11.Os with its feature.

An operating system (OS) is system software that manages computer hardware and software resources, and provides common services for computer programs.

Feature:

1. Protected and Supervisor Mode
2. Program Execution
3. Manipulation of the File System
4. Handling I/O Operations
5. Error Handling
6. Resource Allocation
7. Information and Resource Protection

2.Advantage and disadvantage of OS

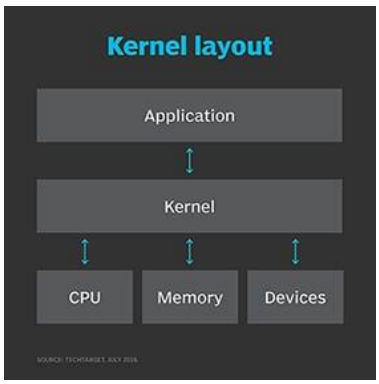
Advantages	Disadvantages
Computing Source	Expensive
User-Friendly Interface	System Failure
Resource Sharing	Highly Complex
Safeguard of Data	Virus Threats

3.PCB?

A Process Control Block in OS (PCB) is a data structure used by an operating system (OS) to manage and control the execution of processes. It contains all the necessary information about a process, including the process state, program counter, memory allocation, open files, and CPU scheduling information.

4.Importance of kernal.

The kernel is the essential foundation of a computer's operating system (OS)



In broad terms, an OS kernel performs three primary jobs.

1.It provides the interfaces needed for users and applications to interact with the computer.

2.It launches and manages applications.

3.It manages the underlying system hardware devices.

5.Feature of unix.

1.Free and Open-Source. ...

2.Extremely Flexible. ...

3.Lightweight Infrastructure. ...

4.Graphical User Interface (GUI) ...

5.End-to-end encryption. ...

6.Portable Environment. ...

7.Shell/ Command-line Interface. ...

8.Customized keyboard.

9. Frequent New Updates

10. Hierarchical File System

11. Multi-user and Multi-programming

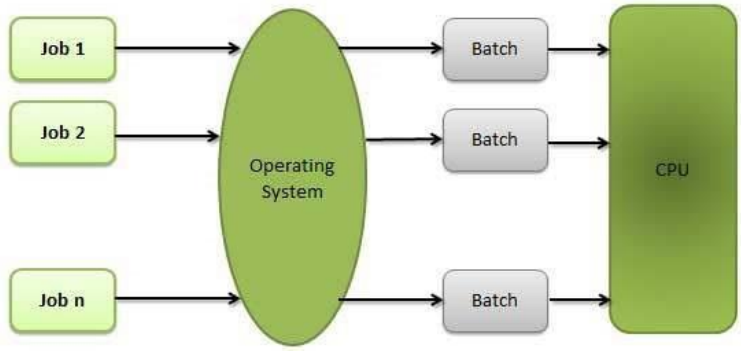
6.Batch processing.

Batch processing is a technique in which an Operating System collects the programs and data together in a batch before processing starts. An operating system does the following activities related to batch processing −

1.The OS defines a job which has predefined sequence of commands, programs and data as a single unit.

2.The OS keeps a number a jobs in memory and executes them without any manual information.

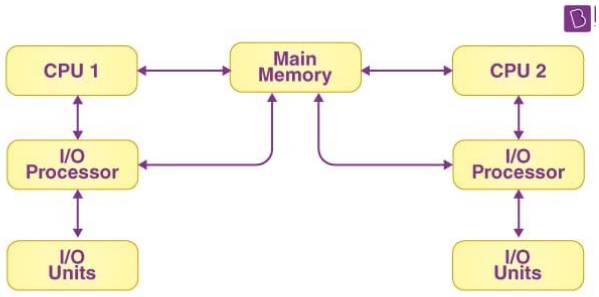
3.Jobs are processed in the order of submission, i.e., first come first served fashion.



7.Multi-processor.

Multiprocessor operating systems are used in operating systems to boost the performance of multiple CPUs within a single computer system.

Multiple CPUs are linked together so that a job can be divided and executed more quickly. When a job is completed, the results from all CPUs are compiled to provide the final output.

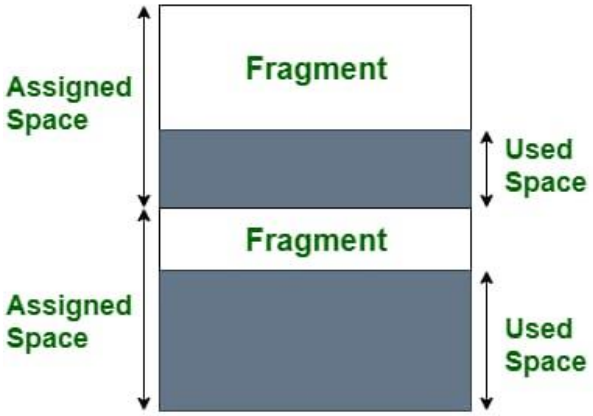


Working of Multiprocessor System

8.Fragmentation Internal and External.

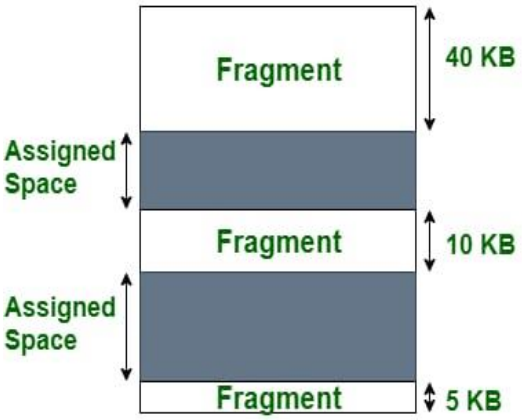
There are two types of fragmentation in OS which are given as Internal fragmentation and External fragmentation.

1. **Internal Fragmentation** - Internal fragmentation happens when the memory is split into mounted-sized blocks. Whenever a method is requested for the memory, the mounted-sized block is allotted to the method.



Internal Fragmentation

2.External Fragmentation- External fragmentation happens when there's a sufficient quantity of area within the memory to satisfy the memory request of a method.



10.important of Semaphore

Semaphores are integer variables that are used to solve the critical section problem by using two atomic operations, wait and signal that are used for process synchronization.

The definitions of wait and signal are as follows −

Wait
The wait operation decrements the value of its argument S, if it is positive. If S is negative or zero, then no operation is performed.

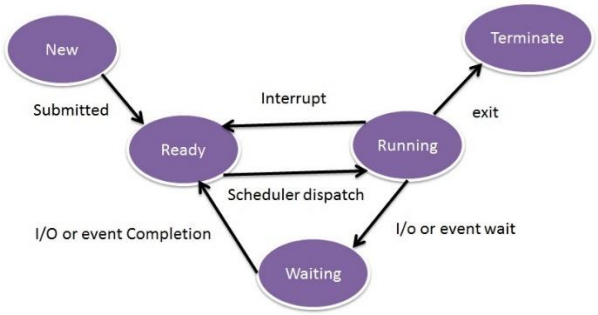
```
1. wait(S)
{
  while (S<=0);

  S--;
}
```

2. Signal
The signal operation increments the value of its argument S.

```
signal(S)
{
  S++;
}
```

14.process state.

When an Operating system process executes, Process may change its state. Process may be in one of the following state:



1.New State- As soon as a new process is created, process is in the initial state.

2.**Ready State-**Process is waiting to be assigned CPU time. Operating system allocates the processor time to the ready process.

3.Running State-Once the process gets the CPU time, process instructions are being executed.

4.Waiting State- Running process needs some resources or some events to occur (such as I/O complete, waiting for user input and waiting for input file).

5.Terminate State- the process has finished execution

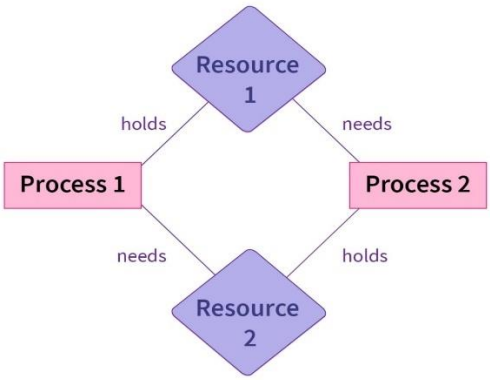
16.Difference paging and segmentation.

Paging	Segmentation
1. A program is divided into fixed-size pages.	1. A program is divided into variable size segments.
2. The operating system is in charge of paging.	2. Segmentation is the responsibility of the user/compiler.
3. In terms of memory access, paging is faster than segmentation.	3. Segmentation is slower than paging
4. It suffers from internal fragmentation.	4. It suffers from external fragmentation.
5. Logical address space is divided into a page number and page offset.	5. Logical memory address space is divided into a segment number and segment offset.
6. To keep track of virtual pages, paging requires a page table.	6. To keep track of virtual pages, segmentation requires a segmentation table.
7. The page table has one entry for each virtual page.	7. The segment table has one entry for each virtual segment.
8. A free frame list must be maintained by the operating system.	8. A list of holes in the main memory must be kept by the operating system.
9. Paging is invisible to the user	9. Segmentation is visible to the user.
10. Page table entry has frame number and additional protected bits for pages	10. Segment table entry has a limit, base, and may contain some bits for the protection of segments.

15.Dead lock,condition.

Deadlock is a situation which involves the interaction of more than one resources and processes with each other.

There are four conditions necessary for the occurrence of a deadlock.



1.Mutual Exclusion: Only one process can use a resource at any given time i.e. the resources are non-sharable.

2.Hold and wait: A process is holding at least one resource at a time and is waiting to acquire other resources held by some other process.

3.No preemption: The resource can be released by a process voluntarily i.e. after execution of the process.

4.Circular Wait: A set of processes are waiting for each other in a circular fashion.

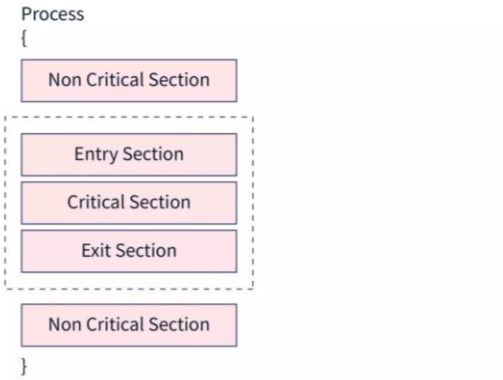
18.Critical section.

1.Critical Section refers to the segment of code or the program which tries to access or modify the value of the variables in a shared resource.

2.The section above the critical section is called the Entry Section. The process that is entering the critical section must pass the entry section.

3.The section below the critical section is called the Exit Section.

4.The section below the exit section is called the Reminder Section and this section has the remaining code that is left after execution.

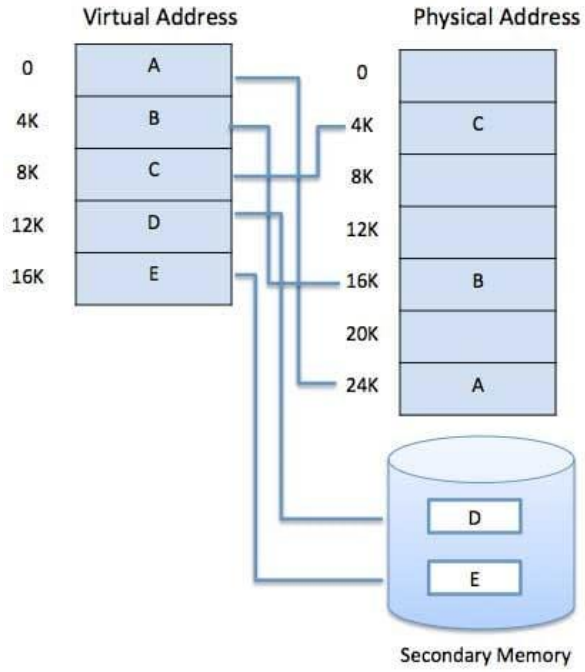


21.Process and thread different.

Parameter	Process	Thread
Definition	Process means a program is in execution.	Thread means a segment of a process.
Lightweight	The process is not Lightweight.	Threads are Lightweight.
Termination time	The process takes more time to terminate.	The thread takes less time to terminate.
Creation time	It takes more time for creation.	It takes less time for creation.
Communication	Communication between processes needs more time compared to thread.	Communication between threads requires less time compared to processes.
Context switching time	It takes more time for context switching.	It takes less time for context switching.
Resource	Process consume more resources.	Thread consume fewer resources.
Treatment by OS	Different process are tread separately by OS.	All the level peer threads are treated as a single task by OS.
Memory	The process is mostly isolated.	Threads share memory.
Sharing	It does not share data	Threads share data with each other.

23.Virtual memory with example.

A computer can address more memory than the amount physically installed on the system. This extra memory is actually called virtual memory and it is a section of a hard disk that's set up to emulate the
computer's RAM.
A basic example is given below –



24.First fit, best fit, worst fit.

1.First fit-Process allocate frist available portion of memory

2.Best fit- Process allocate best available portion of memory due to which memory westage is very much less

3.Worst fit-Process allocate large portion of memory

25.Long term, short term, medium term.

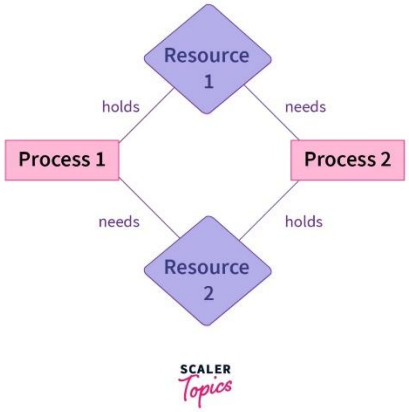
S.N.	Long Term	Short Term	Medium Term
1	It is job scheduler	It is CPU Scheduler	It is swapping
2	Speed is less than short term scheduler	Speed is very fast	Speed is in between both
3	It controls degree of multiprogramming	Less control over degree of multiprogramming	Reduce the degree of multiprogramming.
4	Absent or minimal in time sharing system.	Minimal in time sharing system.	Time sharing system use medium term scheduler.
5	It select processes from pool and load them into memory for execution.	It select from among the processes that are ready to execute.	Process can be reintroduced into memory and its execution can be continued.
6	Process state is (New to Ready)	Process state is (Ready to Running)	swapping
7	Select a good process, mix of I/O bound and CPU bound.	Select a new process for a CPU quite frequently.	

Important question:

1. Reason for dead lock.

In an operating system, a deadlock occurs when a process or thread enters a waiting state because a requested system resource is held by another waiting process, which in turn is waiting for another resource held by another waiting process.

There are four conditions necessary for the occurrence of a deadlock.



1.Mutual Exclusion: Only one process can use a resource at any given time i.e. the resources are non-sharable.

2.Hold and wait: A process is holding at least one resource at a time and is waiting to acquire other resources held by some other process.

3.No preemption: The resource can be released by a process voluntarily i.e. after execution of the process.

4.Circular Wait: A set of processes are waiting for each other in a circular fashion.

2.Semaphore (Binary Semaphore)

Binary semaphores are synchronization mechanisms that have integer values that range from 0 (zero) to 1 (one).

Binary Semaphores have two operations namely wait(P) and signal(V) operations. Both operations are atomic. Semaphore(s) can be initialized to zero or one.

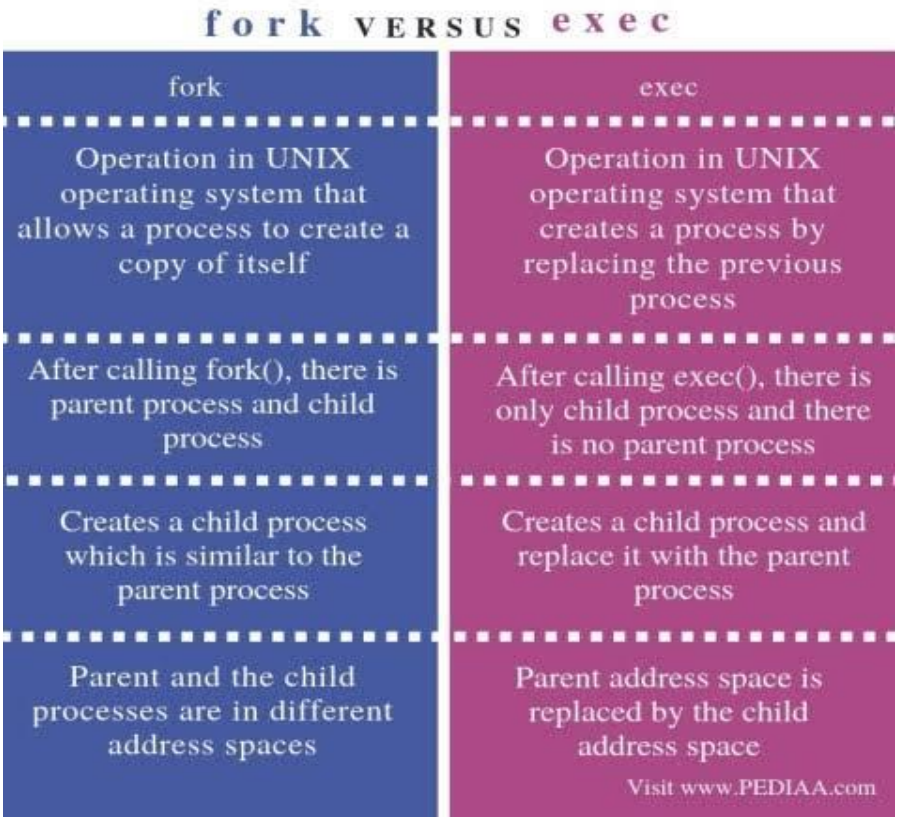
Wait
The wait operation decrements the value of its argument p, if it is positive. If p is negative or zero, then no operation is performed.

```
wait(p)
{
  while (p<=0);

  p--;
}
```

Signal
The signal operation increments the value of its argument v.

```
signal(v)
{
  v++;
}
```

3.differnce between FORK() and Exec().



4.monolithic and microlithic.

Microkernel	vs	Monolithic
In microkernel user services and kernel services are kept in separate address space		In Monolithic kernel user and monolithic services are kept in same address space
Os is complex in design		Os is easy to implement
Microkernel is smaller in size		Os is larger in size
Easyr to add new feature		Difficult to add new feature
Execution Speed is slow		Execution Speed is high
Example Mac os X.		Example . Windows 95

5.primitive and non primitive.

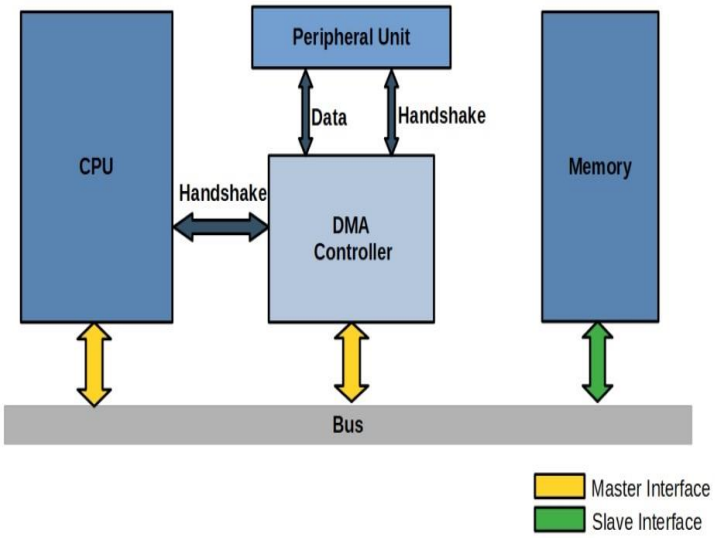
Parameter	Preemptive Scheduling	Non-Preemptive Scheduling
Basic	In this resources (CPU Cycle) are allocated to a process for a limited time.	Once resources (CPU Cycle) are allocated to a process, the process holds it till it completes its burst time or switches to waiting state.
Interrupt	Process can be interrupted in between.	Process cannot be interrupted until it terminates itself or its time is up.
Starvation	If a process having high priority frequently arrives in the ready queue, low priority process may starve.	If a process with long burst time is running CPU, then later coming process with less CPU burst time may starve.
Overhead	It has overheads of scheduling the processes.	It does not have overheads.
Flexibility	flexible	rigid
Cost	cost associated	no cost associated
CPU Utilization	In preemptive scheduling, CPU utilization is high.	It is low in non preemptive scheduling.
Examples	Examples of preemptive scheduling are Round Robin and Shortest Remaining Time First.	Examples of non-preemptive scheduling are First Come First Serve and Shortest Job First.

6.Process from deadlock.

---Same as question no 15

7.DMA

Direct memory access (DMA) is a method that allows an input/output (I/O) device to send or receive data directly to or from the main memory, bypassing the CPU to speed up memory operations.



8.kernal level thread.

Kernel-level threads are handled by the operating system directly and the thread management is done by the kernel.The context information for the process as well as the process threads is all managed by the kernel. Because of this, kernel-level threads are slower than user-level threads.

Advantages of Kernel-Level Threads
Some of the advantages of kernel-level threads are as follows −

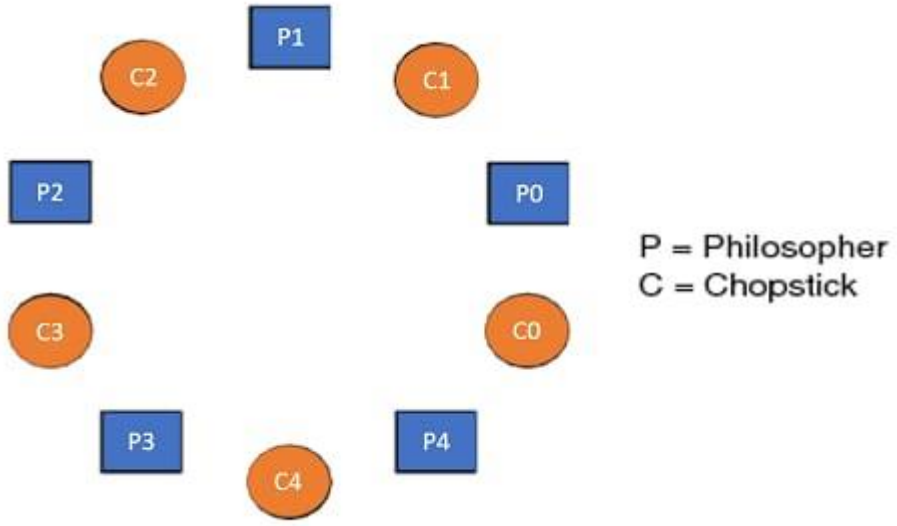
1.Multiple threads of the same process can be scheduled on different processors in kernel-level threads.
2.The kernel routines can also be multithreaded.
3.If a kernel-level thread is blocked, another thread of the same process can be scheduled by the kernel.

Disadvantages of Kernel-Level Threads
Some of the disadvantages of kernel-level threads are as follows −

1.A mode switch to kernel mode is required to transfer control from one thread to another in a process.
2.Kernel-level threads are slower to create as well as manage as compared to user-level threads.

9 er ta sobai nije korbi

The Dining Philosopher Problem – The Dining Philosopher Problem states that K philosophers seated around a circular table with one chopstick between each pair of philosophers. There is one chopstick between each philosopher. A philosopher may eat if he can pick up the two chopsticks adjacent to him. One chopstick may be picked up by any one of its adjacent followers but not both.



Code-

```
do{
   wait(cs[i]);
   wait(cs[i+1]%5);
//eat
signal(cs[i]);
signal(cs[i+1]%5);
}while(true);
```