OSSP-S-2019

1. Answer all questions

a. **Define Process**: A process or a task is an instance of a program running in a computer. A process can initiate a sub process, which is a called a child process

b. **What is Semaphore**: Semaphore is simply a variable that is non-negative and shared between threads. A semaphore is a signaling mechanism, and a thread that is waiting on a semaphore can be signaled by another thread. It uses two atomic operations, 1)wait, and 2) signal for the process synchronization.

c. **Define Page**: A page, memory page, or virtual page is a fixed-length contiguous block of virtual memory, described by a single entry in the page table. It is the smallest unit of data for memory management in a virtual memory operating system.

d. **Write the function of interpreter**: An interpreter translates high-level instructions into an intermediate form, which it then executes. In contrast, a compiler translates high-level instructions directly into machine language. ... The interpreter, on the other hand, can immediately execute high-level programs.

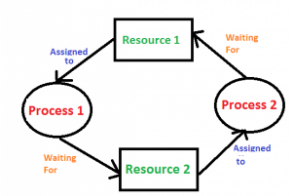
e. **What is context switching**: Context Switching involves storing the context or state of a process so that it can be reloaded when required and execution can be resumed from the same point as earlier

f. **Name two os**: windows, linux

g. **What is spooling**: Spooling is a process in which data is temporarily held to be used and executed by a device, program or the system. Data is sent to and stored in memory or other volatile storage until the program or computer requests it for execution. "Spool" is technically an acronym for simultaneous peripheral operations online.

h. **What is the difference between appl. Software & system s/w**: Application software (app for short) is a program or group of programs designed for end users. System software is a type of computer program that is designed to run a computer's hardware and application programs.

i. **Give an example of deadlock situation**: here the proces1is holding resource 1 and waiting for resource 2 while process2 is holding resource 2 and waiting for resource1.this is the situation of deadlock.



j. **Write two file access method**: Sequential-Access, Direct Access.

2. Answer any five

a. Identify the Function of os and explain each

Ans : Following are some of important functions of an operating System.

Memory Management

An Operating System does the following activities for memory management −

- Keeps tracks of primary memory, i.e., what part of it are in use by whom, what part are not in use.
- In multiprogramming, the OS decides which process will get memory when and how much.
- Allocates the memory when a process requests it to do so.
- De-allocates the memory when a process no longer needs it or has been terminated.

Processor Management

In multiprogramming environment, the OS decides which process gets the processor when and for how much time. This function is called process scheduling. An Operating System does the following activities for processor management −

- Keeps tracks of processor and status of process. The program responsible for this task is known as traffic controller.
- Allocates the processor (CPU) to a process.
- De-allocates processor when a process is no longer required.

Device Management

An Operating System manages device communication via their respective drivers. It does the following activities for device management −

- Keeps tracks of all devices. Program responsible for this task is known as the I/O controller.
- Decides which process gets the device when and for how much time.
- Allocates the device in the efficient way.
- De-allocates devices.

File Management

A file system is normally organized into directories for easy navigation and usage. These directories may contain files and other directions.

An Operating System does the following activities for file management −

- Keeps track of information, location, uses, status etc. The collective facilities are often known as file system.
- Decides who gets the resources.
- Allocates the resources.
- De-allocates the resources.

Other Important Activities

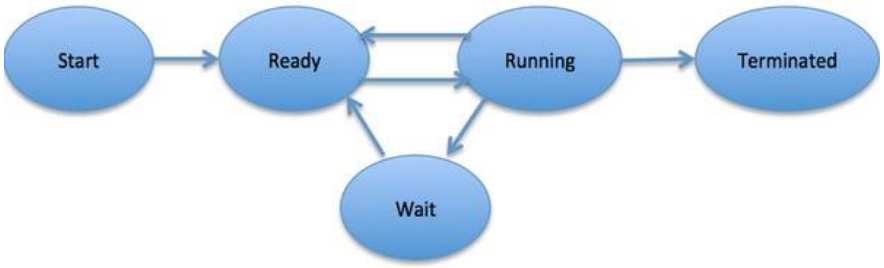
Following are some of the important activities that an Operating System performs −

- Security − By means of password and similar other techniques, it prevents unauthorized access to programs and data.
- Control over system performance − Recording delays between request for a service and response from the system.
- Job accounting − Keeping track of time and resources used by various jobs and users.
- Error detecting aids − Production of dumps, traces, error messages, and other debugging and error detecting aids.
- Coordination between other softwares and users − Coordination and assignment of compilers, interpreters, assemblers and other software to the various users of the computer systems.

b. What is Process and explain the process state diagram.

Ans: A process or a task is an instance of a program running in a computer. A process can initiate a subprocess, which is a called a child process. When a process executes, it passes through different states. These stages may differ in different operating systems, and the names of these states are also not standardized.

In general, a process can have one of the following five states at a time.



Start

This is the initial state when a process is first started/created.

Ready

The process is waiting to be assigned to a processor. Ready processes are waiting to have the processor allocated to them by the operating system so that they can run. Process may come into this state after Start state or while running it by but interrupted by the scheduler to assign CPU to some other process.

Running

Once the process has been assigned to a processor by the OS scheduler, the process state is set to running and the processor executes its instructions.

Waiting

Process moves into the waiting state if it needs to wait for a resource, such as waiting for user input, or waiting for a file to become available.

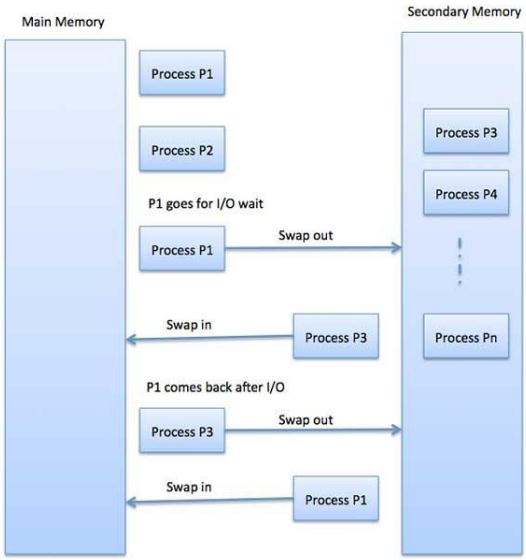
Terminated or Exit

Once the process finishes its execution, or it is terminated by the operating system, it is moved to the terminated state where it waits to be removed from main memory.

c. Define Swapping. Explain with suitable diagram

Ans: Swapping is a method in which the process should be swapped temporarily from the main memory to the backing store. It will be later brought back into the memory for continue execution.

Backing store is a hard disk or some other secondary storage device that should be big enough inorder to accommodate copies of all memory images for all users. It is also capable of offering direct access to these memory images.



Benefits of Swapping

Here, are major benefits/pros of swapping:

- It offers a higher degree of multiprogramming.
- Allows dynamic relocation. For example, if address binding at execution time is being used, then processes can be swap in different locations. Else in case of compile and load time bindings, processes should be moved to the same location.
- It helps to get better utilization of memory.
- Minimum wastage of CPU time on completion so it can easily be applied to a priority-based scheduling method to improve its performance.

d. State and explain The Banker algorithm

Ans: The Banker algorithm is a resource allocation and deadlock avoidance algorithm that tests for safety by simulating the allocation of predetermined maximum possible amounts of all resources, and then makes an "safty-state" check to test for possible deadlock conditions for all other pending activities, before deciding whether allocation should be allowed to continue.

The Data structures used by the Banker's Algorithm are:

Let 'n' be the number of processes in the system and 'm' be the number of resources types.

Available :

- It is a 1-d array of size 'm' indicating the number of available resources of each type.
- Available[j] = k means there are 'k' instances of resource type R_j

Max :

- It is a 2-d array of size 'n*m' that defines the maximum demand of each process in a system.
- Max[i, j] = k means process P_i may request at most 'k' instances of resource type R_j.

Allocation :

- It is a 2-d array of size 'n*m' that defines the number of resources of each type currently allocated to each process.
- Allocation[i, j] = k means process P_i is currently allocated 'k' instances of resource type R_j

Need :

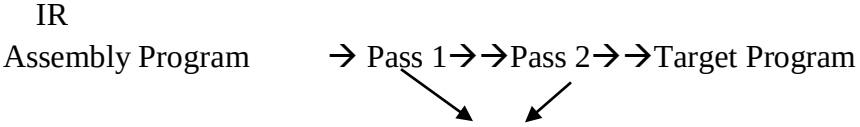
- It is a 2-d array of size 'n*m' that indicates the remaining resource need of each process.
- Need [i, j] = k means process P_i currently need 'k' instances of resource type R_j for its execution.

- Need [i, j] = Max [i, j] – Allocation [i, j]

$Allocation_i$ specifies the resources currently allocated to process P_i and $Need_i$ specifies the additional resources that process P_i may still request to complete its task.

e. What is an Assembler and define its function

Ans: An assembler is a program that converts assembly language into machine code. It takes the basic commands and operations from assembly code and converts them into binary code that can be recognized by a specific type of processor. Assemblers are similar to compilers in that they produce executable code. It generates instructions by evaluating the mnemonics (symbols) in operation field and find the value of symbol and literals to produce machine code. Now, if assembler do all this work in one scan then it is called single pass assembler, otherwise if it does in multiple scans then called multiple pass assembler.



Here assembler divide tasks in two passes:

Pass-1:

i. Define symbols and literals and remember them in symbol table and literal table respectively.
ii. Keep track of location counter
iii. Process pseudo-operations

Pass-2:

iv. Generate object code by converting symbolic op-code into respective numeric op-code
v. Generate data for literals and look for values of symbols

f. What is File ? explain file organization.

Ans: A file is a computer resource for recording data discretely in a computer storage device.

- File organization: There are four methods of organizing files on a storage media. This include:

- sequential,
- random,
- serial and
- indexed-sequential

1. Sequential file organization

- Records are stored and accessed in a particular order sorted using a key field.
- Retrieval requires searching sequentially through the entire file record by record to the end.
- Because the record in a file are sorted in a particular order, better file searching methods like the binary search technique can be used to reduce the time used for searching a file .
- Since the records are sorted, it is possible to know in which half of the file a particular record being searched is located, Hence this method repeatedly divides the set of records in the file into two halves and searches only the half on which the records is found.

Disadvantages of sequential file organization

- The sorting does not remove the need to access other records as the search looks for particular records.
- Sequential records cannot support modern technologies that require fast access to stored records.
- The requirement that all records be of the same size is sometimes difficult to enforce.

1. Random or direct file organization

- Records are stored randomly but accessed directly.
- To access a file stored randomly, a record key is used to determine where a record is stored on the storage media.
- Magnetic and optical disks allow data to be stored and accessed randomly.

Advantages of random file access

- Quick retrieval of records.
- The records can be of different sizes.

1. Serial file organization

- Records in a file are stored and accessed one after another.
- The records are not stored in any way on the storage medium this type of organization is mainly used on magnetic tapes.

Advantages of serial file organization

- It is simple
- It is cheap

Disadvantages of serial file organization

- It is cumbersome to access because you have to access all proceeding records before retrieving the one being searched.
- Wastage of space on medium in form of inter-record gap.
- It cannot support modern high speed requirements for quick record access.

1. Indexed-sequential file organization method

- Almost similar to sequential method only that, an index is used to enable the computer to locate individual records on the storage media. For example, on a magnetic drum, records are stored sequential on the tracks. However, each record is assigned an index that can be used to access it directly.

3. What is the function of a compiler and describe the phases of compiler.

Ans: A compiler is a special program that processes statements written in a particular programming language and turns them into machine language or "code" that a computer's processor uses. The compilation process is a sequence of various phases. Each phase takes input from its previous stage, has its own representation of source program, and feeds its output to the next phase of the compiler. Different phases of a compiler are

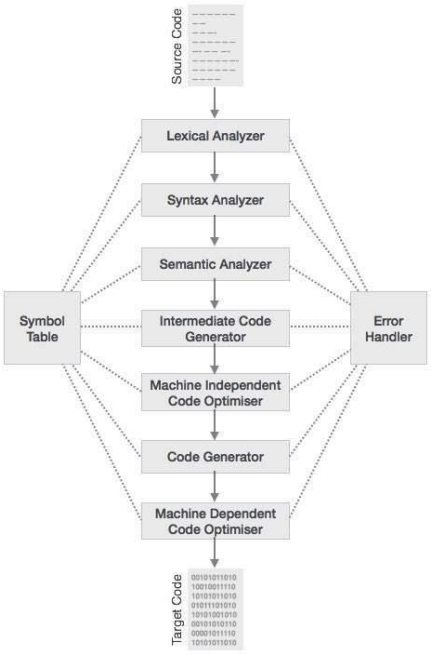
Lexical Analysis

The first phase of scanner works as a text scanner. This phase scans the source code as a stream of characters and converts it into meaningful lexemes. Lexical analyzer represents these lexemes in the form of tokens as:

<token-name, attribute-value>

Syntax Analysis

The next phase is called the syntax analysis or parsing. It takes the token produced by lexical analysis as input and generates a parse tree (or syntax tree). In this phase, token arrangements are checked against the source code grammar, i.e. the parser checks if the expression made by the tokens is syntactically correct.



Semantic Analysis

Semantic analysis checks whether the parse tree constructed follows the rules of language. For example, assignment of values is between compatible data types, and adding string to an integer. Also, the semantic analyzer keeps track of identifiers, their types and expressions; whether identifiers are declared before use or not etc. The semantic analyzer produces an annotated syntax tree as an output.

Intermediate Code Generation

After semantic analysis the compiler generates an intermediate code of the source code for the target machine. It represents a program for some abstract machine. It is in between the high-level language and the machine language. This intermediate code should be generated in such a way that it makes it easier to be translated into the target machine code.

Code Optimization

The next phase does code optimization of the intermediate code. Optimization can be assumed as something that removes unnecessary code lines, and arranges the sequence of statements in order to speed up the program execution without wasting resources (CPU, memory).

Code Generation

In this phase, the code generator takes the optimized representation of the intermediate code and maps it to the target machine language. The code generator translates the intermediate code into a sequence of (generally) re-locatable machine code. Sequence of instructions of machine code performs the task as the intermediate code would do.

Symbol Table

It is a data-structure maintained throughout all the phases of a compiler. All the identifier's names along with their types are stored here. The symbol table makes it easier for the compiler to quickly search the identifier record and retrieve it. The symbol table is also used for scope management.

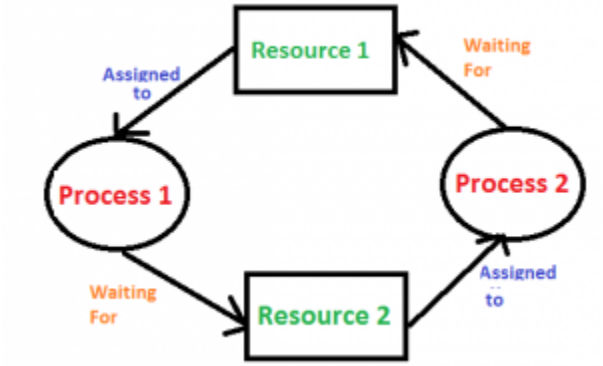
4. How deadlock occurs? How is it detected,recovered and prevented.

Ans: A deadlock is a situation in which two computer programs sharing the same resource are effectively preventing each other from accessing the resource, resulting in both programs ceasing to function. The earliest computer operating systems ran only one program at a time.

Deadlock detection and recovery: Deadlock Detection

1. If resources have single instance:

In this case for Deadlock detection we can run an algorithm to check for cycle in the Resource Allocation Graph. Presence of cycle in the graph is the sufficient condition for deadlock.



2. In the above diagram, resource 1 and resource 2 have single instances. There is a cycle R1 → P1 → R2 → P2. So, Deadlock is Confirmed.

3. If there are multiple instances of resources:

Detection of the cycle is necessary but not sufficient condition for deadlock detection, in this case, the system may or may not be in deadlock varies according to different situations.

Deadlock Recovery

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

Recovery method

1. Killing the process: killing all the process involved in the deadlock. Killing process one by one. After killing each process check for deadlock again keep repeating the process till system recover from deadlock.

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.

Deadlock Preventation: Deadlock prevention is a set of methods for ensuring that at least one of these necessary conditions cannot hold.

Mutual Exclusion: The mutual exclusion condition holds for non sharable devices. Sharable resources do not require mutual exclusive access and thus cannot be involved in a dead lock.

Hold and wait: To ensure that the hold and wait condition never occurs in the system, we must guaranty that whenever a process requests a resource it does not hold any other resources. There are two protocols to handle these problems such as one protocol that can be used requires each process to request and be allocated all its resources before it begins execution. The other protocol allows a process to request resources only when the process has no resource. These protocols have disadvantages like resource utilization may be low. And also starvation may be possible.

No Preemption:Alternatively if a process requests some resources, the operating system first check whether they are available. If they are, the operating system allocate them. If they are not available, operating system check whether they are allocated to some other process that is waiting for additional resources. If so, operating system preempt the desired resources from the waiting process and allocate them to the requesting process. If the resources are not either available or held by a waiting process, the requesting process must wait.

Circular Wait: Let R = {R1, R2, …….Rn}be the set of resource types. We assign to each resource type a unique integer number, which allows us to compare two resources and to determine whether one precedes another in our ordering. This can be ensure that this condition never holds by ordering of all resource type and to require that each process requests resource in an increasing order of enumeration.

5. Define device allocation technique. Explain the function of I/O sechduler

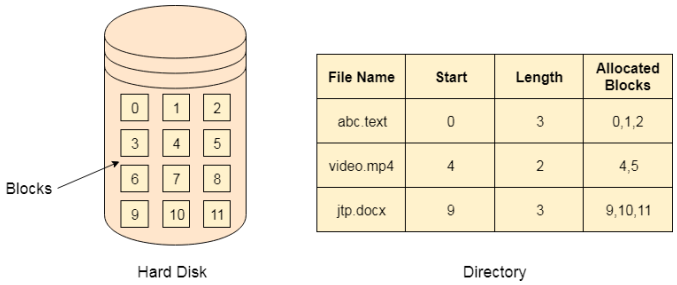
Ans: Devive allocation: There are various methods which can be used to allocate disk space to the files. Selection of an appropriate allocation method will significantly affect the performance and efficiency of the system. Allocation method provides a way in which the disk will be utilized and the files will be accessed.

There are following methods which can be used for allocation.

1. Contiguous Allocation.
2. Linked Allocation
3. FAT
4. Indexed Allocation
5. Linked Indexed Allocation
6. Multilevel Indexed Allocation
7. Inode

Contiguous Allocation:

If the blocks are allocated to the file in such a way that all the logical blocks of the file get the contiguous physical block in the hard disk then such allocation scheme is known as contiguous allocation. In the figure, there are three files in the directory. The starting block and the length of each file are mentioned in the table. We can check in the table that the contiguous blocks are assigned to each file as per its need

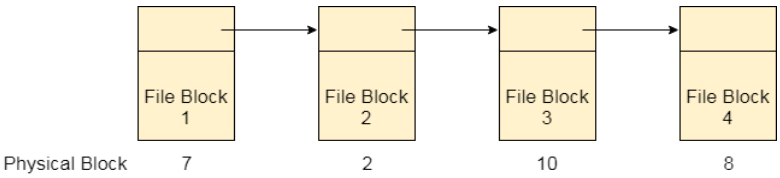


File Name	Start	Length	Allocated Blocks
abc.text	0	3	0,1,2
video.mp4	4	2	4,5
jtp.docx	9	3	9,10,11

Directory

Contiguous Allocation

Linked List allocation solves all problems of contiguous allocation. In linked list allocation, each file is considered as the linked list of disk blocks. However, the disks blocks allocated to a particular file need not



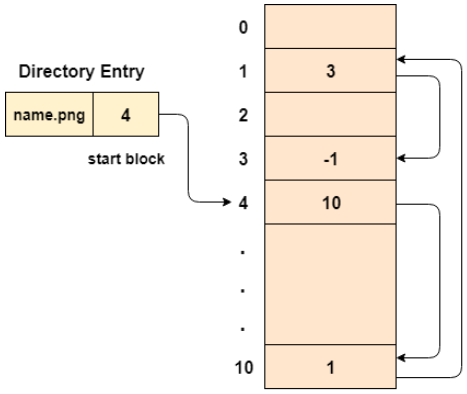
Linked List Allocation

to be contiguous on the disk. Each disk block allocated to a file contains a pointer which points to the next disk block allocated to the same file.

File Allocation Table :

Here a file allocation table is maintained, which gathers all the disk block links. The table has one entry for each disk block and is indexed by block number.

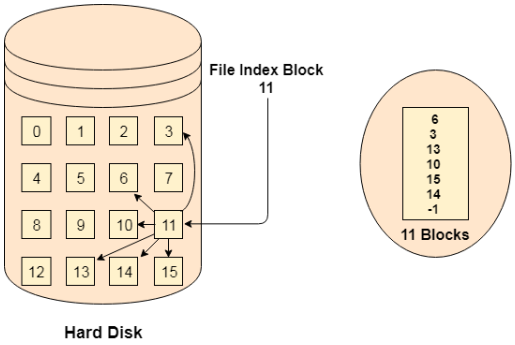
File allocation table needs to be cached in order to reduce the number of head seeks. Now the head doesn't need to traverse all the disk blocks in order to access one successive block.



File Allocation Table

Indexed Allocation

Indexed allocation scheme stores all the disk pointers in one of the blocks called as indexed block. Indexed block doesn't hold the file data, but it holds the pointers to all the disk blocks allocated to that particular file. Directory entry will only contain the index block address.



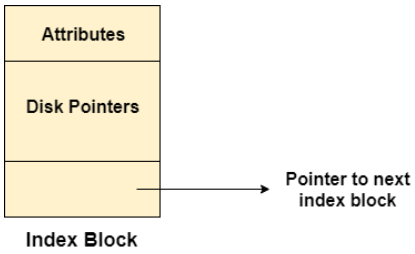
Linked Index Allocation

Single level linked Index Allocation

In index allocation, the file size depends on the size of a disk block. To allow large files, we have to link several index blocks together. In linked index allocation,

- Small header giving the name of the file
- Set of the first 100 block addresses
- Pointer to another index block

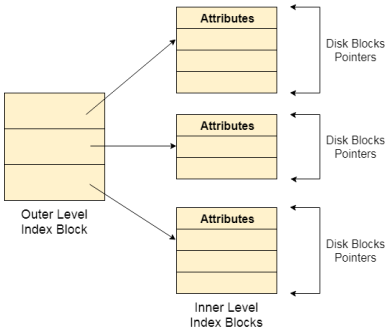
For the larger files, the last entry of the index block is a pointer which points to another index block. This is also called as linked schema.



Multilevel Index Allocation

In Multilevel index allocation, we have various levels of indices. There are outer level index blocks which contain the pointers to the inner level index blocks and the inner level index blocks contain the pointers to the file data.

- The outer level index is used to find the inner level index.
- The inner level index is used to find the desired data block.

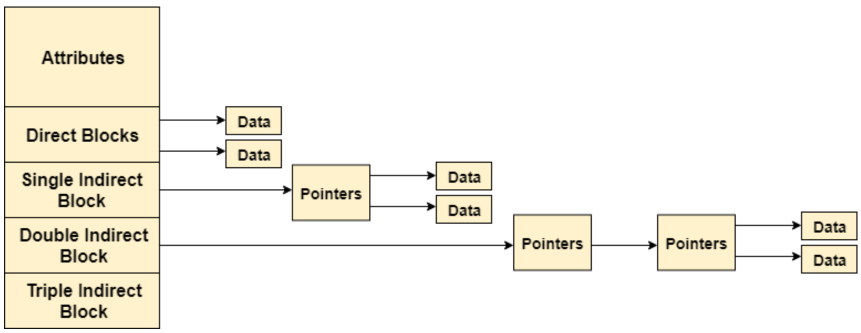


Inode

In UNIX based operating systems, each file is indexed by an Inode. Inode are the special disk block which is created with the creation of the file system. The number of files or directories in a file system depends on the number of Inodes in the file system.

An Inode includes the following information

1. Attributes (permissions, time stamp, ownership details, etc) of the file
2. A number of direct blocks which contains the pointers to first 12 blocks of the file.
3. A single indirect pointer which points to an index block. If the file cannot be indexed entirely by the direct blocks then the single indirect pointer is used.



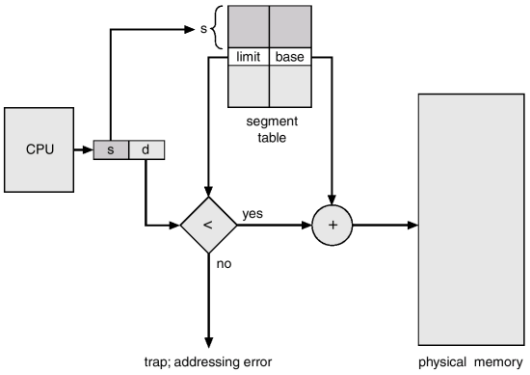
I/O schedulers can have many purposes depending on the goals; common purposes include the following:

- To minimize time wasted by hard disk seeks
- To prioritize a certain processes' I/O requests
- To give a share of the disk bandwidth to each running process
- To guarantee that certain requests will be issued before a particular deadline

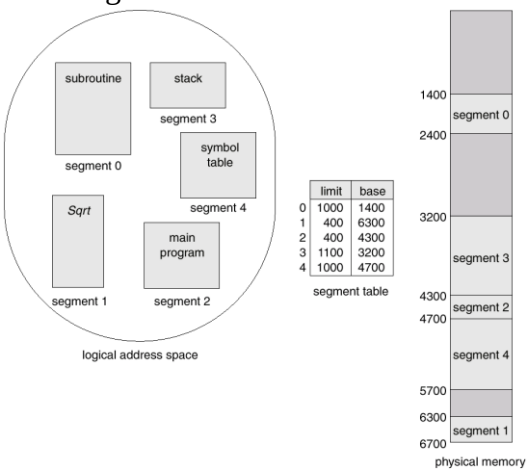
6. What is virtual memory? Explain using virtual memory in segmentation.

Ans: Virtual memory is a memory management capability of an operating system (OS) that uses hardware and software to allow a computer to compensate for physical memory shortages by temporarily transferring data from random access memory (RAM) to disk storage.
Segmentation is a memory management scheme that supports this user view of memory.
· A logical address space is a collection of segments. Each segment has a name and a length.
· The addresses specify both the segment name and the offset within the segment.
· The user therefore specifies each address by two quantities such as segment name and an offset.
For simplicity of implementation, segments are numbered and are referred to by a segment number, rather than by a segment name.
· Logical address consists of a two tuples:
$<$segment-number, offset$>$
· Segment table – maps two-dimensional physical addresses; each table entry has:
o Base – contains the starting physical address where the segments reside in memory.
o Limit – specifies the length of the segment.
· Segment-table base register (STBR) points to the segment table’s location in memory.
· Segment-table length register (STLR) indicates number of segments used by a program;

Segment number s is legal if s< STLR.



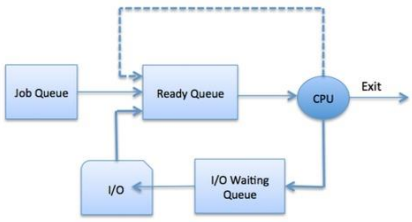
When the user program is compiled by the compiler it constructs the segments.
· The loader takes all the segments and assigned the segment numbers.
· The mapping between the logical and physical address using the segmentation technique is shown in above figure.
· Each entry in the segment table as limit and base address.
· The base address contains the starting physical address of a segment where the limit address specifies the length of the segment.
· The logical address consists of 2 parts such as segment number and offset.
· The segment number is used as an index into the segment table. Consider the example is given below.



7. State and explain job scheduling.

Ans: Job scheduling is the process of allocating system resources to many different tasks by an operating system (OS). The system handles prioritized job queues that are awaiting CPU time and it should determine which job to be taken from which queue and the amount of time to be allocated for the job. The Operating System maintains the following important process scheduling queues −

- Job queue − This queue keeps all the processes in the system.
- Ready queue − This queue keeps a set of all processes residing in main memory, ready and waiting to execute. A new process is always put in this queue.
- Device queues − The processes which are blocked due to unavailability of an I/O device constitute this queue.



Job Schedulers are special system software which handle process scheduling in various ways. Their main task is to select the jobs to be submitted into the system and to decide which process to run. Job Schedulers are of three types −

- Long-Term Scheduler
- Short-Term Scheduler
- Medium-Term Scheduler

Long Term Scheduler

It is also called a job scheduler. A long-term scheduler determines which programs are admitted to the system for processing. It selects processes from the queue and loads them into memory for execution. Process loads into the memory for CPU scheduling.

The primary objective of the job scheduler is to provide a balanced mix of jobs, such as I/O bound and processor bound. It also controls the degree of multiprogramming. If the degree of multiprogramming is stable, then the average rate of process creation must be equal to the average departure rate of processes leaving the system.

Short Term Scheduler

It is also called as CPU scheduler. Its main objective is to increase system performance in accordance with the chosen set of criteria. It is the change of ready state to running state of the process. CPU scheduler selects a process among the processes that are ready to execute and allocates CPU to one of them.

Short-term schedulers, also known as dispatchers, make the decision of which process to execute next. Short-term schedulers are faster than long-term schedulers.

Medium Term Scheduler

Medium-term scheduling is a part of swapping. It removes the processes from the memory. It reduces the degree of multiprogramming. The medium-term scheduler is in-charge of handling the swapped out-processes.

A running process may become suspended if it makes an I/O request. A suspended processes cannot make any progress towards completion. In this condition, to remove the process from memory and make space for other processes, the suspended process is moved to the secondary storage. This process is called swapping, and the process is said to be swapped out or rolled out. Swapping may be necessary to improve the process mix.