

LECTURE-6

Topics:

- Introduction to Operating system.
- Functions of an Operating system.
- Classification of Operating systems.

WHAT IS OPERATING SYSTEM?

- Operating system is a software ,which makes a computer to actually work
- A program that controls the execution of application programs
- An interface between applications and hardware
- The OS organizes and control the hardware
- Examples: Windows, Linux, Unix ,etc.

WHAT IS OPERATING SYSTEM?

Operating system – part of the computing system that manages all of the hardware and all of the software

Controls every file, device, section of main memory & every nanosecond of processing time

**Operating
system consists
of**

- Memory manager
- Processor manager
- Device manager
- File manager

WHAT OS DOES

An operating system performs basic tasks such as

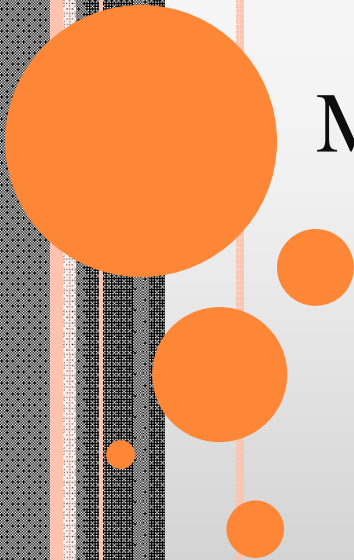
- ❑ Controlling and allocating memory
- ❑ Prioritizing system requests
- ❑ Controlling input and output devices
- ❑ Facilitating networking
- ❑ Managing file systems

CLASSIFICATION OF OS

Operating System can also be classified as,-

Single User Systems

Multi User Systems



SINGLE USER SYSTEMS:

- ❑ Provides a platform for only one user at a time.
- ❑ They are popularly associated with Desk Top operating system which run on standalone systems where no user accounts are required.
- ❑ Example: DOS

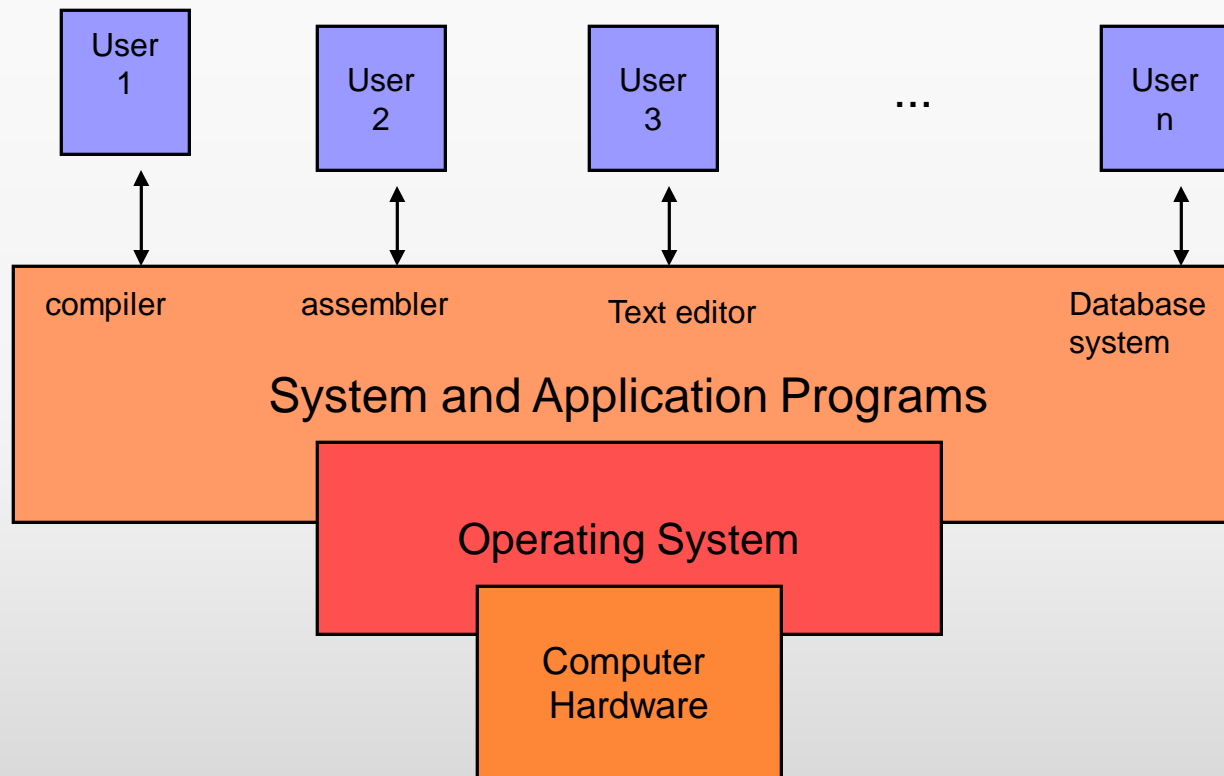


MULTI-USER SYSTEMS:

- ❑ Provides regulated access for a number of users by maintaining a database of known users.
- ❑ Refers to computer systems that support two or more simultaneous users.
- ❑ Another term for *multi-user* is *time sharing*.
- ❑ Ex: All mainframes and are multi-user systems.
- ❑ Example: Unix



STRUCTURE OF OS



STRUCTURE OF OS

The structure of OS consists of 4 layers:

1. hardware

Hardware consists of CPU, Main memory, I/O devices, etc .

2. Software(Operating System)

Software includes process management, memory management, I/O control, file management.

3. System programs

This layer consists of compilers, assemblers, linkers etc.

4. Application programs

This is dependent on user need .Ex. Railway reservation system, Bank database management etc.

OPERATING SYSTEM FUNCTION

The main function of operating systems are:

1. Program creation
2. Program execution
3. Input/output operation
4. Error detection
5. Resource allocation
6. Accounting
7. Protection

EVOLUTION OF OS

Operating system timeline

First generation: 1945 – 1955

Vacuum tubes
Plug boards

Second generation: 1955 – 1965

Transistors
Batch systems

Third generation: 1965 – 1980

Integrated circuits
Multiprogramming

Fourth generation: 1980 – present

Large scale integration
Personal computers

Next generation: ???

Systems connected by high-speed networks?
Wide area resource management?

FIRST GENERATION: DIRECT INPUT

❑ **Run one job at a time**

- Enter it into the computer (might require rewiring!)
- Run it
- Record the results

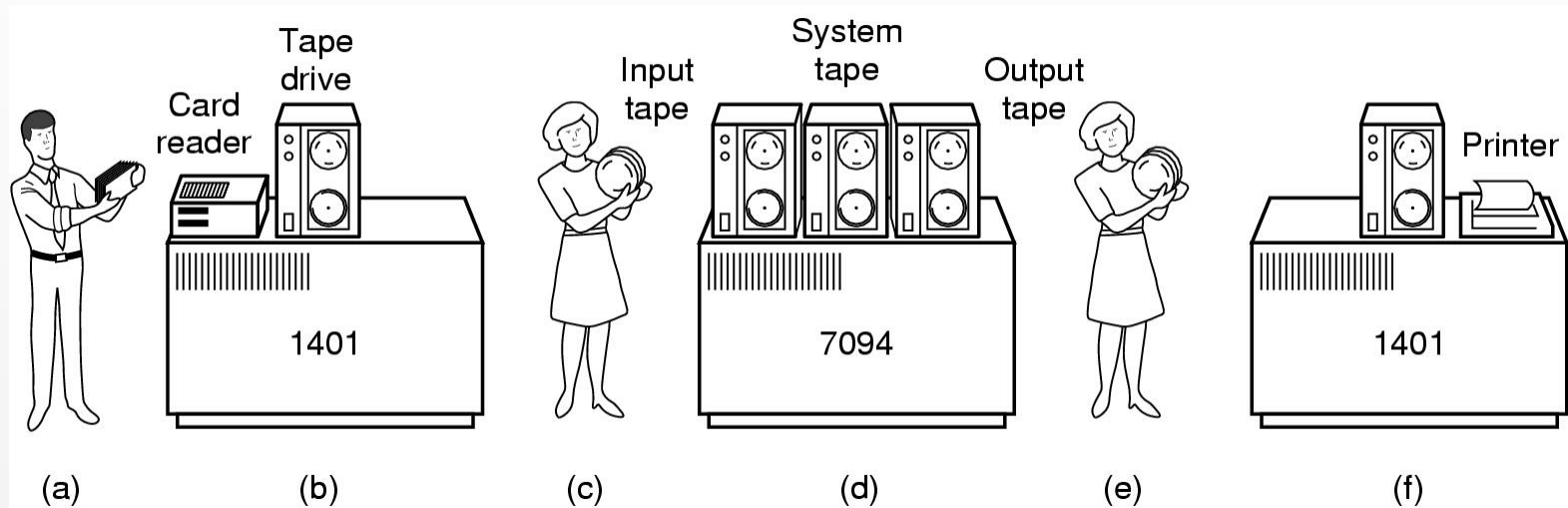
❑ **Problem: lots of wasted computer time!**

- Computer was idle during first and last steps
- Computers were *very* expensive!

❑ **Goal: make better use of an expensive commodity: computer time**



SECOND GENERATION: BATCH SYSTEMS



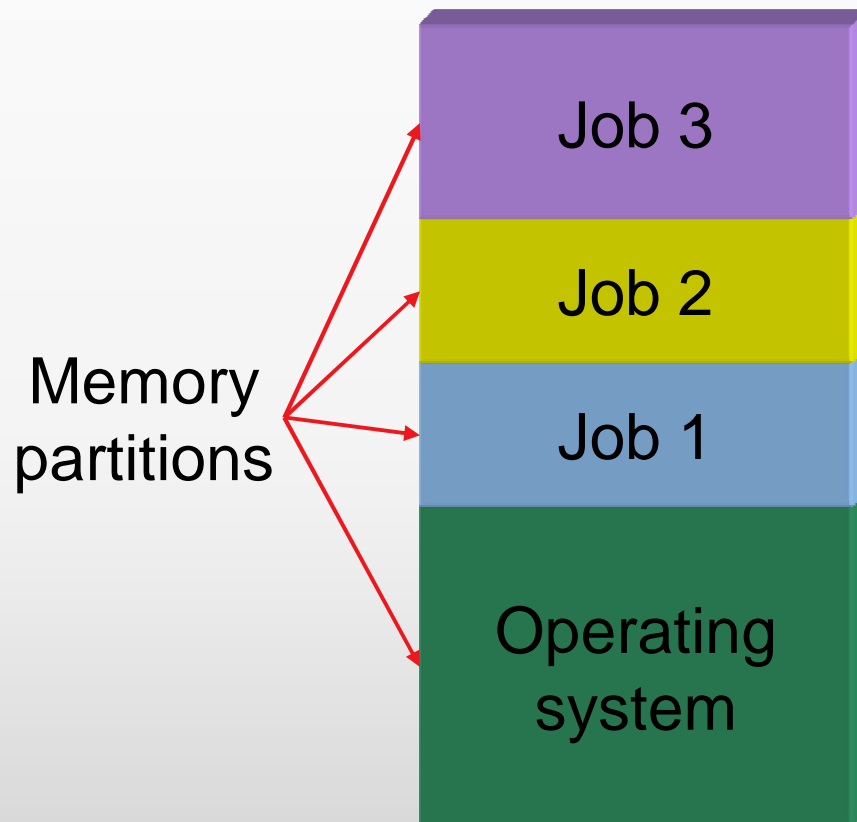
- ❑ Bring cards to 1401
- ❑ Read cards onto input tape
- ❑ Put input tape on 7094
- ❑ Perform the computation, writing results to output tape
- ❑ Put output tape on 1401, which prints output

SPOOLING

- ❑ Original batch systems used tape drives
- ❑ Later batch systems used disks for buffering
 - Operator read cards onto disk attached to the computer
 - Computer read jobs from disk
 - Computer wrote job results to disk
 - Operator directed that job results be printed from disk
- ❑ Disks enabled simultaneous peripheral operation on-line (spooling)
 - Computer overlapped I/O of one job with execution of another
 - Better utilization of the expensive CPU
 - Still only one job active at any given time



THIRD GENERATION: MULTIPROGRAMMING



- ❑ Multiple jobs in memory
 - Protected from one another
- ❑ Operating system protected from each job as well
- ❑ Resources (time, hardware) split between jobs
- ❑ Still not interactive
 - User submits job
 - Computer runs it
 - User gets results minutes (hours, days) later

TIMESHARING

- ❑ Multiprogramming allowed several jobs to be active at one time
 - Initially used for batch systems
 - Cheaper hardware terminals -> interactive use
- ❑ Computer use got much cheaper and easier
 - No more “priesthood”
 - Quick turnaround meant quick fixes for problems



BRIEF HISTORY OF OPERATING SYSTEMS DEVELOPMENT

Second Generation

Job scheduling, JCL,
faster I/O, spooling,
batch, files

Recent Developments

Distributed computing,
personal computers,
high-speed
communication, multi-
media

First Generation

Vacuum tube, single
user, early operating
systems

Third Generation

Shared processing,
multiprogramming, virtual
memory, DBMS



OPERATING SYSTEM CLASSIFICATION

Operating Systems can be classified as:

- *GUI*: Graphical User Interface operating systems are operating systems that have the capability of using a mouse and are graphical
- *Multi user*: allows multiple users to utilize the computer and run programs at the same time
- *Multi processing*: allows multiple processors to be utilized
- *Multi tasking*: allows multiple software processes to be run at the same time
- *Multi threading*: allows different parts of a software program to run concurrently



LECTURE-7

Topics:

- Introduction to DOS.
- Introduction to UNIX/LINUX OS.
- Introduction to Windows.



BASIC INTRODUCTION TO DOS OPERATING SYSTEM

DISK OPERATING SYSTEM (DOS)

- ❑ In the 1980s or early 1990s, the operating system that shipped with most PCs was a version of the *Disk Operating System (DOS)* created by Microsoft: *MS-DOS*.
- ❑ MS-DOS is a disk operating system for IBM PC-compatible computers.
- ❑ In its day, it was easily the most popular operating system in the world.



MS-DOS, PC-DOS OR DOS

- ❑ **Developed to run single-user, stand-alone desktop computers.**
 - Exemplifies early Operating Systems because it manages jobs sequentially from single user.
- ❑ **Advantages:**
 - Simple operation & straight-forward user commands.
- ❑ **Disadvantages:**
 - Lack of flexibility & limited ability to meet needs of programmers & experienced users.
 - Written for a single family of microprocessors (Intel family of chips: 8086, 8088, 80186, and 80286).

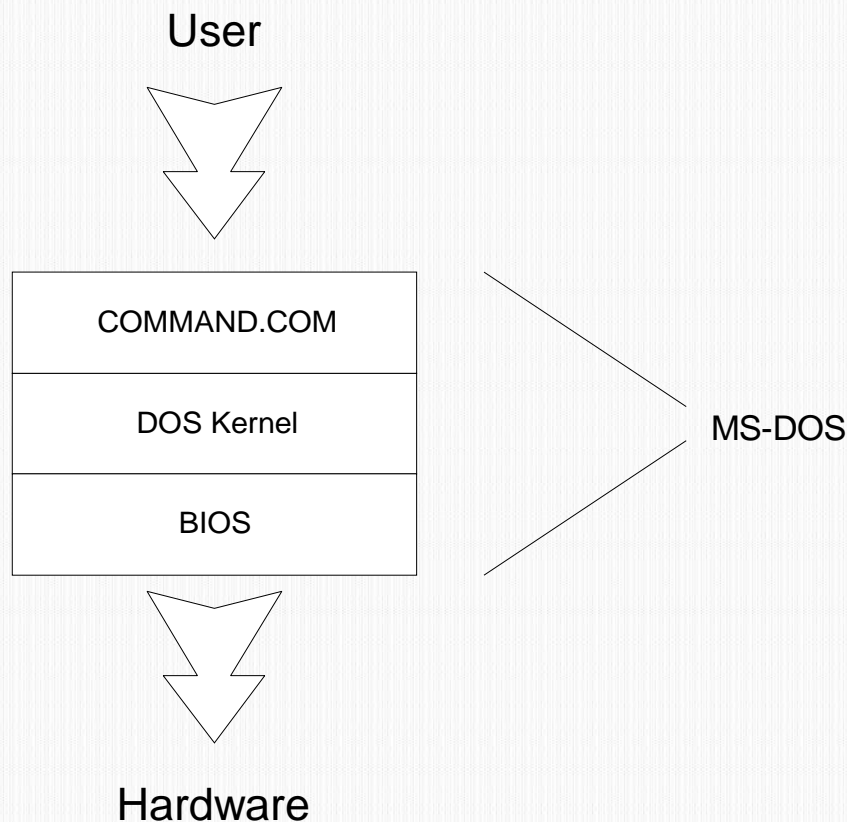


THE MAIN FUNCTIONS OF DOS

- ❑ DOS translate the command issued of the user in the format that is understood by the computer to execute it, also error message in the format for the user to understand.
- ❑ Manage disk file.
- ❑ Manage memory efficiently.
- ❑ Allocate system resources according to the requirement.
- ❑ DOS provides features essential to control hardware devices such as Keyboard, Screen, Disk Devices, Printers, Modems and programs.



THE STRUCTURE OF DOS



- ❑ Accommodate single novice user in single-process environment.
- ❑ Standard I/O support includes keyboard, monitor, printer, & secondary storage unit.
- ❑ User commands are based on English words/phrases indicative of action to be performed.
- ❑ Commands are interpreted by command processor.
- ❑ Layering approach “protects” user from hardware.

FIRST 2 DOS LAYERS : BIOS & KERNEL

1. **BIOS (Basic Input/Output System)** -- interfaces directly with various I/O devices.
 - Device drivers (control flow of data to/from each device).
 - Receives status each I/O operation & passes it on to the processor about success/failure of operation.
2. **DOS kernel** -- routines needed to interface with the disk drives.
 - Read into memory at initialization time from MSDOS.SYS file on boot disk.
 - Accessed by application programs.
 - Provides a collection of hardware-independent services, such as memory management, & file & record management (system functions).



THIRD DOS LAYER : COMMAND PROCESSOR

Command processor (shell) -- sends prompts to user, accepts commands, executes commands, & issues appropriate responses.

- Resides in a file called COMMAND.COM, which consists of 2 parts stored in 2 different sections of main memory.
- Not interpretive.
- MS-DOS Version 4.0 -- menu-driven DOS shell.
- OS/2 -- designed to replace MS-DOS.



USER INTERFACE

- ❑ Command-driven operating system.
- ❑ When user presses Enter key, shell (COMMAND.COM) interprets command & calls on next lower level routine to satisfy request.
- ❑ User commands include some or all of these elements in this order:
command source-file destination-file switches



Command	Stands for	Action to be performed
DIR	Directory	List what's in this directory.
CD or CHDIR	Change Directory	Change working directory.
COPY	Copy	Copy a file. Append one to another.
DEL or ERASE	Delete	Delete the following file or files.
RENAME	Rename	Rename a file.
TYPE	Type	Display text file on screen.
PRINT	Print	Print one or more files on printer.
DATE	Date	Display and/or change system date.
TIME	Time	Display and/or change system time.
MD or MKDIR	Make Directory	Create a new directory or subdirectory.
FIND	Find	Find a string. Search files for a string.
FORMAT	Format Disk	Logically prepare disk for file storage.
CHKDSK	Check Disk	Check disk for disk/file/directory status.
PROMPT	System Prompt	Change system prompt symbol.
DEFRAG	Defragment Disk	Compact fragmented files.
(filename)		Run, execute, file.



DOS COMMANDS

- ❑ **Internal Commands**

- ❑ **External Commands**



INTERNAL DOS COMMANDS

- ❑ Internal Commands are located in memory and do not require and further disk access when they are used.
- ❑ COPY, DEL, DIR, VER, TYPE, PRINT

EXTERNAL DOS COMMANDS

- ❑ they reside on the DOS disk and require a disk access to be used
- ❑ CHKDSK, FORMAT, DISKCOPY ,MD,CD,RD



FORMAT

- ❑ NEW DISKS MAY BE FORMATTED ALREADY OR NOT
- ❑ USING THE FORMAT COMMAND INITIALIZES; BY WRITING CODE ON IT TO ASSIGN TRACKS AND SECTORS WHERE YOUR DATA WILL BE STORED.
- ❑ FORMAT ERASES AND CHECKS OLD DISKS FOR ERRORS
- ❑ FORMAT A:/S/Q/U

DIRECTORY

- ❑ THE DIR COMMAND IS AN INTERNAL DOS COMMAND USED TO LIST CONTENT
- ❑ DIR
- ❑ DIR/W
- ❑ DIR/P



COPY COMMAND

- ❑ THE COPY COMMAND MAKES A SECOND COPY OF A FILE ONTO ANOTHER DESTINATION
- ❑ IF YOU WANT TO COPY SOMETHING FROM ONE DIRECTORY TO A DISK YOU MUST BE IN THAT DIRECTORY
- ❑ COPY DOES NOT ERASE THE ORIGINAL

DEL OR ERASE

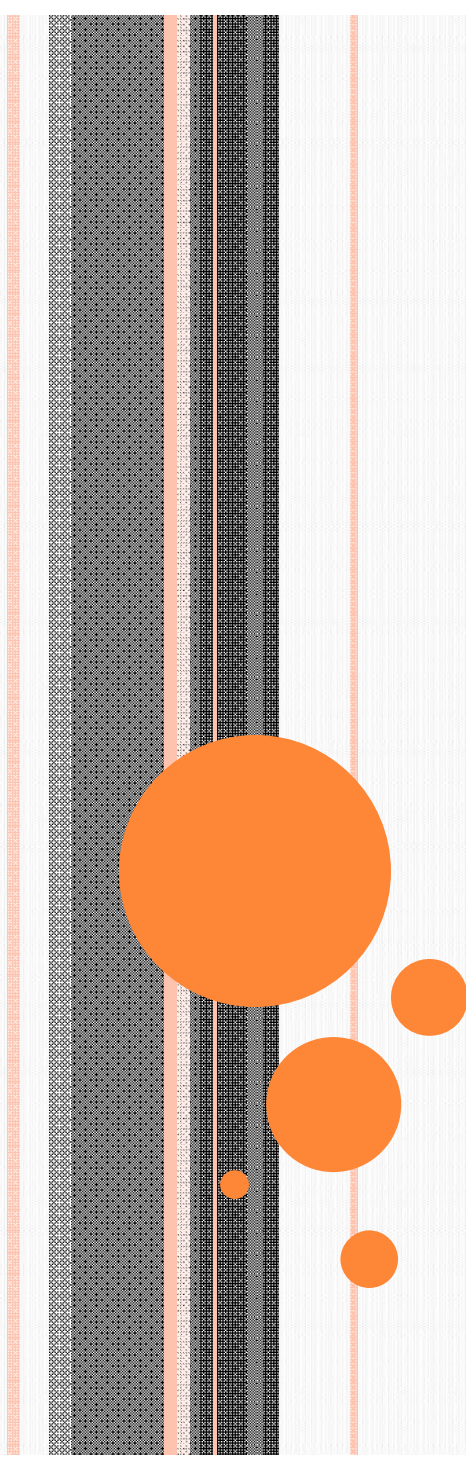
- ❑ REMOVES DATA
- ❑ WINDOWS 95 IS HELPFUL FOR THIS
- ❑ ONCE ERASED , A FILE CANNOT BE RECOVERD UNLESS USE A SPECIAL UTILITY LIKE NORTON UTILITIES



MKDIR(MD), CHDIR(CD) , RMDIR(RD)

- ❑ MD MAKES A DIRECTORY
- ❑ CD CHANGES DIRECTORY
- ❑ RD REMOVES A DIRECTORY





BASIC INTRODUCTION TO UNIX/LINUX OPERATING SYSTEM

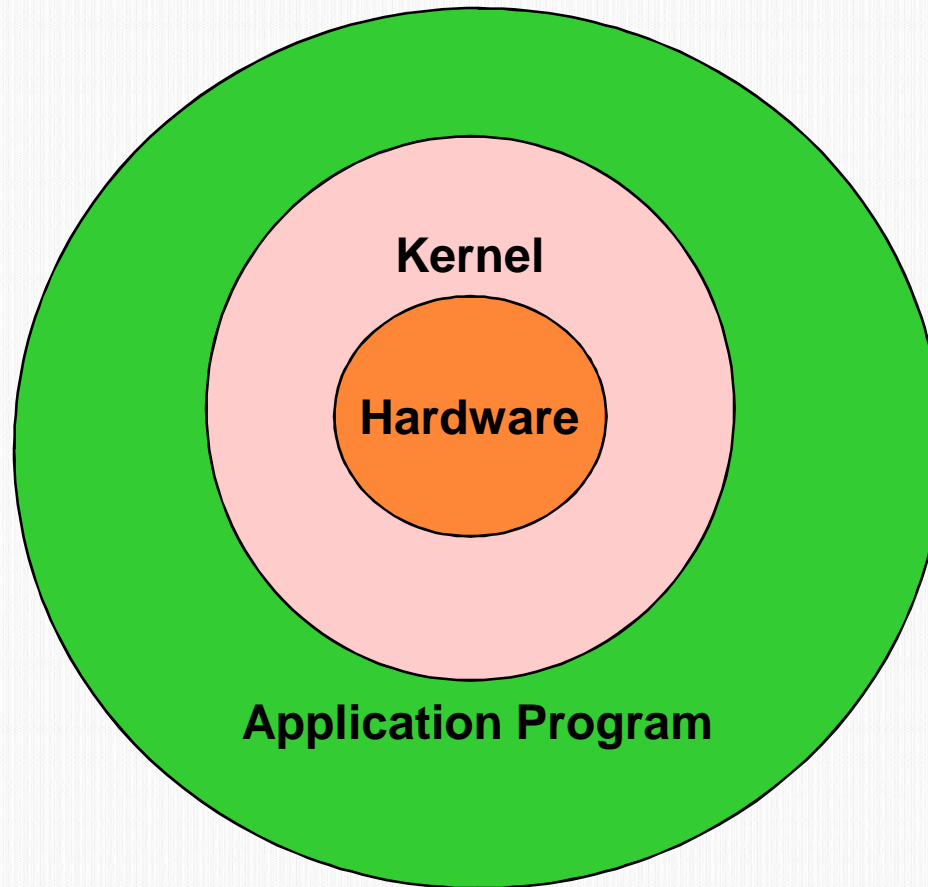
AN OVERVIEW OF THE UNIX OS

At first, unix was designed to let a number of programmer access the same time and share its resource. The important features that make unix favorite are:

- ❑ Multitasking
- ❑ Multiuser
- ❑ Portability
- ❑ Unix Tools
- ❑ Unix Communication
- ❑ Library of application software



UNIX STRUCTURE



START UP

- ❑ Before the UNIX operating system can perform correctly, there are a number of steps that must be followed
- ❑ The failure of any one of these can mean that the system will not start.
- ❑ It is important for the Systems Administrator to be aware of what happens during system startup so that any problems that occur can be remedied.



BOOTING - CONSISTS OF THE FOLLOWING STEPS

- ❑ ROM Phase
- ❑ Boot Program Phase
- ❑ Kernel initialization Phase
- ❑ Init Phase



UNIX COMMAND

- ❑ A command is a program that tells the Unix system to do something.
- ❑ Usually has the form: **command** *[options] [arguments]*
- ❑ Commands are case sensitive. *command* and *Command* are not the same.
- ❑ Options and syntax for a command are listed in the *man page*



Navigation and Directory Control Commands

Command/Syntax	What it will do
ls [options] [directory or file]	list directory contents or file permissions
cd [directory]	change directory
mkdir [options] directory	make a directory
pwd	print working (current) directory
rmdir [options] directory	remove a directory



FILE MAINTENANCE COMMANDS

Command/Syntax	What it will do
<code>chgrp[options] groupfile</code>	change the group of the file
<code>chmod[options] file</code>	change file or directory access permissions
<code>cp[options] file1 file2</code>	copy file1 into file2, file2 shouldn't already exist.
	This command creates or overwrites file2
<code>mv[options] file1 file2</code>	move file1 into file2
<code>rm[options] file</code>	remove (delete) a file or directory
	(-r recursively deletes the directory and its contents)



MISCELLANEOUS COMMANDS

Command/Syntax	What it will do
cat [options] file	concatenate(list) a file
more(or less or pg) [options] file	page through a text file
tail [options] file	display the last few lines (or parts) of a file
head [-number] file	display the first 10 (or number of) lines of a file
which command	reports the path to the command or the shell alias in use
who or w	report who is logged in and what processes are running
date [options]	report the current date and time
grep [options] file	search specify pattern in the file
tar [options] file or directory	pack and zip file



UNIX FILE TYPE

- **Regular file.** A common file which contains data in text or binary.
- **Directory file.** A file that contains name of other files and point to information on these files.
- **Symbolic link.** A type of file that point to another file.



UNIX FILE TYPE

- **Character devices file.** A type of file use for certain type of devices(read-write data in one character).
- **Block devices file.** A type of file use for certain type of devices(read-write data in block).
- **FIFO.** A type of file that use for interprocess communication,some time call named pipe.
- **Socket.** A type of file that use for network communication.



File Access Permission

Symbol	Permission	Means
r	Read	Can open and read content of file
w	Write	Can write to file(Modify content) or delete it
x	Execute	Can execute the file(if it program or shell script)
-	Denied	Cannot read, write or execute file



LECTURE-8

Topics:

- Introduction to computer languages.
 - Low level languages
 - High level languages

Types of high and low level languages

PROGRAMMING LANGUAGES

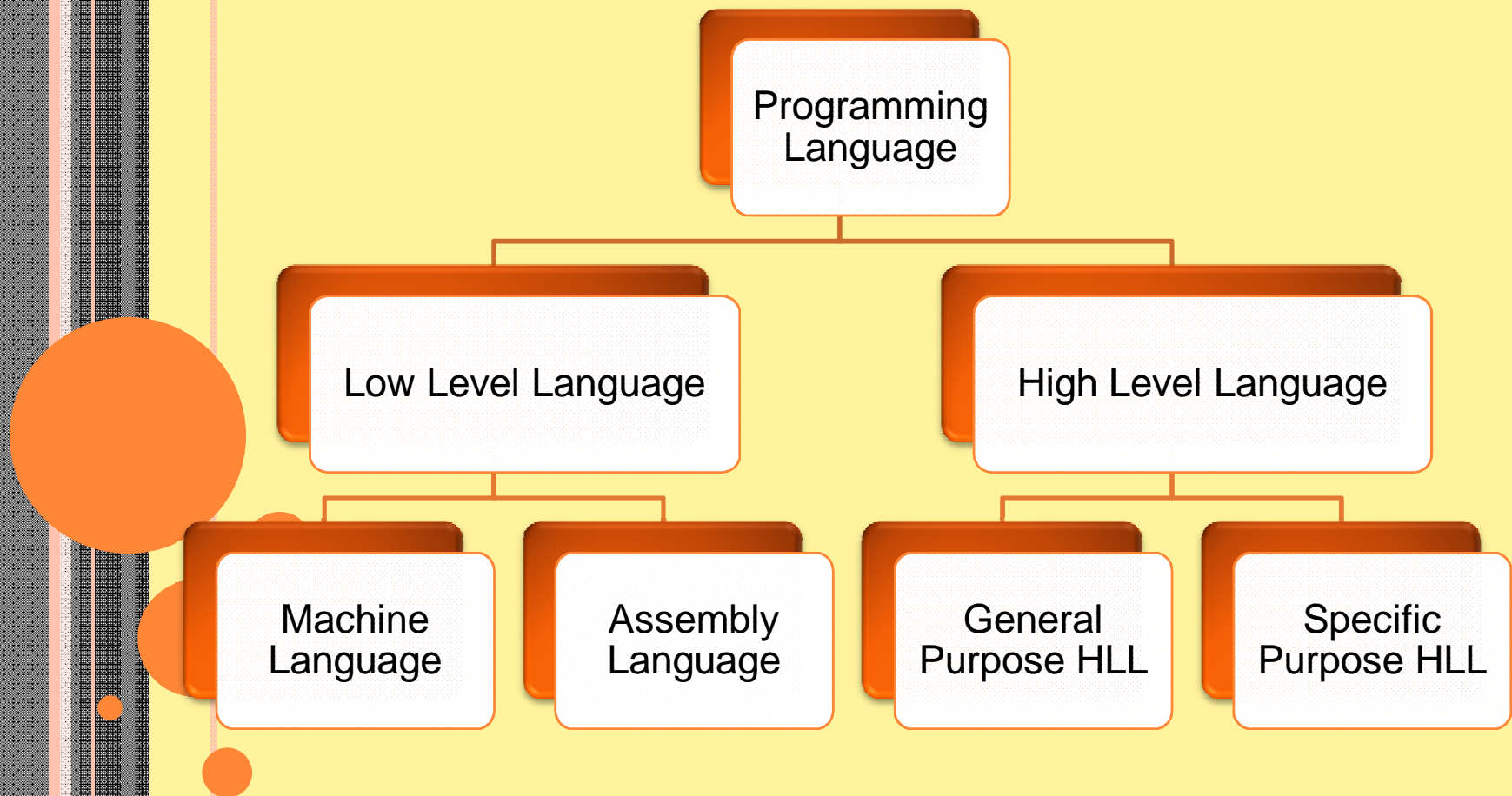
The art of writing instructions for a computer to solve the specific task is known as programming.

or

A vocabulary and set of grammatical rules for instructing a computer to perform specific tasks

- ✓ The output of programming is a well defined set of instructions.
- ✓ This is called a **program**
- ✓ A programming language is the medium of communication between the man and the machine.
- ❑ Computer languages are classified into two levels:
 - ❑ **Low level languages**
 - ❑ **High level languages**

PROGRAMMING LANGUAGES



LOW LEVEL LANGUAGES

- These are easily understood by computers.
- They are machine dependent languages.
- Programs written in these languages are not transferable from one computer to another.
- The machine language and assembly language are examples of low level languages.

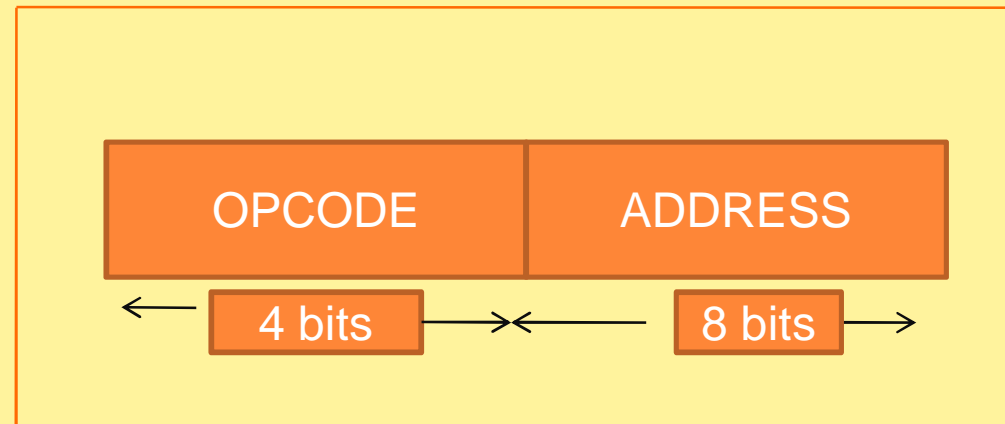
MACHINE LANGUAGES

- ❑ As the name itself implies, programming is done at machine level.
- ❑ The vocabulary of computer consists of only two words ,zero and one.
- ❑ The digit of a binary number system.
- ❑ A 0 indicates low voltage level and 1 indicates high voltage level
- ❑ The machine level language is considered as a first generation language
- ❑ Any sequence of 0's and 1's forms an instruction in this language.
For example: 1010101011
- ❑ Each instruction has a specific format, consisting of two fields.
- ❑ First field is opcode and
- ❑ Second field is address



MACHINE LANGUAGES

Instruction Format



Opcode- (it stands for operation code). It indicates what operation is to be done .such as addition, subtraction, multiplication etc.

Address –(memory location where in the data is referenced)
The length of each instruction is 12 bits;4bit for opcode and 8bits for address.

MACHINE LANGUAGES

Advantages

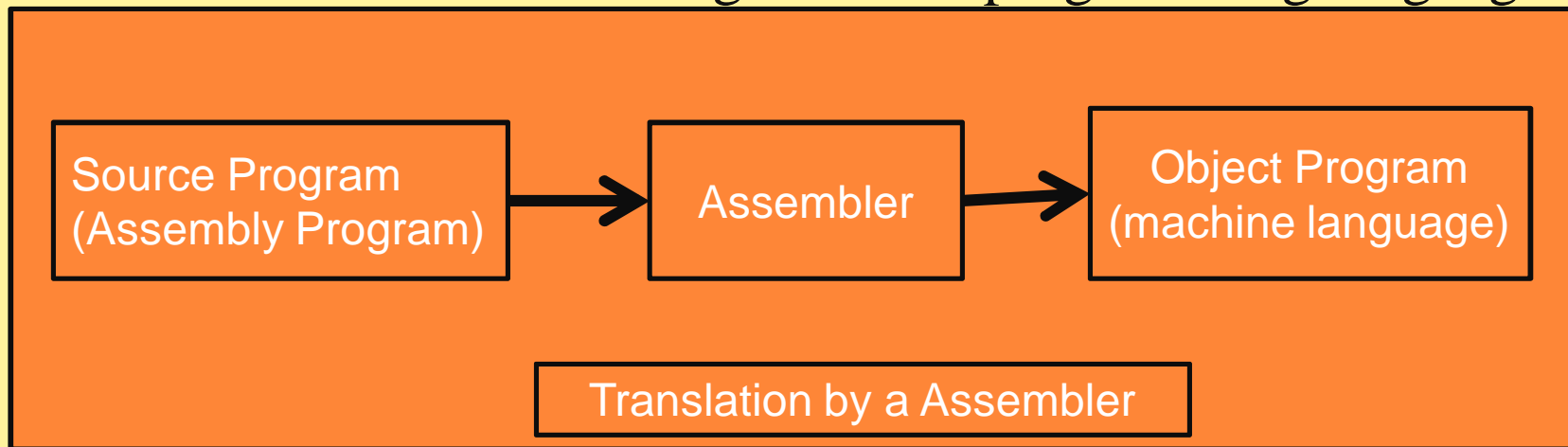
- can be directly typed and executed
- No translator program is required.

Disadvantages

- Difficult to remember machine instructions
- Difficult to understand , modify, and debug errors
- Difficult to remember address of data and instructions
- Each and every instruction is numerical

ASSEMBLY LANGUAGE

- ❑ To overcome the drawbacks of the machine language ,computer engineers developed a new programming language which uses symbolic instructions.
- ❑ This symbolic instruction oriented programming language is known as assembly language.
- ❑ This is called as the second generation programming language.



ASSEMBLY LANGUAGE

Symbolic words used in this language are referred to as mnemonics. The meaning of mnemonic is memory sake or to remember.

For example,

ADD	For Addition
SUB	For Subtraction
MUL	For Multiplication
STA	For store at Accumulator
HALT	For Halt
JMP	For jump
INR	For increment and so on

ASSEMBLY LANGUAGES

Advantages

- Easy to remember operation codes
- Easy to write the programs, modify and debug as compared to machine language.
- Need not remember addresses of operands and instruction locations

Disadvantages

- The mnemonics are machine dependent.
- Not standardized.
- Less efficient than machine language.

HIGH LEVEL LANGUAGES

- ❑ Higher-level languages are more powerful than assembly language and allow the programmer to work in a more English-like environment.
- ❑ Higher-level programming languages are divided into three "generations," each more powerful than the last:

- **Third-generation languages**
- **Fourth-generation languages**
- **Fifth-generation languages**

Source Program
(HLL Language)



Translator
(Interpreter)/(Compiler)



Object Program
(Machine Language)

Translation by Compiler to Interpreter

Higher-Level Languages - Third-Generation Languages

- ❑ Third-generation languages (3GLs) are the first to use true English-like phrasing, making them easier to use than previous languages.
- ❑ 3GLs are portable, meaning the object code created for one type of system can be translated for use on a different type of system.
- ❑ The following languages are 3GLs:

FORTAN

COBOL

BASIC

C

C++

Java



Higher-Level Languages - Fourth-Generation Languages

- ❑ Fourth-generation languages (4GLs) are even easier to use than 3GLs.
- ❑ 4GLs may use a text-based environment (like a 3GL) or may allow the programmer to work in a visual environment, using graphical tools.
- ❑ The following languages are 4GLs:

Visual Basic (VB)

VisualAge

Authoring environments



Higher-Level Languages - Fifth-Generation Languages

- ❑ Fifth-generation languages (5GLs) are an issue of debate in the programming community – some programmers cannot agree that they even exist.
- ❑ These high-level languages would use artificial intelligence to create software, making 5GLs extremely difficult to develop.
- ❑ Solve problems using constraints rather than algorithms, used in Artificial Intelligence

Prolog



LECTURE-9

Topics:

- Compiler
- Interpreter
- Loader
- Linker

.Types of Software.

COMPILER/INTERPRETER

Regardless of what language you use, you eventually need to convert your program into a language that the computer can understand

Two ways for doing that:

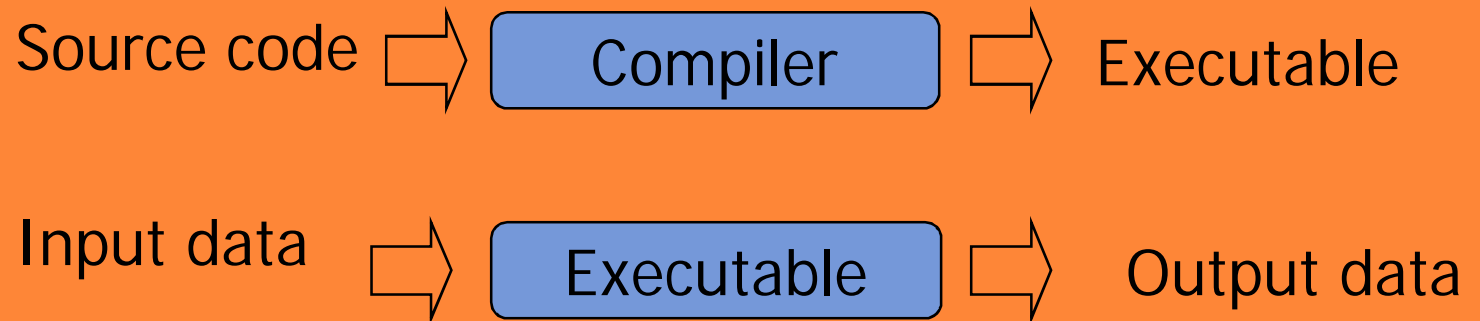
**compile the program or
interpret the program**

COMPILER

A compiler is a computer program that translates a program in a *source language* into an equivalent program in a *target language*.

or

Compilers: Translate a source (human-writable) program to an executable (machine-readable) program



COMPILER

- Translate the entire program.
- Convert the entire program to machine code, when the syntax errors are removed then converted into the object code
- Requires more main memory
- Neither source nor the compiler are required for execution.
- Slow for debugging and testing.
- Execution time is less.
- Security of source code .

INTERPRETER

Interpreter is a program that executes instructions written in a high-level language

or

Interpreters: Convert a source program and execute it at the same time.

Source code

Input data



Interpreter



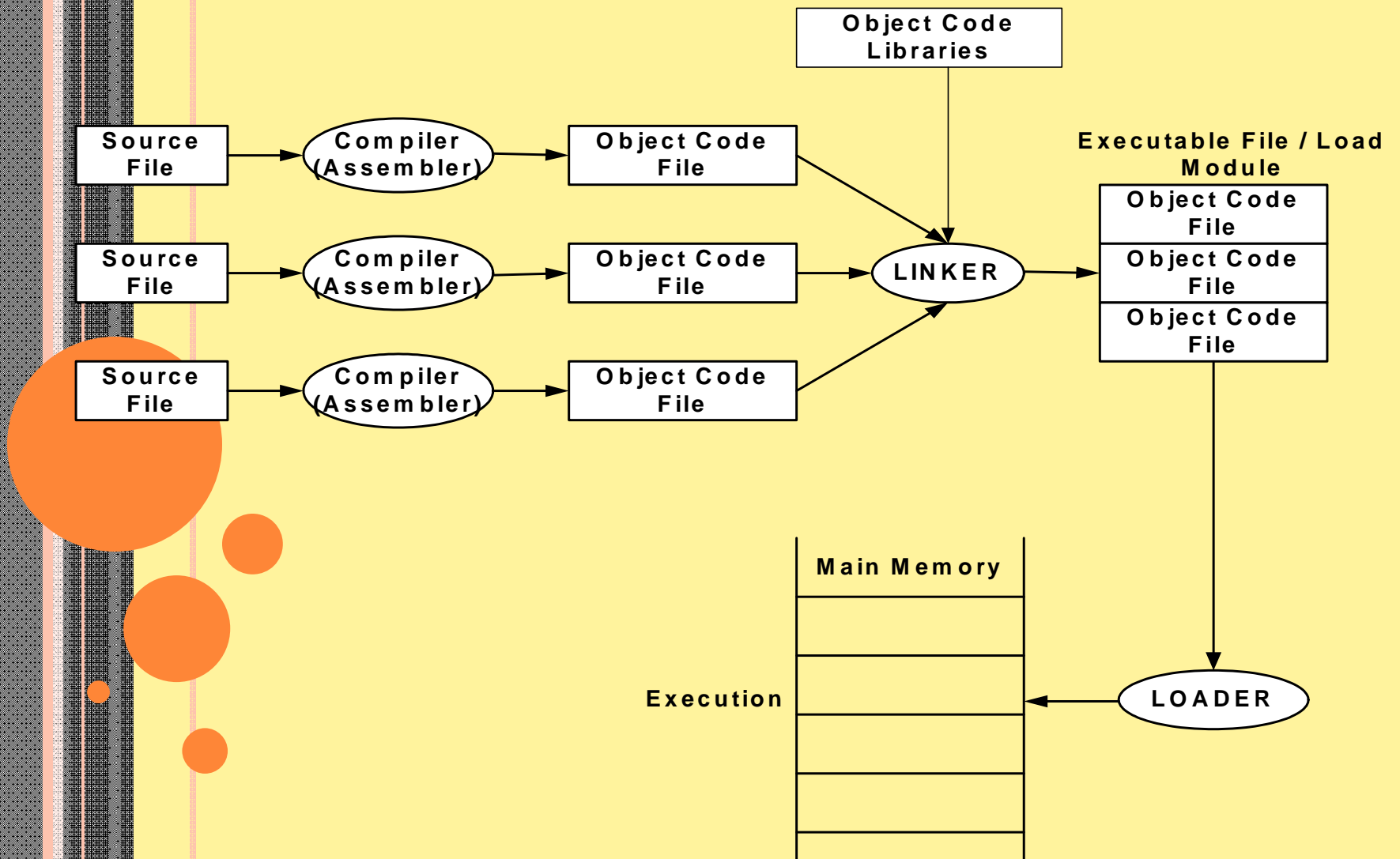
Output data

INTERPRETER

- Translate the program line by line.
- each time the program is executed ,every line is checked for syntax error & then converted to equivalent machine code directly.
- Requires less main memory
- Source program and the interpreter are required for execution.
- Good for fast debugging and testing.
- Execution time is more.
- No Security of source code .

LINKER/LOADER

INTODUCATION



DEFINITION OF LINKER

A program that takes as input the object files of one or more separately compiled program modules, and links them together into a complete executable program, resolving reference from one module to another.

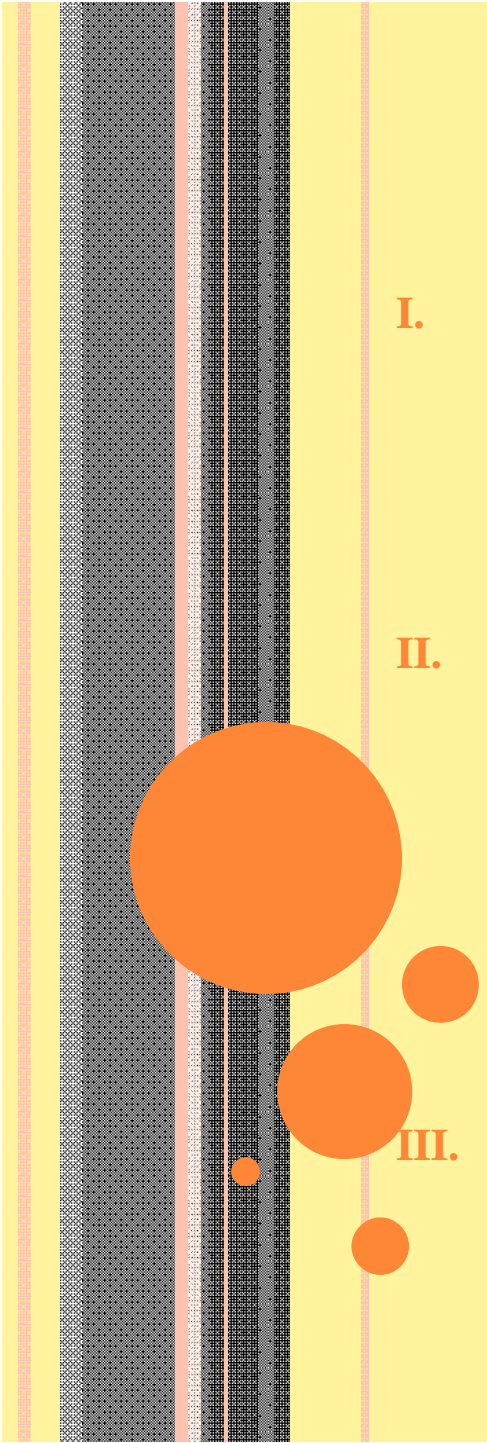
DEFINITION OF LOADER

- ❑ A program that takes an input an executable program, loads it into main memory, and causes execution to begin by loading the correct starting address into the computer register.
- ❑ When the program finished, control must somehow be returned to the operating system.

OBJECT FILE & EXECUTABLE FILE

- ❑ A single object file might contain machine code for only one procedure or a set of procedures.
- ❑ An executable file must contain all the machine code needed for a particular program; it must contain the address of the first instruction to be executed.

TYPES OF LOADER

- 
- I. **Absolute loader:** absolute loader is a primitive type of loader which does only the loading function. It does not perform linking and program relocation.
 - II. **Bootstrap loader:** when the computer is turned on absolute loader is executed. this loader is responsible for loading the operating system and transferring control to it. This loader is present in the ROM area of main memory.
 - III. **Relocateabel loader:** this loader is responsible for relocation and loading.

TYPES OF LINKER

- I. **Linking loader:** this linker performs all the linking and relocation operations and load the linked program directly into the main memory.
- II. **Linkage editor:** this linker produces a linked version of the program called as a load module or an executable image.
- III. **Dynamic linker:** this. Scheme postpone the linking function until execution time. Any subroutine is loaded and linked to the rest of program when it is first called.

SOFTWARE

What is software?

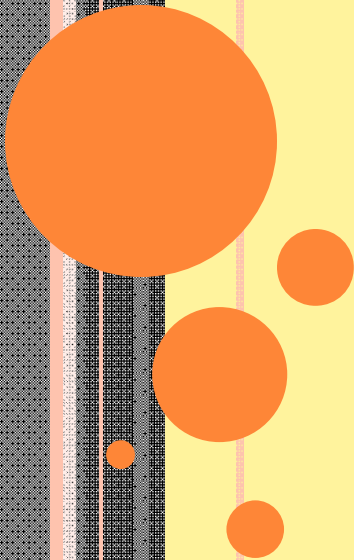
Computer Instructions or data, anything that can be stored electronically is Software.

Types of Software

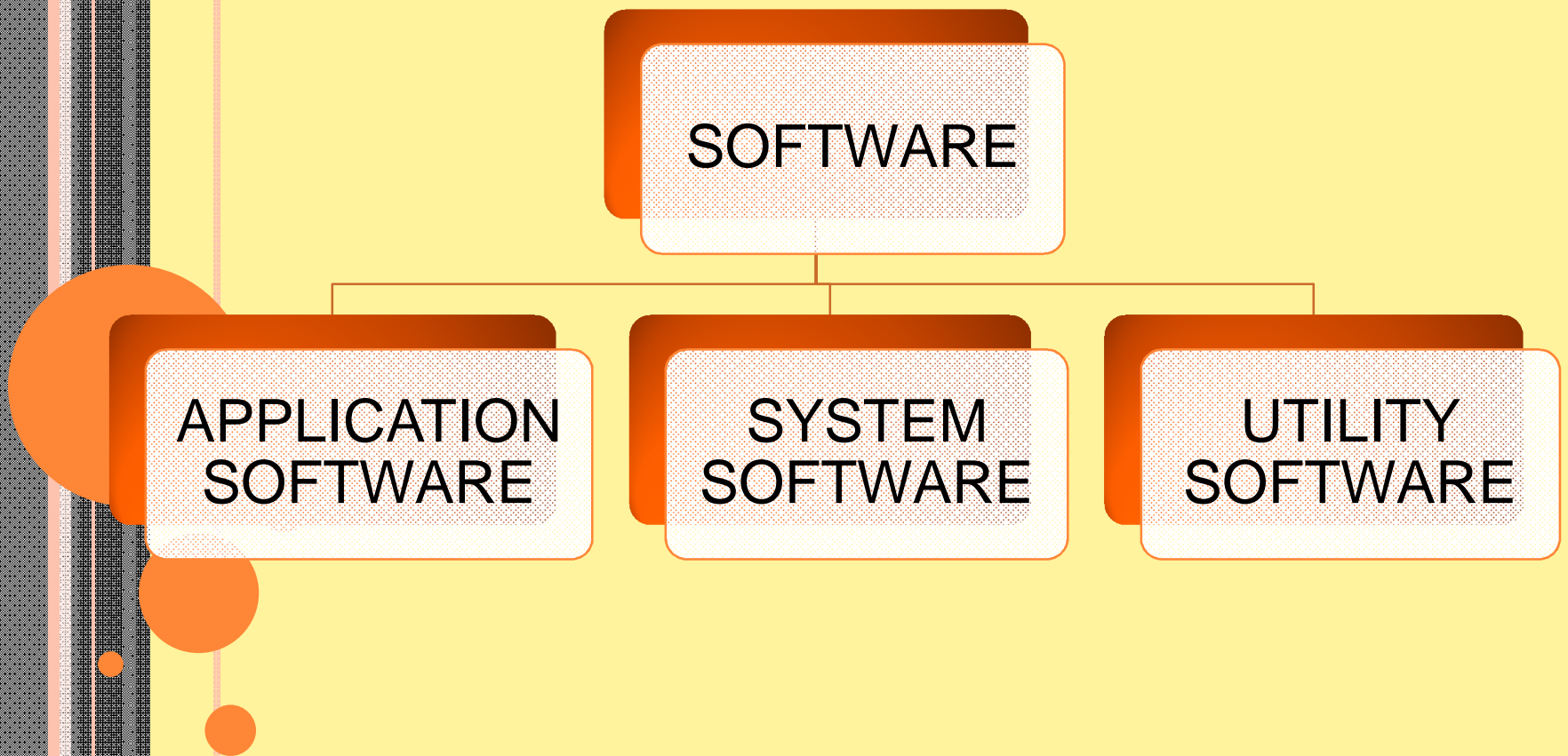
System Software

Application Software

Utility Software



CLASSIFICATION OF SOFTWARE



SYSTEM SOFTWARE

System Software includes the Operating System and all the utilities that enable the computer to function.

System software is a term referring to any computer software which manages and controls the hardware so that application software can perform a task.

Example:

Operating Systems, Compiler, Loader, Linker, Interpreter.



APPLICATION SOFTWARE:

This is a general purpose program or a collection of programs written by the user to solve a particular problem.

Application Software includes programs that do real work for user.

Example:

Payroll systems, Inventory Control, Manage student database, Word Processor, Spreadsheet and Database Management System etc.,



UTILITY SOFTWARE:

these are readymade routines that are incorporated into the computer utility software are generally called as application oriented readymade system programs

Example:

Soundblaster , sidekick etc.





Lecture-10

Introduction

- **Data Communication**
 - **Networks**
 - **Protocols**
-

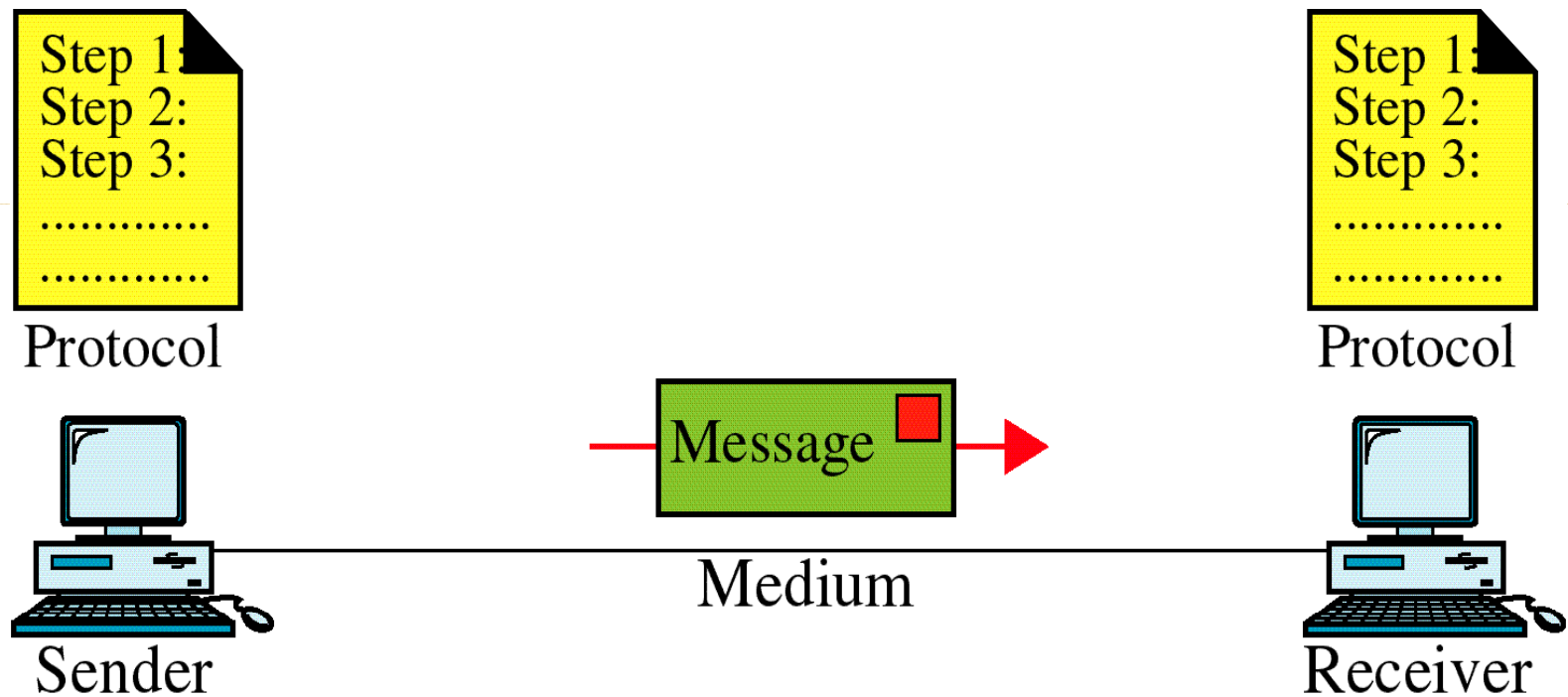
Data Communication System Components

Data communication is the exchange of data between two devices via some form of transmission medium such as a wire cable

Data communication system has five component

1. Message
2. Sender
3. Receiver
4. Transmission Medium
5. Protocol

Data Communication System Components





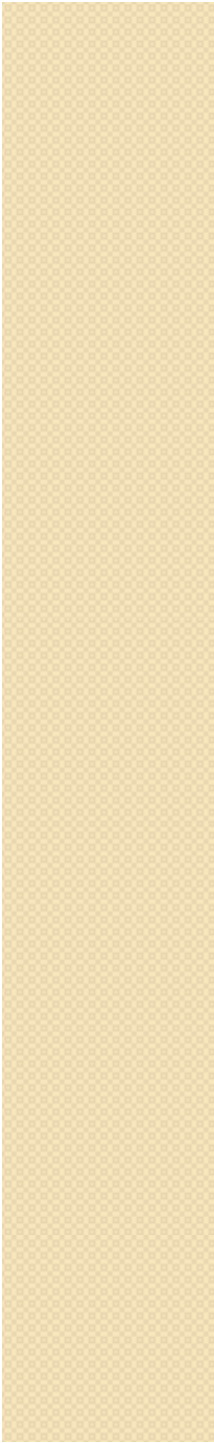
Type of connection

A network is two or more devices connected through links.

A link is a communications pathway that transfer data from one device to another.

For communication to occur, two devices must be connected in some way to the same time. There are two possible types of connections:

1. point-to-point
2. multipoint



Type of Connection

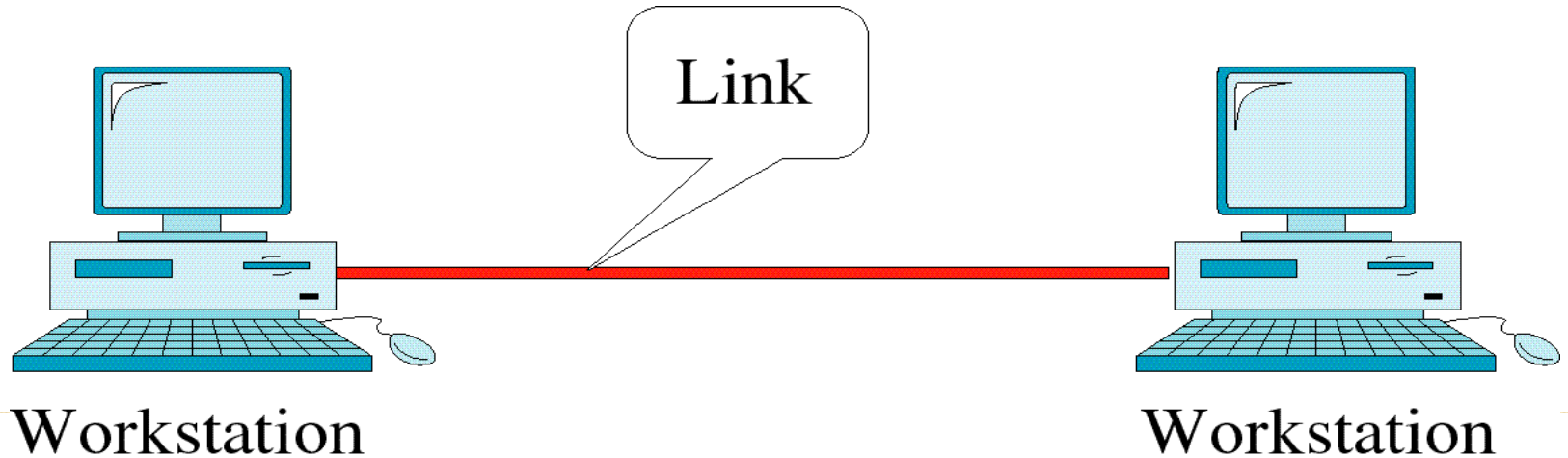


```
graph TD; A[Type of Connection] --> B[Point-to-point]; A --> C[Multipoint]
```

Point-to-point

Multipoint

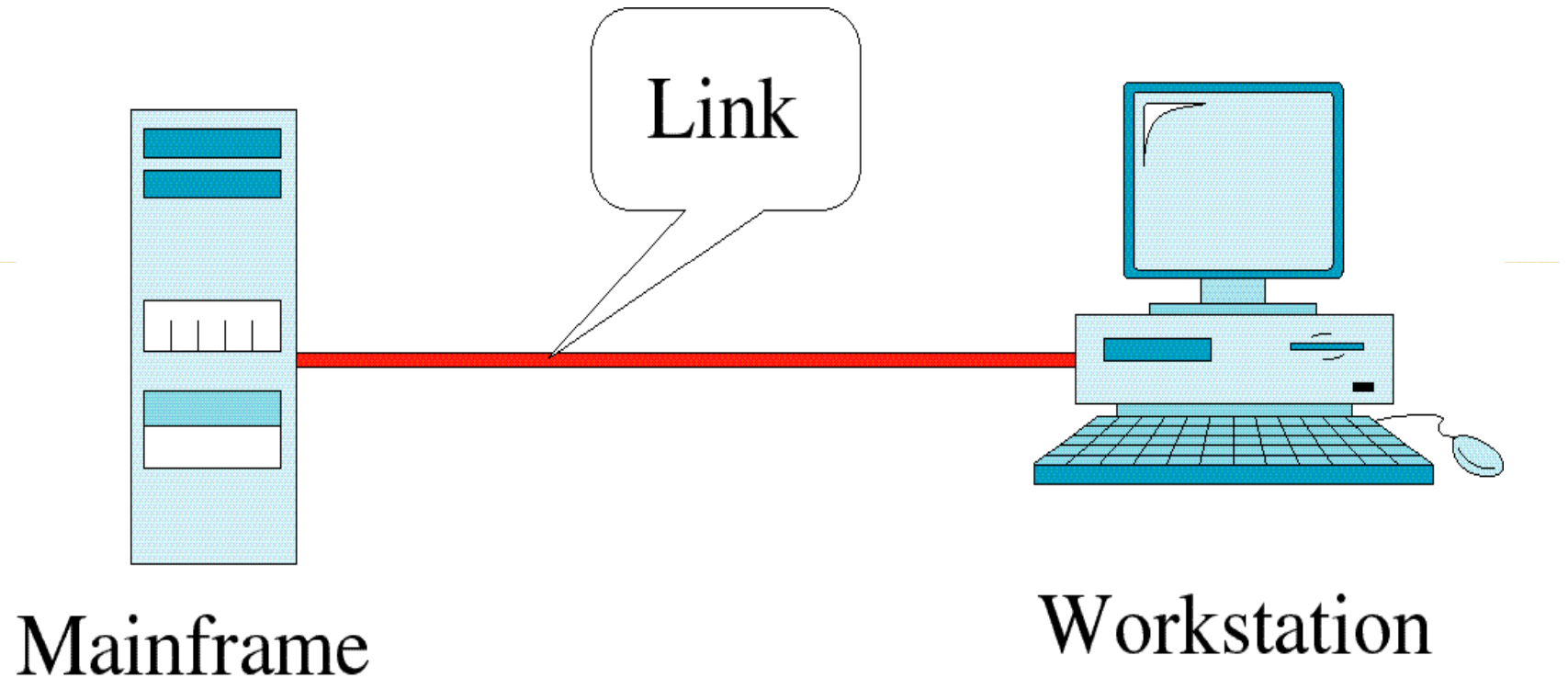
Point-to-Point Line Configuration



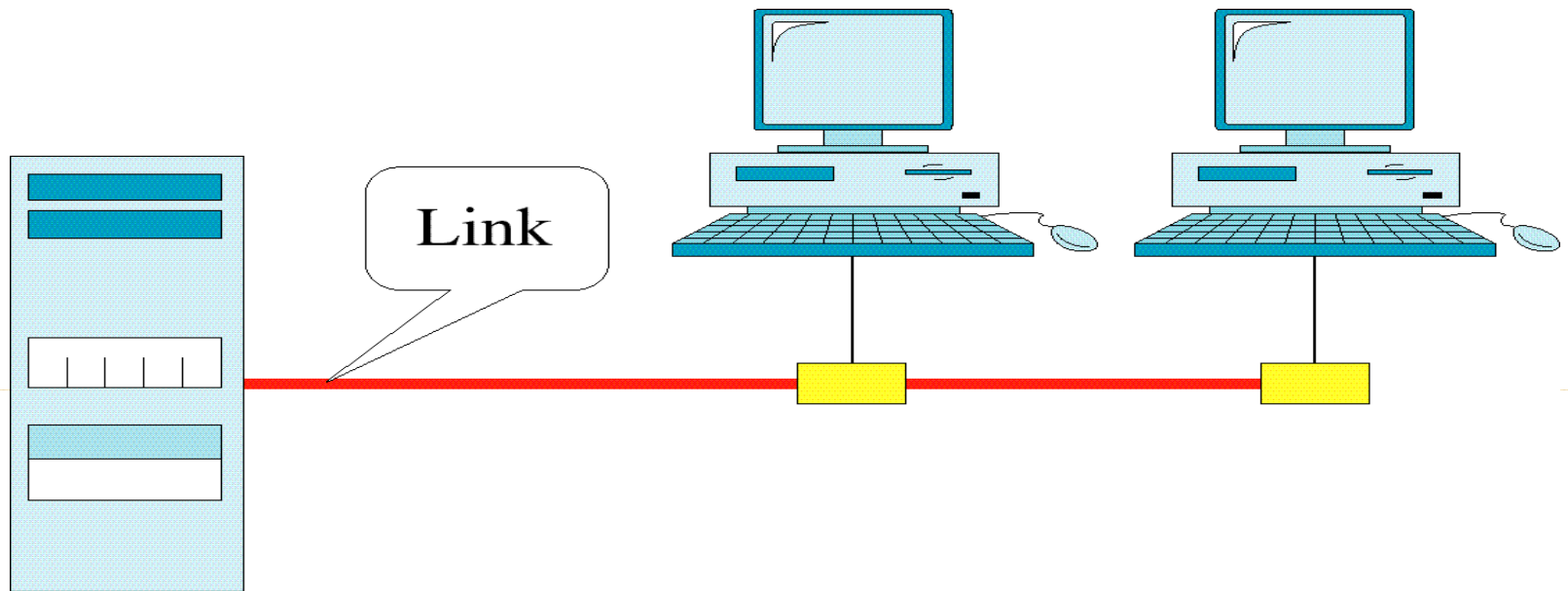
P2P provide a dedicated link b/w two devices. The entire capacity of the link is reserved for transmission between those devices

e.g when you change television channel by infrared remote control, you are establishing a point-to-point connection between the remote control and the television's control system.

Point-to-Point Line Configuration



Multipoint Connection



A **multipoint** (also called **Multidrop**) connection is one in which more than two specific devices share a single link.

In a multipoint environment, the capacity of the channel is shared, either regularly or temporally. If several devices can use the link simultaneously it is a shared connection.



Direction of Data Flow

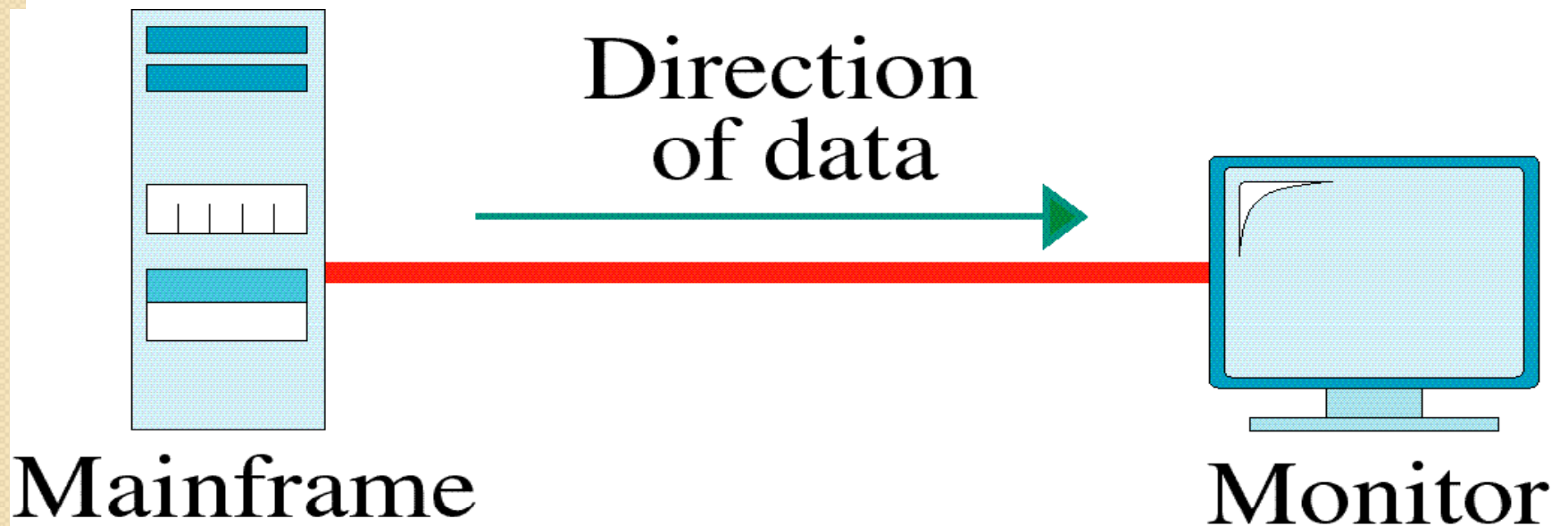
Communication between two devices can be

-
- ✓ simplex
 - ✓ half-duplex
 - ✓ full-duplex

Simplex

In simplex mode, the communication is unidirectional, as on a one-way street. Only one of the two devices on a link can transmit, the other can only receive.

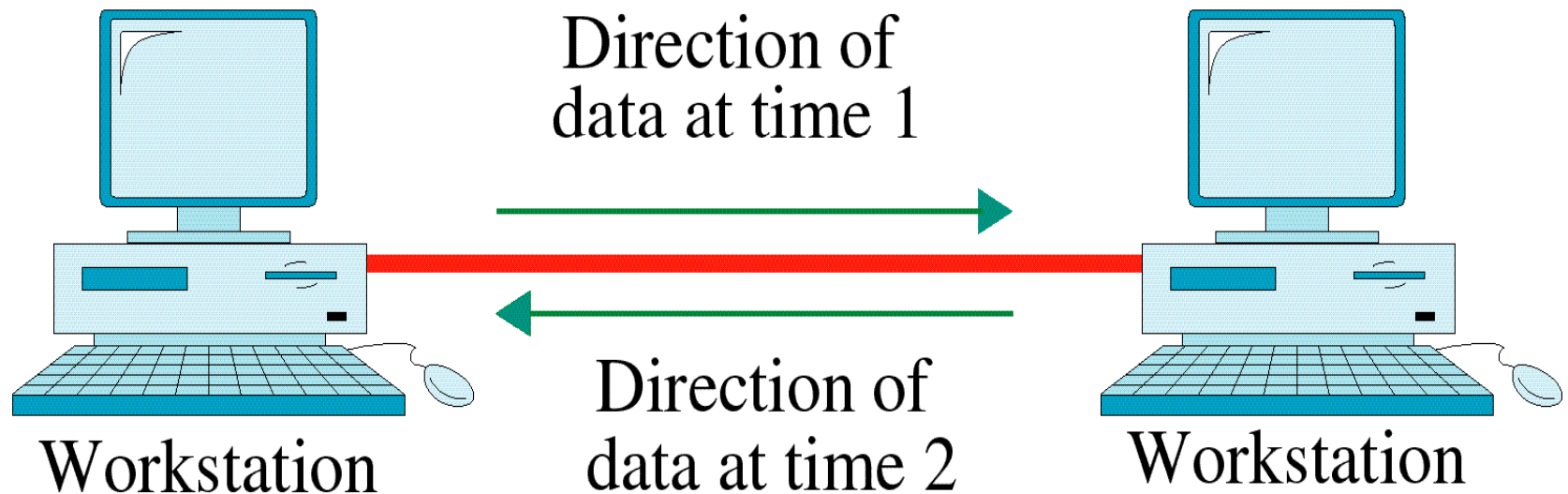
e.g Keyboards



Half-Duplex

In half- duplex mode, each station can transmit and receive ,but not at the same time. when one device is sending the other can only receive and vice versa.

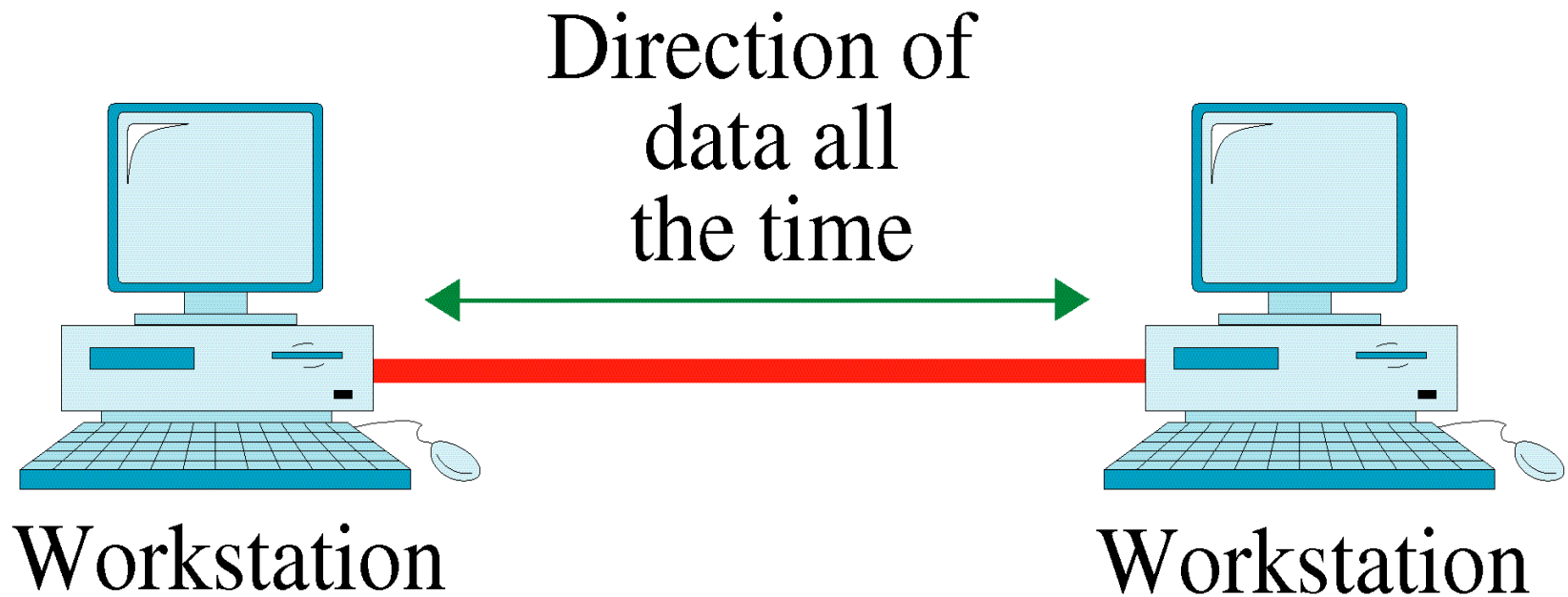
e.g walkies-talkies



Full-Duplex

In full-duplex mode both stations can transmit and receive simultaneously. The message flow in both directions at the same time.

e.g communication through chat .



Computer Network

• A **computer network** is a group of interconnected computers.

Two computers are said to be interconnected, if they are able to exchange information. The connection need not be via a copper wire, fiber optics, microwaves and satellites can also be used.

The Advanced Research Projects Agency (ARPA) designed “Advanced Research Project Agency Network”(ARPANET) for the United States Department of Defense. It was the first computer network in the world in late 1960s and early 1970s.

Categories of Networks

LAN

MAN

WAN



Network



```
graph TD; Network[Network] --> LAN[Local area network (LAN)]; Network --> MAN[Metropolitan area network (MAN)]; Network --> WAN[Wide area network (WAN)];
```

Local area
network
(LAN)

Metropolitan area
network
(MAN)

Wide area
network
(WAN)

LAN

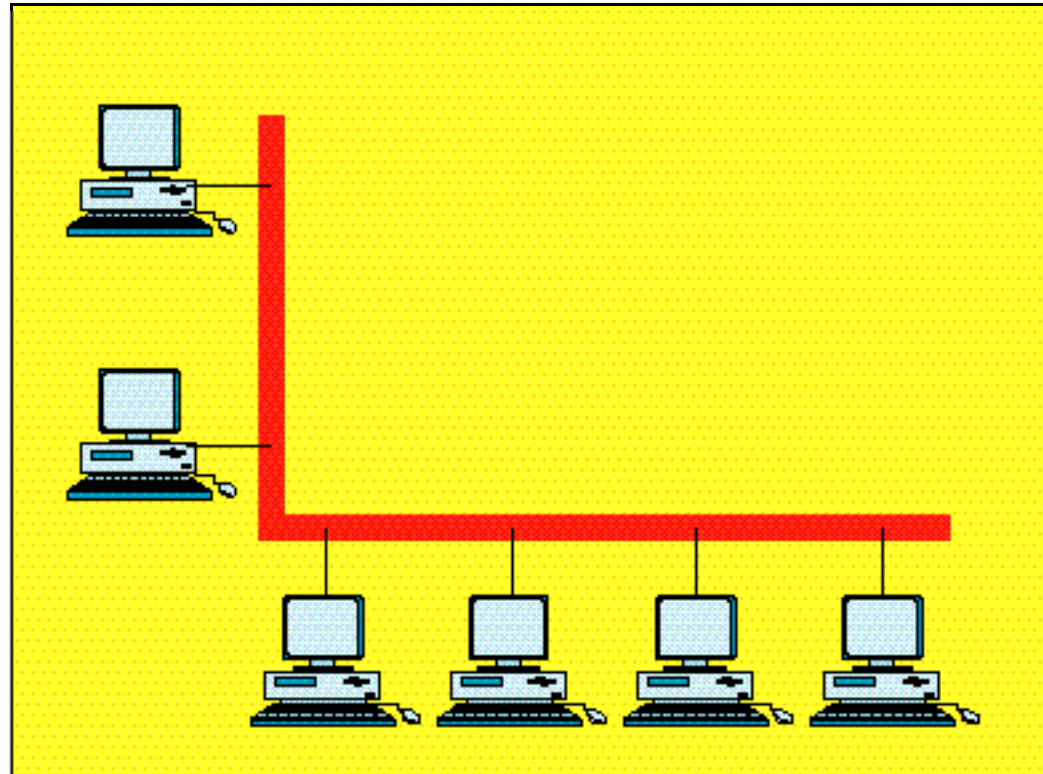
A local area network is a network used for connecting a business or organizations computers to one another.

A local area network usually links computers using a wired transmission medium over a area of about a hundred meters.

Components of a LAN

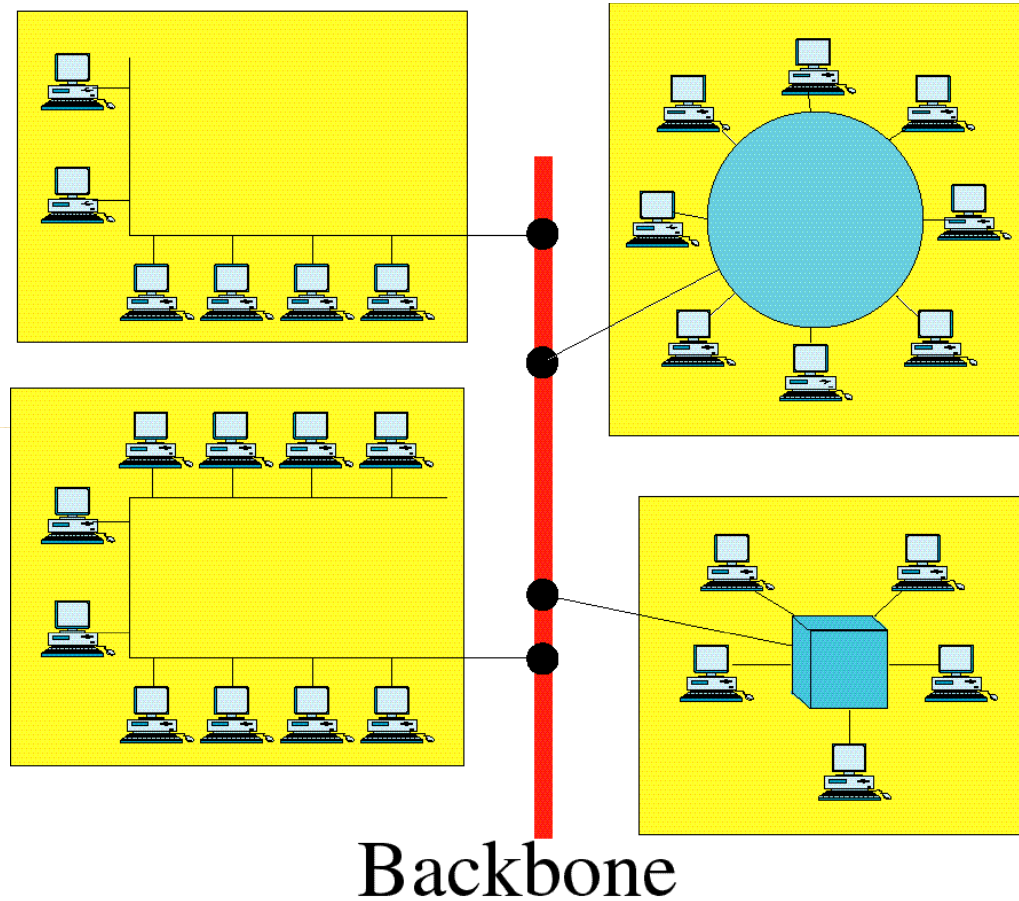
- The network card
- The transceiver
- The socket
- The physical connection medium
- Topologies

Local Area Network



Single building LAN

Local Area Network



Multiple building LAN



Connection Equipments

The primary hardware set up in local area networks is:

Repeaters

Hubs

Bridges

Switches

Gateways

Routers

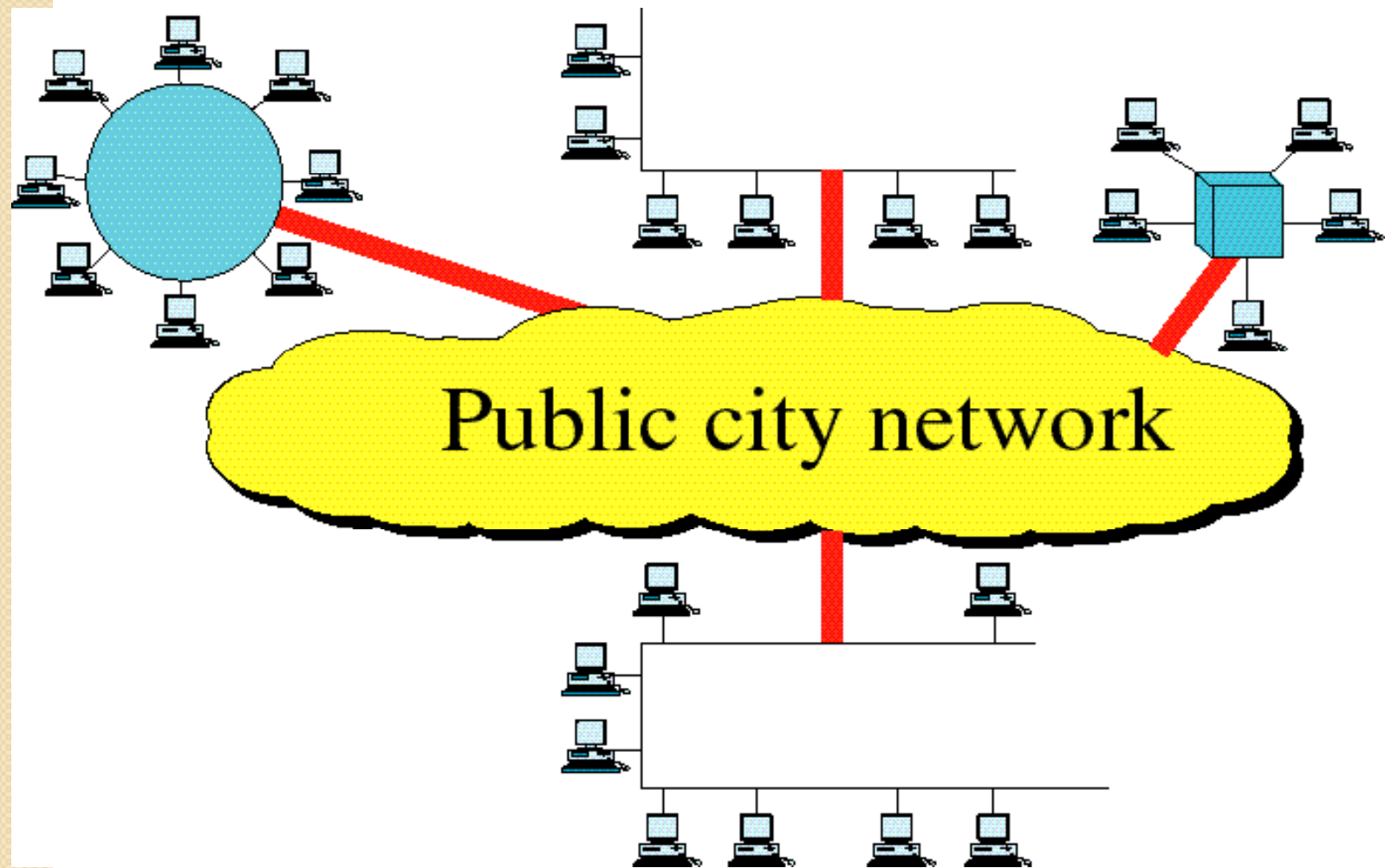


MAN

Metropolitan Area Network (MAN) is a computer networks usually spanning a campus or a city, which typically connect a few local area networks using high speed backbone technologies. A MAN often provides efficient connections to a wide area networks.

MAN adopted technologies from both LAN and WAN to serve its purpose.

Metropolitan Area Network



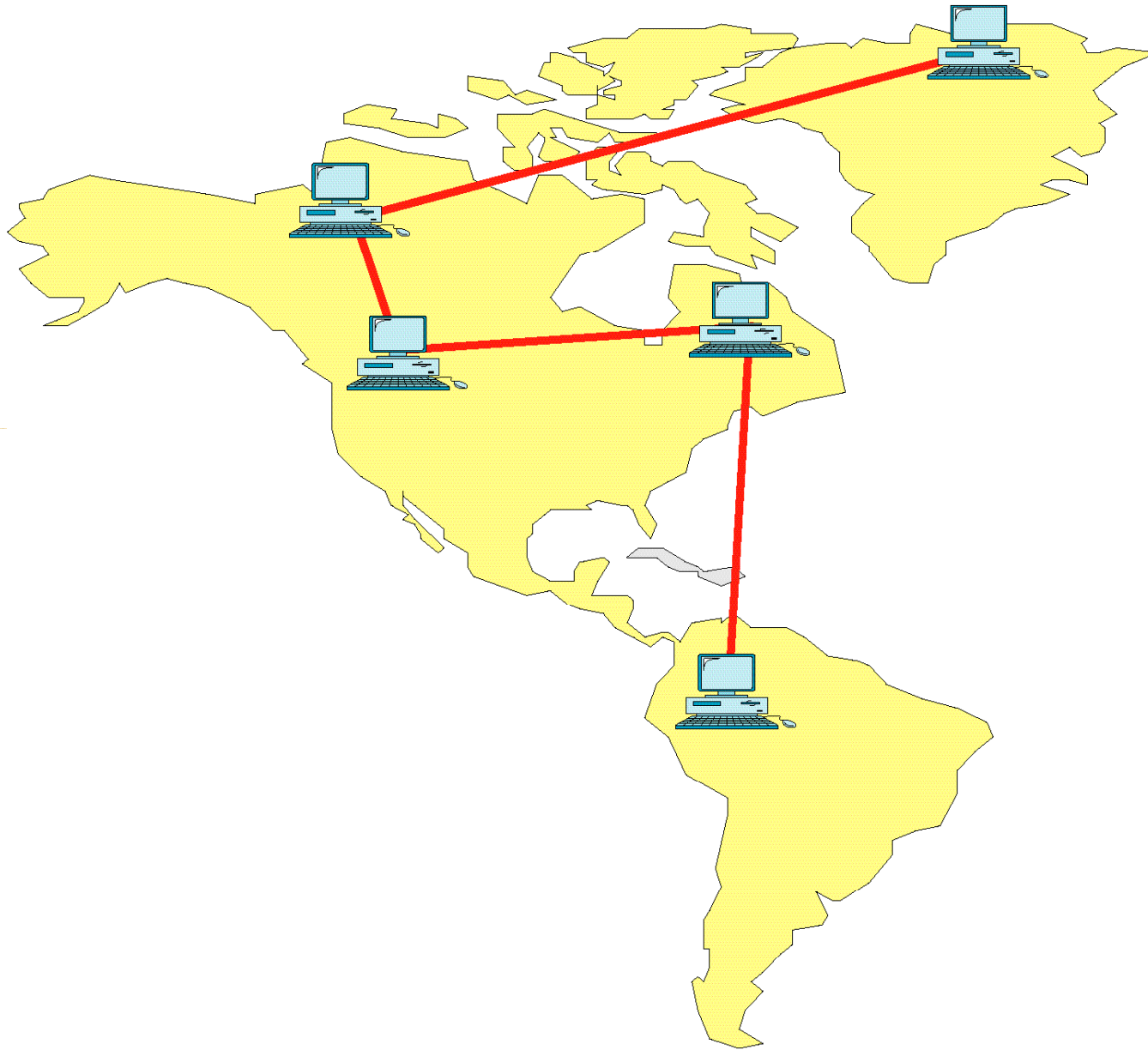


WAN

Wide Area Networks (WANs) connect large geographic areas such as United States, or the world. Dedicated cabling or satellite uplinks may be used to connect this type of network.

A WAN is complicated .it uses multiplexers to connect LAN and MAN to global communications networks like the internet.

Wide Area Network



Internetwork (Internet)

