

FUNDAMENTALS OF OPERATING SYSTEM

PGDCA 104

BLOCK 1: INTRODUCTION TO OPERATING SYSTEMS

**Dr. Babasaheb Ambedkar Open University
Ahmedabad**



FUNDAMENTALS OF OPERATING SYSTEM



Knowledge Management and
Research Organization
Pune



Editorial Panel

Author

Er. Nishit Mathur

Language Editor

Prof. Jaipal Gaikwad

Graphic and Creative Panel

Ms. K. Jamdal

Ms. Lata Dawange

Ms. Pinaz Driver

Ms. Ashwini Wankhede

Mr. Kiran Shinde

Mr. Prashant Tikone

Mr. Akshay Mirajkar

Copyright © 2015 Knowledge Management and Research Organization.

All rights reserved. No part of this book may be reproduced, transmitted or utilized in any form or by a means, electronic or mechanical, including photocopying, recording or by any information storage or retrieval system without written permission from us.

Acknowledgment

Every attempt has been made to trace the copyright holders of material reproduced in this book. Should an infringement have occurred, we apologize for the same and will be pleased to make necessary correction/amendment in future edition of this book.

The content is developed by taking reference of online and print publications that are mentioned in Bibliography. The content developed represents the breadth of research excellence in this multidisciplinary academic field. Some of the information, illustrations and examples are taken "as is" and as available in the references mentioned in Bibliography for academic purpose and better understanding by learner.'



ROLE OF SELF INSTRUCTIONAL MATERIAL IN DISTANCE LEARNING

The need to plan effective instruction is imperative for a successful distance teaching repertoire. This is due to the fact that the instructional designer, the tutor, the author (s) and the student are often separated by distance and may never meet in person. This is an increasingly common scenario in distance education instruction. As much as possible, teaching by distance should stimulate the student's intellectual involvement and contain all the necessary learning instructional activities that are capable of guiding the student through the course objectives. Therefore, the course / self-instructional material are completely equipped with everything that the syllabus prescribes.

To ensure effective instruction, a number of instructional design ideas are used and these help students to acquire knowledge, intellectual skills, motor skills and necessary attitudinal changes. In this respect, students' assessment and course evaluation are incorporated in the text.

The nature of instructional activities used in distance education self-instructional materials depends on the domain of learning that they reinforce in the text, that is, the cognitive, psychomotor and affective. These are further interpreted in the acquisition of knowledge, intellectual skills and motor skills. Students may be encouraged to gain, apply and communicate (orally or in writing) the knowledge acquired. Intellectual-skills objectives may be met by designing instructions that make use of students' prior knowledge and experiences in the discourse as the foundation on which newly acquired knowledge is built.

The provision of exercises in the form of assignments, projects and tutorial feedback is necessary. Instructional activities that teach motor skills need to be graphically demonstrated and the correct practices provided during tutorials. Instructional activities for inculcating change in attitude and behavior should create interest and demonstrate need and benefits gained by adopting the required change. Information on the adoption and procedures for practice of new attitudes may then be introduced.

Teaching and learning at a distance eliminates interactive communication cues, such as pauses, intonation and gestures, associated with the face-to-face method of teaching. This is particularly so with the exclusive use of print media. Instructional activities built into the instructional repertoire provide this missing interaction between the student and the teacher. Therefore, the use of instructional activities to affect better distance teaching is not optional, but mandatory.

Our team of successful writers and authors has tried to reduce this.

Divide and to bring this Self Instructional Material as the best teaching and communication tool. Instructional activities are varied in order to assess the different facets of the domains of learning.

Distance education teaching repertoire involves extensive use of self-instructional materials, be they print or otherwise. These materials are designed to achieve certain pre-determined learning outcomes, namely goals and objectives that are contained in an instructional plan. Since the teaching process is affected over a distance, there is need to ensure that students actively participate in their learning by performing specific tasks that help them to understand the relevant concepts. Therefore, a set of exercises is built into the teaching repertoire in order to link what students and tutors do in the framework of the course outline. These could be in the form of students' assignments, a research project or a science practical exercise. Examples of instructional activities in distance education are too numerous to list. Instructional activities, when used in this context, help to motivate students, guide and measure students' performance (continuous assessment)



PREFACE

We have put in lots of hard work to make this book as user-friendly as possible, but we have not sacrificed quality. Experts were involved in preparing the materials. However, concepts are explained in easy language for you. We have included many tables and examples for easy understanding.

We sincerely hope this book will help you in every way you expect.

All the best for your studies from our team!



FUNDAMENTALS OF OPERATING SYSTEM

Contents

BLOCK 1: INTRODUCTION TO OPERATING SYSTEMS

UNIT 1 BASICS OF OS

Definition and Function of operating systems, Evolution of operating system, Operating system structure-monolithic layered, virtual machine and Client server

UNIT 2 TYPES OF OPERATING SYSTEM

Different types of operating system-real time systems, multi-user System, distributed system

UNIT 3 BATCH OPERATING SYSTEM

Introduction to basic terms and batch processing system: Jobs, Processes files, command interpreter

BLOCK 2: MEMORY MANAGEMENT AND PROCESS SCHEDULING

UNIT 1 MEMORY MANAGEMENT

Logical and Physical address protection, paging, and segmentation, Virtual memory, Page replacement algorithms, Cache memory, hierarchy of memory types, Associative memory

UNIT 2 PROCESS SCHEDULING

Process states, virtual processor, Interrupt mechanism, Scheduling algorithms Performance evaluation of scheduling algorithm, Threads



BLOCK 3: FILE AND I/O MANAGEMENT

UNIT 1 FILE SYSTEM

File systems-Partitions and Directory structure, Disk space allocation, Disk scheduling

UNIT 2 I/O MANAGEMENT

I/O Hardware, I/O Drivers, DMA controlled I/O and programmed I/O, I/O Supervisors

BLOCK 4: BASICS OF DISTRIBUTED OPERATING SYSTEM

UNIT 1 DISTRIBUTED OPERATING SYSTEM

Introduction and need for distributed OS, Architecture of Distributed OS, Models of distributed system

UNIT 2 MORE ON OPERATING SYSTEM

Remote procedure Calls, Distributed shared memory, Unix Operating System: Case Studies



Dr. Babasaheb
Ambedkar
Open University

PGDCA 104

FUNDAMENTALS OF OPERATING SYSTEM

BLOCK 1: INTRODUCTION TO OPERATING SYSTEM

UNIT 1

BASICS OF OS

UNIT 2

TYPES OF OPERATING SYSTEM

UNIT 3

BATCH OPERATING SYSTEM

BLOCK 1: INTRODUCTION TO OPERATING SYSTEMS

Block Introduction

An operating system is important software which makes the computer to run. It handles all the computer processes and runs the hardware. It makes you to communicate with computer without having command on its language. It is seen that your computer operating system manages all software and hardware functions. The main idea of operating system is to coordinate will all processes and links these processes with central processing unit (CPU), memory and storage.

In this block, we will detail about the basic of Operating System and different types of Operating System. The block will focus on the study and concept that led to explanation about Operating System structure. The students will give with the idea about Batch processing system.

In this block, the student will make to learn and understand about the basic of operating system and its functions. The student will be demonstrated practically and theoretically about different types of operating system used.

Block Objective

After learning this block, you will be able to understand:

- About Operating system and its features
- Detailed about different types of O/S and their structure
- Knowledge about batch processing system

Block Structure

Unit 1: Basics of OS

Unit 2: Types of Operating System

Unit 3: Batch Operating System

UNIT 1: BASICS OF OS

Unit Structure

- 1.0 Learning Objectives**
- 1.1 Introduction**
- 1.2 Definition and Function of operating systems**
- 1.3 Evolution of operating system**
- 1.4 Operating system structure-monolithic layered**
- 1.5 Virtual machine and Client server**
- 1.6 Let Us Sum Up**
- 1.7 Answers for Check Your Progress**
- 1.8 Glossary**
- 1.9 Assignment**
- 1.10 Activities**
- 1.11 Case Study**
- 1.12 Further Readings**

1.0 Learning Objectives

After learning this unit, you will be able to understand:

- The various task of an Operating System
- Concept of application program
- Idea about Internal Parts of Operating System
- Idea about DOS Operating System
- Study about Client/Server program

1.1 Introduction

Earlier in 1960's, operating system is software which handles the hardware. Presently, we see operating system as set of programs that create the hardware to work. Generally, operating system is set of programs to facilitate controls of a computer. There are different types of operating systems as UNIX, MS-DOS, MS-Windows, Windows/NT, and VM.

Over protecting of computer engage software at numerous levels. We will distinguish kernel services, library services, as well as application-level services, all of which are division of an operating system. Processes run Applications, which are related together by means of libraries that carry out standard services. The kernel supports the development by providing a path to the peripheral devices. The kernel reacts to service calls as of the processes as well as interrupts from the devices. The centre of the operating system is the kernel, a organize program with the purpose to function in restricted state, act in response to interrupts from external devices as well as service requests along with traps from processes. In order to run Computer hardware, we require an Operating System that will be able to recognise all hardware components and enable us to work on it. In this unit, we will study about Operating system and its evolution along with its necessary role.

1.2 Definition And Function Of Operating Systems

An operating system also known as OS is a software program that enables the computer hardware to communicate and operate with the computer software. Operating systems perform basic tasks:

- Recognizing input from the keyboard
- Sending output to Monitor
- keeping track of files and directories
- Controlling peripheral such as disk drives and printers.

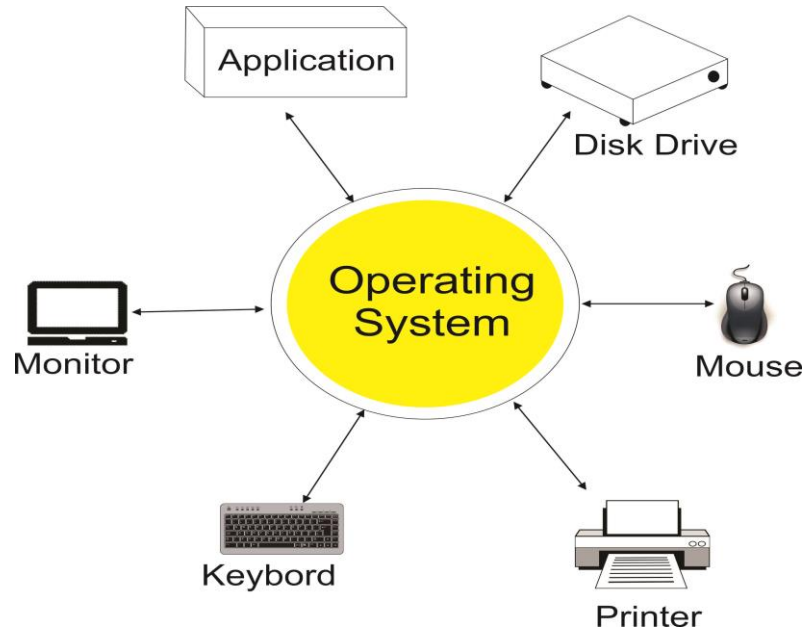


Fig 1.1 Operating System with Computer hardware

The operating system is system software that is stored on the storage device such as hard disk, CD-ROM or floppy disk. When a computer is switched on, the operating system is transferred from the storage device into main memory through ROM.

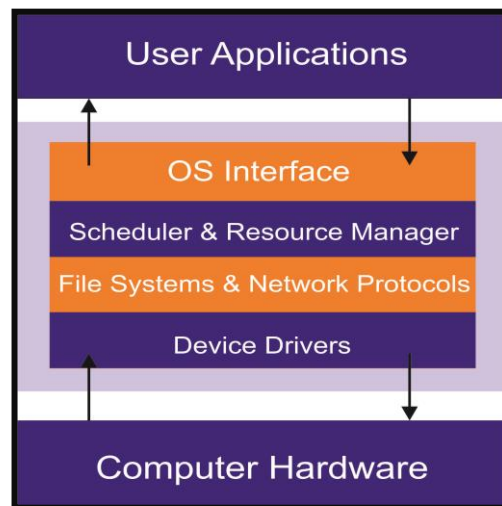


Fig 1.2 Position of Operating System

An operating system controls and coordinates the operations of the computer system. It manages the computer hardware, controls the execution of application programs and provides the set of services to the users. It acts as an interface between user and the computer. The users interact with the operating system indirectly through application program.

The work of the operating system involves:-

- Managing the processor
- Managing Random Access Memory
- Managing Input/output
- Managing execution of applications
- Managing Files
- Controlling Information management

Parts of Operating System

i) Resident part-

It is called as kernel that contains critical functions. It is loaded inside the main memory during the booting. It performs various functions residing in the main memory.

ii) Non-resident part-

This part of operating system is loaded into main memory when required. It includes:

- Disk Operating System (DOS) developed by Microsoft.
- Operating System 2 (OS/2) developed by IBM.
- XENIX or ZENIX developed by Microsoft.
- WTNOWS developed by Microsoft
- WINDOWS- NE

Check your progress 1

1. It is studied that an Operating System is a _____.
 - a. System software
 - b. Stores information on the storage device
 - c. Controls and coordinates the operations of the computer system
 - d. All of above

2. Which among the following is not a function of an operating system?
- a. recognize input from keyboard
 - b. shows output on monitor
 - c. loads keyboard
 - d. track of files

1.3 Evolution of Operating System

Initially, the computer utilises batch operating systems where batches of jobs are run without taking a break. These programs are punched into cards where the processing was performed by copied into tape. After finishing the first job, the computer would soon start with the next job on the tape.

Professional operators when interacted with computer found that users drop such jobs and finally returned to hold the result soon after running of particular job. It was quiet difficult for users as expensive computer were made to involve in such type of processing of jobs.

During late 1960s, invent of timesharing operating systems led to replacement of batch systems. Users when involved directly by way of printing terminal found that Western Electric Teletype shown was ok.

With this time sharing OS, many users shared the computer and then spent only a fraction of second on every job before moving to the next job. It is found that a fast computer will work for many user's jobs at the same time thereby making the illusion that they were full attentive while receiving such jobs.

Printing terminals found that the programs were set of characters or command line user interfaces (CLI) where user had to type responses in order to typed commands which led to scrolled down the instructions on paper.

During mid-1970, the personal computers allows pockets and Altair 8800 were initially used for commercial purposes for an individuals. In the start of 1975, the Altair was sold to hobbyists in kit form. It was without the operating system because it has only toggle switches and light emitting diodes which serves as input and output.

After sometimes, people started connected terminals and floppy disk drives to Altairs. During the year 1976, Digital Research introduced CP/M operating

system for such Computer. CP/M and later on DOS had CLIs which were similar to timeshared operating systems where computer was only for a particular user.

With the success of Apple Macintosh in 1984, the particular system pushed the state of hardware art which were restricted to small with black and white display. As hardware continued to develop, many colour Macs were under developed position and soon Microsoft introduced Windows as its GUI operating system.

It was found that the Macintosh operating system was based on decades of research on graphically-oriented personal computer operating systems and applications. Computer applications today require a single machine to perform many operations and the applications may compete for the resources of the machine. This demands a high degree of coordination which can be handled by system software known as an operating system

The internal part of the OS is often called the kernel which comprises of:

- File Manager
- Device Drivers
- Memory Manager
- Scheduler
- Dispatcher

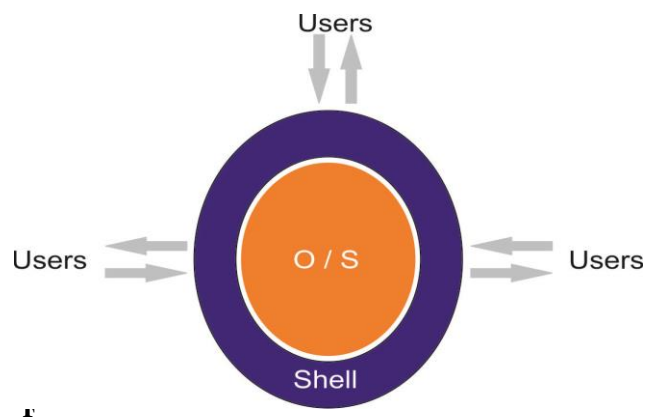


Fig 1.3 Interfaces of OS

Check your progress 2

1. The commercial computer Altairs was developed in the year:
 - a. 1980
 - b. 1970
 - c. 1985
 - d. 1955

1.4 Operating system structure-monolithic layered

The mean of operating system architecture usually follows the leave-taking of particular principle. Such principle guide to re-structure the operating system mainly into relatively independent parts that can be easily provide basic independent features by keeping complicated designs in manageable conditions.

Apart from controlling complexity, the architecture of operating system influences key features that are in terms of robustness or efficiency as:

- The OS receives importance which allows to work if not then protected resources like physical devices or application memory. With such importance, the various related parts of OS or OS as a whole will be both accidental and malicious privileges misuse gets lowered.
- By breaking OS into different parts will led to adverse effect on efficiency since the overhead linked with communication among individual parts be exacerbated when coupled with hardware mechanisms?

Monolithic Systems

Aboriginal concept of the operating system arrangement brings about no definite accommodation for the discriminating nature of the operating system. Furthermore the concept follows the separation of concerns; no action is acted to limit the blessings granted to the single parts of the operating system. The complete operating system acts with maximum approvals. The communication overhead inside the basic operating system is the identical as the communication overhead inside numerous other software, considered relatively low.

It is seen that CP/M and DOS are examples of monolithic operating systems that share common address space with certain applications. It is found in CP/M, 16 bit address space will begins with system variables along with application area additionally ends with 3 parts of O/S which are known as:

- CCP or Console Command Processor
- BDOS or Basic Disk Operating System
- BIOS or Basic Input/output System

If we see that in a DOS Operating System, there exists a 20 bit address space that begins with an array of interrupt vectors along with system variables that are followed by local DOS and its application area which will end with memory block utilised by video card and BIOS as shown in fig 1.4.

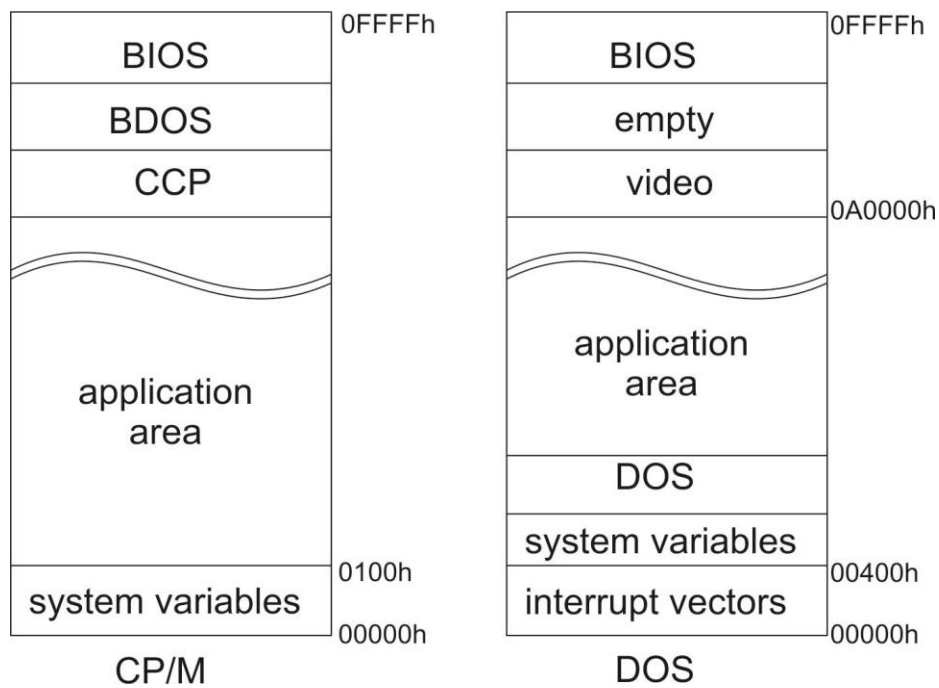


Fig 1.4: Monolithic Operating Systems

Check your progress 3

1. Which is not a part of Operating System?

- a. CCP
- b. BDOS
- c. BIOS
- d. DOS

1.5 Virtual machine and Client server

Virtual machine

A virtual machine (VM) abides an operating system OS or conduct environment that is embedded on software which copies consecrated hardware. The end user embraces the equivalent experience on a virtual machine as they would acquire on dedicated hardware.

Individualized software designated a hypervisor copies the PC client or server’s CPU, memory, hard disk and network as well as other hardware resources collectively, allowing virtual appliances to participate the resources. The hypervisor can copy integral virtual hardware platforms that are occasional from each other, assigning virtual machines to run Linux as well as Windows server operating systems on the identical underlying physical aggregation. Virtualization conserves costs by depreciating the need for physical hardware systems. Virtual machines additional desirably use hardware, which lowers the quantities of

hardware as well as associated maintenance costs, along with reduces power furthermore cooling demand. They also allay management due to virtual hardware does not collapse. Administrators can take advantage of virtual circumstances to simplify backups, disaster recovery, new deployments as well as elementary system administration tasks.

Virtual machines do not constrain distinguished hypervisor-specific hardware. Virtualization appears although require more bandwidth, storage along with processing capacity than a conventional server or desktop if the physical hardware is going to host multiple running virtual machines. VMs can easily move, be copied and reassigned between host servers to optimize hardware resource utilization. Because VMs on a physical host can consume unequal resource quantities (one may hog the available physical storage while another stores little), IT professionals must balance VMs with available resources.

Client server

Client/server is a program relationship in which one program (the client) requests a service or resource from another program (the server).

It is seen that in client/server model, the programs are used by single computer only. It serves as an important concept for networking. Here, the client makes a connection with the server through local area network (LAN) or wide-area network (WAN) like Internet. After clearing the client's request, the connection gets terminated. In this case the Web browser serves as a client program which further appeals for a service from the server. The service and resource of the server will show the delivery of such Web page.

Computer assignments in which the server accomplishes a request created by a client are very customary furthermore the client/server model has served one of the main concepts of network computing. Most business approaches facilitate the client/server model as appears acts the Internet's core program, TCP/IP. For exemplary, when you examine your bank account from your computer, a client approximation in your computer overtures a request to a server program at the bank. That program may in twist forward an approach to its own client program, which that time conveys, a request to a database server at another bank computer. Once your account balance sheet has been acquired from the database, it is acknowledged back to the bank data client, which in turn applies it back to the client in your personal computer, which that time displays the information to you.

Both client programs as well as server programs are usual constituent of a larger program or application. On account of multiple client programs participate

the services of the equivalent server program, a special server identified a daemon may be charged due to anticipate client requests. In marketing, the client/server had been once used to differentiate allocated computing by personal computers (PCs) from the monolithic, concentrated computing model exercised by mainframes. This differentiation has largely evaporated, although, as mainframes along with their applications possess additionally turned to the client/server model further become part of network computing.

Check your progress 4

1. Virtual machine can run:
 - a. Windows
 - b. Linux
 - c. DOS
 - d. all
2. In a Client/server program:
 - a. one program requests a service from another program
 - b. one program requests a copy to another program
 - c. one program requests a hardware to run another program
 - d. one program requests a software to load operating system

1.6 Let Us Sum Up

In this unit, we have learned:

- That an operating system or OS is software program that enables the computer hardware to communicate and operate with the computer software.
- We see that there are many functions of an operating system which will help in managing:
 1. Processor.
 2. Random Access Memory:
 3. Input/output
 4. Execution of applications

5. Files
6. Information management

- A virtual machine (VM) abides an operating system OS or conduct environment that is embedded on software which copies consecrated hardware.
- Client/server is a program relationship in which one program (the client) requests a service or resource from another program (the server).

1.7 Answers for Check Your Progress

Check your progress 1

Answers: (1-a), (2-c)

Check your progress 2

Answers: (1-b)

Check your progress 3

Answers: (1-d)

Check your progress 4

Answers: (1-a), (2-a)

1.8 Glossary

1. **Shell** - It makes interface with user
2. **File Manager** - It manages mass memory
3. **Device Drivers** - These carry different drives for various peripherals
4. **Memory Manager** - It handles the main memory
5. **Scheduler and Dispatcher** - It helps in managing the processes

1.9 Assignment

Write note on Client Operating System?

1.10 Activities

Establish a Client System in Linux Operating System.

1.11 Case Study

Can an Operating System be able to handle all Computer Hardware?

1.12 Further Readings

1. The Operating system by Andrew Tannenbaum.
2. Operating System by Mach.

UNIT 2: Types of Operating System

Unit Structure

- 2.0 Learning Objectives**
- 2.1 Introduction**
- 2.2 Different types of operating system**
 - 2.2.1 Real time Systems
 - 2.2.2 Multi-user System
 - 2.2.3 Distributed system.
- 2.3 Let Us Sum Up**
- 2.4 Answers for Check Your Progress**
- 2.5 Glossary**
- 2.6 Assignment**
- 2.7 Activities**
- 2.8 Case Study**
- 2.9 Further Readings**

2.0 Learning Objectives

After learning this unit, you will be able to understand:

- Various types of Operating System
- About Batch processing system
- Concept of Multi-user System
- Distributed System

2.1 Introduction

There are abundant Operating Systems those monopolize be constructed for functioning the performances those are demanded by the user. There are ample Operating Systems which acquire the ability to behave the entreaties those are acquired from the approach. The Operating system can behave a unique Operation

furthermore also multiple movements at duration. Hence there are numerous categories of Operating systems those are arranged by utilizing their Working mechanisms.

2.2 Different Types of Operating System

There are many types of operating system such as:

- 1) Serial Processing
- 2) Batch Processing
- 3) Multi-Programming
- 4) Real Time System
- 5) Distributed Operating System
- 6) Multiprocessing
- 7) Parallel operating systems

2.2.1 Real time Systems

There occurs additionally an Operating System which is comprehended as Real Time Processing System. In this acknowledgment duration is already adjusted. Indicates duration to show the after-effects after acquiring has adjusted by the Processor or CPU. Real Time System is exercised at those areas in which we binds higher along with well-timed return. These categories of approaches are exercised in reservation. Hence when we discriminate the demand, the CPU will conduct at that duration. There are two Types of Real Time System:

- **Hard Real Time System:** In the Hard Real Time System, Time is fixed and we can't Change any Moments of the Time of Processing. Means CPU will Process the data as we Enters the Data.
- **Soft Real Time System:** In the Soft Real Time System, some Moments can be Change. Means after giving the Command to the CPU, CPU Performs the Operation after a Microsecond.

2.2.2 Multi-user System

As we comprehend that in the Batch Processing System there are multiple jobs appear by the System. The System foremost compose a batch furthermore later that he will accomplish all the jobs those are saved into the Batch. Furthermore the innermost difficulty is that if a mechanism or job needs an Input as well as Output Operation, that time it is not achievable and second there will be the wastage of the duration when we are composing the batch as well as the CPU will continue idle at that duration.

Although with the help of Multi programming we can achieve Multiple Programs on the System at a duration besides in the Multi-programming the CPU determination never get idle, hence with the help of Multi-Programming we can achieve ample algorithms on the System including When we are functioning with the Program that time we can additionally acknowledge the supplement or Another Program for sprinting additionally the CPU will that time behave the secondary Program following the completion of the original Program. Additionally in this we can further differentiate our Input means a user can additionally interact with the System.

The Multi-programming Operating Systems never utilize numerous cards on account of the approach is accessed on the Spot by the user. But the Operating System also utilizes the Process of Allocation and De-allocation of the Memory Means he will provide the Memory Space to all the Running and all the Waiting Processes. There must be the Proper Management of all the Running Jobs.

2.2.3 Distributed system

Distributed Means Data is Stored and Processed on Multiple Locations. When a Data is stored on to the Multiple Computers, those are placed in Different Locations. Distributed means In the Network, Network Collections of Computers are connected with Each other.

Then if we want to Take Some Data from other Computer, Then we use the Distributed Processing System. And we can also Insert and Remove the Data from out Location to another Location. In this Data is shared between many users. And we can also Access all the Input and Output Devices are also accessed by Multiple Users.

Check your progress 1

1. In Hard Real Time System, Time:
 - a. varies
 - b. fixed
 - c. zero
 - d. none of these
2. Real Time System is used when we require:
 - a. delay time return
 - b. no time constrain
 - c. timely return
 - d. none of these

2.3 Let Us Sum Up

In this unit, we have learned:

- About Different Types of Operating System
 1. Real Time System is an Operating system which works to achieve timely return
 2. Soft Real Time System is a part of Real Time O/S where each moment changes
 3. Multi programming is a programming technique where many programs perform

2.4 Answers for Check Your Progress

Check your progress 1

Answers: (1-b), (2-c)

2.5 Glossary

1. **Real Time System** - It is an Operating system which is exercised to get timely return
2. **Soft Real Time System** - It is a type of Real Time O/S where each moment changes

3. **Multi programming** - it a programming techniques where many programs perform

2.6 Assignment

Write note on Batch processing Operating System?

2.7 Activities

Explain the cycle of operation of Real Time Operating System

2.8 Case Study

Can a multiuser Operating System be installed on the server?

2.9 Further Readings

1. The Operating system by Andrew Tannenbaum.
2. Operating System by Mach.

UNIT 3: BATCH OPERATING SYSTEM

Unit Structure

3.0 Learning Objectives

3.1 Introduction

3.2 Basic terms

3.3 Batch processing system

3.3.1 Jobs

3.3.2 Processes files

3.3.3 Command interpreter

3.4 Let Us Sum Up

3.5 Answers for Check Your Progress

3.6 Glossary

3.7 Assignment

3.8 Activities

3.9 Case Study

3.10 Further Readings

3.0 Learning Objectives

After learning this unit, you will be able to understand:

- The concept of Batch
- Concept about Batch Processing
- Idea about File Processing

3.1 Introduction

Batch is the term which is given to the work of doing similar jobs continuously again and again but with a difference as in this the input data is shown for every iteration of the job and probably the output file.

Batch operating system is a kind of operating system requirement which mainly involves in mainframe computer that was used with the intention that it may perform great repetitive data processing work. It is analysed that a mainframe set is used to process 30 million pension statements which belongs to individual customers.

It is found that a batch job requires no intervention by a person once the initial commands are set up. Setting up a batch job is similar to filling out a form, with specific details required to be shown.

3.2 Basic terms

Databases

Batch processing is additionally applied for capable bigness database updates as well as automated transaction processing, as compared to collaborative online transaction processing (OLTP) approaches. The extract, transform, load (ETL) extent in populating data warehouses is inherently a batch approach in most implementations.

Images

Batch processing is often used to perform various operations with digital images such as resize, convert, watermark, or otherwise edit image files.

Conversions

Batch processing may additionally be applied for altering computer files from one format to another. For exemplary a batch job may change proprietary as well as legacy files to conventional standard formats for end-user queries along with display.

Batch window

A batch window continues “a duration of less-intensive online assignment”, when the computer system is accomplished to plunge batch jobs without obstruction from online systems.

Numerous untimely computer systems granted sole batch processing; hence jobs could be plunge any time within a 24-hour day. With the accomplishment of assignment processing the online approaches might singular is required from 9:00 a.m. to 5:00 p.m., abstracting dual shifts obtainable for batch work, in this casket the batch window would be sixteen hours. The difficulty is not normally that the computer system is inadequate of admitting combined online along with batch work, although that the batch systems normally constrain approach to data in an integrated state, released from online updates until the batch processing is complete.

In a bank, for exemplary, so-called end-of-day (EOD) jobs constitute interest calculation, duration of reports as well as data sets to disjoint systems, printing statements, as well as payment processing.

Check your progress 1

1. Which is not a batch job?
 - a. Priority of the Job
 - b. Uses CPU hogging
 - c. Avoids infinite printouts
 - d. Data details for input and output

3.3 Batch processing system

A batch processing approach is one where facts are assorted together in a batch preceding processing begins. A batch procedure behaves as an account of commands in arrangement. It be scramble by a computer's operating system facilitating a script or batch file, or may be accomplished within a system utilizing a macro or inner scripting tool. The mechanism of data entry for premature computers existed facilitating punched cards, which were experienced in batches, further on account of the term batch processing. Each bit of work for a batch processing system is designated a job.

Jobs are assigning up so they can race to accomplishment without human intercommunication. All entryway parameters are predefined through scripts, command-line arguments, control files, or job control jargon. This is in contradicting to "online" or to-and-fro programs which advice the user for similar

input. A program acquires a portion of data files as input, processes the data, as well as brings about a set of output facts files. This operating arrangement is identified as “batch processing” since the input data are acquired into batches or sets of records as well as each batch endures processed as a unit. The output exists another batch that can be reused for assessment.

Batch processing has been affiliated with mainframe computers owing to the earliest decades of electronic computing in the 1950s. There was a multifariousness of reasons why batch processing commanded premature computing. One logic is that the foremost bustling business problems for analyses of profitability as well as competitiveness were initially accounting problems, like as billing. Billing may effectively be appeared as a batch-oriented business process, along with appropriately every business constraining bill, reliably as well as on-time. Furthermore, every computing resource had been costly; hence consecutive submission of batch jobs on punched cards matched the resource constraints as well as technology evolution at the duration. Later, interactive sessions with coupled text-based computer terminal interfaces or graphical user interfaces became additional common. Furthermore, computers originally were not even cogent of having multiple programs loaded into the main memory.

Batch processing is still pervasive in mainframe computing, but practically all types of computers are now capable of at least some batch processing, even if only for “housekeeping” tasks. These include UNIX-based computers, Microsoft Windows, Mac OS X and even Smartphone’s. Increasingly, as computing in general becomes more pervasive batch processing is unlikely to lose its significance.

Batch approaches are still fault-finding in maximum organizations in big part on account of many common business processes are adaptable to batch processing. While online systems can additionally function when manual facilitation is not expected, they are not definitely optimized to harmonize high-volume, consecutive tasks. hence, even new systems commonly contain one or more batch approaches for updating information at the accomplishment of the day, generating reports, printing documents, as well as other non-interactive efforts that inevitable fulfil reliably within assured business deadlines.

Modern batch applications make utilize of modern batch architectures like as Jem the Bee or Spring Batch, which is composed for Java, as well as irrelevance frameworks for external programming languages, to deliver the defect tolerance as well as scalability necessary for high-volume processing. In steadiness to promise high-speed processing, batch applications are habitual

integrated with grid computing solutions to measure a batch job above a large number of processors; however there are relevant programming conflicts in doing so? High volume batch processing grounds particularly heavy demands on system along with application architectures as well. Architectures that feature energetic input/output performance as well as vertical scalability, along with modern mainframe computers, tend to cater better batch performance than alternatives.

Batch processing is most suitable for tasks where a large amount of data has to be processed on a regular basis.

Examples

- A. payroll systems
- B. examination report card systems

Advantages

- Once the data are submitted for processing, the computer may be left running without human interaction.
- The computer is only used for a certain period of time for the batch job.
- Jobs can be scheduled for a time when the computer is not busy.

Disadvantages

- There is always a delay before work is processed and returned.
- Batch processing usually involves an expensive computer and a large number of trained staff.

3.3.1 Jobs

In batch processing, job contains relevantly common group of processing along with calculation actions that utilizes small or very less cooperation among you along with the computer system. When a batch job is acknowledged, that time the job will primarily enter in a job queue where it will functionally halt till the system captures ready to process the next job. Here the system formed off its processing mechanism of job when it acquires the job from the job sequence. A batch job is put in a job queue by:

- Choosing a menu option that submits a batch job
- Submitting a job into the system using the SBMJOB command

It is found that a job queue carries several work or jobs which are halted for the system to process them. Your job waits while the system processes other jobs that other users submit prior to your job or have a higher priority. When system resources are available, the system processes your job.

3.3.2 Processes files

A conventional activity is to conduct a set of equivalent operations on data sets in a group of files. This endures batch processing. For exemplary, you might desire to read multiple Excel files from a plate reader approximate concentrations from a common curve, fit a four parameter logistic application to duplicate data in each file, construct a graph of the data besides fit, along with export assured fit parameters to an Excel results file. On account of the automation language in Sigma Plot is transcribed for Visual Basic that time implementing a batch processing activity is purely a factor of writing a Sigma Plot macro or a Visual Basic program.

An exceptional exemplary of a batch processing program is the Batch Process Excel Files macro in Sigma Plot. The dialog from this macro is demonstrated. It is considered as a basic design around which you can construct your own macro to appear a definite task. It approves you select a group of Excel files by clicking on the Add File button that invokes the file open argument, fit an equation to a data domain in the Excel file that you discriminate as well as bring about a graph along with report of the results. In this casket the results for each file are accredited in a distinct section of a Sigma Plot notebook. This can be adjusted to establish the results in an Excel file if you desire.

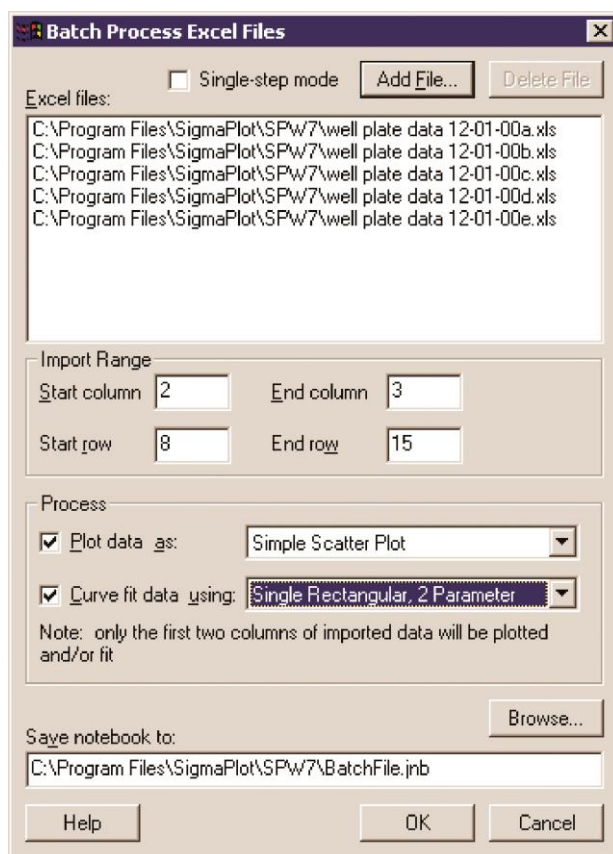


Fig 3.1 Dialog for Batch Process Excel Files macro.

Each of the Excel files from a well plate reader observes prefer the one demonstrated in Figure 3.2. Five duplicate measurements of definite binding are demonstrated in columns C through G. For tutorial approaches, the liberated radio ligand entrancement has been accumulated in column B. The macro has been written to appropriate an equation to two columns of data so for this example we will ignore the replicates. It is at ease to change the macro to encompass the row wise duplicate format in the curve fit.

	A	B	C	D	E	F	G	H
1								
2	Machine Name:	X123	User Name:	SPSS				
3	Protocol:	Gen1						
4	Comments:	demo						
5	Plate Number:	2						
6	Time:	12:01:00	1/23/2001					
7			1	2	3	4	5	6
8	A	5	34.81	29.93	38.49	26.69	33.42	
9	B	10	49.97	51.03	51.88	48.17	47.79	
10	C	30	76.75	79.18	75.57	72.73	74.47	
11	D	50	89.03	80.75	84.79	87.36	84.52	
12	E	70	93.09	90.10	88.60	84.93	90.65	
13	F	90	97.95	84.49	80.35	93.47	91.53	
14	G	110	90.49	94.15	97.24	94.07	91.81	
15	H	130	94.84	90.13	92.34	98.10	94.38	
16								

Fig 3.2 one of the excel files to analyse.

You can then select the appropriate region of the Excel file containing the data to fit. This is shown in Figure 3.3 for the data in Figure 3.2.

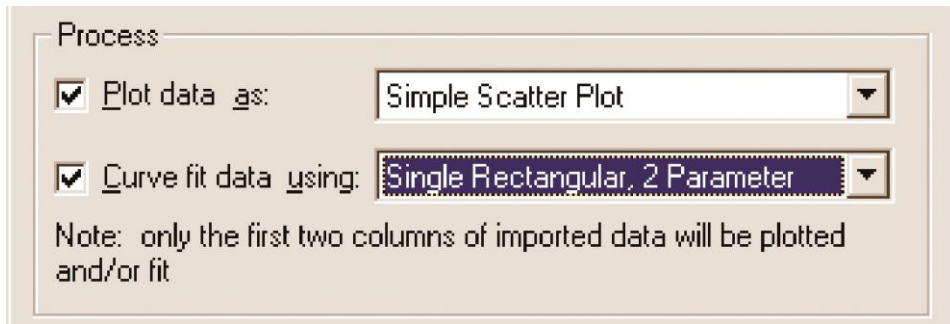


Fig 3.3 data in Figure

The distorted function is chosen to fit each data set additionally a simple scatter plot continues used to demonstrate the results. Note that every equation in the Sigma Plot curve suited library is accounted in the dropdown box in Figure 3.4. It is comfortable to conduct this on account of Sigma Plot Automation authorizes you to look for a notebook (in this case the standard.jfl notebook containing all the curve fit equations) for benefit objects (or objects of any type) as well as create a list of them. If you expect you can displaced in this macro a contrary notebook with another group of fit equations. The new equations will that time display in the dropdown list. If a user-defined equation is acquired to common. jf l that time it will display in the list. The batch process effects are that time saved in a notebook. You may browse to select the suitable file.



Fig 3.4 Specify the notebook to save the results.

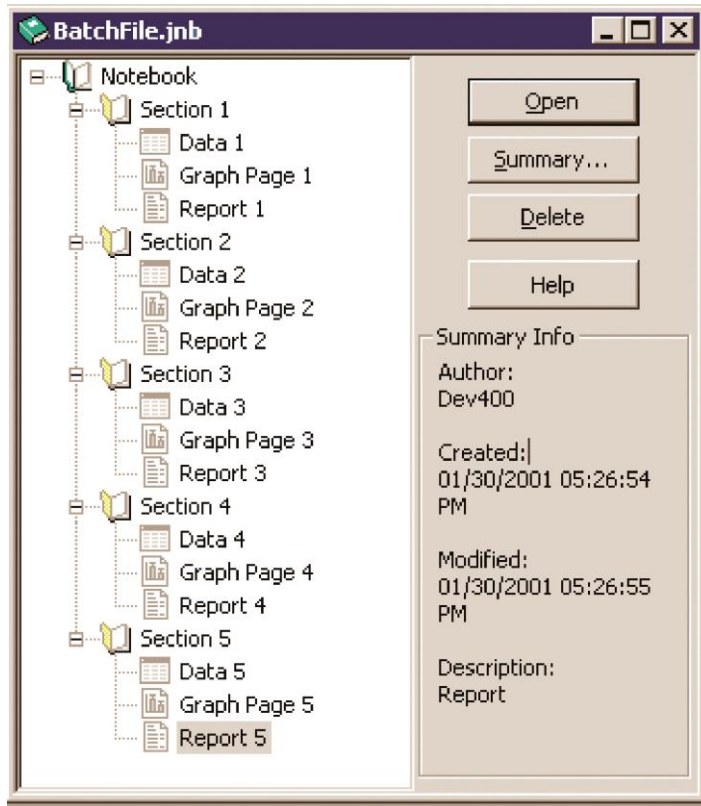


Fig 3.5 Batch File

For the five files shown in Figure 3.5, the notebook contains five sections each with worksheets with individual data sets, scatter plots of the data, fit results and detailed curve fit reports.

3.3.3 Command interpreter

A command interpreter, or command processor, occurs that critical component of the operating system software that analyses, or programs, the commands you allocate, additionally that time delivers them out for you. In DOS, the command processor is normally COMMAND. COM, furthermore DOS allows you alternate another command processor if you expect. This cries out slightly technical as well as scary, however it really isn't complicated.

Check your progress 2

1. In an operating system, a batch procedure works as:

- | | |
|-----------------|----------|
| a. instructions | c. rules |
| b. commands | d. all |

- | | |
|--|------------------|
| 2. In a program, the data files acts as: | |
| a. input | c. processes |
| b. output | d. none of these |
| 3. Batch processing is concern with_____computers. | |
| a. analog | c. mainframe |
| b. hybrid | d. all |

3.4 Let Us Sum Up

In this unit, we have learned:

- Batch operating system is a kind of operating system requirement which mainly involves in mainframe computer
- Batch processing may additionally be applied for altering computer files from one format to another
- Batch processing has been affiliated with mainframe computers owing to the earliest decades of electronic computing

3.5 Answers for Check Your Progress

Check your progress 1

Answers: (1- b)

Check your progress 2

Answers: (1 - d), (2 - a), (3 - c)

3.6 Glossary

1. **Batch** - It is term that describes the amount of the work to do particular jobs continuously.
2. **Batch operating system** - A kind of operating system need mainly involves in mainframe computer.

3.7 Assignment

Write short details on Batch data.

3.8 Activities

Study the different techniques of Batch jobs.

3.9 Case Study

How will the Computer be able to perform Batch Jobs?

3.10 Further Readings

1. The Operating system by Andrew Tannenbaum.
2. Operating System by Mach.

Block Summary

In this block, we have studied about the basic of Operating System and different types of Operating System. We have an idea about the necessary function and advantages of using an operating system. There are abundant Operating Systems those monopolize be constructed for functioning the performances those are demanded by the user. Batch is the term which is given to the work of doing similar jobs continuously again and again but with a difference as in this the input data is shown for every iteration of the job and probably the output file.

The block detailed about the concept that explains the usability and structuring about Operating System. In this block students have given an idea about Batch processing system. The block focuses on practical implications of operating system.

Block Assignment

Short Answer Questions

1. What is an Operating System?
2. Explain Real time Systems?
3. What is batch processing?
4. What is Distributed system?
5. Explain Multi-user System?

Long Answer Questions

1. Discuss the different types of Operating System?
2. What is the need of Multi-programming Operating System?
3. What are the advantages of Batch processing System?

Enrolment No.

1. How many hours did you need for studying the units?

Unit No	1	2	3	4
Nos of Hrs				

2. Please give your reactions to the following items based on your reading of the block:

Items	Excellent	Very Good	Good	Poor	Give specific example if any
Presentation Quality	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____ _____
Language and Style	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____ _____
Illustration used (Diagram, tables etc)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____ _____
Conceptual Clarity	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____ _____
Check your progress Quest	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____ _____
Feed back to CYP Question	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____ _____

3. Any Other Comments

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....



“

*Education is something
which ought to be
brought within
the reach of every one.*

”

- Dr. B. R. Ambedkar



Dr. Babasaheb Ambedkar Open University
Jyotirmay' Parisar, Opp. Shri Balaji Temple, Sarkhej-Gandhinagar Highway, Chharodi,
Ahmedabad-382 481.

FUNDAMENTALS OF OPERATING SYSTEM

PGDCA 104

BLOCK 2: MEMORY MANAGEMENT AND PROCESS SCHEDULING

Dr. Babasaheb Ambedkar Open University



FUNDAMENTALS OF OPERATING SYSTEM



Knowledge Management and
Research Organization
Pune



Editorial Panel

Author

Er. Nishit Mathur

Language Editor

Prof. Jaipal Gaikwad

Graphic and Creative Panel

Ms. K. Jamdal

Ms. Lata Dawange

Ms. Pinaz Driver

Ms. Ashwini Wankhede

Mr. Kiran Shinde

Mr. Prashant Tikone

Mr. Akshay Mirajkar

Copyright © 2015 Knowledge Management and Research Organization.

All rights reserved. No part of this book may be reproduced, transmitted or utilized in any form or by a means, electronic or mechanical, including photocopying, recording or by any information storage or retrieval system without written permission from us.

Acknowledgment

Every attempt has been made to trace the copyright holders of material reproduced in this book. Should an infringement have occurred, we apologize for the same and will be pleased to make necessary correction/amendment in future edition of this book.

The content is developed by taking reference of online and print publications that are mentioned in Bibliography. The content developed represents the breadth of research excellence in this multidisciplinary academic field. Some of the information, illustrations and examples are taken "as is" and as available in the references mentioned in Bibliography for academic purpose and better understanding by learner.'



ROLE OF SELF INSTRUCTIONAL MATERIAL IN DISTANCE LEARNING

The need to plan effective instruction is imperative for a successful distance teaching repertoire. This is due to the fact that the instructional designer, the tutor, the author (s) and the student are often separated by distance and may never meet in person. This is an increasingly common scenario in distance education instruction. As much as possible, teaching by distance should stimulate the student's intellectual involvement and contain all the necessary learning instructional activities that are capable of guiding the student through the course objectives. Therefore, the course / self-instructional material are completely equipped with everything that the syllabus prescribes.

To ensure effective instruction, a number of instructional design ideas are used and these help students to acquire knowledge, intellectual skills, motor skills and necessary attitudinal changes. In this respect, students' assessment and course evaluation are incorporated in the text.

The nature of instructional activities used in distance education self-instructional materials depends on the domain of learning that they reinforce in the text, that is, the cognitive, psychomotor and affective. These are further interpreted in the acquisition of knowledge, intellectual skills and motor skills. Students may be encouraged to gain, apply and communicate (orally or in writing) the knowledge acquired. Intellectual-skills objectives may be met by designing instructions that make use of students' prior knowledge and experiences in the discourse as the foundation on which newly acquired knowledge is built.

The provision of exercises in the form of assignments, projects and tutorial feedback is necessary. Instructional activities that teach motor skills need to be graphically demonstrated and the correct practices provided during tutorials. Instructional activities for inculcating change in attitude and behavior should create interest and demonstrate need and benefits gained by adopting the required change. Information on the adoption and procedures for practice of new attitudes may then be introduced.

Teaching and learning at a distance eliminates interactive communication cues, such as pauses, intonation and gestures, associated with the face-to-face method of teaching. This is particularly so with the exclusive use of print media. Instructional activities built into the instructional repertoire provide this missing interaction between the student and the teacher. Therefore, the use of instructional activities to affect better distance teaching is not optional, but mandatory.

Our team of successful writers and authors has tried to reduce this.

Divide and to bring this Self Instructional Material as the best teaching and communication tool. Instructional activities are varied in order to assess the different facets of the domains of learning.

Distance education teaching repertoire involves extensive use of self-instructional materials, be they print or otherwise. These materials are designed to achieve certain pre-determined learning outcomes, namely goals and objectives that are contained in an instructional plan. Since the teaching process is affected over a distance, there is need to ensure that students actively participate in their learning by performing specific tasks that help them to understand the relevant concepts. Therefore, a set of exercises is built into the teaching repertoire in order to link what students and tutors do in the framework of the course outline. These could be in the form of students' assignments, a research project or a science practical exercise. Examples of instructional activities in distance education are too numerous to list. Instructional activities, when used in this context, help to motivate students, guide and measure students' performance (continuous assessment)



PREFACE

We have put in lots of hard work to make this book as user-friendly as possible, but we have not sacrificed quality. Experts were involved in preparing the materials. However, concepts are explained in easy language for you. We have included many tables and examples for easy understanding.

We sincerely hope this book will help you in every way you expect.

All the best for your studies from our team!



FUNDAMENTALS OF OPERATING SYSTEM

Contents

BLOCK 1: INTRODUCTION TO OPERATING SYSTEMS

UNIT 1 BASICS OF OS

Definition and Function of operating systems, Evolution of operating system, Operating system structure-monolithic layered, virtual machine and Client server

UNIT 2 TYPES OF OPERATING SYSTEM

Different types of operating system-real time systems, multi-user System, distributed system

UNIT 3 BATCH OPERATING SYSTEM

Introduction to basic terms and batch processing system: Jobs, Processes files, command interpreter

BLOCK 2: MEMORY MANAGEMENT AND PROCESS SCHEDULING

UNIT 1 MEMORY MANAGEMENT

Logical and Physical address protection, paging, and segmentation, Virtual memory, Page replacement algorithms, Cache memory, hierarchy of memory types, Associative memory

UNIT 2 PROCESS SCHEDULING

Process states, virtual processor, Interrupt mechanism, Scheduling algorithms Performance evaluation of scheduling algorithm, Threads



BLOCK 3: FILE AND I/O MANAGEMENT

UNIT 1 FILE SYSTEM

File systems-Partitions and Directory structure, Disk space allocation, Disk scheduling

UNIT 2 I/O MANAGEMENT

I/O Hardware, I/O Drivers, DMA controlled I/O and programmed I/O, I/O Supervisors

BLOCK 4: BASICS OF DISTRIBUTED OPERATING SYSTEM

UNIT 1 DISTRIBUTED OPERATING SYSTEM

Introduction and need for distributed OS, Architecture of Distributed OS, Models of distributed system

UNIT 2 MORE ON OPERATING SYSTEM

Remote procedure Calls, Distributed shared memory, Unix Operating System: Case Studies



Dr. Babasaheb
Ambedkar
Open University

PGDCA 104

FUNDAMENTALS OF OPERATING SYSTEM

BLOCK 2: MEMORY MANAGEMENT AND PROCESS SCHEDULING

UNIT 1

MEMORY MANAGEMENT

UNIT 2

PROCESS SCHEDULING

BLOCK 2: MEMORY MANAGEMENT AND PROCESS SCHEDULING

Block Introduction

An operating system is important software which makes the computer to run. It handles all the computer processes and runs the hardware. It makes you to communicate with computer without having command on its language. It is seen that your computer operating system manages all software and hardware functions. The main idea of operating system is to coordinate will all processes and links these processes with central processing unit (CPU), memory and storage.

In this block, we will discuss detail about the basic of memory management and process scheduling of Operating System. The block will focus on the study and concept of virtual memory, paging and segmentation. The students will give with the idea about Catch memory and virtual processor.

In this block, the student will make to learn and understand about the basic of memory management techniques and its techniques. The concept related to memory hierarchy, process state and interrupt mechanism will also be explained to the students. The student will be demonstrated practically about the working of page replace algorithm and its technique.

Block Objective

After learning this block, you will be able to understand:

- About Logical and Physical address protection of memory.
- Study about paging and segmentation.
- Idea about virtual memory.
- Detailed on page replace algorithms.
- Knowledge about different type of memory hierarchy.
- Concept of process states.
- Generalization of virtual processor.

Memory
Management and
Process
Scheduling

- Basic of interrupt mechanism.

Block Structure

Unit 1: Memory Management

Unit 2: Process Scheduling

UNIT 1: MEMORY MANAGEMENT

Unit Structure

- 1.0 Learning Objectives
- 1.1 Introduction
- 1.2 Logical and Physical Address Protection
- 1.3 Paging and Segmentation
- 1.4 Virtual Memory
- 1.5 Page Replacement Algorithms
- 1.6 Cache Memory
- 1.7 Hierarchy of Memory Types
- 1.8 Associative Memory
- 1.9 Let Us Sum Up
- 1.10 Answers for Check Your Progress
- 1.11 Glossary
- 1.12 Assignment
- 1.13 Activities
- 1.14 Case Study
- 1.15 Further Readings

1.0 Learning Objectives

After learning this unit, you will be able to understand:

- Memory Management Unit
- External and internal fragmentation
- Virtual page number
- Paging address Translation
- Virtual and physical memory
- Importance of Cache memory
- Associative memory

1.1 Introduction

Memory management is a type of subsystem which is an important part of an operating system. During the computing period there was continuously need of more memory in computer systems. Strategies have been developed to overcome this limitation and the most successful of these is virtual memory. Virtual memory makes the system appear to have more memory than it actually has by sharing it between competing processes as they need it. With earlier computing it is found that:

- Program should be carryout into the memory and is kept inside a process for further working.
- Input queue which is collecting of processes information on the disk are brought into the memory for implementation.
- The single process implementation will do from input queue which is loaded inside the memory for implementation.
- Once the implementation is done, then the implementation memory space will become free.
- In computers, the address space starts with 00000, which was first address of the user process that cannot be all 0.

1.2 Logical and Physical Address Protection

In the meantime, the computer acquires inter communicated through logical as well as physical addressing in order to map its memory. The logical approach is developed by the processor which is additionally designated as virtual address. The program observes this address space. Whereas the physical address exists the real address which continues assumed by the computer hardware like as memory unit. It is determined that the logical to physical address analysis occurrence acted or conveyed off is experienced by the Operating System. Here, virtual memory designates to the absence of estrangement of logical memory which is glanced nearby the process from physical memory which is inspected nearby the processor. On approximate of this separation, the computer programmer desires to be careful of about logical memory space while the operating system affirms two or more levels of physical memory space.

It is found that during the compile time and load time, the address binding schemes makes these two tend similar but they differ in execution time address

binding scheme and MMU (Memory Management Unit) that caters the translation of such addresses.

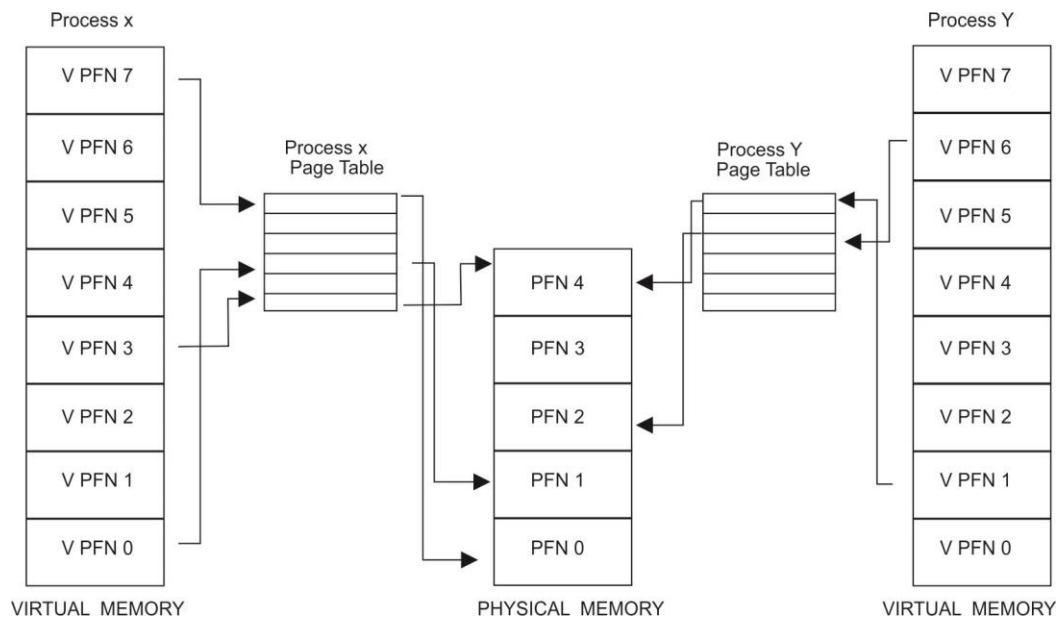


Fig 1.1 Virtual to Physical address mapping

In fig 1.1, it is looked that every procedure in the system comprises its own virtual address space. Similarly virtual address spaces are entirely alienated from each other furthermore on account of a process bounding one application cannot authorize another. Additionally, the hardware virtual memory approaches assign regions of memory to be maintained across writing. This conserves code as well as data from being overwritten by miscreant applications.

In the virtual to physical address mapping demonstrated in fig 1.1, as the processor deliver out the program it understands an instruction from memory furthermore analyse it. While the processor interprets the instruction, it desires to fetch or store the contents of a position in the memory. Following that, the processor will deliver out the instruction furthermore actuate onto the following instruction in the program. With this concept, the processor is consecutively approaching memory to deliver out the instructions or to acquire as well as accumulate the data.

In rack relevantly virtual memory system, complete similar addresses are virtual addresses in addition are not physical addresses. The virtual addresses acquires altered into physical addresses with the support of a processor based on

information that had been conveyed in a set of tables to preserve safe the operating system.

To simplify this, virtual as well as physical memory are allotted as handy sized chunks recognized as pages. Similarly chunks are of identical size where it serves difficult for the system to monitor as Linux on Alpha AXP systems utilizes 8 KB pages as well as on Intel x86 systems it applies 4 KB pages. Every chunk of pages are allotted with a unique number as page frame number (PFN).

Under such model, a virtual address is made of two parts:

- Offset
- Virtual page frame number

If the page size is 4 KB, then the bit ratio will be 11:0 of virtual address which has offset and bits number 12 and over this are virtual page frame number. Every time the processor comes across a virtual address which takes the offset and the virtual page frame number. For this, the processor should translate virtual page frame number into physical frame number and contact the location for correct offset into physical page by using page tables.

It is found that the virtual memory will allow process to be of virtual address spaces, so that there are times when you need processes to share such memory. Now the processor uses virtual page frame number as an index into the processes page table to get back its page table entry. If the entry is valid, then the processor will carry out the physical page frame number from such entry. If the entry is not a valid entry, then the process will try to come out with a non-existing area of its virtual memory. Under such conditions, the processor cannot solve the address and will pass the control to the operating system which is a permanent process.

The concept of logical address space that is bound to a separate physical address space is central to proper memory management.

- Logical address – generated by the CPU; also referred to as virtual address
- Physical address – address seen by the memory unit

Logical and physical addresses are the same in compile-time and load-time address-binding schemes; logical (virtual) and physical addresses differ in execution-time address-binding scheme

Check your progress 1

- In logical address protection, the logical address is generated by the _____.
 a. memory
 b. processor
 c. memory address
 d. none of these
- Input queue is a collection of process _____ on the disk
 a. data
 b. information
 c. both a and b
 d. neither a nor b
- Linux on Alpha AXP systems utilizes ___KB pages
 a. 8
 b. 16
 c. 32
 d. 64

1.3 Paging and Segmentation

Paging

Paging is a process that will help in solving the problem that was seen in case of variable sized partitions such as external fragmentation. In paged system, the logical memory is sliced into number of constant sizes chunks called as pages. Further, the physical memory is pre-divided in certain constant sized blocks which are known as page frames. The page sizes or the frame sizes will be of power 2, and fluctuates between 512 bytes to 8192 bytes per page. They have certain bytes per page because of the implementation of paging mechanism with page number and page offset.

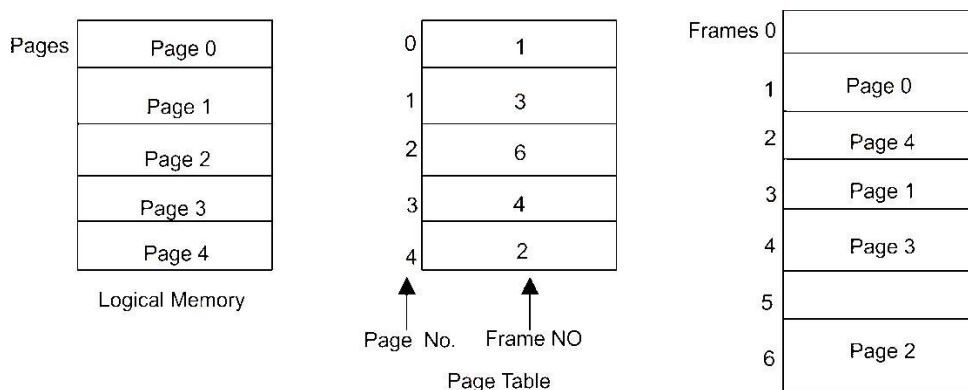


Fig 1.2 Paging operation

In fig 1.2, the process page gets loaded to particular memory frame. Such pages will further be loaded into neighboring frames or in non-neighboring memory as highlighted in the figure 1.2 It is seen that the outside fragmentation gets improved because the processes gets inside in a separate holes.

Page Allocation

With variable sized partitioning of memory, it is seen that every time a process of n size is loaded which is the best location from the list of available/free holes. Such type of dynamic storage allocation is required as it increases the efficiency and throughput of system. This type of selection can be done by using:

- 1) Best-fit Policy: It allocates the hole where the process is tight as the difference between whole size and process size is lowest.
- 2) First-fit Policy: This will allocates the initial found hole that can be big enough to fit in the new process.
- 3) Worst-fit Policy: It allocates the maximum size hole which leaves the full amount of unused space.

From the above three listed strategies it seems that the strategy best-fit and first-fit are better as compared to the worst-fit. Both best-fit and first-fit strategies are efficient in terms of time and storage capacity. In case of best-fit strategy, minimum leftover space is seen which will create the smallest hole which are not used frequently. In case of first-fit strategy, it uses least overheads in order to work because it is the simplest strategy to work upon. Possibly worst-fit also sometimes leaves large holes that could further be used to accommodate other processes. Thus all these policies have their own merits and demerits.

Hardware Support for Paging

It is seen that all logical page in paging scheme is further divided as:

- Page number (p) in logical address space
- Displacement in page pat which item resides

Such arrangement is known as Address Translation scheme as it shows that in case of a 16-bit address, we can divide the address as:

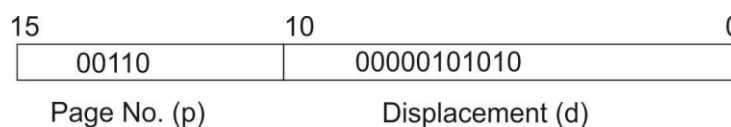


Fig 1.3 dividing the address

From the figure 1.3, it is seen that a page number will take 5 bits with its range starts from value 0 to 31 that can be 2^5-1 . Likewise, if we consider an offset value of having 11-bits, then the range will be from 0 to 2023 which is $2^{11}-1$. Totally, we see that the paging scheme uses 32 pages, each having 2024 locations. Also, the table that keeps virtual address to physical address translations is further classified as page table. It is found that as the displacement is fixed, the translation of virtual page number to physical page exists which can be seen in the figure 1.4.

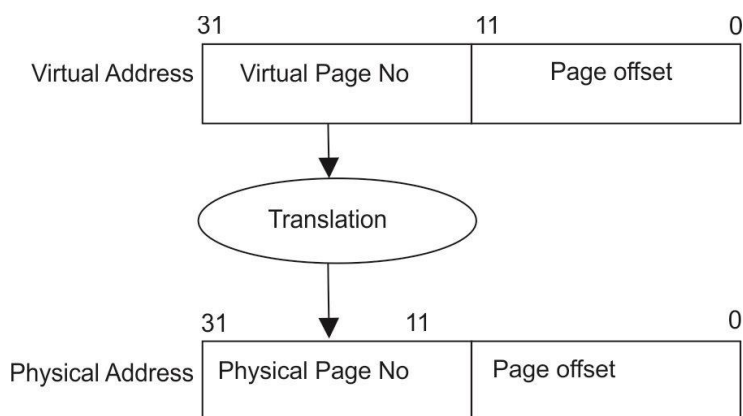


Fig 1.4 Address Translation scheme

It is seen that the page number is required in shape of an index which is into the page table containing base address for every corresponding physical memory page number. This arrangement will lowers the dynamic relocation efforts which are shown by the paging hardware support as in figure 1.5.

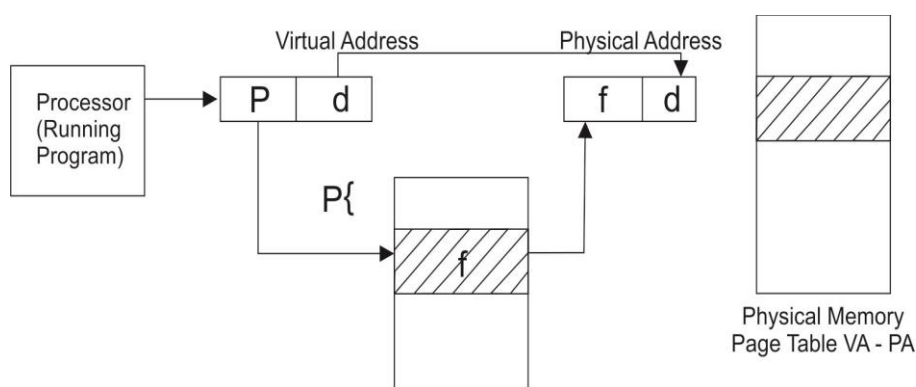


Fig 1.5 Direct Mapping

Paging address Translation by direct mapping

Consider a case of direct mapping as shown in fig 1.5 where a page table sends directly to physical memory page. In this, the drawback is that the speed of translation decreases because the page table is put in primary storage place having a considerably larger in size that increases the instruction execution time and led to lowering of system speed. In order to conquer such situation, the use of extra hardware such as registers and buffers are used.

Paging Address Translation with Associative Mapping

It is based on the utilization of fixed registers that has high speed and efficiency. Such small, fast-lookup Catch will help to put the whole page table in content-addresses associative storage place thereby making the speed to improve and further to care for the lookup problem of the Catch. These are known as associative registers or Translation Look-aside Buffers (TLB's). It is found that each register consists of two entries:

- 1) Key, which is matched with logical page.
- 2) Value which returns page frame number corresponding top.

Such arrangement is same as direct mapping scheme but only difference is that we have associative registers having few page table entries that made the search fast. It is quite expensive due to the presence of register support. Hence it is found that both direct and associative mapping schemes will combine to result in more benefits. In this, that page number is coordinated with associative registers at the same time. Also the percentage of number of times the page is found in TLB's is further termed as hit ratio. If it is not found, it is seek out in page table and added into TLB. In case if the TLB is full, then the page replacement policies will come into effect. It is found that the entry in TLB is limited only. Such type of combined scheme is shown in Figure 1.6.

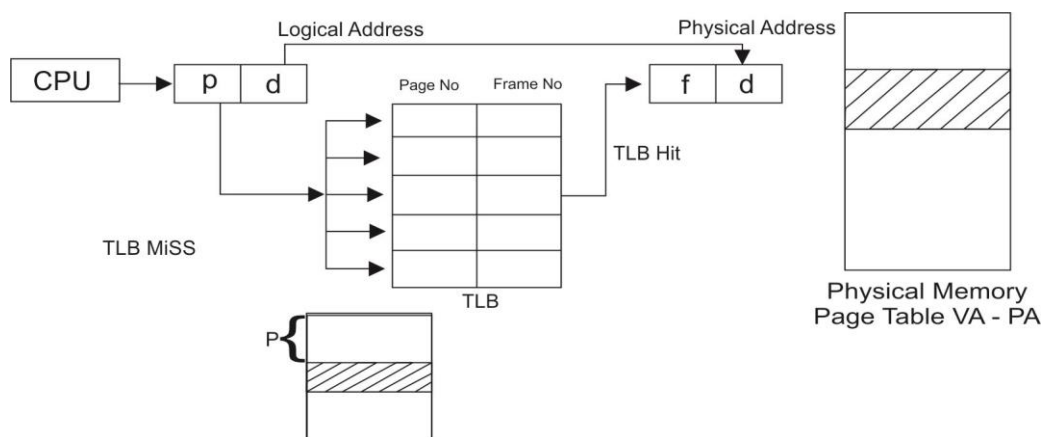


Fig 1.6 Type of combined scheme

It is seen that in paging hardware, there is a presence of some protection mechanism. Inside the page table there exists corresponding frame where a protection bit is linked. Such type of bit will show whether the page is read-only or read-write. In this, sharing code and data will take place only when two page table entries in different process show the similar physical page where every process shares the memory. It is seen that if one process writes the data, then the second process will locate for the changes. Such type of an arrangement is quite efficient while in communicating. Sharing is required to control in order to protect modification and admission of data in a single process with the help of second process. Such type of programs is kept independent as procedures and data where procedures and data that are pure/reentrant code get shared. Re-entrant code will not be able to change itself and should make sure that it contains a separate copy of per-process global variables. It is predicted that modifiable data and procedures will not share without the intervention of concurrency controls. Such type of non-modifiable procedures sometimes are called as pure procedures or reentrant codes. In case of an example, it is illustrated that in such system only single copy of editor or compiler code be kept in the memory, and all editor or compiler processes and executes sit with the help of single copy of code which will help in memory utilization.

Advantages

There are certain advantages of paging scheme such as:

1. Virtual address space must be greater than main memory size. i.e., can execute program with large logical address space as compared with physical address space.
2. Avoid external fragmentation and hence storage compaction.
3. Full utilization of available main storage.

Disadvantages

The disadvantages of paging scheme include:

1. Internal fragmentation problem led to wastage inside the allocated page
2. Extra resource consumption
3. Overheads for paging hardware
4. Virtual address to physical address translation takes place

Segmentation

In generic, a consumer or a programmer likes to observe system memory as an assembly of variable-sized allocations rather than as a linear arrangement of words. Segmentation occurs as a memory management arrangement that accepts this glance of memory.

Principles of Operation

Segmentation demonstrates an exchange arrangement for memory management. This arrangement bisects the logical address space into variable length allocations, named segments, with no appropriate sequencing among them. Each allotment has a name and a length. For clarity, segments are acknowledged by a segment number, rather than by a name. Hence, the logical addresses are acknowledged as a pair of segment number as well as offset within segment. It empowers a program to be broken down into feasible parts according to the user opinion of the memory, which is that time mapped into physical memory. Furthermore logical addresses are two-dimensional although actual physical addresses are still one-dimensional arrangement of bytes only.

Address Translation

This mapping between two is done by segment table, which contains segment base and its limit. The segment base has starting physical address of segment, and segment limit provides the length of segment. This scheme is depicted in Figure 1.7.

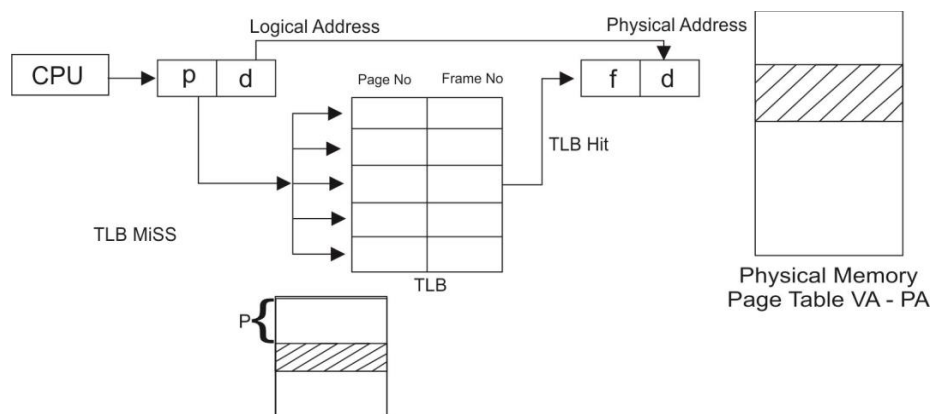


Fig 1.7 Address translation

The offset d must range between 0 and segment limit/length, otherwise it will generate address error. For example, consider situation shown in Figure 1.8.

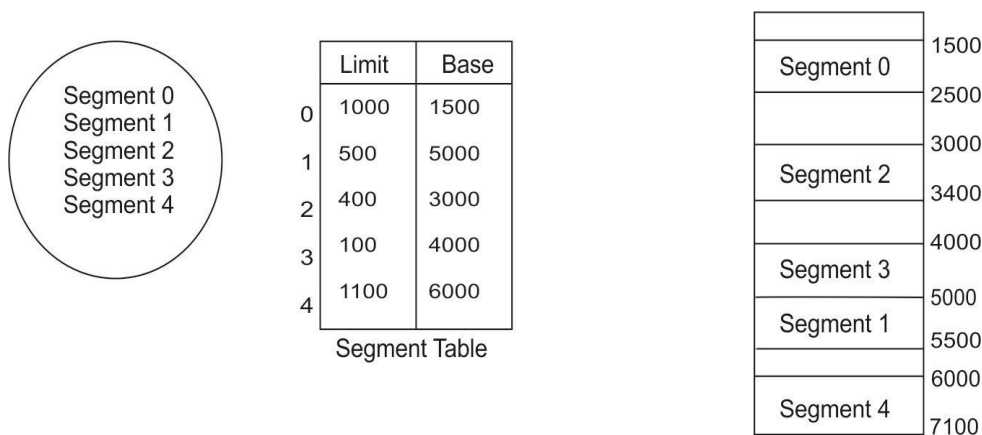


Fig 1.8 Principle of operation of representation

This approximation is comparable to adaptable partition allocation method with advancement that the process is bifurcated into parts. For quick retrieval we can utilize registers as in paged approach. This is comprehended as a segment-table length register (STLR). The segments in a segmentation mechanism dispatch to logical divisions of the process additionally are described by program names. Extract the segment number along with offset from logical address originally so that time the use of segment number as index into segment table gets capture segment base address along with its limit /length. Additionally, contemplate that the offset is not greater than allocated limit in segment table. Today, normally physical address is acquired by adding the offset to the base address.

Protection and Sharing

This approach in addition enables segments that are read-only to be allotted, so that two approaches can utilize shared code for advance memory efficiency. The intervention is comparable that no program can read from or write to chunks belonging to another program, except the allocations that have been set up to be apportioned. With each segment-table entry safety bit differentiating segment as read-only or execute unique can be employed. So fallacious attempt to write into a read-only segment can easily be preserved.

Sharing of segments can be accomplished by constructing common /same entries in segment tables of two asymmetric processes which point to equivalent physical location. Segmentation may continue from external fragmentation i.e., when blocks of released memory are not sufficient to adjust a segment. Storage compression as well as coalescing can shorten this barrier.

Check your progress 2

1. The page sizes or frame sizes is in the range of-
 - a. 512 bytes to 8192 bytes per page
 - b. 128 bytes to 512 bytes per page
 - c. 1024 bytes to 2048 bytes per page
 - d. 2048 bytes to 4098 bytes per page
2. In Best-fit Policy, the difference between hole size and process size is-
 - a. maximum
 - b. lowest
 - c. half
 - d. none of these

1.4 Virtual Memory

Virtual memory is an approach that empowers the accomplishment of processes which are not comprehensively obtainable in memory. The core clear benefit of this approach is that programs can be extended than physical memory. Virtual memory is the division of user logical memory from physical memory.

This division authorizes an intensely large virtual memory to be delivered for programmers when only a smaller physical memory is obtainable. Following are the circumstances, when complete program is not essential to be loaded completely in main memory.

- User written error handling routines are used only when an error occurred in the data or computation.
- Certain options and features of a program may be used rarely.
- Many tables are assigned a fixed amount of address space even though only a small amount of the table is actually used.
- The ability to execute a program that is only partially in memory would counter many benefits.
- Less number of I/O would be needed to load or swap each user program into memory.

- A program would no longer be constrained by the amount of physical memory that is available.
- Each user program could take less physical memory; more programs could be run the same time, with a corresponding increase in CPU utilization and throughput.

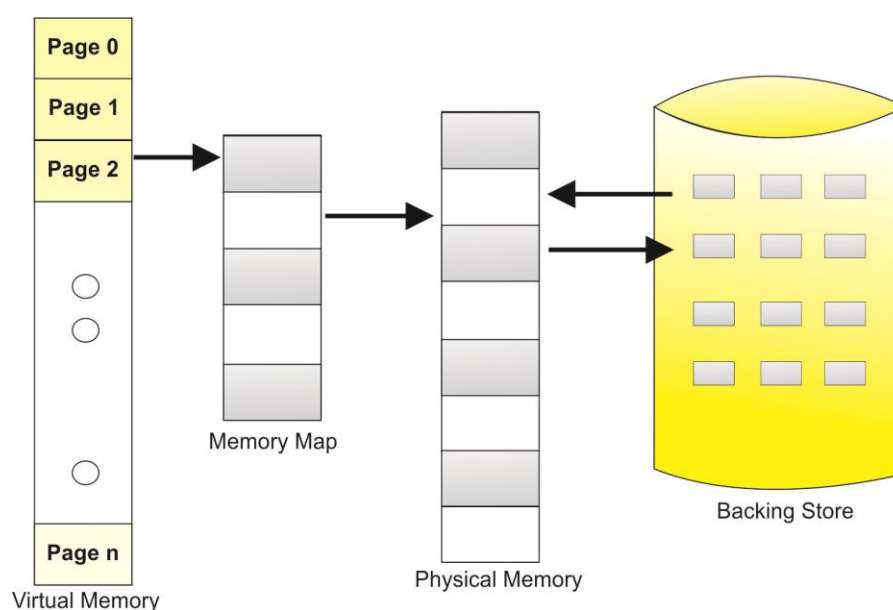


Fig 1.9 Virtual memory

Virtual memory is frequently exercised by demand paging. It can additionally be exercised in a segmentation system. Demand segmentation can further be utilized to supply virtual memory.

A demand paging mechanism is quite comparable to a paging system with exchanging. When we expect to achieve a process, we exchange it into memory. Rather than exchanging the complete process into memory, furthermore, we facilitate a lazy swapper called pager.

When a process is to be exchanged in, the pager conceives which pages will be facilitated before the process is exchanged out again. Instead of exchanging in a whole process, the pager carries only those essential pages into memory. So, it bypasses reading into memory pages that will not be used in anyway, shortening the swap time as well as the amount of physical memory expected.

Hardware support is essential to discriminate between those pages that are in memory as well as those pages that are on the disk employing the valid-invalid character scheme where correct as well as defective pages can be examined by checking the bit. Marking a page will hold no effect if the process never

approaches to approach the page. While the process achieves as well as accesses pages that are memory resident, execution approaches predominantly.

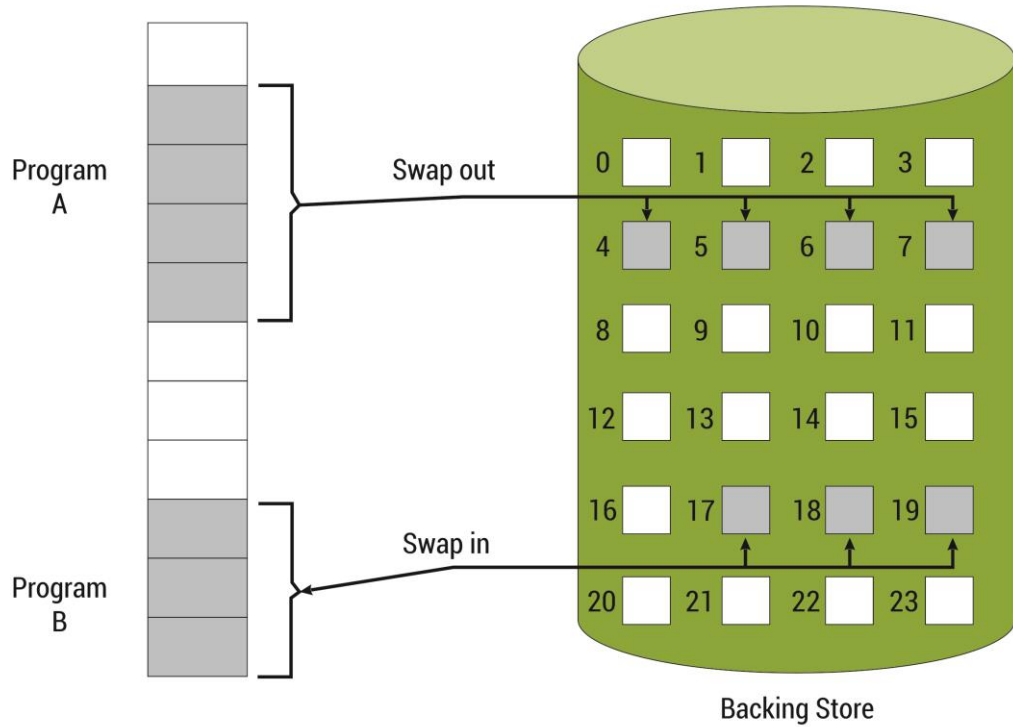


Fig 1.10 Demand paging system

Check your progress 3

1. The presence of virtual memory helps to share the memory among them.
 - a. processes
 - b. threads
 - c. instructions
 - d. none of the mentioned
2. _____ is the concept where a process is copied into main memory from secondary memory as required.
 - a. Paging
 - b. Demand paging
 - c. Segmentation
 - d. Swapping
3. Swap space is present in:
 - a. primary memory
 - b. secondary memory
 - c. CPU
 - d. none of the mentioned

1.5 Page Replacement Algorithms

Page replacement algorithms are the mechanisms exercising which Operating System determines which memory pages to exchange out, write to disk when a page of memory expects to be assigned. Paging occurs whenever a page fault arises also a free page cannot be facilitated for allocation approach accounting to analysis that pages are not obtainable or the number of free pages is shorten than necessary pages.

When the page that was chosen for exchanged and was paged out, is referenced again that time it has to read in from disk, furthermore this stipulates for I/O completion. This approach considers the quality of the page replacement algorithm: the lesser the time subsiding for page-ins, the better is the algorithm. A page replacement algorithm beholds at the edged information about approaching the pages delivered by hardware, additionally tries to choose which pages should be replaced to shorten the total number of page lacks, while balancing it with the costs of primary storage along with processor time of the algorithm itself. There are abundant different page replacement algorithms. We calculate an algorithm by bounding it on a definite string of memory reference as well as assessing the number of page faults.

RAND (Random)

- Determine some page to change at random.
- Affirms the following page to be referenced exists random.
- Can check breach algorithms across random page exchanged.

MIN (minimum) or OPT (optimal)

- Belady's optimal algorithm for the minimal number of page defects.
- Change the page that will be referenced best in the future or not at all.
- Problem: we cannot apply it, since we cannot forecast the future.
- This is the better case.
- Can exercise it to match external algorithms against

FIFO (First In, First Out)

- Choose the page that has been in main memory the longest.
- Exercise a chain (data structure).

- Problem: however a page has been residing for a long time, it may be absolutely useful.
- Windows NT as well as Windows 2000 utilize this algorithm, as a local page replacement algorithm (explained distinctly), with the pool approach (described in more detail separately).
- Construct a bay of the pages that have been labeled for removal.
- Manage the pool in the identical way as the rest of the pages.
- If a latest page is expected, take a page from the pool.
- If a page in the bay is referenced again foregoing being replaced in memory, it is clearly reactivated.
- This is relatively efficient.

LRU (Least Recently Used)

- Select the page that was final referenced the longest time ago.
- Affirms current behavior is a good guru of the immediate future.
- Can control LRU with a list identify the LRU stack or the paging stack (data structure).
- In the LRU stack, the initial entry explains the page referenced least recently, the last entry describes to the last page referenced.
- If a page is referenced, proceed it to the end of the list.
- Problem: stipulates updating on every page referenced.
- Too slow to be used in practice for controlling the page table, however many systems use assessments to LRU.

NRU (Not Recently Used)

- As an appraisal to LRU, choose one of the pages that has not been exercised currently (as opposed to identifying exactly which one has not been employed for the longest amount of time).
- Save one bit identified the "used bit" or "reference bit", where 1 => used recently and 0 => not used recently.
- Variants of this scheme are exercised in numerous operating systems, involving UNIX along with Macintosh.

- Most variations facilitate a scan pointer and pass through the page frames one by one, in some order, inspecting for a page that has not been used currently.

Working Set (WS)

- Clearly address the problem of thrashing.
- Thrashing: when the computer system is compulsive with paging, i.e., CPU has little to do but there is heavy disk traffic moving pages to and from memory with little use of those pages.
- Working set: the pages that a process has used in the last w time intervals.
- Choose any page that is not in the working set.
- Global: not in the working set of much ready process.
- If no such page exists, swap out some process.
- The medium-term scheduler places a process in the waiting-for-memory queue.

Working Set Policy

- Block the number of processes in the develop list so that whole can have their functioning set of pages in memory.
- Before beginning a process, make sure it's working set is in main memory.
- Too costly in practice, however there are some good approximations.

Page Fault Frequency algorithm (PFF) -- dissimilarity of Working Set

- When a page fault exists, if the last page fault for that process was fresh, that time increase the size of its working set (up to a maximum).
- All processes begin with a default ws dimension.
- Load original code pages, original data pages, original stack pages.
- If a process holds not faulted currently, ease the size of its ws i.e. exclude all pages not used currently ("used bit").
- Exercised in Windows NT as well as Windows 2000 as a complete page replacement algorithm (described separately).
- They assign to it as automatic working-set trimming.
- Additionally, in WinNT, can call the process object service to alter working-set min as well as max for a process, up to a defined max as well as min .

Check your progress 4

1. Page replacement algorithms decide:
 - a. which memory pages to be exchanged
 - b. which segment pages to be exchanged
 - c. which data pages to be exchanged
 - d. all
2. In FIFO page replacement algorithm, the page to be replaced_____.
 - a. with oldest page selected
 - b. with new page selected
 - c. random page selected
 - d. none
3. Which algorithm selects a page that was not used for a long period whenever a page is replaced?
 - a. first in first out algorithm
 - b. additional reference bit algorithm
 - c. least recently used algorithm
 - d. counting based page replacement algorithm

1.6 Cache Memory

The Cache Memory exists in the Memory which is very nearest to the CPU, complete the current Instructions are saved into the Cache Memory. The Cache Memory is connected for storing the input which is allotted by the user additionally which is essential for the CPU to play a work. But the size of the Cache Memory is additionally low in contrast to Memory as well as Hard Disk.

Importance of Cache memory

The Cache memory lies in the direction between the processor as well as the memory. The Cache memory hence, has a shorter access time than memory further is faster than the main memory. A Cache memory acquires an access time of 100ns, while the main memory may acquire an access time of 700ns.

The Cache memory is very costly moreover owing to subsists limited in capacity. Earlier Cache memories were practicable individually but the microprocessors include the Cache memory on the chip itself.

Expectation for the Cache memory is just to the mismatch between the speeds of the main memory as well as the CPU. The CPU clock as lectured earlier is very fast, whereas the main memory access time is contrastingly slower. Therefore, no matter how fast the processor is, the processing speed depends additionally on the speed of the main memory (the energy of a chain is the energy of its weakest link). It is on account of this analysis that a Cache memory acquires access time closer to the processor speed that is created.

The Cache memory stores the program (or its part) currently being executed or which may be executed within a short period of time. The Cache memory additionally accumulates temporary data that the CPU may commonly stipulate for manipulation.

The Cache memory performs according to diversified algorithms, which determine what information it acquires to store. These algorithms work out the chance to adopt which data would be most repeatedly expected. This probability is worked out on the basis of past attestations.

It appears as a high speed buffer between CPU along with main memory additionally is used to temporary store very energetic data and action all along processing because the Cache memory is faster than main memory, the processing speed is elevated by making the data as well as instructions desired in current processing available in Cache. The Cache memory is very costly and therefore is bordered in capacity.

Check your progress 5

1. The closest memory to the CPU is:

- | | |
|--------|----------|
| a. RAM | c. Cache |
| b. ROM | d. all |

2. A Cache memory acquires an access time of:

- | | |
|----------|----------|
| a. 100ns | c. 350ns |
| b. 700ns | d. 500ns |

1.7 Hierarchy of Memory Types

Memory hierarchy is employed in computer architecture when chattering behaviour events in computer architectural idea, algorithm predictions, as well as the compact level programming composes such as confounding locality of reference. A "memory hierarchy" in computer storage discriminates each level in the "hierarchy" by response time. There are physically various brands of memory acquiring significant asymmetries in the time to read or write the contents of a peculiar position in memory, the measure of information that is read or written on an allotted condition, the complete volume of information that can be stored, along with the unit amounts of storing an assigned amount of information. To optimize its operation as well as to capture greater efficiency along with economy, memory is arranged in a hierarchy with the greatest performance as well as in universal the best high-priced devices at the top, as well as with progressively lessen performance additionally less costly devices in following layers as shown in fig 1.11 The contents of a certain memory hierarchy, along with the way in which data flows between immediate layers, might be arranged as results.

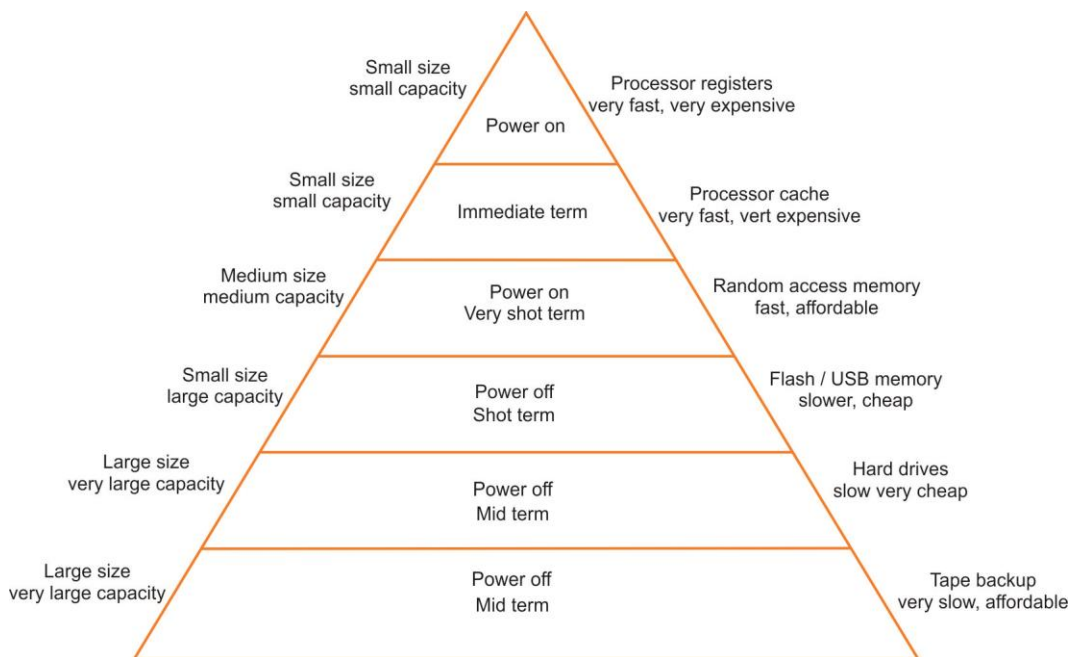


Fig 1.11 Memory hierarchy

Register

A single word confined in each register of the processor; definitely a word includes 4 bytes. This is sometimes not considered of as chunk of the hierarchy.

Catch

These are bunches of words within the Catch; definitely an individual group in the Catch will gather 64 words (say 256 bytes), along with there will be, say, 1024 alike groups, assigning a complete Catch of 256 KBs. Individual word flows between the Catch as well as registers within the processor. All transfers into further out of the Catch are controlled completely by hardware.

Main memory

Words within the main (random-access) memory. On a very high performance system, groups of words acknowledging to a group within the Catch are conveyed between the Catch as well as the main memory in a unique cycle of main memory. On lower-performance systems the dimension of the group of words in the Catch is more than the width of the memory bus, along with the transfer takes the category of a chain of memory cycles. The algorithm that administers this movement is exercised completely in hardware. Main memory measurements are very variable – from as little as 1 GB on a compact system up to numerous GB on a high-performance system.

Online backing store

Blocks of words confined on permanently connected backing store. There may be bilateral somewhat abnormal forms of activity here:(a) swapping device – pages (of say 4 KBs) or segments (up to many GBs) of memory acquired on a swapping device are carried as comprehensive units between their backing-store home along with a page frame or segment domain in main memory, underneath the control of an algorithm applied by the software of the operating system furthermore with hardware assistance to denote when pages or segments are to be actuated;(b) backing store – comprehensive files, or apparently identifiable subsections of big files, are coursed between the backing-store device along with the main memory in acknowledgment to accurate actions by the programmer, normally by a supervisor call to the operating system.

Demountable storage

Comprehensive files, assisted up onto removable disks or magnetic tape within the file archive system along with the archiving system. Accomplished files are preceded in both directions. The development of backup copies along

with the improvement of a backed-up file may be automatic, or may stipulate direct facilitation by the end user. For additional systems the backup agency is definitely a changed feature of a video or audio cassette system, perchance elevated in some structure of computer-controlled cassette-handling robot assessment. Smaller systems may conduct a cassette system or floppy disks.

Read-only library

Accomplished files, as well as collections of affiliated files connecting to an individual application, contained on read-only devices alike as CD-ROM, or on a machine with numerous appearance of write-protection discipline. Complete measures of files are read into the mechanism from the read-only device, however for distinct reasons there are never any transits from the system to the device.

Check your progress 6

1. Register is of:

- | | |
|------------|-------------|
| a. 5 bytes | c. 7 bytes |
| b. 4 bytes | d. 16 bytes |

2. Cache memory contains _____ bytes.

- | | |
|--------------|---------------|
| a. 128 bytes | c. 256 bytes |
| b. 512 bytes | d. 1024 bytes |

1.8 Associative Memory

Memory that is approached by content rather than by address; content addressable is for the time being applied synonymously. An Associative Memory authorizes its users to discriminate part of a pattern or key as well as acquire the values affiliated with that model.

An associative memory is a content-addressable architecture that maps a portion of input prototypes to a set of output prototypes. There are two categories of associative memory:

- auto associative
- hetero associative

An auto associative memory accumulates a formerly stored prototype that most immediately looks like the today's prototype. In a hetero associative memory, the accumulated prototype is, in general, distinct from the input prototype not only in content yet perhaps furthermore in type as well as format.

In 1988, Kosko enlarged the Hopfield example by encompassing an incremented layer to act ceaseless auto associations as favourably as hetero associations on the Catch memories. The network architecture of the bi-directional associative memory (BAM) example resembles to that of the linear associate although the connections are bi-directional, i.e. BAM allocates forward as well as backward transfer of information between the layers. The BAM example can conduct either auto associative as well as hetero associative remembers of stored information.

Check your progress 7

1. In auto associative memory, the stored prototype looks like:
 - a. current prototype
 - b. defined prototype
 - c. accumulated prototype
 - d. all

1.9 Let Us Sum Up

In this unit, we have learned:

- That memory management is a type of subsystem which is an important part of an operating system.
- Input queue which is collecting of processes information on the disk.
- The page sizes or the frame sizes will be of power 2, and fluctuates between 512 bytes to 8192 bytes per page.
- Segmentation occurs as a memory management arrangement that accepts this glance of memory.
- Virtual memory is frequently exercised by demand paging.
- Page replacement algorithms are the mechanisms exercising which Operating System determines which memory pages to exchange out.
- The Catch Memory exists as Memory which is very nearest to the CPU.

1.10 Answers for Check Your Progress

Check your progress 1

Answers: (1-b), (2-c), (3-a)

Check your progress 2

Answers: (1-a), (2-b)

Check your progress 3

Answers: (1-a), (2-b), (3-c)

Check your progress 4

Answers: (1-a), (2-a), (3-c)

Check your progress 5

Answers: (1-c), (2-a)

Check your progress 6

Answers: (1-b), (2-c)

Check your progress 7

Answers: (1-c)

1.11 Glossary

1. **Memory hierarchy** - Refers to different types of memory.
2. **Catch Memory** - It is the closest memory available for the CPU.

1.12 Assignment

What are the four important tasks of a memory manager?

1.13 Activities

What are the three tricks used to resolve absolute addresses?

1.14 Case Study

What are the problems that arise with absolute addresses in terms of swapping?

1.15 Further Readings

1. The Operating system by Andrew Tannenbaum.
2. Operating System by Mach.

UNIT 2: PROCESS SCHEDULING

Unit Structure

- 2.0 Learning Objectives**
- 2.1 Introduction**
- 2.2 Process States**
- 2.3 Virtual Processor**
- 2.4 Interrupt Mechanism**
- 2.5 Scheduling Algorithms And Its Performance**
- 2.6 Threads**
- 2.7 Let Us Sum Up**
- 2.8 Answers For Check Your Progress**
- 2.9 Glossary**
- 2.10 Assignment**
- 2.11 Activities**
- 2.12 Case Study**
- 2.13 Further Readings**

2.0 Learning Objectives

After learning this unit, you will be able to understand:

- Concept of process scheduling
- Idea about primitive operating systems
- Basic of process states
- Concept of parallel processing
- Brief on scheduling algorithms

2.1 Introduction

Maximum systems have a great figure of processes with abrupt CPU bursts bracketed between I/O requests as well as a little figure of processes with elongated CPU bursts. To allow good time-sharing behaviour, we may pre-empt a moving process to allow another one flow. The arrange list, additionally comprehended as a run chain, in the operating system preserves a history of complete processes that are eager to run moreover not blocked on input/output or another blocking system demand, alike as a semaphore. The entries in this document are pointers to a procedure control block, which accumulates all information besides state about a process.

When an I/O approach for a process is accomplish, the process behaves from the waiting state to the ready state further acquires placed on the run chain.

The process scheduler is the constituent of the operating system that is concerned for adopting whether the recently running process should extend running moreover, if not, which process should flow next. There are four conditions that may happen where the scheduler needs to place in too make this decision:

The fresh process flows from the running to the waiting condition due to it issues an I/O request or numerous operating system demand that cannot be satisfied currently. The recent process halts.

A timer interrupt drives the scheduler to run as well as decide that a process acquires run for its allocated duration of time as well as it is time to proceed it from the active to the develop state.

An I/O operation is accomplished for an approach that demanded it besides the process here and now moves from the halting to the warm up state. The scheduler may that time choose to pre-empt the currently-running process as well as move this newly-ready process into the running state.

A scheduler is a pre-emptive scheduler if it acquires the aptitude to acquire invoked by an interrupt as well as delivers a process out of a moving state to allow another process flow. The last two events in the furthestmost list may drive this to occur. If a scheduler cannot abduct the CPU elsewhere from a process that time it is an adaptable or non-pre-emptive scheduler. Primitive operating systems like as Microsoft Windows 3.1 or Apple Mac OS following to OS X are examples of cooperative schedulers. Older batch processing systems adhered run-to-completion schedulers where a mechanism raced to abandonment before numerous foreign processes would be assigned to run.

The judgments that the scheduler brings about concerning the sequence as well as length of time that mechanisms may run is designated the scheduling algorithm (or scheduling policy). These judgments are not contend ones, as the scheduler acquires only a restricted number of information about the processes that are develop to run. An excellent scheduling algorithm should:

Be attractive – allocate each process a pretty share of the CPU, permit each process to proceed in a feasible measure of time.

Be accommodating – preserve the CPU busy whole the time.

Enlarge throughput – service the largest feasible notation of jobs in an allotted measure of time; decrease the measure of time user’s essential wait for their outcomes.

Abbreviate response time – collaborative users should inspect good performance.

Be predictable – an allotted job should appropriate about the equal number of time to run when run multiple times. This preserves users realistic.

Minimize overhead – don’t excrete too many means. Keep approximating time as well as context switch time at a minimal.

Maximize resource utilize – contribute processes that will utilize underutilized means. There are two causes for this. Maximum devices are sluggish matched to CPU actions.

2.2 Process States

The process state complying of all imperative to begin again the process accomplishment if it is somehow laid monologue impermanent. The process state consists of at least resulting:

- Code for the program.
- Programs fixed data.
- Program’s active data.
- Program’s approach call stack.
- Contents of general purpose register.
- Contents of program counter (PC)
- Contents of program status word (PSW).

- Operating Systems resource in application.

A process flows through an arrangement of different process states.

- New State: The process being created.
- Running State: A process is discussed to be running if it holds the CPU, that is, process accurately employing the CPU at that definite condition.
- Blocked (or halting) State: A process is lectured to be blocked if it is halting for several conditions to occur like that as an I/O achievement preceding it can precede. Record that a process is unable to flow until several external condition occurs.
- Ready State: A process is discussed to be ready if it utilizes a CPU if one were suitable. A ready state process is run able however transiently stopped flowing to allow another approach run.
- Terminated state: The process seizes finished execution.

Check your progress 1

1. In a New State, the starts _____.

a. developing	c. writing
b. reading	d. all
2. The CPU is used in _____state.

a. new	c. halting
b. running	d. none of these

2.3 Virtual Processor

A virtual CPU (vCPU) additionally recognized as a virtual processor, is an actual central processing unit (CPU) that is allocated to a virtual machine (VM).By shortfall, virtual machines are assigned one vCPU each. If the actual host acquires multiple CPU cores at its desertion, nevertheless, that time a CPU scheduler allocates completion contexts as well as the vCPU centrally serves a series of duration slots on logical processors.

Since processing time is billable, it is notable for an administrator to comprehend how his cloud donator documents vCPU application in an invoice. It is additionally important for the administrator to determine that accumulating

more vCPUs will not automatically advance action. This is due to as the notation of vCPUs flows up, it serves increased complicated for the scheduler to arrange time slots on the real CPUs, along with the wait time can disgrace performance.

In VMware, vCPUs are component of the symmetric multi-processing (SMP) multi-threaded approximate model. SMP additionally allocates threads to be break across multiple actual or feasible cores to alter performance of additional parallel virtualized works. vCPUs allow multitasking to be acted consecutively in a multi-core ambience.

- (1) In virtualized server surroundings, a virtual processor is a CPU core that is apportioned to a virtual machine. There can be additional virtual processors allocated than real cores feasible, which allocates virtual machines to participate the equivalent core.
- (2) In parallel processing surroundings that adheres more data components than processors, a virtual processor is a duplicated processor. Virtual processors conduct in series, not in parallel, although authenticate applications that need a processor for each data component to flow in a computer with fewer processors.

Check your progress 2

1. A virtual processor is a _____core.

a. hard disk

c. memory

b. CPU

d. none

2.4 Interrupt Mechanism

An interrupt is a signal from equipment affixed to a computer or from an approach within the computer that brings about the core program that conducts the computer (the operating system) to stop as well as figure out what to conduct next. Almost entire personal (or larger) computers here and now are interrupt-driven - that is, they begin down the index of computer instructions in one program (maybe an application like as a word processor) further preserve running the instructions until both

1. Actuate any further
2. Interrupt signal is detected

Following the interrupt signal is perceived, the computer either begins again running the program it endured running or commences running another program.

Acutely, an individual computer can function only one computer instruction at an interval. Although, since it can be delayed, it can acquire turns in which programs or sets of instructions that it functions. This is comprehended as multitasking. It assigns the user to execute a enumerate of contradictory things at the same time. The computer clearly acquires turns managing the programs that the user consequentially starts. Of course, the computer conducts at speeds that generate it seem as although all of the user's works are being acted at the same time. (The computer's operating system is beneficial at using compact pauses in operations besides user think time to work on external programs.)

An operating system usually acquires several code that is identified an interrupt handler. The interrupt handler prioritizes the interrupts as well as preserves them in a chain if more than one is halting to be handled. The operating system acquires another brief program, sometimes termed a scheduler, which circumscribes away which program to assign control to next.

In common, there are hardware interrupts as well as software interrupts. A hardware interrupt arises, for exemplary, when an I/O operation is accomplished like as reading some data into the computer from a tape drive. A software interrupt arises when an application program ceases or appeals assured services from the operating system. In a personal computer, a hardware interrupt request (IRQ) acquires a value affiliated with it that associates it with a definite device.

Five conditions must be true for an interrupt to be generated:

- 1) Device arm,
- 2) NVIC enables,
- 3) Global enable,
- 4) Interrupt priority level must be higher than current level executing, and
- 5) Hardware event trigger.

Check your progress 3

1. Interrupt mechanism uses_____.
 - a. one programs
 - b. two programs
 - c. many programs
 - d. all
2. Which is not a valid condition for an interrupt to be generated?
 - a. device arm
 - b. NVIC enable
 - c. global enable
 - d. interrupt priority should be low

2.5 Scheduling Algorithms and Its Performance

Previously when a set of preference relations for a project is known, then the necessary scheduling trouble turn out to be formation of a Priority List. There are a group of potential strategies to facilitate which lead to formation of a Priority List. At this time, we will think only two of these strategies:

- Decreasing-Time Algorithm
- Critical-Path Algorithm

Decreasing-Time Algorithm

Decreasing-Time Algorithm (DTA) is based on simple strategy:

Perform the longer jobs initially as well as save the shorter jobs for final. Basically it places the DTA to make a Priority List by listing the everyday jobs in declining order of dispensation times. Tasks through equal processing times are capable of listing in any order. A Priority List produced by the DTA is over and over again a decreasing-time list as shown in fig 2.1.

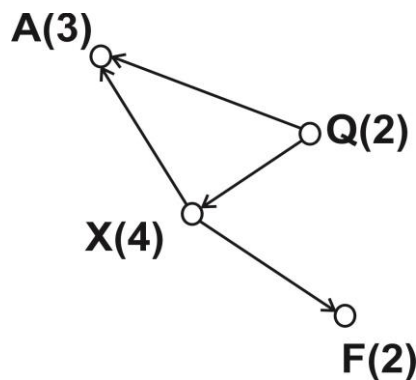


Fig 2.1 decreasing-time Algorithm

One time, it is seen that the precedence relations always overrule the Priority List as soon as there is a conflict involving the two. As a result, for example, at this time the task X cannot in fact be assigned first despite of the fact that it is first on the Priority List from the time when precedence relations insist to facilitate task Q lead task X.

Even if the approach of scheduling says that the longer tasks first are good, it does have a major defect. The DTA pay no attention to any information in the project diagram that shows that one or more tasks ought to be done near the beginning rather than late. For illustration, if one or more tasks by way of long processing times can't commence in anticipation of task X to get finished, at that time passing on task X early will almost certainly result in a shorter finishing time still however assigning task X early go against the DTA.

Critical-Path Algorithm

Formerly, the theory of critical time is known, now we will study about Critical-Path Algorithm. The Critical-Path Algorithm (CPA) is based on an approach comparable to with the aim of Decreasing-Time Algorithm:

It performs the work with high critical times first as well as keeps the jobs with shorter critical times for final. It is seen that, the CPA produce a Priority List by listing the work in declining order of significant times. It is found that work with equal critical times can be listed in any manner. A Priority List created by the CPA is often called a critical-path list.

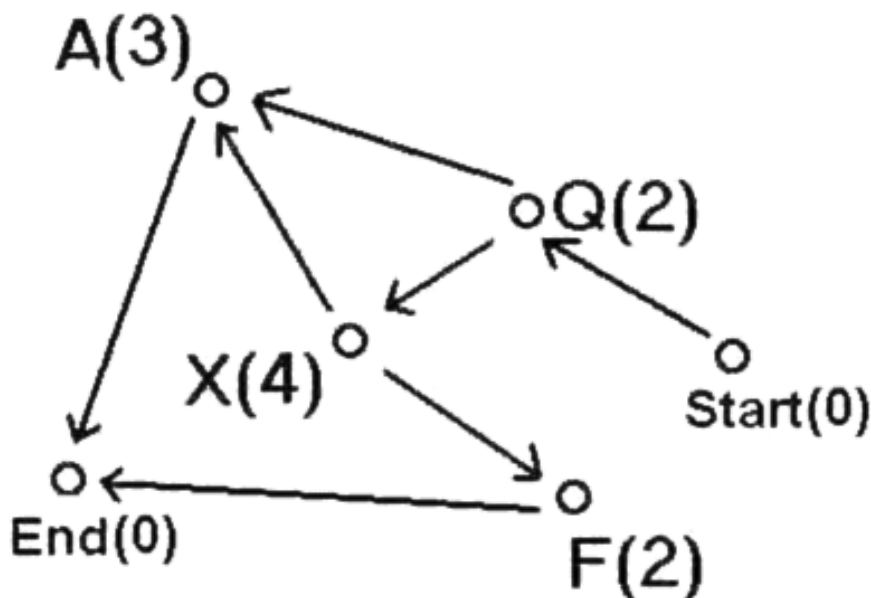


Fig 2.2 Critical-Path Algorithm

The initial step in applying the CPA to a project diagram is to understand the Backflow Algorithm to return all processing times with critical times. Although the Critical-Path Algorithm is usually enhanced as compared to Decreasing-Time Algorithm, neither is guaranteed to produce an optimal schedule. In fact, no efficient scheduling algorithm is presently known that always gives an optimal schedule. However, the Critical-Path Algorithm is the best general-purpose scheduling algorithm currently known.

Check your progress 4

1. Scheduling is :

- | | |
|--|------------------|
| a. allowing a job to use the processor | c. Both a and b |
| b. making proper use of processor | d. None of these |

2.6 Threads

A thread is a particular sequence stream surrounded by a process. For the reason that threads have a number of properties of processes, they are occasionally called light weight processes. In a process, threads permit multiple implementations of streams. In numerous reverences, threads are accepted way to get better application through parallelism. The CPU switches quickly back as well as forth in the middle of the threads giving false impression that the threads are running in parallel. Like a conventional process i.e., process with one thread, a thread can be in any of several states. Each thread has its individual stack. In view of the fact that thread will usually call different procedures moreover thus a different execution history. This is why thread needs its individual stack. An operating system that has thread facility, the fundamental unit of CPU operation is a thread. A thread has or consists of a program counter (PC), a register set as well as stack space. Threads are not self-governing of one other like processes as a result threads distribute with other threads their code section, data section, OS resources also known as task, such as open files and signals.

Threads are used in designing operating systems because:

- A process with multiple threads makes a great server for example printer server.

- Because threads can share common data, they do not need to use interprocess communication.
- Because of the very nature, threads can take advantage of multiprocessors.

Threads are cheap in the intelligence because:

- They only need a stack along with storage for registers as a result, threads are cheap to create.
- Threads use very small resources of an operating system in which they are working. That is, threads do not require new address space, global data, program code or operating system resources.
- Context switching is fast as soon as working with threads. The reason is that we only have to save and/or restore PC, SP and registers.

As shown in Figure 2.3, multi-threaded applications contain multiple threads contained by single process, all having their individual program counter, stack and set of registers, other than sharing common code, data as well as certain structures such as open files

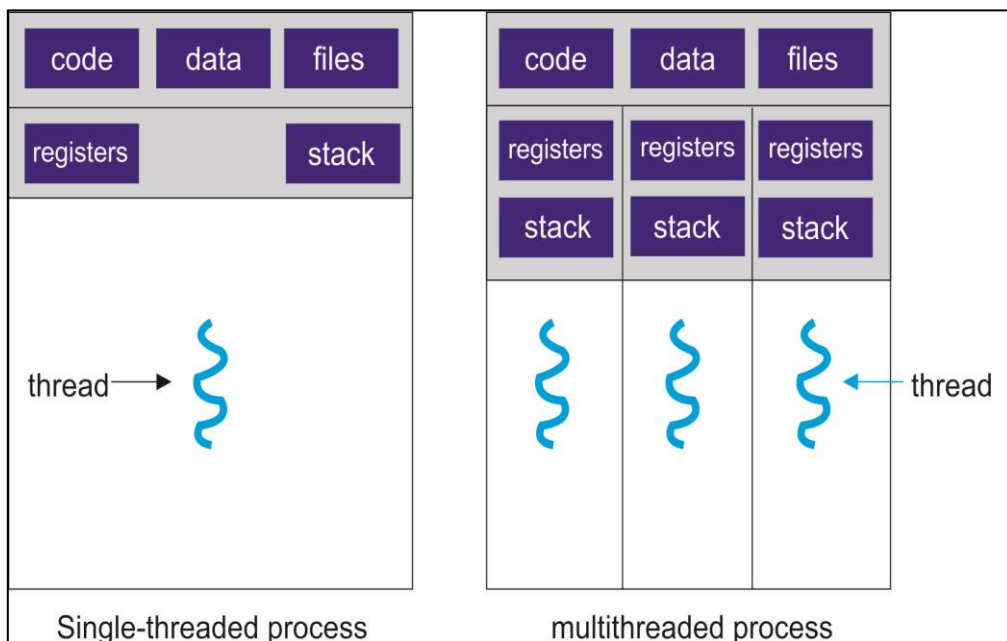


Fig 2.3 Single and multi-threaded

Architecture

Threads are extremely useful in modern programming at any time the process has multiple tasks to carry out independently of the others.

This is on the whole true when one of the tasks possibly will block, furthermore it is required to allow the other tasks to proceed with no blocking.

For instance in a word processor, a surroundings thread may ensure spelling as well as grammar while a centre thread processes user input, while however a third thread loads images from the hard drive, as well as a fourth does periodic automatic backups of the file being condensed.

An additional instance is a web server - Multiple threads permit for multiple requests to be fulfilled simultaneously, with no service requests sequentially or to fork off separate processes for each incoming request.

Benefits

There are four major categories of benefits to multi-threading:

Responsiveness - One thread may give rapid reply at the same time other threads are blocked-up or slow down doing serious calculations.

Resource sharing - By logic, threads contribute to common code, data, as well as other resources, which allows numerous tasks to be performed at the same time in a single address space.

Economy - Creating as well as managing threads is a lot faster than performing the same tasks for processes.

Scalability, i.e. Utilization of multiprocessor architectures - A single threaded process be able to only run on one CPU, no issue how many may be accessible, while the execution of a multi-threaded application might be split in the middle of available processors.

Check your progress 5

1. A process can be _____.
 - a. single threaded
 - b. multithreaded
 - c. both (a) and (b)
 - d. none of the mentioned
2. Which of the following is not a valid state of a thread?
 - a. running
 - b. parsing
 - c. ready
 - d. blocked

2.7 Let Us Sum Up

In this unit we have learned:

- That a virtual CPU also called as CPU is called as a virtual processor.
- In this, an interrupt is a signal from equipment affixed to a computer.
- We see that thread is the smallest unit of processing that can be performed in an operating system.

2.8 Answers for Check Your Progress

Check your progress 1

Answers: (1-a), (2-b)

Check your progress 2

Answers: (1-b)

Check your progress 3

Answers: (1-c), (2-d)

Check your progress 4

Answers: (1-c)

Check your progress 5

Answers: (1-c), (2-b)

2.9 Glossary

1. **Virtual reality** - Virtual reality is an artificial environment that is created with software and presented to the user in such a way that the user suspends belief and accepts it as a real environment.
2. **VMware Storage Policy-Based Management** - Storage Policy-Based Management is a feature that allows for automatic provisioning of virtual machines in a VMware environment.

3. **VMware Platform Services Controller (PSC)** - VMware Platform Services Controller (PSC) is a new service in vSphere 6 that handles the infrastructure security functions.
4. **Virtualization** - Terms related to virtualization, including definitions about virtualization technologies and words and phrases about server virtualization, desktop virtualization and storage virtualization.

2.10 Assignment

Write detail on Page replacement algorithms.

2.11 Activities

Explain Paging address Translation by direct mapping.

2.12 Case Study

Write the different types of thread mechanism.

2.13 Further Reading

1. The Operating system by Andrew Tannenbaum.
2. Operating System by Mach.

Block Summary

In this block, the students have learnt about the basic of memory management and process scheduling that occurs in Operating System. The block focuses more on the concept of virtual memory, paging and segmentation. The understanding about Cache memory and virtual processor along with its necessary techniques has also been explained.

After completing this block, students will be able to learn and work on variety of operating system available today. The use of operating system with various processing techniques will allow them to gain practical knowledge on processor and its interference with operating system. The authors have made every possible effort in learning and designing about basic of memory management techniques and its related concepts with more knowledge on memory hierarchy. The students will explained diagrammatically about different process involves along with interrupt mechanism. The students will be demonstrated with working of page replace algorithm and its technique.

Block Assignment

Short Answer Questions

1. What is paging?
2. What do you mean by an interrupt?
3. What is segmentation?
4. Explain the types of processes?
5. What is Virtual memory?

Long Answer Questions

1. Write short note on memory hierarchy?
2. What is the importance of Cache memory?
3. Write detail on Virtual processor?

Enrolment No.

1. How many hours did you need for studying the units?

Unit No	1	2	3	4
Nos of Hrs				

2. Please give your reactions to the following items based on your reading of the block:

Items	Excellent	Very Good	Good	Poor	Give specific example if any
Presentation Quality	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____
Language and Style	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____
Illustration used (Diagram, tables etc)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____
Conceptual Clarity	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____
Check your progress Quest	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____
Feed back to CYP Question	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____

3. Any Other Comments

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....



“

*Education is something
which ought to be
brought within
the reach of every one.*

”

- Dr. B. R. Ambedkar



Dr. Babasaheb Ambedkar Open University
Jyotirmay' Parisar, Opp. Shri Balaji Temple, Sarkhej-Gandhinagar Highway, Chharodi,
Ahmedabad-382 481.

FUNDAMENTALS OF OPERATING SYSTEM

PGDCA 104

BLOCK 3: FILE AND I/O MANAGEMENT

**Dr. Babasaheb Ambedkar Open University
Ahmedabad**



FUNDAMENTALS OF OPERATING SYSTEM



Knowledge Management and
Research Organization
Pune



Editorial Panel

Author

Er. Nishit Mathur

Language Editor

Prof. Jaipal Gaikwad

Graphic and Creative Panel

Ms. K. Jamdal

Ms. Lata Dawange

Ms. Pinaz Driver

Ms. Ashwini Wankhede

Mr. Kiran Shinde

Mr. Prashant Tikone

Mr. Akshay Mirajkar

Copyright © 2015 Knowledge Management and Research Organization.

All rights reserved. No part of this book may be reproduced, transmitted or utilized in any form or by a means, electronic or mechanical, including photocopying, recording or by any information storage or retrieval system without written permission from us.

Acknowledgment

Every attempt has been made to trace the copyright holders of material reproduced in this book. Should an infringement have occurred, we apologize for the same and will be pleased to make necessary correction/amendment in future edition of this book.

The content is developed by taking reference of online and print publications that are mentioned in Bibliography. The content developed represents the breadth of research excellence in this multidisciplinary academic field. Some of the information, illustrations and examples are taken "as is" and as available in the references mentioned in Bibliography for academic purpose and better understanding by learner.'



ROLE OF SELF INSTRUCTIONAL MATERIAL IN DISTANCE LEARNING

The need to plan effective instruction is imperative for a successful distance teaching repertoire. This is due to the fact that the instructional designer, the tutor, the author (s) and the student are often separated by distance and may never meet in person. This is an increasingly common scenario in distance education instruction. As much as possible, teaching by distance should stimulate the student's intellectual involvement and contain all the necessary learning instructional activities that are capable of guiding the student through the course objectives. Therefore, the course / self-instructional material are completely equipped with everything that the syllabus prescribes.

To ensure effective instruction, a number of instructional design ideas are used and these help students to acquire knowledge, intellectual skills, motor skills and necessary attitudinal changes. In this respect, students' assessment and course evaluation are incorporated in the text.

The nature of instructional activities used in distance education self-instructional materials depends on the domain of learning that they reinforce in the text, that is, the cognitive, psychomotor and affective. These are further interpreted in the acquisition of knowledge, intellectual skills and motor skills. Students may be encouraged to gain, apply and communicate (orally or in writing) the knowledge acquired. Intellectual-skills objectives may be met by designing instructions that make use of students' prior knowledge and experiences in the discourse as the foundation on which newly acquired knowledge is built.

The provision of exercises in the form of assignments, projects and tutorial feedback is necessary. Instructional activities that teach motor skills need to be graphically demonstrated and the correct practices provided during tutorials. Instructional activities for inculcating change in attitude and behavior should create interest and demonstrate need and benefits gained by adopting the required change. Information on the adoption and procedures for practice of new attitudes may then be introduced.

Teaching and learning at a distance eliminates interactive communication cues, such as pauses, intonation and gestures, associated with the face-to-face method of teaching. This is particularly so with the exclusive use of print media. Instructional activities built into the instructional repertoire provide this missing interaction between the student and the teacher. Therefore, the use of instructional activities to affect better distance teaching is not optional, but mandatory.

Our team of successful writers and authors has tried to reduce this.

Divide and to bring this Self Instructional Material as the best teaching and communication tool. Instructional activities are varied in order to assess the different facets of the domains of learning.

Distance education teaching repertoire involves extensive use of self-instructional materials, be they print or otherwise. These materials are designed to achieve certain pre-determined learning outcomes, namely goals and objectives that are contained in an instructional plan. Since the teaching process is affected over a distance, there is need to ensure that students actively participate in their learning by performing specific tasks that help them to understand the relevant concepts. Therefore, a set of exercises is built into the teaching repertoire in order to link what students and tutors do in the framework of the course outline. These could be in the form of students' assignments, a research project or a science practical exercise. Examples of instructional activities in distance education are too numerous to list. Instructional activities, when used in this context, help to motivate students, guide and measure students' performance (continuous assessment)



PREFACE

We have put in lots of hard work to make this book as user-friendly as possible, but we have not sacrificed quality. Experts were involved in preparing the materials. However, concepts are explained in easy language for you. We have included many tables and examples for easy understanding.

We sincerely hope this book will help you in every way you expect.

All the best for your studies from our team!



FUNDAMENTALS OF OPERATING SYSTEM

Contents

BLOCK 1: INTRODUCTION TO OPERATING SYSTEMS

UNIT 1 BASICS OF OS

Definition and Function of operating systems, Evolution of operating system, Operating system structure-monolithic layered, virtual machine and Client server

UNIT 2 TYPES OF OPERATING SYSTEM

Different types of operating system-real time systems, multi-user System, distributed system

UNIT 3 BATCH OPERATING SYSTEM

Introduction to basic terms and batch processing system: Jobs, Processes files, command interpreter

BLOCK 2: MEMORY MANAGEMENT AND PROCESS SCHEDULING

UNIT 1 MEMORY MANAGEMENT

Logical and Physical address protection, paging, and segmentation, Virtual memory, Page replacement algorithms, Cache memory, hierarchy of memory types, Associative memory

UNIT 2 PROCESS SCHEDULING

Process states, virtual processor, Interrupt mechanism, Scheduling algorithms Performance evaluation of scheduling algorithm, Threads

BLOCK 3: FILE AND I/O MANAGEMENT

UNIT 1 FILE SYSTEM

File systems-Partitions and Directory structure, Disk space allocation, Disk scheduling

UNIT 2 I/O MANAGEMENT

I/O Hardware, I/O Drivers, DMA controlled I/O and programmed I/O, I/O Supervisors

BLOCK 4: BASICS OF DISTRIBUTED OPERATING SYSTEM

UNIT 1 DISTRIBUTED OPERATING SYSTEM

Introduction and need for distributed OS, Architecture of Distributed OS, Models of distributed system

UNIT 2 MORE ON OPERATING SYSTEM

Remote procedure Calls, Distributed shared memory, Unix Operating System: Case Studies



Dr. Babasaheb
Ambedkar
Open University

PGDCA 104

FUNDAMENTALS OF OPERATING SYSTEM

BLOCK 3: FILE AND I/O MANAGEMENT

UNIT 1

FILE SYSTEM

UNIT 2

I/O MANAGEMENT

BLOCK 3: FILE AND I/O MANAGEMENT

Block Introduction

An operating system is important software which makes the computer to run. It handles all the computer processes and runs the hardware. It makes you to communicate with computer without having command on its language. It is seen that your computer operating system manages all software and hardware functions. The main idea of operating system is to coordinate will all processes and links these processes with central processing unit (CPU), memory and storage.

In this block, we will discuss about the basic of file system management and input output memory management. The block will focus on the study and concept of disk space allocation, disk scheduling and input out device drivers. The students will give an idea on DMA control input output and basic programmed input output.

In this block, the student will made to learn and understand about the basic of programmed DMA input output management techniques. The concept related to input output supervisors and input output drivers will also be explained to the students. The student will be demonstrated practically about programmed input output technique.

Block Objective

After learning this block, you will be able to understand:

- About File systems-structure and partition
- Basic of Disk space allocation
- Features of Disk scheduling
- Concept of I/O Hardware and Drivers
- Detailed about DMA controlled I/O
- Basic of Programmed I/O
- Idea of I/O Supervisors

File and I/O
Management

Block Structure

Unit 1: File System

Unit 2: I/O Management

UNIT 1: FILE SYSTEM

Unit Structure

- 1.0 Learning Objectives**
- 1.1 Introduction**
- 1.2 File Systems**
 - 1.2.1 Partitions
 - 1.2.2 Directory structure
- 1.3 Disk Space Allocation**
- 1.4 Disk Scheduling**
- 1.5 Let Us Sum Up**
- 1.6 Answers For Check Your Progress**
- 1.7 Glossary**
- 1.8 Assignment**
- 1.9 Activities**
- 1.10 Case Study**
- 1.11 Further Readings**

1.0 Learning Objectives

After learning this unit, you will be able to understand:

- Basic of File systems
- Structure and file partition concepts
- Types of files
- Non-contiguous and contiguous storage allocation
- Idea about Disk scheduling

1.1 Introduction

A file system is the methods and data structures that an operating system uses to keep track of files on a disk or partition; that is, the way the files are organized on the disk. The word is also used to refer to a partition or disk that is used to store the files or the type of the file system. Thus, one might say “I have two file systems” meaning one has two partitions on which one stores files, or that one is using the “extended file system”, meaning the type of the file system.

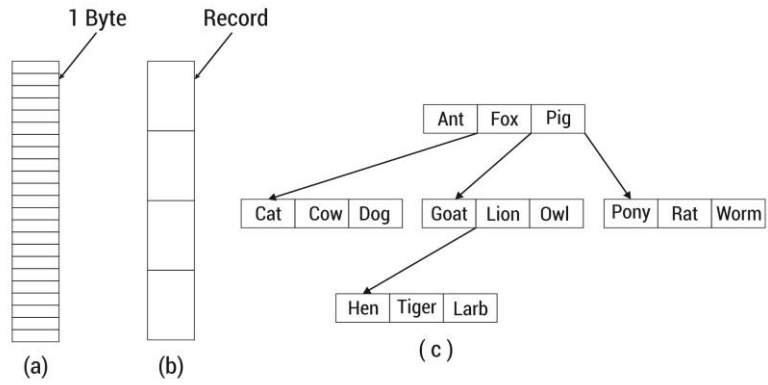
The difference between a disk or partition and the file system it contains is important. A few programs (including, reasonably enough, programs that create file systems) operate directly on the raw sectors of a disk or partition; if there is an existing file system there it will be destroyed or seriously corrupted. Most programs operate on a file system, and therefore won't work on a partition that doesn't contain one (or that contains one of the wrong types).

Before a partition or disk can be used as a file system, it needs to be initialized, and the bookkeeping data structures need to be written to the disk. This process is called making a file system.

1.2 File Systems

When preparing a segment such as this, an adequate discussion tends to blur the line between hardware issues associated with hard disks and the software issues that control what is placed on them and in what manner. As manufacturers and operating system developers strive for performance and security, this line tends to blur even more. The very nature of the logical structures on a hard disk influences their performance, reliability, expandability and compatibility.

In spite of all of the media hype about them, a hard disk is merely a medium for storing information. A replacement for the limited capacity of the floppy disk, which was the first type of disk storage media available on small computers. As hard disks grow in capacity, becoming larger and larger every year, it is becoming increasingly difficult for operating systems and their companion file systems, to use them in an efficient manner.



Three kinds of files
 • byte sequence • record sequence • tree

Fig 1.1 file system

The file system employed by most operating systems today is a generic name given to the software routines and logical structures used to prepare the given hard disk to store data as well as control access to that particular storage space. Different operating systems use different methods of organizing and controlling access to the data on the hard disk, which is entirely independent of the specific hardware in use. A single hard disk can be prepared in many different ways to store data, and under given circumstances a hard disk may even be prepared multiple ways on the same disk.

File systems will find about the naming particular files that are having maximum name characters to be utilised in certain systems and till what time the file name suffix can be applied. It shows a way to specify path to a file by the use of directory structure. It uses metadata to keep and retrieve files that will cover:

- Date created
- Date modified
- File size

Such type of file system example can be OS X that are utilised by Macintosh hardware by allowing various optimization features that will cover file names with 255 characters.

For certain group of user, such type of file system is constraints as they will not provide read / write access. The best way is to wither put a password or to encrypt the files so that the user can't access. While encrypting, a key is provided

to encrypt the file which can further decrypt the encrypted file text. The user with definite key can only access the required file.

1.2.1 Partitions

When referring to a computer hard drive, a disk partition or partition is a segment of the hard drive that is separated from other portions of the hard drive. Partitions help enable users to divide a computer hard drive into different drives or into different portions for multiple operating systems to run on the same drive.

With older file allocation tables, such as FAT16, creating smaller partitions allows a computer hard drive to run more efficiently and save more disk space. However, with new file allocation tables, such as FAT32, this is no longer the case.

There are different types of partitions that exist in Operating System:

- AIX: Partition used with the AIX operating system.
- Boot: It is a partition that contains the files required for a system start up.
- BSD/OS: This partition is used with the BSD operating system.
- DOS: It is used with older versions of MS-DOS.
- DOS Ext: It is extended from one or more original MS-DOS partitions.
- DRDOS: It is used with DR. DOS operating system.
- Extended: It is extended from one or more of the primary partitions.
- Hibernation: It is used with older hibernation programs.
- HPFS: This is used with IBM OS/2 and Microsoft NT 3.x
- Linux: This partition is used with several variants of Linux O/S.
- MINIX: This is used with MINIX operating system.
- NON-DOS: It is used in Microsoft fdisk partition which is not native to Microsoft operating system.
- NEC DOS: It is used with earlier NEC DOS variant.
- NEXTSTEP: It is used with Next step operating system.
- Novell Netware: It is used with Novell Netware operating system.

- NTFS: It is used with Microsoft Windows NT 4.x, Windows 2000 and Windows XP.
- Partition Magic: It is created using Partition Magic utility by PowerQuest.
- PC-ARMOUR: It is created by PC ARMOUR security utility.
- Solaris X86: This is used with Sun Solaris X86 platform operating system.
- System: This partition contains system32 directory.
- Tandy DOS: It is used with earlier Tandy DOS variant.
- Unix System V: This is used with various Unix Operating systems.
- VMWare: This is used by VMWare.
- XENIX: It is used with Xenix operating system.

1.2.2 Directory structure

There are many types of directory structure in Operating System. They are as follows:-

- 1) Single Level Directory
- 2) Two Level Directory
- 3) Tree Structured Directory
- 4) Acyclic Graph Directory
- 5) General Graph Directory

Single Level Directory

In this type of directory structure as shown in fig 1.2, all files are in the same directory.

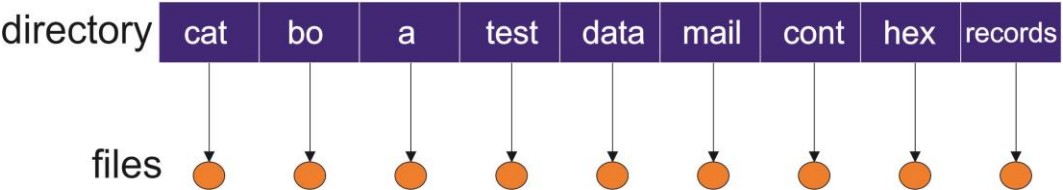


Fig 1.2 Single level Directory

It has certain limitations as:

- Being all files in same directory, they possess unique name.
- If two users call their data free test, then unique name rule is violated.
- Files are limited in length.
- Even a single user may find it difficult to remember the names of all files as the number of file increases.
- Keeping track of so many file is daunting task.

Two Level Directories

In this type of directory system as shown in fig 1.3:

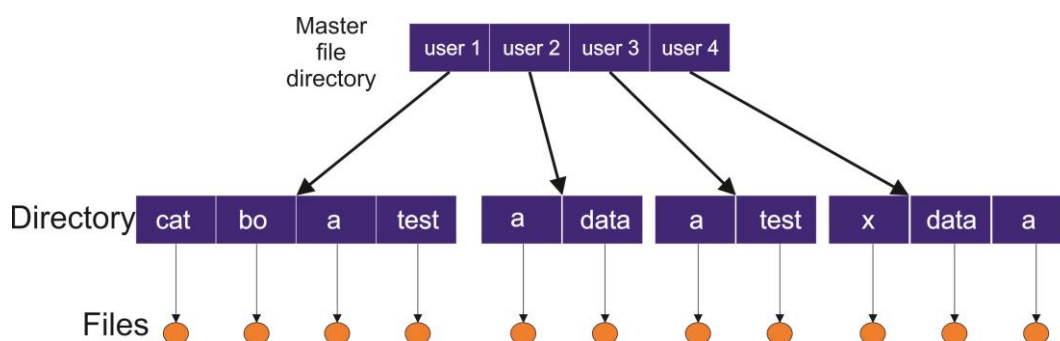


Fig 1.3 Two level Directory Structure

- i. Every user has its own User File Directory (UFD).
- ii. On account of user job start or log in, system Master File Directory (MFD) gets searched. The Master File Directory gets indexed by user name or Account Number.
- iii. In case when the user refers to certain file, then at that time its own UFD gets searched.

It is seen that different users contains files in similar name. To obtain a different partition, in a two level directory, we assign user name and file name. In case of two level directories, we see that it could be a tree or an inverted tree of height 2. Also the root of a tree in this is a Master File Directory (MFD) which gets direct descendants as User File Directory (UFD). Descendants of UFD's are its file. Such files are the leaves of a tree. This type of file structure has certain limitations as the structure become isolated from one user to another.

Tree Structured Directory

It is another type of directory or Sub directory structure as shown in fig 1.4 carries set of files or sub directories.

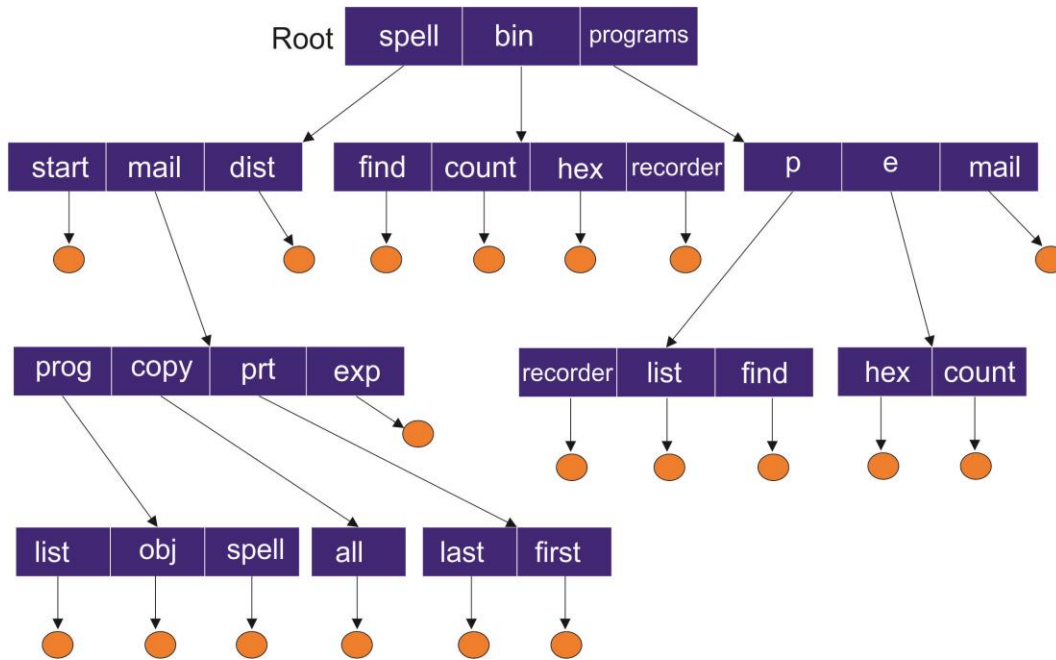


Fig 1.4 Tree Level Directory Structure

In this, complete directories contains similar internal format. It has specific features:

1. In this, single bit in each directory entry shows the entry.
2. Here the special calls are used to form and delete directories.
3. In this all process has current directory that carries many files which are of current interest to the process.
4. In this, when a reference is made to a file, the current directory gets searched.
5. In this, the user will amend his current directory whenever he wants.
6. In case, if the file is not required in present directory then the user normally either shows a path name or practically amends the present directory. Here the paths can be of two types:-
 - a) Absolute Path: This will start at root and follows a path down to particular file.
 - b) Relative Path: this will explain a path from present directory.
7. In this type of directory structure, if deleted directory is empty, then its entry in directory containing will get deleted. On the other hand, if the directory is not empty, then either the two approaches exist:-
 - a) User must delete all the files in the directory.

b) If any sub directories exist, same procedure must be applied.

Here the UNIX rm command is used, whereas MS dos will not delete a directory unless it is empty.

Acyclic Graph Directory

Fig 1.5 shows another type of directory which is a graph having no cycles.

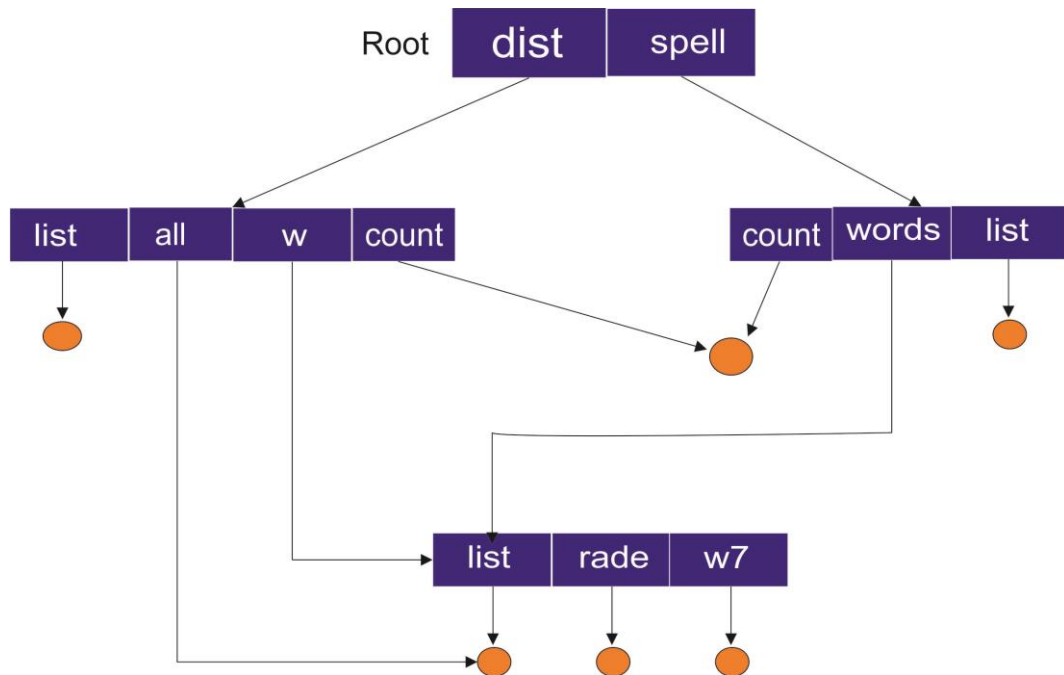


Fig 1.5 Acyclic Graph Directories

This type of directory structure allows directories to contribute sub directories and files. In sharing mode, only actual file exists, so for any changes by one person gives visibility to another. Here the implementation of shared files and directories takes place:

A. Creating a link

- A link is effectively a pointer to another file or sub directory.
- Duplicate all the information about them in both sharing directories.

B. Deleting a link

- Deletion of the link will not affect the original file, only link is removed.
- To preserve the file until all references to it are deleted.

Check your progress 1

1. Which is not a file type?
 - a. tree
 - b. leaf sequence
 - c. byte sequence
 - d. record sequence
2. Which is an example of metadata tags?
 - a. Date created
 - b. Date modified
 - c. File size
 - d. all
3. Which partition is used with older versions of MS-DOS.
 - a. DOS
 - b. DOS Ext
 - c. DRDOS
 - d. none
4. In which type of directory arrangement, all files are placed in single directory.
 - a. Single Level Directory
 - b. Two Level Directory
 - c. Tree Structured Directory
 - d. Acyclic Graph Directory

1.3 Disk Space Allocation

It is noted that an important function of a file system is to handle the space present on the secondary storage with the use of safe tracking of disk blocks which are assigned to files along with free blocks which is made available for allocation. The process of allocation of space to files carries the following problems:

1. Efficient disk space usage
2. Quick file access

Disk block management appears to be a problem where the secondary storage arises with two additional problems as:

1. Low disk access time
2. Dealing of more blocks

With such problems, there appears to be plenty of conclusions which is present in both environments such as contiguous file allocation as well as non-contiguous file allocation. With this, there exists three another allocation

techniques like contiguous, linked along with indexed. It is finding that every method has its own merits and demerits.

Disk Allocation Methods

By using direct applications of disks and storing of files in neighbouring part the disk is recommended. The only problem arises as how to give space to files in order to have efficient disk space application with faster response. Once the files are allotted and not tied up, the space available on the disk gets splitted up. There are two important methods, where the disk space gets allotted which are:

1. Continuous
2. Non-continuous

Contiguous Allocation

The contiguous allocation involves assigning files to contiguous secondary storage space. For this, the programmer or user needs to present in advance the size of area that can keep a file that is to be framed. If the particular portion of the space is not free, then the file will not be formed.

The benefit in case of contiguous allocation is that nearly every records of a file will appear closely to each other directly which will increases the entry speed of particular records. This will explain that, if the records are placed here and there across the disk, then its accessing speed will be slow. For a contiguously allocated file, the addressing is easy. In case of sequential access, the file system will keep disk address of last block and if required will reads the next block. In the diagram shown in fig 1.6, in order to access directly to block B of a file having location L, it is defined that it will quickly shows the block L+B. So with this, we can say that contiguous allocation is applicable to both sequential as well as direct accessing.

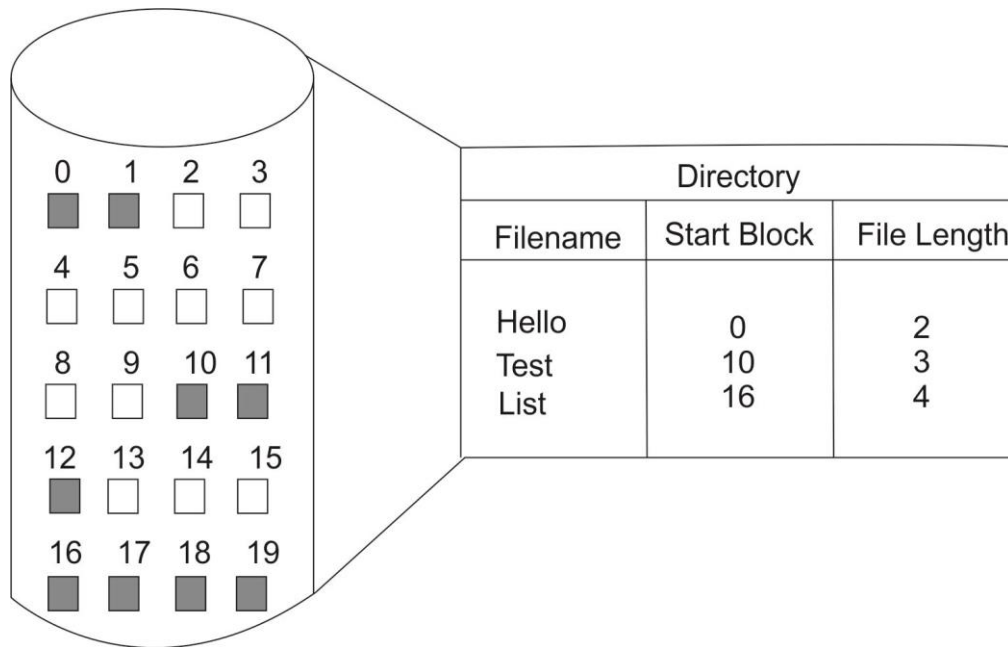


Fig 1.6 contiguous allocation systems

Also it is noted that the file directories available inside contiguous allocation systems is clear to implement. It is found that every file is required to keep the initial location of particular file along with required file size. Consider again the diagram shown in fig 1.6, where the file size is of N blocks long which originates from location L , then it will gather blocks $L, L+1, \dots, L+N-1$. In this the directory entry shows the location or address of initial block along with the file size length.

The only problem with in the contiguous allocation is locate for the space inside a new file when process of free space listing is performed with the help of bit map method. In order to create n -bytes which are quiet long file, we should first locate for n 0 bits in row. For better understanding of contiguous storage allocation problem, we need to assume disk space as a mixture of free as well as occupied portion. With this, we feel that each segment is contiguous group of disk blocks.

In order to agree a request of file for n free contiguous blocks, a group of unused blocks can be find out which is find and decide that which hole will be best for allocation. Such type of problem refers to particular application of dynamic storage allocation which will explain how a request is to satisfied with N size request from a particular list that carries free holes. For this you need to follow two common strategies which can be

1. First – fit

In case of first fit strategy as soon as first hole appears, you will see that the searching gets paused moreover the memory is billed for developing a file. Here, searching will start at the start of set of holes or from the place where earlier first fit search gets halted.

2. Best – fit

In case of best fit strategy, the searching of complete list started for smallest hole, which is quiet big that builds for developing a particular file.

From the above two strategies, we see that both these are not fit in case of storage applications. It is seen that out of the two strategies, first-fit is normally faster as compared to best fit.

With this reason, such type of algorithms will lack from external fragmentations. Once the files are billed and erased, the empty space on the disk space gets broken down into smaller pieces. So we can say that external fragmentations is basically scattered groups that has free blocks which are very tiny for allocation that on collection will show big disk size.

As per the full disk storage and related file size, it seems that external fragmentation could result in minor or major problem. If we see the major problem in case of contiguous allocation, we will find that it is difficult to find the space which is required by the file system. Such type of problems will not arise in case of copying files where exact determination of file size is hard and is not correct.

When the expected file size is much larger in a way that its extension is kept in different disk area, then such of mechanism is called as file overflow. It is seen that locating overflowed contiguous area is quiet boring and difficult that will lost a feel in regard of contiguous allocation.

Non Contiguous Allocation

As the user are unaware about the file storage capacity and data present to contiguous , so nowadays, the storage allocation systems are changed with more dynamic non-contiguous storage allocation systems, that can be:

- Linked allocation
- Indexed allocation

Linked Allocation

It is seen that linked allocation is normally a disk based description of linked list, where the disk blocks are placed here and there on the disk. In this, a directory consists of pointer which is placed in first and last block of file. Also each block contains pointers to the next block, which are not made available to the user.

It can be used effectively for sequential access only but there also it may generate long seeks between blocks. Another issue is the extra storage space required for pointers. Yet the reliability problem is also there due to loss/damage of any pointer. The diagram 1.7 shows linked /chained allocation where each block contains the information about the next block.

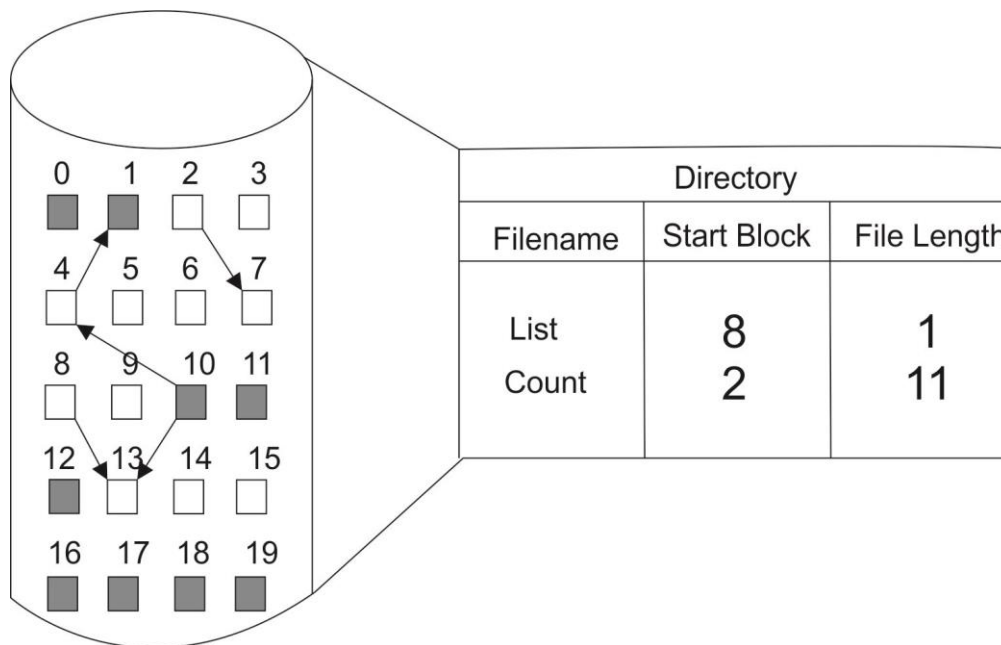


Fig 1.7 linked /chained allocation

MS-DOS and OS/2 use another variation on linked list called FAT (File Allocation Table). The beginning of each partition contains a table having one entry for each disk block and is indexed by the block number.

The directory entry contains the block number of the first block of the file. The table entry indexed by block number contains the block number of the next block in the file.

The Table pointer of the last block in the file has an EOF pointer value. This chain continues until EOF (end of file) table entry is encountered.

With this, we will pass to the next pointers without entering inside the disk for all. The 0 table value shows the presence of unused block; hence allocation of

free blocks with the help of FAT arrangement is clear with simple searching of first block along with 0 table pointer. MS-DOS and OS/2 use this scheme. The figure 1.8 shows file allocation table (FAT).

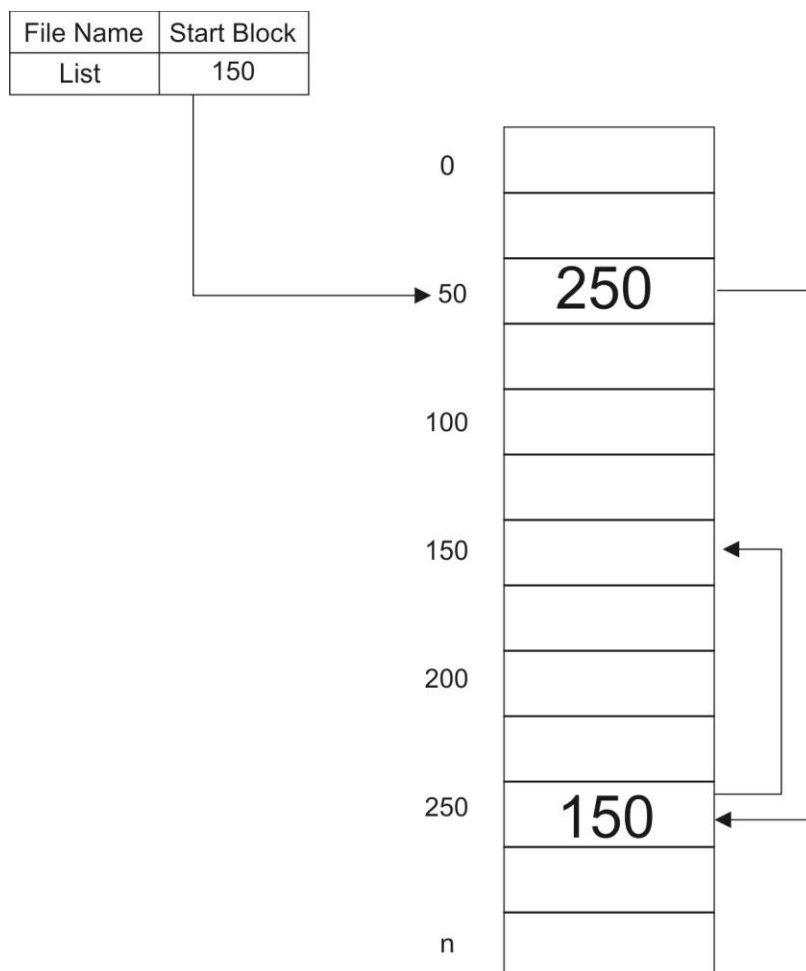


Fig 1.8 Directories

Indexed Allocation:

Index allocation addresses many of the problems of contiguous and chained allocation. In this case, the file allocation table contains a separate one-level index for each file; the index has one entry for each portion allocated to file.

Typically, the file indexes are not physically stored as part of the file allocation table. Rather, the index for a file is kept in a separate block, and entry for the file in the allocation table points to that block. The allocation may be on the basis of either fixed size blocks or variable size portions. The indexed allocation scheme is shown in fig 1.9.

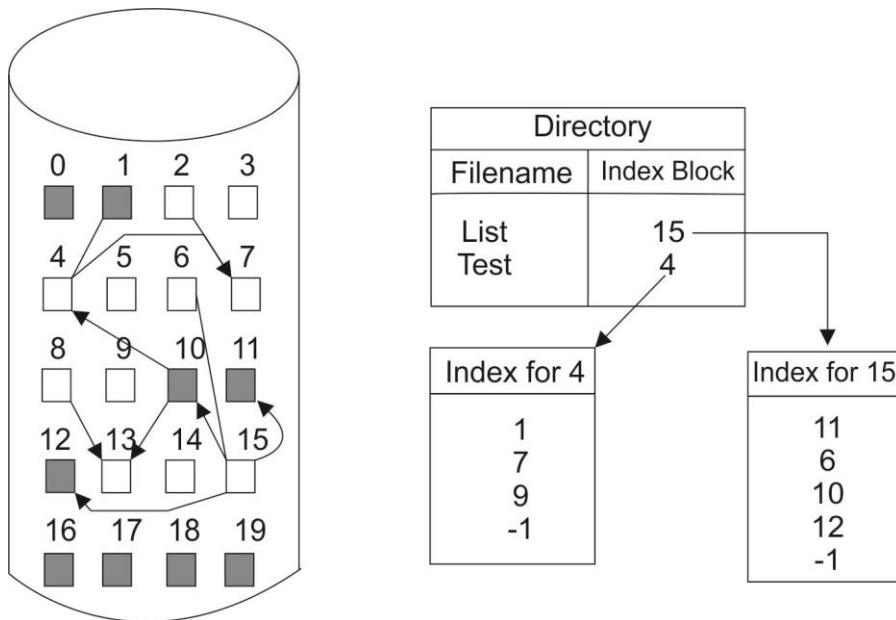


Fig 1.9 indexed allocation

The advantage of this scheme is that it supports both sequential and random access. The searching may take place in index blocks themselves. The indexes blocks may be kept close together in secondary storage to minimize seek time. Also space is wasted only on the index which is not very large and there's no external fragmentation.

Check your progress 2

- Which is a hindrance of secondary storage?
 - slows disk access time
 - larger number of blocks to deal with
 - both a and b
 - neither a nor b
- Which is incorrect in case of Contiguous Allocation?
 - files are assigned to contiguous areas of secondary storage
 - specific size by the user
 - file can be created with any space requirement
 - successive records of file are adjacent to each other

1.4 Disk Scheduling

It is seen that in an individual Computer, there can be many operations at a particular time; hence the management is required on all running processes that are running on particular system at a time. The idea of multi-programming is to perform multiple programs at the same time. To control and distribute memory to related processes, operating system will try to utilise the disk scheduling procedure.

With this process, the CPU time is distributed among various related processes which will describe all procedures in order to perform well. These processes will be specified by disk scheduling that specifies which process to be executed initially by the CPU. Normally, scheduling is concerned with processes that are taken care by the CPU during the particular time.

The idea of CPU scheduling is to examine the complete time required by CPU which carries certain number of processes which can occur simultaneously during a particular period of time. In order to share or divide the complete time of CPU, the CPU will make use of following scheduling processes:

FCFC or First Come First Serve -

In First Come First Serve process, Job or Processes which are undertaken gets arranged as per their order to entry inside the Computer system. The role of Operating System which is located in a queue has series of order which will be acted and describe for the whole process. In this, the jobs are carried out in the same manner as they entered inside the computer system.

SSTF or Shortest Seek Time First -

This type of technique is basically employed by an operating system in order to find for the minimum time. With this technique, searching will takes less time in order to find for a job. After the final examination, all jobs gets arranged in certain sequence as per the priority. The priority in this will signify the total processing time required by particular job to work out. In this the shortest seek time will encounter all time which will takes up the time that to be entered and completes the process.

C-Scan Scheduling -

In such type of scheduling, the processes get arranged by using particular circular order list. Circular List contains a process having no start and end point. In this, the end of List will actually know as the starting point of list. This scheduling involves CPU searching from start to end of a particular process and if

an end was taken, then it will again start its process from starting process. This happens as many times, when a CPU is working on a process, and then the user may wish to enter some data. It shows that the user if required can enter some data as the situation arises where the CPU will again execute the process soon after the Input operation. This type of scheduling is mostly applied in order to process same process in a cycle.

Look Scheduling -

This scheduling involves complete CPU scanning of list from start to end of Disk along with certain other procedures. This scheduling involves continuous CPU scanning of complete disk again and again from one point end to another end point.

Round Robin -

This scheduling is distributed proportionately which is known as Quantum Time. In this, each process that ask for execution gets similar amount of CPU times with standard quantum time. In this the initial process is present where CPU enters straight into next process state. Such type of scheduling is not favourable where after completion of process; the time also gets used by the process. This will shows that due to the presence of dissimilar operation, CPU time gets absorbed by CPU itself which further is wastage.

Priority Scheduling -

In this scheduling, every prioritized process gets checked with the help of total time which is carried out by such processes. This scheduling will examine the complete process time along with total number of Input/output process so that to stop priorities of processes.

Multilevel Queue:

This scheduling is applicable when there are many queues for definite number of processes. This happens as we know that there are many works which are to be performed with computers during the particular time. To arrange different Queues, the CPU here will arrange such Queues by using certain type of approach. The queues will assemble and are arranged in definite functions whose request is there to work.

Check your progress 3

1. Which technique is used by the Operating System to search for the shortest time?
 - a. FCFC
 - b. SSTS
 - c. C-Scan
 - d. Look Scheduling
2. In which scheduling, the processes get arranged by using particular circular order list.
 - a. FCFC
 - b. SSTS
 - c. C-Scan
 - d. Look Scheduling
3. In case of _____scheduling, the time of CPU is shared among equal numbers as Quantum Time.
 - a. Round Robin
 - b. Look
 - c. C-Scan
 - d. none

1.5 Let Us Sum Up

In this unit, we have learned:

- MS-DOS and OS/2 use another variation on linked list called FAT.
- Index allocation addresses many of the problems of contiguous and chained allocation.
- C-Scan Scheduling is a type of scheduling, where the processes get arranged by using particular circular order list.
- Round Robin is a type of scheduling where the time of CPU is shared into equal numbers which is called as Quantum Time.

1.6 Answers for Check Your Progress

Check your progress 1

Answers: (1-b), (2-d), (3-a), (4-a)

Check your progress 2

Answers: (1-d), (2-c)

Check your progress 3

Answers: (1-b), (2-c), (3-a)

1.7 Glossary

1. **File** - A file is a collection of records.
2. **File Organisation** - It is way by which the records get accessed on the disk.
3. **Sequential File** - It is a simplest file organisation where sequential access is present instead of individual file.
4. **Index allocation** - Type of file system that addresses many problems of contiguous and chained allocation.
5. **C-Scan Scheduling** - A scheduling where the processes get arranged by using particular circular order list.
6. **Round Robin** - It is a scheduling where the time of CPU is shared into equal numbers which is called as Quantum Time.

1.8 Assignment

Explain the Operating System File structure?

1.9 Activities

Study file organisation in Operating System.

1.10 Case Study

Study the types of partition in Windows Operating System.

1.11 Further Readings

1. Operating System Concept by Abraham Silberschatz, Peter Baer Galvin, Greg Gagne.
2. Programming Be Operating System by Dan Sydow.

File and I/O
Management

3. Computer Science & Application by Dr. Arvind Mohan Parashar, Chandresh Shah, Saurab Mishra.
4. An Integrated Approach to Software Engineering by Pankaj Jalote.
5. An Operating Systems by Raphael A.

UNIT 2: I/O MANAGEMENT

Unit Structure

- 2.0 Learning Objectives**
- 2.1 Introduction**
- 2.2 I/O Hardware**
- 2.3 I/O Drivers**
- 2.4 DMA Controlled I/O**
- 2.5 Programmed I/O**
- 2.6 I/O Supervisors**
- 2.7 Let Us Sum Up**
- 2.8 Answers For Check Your Progress**
- 2.9 Glossary**
- 2.10 Assignment**
- 2.11 Activities**
- 2.12 Case Study**
- 2.13 Further Readings**

2.0 Learning Objectives

After learning this unit, you will be able to understand:

- Concept of I/O devices
- About Bus Architecture
- Detailed regarding features of DMA controlled I/O
- Basic of Input Output Programmed
- Idea of DMA Channels

2.1 Introduction

Management of I/O devices is one of the important parts of an operating system. It is so important and varied that the entire I/O subsystems will be focussed particularly in its operation. On considering certain devices such as mouse, keyboards, disk drives, display adapters, USB devices, network connections, audio I/O, printers etc., we find that Input/Output subsystems will work on following principles:

- The focus of using devices that gets attached will help to get new developed devices for old systems
- The creation of latest devices that gets interfaced with original standard which are not easy and compatible.

Further we see that for every hardware device there is a device driver which works in support of Operating System which will handle the complete hardware.

Goals for I/O

- Users will be able to access all devices in uniform manner.
- Devices to be named in particular order.
- Operating System without the interruption of user program can able to control recoverable errors.
- Operating System should control security of the devices.
- Operating System to optimize performance of I/O system.

2.2 I/O Hardware

If we talk about I/O devices, we mean to say as storage, communications, and user-interface devices that work with computer by using signals which often can be through wires or by air. These devices get attached with the computer through different ports which can be serial or parallel. It is seen a common set of wires that will connect many such devices is known as bus. The buses in computer architecture cover rigid protocols for certain different messages that can be sent across the bus and procedures in order to solve conflict issues.

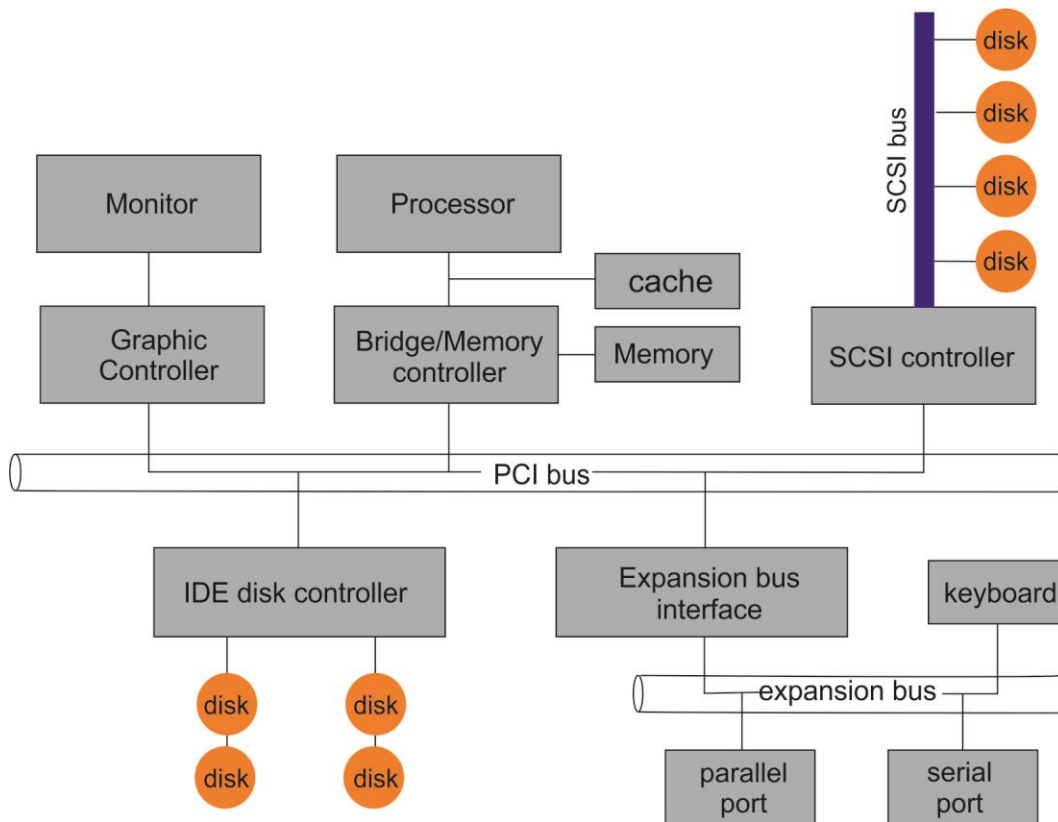


Fig 2.1 Buses

Figure 2.1 shows three of four types of buses that are usually found in modern computers:

- PCI bus that connects high speed having high band width devices to the memory subsystem and the processor or CPU.
- Expansion bus will connect low band width devices that normally transmit data single character at a particular time by use of buffering.
- SCSI bus which connects many SCSI devices to particular SCSI controller.
- Daisy chain bus is a type of bus that will show when a string of devices is connected each other as beads on chain where only single device is directly connected to host computer.

It is seen that a particular way of communicating with devices is using registers that is connected with each port. Registers can be one to four bytes in size which includes a subset of:

- Data-in register which is read by host to have input from device.
- Data-out register is that which is written by host that sends an output.
- Status register having bits works by host to have a position of certain device to feel as idle, ready, busy, error, etc.

- Control register have bits applied by host which determines commands that can also alter settings of various device.

Apart from Input Output, there is memory-mapped I/O, which also communicates with computer which is shown in fig 2.2.

I/O address range (hexadecimal)	device
000–00F	DMA controller
020–021	interrupt controller
040–043	timer
200–20F	game controller
2F8–2FF	serial port (secondary)
320–32F	hard-disk controller
378–37F	parallel port
3D0–3DF	graphics controller
3F0–3F7	diskette-drive controller
3F8–3FF	serial port (primary)

Fig 2.2 Device I/O port locations on PCs

In memory mapped I/O several processor's address space parts are mapped with device where communications is done through reading as well as writing directly from the memory location.

It is beneficial for devices which travel with large quantities of data quickly. These I/O are application with additional mixture of original registers. It is seen that possibly the problem with memory-mapped I/O, is that a process is allowed to write straight to address space that are used by memory-mapped I/O device.

Check your progress 1

1. Which among the following is a category of I/O device?
 - a. storage
 - b. communications
 - c. user-interface
 - d. all

2. Buses are set of_____.
- a. rules
 - b. protocols
 - c. information
 - d. data
3. _____bus connects high speed bandwidth devices to memory subsystem and processor.
- a. PCI bus
 - b. Expansion bus
 - c. SCSI bus
 - d. Daisy Chain

2.3 I/O Drivers

The CPU is not the single brilliant device in the system; every corporal device acquires its own hardware controller. The keyboard, mouse as well as serial ports are controlled proximate a Super IO chip, the IDE disks nearby an IDE controller, SCSI disks nearby a SCSI controller additionally so on. Each hardware controller holds its own control as well as status registers (CSRs) furthermore these contrasts between devices. The CSRs for an Adaptec 2940 SCSI controller are entirely differing from those relevantly an NCR 810 SCSI controller. The CSRs are utilized to start as well as stop the device, to begin it furthermore to diagnose several dilemmas with it. Instead of embedding code to coordinate the hardware controllers in the system into every exercise, the code continues conserved in the Linux kernel. The software that triggers or commands a hardware controller is comprehended as a device driver. The Linux kernel device drivers are, centrally, an assigned library of excepted, memory incumbent, bottom level hardware experiencing routines. It is Linux's device drivers that experience the features of the devices they are administering.

One of the elementary features of subsists that it evacuates the responding of devices. complete hardware devices observe like average files; they can be exposed, closed, read as well as written applying the identical, standard, system calls that are exercised to touch files. Every device in the system is circumscribed nearby a device definite file, for exemplary the early IDE disk in the discipline is represented by /dev/hda. It is found that for block (disk) as well as character devices, such device serves as particular files that are created by mknod command which can be describe as device that uses more and less device numbers. Network devices are shown with the help of special files since they are developed by Linux, as it locates and initialize network controllers in system. Such devices controlled by single device driver have common major device number. The minor

device numbers are used to differentiate among various devices and their controllers having each partition on primary IDE disk contains different minor device number. So, we see that in /dev/hda2, the second partition of primary IDE disk contains major number 3 and minor number 2. In this, Linux maps the device special file that is passed in system calls to device driver that uses major device number and number of system tables as character device table labelled as chrdevs.

Device driver is a sort of a program which is constructed for I/O device. These drivers will make use of I/O operations on certain devices. Here the arrangement will take care of number of certain terminals that contains dissimilar terminal driver. The work of device driver is to:

- Take the request from individual software.
- Manage the request send by the device.

In order to handle a request, a device driver will take care of the request as:

- If the request appears to read block N and the driver is ease at said time, then it will process the request immediately
- If the driver is busy, then the request so appears will be placed in the queue.

Check your progress 2

1. Drivers are the_____.

- | | |
|------------|----------------|
| a. data | c. information |
| b. program | d. all |

2.4 DMA Controlled I/O

In order to exercise an interrupt obtained from device drivers, the data is assigned from the hardware which will work correctly if the data appear is less. If the interrupt is not recognised, then number of time it will achieve in hardware device gets shoot up the interrupt with routine occurrence with result in stumpy transferring of data. In case of high speed devices such as hard disk or Ethernet, the data transfer rate will be much higher. If you see the data transfer rate in case of SCSI device, you will find that it will be above 45 Mbytes of data transfer per second.

To see such variation in data transfer, Direct Memory Access is applied will be breakthrough such type of problems. As seen, a PC's ISA DMA controller will carry 8 DMA channels out of which, 7 channels are feasible by device drivers. In order to begin with data transfer, device driver will allot DMA channel's address along with count registers in the direction of data transfer that will read or write data. During that period, the device will start the DMA when it is required.

It is found that while conducting DMA, the device drivers needs to be careful. Initially, the DMA controller will know nothing about virtual memory that has appeared singly in the system. With this effect, the memory will go through continuing DMA from required contiguous block of physical memory which explains that the DMA cannot be directly placed inside virtual address space.

It is seen that the DMA channels are restricted, as they possess only 7 channels out of 16 which are difficult while sharing among device drivers. Similarly like interrupts, the device drivers will able to judge and manage which DMA channel should be applied. Many device drivers carry standard DMA channel, which is same in case of interrupts.

Linux operating system will takes care of all DMA channels applications with the help of vector `dma_chan` data structures. Such DMA data structure carry pointer which shows presence of DMA channel owner and a flag that will reflect on allocation of DMA channel.

- The devices which transfer big data will be of no use of data transfer in CPU during data input to registers.
- Such type of work can be carried off by Direct Memory Access and Controller.
- Initially the command is sent to DMA controller with data storage location and data transfer with number of bytes of data to transfer.
- DMA controller is independent component of computer where different bus mastering I/O cards has their own DMA hardware.
- Handshaking exist among DMA controllers and certain devices will be done through double DMA request and DMA acknowledge wires.

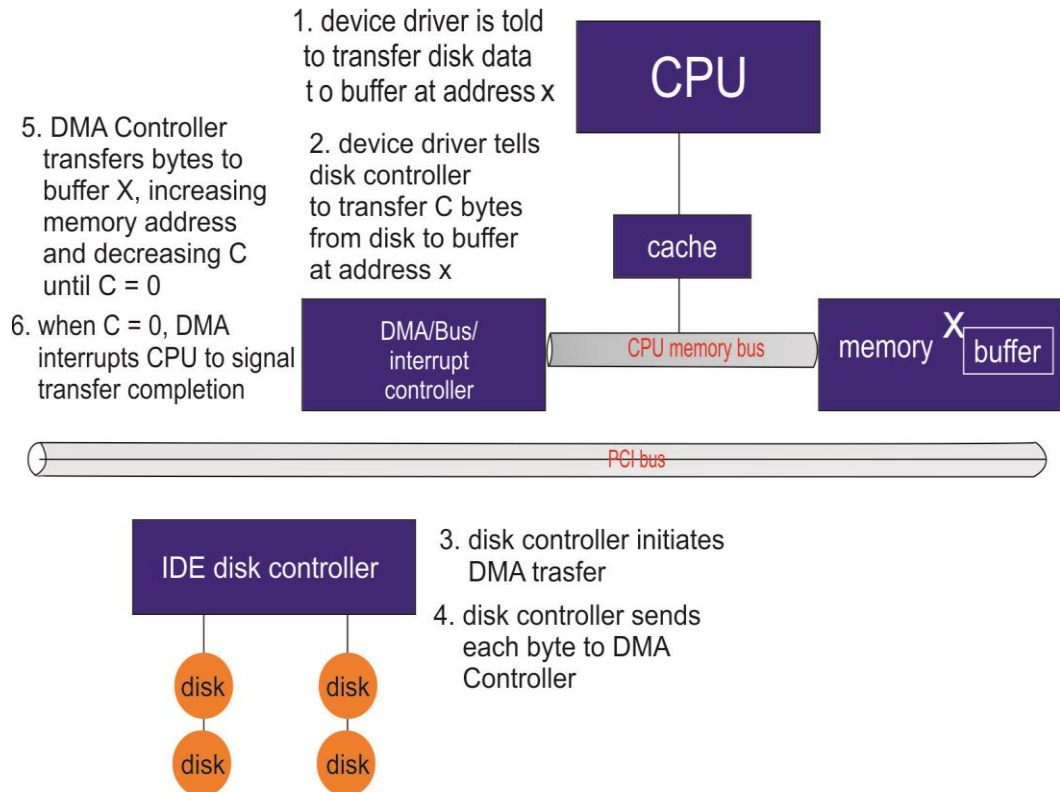


Fig 2.3 Illustrates the DMA process.

Check your progress 3

1. A PC's ISA DMA controller holds _____DMA channels.
 - a. 4
 - b. 8
 - c. 16
 - d. 32

2.5 Programmed I/O

The basic computer structure is shown in figure 2.4.

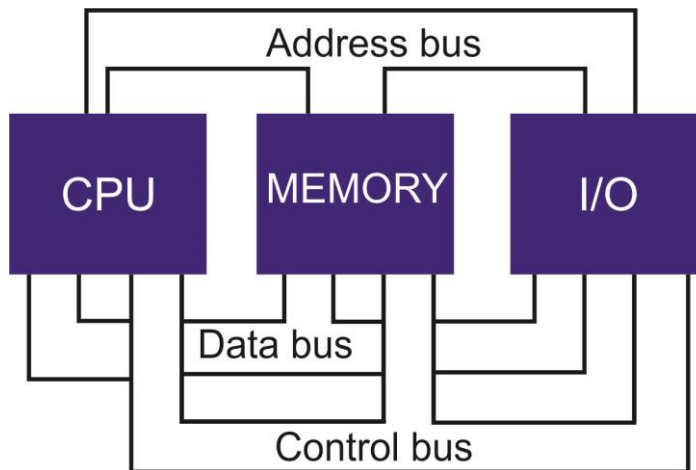


Fig 2.4 basic computer structure

Programmed I/O is the method of transferring of data done by the CPU in supervision with driver software control so as to make use of registers or memory on a certain device. Since in case of Direct Memory Access (DMA), the data is transferred through certain device in order to work with system memory. Similarly, the work of PIO is to take care of data transfer through normal memory that will be loaded and kept on others.

In case of UDI, the Programmed I/O working is done with the support of environmental service calls which is coded as function calls instead of direct memory references in certain drivers.

It is found that PIO will keep an unclear data type which will carry addressing, data translation as well as access restriction information which is needed to work device or memory address in certain address space.

In case of transaction which will work a device where PIO offset is shown will highlight the offset that carries a space which is referenced by PIO handle with which I/O operations occur.

Synchronization that exists among PIO transaction lists can be demonstrated with syntax serialization domain argument which will map PIO call. Working of PIO transaction list can be sequenced due to working of other PIO transaction also are mapped with certain device as for a particular device and serialization domain, at least one thread will be actively working corresponds to transaction list and these transaction list will work to complete before other transaction list at the start of serial domain execution.

In such cases, there will no ordering confirmation with regards to working of transactions lists having a syntax `udi_pio_trans` calls instead of the calls made from similar portion exist in similar serialization domain as processed in FIFO order.

Check your progress 4

1. Programmed I/O is basically data transfers by CPU under software control to access_____.
- | | |
|--------------|----------|
| a. registers | c. files |
| b. data | d. all |

2.6 I/O Supervisors

The operating systems will keep track of all input/output devices which are attached to a computer system.

Device drivers accompany operating systems and enable a computer system to be configured for specific hardware. Most hardware peripheral devices have their own device drivers which need to be installed for the operating system to communicate with these devices.

Check your progress 5

1. Device drives are need to_____.
- | | |
|------------|---------|
| a. run | c. copy |
| b. install | d. all |

2.7 Let Us Sum Up

In this unit we have learned:

- That I/O devices are concerning with storage, communications and user-interface which will work and interface with the help of computer by using analog and digital signals which is there through wires or through air.

- The buses in computer architecture cover rigid protocols for certain different messages that can be sent across the bus and procedures in order to solve conflict issues.
- Drivers or Device driver is a program or routine that for designed for an I/O device.

2.8 Answers for Check Your Progress

Check your progress 1

Answers: (1-d), (2-b), (3-a)

Check your progress 2

Answers: (1-b)

Check your progress 3

Answers: (1-b)

Check your progress 4

Answers: (1-a)

Check your progress 5

Answers: (1-b)

2.9 Glossary

1. **I/O devices** - These can be storage, communications, user-interface devices that communicate by using the computer through signals that used to send by wires or through air.
2. **Buses** - Are protocols for certain different messages that can be sent across the bus and procedures in order to solve conflict issues.
3. **Drivers or Device driver** - Program or routine that for designed for an I/O device.

2.10 Assignment

Write short note on directory structure of Operating System.

2.11 Activities

Collect some information on I/O devices.

2.12 Case Study

Generalized the basic computer architecture and discuss.

2.13 Further Readings

1. Operating System Concept by Abraham Silberschatz, Peter Baer Galvin, Greg Gagne.
2. Programming Be Operating System by Dan Sydow.
3. Computer Science & Application by Dr. Arvind Mohan Parashar, Chandresh Shah, Saurab Mishra.
4. An Integrated Approach to Software Engineering by Pankaj Jalote.
5. An Operating Systems by Raphael A.

Block Summary

In this block, students have learnt and understand about the basic of file system management and input output memory management. The block gives an idea on the study and concept of disk space allocation, disk scheduling and input out device drivers. The students have be well explained on the concepts of DMA controlled input output and basic programmed input output.

The block detailed about the basic of programmed DMA input output management techniques. The concept related to input output supervisors and input output drivers will also be explained to the students. The student will be demonstrated practically about programmed input output technique.

Block Assignment

Short Answer Questions

1. What is Disk scheduling?
2. Explain the function of disk space allocation?
3. Write note on I/O Hardware?
4. Write short note on Contiguous Allocation?

Long Answer Questions

1. Write short notes on types of buses?
2. Write short note on meta data tags?
3. Write note on DMA controlled I/O?

Enrolment No.

1. How many hours did you need for studying the units?

Unit No	1	2	3	4
Nos of Hrs				

2. Please give your reactions to the following items based on your reading of the block:

Items	Excellent	Very Good	Good	Poor	Give specific example if any
Presentation Quality	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____
Language and Style	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____
Illustration used (Diagram, tables etc)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____
Conceptual Clarity	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____
Check your progress Quest	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____
Feed back to CYP Question	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____

3. Any Other Comments

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....



“

*Education is something
which ought to be
brought within
the reach of every one.*

”

- Dr. B. R. Ambedkar



Dr. Babasaheb Ambedkar Open University
Jyotirmay' Parisar, Opp. Shri Balaji Temple, Sarkhej-Gandhinagar Highway, Chharodi,
Ahmedabad-382 481.

FUNDAMENTALS OF OPERATING SYSTEM

PGDCA 104

BLOCK 4: BASICS OF DISTRIBUTED OPERATING SYSTEM

**Dr. Babasaheb Ambedkar Open University
Ahmedabad**



FUNDAMENTALS OF OPERATING SYSTEM



Knowledge Management and
Research Organization
Pune



Editorial Panel

Author

Er. Nishit Mathur

Language Editor

Prof. Jaipal Gaikwad

Graphic and Creative Panel

Ms. K. Jamdal

Ms. Lata Dawange

Ms. Pinaz Driver

Ms. Ashwini Wankhede

Mr. Kiran Shinde

Mr. Prashant Tikone

Mr. Akshay Mirajkar

Copyright © 2015 Knowledge Management and Research Organization.

All rights reserved. No part of this book may be reproduced, transmitted or utilized in any form or by a means, electronic or mechanical, including photocopying, recording or by any information storage or retrieval system without written permission from us.

Acknowledgment

Every attempt has been made to trace the copyright holders of material reproduced in this book. Should an infringement have occurred, we apologize for the same and will be pleased to make necessary correction/amendment in future edition of this book.

The content is developed by taking reference of online and print publications that are mentioned in Bibliography. The content developed represents the breadth of research excellence in this multidisciplinary academic field. Some of the information, illustrations and examples are taken "as is" and as available in the references mentioned in Bibliography for academic purpose and better understanding by learner.'



ROLE OF SELF INSTRUCTIONAL MATERIAL IN DISTANCE LEARNING

The need to plan effective instruction is imperative for a successful distance teaching repertoire. This is due to the fact that the instructional designer, the tutor, the author (s) and the student are often separated by distance and may never meet in person. This is an increasingly common scenario in distance education instruction. As much as possible, teaching by distance should stimulate the student's intellectual involvement and contain all the necessary learning instructional activities that are capable of guiding the student through the course objectives. Therefore, the course / self-instructional material are completely equipped with everything that the syllabus prescribes.

To ensure effective instruction, a number of instructional design ideas are used and these help students to acquire knowledge, intellectual skills, motor skills and necessary attitudinal changes. In this respect, students' assessment and course evaluation are incorporated in the text.

The nature of instructional activities used in distance education self-instructional materials depends on the domain of learning that they reinforce in the text, that is, the cognitive, psychomotor and affective. These are further interpreted in the acquisition of knowledge, intellectual skills and motor skills. Students may be encouraged to gain, apply and communicate (orally or in writing) the knowledge acquired. Intellectual-skills objectives may be met by designing instructions that make use of students' prior knowledge and experiences in the discourse as the foundation on which newly acquired knowledge is built.

The provision of exercises in the form of assignments, projects and tutorial feedback is necessary. Instructional activities that teach motor skills need to be graphically demonstrated and the correct practices provided during tutorials. Instructional activities for inculcating change in attitude and behavior should create interest and demonstrate need and benefits gained by adopting the required change. Information on the adoption and procedures for practice of new attitudes may then be introduced.

Teaching and learning at a distance eliminates interactive communication cues, such as pauses, intonation and gestures, associated with the face-to-face method of teaching. This is particularly so with the exclusive use of print media. Instructional activities built into the instructional repertoire provide this missing interaction between the student and the teacher. Therefore, the use of instructional activities to affect better distance teaching is not optional, but mandatory.

Our team of successful writers and authors has tried to reduce this.

Divide and to bring this Self Instructional Material as the best teaching and communication tool. Instructional activities are varied in order to assess the different facets of the domains of learning.

Distance education teaching repertoire involves extensive use of self-instructional materials, be they print or otherwise. These materials are designed to achieve certain pre-determined learning outcomes, namely goals and objectives that are contained in an instructional plan. Since the teaching process is affected over a distance, there is need to ensure that students actively participate in their learning by performing specific tasks that help them to understand the relevant concepts. Therefore, a set of exercises is built into the teaching repertoire in order to link what students and tutors do in the framework of the course outline. These could be in the form of students' assignments, a research project or a science practical exercise. Examples of instructional activities in distance education are too numerous to list. Instructional activities, when used in this context, help to motivate students, guide and measure students' performance (continuous assessment)



PREFACE

We have put in lots of hard work to make this book as user-friendly as possible, but we have not sacrificed quality. Experts were involved in preparing the materials. However, concepts are explained in easy language for you. We have included many tables and examples for easy understanding.

We sincerely hope this book will help you in every way you expect.

All the best for your studies from our team!



FUNDAMENTALS OF OPERATING SYSTEM

Contents

BLOCK 1: INTRODUCTION TO OPERATING SYSTEMS

UNIT 1 BASICS OF OS

Definition and Function of operating systems, Evolution of operating system, Operating system structure-monolithic layered, virtual machine and Client server

UNIT 2 TYPES OF OPERATING SYSTEM

Different types of operating system-real time systems, multi-user System, distributed system

UNIT 3 BATCH OPERATING SYSTEM

Introduction to basic terms and batch processing system: Jobs, Processes files, command interpreter

BLOCK 2: MEMORY MANAGEMENT AND PROCESS SCHEDULING

UNIT 1 MEMORY MANAGEMENT

Logical and Physical address protection, paging, and segmentation, Virtual memory, Page replacement algorithms, Cache memory, hierarchy of memory types, Associative memory

UNIT 2 PROCESS SCHEDULING

Process states, virtual processor, Interrupt mechanism, Scheduling algorithms Performance evaluation of scheduling algorithm, Threads



BLOCK 3: FILE AND I/O MANAGEMENT

UNIT 1 FILE SYSTEM

File systems-Partitions and Directory structure, Disk space allocation, Disk scheduling

UNIT 2 I/O MANAGEMENT

I/O Hardware, I/O Drivers, DMA controlled I/O and programmed I/O, I/O Supervisors

BLOCK 4: BASICS OF DISTRIBUTED OPERATING SYSTEM

UNIT 1 DISTRIBUTED OPERATING SYSTEM

Introduction and need for distributed OS, Architecture of Distributed OS, Models of distributed system

UNIT 2 MORE ON OPERATING SYSTEM

Remote procedure Calls, Distributed shared memory, Unix Operating System: Case Studies



Dr. Babasaheb
Ambedkar
Open University

PGDCA 104

FUNDAMENTALS OF OPERATING SYSTEM

BLOCK 4: BASICS OF DISTRIBUTED OPERATING SYSTEM

UNIT 1

DISTRIBUTED OPERATING SYSTEM

UNIT 2

MORE ON OPERATING SYSTEM

BLOCK 4: BASICS OF DISTRIBUTED OPERATING SYSTEM

Block Introduction

An operating system is important software which makes the computer to run. It handles all the computer processes and runs the hardware. It makes you to communicate with computer without having command on its language. It is seen that your computer operating system manages all software and hardware functions. The main idea of operating system is to coordinate will all processes and links these processes with central processing unit (CPU), memory and storage.

In this block, we will detail about the basic of distributed operating system and its modelling techniques. The block will focus on architecture and distribution of distributed operating system with study about their characteristics. The concept of distributed operating system layout and working characteristics are also explained.

In this block, the student will made to learn and understand about the basic of remote procedure calls and its techniques. The concept related to distributed shared memory and Unix operating system will also be explained to the students. The student will be demonstrated practically about Unix architecture.

Block Objective

After learning this block, you will be able to understand:

- Concept of distributed operating system
- About the architecture of distributed OS
- Characteristics models of distributed system.
- Basic of remote procedure calls
- Concept of Unix Operating System

Block Structure

Unit 1: Distributed Operating system

Unit 2: More on Operating System

UNIT 1: DISTRIBUTED OPERATING SYSTEM

Unit Structure

- 1.0 Learning Objectives**
- 1.1 Introduction**
- 1.2 Need for Distributed OS**
- 1.3 Architecture of Distributed OS**
- 1.4 Models of Distributed System**
- 1.5 Let Us Sum Up**
- 1.6 Answers for Check Your Progress**
- 1.7 Glossary**
- 1.8 Assignment**
- 1.9 Activities**
- 1.10 Case Study**
- 1.11 Further Readings**

1.0 Learning Objectives

After learning this unit, you will be able to understand:

- About object-based distributed object systems
- Characteristics models of Content Delivery Network
- About Processor Pool Model

1.1 Introduction

Distributed Operating System as shown in fig 1.1 is a model where distributed applications are running on multiple computers linked by communications. Such type of operating system is an advancement of network operating system which is basically designed for higher communication and integration levels for certain machines which are on network. It will appear as simple centralized operating system to its users where it can handle and perform multiple independent central processing units operations.

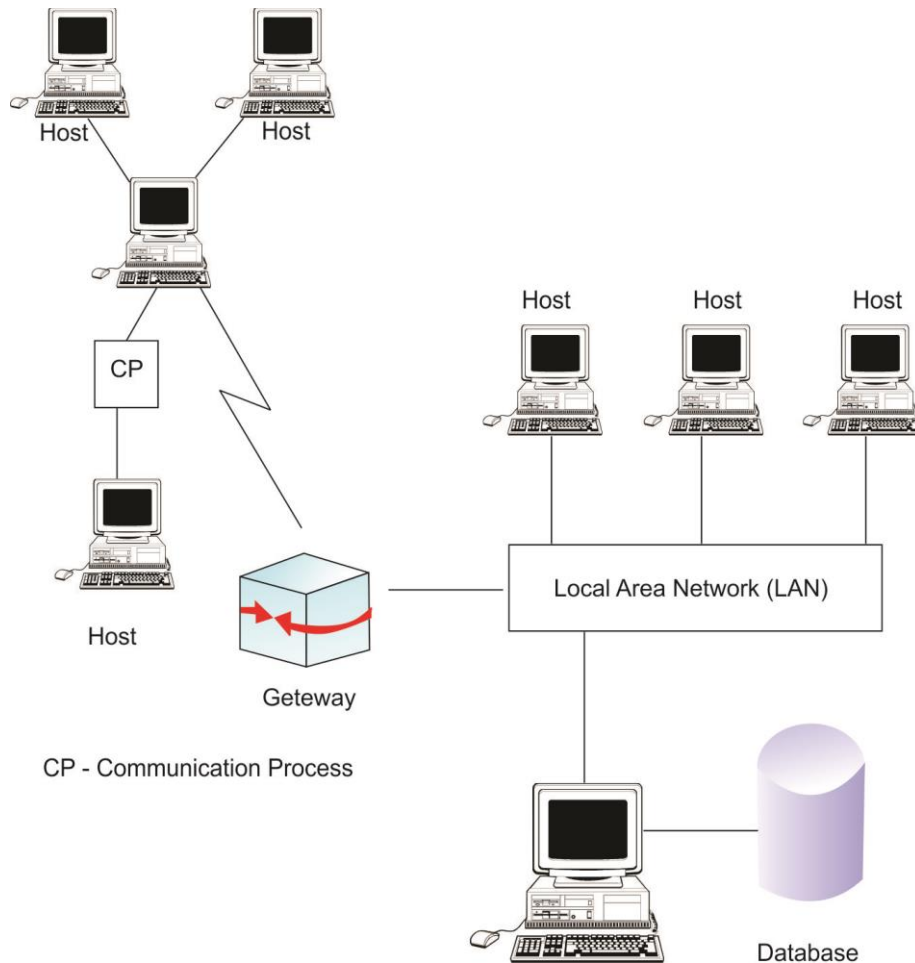


Fig 1.1 Distributed Operating System

These systems are referred as loosely coupled systems where each processor has its own local memory and processors communicate with one another through various communication lines, such as high speed buses or telephone lines. By loosely coupled systems, we mean that such computers possess no hardware connections at the CPU - memory bus level, but are connected by external interfaces that run under the control of software. The Distributed O/S involves a

collection of independent computer systems, competent of communicating and cooperating with each other from side to side by LAN / WAN. A Distributed OS provides a virtual machine abstraction to its users and wide sharing of resources like as computational capacity, I/O and files etc.

1.2 Need for Distributed OS

The ODP standards, and this text, assume a model where distributed applications are running in multiple processes in multiple computers linked by communications. The application programmer will be supported by a programming environment and run-time system that will make many aspects of distribution in the system transparent. For instance the programmer may not have to worry about where the parts of the application are running this can all be taken care of, if required; this is called location transparency.

There is another approach to supporting applications in a distributed system that is by using a distributed operating system. On every computer system with an operating system the O/S provides an interface which the programs use to obtain services, such as input and output.

Distributed systems are potentially more reliable than a central system because if a system has only one instance of some critical component, such as a CPU, disk, or network interface, and that component fails, the system will go down. When there are multiple instances, the system may be able to continue in spite of occasional failures. In addition to hardware failures, one can also consider software failures. Distributed systems allow both hardware and software errors to be dealt with.

A distributed system is a set of computers that communicate and collaborate each other using software and hardware interconnecting components. Multiprocessors (MIMD computers using shared memory architecture), multicomputer connected through static or dynamic interconnection networks (MIMD computers using message passing architecture) and workstations connected through local area network are examples of such distributed systems.

A distributed system is managed by a distributed operating system. A distributed operating system manages the system shared resources used by multiple processes, the process scheduling activity (how processes are allocating on available processors), the communication and synchronization between running processes and so on. The software for parallel computers could be also tightly coupled or loosely coupled. The loosely coupled software allows

computers and users of a distributed system to be independent each other but having a limited possibility to cooperate. An example of such a system is a group of computers connected through a local network. Every computer has its own memory, hard disk. There are some shared resources such files and printers. If the interconnection network broke down, individual computers could be used but without some features like printing to a non-local printer.

Check your progress 1

1. Distributed Operating System is also called as?
 - a. loose coupled systems
 - b. tight coupled systems
 - c. fat coupled systems
 - d. thin coupled systems
2. In Distributed Operating System, the processor has its own_____.
 - a. data
 - b. information
 - c. memory
 - d. all
3. The distributed systems comprises of critical component like_____.
 - a. hard disk
 - b. CPU
 - c. memory
 - d. modem

1.3 Architecture of Distributed OS

The architecture of distributed O/S comprises off our styles such as:-

- Layered architecture
- Object-based architecture
- Data-centered architecture
- Event-based architecture

The architecture is organized into logically different components, and distributed components over various machines.

The layered architecture is an arrangement of client system model as shown in fig 1.2.

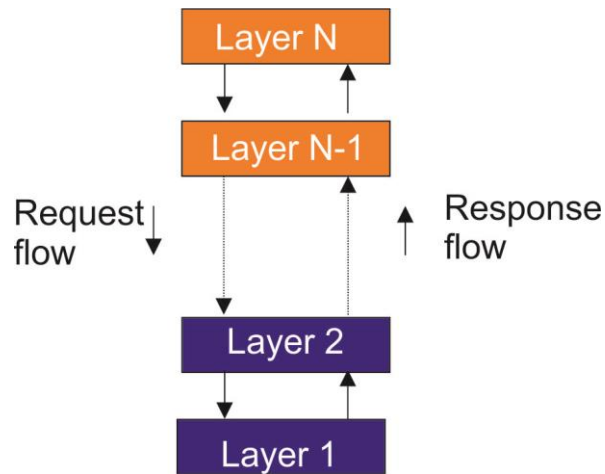


Fig 1.2 layered architecture arrangement

Fig 1.3 shows an object-based style for distributed object systems which is less structured having component similar as object and containing connector as RPC or RMI.

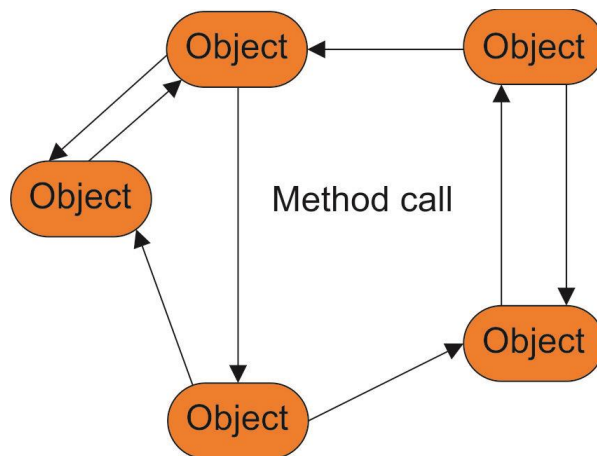


Fig 1.3 object-based distributed object systems

In this, the decoupling processes in space and also time led to various alternative styles as shown in fig 1.4.



Fig 1.4 Styles in object-based distributed object systems

In case of publish/subscribe event-based architecture as shown in fig 1.5 carries-

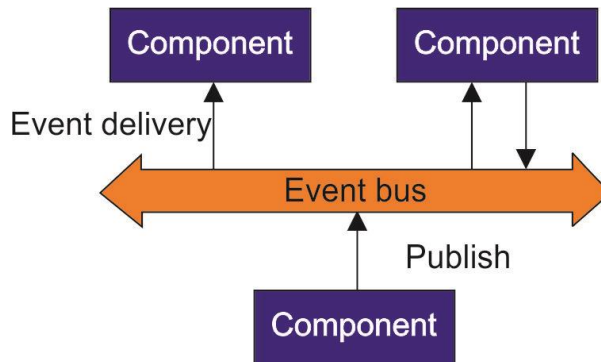


Fig 1.5 publish/subscribe event-based architecture

- Publish-subscribe
- Broadcast
- Point-to-point

Here the decouples sender & receiver acts as asynchronous communication. In the event driven architecture (EDA), the activities such as production, detection, consumption of reaction results in occurrence of various events.

Here we explain an event as an important change in state. The benefit of such type of architecture is that they are loosely coupled since they require no explicitly that belongs to each other.

In case of shared data space which a data centered + event-based architecture as shown in fig 1.6, contains-

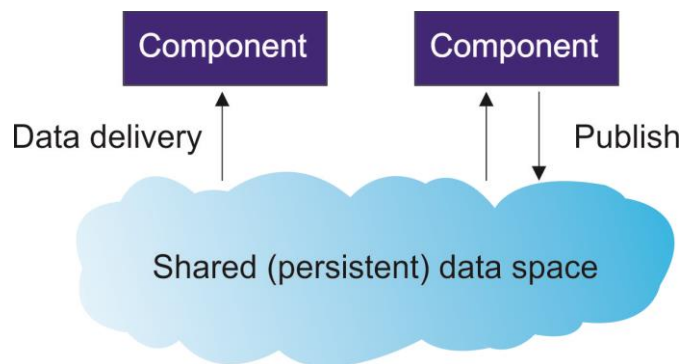


Fig 1.6 data centered

Such type of data centred architecture access and update data that are kept for data-centred system. In this, the processes communicate or exchange the data or information mainly by reading and modifying data in some shared repository. It is seen that a many web-based distributed systems communicate among themselves through the use of shared Web based data services.

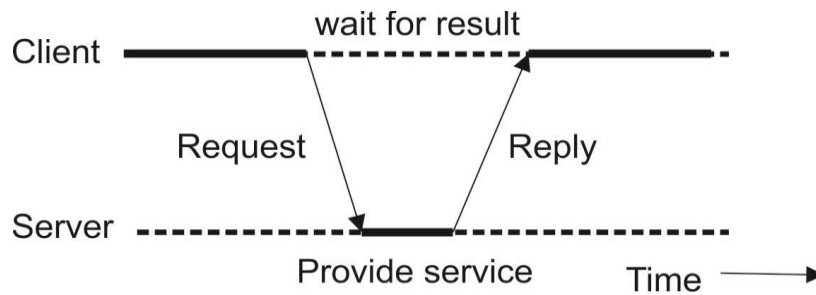


Fig 1.7 Client Server Model

Further it is seen that in basic Client Server Model as shown in fig 1.7, there exists certain main features that-

- There are certain process that offers the server services.
- There are certain process which uses client services.
- There seems that the clients and servers can be on different machines.
- There follows that clients request/reply model be used for services.
- There exists certain synchronous communication where request or reply protocol takes place.
- In case of LANs, it works with connectionless protocol as they are unreliable.
- In case of WANs, the communication exists as connection oriented TCP/IP as they are reliable.

Since many years it is seen that there exists high level development growth in peer-to-peer systems. These can be:-

- Structured P2P: where the nodes are arranged having a particular distributed data structure.
- Unstructured P2P: where the nodes have arbitrarily selected other close nodes.
- Hybrid P2P: where some nodes are presented as special functions in a good organized manner.

In case of structured P2P systems, we see that:-

Nodes are arranged in a structure overlay network like logical ring as shown in fig 1.8 and develop a specific node that are responsible for services based only on their ID.

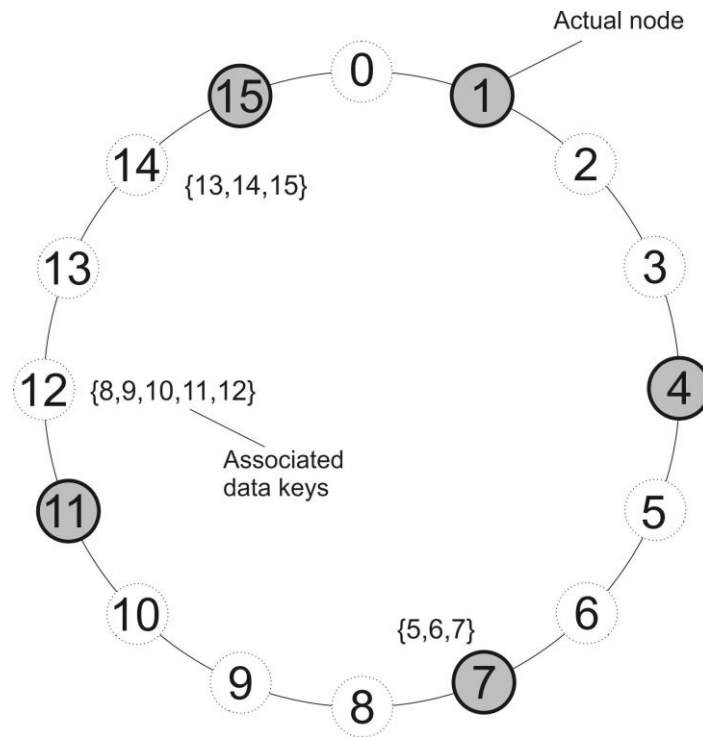


Fig 1.8 Nodes arranged logical ring network

Here the idea is to use a distributed hash table (DHT) to arrange the nodes. The earlier hash functions transform a key to hash value that can be used as an index in hash table:

- Keys are unique –each key shows an object to keep in the table;
- Hash function value is there to insert an object in hash table and can get anytime.

It is seen that in DHT, the data objects and nodes are both given a key which hashes to a random number from a very big identifier space. In this, mapping function gives objects to nodes that are based on hash function value. In such case a lookup based on hash function value will go back to the network address of the node which keeps all requested object.

In case of unstructured P2P systems, it is seen that:-

- The information or data depends on randomized algorithms for developing an overlay network.
- There occur several unstructured P2P systems that will try to maintain a random graph.

The basic idea of unstructured P2P systems is that every node is required to contact a randomly selected other node which:-

- allows each peer to maintain a partial view of network that carries n nodes
- will select each node P periodically and also node Q from its partial view
- will exchange P and Q information and members from their respective partial views

The hybrid architecture of Client-server combined with P2P where edge server architectures used for Content Delivery Networks is shown in fig 1.9

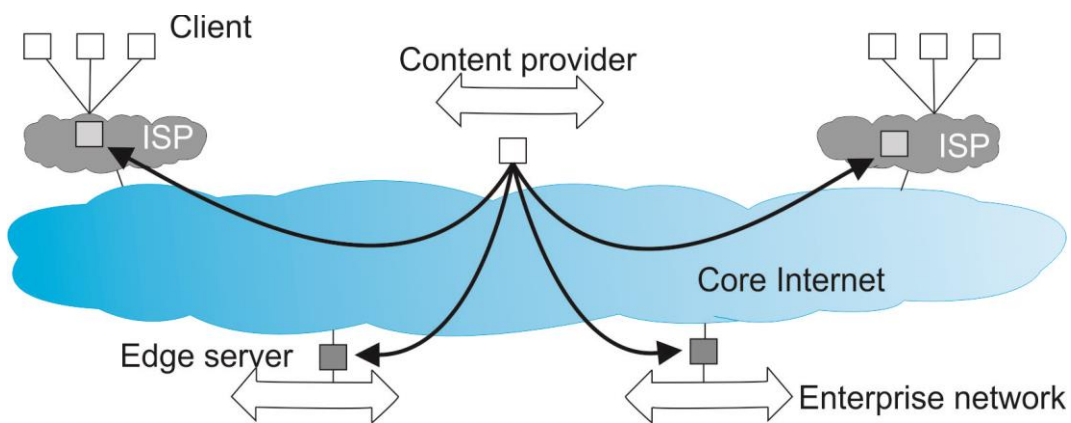


Fig 1.9 Content Delivery Network

In another hybrid architecture of C/S with P2P which is a Bit Torrent, where users cooperate in file distribution is shown in fig 1.10

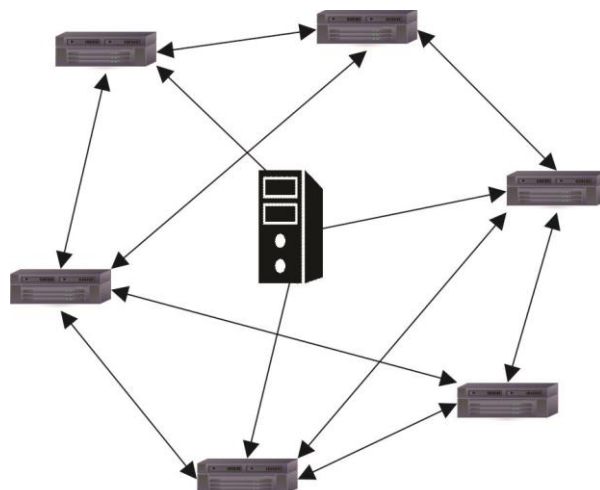


Fig 1.10 hybrid architecture of C/S with P2P

It is seen that once a node finds a path to download a file, it will joins a swarm of downloader which will parallel get file chunks from the source, and will further distribute such chunks among them. It can be shown by fig 1.11.

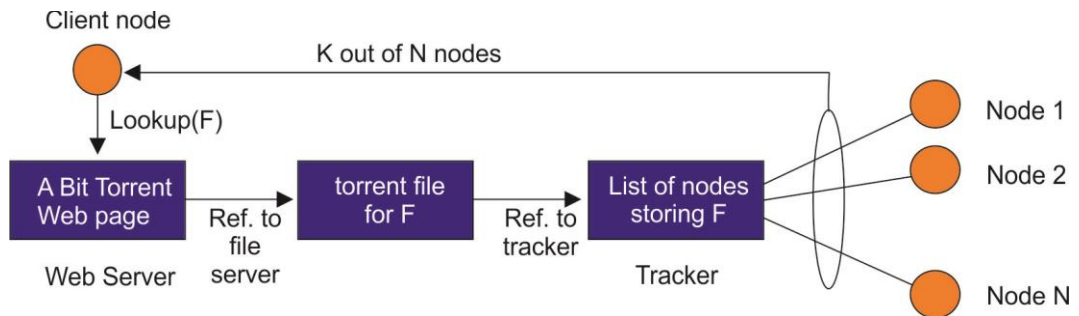


Fig 1.11 architecture of distributed O/S

The architecture of distributed O/S comprises of multiple computers which:

- Will not share memory or Clock.
- Will communicate by exchanging of messages over a network.

The advantages of distributed system are:-

- They can easily share resource related to hardware and software.
- They have better performance that led to rapid response time and higher system throughput.
- They are capable of improved reliability and availability.
- They can expand by modular expansion.
- They have low set up price and have better performance.
- They are traditional time-sharing systems.

The distributed operating system appears as a centralized OS for a single machine that can run on multiple independent computers. They are clear with the transparency when they are handled by users.

The distributed operating system has certain drawbacks:

- They lack in latest and current global knowledge.
- They are poor in naming.
- They possess low scalability.
- They are much compatible.

- They have less process synchronization.
- They have weak resource management.
- They are not secure.
- They possess poor structuring of OS.

There exist numerous levels of compatibility:

1. **Binary level** - They are restrictive if same binary is to be executed on all computers.
2. **Execution level** - In case of an execution level, if similar source code gets compiled then they can execute on all computers.
3. **Protocol level** - In case of protocol level, when each computer can perform on different OS, then they:
 - support common protocols for interaction
 - least restrictive

The structure of distributed operating system comprises of -

Monolithic kernel

In this, every computer OS kernel contains necessary services.

Such type of kernel is not advisable for distributed systems.

Collective kernel

In this, the structure of microkernel or the nucleus runs on many computers.

In this the OS services can be done as individual processes which are not necessarily for all computers.

Here the microkernel supports interaction among processes by way of messages.

Object-oriented OS

In this type of structure, the services performed by OS are implemented in shape of an object.

Check your progress 2

1. In an object-based style system, there exists a connector as____.
 - a. RPC
 - b. RIM
 - c. RCP
 - d. PCR
2. Which is not an activity of event driven architecture?
 - a. production
 - b. organisation
 - c. consumption
 - d. detection
3. The shared data space architecture is_____.
 - a. data cantered architecture
 - b. event-based architecture
 - c. data cantered + event-based architecture
 - d. none
4. In which peer-to-peer system the nodes are arranged having a particular distributed data structure.
 - a. Structured P2P
 - b. Unstructured P2P
 - c. Hybrid P2P
 - d. all

1.4 Models of distributed system

There are various types of distribute operating system models such as:-

- Workstation-server Model - Workstation may be a standalone system or a part of a network
- Processor-pool Model - Provides processing power on a demand basis

- Integrated Hybrid Model - Workstations used as processor pools

Workstation-server Model

The workstation model is a basic arrangement where system comprises of workstations that will be high end personal computers that are spread across the building or campus and are joined or connected through high speed LAN. This type of arrangement is shown in fig 1.12.

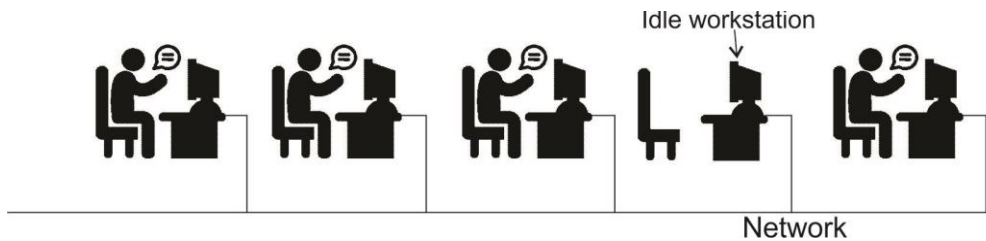


Fig 1.12 workstation server model

Here we see that some of the workstations in offices are particularly for single user, whereas others can be for public during the course of a day. We see that in such cases, a workstation can either have single user logged into it or will remain idle.

It is found that in some systems, the workstations contain local disks and some are without disk. Such type of computers is called as diskless workstations where as workstations containing a disk is often known as diskful or diskly workstations. It is found that the workstations having no disk, will store file in the one or more remote file servers. These workstations will requests for read and write of files that are sent to file server which after performing sends the file back.

Such type of diskless workstations is found in universities and companies for several reasons. The workstation with small or slow disks is normally more expensive as compared to one or two file servers that carries huge, fast disks and accessed over the LAN.

Apart from this, the diskless workstations are popular as their maintenance is easy. When in this case, a compiler comes out, and then system administrator can easily install it on small number of file servers in respective machines. In this case taking backup of files and maintain the hardware is somewhat easy with centrally placed hard disk.

Finally, it is seen that a diskless workstations will supply symmetry and flexibility. Here the user can walk up to any workstation in system and can easily

logged in. It is examined that when all files are stored on local disks, then by accessing any workstation will be easier to get files as compared to getting your own. If the workstations contain private disks, such disks can be used in either of ways:

1. Paging and temporary files.
2. Paging, temporary files, and system binaries.
3. Paging, temporary files, system binaries, and file caching.
4. Complete local file system.

From above, the first design is based on observation that since it is easier to keep all user files on single file servers, disks also requires paging for temporary files. This model is used only for paging and files that are temporary, unshared, and can be leftover during end of login session.

The next arrangement is an alternative of the first one as in this, the local disks keeps the executable programs like compilers, text editors and electronic mail handlers. If a program is called, it will be obtained from the local disk as a replacement for file server which lowers the network load. As such programs rarely changes, they get installed on local disks and can be stored for future use. On obtaining the new release of system program, these will be shown on all machines.

The third option uses local disks for open a cache. In this, users can download files from file servers to its own disks which will be read and write by him and can again be uploaded at the end of login session. The idea of this arrangement is to keep long-term storage in a particular place, but reduce network load by keeping files locally at the time they are used.

The last option shows that every machine will ideally have its own self-contained file system that can be mounted or accessed by other machines file systems. The idea is that every machine is on the whole self-contained and that contact with the outside world is inadequate. This organization makes available a uniform as well as guaranteed reply time for the user further more position little load on the network. The disadvantage is to facilitate sharing is additional difficult, and the resulting system is largely close to network operating system as compared to a true transparent distributed operating system.

The advantages of the workstation model are multiple and comprehensible. The model is undoubtedly easy to comprehend. Users have a predetermined amount of committed computing power, more over accordingly guaranteed

response time. Complicated graphics programs can be extremely fast, in view of the fact that they can have straight access to the screen. Every user has a great degree of independence and be able to distribute his workstation's assets as he sees fit. Local disks put in to this independence, furthermore make it likely to carry on working to a smaller or better degree even in the countenance of file server collides.

Processor Pool Model

Even though by means of idle workstations put in a little computing power to the system, it does not deal with supplementary primary issues. An optional move towards this is to build a processor pool, which is rack full of CPUs in machine room that can be animatedly owed to users on order. The processor pool comes near as shown in Fig. 1.13.

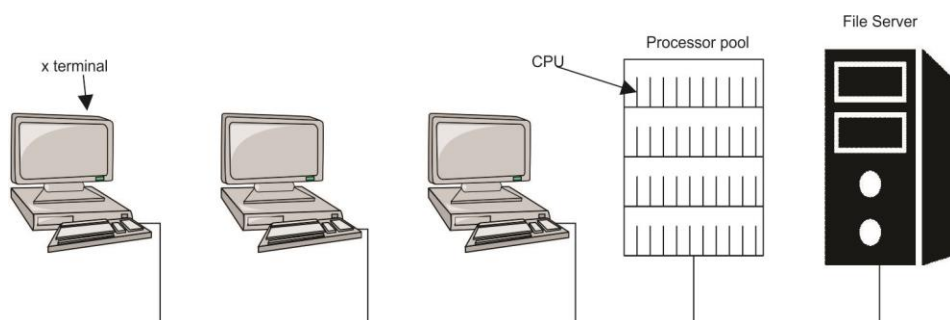


Fig 1.13 processor pool model

In its place, giving users personal workstations, this model gives high-performance graphics terminals like X terminals. This design is based on the inspection that what numerous users actually would like is high-quality graphical interface as well as high-quality performance. Theoretically, it is greatly faster to usual timesharing as compared to personal computer model, even though it is built with up to date technology involving low cost microprocessors.

The inspiration for the processor pool thought comes as of taking the diskless workstation proposal a step extra. If the file system be able to centralise in a little number of file servers to increase economies of scale, it have to be possible to do the similar thing for computer servers. Besides putting all the CPUs in a large rack in the machine room, power supply as well as other packaging costs is able to abridged, giving more computing power for a specified amount of money. In addition, it authorizes the use of cheaper X terminals, and decouples the amount of users as of number of workstations. The model in addition allocate for simple incremental growth. If the computing pack increases by 10%, you can immediately buy 10% additional processors along with them in the pool.

In result, we are converting all computing power into idle workstations with the purpose of to access vigorously. Users can be allocate as several CPUs as they need for little periods, after which they are revisit to the pool as a result that the other users can have them. There is no idea of possession here as all the processors fit in equally to everyone.

The biggest fight for centralizing the computing power in a processor pool comes as of queuing theory. A queuing scheme is a circumstance in which users produce accidental requests for work from a server. When the server is full of activity, the users queue for service as well as process in turn. Frequent examples of queuing systems are:-

- Bakeries
- Airport check-in counters
- Supermarket check-out counters

The basics of such system are shown in the Fig. 1-14.

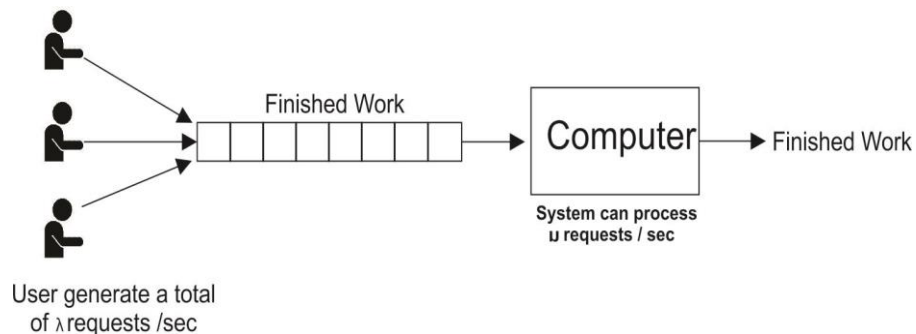


Fig 1.14 queuing systems

Queuing systems are helpful for the reason that they can be easily modelled analytically. Allow us call the whole input rate requests per second, as of all the users combined. Allow us call the rate at which the server can practice the requests. For steady operation, we should have $\lambda < \mu$. If the server be able to handle 100requests/sec, other than the users continuously generate 110requests/sec, then the queue will produce with no bound.

Integrated Hybrid Model

A probable negotiation is to give each user with a personal workstation plus to have a processor pool in addition. While this solution is further costly than

either a clean workstation model or a clean processor pool model, having the advantages of both.

Interactive work can be completed on workstations, giving certain reply. Idle workstations, on the other hand, are not exploiting, making for a simpler system design. They are now left idle. In its place, all non-interactive procedures run on the processor pool, because they do all serious computing in all-purpose. This model makes available for fast interactive response, an efficient use of resources, and a straightforward design.

Check your progress 3

1. A workstation server model is a_____.
 - a. single system
 - b. two systems
 - c. multiple system
 - d. all
2. Which model provides processing power on request?
 - a. Processor-pool Model
 - b. workstation server model
 - c. Integrated Hybrid Model
 - d. none
3. The disk in workstations can be used by_____.
 - a. Paging
 - b. Temporary files
 - c. System binaries
 - d. all

1.5 Let Us Sum Up

In this unit we have learned:

- That a Distributed Operating System is a model where applications are running on multiple computers linked by communications.
- It is studied that a workstation model is a basic arrangement where system comprises of workstations which are high end personal computers spread across the building or campus and are joined or connected through high speed LAN.
- A distributed operating system is an extension of the network operating system.
- Queuing systems are helpful as they can be easily modelled analytically.

1.6 Answers for Check Your Progress

Check your progress 1

Answers: (1-a), (2-c), (3-b)

Check your progress 2

Answers: (1-a), (2-b), (3-c), (4-a)

Check your progress 3

Answers: (1-d), (2-a), (3-d)

1.7 Glossary

1. **Structured P2P** - where the nodes are arranged having a particular distributed data structure.
2. **Unstructured P2P** - where the nodes have arbitrarily selected other close nodes.
3. **Hybrid P2P** - where some nodes are presented as special functions in a good organized manner.
4. **Workstation-server Model** - Workstation may be a standalone system or a part of a network.
5. **Processor-pool Model** - Provides processing power on a demand basis.
6. **Integrated Hybrid Model** - Workstations used as processor pools.

1.8 Assignment

Design a Processor-pool Model in your institute.

1.9 Activities

Create an activity on Unstructured P2P.

1.10 Case Study

Is your institute carries Workstation-server Model.

1.11 Further Readings

1. Distributed Systems, Principles and Paradigms by Tanenbaum.
2. Distributed Systems, Concepts and Design by Coulouris, Dollimore, Kindberg.

UNIT 2: MORE ON OPERATING SYSTEM

Unit Structure

- 2.0 Learning Objectives**
- 2.1 Introduction**
- 2.2 Remote Procedure Calls**
- 2.3 Distributed Shared Memory**
- 2.4 UNIX Operating System: Case Studies**
- 2.5 Let Us Sum Up**
- 2.6 Answers For Check Your Progress**
- 2.7 Glossary**
- 2.8 Assignment**
- 2.9 Activities**
- 2.10 Case Study**
- 2.11 Further Readings**

2.0 Learning Objectives

After learning this unit, you will be able to understand:

- Basic of RPC arrangement
- Shared virtual memory
- Concept of Unix Processing
- About Unix Shell

2.1 Introduction

In that context, by allowing the programmer to access and to share “memory objects” without being in charge of their management, virtually shared memory systems want to propose a trade-off between the easy-programming of shared memory machines and the efficiency and scalability of distributed memory systems. We can say that a procedure is an arrangement of closed sequence of instructions which will be controlled through external source. In this case, the data

approximations are travelled in all directions which indicate the flow of control. At last, the procedure call is an invention of a procedure.

2.2 Remote procedure Calls

RPC is an influential method for constructing distributed, client-server based request. It stands on extending the concept of conservative or local process calling, subsequently called as procedure that should not be present in the similar address space as the calling procedure. The two procedures possibly will be on the identical system, or they possibly will be on different systems with a network linking them. By means of RPC, programmers of distributed applications stay away from the details of interface with the network. The transport autonomy of RPC cut off the application from the physical as well as logical fundamentals of the data communications method in addition to allows the application to exercise a mixture of transports.

RPC is equivalent to a function call. Similar to a function call, as soon as an RPC is made, the calling opinion is passed to the remote procedure plus the caller waits for a reply to be come back from the remote process. Figure 2.1 shows the stream of activity that takes place all through an RPC call among two networked systems. The client creates a procedure call with the aim of sending a request to the server more over and will wait. The thread is blocked-up from processing in anticipation of either a reply is received or it timed out. When the appeal arrives, the server calls a dispatch routine with the intention of performing the requested service, furthermore sends the reply to the client. Following the RPC call is ended, the client program goes on. RPC purposely supports network applications.

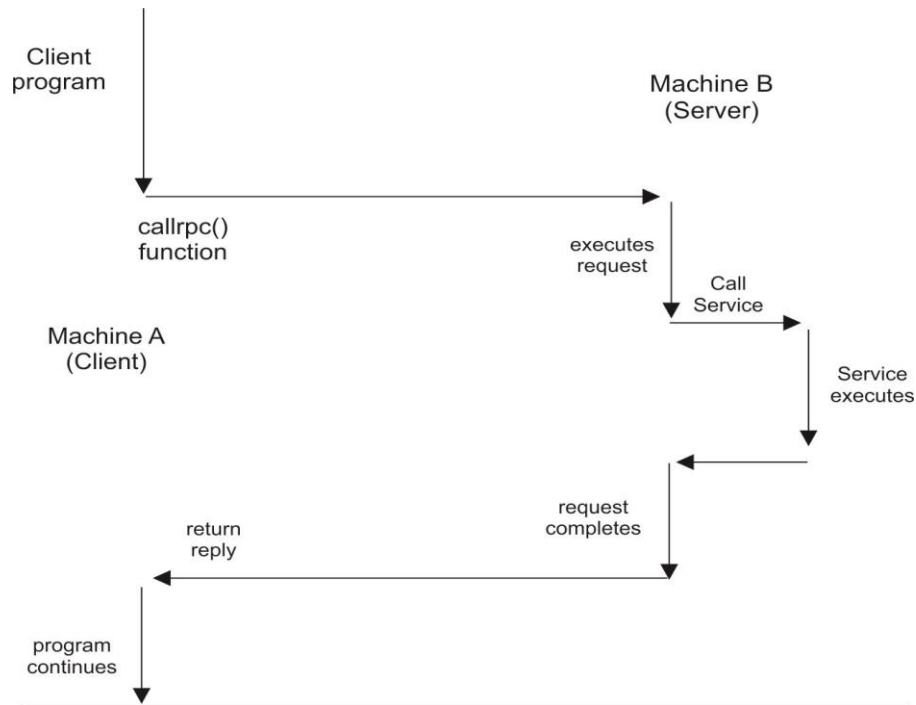


Fig 2.1 Remote procedure Calls

A remote process is exclusively acknowledged by the:-

- program number
- version number
- procedure number

The program number makes out a group of connected remote events, each of which has an exclusive procedure number. A program possibly will consist of one or more versions. Every version comprises of compilation of procedures which are accessible to be called remotely. Version numbers facilitate manifold versions of an RPC protocol to be obtainable at the same time. Every version includes a number of procedures to facilitate remotely. Each procedure has a procedure number.

It is studied that all RPC will arise in the background of a thread which shows sequential amount of control flow through single execution all the time. It is found that the thread is created and handled by certain application code that is present inside an application thread.

The RPC usage makes use of application threads in order to give equal RPCs as well as RPC run time calls. It is found that an RPC client will be gathered by one or more client application threads which applies RPCs.

While calculating remote procedures, RPC server will employ single or many call threads to present RPC run-time system. In the beginning, the server application thread will bring about several simultaneous calls. The single threaded applications will carry out at least single call thread. It is found that an run-time system will produce call threads in server execution background as shown in Fig 2.2.

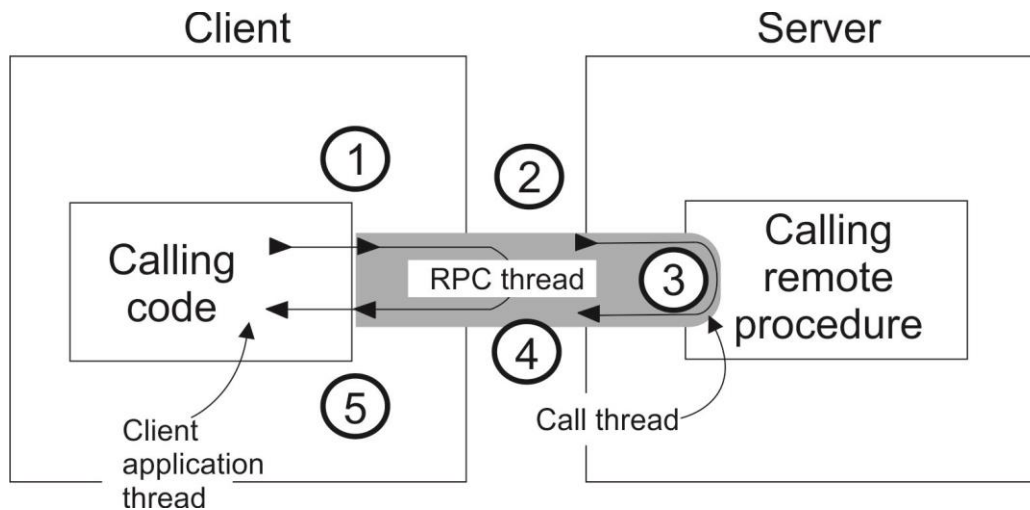


Fig 2.2 RPC arrangement

In the figure, the RPC will get expanded from one corner to corner client along with server execution. When, a client application thread calls a remote process, at that time, it will become the part of rational thread of execution, which will be located as RPC thread. Such types of thread behaves as rational assembly which carry certain portions of RPC that gets increased from one corner point to another corner point with fixed threads of execution in a network. The working part of RPC thread when in execution stage will cover:

- Starting of RPC thread in a client process since client application thread produces RPC.
- Expansion of RPC thread across the diagonals in network to server.
- Movement of RPC thread into call thread in case of remote procedure.
- Execution of remote procedure, where a call thread is the part of RPC thread.
- Giving the network to client by RPC thread.
- Precedence of call result and client application thread by RPC on arrival of RPC thread.

During the lack of RPC, the cancellation and location of thread with local working belong to similar framework. In the presence of RPC, the system will have a remote procedure where both local as well as fraction of cancelled thread's will work.

Check your progress 1

1. Remote procedure calls is a method for creating_____.
 - a. distributed request
 - b. client-server request
 - c. Both a and b
 - d. None
2. Remote procedure calls is equivalent to_____ call.
 - a. data
 - b. information
 - c. message
 - d. function
3. In RPC, the remote process is done by_____.
 - a. program number
 - b. version number
 - c. procedure number
 - d. all
4. RPC run-time system generates _____threads.
 - a. call
 - b. information
 - c. message
 - d. function

2.3 Distributed shared memory

A distributed shared memory is a method permitting the end-users procedure to right to use shared data with no inter-process communications. We can say that, the objective of DSM arrangement is to create inter-process communications see-through to end-users. It is implemented with the help of both hardware and software. With the idea of programming, two approaches have been studied:-

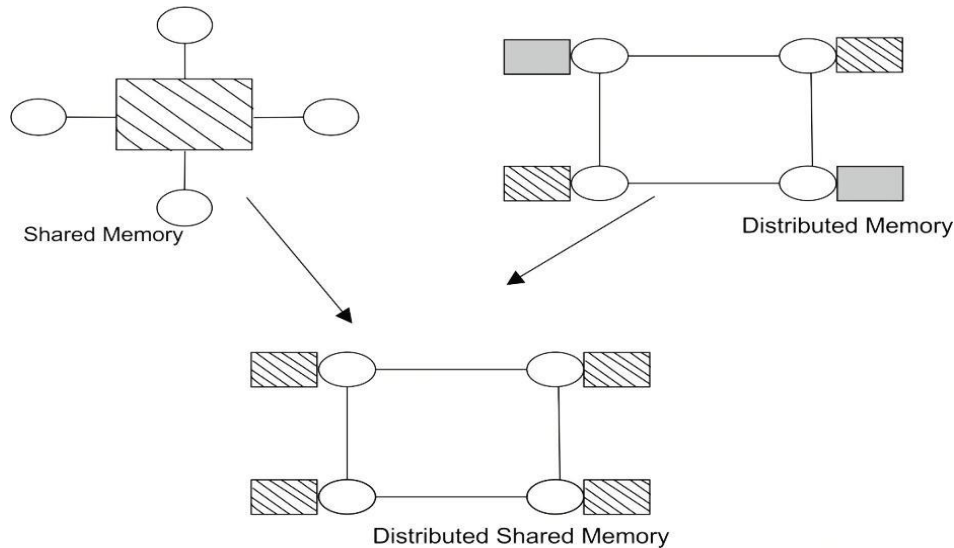


Fig 2.3 distributed shared memory

Shared virtual memory

This idea is very analogous to the well-known thought of paged virtual memory put into practice in mono-processor systems. The essential idea is to set all distributed memories jointly into a single broad address space. Such type of virtual memory has certain drawbacks as they do not allow to take into account the semantics of joint data as well as in this the data granularity is randomly fixed to several page size whatever the type as well as the actual size of the shared data might be. The programmer has no way to give information about such data.

Object DSM

In such class, the joint data such as objects that are variables having an access functions. In his purpose, the user has merely to describe which data (objects) are shared. The complete management of the collective objects (creation, access, modification) is handled by the DSM system. In contradictory of SVM systems which work at operating system layer, objects DSM systems in fact propose a programming model option to the classical message-passing.

In any case, executing a DSM system involves address problems of data position, data access, sharing and locking of data, data coherence. These problems are not definite to parallelism except they have connections through distributed or replicated databases management systems, networks, uniprocessor operating systems and distributed systems.

Methods of Achieving DSM

Hardware - It uses special network interfaces as well as cache coherence circuits.

Software - It modifies OS kernel additionally add a software layer between the operating system as well as in application.

Software DSM Implementation

- Page based –It uses system’s virtual memory.
- Shared variable approach- It uses routines to access shared variables.
- Object based- It shares data within collection of objects and gives access to share data by object oriented discipline.

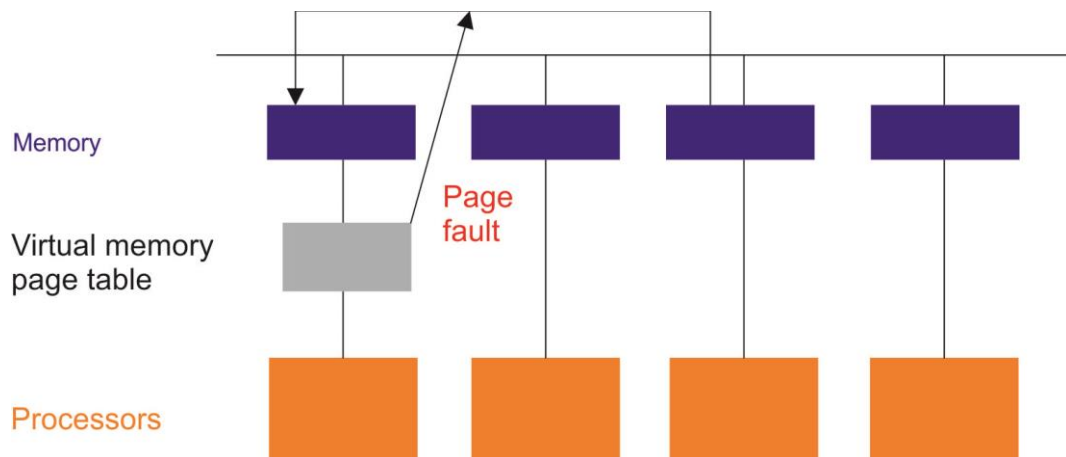


Fig 2.4 objects DSM systems

Advantages of DSM

- It is system scalable.
- It hides message passing and do not open definite sending messages among processes.
- It can use easy extensions to sequential programming.
- It handles difficult and big data bases without replication or sending data to processes.

Disadvantages of DSM

- It may acquire a performance penalty.
- It should be provided for protection against immediate admission to shared data.
- It has small programmer control over real messages individually generated.
- It has a problem in performance of particular that can be difficult.

Consistency Models used on DSM Systems

Release Consistency

An extension of weak consistency shown in fig 2.5 in which the synchronization operations have been specified-

- acquire operation - used before a shared variable or variables are to be read.
- release operation - used after the shared variable or variables have been altered (written) and allows another process to access to the variable(s).

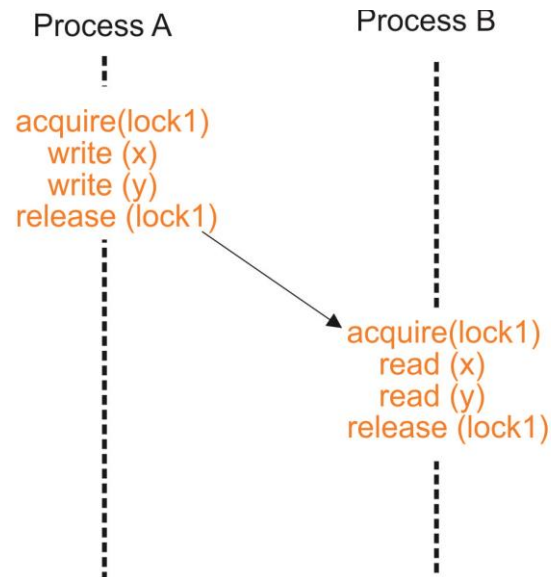


Fig 2.5 release consistency

Typically acquire is done with a lock operation and release by an unlock operation (although not necessarily).

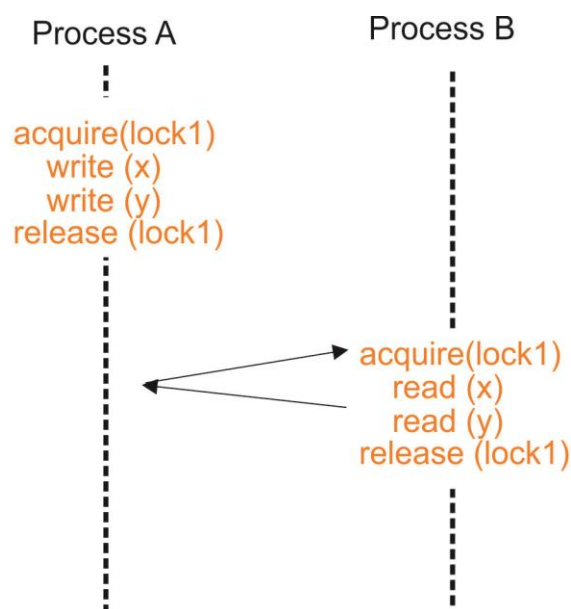


Fig 2.6 Lazy release consistency

Check your progress 2

1. Execution of DSM system involves_____.
 - a. address problems of data position
 - b. data access
 - c. sharing and locking of data_____.
 - d. all
2. Which is an advantage of DSM?
 - a. Acquires performance penalty.
 - b. Should be provided for protection against immediate admission.
 - c. Small programmer control over real messages.
 - d. It is system scalable.

2.4 UNIX Operating System: Case Studies

Shell

- Shell Serves as an interface among command language user and OS.
- The Shell is user interface and comes in many forms.
- User allowed to enter input when prompted (\$ or %)

Unix supports all shells which is running at the same time. The required shell gets loaded at the login position where every alteration can be done by the user. The UNIX command syntax will be executable_file [-options] arguments.

The shell runs a command interpretation loop:-

- Accept command
- Read command
- Process command
- Execute command

Performing a command means developing a child process which is seen working in another shell with the help of forking. In this, the parent process will halt till the child process terminates initially early re-entering command interpretation loop.

It is found that the programs will work in the background by using suffix command line entry which is applied by ampersand (&). The result of this is that the parent will not wait for child process to get completed.

The Processing Environment

Input and Output

During the working of UNIX, there exist three files for a particular process:

- STDIN - Standard input (attached to keyboard)
- STDOUT - Standard output (attached to terminal)
- STDERR - Standard Error (attached to terminal)

As in UNIX, I/O devices results in a unique types of file systems, the STDIO will easily be redirected to different devices along with files who > list _of _users

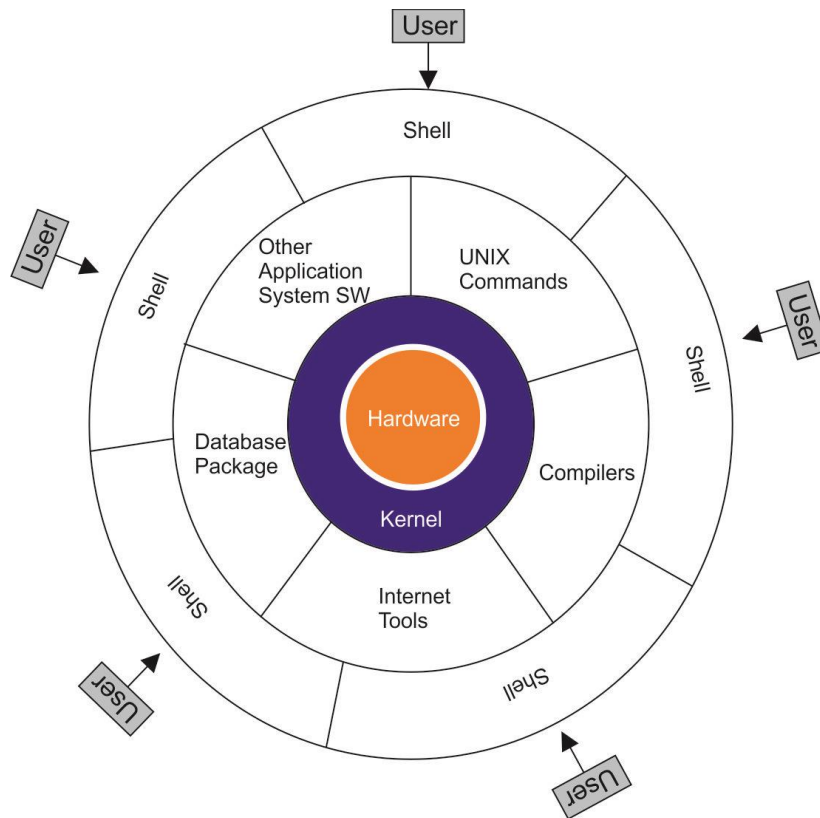


Fig 2.7 Structure of Unix

The Kernel

This is the middle portion of an OS which will give system services to application programs and shell.

Kernel manages certain memory, I/O and Timer processes.

Different process contains different address space for protection.

In Unix sharing of text region is done and changes in process occurs as per environment by calls.

The File System

In UNIX, HDS file is along with root is applied at origin.

The directory in UNIX file has file names and i-nodes.

In this, the subdirectories shows entry of file.

In Unix, directories cannot be directly changed but will change with the help of operating system.

File System is a type of data structure which is present of the disk.

File system contains super block, an arrangement of i-nodes, actual file data blocks as well as free blocks.

In Unix, space allocation is done in case of fixed size blocks.

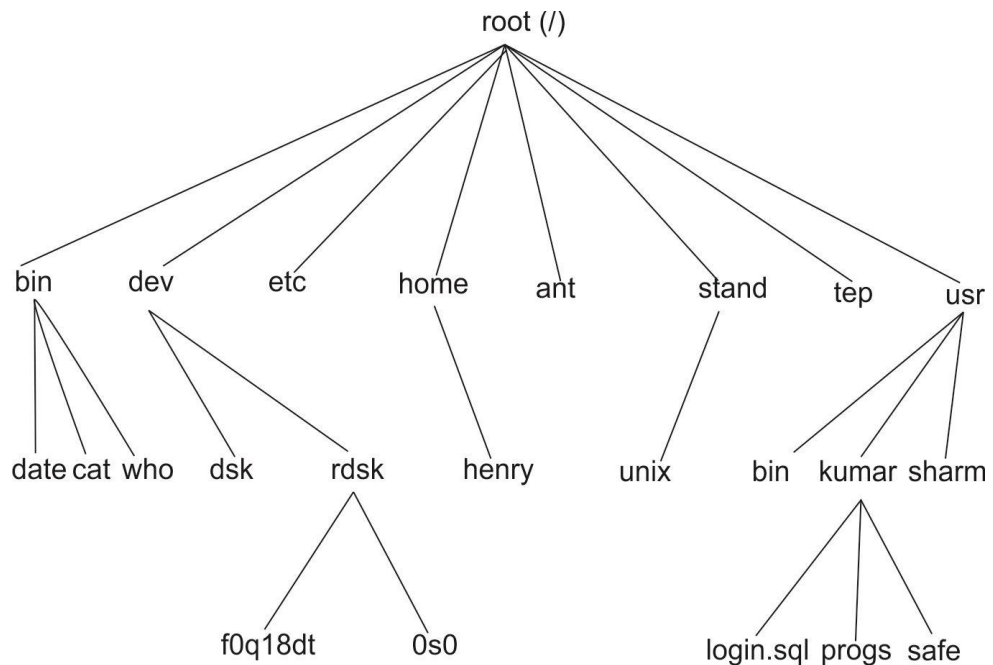


Fig 2.8 File Structure

The i-node contains

- The file owner's user-id and group-id.
- Protection bit for owner, group, and world.
- The block locator.
- File size.
- Accounting information.
- Number of links to the file.
- File type.

The Block Locator

Consists of 13 fields

- First 10 fields point directly to first 10 file blocks.
- 11th field is an indirect block address.
- 12th field is a double-indirect block address.
- 13th field is a triple-indirect block address.

Permissions

- In UNIX, the file and directory contains 3 sets of permission bits.
- These directories are allowed for owner, group and world.
- The system files are copyright by root, wizard, or super user.
- The access of root is restricted to the owners of files.

Setuid

- To modify or change your password, you have to alter/etc/passwd file.
- The root has copyright for passwd command that can be executed by permission to access.
- Setuid is bit which on application with executable file will give similar privileges to user.
- It is similar with many operating system commands.

Process Management

Description of Process Management in SunOS.

Scheduling

- Priority-based pre-emptive Scheduling. Priorities in range -20 to 20.
- Priorities for runnable processes are recomputed every second.
- Allow for ageing, but also increases or decreases process priority based on past behavior.
- I/O – bound processes receive better service.
- CPU – bound processes do not suffer indefinite postponement because the algorithm forgets 90% CPU use in $5 \cdot n$ sec. (where n is the average number of runnable process in the past 60 seconds).

Signals

- Signals are software equivalents to hardware interrupts used to inform processes asynchronously of the occurrence of an event.

Inter-process Communication

- UNIX System V uses semaphores to control access to shared resources.
- For processes to exchange data or communicate, pipes are used.
- A pipe is a unidirectional channel between 2 processes.
- UNIX automatically provides buffering scheduling services and,
- Synchronization to processes in pipe line.

Timers

- There are three interval timers associated to each process.
- Initially it will count to 0 and then produce a signal.
- The initial signal will work regularly, while second work during the process that works with process code.
- The last will work at the time when process works out process code.

Memory Management

- Address Mapping (Virtual Storage) – Paged MMS.
- Virtual address V is dynamically translated to real address (P,D) .
- Direct Mapping is used, with the page map held in a high-speed RAM cache.

- Each page map entry contains a modified bit and accessed bit, a valid bit (if the page is resident in PM) and protection bits.
- The system maintains 8 page maps – 1 for the kernel and 7 for processes.
- 2 context register are used – one points to the running process page map and the other to the kernel's page map.
- The replacement strategy replaces the page that has not been active for longest (LRU).

Paging

- Sum OS maintains 2 data structures to control paging.
- The free list contains empty page frames.
- The free list contains an ordered list of all allocated page frames (except for the kernel).
- The pager ensures that there is always free space in memory.
- When a page is swapped out (not necessarily replaced the system judges whether the page is likely to be used again).
- If the page contains a text region, the page is added to bottom of the free list, otherwise it is added to the top.
- When a page fault occurs, if the page is still in the free list it is reclaimed.

I/O Data

- Data is addressed as byte stream.
- UNIX not put any sort of structure on data, but applications do.
- Manipulation of data appears in any direction.

Devices

- In UNIX, device is unique file type.
- These unique files contain a protection bits to stop read/write operations.
- Certain sensitive devices are not allowed to root while other users uses system calls having setuid bit set.

Generic Unix Command

Command	Function
Date	Used to display the current date and time.
Date + % D	Display date only
Date + % T	Display time only
Date + % Y	Display Year part of the date.
Date + % H	Display the hour part of the time.
Cal	Calendar of the current month
Cal year	Displays calendar for all months for the specified year
Call month year	Displays calendar for the specified month of the year.
Who	Login detail of all user as their IP, Terminal NO, User Name,
Who am I	Used to display the login detail of the under.
tty	Used to display the terminal name
uname	Display the operating system
uname-r	Show version number of the OS (kernel).
uname-n	Display domain name of the server
echo`txt`	Display the given text on the screen
echo\$HOME	Display the user's home directory
Bc	Basic calculator. Press Ctrl+d to quit.
lp file queue	Allows the user to spool a job along with other in a print
mancmdname	Manual for the given command Press q to exit

History	To display the command used by the user since log on.
Exit / out	Exit form a process. If shell is the only process then logs

Check your progress 3

1. Which is not a shell run command?

- | | |
|-------------------|--------------------|
| a. delete command | c. read command |
| b. accept command | d. process command |

2. The processing files of UNIX are_____.

- | | |
|-----------|-----------|
| a. STDIN | c. STDERR |
| b. STDOUT | d. all |

3. The central part of Unix OS is_____.

- | | |
|-----------|--------------|
| a. kernel | c. compilers |
| b. shell | d. database |

4. In Unix, i-node contains_____.

- | | |
|-------------------------|------------------------------|
| a. file owner's user-id | c. protection bits for owner |
| b. group-id | d. all |

2.5 Let Us Sum Up

In this unit we have learned:

- That a procedure is an enclosed series of instructions which is introduced from and returns the control to an external source.
- It is studied that data approximations can be travelled in all directions along with the flow of control.
- RPC is equivalent to a function call
- A thread is a single chronological flow of control by way of one point of execution at whichever moment.

- A thread formed as well as managed by application code is an application thread.
- Distributed shared memory is a method permitting the end-users procedure to right to use shared data with no inter-process communications.

2.6 Answers for Check Your Progress

Check your progress 1

Answers: (1-c), (2-d), (3-d), (4-a)

Check your progress 2

Answers: (1-d), (2-d)

Check your progress 3

Answers: (1-a), (2-d), (3-a), (4-d)

2.7 Glossary

1. **Kernel** - It is the central part of OS which provides system services to application programs and the shell.
2. **File System** - It is a data structure that is resident on disk.

2.8 Assignment

Explain the Consistency Models used on DSM Systems.

2.9 Activities

Write a DSM system in C++ using MPI for the underlying message-passing and process communication.

2.10 Case Study

Compile and run the remote directory example `rls.cand` run both client and server on the network.

2.11 Further Readings

1. Distributed Systems, Principles and Paradigms by Tanenbaum.
2. Distributed Systems, Concepts and Design by Coulouris, Dollimore, Kindberg.

Block Summary

In this block, the student will understand about the basic of distributed operating system and its modelling techniques. The block gives an idea on architecture and distribution of distributed operating system with study about their characteristics. The examples related to concept of distributed operating system layout and working characteristics are also discussed.

In this block, the student will understand about the basic of remote procedure calls and its techniques. The concept related to distribute shared memory and Unix operating system is also detailed. The student will be demonstrated practically about Unix architecture.

Block Assignment

Short Answer Questions

1. What is Workstation Model?
2. What are i-nodes in Unix?
3. What are the advantages and drawbacks of Distributed Operating System Model?
4. What are shell run commands?
5. Explain Remote procedure Calls?

Long Answer Questions

1. Explain Distributed Computing System Models?
2. What are the different methods of achieving DSM?
3. Explain Workstation Server Model?

Enrolment No.

1. How many hours did you need for studying the units?

Unit No	1	2	3	4
Nos of Hrs				

2. Please give your reactions to the following items based on your reading of the block:

Items	Excellent	Very Good	Good	Poor	Give specific example if any
Presentation Quality	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____ _____
Language and Style	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____ _____
Illustration used (Diagram, tables etc)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____ _____
Conceptual Clarity	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____ _____
Check your progress Quest	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____ _____
Feed back to CYP Question	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____ _____

3. Any Other Comments

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....



“

*Education is something
which ought to be
brought within
the reach of every one.*

”

- Dr. B. R. Ambedkar



Dr. Babasaheb Ambedkar Open University
Jyotirmay' Parisar, Opp. Shri Balaji Temple, Sarkhej-Gandhinagar Highway, Chharodi,
Ahmedabad-382 481.