1. Explain about segmentation and paging

A. Segmentation:

It is a memory management scheme that supports the user view of memory. A logical address space is a collection of segments each segment has a name and a length.

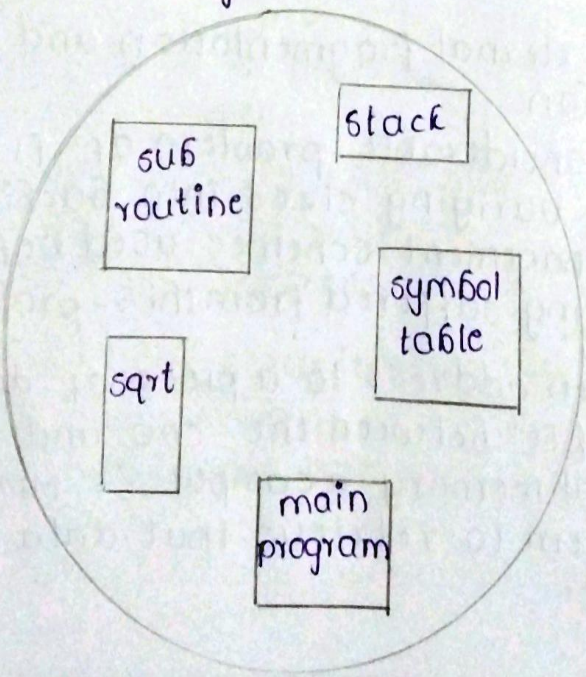
1. As we have already seen, the user's view of memory is not the same as the actual physical memory.
2. The user's view is mapped onto physical memory
3. This mapping allows differentiation between logical memory and physical memory.

→ how you think of a program when you are writing it. You think of it as a main program with a set of methods, procedures or functions.

→ It may also include various data structures, objects, arrays, stacks and variables so on.

→ Each of these modules or data elements is reffered to by name. You talk about the stack "the math library" without caring what addresses in memory these elements occupy.

Logical address



A C-compiler might create separate segments for the following

1. The code
2. Global variables
3. The heap, from which memory is allocated
4. The stack used by each thread
5. The standart C library

Each segmentation is a different virtual address space that directly corresponds to process objects.
When a process executes, segmentation assigns related data into segments for faster processing. The segmentation function maintains a segment table that includes physical address of the segment, size and other data.

→ Segmentation speeds up a computers information retrieval by assigning related data into a "segment table" between the CPU and the physical memory.

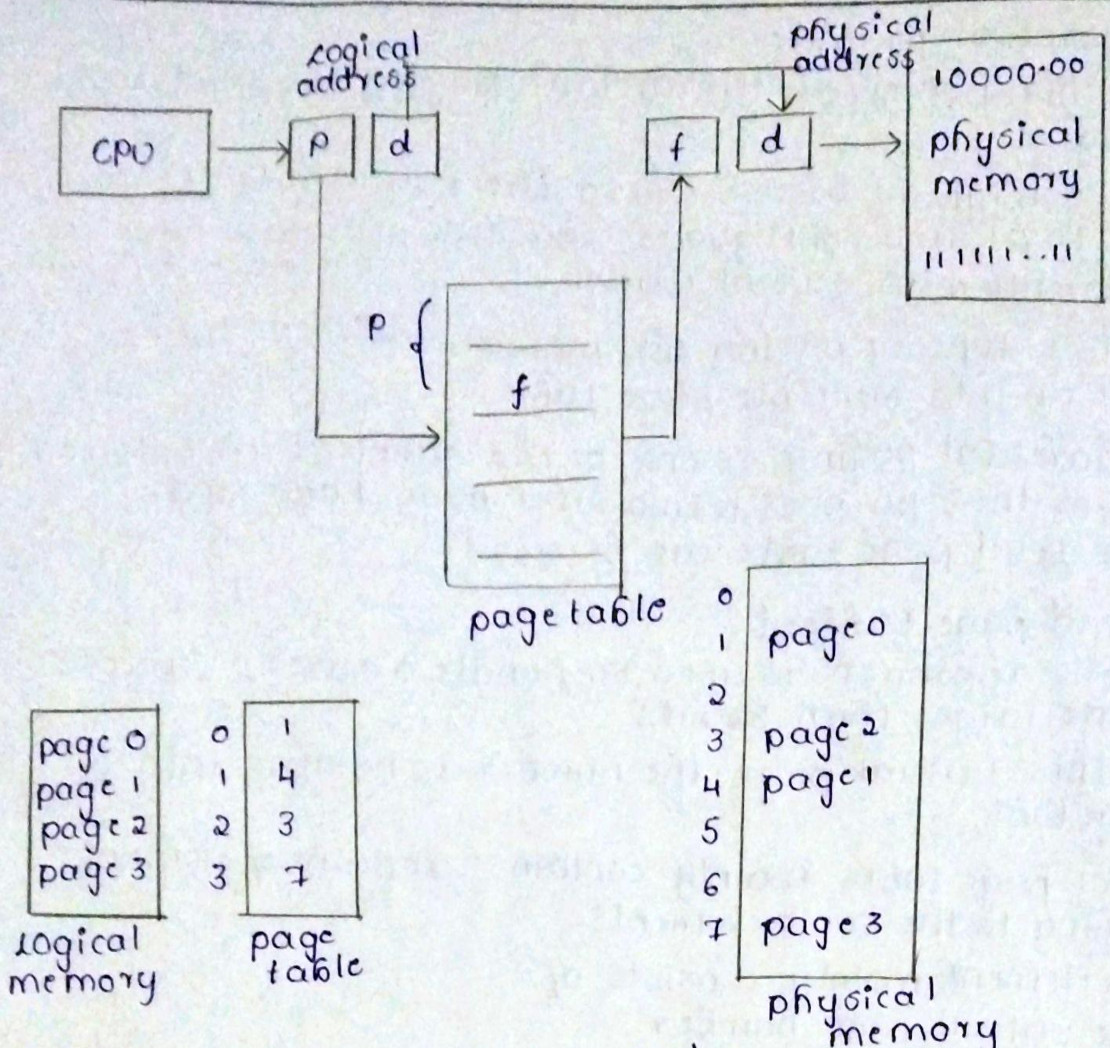
Paging:

Paging is a computer memory management function that presents storage locations to the computer's CPU as additional memory, called virtual memory. Each piece of data needs a storage address.
As mentioned above, the memory management function called paging specifies storage locations to the CPU as additional memory, called virtual memory. The CPU cannot directly access storage disk, so the MMU emulates memory by mapping pages to frames that are in RAM.

→ Paging avoids external fragmentation and the need for compaction

→ It also solves considerable problem of fitting memory chunks of variying sizes into backing store, most memory management schemes used before the introduction of paging suffered from this problem

→ By assigning an address to a piece of data using a "page table" between the CPU and the computer's physical memory, a computer's MMU enables the system to retrieve that data whenever needed.




paging model of logical and physical memory

Structure of page table:

1. Simply defines in how many ways page table can be structured. well, the paging is memory management technique where a large process is divided into pages and is placed in physical memory which is also divided into frames.

2. Frame and page size is equivalent. The OS uses page table to map logical address of page generated by CPU to its physical address in the main memory.

Techiques used for structuring the page table

Some of the common techniques that are used for structuring the page table are as follows

1. Hierarchial paging
2. Hashed page tables
3. Inverted page tables

Scanned by Scanner Go

## Hierarchial paging:

Another name for Hierarchial paging is multilevel paging

→ There might be a case where the page table is too big to fit in a contiguous space, so we may have a hierarchy with several levels.

→ In this type of paging the logical address space is broke up into Multiple page tables.

→ Hierarchial paging is one of the simplest techniques and for this purpose, a two-level page table and three-level page table can be used.

## Hashed page tables:

This approach is used to handle address spaces that are larger than 32 bits

→ In this virtual page, the number is hashed into a page tables

→ This page table mainly contains a chain of elements hashing to the same elements.

Each element mainly consists of

1. The virtual page number
2. The value of the mapped page frame
3. A pointer to the next element in the linked list.

## Inverted page tables:

Inverted page table is a combination of a page table and a frame table. There is one entry for virtual page number and real page of memory.

This method decreases the memory needed to store each page table.

But, it may also increase the amount of time needed to search the table for a page reference.

2. Explain briefly about deadlock prevention, detection avoidance and recovery.

A. Deadlock characteristics:

Deadlock has following characteristics

1. Mutual exclusion
2. Hold and wait
3. No preemption
4. circular wait

Deadlock prevention:

We can prevent deadlock by eliminating any of the above four conditions

Eliminate Mutual Exclusion:

It is not possible to dis-satisfy the mutual exclusion because some resources, such as the tape drive and printer, are inherently non-shareable

Eliminate hold and wait:

Allocate all required resources to the process before the start of its execution, this way hold and wait condition is eliminated but it will lead to low device utilization.

Eliminate No preemption:

preempt resources from the process when resources required by other high priority processes

Eliminate circular wait:

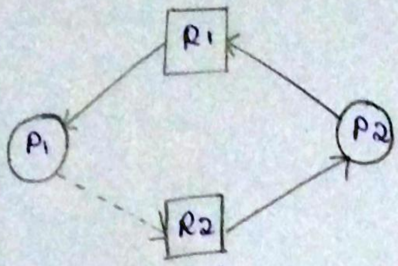
Each resource will be assigned with a numerical number. A process can request the resources increasing or decreasing order of numbering.

Deadlock detection:

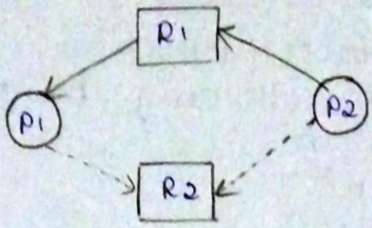
1. Resources contains type with single instances have two graphs Resource allocation graph and wait for graph.
2. Multiple instances - banker's algorithm

If resources have a single instance -

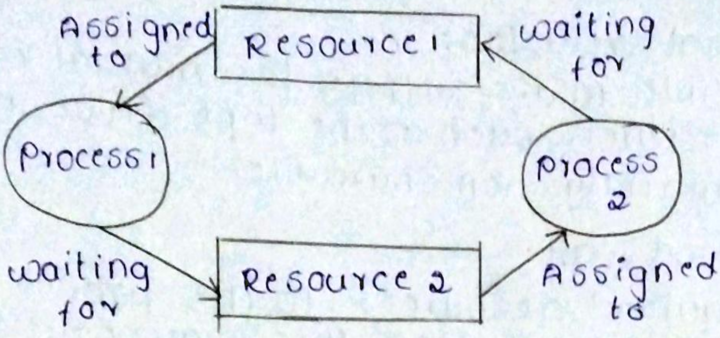
In this case for deadlock detection, we can run an algorithm to check for the cycle in the Resource allocation graph. The presence of a cycle in the graph is a sufficient condition for deadlock.




unsafe stage in allocation resource graph




Resource allocation graph for deadlock avoidance




In the above diagram, resource 1 and resource 2 have single instances. There is a cycle $R1 \rightarrow P1 \rightarrow R2 \rightarrow P2$. so, deadlock is confirmed.

If there are multiple instances of resources -
Detection of the cycle is necessary but not sufficient condition for dead lock, in this case, the system may or may not be in deadlock varies according to different situations.

Deadlock avoidance:
Deadlock avoidance can be done with Banker's algorithm

Banker's Algorithm:
Banker's algorithm is resource allocation and deadlock avoidance algorithm which test all the request made by processes for resources, it checks for the safe state, if after granting request system remains in the safe state it allows the request and if there is no safe state it doesn't allow the request made by the process.

Inputs to Banker's algorithm:

1. Max need of resources by each process.
2. currently, allocated resources by each process.
3. Max free available resources in the system.

The request will only be granted under the below condition:

1. If the request made by the process is less than equal to max need to that process.
2. If the request made by the process is less than equal to the freely available resource in the system.

Deadlock Recovery:

A traditional operating system such as windows doesn't deal with deadlock recovery as it is a time and space-consuming process. Real-time operating systems use Deadlock recovery.

1. Killing the process -

Killing all the processes involved in the deadlock. Killing process one by one. After killing each process check for deadlock again keep repeating the process till the system recovers from deadlock. Killing all the processes one by one helps a system to break circular wait condition.

2. Resource preemption -

Resources are preempted from the processes involved in the deadlock, preempted resources are allocated to other processes so that there is a possibility of recovering the system from deadlock. In this case, the system goes into starvation.

---

Discuss briefly about Disk structure, attachment, scheduling.

Disk structure:

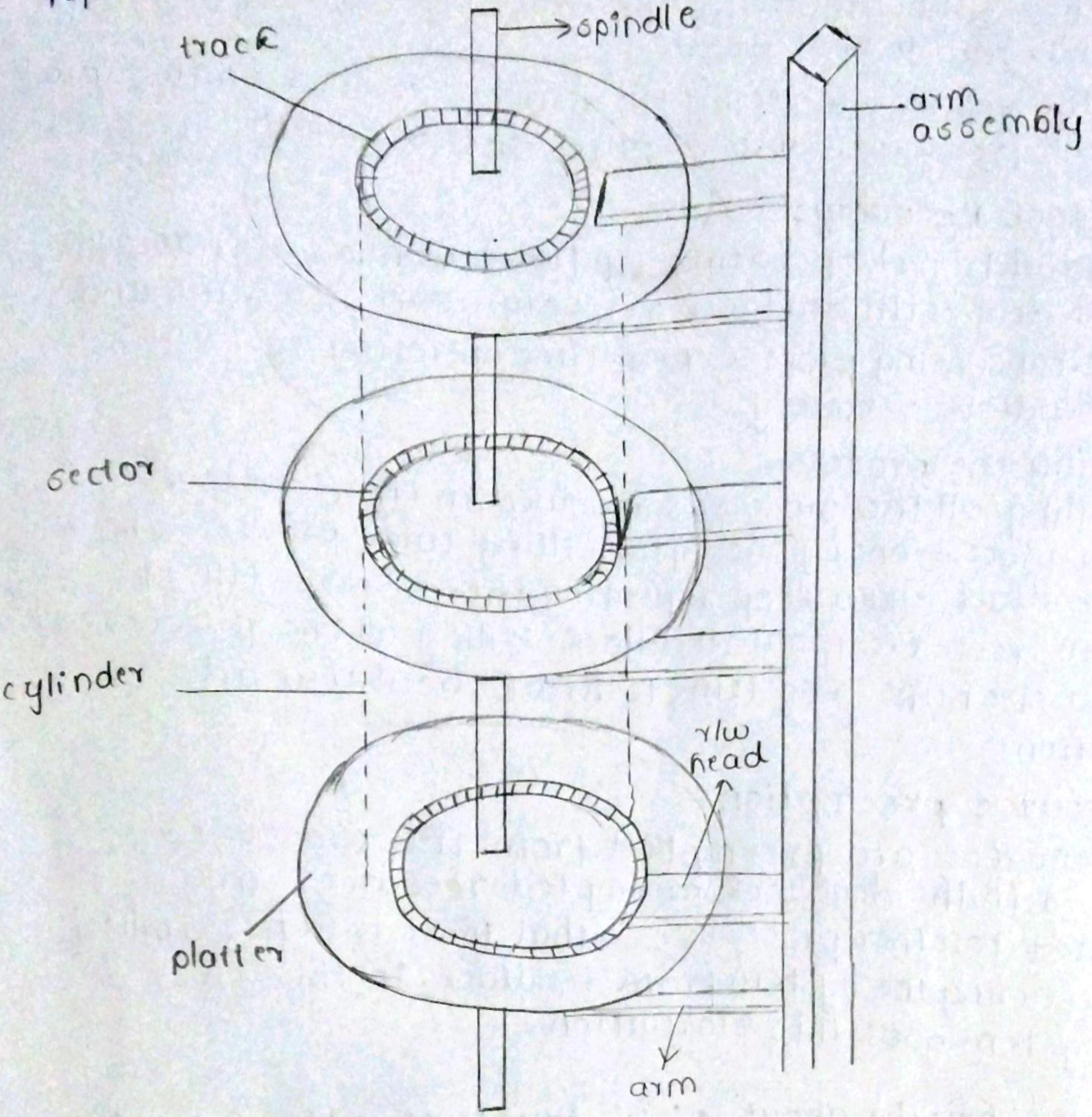
Here the length of the tracks near the center is less than the length of the tracks farther from the centre.

→ each track is further divided into sectors

→ spindle revolves the platters and is controlled by r/w unit of OS. Some advanced spindles have capability to only revolve a particular disk and keep others intact.

→ Arm assembly is there which keeps a pointy r/w head on each disk to read of write on a particular disk.

→ A world cylinder may also be used at times to refer disk stack.




→ Transfer rate:

This is rate at which the data moves from disk to the computer

→ Random access time:

It is the sum of the seek time and rotational latency

Using these two parts the speed of disk is measured

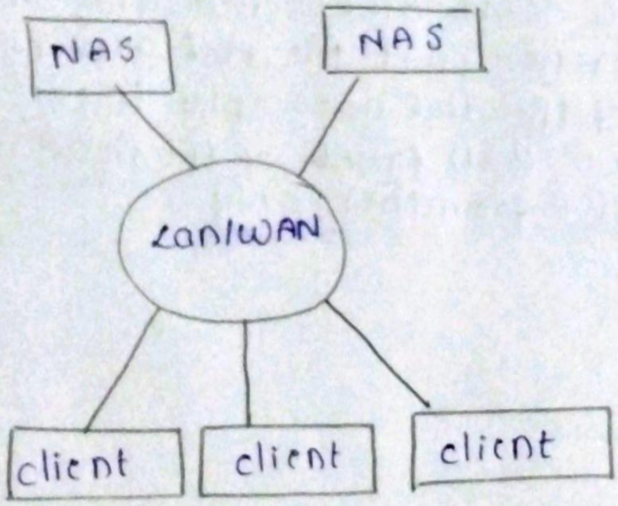
## Disk Attachment:

There are two ways for computers to access disk storage. One method is to use I/O ports also known as host-attached storage, on small systems. Another method is through a remote host in a distributed file system, which is known as network-attached storage.

→ Host-attached storage (HAS) is storage accessed via local I/O ports. These ports make use of various technologies

→ SCSI (Small computer system interface) is bus architecture. Its physical medium is typically a ribbon wire with many conductors.

→ A network-attached storage (NAS) device is a dedicated storage system that can be accessed via the data network.




## Disk scheduling:

Disk scheduling is done by operating systems to schedule I/O requests arriving for the disk. Disk scheduling is also known as I/O scheduling.

## Disk scheduling algorithms

### 1. FCFS:

FCFS is the simplest of all the disk scheduling algorithms. In FCFS the requests are addressed in the order they arrive in the disk queue.

2. SSTF (Shortest seek time first):

In SSTF requests having shortest seek time are executed first. So the seek time of every request is calculated in advance in the queue and then they are scheduled according to their calculated seek time.

3. SCAN:

In SCAN algorithm the disk arm moves into a particular direction and services the requests coming in its path and after reaching the end of disk, it reverses its direction and again services the request arriving in its path.

4. CSCAN:

In CSCAN instead of reversing its direction goes to the other end of the disk and starts servicing the requests from there

5. LOOK:

It is similar to SCAN disk scheduling algorithm except for the difference that the disk arm in spite of going to the end of the disk goes only to the last request to be serviced in front of the head and then reverses its direction from there only.