UNIT 1: FUNDAMENTAL OF COMPUTERS

1.1. Introduction

Computer is the most powerful tool man has ever created. Computers have made a great impact on our everyday life. Today, computer technology has permeated every sphere of existence of modern human being. With the growing information needs the computer has become one of the vital components for the survival of the business houses.

Information technology (IT) is the acquisition, processing, storage and dissemination of vocal, pictorial, textual and numerical information by a microelectronics-based combination of computing and telecommunications. It is the use of computers and software to manage information related to any organization or entity.

1.2 Application of Information Technology

Now a day computer is being used almost in every aspects of life. Every company, small or large, government offices, educational institutions are now directly or indirectly dependent on computers mainly for information processing. Computer based railway and airway reservation system is a common example of computer application. Computer system is helping in the efficient management of the banking sector, hospital records, and payroll records and so on. Some of the areas where computers are being used mostly can be listed as below:

Science:

Scientists are using computers to carry out their research works based on complex computations because of computer's fast speed and the accuracy.

Education:

In schools and colleges, to make education much more interesting, computers are used now days. Computer Aided Education (CAE) and Computer Based Training (CBT) packages are making learning much more interactive.

Health and Medicine:

Starting from diagnosing the illness to monitoring a patient's status during a surgery , in pathological analysis ,in CAT scans or MRI scans etc. , doctors are using computers. Some special purpose computers are available which can even be operated within the human body.

Engineering:

Engineers and architects are using computers in designing machineries, drawing design layouts. Architects can object that can be viewed from all the three dimensions

by using techniques like virtual reality. In manufacturing industries, using computerized robotic arms hazardous jobs can be performed. The packages like Computer Aided Designing (CAD), Computer Aided Manufacturing (CAM) and so on are used in designing the product, ordering the parts and planning production.

Entertainment:

With the use of multimedia facilities, computers are now greatly used in entertainment industry. Computers are used to control and bring special effects on image and sound.

Communication:

Computer network and finally the Internet has brought a drastic change in communication system. Through E-mail or Electronic mail, it is possible to send messages and reports very fast from one person to another or a group of persons with the aid of computers and telephone lines.

Business and Banking

In business sectors, currently computers are used for real time applications such as sales counter, on line booking etc. Other applications are business forecasting, order report generation, pay bill generation, personal record keeping and so on. In the field of banking and finance also computers have many applications.

1.3 Number System

We are familiar with the decimal number system which is used in our day-to-day work. Ten digits are used to four decimal numbers. To represent these decimal digits, ten separate symbols 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9 are used. But a digital computer stores, understands and manipulates information composed of any zeros and ones. So, each decimal digit, letters, symbols etc. written by the programmer (an user) are converted to binary codes in the form of 0's and 1's within the computer. The no. system is divided into some categories according to the base (or radix) of the system as binary octal and hexadecimal. If a number system of base r is a system, then the system has r distinct symbols for r digits. The knowledge of the number system is essential to understand the operation of a computer.

1.3.1 Decimal Number System

Decimal no. system have ten digits represented by 0,1,2,3,4,5,6,7,8 and 9. So, the base or radix of such system is 10.

In this system the successive position to the left of the decimal point represent units, tens, hundreds, thousands etc. For example, if we consider a decimal number 257, then the digit representations are

2	5	7
Hundred Position	Tens Position	Units Position

The weight of each digit of a number depends on its relative position within the number.

Example 1.1:

The weight of each digit of the decimal no. 6472

$6472 = 6000+400+70+2$

$= 6X10^3+4X10^2+7X10^1+2X10^0$

The weight of digits from right hand side are-

Weight of 1st digit $= 2 X 10^0$

Weight of 2nd digit $= 7 X 10^1$

Weight of 3rd digit $= 4 X 10^2$

Weight of 4th digit $= 6 X 10^3$

The above expressions can be written in general forms as the weight of nth digit of the number from the right hand side

$= n^{th}$ *digit* $X\ 10^{n-1}$

$= n^{th}$ *digit* X *(base)*$^{n-1}$

The no. system in which the weight of each digit depends on its relative position within the number is called positional number system. The above form of general expression is true only for positional number system.

1.3.2 Binary Number System

Only two digits 0 and 1 are used to represent a binary number system. So the base or radix of binary system is two (2). The digits 0 and 1 are called bits (Binary Digits). In this number system the value of the digit will be two times greater than its predecessor. Thus the value of the places are-

<-- <-- 32 <-- 16 <--8 <--4 <--2 <--1

The weight of each binary bit depends on its relative position within the number. It is explained by the following example--

Example:

The weight of bits of the binary number 10110 is-

$= 1X2^4+0X2^3+1X2^2+1X2^1+0X2^0$

= 16+0+4+2+0 = 22(decimal number)

The weight of each bit of a binary no. depends on its relative pointer within the no. and explained from right hand side

Weight of 1st bit = 1st bit X 2^0

Weight of 2nd bit = 2nd bit X 2^1

.....

.....

and so on.

The weight of the nth bit of the number from right hand side

$=n^{th}$ *bit X* 2^{n-1}

$=n^{th}$ *bit X (Base)*$^{n-1}$

It is seen that this rule for a binary number is same as that for a decimal number system. The above rule holds good for any other positioned number system. The weight of a digit in any positioned number system depends on its relative positon within the number and the base of the number system.

Table 1.1 shows the binary equivalent numbers for decimal digits .

Table: Binary equivalent of decimal numbers

Decimal Number	*Equivalent Binary Number*
0	*0*
1	*1*
2	*10*
3	*11*
4	*100*
5	*101*
6	*110*
7	*111*
8	*1000*
9	*1001*

1.3.3 Octal Number System

A commonly used positional number system is the Octal Number System. This system has eight (8) digit representations as 0,1,2,3,4,5,6 and 7. The base or radix of this system is 8. The values increase from left to right as 1, 8, 64, 512, 4096 etc. The decimal value 8 is represented in octal as 10, 9 as 11, 10 as 12 and so on. As $8=2^3$, an octal number is represented by a group of three binary bits. For example 3 is represented as 011, 4 as 100 etc.

Table: The octal number and their binary representations.

Decimal Number	*Octal Number*	*Binary Coded Octal No.*
0	*0*	*000*
1	*1*	*001*
2	*2*	*010*
3	*3*	*011*

4	*4*	*100*
5	*5*	*101*
6	*6*	*110*
7	*7*	*111*
8	*10*	*100 000*
10	*12*	*001 010*
15	*17*	*001 111*
16	*20*	*010 000*

1.3.4 Hexadecimal Number System

The hexadecimal number system is now extensively used in computer industry. Its base (or radix) is 16, ie. 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F. The hexadecimal numbers are used to represent binary numbers because of case of conversion and compactness. As $16=2^4$, hexadecimal number is represented by a group of four binary bits. For example, 5 is represented by 0101. Table 2.3 shows the binary equivalent of a decimal number and its hexadecimal representation.

Table: Hexadecimal number and their Binary representation

Decimal No.	***Hexadecimal No.***	***Binary coded Hex. No***
0	*0*	*0000*
1	*1*	*0001*
2	*2*	*0010*
3	*3*	*0011*
4	*4*	*0100*
5	*5*	*0101*
6	*6*	*0110*
7	*7*	*0111*
8	*8*	*1000*

9	*9*	*1001*
10	*A*	*1010*
11	*B*	*1011*
12	*C*	*1100*
13	*D*	*1101*
14	*E*	*1110*
15	*F*	*1111*

1.4 Number System Conversion

As the computer uses different number systems, there is a process of converting generally used decimal number systems to other number systems and vice-versa.

1.4.1 Binary to Decimal Conversion

To convert a binary number to its decimal equivalent we use the following expression.The weight of the n^{th} bit of the number from right hand side

$$=n^{th} \text{ bit } X\ 2^{n-1}$$

First we mark the bit position and then we give the weight of each bit of the number depending on its position. The sum of the weight of all bits gives the equivalent number.

Example: *Convert binary $(110100)_2$ to its decimal equivalent*

Solution:

$$(110100)_2=1X2^5+1X2^4+0X2^3+1X2^2+0X2^1+0X2^0$$

$$=32+16+0+4+0+0$$

$$=(52)_{10}$$

$$(110100)_2=(52)_{10}$$

Example: *Converting binary fraction* $(111011.101)_2$ *to its equivalent decimal fraction.*

Solution:

$(111011.101)_2 = (1X2^5+1X2^4+1X2^3+0X2^2+1X2^1+1X2^0)+(1X\ 2^{-1}+0X2^{-2}+1X2^{-3})$

$= (32+16+8+0+2+1)+(0.5+0+0.125)$

$= (59.625)_{10}$

$(111011.101)_2 = (59.625)_{10}$

1.4.2 Decimal to Binary Conversion

There are different methods used to convert decimal number to binary number. The most common method is, repeated-division method. In this method, the number is successively divided by 2 and its remainders 0's abd 1's are recorded. The final binary result is obtained by assembling the remainders in reverse order to obtain the binary equivalent of the decimal number. In this case, the last remainder will be the most significant bit (MSB).

Example 1.5 Convert $(75)_{10}$ *to its binary equivalent*

2 |75	Remainder	
2|37	1	
2|18	1	
2|9	0	Read in
2|4	1	reverse order
2|2	0	
1	0	

So, $(75)_{10}=(1001011)_2$

The method to convert the fraction decimal number to its binary equivalent, is repeatedly multiply the fraction part by 2 and count the most significant bits in the order they appear.

Example: *Convert decimal fraction $(12.75)_{10}$ to its equivalent binary fraction.*

2|12	*Remainder*	*MSB*	*.75*	
2|6	*0*		*X 2*	
2|3	*0*		*1.50*	*Read*
1	*1*		*X 2*	*the MSB*
			1.00	*bits.*

So, $(12)_{10} = (1100)_2$ *and* $(.75)_{10} = (.11)_2$

Now, $(12.75)_{10} = (1100.11)_2$

1.4.3 Octal to Decimal Conversion

The method of converting octal numbers to decimal numbers is simple. The decimal equivalent of an octal number is the sum of the numbers multiplied by their corresponding weights.

Example: *Find decimal equivalent of octal number* $(153)_8$

Solution: $1X8^2 + 1X8^1 + 1X8^0 = 64 + 40 + 3 = 107$

$So, (153)_8 = (107)_{10}$

Example: *Find decimal equivalent of octal number* $(123.21)_8$

Solution: $(1X8^2 + 2X8^1 + 3X8^0) + (2X8^{-1} + 1X8^{-2})$

$= (64 + 16 + 3) + (0.25 + 0.0156) = 83.2656$

$So, (123.21)_8 = (83.2656)_{10}$

1.4.4 Decimal to Octal Conversion

The procedure for conversion of decimal numbers to octal numbers is exactly similar to the conversion of decimal number to binary numbers except replacing 2 by 8.

Example: *Find the octal equivalent of decimal* $(3229)_{10}$

Solution: *Remainders*

8 | 3229		
8 | 403	5	read in
8 | 50	3	reverse
8 | 6	2	order
0	6	

So, $(3229)_{10}=(6235)_8$

Example: *Find the octal equivalent of* $(.123)_{10}$

Solution: *Octal equivalent of fractional part of a decimal number as follows:*

8 X 0.123 = 0.984	0	
8 X 0.984 = 7.872	7	read in
8 X 0.872 = 6.976	6	forward order
8 X 0.976 = 7.808	7	

Read the integer to the left of the decimal point

The calculation can be terminated after a few steps if the fractional part does not become zero.

The octal equivalent of

$$(0.123)_{10}=(0.0767)_8$$

1.4.5 Hexadecimal to Decimal Conversion

The method of converting hexadecimal numbers to decimal number is simple. The decimal equivalent of an hexadecimal number is the sum of the numbers multiplied by their corresponding weights.

Example: *Find the decimal equivalent of* $(4A83)_{16}$

Solution:

$(4A83)_{16}=(4 \times 16^3) + (10 \times 16^2) + (8 \times 19^{1)} + (3 \times 16^0)$

$=16384+2560+128+3$

$=(19075)_{10}$

$(4A83)_{16}=(19075)_{10}$

Example: *Find the decimal equivalent of* $(53A.0B4)_{16}$

Solution:

$(53A.0B4)16=(5 \times 16^2) + (3 \times 16^1) + (10 \times 16^0) + (0 \times 16^{-1}) + (11 \times 16^{-2}) + (4 \times 16^{-3})$

$=1280+48+10+0+0.04927+0.0009765$

$=(1338.0439)_{10}$

$(53A.0B4)_{16}=(1338.0439)_{10}$

1.4.6 Decimal to Hexadecimal Conversion

To convert a decimal integer number to hexadecimal, successively divide the given decimal number by 16 till the quotient is zero. The last remainder is the MSB (Most Significant Bit). The remainders read from bottom to top give the equivalent hexadecimal integer.

To convert a decimal fraction to hexadecimal, successively multiply the given decimal fraction by 16, till the product is zero or till the required accuracy is obtained, and collect all the integers to the left of decimal point. The first integer is the MSB and the integers read from top to bottom give the hexadecimal fraction.

Example: *Convert decimal* $(1234.675)_{10}$ *to hexadecimal.*

Solution:

1st consider $(1234)_{10}$

	Remainder		
	Decimal	*Hexadecimal*	
16|1234	*2*	*2*	
16|77	*13*	*D*	*read in reverse order*
16|4	*4*	*4*	

$(1234)_{10} = (4D2)_{16}$

Conversion of $(0.675)_{10}$

Hexadecimal conversion of fractional part of a decimal number as follows:

	Decimal	*Hexadecimal*
0.675 X 16 = 10.8	*10*	*A*
0.800 X 16 = 12.8	*12*	*C*
0.800 X 16 = 12.8	*12*	*C*
0.800 X 16 = 12.8	*12*	*C*

$(0.675)_{10} = (0.ACCC)_{16}$

Hence, $(1234.675)_{10} = (4D2.ACCC)_{16}$

If the decimal number is very large, it is tedious to convert the number to binary directly. So it is always advisable to convert the number into hex first, and then convert the hex to binary.

1.5 Error Detection code

The error detection and correction is to add some redundancy (i.e., some extra data) to a message, which receivers can use to check consistency of the delivered message, and to recover data determined to be erroneous.

In Error-detection and correction schemes the transmitter sends the original data, and attaches a fixed number of check bits (or parity data), which are derived from the data bits by some deterministic algorithm. If only error detection is required, a receiver can simply apply

the same algorithm to the received data bits and compare its output with the received check bits; if the values do not match, an error has occurred at some point during the transmission.

Some of the error detection schemes are as follows:

Repetition codes

A repetition code is a coding scheme that repeats the bits across a channel to achieve error-free communication. Given a stream of data to be transmitted, the data is divided into blocks of bits. Each block is transmitted some predetermined number of times. For example, to send the bit pattern "1011", the four-bit block can be repeated three times, thus producing "1011 1011 1011". However, if this twelve-bit pattern was received as "1010 1011 1011" – where the first block is unlike the other two – it can be determined that an error has occurred.

Repetition codes are very inefficient, and can be susceptible to problems if the error occurs in exactly the same place for each group (e.g., "1010 1010 1010" in the previous example would be detected as correct). The advantage of repetition codes is that they are extremely simple, and are in fact used in some transmissions of numbers stations.

Parity bit

A parity bit is a bit that is added to a group of source bits to ensure that the number of set bits (i.e., bits with value 1) in the outcome is even or odd. It is a very simple scheme that can be used to detect single or any other odd number (i.e., three, five, etc.) of errors in the output. An even number of flipped bits will make the parity bit appear correct even though the data is erroneous.

1.6 Memory Representation

You will be strange that - the computer is made up with millions of tiny electric circuits that are not visible to us. The memory elements are also build with these tiny circuitry which are used for storing data. How the data are stored in those electronic circuit? To better understand how a computer stores information let's take a closer look. For every circuit in a computer chip, there are two possibilities:

- *An electric current flows through the circuit or*
- *An electric current does not flow through the circuit.*

When an electric current flows through a circuit, the circuit is on. When no electricity flows, the circuit is off. An "on" circuit is represented by the number one (1) and an off circuit is represented by the number zero (0). The number 1 or 0 is called ***bit****. The word bit comes*

from "binary digit". Each time a computer reads an instruction, it translates that instruction into a series of bits, 1's and 0's. On most computers every character from the keyboard is translated into eight bits, a combination of eight 1's and 0's. Each group of 8 bits is called 1 byte as shown in the following figure. Each group of four bits is called a nibble.

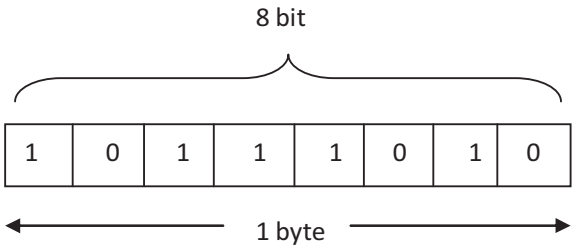

Fig. : Representation of 1-byte

Thus, 8 bits = 1 byte

4 bits = 1 nibble and

1024 byte = 1 kilo byte

We know that, all the quantities, physical or abstract, can be measured in some units. For example, time is measured in second; length is measured in meters and mass in grams. Similarly, the computer memory can be measured in terms of bits, bytes, kilobytes, megabytes, gigabytes etc.

The followings are the measuring units given in terms of bytes :

1 kilobyte (KB) = 1,024 byte

1 megabyte (MB) = 1,048,576 bytes

1 gigabyte (GB) = 1,073,741,824 bytes

1 terabyte = 1,099,511,627,776

1.7 Primary Memory

*The term **memory** usually refers to a form of semiconductor storage and sometimes other forms of fast but temporary storage. Similarly, today the term **storage** more commonly refers to mass storage such as optical discs, forms of magnetic storage like hard disks and other types. All these are slower than memory, but of a more permanent nature. Generally , the computer memory is divided into two category : **primary memory** and **secondary memory**.*

*The term primary memory is used to represent the **memory** and the term secondary memory is used to represent the **storage.***

***Primary memory** is directly accessible to the CPU. The CPU continuously reads instructions stored there and executes them. Any data actively operated on is also stored there in uniform manner. It is a form of semiconductor storage known as random access memory (**RAM**) and sometimes other forms of fast but temporary storage. It is small-sized, light, but quite expensive. This type of memory is divided into the following two types :*

a) *RAM (Random Accress Memory)*

b) *ROM (Read Only Memory)*

1.7.1 Random Access Memory (RAM)

*A computer's **main memory** is often referred to as RAM - an area in the computer system unit that temporarily holds user data, operating system instructions and program instructions. Every time we turn on our computer, a set of operating system instructions is copied from hard disk into RAM. These instructions, which help control basic computer functions, remain in RAM until we turn the computer off.*

RAM features:

a) *RAM is the place which holds the data that has to be processed by the computer's preocessor. Moreover, it hold the operating system and other application programs also.*

b) *RAM is made up of different ICs (Integreted Circuits). They are also called semiconductor memory.*

c) *Each element of RAM is a memory location in which data can be stored. Each location has a **unique address**. Using this address data can be directly retrieved or stored.*

d) *Since RAM must hold both the data to be processed and the instructions for processing, its size or capacity is one of the measure of power of the computer.*

e) *RAM is a volatile memory and loss its data when the computer turned off.*

There are generally two broad categories of random access memory :

i) ***DRAM** memories (Dynamic Random Access Memory)*

ii) ***SRAM** memories (Static Random Access Memory)*

DRAM :

Dynamic RAM is a type of RAM in which data is stored in a storage cell, consisting of transistors and capacitors. A common property of a capacitor is its tendency to get discharge, it means the current goes down in the circuit and finally data cannot holds for long time. To overcome this effect, the DRAM needs to be continuously refreshed with power supply and thus it has one refreshing circuit which r-writes the data in certain time intervals. This refreshing circuit refreshes or rewrites the data several hundred times in a second.

SRAM :

Static RAM is a type of RAM that holds its data as long as power is supplied to the circuit.

The SRAM does not need any refreshing circuit. SRAMs are used for specific applications within the PC. The SRAM has the following properties :

- ***Simplicity :*** *SRAMs don't require external refresh circuitry in order for them to keep their data intact.*
- ***Speed :*** *SRAM is faster than DRAM.*

In contrast, SRAMs have the following weaknesses, compared to DRAMs:

- ***Cost :*** *SRAM is more expensive than DRAM.*
- ***Size :*** *SRAMs take up much more space than DRAMs*

The physical image of a RAM chip is shown in the following figure :

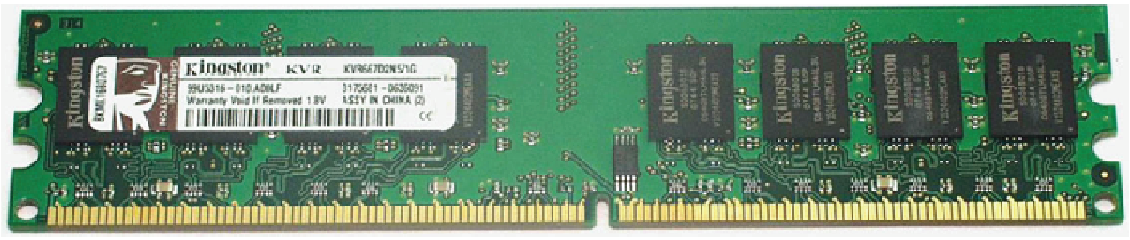

Fig.: A RAM chip

1.7.2 Read Only Memory

One major type of memory that is used in PCs is called read-only memory, (ROM). ROM is the memory that stores the data permanently i.e. the instructions in ROM are

permanent whether the power is on or off. We have no way to change them, unless we remove the ROM chips from the main board and replace them with another set.

Read-only memory is most commonly used to store system-level programs that we want to have available to the PC at all times. For example, the ROM-BIOS (ROM basic input and output services) programs are stored in ROM chips which are used to boot up (start up) the computer.

The following are the different types of ROMs with a description of their relative modifiability:

Programmable ROM (PROM) : *This is a type of ROM that can be programmed using special equipment; it can be written to, but only once.*

Erasable Programmable ROM (EPROM) : *An EPROM is a ROM that can be erased and reprogrammed. Ultraviolet light of a specific frequency is used for erasing the EPROM and allow it to be reprogrammed again.*

Electrically Erasable Programmable ROM (EEPROM) : *The next level of erasability is the EEPROM, which can be erased under software control. This is the most flexible type of ROM, and is now commonly used for holding BIOS programs.*

Another one important types of memory used in a computers are– the ***cache memory and registers****.*

Cache memory*:*

Cache memory is a special type of RAM (static RAM) which is the faster memory used in a computer system in between the CPU and main memory. It means that the accessing speed from the cache memory is faster than the main memory used in computer. So, the use of the cache memory reduces the average access time and helps the processor for fast processing. The Cache memory stores the instructions and data that are frequently needed by the processor for execution. Cache memory chip is a static memory chip. It is much expensive than the main memory. Cache size varies from 32 MB to 128MB or 512MB etc.

Registers:

Registers are the small amount of storage available on the CPU whose contents can be accessed more quickly than any other storage available in a computer system. Registers are the fastest memory element and thus provide the fastest way for a CPU to access data. Registers are normally measured by the number of bits they can hold, for example, an "8-bit register" or a "32-bit register". Registers can be classified as general purpose registers and special purpose registers.

1.8 Secondary Memory

Secondary memory or storage provides the facility of storing information and programs permanently. It differs from primary memory in that it is not directly accessible by the CPU. The computer usually uses its input/output channels to access secondary storage and transfers desired data using intermediate area in primary memory. Secondary storage does not lose the data when the device is kept out of power. It is non-volatile. It is less expensive than primary memory.

In modern computers, hard disks are usually used as secondary storage. Some other examples of secondary storage devices are: optical disk, floppy disks, magnetic tape, Paper tape, Punch Cards, standalone RAM disks, flash memory (e.g. USB sticks or keys), and Zip drives etc.

1.8.1 Magnetic Tape

Magnetic tapes are the plastic tapes with magnetic coating that are used for storing the data, such as text, audio or video. Magnetic tapes are similar to the normal recording tapes, which are used for audio and video recording. Magnetic tape is a medium for magnetic recording, made of a thin magnetizable coating on a long, narrow strip of plastic. A device that stores computer data on magnetic tape is called a tape drive. The data stored on the magnetic tapes can be accessed using the sequential access method. A magnetic tape is show:

Fig.: A magnetic tape

Major advantages of magnetic tapes are:

- *Low cost*
- *Large storage capacity*
- *Easily transportable*

- *Easy to handle and store*

1.8.2 Magnetic Disks

Magnetic disks are widely used popular secondary storage medium. A magnetic disk consists of plastic or metal circular plate or platter. These platters are coated with magnetic oxide layer. The data stored in magnetic disks can be accessed directly. They provide high storage capacity and reliability. Data is represented as magnetized spots on a disk .The presence of a spot represents a 1 and the absence of a magnetized spot represents a 0. The information is accessed using one or more read/write ***heads****.*

A magnetic disk typically contains several rotating disks, or platters. There is one head for each surface, and all the heads move together. The disk rotates at around 3600 rpm (rotation per minutes) with the heads floating microscopic distances above the surfaces. Data on the magnetic disks is organized in a concentric set of circles, called ***tracks****. Each track is divided into* ***sectors****. The following diagram shows the tracks and sectors of a platter.*

We have already mentioned that magnetic disk uses a single read/ write head for each disk surface. The read/write head may be movable or fixed. The following figure shows a movable read/ write heads for each surface:

There are two popular types of magnetic disks:

- *Hard disks*
- *Floppy disks*

Major advantages of magnetic disks are:

- *High storage capacity*
- *Easy, direct access to data*
- *Easily moveable from one place to another*
- *Better data transfer rate as compared to magnetic tapes*
- *Low cost compared to RAM*
- *Less prone to corruption of data as compared to tapes*

1.8.3 Optical Media

The optical media are used for the same purpose as the magnetic storage system. We have already found that in case of magnetic media there is a coating of magnetic material on the surface of the medium. But in case of optical medium laser light is used to retrieve as well as record data. Examples of optical storage mediums are given below:

- *Compact Disk - Read only memory (CD-ROM)*
- *Digital Video Disk (DVD)*
- *Compact Disk - Recordable (CD-R)*
- *Compact Disk - Rewritable (CD-RW)*
- *Digital Video Disk - Recordable (DVD-R)*
- *Digital Video Disk - Rewritable (DVD-RW)*

Like other storage media, optical storage are non-volatile in nature and also they are more reliable as compared to the magnetic medium.

CD-ROM:

We are already familiar with CD-ROM disk is a shiny, silver color metal disk usually of12cm diameter. It is made of polycarbonate plastic and thin layer of pure aluminum is applied to make the surface reflective. For some good quality disks, gold layer is used. A thin layer of lacquer protects it. The surface of an optical disk consists of pits and land. The information is read from pits and lands, like 1s and 0s. It is changed into binary so computers can read it. An optical reader reads the patterns of pits that stands for bytes. One CD can hold 650MB of data or 300,000 pages of text. Most CDs are read only, which means you cannot save data to the disk. This device is usually not used as a primary storage device for data.

An optical disk is mounted on an optical disk drive for reading/writing of information on it. An optical disk drive contains all the mechanical, electrical and electronic components for holding an optical disk and for reading/writing of information on it. That is, it contains the tray on which the disk is kept, read/write ***laser beams assembly****, and the motor to rotate the disk. Access time for optical disks are in the range of 100 to 300 milliseconds.*

Advantages of CD-ROM are –

a) Storage capacity is much more in optical disks.

b) Multiple copies of the disk along with its contents can be made inexpensively.

c) They are transportable from one computer to another very easily.

DVD Disks:

The success of CD technology and the continuing quest for greater storage capability has led to the development of DVD. ***DVD****, also known as* ***Digital Versatile Disk*** *or* ***Digital Video Disk****, is an optical disk storage media format, and was invented and developed by Sony, and Philips in 1995. Its main uses are video and data storage. DVDs are of the same size as compact disks (CDs), but can store more than six times of data as that of a CD.*

Variations of the term DVD often indicate the way data is stored on the disks : DVD-ROM (read only memory) has data that can only be read and not written; DVD-R and DVD+R (recordable) can record data only once, and then function as a DVD-ROM; DVD-RW (re-writable), DVD+RW, and DVD-RAM (random access memory) can all record and erase data multiple times.

RAID TECHNOLOGY: *RAID technology was first defined by a group of computer scientists at the University of California at Berkeley in 1987. The scientists studied the possibility of using two or more disks to appear as a single device to the host system. RAID is a method of combining several hard disk drives into one logical unit (two or more disks grouped together to appear as a single device to the host system). RAID technology was developed to address the fault-tolerance and performance limitations of conventional disk storage. It can offer fault tolerance and higher throughput levels than a single hard drive or group of independent hard drives.*

FLASH MEMORY: *Flash memory is a non-volatile computer storage chip that can be electrically erased and reprogrammed. Flash drives and pen drives are USB storage devices based on flash memory. It is primarily used in memory cards, USB flash drives, MP3 players and solid-state drives for general storage and transfer of data between computers and other digital products.*

1.9 Suggested Readings

1. *Computer System Architecture, M.M. Mano, PHI*
2. *Digital logic and Computer Design, M.M. Mano, PHI*
3. *Computer Fundamentals Architecture and Organization, B. Ram, New Age International (P) Ltd.*

4. *Digital Techniques, P.H. Talukdar*

1.10 Terminal Question

1. *What is Computer Memory?*
2. *How Covert Binary number to Decimal Number?*
3. *What is the different between RAM and ROM?*
4. *What is Mainframe Computer?*
5. *What is magnetic disk and its Size?*
6. *What is a RAM? What are its two types? Differentiate between them.*
7. *What is a ROM? Why is it called so? List differenet types of ROMs and their key features.*
8. *What is a cache memory? How is it different from a primary memory?*
9. *Differentiate between*
    a. *PROM and EPROM.*
    b. *Static RAM and Dynamic RAM*
10. *Write down the difference between a magnetic tape and a magnetic disk.*
11. *Write short note :*
    a. *Primary memory*
    b. *Secondary memory*
    c. *Magnetic tape*
    d. *Magnetic disk*
    e. *Compact disk*

***

Unit 2: Operating Systems

2.1 Introduction

Operating System is an important component of a computer system.

The primary objective of an operating system is to make computer system convenient to use and to utilize computer hardware and various resources in an efficient manner.

An operating system is a large set of software which is also an interface between users and computer systems. It is system software. Operating system also manages resources of the computer system, such as memory, processor, file system and input/output devices.

2.2 Objectives

After going through this unit, you will be able to

1. *Define operating system*
2. *Understand different types of operating system*
3. *Describe different functions of operating system*
4. *Understand files and directories*
5. *Understand the concept of algorithms*
6. *Describe the basics of data structure*

2.3 Operating System

An Operating System is program that makes the computing power available to users by controlling the hardware.

Another definition is that Operating System is a program that controls the execution of application programs. It is a set of processes permanently or transitively resident within the computer that makes the resources of the computer system available to the user in a consistent, reliable, friendly way.

Operating System can be divided into the Kernel and the Operating System. The kernel is the essential centre of a computer operating system, the core that provides basic services for all other parts of the operating system.

A kernel can be contrasted with a shell, the outermost part of an operating system that interacts with user commands. Kernel and shell are terms used more frequently in UNIX. Users interact with operating system through Command Line Interfaces (CLIs) or Graphical User Interfaces (GUIs). The operating system thus contributes to the simplification of the human interaction with the computer hardware. They are responsible for linking application programs with the hardware, thus achieving an easy user access to the computers.

Structure

A Computer System is made up of hardware, software and user interface. Computer Software can be divided into System programs and Application programs.

System software manages the operation of the computer itself and the application programs, which solve problems for their users. Operating System is the most fundamental of all the System programs (or System Software). The Operating System controls the entire Computer's entire resources and provides the base upon which the application programs can be written.

Figure 2.1 shows the different layers of the structure of a computer system. The core of a computer system is its hardware and the higher level is the user who interfaces with the computer with the application program with the help of operating system.

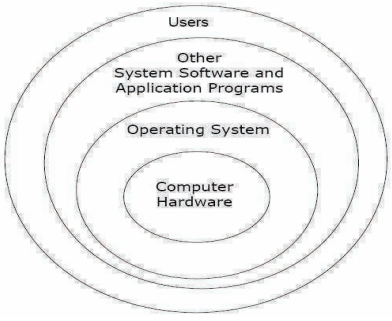

Figure 2.1: Logical structure of a computer system

2.4 Types of Operating System

Operating systems are categorized based on the types of computers they control and the sort of applications they support. The types are:

2.4.1 Single User Operating System

Single User Operating System is the simplest operating system which allows only one user to work on a computer at a time is known as single user operating system. For example MS-DOS, Windows 98 ect.

Single user operating systems, as opposed to a multi user operating system, are usable by a single user at a time. Being able to have multiple accounts on a Windows operating system does not make it a multi user system. Rather, only the network administrator is the real user.

2.4.2 Multi User Operating System

***Multi User Operating System** This type of operating system which is running on a computer will manage the work of all different users, without letting them know that they all are actually working on a single computer. A multi user operating system allows a number to work together on a single computer. Each user will be provided with a terminal and all such terminals will be connected to the single computer. For example, UNIX, Linux, Windows XP etc.*

A multi user operating system allows many different users to take advantage of the computer's resources simultaneously. The operating system must make sure that the requirements of the various users are balanced, and that each of the programs they are using has sufficient and separate resources so that a problem with one user does not affect the entire community of users. UNIX, mainframe operating systems, such as MVS, are examples of multi user operating systems

2.4.3 Single Tasking Operating System

Single Tasking Operating System It is one type of operating system which can execute a single job at a time. So it is known as Single Tasking operating system. For example, MS-DOS is a single tasking operating system because one can open and run only one application at one time in MS-DOS.

When a single program is allowed to run at a time, the system is grouped under a single-tasking system.

2.4.4 Multi-Tasking Operating System

Multi-Tasking Operating System Multi-tasking operating system allows the user to perform more than one job at the same time on a computer. Now a days most of the operating systems such as UNIX, LINUX, OS/2 etc. are multi-tasking operating system.

Multi-tasking can be of two types namely, pre-emptive and co-operative. In pre-emptive multitasking, the operating system slices the CPU time and dedicates one slot to each of the programs. UNIX like operating systems such as Solaris and Linux support pre-emptive multi-tasking. Cooperative multi-tasking is achieved by relying on each process to give time to the other processes in a defined manner. MS Windows prior to Windows 95 used to support cooperative multitasking.

2.4.5 Multiprogramming Operating System

In multiprogramming technique of processing, the CPU runs several programs at the same time. Multiprogramming is implemented in such a way that it appears as if many programs are being executed concurrently. Multiprogramming keeps CPU busy, switching its attention from one program to another. For example, while one program is waiting for some input or output operation, another program can use the processor because of the high speed of the processor.

Multiprogramming is the rapid switching of the CPU between multiple processes in memory. It is done only when the currently running process requests I/O, or terminates. It was commonly used to keep the CPU busy while one or more processes are doing

I/O. It is now mostly superseded by multitasking, in which the processes also lose the CPU when their time quantum expires.

Multiprogramming makes efficient use of the CPU by overlapping the demands for the CPU and its I/O devices from various users. It attempts to increase CPU utilization by always having something for the CPU to execute.

The prime reason for multiprogramming is to give the CPU something to do while waiting for I/O to complete. If there is no DMA, the CPU is fully occupied doing I/O, so there is nothing to be gained (at least in terms of CPU utilization) by multiprogramming. No matter how much I/O a program does, the CPU will be 100% busy. This of course assumes the major delay in the wait while data is copied. A CPU could do other work if the I/O were slow for other reasons (arriving on a serial line, for instance). I/O operations are exceedingly slow (compared to instruction execution). A program containing even a very small number of I/O operations, will spend most of its time waiting for them. Hence, poor CPU usage is seen when only one program is present in memory.

2.4.6 Real Time Operating System

Real time operating systems work towards providing immediate processing and also responding to user's commands in a very short period of time. This type of operating system is more commonly used in chemical industries for process control and scientific processing like airplane control and space vehicle control operations. Success of a real time operating system does not depend only on the correctness of the result but also on the timeliness of the result. For example HP-RT and VTWorks ect. It is a multitasking operating system that aims at executing real-time applications. Real-time operating systems often use specialized scheduling algorithms so that they can achieve adeterministic nature of behavior. The main object of real time operating system is their quick and predictable response to events. They either have an event-driven or a time-sharing design.

An event-driven system switches between tasks based of their priorities while time-sharing operating systems switch tasks based on clock interrupts.

2.4.7 Network Operating System

Network Operating System (NOS) is an operating system specifically designed to support interconnection of several computers. NOS provide support for multiuser operating system as well as administrative, security and network management functions. Some examples of network operating system are Windows NT, Novell's Netware etc.

A network operating system has to acknowledge and respond to requests from many work- stations, managing network access, resource allocation and sharing, data protection as well as error control.

2.4.8 Distributed Operating System

A Distributed Operating System hides the existence of multiple computers which are interconnected by a network from the user. That is, the user remains unaware of the fact that many computers are being used to process the data. These computers may be located at many places around the globe.

Distributed Operating System provides single-system image to its users. All these computers work in close coordination with each other. Processes and system resources are managed globally and controlled from specific locations.

An operating system that manages a group of independent computers and makes them appear to be a single computer is known as a distributed operating system. The development of networked computers that could be linked and that can communicate with each other, gave rise to distributed computing. Distributed computations are carried out on more than one machine. When computers in a group work in cooperation, they make a distributed system.

2.4.9 Embedded Operating System

The operating systems designed for being used in embedded computer systems are known as embedded operating systems. They are designed to operate on small machines like PDAs with less autonomy. They are able to operate with a limited number of resources. They are very compact and extremely efficient by design. Windows CE, Free BSD and Minix 3 are some examples of embedded operating systems.

The operating systems thus contribute to the simplification of the human interaction with the computer hardware. They are responsible for linking application programs with the hardware, thus achieving an easy user access to the computers.

2.5 Functions of the operating system

There are various functions performed by an operating system. The basic functions are Input/output management, Memory management, Process management, File and directory management, Security management, Command interpretation

2.5.1 Input/output management

An operating system makes efficient use of input and output devices and controls them properly. The Input/output management module of the operating system coordinates and assigns different input and output devices, namely terminals, printers, disk drives, tape drives ect. It controls all input and output devices, keeps track of input and output requests, issues commands to these devices and takes measures which would ensure that data is transmitted efficiently and correctly to and from input and output devices.

So far we have studied how resources like processor and main memory are managed. We shall now examine the I/O management. Humans interact with machines by providing information through I/O devices. Also, much of whatever a computer system provides as on-line services is essentially made available through specialized devices such as screen displays, printers, keyboards, mouse, etc. Clearly, management of all these devices can affect the throughput of a system. For this

reason, input output management also becomes one of the primary responsibilities of an operating system.

2.5.2 Memory or storage management

Memory is an important resource for a computer system. So memory allocation and deallocation for programs and data is an another function of an operating system. Memory management module of an operating system takes care of allocation and de-allocatiion of the main memory to various procsses. It allocates the main memory and the secondary memory to the system programs, user programs and data. For executing a program, its binary image must be loaded into the main memory to hold data from disk files, the contnts of which it is reading or manipulating at the executing time.

When an operating system manages the computer's memory there are two broad tasks to be accomplished:

1. *Each process must have enough memory in which to execute, and it can neither run into the memory space of another process nor be run into by another process.*

2. *The different types of memory in the system must be used properly so that each process can run most effectively.*

The first task requires the operating system to set up memory boundaries for types of software and for individual applications.

When applications begin to be loaded into memory, they are loaded in block sizes determined by the operating system. If the block size is 2 kilobytes, then every process that's loaded will be given a chunk of memory that's a multiple of 2 kilobytes in size. Applications will be loaded in these fixed block sizes, with the blocks starting and ending on boundaries established by words of 4 or 8 bytes. These blocks and boundaries help to ensure that applications won't be loaded on top of one another's space by a poorly calculated bit or two. With that being ensured, the larger question is what to do when the 500-kilobyte application space is filled.

In most computers, it is possible to add memory beyond the original capacity. For example, we can expand RAM from 1 to 2 gigabytes. This works fine, but can be relatively expensive. It also ignores a fundamental fact of computing — most of the information that an application stores in memory is not being used at any given moment.

A processor can only access memory one location at a time, so the majority of RAM is unused at any moment. Since disk space is cheap compared to RAM, then moving information in RAM to hard disk can greatly expand RAM space at no cost. This technique is called ***virtual memory management****.*

Disk storage is only one of the memory types that must be managed by the operating system, and it is the slowest process. Ranked in order of speed, the types of memory in a computer system are:

High-speed cache*: This is fast and relatively small amounts of memory that are available to the CPU through the fastest connections. Cache controllers predict which pieces of data the CPU will need next and pull it from the main memory into high-speed cache to speed up system performance.*

Main memory*: This is the RAM that you see measured in megabytes when you buy a computer.*

Secondary memory *: This is most often some sort of rotating magnetic storage that keeps applications and data available to be used, and serves as virtual RAM under the control of the operating system.*

2.5.3 Process management

Process can be defined as a program in execution. It is the job which is currently being executed by the processor (CPU) . During its execution, a process would require certain system resources, such as processor time, main memory, files, input and output devices etc. UNIX operating system supports multiple processes simultaneously. The process management module of UNIX takes care of creation and termination of processes, assigning required resources to different processes

currently running, scheduling h processor's time to different processes and providing mechanisms for synchronization and communication among the processes. A process in execution needs resources like processing resource, memory and I/O resources. Current machines allow several processes to share resources. In reality, one processor is shared amongst many processes.

*As we know a **process** is a program in execution. To understand the importance of this definition, let us imagine that we have written a program called my_prog.c in C language. On execution, this program may read in some data and output some data. When a program is written and a file is prepared, it is still a script i.e., it cannot cause any input, processing or output to happen. Once we compile, and still later when we run this program, the intended operations take place. In other words, a program is a text script with no dynamic behavior. When a program is in execution, the script is acted upon. It can result in engaging a processor for some processing and it can also engage in I/O operations. It is for this reason a process is differentiated from program. While the program is a text script, a program in execution is a process.*

2.5.4 Files and directory management

Files and directories are also properly managed by an operating system. Data is stored in a computer system as files. The file management module of the operating system would manage files as well as directories held on various storage devices as well as transfer of files from one storage device to another device. Thus, file management takes care of organization, storage, retrieval, naming, sharing and protection of different files. It also allows files to be read and modified by using text editors or some other file manipulation software packages.

We emphasized that a computer system processes and stores information. Usually, during processing computers need to frequently access primary memory for instructions and data. However, the primary memory can be used only for only temporary storage of information. This is so because the primary memory of a computer system is volatile. The volatility is evinced by the fact that when we switch

off the power the information stored in the primary memory is lost. The secondary memory, on the other hand, is non-volatile. This means that once the user has finished his current activity on a computer and shut down his system, the information on disks (or any other form of secondary memory) is still available for a later access. The non-volatility of the memory enables the disks to store information indefinitely. This information can also be made available online all the time. Users think of all such information as files. As a matter of fact, while working on a computer system a user is continually engaged in managing or using his files in one way or the other. OS provides support for such management through a file system. File system is the software which empowers users and applications to organize and manage their files. The organization and management of files may involve access, updates and several other file operations. In this chapter our focus shall be on organization and management of files.

File

Suppose, we are developing an application program. A program, which we prepare, is a file. Later we may compile this program file and get an object code or an executable. The executable is also a file. In other words, the output from a compiler may be an object code file or an executable file.

When we store images from a web page we get an image file. If we store some music in digital format it is an audio file. So, in almost every situation we are engaged in using a file. In addition, we saw in the previous module that files are central to our view of communication with Input/Output.

Scheduling

The main motivation for scheduling various OS services is to maximize the usage of CPU resource, memory, and other IO resources. Consider the use of a printer as an output resource. A user takes printouts only once in a while. A printer usage, therefore, can be shared amongst many users. The motivation to

share a resource may come from several reasons. Sharing enhances utilization of resources immensely.

The operating system also establishes and enforces process-priority. That is, it determines and maintains the order in which jobs are to be executed by the computer system. So the most important jobs must be executed first followed by less important jobs. Scheduler selects the most deserving process to run, out of all the runnable processes in the system, based on their relative priority.

Need for Scheduling

Resources may be categorized depending upon the nature of their use.

To enforce temporal sharing of a common resource, the OS needs a policy to schedule its usage. The policy may depend upon the nature of resource, frequency of its use and the context of its usage. In the case of a printer, the OS can spool printout requests. Printing, additionally, requires that once a process is engaged in printing, it must have its exclusive usage till it finishes the current print job. If that is not the case then the printouts from the printer shall be garbled. Some specialized resources, like a flat-bed plotter, require an elaborate initial set-up. So, once assigned to a process, its usage better not be pre-empted. A process that gets such a resource should be permitted to keep it till either the process terminates or releases the resource. This is also true of a transaction which updates a shared data record. The transaction should complete the record's update before another process is given the access to the record.

Processes may need more than one resource. It is quite possible that a process may not be able to progress till it gets all the resources it needs. Let us suppose that a process P1 needs resources r1 and r2. Process P2 needs resources r2 and r3. Process P1 can proceed only when it has both r1 and r2. If process P2 has been granted r2 then process P1 has to wait till process P2 terminates or releases r2. Clearly, the resource allocation policy of an OS can affect the overall throughput of a system.

2.5.5 Security management

It is also one of the most important functions of an operating system to protect its various resources from the outer world. Security management module of the operating system which is executing concurrently in the memory, in such a way that the system ensures data security and integrity. That is, it protects data and program stored in the computer system from destruction and unauthorized access. It keeps different programs and data in such a way that they donot interfere with each other. Moreover, it protects files from being accessed or modified by unauthorized users.

The operating system assigns processors (if a computer has more than processors) to different tasks that must be performed by the computer system. Whenever the operating system finds any processor idle, one of the processors waiting to be executed is assigned to this idle processor. The main objective of the processor management module of the operating system is to ensure that a process is run on each processor at all times.

Computers, with their ubiquitous presence, have ceased to be a wonder they once used to be. Their usage is pervasive. Information access and delivery from, and to, a remote location via internet is common. Today many societal services like railway time-table or election results are rendered through computers. The notion of electronic commerce has given fillip to provisioning commercial services as well. Most individuals use computers to store private information at home and critical professional information at work. They also use computers to access information from other computers anywhere on the net. In this kind of scenario, information is the key resource and needs to be protected.The OS, being the system's resource regulator, must provide for security mechanisms. It must not only secure the nformation to protect the privacy but also prevent misuse of system resources. Unix designers had aimed to support large-scale program development and team work. The main plank of design was flexibility and support tools. The idea was to promote the creation of large programs through cooperative team efforts.

Security has become a bigger issue now. Much of Unix provisioning of services was with the premise that there are hardly, if any, abuses of system. So, Unix leaves much to be desired in respect of security. And yet, Unix has the flexibility to augment mechanisms that primarily protect the users resources like files and programs.

2.6 Algorithm

*An **algorithm** is a step-by-step procedure for solving a problem. Algorithms are a fundamental part of computing. It is thus a sequence of computational steps that transform the input into output. To be an algorithm, a set of rules must be unambiguous and have a clear stopping point. Algorithms can be expressed in any language, from natural languages like English.*

Example: Algorithm for calculating sum, average and product of 3 numbers:

Step 1: Input X, Y, Z

Step 2: Compute Sum (S) as X + Y + Z

Step 3: Compute Average (A) as S / 3

Step 4: Compute Product (P) as X x Y x Z

Step 5: Display the Sum, Average and Product

Following are the factors that designers should give importance while developing an algorithm:

- *Preciseness of the algorithm*
- *All possible circumstances are handled.*
- *The algorithm is executable.*
- *Termination of the algorithm.*
- *Consideration of time and using more resources than required to solve the problem.*

• *Efficiency becomes more important for larger programs.*

Efficiency of algorithm:

In Computer Science, often the question is not how to solve a problem, but how to solve a problem efficiently. For instance, take the problem of sorting. Many sorting algorithms are well-known; the problem is not to find a way to sort words, but to find a way to efficiently sort words.

The ability to analyze a piece of code or an algorithm and understand its efficiency is vital for understanding Computer Science. Efficiency of algorithm depends on two major criteria, first one is ***run time*** *of that algorithm and second is the* ***space****. Run time of algorithm means the time taken by program for execution. While implementing an algorithm we use data structures. Some data structure takes more space but improves the run time and some takes less space but it effects the run time of the algorithm. Usually the resource being considered is running time, but it could also be memory or other resources. Suppose the space is fixed for one algorithm then only run time will be considered for obtaining the efficiency of the algorithm.*

2.7 Data Structures

There is always a tradeoff between the program speeds versus the program storage. Here the storage is the space of the program instruction, normally called as memory and speed is the execution of the program time. Data structures can make memory usage efficient. There are three different types of data structures used to effectively use the program space for the better execution time.

2.7.1 Array

Array implementation is important because, most assembly languages have no concept of arrays. In array, each element is the same size (char = 1 byte, integer = 1 word) and elements are stored contiguously, with the first element stored at the smallest memory address (called the base address). So, the whole trick in assembly language is to allocate the correct amount of space for an array and the address tells

the location of an element. To allocate a portion of memory (more than a single variable's worth) (allocating an array within memory):

Examples:

Array name: .byte 0:8

This gives 8 character-sized elements, numbered 0 - 7, initialized to 0, which is the null character.

Name: .space 18

This gives 18 bytes of memory (with no implied initial contents).

2.7.2 Stack

We often need a data structure that stores data in the reverse order that it is used. Along with this is the concept that the data is not known until the program is executed (run time). A stack allows both properties. Abstractly, here is a stack. Also called Last in First Out, LIFO Data put into the stack is said to be pushed onto the stack. Data taken out of the stack is said to be popped off the stack. These are the 2 operations defined for a stack. Whereas a stack is LIFO, a queue is FIFO (First In, First Out). Getting into the queue is an operation called enquires, and taking something off the queue is operations called DE queue.

2.7.3 Queue

Queues in Computer Science are very similar to queues in real life. A queue in real life would be a line up at a railway ticket counter or a bank teller. The people are dealt with in the order that they arrived.

Queue called FIFO (First-In First-Out) queue. Items can only be removed from the queue in the order that they are inserted into it. Queues are very common in the everyday functions of computers. In a computer, there may be a queue of tasks waiting for execution, few others for printing, and some are for inputting the data and instructions through the keyboard. Unless the task turns and comes on front, it will

not be executed. The most obvious is printing queues. When we send multiple print jobs to a printer, each printing job is inserted at the rear of the queue in the order it was sent. Each job is then printed in the order sent to the printer. Processes waiting to be executed by a CPU are usually in form of a queue also.

Queues are widely used in simulations. In any situation, where the rate at which users' demand for services can exceed the rate at which those services can be supplied, queues are normally used.

Queues are useful in a time sharing computer system where many users share the system simultaneously. For e.g., printer maintains a queue of jobs waiting to get the printer. A queue also stores the keystroke of data as you type on the keyboard. Queues are frequently used in computer programming and a typical example is the creation of job queue by an operating system. If the operating system does not use priorities, then the jobs are processed in the order they enter the system, i.e., first come first served basis. Queues are also used in Internet message packets waiting to be routed.

2.8 Summery

An Operating System is an important component of a computer system. The primary objective of an operating system is to make the computer system convenient to use and utilize computer hardware and various resources in an efficient manner. An operating system is a large set of software which is also an interface between the users and the computer systems. It is also a system software.

2.9 Further Readings

1. *Andrew S Tanenbaum, Albert S. Woodhull, "Operating Systems Design and Implementation", Prentice-Hall India publication.*

2. *Deitel & Deitel, Choffnes, "Operating System", Pearson Education.*

2.10 Terminal Questions

1. *What is Operating system? Give some popular OS name.*
2. *What is process? What are the main objectives of the process-management module of the operating system?*
3. *Differentiate between multiprogramming and multi-tasking operating system.*
4. *What are different functions performed by an operating system?*
5. *Differentiate between network operating system and distributive operating system?*

***

UNIT 3: BASICS OF NETWORKING

3.1 Introduction

There has been tremendous success of World Wide Web. The merging of computers and communications has had a profound influence on the way computer systems are organized.

The concept of the "computer center" as a room with a large computer to which users bring their work for processing is now totally obsolete. The old model of a single computer serving all of the organization's computational needs has been replaced by one in which a large number of separate but interconnected computers do the job. These systems are called computer networks.

One way to categorize the different types of computer network designs is by their scope or scale. For historical reasons, the networking industry refers to nearly every type of design as some kind of area network. In this unit we will discuss about the different networks.

3.2 Types of Network

Whenever we have a set of computers or networking devices to be connected, we make the connections, depending on the physical layout and our requirements. On the basis of geographical area covered, computer networks can be classified under the following three categories:

- *Local Area Network*
- *Metropolitan Area Network*
- *Wide Area Network*

3.2.1 Local Area Network (LAN)

*A **local area network** or **LAN**, is a high-speed data network that covers a relatively small geographic area such as a building, a laboratory, or a school. It typically connects workstations, personal computers, printers, servers, and other*

devices. LANs offer computer users many advantages, including shared access to devices and applications, file exchange between connected users, and communication between users via electronic mail and other applications. LANs differ in the way the computers are connected (i.e., their topology), how information moves around the network (i.e., their transmission technology) and their size.

IEEE (Institute of Electrical and Electronic Engineers) is a US publishing and standards organization responsible for many LAN standards such as the 802 series. IEEE 802.3, popularly called Ethernet, is the most popular LAN, usually operating at 10 Mbps to 10 Gbps and is present in most large organizations and offices. In a typical LAN configuration, one computer is designated as the server. It stores all of the software that controls the network as well as the software that can be shared by the computers attached to the network. Computers connected to the server are called workstations. Two most commonly used LAN implementations are depicted below:

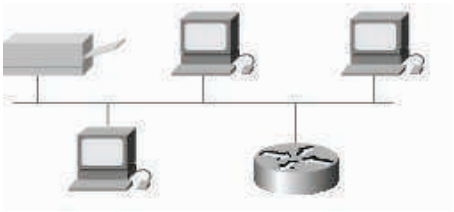

Fig. 3.1(a): Ethernet or IEEE 802.3

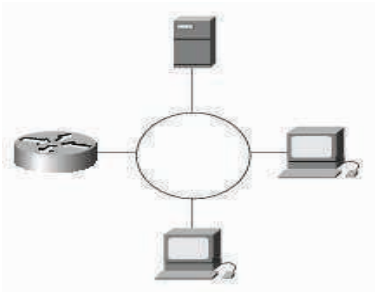

Fig. 3.1(b): Token Ring or IEEE 802.5

LAN protocols function at the lowest two layers, i.e., the physical and the data link layer of the OSI reference model. We will discuss the layer concept of ISO-OSI reference model in the next unit which is "Network models".

3.2.1.1 LAN Transmission Methods

LAN data transmissions fall into three classifications: unicast, multicast, and broadcast. In each type of transmission, a single packet is sent to one or more nodes.

- ***Unicast :*** *Unicast is a one-to-one transmission method in which the network carries a message to one receiver, such as from a server to a LAN workstation. In a unicast environment, even though multiple users might ask for the same information from the same server at the same time, such as a video clip, duplicate data streams are sent. Unicast sends separate data streams to each computer requesting the data, thereby flooding the network with traffic.*

- ***Multicast :*** *Multicast is a one-to-many transmission method in which the network carries a message to multiple receivers at the same time. Multicast is similar to broadcasting, except that multicasting means sending to a specific group, whereas broadcasting implies sending to everybody, whether they want the traffic or not. When sending large amounts of data, multicast saves considerable network bandwidth because the bulk of the data is sent only once. The data travels from its source through major backbones and is then multiplied, or distributed out, at switching points closer to the end users. This is more efficient than a unicast system, in which the data is copied and forwarded to each recipient.*

- ***Broadcast :*** *Concept of broadcast is already discussed in previous section. Broadcast is a one-to-all transmission method in which the network carries a packet to all devices at the same time, but a particular machice for which the packet is intended accepts it.*

3.2.1.2 Peer-to-Peer LAN & Server Based LAN

On a LAN, we expect to share files, programs, or printers, all without being particularly aware of where the physical resources we're using are actually located. LANs providing these types of services are typically set up either as "peer-to-peer" or "client-server" LANs, or perhaps as a combination of the two.

- ***Peer-to-Peer LAN***

  All the machines on a peer-to-peer LAN are equal. Provided that the file's owners give permission, a file on machine A can be accessed from machine B, and vice versa. Peer-to-peer LANs do not require any one machine to be a dedicated, high-performance server; service by a peer-to-peer LAN is often cheaper for this reason. Peer-to-peer LANs work well when only a small number of machines are connected to it. But as the size of the LAN grows, peer-to-peer services can become quite disorganized. To serve all its peers, each machine on the LAN must be powerful enough and for this reason its cost increases. For larger LANs, the dedicated client-server LAN architecture becomes more cost effective.

  Advantages :

  Less initial expense – No need for a dedicated server.

  Setup – An existing operating system (such as Windows XP) of the machine may only need to be reconfigured for peer-to-peer operations.

  Disadvantages :

  Decentralized – No central repository for files and applications.

  Security – Does not provide the security available on a client/server.

- ***Client-Server LAN***

  A client-server LAN consists of one or more server machines on which shared files and programs reside and many client machines where people do their task. The LAN server machines usually have higher configuration

and are fast because they must serve many users, while the client machines need only to be fast enough for one person to use at a time. Shared printers are either attached directly to a server, or to a print server (a specialized computer attached to the network), or to a personal computer on network that acts as a print server.

Advantages :

Centralized – In case of client-server architecture, resources and data security are controlled through the server.

Scalability – Any or all elements can be replaced individually as needs increase.

Flexibility – New technology can be easily integrated into system.

Accessibility – Server can be accessed remotely and across multiple platforms.

Disadvantages :

Expense – Requires initial investment in dedicated server.

Maintenance – Large networks will require a staff to ensure efficient operation.

Dependence – When server goes down, operations will cease across the network.

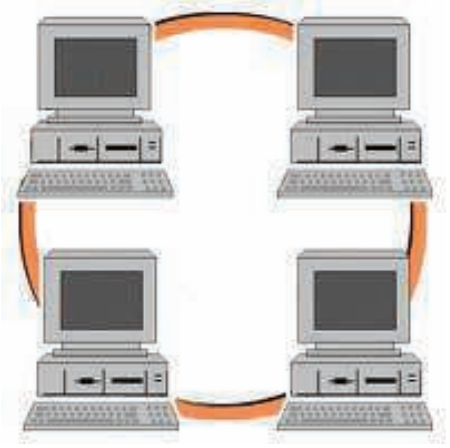

Fig. 3.2: Peer-to-Peer

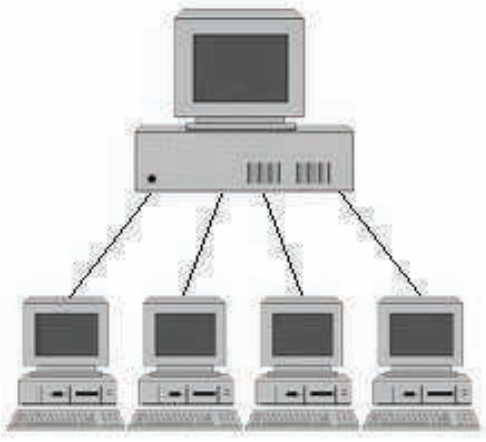

Fig. 3.3: Client-Server

3.2.2 Metropolitan Area Network (MAN)

*A **metropolitan area network**, or **MAN**, is basically a bigger version of a LAN and normally uses similar technology. It might cover a group of nearby corporate offices or a city and might be either private or public.*

A MAN often provides efficient connections as it has high-speed transmission capabilities which uses some types of telecomminication components to handle long-distance transmission. One very common example of MAN is the cable television network. Another important example is the high-speed wireless Internet access, which has been standardized as IEEE 802.16.

3.2.3 Wide Area Network (WAN)

*A **wide area network** or **WAN** is a computer network covering multiple distance areas, which may spread across the entire world. WANs often connect multiple smaller networks, such as local area networks (LANs) or metropolitan area networks (MANs).*

Typically, a WAN consists of a number of interconnected switching nodes. A transmission from any one device is routed through these internal nodes to the specified destination device. These nodes are not concerned with the content of the data; rather, their purpose is to provide a switching facility that will move the data from node to node until they reach their destination. Traditionally, WANs have been implemented using either circuit switching or packet switching technologies. More recently, frame relay and ATM (Asynchronous Transfer Mode) networks have assumed major roles. Frame relay provides higher data rates, lowar costs, efficient handling of bursty data transmission in less expenditure. ATM offers more bandwidth to end users at less cost.

*The **Internet** is the best known example of a WAN. Some segments of the Internet are also WANs in themselves. The Internet is a system of linked networks that are worldwide in scope and facilitate data communication services such as remote login, file transfer, e-mail, the World Wide Web etc. With the rise in demand for*

connectivity, the Internet has become a communications highway for millions of users. The Internet was initially restricted to military and academic institutions, but now it is a full-fledged conduit for any and all forms of information and commerce. Internet websites now provide personal, educational, political and economic resources to every corner of the planet.

3.3 Network Topology

In this section we will discuss how computers and others devices are connected in a network.

In computer networking, ***topology*** *refers to the layout of connected devices. Network topologies can be physical or logical.* ***Physical topology*** *means the physical design of a network including the devices, location and cable installation. It defines how the systems are physically connected. Several physical topologies are in use for networks today. Some of the common topologies include the bus, ring, star, tree and mesh. More complex networks can be built as hybrids of two or more of the above basic topologies. The* ***logical topology*** *defines how the systems communicate across the physical topologies. The two most common types of logical topologies are* ***broadcast*** *and* ***token passing****.*

When we decide which topology we should choose for our network, then there are few basic points that we need to take cares of. The factors that decide which topology we should choose are:

- ***Cost*** *: Cost is a factor that plays an important role for the decision of topology. If we want to create a network for 4-5 computers, we should not expense very much in network.*
- ***Scalability****: What is the size of the network that we need to make and is it possible in that kind of topology*
- ***Bandwidth capacity****: The required speed of the network that can be taken care of by any particular topology.*

- ***Ease of installation****: Is it easy to install the network using selected topology.*
- ***Ease of fault finding and maintenance****: If we have a network we will definitely get the problem also; so will it be easy for network administrator to identify the problem and give the solution with ease and in least possible time.*

3.3.1 Bus Topology

A network of bus topology consists of a long single cable to which all the computers and other devices are connected. Any node attached to the bus can send signals down the cable to all the nodes of the network; that means, a bus is a broadcast medium. When more than one node start sending data through the bus, they mix with each other and the sent data becomes a garbage. This is called collision. To avoid collision there must be some agreement between the nodes so that when one computer starts to send data, others refrain themselves from sending data. To ensure correct data communication, both ends of the cable are terminated by a special device called end terminator.

Fig. 3.4: A Bus Topology

Ethernet bus topologies are relatively easy to install and don't require much cabling compared to the alternatives. 10Base-2 (Thin) and 10Base-5 (Thick) both were popular Ethernet cabling options for bus topologies. Some advantages and disadvantages of the bus topology are listed below:

Advantages :

- *Bus topology is inexpensive in installation*
- *It requires less cable than other topologies*

- *Good for smaller networks not requiring higher speed.*
- *Easy to add systems to network.*

Disadvantage :

- *Out-of-date technology. Bus topology was used in the early days of networking because it was inexpensive to use and relatively easy to set up.*
- *If the backbone cable fails, the entire network effectively becomes unusable.*
- *Unmanageable in a large network. If more than a few dozen computers are added to a network bus, performance problems are likely to occur.*
- *Difficult to troubleshoot.*

3.3.2 Ring Topology

In ring topology the computers are connected between each other in a way that forms a close loop. In practice, a cable connects the first computer to the second computer, another cable connect the second computer to the third and so on until the last computer is connected back to the first to complete the loop. It should be noted that the topology may not physically look like a circle. The ring means the computers are connected with each other in a logical ring. The interconnecting cables may take any shape in practice. All messages travel through a ring in the same direction (either "clockwise" or "counterclockwise").

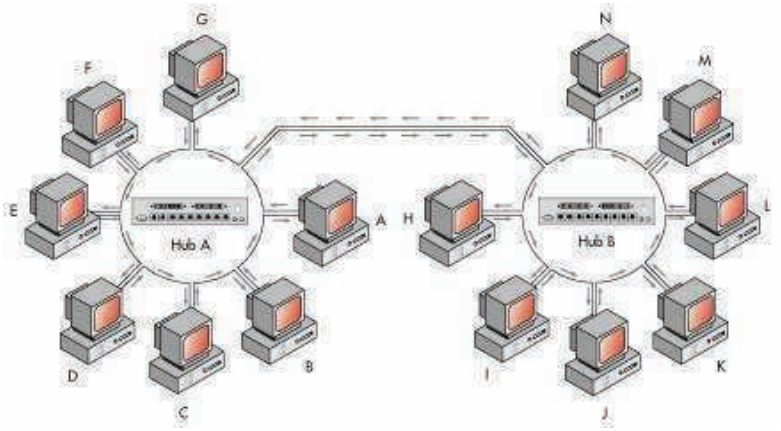

Fig. 3.5: A Dual Ring Topology

In ring topology the communicating computers must follow some agreement between them to avoid collision as it is also a broadcast network. The main difference between the bus and ring is that the ring topology does not require termination. Because the systems are connected all together in a loop, there are no beginning and end point as there is with the bus topology. A failure in any cable or device breaks the loop and can take down the entire network.

Two major network types that use the ring topology are:

i) Fiber Distributed Data Interface (FDDI) where a large, high speed networks use fiber optic cables in a physical ring topology.

ii) Token-Ring networks that use logical ring topology.

Advantages :

- *No collision of data as data travel in one direction only.*
- *Easier to fault find. If any point gets broken we can trace that easily.*
- *No terminator required.*

Disadvantages :

- *Ring topology requires more cable than bus topology*
- *A break in ring will take the whole network down.*
- *Addition or removal of any node can affect the entire network.*

3.3.3 Star Topology

In the star topology, all computers and other network devices connect to a central device(controller) called a ***hub*** *as depicted in the figure.1.11. Each connected device requires a single cable to be connected to the hub. A hub normally accepts data from a sending computer and delivers it to the computer for which the data*

is addressed. Hence, a star network is not a broadcast network, rather a point-to-point network. Using a separate cable to connect to the hub allows the network to be expanded without disruption to the network. Because each computer uses a separate cable to connect to the hub, the failure of a network connection affects only the single machine concerned. The other computers can continue to function normally.

Fast Ethernet (100Base-TX & 100Base-FX) in a star topology is the most commonly used LAN today. 10Base-T Ethernet and 1000Base-TX Gigabit Ethernet also use star topology.

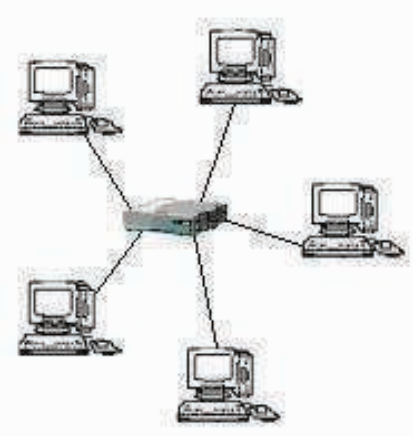

Fig. 3.6: A Star Topology

Adavntages :

- *Star networks are easily expanded without disruption to the network.*
- *Easy to add/remove devices to/from network*
- *One break does not bring the whole network down. Cable failure affects only a single user.*
- *Easy to troubleshoot and isolate problems.*
- *Widely used centralized management*

Disadvantages :

- *Costs are usually higher than with bus or ring networks.*
- *Requires more cable than most of the other topologies.*
- *If the hub fails, any device connected to it will not be able to access the network.*

In the following table, various networks with their cable types, topologies are shown:

Network Type	Standard	Cable Type	Topology
Ethernet	10Base-2	Thin(RG-58)	Bus
Ethernet	10Base-5	Thick(RG-59)coaxial	Bus
Ethernet	10Base-T	CAT3 or CAT5 UTP	Star
Fast Ethernet	100Base-TX	CAT5 UTP	Star
Gibabit Ethernet	1000BaseTX	CAT5, 5e or UTP	Star
Token Ring	all	UTP or STP	Logical Ring

Table 3.7: Network cable types and topologies

3.3.4 Mesh Topology

The mesh topology incorporates a unique network design in which each computer on the network connects to every other, creating a point-to-point connection between every device on the network. A mesh topology is used when there can be absolutely no break in communications; for example, the control systems of a nuclear power plant. The purpose of the mesh topology is to provide a high level of redundancy. If a network cable, computers or other components fail, the data always has an alternative path to get to its destination.

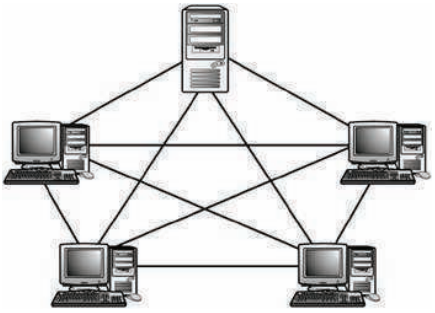

Fig. 3.8: A Mesh Topology

A fully connected mesh network has n(n-1)/2 cables to link 'n' devices. Therefore, every device on the network must have 'n-1' input/output (I/O) ports. For example, in the Fig.1.12, we have five systems that require 10 cables to create a mesh network. This topology is mainly used in environments where high availability outweighs the costs associated with this amount of interconnection. We can see in the figure that the wiring for a mesh network can be very complicated. Further, troubleshooting a failed cable can be tricky. Because of this, the mesh topology is rarely used.

Advantages :

- *Provides alternative paths between devices in the network.*
- *The network can be expanded without disruption to current users.*

Disadvantages :

- *It is expensive as because a large amount of cabling are required.*
- *Routing network traffic can be difficult because of all the different possible paths between nodes.*
- *It is very expensive to wire up.*

There are also partial-mesh networks where some of the nodes are connected to all the others, but others are only connected to nodes with which they exchange the most of the data.

3.3.5 Tree Topology

A tree topology combines the characteristics of bus and star topologies. It consists of groups of star-configured workstations connected to a linear bus backbone cable (Fig. 1.13). This bus/star hybrid approach supports future expandability of the network much better than a bus (limited in the number of devices due to the broadcast traffic it generates) or a star (limited by the number of hub connection points) alone.

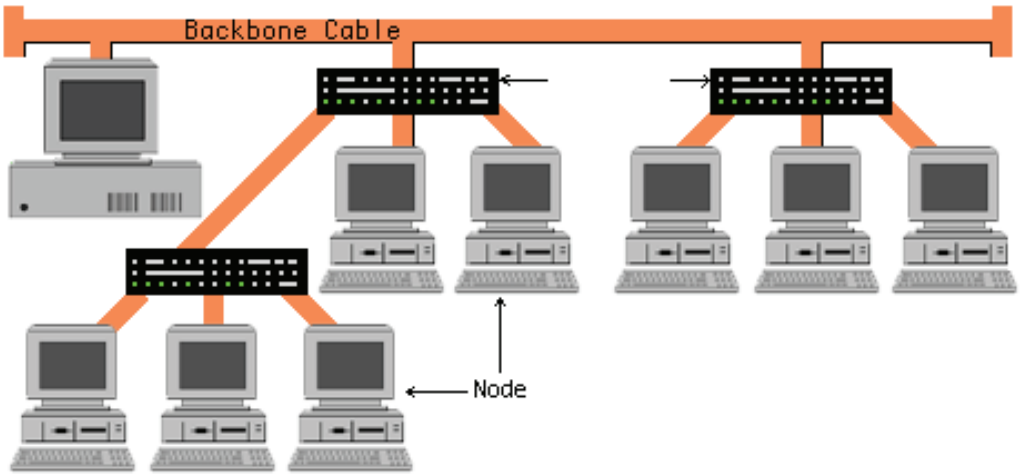

Fig. 3.9: A Tree Topology

Advantages of Tree Topology :

- *The network is easy to extend by just adding another branch.*
- *Fault isolation is relatively easy.*

Disadvantages of Tree Topology :

- *If the backbone cable breaks, the entire network goes down.*
- *More difficult to configure and wire than other topologies.*
- *If any hub goes down, all branches of that hub go down.*

3.5 ISO-OSI Reference Model

The OSI reference model is shown in Fig 3.10. The International Standard Organization (ISO) developed a proposal for a network model and the resulting model is known as ISO – OSI reference model. This model has seven layers and the layers are :

i) Physical layer

ii) Data link layer

iii) Network layer

iv) Transport layer

v) Session layer

vi) Presentation layer

vii) Application layer

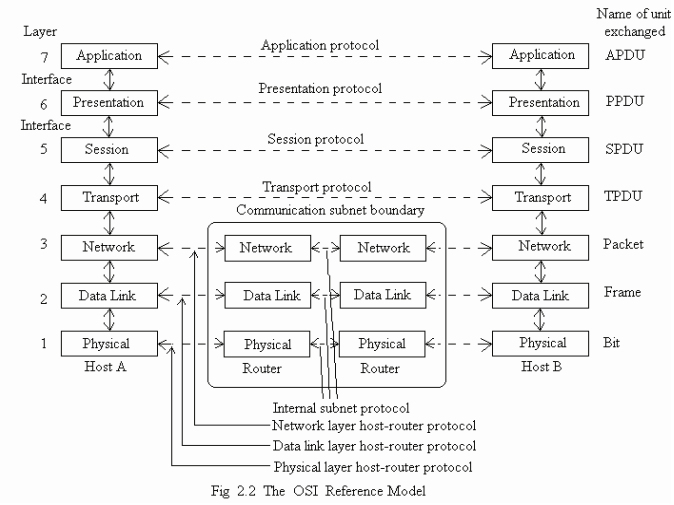

Fig. 3.10: Layers, protocols and interfaces

(i) The Physical Layer : *The physical layer is responsible for transmitting raw bits over the communication channel. This layer is to ensure how to send a 1 bit from the sending computer as 1 bit to the receiving computer; not as 0 bit.*

The physical layer also deals with the issues of how many bits per second will be transmitted, what level of voltage will be used to represent 1 and 0, whether transmission will be unidirectional or bidirectional, how the initial connection be established and terminated at the end, how many pins of network the connector has and which pin is for what etc etc. Hence, the design issues of the physical layer is mostly mechanical, electrical and procedure oriented.

(ii) The Data Link Layer : *The data link layer use the raw transmission facility and transforms it to an apparently error- free facility to be used by the network layer. This layer breaks the input data into frames by inserting appropriate frame boundary, transmits the frames sequentially and processes the acknowledgment sent back by the receiving computer. If a frame is completely destroyed by noise burst, it is the duty of data link layer to retransmit it from the source machine. Data link layer also ensures that a fast sender is not allowed to swamp a slow receiver by sending data at a higher rate than it can be handled by the receiver. This is called* ***flow control****. In a broadcast network it is the duty of MAC (Medium Access Control) sub-layer of data link layer to decide who will access the transmission medium at a particular time.*

(iii) The Network Layer : *The network layer controls the operation of the subnet. The layer is to determine how packets are routed from source to destination. The routing may be static or dynamic depending on traffic load and availability of channel.*

Too many packets may cause congestion (traffic-jam) and control of such congestion is also a duty of the network layer.

The subnet operation requires cost; hence, some accounting function is also there being built into the network layer. When a packet crosses national boundary some other aspects of accounting has to be dealt with by network layer.

Packets have to travel in between heterogeneous network running on different platforms using different network protocol and the network layer is also responsible to resolve all the problems arising out of such situations.

(iv) The Transport Layer : *The transport layer is to accept the data from the session layer, breaks it into smaller units if necessary, hands over these to the network layer and ensures that the pieces all delivered correctly to the receiver. The above duties must be done efficiently and in such a way that it will not affect the upper layer in case there is any change in the hardware.*

Normally, the transport layer creates individual connection for each session. If high throughput is required, the transport layer may establish multiple network connections, dividing the data among individual connections, thereby improving the throughput.

To reduce cost the transport layer may also multiplex several transport connections onto the network connection. However, multiple connections or multiplexing must not be seen by the session layer.

The transport layer also determines the type of service given to the user of the network. Error free point-to-point connection is the most popular transport layer service where messages or bytes are delivered in order in which they were sent. The transport layer also performs flow control.

(v) The Session Layer : *This layer offers facility to different users on different computers to establish session between them. A session allows an user to remotely log into a distant machine and transfer file between the two machines. Session layer perform token management to provide unidirectional communication. It also provides a service called synchronization.*

(vi) The Presentation Layer : *This layer performs data presentation job by following syntax and semantics rules. Before presenting data to the user, it transforms data into their acceptable form.*

(vii) The Application Layer : *This layer is the nearest layer to all the network users. It offers variety of protocols that are commonly needed. It helps to transfer file. Different file systems have different meanings in different machines with different data formats etc. When files are transferred from one machine to another with different file systems, the application layer takes the necessary steps to resolve the abnormalities.*

3.6 TCP/IP Reference Model

TCP/IP reference model was developed much earlier than the OSI reference model. It was evolved from the US Department of Defense's (DoD) research network- The ARPANET. Eventually the ARPANET connected many universities and other government organizations. During the process of interconnecting all these through the existing telephone lines, satellite link and radio, the protocol used for the ARPANET had trouble and therefore, a new architecture was needed to overcome that. The model has four layers, namely :

1. *The Host-to-Network Layer.*
2. *The Internet Layer.*
3. *The transport Layer.*
4. *The Application Layer.*

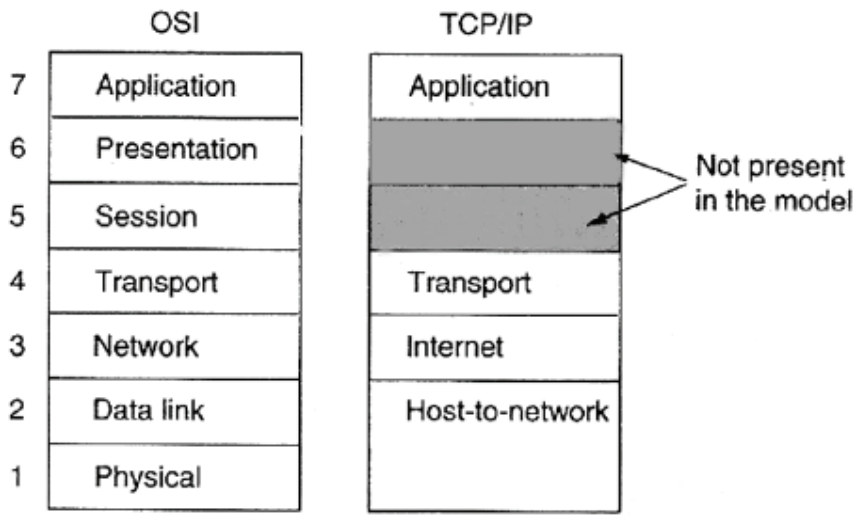

Fig. 3.11: The TCP/IP reference model

1. ***The Host-to-Network Layer :*** *In TCP/IP reference model the bottom layer is not defined clearly. Yet we can consider that whatever is there below the internet layer is the bottom layer in the TCP/IP model. In the model, the host has to be connected to the network using some protocol so that it can send IP packets over it.*

2. ***The Internet Layer :*** *The DoD planned to set up their inter-network in such a way that it had to survive even if a particular link failed in a probable war. So, this requirement led to connectionless packet-switching network instead of a connection-oriented circuit-switching network. The internet layer is designed to fulfil the goal of the architecture in such a way that it becomes the linchpin of the whole architecture. The job of this layer is to pump the packets from the host machine into any network and to help the packets to go to the destination independently. In this style of communication, packets may arrive in a different order than they were sent. This layer has its own protocol called* ***IP (Internet Protocol)*** *and a specified packet format. Packets routing as well as congestion avoidance are the major issues in this layer. This layer is similar to the network layer in OSI reference model.*

3. ***The Transport Layer :*** *The layer immediately above the internet layer is called the transport layer. This layer communicates with the peer layer on the other machine. It has two protocols namely* ***TCP and UDP.***

   (i)TCP: TCP is a reliable connection-oriented protocol that ensures delivery of byte stream from source machine to destination machine. This layer breaks the byte stream received from the upper layer into messages and passes them to the internet layer. At the receiving machine, these messages are reassembled by the TCP which passes them to the upper layer. Flow control is also another job of this layer to restrict a fast sender from swamping a slow receiver.

   (ii) UDP: UDP is another protocol used by this layer which offers an unreliable, connectionless service. Here, unlike TCP, sequencing and flow

control is not done. When prompt delivery is more important than accurate delivery, UDP is used. It is also widely used in one time client-server type communication. Email, video or sound transmissions are some of these applications.

4. ***The Application Layer:*** *In OSI model, there are session and presentation layers above the transport layer. In TCP/IP model these two layers are absent. So, on top of the transport layer, the application layer is present in TCP/IP reference model. All the high level protocols are present here. TELNET, FTP, SMTP are some of the early inclusions. Later, other protocols such as DNS, NNTP, HTTP are added to this layer.*

3.7 Internet

The Internet can be defined as a network of globally connected computers that is decentralized by design. This definition can be broken down into three parts. Let's understand each part of the definition in isolation.

It is a network. *A network is a collection of computers. The Internet can also be referred to as a network because it is a collection of millions of computers.*

Globally connected computers. *This means that you can be connected to the Internet, regardless of your location. The Internet has brought people in the world closer by connecting computers located in the remotest of locations.*

Decentralized design. *The Internet has a decentralized design. That is, there is no centralized body that controls the way in which the Internet functions. The Internet does provide online services that are centrally administered, but as a whole, it would not be incorrect to say that the Internet has a decentralized design. Each computer connected to the Internet is called a host. The operator/ user of a particular host can choose from the millions of available Internet services and can also make services available through the Internet.*

You can consider Internet to have the following characteristics:

- *A complex network - with simplified definition as a 'network of network'*
- *Disorganized - Internet can be cumbersome and confusing, even for experienced users*
- *A decentralized system - millions of individual networks and over 200 million individual computers connected through the world*
- *Composed of many billions of files(web pages).*
- *Dynamic - changing every minute of every day. On an average, a new network is connected to the Internet every 30 minutes.*
- *Expanding exponentially - the Internet is growing at the rate not less than 15% per month.*

3.7.1 Internet Architecture

It is important to understand what the term "architecture" means. The notion of network architecture was introduced during the Internet research phase by the research community that had developed the ARPAnet protocols. Network architecture is a set of high-level design principles that guides the technical design of the network, especially the engineering of its protocols and algorithms. There are two most commonly used architecture in Internet technology: peer-to-peer and client server architecture.

3.7.1.1 Peer-to-Peer

Peer-to-peer is a communication model in which each and every node is capable of sharing information and can initiate a communication session. On the Internet, peer-to-peer (referred to as P2P) is a type of transient Internet network that allows a group of computer users with the same networking program to connect with each other and directly access files from one another's hard drives. ***Napster*** *and* ***Gnutella*** *are examples of this kind of peer-to-peer software.*

3.7.1.2 Client-Server

The Client-Server Architecture is based on the principle where the client computer requests for some data and the data are sent by the server computer through the network. The concept of client/server computing has particular importance on the Internet because most of the programmes are built using this design. Servers are powerful computers or processes dedicated to managing disk drives (file servers), printers (print servers), or network traffic (network servers). Clients are PCs or workstations on which users run applications. Clients rely on servers for resources, such as files, devices, and even processing power.

The following figure shows the two architectures:

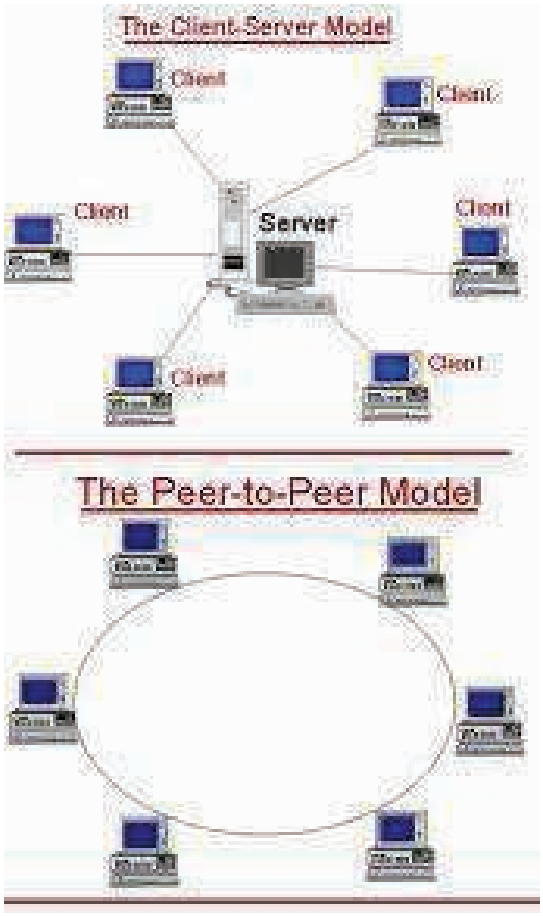

Fig. 3.12: Client Server and peer-to-peer architecture

3.8 Intranet

*An **Intranet** is a Web-based architecture used for managing internal information of an office, company, university or college or even a large library that runs on IP protocols. It means that an intranet is a private network that is contained within an enterprise.. It uses Internet protocols such as **TCP/IP** (Transmission Control Protocol/Internet Protocol), **HTTP** (Hypertext Transfer Protocol), **HTML** (Hyper Text Markup Language) **SMTP** (Small Message Transfer Protocol), **FTP** (File Transfer Protocol), and **AAA** (Access, Authorization, and Authentication). It may consist of many interlinked Local Area Networks and also use leased lines in the Wide Area Network. The main purpose of an intranet is to share information within the organization and computing resources among employees. An intranet can also be used to facilitate working in groups and for teleconferences.*

3.9 Network Devices

The internetworking devices are the vital tools for communication. Here, we will gain some information about the various hardware that is used to maintain connections between the networks.

- ***Network Interface Card (NIC) :** A **network interface card**, also known as a **network adapter**, is a very important hardware which is responsible for connecting a particular computer to a network. The NIC mediates between the computer(and its user) and the network. Usually, an NIC consists of a printed circuit boardcontaining different electronic components. It possesses a ROM chip that contains a unique number, known as the Media Access Controll (MAC) address. The MAC address identifies the device uniquely on the LAN. It also contains DMA (Direct Memory Access) circuit that allows the NIC to transmit or receive bits from memory without the involvement of the CPU. A NIC contains connector which provides plug-in facility to the network cable. It exists on the Data Link Layer of the ISO-OSI reference model. NIC handles all the details of packet transmission and reception.*

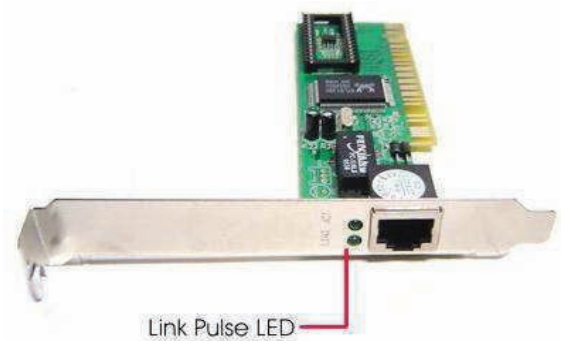

Fig. 3.13: A Network Interface Card

- ***Modem :*** *A modem (**mo**dulation-**dem**odulation) is a communication device that converts binary electrical signals into analog signals for transmission over telephone lines and converts these signals back into binary form at the receiving end. Conversion to analog form is called modulation; and the reverse process is known as demodulation.*

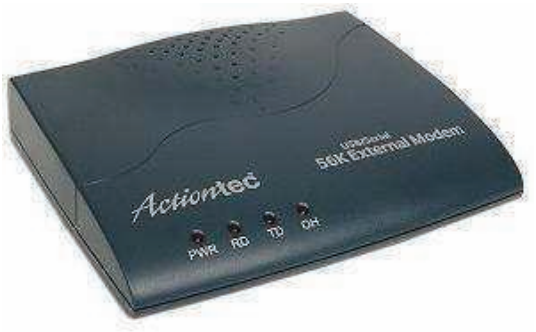

Fig. 3.14: A Modem

At the transmitting end, a bits stream arrives at the modem from computer's serial port. These bits are converted into analog form (modulation) using a predefined modulation technique. This analog signal is transmitted along telephone lines. At the receiving end, the analog signal is converted back to digital form(demodulation). The resulting serial stream is sent on to the computer at the receiving end.

- ***Repeater :*** *Signals can only travel so far through media before they weaken and become garbled. This weakening of signals is called attennuation.*

Repeaters are used to regenerate the analog as well as the digital signals which are distorted by transmission loss. These are Layer 1(Physical layer) internetworking devices used to combat attenuation. Repeaters take in weakened signals, clean them up, regenerates them, and send them on their way along the network. By using repeaters, the distance over which a network can operate is extended.

For example, 10Base-T (a wiring standard) is allowed to run 100 meters. One repeater can double this distance to 200 meters, as decpicted in Fig.3.15.

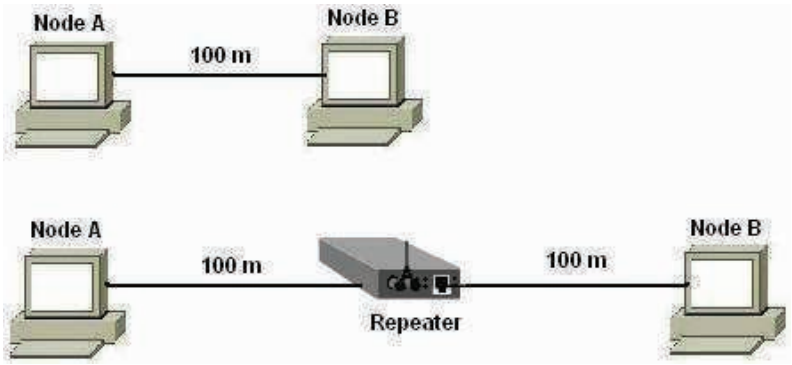

Fig. 3.15: A repeater extending network without attenuation

- ***Bridges and Gateways :*** *As the size of local area networks grew, the management of the networks became more difficult. Controlling traffic flow, maintaining fault monitoring, and adding and deleting stations from the system required more and more sophisticated system management. An alternate method to the computer at the receiving end.*

- ***Repeater :*** *Signals can only travel so far through media before they weaken and become garbled. This weakening of signals is called attennuation. Repeaters are used to regenerate the analog as well as the digital signals which are distorted by transmission loss. These are Layer 1(Physical layer) internetworking devices used to combat attenuation. Repeaters take in weakened signals, clean them up, regenerates them, and send them on their*

way along the network. By using repeaters, the distance over which a network can operate is extended.

For example, 10Base-T (a wiring standard) is allowed to run 100 meters. One repeater can double this distance to 200 meters, as decpicted in Fig.3.16.

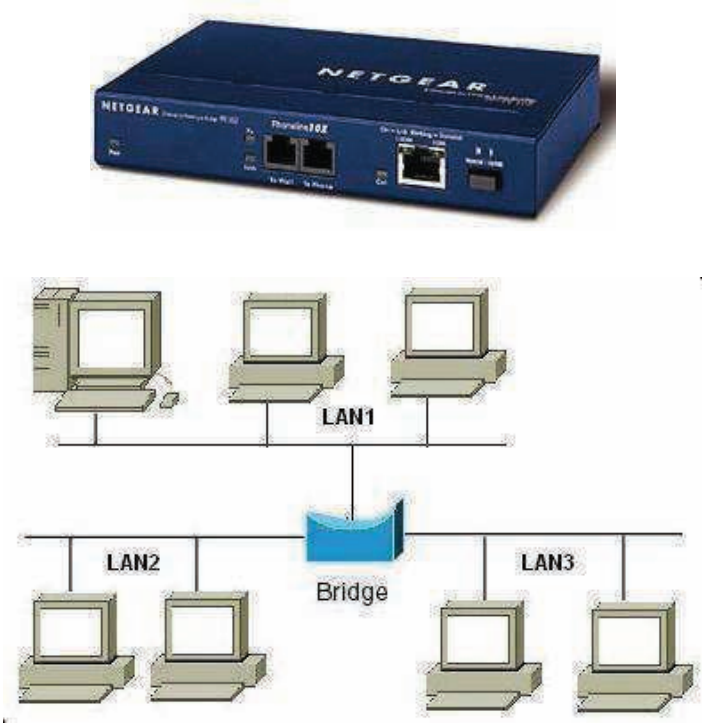

Fig. 3.16: A repeater extending network without attenuation

- ***Bridges and Gateways :*** *As the size of local area networks grew, the management of the networks became more difficult. Controlling traffic flow, maintaining fault monitoring, and adding and deleting stations from the system required more and more sophisticated system management. An alternate method to handling larger networks is to divide them into smaller networks that are interconnected using hardware/software systems called bridges and gateways.*

  Bridges *are mainly used to coonect multiple LANs which operate in the Data Link Layer to construct a larger LAN. The type of data transferred between the LANs must have the same format. A bridge can regenerate a signal it*

receives like a repeater. The basic function of a bridge is to transfer the frames from one LAN to another LAN. Bridge also has the capacity of frame filtering. It examines the header of each incloming frame frame and based on this information bridge decides whether to discard or forward this frame. Bridge stores the MAC address of received frame, keeps record on which port it has received, and which address belongs to which computer on the LAN.

Fig. 3.17(a) : A typical bridge

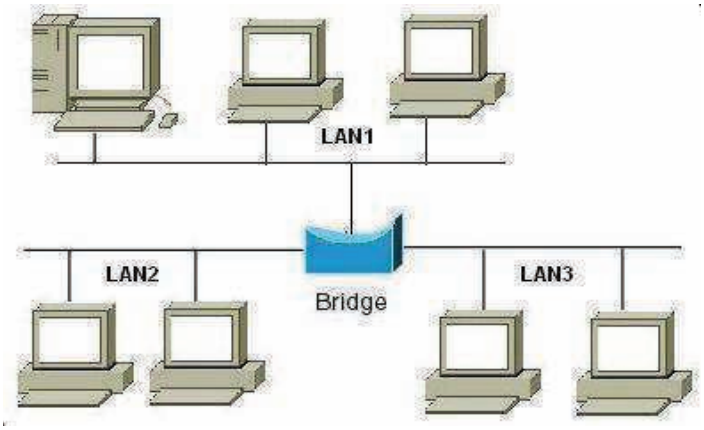

Fig. 3.17(b) : A bridge connecting 3 LANs

Gateways *are similar in functions to bridges in that they connect more than one networks. The main difference between bridges and gateways is that the latter can interconnect networks with differing message formats. A bridge does not perform any protocol translation in transferring packets between networks. whereas gateway takes an application message, reads it and interprets it. Hence, it can be used as a connecting device between two inter*

networks that use different model of references. One of them may use OSI model whereas the other uses TCP/IP model. A gateway is capable of receiving a frame coming from one model, makes necessary conversions and sends it to the other system of model. Thus, gateway relays packets among networks that have different protocols (e.g., between a LAN and a WAN). A gateway can provide security also.

- ***Hub :*** *Hubs are Physical layer(Layer 1) devices. Data that comes in one port is sent out to all other ports, except for the port it came in on. Hub does not read data passing through it and does not know the source and destination points. It simply receives the data packets, amplifes the electrical signal and then retransmit data packets on the network. Hubs usually accomodate four or eight nodes, and many hubs include connectors for linking to other hubs. A hub connects nodes that have a common network architecture.*

  The types of hubs that exist in the network are: Passive hub and active hub. Passive hub receives data packet and retransmits them on the network without amplifying the electrical signal. A multiport repeater is called an active hub. They receives the data packets, amplify them and retransmit on the network. For amplification they need the electrical signal.

Fig. 3.18 : A hub

- ***Switch :*** *Switches are the core devices in today's modern LANs. A network* ***switch*** *is a hardware device that is used to connect computers that require high bandwidth and is also used to connect hubs to form large network. Switches are more expensive but provide better performance to hubs. Unlike hubs, network switches are capable of inspecting data packets as they are received, determining the source and destination device of each packet, and*

forwarding them appropriately. By delivering messages only to the connected device intended, a network switch conserves network bandwidth and offers generally better performance than a hub.

Fig. 3.19 : A Switch

Two types of switches exist, which are :

Layer- 2 switches : These types of switches are based on the bridging technology. They establish connection between the ports based on the MAC address stored in the address table.

Layer-3 switches : These types of switches are based on the routing technology and are responsible for seting up connection between the ports based on the network address.

- ***Router :*** *A router is the primary device responsible for connecting multiple networks together. The function of a router is to provide a path from a node on one network to a node on another network. The two networks may be separated by several intervening networks and possibly, by many miles. The purpose of a router is to forward data packets between networks. It examine incoming packets, chooses the best path for them through the network, and then switches them to the proper outgoing port. Routers are normally used to connect LANs and WANs in the Internet. It performs routing of packets based on their logical addresses. It has a routing table to take decision about the routes of packets. The table is constructed and updated dynamically by following some routing algorithm. Routers work at Layer 3, the Network Layer. In the Fig. 3.20(b), routers connect nodes on different networks.*

Fig. 3.20(a) : Some images of Cisco routers

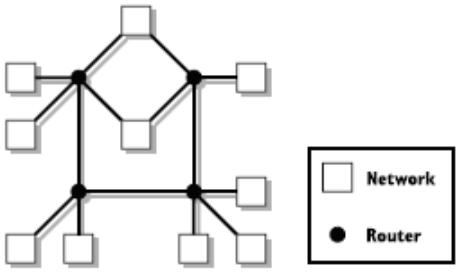

Fig. 3.20(b) : Networks connected with routers

3.10 Firewall

A firewall is a device or set of devices designed to permit or deny network transmissions based upon a set of rules and is frequently used to protect networks from unauthorized access while permitting legitimate communications to pass.

Many personal computer operating systems include software-based firewalls to protect against threats from the public Internet. Many routers that pass data between networks contain firewall components and, conversely, many firewalls can perform basic routing functions

3.11 Suggested readings

1. *Data and Computer Communications by William Stallings, Pearson Education.*

2. *Data Communications and networking by Behrouz A. Forouzan, Tata McGrawHill.*

3. *"Web technologies: a computer science perspective", Jeffrey C. Jack son, Pearson Prentice Hall.*

4. *"Web Technology : A Developer S Perspective", Gopalan, Gopalan/akilandeswari, Pearson Prentice Hall.*

3.12 Terminal questions

1. *There are many application areas of Internet. Identify four of them and describe two of them.*

2. *What is Internet? Explain why Internet is called a 'network of networks'?*

3. *Define computer network. What are the goals of computer networks?*

4. *Describe ISO-OSI reference model of computer network.*

***

UNIT 4: DATABASE MANAGEMENT SYSTEMS

4.1 Introduction

Data is one of the most important components in the information systems. In the early days data used to maintain using the traditional or manual record keeping system. This is also sometimes called as paper based record keeping system. However, there are some drawbacks in this manual record keeping system in terms of data handling and selection. In early days of 1960s the computer applications were developed directly on top of the file systems. Again there are some drawbacks on these file systems, such as multiple file formats and duplication of information in different files. After that it is realized that the purpose of any database is to meet the requirement of the users in terms of data storage, data access.

Now the database is a repository of a structured data to store its attributes in a proper way. Database Management System (DBMS) it is nothing but a computer-based record keeping system: that is, a system whose overall purpose is to record and maintain information. The information concerned can be anything that is deemed to be of significance to the organization the system is serving anything, in other words, that may be necessary to the decision-making processes involved in the management of that organization.

In this chapter we will discuss each one in rather detail.

4.2 Objectives

After reading this unit the student will be able to:

1. *Define Database concept*
2. *Define architecture of a database system*
3. *Understand principles of database*
4. *Discuss normalization*
5. *Understand structure query language*

4.3 DBMS

The database management system is the interface between the users (application programmers) and the database (the data). A database management system (DBMS) is a program that allows user to define, manipulate and process the data in a database, in order to produce meaningful information.

A DBMS is a set of software programs that controls the organization, storage, management, and retrieval of data in a database. It is a set of pre-written programs that are used to store, update and retrieve a database. The DBMS accepts requests for data from the application program and instructs the operating system to transfer the appropriate data. The followings are the examples of DBMS software :

Microsoft Visual FoxPro, MonetDB, MySQL , Oracle Database, PostgreSQL, SQL Anywhere, SQLite, FileMaker, IBM DB2, IBM UniVerse, Firebird, Microsoft Access, Microsoft SQL Server etc.

The following are examples of database applications:

- *Reservation systems, banking systems*
- *Record/book keeping (corporate, university, medical), statistics*
- *Bioinformatics, e.g., gene databases*
- *Criminal justice, e.g. fingerprint matching*
- *Multimedia systems, e.g. image/audio/video retrieval*
- *Satellite imaging; require petabytes (10^{15} bytes) of storage*
- *The web*
- *Data mining (Knowledge Discovery in Databases) etc.*

To complete our initial definitions, we will call the database and DBMS software together as a database system. The following figure depicts a database system.

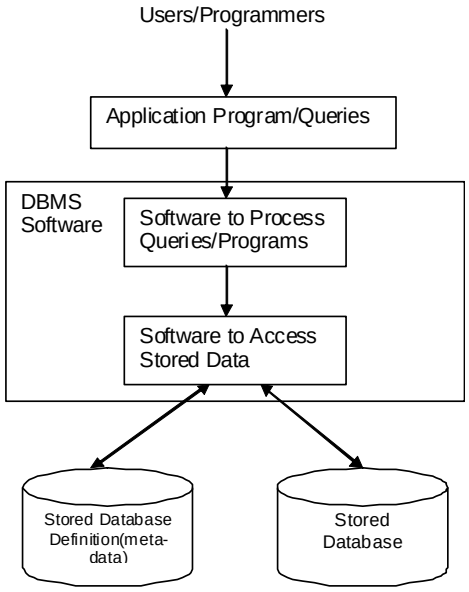

Fig. A simplified database system

The purpose of the database system is:

***Redundancy can be reduced** – duplication of records in different database can be avoided by normalizing the database.*

***Inconsistency can be avoided** – the data can be entered to represent a valid record with proper updating mechanisms without the duplication of records.*

***Data can be shared** – the existing data can be shared and also the database also can be shared to develop new applications*

***Standards can be enforced** – since data is centrally designed and maintained, all standards in the data are followed in the representation of data.*

***Security restrictions can be applied** – the database can be enforced with authorization privileges to its users.*

***Integrity can be maintained** – Number of rules and regulations can be applied on to the database for accurate data in the database.*

Conflicting requirements can be balanced *– the database can be structured to provide overall access to the organization.*

Database systems offer convenient solutions to overcome the above problems. The system takes care of the data part from the design phase to selection phase for effectively maintaining the data of any organization.

4.4 Architecture of a database system

The architecture is divided into three general levels:

1. *Internal level*
2. *Conceptual level*
3. *External level*

Internal level *is the one closest to physical storage, that is, the one concerned with the way in which the data is actually stored.*

Conceptual level *describes the logical structure of an entire database, including descriptions of the data and relationships among the data.*

External level *is the users view to the database. Depending on their needs, different users access different parts of the database.*

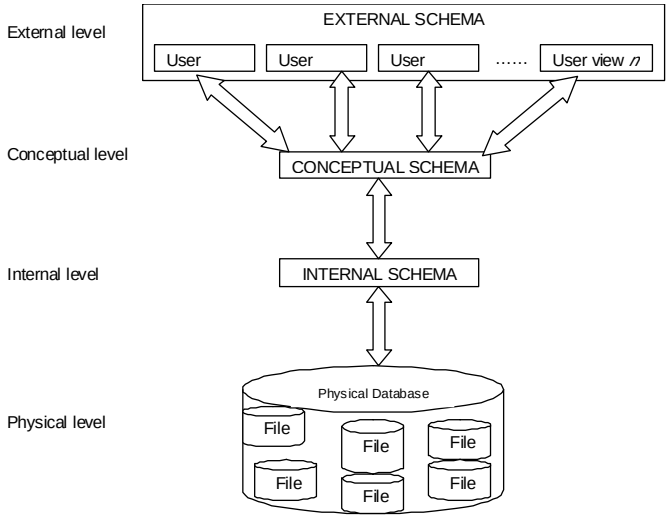

Fig.: Three level view of Database architecture

4.5 TYPES OF DBMS

The modern business environment revolves around the accuracy and integrity of information. The advancements in computer technology and rapid development of graphical user interface (GUI)-based applications, networking and communications have resulted in new dimensions in database management systems. A DBMS can be classified according to the number of users, the database site locations and the type and extent of use.

On the basis of number of users :

- *Single-user DBMS*
- *Multi-user DBMS*

On the basis of site locations :

- *Centralised DBMS*
- *Parallel DBMS*
- *Distributed DBMS*
- *Client/Server DBMS*

On the basis of the type and extent of use

- *Transactional DBMS*
- *Decision support DBMS*
- *Data Warehouse*

4.5.1 Centralized DBMS

The centralized database system consists of a single computer system associated with its peripherals, physically located in a single location. The computer system offers data processing facilities to the users located either at the same site, or, at geographically dispersed sites, through remote terminals. The management of the system and its data are controlled centrally from any one or central site. The following figure shows a centralized database system.

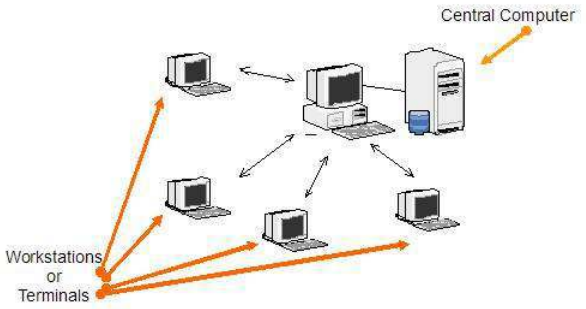

Fig. Centralized DBMS

Advantage of such system is given below:

- *Most of the functions such as update, backup, query, control access etc. are easier with this system.*
- *Single database is shared across the several different users.*

Of course, when the central site computer goes down, then every user is blocked from using the system until it recovers.

4.5.2 Parallel Database System

Parallel database systems architecture consists of a multiple central processing units (CPU) and data storage disk in parallel as shown in the figure. Hence, in such a system data processing speed is fast as well as input/output speed is also fast. The system, in which it needs to process an extremely large number of transactions per second, in such a system parallel database system is used.

Advantage of such sytem is given below :

- *Useful in the applications, which have to process an extremely large number of transactions per seconds (of the order of thousands of transactions per seconds)*
- *Performance of such database system is very high.*

The following figure shows a parallel database system.

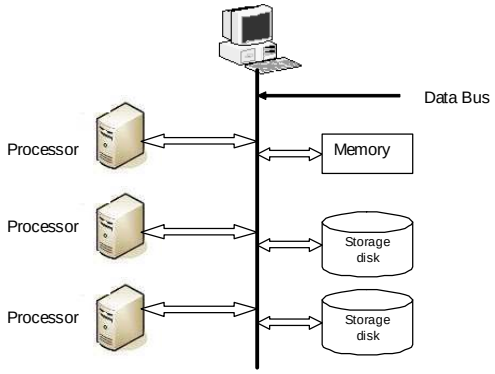

Fig. Parallel database system

4.5.3 Distributed DBMS

In a distributed database system, data are spread across a variety of different databases. These are managed by a variety of different DBMS softwares running on a variety of different computing machines having different operating systems. These machines are actlually located on different sites and connected with some kind of communication networks as shown in the figure below. Thus, in a distributed database system, the organizations data might be distributed on different computers in such a way that data for one portion (or department) of the organization is stored in one computer and data for another department is stored in another computer. Each machine can have own data and applications and users of one machine can access the data of several other computers.

The following figure shows a distributed database system.

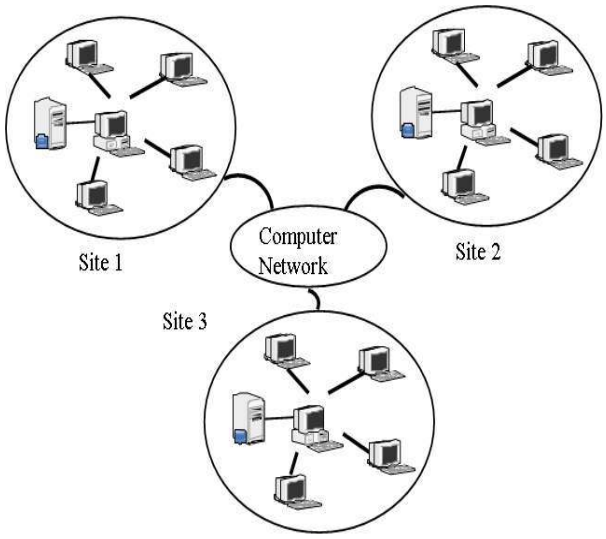

Fig. Distributed database

Advantage of such sytem is given below :

- *efficiency and performancce of this system is high.*
- *a single database can be shared across several distinct client systems.*
- *As data volumes and transaction rates increase, users can grow the system incrementally.*

4.5.4 Client-Server DBMS

The client-server database system has two logical components namely - client and server. Clients are generally personal computers or workstations and the servers are the large workstations or mainframe computer system. The applications and tools of DBMS run on one or more client plateforms, while the DBMS softwares reside on the server. The server computer is called backend and the client computer is called front-end. The server and the clients are connected through networks. The clients send request to the server for performing some special tasks. The DBMS in the server side, in turn, process these requests and returns the results to the clients. The server handles parts of the job that are common to many clients, for example, database access and updates.

The following figure shows a client-server database model.

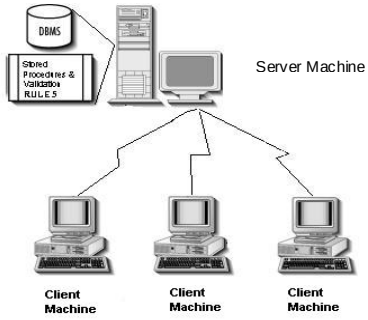

Fig. Client-Server database model

Advantage of such sytem is given below :

- *Performance is high.*
- *A single database (on server) can be shared accross several distinct client system.*
- *More flexible as compared to the centralised system.*
- *Facilitates in more productive work by the users and making better use of existing data.*

4.6 Data Models

Database model are designed to use one of database structures to provide simplistic access to information stored in databases. The main database structures are:

- *Relational data model*
- *Network data model*

4.6.1 Relational data model

This is the most widely used data base model that represents data as well as relationship among data in the form of tables. It looks like a file. Constraints are stored in a meta-data table. This is a very simple model and is based on a proven

mathematical theory. Most relational data bases have high-level query languages and support a limited form of user views. Usually, the conceptual and internal schemas are not distinguishable, and a single Data Definition Language is used to describe all aspects of the database structure. Figure 4.2 shows the relational representation of student database.

Table: STUDENT

Name	*Roll_No*	*Class*	*DOB*
Raj	*1*	*12*	*12/01/1990*
Ramesh	*2*	*12*	*103/04/1991*

Fig. : Relational representation of student data base

Advantages of Relational data model:

1. *The relational data model can be implemented with a personal computer having limited main memory and processing capability.*
2. *Very effective for small databases.*
3. *Much easier to use because it enables a computer system to accommodate a variety of enquiries in an efficient manner.*
4. *Very easy to represent the logical relationship among the data items since it use primary key to represent record relationships instead of pointers.*
5. *Relational model is very useful for representing most of the real world objects and the relationships among them.*

Disadvantages of Relational data model:

In this model, as the size of the database increases, several problems may come into existence – system slowdown, performance degradation and data corruption.

4.6.2 Network data model

In this model, data is represented as records and relationship as links. Figure 3.2 shows the network representation for the database STUDENT with grade, course and section. In the figure, record types are shown as rectangles and set types are shown as labeled directed arcs. The network model is also known as the Computer Data System Language Data Base Task Group (CODASYL DBTG) model. It has an associated record-at-a-time language that must be embedded in a host programming language.

Advantages of network model:

1. *Useful to represent the records having many-to-many relationships.*
2. *Problem of inconsistency does not exist in this model because a data element is physically located at just one place.*
3. *Searching a record is easy since there are multiple access paths to a data element.*

4.7 INTEGRITY CONSTRAINTS

*The term **integrity** refers to the accuracy or correctness of data in the database schema and is expected to hold on every database instance of that schema. Relational model includes two general integrity constraints. They are:*

4.7.1 Entity Integrity Constraints (Rule 1)

Entity Integrity Constraints states that no primary key value can be NULL. This is because we use the primary key value to identify individual tuples in a relation. It ensures that instances of the entities are distinguishable i.e., they must have a unique identification of some kind. Primary keys perform that unique identification function in a relational database.

4.4.2 Referential Integrity Constraints (Rule 2)

Referential Integrity Constraint is specified between two relations and is used to maintain the consistency among tuples of the two relations (not necessarily be distinct). It uses a concept of foreign key which will be explained more details in the next unit. Informally, it states that a tuple in one relation that refers to another relation must refer to an existing tuple in that relation. Considering the following relations,

EMPLOYEE

PRIMARY KEY *FORIGN KEY*

ENO	***ENAME***	***DNO***
101	*Robert*	*10*
102	*Smith*	*12*
103	*Robindra*	*12*
104	*John*	*10*

DEPARMENT

PRIMARY KEY

DNO	***DNAME***	***LOCATION***
10	***Comp. Sc.***	*Jalukbari*
12	***Electronic Sc.***	*Guwahati*

Fig. : relational database table showing referential integrity

In the above figure EMPLOYEE and DEPARTMENT are two relations where ENO and DNO are primary keys respectively. Here the attribute DNO of EMPLOYEE table is a foreign key that gives the department number for which each employee works. Hence its value in each EMPLOYEE tuple must match the DNO value of some tuple in the DEPARTMENT relation.

4.8 DOMAIN CONSTRAINTS

It specifies that each attribute in a relation must contain an atomic value only from the corresponding domains. The data types for commercial RDBMS domains are:

- *Standard numeric data types for integer*
- *Real numbers*
- *Characters*
- *Fixed length and variable length strings*

Thus, domain constraint specifies the condition that we want to put on each instance of the relation. So, the values that appear in each column must be drawn from the domain associated with that column.

4.9 THE CODD COMMANDMENTS

There are twelve (12) rules formulated by E.F. Codd for RDBMS in 1970 to define the requirements more rigorously within a single product. In reality it is true to say that they do not all carry the same degree of importance, but can be obtained a good result if an RDBMS satisfies all these twelve rules. The rules are:

Rule 1: The information rule

All information is explicitly and logically represented in exactly one way – by data values in tables. In simple terms this means that if an item of data does not reside somewhere in a table in the database then it does not exist and this should be extended to the point where even such information as table, view and column names to mention just a few, should be contained somewhere in table form.

Rule 2: The rule of guaranteed access

Every item of data must be logically addressable by resorting to a combination of table name, primary key value and column name. For a table like storage structure, this rule says that at the insertion of a column and row it is necessarily find one value of a data item or null.

Rule 3: Systematic treatment of null values

In DBMS NULL values are supported in the representation of missing and inapplicable information. This support for null values must be consistent throughout the DBMS and independent of all data type.

Rule 4: Database description rule

The description of the database is held and maintained using the same logical structures used to define the data, thus allowing users with appropriate authority to query such information in the same way and using the same languages as they would any other data in the database. It implies that there must be a data dictionary within the RDBMS that is constructed of tables and/or views that can be examined using SQL. Therefore a dictionary is mandatory for RDBMS.

Rule 5: Comprehensive sub-language rule

There must be at least one language whose statements can be expressed as character strings conforming to some well-defined syntax. In real terms, the RDBMS must be completely manageable through its own extension of SQL.

Rule 6: View updating rule

All views that can be defined using combination of base tables, and theoretically updatable, must also be capable of being updated by the system. This is quite a difficult rule to interpret and with all sorts of aggregates and virtual columns, it is obviously not possible to update through some of them.

Rule 7: Insert and update rule

An RDBMS do more than just be able to retrieve relational data sets. It has to be capable of inserting, updating and deleting data as a relational set.

Rule 8: Physical independence rule

User access to the database, via monitors and application programs, must remain logically consistent whenever changes to the storage representation, or access methods to the data, are changed. For example, if an index is built and destroyed by the DBA on a table, any user should still retrieve the same data from that table.

Rule 9: Logical data independence

Application programs must be independent of changes made to the base tables. This allows many types of database design change to be made dynamically, without users being aware of them.

Rule 10: Integrity rule

The relational model includes two general integrity rules which we have discussed in already in this unit. These integrity rule implicitly or explicitly define the set of consistent database states, or changes of state, or both. Other integrity constraints can be specified during database design.

Rule 11: Distribution rule

An RDBMS must have distribution independence. Thus, RDBMS package must make it possible for the database to be distributed across multiple computers even though they are having heterogeneous platforms both for hardware and operating system.

This is one of the most attractive aspects of RDBMSs, database system built on the relational framework are well suited to today's client/server database design.

Rule 12: No subversion rule

If an RDBMS supports a lower level language that permits for example, row at-a-time processing, then this language must not be able to bypass any integrity rules or constraints of the relational language.

4.10 KEYS

In a relational model, a database consists of relations (tables), which consists of tuples (or records/rows), which further consist of attributes (or fields/columns). We must have a way to specify how tuples within a relation are distinguised. Each relation in a relational database must have an attribute or combination of attributes such that they can uniquely identify the tuple. This unique identifier is called ***key****. A key is that data item that exclusively identifies a record or tuple. It may consists of one or more attributes. We can split related data into different relations or tables and logically linked them together with the help of keys. Without this unique identifier, there is no way to retrieve the unique tuple from a relation.*

For example, let us consider the following relation (table). In this unit we may use the terminologies table, row or record and field in place of relation, tuple and attribute respecively.

STUDENT

Roll_no	*Name*	*Marks*	*Grade*
1	*Monirupa Misra*	*360*	***A***
2	*Ranjita Dutta*	*180*	*C*
3	*Rajib Sharma*	*310*	*A*
4	*Kaustab Baruah*	*265*	*B*
5	*Apurba Bora*	*310*	*A*
6	*Rajib Sharma*	*210*	*B*

Table: Student

4.10.1 DIFFERENT TYPES OF KEYS

Every key which has the property of uniqueness can be distinguished as follows :

- *Super Key*
- *Candidate Key*
- *Primary Key*
- *Alternate Key*
- *Composite Key*
- *Foreign Key*

4.10.1.1 Super Key

A superkey is a set of columns that uniquely identifies every row in a table. For example, if there is a table ***STUDENT*** *with only two columns* ***Roll_no*** *and* ***Name****, then the super key will be if we assume that there are no two student in the class with the same Roll_no as well as Name.*

{ Roll_no, Name}

4.10.1.2 Candidate Key

A table can have more than one columns that could be chosen as the key because they individually have the capability to identify a record uniquely. These fields are termed as ***candidate keys****. In other words, a candidate key is any set of one or more columns whose combined values are unique among all occurrences (i.e., tuples or rows or record). Since a NULL value is not guaranteed to be unique, no component of a candidate key is allowed to be NULL. Candidate keys are those attributes of a relation, which have the properties of uniqueness and irreducibility. This two properties are explained below:*

Let K be a set of attributes of relation R. Then K is a candidate key for R if and only if it possesses both of the following properties:

Uniqueness: *No legal value of R ever contains two distinct tuples with the same value for K.*

Irreducibility: *No proper subset of K has the uniqueness property.*

Let us consider the following relation ***EMP_INFO*** *containing some personal information of employees working in an office. Suppose all of them have passport.*

EMP_INFO

Emp_ID	***Name***	***Passport_no***	***Blood Group***
12341	*Kunal Kashyap*	*M 9523421*	*A+*
12342	*Rajib Sharma*	*M 9515212*	*O+*
12343	*Ankur Chakraborty*	*F 9515456*	*O+*
12344	*Niharika Bora*	*F 9643521*	*B+*

Table: EMP_INFO

The attribute ***Emp_ID*** *and* ***Passport_no*** *posseses unique data item for each employee. Therefore, any of these two attribute can be chosen as the key. These two are examples of candidate keys in the above relation. The attribute* ***Name*** *cannot be a candidate key as more than one employee might have identical name. Similary, several employees might have same blood group. So* ***Blood Group*** *cannot be chosen as key.*

4.10.1.3 Primary Key

Every database table should have one or more columns designated as the primary key. The value this key holds should be unique for each record in the database. In a database, there can be multiple candidate keys. Out of all the available candidate keys, a database designer can identify a primary key. The primary key should be chosen such that its attributes are never or very rarely changed.

A ***primary key*** *is a field or combination of fields that uniquely identify a record in a table, so that an individual record can be located without confusion. Depending on its design, a table or relation may have arbitrarily many unique keys but at most*

one primary key. For example, let us assume we have a table called ***EMPLOYEE_ADDRESS*** *that contains some information for every employee in an organization. We should need to select an appropriate primary key that would uniquely identify each employee. Our first thought might be to use the employee's name i.e,* ***Emp_Name****. But this would not work properly because two or more employees with the same name might be possible in the organization. The* ***Location*** *field of a person cannot be chosen as primary key since it is likely to change. A better choice might be to use a unique* ***Emp_ID*** *number that the organization assign to each employee when they are appointed.* ***Emp_ID*** *can be a primary key as it does not changed till the person is working in the same organization.*

EMPLOYEE_ADDRESS

Primary Key

Emp_ID	***Emp_Name***	***Location***
1231	*Gautam Baruah*	*GS Road Guwahati*
1232	*Arindam Dutta*	*RG Barua Road*
1233	*Meghali Gogoi*	*Chandmari Guwahati*
1235	*Arindam Dutta*	*Guwahati*

Table: EMPLOYEE_ADDRESS

In the table 5.1., student's ***Roll_no*** *would be a good choice for a primary key in the* ***STUDENTS*** *table. The student's name would not be a good choice, as there is always the chance that more than one student with same name. Some other examples of primary*

*keys are **Social Security Numbers** (associated with a specific person) , **ISBN_no** (associated with a specific book).*

*A primary key is a special case of unique keys. Unique key constraint is used to prevent the duplication of key values within the rows of a table and allow NULL values. Primary key allows each row in a table to be uniquely identified and ensures that no duplicate rows exist and no NULL values are entered. Thus **primary key constraint** can be defined as a rule that says that the primary key fields cannot be NULL and cannot contain duplicate data.*

Once we decide upon a primary key and set it up in the database, the database management system (DBMS) will enforce the uniqueness of the key. If we try to insert a record into a table with a primary key that duplicates an existing record, the insert will fail. Sometimes, a table just does not have a primary key. In such cases, we may need to introduce an additional column which contains unique values. Most databases are also capable of generating their own primary keys. Microsoft Access, for example, may be configured to use the AutoNumber data type to assign a unique ID to each record in the table. While effective, this is a bad design practice because it leaves us with a meaningless value in each record in the table. It is better to use that space by storing some useful data.

Properties of Primary Key

To qualify as a primary key for an entity, an attribute must have the following properties:

Stable:

The value of a primary key must not change or should not become NULL throughout the file of an entity. A stable primary key helps to keep the model stable. For example, if we consider a patient record, the value for the primary key (Patient number) must not change with time as would happen with the age field.

Minimal:

The primary key should be composed of the minimum number of fields that ensures the occurrences are unique.

Definitive:

A value must exist for every record at creation time. Because an entity occurrence cannot be substantiated unless the primary key value also exists.

Accessible:

Anyone who wants to create, read or delete a record must be able to see the Primary key value.

4.10.1.4 Alternate Key

As we have seen, it is possible for a relation to have two or more candidate keys. If we chose any one of them as primary key, then the remaining keys will be termerd as alternate key. The alternate key (or secondary key) is any candidate key which is not selected to be the primary key. For the illustration of alternate key, let us consider the following table ELEMENT which stores some information like element name, symbol, atomic number of the elements of periodic table.

ELEMENT

Name	***Symbol***	***Atomic_no***
Hydrogen	*H*	*1*
Helium	*He*	*2*
Lithium	*Li*	*3*
Berylium	*Be*	*4*

Table: ELEMENT

All the three fields can individually identify each element in the table. So any of these three fields can be chosen as the primary key . If we choose ***Symbol*** *as the primary key;* ***Name*** *and* ***Atomic_no*** *would then be alternate keys. Similarly, in the* ***EMP_INFO*** *(table 5.3), if we consider* ***Emp_ID*** *as the primary key then* ***Passport_no*** *will be the alternate key.*

4.10.1.5 Composite Key

In some situations, while designing a database, there may not be a particular column or field that can individually identify a record uniquely in a table. In such cases, we may require to select two or more fields so that combination of those can identify each record uniquely. These combination of fields is known as ***composite key****. It is used when a record cannot be uniquely identified by a single field.*

For the illustration of composite key, let us consider the following table ***ITEM*** *with the fields* ***Supplier_ID, Item_ID, Item_Name*** *and* ***Quantity****. This table gives us the information which supplier sells which item. As we can see, any of these fields indivisually cannot identify a row in the table uniquely. But if we combine* ***Supplier_ID*** *and* ***Item_ID****, then these together can easily identify any row in the table uniquely. Thus,* ***Supplier_ID*** *and* ***Item_ID*** *together becomes a composite key.*

ITEM

Supplier_ID	***Item_ID***	***Item_Name***	***Quantity***
S_1	I_1	*AC*	*5*
S_1	I_2	*Inverter*	*8*
S_2	I_2	*Inverter*	*4*
S_2	I_3	*UPS*	*15*

Table: ITEM

4.10.1.6 Foreign Keys

One important type of key that we will discuss in this unit is the ***foreign key****. These keys are used to create relationships between tables.*

A foreign key is a field in a relational table that matches the primary key column of another table. It identifies a column or a set of columns in one (referencing) table that refers to a column or set of columns in another (referenced) table. The columns in the referencing table must be the primary key or other candidate key in the

referenced table. The values in one row of the referencing columns must occur in a single row in the referenced table. Thus, a row in the referencing table cannot contain values that donnot exist in the referenced table. This way references can be made to link information together and it is an essential part of database normalization. Multiple rows in the referencing table may refer to the same row in the referenced table.

For example in an employees database, let us imagine that we wanted to add a table ***DEPARTMENT*** *containing departmental information to the database. We would also want to include information about the employees in the department, but it would be redundant to have the same information in two tables (****EMPLOYEE*** *and* ***DEPARTMENT****). Instead, we can create a relationship between the two tables. Let us assume that the* ***DEPARTMENT*** *table uses the* ***Department_Name*** *column as the primary key. To create a relationship between the two tables, we add a new column to the* ***EMPLOYEE*** *table called* ***Department_Name****. We then fill in the name of the department to which each employee belongs. The* ***Department_Name*** *column in the* ***EMPLOYEE*** *table is a foreign key (FK) that references the* ***DEPARTMENT*** *table. The database will then enforce referential integrity by ensuring that all of the values in the* ***Department*** *column of the* ***EMPLOYEES*** *table have corresponding entries in the* ***DEPARTMENT*** *table.*

EMPLOYEE

PK *FK*

Name	***Post***	***Department_Name***
Goutam Bora	*Accountant*	*Sales*
Manash Saikia	*Manager*	*Marketing*
Himangshu Das	*Financial Officer*	*Human Resource*
Niharika Baruah	*Accountant*	*Production*
Pranjal Hazarika	*Law Officer*	*Human Resource*

Table: EMPLOYEE

DEPARTMENT

PK

Department_Name	***Manager***	***Dept_Code***
Human Resource	*Manash Saikia*	*D1*
Marketing	*Dhruba Sharma*	*D2*
Production	*Diganata Bora*	*D3*

Table: DEPARTMENT

*Again, let us consider a book database. The **BOOKS** table has a link to the **PUBLISHERS** table. The **Pub_ID** column is the primary key for the **PUBLISHERS** table and **ISBN_no** is the primary key for the **BOOKS** table. The **BOOKS** table also contains a **Pub_ID** column which matches the primary key column of the **PUBLISHERS** table. This **Pub_ID** is the foreign key in the **BOOKS** table. The **Pub_ID** field in the **BOOKS** table indicates which publisher a book belongs to.*

PUBLISHERS

PK

Pub_ID	*Pub_Name*	*City*	*State*

Table: PUBLISHERS

BOOKS

PK *FK*

ISBN_no	*Book_name*	*Author_name*	*Pub_ID*	*Pub_date*	*Price*

Table: BOOKS

*Although the primary purpose of a **foreign key constraint** is to control the data that can be stored in the foreign key table, it also controls changes to data in the primary key*

*table. For example, if the row for a publisher is deleted from the **PUBLISHERS** table, and the publisher's ID is used for books in the **BOOKS** table, the relational integrity between the two tables is broken; the deleted publisher's books are orphaned in the **BOOKS** table without a link to the data in the **PUBLISHERS** table. A foreign key constraint prevents this situation. The constraint enforces referential integrity by ensuring that changes cannot be made to data in the primary key table if those changes invalidate the link to data in the foreign key table. If an attempt is made to delete the row in a primary key table or to change a primary key value, the action will fail if the deleted or changed primary key value corresponds to a value in the foriegn key constraint of another table. To change or delete a row in a foreign key constraint successfully, we must first either delete the foreign key data in the foreign key table or change the foreign key data in the foreign key table, thereby linking the foreign key to different primary key data. i.e., a primary key constraint cannot be deleted if referenced by a foreign key constraint in another table; the foreign key constraint must be deleted first.*

4.11 NORMAL FORMS

Whenever the simple rules of functional dependencies are applied to a relations, it transforms the relations into a state which called normal form. The normal forms are used to ensure that various types of anomalies and inconsistencies are not introduced into the database. Various types of normal forms are used in relational data base, they are :

- *First normal form (1 NF)*
- *Second normal form (2 NF)*
- *Third normal form (3 NF)*

4.11.1 First Normal Form (1NF)

A relation is said to be in first normal form if the values in the domain of each attribute of the relation are atomic. The first normal form prohibits multivalued attributes, composite attributes and their combinations. It means that, 1NF

disallows having a set of values, a tuple of values, or combination of both as an attribute value for a single tuple.

Let us consider the relation TRAVEL_INFO as shown in the figure.

PERSON	*VISITED_CITY*	
HEMANGA	*CITY*	*JOURNEY_DATE*
	DELHI	*04-07-02*
	MUMBAI	*15-09-03*
	AGRA	*02-03-04*
BIKASH	*CITY*	*JOURNEY_DATE*
	CHENNAI	*05-01-02*
	KOLKATA	*13-02-03*
	PUNE	*21-08-04*

Un-normalized relation TRAVEL_INFO

Here, in the relation the domain VISITED_CITY is not simple. Hence, the relation is un-normalized. Now, let us combine the respective rows in VISITED_CITY with the value of the attribue PERSON and the resultant relation is shown below -

PERSON	*CITY*	*JOURNEY_DATE*
HEMANGA	*DELHI*	*04-07-02*
HEMANGA	*MUMBAI*	*15-09-03*
HEMANGA	*AGRA*	*02-03-04*
BIKASH	*CHENNAI*	*05-01-02*
BIKASH	*KOLKATA*	*13-02-03*
BIKASH	*PUNE*	*21-08-04*

Table Modified relation TRAVEL_INFO

Let us consider another relation PATIENT_DOCTOR, which keeps the records of appointment details bhetween patient and doctors. This relation is in 1NF. The relational table can be depicted as -

PATIENT_DOCTOR *(P_NAME, DOB, D_NAME, DATE_TIME, PHONE, DURATION)*

P__NAME	DOB	D__NAME	PHONE	DATE__TIME	DURATION
MUKUL	10-02-1975	Dr Deepak	0361-223470	10-01-05 10:00	15 (minutes)
RUHIT	27-01-1971	Dr Kalyan	0361-270512	10-01-05 11:00	10
JAMES	30-03-1977	Dr Dhiraj	0361-270615	10-01-05 10:30	10
JAMES	30-03-1977	Dr Dhiraj	0361-270615	10-01-05 09:00	10
RUHIT	27-01-1971	Dr Kalyan	0361-223470	10-01-05 10:15	15
MUKUL	10-02-1975	Dr Dhiraj	0361-270615	10-01-05 10:50	20
RANA	02-11-1980	Dr Kalyan	0361-270512	10-01-05 11:10	20
MUKUL	10-02-1975	Dr Deepak	0361-223470	06-02-05 16:00	15

Table Relation PATIENT_DOCTOR

From the relational table we have observed that a doctor cannot have two simultaneous appointments so D_NAME and DATE_TIME is a compolsite key. Similarly, a patient cannot have same time from two different doctors. Therefore, P_NAME and DATE_TIME attributes are also a candidate key.

Problems in 1 NF :

1 NF *contains redundant information. In our example, PATIENT_DOCTOR relation has the following errors :*

a) *There exists redundant information in patients date of birth and phone number.*

b) *A doctor, who does not currently have an appointment with a patient, cannot be represented.*

c) *A patient, who does not currently have an appointment with a doctor, cannot be represented.*

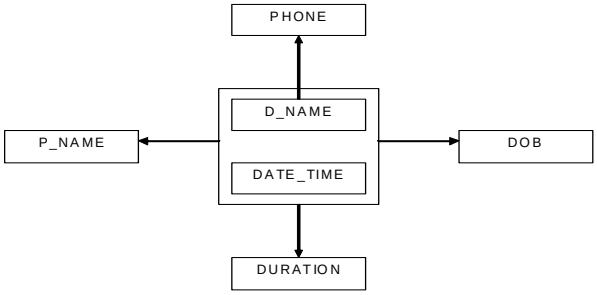

Fig. FDD of relation PATIENT_DOCTOR

Functional dependency diagram of the relation PATIENT_DOCTOR is shown. Here, P_NAME, DURATION, DOB are dependent on the key D_NAME & DATE_TIME and doctor's contact number i.e. PHONE is only dependent on the D_NAME.

4.11.2 Second Normal Form (2NF)

A relation or table is said to be in second normal form (2NF) if and only if -

a) it is in 1NF,

b) each non-primary key attribute is fully functionally dependent on primary key of that relation.

2NF is an intermediate step towards higher normal forms. 2NF is based on the concept of full functional dependency. It eliminates the problems of 1NF.

So, we come to know that, no attributes of the relation (or table) should be functionally dependent on only one part of a concatenated primary key. In our example, we have seen from the functional dependancy diagram that, PHONE is partiallly dependent only on D_NAME, for which the relation is not in 2NF.

Therefore, to bring the relation into 2NF, the information about doctor and their contact numbers have to be separated from information about patient and their appointments with doctors. Thus, the relation is decomposed into two tables,

namely PATIENT_DOCTOR and DOCTOR as shown below. The relational table can be depicted as :

***PATIENT_DOCTOR** (P_NAME, DOB, D_NAME, DATE_TIME, DURATION)*

***DOCTOR** (D_NAME, PHONE)*

P__NAME	DOB	D__NAME	DATE__TIME	DURATION
MUKUL	10-02-1975	Dr Deepak	10-01-05 10:00	15 (minutes)
RUHIT	27-01-1971	Dr Kalyan	10-01-05 11:00	10
JAMES	30-03-1977	Dr Dhiraj	10-01-05 10:30	10
JAMES	30-03-1977	Dr Dhiraj	10-01-05 09:00	10
RUHIT	27-01-1971	Dr Kalyan	10-01-05 10:15	15
MUKUL	10-02-1975	Dr Dhiraj	10-01-05 10:50	20
RANA	02-11-1980	Dr Kalyan	10-01-05 11:10	20
MUKUL	10-02-1975	Dr Deepak	06-02-05 16:00	15

Table 7.4 Relation PATIENT_DOCTOR

D__NAME	PHONE
Dr Deepak	0381-223470
Dr Kalyan	0381-270572
Dr Dhiraj	0381-270875
Dr Dhiraj	0381-270875
Dr Kalyan	0381-223470
Dr Dhiraj	0381-270875
Dr Kalyan	0381-270572
Dr Deepak	0381-223470

Table 7.5 Relation DOCTOR

The functional dependency diagram of the above two relations are shown below :

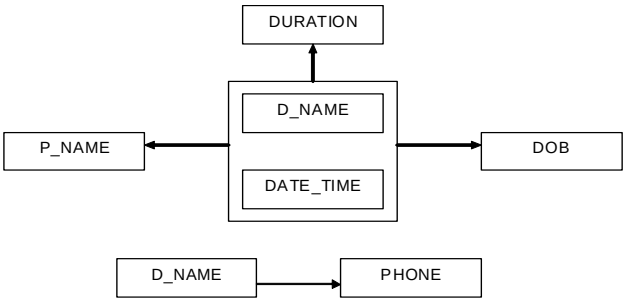

Fig. FFD of relation PATIENT_DOCTOR

4.11.3 Third Normal Form (3NF)

A relation or table is said to be in third normal form (3NF) if the relation is in 2NF and the non-prime attributes are -

a) mutually independent,

b) functionally dependent on the primary key.

It means that, no attributes of the relation should be transitively functionally dependent on the primary key. Thus, in 3NF, no non-prime attribue is functionally dependent on another non-prime attribute. This means that a relation in 3NF consists of the primary key and a set of independent nonprime attributes. 3NF is based on the problem of transitive dependency. The 3NF eliminates the problem of 2NF.

In our example, in the Fig 8.5, relation PATIENT_DOCTOR , there is no dependency between the attributes P_NAME and DURATION. Again, P_NAME and DOB are mutually dependent. So, the relation is not in 3NF.

To bring the relation into 3NF, it has to be decomposed and remove the attributes that are not directly dependent on the primary key. Now, using the transitive dependency, DOB can be linked to the primary key, through its dependency on the P_NAME. The functional dependency diagram is shown below. Now, the relations uses are -

***PATIENT_DOCTOR**(P_NAME,D_NAME, DATE_TIME,DURATION)*

***DOCTOR** (D_NAME, PHONE)*

***PATIENT** (P_NAME, DOB)*

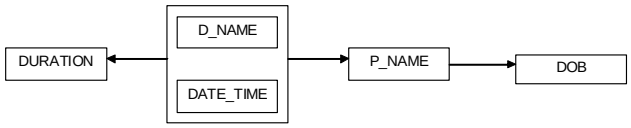

Fig. FDD for relation PATIENT_DOCTOR

Now, let us consider an another relation namely STUDENT.

***STUDENT** (Roll_No, Name, Dept, Year, Hostel_Name)*

Roll_No	Name	Dept	Year	Hostel_Name
1784	Ruhit	Physics	1	Ganga
1648	Keshab	Chemistry	1	Ganga
1768	Gopal	Maths	2	Kaveri
1848	Raja	Botany	2	Kaveri
1682	Mrinal	Geology	3	Krishna
1485	Sumit	Zology	4	Godavari

Table STUDENT relation

Here, the dependency Roll_No -> Hostal_Name is transitive through the following two dependencies :

Roll_No -> Year,

Year -> Hostal_Name

Thus, the STUDENT relation is not in 3NF. To bring the relation into 3NF we can decompose the relation into two relation STUD1 and STUD2, as shown below.

***STUD1** (Roll_No, Name, Dept)*

***STUD2** (Year, Hostel_Name)*

Roll_No	Name	Dept	Year
1784	Ruhit	Physics	1
1648	Keshab	Chemistry	1
1768	Gopal	Maths	2
1848	Raja	Botany	2
1682	Mrinal	Geology	3
1485	Sumit	Zology	4

Table Relation STUD1

Year	Hostel_Name
1	Ganga
1	Ganga
2	Kaveri
2	Kaveri
3	Krishna
4	Godavari

Table Relation STUD2

In the above examples, the conversion into 3NF is not hard, but when-ever a relation has more than one combination of attributes that may be considered as primary key then the conversion becomes proble-matic. Let us consider the following relation UTILIZE, shown below.

Project	Proj_Manager	Machine	Qty_Used
P1	Thomas	Excavator	5
P3	John	Shovel	2
P2	Abhishek	Drilling	5
P4	Avinash	Dumper	10
P3	John	Welding	3
P1	Thomas	Drilling	4

Table Relation UTILIZE

The relation stores the machines information used by both projects and project managers. Each project has one prooject manager and each project manager manages one project. Now it is obvious from the table that , we can consider any one of the combination of attributes as primary key, namely, {Project, Machine} or {Proj_Manager, Machine}. The FDD for relation UTILIZE is shown below.

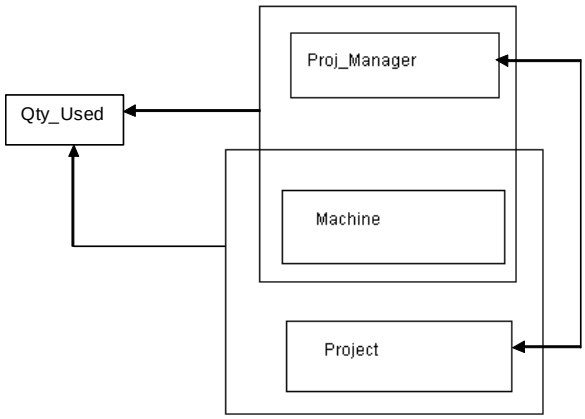

Fig. FDD of UTILIZE

In the relation, there is only one non-prime attribute called, QTY_Used, which is fully functionally dependent on each of the two relations. Thus, the relation UTILIZE is in 2NF. Moreover, there is only one non-prime attribute Qty_Used, there can be no dependencies between non-prime attributes. Thus the relation is also in 3NF.

Problems in 3NF :

If we consider the above relation i.e. UTILIZE, which is in 3NF, has the following undesirable properties :

a) the project manager of each project is stored more than once.

b) A project manager cannot be stored until the project has ordered some machines

c) A project cannot be entered unless that project's manager is known.

d) If a project's manager changes, some rows also must be changed.

4.12 SQL

The following are the important features of SQL:

i) *SQL can be used by a range of users, including those with little or no programming experience.*

ii) *It is non procedural language.*

iii) *It reduces the amount of time required for creating and maintaining systems.*

iv) *It is an English-like language.*

4.12.1 CLASSIFICATION OF SQL STATEMENTS

SQL statements are classified into three categories. They are–

Data Definition Language (DDL)

Data Manipulation Language (DML)

Data Control Language (DCL)

Data Definition Language

With the help of DDL statements, you can create and alter database objects like Tables, Rules etc like Create Table, Drop Table, and Alter Table. We will examine these in a next section of this unit.

Data Manipulation Language

DML statements enable you to change data and retrieve information from tables like Select, Insert, Update and Delete.

Data Control Language

DCL statements allow you to set, change permissions for security purposes like Grant, Deny and Revoke.

In order to apply SQL statements, you have to learn SQL syntax. For example, to create a database and a table, you have to apply the syntax for creating the same. For each and every purpose of database operations there are syntaxes.

4.12.2 SQL QUERY

Basically, all SQL code is written in the form of a query statement. All SQL queries perform some type of data operation such as selecting data, inserting/updating data, or creating data objects such as SQL databases and SQL tables. Each query statement begins with a clause. Structurally, each SQL query is similar.

SQL Expressions are the pieces of a SQL query that compare values against other values or perform arithmetic calculations. Expressions can be found inside of any SQL command usually in the form of a conditional statement. In the SQL world, conditional statements and expressions test or compare values against other values. Boolean expressions return rows (results) when a single value is matched with other values. SQL also offers several built-in functions to perform what is known as aggregate data calculations against a table or a specific table column. For Example following functions executed with query processing statements

AVG() — Returns the average value of a stated column.

COUNT(*) — Returns a count of the number of rows of table.

SUM() — Returns the sum of a given column

Syntax, by definition, means the study of linguistic rules and patterns. Every programming language, including SQL, must follow a unique set of guidelines termed syntax. Punctuation, spaces, mathematical operators, and special characters have special meaning when used inside of SQL commands and query statements.

By using the following rules, you can construct valid SQL statements that are both easy to read and edit.

- *SQL statements are not case sensitive.*

· *SQL statements can be on one or more lines.*

· *Keywords cannot be abbreviated or split across lines.*

· *Clauses are usually place on separate lines.*

· *Tabs and indents are used to enhance readability.*

· *Keywords (i.e. CREATE, ALTER …etc) are generally entered in the upper case.*

· *Only one statement can be current at any time within the buffer, and it can be run in continuous line by placing a semicolon (;) at the end of last clause.*

4.13 Suggested Readings

1. *"Database Systems:Concepts, Design and Applications", by S.K.Singth, Pearson Education*

2. *"Introduction to Database Management Systems", by Atul Kahate, Pearson Education*

3. *"Fundamentals of Database Systems", by Elmasri Navathe, Somayajulu & Gupta , Pearson Education Publication.*

4. *"An Introduction to Database Systems, by C.J. Date, Eighth Edition, Addison Wesley, 2003.*

4.14 Terminal Questions

- *What do you mean by DBMS?*
- *Write DBMS Database Structure Name.*
- *What is SQL?*
- *What is the different between Primary keys and foreign key?*

***