MRS. LEENA J JACOB

CLASS : XI

SUBJECT : COMPUTER SCIENCE

NOTE:

***FOR NOW STUDENTS NEED NOT WORRY ABOUT THE TEXT BOOK OR COPY.**

***CONTENTS OF THE THIS FILE SHOULD BE WRITTEN IN ANY ROUGH COPY.**

UNIT 1

CHAPTER 4 - INSIGHT IN TO PROGRAM EXECUTION

CONTENTS -

- **Basic concepts of Flowchart**
- **Concept of Compiler & Interpreter**
- **Running a program: Notion of an operating system, how an operating system runs a program, idea of loading, operating system as a resource manager.**
- **Concept of cloud computing, cloud (public/private), introduction to parallel computing.**

FLOWCHART :

A flowchart is basically pictorial or diagrammatic representation of an algorithm using standard symbols. In other words, it is a graphical representation that explains the sequence of operations to be performed in order to solve a problem under consideration.

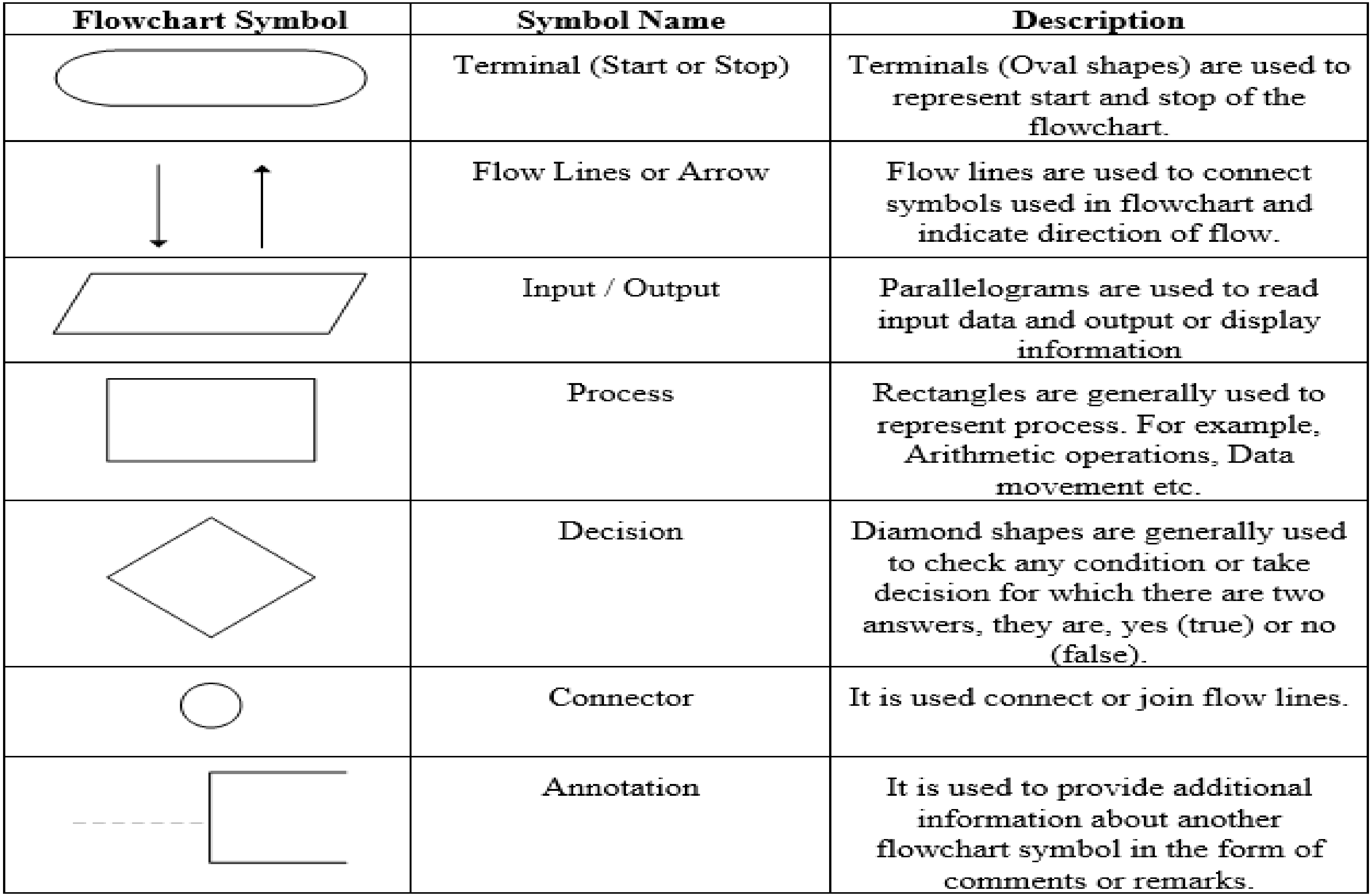
Following guidelines must be followed while preparing the flowcharts:

- **Standard symbols should be used while drawing flowchart.**
- **Ensure that flowchart has START (or BEGIN) and STOP (or END).**
- **Flowchart should be neat, clean and easy to follow. There should be no any ambiguity.**
- **The usual direction of flowchart is from top to bottom or from left to right.**
- **The terminal symbol, that is, START/BEGIN or STOP/END should have only one flow line.**
- **Only one flow line should come out from process symbol.**
- **Only one flow line should enter a decision symbol, but two or three flow-lines, one for each possible answer, can leave the decision symbol.**
- **If the flowchart is lengthy and complex connector symbol should be used to reduce the number of flow lines. Avoid intersection of flow lines. .**
- **Use annotation symbol to describe steps more clearly.**

Standard Flowchart Symbols

To express different operations in the flowchart various standard symbols are used. All symbols are connected among themselves in order to show the flow of information and processing.

Different symbols as prescribed by American National Standard Institute (ANSI) which are frequently required while drawing flowchart are tabulated below:



Flowchart Symbol	Symbol Name	Description
⬭	Terminal (Start or Stop)	Terminals (Oval shapes) are used to represent start and stop of the flowchart.
↓ ↑	Flow Lines or Arrow	Flow lines are used to connect symbols used in flowchart and indicate direction of flow.
▱	Input / Output	Parallelograms are used to read input data and output or display information
▭	Process	Rectangles are generally used to represent process. For example, Arithmetic operations, Data movement etc.
◇	Decision	Diamond shapes are generally used to check any condition or take decision for which there are two answers, they are, yes (true) or no (false).
○	Connector	It is used connect or join flow lines.
- - - [	Annotation	It is used to provide additional information about another flowchart symbol in the form of comments or remarks.

CONCEPT OF INTERPRETER AND COMPILER: Basically there are three types of translators interpreter , complier and assembler.

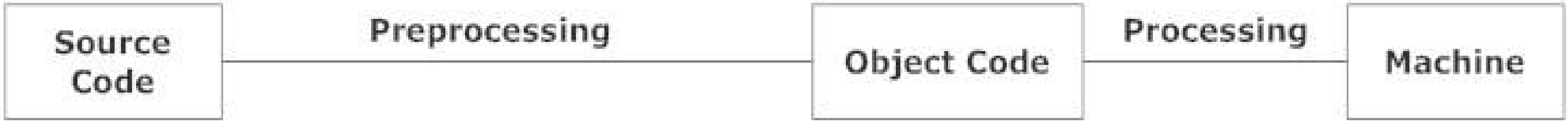


Figure: Compiler

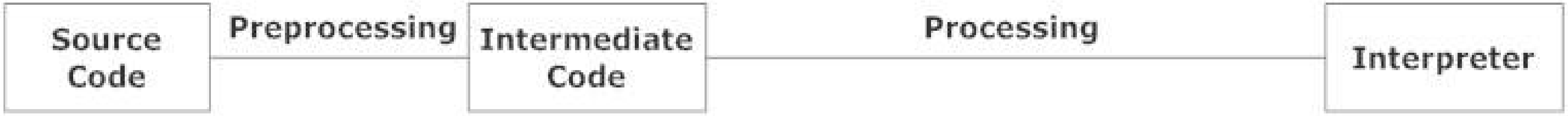
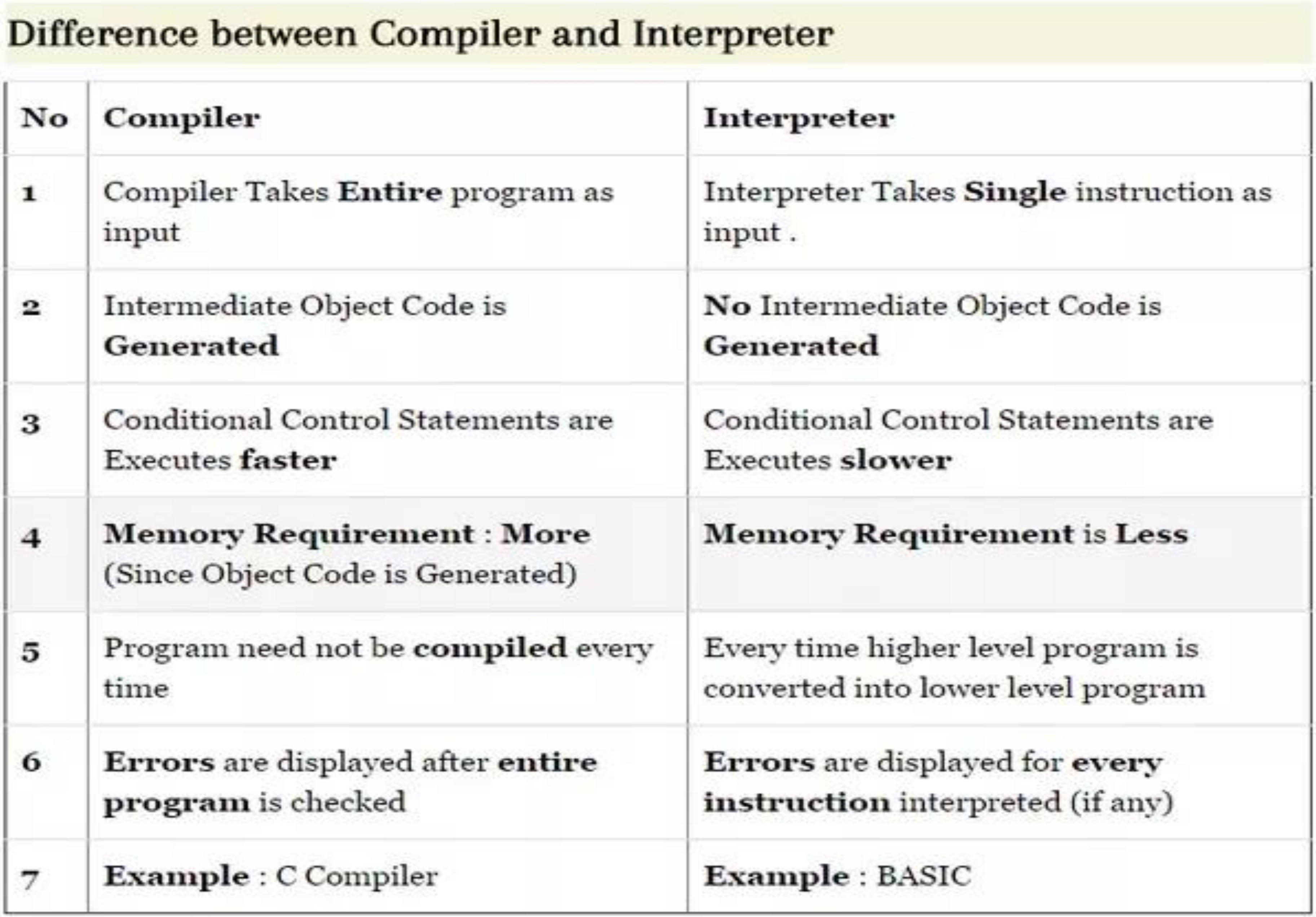


Figure: Interpreter

COMPILER VS INTERPRETER VS ASSEMBLER

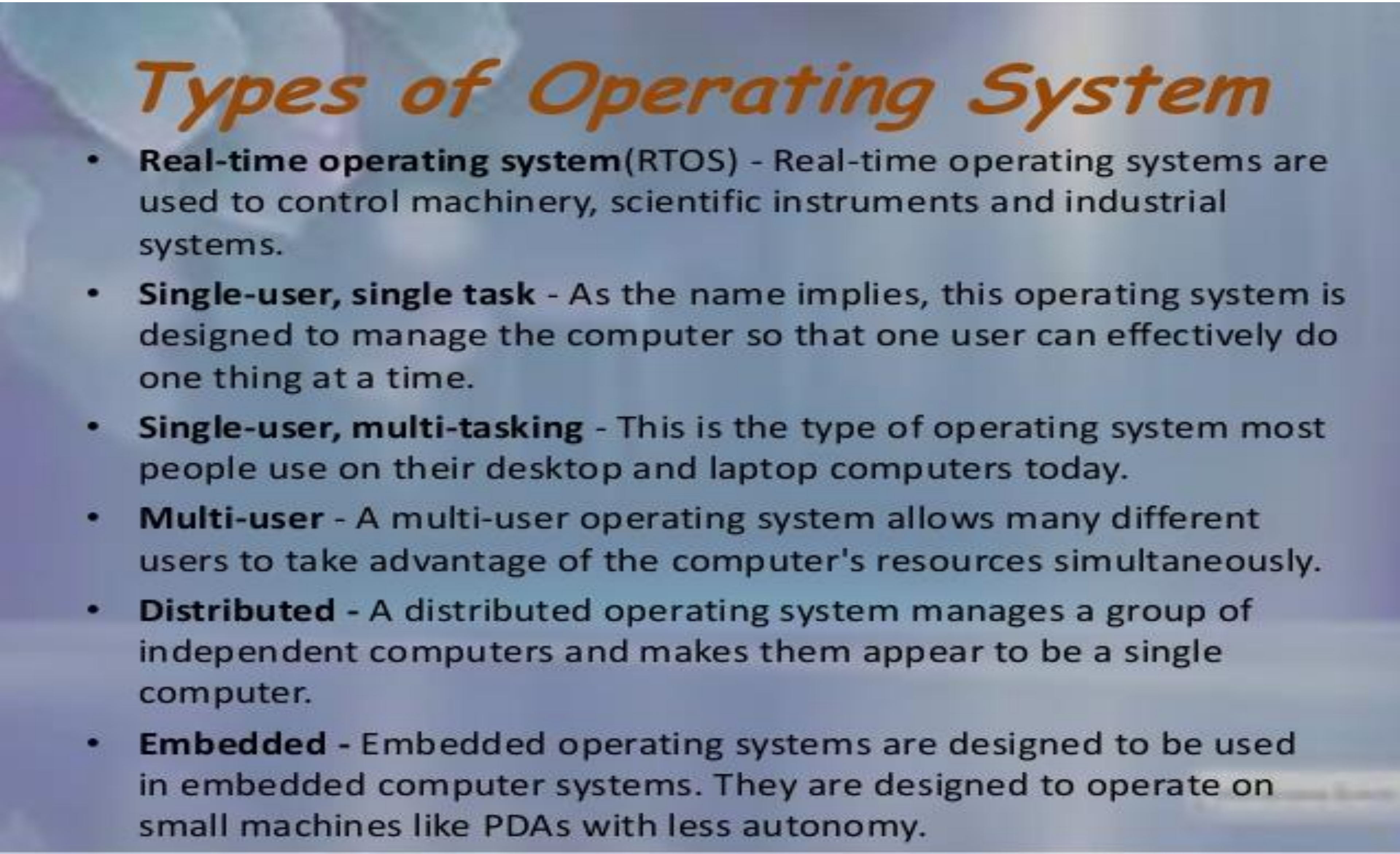
Software that converts programs written in a high level language into machine language	Software that translates a high level language program into machine language	Software that converts programs written in assembly language into machine language
Converts the whole high level language program to machine language at a time	Converts the high level language program to machine language line by line	Converts assembly language program to machine language
Used by C, C++	Used by Ruby, Perl, Python, PHP	Used by assembly language



Difference between Compiler and Interpreter

No	Compiler	Interpreter
1	Compiler Takes **Entire** program as input	Interpreter Takes **Single** instruction as input .
2	Intermediate Object Code is **Generated**	**No** Intermediate Object Code is **Generated**
3	Conditional Control Statements are Executes **faster**	Conditional Control Statements are Executes **slower**
4	**Memory Requirement** : **More** (Since Object Code is Generated)	**Memory Requirement** is **Less**
5	Program need not be **compiled** every time	Every time higher level program is converted into lower level program
6	**Errors** are displayed after **entire program** is checked	**Errors** are displayed for **every instruction** interpreted (if any)
7	**Example** : C Compiler	**Example** : BASIC

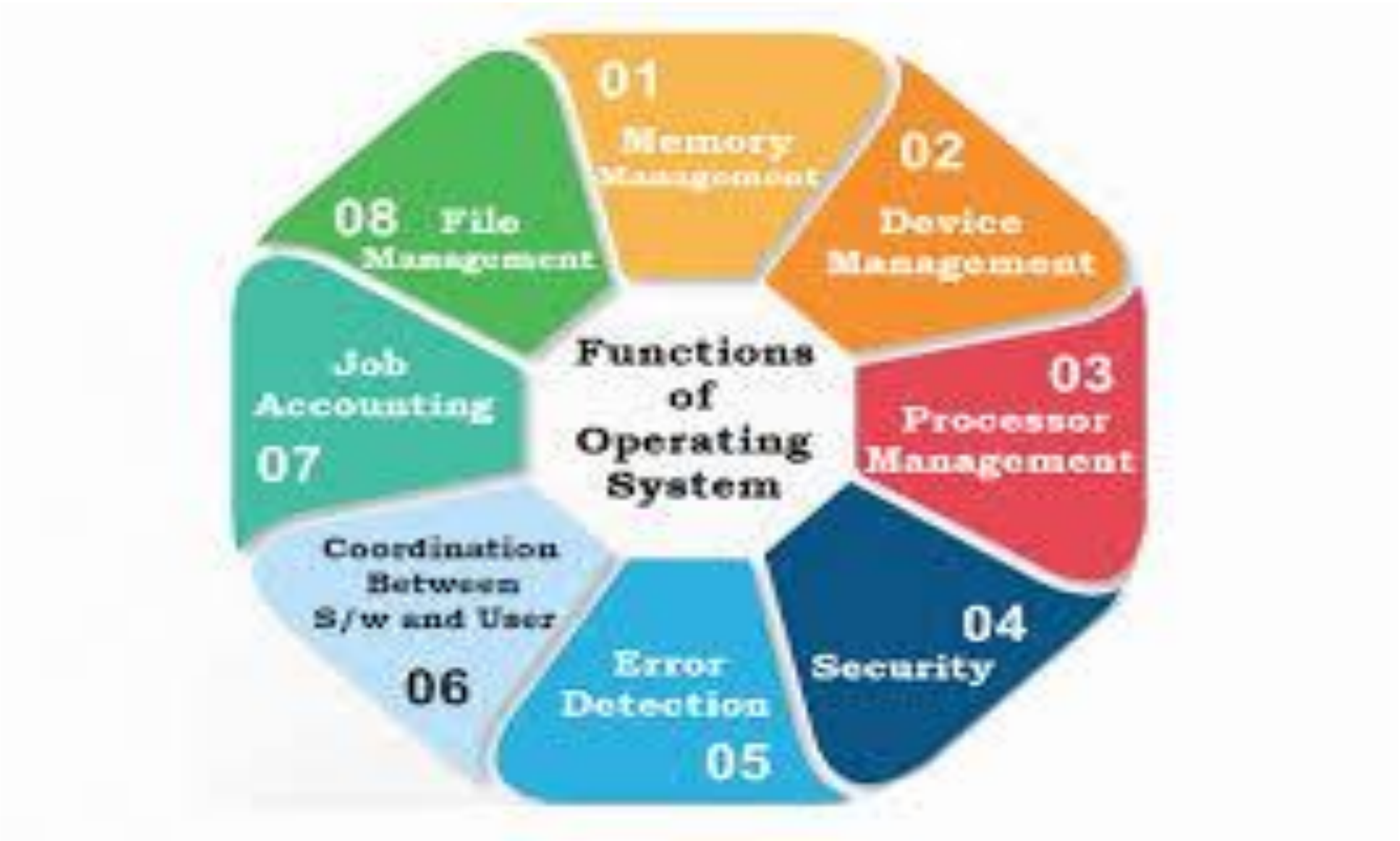
An Operating System (OS) is an interface between a computer user and computer hardware. An operating system is a software which performs all the basic tasks like file management, memory management, process management, handling input and output, and controlling peripheral devices such as disk drives and printers.



Types of Operating System

- **Real-time operating system**(RTOS) - Real-time operating systems are used to control machinery, scientific instruments and industrial systems.
- **Single-user, single task** - As the name implies, this operating system is designed to manage the computer so that one user can effectively do one thing at a time.
- **Single-user, multi-tasking** - This is the type of operating system most people use on their desktop and laptop computers today.
- **Multi-user** - A multi-user operating system allows many different users to take advantage of the computer's resources simultaneously.
- **Distributed -** A distributed operating system manages a group of independent computers and makes them appear to be a single computer.
- **Embedded -** Embedded operating systems are designed to be used in embedded computer systems. They are designed to operate on small machines like PDAs with less autonomy.

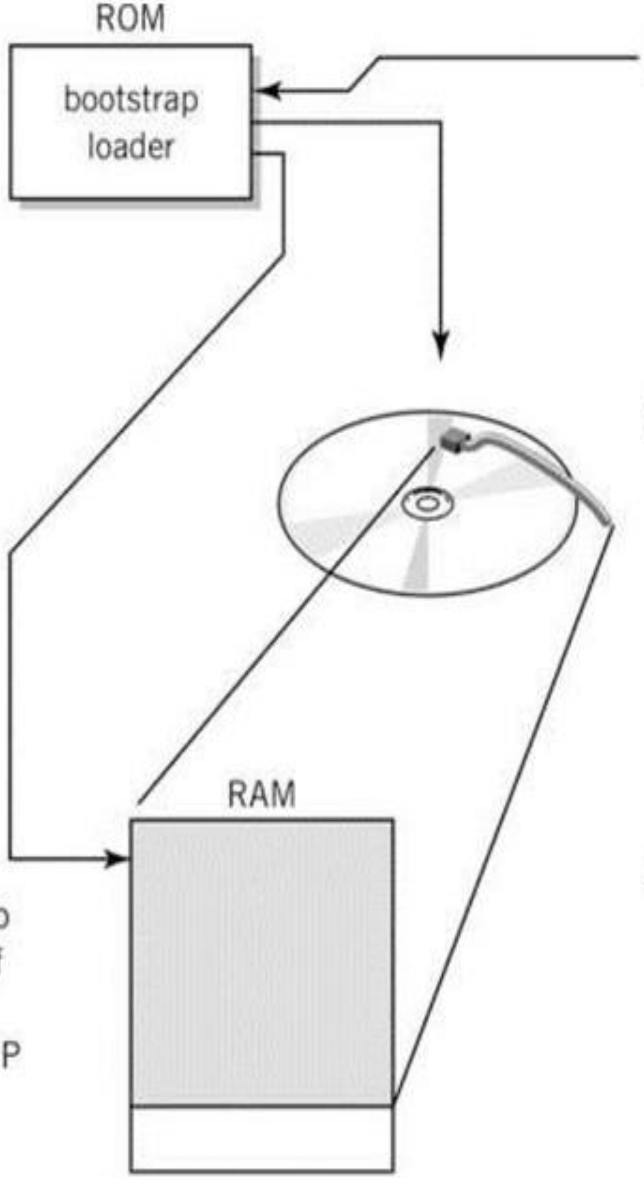
MAIN FUNCTIONS OF AN OPERATING SYSTEM



FUNCTIONS OF AN OPERATING SYSTEM

- Main functions of an operating system include:
  - ***Process Management :*** allowing processes to co-exist in the system, allocating job priorities, scheduling processor time
  - ***Memory Management:*** allocating main memory space to each process, protecting processes from one another, enabling different processes to share the same memory.
  - ***Input/output control:*** Channelling data to and from peripherals of the computer
  - ***File Management:*** allocating space on secondary storage media, keep track of data stored on secondary storage, control file access rights and privileges
  - ***Error handling and protection:*** The detection and reporting of errors and minimising their effect. Also it involves protection against errors and against deliberate abuse of the system.

STARTING THE COMPUTER SYSTEM: THE BOOTSTRAP



1. When computer is started, execution begins with bootstrap loader, permanently stored in ROM.

2. Bootstrap loader locates operating system program, usually at a fixed disk location.

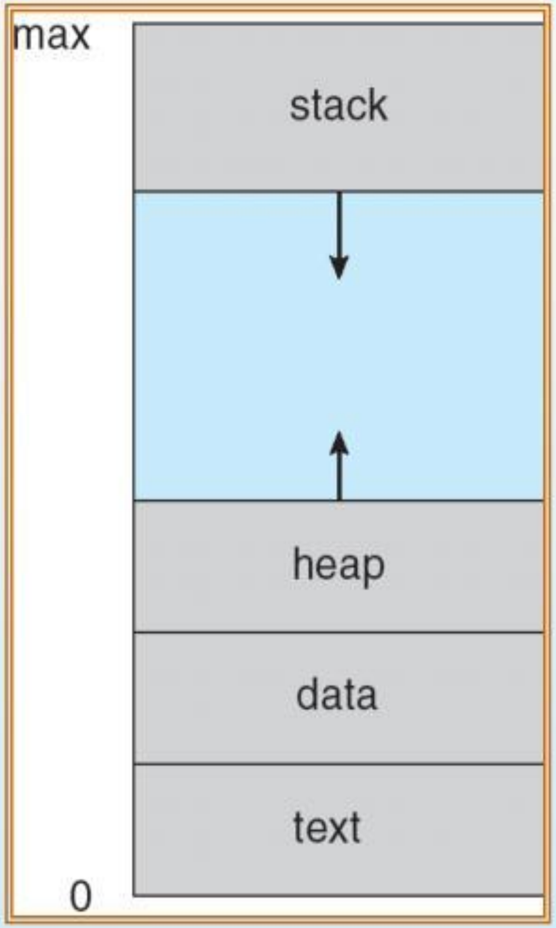
3. Loads it into RAM.

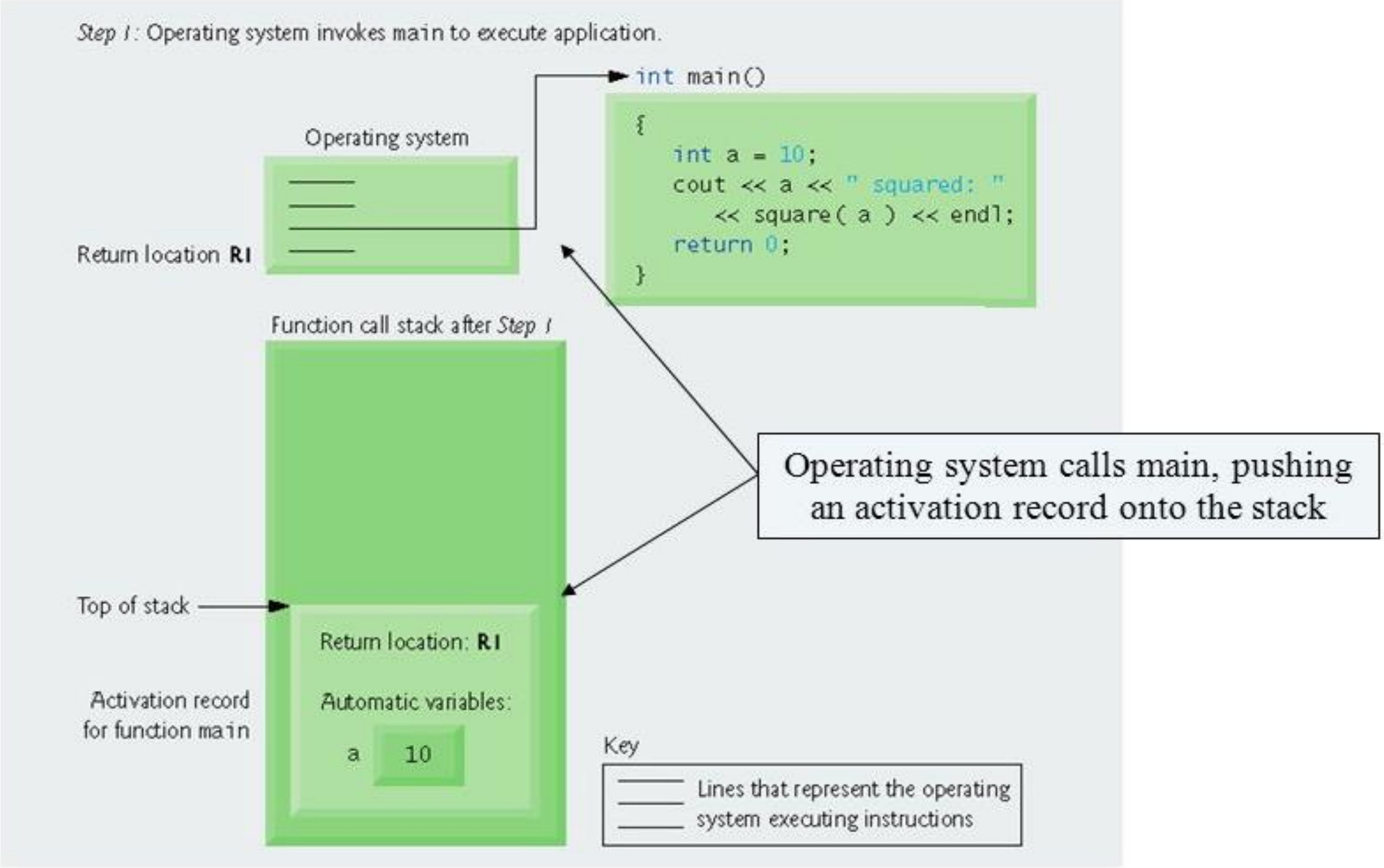
4. Transfers control to starting location of operating system program with a JMP instruction.

Note: Loader program in OS can then be used to load and execute user programs.

Process Concept

- An operating system executes a variety of programs:
  - Batch system – jobs
  - Time-shared systems – user programs or tasks
- Textbook uses the terms *job* and *process* almost interchangeably
- Process – a program in execution; process execution must progress in sequential fashion
- A process includes:
  - program counter
  - stack
  - data section





Step 1: Operating system invokes main to execute application.
Operating system
Return location R1
Function call stack after Step 1
Top of stack
Return location: R1
Automatic variables:
a
10
Activation record for function main
Key
Lines that represent the operating system executing instructions

```
int main()
{
   int a = 10;
   cout << a << " squared: "
      << square( a ) << endl;
   return 0;
}
```

Operating system calls main, pushing an activation record onto the stack

Function call stack after the operating system invokes `main` to execute the application.

OPERATING SYSTEM ACTS AS A RESOURCE MANAGER

Modern computers consist of processors, memories, timers, disks, mice, network interfaces, printers, and a wide variety of other devices. In the alternative view, the job of the operating system is to provide for an orderly and controlled allocation of the processors, memories, and input/output devices among the various programs competing for them.

When a computer (or network) has multiple users, the need for managing and protecting the memory, input/output devices, and other resources is even greater, since the users might otherwise interface with one another. In addition, users often need to share not only hardware, but information (files, databases, etc.) as well. In short, this view of the operating system holds that its primary task is to keep track of which programs are using which resources, to grant resource requests, to account for usage, and to mediate conflicting requests from different programs and users.

Resource management includes multiplexing (sharing) resources in two different ways:

Time Multiplexing

Space Multiplexing

1. Time Multiplexing

When the resource is time multiplexed, different programs or users take turns using it. First one of them gets to use the resource, then another, and so on.

For example:

With only one CPU and multiple programs that want to run on it, operating system first allocates the CPU to one long enough, another one gets to use the CPU, then another and ten eventually the first one again.

Determining how the resource is time multiplexed – who goes next and for how long – is the task of the operating system.

2. Space Multiplexing

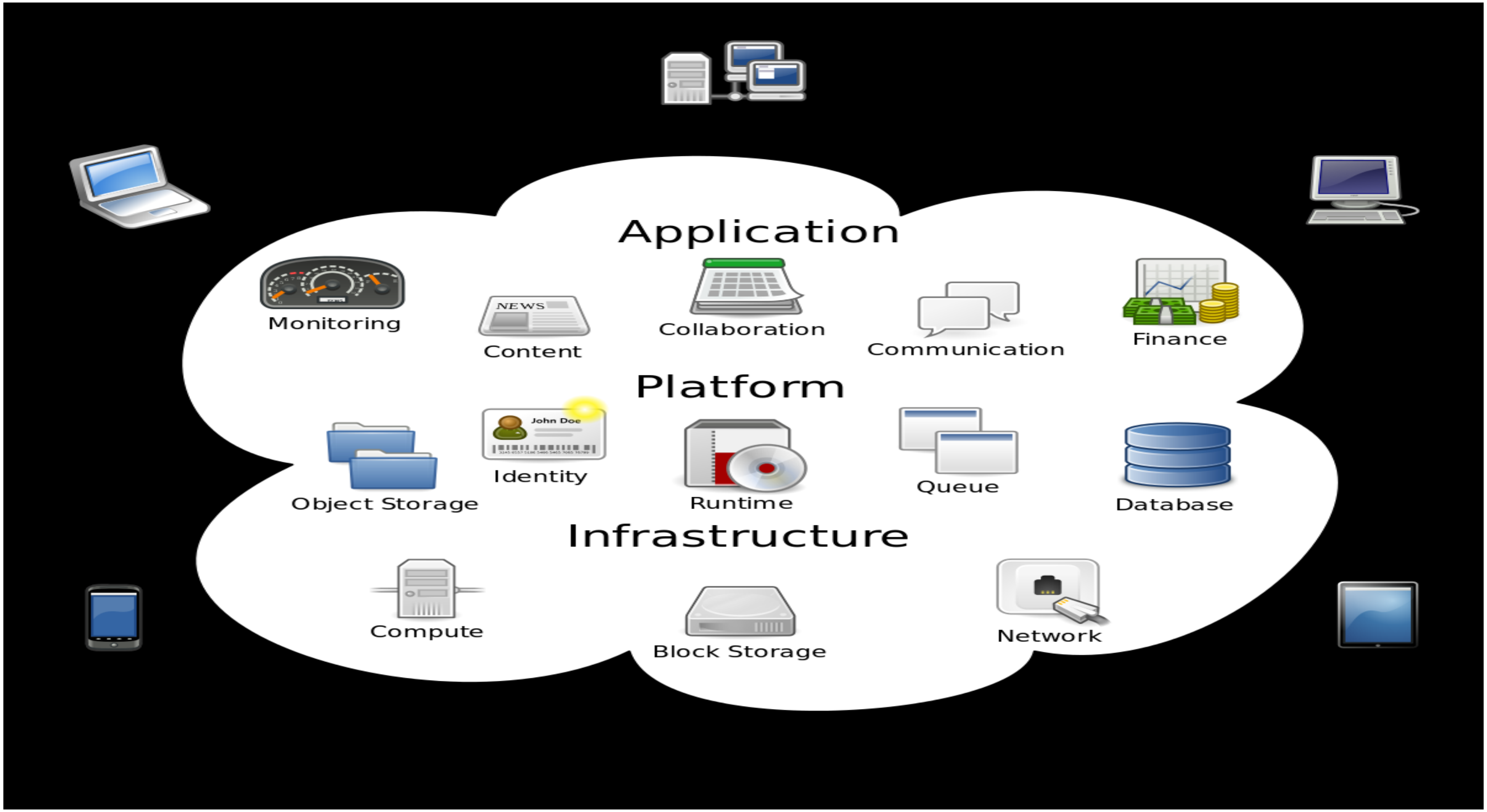
In space multiplexing, instead of the customers taking turns, each one gets part of the resource.

For example:

Main memory is normally divided up among several running programs, so each one can be resident at the same time (for example, in order to take turns using the CPU). Assuming there is enough memory to hold multiple programs, it is more efficient to hold several programs in memory at once rather than give one of them all of it, especially if it only needs a small fraction of the total. Of course, this raises issues of fairness, protection, and so on, and it is up to the operating system to solve them.

CONCEPT OF CLOUD COMPUTING, CLOUD (PUBLIC/PRIVATE)

Cloud computing is a type of computing that relies on shared computing resources rather than having local servers or personal devices to handle applications. In its most simple description, cloud computing is taking services ("cloud services") and moving them outside an organization's firewall.

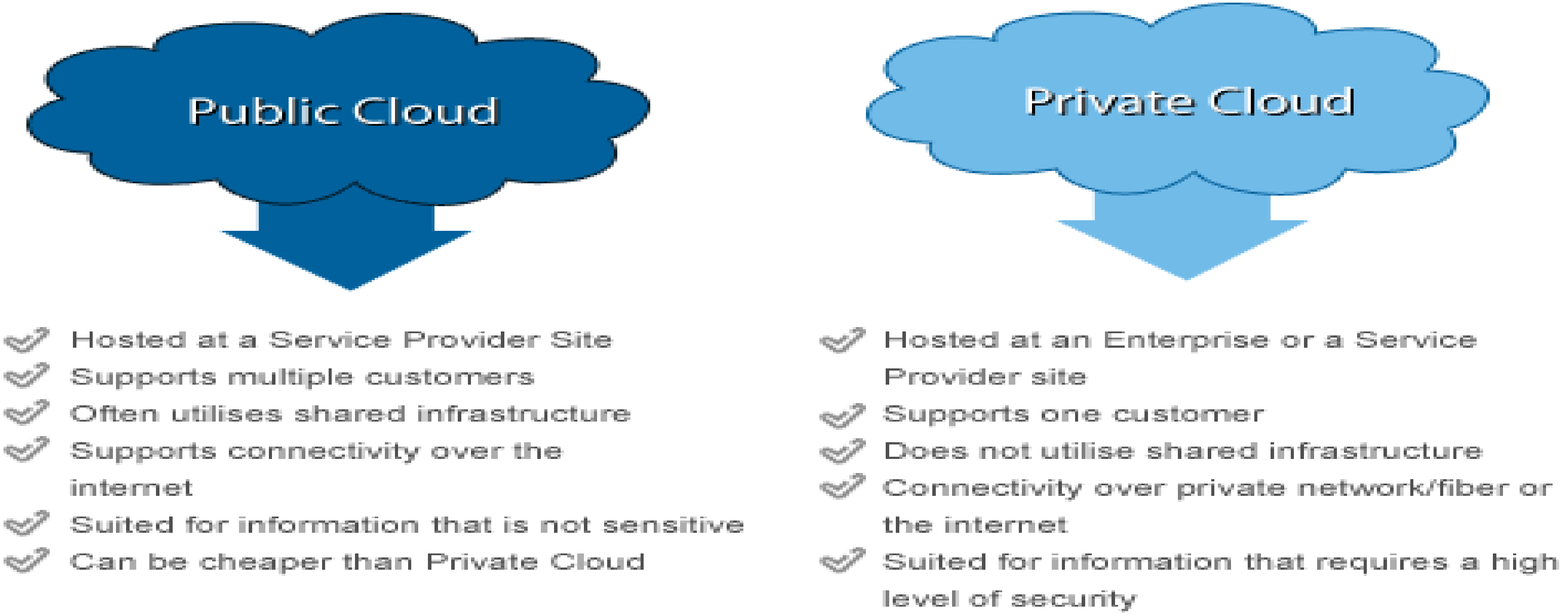


PRIVATE CLOUD

The private cloud is defined as computing services offered either over the Internet or a private internal network and only to select users instead of the general public. ... So private clouds require the same staffing, management and maintenance expenses as traditional datacenter ownership.

PUBLIC CLOUD

The public cloud is defined as computing services offered by third-party providers over the public Internet, making them available to anyone who wants to use or purchase them. They may be free or sold on-demand, allowing customers to pay only per usage for the CPU cycles, storage, or bandwidth they consume.



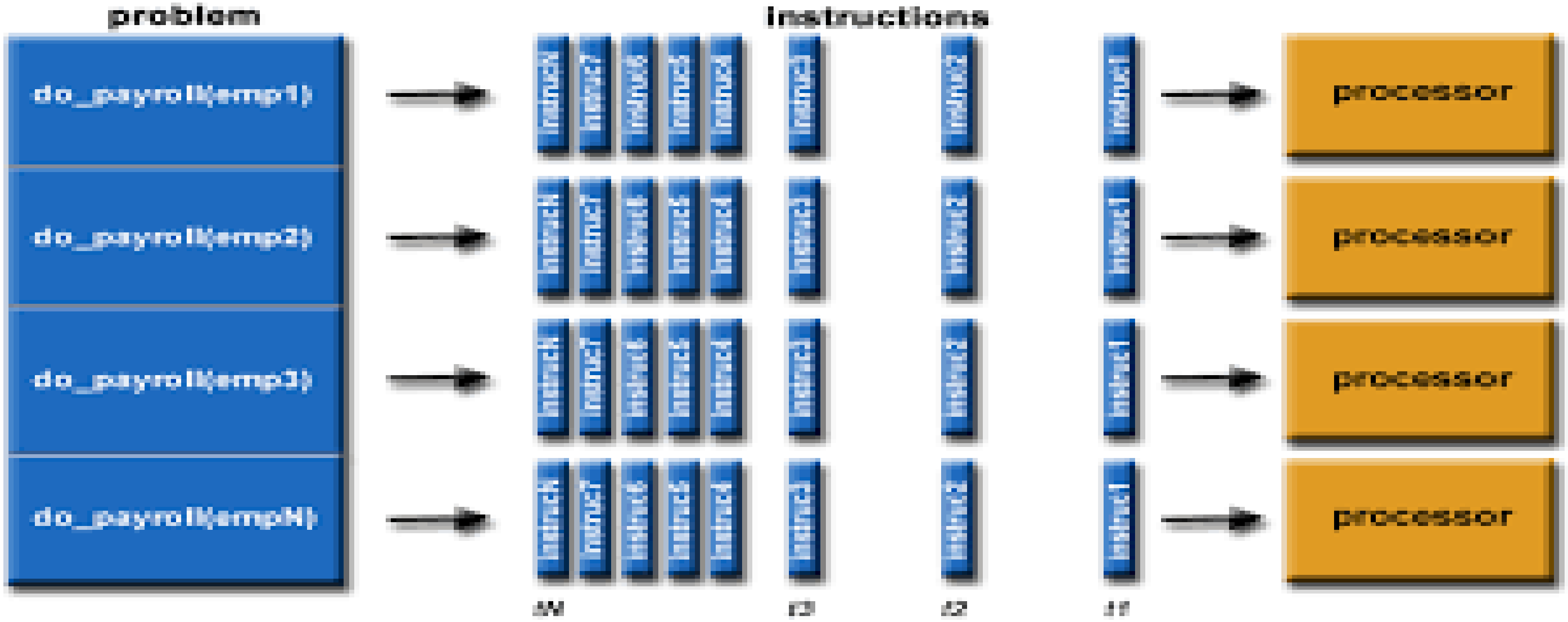
- ✓ Hosted at a Service Provider Site
- ✓ Supports multiple customers
- ✓ Often utilises shared infrastructure
- ✓ Supports connectivity over the internet
- ✓ Suited for information that is not sensitive
- ✓ Can be cheaper than Private Cloud

- ✓ Hosted at an Enterprise or a Service Provider site
- ✓ Supports one customer
- ✓ Does not utilise shared infrastructure
- ✓ Connectivity over private network/fiber or the internet
- ✓ Suited for information that requires a high level of security

INTRODUCTION TO PARALLEL COMPUTING:

Parallel computing stands for the ability of computer systems to perform multiple operations simultaneously. The main driver behind parallel computing is the fact that large problems can be divided to smaller ones which can be then solved in parallel — i.e. executed concurrently on the available computing resources.

Some examples of parallel computing include weather forecasting, movie special effects, and desktop computer applications.



****