MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

L 1

LECTURE HANDOUTS

AI&DS

II\III

Course Name with Code : **19ADC07&OPERATING SYSTEMS**

Course Faculty : **R.NIVETHITHA**

Unit : **I OPERATING SYSTEMS OVERVIEW**

Date of Lecture:

Topic of Lecture: Operating system Functions, Operating system Structure

Introduction :

A program that acts as an intermediary between a user of a computer and the computer hardware

Operating system goals:

- Execute user programs and make solving user problems easier
- Make the computer system convenient to use
- Use the computer hardware in an efficient manner

Prerequisite knowledge for Complete understanding and learning of Topic:

- Operating Systems
- Hardware
- Software
- Process

Detailed content of the Lecture:

OPERATING SYSTEM:

A program that acts as an intermediary between a user of a computer and the computer hardware

Operating system goals:

- Execute user programs and make solving user problems easier
- Make the computer system convenient to use
- Use the computer hardware in an efficient manner

An operating system is a construct that allows the user application programs to interact with the system hardware. Since the operating system is such a complex structure, it should be created with utmost care so it can be used and modified easily. An easy way to do this is to create the operating system in parts. Each of these parts should be well defined with clear inputs, outputs and functions.

FUNCTIONS OF AN OPERATING SYSTEM:

1. **Security**
   The operating system uses password protection to protect user data and similar other techniques. it also prevents unauthorized access to programs and user data.

2. **Control over system performance**
   Monitors overall system health to help improve performance. records the response time between service requests and system response to have a complete view of the system health. This can help improve performance by providing important information needed to troubleshoot problems.

3. **Job accounting**
   Operating system Keeps track of time and resources used by various tasks and users, this information can be used to track resource usage for a particular user or group of user.

4. **Error detecting aids**
   Operating system constantly monitors the system to detect errors and avoid the malfunctioning of computer system.

5. **Coordination between other software and users**
   Operating systems also coordinate and assign interpreters, compilers, assemblers and other software to the various users of the computer systems.

6. **Memory Management**
   The operating system manages the Primary Memory or Main Memory. Main memory is made up of a large array of bytes or words where each byte or word is assigned a certain address. Main memory is a fast storage and it can be accessed directly by the CPU. For a program to be executed, it should be first loaded in the main memory. An Operating System performs the following activities for memory management:

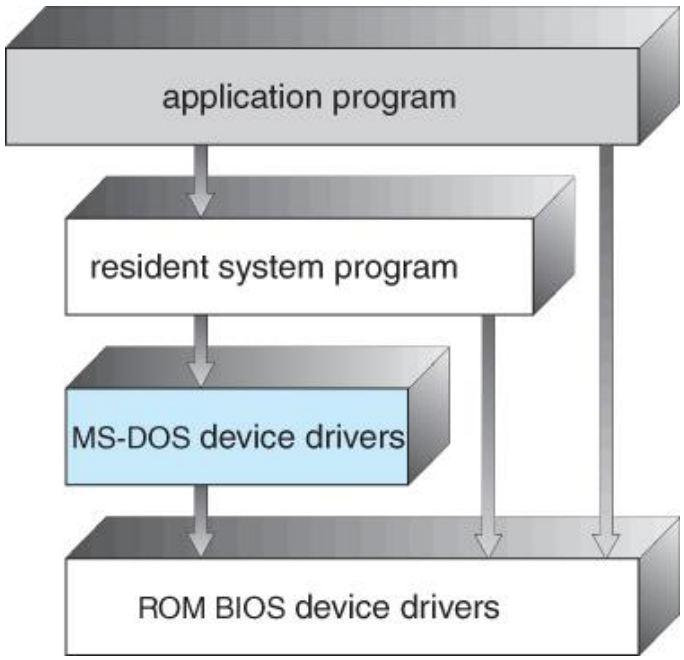
7. **Processor Management**
   In a multi programming environment, the OS decides the order in which processes have access to the processor, and how much processing time each process has. This function of OS is called process scheduling.

8. **Device Management**
   An OS manages device communication via their respective drivers. It performs the following activities for device management. Keeps tracks of all devices connected to system. designates a program responsible for every device known as the Input/Output controller.

9. **File Management**
   A file system is organized into directories for efficient or easy navigation and usage. These directories may contain other directories and other files. An Operating System carries out the following file management activities

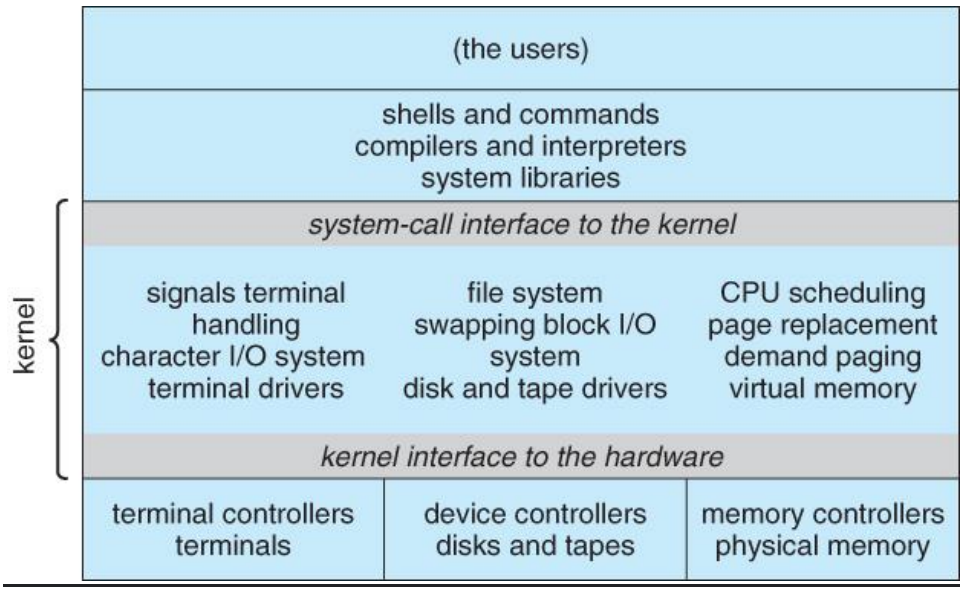
SIMPLE STRUCTURE:

When DOS was originally written its developers had no idea how big and important it would eventually become. It was written by a few programmers in a relatively short amount of time, without the benefit of modern software engineering techniques, and then gradually grew over time to exceed its original expectations. It does not break the system into subsystems, and has no distinction between user and kernel modes, allowing all programs direct access to the underlying hardware. (Note that user versus kernel mode was not supported by the 8088 chip set anyway, so that really wasn't an option back then.)



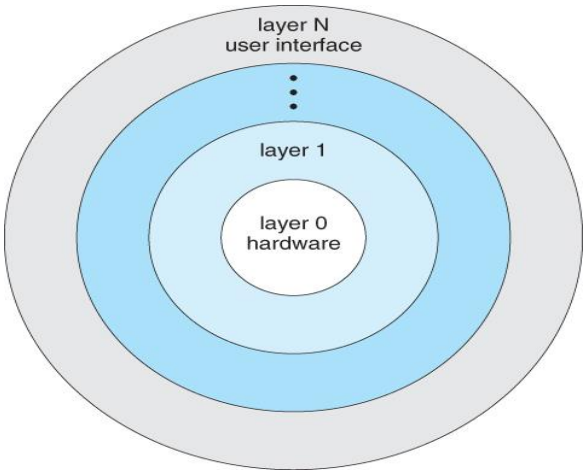
MS-DOS layer structure

The original UNIX OS used a simple layered approach, but almost all the OS was in one big layer, not really breaking the OS down into layered subsystems:



LAYERED APPROACH:

- Another approach is to break the OS into a number of smaller layers, each of which rests on the layer below it, and relies solely on the services provided by the next lower layer.
- This approach allows each layer to be developed and debugged independently, with the assumption that all lower layers have already been debugged and are trusted to deliver proper services.
- The problem is deciding what order in which to place the layers, as no layer can call upon the services of any higher layer, and so many chicken-and-egg situations may arise.
- Layered approaches can also be less efficient, as a request for service from a higher layer has to filter through all lower layers before it reaches the HW, possibly with significant processing at each step.



LAYERED OPERATING SYSTEM

Video Content / Details of website for further learning (if any):
http://www.cs.nthu.edu.tw/~ychung/slides/CSC3150/Abraham-Silberschatz-Operating-System-Concepts---9th2012.12

Important Books/Journals for further learning including the page nos.:
Book: Abraham Silberschatz,Peter Bear Galvin and Greg Gagne, ," Operating system concepts",2015, John Wiley & Sons (ASIA) ,9th Edition,Page no(351-466**).**

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

L 2

LECTURE HANDOUTS

AI&DS

II\III

Course Name with Code : **19ADC07&OPERATING SYSTEMS**

Course Faculty : **R.NIVETHITHA**

Unit : **I OPERATING SYSTEMS OVERVIEW**

Date of Lecture:

Topic of Lecture: Operating system Operations.

Introduction :

- A program that acts as an intermediary between a user of a computer and the computer hardware
- An operating system is a construct that allows the user application programs to interact with the system hardware
- The major operations of the operating system are process management, memory management, device management and file management

Prerequisite knowledge for Complete understanding and learning of Topic:

- Operating Systems
- Hardware
- Software
- Process

Detailed content of the Lecture:

OPERATING SYSTEM OPERATIONS:

An operating system is a construct that allows the user application programs to interact with the system hardware. Operating system by itself does not provide any function but it provides an atmosphere in which different applications and programs can do useful work.

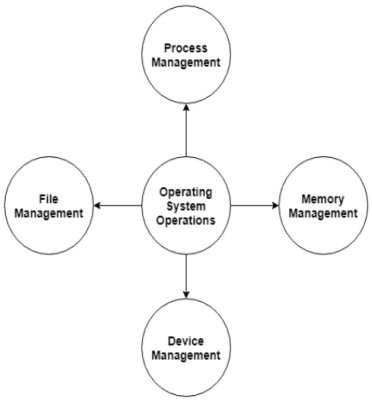
The major operations of the operating system are process management, memory management, device management and file management. These are given in detail as follows:

PROCESS MANAGEMENT:

The operating system is responsible for managing the processes i.e assigning the processor to a process at a time. This is known as process scheduling. The different algorithms used for process scheduling are FCFS (first come first served), SJF (shortest job first), priority scheduling, round robin scheduling etc.

There are many scheduling queues that are used to handle processes in process management. When the processes enter the system, they are put into the job queue. The processes that are ready to execute in the main memory are kept in the ready queue. The processes that are waiting for the I/O

device are kept in the device queue.



OPERATING SYSTEM OPERATIONS

MEMORY MANAGEMENT:

Memory management plays an important part in operating system. It deals with memory and the moving of processes from disk to primary memory for execution and back again.

The activities performed by the operating system for memory management are

- The operating system assigns memory to the processes as required. This can be done using best fit, first fit and worst fit algorithms.
- All the memory is tracked by the operating system i.e. it nodes what memory parts are in use by the processes and which are empty.

DEVICE MANAGEMENT:

There are many I/O devices handled by the operating system such as mouse, keyboard, disk drive etc. There are different device drivers that can be connected to the operating system to handle a specific device. The device controller is an interface between the device and the device driver. The user applications can access all the devices using the device drivers, which are device specific codes.

FILE MANAGEMENT:

Files are used to provide a uniform view of data storage by the operating system. All the files are mapped onto physical devices that are usually non volatile so data is safe in the case of system failure.

The files can be accessed by the system in two ways i.e. sequential access and direct access

- **Sequential Access**

  The information in a file is processed in order using sequential access. The files records are accessed on after another. Most of the file systems such as editors, compilers etc.

- **Direct Access**

  In direct access or relative access, the files can be accessed in random for read and write

operations. The direct access model is based on the disk model of a file, since it allows random accesses.

Modern operating systems are interupt driven.

- If there are no processes to execute,
- no I/O devices to service, and no users to whom to respond,

An operating system will sit quietly, waiting for something to happen. Events are almost always signaled by the occurrence of an **interrupt** or a **trap**.

DUAL-MODE OPERATION:

we need two separate modes of operation: **user mode** and **kernel mode** (also called **supervisor mode**, **system mode**, or **privileged mode**). A bit, called the **mode bit**, is added to the hardware of the computer to indicate the current mode: kernel (0) or user (1). When the computer system is executing on behalf of a user application, **the system is in user mode**.

- However, when a user application requests a service from the operating system (via a system call), it must **transition from user to kernel mode** to fulfill the request. As we shall see, this architectural enhancement is useful for many other aspects of system operation as well.
- At system boot time, the hardware starts in **kernel mode.**
- The operating system is then loaded and starts user applications in **user mode**.
- Whenever a trap or interrupt occurs, the hardware switches from **user mode to kernel mode** (that is, changes the state of the mode bit to 0).
- Thus, whenever the operating system gains control of the computer, it is in **kernel mode**.
- The system always **switches to user mode** (by setting the mode bit to 1) before passing control to a user program.

Video Content / Details of website for further learning (if any):
http://www.cs.nthu.edu.tw/~ychung/slides/CSC3150/Abraham-Silberschatz-Operating-System-Concepts---9th2012.12
https://padakuu.com/article/57-operating-system-operations-dual-mode-operation-timer

Important Books/Journals for further learning including the page nos.:
Book: Abraham Silberschatz,Peter Bear Galvin and Greg Gagne, ," Operating system concepts",2015, John Wiley & Sons (ASIA) ,9th Edition,Page no(351-466**).**

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

L 3

LECTURE HANDOUTS

AI&DS

II\III

Course Name with Code : **19ADC07&OPERATING SYSTEMS**

Course Faculty : **R.NIVETHITHA**

Unit : **I OPERATING SYSTEMS OVERVIEW**

Date of Lecture:

Topic of Lecture: Protection and Security , Computing Environments.

Introduction :

- A program that acts as an intermediary between a user of a computer and the computer hardware
- A threat is a program that is malicious in nature and leads to harmful effects for the system. Some of the common threats that occur in a system
- These passwords provide a lot of security for authentication purposes
- There are many scheduling queues that are used to handle processes in process management

Prerequisite knowledge for Complete understanding and learning of Topic:

- Operating Systems
- Hardware
- Software
- Process

Detailed content of the Lecture:

PROTECTION AND SECURITY:

A threat is a program that is malicious in nature and leads to harmful effects for the system. Some of the common threats that occur in a system are −

Virus

Viruses are generally small snippets of code embedded in a system. They are very dangerous and can corrupt files, destroy data, crash systems etc. They can also spread further by replicating themselves as required.

Trojan Horse

A trojan horse can secretly access the login details of a system. Then a malicious user can use these to enter the system as a harmless being and wreak havoc.

Trap Door

A trap door is a security breach that may be present in a system without the knowledge of the users. It can be exploited to harm the data or files in a system by malicious people.

Worm

A worm can destroy a system by using its resources to extreme levels. It can generate multiple copies which claim all the resources and don't allow any other processes to access them. A worm can shut down a whole network in this way.

Denial of Service

These type of attacks do not allow the legitimate users to access a system. It overwhelms the system with requests so it is overwhelmed and cannot work properly for other user.

PROTECTION AND SECURITY METHODS:

The different methods that may provide protect and security for different computer systems are −

Authentication

This deals with identifying each user in the system and making sure they are who they claim to be. The operating system makes sure that all the users are authenticated before they access the system. The different ways to make sure that the users are authentic are:

- **Username/ Password**

  Each user has a distinct username and password combination and they need to enter it correctly before they can access the system.

- **User Key/ User Card**

  The users need to punch a card into the card slot or use they individual key on a keypad to access the system.

- **User Attribute Identification**

  Different user attribute identifications that can be used are fingerprint, eye retina etc. These are unique for each user and are compared with the existing samples in the database. The user can only access the system if there is a match.

One Time Password

These passwords provide a lot of security for authentication purposes. A one time password can be generated exclusively for a login every time a user wants to enter the system. It cannot be used more than once. The various ways a one time password can be implemented are

- **Random Numbers**

  The system can ask for numbers that correspond to alphabets that are pre arranged. This combination can be changed each time a login is required.

- **Secret Key**

  A hardware device can create a secret key related to the user id for login. This key can change each time.

COMPUTING ENVIRONMENTS :

A brief overview on how computer-system organization and major operating-system components are used in a variety of computing environments.

TRADITIONAL COMPUTING

As computing matures, the lines separating many of the traditional computing environments are blurring. Consider the "typical office environment." Just a few years ago, this environment consisted of PCs connected to a network, with servers providing file and print services. Remote access was awkward, and portability was achieved by use of laptop computers. Terminals attached to mainframes were prevalent at many companies as well, with even fewer remote access and portability options The current trend is toward providing more ways to access these computing environments. Web technologies are stretching the boundaries of traditional computing. Companies establish portals, which provide web accessibility to their internal servers.

Network computers are essentially terminals that understand web-based computing. Handheld computers can synchronize with PCs to allow very portable use of company information. Handheld PDAs can also connect to wireless networks to use the company's web portal (as well as the myriad other web resources). At home, most users had a single computer with a slow modem connection to the office, the Internet, or both.

CLIENT-SERVER COMPUTING

As PCs have become faster, more powerful, and cheaper, designers have shifted away from centralized system architecture. Terminals connected to centralized systems are now being supplanted by PCs. Correspondingly, userinterface functionality once handled directly by the centralized systems is increasingly being handled by the PCs. As a result, many of todays systems act as server systems to satisfy requests generated by client systems. This form of specialized distributed system, called client-server system,

PEER-TO-PEER COMPUTING

Another structure for a distributed system is the peer-to-peer (P2P) system model. In this model, clients and servers are not distinguished from one another; instead, all nodes within the system are considered peers, and each may act as either a client or a server, depending on whether it is requesting or providing a service.

1. When a node joins a network, it registers its service with a centralized lookup service on the network. Any node desiring a specific service first contacts this centralized lookup service to determine which node provides the service. The remainder of the communication takes place between the client and the service provider.
2. A peer acting as a client must first discover what node provides a desired service by

broadcasting a request for the service to all other nodes in the network. The node (or nodes) providing that service responds to the peer making the request.

WEB-BASED COMPUTING

The Web has become ubiquitous, leading to more access by a wider variety of devices than was dreamt of a few years ago. PCs are still the most prevalent access devices, with workstations, handheld PDAs, and even cell phones also providing access. Web computing has increased the emphasis on networking. Devices that were not previously networked now include wired or wireless access. Devices that were networked now have faster network connectivity, provided by either improved networking technology, optimized network implementation code, or both.

Video Content / Details of website for further learning (if any):

http://www.cs.nthu.edu.tw/~ychung/slides/CSC3150/Abraham-Silberschatz-Operating-System-Concepts---9th2012.12

https://www.tutorialspoint.com/Protection-and-Security-in-Operating-System

https://padakuu.com/article/98-computing-environments-traditional-computing-client-server-computing-peer-to-peer-computing-web-based-computing

Important Books/Journals for further learning including the page nos.:

Book: Abraham Silberschatz,Peter Bear Galvin and Greg Gagne, ," Operating system concepts",2015, J Wiley & Sons (ASIA) ,9th Edition,Page no(351-466**).**

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

L4

LECTURE HANDOUTS

AI&DS

II\III

Course Name with Code : **19ADC07&OPERATING SYSTEMS**

Course Faculty : **R.NIVETHITHA**

Unit : **I OPERATING SYSTEMS OVERVIEW**

Date of Lecture:

Topic of Lecture: Open source Operating system: Operating system services, user and Operating system Interface

Introduction :

- It provides programs an environment to execute.
- It provides users the services to execute the programs in a convenient manner.
- Operating system provides the access to the required I/O device when required.
- The OS takes an appropriate action to ensure correct and consistent computing

Prerequisite knowledge for Complete understanding and learning of Topic:

- Operating Systems
- Hardware
- Software
- Process

Detailed content of the Lecture:

OPERATING SYSTEM SERVICES:

An Operating System provides services to both the users and to the programs.

- It provides programs an environment to execute.
- It provides users the services to execute the programs in a convenient manner.

Following are a few common services provided by an operating system

- Program execution
- I/O operations
- File System manipulation
- Communication
- Error Detection
- Resource Allocation

- Protection

PROGRAM EXECUTION:

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

I/O OPERATIONS:

- I/O operation means read or write operation with any file or any specific I/O device.
- Operating system provides the access to the required I/O device when required.

FILE SYSTEM MANIPULATION:

- Program needs to read a file or write a file.
- The operating system gives the permission to the program for operation on file.
- Permission varies from read-only, read-write, denied and so on.
- Operating System provides an interface to the user to create/delete files.
- Operating System provides an interface to the user to create/delete directories.
- Operating System provides an interface to create the backup of file system.

COMMUNICATION:

- Two processes often require data to be transferred between them
- Both the processes can be on one computer or on different computers, but are connected through a computer network.
- Communication may be implemented by two methods, either by Shared Memory or by Message Passing.
- Numbers that correspond to alphabets that are pre arranged. This combination can be changed each time a login is required.

ERROR DETECTION:

- The OS constantly checks for possible errors.
- The OS takes an appropriate action to ensure correct and consistent computing.

RESOURCE ALLOCATION

- The OS manages all kinds of resources using schedulers.

- CPU scheduling algorithms are used for better utilization of CPU.

PROTECTION

- The OS ensures that all access to system resources is controlled.
- The OS ensures that external I/O devices are protected from invalid access attempts.
- The OS provides authentication features for each user by means of passwords.

USER OPERATING-SYSTEM INTERFACE

There are two fundamental approaches for users to interface with the operating system. One technique is to provide a command-line interface or command interpreter that allows users to directly enter commands that are to be performed by the operating system. The second approach allows the user to interface with the operating system via a graphical user interface or GUI.

COMMAND INTERPRETER

Some operating systems include the command interpreter in the kernel. Others, such as Windows XP and UNIX, treat the command interpreter as a special program that is running when a job is initiated or when a user first logs on (on interactive systems). On systems with multiple command interpreters to choose from, the interpreters are known as shells. For example, on UNIX and Linux systems, there are several different shells a user may choose from including the Bourne shell, C shell, Bourne-Again shell, the Korn shell, etc.

For example, a command to delete a file may cause the command interpreter to jump to a section of its code that sets up the parameters and makes the appropriate system call. In this case, the number of commands that can be given determines the size of the command interpreter, since each command requires its own implementing code. An alternative approach—used by UNIX, among other operating systems —implements most commands through system programs. In this case, the command interpreter does not understand the command in any way; it merely uses the command to identify a file to be loaded into memory and executed. Thus, the UNIX command to delete a file rm file.tx t would search for a file called rm, load the file into memory, and execute it with the parameter file . txt.

GRAPHICAL USER INTERFACES

A second strategy for interfacing with the operating system is through a userfriendly graphical user interface or GUI. Rather than having users directly enter commands via a command-line interface, a GUI allows provides a mouse-based window-and-menu system as an interface. A GUI provides a desktop metaphor where the mouse is moved to position its pointer on images, or

icons, on the screen (the desktop) that represent programs, files, directories, and system functions. Depending on the mouse pointer's location, clicking a button on the mouse can invoke a program, select a file or directory—known as a folder— or pull down a menu that contains commands. Graphical user interfaces first appeared due in part to research taking place in the early 1970s at Xerox PARC research facility. The first GUI appeared on the Xerox Alto computer in 1973.

However, there has been significant development in GUI designs from various opensource projects such as K Desktop Environment (or KDE) and the GNOME desktop by the GNU project. Both the KDE and GNOME desktops rim on Linux and various UNIX systems and are available under open-source licenses, which means their source code is in the public domain. The choice of whether to use a command-line or GUI interface is mostly one of personal preference. As a very general rule, many UNIX users prefer a command-line interface as they often provide powerful shell interfaces. Alternatively, most Windows users are pleased to use the Windows GUI environment and almost never use the MS-DOS shell interface. The various changes undergone by the Macintosh operating systems provides a nice study in contrast. Historically, Mac OS has not provided a command line interface, always requiring its users to interface with the operating system using its GUI.

Video Content / Details of website for further learning (if any):

http://www.cs.nthu.edu.tw/~ychung/slides/CSC3150/Abraham-Silberschatz-Operating-System-Concepts---9th2012.12

https://www.tutorialspoint.com/operating_system/os_services.htm

https://padakuu.com/article/84-user-os-interface-command-interpreter-and-graphical-user-interfaces

Important Books/Journals for further learning including the page nos.:

Book: Abraham Silberschatz,Peter Bear Galvin and Greg Gagne, ," Operating system concepts",2015, John Wiley & Sons (ASIA) ,9th Edition,Page no(351-466**).**

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

L5

LECTURE HANDOUTS

AI&DS

II\III

Course Name with Code : **19ADC07&OPERATING SYSTEMS**

Course Faculty : **R.NIVETHITHA**

Unit : **I OPERATING SYSTEMS OVERVIEW**

Date of Lecture:

Topic of Lecture: System Call, Types of System Call, System Program

Introduction :

- A system call is a mechanism that provides the interface between a process and the operating system
- Reading and writing from files demand system calls.
- If a file system wants to create or delete files, system calls are required.
- System calls are used for the creation and management of new processes

Prerequisite knowledge for Complete understanding and learning of Topic:

- Operating Systems
- Hardware
- Software
- Process

Detailed content of the Lecture:

SYSTEM CALL :

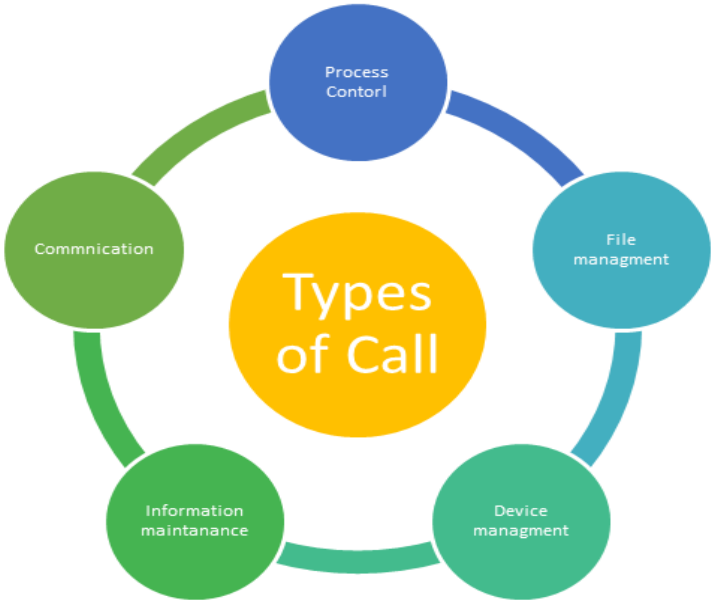
A **system call** is a mechanism that provides the interface between a process and the operating system. It is a programmatic method in which a computer program requests a service from the kernel of the OS. System call offers the services of the operating system to the user programs via API (Application Programming Interface). System calls are the only entry points for the kernel system.

TYPES OF SYSTEM CALLS

Here are the five types of System Calls in OS:

- Process Control
- File Management
- Device Management
- Information Maintenance

- Communications



Types of System calls in OS

Process Control

This system calls perform the task of process creation, process termination, etc.

Functions:

- End and Abort
- Load and Execute
- Create Process and Terminate Process
- Wait and Signal Event
- Allocate and free memory

File Management

File management system calls handle file manipulation jobs like creating a file, reading, and writing, etc.

Functions:

- Create a file
- Delete file
- Open and close file
- Read, write, and reposition
- Get and set file attributes

Device Management

Device management does the job of device manipulation like reading from device buffers, writing into device buffers, etc.

Functions:

- Request and release device
- Logically attach/ detach devices
- Get and Set device attributes

Information Maintenance

It handles information and its transfer between the OS and the user program.

Functions:

- Get or set time and date
- Get process and device attributes

Communication:

These types of system calls are specially used for interprocess communications.

Functions:

- Create, delete communications connections
- Send, receive message
- Help OS to transfer status information
- Attach or detach remote devices

- The OS ensures that all access to system resources is controlled.
- The OS ensures that external I/O devices are protected from invalid access attempts.
- The OS provides authentication features for each user by means of passwords.

SYSTEM PROGRAMMING :

It can be defined as the act of building Systems Software using System Programming Languages. According to Computer Hierarchy, one which comes at last is Hardware. Then it is Operating System, System Programs, and finally Application Programs. Program Development and Execution can be done conveniently in System Programs. Some of the System Programs are simply user interfaces, others are complex. It traditionally lies between the user interface and system calls. System Programs can be divided into these categories :

- **File Management**

  A file is a collection of specific information stored in the memory of a computer system. File management is defined as the process of manipulating files in the computer system, its management includes the process of creating, modifying and deleting files.

- **Status Information**
  Information like date, time amount of available memory, or disk space is asked by some

users. Others providing detailed performance, logging, and debugging information which is more complex. All this information is formatted and displayed on output devices or printed. Terminal or other output devices or files or a window of GUI is used for showing the output of programs.

- **File Modification**
  For modifying the contents of files we use this. For Files stored on disks or other storage devices, we used different types of editors. For searching contents of files or perform transformations of files we use special commands.

- **Programming-Language support**
  For common programming languages, we use Compilers, Assemblers, Debuggers, and interpreters which are already provided to users. It provides all support to users. We can run any programming language. All languages of importance are already provided.

- **Program Loading and Execution**
  When the program is ready after Assembling and compilation, it must be loaded into memory for execution. A loader is part of an operating system that is responsible for loading programs and libraries. It is one of the essential stages for starting a program. Loaders, relocatable loaders, linkage editors, and Overlay loaders are provided by the system.

- **Communications**
  Virtual connections among processes, users, and computer systems are provided by programs. Users can send messages to another user on their screen, User can send e-mail, browsing on web pages, remote login, the transformation of files from one user to another.

Video Content / Details of website for further learning (if any):
http://www.cs.nthu.edu.tw/~ychung/slides/CSC3150/Abraham-Silberschatz-Operating-System-Concepts---9th2012.12
https://www.guru99.com/system-call-operating-system.html
https://www.geeksforgeeks.org/system-programs-in-operating-system/

Important Books/Journals for further learning including the page nos.:
Book: Abraham Silberschatz,Peter Bear Galvin and Greg Gagne, ," Operating system concepts",2015, John Wiley & Sons (ASIA) ,9th Edition,Page no(351-466**).**

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

LECTURE HANDOUTS

L6

AI&DS

II\III

Course Name with Code : **19ADC07&OPERATING SYSTEMS**

Course Faculty : **R.NIVETHITHA**

Unit : **I OPERATING SYSTEMS OVERVIEW**

Date of Lecture:

Topic of Lecture: Operating System Structure, Operating System debugging, System Boot

Introduction :

- Debugging is the process of finding the problems in a computer system and solving them.
- The core dump files contain the memory address space of a process that terminates
- The trace listing record information about a program execution using logging unexpectedly
- Booting the system is done by loading the kernel into main memory, and starting its execution.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Operating Systems
- Hardware
- Software
- Process

Detailed content of the Lecture:

OPERATING SYSTEM DEBUGGING:

Debugging is the process of finding the problems in a computer system and solving them. There are many different ways in which operating systems perform debugging. Some of these are −

Log Files

The log files record all the events that occur in an operating system. This is done by writing all the messages into a log file. There are different types of log files. Some of these are given as follows −

Event Logs

These stores the records of all the events that occur in the execution of a system. This is done so that the activities of all the events can be understood to diagnose problems.

Transaction Logs

The transaction logs store the changes to the data so that the system can recover from crashes and other errors. These logs are readable by a human.

Message Logs

These logs store both the public and private messages between the users. They are mostly plain text

files, but in some cases they may be HTML files.

Crash Dump Files

In the event of a total system failure, the information about the state of the operating system is captured in crash dump files. There are three types of dump that can be captured when a system crashes. These are

Complete Memory Dump

The whole contents of the physical memory at the time of the system crash are captured in the complete memory dump. This is the default setting on the Windows Server System.

Kernel Memory Dump

Only the kernel mode read and write pages that are present in the main memory at the time of the system crash are stored in the kernel memory dump.

SYSTEM BOOT:

- Booting the system is done by loading the kernel into main memory, and starting its execution.
- The initial bootstrap program is found in the BIOS read-only memory.
- This program can run diagnostics, initialize all components of the system, loads and starts the Operating System loader. (Called **boot strapping**)
- The loader program loads and starts the operating system.

Video Content / Details of website for further learning :

http://www.cs.nthu.edu.tw/~ychung/slides/CSC3150/Abraham-Silberschatz-Operating-System-Concepts---9th2012.12

https://www.tutorialspoint.com/operating-system-debugging

http://faculty.salina.k-state.edu/tim/ossg/Introduction/boot.html

Important Books/Journals for further learning including the page nos.:

Book: Abraham Silberschatz,Peter Bear Galvin and Greg Gagne, ," Operating system concepts",2015, John Wiley & Sons (ASIA) ,9th Edition,Page no(351-466**).**

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

L7

LECTURE HANDOUTS

AI&DS

II\III

Course Name with Code : **19ADC07&OPERATING SYSTEMS**

Course Faculty : **R.NIVETHITHA**

Unit : **I OPERATING SYSTEMS OVERVIEW**

Date of Lecture:

Topic of Lecture: Processes: Process concept, process scheduling

Introduction :

- A process is basically a program in execution. The execution of a process must progress in a sequential fashion.
- A process is defined as an entity which represents the basic unit of work to be implemented in the system.
- The processes which are blocked due to unavailability of an I/O device constitute this queue.

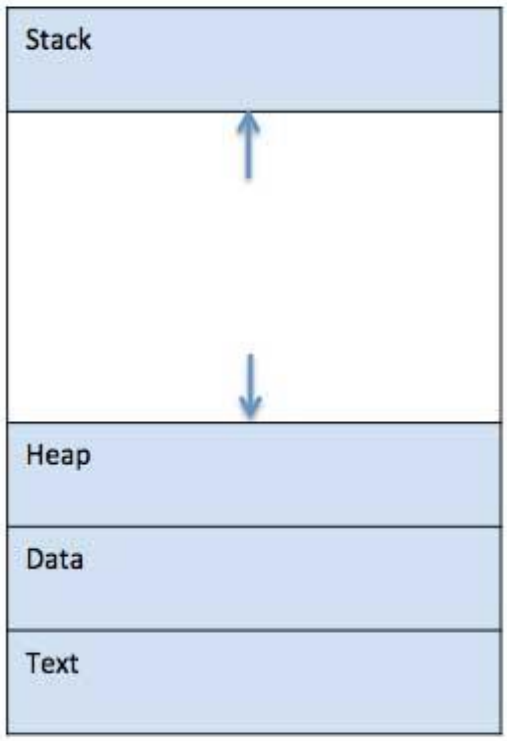
Prerequisite knowledge for Complete understanding and learning of Topic:

- Operating Systems
- Hardware
- Software
- Process

Detailed content of the Lecture:

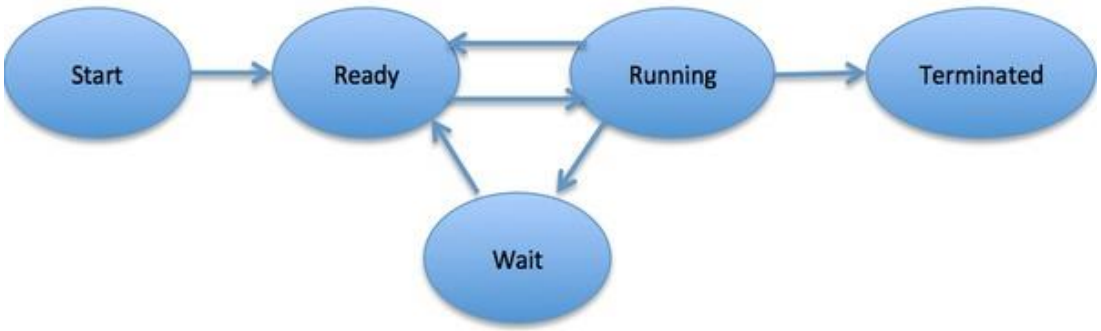
PROCESS:

- A process is basically a program in execution. The execution of a process must progress in a sequential fashion.
- A process is defined as an entity which represents the basic unit of work to be implemented in the system.
- To put it in simple terms, we write our computer programs in a text file and when we execute this program, it becomes a process which performs all the tasks mentioned in the program.
- When a program is loaded into the memory and it becomes a process, it can be divided into four sections stack, heap, text and data.
- The following image shows a simplified layout of a process inside main memory



PROCESS CONTROL BLOCK (PCB)

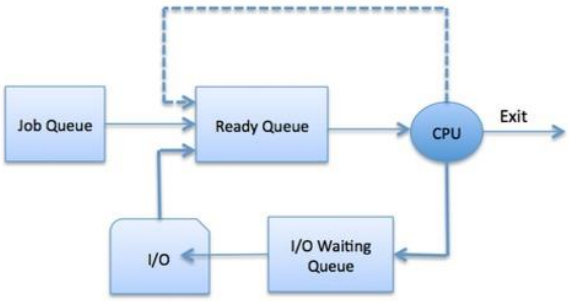
A Process Control Block is a data structure maintained by the Operating System for every process.The PCB is identified by an integer process ID (PID).



PROCESS SCHEDULING:

The Operating System maintains the following important process scheduling queues

- **Job queue** − This queue keeps all the processes in the system.
- **Ready queue** − This queue keeps a set of all processes residing in main memory, ready and waiting to execute. A new process is always put in this queue.
- **Device queues** − The processes which are blocked due to unavailability of an I/O device constitute this queue.



SCHEDULERS

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

LONG TERM SCHEDULER

It is also called a **job scheduler**. A long-term scheduler determines which programs are admitted to the system for processing. It selects processes from the queue and loads them into memory for execution. Process loads into the memory for CPU scheduling.

SHORT TERM SCHEDULER

It is also called as **CPU scheduler**. Its main objective is to increase system performance in accordance with the chosen set of criteria. It is the change of ready state to running state of the process.

MEDIUM TERM SCHEDULER

Medium-term scheduling is a part of **swapping**. It removes the processes from the memory. It reduces the degree of multiprogramming. The medium-term scheduler is in-charge of handling the swapped out-processes.

Video Content / Details of website for further learning :

http://www.cs.nthu.edu.tw/~ychung/slides/CSC3150/Abraham-Silberschatz-Operating-System-Concepts---9th2012.12

https://www.tutorialspoint.com/operating_system/os_processes.htm

Important Books/Journals for further learning including the page nos.:

Book: Abraham Silberschatz,Peter Bear Galvin and Greg Gagne, ," Operating system concepts",2015, John Wiley & Sons (ASIA) ,9th Edition,Page no(351-466**).**

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

L8

LECTURE HANDOUTS

AI&DS

II\III

Course Name with Code : **19ADC07&OPERATING SYSTEMS**

Course Faculty : **R.NIVETHITHA**

Unit : **I OPERATING SYSTEMS OVERVIEW**

Date of Lecture:

Topic of Lecture: Operations on Processes, Inter Process Communication

Introduction :

- A process is basically a program in execution. The execution of a process must progress in a sequential fashion.
- A process is defined as an entity which represents the basic unit of work to be implemented in the system.
- The execution of a process is a complex activity. It involves various operations. Following are the operations that are performed.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Operating Systems
- Hardware
- Software
- Process

Detailed content of the Lecture:

OPERATION ON A PROCESS:

The execution of a process is a complex activity. It involves various operations. Following are the operations that are performed while execution of a process:

1.Creation:

This the initial step of process execution activity. Process creation means the construction of a new process for the execution. This might be performed by system, user or old process itself. There are several events that leads to the process creation. Some of the such events are following:

- When we start the computer, system creates several background processes.
- A user may request to create a new process.
- A process can create a new process itself while executing.
- Batch system takes initiation of a batch job.

2. Scheduling/Dispatching:

The event or activity in which the state of the process is changed from ready to running. It means the operating system puts the process from ready state into the running state. Dispatching is done by operating system when the resources are free or the process has higher priority than the ongoing process. There are various other cases in which the process in running state is preempted and process in ready state is dispatched by the operating system.

3. Blocking:

When a process invokes an input-output system call that blocks the process and operating system put in block mode. Block mode is basically a mode where process waits for input-output. Hence on the demand of process itself, operating system blocks the process and dispatches another process to the processor. Hence, in process blocking operation, the operating system puts the process in 'waiting' state.

4. Preemption:

When a timeout occurs that means the process hadn't been terminated in the allotted time interval and next process is ready to execute, then the operating system preempts the process. This operation is only valid where CPU scheduling supports preemption. Basically this happens in priority scheduling where on the incoming of high priority process the ongoing process is preempted. Hence, in process preemption operation, the operating system puts the process in 'ready' state.

5. Termination:

Process termination is the activity of ending the process. In other words, process termination is the relaxation of computer resources taken by the process for the execution. Like creation, in termination also there may be several events that may lead to the process termination. Some of them are:

- Process completes its execution fully and it indicates to the OS that it has finished.
- Operating system itself terminates the process due to service errors.
- There may be problem in hardware that terminates the process.
- One process can be terminated by another process.

INTER PROCESS COMMUNICATION (IPC)

A process can be of two types:

- Independent process.
- Co-operating process.

An independent process is not affected by the execution of other processes while a co-operating process can be affected by other executing processes. Though one can think that those processes, which are running independently, will execute very efficiently, in reality, there are many situations when co-operative nature can be utilized for increasing computational speed, convenience, and modularity. Inter-process communication (IPC) is a mechanism that allows processes to communicate with each other and synchronize their actions. The communication between these processes can be seen as a method of co-operation between them. Processes can communicate with each other through both:

1. Shared Memory
2. Message passing

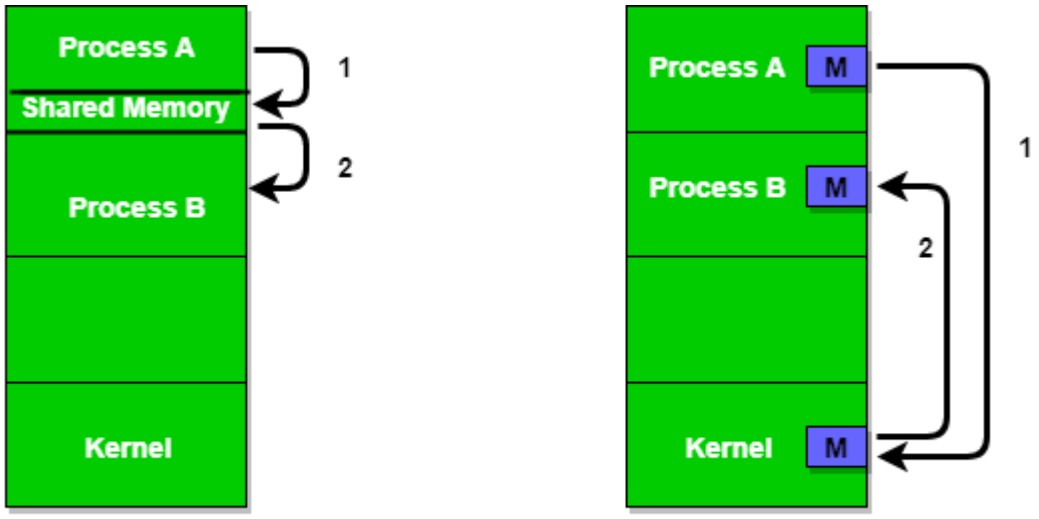


Figure 1 - Shared Memory and Message Passing

Video Content / Details of website for further learning :

http://www.cs.nthu.edu.tw/~ychung/slides/CSC3150/Abraham-Silberschatz-Operating-System-Concepts---9th2012.12

https://www.geeksforgeeks.org/operations-on-processes/

https://www.geeksforgeeks.org/inter-process-communication-ipc/

Important Books/Journals for further learning including the page nos.:

Book: Abraham Silberschatz,Peter Bear Galvin and Greg Gagne, ," Operating system concepts",2015, John Wiley & Sons (ASIA) ,9th Edition,Page no(351-466**).**

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

L9

LECTURE HANDOUTS

AI&DS

II\III

Course Name with Code : **19ADC07&OPERATING SYSTEMS**

Course Faculty : **R.NIVETHITHA**

Unit : **I OPERATING SYSTEMS OVERVIEW**

Date of Lecture:

Topic of Lecture: Examples of Inter Process Communication

Introduction :

- Inter-process communication (IPC) is a mechanism that allows processes to communicate with each other and synchronize their actions.
- The communication between these processes can be seen as a method of co-operation between them
- Communication between processes using shared memory requires processes to share some variable

Prerequisite knowledge for Complete understanding and learning of Topic:

- Operating Systems
- Hardware
- Software
- Process

Detailed content of the Lecture:

EXAMPLES OF INTER PROCESS COMMUNICATION:

- Posix : uses shared memory method.
- Mach : uses message passing.
- Windows XP : uses message passing using local procedural calls.

POSIX:

POSIX (Portable Operating System Interface) is a set of standard operating system interfaces based on the Unix operating system. The need for standardization arose because enterprises using computers wanted to be able to develop programs that could be moved among different manufacturer's computer systems without having to be recoded. Unix was selected as the basis for a standard system interface partly because it was "manufacturer-neutral." However, several major versions of Unix existed so there was a need to develop a common denominator system.

The Portable Operating System Interface evolved from the requirement of different organizations to have applications and programs that could be moved to different computer systems

without the need for recoding. The UNIX operating system was selected as the standard system interface as it was capable of being "manufacturer neutral." Although there have been many releases for portable operating system interface, POSIX 1 which defines system calls and POSIX 2 which defines command line interfaces are important ones.

The Portable Operating System Interface is recognized by standards such as the International Organization for Standardization (ISO) and American National Standards Institute (ANSI). One of the biggest roles of POSIX is to ensure code portability and compatibility between the different systems and hardware platforms. This is increasingly checked for many government agreements and commercial applications. Modern commercial implementations based on UNIX and even free ones based on UNIX are mostly POSIX compliant.

MACH:

The Mach operating system traces its ancestry to the Accent operating system developed at Carnegie Mellon University (CMU) (Rashid and Robertson [1981]). Mach's communication system and philosophy are derived from Accent, but many other significant portions of the system (for example, the virtual memory system, task and thread management) were developed from scratch (Rashid [1986], Tevanian et al. [1989], and Accetta et al. [1986]). The Mach scheduler was described in detail by Tevanian et al. [1987a] and Black [1990].

An early version of the Mach shared memory and memory-mapping system was presented by Tevanian et al. [1987b]. The Mach operating system was designed with the following three critical goals in mind:

- Emulate 4.3BSD UNIX so that the executable files from a UNIX system can run correctly under Mach.
- Be a modern operating system that supports many memory models, as well as parallel and distributed computing.
- Have a kernel that is simpler and easier to modify than is 4.3BSD. Mach's development followed an evolutionary path from BSD UNIX systems. Mach code was initially developed inside the 4.2BSD kernel, with BSD kernel components replaced by Mach components as the Mach components were completed.

The BSD components were updated to 4.3BSD when that became available. By 1986, the virtual memory and communication subsystems were running on the DEC VAX computer family, including multiprocessor versions of the VAX. Versions for the IBM RT/PC and for SUN 3 workstations followed shortly. Then, 1987 saw the completion of the Encore Multimax and Sequent Balance multiprocessor versions, including task and thread support, as well as the first official releases of the system, Release 0 and Release

WINDOWS XP :

Windows XP is an operating system (OS) developed and exclusively distributed by

Microsoft Corporation and targeted to owners of personal computers, laptops and media centers. The "XP" stands for eXPerience. Windows XP was released to manufacturers in August 2001 and publicly released in October 2001. Because of its installed user base, it is the second most popular Windows version.

During the early 2000s, Windows XP was Microsoft's most important OS release since Windows 95. Built on the improved stability of the Windows 2000 kernel, Windows XP offers a number of Windows system upgrades, including visual user interface quality and multiple features for streamlining multimedia, connectivity, and device management.

The three main Windows XP versions are (were) as follows:

- Home Edition: For basic home users
- Professional Edition: For power users and professionals requiring more advanced features
- Media Center Edition: Not released to retail, this version was made exclusively available to computer manufacturers for installation on desktops/laptops marketed as media center PCs.

Video Content / Details of website for further learning :

http://www.cs.nthu.edu.tw/~ychung/slides/CSC3150/Abraham-Silberschatz-Operating-System-Concepts---9th2012.12

https://padakuu.com/article/161-the-mach-operating-system

Important Books/Journals for further learning including the page nos.:

Book: Abraham Silberschatz,Peter Bear Galvin and Greg Gagne, ," Operating system concepts",2015, John Wiley & Sons (ASIA) ,9th Edition,Page no(351-466**).**

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

L10

LECTURE HANDOUTS

AI&DS

II/III

Course Name with Code : **19ADCO7&OPERATING SYSTEM**

Course Faculty : **R.NIVETHITHA.**

Unit : **II THREADS AND SCHEDULING ALGORITHM**

Date of Lecture:

Topic of Lecture: Multicore Programming, Multithreading Models.

Introduction :

- Multicore programming helps to create concurrent systems for deployment on multicore processor and multiprocessor systems
- Multicore and FPGA processing helps to increase the performance of an embedded system.
- The system can take advantage of increasing numbers of cores and FPGA processing power over time.

Prerequisite knowledge for Complete understanding and learning of Topic:

- **Operating Systems**
- Hardware
- Software
- Process

Detailed content of the Lecture:

MULTICORE PROGRAMMING:

Multicore programming helps to create concurrent systems for deployment on multicore processor and multiprocessor systems. A multicore processor system is basically a single processor with multiple execution cores in one chip. It has multiple processors on the motherboard or chip. A Field-Programmable Gate Array (FPGA) is might be included in a multiprocessor system. A FPGA is an integrated circuit containing an array of programmable logic blocks and a hierarchy of reconfigurable interconnects. Input data is processed by to produce outputs. It can be a processor in a multicore or multiprocessor system, or a FPGA.

The multicore programming approach has following advantages −

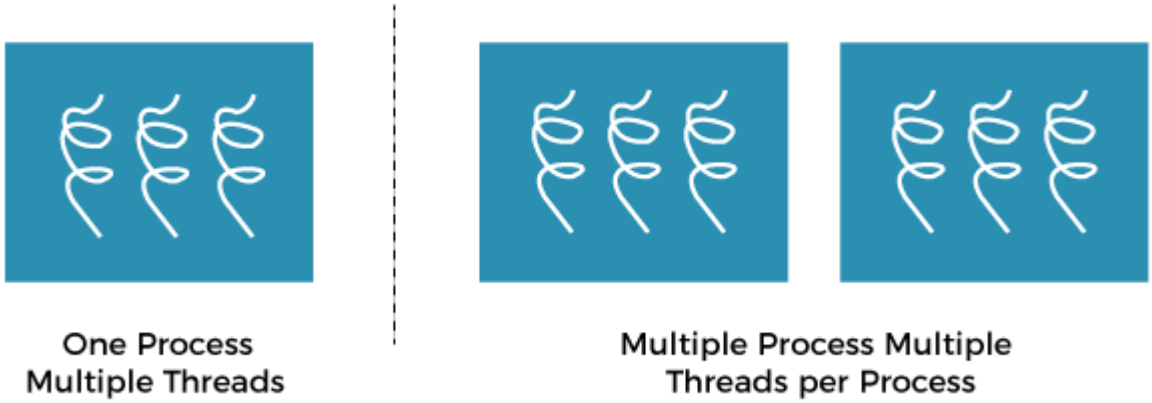
- Multicore and FPGA processing helps to increase the performance of an embedded system.
- Also helps to achieve scalability, so the system can take advantage of increasing numbers of cores and FPGA processing power over time.

Concurrent systems that we create using multicore programming have multiple tasks executing in

parallel. This is known as concurrent execution. When multiple parallel tasks are executed by a processor, it is known as multitasking. A CPU scheduler, handles the tasks that execute in parallel. The CPU implements tasks using operating system threads. So that tasks can execute independently but have some data transfer between them, such as data transfer between a data acquisition module and controller for the system. Data transfer occurs when there is a data dependency.

MULTITHREADING MODEL:

Multithreading allows the application to divide its task into individual threads. In multi-threads, the same process or task can be done by the number of threads, or we can say that there is more than one thread to perform the task in multithreading. With the use of multithreading, multitasking can be achieved.

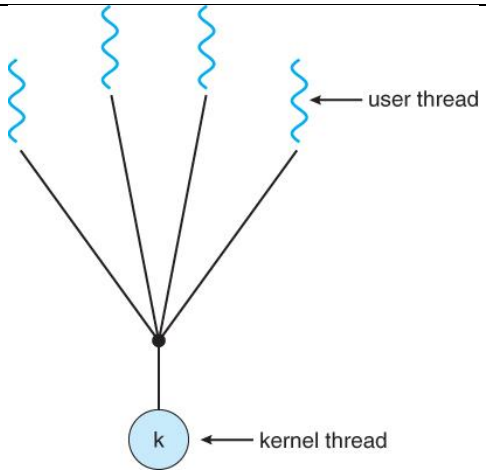


MULTITHREADING MODELS:

- There are two types of threads to be managed in a modern system: User threads and kernel threads.
- User threads are supported above the kernel, without kernel support. These are the threads that application programmers would put into their programs.
- In a specific implementation, the user threads must be mapped to kernel threads, using one of the following strategies.

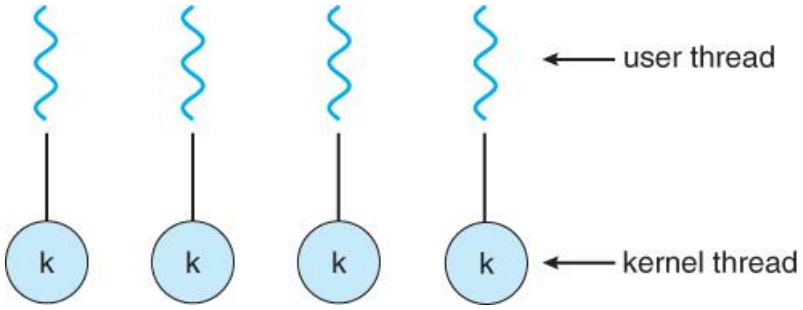
Many-To-One Model

- In the many-to-one model, many user-level threads are all mapped onto a single kernel thread.
- Thread management is handled by the thread library in user space, which is very efficient.
- However, if a blocking system call is made, then the entire process blocks, even if the other user threads would otherwise be able to continue.
- Because a single kernel thread can operate only on a single CPU, the many-to-one model does not allow individual processes to be split across multiple CPUs.
- Green threads for Solaris and GNU Portable Threads implement the many-to-one model in the past, but few systems continue to do so today.



One-To-One Model

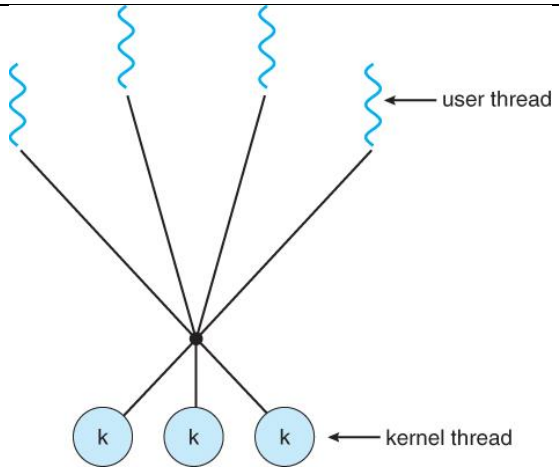
- The one-to-one model creates a separate kernel thread to handle each user thread.
- One-to-one model overcomes the problems listed above involving blocking system calls and the splitting of processes across multiple CPUs.
- However the overhead of managing the one-to-one model is more significant, involving more overhead and slowing down the system.
- Most implementations of this model place a limit on how many threads can be created.
- Linux and Windows from 95 to XP implement the one-to-one model for threads.



One-to-one model

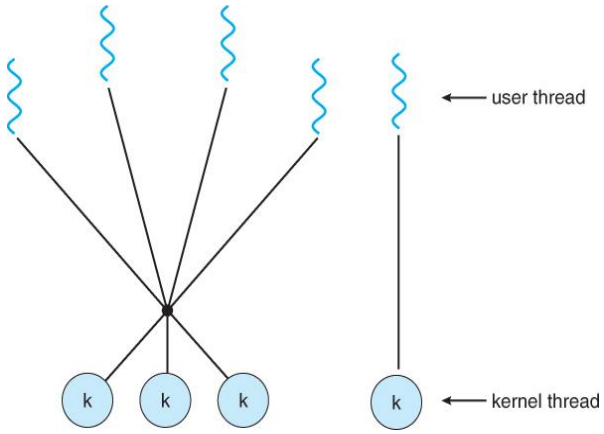
Many-To-Many Model

- The many-to-many model multiplexes any number of user threads onto an equal or smaller number of kernel threads, combining the best features of the one-to-one and many-to-one models.
- Users have no restrictions on the number of threads created.
- Blocking kernel system calls do not block the entire process.
- Processes can be split across multiple processors.
- Individual processes may be allocated variable numbers of kernel threads, depending on the number of CPUs present and other factors.



Many-to-many model

- One popular variation of the many-to-many model is the two-tier model, which allows either many-to-many or one-to-one operation.
- IRIX, HP-UX, and Tru64 UNIX use the two-tier model, as did Solaris prior to Solaris 9.



Video Content / Details of website for further learning (if any):
http://www.cs.nthu.edu.tw/~ychung/slides/CSC3150/Abraham-Silberschatz-Operating-System-Concepts---9th2012.12
https://www.cs.uic.edu/~jbell/CourseNotes/OperatingSystems/4_Threads.html

Important Books/Journals for further learning including the page nos.:
Book:
Abraham Silberschatz,Peter Bear Galvin and Greg Gagne, ," Operating system concepts",2015, John Wiley & Sons (ASIA) ,9th Edition,Page no(467-624).

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

L11

LECTURE HANDOUTS

AI&DS

II/III

Course Name with Code : **19ADCO7&OPERATING SYSTEM**

Course Faculty : **R.NIVETHITHA.**

Unit : **II THREADS AND SCHEDULING ALGORITHM**

Date of Lecture:

Topic of Lecture: Thread Libraries, Threading Issuses.

Introduction :

- A thread library provides the programmer an API for creating and managing threads. There are two primary ways of implementing a thread library
- The threads extension of the POSIX standard, may be provided as either a user- or kernel-level library
- The thread performing this function sets the global data Sum to the value of the summation from 0 to the parameter passed to SummationO

Prerequisite knowledge for Complete understanding and learning of Topic:

- Operating Systems
- Hardware
- Software
- Process

Detailed content of the Lecture:

THREAD LIBRARIES:

A thread library provides the programmer an API for creating and managing threads. There are two primary ways of implementing a thread library. The first approach is to provide a library entirely in user space with no kernel support. All code and data structures for the library exist in user space. This means that invoking a function in the library results in a local function call in user space and not a system call.

The second approach is to implement a kernel-level library supported directly by the operating system. In this case, code and data structures for the library exist in kernel space. Invoking a function in the API for the library typically results in a system call to the kernel.

Three main thread libraries are in use today:

1. POSIX Pthreads,

2. Win32, and
3. Java. Pthreads,

The threads extension of the POSIX standard, may be provided as either a user- or kernel-level library. The Win32 thread library is a kernel-level library available on Windows systems. The Java thread API allows thread creation and management directly in Java programs. However, because in most instances the JVM is running on top of a host operating system, the Java thread API is typically implemented using a thread library available on the host system. This means that on Windows systems, Java threads are typically implemented using the Win32 API; UNIX and Linux systems often use Pthreads.

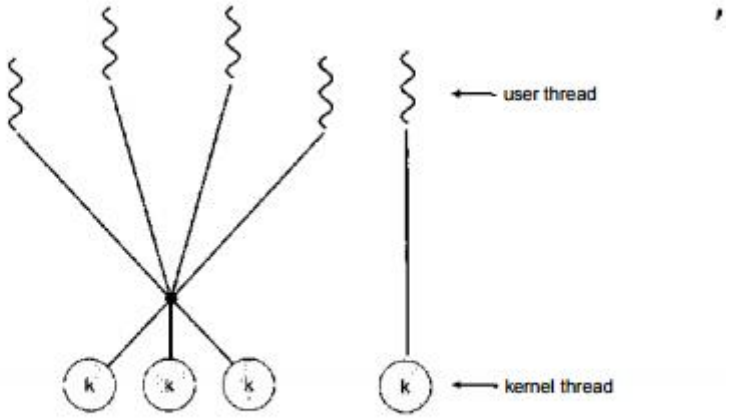


Figure 4.5 Two-level model.

In the remainder of this section, we describe basic thread creation using these three thread libraries. As an illustrative example, we design a multithreaded program that performs the summation of a non-negative integer in a separate thread using the well-known summation function. For example, if N were 5, this function would represent the summation from 0 to 5, which is 15. Each of the three programs will be run with the upper bounds of the summation entered on the command line; thus, if the user enters 8, the summation of the integer values from 0 to 8 will be output.

Pthreads

Pthreads refers to the POSIX standard (IEEE 1003.1c) defining an API for thread creation and synchronization. This is a specification for thread behavior, not an implementation. Operating system designers may implement the specification in any way they wish. Numerous systems implement the Pthreads specification, including Solaris, Linux, Mac OS X, and Tru64 UNIX. Shareware implementations are available in the public domain for the various Windows operating systems as well. In a Pthreads program, separate threads begin execution in a specified function, this is the runner () function. When this program begins.

Win32 Threads

The technique for creating threads using the Win32 thread library is similar to the Pthreads

technique in several ways. we must include the windows.h header file when using the Win32 API. Just as in the Pthreads, data shared by the separate threads—in this case, Sum—are declared globally (the DWORD data type is an unsigned 32-bit integer. We also define the SummationO function that is to be performed in a separate thread. This function is passed a pointer to a void, which Win32 defines as LPVOID. The thread performing this function sets the global data Sum to the value of the summation from 0 to the parameter passed to SummationO

THREADING ISSUES:

The fork( ) and exec( ) System Calls

- Q: If one thread forks, is the entire process copied, or is the new process single-threaded?
- A: System dependant.
- A: If the new process execs right away, there is no need to copy all the other threads. If it doesn't, then the entire process should be copied.
- A: Many versions of UNIX provide multiple versions of the fork call for this purpose.

Signal Handling

- Q: When a multi-threaded process receives a signal, to what thread should that signal be delivered?
- A: There are four major options:
    1. Deliver the signal to the thread to which the signal applies.
    2. Deliver the signal to every thread in the process.
    3. Deliver the signal to certain threads in the process.
    4. Assign a specific thread to receive all signals in a process.
- The best choice may depend on which specific signal is involved.
- UNIX allows individual threads to indicate which signals they are accepting and which they are ignoring. However the signal can only be delivered to one thread, which is generally the first thread that is accepting that particular signal.
- UNIX provides two separate system calls, **kill(pid, signal)** and **pthread_kill(tid, signal)**, for delivering signals to processes or specific threads respectively.
- Windows does not support signals, but they can be emulated using Asynchronous Procedure Calls (APCs). APCs are delivered to specific threads, not processes.

THREAD CANCELLATION

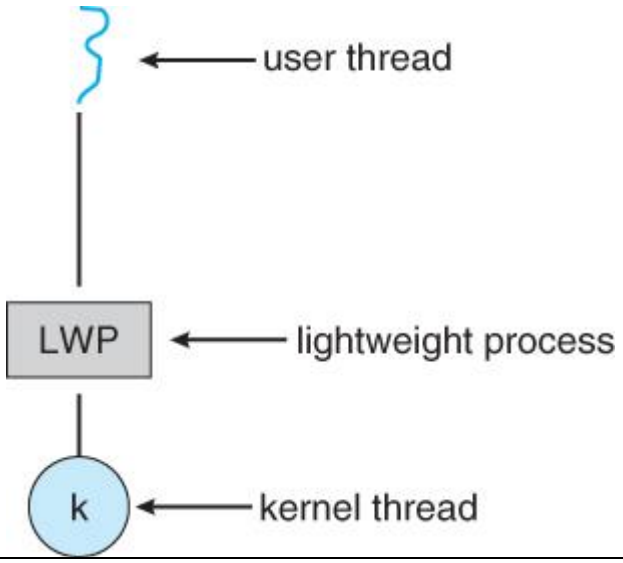
- Threads that are no longer needed may be cancelled by another thread in one of two ways:
    1. **Asynchronous Cancellation** cancels the thread immediately.
    2. **Deferred Cancellation** sets a flag indicating the thread should cancel itself when it is convenient. It is then up to the cancelled thread to check this flag periodically and exit nicely when it sees the flag set.
- (Shared) resource allocation and inter-thread data transfers can be problematic with asynchronous cancellation.

Thread-Local Storage

Most data is shared among threads, and this is one of the major benefits of using threads in the

first place.

- However sometimes threads need thread-specific data also.
- Most major thread libraries (pThreads, Win32, Java) provide support for thread-specific data, known as **thread-local storage** or **TLS.** Note that this is more like static data than local variables,because it does not cease to exist when the function ends.



Video Content / Details of website for further learning (if any):
http://www.cs.nthu.edu.tw/~ychung/slides/CSC3150/Abraham-Silberschatz-Operating-System-Concepts---9th2012.12
https://www.cs.uic.edu/~jbell/CourseNotes/OperatingSystems/4_Threads.html
https://padakuu.com/article/101-thread-libraries

Important Books/Journals for further learning including the page nos.:
Book:
Abraham Silberschatz,Peter Bear Galvin and Greg Gagne, ," Operating system concepts",2015, John Wiley & Sons (ASIA) ,9thEdition,Page no(351-466).

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

L12

LECTURE HANDOUTS

AI&DS

II/III

Course Name with Code : **19ADCO7&OPERATING SYSTEM**

Course Faculty : **R.NIVETHITHA**

Unit : **II THREADS AND SCHEDULING ALGORITHM**

Date of Lecture:

Topic of Lecture: Process synchronization:The Critical-Section Problem, Peterson's Solution

Introduction :

- Process synchronization is the technique to overcome the problem of concurrent access to shared data which can result in data inconsistency.
- Process synchronization is required for consistency of data.
- Every process has a reserved segment of code which is known as **Critical Section**.
- **Mutual Exclusion** − one process at a time can access the critical section

Prerequisite knowledge for Complete understanding and learning of Topic:

- Operating Systems
- Hardware
- Software
- Process

Detailed content of the Lecture:

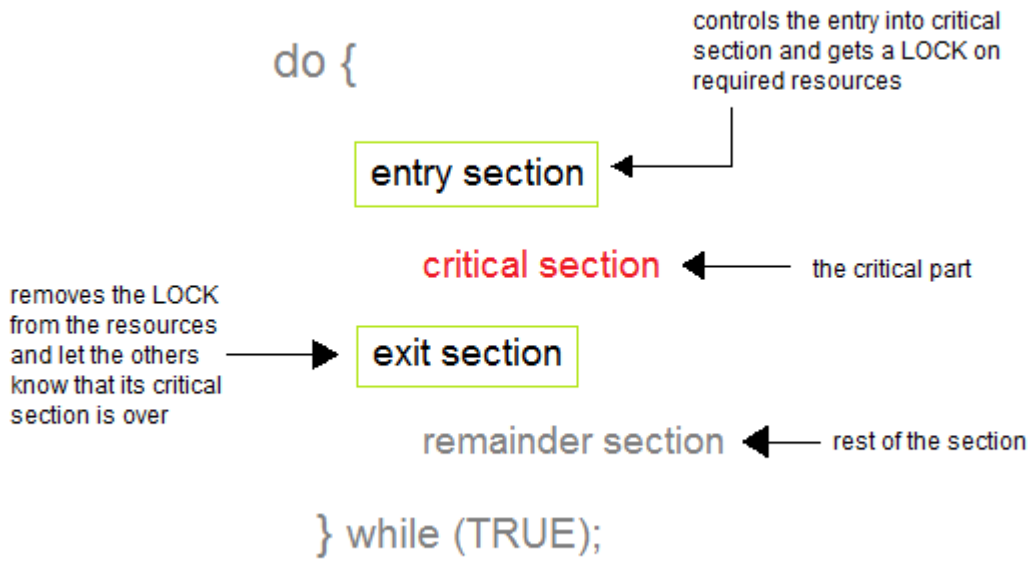
PROCESS SYNCHRONIZATION:

Process synchronization is the technique to overcome the problem of concurrent access to shared data which can result in data inconsistency. A cooperating process is the one which can affect or be affected by other process which will lead to inconsistency in processes data therefore Process synchronization is required for consistency of data.

CRITICAL SECTION PROBLEM:

A Critical Section is a code segment that accesses shared variables and has to be executed as an atomic action. It means that in a group of cooperating processes, at a given point of time, only one process must be executing its critical section. If any other process also wants to execute its critical section, it must wait until the first one finishes. The entry to the critical section is mainly handled

by wait() function while the exit from the critical section is controlled by the signal() function.



Entry Section

In this section mainly the process requests for its entry in the critical section.

Exit Section

This section is followed by the critical section.

The solution to the Critical Section Problem

A solution to the critical section problem must satisfy the following three conditions:

1. Mutual Exclusion

Out of a group of cooperating processes, only one process can be in its critical section at a given point of time.

2. Progress

If no process is in its critical section, and if one or more threads want to execute their critical section then any one of these threads must be allowed to get into its critical section.

3. Bounded Waiting

After a process makes a request for getting into its critical section, there is a limit for how many other processes can get into their critical section, before this process's request is granted. So after the limit is reached, the system must grant the process permission to get into its critical section.

Solutions for the Critical Section

The critical section plays an important role in Process Synchronization so that the problem must be solved.

PETERSON'S SOLUTION:

Peterson's solution is a classic based software solution to the critical-section problem. It is restricted to two processes that alternate execution between their critical sections and remainder sections. Peterson' section requires two data items to be shared between the two processes i.e.

- Int turn;
- Boolean flag[2];

Here, variable turn indicates whose turn is to enter its critical section and flag array indicated whether the process is ready to enter its critical section.

If turn == i, it means process Pi is allowed to enter in its critical section.

If flag[j] is TRUE, it means process j is ready to enter in its critical section

Given below is the structure of process P in Peterson's solution

```
do
{
    Flag[i] = TRUE;
    Turn = j;
    while(flag[j] && turn = = j)
        Critical Section
    Flag[i] = FALSE;
        Remainder section
}while(TRUE);
```

Peterson's Solution preserves all three conditions

- **Mutual Exclusion** − one process at a time can access the critical section.
- **Progress** − A process outside the critical section does not block other processes from entering the critical section.
- **Bounded Waiting** − Every process will get a chance to enter its critical section without waiting indefinitely.

Video Content / Details of website for further learning (if any):
http://www.cs.nthu.edu.tw/~ychung/slides/CSC3150/Abraham-Silberschatz-Operating-System-Concepts---9th2012.12
https://www.tutorialspoint.com/process-synchronization-in-c-cplusplus

Important Books/Journals for further learning including the page nos.:
Book:
Abraham Silberschatz,Peter Bear Galvin and Greg Gagne, ," Operating system concepts",2015, John Wiley & Sons (ASIA) ,9th Edition,Page no(351-466).

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

L13

LECTURE HANDOUTS

AI&DS

II/III

Course Name with Code : 19ADCO7&OPERATING SYSTEM

Course Teacher : R.NIVETHITHA

Unit : II THREADS AND SCHEDULING ALGORITHM

Date of Lecture:

Topic of Lecture: Synchronization Hardware, Mutex Locks, Semaphores

Introduction :

- Test and Set () is a hardware solution to solve the problem of synchronization the file system is the most visible aspect of an operating system
- Whenever the process is trying to enter their critical sections they need to enquire about the value of lock.
- Semaphore is a synchronization tool that is used to overcome the problems generated by TestAndSet() and Swap() instructions

Prerequisite knowledge for Complete understanding and learning of Topic:

- Operating Systems
- Hardware
- Software
- Process

Detailed content of the Lecture:

SYNCHRONIZATION HARDWARE

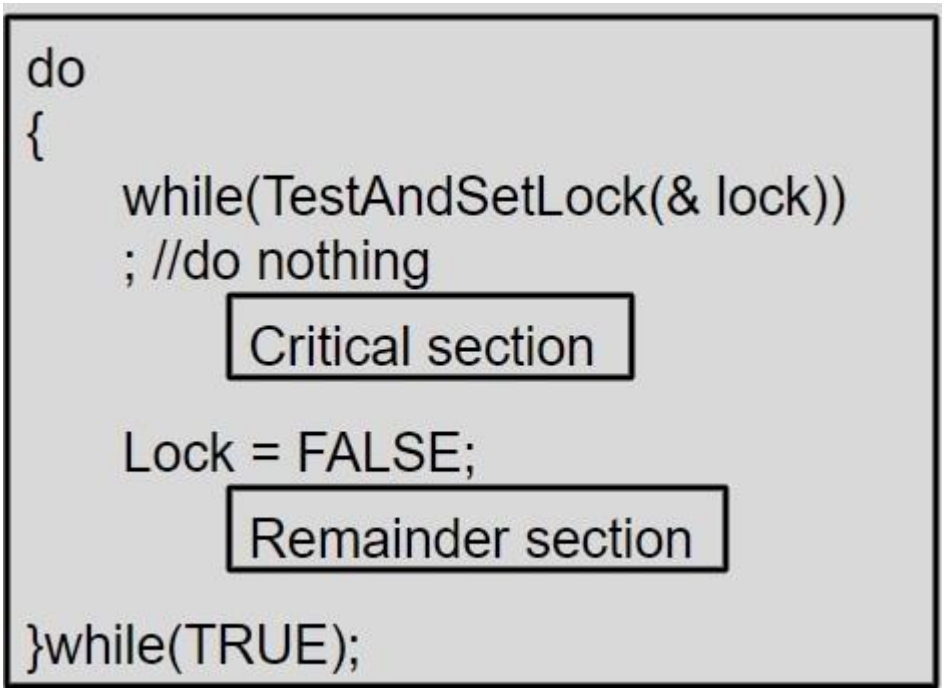
It is implemented using two types of instructions

- Test and Set()
- swap()

Test and Set () is a hardware solution to solve the problem of synchronization. In this, there is a shared variable which is shared by multiple processes known as Lock which can have one value from 0 and 1 where 1 represents Lock gained and 0 represents Lock released.

Whenever the process is trying to enter their critical sections they need to enquire about the value of lock. If the value of lock is 1 then they have to wait until the value of lock won't get changed to 0

Given below is the mutual-exclusion implementation with TestAndSet()



MUTEX LOCKS

As the synchronization hardware solution is not easy to implement for everyone, a strict software approach called Mutex Locks was introduced. In this approach, in the entry section of code, a LOCK is acquired over the critical resources modified and used inside the critical section, and in the exit section that LOCK is released.

As the resource is locked while a process executes its critical section hence no other process can access it.

Semaphores are integer variables that are used to solve the critical section problem by using two atomic operations, wait and signal that are used for process synchronization.

The definitions of wait and signal are as follows −

- **Wait**

  The wait operation decrements the value of its argument S, if it is positive. If S is negative or zero, then no operation is performed.

```
wait(S)
{
  while (S<=0);

  S--;
}
```

- **Signal**

  The signal operation increments the value of its argument S.

```
signal(S)
{
  S++;
}
```

SEMAPHORES:

Semaphore is a synchronization tool that is used to overcome the problems generated by TestAndSet() and Swap() instructions. A semaphore S is an integer variable that can be accessed through two standard atomic operations that are wait() and signal()

Function for wait():

```
wait(S) {
  While S <= 0
  ; // no operation
  S--;
}
```

Function for Signal():

```
signal(S) {
  S++;
}
```

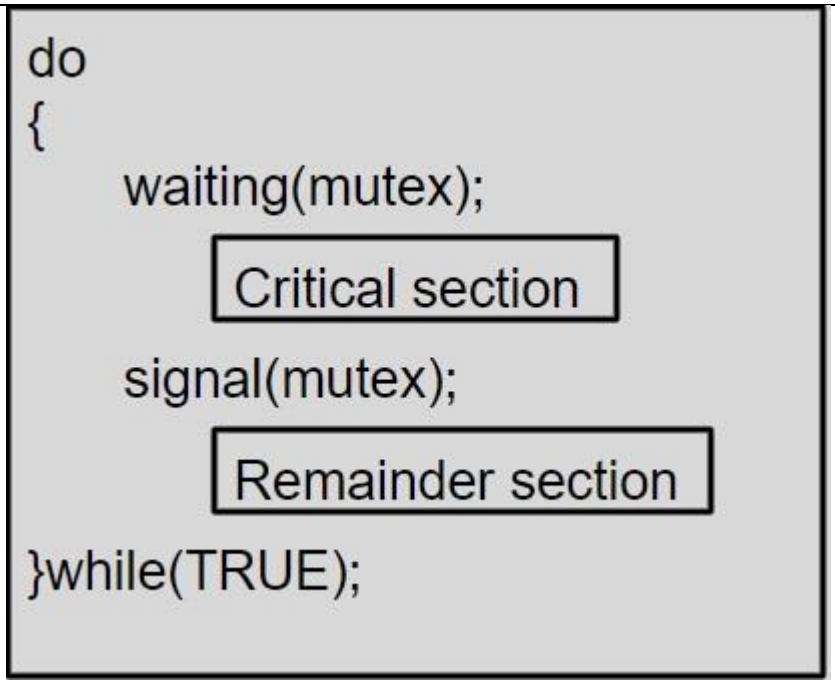
When one process is modifying the value of semaphore then no other process can simultaneously manipulate the same semaphore value

Operating system use two types of semaphores that are

Counting Semaphore:

The value of this type of semaphore can over an unrestricted domain

Given below is the mutual-exclusion implementation with semaphore



Binary Semaphore − the value of this type of semaphore can over between 0 and 1. They are also known as Mutex Locks. Operating system makes use of this for resolving the issues with critical section in multiple processes.

Video Content / Details of website for further learning (if any):
http://www.cs.nthu.edu.tw/~ychung/slides/CSC3150/Abraham-Silberschatz-Operating-System-Concepts---9th2012.12
https://www.tutorialspoint.com/process-synchronization-in-c-cplusplus

Important Books/Journals for further learning including the page nos.:
Book:
Abraham Silberschatz,Peter Bear Galvin and Greg Gagne, ," Operating system concepts",2015, John Wiley & Sons (ASIA) ,9th Edition,Page no(351-466).

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

L14

LECTURE HANDOUTS

AI&DS

II/III

Course Name with Code : **19ADCO7&OPERATING SYSTEM**

Course Teacher : **R.NIVETHITHA.**

Unit : **II THREADS AND SCHEDULING ALGORITHM**

Date of Lecture:

Topic of Lecture: Classic Problems Of Synchronization, Monitors

Introduction :

- Semaphore can be used in other synchronization problems besides Mutual Exclusion.
- The main complexity of this problem is that we must have to maintain the count for both empty and full containers that are available.
- Monitors are a synchronization construct that were created to overcome the problems caused by semaphores such as timing errors.
- The shared data variables cannot be directly accessed by a process and procedures are required to allow a single process to access the shared data variables at a time.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Operating Systems
- Hardware
- Software
- Process

Detailed content of the Lecture:

CLASSIC PROBLEMS OF SYNCHRONIZATION

Semaphore can be used in other synchronization problems besides Mutual Exclusion.

Below are some of the classical problem depicting flaws of process synchronaization in systems where cooperating processes are present.

We will discuss the following three problems:

1. Bounded Buffer (Producer-Consumer) Problem
2. Dining Philosophers Problem
3. The Readers Writers Problem

Bounded Buffer Problem

Because the buffer pool has a maximum size, this problem is often called the **Bounded buffer problem**.

- This problem is generalised in terms of the **Producer Consumer problem**, where a **finite** buffer pool is used to exchange messages between producer and consumer processes.
- Solution to this problem is, creating two counting semaphores "full" and "empty" to keep track of the current number of full and empty buffers respectively.
- In this Producers mainly produces a product and consumers consume the product, but both can use of one of the containers each time.
- The main complexity of this problem is that we must have to maintain the count for both empty and full containers that are available.

Dining Philosophers Problem

- The dining philosopher's problem involves the allocation of limited resources to a group of processes in a deadlock-free and starvation-free manner.
- There are five philosophers sitting around a table, in which there are five chopsticks/forks kept beside them and a bowl of rice in the centre, When a philosopher wants to eat, he uses two chopsticks - one from their left and one from their right. When a philosopher wants to think, he keeps down both chopsticks at their original place.

The Readers Writers Problem

- In this problem there are some processes(called **readers**) that only read the shared data, and never change it, and there are other processes(called **writers**) who may change the data in addition to reading, or instead of reading it.
- There are various type of readers-writers problem, most centred on relative priorities of readers and writers.
- The main complexity with this problem occurs from allowing more than one reader to access the data at the same time.

MONITORS

Monitors are a synchronization construct that were created to overcome the problems caused by semaphores such as timing errors.

Monitors are abstract data types and contain shared data variables and procedures. The shared data variables cannot be directly accessed by a process and procedures are required to allow a single process to access the shared data variables at a time.

This is demonstrated as follows:

```
monitor monitorName
{
  data variables;
```

```
  Procedure P1(....)
  {

  }

  Procedure P2(....)
  {

  }

  Procedure Pn(....)
  {

  }

  Initialization Code(....)
  {

  }
}
```

Only one process can be active in a monitor at a time. Other processes that need to access the shared variables in a monitor have to line up in a queue and are only provided access when the previous process release the shared variables.

Video Content / Details of website for further learning (if any):

http://www.cs.nthu.edu.tw/~ychung/slides/CSC3150/Abraham-Silberschatz-Operating-System-Concepts---9th2012.12

https://www.studytonight.com/operating-system/classical-synchronization-problems

https://www.tutorialspoint.com/monitor

Important Books/Journals for further learning including the page nos.:

Book:

Abraham Silberschatz,Peter Bear Galvin and Greg Gagne, ," Operating system concepts",2015, John Wiley & Sons (ASIA) ,9th Edition,Page no(351-466).

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

L15

LECTURE HANDOUTS

AI&DS

II/III

Course Name with Code : **19ADCO7&OPERATING SYSTEM**

Course Teacher : **R.NIVETHITHA.**

Unit : **II THREADS AND SCHEDULING ALGORITHM**

Date of Lecture:

Topic of Lecture: Synchronization Examples, Alternative Approaches

Introduction :

- Interrupt masks, which protect access to global resources (critical section) on uniprocessor systems
- Readers–writer locks, for the longer section of codes which are accessed very frequently but don't change very often.
- A memory transaction is a sequence of read-write operations to memory that are performed atomically.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Operating Systems
- Hardware
- Software
- Process

Detailed content of the Lecture:

SYNCHRONIZATION EXAMPLES

Following are some synchronization examples with respect to different platforms.

Synchronization in Windows

Windows provides:

- Interrupt masks, which protect access to global resources (critical section) on uniprocessor systems;
- Spinlocks, which prevent, in multiprocessor systems, spinlocking-thread from being preempted;
- Dispatchers, which act like mutexes, semaphores, events, and timers.

Synchronization in Linux

Linux provides:

- semaphores;
- spinlock;
- barriers
- mutex
- readers–writer locks, for the longer section of codes which are accessed very frequently but don't change very often.
- Read-copy-update (RCU)

Enabling and disabling of kernel preemption replaced spinlocks on uniprocessor systems. Prior to kernel version 2.6, Linux disabled interrupt to implement short critical sections. Since version 2.6 and later, Linux is fully preemptive.

Synchronization in Solaris

Solaris provides:

- Semaphores;
- Condition variables;
- Adaptive mutexes, binary semaphores that are implemented differently depending upon the conditions;
- Readers–writer locks: turnstiles, queue of threads which are waiting on acquired lock

Pthreads synchronization

Pthreads is a platform-independent API that provides:

- Mutexes;
- Condition Variables;
- Readers–Writer Locks;
- Spinlocks;
- Barriers.

ALTERNATIVE APPROACHES:

- Transactional Memory
- OpenMP
- Functional Programming Languages

Consider a function update() that must be called atomically. One option is to use mutex locks:

Transaction memory:

- A memory transaction is a sequence of read-write operations to memory that are performed atomically. A transaction can be completed by adding atomic{S} which ensure statements in S are executed atomically

OpenMP:

- OpenMP is a set of compiler directives and API that support parallel progamming. void update(int value) { #pragma omp critical { count += value } }
- The code contained within the #pragma omp critical directive is treated as a critical section and performed atomically.

Functional programming languages:

- Functional programming languages offer a different paradigm than procedural languages in that they do not maintain state
- Variables are treated as immutable and cannot change state once they have been assigned a value
- .There is increasing interest in functional languages such as Erlang and Scala for their approach in handling data races.

Video Content / Details of website for further learning (if any):
http://www.cs.nthu.edu.tw/~ychung/slides/CSC3150/Abraham-Silberschatz-Operating-System-Concepts---9th2012.12
https://en.wikipedia.org/wiki/Synchronization_(computer_science)#Synchronization_examples

Important Books/Journals for further learning including the page nos.:
Book:
Abraham Silberschatz,Peter Bear Galvin and Greg Gagne, ," Operating system concepts",2015, John Wiley & Sons (ASIA) ,9th Edition,Page no(351-466).

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

L16

LECTURE HANDOUTS

AI&DS

II/III

Course Name with Code : **19ADCO7&OPERATING SYSTEM**

Course Teacher : **R.NIVETHITHA.**

Unit : **II THREADS AND SCHEDULING ALGORITHM**

Date of Lecture:

Topic of Lecture: CPU Scheduling: Scheduling- Criteria, Scheduling Algorithms.

Introduction :

- To make out the best use of the CPU and not to waste any CPU cycle, the CPU would be working most of the time
- It is the total number of processes completed per unit of time or rather says the total amount of work done in a unit of time
- A Process Scheduler schedules different processes to be assigned to the CPU based on particular scheduling algorithms.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Operating Systems
- Hardware
- Software
- Process

Detailed content of the Lecture:

CPU Scheduling: Scheduling Criteria

There are many different criteria to check when considering the **"best"** scheduling algorithm, they are:

CPU Utilization

- To make out the best use of the CPU and not to waste any CPU cycle, the CPU would be working most of the time(Ideally 100% of the time).
- Considering a real system, CPU usage should range from 40% (lightly loaded) to 90% (heavily loaded.)

Throughput

It is the total number of processes completed per unit of time or rather says the total amount of work done in a unit of time. This may range from 10/second to 1/hour depending on the specific processes.

Turnaround Time

It is the amount of time taken to execute a particular process, i.e. The interval from the time of submission of the process to the time of completion of the process(Wall clock time).

Waiting Time

The sum of the periods spent waiting in the ready queue amount of time a process has been waiting in the ready queue to acquire get control on the CPU.

Load Average

It is the average number of processes residing in the ready queue waiting for their turn to get into the CPU.

Response Time

Amount of time it takes from when a request was submitted until the first response is produced. Remember, it is the time till the first response and not the completion of process execution(final response). In general CPU utilization and Throughput are maximized and other factors are reduced for proper optimization.

SCHEDULING ALGORITHMS:

A Process Scheduler schedules different processes to be assigned to the CPU based on particular scheduling algorithms. There are six popular process scheduling algorithms which we are going to discuss in this chapter

- First-Come, First-Served (FCFS) Scheduling
- Shortest-Job-Next (SJN) Scheduling
- Priority Scheduling
- Shortest Remaining Time
- Round Robin(RR) Scheduling
- Multiple-Level Queues Scheduling

These algorithms are either **non-preemptive or preemptive**. Non-preemptive algorithms are designed so that once a process enters the running state, it cannot be preempted until it completes its allotted time, whereas the preemptive scheduling is based on priority where a scheduler may preempt

a low priority running process anytime when a high priority process enters into a ready state.

First Come First Serve (FCFS)

- Jobs are executed on first come, first serve basis.
- It is a non-preemptive, pre-emptive scheduling algorithm.
- Easy to understand and implement.
- Its implementation is based on FIFO queue.
- Poor in performance as average wait time is high.

Wait time of each process is as follows

Process	Wait Time : Service Time - Arrival Time
P0	0 - 0 = 0
P1	5 - 1 = 4
P2	8 - 2 = 6
P3	16 - 3 = 13

Average Wait Time: (0+4+6+13) / 4 = 5.75

Shortest Job Next (SJN)

- This is also known as **shortest job first**, or SJF
- This is a non-preemptive, pre-emptive scheduling algorithm.
- Best approach to minimize waiting time.
- Easy to implement in Batch systems where required CPU time is known in advance.
- Impossible to implement in interactive systems where required CPU time is not known.
- The processer should know in advance how much time process will take.

Given: Table of processes, and their Arrival time, Execution time

Process	Arrival Time	Execution Time	Service Time
P0	0	5	0
P1	1	3	5
P2	2	8	14
P3	3	6	8

Waiting time of each process is as follows

Process	Waiting Time
P0	0 - 0 = 0
P1	5 - 1 = 4
P2	14 - 2 = 12
P3	8 - 3 = 5

Average Wait Time: (0 + 4 + 12 + 5)/4 = 21 / 4 = 5.25

Priority Based Scheduling

- Priority scheduling is a non-preemptive algorithm and one of the most common scheduling

algorithms in batch systems.

- Each process is assigned a priority. Process with highest priority is to be executed first and so on.
- Processes with same priority are executed on first come first served basis.
- Priority can be decided based on memory requirements, time requirements or any other resource requirement.

Given: Table of processes, and their Arrival time, Execution time, and priority. Here we are considering 1 is the lowest priority.

Process	Arrival Time	Execution Time	Priority	Service Time
P0	0	5	1	0
P1	1	3	2	11
P2	2	8	1	14
P3	3	6	3	5

Waiting time of each process is as follows

Process	Waiting Time
P0	0 - 0 = 0
P1	11 - 1 = 10
P2	14 - 2 = 12
P3	5 - 3 = 2

Average Wait Time: (0 + 10 + 12 + 2)/4 = 24 / 4 = 6

Shortest Remaining Time

- Shortest remaining time (SRT) is the preemptive version of the SJN algorithm.
- The processor is allocated to the job closest to completion but it can be preempted by a newer ready job with shorter time to completion.
- Impossible to implement in interactive systems where required CPU time is not known.
- It is often used in batch environments where short jobs need to give preference.

Round Robin Scheduling

- Round Robin is the preemptive process scheduling algorithm.
- Each process is provided a fix time to execute, it is called a **quantum**.
- Once a process is executed for a given time period, it is preempted and other process executes for a given time period.
- Context switching is used to save states of preempted processes.

Wait time of each process is as follows

Process	Wait Time : Service Time - Arrival Time
P0	(0 - 0) + (12 - 3) = 9
P1	(3 - 1) = 2
P2	(6 - 2) + (14 - 9) + (20 - 17) = 12
P3	(9 - 3) + (17 - 12) = 11

Average Wait Time: (9+2+12+11) / 4 = 8.5

Multiple-Level Queues Scheduling

Multiple-level queues are not an independent scheduling algorithm. They make use of other existing algorithms to group and schedule jobs with common characteristics.

- Multiple queues are maintained for processes with common characteristics.
- Each queue can have its own scheduling algorithms.
- Priorities are assigned to each queue.

For example, CPU-bound jobs can be scheduled in one queue and all I/O-bound jobs in another queue. The Process Scheduler then alternately selects jobs from each queue and assigns them to the CPU based on the algorithm assigned to the queue.

Video Content / Details of website for further learning (if any):
http://www.cs.nthu.edu.tw/~ychung/slides/CSC3150/Abraham-Silberschatz-Operating-System-Concepts---9th2012.12
https://www.studytonight.com/operating-system/cpu-scheduling
https://www.tutorialspoint.com/operating_system/os_process_scheduling_algorithms.htm

Important Books/Journals for further learning including the page nos.:
Book:
Abraham Silberschatz,Peter Bear Galvin and Greg Gagne, ," Operating system concepts",2015, John Wiley & Sons (ASIA) ,9th Edition,Page no(351-466).

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

L17

LECTURE HANDOUTS

AI&DS

II/III

Course Name with Code : 19ADCO7&OPERATING SYSTEM

Course Teacher : R.NIVETHITHA.

Unit : II THREADS AND SCHEDULING ALGORITHM

Date of Lecture:

Topic of Lecture: Thread Scheduling, Multiple Process Scheduling

Introduction :

- Scheduling of user level threads (ULT) to kernel level threads (KLT) via leightweight process (LWP) by the application developer.
- Scheduling of kernel level threads by the system scheduler to perform different unique os functions.
- These systems have multiple processors working in parallel that share the computer clock, memory, bus, peripheral devices etc

Prerequisite knowledge for Complete understanding and learning of Topic:

- Operating Systems
- Hardware
- Software
- Process

Detailed content of the Lecture:

THREAD SCHEDULING:

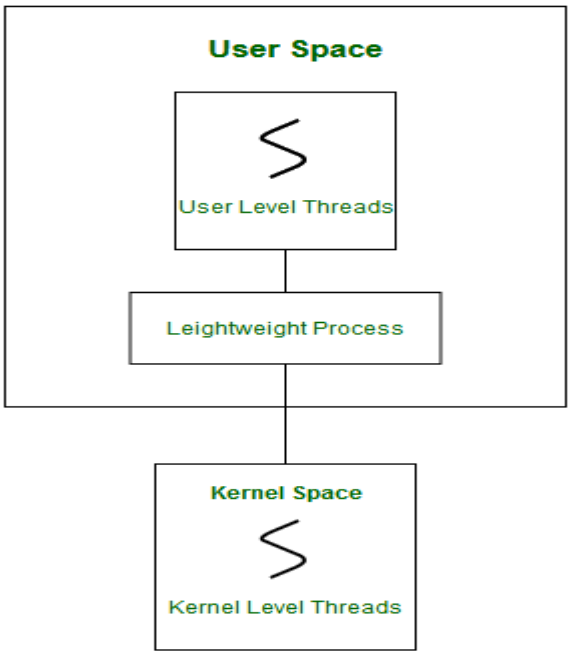
Scheduling of threads involves two boundary scheduling,

- Scheduling of user level threads (ULT) to kernel level threads (KLT) via leightweight process (LWP) by the application developer.
- Scheduling of kernel level threads by the system scheduler to perform different unique os functions.

Lightweight Process (LWP) :

Light-weight process are threads in the user space that acts as an interface for the ULT to access the physical CPU resources. Thread library schedules which thread of a process to run on which LWP and how long. The number of LWP created by the thread library depends on the type of application. In the case of an I/O bound application, the number of LWP depends on the number of user-level threads. This is because when an LWP is blocked on an I/O operation, then to invoke the other ULT the thread library needs to create and schedule another LWP.

Thus, in an I/O bound application, the number of LWP is equal to the number of the ULT. In the case of a CPU bound application, it depends only on the application. Each LWP is attached to a separate kernel-level thread.



In real-time, the first boundary of thread scheduling is beyond specifying the scheduling policy and the priority. It requires two controls to be specified for the User level threads: Contention scope, and Allocation domain. These are explained as following below.

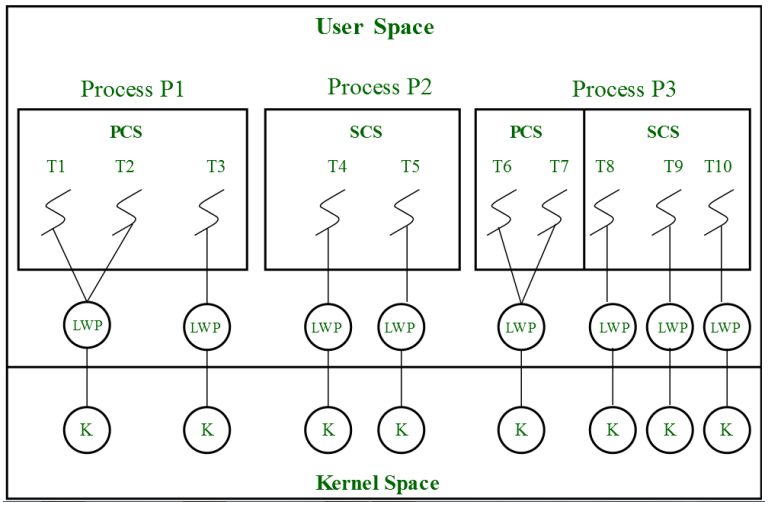
Contention Scope :

The word contention here refers to the competition or fight among the User level threads to access the kernel resources. Thus, this control defines the extent to which contention takes place. It is defined by the application developer using the thread library. Depending upon the extent of contention it is classified as **Process Contention Scope** and **System Contention Scope**.

1. **Process Contention Scope (PCS)**
   The contention takes place among threads **within a same process**. The thread library schedules the high-prioritized PCS thread to access the resources via available LWPs (priority as specified by the application developer during thread creation).
2. **System Contention Scope (SCS)**
   The contention takes place among **all threads in the system**. In this case, every SCS thread is associated to each LWP by the thread library and are scheduled by the system scheduler to access the kernel resources.
   In LINUX and UNIX operating systems, the POSIX Pthread library provides a function *Pthread_attr_setscope* to define the type of contention scope for a thread during its creation.
   int Pthread_attr_setscope(pthread_attr_t *attr, int scope)

3.Allocation Domain :

The allocation domain is **a set of one or more resources** for which a thread is competing. In a multicore system, there may be one or more allocation domains where each consists of one or more cores. One ULT can be a part of one or more allocation domain. Due to this high complexity in dealing with hardware and software architectural interfaces, this control is not specified. But by default, the multicore system will have an interface that affects the allocation domain of a thread.
Consider a scenario, an operating system with three process P1, P2, P3 and 10 user level threads (T1 to T10) with a single allocation domain. 100% of CPU resources will be distributed among all the three processes. The amount of CPU resources allocated to each process and to each thread depends on the contention scope, scheduling policy and priority of each thread defined by the application developer using thread library and also depends on the system scheduler. These User level threads are of a different contention scope.



MULTIPROCESSOR SYSTEMS

Most computer systems are single processor systems i.e they only have one processor. However, multiprocessor or parallel systems are increasing in importance nowadays. These systems have multiple processors working in parallel that share the computer clock, memory, bus, peripheral devices etc.

Types of Multiprocessors

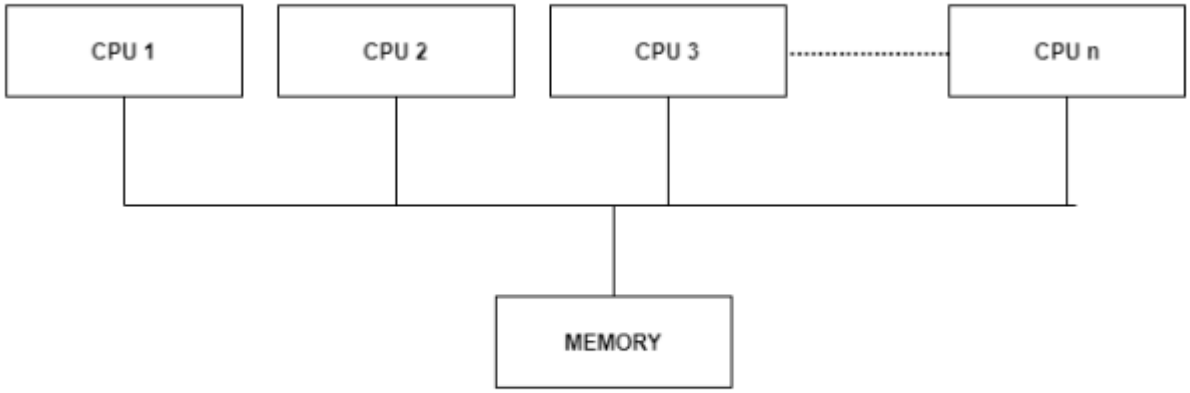
There are mainly two types of multiprocessors i.e. symmetric and asymmetric multiprocessors. Details about them are as follows

Symmetric Multiprocessors

In these types of systems, each processor contains a similar copy of the operating system and they all communicate with each other. All the processors are in a peer to peer relationship i.e. no master - slave relationship exists between them.

An example of the symmetric multiprocessing system is the Encore version of Unix for the Multimax Computer.

An image demonstrating the multiprocessor architecture is



Multiprocessing Architecture

Asymmetric Multiprocessors

In asymmetric systems, each processor is given a predefined task. There is a master processor that gives instruction to all the other processors. Asymmetric multiprocessor system contains a master slave relationship.

Asymmetric multiprocessor was the only type of multiprocessor available before symmetric multiprocessors were created. Now also, this is the cheaper option.

Video Content / Details of website for further learning (if any):
http://www.cs.nthu.edu.tw/~ychung/slides/CSC3150/Abraham-Silberschatz-Operating-System-Concepts---9th2012.12
https://www.geeksforgeeks.org/thread-scheduling/
https://www.tutorialspoint.com/Multiprocessor-Systems

Important Books/Journals for further learning including the page nos.:
Book:
Abraham Silberschatz,Peter Bear Galvin and Greg Gagne, ," Operating system concepts",2015, John Wiley & Sons (ASIA) ,9th Edition,Page no(351-466).

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

L18

LECTURE HANDOUTS

AI&DS

II/III

Course Name with Code : 19ADCO7&OPERATING SYSTEM

Course Teacher : R.NIVETHITHA.

Unit : II THREADS AND SCHEDULING ALGORITHM

Date of Lecture:

Topic of Lecture: Real-Time CPU Scheduling , Algorithm Evaluation

Introduction :

- The allocation methods define how the files are stored in the disk blocks. There are three main disk space or file allocation methods.
- The directory entry for a file with contiguous allocation contains address of starting block
- The directory entry contains a pointer to the starting and the ending file block
- A file system is responsible to allocate the free blocks to the file therefore it has to keep track of all the free blocks present in the disk.

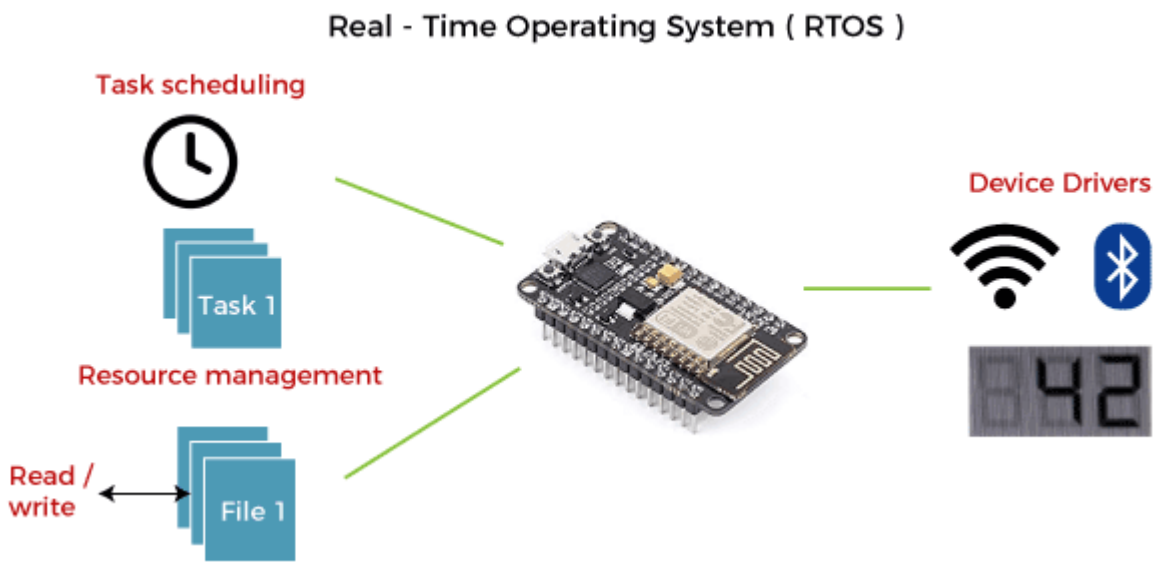
Prerequisite knowledge for Complete understanding and learning of Topic:

- Operating Systems
- Hardware
- Software
- Process

Detailed content of the Lecture:

A real-time operating system (RTOS)

It is a special-purpose operating system used in computers that has strict time constraints for any job to be performed. It is employed mostly in those systems in which the results of the computations are used to influence a process while it is executing. Whenever an event external to the computer occurs, it is communicated to the computer with the help of some sensor used to monitor the event. The sensor produces the signal that is interpreted by the operating system as an interrupt. On receiving an interrupt, the operating system invokes a specific process or a set of processes to serve the interrupt.



This process is completely uninterrupted unless a higher priority interrupt occurs during its execution. Therefore, there must be a strict hierarchy of priority among the interrupts. The interrupt with the highest priority must be allowed to initiate the process , while lower priority interrupts should be kept in a buffer that will be handled later. Interrupt management is important in such an operating system.

8.5M
169
Java Try Catch

Real-time operating systems employ special-purpose operating systems because conventional operating systems do not provide such performance.

The various examples of Real-time operating systems are:

- MTS
- Lynx
- QNX
- VxWorks etc.

Applications of Real-time operating system (RTOS):

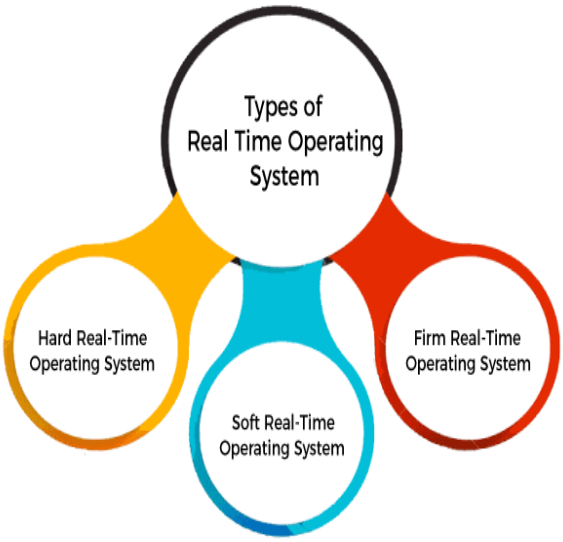
RTOS is used in real-time applications that must work within specific deadlines. Following are the common areas of applications of Real-time operating systems are given below.

- Real-time running structures are used inside the Radar gadget.
- Real-time running structures are utilized in Missile guidance.
- Real-time running structures are utilized in on line inventory trading.
- Real-time running structures are used inside the cell phone switching gadget.
- Real-time running structures are utilized by Air site visitors to manipulate structures.

- Real-time running structures are used in Medical Imaging Systems.
- Real-time running structures are used inside the Fuel injection gadget.
- Real-time running structures are used inside the Traffic manipulate gadget.
- Real-time running structures are utilized in Autopilot travel simulators.

Types of Real-time operating system

Following are the three types of RTOS systems are:



Hard Real-Time operating system:

In Hard RTOS, all critical tasks must be completed within the specified time duration, i.e., within the given deadline. Not meeting the deadline would result in critical failures such as damage to equipment or even loss of human life.

Soft Real-Time operating system:

Soft RTOS accepts a few delays via the means of the Operating system. In this kind of RTOS, there may be a closing date assigned for a particular job, but a delay for a small amount of time is acceptable. So, cut off dates are treated softly via means of this kind of RTOS.

ALGORITHM EVALUATION IN OPERATING SYSTEM:

There are a number of scheduling algorithms and strategies. The choice of algorithms depends heavily on the most important criteria. Once an algorithm is selected, it should evaluated. Here are some ways to evaluate an algorithm:

1. Deterministic Modeling:

Take a particular predetermined workload (set of processes) and define the performance of the algorithm dependant on that workload. This form of evaluation is simple 'and can be done quickly. However, it is not very general. We might get ft completely different idea about the

performance of an algorithm if we chose a different set of processes.

2. Queuing Models:

We can examine the distribution of CPU burst and I/O wait times. We can apply a formula and compute one of three things.

Average length of the queue

Average waiting time of the queue

Average arrival rate

3 Simulation:

A lot of processes are required to simulate an operating system. The processes should mimic those of typical users and the operating system itself that can be difficult. The simulation may take a long time to complete. For more accurate results, it may be necessary to use process information from a real system.

4.Implementation:

The only real way to test an operating system is to write the code and run it. However, this approach is very expensive.

Video Content / Details of website for further learning (if any):
http://www.cs.nthu.edu.tw/~ychung/slides/CSC3150/Abraham-Silberschatz-Operating-System-Concepts---9th2012.12
https://www.javatpoint.com/real-time-operating-system
https://taqi-shah.blogspot.com/2012/06/algorithm-evaluation-in-operating.html

Important Books/Journals for further learning including the page nos.:
Book:
Abraham Silberschatz,Peter Bear Galvin and Greg Gagne, ," Operating system concepts",2015, John Wiley & Sons (ASIA) ,9th Edition,Page no(351-466).

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

L19

LECTURE HANDOUTS

AI&DS

II/III

Course Name with Code : 19ADCO7&OPERATING SYSTEM

Course Faculty : R.NIVETHITHA.

Unit : III MEMORY MANAGEMENT

Date of Lecture:

Topic of Lecture: Management Strategies: Swapping, contiguous memory allocation.

Introduction :

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

Prerequisite knowledge for Complete understanding and learning of Topic:

- Operating Systems
- Hardware
- Software
- Process

Detailed content of the Lecture:

- Memory management is concerned with managing the primary memory. Memory consists of array of bytes or words each with their own address. The instructions are fetched from the memory by the CPU based on the value program counter.

FUNCTIONS OF MEMORY MANAGEMENT:

- Keeping track of status of each memory location.
- Determining the allocation policy.
- Memory allocation technique.
- De-allocation technique.

ADDRESS BINDING:

- Programs are stored on the secondary storage disks as binary executable files. When the programs are to be executed they are brought in to the main memory.

.

- One of the processes which are to be executed is fetched from the queue and placed in the main memory.
- During the execution it fetches instruction and data from main memory. After the process terminates it returns back the memory space.
- During execution the process will go through different steps and in each step the address is represented in different ways.
- In source program the address is symbolic.
- The compiler converts the symbolic address to re-locatable address. The loader will convert this re-locatable address to absolute address.
- Binding of instructions and data can be done at any step along the way:
- Compile time:-If we know whether the process resides in memory then absolute code can be
- Generated. If the static address changes then it is necessary to re-compile the code from the beginning.

DYNAMIC LOADING:

- For a process to be executed it should be loaded in to the physical memory. The size of the process is limited to the size of the physical memory.
- Dynamic loading is used to obtain better memory utilization. In dynamic loading the routine or procedure will not be loaded until it is called.
- Whenever a routine is called, the calling routine first checks whether the called routine is already loaded or not.
- If it is not loaded it cause the loader to load the desired program in to the memory and updates the programs address table to indicate the change and control is passed to newly called routine.

SWAPPING:

Swapping is a memory management scheme in which any process can be temporarily swapped from main memory to secondary memory so that the main memory can be made available for other processes. It is used to improve main memory utilization.

There are two types in swapping:

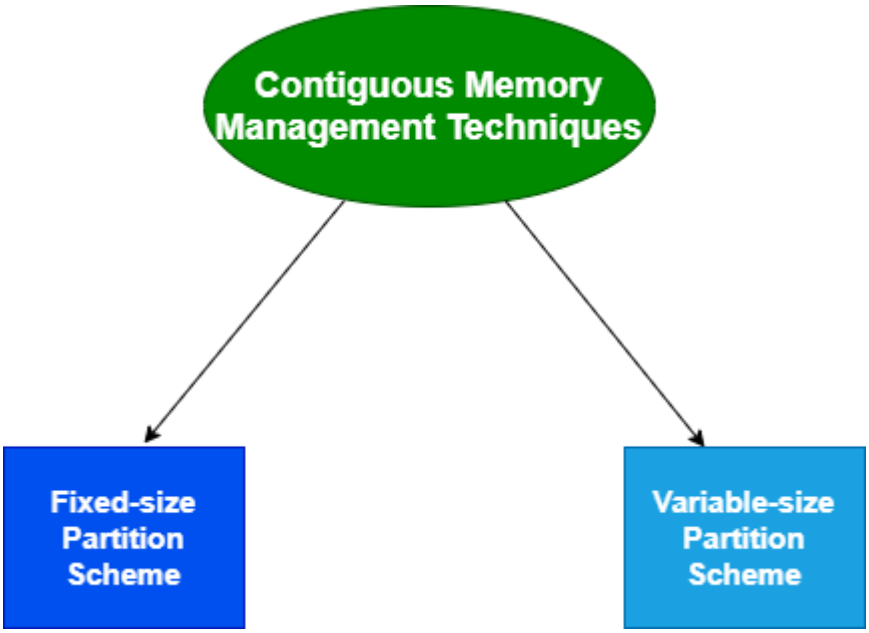
- Swap out means **to take the program from RAM and to bring them in Hard disk**.
- Swap in means to take the program from Hard disk and again bring them to the RAM.

The purpose of the swapping in operating system is **to access the data present in the hard disk and bring it to RAM so that the application programs can use it**. The thing to remember is that swapping is used only when data is not present in RAM.

CONTIGUOUS MEMORY ALLOCATION:

In the Contiguous Memory Allocation, each process is contained in a **single contiguous section of memory**. In this memory allocation, all the available memory space remains together in one place which implies that the freely available memory partitions are not spread over here and there across

the whole memory space.



FIXED SIZED PARTITION

The memory is partitioned into pre-determined, fixed-size memory chunks. All the user processes are allocated in one partition.

Partitioning the memory into fixed sizes allows you fast access to data in memory and smoother memory management. However, this efficiency is achieved at the cost of internal fragmentation.

VARIABLE SIZED PARTITION

Instead of the memory being divided into fixed sizes, it is partitioned into consecutive, but variable-sized memory chunks for the user processes.

Although having memory partitioned helps you counter internal fragmentation, it also results in sluggish access to data in the memory and complicated memory management.

Video Content / Details of website for further learning (if any):
http://www.cs.nthu.edu.tw/~ychung/slides/CSC3150/Abraham-Silberschatz-Operating-System-Concepts---9th2012.12

Important Books/Journals for further learning including the page nos.:
Book: Abraham Silberschatz,Peter Bear Galvin and Greg Gagne, ," Operating system concepts",2015, John Wiley & Sons (ASIA) ,9th Edition,Page no(351-466**).**

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

L20

LECTURE HANDOUTS

AI&DS

II/III

Course Name with Code : **19ADCO7&OPERATING SYSTEM**

Course Faculty : **R.NIVETHITHA.**

Unit : **III MEMORY MANAGEMENT**

Date of Lecture:

Topic of Lecture: Segmentation – Paging, structure of the page table

Introduction :

- Most users do not think memory as a linear array of bytes rather the users thinks memory as a collection of variable sized segments which are dedicated to a particular use such as code, data, stack, heap etc.
- A logical address is a collection of segments. Each segment has a name and length. The address specifies both the segment name and the offset within the segments.
- The users specify address by using two quantities: a segment name and an offset. For simplicity the segments are numbered and referred by a segment number

Prerequisite knowledge for Complete understanding and learning of Topic:

- Operating Systems
- Hardware
- Software
- Process

Detailed content of the Lecture:

SEGMENTATION:

Segmentation is **a memory management technique in which the memory is divided into the variable size parts**. Each part is known as a segment which can be allocated to a process. The details about each segment are stored in a table called a segment table.

PAGING:

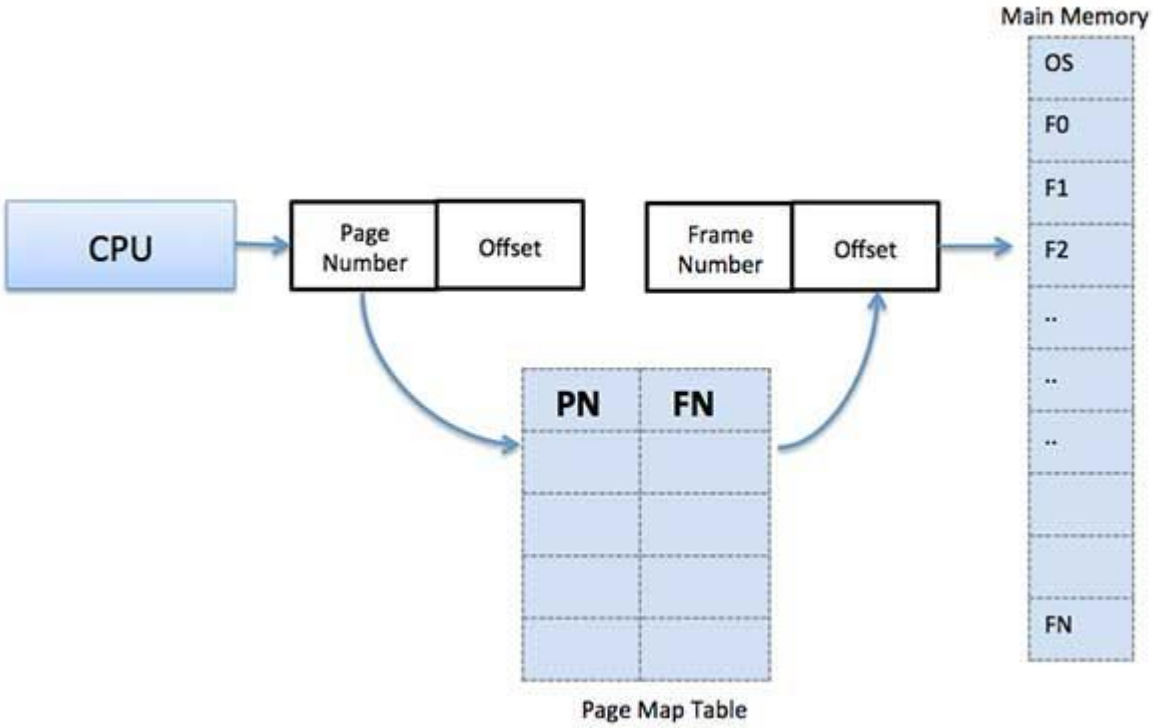
A computer can address more memory than the amount physically installed on the system. This extra memory is actually called virtual memory and it is a section of a hard that's set up to emulate the computer's RAM. Paging technique plays an important role in implementing virtual memory.

Paging is a memory management technique in which process address space is broken into blocks of the same size called **pages** (size is power of 2, between 512 bytes and 8192 bytes). The size of the process is measured in the number of pages.

Similarly, main memory is divided into small fixed-sized blocks of (physical) memory called **frames** and the size of a frame is kept the same as that of a page to have optimum utilization of the main memory and to avoid external fragmentation.

STRUCTURE OF PAGE TABE:

A data structure called **page map table** is used to keep track of the relation between a page of a process to a frame in physical memory.



When the system allocates a frame to any page, it translates this logical address into a physical address and create entry into the page table to be used throughout execution of the program.

When a process is to be executed, its corresponding pages are loaded into any available memory frames. Suppose you have a program of 8Kb but your memory can accommodate only 5Kb at a given point in time, then the paging concept will come into picture. When a computer runs out of RAM, the operating system (OS) will move idle or unwanted pages of memory to secondary memory to free up RAM for other processes and brings them back when needed by the program

Video Content / Details of website for further learning (if any):

http://www.cs.nthu.edu.tw/~ychung/slides/CSC3150/Abraham-Silberschatz-Operating-System-Concepts---9th2012.12

https://www.tutorialspoint.com/operating_system/os_memory_management.htm

Important Books/Journals for further learning including the page nos.:

Book:

Abraham Silberschatz,Peter Bear Galvin and Greg Gagne, ," Operating system concepts",2015, John Wiley & Sons (ASIA) ,9th Edition,Page no(351-466).

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

L21

LECTURE HANDOUTS

AI&DS

II/III

Course Name with Code : **19ADCO7&OPERATING SYSTEM**

Course Faculty : **R.NIVETHITHA.**

Unit : **III MEMORY MANAGEMENT**

Date of Lecture:

Topic of Lecture: Virtual memory: demand paging, page-replacement

Introduction :

- A computer can address more memory than the amount physically installed on the system. This extra memory is actually called **virtual memory** and it is a section of a hard disk that's set up to emulate the computer's RAM.
- The main visible advantage of this scheme is that programs can be larger than physical memory.
- The basic idea behind demand paging is that when a process is swapped in, its pages are not swapped in all at once.
- Rather they are swapped in only when the process needs them. (on demand.) This is termed a lazy swapper, although a pager is a more accurate term.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Operating Systems
- Hardware
- Software
- Process

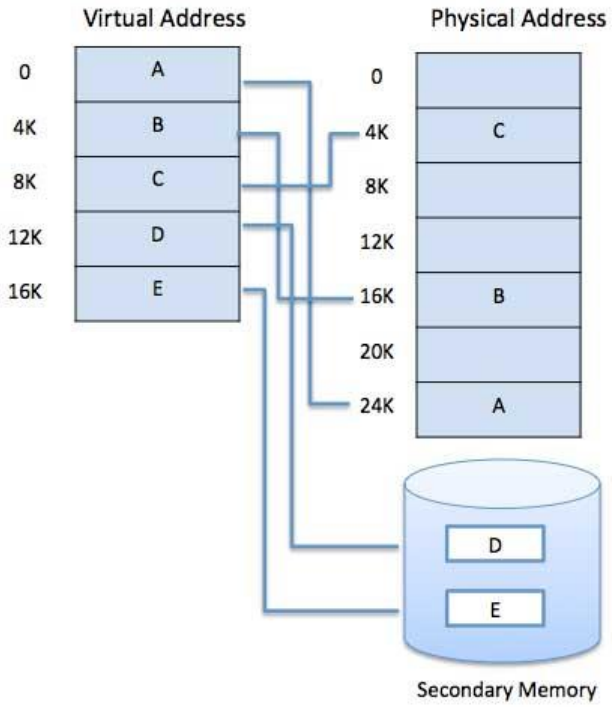
Detailed content of the Lecture:

VIRTUAL MEMORY:

The ability to load only the portions of processes that were actually needed (and only when they were needed) has several benefits:

o Programs could be written for a much larger address space (virtual memory space) than physically exists on the computer.
o Because each process is only using a fraction of their total address space, there is more memory left for other programs, improving CPU utilization and system throughput.
o Less I/O is needed for swapping processes in and out of RAM, speeding things up.

Figure below shows the general layout of **virtual memory**, which can be much larger than physical memory:



DEMAND PAGING:

A demand paging system is quite similar to a paging system with swapping where processes reside in secondary memory and pages are loaded only on demand, not in advance. When a context switch occurs, the operating system does not copy any of the old program's pages out to the disk or any of the new program's pages into the main memory Instead, it just begins executing the new program after loading the first page and fetches that program's pages as they are referenced.

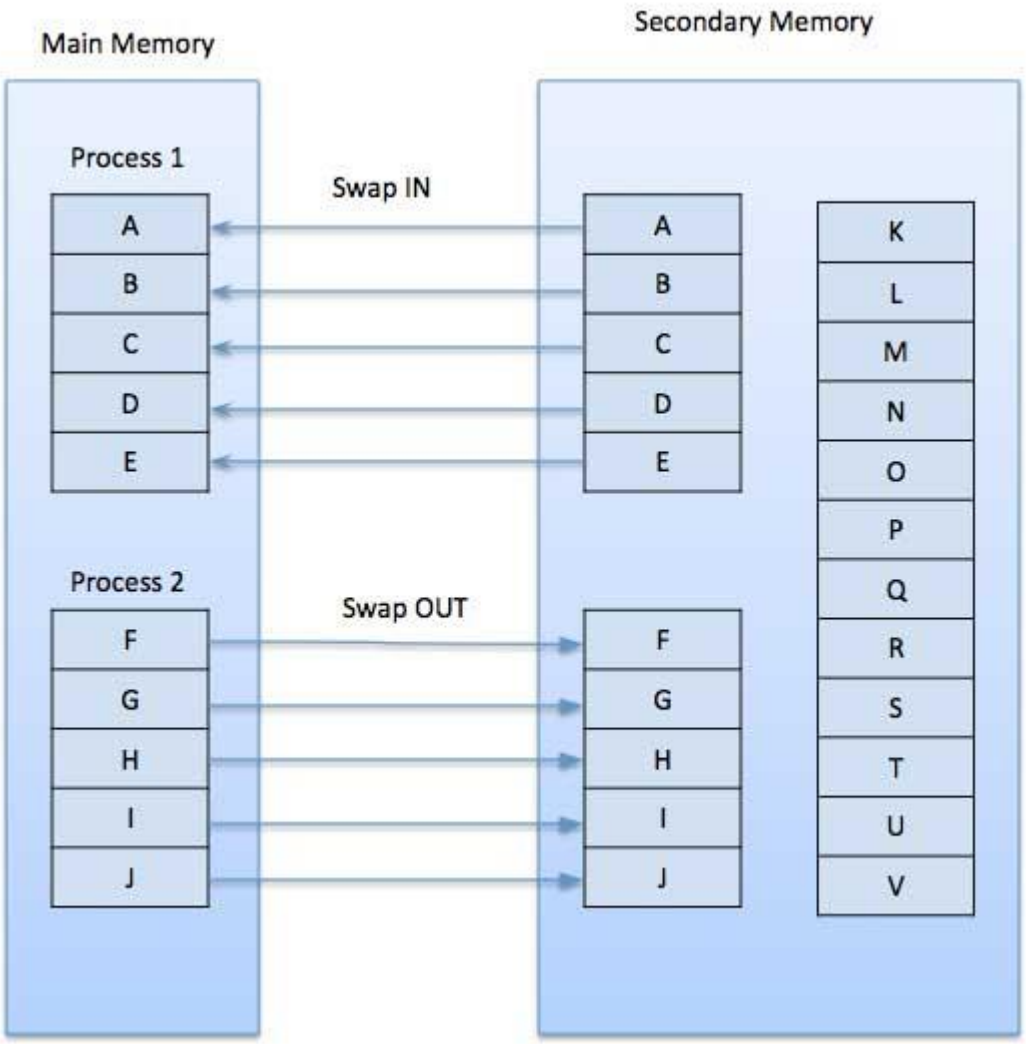
While executing a program, if the program references a page which is not available in the main memory because it was swapped out a little ago, the processor treats this invalid memory reference as a **page fault** and transfers control from the program to the operating system to demand the page back into the memory.

PAGE REPLACEMENT ALGORITHM:

Page replacement algorithms are the techniques using which an Operating System decides which memory pages to swap out, write to disk when a page of memory needs to be allocated. Paging happens whenever a page fault occurs and a free page cannot be used for allocation purpose accounting to reason that pages are not available or the number of free pages is lower than required pages.

When the page that was selected for replacement and was paged out, is referenced again, it has to read in from disk, and this requires for I/O completion. This process determines the quality of the page replacement algorithm: the lesser the time waiting for page-ins, the better is the algorithm.

A page replacement algorithm looks at the limited information about accessing the pages provided by hardware, and tries to select which pages should be replaced to minimize the total number of page misses, while balancing it with the costs of primary storage and processor time of the algorithm itself. There are many different page replacement algorithms. We evaluate an algorithm by running it on a particular string of memory reference and computing the number of page faults,



Video Content / Details of website for further learning (if any):
http://www.cs.nthu.edu.tw/~ychung/slides/CSC3150/Abraham-Silberschatz-Operating-System-Concepts---9th2012.12

Important Books/Journals for further learning including the page nos.:
Book:
Abraham Silberschatz,Peter Bear Galvin and Greg Gagne, ," Operating system concepts",2015, John Wiley & Sons (ASIA) ,9th Edition,Page no(351-466).

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

L22

LECTURE HANDOUTS

AI&DS

II/III

Course Name with Code : **19ADCO7&OPERATING SYSTEM**

Course Faculty : **R.NIVETHITHA.**

Unit : **III MEMORY MANAGEMENT**

Date of Lecture:

Topic of Lecture: Allocation of Frames, Thrashing, Memory Mapped Files

Introduction :

- An important aspect of operating systems, virtual memory is implemented using demand paging. Demand paging necessitates the development of a page-replacement algorithm and a **frame allocation algorithm**.
- Frame allocation algorithms are used if you have multiple processes; it helps decide how many frames to allocate to each process.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Operating Systems
- Hardware
- Software
- Process

Detailed content of the Lecture:

EQUAL ALLOCATION:

In a system with x frames and y processes, each process gets equal number of frames, i.e. x/y. For instance, if the system has 48 frames and 9 processes, each process will get 5 frames. The three frames which are not allocated to any process can be used as a free-frame buffer pool.

PROPORTIONAL ALLOCATION:

Frames are allocated to each process according to the process size.

For a process p_i of size s_i, the number of allocated frames is $\mathbf{a_i = (s_i/S)*m}$, where S is the sum of the sizes of all the processes and m is the number of frames in the system. For instance, in a system with 62 frames, if there is a process of 10KB and another process of 127KB, then the first process will be allocated (10/137)*62 = 4 frames and the other process will get (127/137)*62 = 57 frames.

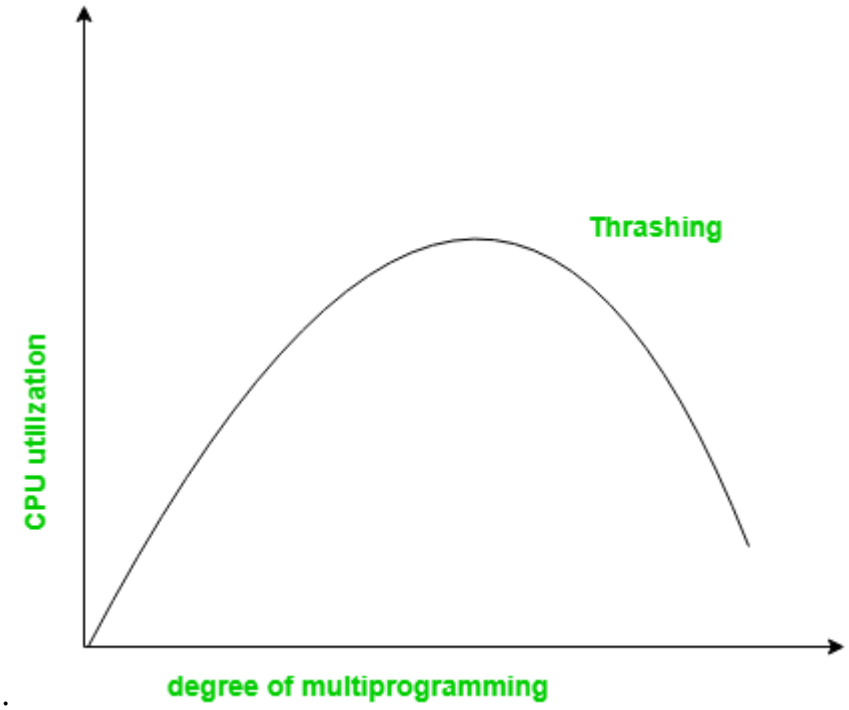
ADVANTAGE:

All the processes share the available frames according to their needs, rather than equally.

THRASHING:

Thrashing is **when the page fault and swapping happens very frequently at a higher rate**, and then the operating system has to spend more time swapping these pages. This state in the operating

system is known as thrashing. Because of thrashing, the CPU utilization is going to be reduced or negligible.
During thrashing, the CPU spends less time on some actual productive work spend more time swapping.



CAUSES OF THRASHING:

Thrashing affects the performance of execution in the Operating system. Also, thrashing results in severe performance problems in the Operating system. When the utilization of CPU is low, then the process scheduling mechanism tries to load many processes into the memory at the same time due to which degree of Multiprogramming can be increased. Now in this situation, there are more processes in the memory as compared to the available number of frames in the memory. Allocation of the limited amount of frames to each process.

EFFECT OF THRASHING

At the time, when thrashing starts then the operating system tries to apply either the **Global page replacement** Algorithm or the **Local page replacement** algorithm.

Video Content / Details of website for further learning (if any):
http://www.cs.nthu.edu.tw/~ychung/slides/CSC3150/Abraham-Silberschatz-Operating-System-Concepts---9th2012.12
https://www.studytonight.com/operating-system/thrashing-in-operating-system

Important Books/Journals for further learning including the page nos.:
Book:
Abraham Silberschatz,Peter Bear Galvin and Greg Gagne, ," Operating system concepts",2015, John Wiley & Sons (ASIA) ,9th Edition,Page no(351-466).

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

L23

LECTURE HANDOUTS

AI&DS **II/III**

Course Name with Code : **19ADCO7&OPERATING SYSTEM**

Course Faculty : **R.NIVETHITHA.**

Unit : **III MEMORY MANAGEMENT**

Date of Lecture:

Topic of Lecture: Allocating Kernel Memory Deadlocks: System Model, Deadlock Characterization

Introduction :

- When a process running in user mode requests additional memory, pages are allocated from the list of free page frames maintained by the kernel. This list is typically populated using a page-replacement algorithm
- The kernel requests memory for data structures of varying sizes, some of which are less than a page in size
- A deadlock occurs when a set of processes is stalled because each process is holding a resource and waiting for another process to acquire another resource.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Operating Systems
- Hardware
- Software
- Process

Detailed content of the Lecture:

KERNEL MEMORY ALLOCATION:

The Solaris kernel memory allocator **distributes chunks of memory for use by entities inside the kernel**. The allocator creates a number of caches of varying size for use by its clients. use of the flags incurs additional overhead and memory usage during system operations.

When a process running in user mode requests additional memory, pages are allocated from the list of free page frames maintained by the kernel. This list is typically populated using a page-replacement algorithm such as those discussed in Section 9.4 and most likely contains free pages scattered throughout physical memory, as explained earlier. Remember, too, that if a user process requests a single byte of memory, internal fragmentation will result, as the process will be granted, an entire page frame. Kernel memory, however, is often allocated from a free-memory pool different from the list used to satisfy ordinary user-mode processes. There are two primary reasons for this:

1. The kernel requests memory for data structures of varying sizes, some of which are less than a page in size. As a result, the kernel must use memory conservatively and attempt to minimize waste due to

fragmentation. This is especially important because many operating systems do not subject kernel code or data to the paging system.

2. Pages allocated to user-mode processes do not necessarily have to be in contiguous physical memory. However, certain hardware devices interact directly with physical memory—-without the benefit of a virtual memory interface—and consequently may require memory residing in physically contiguous pages. In the following sections, we examine two strategies for managing free memory that is assigned to kernel processes.

DEADLOCK SYSTEM MODEL:

A deadlock occurs when a set of processes is stalled because each process is holding a resource and waiting for another process to acquire another resource. In the diagram below, for example, Process 1 is holding Resource 1 while Process 2 acquires Resource 2, and Process 2 is waiting for Resource 1.

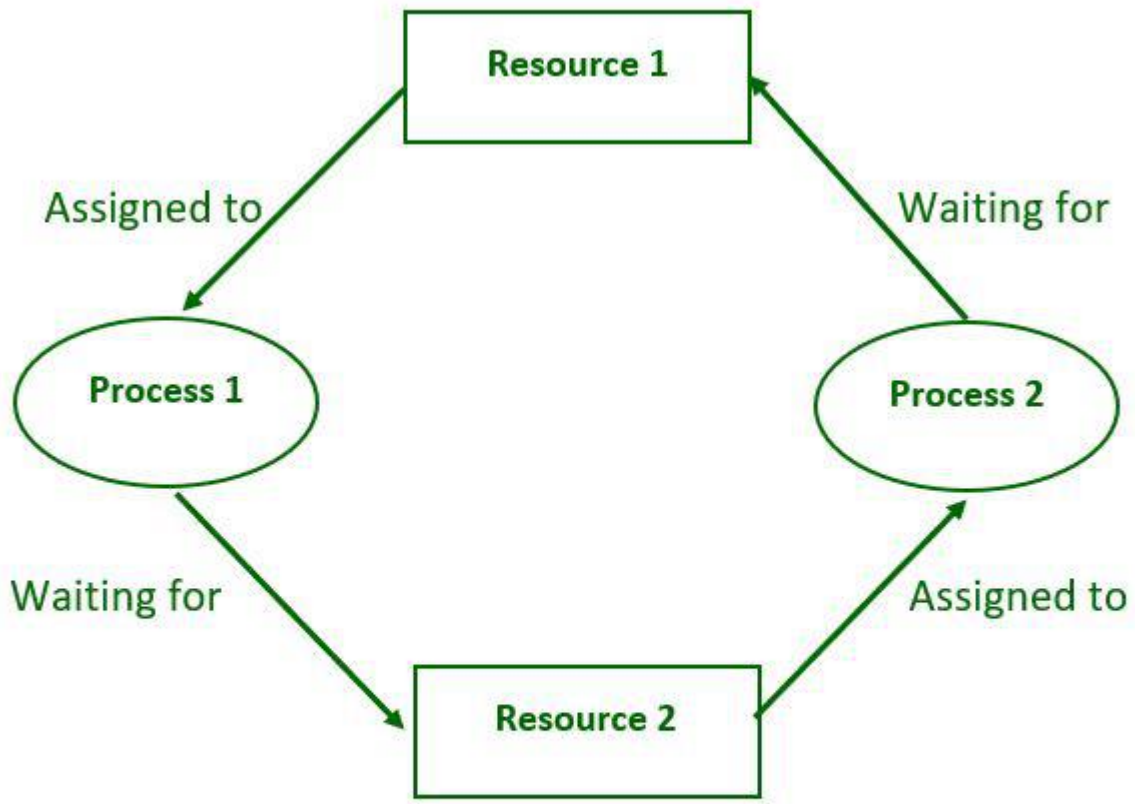


Figure: Deadlock in Operating system

SYSTEM MODEL :

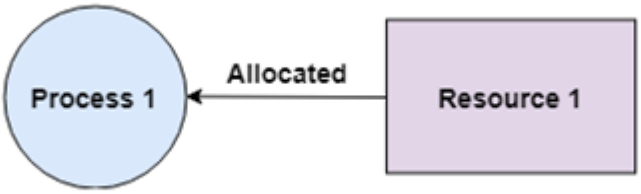
- For the purposes of deadlock discussion, a system can be modeled as a collection of limited resources that can be divided into different categories and allocated to a variety of processes, each with different requirements.
- Memory, printers, CPUs, open files, tape drives, CD-ROMs, and other resources are examples of resource categories.
- By definition, all resources within a category are equivalent, and any of the resources within that category can equally satisfy a request from that category. If this is not the case (i.e. if there is some difference between the resources within a category), then that category must be subdivided further. For example, the term “printers” may need to be subdivided into “laser printers” and “color inkjet printers.”

DEADLOCK CHARACTERIZATION:

- A deadlock happens in operating system when two or more processes need some resource to complete their execution that is held by the other process.
- A deadlock occurs if the four Coffman conditions hold true. But these conditions are not mutually exclusive. They are given as follows

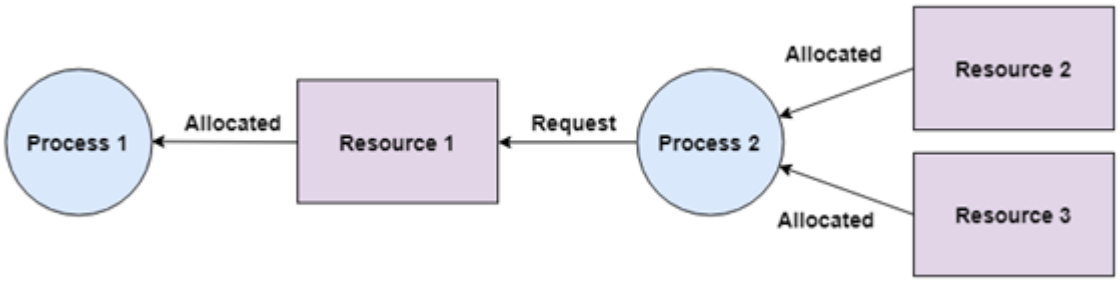
Mutual Exclusion

There should be a resource that can only be held by one process at a time. In the diagram below, there is a single instance of Resource 1 and it is held by Process 1 only.



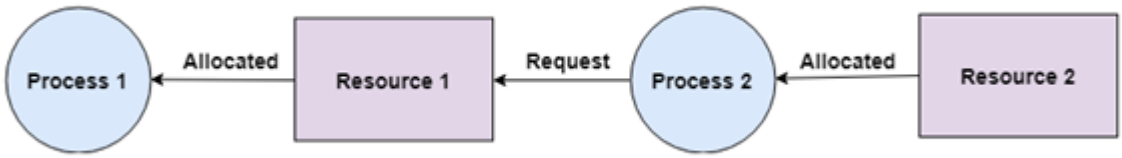
Hold and Wait

A process can hold multiple resources and still request more resources from other processes which are holding them. In the diagram given below, Process 2 holds Resource 2 and Resource 3 and is requesting the Resource 1 which is held by Process 1.



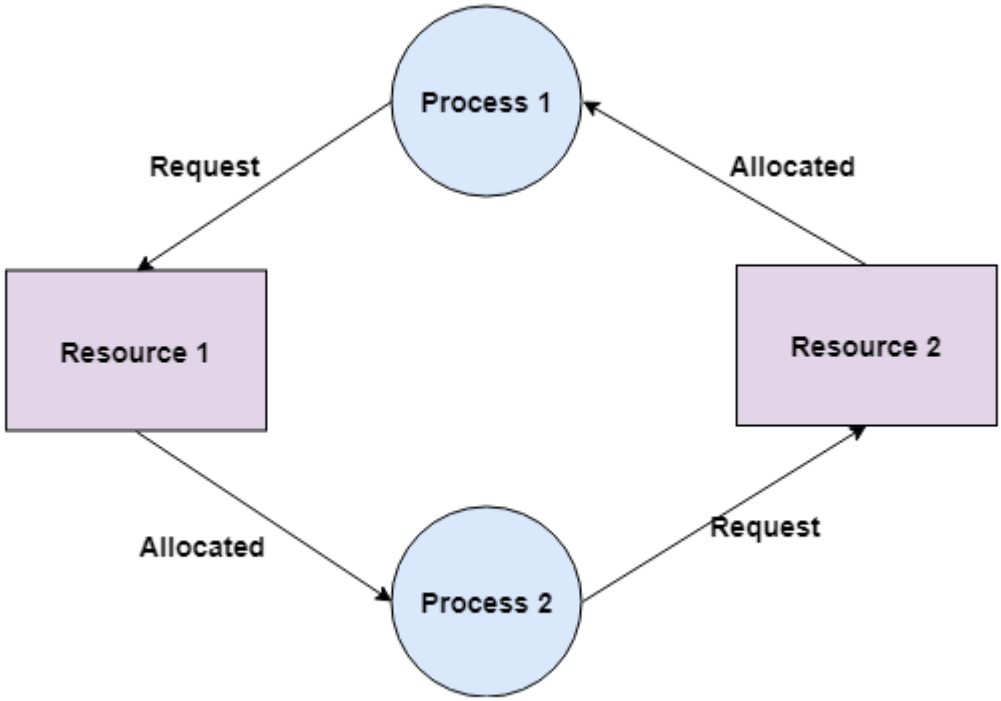
No Preemption

A resource cannot be preempted from a process by force. A process can only release a resource voluntarily. In the diagram below, Process 2 cannot preempt Resource 1 from Process 1. It will only be released when Process 1 relinquishes it voluntarily after its execution is complete.



Circular Wait

A process is waiting for the resource held by the second process, which is waiting for the resource held by the third process and so on, till the last process is waiting for a resource held by the first process. This forms a circular chain. For example: Process 1 is allocated Resource2 and it is requesting Resource 1. Similarly, Process 2 is allocated Resource 1 and it is requesting Resource 2. This forms a circular wait loop.



Video Content / Details of website for further learning (if any):
http://www.cs.nthu.edu.tw/~ychung/slides/CSC3150/Abraham-Silberschatz-Operating-System-Concepts---9th2012.12
https://padakuu.com/article/69-allocating-kernel-memory
https://www.geeksforgeeks.org/deadlock-system-model/

Important Books/Journals for further learning including the page nos.:
Book:
Abraham Silberschatz,Peter Bear Galvin and Greg Gagne, ," Operating system concepts",2015, John Wiley & Sons (ASIA) ,9th Edition,Page no(351-466).

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

L24

LECTURE HANDOUTS

AI&DS

II/III

Course Name with Code : **19ADCO7&OPERATING SYSTEM**

Course Faculty : **R.NIVETHITHA.**

Unit : **III MEMORY MANAGEMENT**

Date of Lecture:

Topic of Lecture: Methods for Handling Deadlocks

Introduction :

- Deadlock detection, deadlock prevention and deadlock avoidance are the main methods for handling deadlocks. Details about these are given as follows −

Prerequisite knowledge for Complete understanding and learning of Topic:

- Operating Systems
- Hardware
- Software
- Process

Detailed content of the Lecture:

There are mainly four methods for handling deadlock.

- Deadlock ignorance. It is the most popular method and it acts as if no deadlock and the user will restart.
- Deadlock prevention. It means that we design such a system where there is no chance of having a deadlock.
- Deadlock avoidance.
- Detection and recovery.

Deadlock detection, deadlock prevention and deadlock avoidance are the main methods for handling deadlocks. Details about these are given as follows

DEADLOCK DETECTION:

Deadlock can be detected by the resource scheduler as it keeps track of all the resources that are allocated to different processes. After a deadlock is detected, it can be handed using the given methods

- All the processes that are involved in the deadlock are terminated. This approach is not that useful as all the progress made by the processes is destroyed.
- Resources can be preempted from some processes and given to others until the deadlock situation is resolved.

DEADLOCK PREVENTION:

It is important to prevent a deadlock before it can occur. So, the system checks each transaction before it is executed to make sure it does not lead to deadlock. If there is even a slight possibility that a transaction may lead to deadlock, it is never allowed to execute.

Some deadlock prevention schemes that use timestamps in order to make sure that a deadlock does not occur are given as follows

Wait - Die Scheme

- In the wait - die scheme, if a transaction T1 requests for a resource that is held by transaction T2, one of the following two scenarios may occur −
    - TS(T1) < TS(T2) - If T1 is older than T2 i.e T1 came in the system earlier than T2, then it is allowed to wait for the resource which will be free when T2 has completed its execution.
    - TS(T1) > TS(T2) - If T1 is younger than T2 i.e T1 came in the system after T2, then T1 is killed. It is restarted later with the same timestamp.

Wound - Wait Scheme

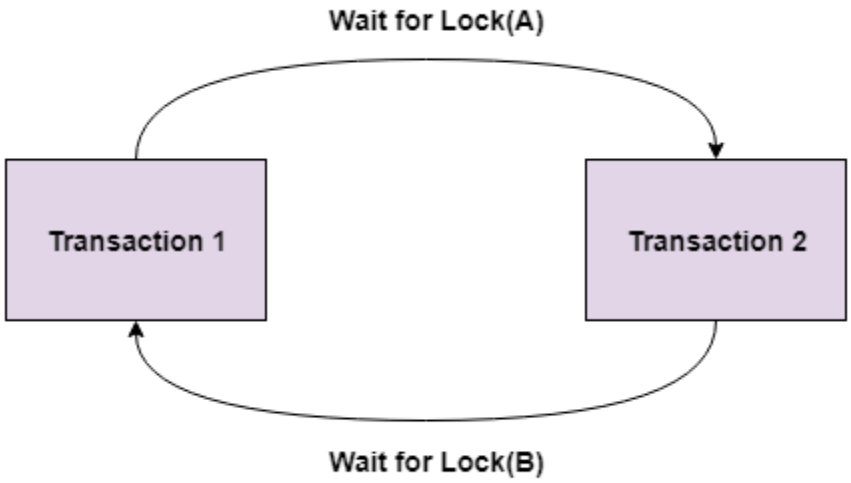
- In the wound - wait scheme, if a transaction T1 requests for a resource that is held by transaction T2, one of the following two possibilities may occur −
    - TS(T1) < TS(T2) - If T1 is older than T2 i.e T1 came in the system earlier than T2, then it is allowed to roll back T2 or wound T2. Then T1 takes the resource and completes its execution. T2 is later restarted with the same timestamp.
    - TS(T1) > TS(T2) - If T1 is younger than T2 i.e T1 came in the system after T2, then it is allowed to wait for the resource which will be free when T2 has completed its execution.

DEADLOCK AVOIDANCE:

It is better to avoid a deadlock rather than take measures after the deadlock has occurred. The wait for graph can be used for deadlock avoidance. This is however only useful for smaller databases as it can get quite complex in larger databases.

Wait for graph

The wait for graph shows the relationship between the resources and transactions. If a transaction requests a resource or if it already holds a resource, it is visible as an edge on the wait for graph. If the wait for graph contains a cycle, then there may be a deadlock in the system, otherwise not.



Ostrich Algorithm

- ✓ The ostrich algorithm means that the deadlock is simply ignored and it is assumed that it will never occur. This is done because in some systems the cost of handling the deadlock is much higher than simply ignoring it as it occurs very rarely.
- ✓ So, it is simply assumed that the deadlock will never occur and the system is rebooted if it occurs by any chance.

Video Content / Details of website for further learning (if any):
http://www.cs.nthu.edu.tw/~ychung/slides/CSC3150/Abraham-Silberschatz-Operating-System-Concepts---9th2012.12

Important Books/Journals for further learning including the page nos.:
Book:
Abraham Silberschatz,Peter Bear Galvin and Greg Gagne, ," Operating system concepts",2015, John Wiley & Sons (ASIA) ,9th Edition,**Page no**(203-260).

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

L25

LECTURE HANDOUTS

AI&DS

II/III

Course Name with Code : **19ADCO7&OPERATING SYSTEM**

Course Faculty : **R.NIVETHITHA.**

Unit : **III MEMORY MANAGEMENT**

Date of Lecture:

Topic of Lecture: Deadlock Prevention

Introduction :

Deadlock Characteristics:

As discussed in the previous post, deadlock has following characteristics.

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

Prerequisite knowledge for Complete understanding and learning of Topic:

- Operating Systems
- Hardware
- Software
- Process

Detailed content of the Lecture:

DEADLOCK CHARACTERISTICS:

As discussed in the previous post, deadlock has following characteristics.

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

DEADLOCK PREVENTION:

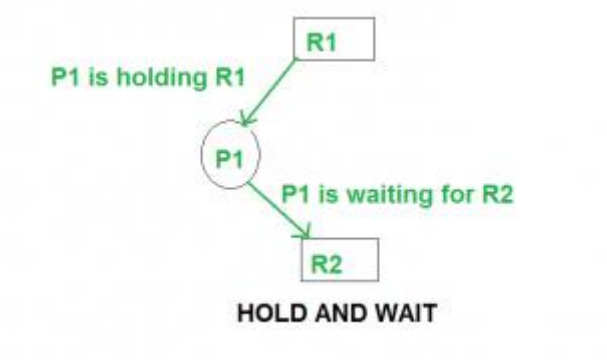
We can prevent Deadlock by eliminating any of the above four conditions.

Eliminate Mutual Exclusion:

It is not possible to dis-satisfy the mutual exclusion because some resources, such as the tape drive and printer, are inherently non-shareable.

Eliminate Hold and wait:

1. Allocate all required resources to the process before the start of its execution, this way hold and wait condition is eliminated but it will lead to low device utilization. for example, if a process requires printer at a later time and we have allocated printer before the start of its execution printer will remain blocked till it has completed its execution.
2. The process will make a new request for resources after releasing the current set of resources. This solution may lead to starvation.



Eliminate No Preemption:

- Preempt resources from the process when resources required by other high priority processes.
- **Deadlock Avoidance**
- Deadlock avoidance can be done with Banker's Algorithm.

Banker's Algorithm:

Bankers's Algorithm is resource allocation and deadlock avoidance algorithm which test all the request made by processes for resources, it checks for the safe state, if after granting request system remains in the safe state it allows the request and if there is no safe state it doesn't allow the request made by the process.

Inputs to Banker's Algorithm:

1. Max need of resources by each process.
2. Currently allocated resources by each process.
3. Max free available resources in the system.

The request will only be granted under the below condition:

1. If the request made by the process is less than equal to max need to that process.
2. If the request made by the process is less than equal to the freely available resource in the system.

Video Content / Details of website for further learning (if any):
http://www.cs.nthu.edu.tw/~ychung/slides/CSC3150/Abraham-Silberschatz-Operating-System-Concepts---9th2012.12

Important Books/Journals for further learning including the page nos.:
Book:
Abraham Silberschatz,Peter Bear Galvin and Greg Gagne, ," Operating system concepts",2015, John Wiley & Sons (ASIA) ,9th Edition,Page no(203-260).

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

L26

LECTURE HANDOUTS

AI&DS

II/III

Course Name with Code : 19ADCO7&OPERATING SYSTEM

Course Faculty : R.NIVETHITHA.

Unit : III MEMORY MANAGEMENT

Date of Lecture:

Topic of Lecture: Deadlock Detection and Avoidance

Introduction :

- ✓ In deadlock avoidance, the request for any resource will be granted if the resulting state of the system doesn't cause deadlock in the system.
- ✓ The state of the system will continuously be checked for safe and unsafe states.In order to avoid deadlocks, the process must tell OS, the maximum number of resources a process can request to complete its execution.

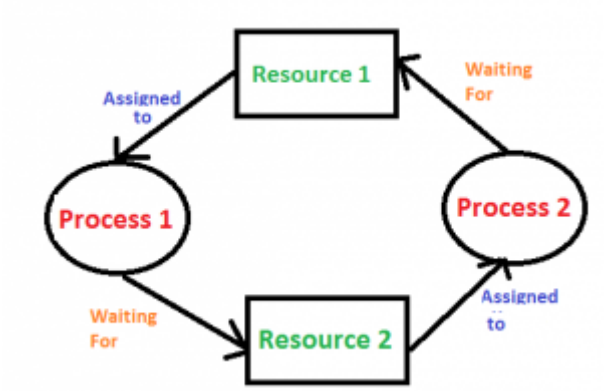
Prerequisite knowledge for Complete understanding and learning of Topic:

- Operating Systems
- Hardware
- Software
- Process

Detailed content of the Lecture:

DEADLOCK DETECTION:

1. If resources have single instance:
   In this case for Deadlock detection we can run an algorithm to check for cycle in the Resource Allocation Graph. Presence of cycle in the graph is the sufficient condition for deadlock.



   In the above diagram, resource 1 and resource 2 have single instances. There is a cycle R1 → P1 → R2 → P2. So, Deadlock is Confirmed.

2. If there are multiple instances of resources:
   Detection of the cycle is necessary but not sufficient condition for deadlock detection, in this
3. case, the system may or may not be in deadlock varies according to different situations.

DEADLOCK RECOVERY

A traditional operating system such as Windows doesn't deal with deadlock recovery as it is time and space consuming process. Real-time operating systems use Deadlock recovery.

Recovery method

- **Killing the process:** killing all the process involved in the deadlock. Killing process one by one. After killing each process check for deadlock again keep repeating the process till system recover from deadlock.

Resource Preemption: Resources are preempted from the processes involved in the deadlock, preempted resources are allocated to other processes so that there is a possibility of recovering the system from deadlock. In this case, the system goes into starvation.

DEADLOCK AVOIDANCE:

- In deadlock avoidance, the request for any resource will be granted if the resulting state of the system doesn't cause deadlock in the system. The state of the system will continuously be checked for safe and unsafe states.In order to avoid deadlocks, the process must tell OS, the maximum number of resources a process can request to complete its execution.
- The simplest and most useful approach states that the process should declare the maximum number of resources of each type it may ever need. The Deadlock avoidance algorithm examines the resource allocations so that there can never be a circular wait condition.

Safe and Unsafe States:

- The resource allocation state of a system can be defined by the instances of available and allocated resources, and the maximum instance of the resources demanded by the processes.
- Above tables and vector E, P and A describes the resource allocation state of a system. There are 4 processes and 4 types of the resources in a system. Table 1 shows the instances of each resource assigned to each process.
- The instances of the resources, each process still needs. Vector E is the representation of total instances of each resource in the system.Vector P represents the instances of resources that have been assigned to processes. Vector A represents the number of resources that are not in use.
- A state of the system is called safe if the system can allocate all the resources requested by all the processes without entering into deadlock.If the system cannot fulfill the request of all processes then the state of the system is called unsafe.

Video Content / Details of website for further learning (if any):
http://www.cs.nthu.edu.tw/~ychung/slides/CSC3150/Abraham-Silberschatz-Operating-System-Concepts---9th2012.12

Important Books/Journals for further learning including the page nos.:
Book:
Abraham Silberschatz,Peter Bear Galvin and Greg Gagne, ," Operating system concepts",2015, John Wiley & Sons (ASIA) ,9th Edition,Page no(203-260).

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

L27

LECTURE HANDOUTS

AI&DS

II/III

Course Name with Code : **19ADCO7&OPERATING SYSTEM**

Course Faculty : **R.NIVETHITHA.**

Unit : **III MEMORY MANAGEMENT**

Date of Lecture:

Topic of Lecture: Recovery from Deadlocks.

Introduction :

- When a Deadlock Detection Algorithm determines that a deadlock has occurred in the system, the system must recover from that deadlock
- Aborting all the processes will certainly break the deadlock, but with a great expense
- Deadlock recovery performs when a deadlock is detected. When deadlock detected, then our system stops working

Prerequisite knowledge for Complete understanding and learning of Topic:
(Max. Four important topics)

- Operating Systems
- Hardware
- Software
- Process

Detailed content of the Lecture:

DEADLOCK DETECTION AND RECOVERY:

Deadlock recovery performs when a deadlock is detected. When deadlock detected, then our system stops working, and after the recovery of the deadlock, our system start working again. Therefore, after the detection of deadlock, a method/way must require to recover that deadlock to run the system again. The method/way is called as deadlock recovery.

DEADLOCK RECOVERY:

A traditional operating system such as Windows doesn't deal with deadlock recovery as it is time and space consuming process. Real-time operating systems use Deadlock recovery.

Recovery method

1. **Killing the process:** killing all the process involved in the deadlock. Killing process one by one. After killing each process check for deadlock again keep repeating the process till system recover from deadlock.
2. **Resource Preemption:** Resources are preempted from the processes involved in the deadlock, preempted resources are allocated to other processes so that there is a possibility of recovering the system from deadlock. In this case, the system goes into starvation.

1. Process Termination:

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

(a). Abort all the Deadlocked Processes:

Aborting all the processes will certainly break the deadlock, but with a great expenses. The deadlocked processes may have computed for a long time and the result of those partial computations must be discarded and there is a probability to recalculate them later.

(b). Abort one process at a time untill deadlock is eliminated:

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

2. Resource Preemption:

To eliminate deadlocks using resource preemption, we preepmt some resources from processes and give those resources to other processes. This method will raise three issues –

- **(a). Selecting a victim:**
  We must determine which resources and which processes are to be preempted and also the order to minimize the cost.
- **(b). Rollback:**
  We must determine what should be done with the process from which resources are preempted. One simple idea is total rollback. That means abort the process and restart it.
- **(c). Starvation:**
  In a system, it may happen that same process is always picked as a victim. As a result, that process will never complete its designated task. This situation is called **Starvation** and must be avoided.

Video Content / Details of website for further learning (if any):

http://www.cs.nthu.edu.tw/~ychung/slides/CSC3150/Abraham-Silberschatz-Operating-System-Concepts---9th2012.12

Important Books/Journals for further learning including the page nos.:

Book:

Abraham Silberschatz,Peter Bear Galvin and Greg Gagne, ," Operating system concepts",2015, John Wiley & Sons (ASIA) ,9th Edition,Page no(203-260).

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

L28

LECTURE HANDOUTS

AI&DS

II/III

Course Name with Code : **19ADCO7&OPERATING SYSTEM**

Course Faculty : **R.NIVETHITHA.**

Unit : **IV STORAGE AND FILE SYSTEM.**

Date of Lecture:

Topic of Lecture: Mass Storage Structure, Disk structure.

Introduction :

- Each modern disk contains concentric tracks and each track is divided into multiple sectors. The disks are usually arranged as a one dimensional array of blocks, where blocks are the smallest storage unit.
- Blocks can also be called as sectors. For each surface of the disk, there is aread/write desk available. The same tracks on all the surfaces is known as a cylinder.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Operating Systems
- Hardware
- Software
- Process

Detailed content of the Lecture:

MASS STORAGE STRUCTURE:

1. Main memory is usually to small to store all needed programs and data permanently.
2. Main memory is a volatile storage device that loses its contents when power is turned off or otherwise lost.

There are two types of storage devices:-

- **Volatile Storage Device –**
  It loses its contents when the power of the device is removed.
- **Non-Volatile Storage device –**
  It does not loses its contents when the power is removed. It holds all the data when the power is removed.

Secondary Storage is used as an extension of main memory. Secondary storage devices can hold the data permanently.Storage devices consists of Registers, Cache, Main-Memory, Electronic-Disk, Magnetic-Disk, Optical-Disk, Magnetic-Tapes. Each storage system provides the basic system of storing a datum and of holding the datum until it is retrieved at a later time. All the storage devices differ in speed, cost, size and volatility. The most common Secondary-storage device

is a Magnetic-disk, which provides storage for both programs and data.

In this hierarchy all the storage devices are arranged according to speed and cost. The higher levels are expensive, but they are fast. As we move down the hierarchy, the cost per bit generally decreases, where as the access time generally increases.

The storage systems above the Electronic disk are Volatile, where as those below are Non-Volatile. An Electronic disk can be either designed to be either Volatile or Non-Volatile. During normal operation, the electronic disk stores data in a large DRAM array, which is Volatile. But many electronic disk devices contain a hidden magnetic hard disk and a battery for backup power. If external power is interrupted, the electronic disk controller copies the data from RAM to the magnetic disk.

When external power is restored, the controller copies the data back into the RAM. The design of a complete memory system must balance all the factors. It must use only as much expensive memory as necessary while providing as much inexpensive, Non-Volatile memory as possible. Caches can be installed to improve performance where a large access-time or transfer-rate disparity exists between two components.

DISK STRUCTURE:

A hard disk is a memory storage device which looks like this: **The disk is divided into tracks**. Each track is further divided into sectors. ... It is this Read-Write head that performs all the read and write operations on the disk and hence, position of the R-W head is a major concern.

Disk storage (also sometimes called **drive storage**) is a general category of storage mechanisms where data is recorded by various electronic, magnetic, optical, or mechanical changes to a surface layer of one or more rotating disks. A **disk drive** is a device implementing such a storage mechanism. Notable types are the hard disk drive (HDD) containing a non-removable disk, the floppy disk drive (FDD) and its removable floppy disk, and various optical disc drives (ODD) and associated optical disc media.

Video Content / Details of website for further learning (if any):

http://www.cs.nthu.edu.tw/~ychung/slides/CSC3150/Abraham-Silberschatz-Operating-System-Concepts---9th2012.12

https://www.geeksforgeeks.org/storage-structure-in-operating-systems/

Important Books/Journals for further learning including the page nos.:

Book:

Abraham Silberschatz,Peter Bear Galvin and Greg Gagne, ," Operating system concepts",2015, John Wiley & Sons (ASIA) ,9th Edition,Page no(467-624).

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

L29

LECTURE HANDOUTS

AI&DS

II/III

Course Name with Code : **19ADCO7&OPERATING SYSTEM**

Course Faculty : **R.NIVETHITHA.**

Unit : **IV STORAGE AND FILE SYSTEM.**

Date of Lecture:

Topic of Lecture: Disk Attachments, Disk Scheduling, Swap-space Management

Introduction :

- Hard drives are one of the slowest parts of the computer system and thus need to be accessed in an efficient manner.
- Seek Time: Seek time is the time taken to locate the disk arm to a specified track where the data is to be read or write.
- Transfer time is the time to transfer the data. It depends on the rotating speed of the disk and number of bytes to be transferred

Prerequisite knowledge for Complete understanding and learning of Topic:

- Operating Systems
- Hardware
- Software
- Process

Detailed content of the Lecture:

DISK SCHEDULING ALGORITHMS:

Disk scheduling is done by operating systems to schedule I/O requests arriving for the disk. Disk scheduling is also known as I/O scheduling.
Disk scheduling is important because:

- Multiple I/O requests may arrive by different processes and only one I/O request can be served at a time by the disk controller. Thus other I/O requests need to wait in the waiting queue and need to be scheduled.
- Two or more request may be far from each other so can result in greater disk arm movement.
- Hard drives are one of the slowest parts of the computer system and thus need to be accessed in an efficient manner.

There are many Disk Scheduling Algorithms but before discussing them let's have a quick look at some of the important terms:

- **Seek Time:**Seek time is the time taken to locate the disk arm to a specified track where the data is

to be read or write. So the disk scheduling algorithm that gives minimum average seek time is better.

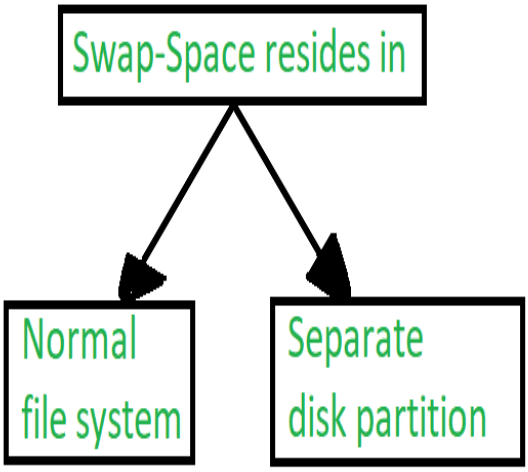
- **Rotational Latency:** Rotational Latency is the time taken by the desired sector of disk to rotate into a position so that it can access the read/write heads. So the disk scheduling algorithm that gives minimum rotational latency is better.
- **Transfer Time:** Transfer time is the time to transfer the data. It depends on the rotating speed of the disk and number of bytes to be transferred.
- **Disk Access Time:** Disk Access Time is:

Disk Access Time = Seek Time + Rotational Latency + Transfer Time

Disk Response Time: Response Time is the average of time spent by a request waiting to perform its I/O operation. *Average Response time* is the response time of the all requests. *Variance Response Time* is measure of how individual request are serviced with respect to average response time. So the disk scheduling algorithm that gives minimum variance response time is better.

SWAP-SPACE :

The area on the disk where the swapped-out processes are stored is called swap space.



SWAP-SPACE MANAGEMENT :

Swap-Swap management is another low-level task of the operating system. Disk space is used as an extension of main memory by the virtual memory. As we know the fact that disk access is much slower than memory access, In the swap-space management we are using disk space, so it will significantly decreases system performance.

Basically, in all our systems we require the best throughput, so the goal of this swap-space implementation is to provide the virtual memory the best throughput. In these article, we are going to discuss how swap space is used, where swap space is located on disk, and how swap space is managed.

System	Swap-Space
1. Solaris	Equal amount of physical memory
2. Linux	Double the amount of physical memory

Explanation of above table :

- Solaris, setting swap space equal to the amount by which virtual memory exceeds page-able physical memory.
- In the past Linux has suggested setting swap space to double the amount of physical memory.
- Today, this limitation is gone, and most Linux systems use considerably less swap space.
- Including Linux, some operating systems; allow the use of multiple swap spaces, including both files and dedicated swap partitions.
- The swap spaces are placed on the disk so the load which is on the I/O by the paging and swapping will spread over the system's bandwidth.

Video Content / Details of website for further learning (if any):
http://www.cs.nthu.edu.tw/~ychung/slides/CSC3150/Abraham-Silberschatz-Operating-System-Concepts---9th2012.12
https://www.geeksforgeeks.org/swap-space-management-in-operating-system/

Important Books/Journals for further learning including the page nos.:
Book:
Abraham Silberschatz,Peter Bear Galvin and Greg Gagne, ," Operating system concepts",2015, John Wiley & Sons (ASIA) ,9th Edition,Page no(351-466).

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

L30

LECTURE HANDOUTS

AI&DS

II/III

Course Name with Code : **19ADCO7&OPERATING SYSTEM**

Course Faculty : **R.NIVETHITHA.**

Unit : **IV STORAGE AND FILE SYSTEM.**

Date of Lecture:

Topic of Lecture: RAID Structure, Stable- storage implementation

Introduction :

- RAID or **R**edundant **A**rray of **I**ndependent **D**isks, is a technology to connect multiple secondary storage devices and use them as a single storage media.
- RAID consists of an array of disks in which multiple disks are connected together to achieve different goals. RAID levels define the use of disk arrays.
- Information residing in stable storage is never lost.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Operating Systems
- Hardware
- Software
- Process

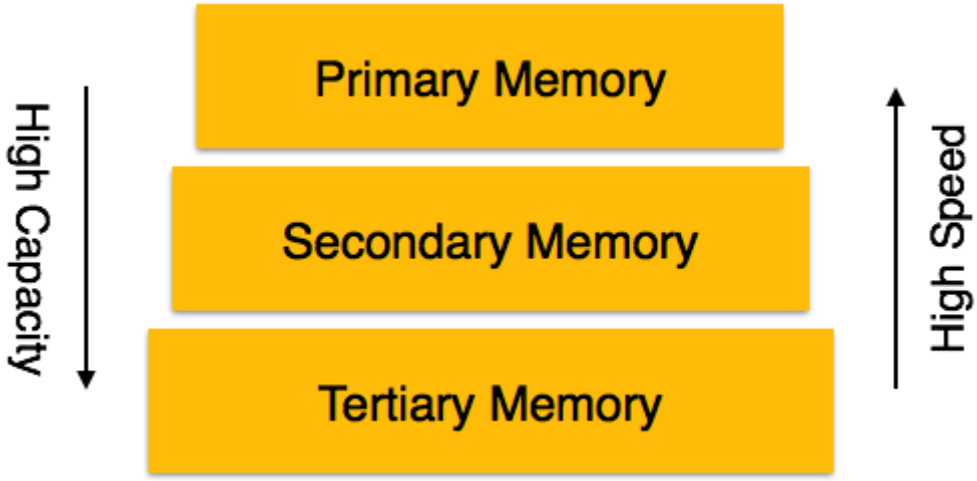
Detailed content of the Lecture:

RAID STRUCTURE:

Databases are stored in file formats, which contain records. At physical level, the actual data is stored in electromagnetic format on some device. These storage devices can be broadly categorized into three types

- **Primary Storage** − The memory storage that is directly accessible to the CPU comes under this category. CPU's internal memory (registers), fast memory (cache), and main memory (RAM) are directly accessible to the CPU, as they are all placed on the motherboard or CPU chipset. This storage is typically very small, ultra-fast, and volatile. Primary storage requires continuous power supply in order to maintain its state. In case of a power failure, all its data is

lost.



- **Secondary Storage** − Secondary storage devices are used to store data for future use or as backup. Secondary storage includes memory devices that are not a part of the CPU chipset or motherboard, for example, magnetic disks, optical disks (DVD, CD, etc.), hard disks, flash drives, and magnetic tapes.
- **Tertiary Storage** − Tertiary storage is used to store huge volumes of data. Since such storage devices are external to the computer system, they are the slowest in speed. These storage devices are mostly used to take the back up of an entire system. Optical disks and magnetic tapes are widely used as tertiary storage.

REDUNDANT ARRAY OF INDEPENDENT DISKS:

RAID or **R**edundant **A**rray of **I**ndependent **D**isks, is a technology to connect multiple secondary storage devices and use them as a single storage media.

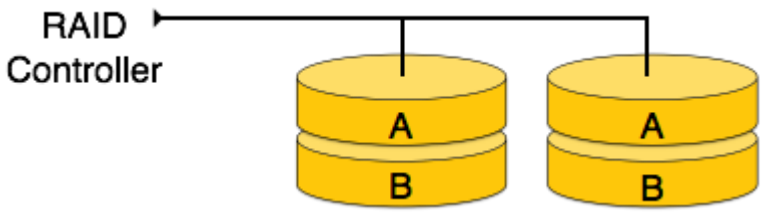
RAID consists of an array of disks in which multiple disks are connected together to achieve different goals. RAID levels define the use of disk arrays.

RAID 0

In this level, a striped array of disks is implemented. The data is broken down into blocks and the blocks are distributed among disks. Each disk receives a block of data to write/read in parallel. It enhances the speed and performance of the storage device. There is no parity and backup in Level 0.

RAID 1

RAID 1 uses mirroring techniques. When data is sent to a RAID controller, it sends a copy of data to all the disks in the array. RAID level 1 is also called **mirroring** and provides 100% redundancy in case of a failure.

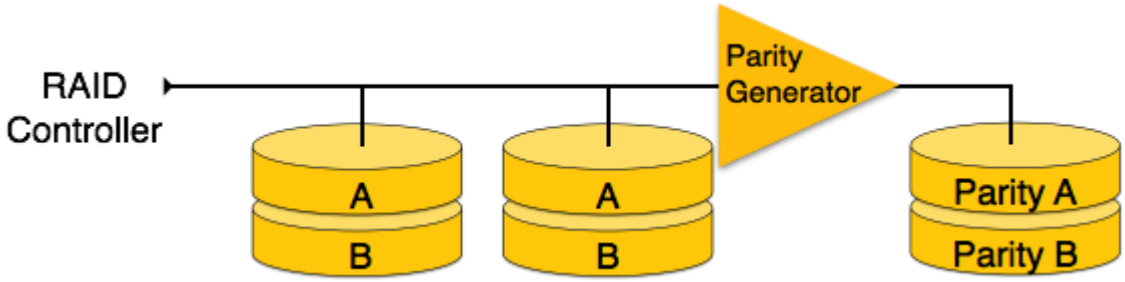


RAID 2

RAID 2 records Error Correction Code using Hamming distance for its data, striped on different disks. Like level 0, each data bit in a word is recorded on a separate disk and ECC codes of the data words are stored on a different set disks. Due to its complex structure and high cost, RAID 2 is not commercially available.

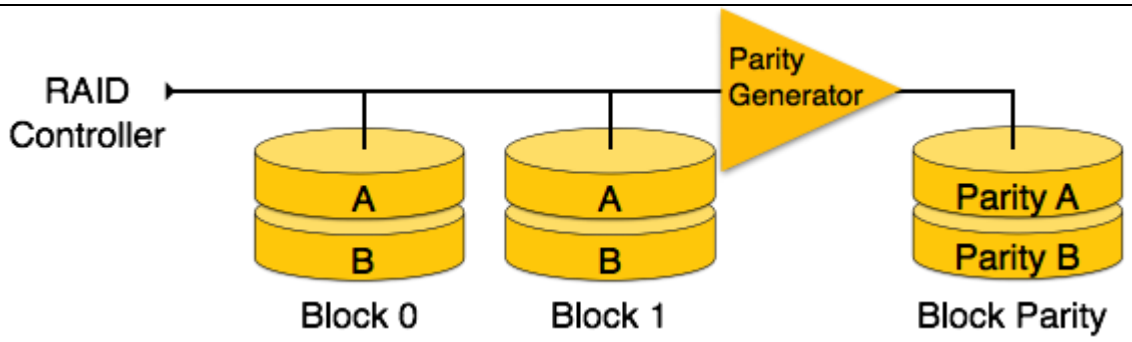
RAID 3

RAID 3 stripes the data onto multiple disks. The parity bit generated for data word is stored on a different disk. This technique makes it to overcome single disk failures.



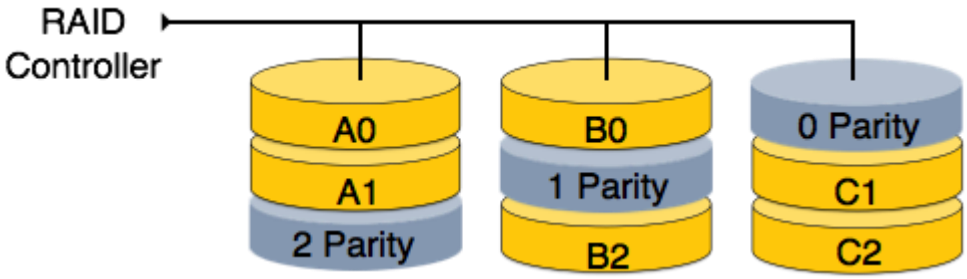
RAID 4

In this level, an entire block of data is written onto data disks and then the parity is generated and stored on a different disk. Note that level 3 uses byte-level striping, whereas level 4 uses block-level striping. Both level 3 and level 4 require at least three disks to implement RAID.



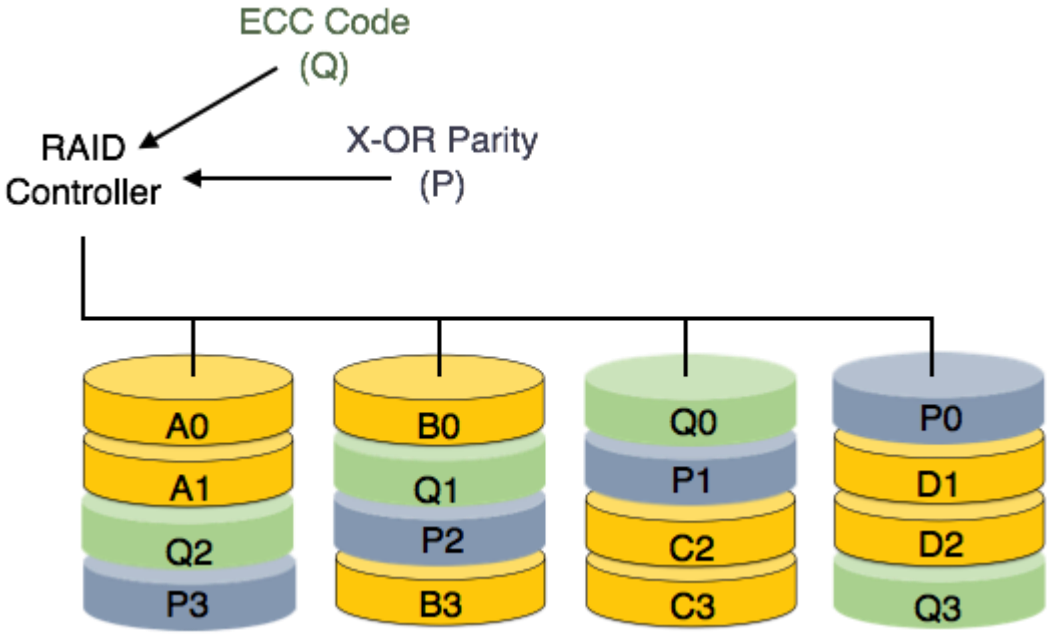
RAID 5

RAID 5 writes whole data blocks onto different disks, but the parity bits generated for data block stripe are distributed among all the data disks rather than storing them on a different dedicated disk.



RAID 6

RAID 6 is an extension of level 5. In this level, two independent parities are generated and stored in distributed fashion among multiple disks. Two parities provide additional fault tolerance. This level requires at least four disk drives to implement RAID.



STABLE-STORAGE IMPLEMENTATION:

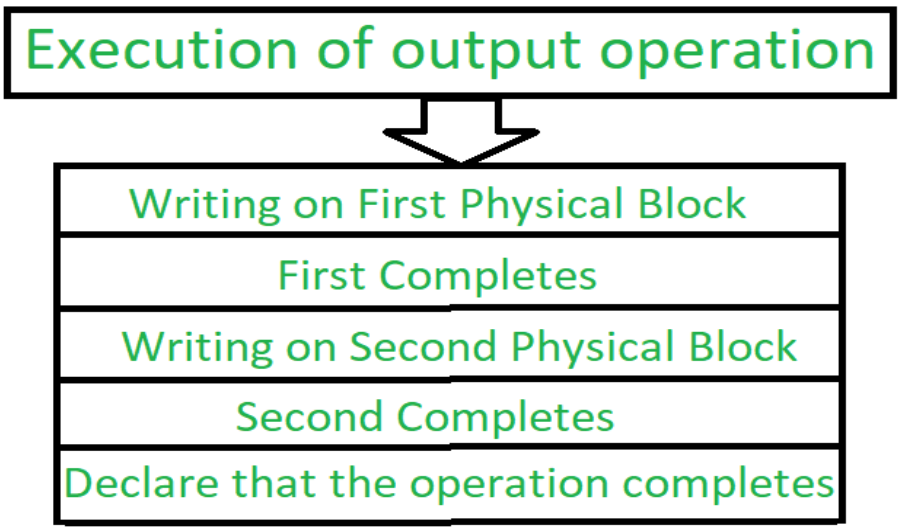
To implement stable storage:–Replicate information on more than one nonvolatile storage media with independent failure modes.–Update information in a controlled manner to ensure that we can recover thestable data after any failure during data transfer or recovery.

A disk write results in one of 3 outcomes

1) Successful completion : The data were written correctly on disk

2) Partial failure : A failure occurred in the middle of transfer so only some of the sectors were written with the new data ,and the sector being written during the failure may have been corrupted.

3) Total failure : The failure occurred before the disk writing started ,so the previous data values on the disk remain intact



Video Content / Details of website for further learning (if any):
http://www.cs.nthu.edu.tw/~ychung/slides/CSC3150/Abraham-Silberschatz-Operating-System-Concepts---9th2012.12
https://www.tutorialspoint.com/dbms/dbms_storage_system.htm

Important Books/Journals for further learning including the page nos.:
Book:
Abraham Silberschatz,Peter Bear Galvin and Greg Gagne, ," Operating system concepts",2015, John Wiley & Sons (ASIA) ,9th Edition,Page no(351-466).

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

L31

LECTURE HANDOUTS

AI&DS

II/III

Course Name with Code : **19ADCO7&OPERATING SYSTEM**

Course Faculty : **R.NIVETHITHA.**

Unit : **IV STORAGE AND FILE SYSTEM.**

Date of Lecture:

Topic of Lecture: Files system Interface: The concept of a file, Access Methods.

Introduction :

- A collection of files, that store related data, and
- A directory structure, which organizes and provides information about all the files in the system.
- the file system is the most visible aspect of an operating system
- An object file is a sequence of bytes organized into blocks that are understandable by the machine.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Operating Systems
- Hardware
- Software
- Process

Detailed content of the Lecture:

FILE SYSTEM INTERFACE:

The file system is **the most visible aspect of an operating system**. It provides the mechanism for on-line storage of and access to both data and programs of the operating system and all the users of the computer system.

File:

A file is a named collection of related information that is recorded on secondary storage such as magnetic disks, magnetic tapes and optical disks. In general, a file is a sequence of bits, bytes, lines or records whose meaning is defined by the files creator and user.

File Structure:

A File Structure should be according to a required format that the operating system can understand.

- A file has a certain defined structure according to its type.
- A text file is a sequence of characters organized into lines.
- A source file is a sequence of procedures and functions.
- An object file is a sequence of bytes organized into blocks that are understandable by the machine.
- When operating system defines different file structures, it also contains the code to support these file structure. Unix, MS-DOS support minimum number of file structure.

File Type:

File type refers to the ability of the operating system to distinguish different types of file such as text files source files and binary files etc. Many operating systems support many types of files. Operating system like MS-DOS and UNIX have the following types of files −

Ordinary files

- These are the files that contain user information.
- These may have text, databases or executable program.
- The user can apply various operations on such files like add, modify, delete or even remove the entire file.

Directory files

- These files contain list of file names and other information related to these files.

Special files

- These files are also known as device files.
- These files represent physical device like disks, terminals, printers, networks, tape drive etc.

These files are of two types −

- **Character special files** − data is handled character by character as in case of terminals or printers.
- **Block special files** − data is handled in blocks as in the case of disks and tapes.

FILE ACCESS MECHANISMS

File access mechanism refers to the manner in which the records of a file may be accessed. There are several ways to access files −

- Sequential access
- Direct/Random access
- Indexed sequential access

Sequential access

A sequential access is that in which the records are accessed in some sequence, i.e., the information in the file is processed in order, one record after the other. This access method is the most primitive one. Example: Compilers usually access files in this fashion.

Direct/Random access

- Random access file organization provides, accessing the records directly.
- Each record has its own address on the file with by the help of which it can be directly accessed for reading or writing.
- The records need not be in any sequence within the file and they need not be in adjacent locations on the storage medium.

Indexed sequential access

- This mechanism is built up on base of sequential access.
- An index is created for each file which contains pointers to various blocks.
- Index is searched sequentially and its pointer is used to access the file directly.

Video Content / Details of website for further learning (if any):
http://www.cs.nthu.edu.tw/~ychung/slides/CSC3150/Abraham-Silberschatz-Operating-System-Concepts---9th2012.12
https://www.w3schools.in/operating-system-tutorial/file-system-interface/

Important Books/Journals for further learning including the page nos.:
Book:
Abraham Silberschatz,Peter Bear Galvin and Greg Gagne, ," Operating system concepts",2015, John Wiley & Sons (ASIA) ,9th Edition,Page no(351-466).

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

L32

LECTURE HANDOUTS

AI&DS

II/III

Course Name with Code : 19ADCO7&OPERATING SYSTEM

Course Faculty : R.NIVETHITHA.

Unit : IV STORAGE AND FILE SYSTEM.

Date of Lecture:

Topic of Lecture: Directory and Disk structure

Introduction :

- A **directory** is a container that is used to contain folders and file. It organises files and folders into hierarchical manner
- Single level directory is simplest directory structure. In it all files are contained in same directory which make it easy to support and understand.
- In general graph directory structure, cycles are allowed within a directory structure where multiple directories can be derived from more than one parent directory.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Operating Systems
- Hardware
- Software
- Process

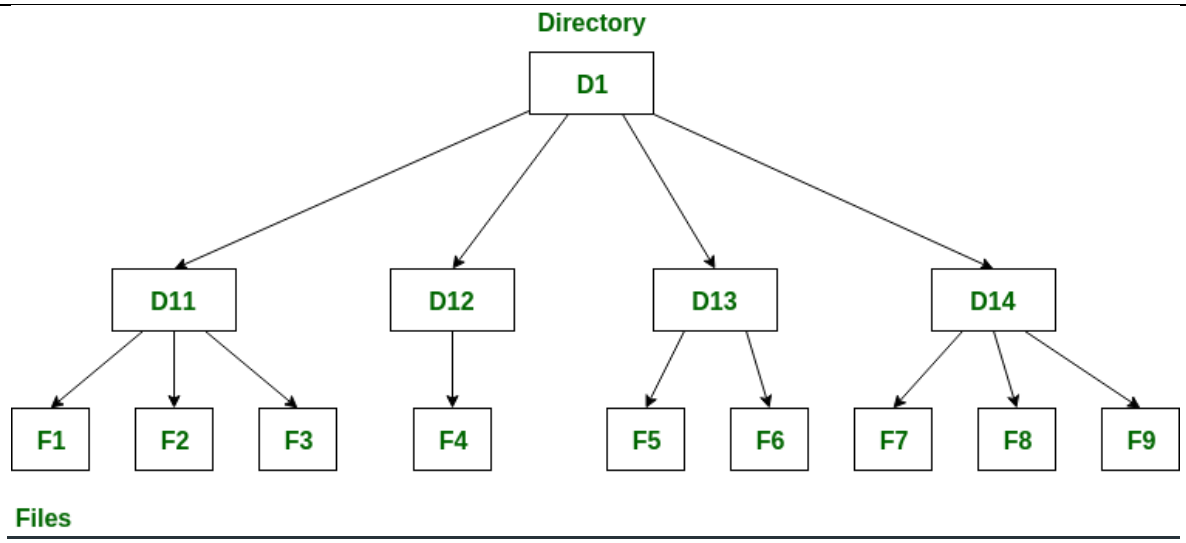
Detailed content of the Lecture:

DIRECTORY :

A **directory** is a container that is used to contain folders and file. It organises files and folders into hierarchical manner.

Operating System | Structures of Directory

- Single-level directory
- Two-level directory
- Tree-structured directory
- Acyclic graph director
- General graph directory structure

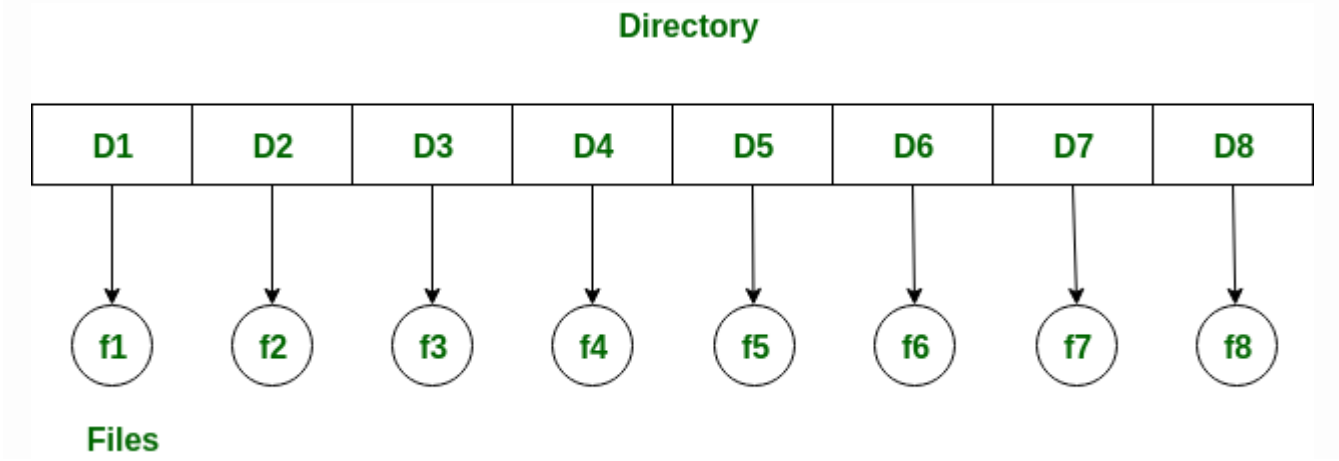


There are several logical structures of directory, these are given as below.

Single-level directory :

Single level directory is simplest disectory structure.In it all files are contained in same directory which make it easy to support and understand.

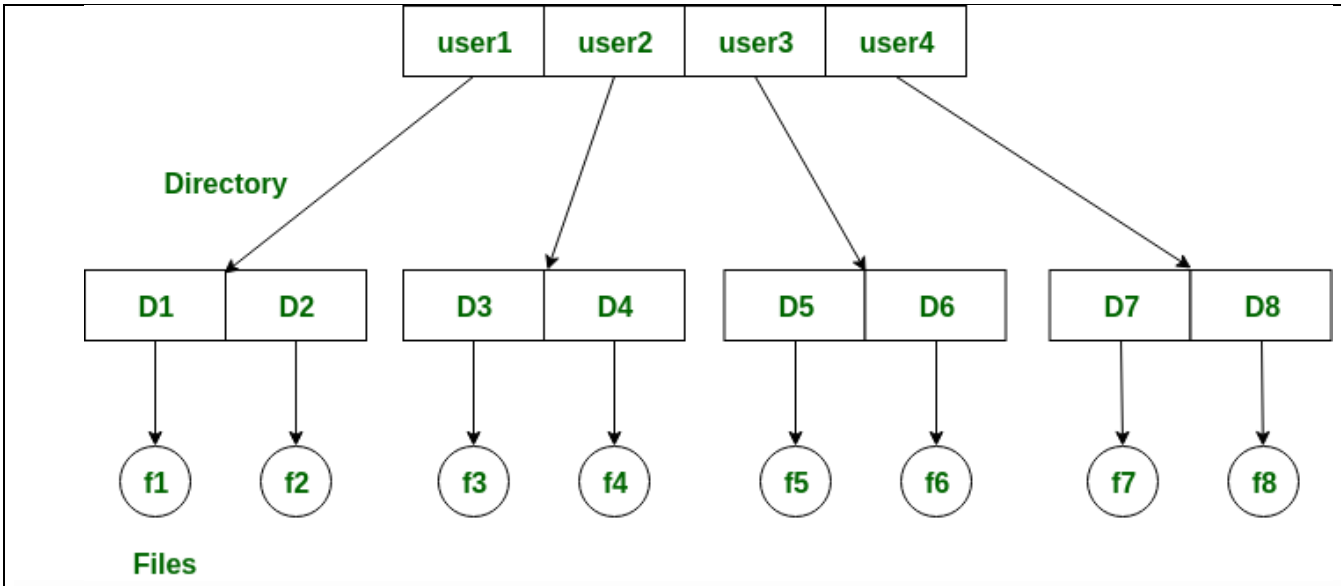
A single level directory has a significant limitation, however, when the number of files increases or when the system has more than one user. Since all the files are in the same directory, they must have the unique name . if two users call there dataset test, then the unique name rule violated.



Two-level directory :

As we have seen, a single level directory often leads to confusion of files names among different users. the solution to this problem is to create a separate directory for each user.

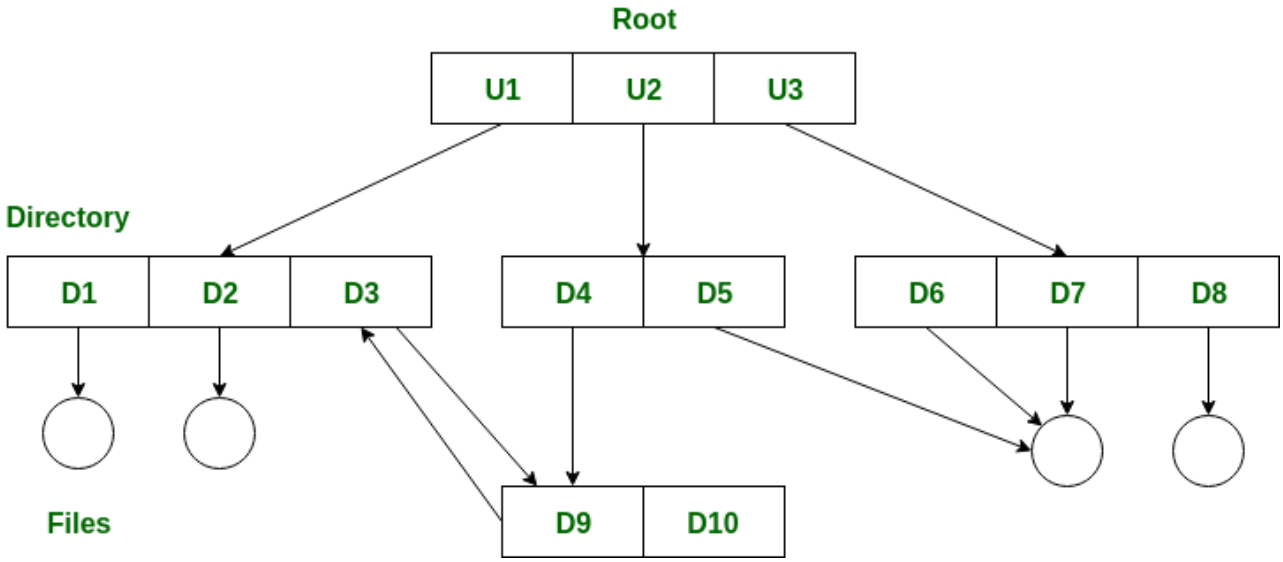
In the two-level directory structure, each user has their own *user files directory (UFD)*. The UFDs have similar structures, but each lists only the files of a single user. system's *master file directory (MFD)* is searches whenever a new user id=s logged in. The MFD is indexed by username or account number, and each entry points to the UFD for that user.



General graph directory structure:

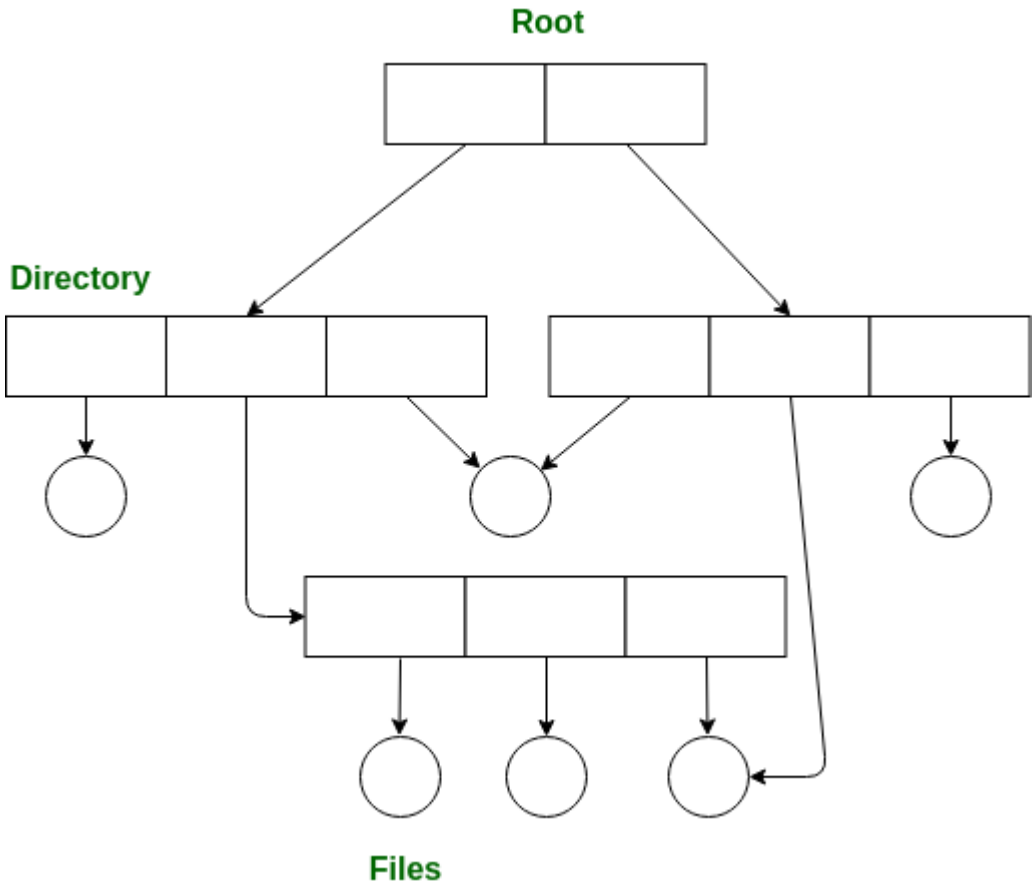
In general graph directory structure, cycles are allowed within a directory structure where multiple directories can be derived from more than one parent directory.

The main problem with this kind of directory structure is to calculate the total size or space that has been taken by the files and directories.



Acyclic graph directory:

An acyclic graph is a graph with no cycle and allows us to share subdirectories and files. The same file or subdirectories may be in two different directories. It is a natural generalization of the tree-structured directory.



Tree-structured directory:

Once we have seen a two-level directory as a tree of height 2, the natural generalization is to extend the directory structure to a tree of arbitrary height.

This generalization allows the user to create their own subdirectories and to organize their files accordingly.

DISK STRUCTURE:

Each modern disk contains concentric tracks and each track is divided into multiple sectors. The disks are usually arranged as a one dimensional array of blocks, where blocks are the smallest storage unit.Blocks can also be called as sectors. For each surface of the disk, there is a read/write desk available. The same tracks on all the surfaces is known as a cylinder.

There are various on disk data structures that are used to implement a file system. This structure may vary depending upon the operating system.

1. **Boot Control Block:**

   Boot Control Block contains all the information which is needed to boot an operating system from that volume. It is called boot block in UNIX file system. In NTFS, it is called the partition boot sector.

2. **Volume Control Block:**

   Volume control block all the information regarding that volume such as number of blocks, size of each block, partition table, pointers to free blocks and free FCB blocks. In UNIX file system, it is known as super block. In NTFS, this information is stored inside master file table.

3. **Directory Structure (per file system):**

   A directory structure (per file system) contains file names and pointers to corresponding FCBs. In UNIX, it includes inode numbers associated to file names.

4. **File Control Block:**

   File Control block contains all the details about the file such as ownership details, permission details, file size,etc. In UFS, this detail is stored in inode. In NTFS, this information is stored inside master file table as a relational database structure. A typical file control block is shown in the image below.

Video Content / Details of website for further learning (if any):
http://www.cs.nthu.edu.tw/~ychung/slides/CSC3150/Abraham-Silberschatz-Operating-System-Concepts---9th2012.12
https://www.geeksforgeeks.org/structures-of-directory-in-operating-system/

Important Books/Journals for further learning including the page nos.:
Book:
Abraham Silberschatz,Peter Bear Galvin and Greg Gagne, ," Operating system concepts",2015, John Wiley & Sons (ASIA) ,9th Edition,Page no(351-466).

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

L33

LECTURE HANDOUTS

AI&DS | **II/III**

Course Name with Code : 19ADCO7&OPERATING SYSTEM

Course Faculty : R.NIVETHITHA.

Unit : IV STORAGE AND FILE SYSTEM.

Date of Lecture:

Topic of Lecture: File system mounting, File sharing, Protection.

Introduction :

- Mounting a file system attaches that file system to a directory (mount point) and makes it available to the system
- File sharing is the practice of sharing or offering access to digital information or resources
- Centralized file hosting server installations on networks
- The files which have direct access of the any user have the need of protection.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Operating Systems
- Hardware
- Software
- Process

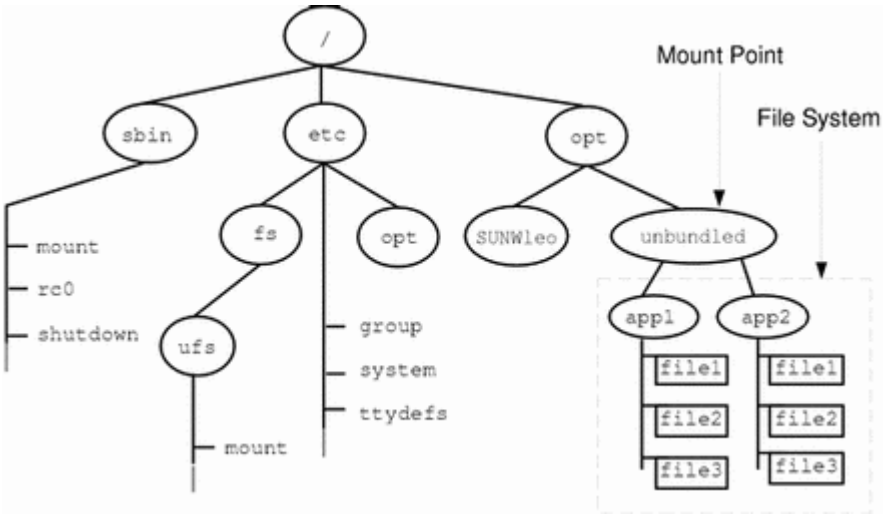
Detailed content of the Lecture:

FILE SYSTEM MOUNTING

Mounting a file system **attaches that file system to a directory (mount point) and makes it available to the system**. The root (/) file system is always mounted. Any other file system can be connected or disconnected from the root (/) file system.

Before you can access the files on a file system, you need to mount the file system. Mounting a file system attaches that file system to a directory (**mount point**) and makes it available to the system. The root (/) file system is always mounted. Any other file system can be connected or disconnected from the root (/) file system.

When you mount a file system, any files or directories in the underlying mount point directory are unavailable as long as the file system is mounted. These files are not permanently affected by the mounting process, and they become available again when the file system is unmounted. However, mount directories are typically empty, because you usually do not want to obscure existing files.



FILE SHARING:

File sharing is the practice of sharing or offering access to digital information or resources, including documents, multimedia (audio/video), graphics, computer programs, images and e-books. It is the private or public distribution of data or resources in a network with different levels of sharing privileges.

File sharing can be done using several methods. The most common techniques for file storage, distribution and transmission include the following:

- Removable storage devices
- Centralized file hosting server installations on networks
- World Wide Web-oriented hyperlinked documents
- Distributed peer-to-peer networks

PROTECTION:

In computer systems, alot of user's information is stored, the objective of the operating system is to keep safe the data of the user from the improper access to the system. Protection can be provided in number of ways. For a single laptop system, we might provide protection by locking the computer in a desk drawer or file cabinet. For multi-user systems, different mechanisms are used for the protection.

Types of Access :

The files which have direct access of the any user have the need of protection. The files which are not accessible to other users doesn't require any kind of protection. The mechanism of the protection provide the facility of the controlled access by just limiting the types of access to the file. Access can be given or not given to any user depends on several factors.

Several different types of operations can be controlled:

- **Read**
  Reading from a file.

- **Write**
  Writing or rewriting the file.
- **Execute**
  Loading the file and after loading the execution process starts.
- **Append**
  Writing the new information to the already existing file, editing must be end at the end of the existing file.
- **Delete**
  Deleting the file which is of no use and using its space for the another data.
- **List**
  List the name and attributes of the file.

Operations like renaming, editing the existing file, copying; these can also be controlled. There are many protection mechanism. each of them mechanism have different advantages and disadvantages and must be appropriate for the intended application.

Access Control :

There are different methods used by different users to access any file. The general way of protection is to associate *identity-dependent access* with all the files and directories an list called access-control list (ACL) which specify the names of the users and the types of access associate with each of the user. The main problem with the access list is their length. If we want to allow everyone to read a file, we must list all the users with the read access

Video Content / Details of website for further learning (if any):
http://www.cs.nthu.edu.tw/~ychung/slides/CSC3150/Abraham-Silberschatz-Operating-System-Concepts---9th2012.12
https://docs.oracle.com/cd/E19455-01/805-7228/6j6q7ueup/index.html

Important Books/Journals for further learning including the page nos.:
Book:
Abraham Silberschatz,Peter Bear Galvin and Greg Gagne, ," Operating system concepts",2015, John Wiley & Sons (ASIA) ,9th Edition,Page no(351-466).

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

L34

LECTURE HANDOUTS

AI&DS

II/III

Course Name with Code : **19ADCO7&OPERATING SYSTEM**

Course Faculty : **R.NIVETHITHA.**

Unit : **IV STORAGE AND FILE SYSTEM.**

Date of Lecture:

Topic of Lecture: File system Structure, File system Implementation

Introduction :

- All disk I/O is performed in units of one block (physical record) size which will exactly match the length of the desired logical record.
- Logical records may even vary in length. Packing a number of logical records into physical blocks is a common solution to this problem.
- A file is a collection of related information..
- There are different methods used by different users to access any file

Prerequisite knowledge for Complete understanding and learning of Topic:

- Operating Systems
- Hardware
- Software
- Process

Detailed content of the Lecture:

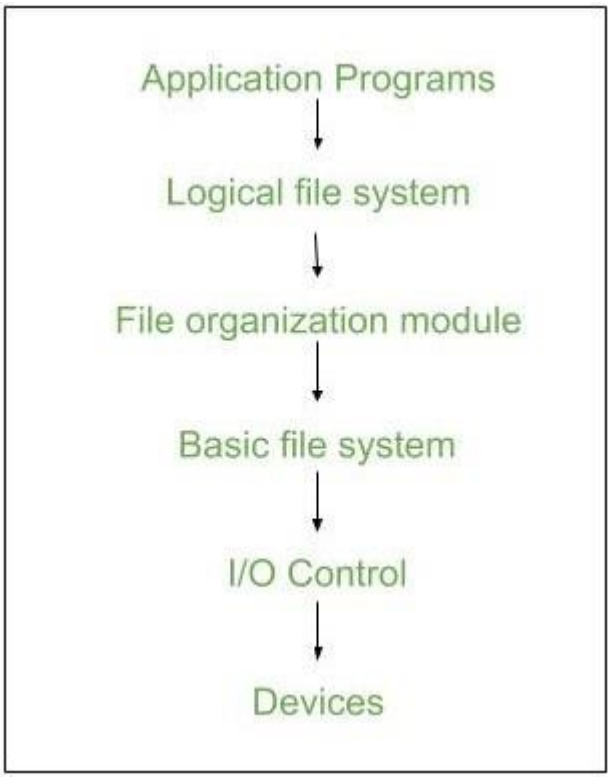
FILE SYSTEM STRUCTURE:

- All disk I/O is performed in units of one block (physical record) size which will exactly match the length of the desired logical record.
- Logical records may even vary in length. Packing a number of logical records into physical blocks is a common solution to this problem.
- For example, the UNIX operating system defines all files to be simply a tream of bytes. Each byte is individually addressable by its offset from the beginning (or end) of the file. In this case, the logical records are 1 byte. The file system automatically packs and unpacks bytes into physical disk blocks – say, 512 bytes per block – as necessary.
- The logical record size, physical block size, and packing technique determine how many logical records are in each physical block. The packing can be done either by the user's application program or by the operating system.

FILE SYSTEM IMPLEMENTATION:

A file is a collection of related information. The file system resides on secondary storage and provides efficient and convenient access to the disk by allowing data to be stored, located, and retrieved.

File system organized in many layers :



- **I/O Control level**

  Device drivers acts as interface between devices and Os, they help to transfer data between disk and main memory. It takes block number a input and as output it gives low level hardware specific instruction.
  /li>
- **Basic file system**

  It Issues general commands to device driver to read and write physical blocks on disk.It manages the memory buffers and caches. A block in buffer can hold the contents of the disk block and cache stores frequently used file system metadata.
- **File organization Module**

  It has information about files, location of files and their logical and physical blocks.Physical blocks do not match with logical numbers of logical block numbered from 0 to N. It also has a free space which tracks unallocated blocks.

- **Logical file system**

  It manages metadata information about a file i.e includes all details about a file except the actual contents of file. It also maintains via file control blocks. File control block (FCB) has information about a file – owner, size, permissions, location of file contents.

and must be appropriate for the intended application.

ACCESS CONTROL :

There are different methods used by different users to access any file. The general way of protection is to associate *identity-dependent access* with all the files and directories an list called access-control list (ACL) which specify the names of the users and the types of access associate with each of the user. The main problem with the access list is their length. If we want to allow everyone to read a file, we must list all the users with the read access

Video Content / Details of website for further learning (if any):

http://www.cs.nthu.edu.tw/~ychung/slides/CSC3150/Abraham-Silberschatz-Operating-System-Concepts---9th2012.12

https://docs.oracle.com/cd/E19455-01/805-7228/6j6q7ueup/index.html

Important Books/Journals for further learning including the page nos.:

Book:

Abraham Silberschatz,Peter Bear Galvin and Greg Gagne, ," Operating system concepts",2015, John Wiley & Sons (ASIA) ,9th Edition,Page no(351-466).

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

L35

LECTURE HANDOUTS

AI&DS | **II/III**

Course Name with Code : 19ADCO7&OPERATING SYSTEM

Course Faculty : R.NIVETHITHA.

Unit : IV STORAGE AND FILE SYSTEM.

Date of Lecture:

Topic of Lecture: Directory Implementation

Introduction :

- There is the number of algorithms by using which, the directories can be implemented. File sharing is the practice of sharing or offering access to digital information or resources
- The directory implementation algorithms are classified according to the data structure they are using.
- The files which have direct access of the any user have the need of protection.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Operating Systems
- Hardware
- Software
- Process

Detailed content of the Lecture:

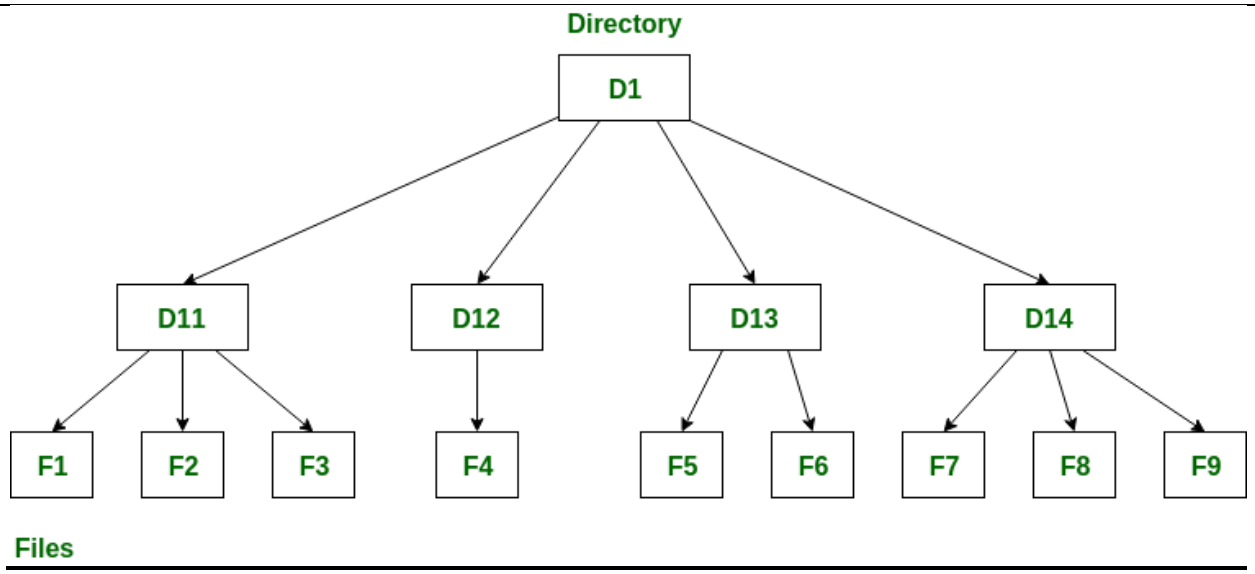
DIRECTORY IMPLEMENTATION:

There is the number of algorithms by using which, the directories can be implemented. However, the selection of an appropriate directory implementation algorithm may significantly affect the performance of the system.

The directory implementation algorithms are classified according to the data structure they are using. There are mainly two algorithms which are used in these days.

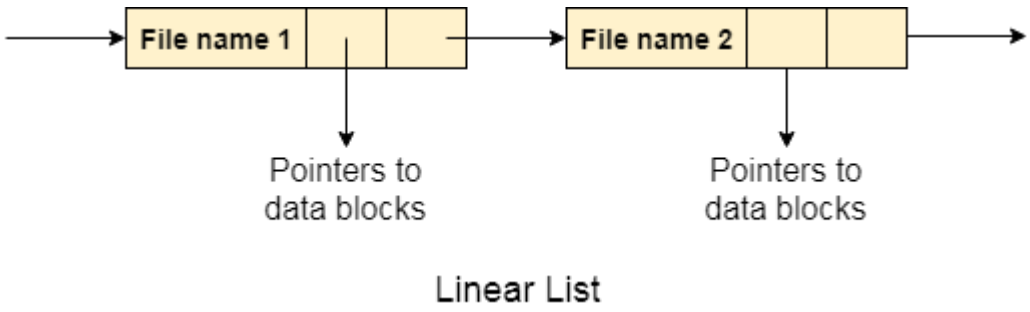
1. Linear List:

In this algorithm, all the files in a directory are maintained as singly lined list. Each file contains the pointers to the data blocks which are assigned to it and the next file in the directory.



Characteristics

1. When a new file is created, then the entire list is checked whether the new file name is matching to a existing file name or not. In case, it doesn't exist, the file can be created at the beginning or at the end. Therefore, searching for a unique name is a big concern because traversing the whole list takes time.
2. The list needs to be traversed in case of every operation (creation, deletion, updating, etc) on the files therefore the systems become inefficient.

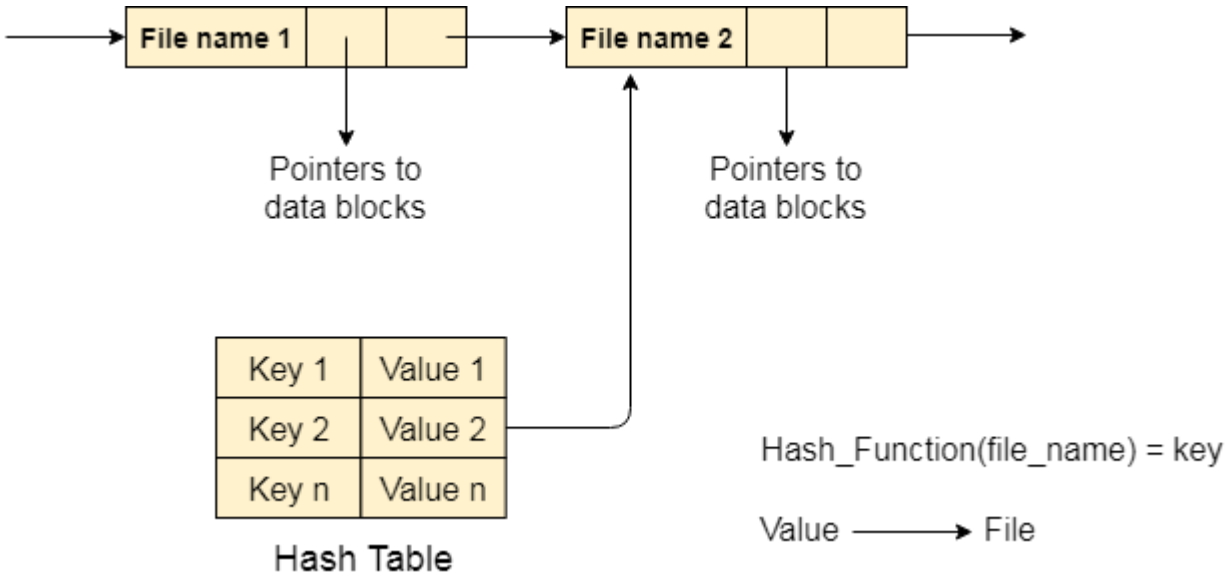


Linear List

2. Hash Table:

To overcome the drawbacks of singly linked list implementation of directories, there is an alternative approach that is hash table. This approach suggests to use hash table along with the linked lists. A key-value pair for each file in the directory gets generated and stored in the hash table. The key can be determined by applying the hash function on the file name while the key points to the corresponding file stored in the directory.

Now, searching becomes efficient due to the fact that now, entire list will not be searched on every operating. Only hash table entries are checked using the key and if an entry found then the corresponding file will be fetched using the value. contents of file. It also maintains via file control blocks. File control block (FCB) has information about a file – owner, size, permissions, location of file contents and must be appropriate for the intended application.



Video Content / Details of website for further learning (if any):
http://www.cs.nthu.edu.tw/~ychung/slides/CSC3150/Abraham-Silberschatz-Operating-System-Concepts---9th2012.12
https://www.javatpoint.com/os-directory-implementation

Important Books/Journals for further learning including the page nos.:
Book:
Abraham Silberschatz,Peter Bear Galvin and Greg Gagne, ," Operating system concepts",2015, John Wiley & Sons (ASIA) ,9th Edition,Page no(351-466).

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

L36

LECTURE HANDOUTS

AI&DS | **II/III**

Course Name with Code : 19ADCO7&OPERATING SYSTEM

Course Faculty : R.NIVETHITHA.

Unit : IV STORAGE AND FILE SYSTEM.

Date of Lecture:

Topic of Lecture: Allocation Method, Free-Space management.

Introduction :

- The allocation methods define how the files are stored in the disk blocks. There are three main disk space or file allocation methods.
- The directory entry for a file with contiguous allocation contains address of starting block
- The directory entry contains a pointer to the starting and the ending file block
- A file system is responsible to allocate the free blocks to the file therefore it has to keep track of all the free blocks present in the disk.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Operating Systems
- Hardware
- Software
- Process

Detailed content of the Lecture:

FILE ALLOCATION METHODS:

The allocation methods define how the files are stored in the disk blocks. There are three main disk space or file allocation methods.

- Contiguous Allocation
- Linked Allocation
- Indexed Allocation

The main idea behind these methods is to provide:

- Efficient disk space utilization.
- Fast access to the file blocks.

All the three methods have their own advantages and disadvantages as discussed below:

1. **Contiguous Allocation:**

In this scheme, each file occupies a contiguous set of blocks on the disk. For example, if a file requires n blocks and is given a block b as the starting location, then the blocks assigned to the file will be: *b, b+1, b+2,……b+n-1.* This means that given the starting block address and the length of the file (in terms of blocks required), we can determine the blocks occupied by the file.

The directory entry for a file with contiguous allocation contains

- Address of starting block
- Length of the allocated portion.

The *file 'mail'* in the following figure starts from the block 19 with length = 6 blocks. Therefore, it occupies *19, 20, 21, 22, 23, 24* blocks.

Directory

file	start	length
count	0	2
tr	14	3
mail	19	6
list	28	4
f	6	2

Advantages:

- Both the Sequential and Direct Accesses are supported by this. For direct access, the address of the kth block of the file which starts at block b can easily be obtained as (b+k).
- This is extremely fast since the number of seeks are minimal because of contiguous allocation of file blocks.

Disadvantages:

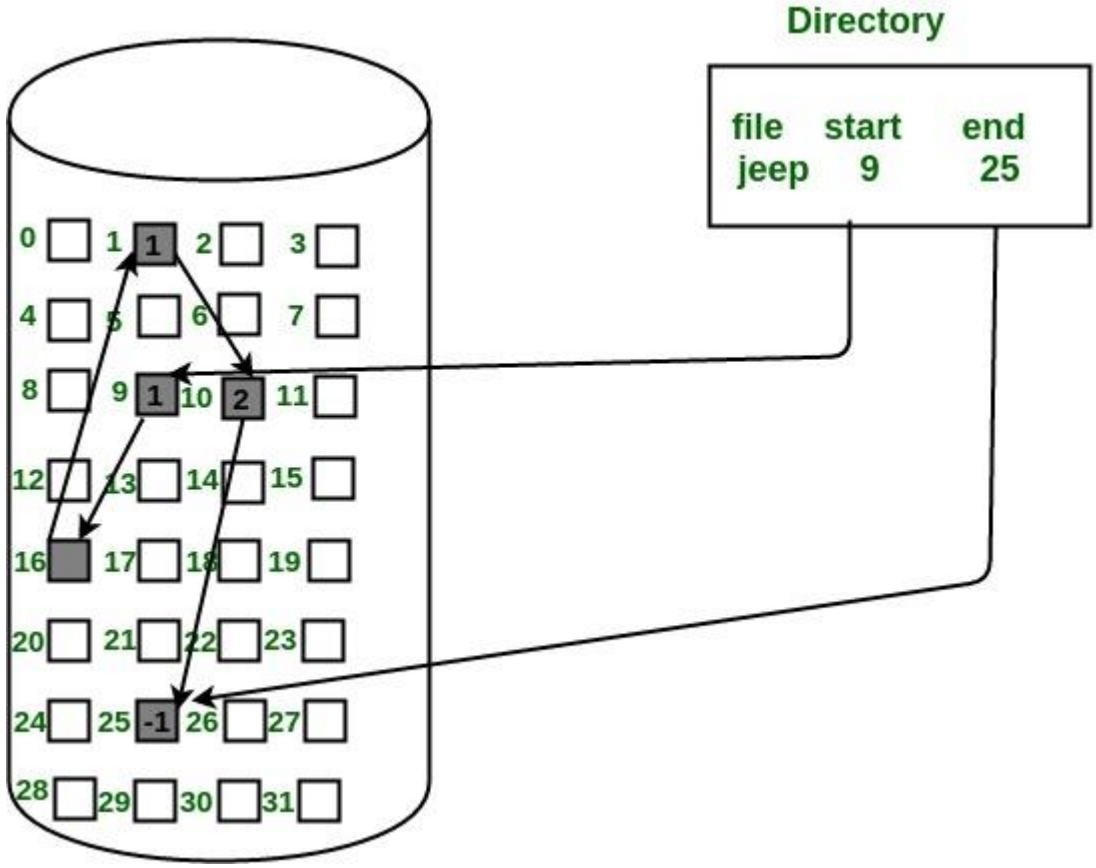
- This method suffers from both internal and external fragmentation. This makes it inefficient in terms of memory utilization.

- Increasing file size is difficult because it depends on the availability of contiguous memory at a particular instance.

2. Linked List Allocation:

In this scheme, each file is a linked list of disk blocks which **need not be** contiguous. The disk blocks can be scattered anywhere on the disk.
The directory entry contains a pointer to the starting and the ending file block. Each block contains a pointer to the next block occupied by the file.

The file 'jeep' in following image shows how the blocks are randomly distributed. The last block (25) contains -1 indicating a null pointer and does not point to any other block.



Advantages:

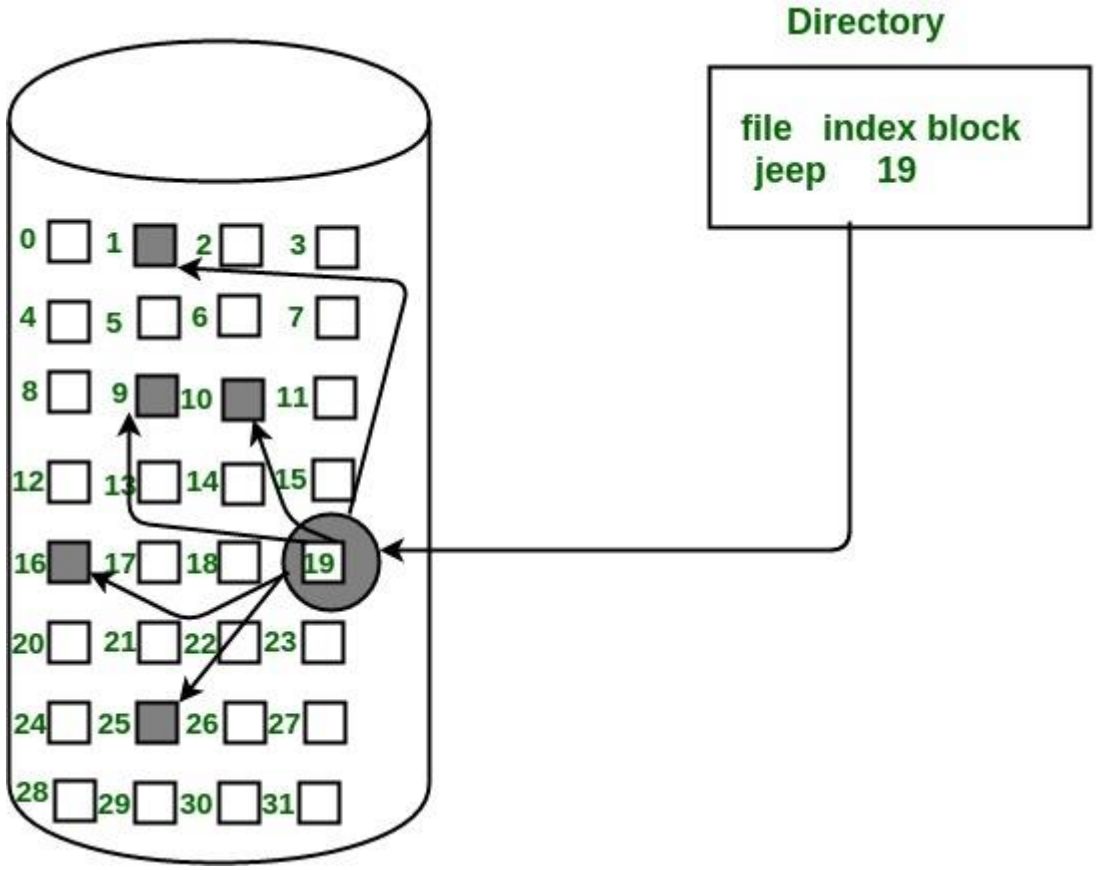
- This is very flexible in terms of file size. File size can be increased easily since the system does not have to look for a contiguous chunk of memory.
- This method does not suffer from external fragmentation. This makes it relatively better in terms of memory utilization.

Disadvantages:

- Because the file blocks are distributed randomly on the disk, a large number of seeks are needed to access every block individually. This makes linked allocation slower.
- It does not support random or direct access. We can not directly access the blocks of a file. A block k of a file can be accessed by traversing k blocks sequentially (sequential access) from the starting block of the file via block pointers.
- Pointers required in the linked allocation incur some extra overhead.

3.Indexed Allocation:

In this scheme, a special block known as the **Index block** contains the pointers to all the blocks occupied by a file. Each file has its own index block. The ith entry in the index block contains the disk address of the ith file block. The directory entry contains the address of the index block as shown in the image:



Advantages:

- This supports direct access to the blocks occupied by the file and therefore provides fast access to the file blocks.
- It overcomes the problem of external fragmentation.

Disadvantages:

- The pointer overhead for indexed allocation is greater than linked allocation.
- For very small files, say files that expand only 2-3 blocks, the indexed allocation would keep one entire block (index block) for the pointers which is inefficient in terms of memory utilization. However, in linked allocation we lose the space of only 1 pointer per block.

For files that are very large, single index block may not be able to hold all the pointers. Following mechanisms can be used to resolve this:

1. **Linked scheme:** This scheme links two or more index blocks together for holding the pointers. Every index block would then contain a pointer or the address to the next index block.
2. **Multilevel index:** In this policy, a first level index block is used to point to the second level index blocks which inturn points to the disk blocks occupied by the file. This can be extended to 3 or more levels depending on the maximum file size.
3. **Combined Scheme:** In this scheme, a special block called the **Inode (information Node)** contains all the information about the file such as the name, size, authority, etc

FREE SPACE MANAGEMENT:

A file system is responsible to allocate the free blocks to the file therefore it has to keep track of all the free blocks present in the disk. There are mainly two approaches by using which, the free blocks in the disk are managed.

1. Bit Vector

In this approach, the free space list is implemented as a bit map vector. It contains the number of bits where each bit represents each block.If the block is empty then the bit is 1 otherwise it is 0. Initially all the blocks are empty therefore each bit in the bit map vector contains 1.LAs the space allocation proceeds,the file system starts allocating blocks to the files and setting the respective bit to 0.

2. Linked List

It is another approach for free space management. This approach suggests linking together all the free blocks and keeping a pointer in the cache which points to the first free block.Therefore, all the free blocks on the disks will be linked together with a pointer. Whenever a block gets allocated, its previous free block will be linked to its next free block.

Video Content / Details of website for further learning (if any):
http://www.cs.nthu.edu.tw/~ychung/slides/CSC3150/Abraham-Silberschatz-Operating-System-Concepts---9th2012.12
https://www.javatpoint.com/os-directory-implementation

Important Books/Journals for further learning including the page nos.:
Book:
Abraham Silberschatz,Peter Bear Galvin and Greg Gagne, ," Operating system concepts",2015, John Wiley & Sons (ASIA) ,9th Edition,Page no(351-466).

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

L37

LECTURE HANDOUTS

AI&DS

II/III

Course Name with Code : **19ADCO7&OPERATING SYSTEM**

Course Faculty : **R.NIVETHITHA.**

Unit : **V CASE STUDY-LINUX SYSTEM**

Date of Lecture:

Topic of Lecture: Linux System- Basic Concepts.

Introduction :

- This chapter does not go into great detail or cover advanced topics. Instead, we want you to hit the ground running.
- We assume very little here about your background, except perhaps that you have some familiarity with personal computer systems, and MS-DOS..
- The main system libraries of Linux were originated by the GNU project

Prerequisite knowledge for Complete understanding and learning of Topic:

- Operating Systems
- Hardware
- Software
- Process

Detailed content of the Lecture:

LINUX SYSTEM:

If you're new to UNIX and Linux, you may be a bit intimidated by the size and apparent complexity of the system before you. This chapter does not go into great detail or cover advanced topics. Instead, we want you to hit the ground running.

We assume very little here about your background, except perhaps that you have some familiarity with personal computer systems, and MS-DOS. However, even if you're not an MS-DOS user, you should be able to understand everything here. At first glance, parts of MS-DOS were modeled on the CP/M operating system, which in turn was modeled on UNIX. However, only the most superficial features of Linux resemble MS-DOS. Even if you're completely new to the PC world, this tutorial should help.

And, before we begin: *Don't be afraid to experiment.* The system won't bite you. You can't destroy anything by working on the system. Linux has built-in security features to prevent ``normal'' users from damaging files that are essential to the system. Even so, the worst thing that can happen is that you may delete some or all of your files and you'll have to re-install the system. So, at this point,

you have nothing to lose.

Linux looks and feels much like any other UNIX system; indeed, UNIX compatibility has been a major design goal of the Linux project. However, Linux is much younger than most UNIX systems. Its development began in1991, when a Finnish university student, Linus Torvalds, began developing a small but self-contained kernel for the 80386 processor, the first true 32-bitprocessor in Intel's range of PC-compatible CPUs. of arbitrary files (but only read-only memory mapping was implemented .

A range of extra hardware support was included in this release. Although still restricted to the Intel PC platform, hardware support had grown to include floppy-disk and CD-ROM devices, as well as sound cards, a range of mice, and international keyboards. Floating-point emulation was provided in the kernel for 80386 users who had no 80387 math coprocessor. System V UNIX-style interprocess communication (IPC), including shared memory, semaphores, and message queues, was implemented.

This sharing of tools has worked in both directions. The main system libraries of Linux were originated by the GNU project, but the Linux community greatly improved the libraries by addressing omissions, inefficiencies, and bugs. Other components, such as the GNU C compiler (gcc), were already of sufficiently high quality to be used directly in Linux. The network administration tools under Linux were derived from code first developed for 4.3 BSD, but more recent BSD derivatives, such as FreeBSD, have borrowed code from Linux in return. Examples of this sharing include the Intel floating-point-emulation math library and the PC sound-hardware device drivers.

The Linux system as a whole is maintained by a loose network of developers collaborating over the Internet, with small groups or individuals having responsibility for maintaining the integrity of specific components.

COMPONENTS OF LINUX SYSTEM:

Linux Operating System has primarily three components

- **Kernel** − Kernel is the core part of Linux. It is responsible for all major activities of this operating system. It consists of various modules and it interacts directly with the underlying hardware. Kernel provides the required abstraction to hide low level hardware details to system or application programs.
- **System Library** − System libraries are special functions or programs using which application programs or system utilities accesses Kernel's features. These libraries implement most of the functionalities of the operating system and do not requires kernel module's code access rights.
- **System Utility** − System Utility programs are responsible to do specialized, individual level tasks.

BASIC FEATURES:

Following are some of the important features of Linux Operating System.

- **Portable** − Portability means software can works on different types of hardware in same way.

Linux kernel and application programs supports their installation on any kind of hardware platform.

- **Open Source** − Linux source code is freely available and it is community based development project. Multiple teams work in collaboration to enhance the capability of Linux operating system and it is continuously evolving.
- **Multi-User** − Linux is a multiuser system means multiple users can access system resources like memory/ ram/ application programs at same time.
- **Multiprogramming** − Linux is a multiprogramming system means multiple applications can run at same time.
- **Hierarchical File System** − Linux provides a standard file structure in which system files/ user files are arranged.
- **Shell** − Linux provides a special interpreter program which can be used to execute commands of the operating system. It can be used to do various types of operations, call application programs. etc.
- **Security** − Linux provides user security using authentication features like password protection/ controlled access to specific files/ encryption of data.

Video Content / Details of website for further learning (if any):

http://www.cs.nthu.edu.tw/~ychung/slides/CSC3150/Abraham-Silberschatz-Operating-System-Concepts---9th2012.12

https://www.tutorialspoint.com/operating_system/os_linux.htm

https://www.brainkart.com/article/Linux-System---Basic-Concepts_9864/

Important Books/Journals for further learning including the page nos.:

Book:

Abraham Silberschatz,Peter Bear Galvin and Greg Gagne, ," Operating system concepts",2015, John Wiley & Sons (ASIA) ,9th Edition,Page no(467-624).

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

L38

LECTURE HANDOUTS

AI&DS

II/III

Course Name with Code : **19ADCO7&OPERATING SYSTEM**

Course Faculty : **R.NIVETHITHA.**

Unit : **V CASE STUDY-LINUX SYSTEM**

Date of Lecture:

Topic of Lecture: System Administration.

Introduction :

- Linux is a major strength in computing technology.
- There are some of the things that a Linux system administrator should know and understand.
- The job of a Linux systems administrator is to manage the operations of a computer system like maintain, enhance, create user account/report,
- It taking backups using Linux tools and command-line interface tools

Prerequisite knowledge for Complete understanding and learning of Topic:

- Operating Systems
- Hardware
- Software
- Process

Detailed content of the Lecture:

SYSYEM ADMINISTATION:

Linux is a major strength in computing technology. Most of the webserver, mobile phones, personal computers, supercomputers, and cloud-servers are powered by Linux. The job of a Linux systems administrator is to manage the operations of a computer system like maintain, enhance, create user account/report, taking backups using Linux tools and command-line interface tools. Most computing devices are powered by Linux because of its high stability, high security, and open-source environment. There are some of the things that a Linux system administrator should know and understand.

Steps to Start the Career as Linux System Admin:

- Install and learn to use Linux environment
- Get Certified in Linux administration
- Learn to do Documentation
- Joining up with a local Linux Users Group or Community for Support and Help

In short, the main role of the **Linux Systems Administrator** is to manage the operations like install, observe the software and hardware systems and taking backup. And also have a good ability to describe an In-depth understanding of technical knowledge. Even freshman-level Professionals has great possibilities for the position of System Administrator with the yearly median salary is around INR 3 Lacs, salary increase with an increase in job experience. To get the experience you need to check for the latest skills and learning in the Linux community.

Linux was designed based on the Unix philosophy of "small, precise tools chained together simplifying larger tasks". Linux, at its root, does not have large single-purpose applications for one specific use a lot of the time. Instead, there are hundreds of basic utilities that when combined offer great power to accomplish big tasks with efficiency.

Unique amongst business class Linux distributions, CentOS stays true to the open-source nature that Linux was founded on. This tutorial gives a complete understanding on Linux Admin and explains how to use it for benefit.

Linux looks and feels are similar to that of any other UNIX system; certainly, UNIX compatibility has been a chief design goal for the Linux project. However, Linux is much younger compared to most UNIX systems. Its development began in the year 1991, when a Finnish student named Linus Torvalds, wrote and launched Linux, a small but self-contained kernel for 80386 processor. Early at the time of its development, the Linux source code was made free on the Internet so that everyone can compose their distro/version. From an initial kernel which partially employs a small subset of the UNIX system services, the Linux system gradually developed to include much of the ifFNIX functionality.

Video Content / Details of website for further learning (if any):

http://www.cs.nthu.edu.tw/~ychung/slides/CSC3150/Abraham-Silberschatz-Operating-System-Concepts---9th2012.12

https://www.geeksforgeeks.org/what-is-linux-system-administration/

https://www.tutorialspoint.com/linux_admin/index.htm

Important Books/Journals for further learning including the page nos.:

Book:

Abraham Silberschatz,Peter Bear Galvin and Greg Gagne, ," Operating system concepts",2015, John Wiley & Sons (ASIA) ,9th Edition,Page no(467-624).

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

L39

LECTURE HANDOUTS

AI&DS

II/III

Course Name with Code : **19ADCO7&OPERATING SYSTEM**

Course Faculty : **R.NIVETHITHA.**

Unit : **V CASE STUDY-LINUX SYSTEM**

Date of Lecture:

Topic of Lecture: Requirements for Linux System Administrator.

Introduction :

- Linux admin, or Linux system administrator, is an IT professional who manages the functionality of a Linux system
- Linux administrators are responsible for maximizing the performance of the operating system.
- Linux works and collaborate with other technology experts, such as software developers or computer engineers

Prerequisite knowledge for Complete understanding and learning of Topic:

- Operating Systems
- Hardware
- Software
- Process

Detailed content of the Lecture:

REQUIREMENTS FOR LINUX SYSTEM ADMINISTRATOR:

To become a Linux administrator, follow these steps:

- Earn a bachelor's degree. A bachelor's degree is the minimum education requirement to become a Linux administrator.
- Pursue a master's degree.
- Enroll in training courses.
- Practice installing Linux.
- Get certified.
- Problem-solving.
- Attention to detail.
- Communication.

LINUX ADMIN:

Linux admin, or Linux system administrator, is an IT professional who manages the functionality of a Linux system. Similar to iOS and Windows, Linux is an operating system that connects the internal hardware in electronics, including phones and computers. Administrators answer technical questions about how Linux works and collaborate with other technology experts, such as software developers or computer engineers. They

may also choose to specialize in cloud computing or automation.

Linux administrators are responsible for maximizing the performance of the operating system. Security is fundamental to their role, especially when approving the installation of new software and enhancement features. Some of their day-to-day responsibilities include:

- Installing Linux servers and systems
- Updating the system after crashes or reboots
- Performing maintenance checks and system backups
- Reading error logs
- Conducting root analyses
- Processing user complaints

Linux and Administration fundamentals such as

- What is the root user?
- The power of the root user
- Basic concept of security groups and users
- Experience using a Linux terminal emulator
- Fundamental networking concepts
- Fundamental understanding of interpreted programming languages (Perl, Python, Ruby)
- Networking protocols such as HTTP, LDAP, FTP, IMAP, SMTP
- Cores that compose a computer operating system: file system, drivers, and the kerne

Video Content / Details of website for further learning (if any):

http://www.cs.nthu.edu.tw/~ychung/slides/CSC3150/Abraham-Silberschatz-Operating-System-Concepts---9th2012.12

https://www.tutorialspoint.com/linux_admin/linux_admin_centos_overview.htm

https://www.brainkart.com/article/Linux-System---Basic-Concepts_9864/

Important Books/Journals for further learning including the page nos.:

Book:

Abraham Silberschatz,Peter Bear Galvin and Greg Gagne, ," Operating system concepts",2015, John Wiley & Sons (ASIA) ,9th Edition,Page no(467-624).

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

L40

LECTURE HANDOUTS

AI&DS

II/III

Course Name with Code : **19ADCO7&OPERATING SYSTEM**

Course Faculty : **R.NIVETHITHA.**

Unit : **V CASE STUDY-LINUX SYSTEM**

Date of Lecture:

Topic of Lecture: Setting up a LINUX Multifunction Server.

Introduction :

1. Confirm that the printer you will use to connect to the DPR-1020 is operating correctly.
2. When you have confirmed that the printer is operating correctly, switch its power OFF.
3. Confirm that your network is operating normally.
4. Using a CAT 5 Ethernet cable, connect the DPR-1020 Ethernet Port (labelled LAN) to the network

Prerequisite knowledge for Complete understanding and learning of Topic:

- Operating Systems
- Hardware
- Software
- Process

Detailed content of the Lecture:

SETTING UP A LINUX MULTIFUNCTION SERVER:

1. Confirm that the printer you will use to connect to the DPR-1020 is operating correctly.
2. When you have confirmed that the printer is operating correctly, switch its power OFF.
3. Confirm that your network is operating normally.
4. Using a CAT 5 Ethernet cable, connect the DPR-1020 Ethernet Port (labelled LAN) to the network.
5. While the printer is turned OFF, connect the USB printer cable to the printer and then to the USB port on the Print Server.
6. Switch on the printer.
7. Insert the power adapter's output plug into the DC 5V power socket on the rear panel of the Print Server.
8. Connect the other end of the power adapter into a power outlet. This will supply power to the Print Server. The blue LED on the Print Server's front panel should turn on and the Print Server's self-test will proceed.

- When the DPR-1020 is powered ON, it automatically performs a Self-Test on each of its major components. The final result of the Self-Test is signaled by the state of the USB LED indicator following the Self-Test. Preliminary to the actual component tests, the three LED indicators are tested to confirm their operation.

- Immediately after power-up, all three of the blue LEDs should illuminate steadily for several seconds. Then the USB LED should light OFF simultaneously. Irregularity of any of the three LEDs during these LED tests may mean there is a problem with the LEDs themselves.

- The actual component tests immediately follow the LED tests. A normal (no fault) result is signaled by simultaneous flashing of the LEDs three times, followed by a quiescent state with all three LEDs dark.

- If the Self-Test routine traps any component error, then following the LED tests the Self-Test will halt and the LEDs will continuously signal the error according to the following table. In the event of any such error signal, contact your dealer for correction of the faulty unit.

Video Content / Details of website for further learning (if any):
http://www.cs.nthu.edu.tw/~ychung/slides/CSC3150/Abraham-Silberschatz-Operating-System-Concepts---9th2012.12
https://www.brainkart.com/article/Setting-Up-a-Linux-Multifunction-Server_9867/

Important Books/Journals for further learning including the page nos.:
Book:

Abraham Silberschatz,Peter Bear Galvin and Greg Gagne, ," Operating system concepts",2015, John Wiley & Sons (ASIA) ,9th Edition,Page no(467-624).

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

L41

LECTURE HANDOUTS

AI&DS

II/III

Course Name with Code : **19ADCO7&OPERATING SYSTEM**

Course Faculty : **R.NIVETHITHA.**

Unit : **V CASE STUDY-LINUX SYSTEM**

Date of Lecture:

Topic of Lecture: Domain Name System.

Introduction :

1. DNS naming system is a hierarchical and logical tree structure called the DNS namespace.
2. The DNS namespace has a unique root that can have any number of sub domains.
3. In turn, each sub domain can have more sub domains.
4. Organizations can also create private networks and use their own private DNS namespaces that are not visible on the Internet.

Prerequisite knowledge for Complete understanding and learning of Topic:

- **Operating Systems**
- Hardware
- Software
- Process

Detailed content of the Lecture:

DOMAIN NAME SYSTEM:

A Domain Name System (DNS) server, or name server, is used to resolve an IP address to a hostname or vice versa. Berkeley Internet Name Domain (BIND) is the most commonly used DNS server on the Internet, especially on Unix-like systems.

DNS naming system is a hierarchical and logical tree structure called the DNS namespace. The DNS namespace has a unique root that can have any number of sub domains. In turn, each sub domain can have more sub domains. The Root in the Internet namespace has many top-level domain names, one of which is com. The domain com can have a sub domain for example,omnisecu.com, can have further sub domains like mcse.omnisecu.com and rhce.omnisecu.com. Organizations can also create private networks and use their own private DNS namespaces that are not visible on the Internet.

Domain: A Domain is any tree or sub-tree within the overall domain namespace.

• Root Domain: Root Domain is the root of the DNS tree. It is unnamed and is represented by a period

(.).

• Top-level Domain: Usually top-level domain name is a two or three character name code that identifies the organizational or geographical status for the domain name.

Example, .com, .biz, .net, .org, .gov, .in, .au etc.

• Second-level Domain: A second level domain is a unique name of variable length assigned to individuals or organizations that connect to the internet. Example: myuniversity.edu. Here second level name refers to ".my university", which is assigned by InterNIC.

• Sub Domains: Large organizations can further subdivide its registered domain name by adding subdivisions that are represented by separate name portion.

FEATURES:

1.Domain Name System (DNS) provides name resolution services to the client computers.

2. Domain Name System (DNS) provides IP address to hostname mapping.

3. Linux DNS Servers can operate in three modes and these modes are, Caching only DNS server, Primary DNS server and Slave (secondary) DNS server.

4. Replication of DNS database between servers

Video Content / Details of website for further learning (if any):
http://www.cs.nthu.edu.tw/~ychung/slides/CSC3150/Abraham-Silberschatz-Operating-System-Concepts---9th2012.12
https://www.omnisecu.com/gnu-linux/redhat-certified-engineer-rhce/introduction-to-domain-name-system-dns.php

Important Books/Journals for further learning including the page nos.:
Book:

Abraham Silberschatz,Peter Bear Galvin and Greg Gagne, ," Operating system concepts",2015, John Wiley & Sons (ASIA) ,9th Edition,Page no(467-624).

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

L42

LECTURE HANDOUTS

AI&DS

II/III

Course Name with Code : **19ADCO7&OPERATING SYSTEM**

Course Faculty : **R.NIVETHITHA.**

Unit : **V CASE STUDY-LINUX SYSTEM**

Date of Lecture:

Topic of Lecture: Setting Up Local Network Server.

Introduction :

1. Linux is increasingly popular in the computer networking/telecommunications industry.
2. Acquiring the Linux operating system is a relatively simple and inexpensive task since virtually all of the source code can be downloaded from several different FTP or HTTP sites on the Internet.
3. This can be very helpful if, for instance, you can not resolve an installation/configuration problem after consulting the Red Hat Linux manuals.
4. The size and scope of a LAN is usually small, covering a single building or group of buildings.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Operating Systems
- Hardware
- Software
- Process

Detailed content of the Lecture:

SETTING UP LOCAL NETWORK SERVER.

Linux is increasingly popular in the computer networking/telecommunications industry. Acquiring the Linux operating system is a relatively simple and inexpensive task since virtually all of the source code can be downloaded from several different FTP or HTTP sites on the Internet. In addition, the most recent version of Red Hat Linux can be purchased from computer retail stores for between $25 and $50, depending on whether you purchase the standard or full version. The retail brand is indeed a worthwhile investment (vs. the free FTP or HTTP versions) since valuable technical support is included directly from the Red Hat Linux engineers for at least a year. This can be very helpful if, for instance, you can not resolve an installation/configuration problem after consulting the Red Hat Linux manuals.

This article describes how to put together a Local Area Network (LAN) consisting of two or more computers using the Red Hat Linux 6.2 operating system. A *LAN* is a communications network that interconnects a variety of devices and provides a means for exchanging information among those

devices. The size and scope of a LAN is usually small, covering a single building or group of buildings. In a LAN, modems and phone lines are not required, and the computers should be close enough to run a network cable between them.

Introduction to TCP/IP:

- TCP/IP is the suite of protocols used by the Internet and most LANs throughout the world. In TCP/IP, every host (computer or other communications device) that is connected to the network has a unique IP address. An IP address is composed of four octets (numbers in the range of 0 to 255) separated by decimal points. The IP address is used to uniquely identify a host or computer on the LAN. For example, a computer with the hostname Morpheus could have an IP address of 192.168.7.127. You should avoid giving two or more computers the same IP address by using the range of IP addresses that are reserved for private, local area networks; this range of IP addresses usually begins with the octets 192.168.
- LAN network address The first three octets of an IP address should be the same for all computers in the LAN. For example, if a total of 128 hosts exist in a single LAN, the IP addresses could be assigned starting with 192.168.1.x, where x represents a number in the range of 1 to 128. You could create consecutive LANs within the same company in a similar manner consisting of up to another 128 computers. Of course, you are not limited to 128 computers, as there are other ranges of IP addresses that allow you to build even larger networks.
- There are different classes of networks that determine the size and total possible unique IP addresses of any given LAN. For example, a class A LAN can have over 16 million unique IP addresses. A class B LAN can have over 65,000 unique IP addresses. The size of your LAN depends on which reserved address range you use and the subnet mask (explained later in the article) associated with that range.

Video Content / Details of website for further learning (if any):
http://www.cs.nthu.edu.tw/~ychung/slides/CSC3150/Abraham-Silberschatz-Operating-System-Concepts---9th2012.12
https://www.brainkart.com/article/Setting-up-Local-Network-Services_9869/

Important Books/Journals for further learning including the page nos.:
Book:

Abraham Silberschatz,Peter Bear Galvin and Greg Gagne, ," Operating system concepts",2015, John Wiley & Sons (ASIA) ,9th Edition,Page no(467-624).

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

L43

LECTURE HANDOUTS

AI&DS

II/III

Course Name with Code : **19ADCO7&OPERATING SYSTEM**

Course Faculty : **R.NIVETHITHA.**

Unit : **V CASE STUDY-LINUX SYSTEM**

Date of Lecture:

Topic of Lecture: Virtualization-Basic concept.

Introduction :

- Linux virtualization is a process through which one or more virtual machines can be installed,
- Executed and maintained on top of the Linux operating system.
- Linux virtualization enables the consolidation of hardware and software resources used by the Linux OS.
- Allows them to be shared and divided across several virtual machines and their associated processes

Prerequisite knowledge for Complete understanding and learning of Topic:

- Operating Systems
- Hardware
- Software
- Process

Detailed content of the Lecture:

VIRTUALIZATION:

Linux virtualization is designed to achieve virtualization on a system running the Linux operating system. Linux virtualization is accomplished through the installation of a virtual machine application on the target system that can create some or more virtual machines depending on the back-end system resources. Each virtualized machine shares the underlying hardware resources but runs independently of the parent Linux OS. However, Linux virtualization allows the creation and execution of Windows, Mac OS X and other virtual machines powered by operating systems other than Linux. Xen, KVM, VirtualBox and VMware are among the popular applications for Linux virtualization.

Linux virtualization refers to a process where more than one virtual machine can be installed in the Linux operating system.It helps in the collaboration of the hardware and software resources of the Linux operating Software. It provides the sharing of resources among the virtual machines installed on the operating system. Despite sharing its capabilities, hardware can still perform its operational functions independently.

Linux virtualization in cloud computing can assist in delivering the virtualized environment at economical cost with greater efficiency. The advantage of adopting it is that it allows Windows, Mac OS X, and other virtual machines to implement operating systems other than Linux.

TOP VIRTUALISATION SOFTWARE:

A. Open Source Linux Virtualization

OpenVZ is an operating system whose functioning is based on the Linux kernel.

Xen is a virtual monitor that incorporates 32/64 bit intel /AMD (IA 64) and PowerPC 970 architectures. It allows the execution of the various guest operating systems on the same hardware simultaneously. Xen is equipped with the most popular Linux products such as Debian, CentOS, Fedora, Ubuntu, and others.

Kernel-based Virtual Machine (KVM): It has a Linux kernel virtualization infrastructure that currently supports native virtualization. It uses Intel VT or AMD-V. KVM works with many guest operating systems including Linux, BSD, Windows, and Solaris.

Linux-VServer, VirtualBox, Bochs, User Mode Linux(UML) are the other open-source software.

B. Proprietary Linux Virtualization Software

VMware ESX Server and VMWare Server: It is an entry-level Linux virtualization server software. It is beneficial for the enterprise as it provides data center virtualization. It enables the enterprise to run guest operating systems such as Solaris, Linux, FreeBSD, Windows, and various others.

a. Citric XenServer: It utilizes an open-source Xen hypervisor that divides the server into logical parts. It is a technology that provides low overhead along with primary performance.

b. Oracle VM: It also employs open-source Xen hypervisor technology, that carries both Windows and Linux guests. It includes an integrated Web browser-based on Management console.

Video Content / Details of website for further learning (if any):

http://www.cs.nthu.edu.tw/~ychung/slides/CSC3150/Abraham-Silberschatz-Operating-System-Concepts---9th2012.12

https://www.techopedia.com/definition/28404/linux-virtualization

https://www.jigsawacademy.com/blogs/cloud-computing/linux-virtualization

Important Books/Journals for further learning including the page nos.:

Book:

Abraham Silberschatz,Peter Bear Galvin and Greg Gagne, ," Operating system concepts",2015, John Wiley & Sons (ASIA) ,9th Edition,Page no(467-624).

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

L44

LECTURE HANDOUTS

AI&DS

II/III

Course Name with Code : **19ADCO7&OPERATING SYSTEM**

Course Faculty : **R.NIVETHITHA.**

Unit : **V CASE STUDY-LINUX SYSTEM**

Date of Lecture:

Topic of Lecture: Setting Up Xen.

Introduction :

- Xen is an Open Source software that allows you to provision virtualized resources to guests.
- Setting up of Xen Virtualization software involves – installation of Xen kernel, xen server (xend), virt-manager (GUI tool to create, destroy and manage virtual machines).
- The Xen kernel runs on top of Linux kernel and is responsible for launching Dom0 (default and first virtual machine).
- The only VM that interacts directly with the system hardware.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Operating Systems
- Hardware
- Software
- Process

Detailed content of the Lecture:

XEN:

Xen is the open source hypervisor included in the Linux kernel and, as such, it is available in all Linux distributions. The Xen Project is one of the many open source projects managed by the Linux Foundation.

XEN COMPONENTS:

A typical environment running Xen consists of different parts. To start with, there's Domain 0. In Xen, this is how you refer to the host operating system (OS), as it's not really a host OS in the sense that other virtual machines (VMs) -- domains in Xen terminology -- don't have to use it to get access to the host server hardware. Domain 0 is only responsible for access to the drivers, and if any coordination has to be done, it will be handled by Domain 0. Apart from Domain 0, there are the other

VMs that are referred to as Domain U.

PARAVIRTUALIZATION:

Xen offers two types of virtualization: paravirtualization and full virtualization. In paravirtualization, the virtualized OS runs a modified version of the OS, which results in the OS knowing that it's virtualized. This enables much more efficient communication between the OS and the physical hardware, as the hardware devices can be addressed directly. The only drawback of paravirtualization is that a modified guest OS needs to be used, which isn't provided by many vendors.

The counterpart of paravirtualization is full virtualization. This is a virtualization mode where the CPU needs to provide support for virtualization extensions. In full virtualization, unmodified virtualized OSes can efficiently address the hardware because of this support.

Commercial versions of Xen:

Although Xen is included in the Linux kernel, only a few Linux distributions, such as Oracle Unbreakable Linux and SUSE Linux Enterprise Server, offer a supported Xen stack. Red Hat included Xen up to Red Hat Enterprise Linux (RHEL) 5, but switched to KVM with the release of RHEL 6.

Video Content / Details of website for further learning (if any):

http://www.cs.nthu.edu.tw/~ychung/slides/CSC3150/Abraham-Silberschatz-Operating-System-Concepts---9th2012.12

https://techglimpse.com/xen-installation-steps-commands-linux/

https://searchservervirtualization.techtarget.com/definition/Xen

Important Books/Journals for further learning including the page nos.:

Book:

Abraham Silberschatz,Peter Bear Galvin and Greg Gagne, ," Operating system concepts",2015, John Wiley & Sons (ASIA) ,9th Edition,Page no(467-624).

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

L45

LECTURE HANDOUTS

AI&DS

II/III

Course Name with Code : **19ADCO7&OPERATING SYSTEM**

Course Faculty : **R.NIVETHITHA.**

Unit : **V CASE STUDY-LINUX SYSTEM**

Date of Lecture:

Topic of Lecture: VMware on Linux Host and Adding Guest OS.

Introduction :

- VMware Player is a free desktop application that lets you run virtual machines on a Windows or Linux PC.
- VMware Player is the only product on the market that lets you run virtual machines without investing in virtualization software, making it easier than ever to take advantage of the security, flexibility.
- Portability of virtual machines.
- VMware Player lets you use host machine devices, such as CD and DVD drives, from the virtual machine.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Operating Systems
- Hardware
- Software
- Process

Detailed content of the Lecture:

Use and evaluate prebuilt applications:

Download and safely run prebuilt application environments in virtual machines that are available from the Virtual Appliance Marketplace at http://vam.vmware.com.The Virtual Appliance Marketplace includes virtual machines from leading software vendors, including Oracle, Red Hat, Novell, BEA, SpikeSource, IBM, and MySQL, as well as virtual machines that are preconfigured with popular open source software.

Transform software distribution:

Simplify software distribution byshipping preconfigured software in virtual machines. End users can experience the benefits of your products immediately, without setup hassles. VMware Player

is ideal for shipping evaluation copies or beta software. You can package complex, sophisticated applications, complete with a full working environment, in a virtual machine that can be used by anyone who downloads VMware Player.

Features in VMware Player

- VMware Player is a free desktop application for running virtual machines. VMware Player does not include features found in other VMware products, such as the ability to create virtual machines.
- VMware Player provides the following features:
- You can connect, disconnect, and use configured host devices, including USB devices, in the virtualmachine.
- You can set preferences, such as how devices are displayed in VMware Player.
- You can change the amount of memory allocated to the virtual machine.
- You can drag and drop files between a Linux or Windows host and a Linux, Windows, or Solaris guest.(Linux hosts and Linux and Solaris guests must be running X Windows.) You can use this feature if theperson who created the virtual machine you are running also installed VMware Tools in it.
- You can copy and paste text between a Windows or Linux host and a Windows, Linux, or Solaris guest.
- You can use this feature if the person who created the virtual machine you are running also installed VMware Tools in it.
- You can copy and paste files between a Windows or Linux host and a Windows, Linux, or Solaris guest.
- You can use this feature if the person who created the virtual machine you are running also installed VMware Tools in it

Video Content / Details of website for further learning (if any):
http://www.cs.nthu.edu.tw/~ychung/slides/CSC3150/Abraham-Silberschatz-Operating-System-Concepts---9th2012.12

Important Books/Journals for further learning including the page nos.:
Book:

Abraham Silberschatz,Peter Bear Galvin and Greg Gagne, ," Operating system concepts",2015, John Wiley & Sons (ASIA) ,9th Edition,Page no(467-624).

Course Faculty

Verified by HOD