Notes

Operating System

Chapter1 : Introduction

A. OS Structure
   a. Simple Structure
   b. Layered Approach
   c. Micro Kernels
   d. Modules
B. Virtual Machines: Creation of no. of different identical execution environments on single computer each of which emulates the host computer.
   Benefits: Protection, Easy Development, Sharing H/W , Easy Communication, Consolidation, Portability, Usefulness.
   Dis Adv**.:** Processing tome is slower, H/W support needed

Chapter 2 : Process Management

- Process is Program in execution .
- Process includes Text section, Stack , Data and Heap
- Process undergoes various states New, Ready, Run, Wait, Halt.
- Process Control Bloc(PCB) is I-Card of process . It is a Data Stuct. This stores all the info of process.
- Diff. Types of schedulers
  a. Long term or Job scheduler
  b. Middle term scheduler
  c. Short term or Process Scheduler
- Switching the CPU from one process to another process saving the state of old process and loading new process. This task is known as Context Switch.
- Process Creation and termination: Process is created by call of Creat process system call every process is identified by unique integer value called PID. After completion process is terminated by EXIT system call
- Inter Process Communication – Shared memory model, Message Passing Model

Chapter 3: Multithreaded Programming

- A thread is a flow of execution through the process code which is also called as lighr weight process. It is used to utilize CPU more effectively

- Thread contains Thread Id, Saved REG, Stack ptr, instruction ptr, signal mask, priority
- Benefits of Thread are Responsiveness, Resource Sharing, Economy, Scalability
- Kernel Level Threads : Thread mgmt. done by Kernel
- User Level Threads: Application manages thread mgmt..
- Multithreading Models: Many-to-Many(Many User Threads to Many Kernel Threads) Many-to-One(Many User Threads to One Kernel Thread) One-to-One(One User thread to one Kernel Thread)

Chapter 4: Process Scheduling

- Scheduling Schemes:-
  - NON-Preemptive Scheduling: Once the process submitted to CPU though higher priority process enters into ready queue currently executing process doesn't stop it's execution
  - Preemptive Scheduling: If Higher Priority Process enters into the ready queue currently executing process stop it's execution and Higher Priority Process starts it's execution.
- Dispatcher is the module which actually gives control of the CPU to the process selected by the short term scheduler.
- Scheduling Criteria:
  - Arrival Time: Time at which process enters into Ready Queue
  - Burst Time: Time required by the process to compete it's execution
  - Start Time: Time at which process starts its execution
  - Finish Time : Time at which process completes it's execution
  - Turn Around time: Total time taken by the process to complete it's execution(Finish Time – Arrival Time)
  - Wait Time: Time for which process resides into ready queue.(Start Time – Arrival Time)
- Scheduling Algorithms:
  - FCFS: -Only Non preemptive scheduling
    - Scheduling criteria Minimum AT.
    - If AT is same then consider Process ID.
  - SJF- Both Preemptive and Non Preemptive scheduling
    - Scheduling Criteria Min BT
    - First process scheduled by default process with min AT. Then next process with min BT.
    - If BT of two or more process are same then consider min AT

        - For preemptive scheduling first process scheduled is the process with Min AT then check the next process arriving into ready queue. If the BT of newly arrived process is less than the remaining BT of currently executing process then preempt the currently executing process.
    - Priority- Both Preemptive and Non Preemptive scheduling
        - Scheduling Criteria High Priority Value
        - First process scheduled by default process with min AT. Then next process highest priority.
        - If Priority of two or more process are same then consider min AT
        - For preemptive scheduling first process scheduled is the process with Min AT then check the next process arriving into ready queue. If the Priority of newly arrived process is higher than the priority of currently executing process then preempt the currently executing process.
    - Round Robin:
        - Only Preemptive scheduling.
        - Scheduling criteria Min AT.
        - Also called as preemptive FIFO
        - Mostly used in Time Sharing Systems.
        - Each process has assigned a unique Time Slice when this time slice is over currently executing process stops it's execution and next process starts it's execution.
    - Multilevel Queue: Process are distributed in different queues depending upon their type. Once the process enters into in one queue can't shifted to another queue. Scheduling of queue done as per the preemptive priority.
    - Multilevel Feedback Queue:
- Thread Scheduling: Preemptive Priority scheduling is used.

Chapter 5: Process Synchronization

- Race Condition- A situation where multiple process access and manipulate the same data concurrently.
- Critical Section-In set of co-operating process each process has shared data or code segment it is called critical section
- Critical Section Problem- It is a protocol that process can use to cooperate. Each process must request permission to enter it's critical

section. Solution to this problem must satisfy Mutual Exclusion, Progress and Bounded waiting. It is implemented by Peterson's solution and Bakery algorithm for more than two processes

- Semaphore- It is a integer variable used for process synchronization. It works two operations Signal() i.e. V and Wait()i.e P
- Problems of Synchronization
    - Bounded –Buffer Problem(Producer and Consumer Problem)
    - Reader-Writer Problem
    - Dining Philosopher Problem.

Chapter 6: Deadlock

- A set of process is in deadlock state when every process in the set is waiting for an event hold by one another.
- Deadlock Characterization-
    - Mutual Exclusion
    - Hold and Wait
    - NO Preemption
    - Circular Wait
- Resource Allocation Graph – It is a directed graph shows process and resource relation to identify that system will suffer from deadlock or not. Process shown in oval and resources in rectangle. Allocation edge- Edge from Resource to Process. Request Edge – Edge from Process to Resource.
- Deadlock Prevention- Avoid the deadlock characterization points.
- Deadlock Avoidance- Implemented by Bankers Algorithm or Resource Request algorithm. It finds the safe sequence to avoid the system from deadlock state.
- Deadlock Recovery-
    - Process Termination-
        - Terminate the process one by one
        - Terminate the process one by one until system get recovered from deadlock.
        - Terminate all the process from Deadlock state.

    - Resource Termination
        - Select Victim
        - Roll Back
        - Starvation- Avoided by Lock

Chapter 7: Memory Management

- The address generated by CPU is known as Logical Address or Virtual Address.
- The address seen by the memory unit that is loaded into the memory-address-register is referred as Physical address.
- Memory Management Techniques
    - Contiguous Memory Management.
        - Single Contiguous- Only one partition is created. Used in Uni-Programming OS
        - Multiple Contiguous
            - MFT(Multiprogramming with Fixed No of Tasks)- Memory Partitions are creates statically. No. of partitions are fixed separated by a H/W lock. As no. of partitions are fixed degree of multiprogramming is also fixed. Suffer from Internal and External Fragmentation
            - MVT (Multiprogramming with Variable No. Tasks)- Memory Partitions are created as per the size of the process. Partitions are created dynamically. As no. of partitions are not fixed degree of Multiprogramming is not fixed.
    - Non Contiguous Memory Management- Implemented by Paging, Segmentation, Virtual Memory.
- Paging- Physical memory braked in to fix sized partitions.
- Page Fault- If required page is not available in memory called page fault.
- Page Replacement Algorithms
    - FIFO- The page which is inserted first will be replaced first.
    - LRU(Least Recently Used)- The page which is not used for longer time will be replaced first.
    - Optimal Replacement- The page which is not required in future reference or required at last in future reference will be removed first.
- Segmentation-It is a memory management scheme that supports user's view of memory. A program is a collection of segments.

Chapter 8: File Management

- A File is a named collection of related information that is recorded on a secondary storage.

- A File is a sequence of bits, bytes, lines or records the meaning of which defined by the file's creator and user.
- File Systems- Tape Based, Disk Based.
- File Attributes- Name, Identifier, Type, Location, Size, Protection, Time Date.
- File Operations- Creating File, Writing File, Reading File, Repositioning, Deleting, Truncating.
- File Types- File type is identified by it's extension. File types are Executable, Object, Source Code, Batch, Text, Library, Archive and Multimedia.
- File Access Methods: Sequential Access, Direct Access, Indexed Access.
- Directory Structure-
    - Single Level
    - Two level directory
    - Tree Structured
    - Acyclic Graph Directories
    - General Graph Directories
- File Allocation Methods
    - Contiguous Allocation- Contains Array of Disk Blocks, Accessed sequential and Direct, Suffers from internal and external fragmentation
    - Linked Allocation- Uses Linked List of disk blocks that are scattered throughout the disk, Only sequential Access, No external fragmentation.
    - Indexed Allocation- All pointers to blocks are placed together in one block known as Index Block, No external fragmentation
- Free Space Management is implemented by Bit Map/Bit Vector , Linked List, Grouping, Counting, Space Map