Operating System

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

Processor Management

- **Error detecting aids**

- Question of security and integrity of user programs and data.
- Problem of data communication.

A process includes the complete execution context (code to execute, data to manipulate, registers, OS resources in use). Following are the major activities of an operating system with respect to program management.

- Loads a program into memory.
- Executes the program.
- Handles program's execution.
- Provides a mechanism for process synchronization.
- Provides a mechanism for process communication.
- Provides a mechanism for deadlock handling.

I/O Operation

I/O subsystem comprised of I/O devices and their corresponding driver software. Drivers hides the peculiarities of specific hardware devices from the user as the device driver knows the peculiarities of the specific device.

4. Operating System Properties

₹

5. Process

A process is a program in execution. The execution of a process must progress in a sequential fashion. Definition of process is following.

∉

Process Control Block, PCB

Each process is represented in the operating system by a process control block (PCB) also called a task control

Queues are of two types

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

Long Term Scheduler

7. Scheduling Algorithms

- Processer should know in advance how much time process will take.

P2	8 - 2 = 6
P3	16 - 3 = 13

Average Wait Time: (0+2+6+13) / 4 = 5.25

Round Robin Scheduling

- Each process is provided a fix time to execute called quantum.
- Once a process is executed for given time period. Process is preempted and other process executes for given time period.
- Context switching is used to save states of preempted processes.

Wait time of each process is following

Process	**Wait Time : Service Time - Arrival Time**

8. MULTI-THREADING

What is Thread?

A thread is a flow of execution through the process code, with its own program counter, system registers and stack. A

ADVANTAGES

- Thread switching does not require Kernel mode privileges.
- User level thread can run on any operating system.
- Scheduling can be application specific in the user level thread.
- User level threads are fast to create and manage.

DISADVANTAGES

- In a typical operating system, most system calls are blocking.
- Multithreaded application cannot take advantage of multiprocessing.

Kernel Level Threads

In this case, thread

pãPc@Î£rî5<í4#-9VT1piV4XDp^2rïe•d#qB6T2”bt¬éF'N-®î>Î®-ªä®î-áãtvr2F„•Žð ±@9 @9 " D-ªî %a•d˜˜$ï ñqù– ™`•A…d x

- Transfer of control from one thread to another within same process requires a mode switch to the Kernel.

Multithreading Models

One to One Model

There is one to one relationship of user level thread to the kernel level thread.This model provides more concurrency

9. MEMORY MANAGEMENT

Memory management is the functionality of an operating system which handles or manages primary memory. Memory management keeps track of each and every memory location either it is allocated to some process or it is free. It checks how much memory is to be allocated to processes. It decides which process will get memory at what time. It tracks whenever some memory gets freed or unallocated and correspondingly it updates the status.

Memory management provides protection by using two registers, a base register and a limit register. The base register holds the smallest legal physical memory address and the limit register specifies the size of the range. For example, if the base register holds 300000 and the limit register is 1209000, then the program can legally access all addresses from 300000 through 411999.

Instructions and data to memory addresses can be done in following ways

- **Compile time** -- When it is known at compile time where the process will reside, compile time binding is used to generate the absolute code.
- **Load time** -- When it is not known at compile time where the process will reside in memory, then the compiler

- **Low Memory** -- Operating system resides in this memory.
- **High Memory** -- User processes then held in high memory.

¢ **Segment number (s)** -- segment number is used as an index into a segment table which contains base address of

Virtual memory is commonly implemented by demand paging. It can also be implemented in a segmentation system. Demand segmentation can also be used to provide virtual memory.

Demand Paging

A demand paging system is quite similar to a paging system with swapping. When we want to execute a process, we swap it into memory. Rather than swapping the entire process into memory, however, we use a lazy swapper called pager.

When a process is to be swapped in, the pager guesses which pages will be used before the process is swapped out again. Instead of swapping in a whole process, the pager brings only those necessary pages into memory. Thus, it

Step	Description

Least Recently Used (LRU) algorithm

11. LINUX

Linux System Architecture is consists of following layers

- **Hardware layer** - Hardware consists of all peripheral devices (RAM/ HDD/ CPU etc).
- **Kernel**

- These are the files that contain user information.
- These may have text, databases or executable program.