

# Insight into Program execution

Based on CBSE curriculum  
Class 11



## Chapter - 15

By-

Neha Tyagi

PGT CS

KV 5 Jaipur II Shift, Jaipur Region

# Introduction

- As we know that computer works on machine level language which is not understandable to a programmer, similarly programmer works with a language which is not understandable by a computer system.
- Programmer uses a specific language to write a program like – C, C++, Java etc. These languages are not directly understandable by computer system.
- Therefore, it is required to convert the instructions written in programmer's language into machine level language.
- In this chapter, we will have a look of internal processing of a program code, how a program reaches to execution state and what are the contributions of an operating system in all these processes.

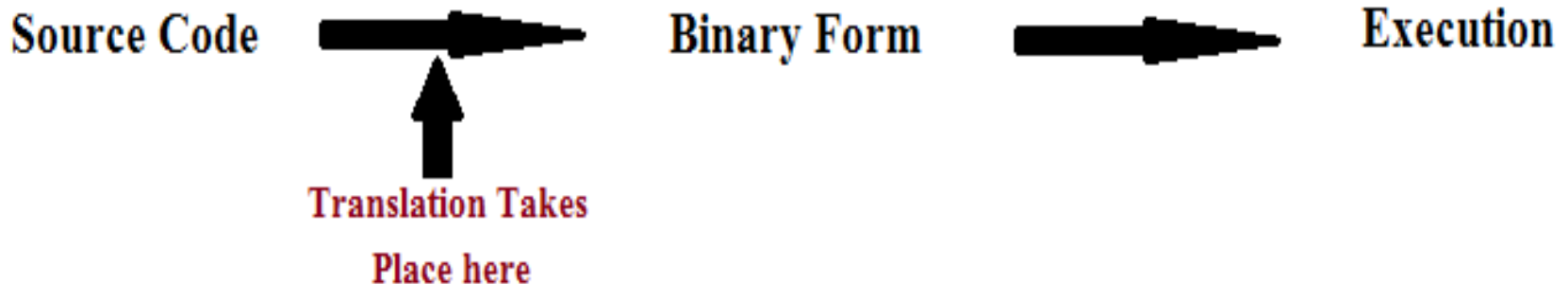
# Basic Flow of Compilation

- Program written in a language is known as *source code*.
- Source code is to be written as per the rules of a programming language.
- This source code is then converted to machine level language, known as *binary code*, in order to be understood by a computer.
- We can say that a programmer develops a source code which is later converted into binary code so that it can be executed by a computer.



# Translator

- A special software is needed to convert a Source Code into binary code which are as under-
  - Compiler
  - Interpreter
- Both these software converts source code into binary code but their way of working is different.
- Compiler works in exactly opposite manner from interpreter.



# Compilation

Compiler takes source code and in few stages generates low-level-code. Compilation process stages are-

1. Processing
2. Compilation
  - I. Analysis [ front end phase ]
  - II. Synthesis [ back end phase ]
3. Assembly
4. Linking
5. Loader

## 1. Processing

This stage removes code other than source code like comments and adds the code which is necessary for the execution of a program.

# 2. Compilation

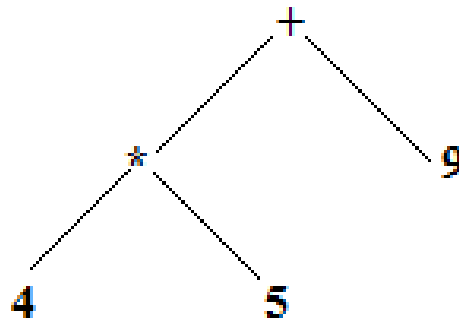
Compilation phase has two sub stages-

## 1. Analysis [ front end phase ]

### I. Synthesis [ back end phase ]

- (i) Analysis Phase : Analysis phase develops a symbol table by all the tokens of source code.
- (ii) Synthesis Phase: This phase creates a syntax tree after parsing the source code. A syntax tree is a syntactic tree of source code.

For ex- *syntax tree of  $4 * 5 + 9$  will be like as -*



## 3. Assembly Phase

This phase develops object code with the help of instructions received from assembly level, an object code is a kind of machine code.

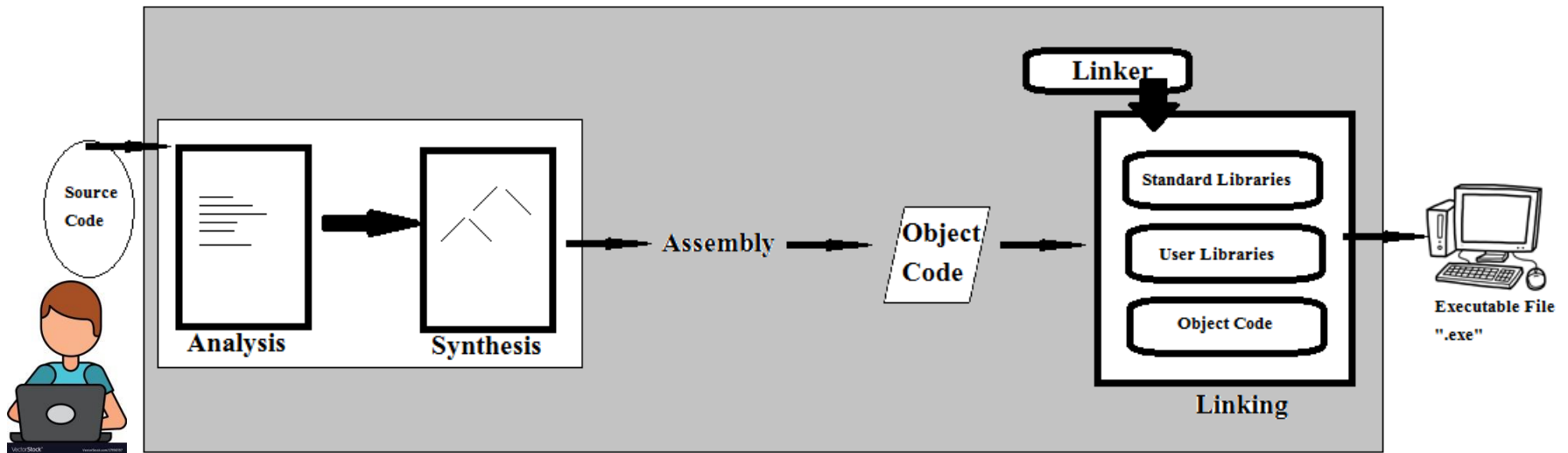
## 4. Linking

This code generated from this phase is actually binary code but still computer is not able to execute the code because the important libraries required for execution are still not linked with the code. This work is carried out by *linker*.

In this phase, operating system resolve all memory references and develops executable (.exe file) file.

# 5. Loader

- Loader is a part of compiler which loads an executable file in memory.
- Now computer can run this executable file without the help of any additional software.

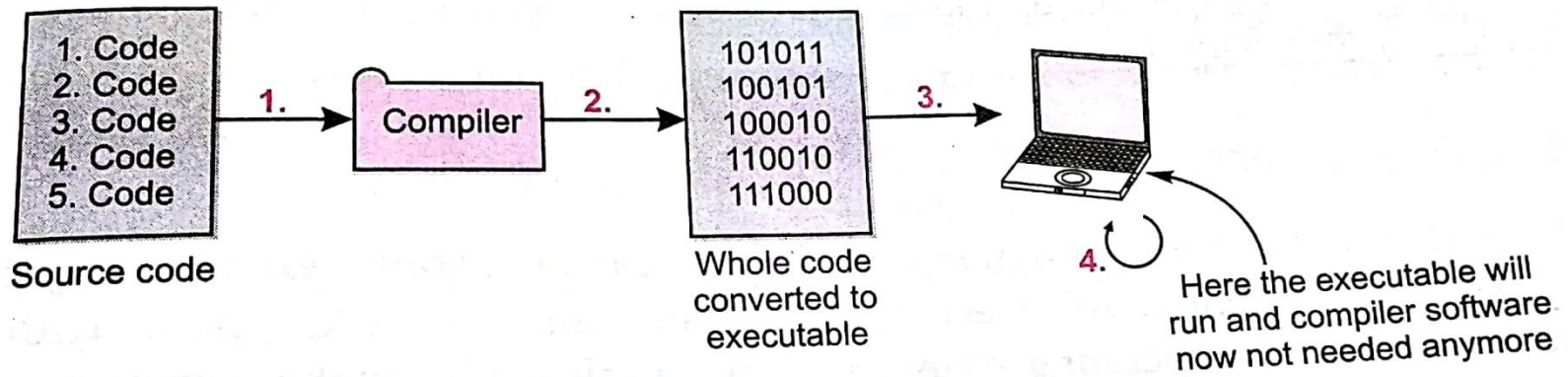




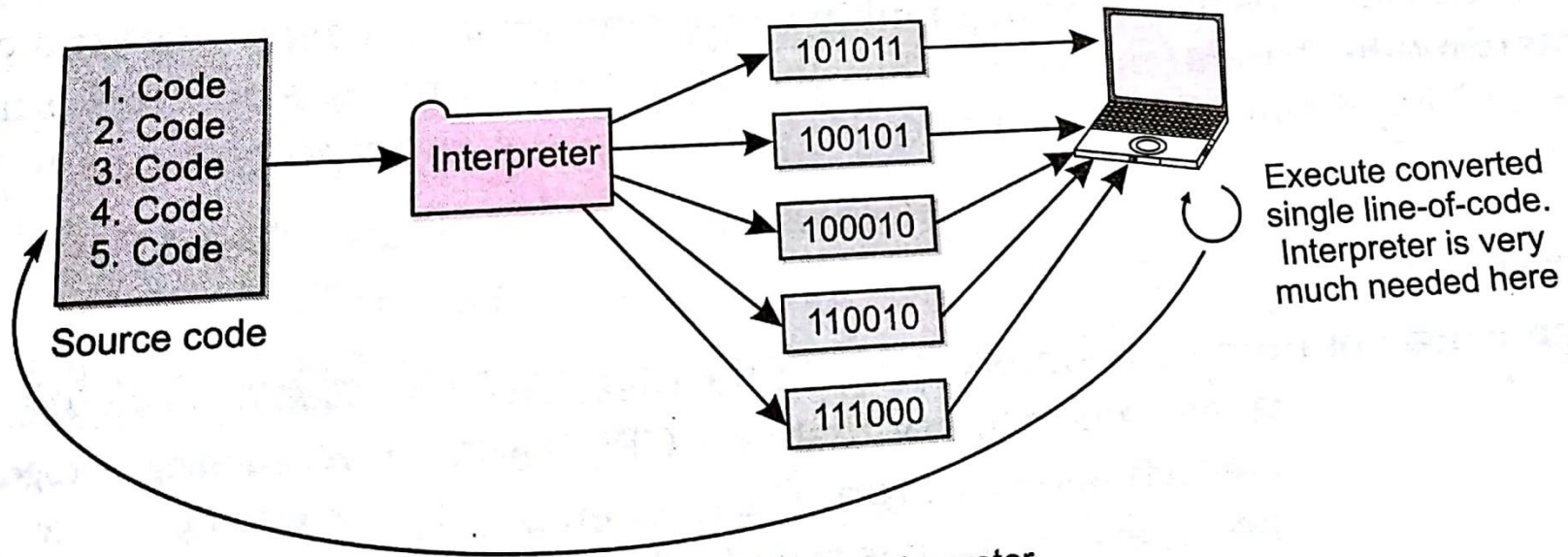
# The Interpretation Process

- Working of an interpreter is like Compiler. It also converts source code to machine understandable code but it is different from compiler in some definite manner.
- It converts a program line by line by adding required libraries.
- Meaning, before the conversion of second line, first line gets completely executed.
- Interpreter follows same manner of conversion and execution for the whole program.

# Compiler Vs Interpreter



(a) Working of a compiler



(b) Working of an interpreter

# Compiler Vs Interpreter

Aspects	Compiler	Interpreter
Input	It takes the whole program for input.	It takes one line as input.
Output	It develops a middle object code of the program.	It does not create middle object code.
Memory	This process requires much memory.	This process does not require much memory.
Errors	It shows all errors together with line number.	It continues by showing error from one line at a time.
Always Required	After successful compilation of a program, compiler is not further needed.	For every run of the program interpreter is needed even after successful interpretation.
Workload	It decreases workload because compiler is not needed again and again for a program.	It's continues requirement increases workload.

# Role of Operating System in program running

- As we know that execution of an .exe file is free from any requirement for a computer system.
- This executable file executes with the help of operating system.
- Now a days, importance and complexity of operating system has been increased much because of multi-user and multi-processing concepts.
- Operating system works as an interface between a user and hardware.
- Also, operating system provides platform to a program to run.

# OS as a Resource Manager

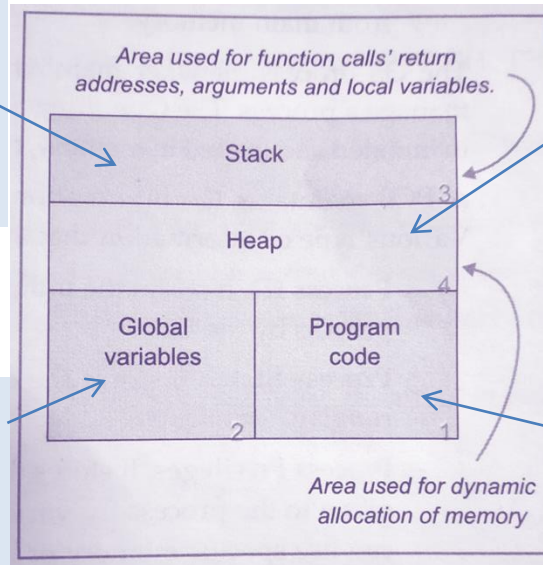
- A program, in its static state is known as a program and in running state it is known as process.
- When many program executes at the same time, OS improves or enhance the capacity or capability of system by managing resources. Which is to be measured in the terms of throughput.
- $$\text{Throughput} = \frac{\text{Number of jobs done}}{\text{Time taken}}$$
- For Multiple Processing, OS provides following function:-
  - Process Management
  - Process Scheduling
  - Memory Management
  - I/O Management

# Process Management

- A program for execution reaches to operating system.
- Then, *process scheduler of OS loads this program into memory. Now, the program becomes process. Which executes as per instructions.*
- For a program, memory is distributed into following blocks-

This block contains all local variables, parameters, return addresses of function calls along with CPU state. This block is also known as Stack.

This block contains all Global variables. These variables remain in the memory till the end of program.

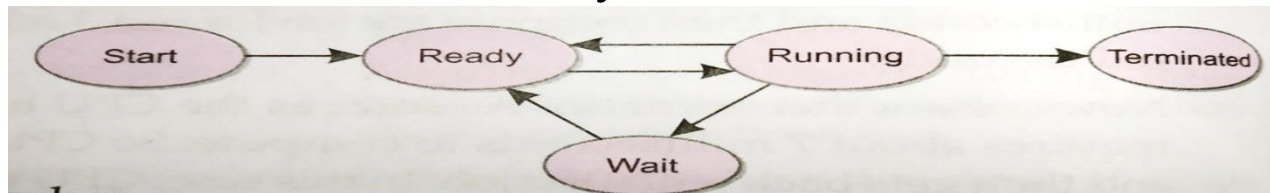


This block provides memory for dynamic memory allocation

This block contains compiled code of program.

# Process Management

- When a program gets ready for execution then it is transferred into memory and becomes process. Now, this process follows following states.-
  - **Start State:** This state comes when a program first time transferred to memory.
  - **Ready State:** This state comes when process gets ready for execution and waits for CPU as per its turn .
  - **Running State:** This state comes when CPU takes the process for execution.
  - **Waiting State:** This state comes when process waits for some resource for execution for ex- input from user.
  - **Terminated OR Exit:** Whn CPU follws all instructions of a process, its exceution gets completed and now it is terminated by OS. This terminated process now gets removed from memory.



# Process Management

- OS process manager develops a data structure to manage a process which is known as Process Control Block (PCB).
- OS develops PCB for every process.
- When a process loads into memory, it gets an ID which is known as Process ID or PID and then its PCB is created.
- PCB keeps important information about process to keep track of process. PCB keeps following information-

Process ID	Stores Process ID.
State	State -> ready, running, waiting ....etc
Pointer	Stores address of sub-processes
Priority	Data of higher or less priority processes
Program Counter	It stores the address of next instruction
CPU Registers	It stores information for execution
Etc . . .	Etc . . .

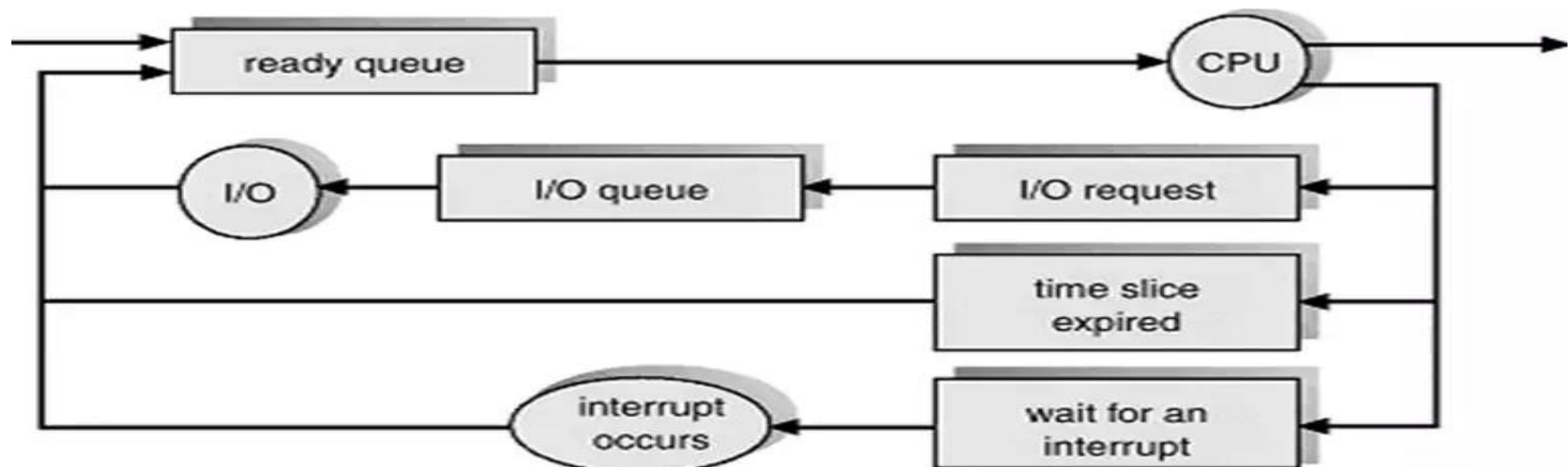


# Process Scheduling by OS

- As we know that many programs loads together in the computer memory for execution. It is very difficult to manage these processes. Following points are important to know for this-
  - Which process will get CPU time?
  - Which process will be deleted from CPU?
  - Which process is to be shifted for I/O?
  - How to handle an interruption?
- Here interruption is signal for CPU from a high priority process to get CPU time.
- This completes task is of OS process scheduler.
- Process scheduling is an activity of process manager in which process manager removes the running processes from CPU and selects the next process.

# Process Scheduling by OS. . . .

- There are some strategies for selection-
  - First come first serve (FCFS)
  - Round robin scheduling (Round Robin Scheduling)
  - Smallest job first (Shortest Job Next ) etc
- There are other strategies too. Process scheduler also prepares some Process Scheduling Queues.
- **Job Queue** : All lines of process in a system.
- **Ready queue**: Line of process with read state.
- **Device Queues** : like I/O line



# Memory Management

Because anything to be processed, needs to load in the memory therefore Memory Management part of OS is also very important.

Memory manager perform following tasks-

- Stores address of every memory location.
- It keeps track of allocated memory to a process and required memory also.
- It decides that when the memory is to be released to a process.
- Responsibility for the safety of data in allocated memory.

Some memory management techniques are-

- Single Memory Allocation
- Partitioned memory allocation
- Paged Memory Management
- Segmented Memory Management

# I/O Management

- When a process requiring an input or output, OS takes its responsibility.
- I/O Manager sends a request to CPU which creates an interrupt for CPU.
- CPU sends running process for I/O meanwhile CPU completes another process. As soon as I/O of first process gets over, CPU again takes it for completion.

# PARALLEL COMPUTING

- At the time of invention of computers, there was only one processor in a system which could perform only one job at a time. It performed one task after previous task. With the improvement in technique and development, processors became more able and multi-processor computers were also developed.
- Multi-processor system can perform more than one task at a time, which is known as *PARALLEL COMPUTING*.
- In a parallel computing system -
  - A problem is distributed into several individual units.
  - Every unit executes its own instructions.
  - Instructions of each unit is completed by separate processor.
- For this, either multi-processor system is to be used or multiple systems are to be used in network.
- This saves time, makes a complex problem easy, makes full use of hardware and use peripherals optimally.

# CLOUD COMPUTING

- Cloud Computing is actually an internet based process which uses computer application.
- [Google apps is an example of cloud computing which provides business applications online.](#)

It is of two types-

- Public Cloud like Google Drive, iCloud etc.
- Private Cloud like KVS Shaala Darpan.

# Thank you

Please follow us on our blog-

[www.pythontrends.wordpress.com](http://www.pythontrends.wordpress.com)