UNIT 4

Chapter 2

SOFTWARE DEVELOPMENT SYSTEM AND ASSEMBLERS

2.1 INTRODUCTION

- It is a simply a computer that enables the user write, modify, debug and test programs
- In microprocessor-based development system, a microcomputer is used to develop software for particular microprocessor
- Generally, the microcomputer has a large R/W memory, disc storage and video terminal with an ASCII keyboard
- The system is controlled by a program called operating system
- The hardware and software features of software development system are described in the next sections

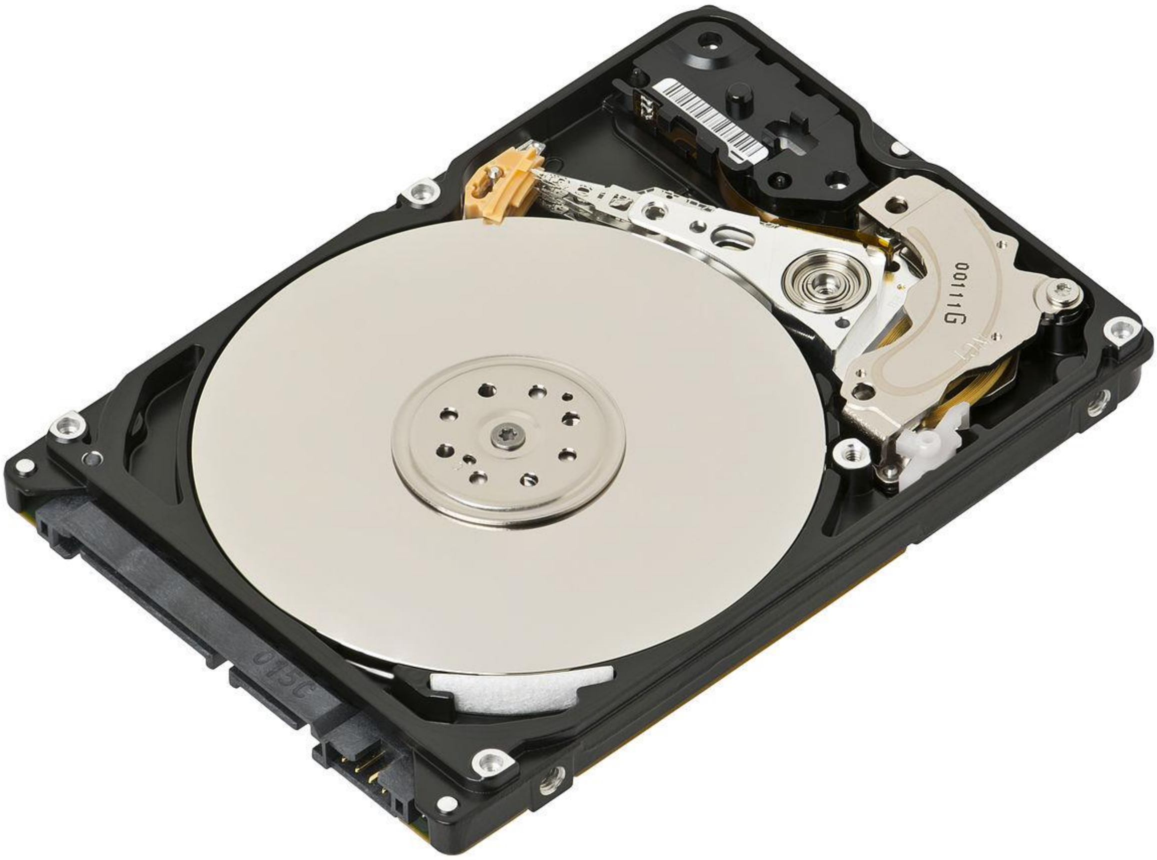
2.2 SYSTEM HARDWARE AND STORAGE MEMORY

- Software development system includes an ASCII keyboard, a CRT terminal, an MPU board with 8M to 64M R/W memory, disc controller and disc drives.
- The disc controller is an interfacing circuit through which the MPU can access a disc and provide R/W control signals
- The disc drives have R/W elements, which are responsible for reading and writing data on disc
- At present, most systems are equipped with a 3.5 inch disc, stores 1.44MB of information
- The storage capacity of a typical hard disc in a PC is 2.2GB or higher

Floppy disk

- A floppy disk, also called a floppy, diskette or just disk, is a type of disk storage composed of a disk of thin and flexible magnetic storage medium, sealed in a rectangular plastic enclosure lined with fabric that removes dust particles.
- Floppy disks are read and written by a floppy disk drive (FDD).
- Floppy disks, initially as 8-inch (200 mm) media and later in 5¼-inch (133 mm) and 3½-inch (90 mm) sizes, were a ubiquitous form of data storage and exchange from the mid-1970s into the mid-2000s.
- By the late 2000s, computers were rarely manufactured with installed floppy disk drives; 3½-inch floppy disks can be used with an external USB floppy disk drive, but USB drives for 5¼-inch, 8-inch, and non-standard diskettes are rare to non-existent.

Hard disk:



Source: https://upload.wikimedia.org/wikipedia/commons/thumb/f/f8/Laptop-hard-drive-exposed.jpg/1200px-Laptop-hard-drive-exposed.jpg

- A hard disk drive (HDD), hard disk, hard drive or fixed disk is a data storage device that uses magnetic storage to store and retrieve digital information using one or more rigid rapidly rotating disks (platters) coated with magnetic material.
- The platters are paired with magnetic heads, usually arranged on a moving actuator arm, which read and write data to the platter surfaces.

- Data is accessed in a random-access manner, meaning that individual blocks of data can be stored or retrieved in any order and not only sequentially.
- HDDs are a type of non-volatile storage, retaining stored data even when powered off.

CD-ROM:

- A CD-ROM a pre-pressed optical compact disc which contains data.
- The name is an acronym which stands for "Compact Disc Read-Only Memory".
- Computers can read CD-ROMs, but cannot write to CD-ROMs which are not writable or erasable.
- Data is stored on the disc as a series of microscopic indentations.
- A laser is shone onto the reflective surface of the disc to read the pattern of pits and lands ("pits", with the gaps between them referred to as "lands").
- Because the depth of the pits is approximately one-quarter to one-sixth of the wavelength of the laser light used to read the disc, the reflected beam's phase is shifted in relation to the incoming beam, causing destructive interference and reducing the reflected beam's intensity.
- This pattern of changing intensity of the reflected beam is converted into binary data.

2.3 OPERATING SYSTEMS:

- An operating system (OS) is system software that manages all the operations of the computer.

- The kernel is a major component of an operating system. The kernel provides the basic communication between hardware and application programs.It manages memory access for programs, files on storage devices and the communication between the computer and its peripherals.
- All computer programs, excluding firmware, require an operating system to function.
- For hardware functions such as input and output and memory allocation, the operating system acts as an intermediary between programs and the computer hardware, although the application code is usually executed directly by the hardware and frequently makes system calls to an OS function or is interrupted by it.

MS-DOS Operating System:

- Used in 16-bit microcomputers, such as IBM PC, XT and AT
- Designed to handle16-bit data word and 640 Kbyte ystem memory
- Can support hard disks and includes a hierarchical file directory.

OS/2 (Operating System2):

- 32-bit single user operating system designed by IBM
- Exploit various features of 32-bit and 64-bit microprocessors.
- Can assign 512Mbyte of virtual memory space to each application program.
- Compatible with DOS and Windows application programs
- Support various telecommunication features such as e-mail, fax and telephone voice mail.
- Support CD-ROM.

WINDOWS 95 TO WINDOWS 98:

- Windows 95 is a 32-bit single user multitasking operating system with graphic interface.
- Supports the use of mouse, icons and menus.
- Windows 98 is upgraded version of Windows 95
- It was followed by Windows NT.

Unix Based Operating Systems:

- Unix is multiuser, multitasking operating system designed for time-shaping environment.
- Initially it was proprietary and designed for main-frame computers, but now also used on powerful microcomputers and supercomputers.
- Open source software. It is free and its code is freely distributed.
- Used in high end microcomputers such as workstations and network servers.

2.4 PROGRAMMING TOOLS

- A programming tool or software development tool is a computer program that software developers use to create, debug, maintain, or otherwise support other programs and applications.
- The term usually refers to relatively simple programs, that can be combined together to accomplish a task, much as one might use multiple hand tools to fix a physical object.
- The most basic tools are a source code editor and a compiler or interpreter, which are used ubiquitously and continuously.
- Other tools are used more or less depending on the language, development methodology, and individual engineer, and are often used for a discrete task, like a debugger or profiler.

2.4.1 Assembler and Cross-Assembler:

- A computer will not understand any program written in a language, other than its machine language which is binary.
- **Assembler is a program that translates source code or mnemonics into binary code, called object code.**
- In addition to translating mnemonics, the assembler also performs various functions like error checking, memory allocations..
- The assembler also generates a list file (LST) and a hex file (HEX).
- The list file is used for documentation while hex file is used to download program from PC to target system.
- If an assembler which runs on a computer and produces the machine codes for the same computer then it is called **self-assembler or resident assembler**.
- If an assembler that runs on a computer and produces the machine codes for other computer then it is called **Cross Assembler**.
- A cross assembler is an assembler that runs on a computer with one type of processor but generates machine code for a different type of processor.

Assemblers are further divided into two types: One Pass Assembler and Two Pass Assembler.

- **One pass assembler** is the assembler which assigns the memory addresses to the variables and translates the source code into machine code in the first pass simultaneously.
- **Two Pass Assembler** is the assembler which reads the source code twice. In the first pass, it reads all the variables and assigns them memory addresses. In the second pass, it reads the source code and translates the code into object code.

2.4.2 Linker:

- **The Linker is a program that takes the object file generated by the assembler and generates another file called the EXE file or COM file.**
- The EXE or COM file is the only executable file and executed by microprocessor.
- In different assemblers the EXE or COM file may be labelled by other names. stored.

2.4.3 Simulation:

- **A simulation enables the user to execute a program on a PC without having a target system..**
- The simulator emulates the execution of the instructions with the operational speed of the PC.
- It is not a real time execution as in hardware.

2.4.4 Debugger:

- **The Debugger is a program that allows the user to test and debug the object file.**
- The user can employ this program to perform the following functions –
    - Make changes in object code
    - Examine and modify the contents of memory.
- Set breakpoints, execute a segment of the program and display the register and memory contents after execution of each instruction.

2.5 WRITING ASSEMBLY LANGUAGE PROGRAM:

After the end of the assembly process, you will have the following files on your PC disk.

- ASM File: saved with extension '.ASM'. This extension suggests that this is an assembly language file.
- OBJ file: saved with extension '.OBJ'. This is intermediate file generated by assembler.
- LST file: saved with extension '.LST'. This is list file generated by assembler for documentation purpose. It contains memory locations, Hex codes, mnemonics and comments.
- HEX file: saved with extension '.HEX'. This is executable file generated by linker. It contains binary code that can be understood and executed by 8085 processor.

2.6 ADVANTAGES OF ASSEMBLERS:

- Assembler translates mnemonics into binary code with speed and accuracy, thus eliminating human errors in entering codes.
- The assembler assigns appropriate values to the symbols in a program. This facilitates specifying jump locations.
- It is easy to insert or delete instructions in a program. This avoids rewriting the whole program manually
- Assembler checks syntax errors and provides error messages.
- The assembler can reserve memory locations for data or results.
- The assembler can provide files for documentation.

2.7 How to Use the Cross-Assembler Package

- A cross-assembler is just like any other assembler except that it runs on some CPU other than the one for which it assembles code.
- For example, this package assembles 8085 source code into 8085 object code, but it runs on an 8088, a 68000, or whatever other CPU you happen to have a C compiler for.
- The reason that cross-assemblers are useful is that you probably already have a CPU with memory, disk drives, a text editor, an operating system, and all sorts of hard-to-build or expensive facilities on hand.
- A cross-assembler allows you to use these facilities to develop code for an 8085.
- This program requires one input file (your 8085 source code) and zero to two output files (the listing and the object).
- The input file MUST be specified, or the assembler will bomb on a fatal error. The listing and object files are optional.
- If no listing file is specified, no listing is generated, and if no object file is specified, no object is generated. If the object file is specified, the object is written to this file in "Intel hexadecimal" format.

EXTRA

SOFTWARE TOOLS FOR HIGH LEVEL LANGUAGE:

Compiler:

- **Software which translates a high level language program into a machine language program is called compiler.**
- A compiler is more intelligent than an assembler. It checks all kinds of limits, ranges, errors etc.
- But its program run time is more and occupies a larger part of the memory.
- It has slow speed. Because a compiler goes through the entire program and then translates the entire program into machine codes. (A compiler is 5 to 25 times faster than an interpreter).
- If a compiler runs on a computer and produces the machine codes for the same computer then it is known as a **self-compiler or resident compiler**.
- On the other hand, if a compiler runs on a computer and produces the machine codes for other computer then it is known as a **cross compiler**.
- By the compiler, the machine codes are saved permanently for future reference

Interpreter:

- **Software which translates statements of a program into machine code is called interpreter.**
- It translates only one statement of the program at a time. It reads only one statement of program, translates it and executes it.
- Then it reads the next statement of the program again translates it and executes it.
- In this way it proceeds further till all the statements are translated and executed.
- Unlike compiler the machine codes produced by interpreter are not saved.
- An interpreter is a small program as compared to compiler. It occupies less memory space, so it can be used in a smaller system which has limited memory space.

Loader:

- **Loader is a program that loads machine codes of a program into the system memory**.
- In Computing, a loader is the part of an Operating System that is responsible for loading programs. It is one of the essential stages in the process of starting a program. Because it places programs into memory and prepares them for execution.
- Loading a program involves reading the contents of executable file into memory.
- Once loading is complete, the operating system starts the program by passing control to the loaded program code.

- All operating systems that support program loading have loaders.
- In many operating systems the loader is permanently resident in memory.