

# Chapter 2

# Computer Organisation

# Lesson 6

## Program development process and tools

**Objective —**

- **Understand process of program development and what are the compilers, assemblers, linkers, debuggers and interpreter**

# Outline

- Process of Program development and execution
- Program development tools

# Process of developing in a high-level language.

1. Write program in high-level language



2. Compile program into assembly

# Process of developing in a high-level language.

3. Assemble program into machine language



4. Link multiple machine-language programs into one application

# Executing a loaded program in Memory

5. Load program into computer's memory



6. Execute program

# Outline

- Process of Program development and execution
- Program development tools

# Compilers

These include *compilers* that convert programs from high-level languages such as C or C++ into assembly language

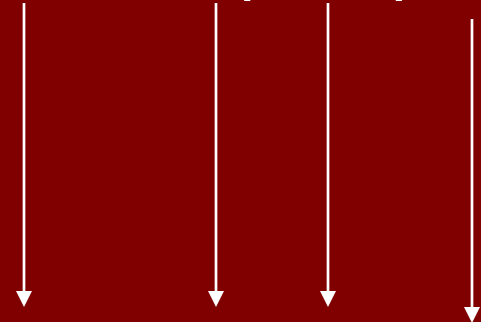


# Assemblers

*Assemblers* converts assembly-language instructions into the numeric representation used by the processor

# Assembly Language

Assembly Language: ADD r1, r2, r3



Machine Language: 0x04 01 02 03

■

# Linkers

*Linkers* that join multiple machine language programs into a single executable file.

# Debuggers

Programs that display the state of another program as it executes to allow programmers to track the progress of a program and find errors.

# Interpreter

- An alternative to compiling a program is to use an interpreter to execute the high-level language version of the program.

# Interpreter

- Interpreters are programs that take high-level language programs as inputs and perform the steps defined by each instruction in the high-level language program

# Interpreter

- Interpreter generates the same result as compiling the program and then executing the compiled version.
- Interpreted programs tend to be much slower than compiled programs,

# Interpreter

- The interpreter has to examine (at run time) each instruction in the source program as it occurs and then jump to a routine that performs the instruction.



# Summary

# We learnt

- **Programs are written in high level language**
- **Programs are compiled, assembled, linked**
- **Program in numeric form loaded into memory in order to execute**

# We also learnt

- **Use of Interpreter**

End of Lesson 6 on  
**Program development process**  
**and tools**

The McGraw-Hill Companies

SCHAUM'S  
ouTlines

# COMPUTER ARCHITECTURE

Nicholas Carter

- ▲ Overview of hardware and software design elements in computer systems
- ▲ Enhanced coverage of topics like instruction sets, processor design, and memory systems.
- ▲ Concise explanations of hardware and software interfaces.
- ▲ Over 600 solved problems and objective questions.



For sale in  
India, Pakistan,  
Nepal, Bangladesh,  
Sri Lanka and  
Bhutan Only

Adapted by: RAJ KAMAL

THANK YOU