumbers:

Get First Number into calculator;

Store First Number in memory;

Get Second Number into calculator;

Add First Number and Second Number;

Output Answer;

END.

901	Get first input
308	Put that input into memory location 8
901	Get next input
108	Add the value from memory location 8 to the calculator
902	Output the answer
000	Halt

Question 2. Rough Work

Q2.(a)(i) Magnetic Tape = 4 marks (3.5 minutes)

Q2.(a)(ii) Magnetic Disk Storage = 4 marks (3.5 minutes)

Q2.(a)(iii) CD-Rewritable Storage = 4 marks (3.5 minutes)

Q2.(a)(iv) Flash Memory = 4 marks (3.5 minutes)

Q2.(a) = 14 minutes

Q2.(b) Disk Access Speed = 5 marks (4 minutes)

Q2.(b) = 4 minutes

Q2.(c)(i) Field = 2 marks (1.5 minutes)

Q2.(c)(ii) Record = 2 marks (1.5 minutes)

Q2.(c)(iii) Field = 2 marks (1.5 minutes)

Q2.(c) = 4.5 minutes

Q2.(d) Physical File Organisation = 6 marks (5 minutes)

Q2.(d) = 5 minutes

Question 2. Rough Work

Q2.(a)(i) Magnetic Tape – plastic strip, tape drive, sticky-shed syndrome.

Q2.(a)(ii) Magnetic Disk Storage - magnetized disk, non-volatile memory, read-and-write heads.

Q2.(a)(iii) CD-Rewritable Storage - CD-RW, written, read arbitrarily many times, laser optics, blanked.

Q2.(a)(iv) Flash Memory - non-volatile, based on EEPROM, NAND and NOR logic gates. NAND in blocks or pages, NOR in bytes.

Q2.(b) Disk Access Speed - Access time, Seek time, Rotational latency, Command processing time, Settle time

Q2.(c)(i) Field – part of a record

Q2.(c)(ii) Record – part of a file

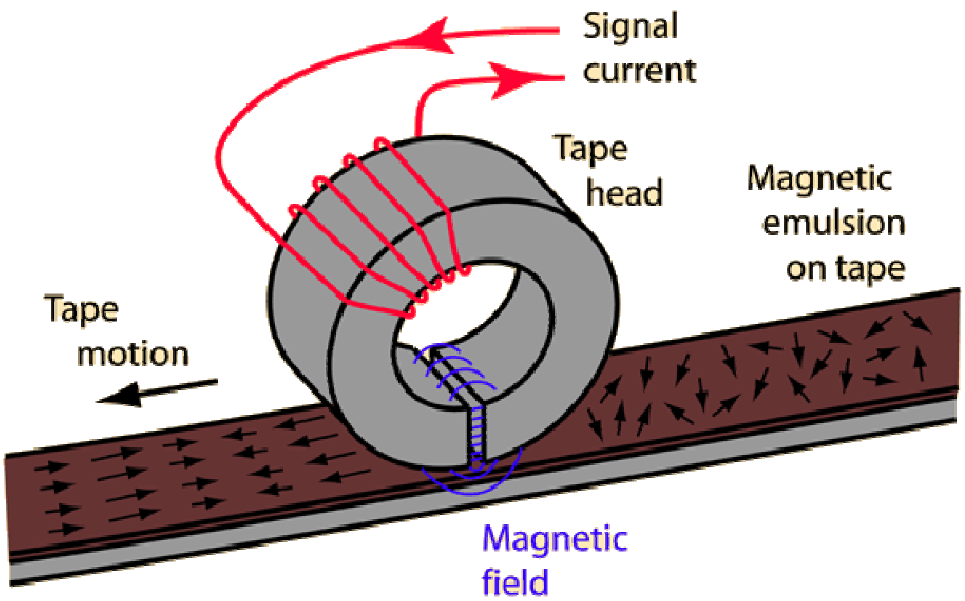
Q2.(c)(iii) File – durable storage

Q2.(d) Physical File Organisation - Sequential File Organization, Heap File Organization.

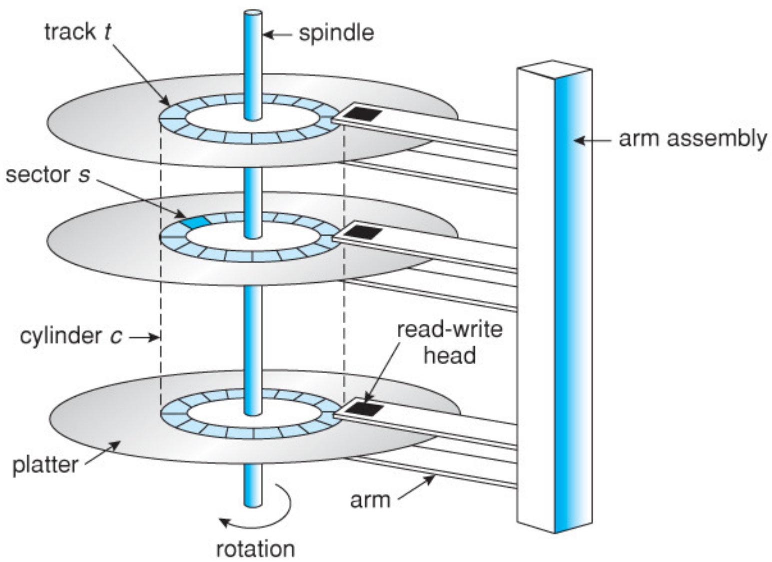
Question 2

Q2. (a) Data Storage Devices

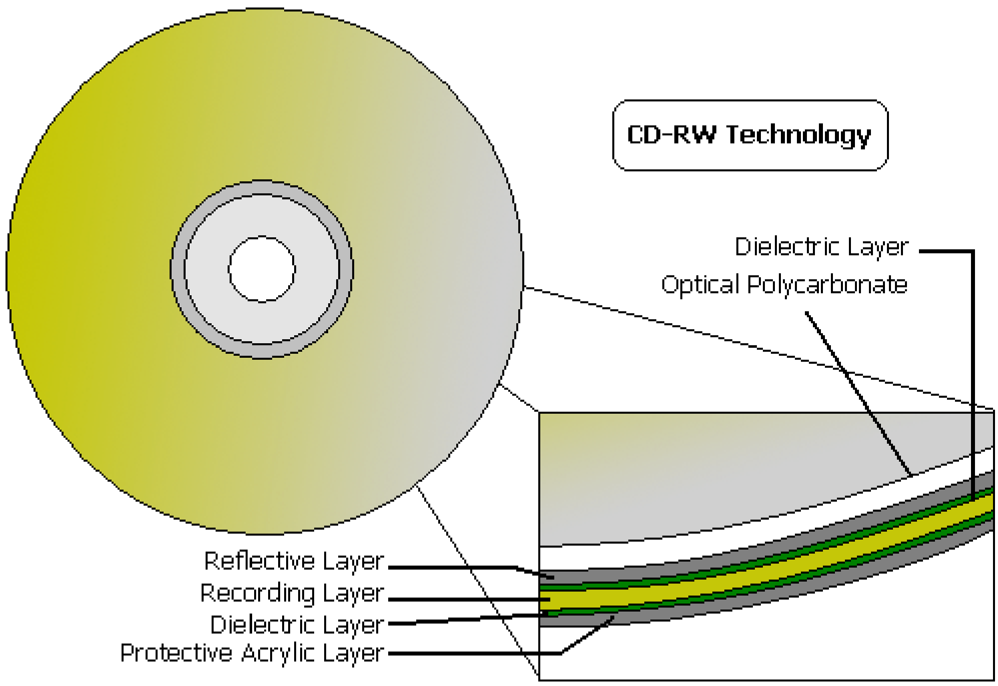
Q2.(a)(i) Magnetic Tape is a medium for magnetic recording, made of a thin magnetizable coating on a long, narrow strip of plastic film. It was developed in Germany, based on magnetic wire recording. Devices that record and play back audio and video using magnetic tape are tape recorders and video tape recorders. A device that stores computer data on magnetic tape is a tape drive (tape unit, streamer). Over years, magnetic tape can suffer from deterioration called sticky-shed syndrome. Caused by absorption of moisture into the binder of the tape, it can render the tape unusable.



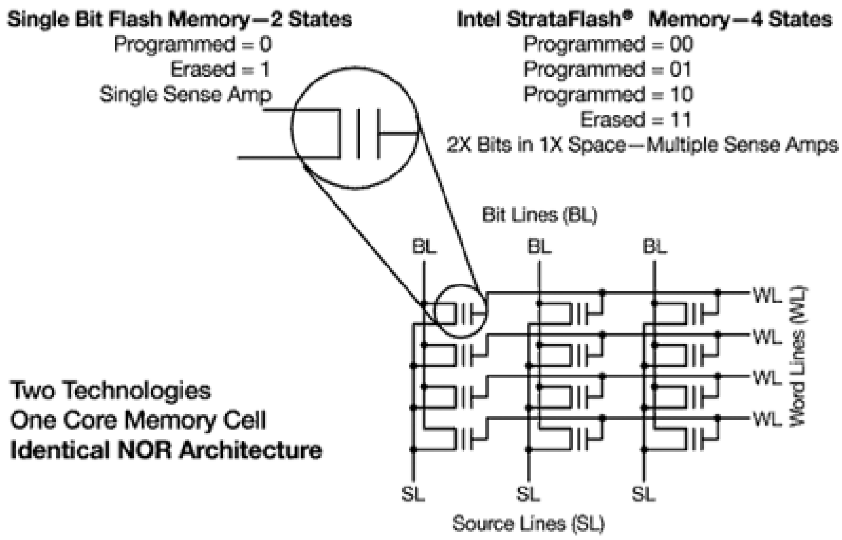
Q2.(a)(ii) Magnetic Disk Storage is the storage of data on a magnetized disk. It uses different patterns of magnetization in a magnetizable material to store data and is a form of non-volatile memory. The information is accessed using one or more read-and-write heads. It is widely used to store computer data as well as audio and video signals. The read-and-write head is used to detect and modify the magnetization of the material immediately under it. There are two magnetic polarities, each of which is used to represent either 0 or 1.



Q2.(a)(iii) CD-Rewritable Storage (CD-RW) is a digital optical disc storage format. A CD-RW disc is a compact disc that can be written, read arbitrarily many times, erased and written again. CD-RW discs (CD-RWs) require readers that have more sensitive laser optics than plain CDs. Consequently, CD-RWs cannot be read in many CD readers built prior to the introduction of CD-RW. CD-ROM drives that bear a "MultiRead" certification claim compatibility. CD-RW discs need to be blanked before reuse.



Q2.(a)(iv) Flash Memory is an electronic non-volatile computer storage medium that can be electrically erased and reprogrammed. Developed from EEPROM (electrically erasable programmable read-only memory). There are two main types of flash memory, which are named after the NAND and NOR logic gates. The internal characteristics of the individual flash memory cells exhibit characteristics similar to those of the corresponding gates. Whereas EPROMs had to be completely erased before being rewritten, NAND type flash memory may be written and read in blocks (or pages) which are generally much smaller than the entire device. NOR type flash allows a single machine word (byte) to be written—to an erased location—or read independently.



Q2.(b) Disk Access Speed

The factors affecting magnetic disk data access speed, include:

The ACCESS TIME or response time of a rotating drive is a measure of the time it takes before the drive can actually transfer data. The factors that control this time on a rotating drive are mostly related to the mechanical nature of the rotating disks and moving heads.

With rotating drives, the SEEK TIME measures the time it takes the head assembly on the actuator arm to travel to the track of the disk where the data will be read or written.

ROTATIONAL LATENCY (sometimes called rotational delay or just latency) is the delay waiting for the rotation of the disk to bring the required disk sector under the read-write head.

The COMMAND PROCESSING TIME or command overhead is the time it takes for the drive electronics to set up the necessary communication between the various components in the device so it can read or write the data.

The SETTLE TIME is the time it takes the heads to settle on the target track and stop vibrating so they do not read or write off track. This time is usually very small, typically less than 0.1 ms, and modern HDD manufacturers account for it in their seek time specifications.

Q2.(c)(i) Field

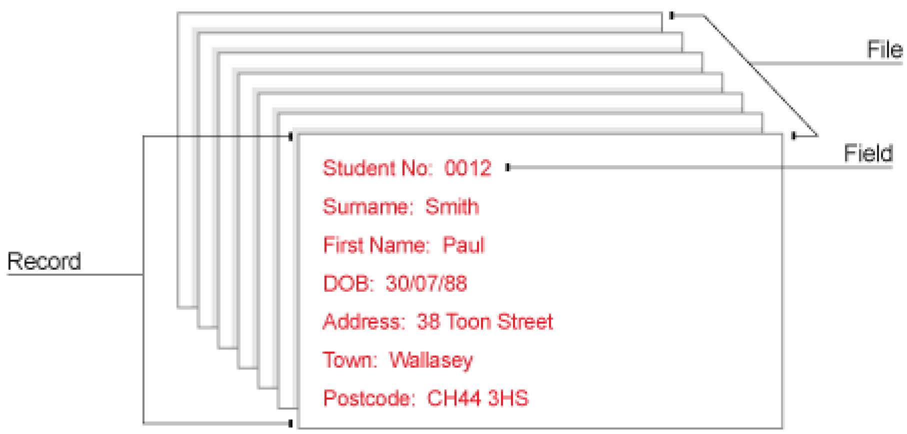
Data that has several parts, known as a record, can be divided into fields. Relational databases arrange data as sets of database records, also called rows. Each record consists of several fields; the fields of all records form the columns.

Q2.(c)(ii) Record

A record (also called struct or compound data) is a basic data structure. A record is a collection of elements, typically in fixed number and sequence and typically indexed by serial numbers or identity numbers.

Q2.(c)(iii) File

A file is a resource for storing information, which is available to a computer program and is usually based on some kind of durable storage. A file is "durable" in the sense that it remains available for other programs to use after the program that created it has finished executing.



Q2.(d) Physical File Organisation

Two methods of physical file organization on a magnetic disk:

Sequential File Organization: Every file record contains a data field (attribute) to uniquely identify that record. In sequential file organization mechanism, records are placed in the file in the some sequential order based on the unique key field or search key. Practically, it is not possible to store all the records sequentially in physical form.

Heap File Organization: When a file is created using Heap File Organization mechanism, the Operating Systems allocates memory area to that file without any further accounting details. File records can be placed anywhere in that memory area. It is the responsibility of software to manage the records. Heap File does not support any ordering, sequencing or indexing on its own.

Question 3. Rough Work

Q3.(a) 3 pre-emptive scheduling policies = 9 marks (8 minutes)

Q3.(b) Unix and NT access control for users = 10 marks (9 minutes)

Q3.(c) Process Control Block = 5 marks (4 minutes)

Q3.(d) Principle of Locality = 5 marks (4 minutes)

Q3.(e) Two Critical Regions = 4 marks (3 minutes)

Q3.(a) 3 pre-emptive scheduling policies - Shortest Remaining Time (SRT), Round Robin, Multi-Level Queues.

Q3.(b) Unix and NT access control for users- Unix (r, w, x) NT (r, w, x, d, p, o)

Q3.(c) Process Control Block - ProcessID, Process Status, Process State (PC, Reg, Main Mem, Resources, Priority), Accounting.

Q3.(d) Principle of Locality - the same value, or related storage locations, being frequently accessed. 2 Examples – loops and recursion

Q3.(e) Two Critical Regions - code that accesses a shared resource that must not be concurrently accessed, terminate in fixed time, synchronization mechanism is required.

Q3.(a) 3 pre-emptive scheduling policies

Shortest Remaining Time (SRT)

A pre-emptive scheduling version of the Shortest Job Next (SJN) algorithm. An algorithm that schedules processes based on the one which is nearest to completion. It can only be implemented on systems that are only Batch Processes, since you have to know the CPU time required to complete each job.

Round Robin

A pre-emptive scheduling algorithm that is used extensively in interactive systems. All active processes are given a pre-determined slice of time ("time quantum"). Choosing the time quantum is the key decision, for interactive systems the quantum must be small, whereas for batch systems it can be longer.

Multi-Level Queues

This isn't really a separate scheduling algorithm, it can be used with others. Jobs are grouped together based on common characteristics. For example, CPU-bound jobs based in one queue, and the I/O-bound jobs in another queue, and the process scheduler can select jobs from each queue based on balancing the load.

Q3.(b) Unix and NT access control for users

UNIX -- Access Control

UNIX uses access control lists. A user logs into UNIX and has a right to start processes that make requests. Each process has an identity(uid). This uid is obtained from the file that stores user passwords: /etc/passwd. Every process inherits its uid based on which user starts the process. Every process also has an effective uid, also a number, which may be different from the uid. Finally, each UNIX process is a member of some groups.

12 mode bits to encode protection privileges -- equivalent to encoding a set of access rights. Nine of the 12 mode bits are used to encode access rights. These access bits can be thought of as the protection matrix entry. They are divided into three groups of three:

- the first triplet (u) is for the user,
- the second (g) for the group and
- the third (o) for anyone else.

Each group can grant

- r: read,
- w: write, and
- x: execute.

Unix access control is very simple.

Windows NT -- Access Control

Windows NT supports multiple file systems, but the protection issues we will consider are only associated with one: NTFS. In NT there is the notion of an item, which can be a file or a directory. Each item has an owner, who can; change the access control list, allow other accounts to change the access control list and allow other accounts to become owner. NTFS is structured so that a file is a set of properties, the contents of the file being just one of those properties. An Access Control List (ACL) is a property of an item. The ACL itself is a list of entries: (user or group, permissions). The permission offer a rich set of possibilities:

• R - read • W - write • X - execute	• D - delete • P - modify the ACL • O - make current account the new owner

The owner is allowed to change the ACL. A user with permission P can also change the ACL. A user with permission O can take ownership. There is also a packaging of privileges known as permissions sets:

- no access
- read - RX
- change - RWXO
- full control - RWXDPO

NT access control is richer than UNIX, but not fundamentally different.

Q3.(c) Process Control Block

The elements of a Process Control Block are:

ProcessID

Each process is uniquely identified by both the user's identification, and a pointer connecting it to its descriptor.

Process Status

The current status of the process:

- Ready
- Running
- Waiting
- Hold
- Finsihed

Process State

- PC: Record the current value of the program counter
- Reg: Record the values of the data registers
- Main Mem: Record all important information from memory, including most importantly the process location
- Resources: Record the resources that have been allocated to this job, including files, disk drives, and printers.
- Priority: The process is assigned a priority, and if the operating system using priorities to schedule processes.

Accounting:

This section records some of the performance statistics associated with this process, including:

- CPU time used so far
- Time process has been in the system
- Time taken in memory (Main and secondary)
- Space taken in memory (Main and secondary)
- System programs used, compliers, editors, etc.
- Number and type of I/O operations
- Time spent waiting for I/O operations completion
- Number of Input records read and Output records written

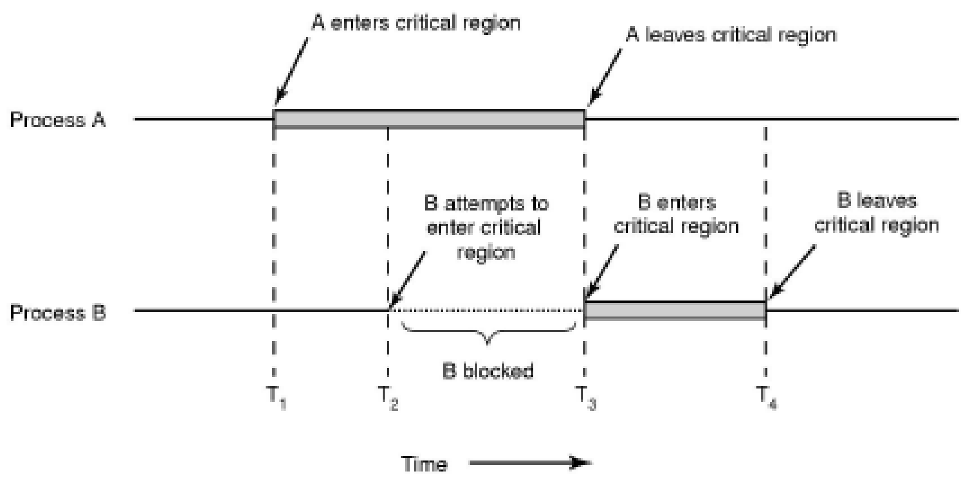
Q3.(d) Principle of Locality

The principle of locality (also known as locality of reference) is a phenomenon describing the same value, or related storage locations, being frequently accessed. There are two basic types of reference locality – temporal and spatial locality. Temporal locality refers to the reuse of specific data, and/or resources, within a relatively small time duration. Spatial locality refers to the use of data elements within relatively close storage locations. Sequential locality, a special case of spatial locality, occurs when data elements are arranged and accessed linearly, such as, traversing the elements in a one-dimensional array.

A loop is a good example, where the same commands are executed repeatedly. Recursion is also an example where the same commands are executed repeatedly.

Q3.(e) Two Critical Regions

A critical region is a piece of code that accesses a shared resource (data structure or device) that must not be concurrently accessed by more than one thread of execution. A critical region will usually terminate in fixed time, and a thread, task, or process will have to wait for a fixed time to enter it (aka bounded waiting). Some synchronization mechanism is required at the entry and exit of the critical region to ensure exclusive use, for example a semaphore. By carefully controlling which variables are modified inside and outside the critical region, concurrent access to that state is prevented.



SOME SUGGESTIONS:

Rough Work

- Always do rough work, I have to mark it
- Get as much as you can get down as quickly as you can, in as few words as possible

For Description Questions

- Try to include a diagram or an example (or both), it gives the answer a bit of life for the reader
- Assume about one sentence per mark, so for 10 marks write 10 sentences
- Underline phrases that are key
- Put headings on the start of paragraphs to tell me what they are about

For Problem-Solving Questions

- Use PseudoCode to design the answer if it helps
- Use Flowcharts as well to clarify in your mind
- Write out a simpler example, or one we did in class, that’s like the question being asked if that helps