Operating Systems Advanced Exercise

1. Briefly explain what an operating system is.

An operating system manages the whole computer system. It organises running programs in the RAM, it keeps track of where each file is stored in secondary storage, it offers a user-interface, coordinates I/O transfers and is also responsible for protection and security.

2. A computer can be looked at as having four levels: User (at the upper level), Applications, Operating System and Hardware (at the lowest level). Explain.

Starting from the upper level, the user communicates with the application program. Any request by the user (e.g. printing) will be passed to the operating system which will direct the hardware how to carry out the request.

3. Name and briefly describe three user-interfaces.

Three of the following:

- Command driven interface (CLI): In a CLI the user types text to communicate with the operating system or application. This is a primitive user interface.
- Menu driven interface: This interface presents a menu of options and the user can choose an option by using the arrow keys.
- Graphical user interface (GUI): A GUI presents icons and a mouse can be dragged on the icon and clicked (or double clicked) for the user to choose an option. Drop-down and pop-up menus are also used in a GUI.
- Form-based interface: This interface has the form of a form to be filled. It can contain text-boxes, menus and check boxes.
- Natural language interface: This interface can understand the spoken word.
- Touch-screen interface: Commands are entered by touching the screen with a stylus or with a finger.
- Hardware interface: examples are knobs, buttons and switches.
- Gesture interface: this interface can interpret movements by hand or whole body that represent particular commands.

4. Name four types of operating systems and describe them.

Four of the following:

- Batch operating system: The programs are executed in sequence and thus it cannot run interactive programs. JCL (Job Control Language) is a command language used with the batch OS. It specifies, for example, priority, program size as well as the files and databases used.
- Multiprogramming Operating System: This is an improvement on the batch OS. In the processor's wait states another process is executed.
- Time-sharing operating system: It enables many users to use the same computer at the same time. The processing of the programs seems

concurrent however the CPU only processes each program individually for a quantum of time before passing one by one to the other programs and then repeats the process forever. For some it is identical to multitasking, for others it is synonymous to multiuser.

- Multitasking: Multitasking is like time-sharing but it acts only on one processor. Multitasking can be pre-emptive or cooperative.
- Distributed operating system: It makes use of resources found on a number of linked computers like processors and RAM. Processors in a distributed system may vary in size and function so the OS dispatches jobs to the most appropriate processor.
- Network operating system: it manages applications on the server, data and users e.g. checks the users logging on. It manages security.
- Real Time operating system (RTOS): the operating system is bound with a time interval to give an output. There are two types of RTOSs: hard (critical) and soft (non-critical). In the hard RTOS response from the system cannot have any delays while soft systems are less restrictive.
- Single-user operating system: it can have only a single application running (e.g. on a mobile phone) or it can allow many applications to be open (e.g. a desktop with multitasking properties.
- Multi-user operating system: This is an operating system that supports two or more simultaneous users e.g. on mainframes or minicomputers. This is synonymous with timesharing.
- Multiprocessing operating system: This is time-sharing on multiple CPUs. Multiprocessing is divided into (1) Symmetric multiprocessing (SMP) (where the processors share memory and the I/O bus or data path and (2) Massively parallel processing (MPP) where each processor has its own operating system and memory.
- Multithreading operating system: It has the structure to execute threads in parallel taking care of the usual memory management, process management, security etc. Threads can belong to an application or to the operating system.
- Online operating system: This operating system downloads applications, checks who the connecting users are, on your computer is responsible for the installation and removal of applications. It enables you to use the programs by downloading the application modules into memory and by uploading requests and necessary all the necessary files to the sender.
- Embedded operating systems: These are usually used for hardware that have very little computing power, little RAM/ROM and a slow CPU, so they tend to be very specific in their applications and scope. Embedded OSs can be found in cars, large laser printers, some home appliances, and even military systems.

5. Name two different network architectures and give a brief description of them.

Two different architectures are:

- Client server: Each computer in the network is either a client (asks for a service e.g. to read a file) or a server (provides the service e.g. provides the file to be read) but not both.
- Peer-to-peer: All the computers within the network can be both clients and servers.

6. Explain Batch processing and Transaction processing.

Batch processing involves executing a sequence of non-interactive jobs all at one time. Batch jobs can be stored up during working hours and then executed during the evening or whenever the computer is idle e.g. updating a master file or working the salary of each employee. Once a batch job begins, it continues until it is done or until an error occurs. There is no interaction with the user.

The opposite of batch processing is transaction processing or interactive processing. In interactive processing, the application responds to commands as soon as you enter them.

7. Name the five main functions of an operating system.

- Process management
- Memory management
- Storage management
- Protection and security
- User interface

8. Explain the following terms: process, pre-emption, dispatch, dispatcher, dispatch latency and context switch.

Process: a program module during execution
Pre-emption: stopping a process from using the CPU so that it can be assigned to another process.
Dispatch: a dispatch occurs when the operating system assigns the CPU to a process.
Dispatcher: the program module of the operating system responsible for the dispatching of processes.
Dispatch latency: The time it takes for the dispatcher to stop one process and start another running.
Context switch: this refers the switching of the CPU from one process or thread to another.

9. State and give a brief description of the three states of a process.

- Executing: when the process is being executed by the CPU.
- Ready (waiting): when the process is waiting to be dispatched to the CPU to be executed.

- Blocked (suspended): when the process cannot proceed because it is waiting for an event to happen e.g. an input.

10. Write and explain three objectives of a process scheduler.

(i) Throughput: to execute the most number of programs possible in the least time.
(ii) CPU utilization: if possible do not let the CPU idle
(iii) Fairness: no process should be left to starve waiting for CPU time; CPU time should be shared fairly.
(iv) Smallest possible turnaround time: the turnaround time is the interval between the submission of a process and the completion of its execution, including the actual running time, plus the time spent sleeping before being dispatched or while waiting to access various resources.
(v) Policy Enforcement: The scheduler has to make sure that system's policy is enforced. For example, if the local policy is safety then the safety control processes must be able to run whenever they want to, even if it means delay in payroll processes.

11.

a. What do we mean by process scheduling?
b. Name five scheduling algorithms and describe two of them.

a. Process scheduling is the policy which is used to decide when a process should be pre-empted and which process should be dispatched.
b. Five scheduling algorithms are the following:
    i. First Come First Served (FCFS): The jobs are run in the sequence they arrive i.e. following the FIFO policy. Since the procedure of doing things is non-pre-emptive it is used only for batch processes.
    ii. Shortest Job First (SJF): The jobs that take less time are executed first. This procedure requires that the process' time be known beforehand. There are two versions (i) the pre-emptive and (ii) the non-pre-emptive.
    iii. Priority Scheduling: In priority scheduling each process is assigned a priority and the process with the highest priority is dispatched first. Processes with the same priority are dispatched on a FCFS bases. One problem in this type of scheduling is starvation which can be overcome by means of aging. There are two versions: the pre-emptive and the non-pre-emptive.
    iv. Round Robin (RR): The Round Robin policy is to execute each process in sequence for a time-slice each and when the last process is given its time-slice the procedure starts all over again.

v. Multilevel Scheduling: This scheduling algorithm incorporates a multi-level queue with different scheduling algorithms within each queue, each suitable for the type of process queued.

12.

a. Explain the following terms: (i) deadlock, (ii) critical region and (iii) circular wait.
b. What solution is used to avoid concurrent execution by critical regions?
c. If a process is in the middle of a critical region can it be pre-empted?

a. Deadlock: this is a situation where two or more processes mutually hold each other from proceeding because each one of them is requesting a resource that is held by some other process which itself is waiting for a resource.
Critical region: This is a section in a program that accesses a resource that cannot be shared. So the region cannot run concurrently with another critical region in another program that accesses the same resource.
Circular wait occurs when a closed chain of 'blocked' processes each hold a resource that is required by another process in the chain in such a way that if one of the processes does not release the resource the processes will keep on waiting forever.
b. Semaphores. A semaphore is a hardware or software flag that indicates whether a desired resource is in use or not. If it is not in use a process can lock it before entering the critical region. When it is ready it unlocks it so that other processes can use it. The semaphore is usually called mutex and to lock the resource one would write "wait (mutex)" and to release it one would write signal(mutex).
c. Yes it can but while the process is in the 'ready' state no other process can access the resource.

13. Name and explain the conditions that should hold to create a deadlock.

1. *Mutual Exclusion:* At least one resource must be nonshareable. Only one process can use the resource at any given instant of time.
2. *Hold and Wait* (*Resource Holding):* A process is currently holding at least one resource and requesting additional resources which are being held by other processes.
3. *No Deallocation*: The operating system does not deallocate resources once they have been allocated; they must be released by the holding process voluntarily.

However, 1, 2 and 3 may be present (momentarily) without finally causing a deadlock. For a deadlock the following fourth condition must be present (with the other three).

4. *Circular Wait:* A process must be waiting for a resource which is being held by another process, which in turn is

waiting for the first process to release the resource. In general, there is a set of waiting processes, P = {P_1, P_2, …, P_N}, such that P_1 is waiting for a resource held by P_2, P_2 is waiting for a resource held by P_3 and so on till P_N is waiting for a resource held by P_1.

14. What is a resource allocation graph?

It is a diagram that represents the resources processes are holding or requesting. A closed circuit in it will indicate a deadlock.

15. There are four methods of handling deadlocks. Name and explain them briefly.

Deadlock Prevention: Disallow one of the four necessary conditions for deadlock i.e. disallow one of the following (i) mutual exclusion, (ii) hold and wait, (iii) no deallocation and (iv) circular wait. Disallowing mutual exclusion is practically impossible (a printer can be used only with mutual exclusion). One can disallow 'hold and wait' by making processes request all resources before beginning execution. This will lead to low resource utilization and may also lead to starvation. Imposing deallocation may be possible in certain cases e.g. if a process is holding a table from a database this table can be released from the process however before doing so the operations performed by the process on the table have to be rolled back. Circular wait can be prevented if the resources are ordered and each process has to request resources only in a certain order. This would impose restrictions on programmers.

Deadlock Avoidance: Do not grant a resource request if the allocation has the potential to lead to a deadlock. This would require that the maximum requirements of each resource be stated in advance. It would also add overhead on the operating system as checks would have to be made each time a resource is required.

Deadlock Detection (and Recovery): In this method the operating system will always grant a resource when this is requested and it is free. Periodically it checks for deadlocks. If a deadlock is discovered the operating system will recover from it by aborting a process (thus releasing all its resources it held) and rolling back to undo all operations done by the process. The process will be executed some time later.

Ignore the problem altogether: If deadlocks only occur once a year or so, it may be better to simply let them happen and reboot as necessary than to incur the constant overhead and system performance penalties associated with deadlock prevention or detection. This is the approach that both Windows and UNIX take.

16. Describe the following terms: (i) uniprogramming, (ii) kernel, (iii) logical and physical addresses, (iv) relocation, (v) process control block

(i) In uniprogramming there is only one program (besides the OS kernel) present in RAM.
(ii) A kernel is the core component of an operating system. It acts as a bridge between applications and the data processing performed at the hardware level. When an operating system is loaded into memory, the kernel loads first and remains in memory until the operating system is shut down again. The kernel is responsible for low-level tasks such as disk management, task management and memory management.
(iii) A logical address is an address reference in a program that has not yet been loaded in memory. The physical address is the address that the 'reference' will hold in memory after loading.
(iv) Relocation is the process of converting a logical address to a physical address. Its purpose is to let a program be executed from different areas of main memory at different times.
(v) A PCB contains information needed to manage processes e.g. identifier for the process, user identifier, the content of the CPU general-purpose registers, priority value etc.

17.
a. "Performance is poor if the OS has to load a process each time it is scheduled." Explain.
b. Therefore, what can be done to improve the performance of a computer?

a. Loading takes a lot of time (in computer terms) because it involves secondary storage which is much slower than main memory. In this case the CPU is idle while waiting for the process to load.
b. Have as large a RAM as possible and make it hold as many programs as possible.

18. "The OS sees that processes do not reference memory locations belonging to other processes." Explain.

Each address reference is checked (by hardware) during execution to see that it is between the ranges of addresses corresponding to the same process.

19. Explain what is meant by (i) shared memory, (ii) logical and physical organisation

(i) It means that different processes are using a common part in RAM e.g. if three users are accessing the same database then this database is shared between them, or if two users are using the same thesaurus on a network there may be only one thesaurus in memory shared by their two programs.

(ii) The physical organisation of memory is linear i.e. it consists of a sequence of memory cells. The logical organisation of a data structure is not necessarily linear e.g. a tree. The logical organisation will always occupy a linear physical organisation.

20.

a. What is memory overlaying?
b. Name one disadvantage of memory overlaying.

a. Memory overlaying occurs when a program replaces one of its modules in memory with another of its own modules at the same physical location. Each transfer is specifically programmed. Overlaying involves dividing a large program into a main file and one or more overlay files.
b. This solution places a burden on the programmer who has to divide the program into units each of which cannot be larger than the prescribed overlay. This can cause 'disk trashing' (excessive disk transfers).

21.

a. What is memory partitioning?
b. Issues with memory partitioning are (i) fragmentation (internal and external), (ii) placement, (iii) replacement and (iv) free space management. Explain them briefly.

a. Memory partitioning is the division of memory in spaces to hold processes in them. Each partition will hold a single process. Partitions may be fixed (with equal partitions or unequal partitions) or dynamic (allocated immediately before loading).

b.

i. This is a condition in which small blocks of unusable memory are created, either within a partition (internal) or between partitions (external).
ii. This concerns the place where a process should be placed?
iii. If all processes are blocked, it may be desirable to swap out one or more to make room for a new process or one in the 'ready' state. How do you decide which one(s) to swap?
iv. How does the system know which blocks of memory are free (not assigned to a process)?

22. Describe two problems associated with fixed partitioning.

i. If a program is larger than any partition, it must be run using user-managed overlays.
ii. The maximum number of processes in memory is limited to the number of partitions.
iii. It causes internal fragmentation.

23.

a. What is Segmentation?
b. Explain the terms (i) hardware exception, (ii) segmentation fault.

a. Memory segmentation is the division of a computer's primary memory into segments (sections). Using segmentation, a reference to a memory location includes a value that identifies a segment and an offset within that segment. Segments or sections are also used in programs. Segments usually correspond to natural divisions of a program such as individual routines or data tables. Certain segments may be shared between programs.
b. (i) A hardware exception is generated whenever some unexpected critical event occurs e.g. reading or writing to an inaccessible memory location or trying to divide by 0. (ii) A segmentation fault is a kind of hardware exception that is generated when a process makes a reference to an address residing outside the range of the segment. It is also generated if a reference points towards an area of which the user has no permission to access.

24. In segmentation a logical address is expressed as (s, d). Explain.

In the address (s, d) s is the segment number and d is the offset into the segment.

25. Four placement algorithms used with segmentation are the following: best fit, worst fit, first fit and next fit. Describe them briefly and list them in order of efficiency.

Best fit: choose smallest hole
Worst fit: choose the largest hole
First fit: choose first hole (that fits) from beginning
Next-fit: choose first hole from last choice

(1) (fastest) first-fit, (2) next-fit, (3) worst-fit, (4) best-fit

26.
a. Why is compaction used?
b. What is a relative address?
c. What is address translation (address mapping)?

a. It is used to eliminate external fragmentation.
b. A memory address that represents some distance from a starting point (base address), such as the first byte of a program or table. The absolute address is derived by adding it to the base address.
c. The procedure of deriving the physical address from the logical address.

27.
a. How does the paging system work (virtual memory)?
b. What is a virtual address?
c. What is a page fault?

d. Name and explain four replacement algorithms used to choose the page to be swapped.

a. The paging system works by dividing the memory in frames (equal size). The process space is divided into equal pages. The size of the frames is equal to the size of the pages. During execution a program will not be wholly present in memory. Only the necessary parts will. If a program requires a page that is not present in memory this is copied from storage to memory. If there is no empty frame in memory a page is downloaded temporarily to secondary storage to make space for the necessary page. This is called swapping.
b. A virtual address is a logical address.
c. A page fault is an interrupt that occurs when a program requests data that is not currently in real memory. The interrupt triggers the operating system to fetch the data from a virtual memory and load it into RAM.
d.
   i. Random Replacement algorithm: Page chosen in random order. This algorithm would not be really suitable for real programs which normally access locally, i.e. in sequential order.
   ii. First-In-First-Out: The page existing in the main memory for the longest time is transferred out.
   iii. First-in-not-used-first-out: Replaces the oldest page which has not been used recently.
   iv. Least Recently Used: The page which has not been used for the longest time is transferred out.

28. Give a brief description of the following basic file organisations: (i) serial file, (ii) sequential file, (iii) indexed file, (iv) indexed sequential file, (v) direct file, (vi) byte-stream files.

   i. In this kind of file there is no uniform structure to the data. The records are not sorted. Record access is done by exhaustive search.
   ii. In the sequential file a key field uniquely identifies each record. The file is sorted on a given key.
   iii. This file system builds an index to help locate specific records within a file. Index entries consist of key value and a pointer into the file. The index may be a complex tree structure, or a simple list.
   iv. This file is used sequentially in normal cases but occasionally (e.g. to change the address of a member) they are used randomly.
   v. Direct files can be Indexed or Hashed. In an indexed file the address of the required record is searched in the index. The hash file will require to work a mathematical expression to get the desired address.
   vi. Some file systems treat files as a sequential stream of bytes. Each file has a file pointer holds the index of the next byte to be read.

29. What is the difference between the following file directory structures: (i) single-level, two-level and tree-structured.

(i) A Single-Level is formed by a list of all files. Search in such a directory is slow.
(ii) In a two-level directory, on level 1 we find the master directory for the entire system while on level 2 we find the personal directory for each user.
(iii) Tree structured directories allow multiple hierarchical subdirectories in each user directory. Pathnames are used to locate files (as in any tree structure) and search is fairly efficient.

30. Name three access rights on files that may be given to users.

- read: user can read and use the file, but can't change it
- write: user can modify, delete or add to the file
- execute: user can load and run the program, cannot change or copy it
- append: add new records in the file
- delete: delete a file

31. What does a file manipulation system do?

It performs 'file open operation' tasks before an application program can read or write a file's contents e.g. searching the file in an index to find out where it is, see if it is already open, see if the user has the right to open that file etc.

32.
a. What do we mean by (i) a block (physical record) and (ii) a logical record?
b. One meaning of 'blocking' refers to the process of packing logical records into physical blocks on secondary storage. What is the meaning of a blocking factor of 3?
c. What do we mean by the terms (i) deblocking, (ii) portion?

a. (i) A block (physical record) is the unit of data for transfer to and from an I/O medium for example 512 bytes. (ii) A logical record is the unit of data processed by a process e.g. a record.
b. It means that a physical record contains 3 logical records.
c. (i) The process of extracting the required record from within the physical record (which contains more than one logical record). (ii) A portion is a contiguous set of allocated blocks. Portions can be fixed size and variable size.

33. Name three blocking methods.

- Fixed blocking: all records have the same size; each block contains an integral number of records. It may have waste space at the end of the block.
- Variable length spanned blocking: records are of different sizes; they are stored in blocks with no waste space. Some records span blocks.

- Variable length un-spanned blocking: variable-length records with no records divided across blocks. Space may be wasted.

34. When managing the secondary storage the operating system designer has two management issues: (i) file allocation and (ii) free space management. Explain.

The system's analyst has to decide about (i) file allocation i.e. how to allocate secondary storage blocks to files (pre-allocation or dynamic allocation; pre-allocation requires that the eventual size of the file be known (or estimated) in advance so all space can be allocated at once; over-estimation wastes disk space; dynamic allocation allocates storage in portions as needed); how large should a portion be (should it be variable or fixed?); what data structure will be used as an index to keep track where files are. (ii) free space management requires itself a data structure to keep track of the free blocks of storage.

35. Describe three file-allocation methods.

Three allocation methods are the following:

a. Contiguous: a contiguous set of blocks is allocated to a file at the time of creation. The FAT (file allocation table) contains one entry for each file. This consists in the starting block and length of the file.
b. Chained (linked): in this organisation blocks are allocated one at a time (dynamic allocation). Each block has a pointer to the next one in the chain. There is one entry in the file allocation table which consists of the starting block and length of file.
c. Indexed: in this method for each file the FAT records an index of a portion on disk. This portion itself holds the addresses of the portions that form the file.

36.

a. What does the Disk Allocation Table contain?
b. Name and describe three ways in which this table can contain its information.

a. The Disk Allocation Table shows where the unallocated blocks of memory are located.
b. This table can have one of the following organisations:
   i. Bit vectors/tables: contain one bit for each disk block. 0/1 indicates free/used.
   ii. Chained free portions: keep a pointer to a free portion on disk. In that portion, store a pointer to the next free portion etc.
   iii. Indexing: treats free space as a file, and has an index with one entry for each free portion on disk.

37.

a. Define (i) efficiency, (ii) reliability, (iii) security and (iv) fault tolerance.

b. Mention one technique that enhances efficiency when reading or writing to secondary storage.
c. Mention and explain three techniques that raise the level of fault tolerance.

a. (i) Efficiency refers to quickness in doing a job, (ii) reliability refers to assurance that the job or the results derived from it are correct, (iii) security is the work done in guaranteeing that the data and programs are not changed, deleted or accessed without permission, (iv) fault tolerance is the degree by which a system can suffer a fault but because of support mechanisms the fault will not stop the system from working and it will not cause information to be lost or suffer corruption.
b. Efficiency will increase if the indices are kept in memory instead of secondary storage.
c.
    i. File backup: this can be full backup (the FMS copies all files and directories for an entire storage volume); incremental backup (only files that have been modified are archived); differential backup (only the changed portions of the files are archived).
    ii. Transaction logging and (eventual) file recovery: this means that every update is recorded in detail together with information such as what it did, the time it did it and by whom.
    iii. Mirroring: here identical secondary storage devices are updated concurrently such that if one stops functioning the other will have all the information intact. It is very improbable that both secondary devices will fail at the same time.

38. Describe the two I/O addressing modes.

These are (i) isolated/separated I/O where the I/O address space is distinct from physical memory, and (ii) memory-mapped I/O: where there is one address space which includes both memory and the I/O addresses. The second scheme is more flexible.

39. What do we mean by handshaking in digital communications?

Handshaking is the process by which two devices initiate communications. Handshaking begins when one device sends a message to another device indicating that it wants to establish a communications channel. The two devices then send several messages back and forth that enable them to agree on a communications protocol.

40.
a. What do we mean by an interrupt?
b. What is the name of the program that is executed when an interrupt is signalled to the processor?
c. What is the difference between a hardware and a software interrupt?

d. What is an IRQ?

a. An interrupt is a signal to the processor indicating that an event needs immediate attention.
b. Interrupt handler or interrupt service routine (ISR).
c. A hardware interrupt is a signal coming from a device while a software interrupt is triggered from a program. An example of a hardware interrupt is a signal to the CPU indicating that a key on the keyboard has been pressed. An example of a software interrupt is when a program is asked to divide a number by zero.
d. The act of initiating a hardware interrupt is referred to as an interrupt request (IRQ).

41. What are the steps that the processor performs when it receives an interrupt?

The processor:

(i) Suspends its current activities and saves the state of the current program.
(ii) It executes the interrupt handler associated with the event (this interruption is temporary, and after the interrupt handler finishes, the processor resumes execution of the previous thread).

42. What is an interrupt-driven system?

It is a system like multitasking and real-time systems where interrupts play an important role in the functioning of the whole system.

43. What is the difference between polled and vectored interrupts?

A polled interrupt is a specific type of I/O interrupt that notifies the part of the computer containing the I/O interface that a device sent an interrupt but does not indicate which device. The interrupt controller must poll (send a signal out to) each device to determine which one made the request.

The alternative to a polled interrupt is a vectored interrupt, an interrupt signal that includes the identity of the device sending the interrupt signal.

44. What is an interrupt vector table?

It is a generic term for a table that associates a list of interrupt handlers with a list of interrupt requests.

45. What is a system stack?

The system stack is used to store information about subroutine calls. When a subroutine is temporarily halted (because another subroutine has to be executed)

all its information is stored in the stack so that it can be called back when the previously-halted subroutine proceeds in its execution.

46.
   a. What two techniques can be applied with multiple interrupts?
   b. What is interrupt latency?

a. (i) Disable interrupts while an interrupt is being processed, (ii) define priorities for interrupts and allow an interrupt of higher priority to interrupt an interrupt-handler of lower priority.
b. It is the time that elapses from when an interrupt is generated to when the source of the interrupt is serviced.

47.
   a. What is the purpose of the interrupt mask register?
   b. How does it work?

a. Its purpose is to disable temporarily part or all of the interrupts so that the concurrent process would not be interrupted.
b. It works in such a way that the CPU does not read the interrupt vector. Instead it reads another vector formed by performing the AND operation of bits from the interrupt vector with bits from the mask vector. At the end of the process the interrupts will be enabled again.

48.
   a. What is the purpose of the DMA?
   b. Describe an alternative for the DMA.

a. Direct Memory Access (DMA) is a capability provided by some computer bus architectures that allows data to be sent directly from an attached device (such as a disk drive) to the memory on the computer's motherboard. The microprocessor is freed from involvement with the data transfer, thus speeding up overall computer operation. Usually a specified portion of memory is designated as an area to be used for direct memory access.
b. An alternative to DMA is the Programmed Input/Output (PIO) interface in which all data transmitted between devices goes through the processor.