## 1. What is deadlock? Characteristics of deadlock

Deadlock is a situation where a set of processes are blocked because each process is holding a resource and waiting for another resource acquired by some other process.

The four necessary characteristics for deadlock in OS are :

- Mutual Exclusion
  Mutual exclusion means that a resource can be used by only one process at a time.
- Hold and Wait
  The processes involved in deadlock acquires at least one resource and wait for at least another resource that has been acquired by another process.
- No preemption
  Resources held by a process cannot be preempted. A resource can be released only voluntarily by the process holding it, after that process has completed its task.
- Circular Wait
  A circular chain or a circular relationship exists among the processes involved in a deadlock. In such a chain each process waits for a resource that is held by the next process in the chain.

## 2. Recovery from deadlock

**1. Process Termination:**
To eliminate the deadlock, we can simply kill one or more processes. For this, we use two methods:

- **(a). Abort all the Deadlocked Processes:**
  Aborting all the processes will certainly break the deadlock, but with a great expense. The deadlocked processes may have computed for a long time and the result of those partial computations must be discarded and there is a probability to recalculate them later.

- (**b). Abort one process at a time until deadlock is eliminated:**
  Abort one deadlocked process at a time, until deadlock cycle is eliminated from the system. Due to this method, there may be considerable overhead, because after aborting each process, we have to run deadlock detection algorithm to check whether any processes are still deadlocked.

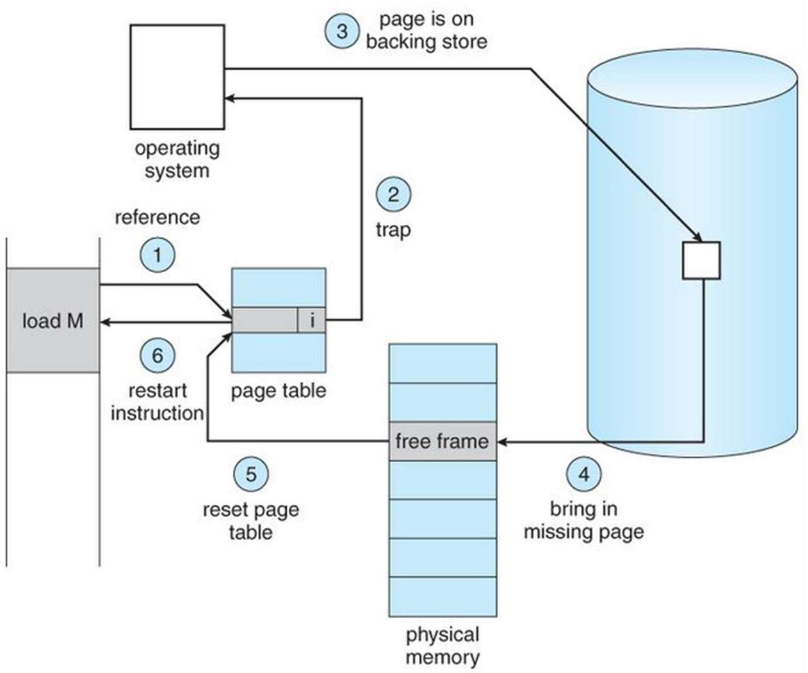
**2. Resource Preemption:**
To eliminate deadlocks using resource preemption, we preempt some resources from processes and give those resources to other processes. This method will raise three issues –
**(a). Selecting a victim:**
We must determine which resources and which processes are to be preempted and also the order to minimize the cost.
(**b). Rollback:**
We must determine what should be done with the process from which resources are preempted. One simple idea is total rollback. That means abort the process and restart it.
**(c). Starvation:**
In a system, it may happen that same process is always picked as a victim. As a result, that process will never complete its designated task. This situation is called Starvation and must be avoided.

## 3. Bankers algorithm

## 4. Resource allocation and wait for graph

## 5. With diagram explain steps involved in page fault

A page fault occurs when a program attempts to access data or code that is in its address space, but is not currently located in the system RAM. So when page fault occurs then following sequence of events happens :

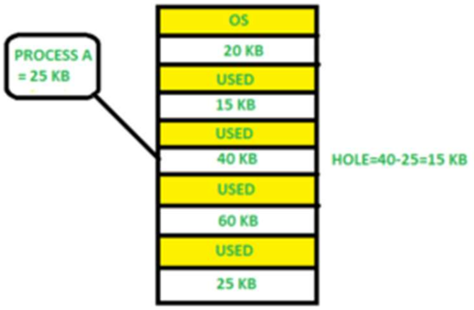


- The computer hardware traps to the kernel and program counter (PC) is saved on the stack. Current instruction state information is saved in CPU registers.
- An assembly program is started to save the general registers and other volatile information to keep the OS from destroying it.
- Operating system finds that a page fault has occurred and tries to find out which virtual page is needed. Some times hardware register contains this required information. If not, the operating system must retrieve PC, fetch instruction and find out what it was doing when the fault occurred.
- Once virtual address caused page fault is known, system checks to see if address is valid and checks if there is no protection access problem.
- If the virtual address is valid, the system checks to see if a page frame is free. If no frames are free, the page replacement algorithm is run to remove a page.
- If frame selected is dirty, page is scheduled for transfer to disk, context switch takes place, fault process is suspended and another process is made to run until disk transfer is completed.
- As soon as page frame is clean, operating system looks up disk address where needed page is, schedules disk operation to bring it in.
- When disk interrupt indicates page has arrived, page tables are updated to reflect its position, and frame marked as being in normal state.
- Faulting instruction is backed up to state it had when it began and PC is reset. Faulting is scheduled, operating system returns to routine that called it.

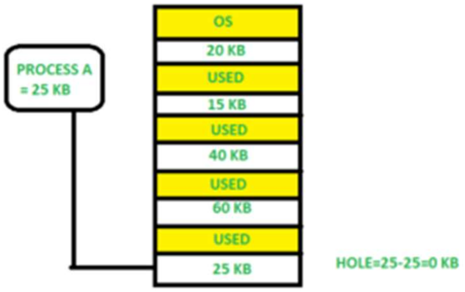
- Assembly Routine reloads register and other state information, returns to user space to continue execution.

## 6. Memory allocation stratergies

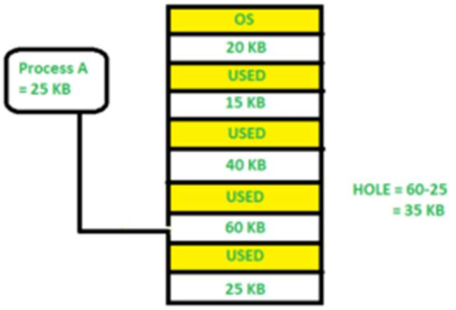
**1. First Fit**: In the first fit, the partition is allocated which is the first sufficient block from the top of Main Memory. It scans memory from the beginning and chooses the first available block that is large enough. Thus it allocates the first hole that is large enough.



**2. Best Fit** Allocate the process to the partition which is the first smallest sufficient partition among the free available partition. It searches the entire list of holes to find the smallest hole whose size is greater than or equal to the size of the process.



**3. Worst Fit** Allocate the process to the partition which is the largest sufficient among the freely available partitions available in the main memory. It is opposite to the best-fit algorithm. It searches the entire list of holes to find the largest hole and allocate it to process.



## 7. Write a shell script using for loop to reverse a string.

```
s=”Hello”
strlen=${#s}
for (( i=$strlen-1; i>=0; i--));
do
    revstr=$revstr${s:$i:1}
done
echo “Original String : $s”
echo “Reversed String : $revstr”
```

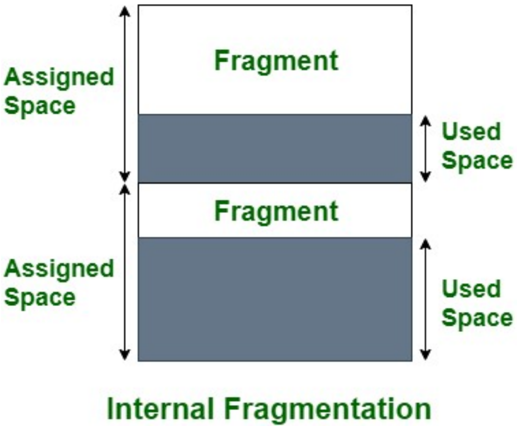
## 8. Write a script to check remote server connectivity at different frequencies.

## 9. Write a note on fragmentation.

As processes are loaded and removed from memory, the free memory space is broken into little pieces. It happens after sometimes that processes cannot be allocated to memory blocks considering their small size and memory blocks remains unused. This problem is known as Fragmentation.

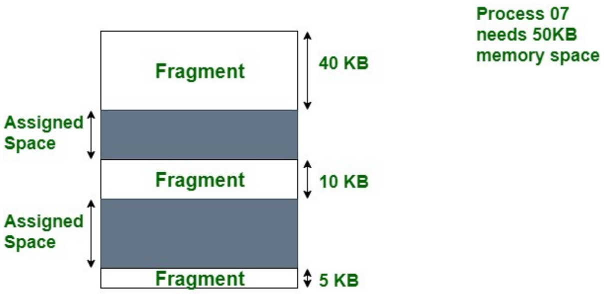
There are two types of fragmentation in OS which are given as Internal fragmentation, and External fragmentation.

**1. Internal Fragmentation:**
The difference between allotted and requested memory is called internal fragmentation.



***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. However, the process’s memory request cannot be fulfilled because the memory offered is in a non-contiguous manner. Whether you apply a first-fit or best-fit memory allocation strategy it’ll cause external fragmentation.

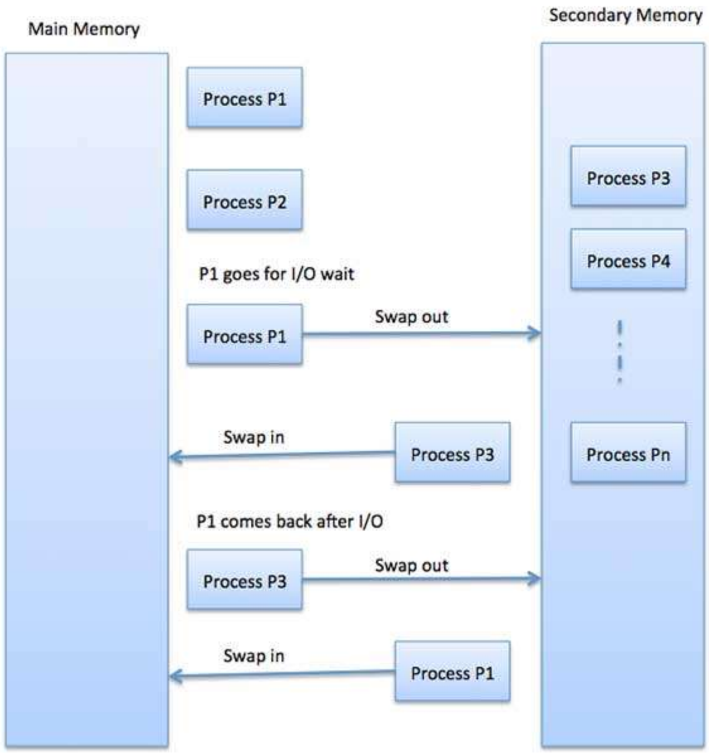


1. *External Fragmentation*

## 10. Explain the concept of swapping with diagram.

Swapping is a mechanism in which a process can be swapped temporarily out of main memory (or move) to secondary storage (disk) and make that memory available to other processes. At some later time, the system swaps back the process from the secondary storage to main memory.

Though performance is usually affected by swapping process but it helps in running multiple and big processes in parallel and that's the reason **Swapping is also known as a technique for memory compaction**.



The total time taken by swapping process includes the time it takes to move the entire process to a secondary disk and then to copy the process back to memory, as well as the time the process takes to regain main memory.

## 11. What is multithreading? Explain different multithreading modules with diagram.

Multi threading-It is a process of multiple threads executes at same time.

Multi threading model are of three types.

Many to many model.

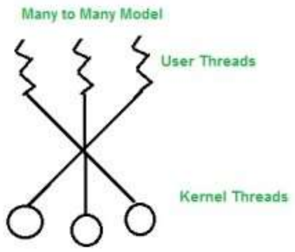
Many to one model.

one to one model.

**Many to Many Model**

In this model, we have multiple user threads multiplex to same or lesser number of kernel level threads. Number of kernel level threads are specific to the machine, advantage of this model is if a user thread is blocked we can schedule others user thread to other kernel thread. Thus, System doesn't block if a particular thread is blocked.

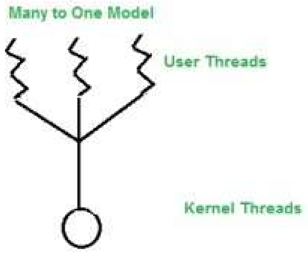
It is the best multi threading model.



**Many to One Model**

In this model, we have multiple user threads mapped to one kernel thread. In this model when a user thread makes a blocking system call entire process blocks. As we have only one kernel thread and only one user thread can access kernel at a time, so multiple threads are not able access multiprocessor at the same time.

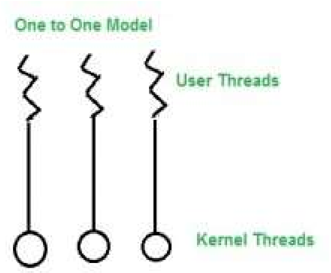
The thread management is done on the user level so it is more efficient.



**One to One Model**

In this model, one to one relationship between kernel and user thread. In this model multiple thread can run on multiple processor. Problem with this model is that creating a user thread requires the corresponding kernel thread.

As each user thread is connected to different kernel , if any user thread makes a blocking system call, the other user threads won’t be blocked.



## 12.Demonstrate decision making and iterative scripts with an example.

### if else statement

```
echo “enter your age”

read age

if [“$age” –ge 18];

then

        echo  ”You are eligible to vote”

else

        echo ”You are younger!!”

fi
```

### output

```
Enter your age:

19

You are eligible to vote
```

## For loop

```
for i in { 1..10 }

do

        echo $i

done
```

## output

```
1

2

3

4

.

.

.

8

9

10
```

## Do while loop

```
i=10

while [ $i –ge 0 ];

do

        echo $i

        let i--;

done
```

## output

```
10

9

8
```

```
.
.
1
0
```

## Case statement

```
echo “Enter the name of the state:”
read state
case $state in
        “Uttar Pradesh”)
                echo “Capital is Lucknow”
        ;;
        “Punjab”)
                echo “Capital is chandigarh”
        ;;
        “Bihar”)
                echo “Capital is Patna”
        ;;
        “Rajasthan”)
                echo “Capital is Jaipur”
        ;;
        “Madhya Pradesh”)
                echo “Capital is Bhopal”
        ;;
        *)
                echo “You discovered an unknown state”
        ;;
esac
```

## Output

```
Enter the name of the State:

Rajasthan

Capital is Jaipur
```