# Ch: 1

Harsh Porwal

(1) Explain different types of service provided by operating System.

-> Operating system provides different types of service for different. users.
-> It provides service to program like load of data into memory, allocating disk for storage, Open and read files etc.
-> Service will change according to the operating system.
-> Two different types of OS provides same types of service with different names.
-> OS make programmer job easy to provide different types of services.
-> There are some list of service provided by OS:

(a) Program: execution
(b) Input - Output operation
(c) Error Detection
(d) File and directory operation
(e) Communication
(f) Graphical user Intearface.

(a) Program execution :

Before executing the program it is loaded into memory by OS.

- Once program loads into memory its start execution.
- Program finishis its execution with or without the ~~error~~ error.

(b) Input - Output Operation :

Program is a combination of input and output statment.

- When executing the program it requires I/o devices.
- OS provides the I/o devices to the program.

(c) Error - Detection :

Error is related to the memory, CPU, I/o device and in the user program.

- Memory is full, stack overflow, stack underflow, file not found, directory is not exist, printer is not ready are the example of error-detection.

(d) File and Directory Operation :

User wants to read and writes the file and directory.

- User wants to serach the file, rename and modify the file etc.
- User wants to create the file.
- All these operation perfomed by user with help of OS.

(e) Communication :

Communication may be inter-process communication and other type of communication.

- Data transfer is one type of communication.

Harsh Porwal

- Communication is in between two process of same machine or different machine.
- Pipe, shared memory, message passing are different types of communication.

(f) Graphical user intearface :

User intearacts with OS using user interface. All OS provides user interface.

- There are two types of interface.
  (i) Command line (ii) Batch intearface.
- In command line interface user enter text command for performing operation.
- Batch interface uses files. A files contains the command of various operation.
- When file executes command output is displayed on screen.

→ All of the above service is provided by OS for single user.

→ If there are multiple user then OS provides some additional services like allocation, accounting, protection of data and security.

→ Multiple user requires different resources for their execution. One type of resource may be required two type of user. So resouce allocation necessary.

→ Accounting means keeping information of resources and user. Which type of resources are allocated to which user.
It checks wheather user has permission for using this type of resources.

(2) Explain Time - sharing.

→ Processor's time which is shared among multiple users at same time is termed as time-sharing.

→ In interactive system many users directly interact with the computer from terminals connected to the computer.

→ It is logical extension of multiprogramming Operating System.

→ It is a method that allows multiple users to share resources at the same time.

→ Multiple user in various locations can use a specific computer system at a time.

→ In this time-sharing OS. many processes are allocated with computer resources in respective time slot.

→ In this processor's time is shared with multiple users that's why it is called time-sharing OS.

Harsh Porwal

-> It has a fixed time slice for different processes. It's main purpose is interactive response time.

-> The CPU executes multiple Jobs by switching between them, but the switches occurs so frequently. Thus user can recive an immediate response.

-> The main difference between multiprogrammed Batch system and Time-sharing system is that in multiprogrammed batch system is used to maximize processor use. And Time-sharing system is used to minimize response time.

* **Features:**

- Each user have dedicated time for all operation.
- Multiple online user can use same computer at same time.
- Better interaction between users and computers.
- User requests can make in small time response.
- It make quick processing with a lot of task.

* **Advantage:**

- It provides quick response.
- Reduce CPU idle time.
- Less probability of duplication of software.
- Easy to use and user friendly.
- All the task are given a specific time.

* **Disadvantages :**

- It consume many resources.
- Requires high specification of hardware.
- Problem of reliability.
- An issue with security and integrity of data.
- Probability of data communication problem.

(3) **Difference between Time-Sharing and multiprogramming OS.**

→

| Time - Sharing OS | Multiprogramming OS |
|---|---|
| It enables execution of multiple task and process at same process to increase CPU perfomance. | Multiple programs store in main memory at same time to improve CPU Utilization. |
| It is based on concept of time-sharing. | It is based on the concept of context switching. |
| It allows multiple process to run at same time via time-sharing. | It reduce CPU idle time for as long as possible. |
| Takes less time to execute the task. | Takes more time to execute the process. |
| Principle objective is minimize response time. | Principle objective is maximize processor use. |
| It has a fixed time slice. | It has not fixed time slice. |
| Ex. Mac OS, Windows OS. | Ex. Linux, Unix |

classmate
Date
Page
Harsh Porwal

(4) What is Operating System? Explain OS as Resource manager. and as a User Interface.

-> Operating system is a program that controls the execution of application programs. It is an interface between application and hardware.

* OS as a resource manager:

-> A computer is a set of resources. These resources provides various functions to the user like storing of data and program.

-> The OS is responsible for managing all the resources.

-> A portion of OS is in main memory This portion of OS is called Kernel.

-> User program and data is also stored in remaining part of memory.

-> Allocation of main memory is controlled by OS with help of memory managment hardware.

-> I/O devices is controlled by OS and it decides when an I/O device can be use by program.

-> Processor is one type of resource and OS controll the execution of user program on the processor

→ Modern OS allows multiple programs to run at same time.

→ If multiple user are using computer so there is need of managig and protecting the memory, I/o device and other resources.

→ There are two concept of resource sharing. (i) Time (ii) Space

→ Time : Time slot allocated to each program first one gets to use the resources then another and so on.

→ Space : Consider the example of main memory. It is divided among serval running programs, so each one can be resident at the same time.

Harsh Porwal

## * OS as a user interface :

-> Computer System consist of software and hardware to solve specific problems.

-> User, Application programs, computer system are components of computer system.

-> Application program used to solve specific program. Student attendance monitoring is the example of application program.

-> OS is a subset of system software. OS directly interacts directly with the hardware to provide an interface to other system software.

-> Resource sharing and Resource abstraction are two key aspect of OS.

-> Purpose :

- OS provides an interface between the computer hardware and user. It simplifies programmer Job like editing, coding, Creation.

- Allocation and use of computer resources among the programmer is controlled by OS.

Harsh Porwal

(6) Explain System calls.

-> A System call is a method for computer program to request a service from the kernel of the OS on which it is running.

-> A system call is a method of interacting with OS via programs.

-> A System call is a request from computer software to an OS kernel.

-> Application Program Interface (API) connects the operating system's function to user programs.

-> It act as a link between OS and process, allowing user level program to request OS services.

-> Kernel system can be access using system call.

-> System calls required for any programs that use resources.

-> The system call uses an API to expose the OS services to user programs. It is the only method to access the kernel system.

-> System call has greater authority than a standard subroutine.

-> System calls are not permited to use shared libraries or any symbols that are not present in the kernel.

-> The code and data for system calls are stored in global kernel memory.

-> The application run in an area of memory known as user space.

-> A system call connects to the OS kernel, which executes in kernel space.

-> When an application create a system call. it must first obtain permission from the kernel.

-> There are five types of system calls:
(a) Process Control
(b) File Managment
(c) Device Managment
(d) Information maintenance
(e) Communication.

(a) Process control:

It is used to direct the processes. Ex. Creating, Load, abort, end, execute, process, terminate etc.

classmate
Date
Page
Harsh Porwal

(b) File Managment :

It is a System call that is used to ~~main~~ handle the files.
User can creat file using create ( ) system call.
Every file has file attribute. File attribute includes name of file, type of file etc.
Ex. Creating, deleting, open, close, read, write, etc.

(c) Device Managment :

It is a system call that is used to deal with devices.
When process request for resources, request is granted if it is free otherwise it is rejected.
Ex. read, device, write, get device attribute etc.

(d) Information Maintanance :

It is a system call that is used to maintain information.
System call for this category is get time and date, set time and date, get system data.

(e) Communications :

It is a system call that is used for communication.

- Shared memory : A process using memory for communication. One process will creat a memory portion which other processes can access.
- Two process communicate with each other by passing message. It is direct and indirect communication.
Ex. Create, delete, send, recive message etc.

(6) Explain Distributed Operating System.

→ A DOS is an essential type of Operating system.

→ Distributed systems use many central processors to serve multiple real-time application and user.

→ As a result data processing jobs are distributed between the processor.

→ It connects multiple computers via a single communication channel.

→ Each of these systems has its own processor and memory.

→ Individual systems that communicate via single ~~channel~~ channel are called as single entity.

→ A distributed computer system consist of multiple software component that are on multiple computers but run as a single system.

→ Distributed OS owns the whole network and makes it look like a virtual uni-processor or may be virtual multiprocessor.

Harsh Porwal

-> A LAN is a network that is confind to a relatively small area. It is genrally limited to a geographic area such as college.

-> WAN provides long distance transmission of data and voice. Computers connected to wide area network are often connected through public network such as telephone system.

-> A MAN often acts as a high-speed network to allow sharing of regional resources. MAN provides transfer speed from 34 to 150 Mbps.
MAN typically cover an area of between 5 and 50 km diameter. It may also cover an area the size of city.

* **Advantage:**

-> Sharing of software resources such as sofware liabraries, database and hardware resources such as hard disk, printer.
-> It is higher reliability. Reliability refers to the degree of tolerance against error. Avalaibility is one of important aspect of reliability.
-> Reduction in price of microprocessor and increasing computing power good price-perfomance ratio.
-> To exted power and functionality of system by adding additional resources to the system.
-> Shorter response time higher throughput.

* Disadvantage:

-> The database connected to the DOS is relatively complicated and hard to manage.

-> There are no current commercialy succesfull examples.

-> Protocol overhead can dominate computation cost.

-> Hard to build well.

-> Probably impossible to build at the scale of internet.

Harsh Porwal

(7) Explain Real Time OS.

-> Time constraint is a key parameter is real time systems. It controls system such as robots, satelites, air traffic control.

-> When user give input to system, it must process within the time limit and result is sent back. Real time systems fails if it does not give result whthin time limits.

-> There are two types of Real time system
(1) Hard Real time
(2) Soft Real time

-> Critical task is completed within the time limit in hard real time system.

-> Gernal purpose operating system does not support the hard real time system function. Real time task can't keep waiting for longer time without allocatting kernel.

-> Soft Real time is less restrictive real time. It can not guarantee that it will be able to meet deadline under all condition.

-> Real time OS uses priority scheduling algorithm to meet the response requirment of real time application.

classmate

Date
Page

→ Applications :

- Transportation : Air traffic control and Traffic light system
- Communication : Digital telephone
- Process control : petroleum and paper mill
- Detection : Radar System
- Flight Simulation : Auto piolet shuttle mission simulator.

→ Memory managment in Real time OS is less demanding than other type of multiprograming OS.

→ Time critical device managment is one of the important characterstic in this OS.

classmate
Date
Page
Harsh Porwal

(8) Explain Kernel.

→ The kernel is a software code that remain in the central core of OS. It has complet control over the system.

→ A kernel is different than the shell. The shell is outer most part of OS and program that interacts with user commands.

| |
|---|
| User Interface |
| Kernel |
| Computer hardware |

→ The kernel does not directly interact with the user. But it interacts with shell, other program and hardware

→ When OS boots, kernel is the first part of OS to load into memory. Kernel remains in the memory for entire duration of computer system session. The kernel code is loaded into a product area of memory.

→ Kernel perform its task in kernel space. Executing processes performed by kernel in kernel space. User perform its task in user area memory. Memory is divided into system area are user area.

-> When a computer is crashing its accutally mean Kernel is crashed.

-> Single program crash is not kernel crash.

-> The kernel provides service for process managment, memory managmant, I/O device managment.
System calls are used to providing this type of service.

-> Kernel content change according to OS. but typically it includes,
Scheduler, Supervisor, Interrupt handler, Memory manager.

-> Scheduler allocates Kernel's processing time.
-> Supervisior grants permission to use computer system resource to each process.
-> Interrupt handler handles all request from various hardware devices.
-> Memory manager allocates space in memory for all users of the kernel service.

-> There are 4 types of kernel
(1) Monolithic kernel
(2) Micro kernel
(3) Hybrid kernel
(4) Exo - kernel.

Harsh Porwal

(9)

| Monolithic Kernel | Micro Kernel |
|---|---|
| -> Kernel size is large. | Kernel size is small. |
| -> OS is complex to design. | OS is easy to design. |
| -> Request may be serviced faster. | Request may be serviced slower. |
| -> All OS service includes in kernel. | Kernel provides IPC and low level device mangment. |
| -> No message parsing and context switching required | It requires message parsing and context switching |

(10)

| Client - Server model | Peer to peer model |
|---|---|
| -> It firmly divided the role of client and server. | It doesn't have such strict roles. All nodes in system are considered peers. |
| -> Single point of failure | No single point of failure. |
| -> Need for dedicated application and database. | No Need dedicated application and database. |
| -> Resource may be remove at any time. | Resource are not removed from network until they requested |
| -> Storage and bandwith must provided by host. | Storage and bandwith are distributed. |
| -> Provides good security. | Provides poor security. |

classmate
Date
Page
Harsh Porwal

# Ch-2

## Process Managment

(1) Explain Process.

-> Process is an asynchronous activity.
-> Process is an active entity that requires a set of resources including processor, program counter, register.
Multiple process may be associated with one program.
-> Process means program execution. Process execution must progress in sequential order.

-> Each process have own address space.
Address space is divided into three region,
(i) Text, (ii) Data (iii) Stack

(i) Text region : It stores the code that the processor executes.

(ii) Data region : It store variables and dynamical allocated memory that the process uses during execution.

(iii) Stack region : It stores instructions and local variable for active procedure calls.

-> A process run to completion only when all requested hardware and software resources have been allocated to the process.

-> Program become process when executable file loaded into memory. A process may be independent of other process in the system.

-> After booting OS it creates foreground process and background process. foreground process interact with user and and background processes is used by system.

-> An Initialized data region contains the static and global variable that are Initialized by the user.

-> Un-initialized data region contains variable that are initialized by kernel to zero.

-> Heap is used for dynamic memory allocation and stack contains program counter.

Harsh Porwal

(2) Explain Process control block.

→ OS keeps an internal data structure to describe each process it manages. When OS creats a process, it creates process descriptor. In some OS it calls process control block (PCB).

→ PCB will change according to the OS. PCB is also called task control block.

→ PCB is identify by Integer process ID (PID). When the OS stops running a process it saves the register's value in the PCB.

→ When process is created by OS, it allocates a PCB for it.

(i) Process identification: Each process is uniquely identified by user's identification and pointer connecting it to its descriptor.

(ii) Priority Number: OS allocates priority number to each process. According to priority number it allocates resources.

(iii) Program Counter: The PC indicates address of next instruction to be executed for this current process.

(iv) Memory allocation : It contains the value of the base registers, limite registers, and the page table depending on the memor system used by the OS.

(v) I/o status Information : It maintains information about the open files, list of I/o devices allocated to the process.

(vi) List of open file : Process uses number of files for operation. OS keep track of all opened file by process.

(vii) Process state : Process may be in any one of the state : new, ready, running and waiting, terminate.

→ when a process changes state the OS must update information in the procesels PCB.

| |
|---|
| Process Identification |
| Priority number |
| Program counter |
| Memory allocation |
| I/o status information |
| List of open file |
| process state |

Harsh Porwal

(3) Explain Process State.

-> Each process has been an execution state which indicates what process is currently doing.

-> The process descriptor is the basic data structure that represent the specific state for each process.

-> A state diagram is composed of a set of states and transition between states.

-> State diagram is used by process manager to determine type of service provided to process. The process states are as follow, (1) New (2) Ready (3) Running (4) Blocked / waiting (5) Exit / End.

(1) New : OS creates new process by using fork() system call. These process are ~~me~~ newley created process and resources are not allocated.

(2) Ready : The process is competing for the cpu. Process reaches to the head of list.

(3) Running : The process that is currently being executed. OS allocates all the h/w and s/w resources to the process for execution.

(4) Block/waiting : A process is waiting untill some event occure such as the completion of an input-output operation.

(5) Exit/End : A process completes its operation and release all resources.

-> OS maintains a ready list of ready process and blocked list of blocked process. Redy list maintain in priority order and blocked list maintain unorder.

-> The act of assigning a processor to the first on redy state called dispatching and is performed by system entity called dispatcher.

-> The process may request a resource when it is in the running state. In most of the OS if system in running state request an immediatley avalaible resources the process is allowed to continue in running state.

Harsh Porwal

(4) What is thread? Explain thread cycle.

-> Threads is a dispatchable unit of work.

-> It consists of thread ID, Program counter, stack and register set. Thread is also called a Light weight process.
A thread is easy to create and destroye.

-> Threads schedule on processor. Each thread can execute a set of instructions independent of other thread and process.

-> Every program has at least one thread. programs without multithreding executes sequentially.

-> More than one thread can executeing code inside process.
-> When user double click on icon by using mouse, the OS creates a process and that process has one thread that runs icon's code.

-> Each thread belong to exactly one process.
-> When process starts up a remote TCP connection, its thread handles the low-level details of ~~network~~ sending out network packets.

-> Different OS users threads in different ways.

(1) Free memory is managed by kernel thread in LINUX OS.

(2) Solaries uses a kernel thread for interrupt handling.

-> Advantage : context switching time is minimized.

- Thread support for efficient communication.
- Resource sharing possible using thread.
- thread provide concurrency within a process.
- It is more economical to create thread.

* Thread Lifecycle

classmate
Date
Page
Harsh Porwal

→ New : Thread is just created.

→ Ready : Thread's start() method invoked and now it is executing. OS put thread into Ready queue.

→ Running : Highest priority ready thread enters running state. Thread is assign a processor and now is running.

→ Blocked : This is the state when a thread is waiting for a lock to access an object.

→ Waiting : Here thread is waiting idenfinitely for another thread to perform action.

→ Sleeping : Thread sleep for a specified time of period. when sleeping time expires it enters to ready state. CPU is not used by sleeping thread.

→ Dead : When thread completes task and operation.

(5) Difference between User level and kernel level thread.

| User Level thread | Kernel Level thread |
|---|---|
| → These are faster to create and manage. | These are slower to create and manage. |
| → Implemented by thread library | Support directly by operating system. |
| → It can run on any operating system. | These are specific to operating system. |
| → Support provided by user level called | Support provided by Kernel level called |
| → Multithread application can't take advantage of multiprocessing. | Kernel routines themeselves can be multithread. |
| → Ex: POSIX Pthreads and Mach C-thread. | Ex. Windows 95/98/NT, Sun Solaris and Digital Unix. |
| → These are also called one mapping thread. | It supports one to one thread mapping |

classmate
Date
Page
Harsh Porwal

# (6) Sheduling algorithm.

## → * First come first serve scheduling (FCFS)

→ FIFO is also called first come first severe scheduling method. The processes are dispatched according to their arrival time at the ready queue. It is non-preemptive CPU scheduling algorithm.

→ FIFO is simple to implement because it uses a fifo queue. This algorithm is fine for most of the batch OS.

→ When new process enters to the system, its PCB is linked to the end of the ready queue and it is removed from front queue.

→ Turnaround time is unpredictable with FIFO algorithm. Average waiting time of the FCFS is often quite long.

Ex. Consider following set of process that arrive at time 0 with the length of CPU brust given in ms. calculate average waiting time when process arrives in the order : $P_1, P_2, P_3, P_4, P_5$

| Process | Burst time |
|---|---|
| $P_1$ | 4 |
| $P_2$ | 7 |
| $P_3$ | 3 |
| $P_4$ | 3 |
| $P_5$ | 5 |

→ Giant chart :

| $P_1$ | $P_2$ | $P_3$ | $P_4$ | $P_5$ |
|---|---|---|---|---|

0 4 11 14 17 22

→ Waiting time :

| Process | waiting time |
|---|---|
| $P_1$ | 0 |
| $P_2$ | 4 |
| $P_3$ | 11 |
| $P_4$ | 14 |
| $P_5$ | 17 |

Average waiting time = $\frac{0+4+11+14+17}{5} = 9.2$

→ Turn around time : waiting time + Brust time

| Process | Turnaround time |
|---|---|
| $P_1$ | $0+4=4$ |
| $P_2$ | $4+7=11$ |
| $P_3$ | $11+3=14$ |
| $P_4$ | $14+3=17$ |
| $P_5$ | $17+5=22$ |

Average turnaround time = $\frac{4+11+14+17+22}{5}$

$= 13.60$

classmate
Date
Page
Harsh Porwal
Harsh Porwal

# * Shortest Job first scheduling (SJFS)

→ SJFS algorithm is also known as shortest Job Next. Scheduling algorithm. It handles the process based on length of their CPU cycle time. It reduces waiting time over FIFO algorithm.

→ SJF is non-preemptive CPU scheduling algorithm.

→ It does not work in interactive system because user do not estimate in advance the CPU time required to run their process.

→ SJF algorithm is used frequently in long term scheduling.

Ex.

| Process | Burst time |
|---|---|
| $P_1$ | 5 |
| $P_2$ | 2 |
| $P_3$ | 6 |
| $P_4$ | 4 |

→ Giant chart :

| $P_2$ | $P_4$ | $P_1$ | $P_3$ |
|---|---|---|---|

0 2 6 11 17

→ Waiting time :

| Process | waiting time |
|---|---|
| $P_1$ | 6 |
| $P_2$ | 0 |
| $P_3$ | 11 |
| $P_4$ | 2 |

Average waiting time = $\frac{6+0+11+2}{4} = 4.75$

→ Turnaround time : Waiting time + Brust time.

| Process | Turnaround time |
|---|---|
| $P_1$ | 11 |
| $P_2$ | 2 |
| $P_3$ | 17 |
| $P_4$ | 6 |

Average turn around time = $\frac{11+2+17+6}{4}$

$= 9$

Harsh Porwal

## * Priority Scheduling

→ It is preemptive and non-preemptive algorithm. It is one of the most common scheduling algorithm in batch system. Priority can be assign by system admin using characterstic of the process.

→ In this algorithm CPU selected higher priority process first. If priority of two process is same then FCFS algorithm solve this problem.

→ Problem : Waiting time is more for lower priority process even if there required CPU burst time is less. Priority Scheduling algorithm faces starvation problem. It can be solved by using aging technique.

→ In agging the priority of the process will increase which waiting for long time in the ready queue.

ex.

| Process | Burst time | Priority |
|---|---|---|
| $P_1$ | 5 | 3 |
| $P_2$ | 2 | 4 |
| $P_3$ | 6 | 1 |
| $P_4$ | 4 | 2 |

→ Giant chart :

-> waiting time :

| Process | waiting time |
|---|---|
| $P_1$ | 10 |
| $P_2$ | 15 |
| $P_3$ | 0 |
| $P_4$ | 6 |

Average waiting time $= \frac{10 + 15 + 0 + 6}{4}$

$= 7.75$

-> Turnaround time : waiting time + Brust time

| Process | Turnaround time |
|---|---|
| $P_1$ | $10+5 = 15$ |
| $P_2$ | $15+2 = 17$ |
| $P_3$ | $0+6 = 6$ |
| $P_4$ | $6+4 = 10$ |

Average turnaround time $= \frac{15 + 17 + 6 + 10}{4}$

$= 12$

classmate
Date
Page

Harsh Porwal

# * Round Robin Scheduling

→ Round robin is a preemptive scheduling algorithm. It is used in interactive system.

→ Here process are given a limited ammount of time of processor called a time slice or time quantum. If a process does not complete before its quantum expires, the system preempts it and gives the processor to the next waiting process.

→ Processes are placed in the ready queue using a FIFO scheme. with the RR algorithm the principle design issue is the length of the time quantum to be used.

Ex.

| Process | Burst time |
|---|---|
| $P_1$ | 21 ~~16~~ 116 |
| $P_2$ | 3 |
| $P_3$ | 6 1 |
| $P_4$ | 2 |

Time quantum = 5

→ Giant chart :

| $P_1$ | $P_2$ | $P_3$ | $P_4$ | $P_1$ | $P_3$ | $P_1$ | $P_1$ | $P_1$ |
|---|---|---|---|---|---|---|---|---|

0 5 8 13 15 20 21 26 31 32

→ Waiting time :

| Process | waiting time |
|---|---|
| $P_1$ | $(0-0)+(15-5)+(21-20)+(26-26)+(31-31)+(32-32)=0$ |
| $P_2$ | 5 |
| $P_3$ | $8+(20-13)=15$ |
| $P_4$ | 13 |

Average waiting time $= \frac{0+5+15+13}{4}$

$= 8.25$

→ Turnaround time :

| Process | Turnaround time |
|---|---|
| $P_1$ | $0+21=21$ |
| $P_2$ | $5+3=8$ |
| $P_3$ | $15+6=21$ |
| $P_4$ | $13+2=15$ |

Average turnaround time $= \frac{21+8+21+15}{4}$

$= 16.25$

Harsh Porwal

(7) Difference between FCFS and RR

| FCFS | RR |
|---|---|
| -> Decision made is non-preemtive | Decision made is preemptive. |
| -> It has minimum overhed. | It has low ovehead. |
| -> Response time may be high. | Provide good response time for short processes. |
| -> It is troublesome for time sharing system. | It is mainly design for time sharing system. |
| -> The workload is simply process in the order of arrival. | It is similar like FCFS but uses time quantum. |
| No starvation in FCFS. | No starvation in RR. |

classmate
Date
Page
Harsh Porwal

# Ch-3 [Concurrency]

(1) Explain principle of concurrency.

-> Concurrent access to shared data may result in data inconsistency. Maintaining data consistency requires mechnisms to ensure the execution of cooperating process.

-> Concurrency arises in the same way at different level of execution stream.
There are some example of concurrency in different OS.

(1) Concurrency in multiprogramming : An interaction between multiple process running on one CPU.

(2) Concurrency in Multithreading : An interaction between multiple thread running in one process.

(3) Concurrecy in Multiprocessor : An interaction between multiple CPUs running multiple processes or threads.

(4) Concurrecy in multi-computers : An interaction between multiple computer running distributed process of threads.

-> Java is a concurrent programming language.

-> Process of synchronization is required in Uni-processor system, multiprocessor system and network.

-> If more than one thread exist in a system at a same time, then the threads are said to be concurrent.

-> Synchronization problem can be occure whenever two or more concurrent process use any shared resources.

-> Concurrency is possible without parallelism.

-> Concurrency means that more than one process or thread running at same time.

Harsh Porwal

(2) Explain Race Condition and Critical Section problem.

-> Race condition occures when two or more operations occure in an undefined manner.

-> When two or more process are reading or writing some shared data and the final result depended on who runs pricisely when are called Race condition.

-> There is a Race condition if output depend on the order of execution. The outcome depends on CPU scheduling or threads.

-> Race condition should be avoided because they can cause fine errors in application and that are difficult to debug.

-> In banking system balance is shared variable. After depositing the ammount, it updated by banek employee balance = balance + ammount.

-> If some ammount is transferd then it becomes balance = balance - ammount. These two tasks are in race to write variable balance. In this type of process that updates last determines the final value of balance.

-> Different execution of the same time program on the same data may not produce same result.

-> Track of various processes is kept by OS.

-> OS allocates and deallocate software and hardware resources to active process.

-> OS must be protect user data from un-authorized process.

-> Process execution speed does not depend upon othe process execution speed.

* <u>Critical Section Problem</u>

-> A critical section is a code of block that only one process at a time can execute.

-> So, when one process is in its critical section, no other process may be in its critical section.

-> The critical system problem ensure that only one process at a time is allowed to be operating in its critical section.

classmate
Date
Page
Harsh Porwal

→ Criticle section means process may change some common variable, writting files, updating memory location, etc.

→ When process is accesing shared modified data it is said to be critical section.

→ Each process takes permission from OS to enter into the criticle section.
There are some structure of Process:
(1) Entry section
(2) Reminder section
(3) Exit section.

→ (1) Entry section : It is a block of code executed in preparation for entering critical situation.

→ (2) Reminder Section : Rest of the code is Reminder section.

→ (3) Exit section : The code executing ~~leaving~~ upon leaving the criticle situation.

→ Each process cycle through reminder, entry, critical, exit section in this order.

Entry Section

→ To prevent critical section problem, the system should ensure that only one process at a time can execute the instruction in its critical section. If one process attemps to enter its criticle section while another is in its own criticle section, the first process should wait untill the executing process exit the criticle section.

→ If a process in critical section terminated by any reason than OS must release mutual exclusion so that the other Process enter their critical section.

| |
|---|
| Entry section |
| Critical section |
| Exit section |
| Reminder section. |

→ OS handles critical section using kernel. kernel is a classified primptive and non-primitive kernel.

→ If process running in kernel mode then it is preempted.

→ It does not allow process to preempt. it is called non-preemptive kernel.

Harsh Porwal

(3) Explain Mutual Exclusion and peterson's Solution.

→ The need of mutual exclusion comes with concurrency. There are serval type of concurrent execution.

(1) Interrupt handle
(2) Interleaved preempitively schedule process.
(3) Multiprocessor clusters with shared memory
(4) Distributed System.

→ It is used in concurrent programming to avoid the simultaneous use of a common resource, such as a global variable.

→ Requirment of mutual exclusion is that when process $P_1$ is accessing a shared resource $R_1$ then on the other process should be able to access resource $R_1$ untill process $P_1$ has finished its operation with resource $R_1$.

→ Files, I/o devices such as printers are example of resources.

→ (1) Software Method : Leave the responsibility with the Processes them selfves.

This method are usually highly error-prone and carry high overheads.

-> (2) Hardware Method : Special purpose machine instructions are used for accessing shared resources.
This method is faster but can not provide complete solution.

-> (3) Programming Language method : Provide support through the operating system or through the programming language.

* Requirments :

-> at any time one process is allowed to enter in its critical section.

-> Solution is implemented purely in software on machine.

-> A process remains inside its critical section for a bounded time.

-> No assumption can be made about relative speed of asynchronous process.

-> A process can not prevent any other process for entering into critical section

-> A process must not be identifinetly postponed from entering its critical situation.

Harsh Porwal

* Peterson's solution:

-> Peterson's solution is software based solution for critical section problem. This algorithm will not work correctly on modern computer architecture.

-> It is simple process algorithm for two process mutual exclusion with busy waiting.

-> This algorithm is less complicated than ~~Dekk~~ Dekker's algorithm.

-> It solves critical section problem of two process only. It does not require any special hardware.

-> Two processes alternates execution between their critical section and reminder section.

-> Deadlock and indefinite are impossible in peterson's algorithm as long as no process or thread terminates unexpectedly.

(4) Explain Test and Set operation of Mutual exclusion. or TSL.

-> Test and set is a single individual machine instruction. It is known as TS.

-> It was introduced by IBM for its Multiprocessing system.

-> In a one machine cycle, it tests to see if the key is avalaible and, if it is, set it to be unavailable.

-> TS are hardware solution of critical section problem.

-> TS instruction eleminates the probability of preemption occurring during interval.

testAndset (P, q)

-> The instruction read the value of q, which may be either true or false. Then the value is copied into p and instruction set the value of q to true.

-> A process $P_1$ would test the condition code using TS instruction before entering a critical section. If no other process in critical Section then process $P_1$ would be allowed to proceed and the condition code would be change from zero to one.

Harsh Porwal

-> When process $P_1$ exist the critical section the condition code is rest to zero so another process can enter into critical section.

-> If process $P_1$ ~~enter the critical section, the~~ finds a busy condition code then its placed in waiting room where it continues to test the condition code and wait untill it is free.

-> The 'Test AND Set' instruction is tool that programmer use to simplify software solution to mutual exclusion but the instruction it self does not enforce mutual exclusion.

* Advantage:

-> Simple to implement.
-> It works well for a small number of process.
-> It can be used to support multiple critical section.

* Drawback:

-> It suffers from starvation.
-> Possibility of busy waiting.
-> There may be deadclock.

(5) Explain Semaphores with its properties.

→ Semaphore is describe by Dijkstra. Semaphore is nonnegative integer variable that is used as a flag.

→ Semaphore is a OS abstract data type. It takes only integer value. It is used to solve critical section problem.

→ Dijkstra introduce two operation (P and V) to operate on semaphore to solve process synchronization problem.

→ A process call the P operation when entering into critical section and process calls V when it exit its critical section.

→ The P operation called as wait operation and V is called as signal operation.

→ A wait operation decrease its value by one.

$$S := S - 1$$

→ A signal operation increase its value by one.

$$S := S + 1$$

→ A proper semaphore implementation requires that P and V be indivisible operation. A semaphore operation is a atomic.

Harsh Porwal

-> This may be possible by taking hardware support. The operation P and V executed by the operating system.

-> There is no guarantee that no two process can execute wait and signal operation on same semaphore at same time.

* Properties :

-> Semaphores are machine independent.
-> Semaphores are simple to implement.
-> Correctness is easy to determine.
-> Semaphore acquire many resources simultaneously.
-> Can have many different criticle sections with different semaphores.

* Drawbacks :

-> They are essentially shared global variables.
-> Access to semaphores can come from anywhere in program.
-> There is no control and guarantee of proper use.
-> They serve two purpose, mutual exclusion and scheduling constraint.

(6) What is Monitor Explain with Structure.

→ Monitor is an object that contains both data and proccedures needed to perform allocation of shared resources.

→ It is programming language construct that support both data access synchronization and control synchronization.

→ Monitor is implement in programming language like Pascal, Java, and C++.

Harsh Porwal

-> Monitor is an abstract data type for which only one process may be executing a procedure at any given time.

-> Monitor is a collection of procedure, variable and data structure.

-> Data inside the monitor may be either global to all routines within the monitor or local to a specific routine.

-> Monitor data is accessible only within the monitor. A shared data structure can be protected by placing it in a monitor.

-> A proccedure defined within a monitor can access only those variable declared locally within monitor.

-> When process calls a monitor procedure, the first few instruction of the procedure will check to see if any other process is currently active within monitor.
If process is active then calling process will be suspended until the othe process left monitor.
If no other process is active calling process may enter.

-> Monitor support synchronization by use of condition variables that are contained within monitor and accessible only within monitor.

Harsh Porwal

-> Condition variable operator has two function.

cwait , csignal.

-> cwait : It suspent execution of the calling process on condition.

-> csignal : Resume execution of some process blocked after a cwait on the same condition.

-> cwait must be come before csignal.

Harsh Porwal

# Ch: 4 Inter - Process Comunication

(1) What is Semaphore and monitor?

→ **Semaphore:**

A semaphore is an integer variable that allows many processes in a parrallel system to manage access to common resources like multitasking OS.

- Semaphore is non-negative integer variable.
- Semaphore is a OS abstract data type.
- It takes only integer value.
- It is used to solve critical section problem.

→ **Monitor:**

It is a synchronization technique that enable threads to mutual exclusion and wait() for a given condition to become true.

- Monitor is an object that contains both data and proccedure needed to perform allocation of shared resources.
- It is a programming language construct that support both data access synchronization and control synchronization.
- At any particular time only one process may be active in a monitor.

(2) Explain The producer consumer Problem.

-> It is a example of classic problem of synchronization.

-> Producer process produce data iteam that consumer process consume later.

-> Buffer is use between producer and consumer. Buffer size may be fixed or variable. Producer portion of application genrates data and store it in buffer. and consumer read that data from buffer.

-> Produce - consumer is also called bounded buffer problem. There are serval producer and consumer.

-> The producer-consumer problem shows that the need of synchronization in system where many processes share a resource.

-> In the problem two procceses share a fixed-size buffer. One process produces data iteams and store into buffer while other process consume data iteams from buffer.

classmate
Date
Page
Harsh Porwal

-> The producer can't deposit (store) data if the buffer is full. Similarly, a consumer can't retrieve data if buffer is empty.

-> If buffer is not full consumer can deposit its data and consumer should retreive a data iteam if buffer contains.

-> The consumer retrieves data item from buffer so buffer is decreasing if buffer is empty then the producer should be allowed to store data.

-> Problem arise when buffer is full and producer want to add new data iteam into buffer. Solution of this problem is that producer goes to sleep untill consumer collect iteam from buffer.

-> If consumer wants to consume more data but buffer is empty then consumer go to sleep untill producer add data to buffer.

-> The producer is blocked when buffer is full and the consumer is block when buffer is empty.

-> Speed of the producer and consumer is different. The producer process is faster than the consumer process.

-> Buffer have three different type of state:
(1) Full Buffer (2) Empty buffer
(3) Partially empty buffer.

-> When buffer is full, Producer have to wait. Producer do not produce data iteam.
-> When buffer is empty consumer have to wait.
-> Mutual exclusion is required for accessing the buffer.

-> Produce-consumer problem is solved by using semaphore, mutex and monitor. Mutual exclusion must be enforced by buffer it self.
-> Each data iteam produced by producer to the shared buffer must be consumed exectly once.
-> Data iteam must be consumed in same order in which it is out into the buffer.
-> We use variable counter to keep track of the number of data iteams in buffer.

Harsh Porwal

(3) Explain classical IPC problems with Reader-writer problem and Dinning philosopher problem.

→ Interprocess communication (IPC) requires the use of resources such as memory which are shared between process and threads.

→ Classical IPC problem are producer - consumer problem, readers-writers problem and dining philosopher.

→ **Reader-Writer Problem** :

→ when two types of process need to access a shared resources such as a file or database They call this process Readers and writers.

→ Multiple reader can read data from database as a same time but Multiple user can not update data same time it can only one user can update data at a time.

→ More than one reader can read data. When a writer uses shared data all other writers and reader must be excluded.

-> For example in Railway reservation system, Reader are those who want train schedule information.
They call readers because it can not change content of database.

-> The writers are those who making reservation on particular train.
Writer can modified the database.

-> When writer is active there are no other writer and reader are active.

-> If process wants to only read the file then it may be share the file with any other process that also want to read the file.

-> We apply rules for accessing order:

- If writer is waiting for readers to be finished do not allow more readers.
- If reader are waiting to finish writer, give reader priority.

-> If reader having higher priority than writer then there will be starvation with writer. and writer having higher priority than reader than starvation with readers.

classmate
Date
Page

Harsh Porwal

# * Dining Philosopher Problems

→ Dining philosopher problem is one of the classical process synchronization problem.

→ Here 5 philosopher are seated around a circular table. They spend their lives on thinking and eating. Five plates with five forks are kept on the circular table.

→ While philosophers think they ignore the eating, they do not require fork. When philosopher decide to eat then he must be obtain two fork, one from left side and another from right side fork. Philosophers pick up only a single fork at a time.

→ After consuming food the philosophers replace the fork and resume thinking. A philosophers to the left or right of a dinning phillophere can not eat while the dinning philosophere is eating, since forks are shared resources.

Problem analysis

→ Shared Resources : Here each fork is shared by two philosophers so we called it as a shared resource.

→ Race Condition : We do not want a philosophere to pick up a fork that has already being picked up by his neighbour. This is race condition

classmate
Date
Page
Harsh Porwal

→ Solution : We consider each fork as a shared iteam and protected by a mutex lock.

- Before eating each philosopher first lock left fork and then right fork.
- If philosophere acqires both locks succesfully then philosophere have two locks so he can eat food.
- After finishes eating, this philosophere release both forks and thinks.

classmate
Date
Page
Harsh Porwal

# Ch - 5

# Deadlock

## (1) Define Principles of dedlock.

-> A lack of process synchronization becomes result in two extream conditions are deadlock or starvation.

-> Deadlock is the problem of multiprogrammed system.

-> Deadlock can be defined as the permanant blocking of a set of process that either complete for system resources. Deadlock is more serious than starvation.

-> Deadlock can occure in shared resources such as printers, database, memory, CPU cycle etc.

-> A process in deadlock state if it was waiting for a particular event that will not occure.

-> Computer system is collection of finite number of resources. This resources are distributed among number of processes.
There are two type of resources.
(1) Reusable resources
(2) Consumable resources

-> Reusable resources only use by one process at a time. Process can release resources after use.
Ex. Files, Database, Primary and secondary memory, semaphore.

-> Consumable resources one of that can be created and destroyed.
There is no limit of consumable resources of particular type.
Ex. message, signal

-> Process utilize the resources in following sequence.

- Request for resources
- Use of resources
- Resources Release.

-> Process uses system call for requesting the resourcing. After allocating resources to the process it is used or operate resources.

-> The process release resources after use.

-> If resources is not avalaible when it is requested, the requesting process is forced to wait.

classmate
Date
Page
Harsh Porwal

# (2) Explain Resource Allocation Graph. (RAG)

-> Resource allocation graph is introduced by Holt. It is a directed graph that indicates a state of the system of resources and process.

-> Process and resources are represented by node. Graph consist set of vertices (v) and Edge (e).

-> A process of node graphically represented by circle.

-> A resource node graphically represented by rectangle and represented one type of resources.

-> The number of bullet symbol in a resourie node indicates how many units of that source is exist in the system.

-> An arrow from process to resource indicates that process is requesting the resources.
An arrow from resource to process indicates that resource has been allocated to process.

-> Claim edge P->R indicates process P may request resource R in the future. represented by dashed line.

-> Request age converted into an assignment edge when the resource is allocated to the process.
When resource is released by a process, assignment edge reconverts to claim edge.

-> Resource manager and othe OS processor can be involved in a deadlock.
Deadlock is a global condition rather than local.

-> An individual program can not detect deadlock. It is blocked and unable to use the processor to do any work.
Deadlock detection must be handled by operating system.

Harsh Porwal

(3) What is deadlock characteristics.

→ There are four condition are necessary for deadlock occure.

(1) Mutual Exclusion
(2) Hold and wait.
(3) No preemption
(4) Circular wait.

→ (1) Mutual exclusion : A resource may be acquired exclusively by only on process at a time.

→ (2) Hold and Wait : Process currently holding resources that were granted earlier can request new resources.

→ (3) No preemption : Once a process has obtain a resources, the system can not remove it from process until the process is finished.

→ (4) Circular wait : A circular chain of hold and wait condition exist in the system.

→ If one of them is absent, no resource deadlock is possible.

## (4) Deadlock Solution.

→ There are four approach for deadlock solution.

(1) Prevention
(2) Avoidance
(3) Detection
(4) Recovery

→ (1) Prevention : aim is to condition a system to remove any posibility of deadlock occuring.

→ (2) Avoidance : Deadlock can be avoided ~~using~~ by clearly identifying safe state and unsafe state.

→ (3) Detection : Deadlock detection method are use in system in which deadlocks can occure.

→ (4) Recovery : Used to resolve the deadlock from a system.

→ Deadlock prevetion and detection algorithm is used for ignoring the deadlock.

Harsh Porwal

## (5) Deadlock Prevention:

→ To prevent a deadlock, the OS must eliminate one of the four necessary conditions.

(1) Mutual exclusion
(2) Hold and wait
(3) No preemption
(4) Circular wait.

→ (1) Mutual exclusion : It is necessary in any computer system because some resources memory, CPU must be exclusively allocated to one user at a time.

→ (2) Hold and wait : If a process holding certain resources is denied a further request. it must be release its original resources if required request them again.

→ (3) No preemption : It could be by pass by allowing the OS to deallocate resources from processes.

→ (4) Circular wait : Circular wait can be by passed if the OS prevents the formation of circle.

## (6) Deadlock Recovery.

→ Once deadlock has been detected in system the deadlock must be broken by removing one or more of the four necessary conditions.

⇒ **Process termination :**

→ Deadlock removed by aborting a process. But aborting a process is not easy. All deadloked process are aborted.

→ Circular waits eliminated by aborting one by one process. There will be lot of overhead.

→ Following parameters are consider.

- Priority of the process.
- What percentage process finished its execution.
- Resource used by process.
- Need of resources to complete process remaining operation.
- How many process will need to terminated.
- Process type : Batch or Interactive.

⇒ **Resources preemption :**

→ Some time resource temporarily take away from its current process and allocated to other process.

classmate
Date
Page
Harsh Porwal

-> Following factor are considered.
- Priority of the process. Higher priority process are usally not selected.
- CPU time used by process.
- The number of other process that would be affected in this process were selected as victim.

## => Recovery through Rollback.

-> When a process in a system terminates, the system performs a rollback by undoning every operation related to terminal process.

-> Checkpointing a process means that its state is written to a file so that it can be restarted later.

-> Risk in this method is that the original deadlock may recover but the non determinacy of concurrent processing may ensure that this does not happen.

## => Starvation

-> Starvation is one type situation in which a process waits for an event that might never occure in the system.

-> Select the victim only for finite number of times.
-> Use Roll back method for selecting victim process.

## (7) Deadlock Avoidance with Banker's algorithm.

-> Deadlock avoidance depends on additional information about the long term resource needs of each process.

-> The system must be able to decide weather granting a resource is safe or not and only make the allocation when it is safe.

-> When a process is created it must be declare its maximum claim, the maximum number of unit resources. The resource manager can grant request if the resources are avalaible.

-> If process 1 requests a resource held by process 2 then make sure that process 2 is not waiting for resourse head by first process ~~st~~. 1.

* Banker's Algorithm

-> It is deadlock avoidance algorithm.
-> Banker's is named because the algorithm is modeled after a banker who makes loads from a pool of capital and recives payment that are returned to that pool.

Harsh Porwal

-> Algorithm is check to see if granting the request leads to an unsafe state. If it does the request is denied. If granting request to safe state it is carried out.

-> It is best known of the avoidance method. By using avoidance method the system is always kept in safe state.

-> Deadlock avoidance uses the worst-case analysis method to check for future deadlocks.

-> Safe state : There is atleast one sequence of resources allocation to process that does not result in deadlock.

-> System is in safe state only if there is exist of safe sequence. Safe state doesn't deadlock state.

-> Deadlock state is unsafe state. It does mean the system is in deadlock.

-> As long as the state is safe, the resource manager can be guaranteed to avoid a deadlock.

-> If system remains in safe state after allocating resources then only OS allocates resources to process.

→ Banker's Algorithm use following DS :

(i) Allocation : It is a table in which row represents process and column represent resources.

(ii) Max : Max be the maximum number of resources that process requires during its execution.

(iii) Need : It is current claim of a process where a process's claim is equal to its maximum need minus its current allocation.

Need = Max - Allocation.

(iv) Avalaible : There will be number of resources still avalaible for allocation.

= Number of resources - Sum of all allocation to all process.

* Disadvantages :

- It requires that there be a fixed number of resources to allocate.
- Number of user must be remain safe.
- Algorithm requires that the banker grants all requests within finite time.
- Algorithm require process returns all resources within finite time.

classmate
Date
Page
Harsh Porwal

Ex. Determine its need state and apply banker's algorithm.

| Process | Allocation | | | Max | | | Available | | |
|---|---|---|---|---|---|---|---|---|---|
| | A | B | C | A | B | C | A | B | C |
| $P_1$ | 0 | 1 | 0 | 7 | 5 | 3 | 3 | 3 | 2 |
| $P_2$ | 2 | 0 | 0 | 3 | 2 | 2 | | | |
| $P_3$ | 3 | 0 | 2 | 9 | 0 | 2 | | | |
| $P_4$ | 2 | 1 | 1 | 2 | 2 | 2 | | | |
| $P_5$ | 0 | 0 | 2 | 4 | 3 | 3 | | | |

→ Need Matrix :

Need = Max - Allocation

$P_1 = (7, 5, 3) - (0, 1, 0) = (7, 4, 3)$
$P_2 = (3, 2, 2) - (2, 0, 0) = (1, 2, 2)$
$P_3 = (4, 0, 2) - (3, 0, 2) = (6, 0, 0)$
$P_4 = (2, 2, 2) - (2, 1, 1) = (0, 1, 1)$
$P_5 = (4, 3, 3) - (0, 0, 2) = (4, 3, 1)$

| Process | Need | | |
|---|---|---|---|
| | A | B | C |
| $P_1$ | 7 | 4 | 3 |
| $P_2$ | 1 | 2 | 2 |
| $P_3$ | 6 | 0 | 0 |
| $P_4$ | 0 | 1 | 1 |
| $P_5$ | 4 | 3 | 1 |

→ Apply banker's algorithm.

- Step 1: For $P_1$

Need ≤ Avalaible

7, 4, 3 ≤ 3, 3, 2 condition is false.

- Step 2: For $P_2$

1, 2, 2 ≤ 3, 3, 2 condition true

- New Avalaible = avalaible + Allocation

= (3, 3, 2) + (2, 0, 0)

= (5, 3, 2)

- Step 3: For $P_3$

6, 0, 0 ≤ 5, 3, 2 condition is false.

- Step 4: For $P_4$

0, 1, 1 ≤ 5, 3, 2 condition is true

- New avalaible = avalaible + Allocation

= (5, 3, 2) + (2, 1, 1)

= (7, 4, 3)

- Step 5: For $P_5$

4, 3, 1 ≤ 7, 4, 3 condition is true.

- New avalaible = avalaible + allocation

= (7, 4, 3) + (0, 0, 2)

= (7, 4, 5)

Harsh Porwal

- Step 6 : For $P_1$

~~7, 4, 3 ≤ 7, 4, 5~~

7, 4, 3 ≤ 7, 4, 5 condition is true.

- New avalaible = avalaible + allocation
  = (7, 4, 5) + (0, 1, 0)
  = (7, 5, 5)

- Step 7 : For $P_3$

6, 0, 0 ≤ 7, 5, 5 condition is true.

- New avalaible = avalaible + allocation
  = (7, 5, 5) + (3, 0, 2)
  = (10, 5, 7)

→ Hence we apply bankers algorithm and find safe state and safe sequence like $P_2$, $P_4$, $P_5$, $P_1$, $P_3$.

classmate
Date
Page
Harsh Porwal

# Ch: 5 Memory Managment

(i) Explain link list method for dynamic memory managment.

→ Application of linked list concept are useful to model many different abstract data type such as queus, stack and trees.

→ If we restrict the process of insertion to one end of list and deletion ~~of~~ to other end then we have model of a queue.

→ We can insert an item at rear and delete a item at front. This obey the decipliane of FIFO. There are many example of queue in real life application.

→ If we restrict the process of Insertion and deletion to occure only at beginning of the list then we have another data structure known as stack.

→ Stacks are also refered to as push down list. Stacks are sometime reffered as LIFO.

-> List, queus and stacks are all inhernently one dimensional.
A tree represent two dimensional application of linked list.

* Application of linked list

-> Use the sizeof operator to determine the size of a linked list.

-> When using memory allocation functions malloc, calloc, test for a NULL pointer return value.

-> Never call memory allocation function with with a zero size.

-> Release dynamic allocated memory when it no longer required.
-> Using free function to release the memory not allocated dynamically with malloc or calloc.
-> Using a pointer after it's memory has been released is an error.
-> It is an error to assign the return value from malloc or calloc to anything other than pointer.
-> It is an error to release individually the elements of an array created with calloc.
-> It is logic error to fail to set the link field in the last node to null.

Harsh Porwal

(2) Explain multiprogramming with fixed partition.

→ In OS, memory managment is function for allocating and managing computer main memory. It keep tracks of status of each memory location, either allocated or free to ensure effective and efficient use of primary memory.

→ There are two memory managment techniques : Contiguous and Non-Contiguous. Contiguous technique can be divided into (1) Fixed partition (2) Variable partition.

⇒ In a fixed partition of any main memory all partitions are of the same size. The OS divides main memory to number of fixed size partition. Each partition hold program.

→ It is relatively simple to implement. Degree of multiprogramming depends on the number of partition.

→ Normally main memory can be divided in two partition. (1) For resident program (2) For user program.

→ Batch OS using fixed size partition scheme. The OS keeps the record of memory allocation and deallocation in the form of table. Initially all the memory is avalaible for user processes.

-> When the process arrives in the system and needs memory, OS search for large hole for this process.
Hole is one large block of free available memory.
If any free hole is found, process is allocated to the free hole of memory.

-> After allocating number of holes for the process, a set of various size holes is scattered throughout memory at any given time.

-> Fixed size partition are simple to implement. Any process whose size are equal to the partition size can be loaded into any available partition.

-> There are two difficulties
(i) A program may be to big to fit into partition.
(ii) Memory utilization is extremely inefficient.
It can be solved Using overlays.

-> Advantage : Simple to implement, Less overhead, Does not require expertise to understand.

-> Disadvantage : Memory is not fully utilized, Poor utilization of process, User process being limited to size, Requires contiguous loading of entire program.

classmate
Date
Page

Harsh Porwal

(3) What is pagging? Explain pagging mechanism in MMU with example.

-> In OS, paging is a storage mechanism used to retrieve processes from the secondary storage into the main memory in the form of the page.

-> The main idea behind paging is to divide each process in form of page. The main memory will also divided in the form of frames.

-> One page of the process is to be stored in one of the frames of the memory. The pages can be stored at the different locations of the memory but the priority is always find to contiguous frames or holes.

-> Pages of the process are brought ~~to~~ into main memory only when they are required Otherwise they reside in secondary storage.

-> Different OS defines different frame sizes. Size of each frame must be equal. Considering the fact that pages are mapped to the frame into pagging, page size needs to be as same as same frame size.

* Example :

→ Let us consider main memory size of 16 Kb and frame size is 1 Kb therefore the main memory will be divided into the collection of 16 frames. of 1 Kb each.

→ There are 4 processes in system that is $P_1$, $P_2$, $P_3$ and $P_4$ of 4 kb each. Each process is divided into pages of 1 Kb each so that one page can be stored one frame.

Harsh Porwal

# * Memory Managment Unit

-> The purpose of MMU is to convert the logical address into the physical address. The logical address is the address genrated by the CPU for every page while the physical address is the actual address of the frame where each page will be stored.

-> When a page to be accessed by CPU using the logical address, the OS needs to obtain physical address to access that page physically.

-> The logical address has two parts.
(1) Page number (2) Offset

-> MMU of OS needs to convert the page number to the frame number.

## (4) Swapping

→ Keeping all process in memory all the time requires a large ammount of main memory. For execution of process it must be swapped.

→ Swapping is method of temporarily removing inactive process from the physical memory.

→ Inactive process means which is neither executing on CPU, nor performing an I/O operation.

→ Swapping is a main memory managment scheme in which any process can be temporarily swapped from main memory to secondary memory so that main memory can be avalaible for furthere operations.

→ It is used to improve main memory utilization.

→ In secondary memory where the swapped out process is stored is called swap space.

Harsh Porwal

→ The purpose of swapping in OS is to acess the data present in the hard disk and bring it to RAM so that application program can use it.

→ It is important to remember swapping is used only when data is not present in RAM.

→ Although the process of swapping affects the perfomance of the system. it helps to run larger and more than one process. This is reason why swapping also refered as memory compaction.

→ The concept of swapping is divided into two more concept.

- Swap-Out is a method of removing a process from RAM and adding into hard disk.
- Swap-In is a process of removing data from hard disk and adding to RAM.

→ Advantage : Helps CPU to manage multiple process.
- Help to create and use virtual memory.
- Improves main memory utilization.

→ Disadvantage: If computer loses power the user may lose all information related to it.
- If swapping method is not good then composite method can increase number of page fault. and decrease perfomance.

## (5) Virtual Memory

→ Virtual memory is a method of using hard disk space to provide extra memory. It simulates additional main memory.

→ In windows operating system ammount of virtual memory avalaible equals to the ammount of free main memory, plus the ammount of disk space allocate to swap file.

→ A swap file is an area of your hard disk that is asign for virtual memory. Swap files can be either temporary or permanant.

→ Virtual memory is stored in Secondary storage device. It helps to extend additional memory capacity and work with primary memory to load application.

→ It reduce cost of expanding physical memory.

→ The portion of process that is actually in main memory at any time is defined to be the resident set of the process.

→ Virtual adress space is much larger than the physical memory. It works with the help of SS device and it's speed is low as compare to physical storage location.

Harsh Porwal

→ Virtual memory uses two types of addresses: Virtual address and physical address.

→ Virtual address is reffered by process and physical address is ~~re~~ actual address of main memory.

→ Whenever a process access virtual address the system must translate it to a physical address. It uses special purpose hardware. This hardware is called Memory managment unit.

→ Most virtual memory system use method of pagging. The virtual address is divided into fixed size unit called ~~frame~~ pages.
The corresponding in physical memory called page frames. The page and pageframes are ~~m~~ normally same size. The area on hard disk that stores page frames is ussally called the pagging file or swap file.

→ Advantage : degree of multiprogramming will increase.
- User can run large application with less real RAM.
- No need to buy more RAMs.

→ Disadvantage : System becomes slower since swapping takes time.
- Takes more time in switching between application.
- User will have lesser hard disk space for use.

(6) What is called Segmantation ? How it differ from pages.

-> In segmantation a programs data and instructions are divided into blocks called segments.

-> Logical view : A process consist of set of segments

-> Physical view : It consists of non-adjacent area of memory allocated to segmant.

-> All segmant size may be equal or may not be equal. Segmantation support user view of memory.

-> Segmantation can be implemented with or without pagging.

-> Collection of segmant called as logical address space. each segmant is identified by it's name.

-> Processor genrates logical address. These consists of a segmant number and an offset into the segmant.

-> Segmant names are normally symbolic names.

Date
Page

Harsh Porwal

| Segmantation | Paging |
|---|---|
| → Program is divided into variable size segmant | Program is divided into fixed size pages. |
| → User is responsible for dividing program into segmant. | Division of pages perfor by OS. |
| → It is slower than paging | It is faster then Segmantation. |
| → It is visible to user. | It is invisible to user. |
| → eliminates internal fragmantation. | Suffers from internal fragmentation. |

→

| Segmantation | Demand Paging |
|---|---|
| → Segmants may of different size. | Pages are same size. |
| → Segmant can be shared. | Pages can not be shared. |
| → It allows dynamic growth of segmant. | Page size is fixed. |
| → Segmant map table indicates the address of each segmant | Page map table keep track of pages in memory. |
| → It is allocated to program while compliation. | Pages are loaded in memory on demand. |
| → Provides vitual memory. | Provides virtual memory. |

## (7) Page Replacement

→ When a page fault occurs page replacement algorithm are used for loading the page in memory and no free page frame exist in memory.

→ Page faut occurs if a running process refrences a nonresident page.

→ The goal of replacement strategey is to minimize the fault rate. To evalute a replacement algorithm following parameter are used.
(i) Size of Page (ii) Set of string
(iii) Number of page frames.

→ Given these inputs we can determine number of page fault and hence page fault rate.

→ When we find that a page frame reference is in main memory then we have a page hit and when page fault occurs we say it is page miss. A poor choice of page replacement algorithm may result in lot of page misses.

→ UNIX OS uses global page replacement policy.

classmate
Date
Page
Harsh Porwal

## * FIFO :-

-> FIFO page replacement algorithm removes the pages that have been in memory the longest. Here system keep track of the order in which pages enter primary memory.

-> When page is replaced, the oldest page is selected. System maintain FIFO queue of page pages are inserted at tail of the queue.

Ex. Reference String = 1, 2, 3, 4, 2, 5, 3, 4, 2, 6, 7, 8, 7, 9, 7, 8, 2, 5, 4, 9

### With - 3 Frames

| Ref String | 1 | 2 | 3 | 4 | 2 | 5 | 3 | 4 | 2 | 6 | 7 | 8 | 7 | 9 | 7 | 8 | 2 | 5 | 4 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $P_1$ | 1 | 1 | 1 | 4 | 4 | 4 | 4 | 4 | 4 | 6 | 6 | 6 | 6 | 9 | 9 | 9 | 9 | 9 | 4 | 4 |
| $P_2$ | | 2 | 2 | 2 | 2 | 5 | 5 | 5 | 5 | 5 | 7 | 7 | 7 | 7 | 7 | 7 | 2 | 2 | 2 | 9 |
| $P_3$ | | | 3 | 3 | 3 | 3 | 3 | 3 | 2 | 2 | 2 | 8 | 8 | 8 | 8 | 8 | 8 | 5 | 5 | 5 |
| PF | * | * | * | * | | * | | | * | * | * | * | | * | | | * | * | * | * |

P.F = 14

### With 4 - Frames

| Ref String | 1 | 2 | 3 | 4 | 2 | 5 | 3 | 4 | 2 | 6 | 7 | 8 | 7 | 9 | 7 | 8 | 2 | 5 | 4 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $P_1$ | 1 | 1 | 1 | 1 | 1 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 9 | 9 | 9 | 9 | 9 | 9 | 9 |
| $P_2$ | | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 2 | 2 | 2 | 2 |
| $P_3$ | | | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 5 | 5 | 5 |
| $P_4$ | | | | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 4 | 4 |
| PF | * | * | * | * | | * | | | | * | * | * | | * | | | * | * | * | |

P.F = 12

## * LRU [Least Recently used]

→ LRU stands for least recently used. This replacement policy choosen to replace the page which has not been referenced for the longest time.
This policy assume the recent past will approximate the immediate future.

→ System must maintain a time stamp for every page that indicates the time of last access.
When deciding which page to remove the OS must scan all pages of memory for finding oldest stamp.

→ LRU is implemented by maintaining pages in ordered list.

Ex. Reference String = 1, 2, 3, 4, 2, 5, 3, 4, 2, 6, 7, 8, 7, 9, 7, 8, 2, 5, 4, 9

With - 4 Frames

| Ref String | 1 | 2 | 3 | 4 | 2 | 5 | 3 | 4 | 2 | 6 | 7 | 8 | 7 | 9 | 7 | 8 | 2 | 5 | 4 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $P_1$ | 1 | 1 | 1 | 1 | 1 | 5 | 5 | 5 | 5 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 2 | 2 | 2 | 2 |
| $P_2$ | | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 9 | 9 | 9 | 9 | 5 | 5 | 5 |
| $P_3$ | | | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 4 | 4 |
| $P_4$ | | | | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 9 |
| PF | * | * | * | * | | * | | | | * | * | * | | * | | | * | * | * | * |

P.F = 13

→ Check left side which is Last Recently used.

Harsh Porwal

## * Optimal

-> The data which ~~is~~ will not be accessed permanantly and for a longest time should be replace in optimal replacement algorithm.

-> Optimal page algorithm gives less page fault as compared to FIFO and LRU.

-> Use of this page replacement algorithm guarantees the lowest possible page fault rate for a fixed number of pages.

-> It is difficult to implement for furture reference.

Ex. Reference String = 1, 2, 3, 4, 2, 5, 3, 4, 2, 6, 7, 8, 7, 9, 7, 8, 2, 5, 4, 9

| Ref String | 1 | 2 | 3 | 4 | 2 | 5 | 3 | 4 | 2 | 6 | 7 | 8 | 7 | 9 | 7 | 8 | 2 | 5 | 4 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $P_1$ | 1 | 1 | 1 | 4 | 4 | 4 | 4 | 4 | 4 | 6 | 7 | 7 | 7 | 7 | 7 | 7 | 2 | 2 | 4 | 4 |
| $P_2$ | | 2 | 2 | 2 | 2 | 5 | 5 | 5 | 5 | 5 | 5 | 8 | 8 | 8 | 8 | 8 | 8 | 5 | 5 | 5 |
| $P_3$ | | | 3 | 3 | 3 | 3 | 3 | 3 | 2 | 2 | 2 | 2 | 2 | 9 | 9 | 9 | 9 | 9 | 9 | 9 |
| PF | * | * | * | * | | * | | | * | * | * | * | | * | | | * | * | * | |

P.F = 13

| Ref String | 1 | 2 | 3 | 4 | 2 | 5 | 3 | 4 | 2 | 6 | 7 | 8 | 7 | 9 | 7 | 8 | 2 | 5 | 4 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $P_1$ | 1 | 1 | 1 | 1 | 1 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 9 | 9 | 9 | 9 | 9 | 9 | 9 |
| $P_2$ | | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 5 | 5 | 5 |
| $P_3$ | | | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 6 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 4 | 4 |
| $P_4$ | | | | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 |
| PF | * | * | * | * | | * | | | | * | * | * | | * | | | | * | * | |

P.F = 11

(8) Explaining following allocation algorithm.
First-fit, Best-fit, Worst-fit

→ * First-fit : It scans linked list and whenever it finds the first big enough hole to store a process it stop scanning and load process into hole.

- This procedure produce two partitions out of them one partition will be a hole while other will store the process.
- It maintain linked list according to increasing order.
- It is simple to implement among all the algorithm.

* Next-fit : It is similar to first fit exept the fact that next fit scans the link list from the node where it previously allocated a hole.

- It doesn't scan the whole list, it starts scanning the list from the next node.
- The idea behind next fit is the fact that the list has been scanned once therefore the probability of finding the hole is larger in the remaining part of the list.

Harsh Porwal

* **Best-fit** : Best fit algorithm tries to find out the smallest ~~path~~ hole possible in the list that can accommodate the size requirment of the process.

- It is slower because it scans the entire list every time and tries to find out the smallest hole which can satisfy requirment of process.

* **Worst-fit** : It scans the entire list every time and tries to find out the biggest hole in the list which full fill requirment of process.

- This algorithm produces larger holes to load the other processes this is not better approach due to the fact that it is slower because it searches the entire list every time again and again.

(a) Define following terms.

→ Throughput : Number of completed process per time unit.

→ Waiting time : Time a process waits in the ready state.

→ Turnaround time : Time a process taken from submission to completion.

→ Response time : Ammount of time it takes from when a request was sumbitted until the first response is produced not output.

→ Granularity : It is the ratio of computation time ~~an~~ to communicateion time. Where, computation time is the time ~~where~~ required to perform the computation of task. Communication time is the time required to exchang data between processor.

→ Short-term scheduler : It ensures which program is suitable or important for processing.

→ CPU-Utilization : It refers to a computer usage ~~of~~ of processing resources or the ammount of work handled by CPU.

classmate
Date
Page
Harsh Porwal

(10) Explain thrashing.

-> Thrashing occurs when a process doesn't have enough frames allocated to the store the pages it uses repeatedly, the page fault rate will be very high.

-> A process is thrashing if it is spending more time for paging inlout than executing.

-> Thrashing leads to low CPU utilization, the operating system's medium or long term scheduler may step in, detect this and increase the degree of multiprograming.

-> Local replacement algorithm can limit the effects of thrashing. If the degree of multiprograming is increase over a limit, processor utilization falls down considerably because of thrashing.

-> The selection of replacemant policy to implement virtual memory plays an important part in the elimination of potential for thrashing.

classmate
Date
Page
Harsh Porwal

→ A policy based on the global mode is more likely to cause thrashing.

→ All pages of memory are avalaible to all transactions a memor intensive transaction may occupy large portion of memory, making other transactions susceptible to page faults and resulting in system that thrashes.

Date
Page

Harsh Porwal

# Ch-6

# I/O Managment

(1) Explain Goals of I/O software.

-> It is used for interaction with I/O devices like mouse, keyboard, USB devices, printer etc. I/O software organize in following ways.

-> User level libraries : Provides a simple interface to program for input output function.

-> Kernel level Modules : Provides device driver to interact with the device independent I/O modules.

-> Hardware : A layer including hardware controller and actual hardware which interact with device driver.

* Uniform Naming : For example naming of file system in OS is done in a way that user does not have to be aware of underlying hardware name.

* Error handling : Error are mostly genrated by controller and also they are mostly handled by controller itself. When lower level solve the problem it does not rich upper level.

* **Device Independence** : Most important part of I/O software is device independance.
It is always most preferable to write program which open all other I/O devices. For example it is not necessary to write the input taking program again and again for taking input from various files and devices.

* **Buffering** : Data that we enter into a system can not be stored directly in memory. For example data is converted into smaller groups than transferred to outer buffer for examination. Buffer have major impact on I/O software as it is helps storing and copying data. Many device have constraints and just to avoid it some data is always put into the buffer in advance so the buffer rate of getting filled with data and getting empty remains balanced.

* **Sharable - Non sharable devices** : Device like hard disk can be shared among multiple process while device like printer can not be shared. The goal of I/O software is handled ~~by~~ to both types of devices.

classmate
Date
Page
Harsh Porwal

(2) Explain Device drivers.

→ A device driver is a programm that controls a particular type of device that is attached to user computer. There are device drivers for printer, display, CD-ROM readers.

→ Applications do not directly interact with I/O devices. But it will interact with the help of OS. some device driver provides access to user ~~to~~ space ~~help~~ ~~of~~ application and the devices are managed directly by the applications.

→ Genral example of device controller include hard disk controller, display and audio controller that in turn manage devices connected to them.

→ Device driver has two parts.
Device specific, OS specific

→ The device specific portion of a device remain the same accross all OS, and is more about understanding and decoding the device data sheet than software programming
A data sheet is a document with technical details of device, including its operation, performance.

→ The OS specific portion is one that is tightly coupled with the OS mechanism of user interface. and thus differentiate a linux device driver from a windows device driver.

* Genral classifications :

→ Serial port devices : Universal Asynchronous Reciver transmitter (UART) interface is used for connecting host. RS232 serial cable is used in between host and serial port drivers.

→ Bus drivers : Provide hardware specific interfaces for the corresponding hardware protocoles and are the bottom most horizontal software layes on OS.

→ Network device driver : Provide interface ~~in~~ is ethernet interface and wifi interface. This driver connects to the TCP/IP protocol stack within the OS.

→ Serial ATA : Secondary storage disk are connected to this interface. This interface defines an advanced host controller interface (AHCI). The driver provides block services to a file system driver.

Harsh Porwal

→ Flash : Flash is one type of storage for block drivers. Blocks are erase, write and program by using special program.

→ Interrupt controller : It is integral part of OS and it could be considered as device driver.

→ A device driver has three side of interface.

(1) Rest of the kernel
(2) Hardware
(3) User.

classmate
Harsh Porwal
Page
Harsh Porwal

(3) Explain Direct Memory Access.

→ DMA is an I/O technique that allows a control unit to directly access main memory. This means that once reading or writing has begun the remander of the data can be transferred to and from memory without processor intervention.

→ Without DMA, the processor is responsible for the physical movment of data between main memory and a device.

→ A special control unit may be provided to allow transfer of a block of data directly between an external device and the main memory, without continuous intervention by the processor. This approach is called DMA.

→ DMA can be used with either polling or interrupt software. DMA is particularly useful on devices like disks, where many bytes of information can be transfferred in single I/O operations When used in conjuction with an interrupt the CPU is notified only after the entire block of data has been transferred. For each byte or word transfred, it must provide the memory address and all the bus signals that control the data transfer.

→ DMA controller is programmed by CPU and its registers are set to perform required operations. It issues a command to a DMA controller for reading data from the secondary storage disk and stores it in the internal buffer.

→ Data transfer is start by disk controller by issuing read request over the bus. The DMA controller then puts the memory address on the bus.

→ Disk controller fetches the next word from the internal buffer, it knows where to write it.

→ Disk controller sends an acknowledgment to DMA controller when writes completes.

→ DMA controller ~~sends~~ increments the memory address to use and decrements the byte count.

The step 2 to step 4 is repeated until byte count ~~to~~ reach to zero.

Harsh Porwal

→ DMA controller has three register as follow.

- Address register : It contains the address to specify the desired location in memory.

- Word count register : It contains the number of words to be transffered.

- Control register : It specifies the transfer mode.

→ The CPU Initializes the DMA by sending given information through databus

- The starting address of the memory block where the data is avalaible to read or where data is avalaible to write.
- It also sends word count which is the number of words in the memory block to read or write.
- Control to define the mode of transfer such as read or write.
- A control to begin the DMA transfer.

(4) Explain RAID

→ Raid means Redundant array of independent disks. It consist of multiple disks. Raid provides protection against hard disk failures.

→ Raid implementation improves the I/o perfomance of storage system by storing data across multiple hard disk. Data is organize in multiple way because of many hard disk. Data is stored in serval disk instead of single disk.

→ RAID capacity depends upon the type of RAID Array user build and the number size of disk drives.

→ Raid consist of 7 layers. RAID is implemented in two ways Software and Hardware RAID.

* Software RAID

→ It runs on the server processor. It is implemented at the OS level. Dedicated h/w is not required to manage the software RAID.

→ It consume server memory and CPU, so perfomance of server is decreased.

classmate
Date
Page
Harsh Porwal

-> It is simpler to implement than hardware RAID. It does not support all level of RAID.

-> The upgradation of OS means upgradation of software RAID so it may create problem for data processing.

-> Software RAID also work with cheaper IDE disks as well as SCSI disks.

* Hard-ware RAID

-> Hardware RAID is implemented in two ways.
(i) Spcidlized hardware controller on the server.
(ii) Specialized hardware controller on Array.

-> Special hardware controller is required to install on server machine or on the array. It is not dependent on OS. It does not occupy any server memory.

-> Host CPU can execute applications while the array adapter's process simultaneously executes array functions.

-> In hardware RAID on the server, RAID controller interacts with hard disk using the PCI bus.

→ Array based hardware RAID uses external controller. External controller acts as the interface between the host and the hard disks.

* RAID Array Components

→ It consits of multiple hard disk with required hardware and software.
Hardware and software is required to implement RAID array.
It also consists of physical array and logical array.

→ Physical array contains the fixed number of hard disk, power supply and supporting hardware.

→ Logical array contains subset of disks within a RAID array. It is also called RAID set or RAID group.

→ OS identifies the logical volume. Actually logical volume mean physical hard disk.

→ Physical hard disk managed by RAID controller RAID level depends on the ~~computer~~ number of hard disk used in logical array.

classmate
Date
Page
Harsh Porwal

(5) Explain RAID Levels

→

* Raid-0 (striping) : This configuration has striping but no redudancy of data. it offers the best performance, but it does not provide ~~fault~~ fault tolerance.

RAID - 0

* RAID-1 (Mirroring) : Also known as disk mirroring this configuration consists of at least two drives that duplicate the storage data.

There is no striping. Read perfomance is improved, write perfomance same as single disk storage.

RAID-1

* **RAID - 2** : This configuration uses striping across disk, with some disk storing error checking and correcting information. RAID-2 also uses a dedicated Hamming code parity, a linear from of ECC. RAID-2 has no advantage over RAID-3 and is no longer used.

RAID - 2

| Disk-1 | Disk-2 | Disk-3 | Disk-4 | Disk-5 | Disk-6 | Disk-7 |
|---|---|---|---|---|---|---|
| B - $A_1$ | B - $A_2$ | B - $A_3$ | B - $A_4$ | A - $P_1$ | A - $P_2$ | A - $P_3$ |
| B - $B_1$ | B - $B_2$ | B - $B_3$ | B - $B_4$ | B - $P_1$ | B - $P_2$ | B - $P_3$ |
| B - $C_1$ | B - $C_2$ | B - $C_3$ | B - $C_4$ | C - $P_1$ | C - $P_2$ | C - $P_3$ |
| B - $D_1$ | B - $D_2$ | B - $D_3$ | B - $D_4$ | D - $P_1$ | D - $P_2$ | D - $P_3$ |

P = Parity

* **RAID - 3** : This technique uses striping and dedicates one drive to storing parity information. The embedded ECC information is used to detect errors. Data recovery is accomplished by calculating the exclusive information recorded on the other drives. Because an I/O operation addresses all the drives at same time, RAID-3 can't overlap I/O. For this reason RAID-3 is best for single user systems with long record applications.

RAID - 3

| | | | |
|---|---|---|---|
| B - $A_1$ | B - $A_2$ | B - $A_3$ | A - Parity (1-3) |
| B - $A_4$ | B - $A_5$ | B - $A_6$ | A - Parity (4-6) |
| B - $B_1$ | B - $B_2$ | B - $B_3$ | B - Parity (1-3) |
| B - $B_4$ | B - $B_5$ | B - $B_6$ | B - Parity (4-6) |

classmate
Date
Page
Harsh Porwal

* **RAID-4** : This level uses large stripes, which means a user can read record from any single drive.

Overlaped I/O can be used for read operation are required to update the parity drive, no I/O overlapping is possible.

RAID-4

* **RAID-5** : This level is based on parity block level striping the parity information is striped across each drive, enabling the array to function, even if one drive were fail. The array's architecture enables read and write operation to span multiple drives. This result in perfomance better than that of a single drive but not high as RAID-0 Array. RAID-5 requires at least three disks but it is often recommanded to use at lest 5 disk for perfomance reason.

RAID - 5

| | | | |
|---|---|---|---|
| B - A1 | B - A2 | B - A3 | Block - A Parity |
| B - B1 | B - B2 | Block - B Parity | B - B3 |
| B - C1 | Block - C Parity | B - C2 | B - C3 |
| B - D0 | B - D1 | B - D2 | B - D3 |

* RAID - 6 : This technique similar to RAID - 5 but it includes second parity scheme distributed across the drivers in array. The use of additional parity enables the array to continue functioning even if two disk fail ~~siman~~ simultaneously. However this extra protection comes at a cost. RAID - 6 array have slower write perfomance than RAID - 5 arrays.

RAID - 6

| | | | |
|---|---|---|---|
| B - A1 | B - A2 | B - A3 | B - A0 |
| B - B1 | B - B2 | B - B0 | B - B0 |
| B - C1 | B - C0 | B - C0 | B - C2 |
| B - D0 | B - D0 | B - D1 | B - D2 |
| B - E0 | B - E1 | B - E2 | B - E3 |

classmate
Date
Page
Harsh Porwal

(6) Explain Disk scheduling with algorithm.

-> Disk scheduling is done by OS to schedule I/O request arriving for the disk. Disk scheduling is also known as I/O scheduling.

-> Multiple I/O request may arrive by different processes and only one I/O request can be served at a time by disk controller. Thus other I/O request have to wait in waiting queue.

-> Two or more request may be far from each other so can result in greater disk arm moment.

-> Hard drives are one of the slowest parts of the computer system and thus need to be accessed in an efficient manner.

-> Seek time : It is the time taken to locate the disk arm to a specific track where data is to be read or write.

-> Rotational latancy : It is the time taken by the desired sector of disk to rotate into position so that it can access read/write head.

-> Transfer time : It is the time to transfer the data. It depends on rotating speed of disk and number of bytes to be transffered.

-> Disk Response time : It is average time of spent by a request waiting to perform I/O operation. Average response time is the response time of all request.

-> Disk access time : Dat = Seek time + Rotational latency + Transfer time.

* FCFS

-> FCFS is the simplest of all the disk scheduling algorithm. In FCFS the request are addressed in the order they arrive in the disk queue.

-> Advantage : Every request gates a fair chance.
No ~~iden~~ indefinite postponement.

-> Disadvantage : Does not try to optimize seek time
• May not provide the best possible service.

classmate
Date
Page
Harsh Porwal

Ex. (82, 170, 43, 140, 24, 16, 190)
head is : 50

Seek time = (82-50) + (170-82) + (170-43) + (140-43)
+ (140-24) + (24-16) + (190-16)
= 642

Not in book

X → Algorithm :

(i) Let request array represents an array storing indexes of tracks that have been requested in ascending order of their time of arrival. head is the position of disk head.
(ii) Let us one by one take the track in default order and calculate the absolute distance of the track from the head.
(iii) Increment the total seek count with this distance.
(iv) Currently serviced track position now becomes the new head position.
(v) Go to step 2 until all tracks in request array have not been serviced.

## * SSTF

→ Shortest seek time first requests having shortest seek time are executed first.

→ So, the seek time of every request is calculated in advance in the queue and they are schedule according to their calculated seek time.

→ As a result ~~seek time~~ the request near disk arm will execute first.

→ SSTF certainly an improvment over FCFS as it decrease the average response time and increases the throughput of system.

→ Advantage : Average response time decreases
- Throughput increases.

→ Disadvantage : overhead calculate seek time in advance.
- High variance response time as SSTF favours only some request.

Ex. (82, 170, 43, 140, 24, 16, 190)
head is : 50

→ Total seek time = (50 - 43) + (43 - 24) + (24 - 16) + (82 - 16) + (140 - 82) + (170 - 140) + (190 - 170)
= 208

classmate
Date
Page
Harsh Porwal

## * SCAN

→ In this the disk arm moves into a particular direction and services the request coming in its path and after reaching the end of disk it reverse its direction and again service the request arriving in its path.

→ So this algorithm work as an elevator algorithm. As a result request at mid range are serviced more and those arriving behind the disk arm will have to wait.

→ Advantage : High throughput
- low variance of response time
- Average response time.

→ Disadvantage : long waiting time for requests for location just visited by disk arm.

Ex. (82, 170, 43, 140, 24, 16, 190) And head is 50, and disk arm should move towards the larger value.

→ Total seek = (199 - 50) + (199 - 16)
= 332

classmate
Date
Page
Harsh Porwal

## * Look

→ It is similar to SCAN disk algorithm except for the difference that the disk arm in spite of going to the end of the disk goes only to the last request to be served in front of the head and then reverse its direction from there only. Thus it prevents the extra delay which ocurred due to unnecessary traversal to the end of the disk.

Ex. (82, 170, 43, 140, 24, 16, 190) and the head is 50, and it is given that disk arm should move towards the larger value.

→ Total seek time = (190 - 50) + (190 - 16)
= 314

## 8.1 Security

GTU : Winter-13

- Computer security addresses the issue of preventing unauthorized access to resources and information maintained by computers. Computer system must provide mechanism to manage security threats.
- Basic methods used to protect software and hardware is passwords, backups, maintenance of written security policies and user training.
- **Computer security** deals with the prevention and detection of unauthorized actions by users of a computer system.
- How to protect the valuable assets? It is necessary to keep in safe place like a bank to protect the valuable assets. But bank is not a safe place now a day. There are so many example where bank robbery in our country.
- Bank robbery is the crime of stealing from a bank during opening hours. Protecting assets was difficult and not always effective.
- Now a day, protection is easier because many factors working against the potential criminal. Very sophisticated alarm and camera systems silently protect secure places like banks.
- Traditionally information security provided by physical i.e. rugged filing cabinets with locks and administrative mechanisms i.e. personnel screening procedures during hiring process.
- Asset protection systems are designed to recover stolen cash and high value assets, apprehend criminals and deter crime. The system has the capacity to track, protect and manage critical assets in real-time.
- The techniques of criminal investigation have become so effective that a person can be identified by genetic material, voice, retinal pattern, fingerprints etc.
- Use of networks and communications links requires measures to protect data during transmission.
- **Data security** is the science and study of methods of protecting data from unauthorized disclosure and modification.
- Data and information security is about enabling collaboration while managing risk with an approach that balances availability versus the confidentiality of data.
- **Computer security :** Generic name for the collection of tools designed to protect data and to hackers.
- **Network security :** Measures to protect data during their transmission.
- **Internet security :** Measures to protect data during their transmission over a collection of interconnected networks.

## 8.2 User Authentication

GTU : Summer-13

### Authentication

- Authentication techniques are used to verify identity. The authentication of authorized users prevents unauthorized users from gaining access to corporate information systems.
- Authentication method is of validating the identity of user, service or application. The use of authentication mechanisms can also prevent authorized users from accessing information that they are not authorized to view.
- Data authentication means providing data integrity as well as that the data have been received from the individual who claimed to supply this information.

In authentication :

a) A Brute force attack is an automated process of trial and error used to guess a person's user name, password, credit-card number of cryptographic key.

b) Insufficient authentication occurs when a website permits an attacker to access sensitive content or functionality without having to properly authenticate.

c) Weak password recovery validation is when a website permits an attacker to illegally obtain, change or recover another user's password.

### Authorization

- Authorization is a procedure of controlling the access of authenticated users to the system resources. An authorization system provides each user with exactly those rights granted to them by the administrator.
- Besides providing users with access rights to files, directories, printers etc, an authorization system might control user privileges, such as local access to the server, setting the system time, creating backup copies of the data and server shutdown.

In authorization :

a) Credential/session prediction is a method of hijacking or impersonating a website user.

b) Insufficient authorization is when a website permits access to sensitive content or functionality that should require increased access control restrictions.

c) Insufficient session expiration is when a website permits an attacker to reuse old session credentials or session IDs for authorization.

## 8.2.1 Password Based Authentication

- Password is a front line protection against the unauthorized access (intruder) to the system. A password authenticate the identifier (ID) and provides security to the system. Therefore almost all systems are password protected.

### 1. Password vulnerability

- Passwords are more common. Passwords are easily guessed. Use of mechanisms to keep passwords secret does not guarantee that the system security can not be broken. It only says that it is difficult to obtain passwords.
- The intruder can always use a trial and error method. A test of only a limited set of potential strings tends to reveal most passwords because there is a strong tendency for people to choose relatively short and simple passwords that they can remember.
- Some techniques that may be used to make the task of guessing a password difficult are as follows
  1. Longer passwords.
  2. Salting the password table.
  3. System assistance in password selection.
- The length of a password determines the ease with which a password can be found by exhaustion. For example, 3-digit password provides 1000 variations whereas a four digit passwords provides 10,000 variations. Second method is the system assistance.
- Password is generated by system or user.

### 2. Encrypted passwords

- Instead of storing the names and passwords in plain text form, they are encrypted and stored in cipher text form in the table.
- In this case, instead of directly using a user specified name and password for table lookup, they are first encrypted and then the results are used for table lookup.
- If the stored encoded password is read by someone, the password cannot be determined.

### 3. One time passwords

- Password sniffing problem is solved by password pair(login name and password). When a session starts, system selects password pair randomly and user supply remaining part of password pair.
- Here use is challenged for password, he will respond with correct answer.

### 8.3.2 Access Control Lists

- **Access control list (ACL)** is a modification of the access control matrix. It maintains a record of only those entries that specify an access rights. It was used in MULTICS operating system.
- Each object has a security attribute that identifies its access control list. For each object in an operating system, an access control list contains entries for each subject and the privileges associated with that subjects for the object.
- When subject send request for accessing an object, the operating system searches the access control list for that object to find any privileges for the subject.

### 8.3.3 Implementation of Access Matrix

- Access control matrix is implemented by using four different methods :
  a) Global table
  b) Access lists for objects
  c) Capability list
  d) Lock key mechanism.

**1. Global table**

- Global table consists of three parameters i.e. <domain, object, rights-set>. Before performing operation on an object by subject in particular domain, the global table is searched for a triples <domain, object, rights-set>.
- If this is found, the operation is performed otherwise error condition is generated.

Drawbacks :

a) Global table cannot keep in main memory because of its large size.

b) It requires additional I/O operation.

**2. Capability lists**

- Capability list is used for controlling device and files. An object is often represented by its physical name or address called **a capability.**
- Capability is a pointer or token that grants privileges to a subject that possesses. It is implemented as a unique object identifier. Capabilities are granted to a subject, which presents the token for all subsequent accesses to the object.
- Each capability grants the owner certain rights on a certain object. Capability lists are themselves objects and may be pointed to from other capability lists.
- Token based and pointer based capabilities are used in the system.
- Pointer based capabilities : It simplify to the address at which the object is stored.
- Token based capabilities : Capabilities are independent for the objects location in memory.
- Upon creation of an object, a capability for that object is created.
- Capability provides two functions :
  1. It provides very large address space for a resource.
  2. Possession of the capability represents the authorization of the subject to the particular object.

**Implementation of capabilities :**

1. It is implemented within the operating system's address space.
2. Hardware may incorporate specialized support for capabilities.

- Mach operating system provides kernel level capabilities.
- Capabilities allow a process to be encapsulated easily. It do not support selective revocation of rights.

**3. Lock key mechanism**

- **Lock** is a unique bit pattern for each object.
- **Key :** each domain has a list of unique bit patterns.

- When a block of memory is allocated to a process, the lock is set to the same as the process's key. Process key and memory locks can be set only while the system is in supervisor mode.
- The access is checked by hardware on each memory access by keeping the key value in a CPU register. So memory locks can be through of as access control list.
- The value of lock and key is same then only accessing of object is allowed.
- This method is simple to implement. It is efficient at runtime.

## University Questions

1. *Write short note on following :*
   *Access metrics mechanism.* **GTU : Winter-13, Marks 4**
2. *How access control list can be useful for managing file access ?* **GTU : Summer-13, Marks 4**

❑❑❑

classmate
Date
Page
Harsh Porwal

# Ch-9

# UNIX / LINUX OS

(1) Explain linux kernel and its function.

→ The main purpose of a computer is to run a predefined sequence of instruction known as a program. A program under execution is reffered as process.

→ Now, most special purpose computers are meant to run a single process but in sophisticated system such as genral purpose computer, are intended to run many process simultaneously.

→ If computer running many process simultaneously we need a middle layer to manage the distribution of the hardware resources of computer efficienty and fairley among all the various processes running on the computer.
This middle layer referred as kernel.

→ Basically kernel virtualizes the common hardware resources of the computer to provide each process with its own virtual resources.

→ This makes the process seem as it is the sole process running on the machine.

→ The kernel is also responsible preventing conflict between different processes.

→ Coresubsystem of LINUX kernel

(1) The Process scheduler
(2) The Memory Managment Unit
(3) The Virtual File System
(4) The Networking Unit
(5) Inter-process Communication unit.

→ The Process scheduler : This kernel subsystem is responsible for fairly distributing the CPU time among all the processes running on the system simultaneously.

→ The Memory Managment Unit : This kernel sub unit is responsible for proper distribution of the memory resources among the various processes running in the System. The MMU does more than just simply provide seprate virtual address space for each of the processes.

→ The Virtual File System : This subsystem is responsible for providing a unified interface to across stored data across different file system and physical storage media.

Harsh Porwal

(2) Write Shell script to find factorial of given number.

→ Using for loop,

```
# shell script for factorial of number

echo "Enter a number"

read num

fact = 1

for ((i=2 ; i<=num ; i++))
{
  fact = $((fact * i))
}
echo fact
```

→ Output :

```
Enter a number
 5
 120
```

## (3) Components of LINUX System.

→ Linux OS consists of three components.
(1) Kernel (2) System library
(3) System Utility

→ Kernel : Kernel is a heart of Linux OS.
It is responsible for all major activities of Linux OS.
It provides important abstraction to hide low level hardware information to user or application program.
Kernel is combination of various modules and it interacts directly with the underlying hardware.
The Linux kernel consists of servel important paris : Process managment, memory managment, network managment etc.

→ System Library : It is special functions or programs. Libraries are used for accessing kernel's features.
This library implements most of the functionalities of the OS and do not requires kernel modules code access rights.

→ System Utility : These programs are responsible to do specialized, individual level task.
Privileged mode is called kernel mode in Linux system. All kernel code executes in priviledged mode with full access to the hardware resources of the computer.

Harsh Porwal

(4) What is Marshalling and Unmarshalling?

→ Marshalling is the process of gathering data from one or more applications or non-contiguous sources in computer storage, putting the data pices into a message buffer, and organize the data into format that is prescribed for a particular reciver or programming interface.

→ Unmarshalling is just the reverse lother side of process taking a representation of the object created by marshalling and using it reconstitute an object instance with the JVM.

(5) What is Innode?

→ An innode is a control structure that contains the key information needed by the OS for a particular file. Indoes exist in a static form on disk, and the kernel reads them into an in-core inode manipulate them.

→ Serval file names may be associated with a single inode, but an active inode is associated with exactly one file and each file is controlled by exactly one inode.

Date
Page

→ Disk inode consist following field

(1) File owner identifire : Owner area of two types. Individual owner and group owner. The superuser has access rights to all files in the system.

(2) File Type : File types are regular, directory, character or block special or pipes.

(3) File access permission : The system protects file according to classes.
(a) owner (b) Group owner (c) other user.
Each class has access right to read, write and execute the file which can be set individualy.

(4) File acess time : It give the time of last modified.

(5) Number of links to the file : Representating numbers of names the file has in the directory heirarchy.

(6) File size : Data in file is addressablye by the number of bytes from the beginning of the file, starting from byte offset 0, and the file size 1 is greater than the highest byte offset of data in file.

(7) Table of contents for the disk addresses of data in a file.

Harsh Porwal

(6) Explain following Linux commands.
mkdir, touch, cat, rm, sort, grep, cut, chmod, finger, man.

→ mkdir : It is used to create a new directory under any directory.

Syntax : mkdir <directory name>

→ toch : It is used to create empty files. we can create multiple empty file by executing it once.

Syntax : toch <file name>
touch <file 1> <file 2> ....

→ Cat : It is used to creat a file, display content of file, copy content of one file to another file and more. This command is mulipurpose utility in Linux system.

Syntax : cat [OPTION]... [FILE]...

execute → cat > <file name>
Save → "CTRL + D"
Display → cat <file name>

→ rm : It is used to remove a file.

Syntax : rm <file name>

→ **Sort** : It is used to ~~store~~ sort files in alphabatical order.

Syntax : Sort < file name >

→ **grep** : It is most powerful and used filter in Linux system. It stands for 'global regular expression print' It is useful for searching the content from a file. Genrally it is used with pipe.

syntax : command | grep < search word >

→ **Cut** : It is used to select a specific column of file. The '-d' option is used to delimiter and '-f' option is used to specify a column number.

Syntax : cut -d (delimiter) -f (column number) < file name >

→ **Chmod** : It is used to change the access mode of file.

Syntax : chmod [reference] [operator] [mode] file...

Harsh Porwal

→ **finger** : It is command of Unix-like OS, the finger command looks up and display information about system user.

Syntax : finger [-lmsp] [user ...] [user@host]

→ **man** : It is used to display the user manual of any command that we can run on the terminal.

Syntax : $man [OPTION] .... [COMMAND NAME]..

# Physical vCenter vs. Virtual vCenter

| Virtual Machine | Physical Machine |
|---|---|
| • Can use VMware HA Cluster for high availability. | • Need third party clustering solutions for redundancy. |
| • Virtual RAM & vCPU's can be scaled up or down easily. | • Resizing of resources is a cumbersome process. |
| • Load Balanced using DRS | • No load balancing technique. |
| • Easy to backup & Restore | • Primitive backup methodology |
| • Use Snapshots for testing & upgrade management. | • Snapshots cannot be used on a Physical Server. |
| • Increased consolidation ratio and an example for other application owners. | • Another physical server added which beats the purpose of Virtualization. |

# OS Virtualization

With the help of OS virtualization nothing is pre-installed or permanently loaded on the local device and no-hard disk is needed. Everything runs from the network using a kind of virtual disk. This virtual disk is actually a disk image file stored on a remote server, SAN (Storage Area Network) or NAS (Non-volatile Attached Storage). The client will be connected by the network to this virtual disk and will boot with the Operating System installed on the virtual disk.

## How does OS Virtualization works?

Components needed for using OS Virtualization in the infrastructure are given below:

The first component is the OS Virtualization server. This server is the center point in the OS Virtualization infrastructure. The server manages the streaming of the information on the virtual disks for the client and also determines which client will be connected to which virtual disk (using a database, this information is stored). Also the server can host the storage for the virtual disk locally or the server is connected to the virtual disks via a SAN (Storage Area Network). In high availability environments there can be more OS Virtualization servers to create no redundancy and load balancing. The server also ensures that the client will be unique within the infrastructure.

Secondly, there is a client which will contact the server to get connected to the virtual disk and asks for components stored on the virtual disk for running the operating system.

The available supporting components are database for storing the configuration and settings for the server, a streaming service for the virtual disk content, a (optional) TFTP service and a (also optional) PXE boot service for connecting the client to the OS Virtualization servers.

As it is already mentioned that the virtual disk contains an image of a physical disk from the system that will reflect to the configuration and the settings of those systems which will be using the virtual disk. When the virtual disk is created then that disk needs to be assigned to the client that will be using this disk for starting. The connection between the client and the disk is made through the administrative tool and saved within the database. When a client has a assigned disk, the machine can be started with the virtual disk using the following process as displayed in the given below Figure:

### 1) Connecting to the OS Virtualization server:

First we start the machine and set up the connection with the OS Virtualization server. Most of the products offer several possible methods to connect with the server. One of the most popular and used methods is using a PXE service, but also a boot strap is used a lot (because of the disadvantages of the PXE service). Although each method initializes the network interface card (NIC), receiving a (DHCP-based) IP address and a connection to the server.

### 2) Connecting the Virtual Disk:

When the connection is established between the client and the server, the server will look into its database for checking the client is known or unknown and which virtual disk is assigned to the client. When more than one virtual disk are connected then a boot menu will be displayed on the client side. If only one disk is assigned, that disk will be connected to the client which is mentioned in step number 3.

### 3) VDisk connected to the client:

After the desired virtual disk is selected by the client, that virtual disk is connected through the OS Virtualization server . At the back-end, the OS Virtualization server makes sure that the client will be unique (for example computer name and identifier) within the infrastructure.

### 4) OS is "streamed" to the client:

As soon the disk is connected the server starts streaming the content of the virtual disk. The software knows which parts are necessary for

identifier) within the infrastructure.

### 4) OS is "streamed" to the client:

As soon the disk is connected the server starts streaming the content of the virtual disk. The software knows which parts are necessary for starting the operating system smoothly, so that these parts are streamed first. The information streamed in the system should be stored somewhere (i.e. cached). Most products offer several ways to cache that information. For examples on the client hard disk or on the disk of the OS Virtualization server.

### 5) Additional Streaming:

After that the first part is streamed then the operating system will start to run as expected. Additional virtual disk data will be streamed when required for running or starting a function called by the user (for example starting an application available within the virtual disk).