# Need of Memory Management

Memory needs to be allocated to ensure a reasonable supply of ready processes to consume available processor time

## Logical Address

is generated by CPU while a program is running. The logical address is virtual address as it does not exist physically, therefore, it is also known as Virtual Address.

## Physical Address

identifies a physical location of required data in a memory. The user never directly deals with the physical address but can access by its corresponding logical address.

# Memory Management Requirements

## Relocation

The programmer does not know where the program will be placed in memory when it is executed,

## Protection

Usually, a program in one process cannot branch to an instruction in another process or access the data area of another process. The processor must be able to abort such instructions at the point of execution.

## Sharing

Allow several processes to access the same portion of memory cause any protection mechanism must have the flexibility to allow several processes to access the same portion of main memory.

## Logical organisation

Memory is usually organized linearly and program is in modules and instruction in modules is execute type only and the data modules are either read or write and here some of the modules are either public or private

## Physical organisation

it is clear that the task of moving information between the two levels of memory should be a system responsibility. This task is the essence of memory management.

## Memory Allocation

In the multiprogramming, the multiple users can share the memory simultaneously. So, the memory is allocated to various processes depending on their requirements.

## Contiguous Memory Allocation

Contiguous memory allocation is a memory allocation method that allocates a single contiguous section of memory to a process or a file here when any user process request for the memory a single section of the contiguous memory block is given to that process according to its need.

# Fixed Partitioning

Any process whose size is less than or equal to the partition size can be loaded into an available partition

## Problem faced

* Internal n external fragmentation occurs
* Limit process size:
* Limitation on Degree of Multiprogramming:

# Variable/Dynamic Partitioning

Partitions are of variable length and number, here the Process is allocated exactly as much memory as required

## Problem faced

External fragmentation occurs

Difficult Implementation:

# Non-Contiguous Memory allocation

The available free memory space is scattered here and there and all the free memory space is not at one place. So, this is time-consuming. This technique reduces the wastage of memory which leads to internal and external fragmentation.

## Paging

Paging is a memory management scheme that eliminates the need for contiguous allocation of physical memory. This scheme permits the physical address space of a process to be non – contiguous.

### Advantages:

It is independent of external fragmentation.

### Disadvantages:

1. It makes the translation very slow as main memory access two times.
2. A page table is a burden over the system which occupies considerable space.
3. It may suffer from internal fragmentation as pages are of fixed size.

## Segmentation

Is a programmer view of the memory where instead of dividing a process into equal size partition we divided according to program into partition called segments. A logical address space (user program) is a collection of segments.

A segment is a logical unit such as:

main program, procedure, function, method, object, local variables, global variables, common block, stack, symbol table, arrays etc. are different segments of a program

### Advantages of Segmentation –

* No Internal fragmentation.
* Segment Table consumes less space in comparison to Page table in paging.

### Disadvantage of Segmentation –

As processes are loaded and removed from the memory, the free memory space is broken into little pieces, causing External fragmentation.

## Internal fragmentation

Out of total memory the one left out space is internal fragmentation.

## External fragmentation

It arises when free memory is separated into small blocks and is interspersed by allocated memory. It is a weakness of certain storage allocation algorithms, when they fail to order memory used by programs efficiently.

## Compaction

Compaction is a process in which the free space is collected in a large memory chunk to make some space available for processes. In Dynamic partitioning, swapping creates multiple fragments in the memory because of the processes moving in and out. Compaction refers to combining all the empty spaces together and processes.

# Memory Allocation Algorithms

## First-fit:

the first hole that is found to be large enough for a process to accommodate is selected.

## Best-fit:

This method needs the list of free holes to be sorted according to their size. Then the smallest hole that is large enough for the process to accommodate is selected from the list of free holes. This strategy reduces the wastage of memory as it does not allocate a hole of larger size which leaves some amount of memory even after the process accommodates the space.

## Worst-fit:

This method requires the entire list of free holes to be sorted. Here, again the largest hole among the free holes is selected. This strategy leaves the largest leftover hole which may be useful for the other process.

# Virtual Memory

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.

The main visible advantage of this scheme is that programs can be larger than physical memory. Virtual memory serves two purposes. First, it allows us to extend the use of physical memory by using disk. Second, it allows us to have memory protection, because each virtual address is translated to a physical address.

## Demand Paging

when a process is swapped in, its pages are not swapped in all at once. Its mainly implemented by virtual memory. Initially only those pages are loaded which will be required the process immediately.

## Page Fault Service Time

The time taken to service the page fault is called as page fault service time. The page fault service time includes the time taken to perform all the above six steps.

Let Main memory access time is: m, Page fault service time is: s, Page fault rate is: p

Then, Effective memory access time = (p\*s) + (1-p) \*m

# Deadlocks

A Deadlock is a situation where each of the computer process waits for a resource which is being assigned to some another process. In this situation, none of the process gets executed since the resource it needs, is held by some other process which is also waiting for some other resource to be released.

## Necessary conditions for Deadlock prevention

### Mutual exclusion:

only one process at a time can use a resource

### Hold and wait:

a process holding at least one resource is waiting to acquire additional resources held by other processes

### No preemption:

a resource can be released only voluntarily by the process holding it, after that process has completed its task

### Circular wait:

If process are there from P1 to P5 the each wait for each other like P0 to P1 , P1 to P2 like that.

# 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. This is a feature of a multitasking operating system and allows a single CPU to be shared by multiple processes.

the PCB remove the process from CPU to ready state cause a high priority process has come there the address is saved till that point so that if we put that process in CPU again it will start from there only not frm start

# In pre-emptive n non-pre-emptive which is better,

preemptive is better cause CPU utilization is more n Waiting n response time is less as compared n also Preemptive Scheduling is prioritized. The highest priority process is a process that is currently utilized.

# In Contiguous n non-contiguous which is better

non- contiguous is better cause Contiguous memory allocation allocates consecutive blocks of memory to a file/process, Faster in Execution. And It is easier for the OS to control.

# In deadlock prevention and avoidance which is better,

avoidance is better in deadlock avoidance, the system requires information on the existing resources, resource availability and resource requests to find whether the system is in a safe or unsafe state and makes sure it don’t go in unsafe state.

# What is Process Scheduling?

The process scheduling is the activity of the process manager that handles the removal of the running process from the CPU and the selection of another process on the basis of a particular strategy. Process scheduling is an essential part of a Multiprogramming operating system.

# What is Pre-emptive and Non-Pre-emptive scheduling?

In Pre-emptive Priority Scheduling, at the time of arrival of a process in the ready queue, its Priority is compared with the priority of the other processes present in the ready queue as well as with the one which is being executed by the CPU at that point of time. The One with the highest priority among all the available processes will be given the CPU next. In the Non Pre-emptive Priority scheduling, The Processes are scheduled according to the priority number assigned to them. Once the process gets scheduled, it will run till the completion. Generally, the lower the priority number, the higher is the priority of the process.

# What is the problem with priority scheduling?

A major problem with priority scheduling is indefinite blocking or starvation. A solution to the problem of indefinite blockage of the low-priority process is aging. Aging is a technique of gradually increasing the priority of processes that wait in the system for a long period of time.

# Different Process States

NEW- The process is being created.

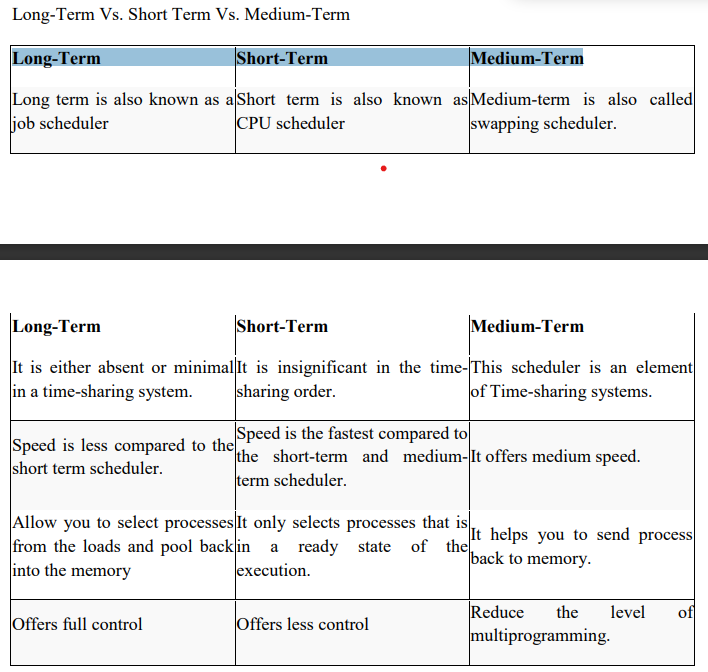
READY- The process is waiting to be assigned to a processor.

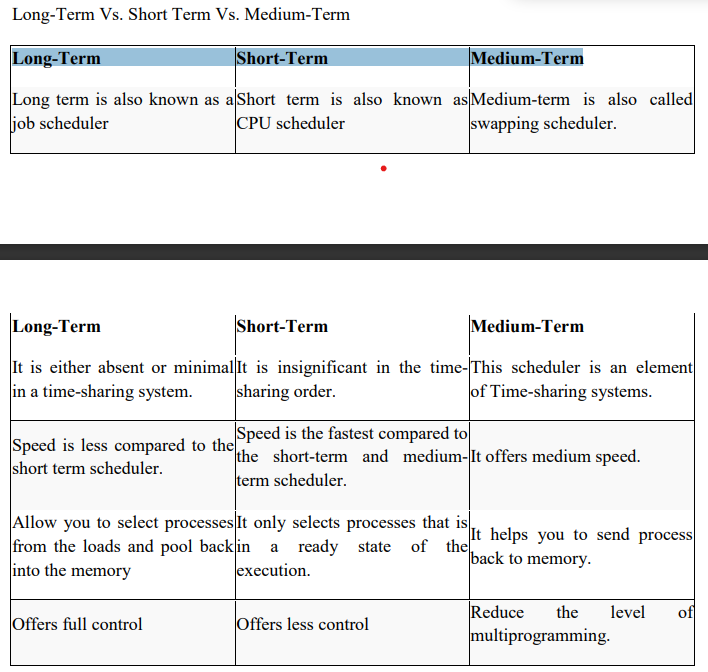
RUNNING- Instructions are being executed.

WAITING- The process is waiting for some event to occur (such as an I/O completion or reception of a signal).

TERMINATED- The process has finished execution.

# Difference between Schedulers

****

****

# In preemptive RR n SRTF

# Non-prremptive SJF n FCFS

# Starvation

A situation in which a runnable process is overlooked in definitely by process scheduler although its capable of proceeding its never chosen.

# Banker’s Algorithm

The banker’s algorithm is a resource allocation and deadlock avoidance algorithm that tests for safety by simulating the allocation for predetermined maximum possible amounts of all resources, then makes an “s-state” check to test for possible activities, before deciding whether allocation should be allowed to continue.

# Producer - Consumer Problem:

Ensure that the producer can’t add data into full buffer and consumer can’t remove data from an empty buffer the producer consumer problem is a synchronization problem.

# Semaphores:

Semaphore is simply a variable that is non-negative and shared between threads. A semaphore is a signalling 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. A semaphore S is an integer variable that can be accessed only through two standard operations

Wait() or down() : decrements the semaphore value by one

Signal() or up() : increments the semaphore value by one

# The Critical Section Problem

Critical Section is the part of a program which tries to access shared resources. That resource may be any resource in a computer like a memory location, Data structure, CPU or any IO device.

# Page Replacement Algorithm:

Page replacement algorithms are the techniques using which an Operating System decides which memory pages to swap out, write to disk when a page of memory needs to be allocated.

## Types of page replacement Algos

* First In First Out (FIFO)
* Optimal
* Least Recently used (LRU)
* Second Chance (SC)
* Not recently used (NRU)

# Reader Writers Problem

There is a reader a writer reader reads writer writes all 4 type of scenario is possible

Reader is reading n writer writes RW

Writer writes then readers read WR

2 writers write WW issue of overwriting

2 readers read RR is the best cause 2 no one writes so can read simultaneously it’s the smooth operation

# TLB

In TLB its mainly the CPU access the pages present in the main memory required by the CPU, if page is not present in main memory then Page Fault occurs.

# The advantages of using TLB are-

* TLB reduces the effective access time.
* Only one memory access is required when TLB hit occurs.

# A major disadvantage of using TLB is-

* After some time of running the process, when TLB hits increases and process starts to run smoothly, a context switching occurs.
* The entire content of the TLB is flushed.
* Then, TLB is again updated with the currently running process.

# Concurrency

Execution of multiple instruction at same time.

# Process Thread Diff

|  |  |  |
| --- | --- | --- |
| Comparison Basis | Process | Thread |
| Definition | A process is a program under execution i.e an active program. | A thread is a lightweight process that can be managed independently by a scheduler. |
| Context switching time | Processes require more time for context switching as they are heavier. | Threads require less time for context switching as they are lighter than processes. |
| Memory Sharing | Processes are totally independent and don’t share memory. | A thread may share some memory with its peer threads. |
| Dependency | Individual processes are independent of each other. | Threads are parts of a process and so are dependent. |