

OVERVIEW OF COMPUTER SYSTEMS

A.1 Introduction

This supplement introduces computer basics. Specifically, it discusses computer hardware components, programming languages, and operating systems.

A.2 What is a Computer?

A computer is an electronic device that stores and processes data. A computer includes both *hardware* and *software*. In general, hardware is the physical aspect of the computer that can be seen, and software is the invisible instructions that control the hardware and make it work. Computer programming consists of writing instructions for computers to perform. You can learn a programming language without knowing computer hardware, but you will be better able to understand the effect of the instructions in the program if you do. This section gives a brief introduction to computer hardware components and their functionality.

A computer consists of the following major hardware components:

- Central Processing Unit (CPU)
- Memory
- Secondary Storage
- Input and Output Devices

Figure A.1 shows the relationship among these components.

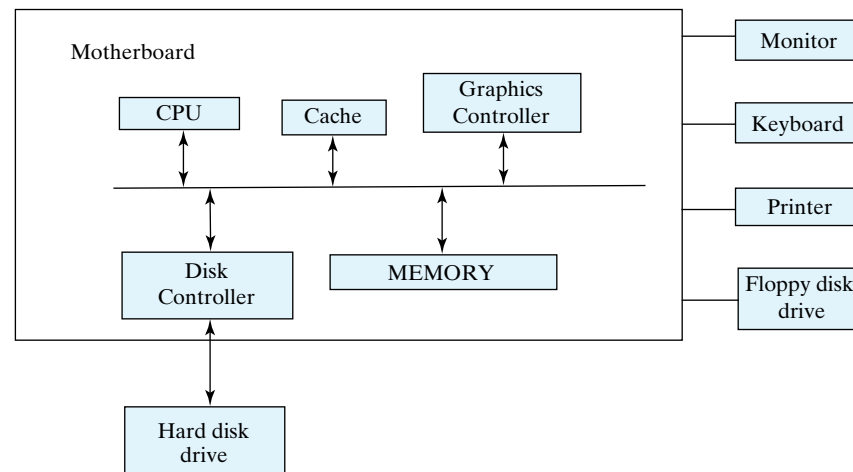


Figure A.1 A computer system consists of a CPU, cache, memory, hard disk, floppy disk, monitor, and printer.

A.2.1 Central Processing Unit

The *central processing unit* (CPU) is the brain of a computer. It retrieves instructions from memory and executes them. The CPU usually has two components: a *control unit* and an *arithmetic/logic unit*. The control unit controls and coordinates the actions of the other components. The arithmetic and logic unit performs numeric operations (addition, subtraction, multiplication, division) and logical operations (comparisons).

The *speed* of the CPU is mainly determined by clock speed. Every computer has an internal clock. The clock emits electronic pulses at a constant rate, and these are used to control and synchronize the pace of operations. The faster the clock speed, the more instructions are executed in a given period of time. The clock speed is measured in megahertz (MHz), with 1 megahertz equaling 1 million pulses per second.

Today's CPU is built on a small silicon semiconductor chip with millions of transistors.

A.2.2 Memory

Memory is used to store information and programs. A memory unit is an ordered sequence of storage cells, each capable of holding a piece of data. Every cell has a unique address. The address is used to locate the cell for storing and retrieving data.

Figure A.2 shows data and instructions stored in memory.

0000	'J'	'a'	'v'	'a'
0004				
0008				
000C				
0010				
0014				
0018				
001C				

Figure A.2 Memory has a physical address for locating data and instructions in the memory.

Since we can access data at any location, the memory is also referred to as *RAM* (random-access memory). Today's personal computers usually have at least 64 megabyte of RAM. A *byte* is 8 bits. A *bit* is a binary digit (0 or 1). A *megabyte* is about 1 million bytes. The data stored in computers are represented by patterns of bits.

A program and its data must be brought to memory before they can be executed. A memory cell is never empty, but its initial content may be meaningless to your program. The current content of a memory cell is lost whenever new information is placed in the cell.

Today's computer memory, like the CPU, is built on silicon semiconductor chips containing thousands of transistors embedded on their surface. Compared to the CPU chips, memory chips are less complicated, slower, and less expensive.

In addition to regular memory chips, most current desktop computers have a special type of RAM called *cache memory*. Cache memory is faster than the regular memory. In consequence, the system can store the most frequently used instructions and data in the cache to improve system performance. Cache memory is more expensive, so its storage size is usually small. Most new PCs have between 128K and 512K cache memory.

A.2.3 Secondary Storage

Secondary storage is the permanent storage for data and programs. Memory is volatile, because information is lost when the power is off. Secondary storage is non-volatile. Your program and data are stored on secondary storage and moved to memory only when the computer actually uses them.

There are three main types of secondary storage devices:

- Disks drives
- CD-ROM drives
- Tape drives

Drives are devices for operating a medium, such as disks, CD-ROMs, and tapes.

A.2.3.1 Disks

There are two kinds of disks: *hard disks* and *floppy disks*. Personal computers usually have a 3.5-inch floppy disk drive and a hard drive. A floppy disk has a fixed capacity of about 1.44 MB. Hard disk capacities vary. The capacity of the hard disks of the latest PCs is in the range of 10 to 30 gigabytes. Access speeds range from 175 to 300 milliseconds for floppy disks and 8 to 20 milliseconds for hard disks. Both disk drives are often encased inside the computer. A floppy disk is easily movable. A hard disk is mounted inside the case of the computer. Removable hard disks are also available.

A.2.3.2 CD-ROMs

There are two types of CD-ROM drives: CD-R and CD-RW. A *CD-R* is read-only permanent storage, and the user cannot modify its contents once they are recorded. A CD-RW can be used like a floppy disk. A single CD can hold up to 650 MB. Most software is distributed through CD-Rs. New PCs all come with a CD-RW drive that can work with both CD-R and CD-W.

A.2.3.3 Tapes

Tapes are now mainly used for backup of data and programs. Unlike disks and CD-ROMs, tapes store information sequentially. The computer must retrieve information in the order it was stored. Tapes are very slow. It would take one to two hours to back up a 1-gigabyte hard disk.

A.2.4 Input and Output Devices

The common input devices are *keyboards* and *mouses*. The output devices are *monitors* and *printers*. Input devices let the user talk to the computer. Output devices let the computer communicate to the user.

A.2.4.1 The Keyboard

A computer *keyboard* resembles a typewriter keyboard except that it has extra keys for certain special functions.

- *Function keys* are located at the top of the keyboard with prefix F. Their use depends on the software.
- *Numeric keypad*, located on the right corner of the keyboard, is a separate set of number keys for quick input of numbers.
- *Arrow keys*, located between the main keypad and the numeric keypad, are used to move the cursor up, down, left, and right.
- *Insert, delete, page up, page down keys*, located above the arrow keys, are used in word processing for performing insert, delete, page up, and page down.

A.2.4.2 The Mouse

A mouse is a pointing device. It is used to move an electronic pointer called a cursor around the screen or to click on an object on the screen to trigger it to respond. In Java GUI programming, you can use the mouse to click on a button to trigger an event.

A.2.4.3 The Monitor

The monitor displays information (text and graphics). The resolution and dot pitch determine the quality of the display.

The *resolution* specifies the number of pixels per square inch. Pixels (short for “picture elements”) are tiny dots that form an image on the screen. The resolution can be set manually. The higher the resolution, the sharper and clearer the image is. However, the image may be very small if you set high resolution on a small screen monitor. PC monitors are usually 15-inch, 17-inch, 19-inch, or 21-inch. For a 15-inch monitor, a comfortable resolution setting would be 640×480 (307,200 pixels).

The *dot pitch* is the amount of space between pixels. The smaller the dot pitch, the better the display.

A.3 Computer Programming

Computer programs are instructions to the computer. You tell a computer what to do through programs. Without programs, a computer is an empty machine. Computers do not understand human languages, so you need to use computer languages to communicate with them.

The language a computer speaks is the computer’s native language or machine language. The *machine language* is a set of primitive instructions built into every computer. The instructions are in the form of binary code, so you have to enter binary codes for various instructions. Programming with native machine language is a tedious process. Moreover the programs are highly difficult to read and modify.

Assembly language is a low-level programming language in which a mnemonic is used to represent each of the machine language instructions. Table 1 shows three assembly language instructions and their corresponding machine codes.

TABLE A.1 Illustrating assembly code and machine code

<i>Assembly Language</i>	<i>Machine Code</i>
CVTLF N, R5	55DBAF4E
MULF2 V, R5	55D3AF44
ADDF3 B, R5, C	C4AF55CBAF41

SUPPLEMENTS

Assembly languages were developed to make programming easy. Since the computer cannot understand assembly language, however, a program called *assembler* is used to convert assembly language programs into machine code, as shown in Figure A.3.

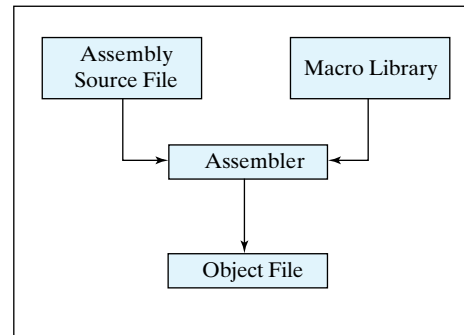


Figure A.3 *Assembler translates assembly language instructions to machine code.*

Since assembly language is machine-dependent, an assembly program can only be executed on a particular machine. Assembly programs are written in terms of machine instructions with easy-to-remember mnemonic names. The high-level languages were developed in order to overcome the platform-specific problem and make programming easier.

The *high-level languages* are English-like and easy to learn and program. There are over one hundred high-level languages. The popular languages used today are:

- COBOL (COMmon Business Oriented Language)
- FORTRAN (FORmula TRANslation)
- BASIC (Beginner All-purpose Symbolic Instructional Code)
- Pascal (named for Blaise Pascal)
- Ada (named for Ada Lovelace)
- C (whose developer designed B first)
- Visual Basic (Basic-like visual language developed by Microsoft)
- Delphi (Pascal-like visual language developed by Borland)
- C++ (an object-oriented language, based on C)

Each of these languages was designed for a specific purpose. COBOL was designed for business applications and now is used primarily for business data processing. FORTRAN was designed for mathematical computations and is used mainly for numeric computations. BASIC, as its name suggests, was designed to be learned and used easily. Ada was developed for the Department of Defense and is mainly used in defense projects. C combines the power of an assembly language with the ease of use and portability of a high-level language. Visual Basic and Delphi are used

in developing graphical user interfaces and in rapid application development. C++ is popular for system software projects like writing compilers and operating systems. Microsoft Windows95 was coded using C++.

A program written in a high-level language is called a *source program*. Since a computer cannot understand a source program, a program called a *compiler* is used to translate the source program into a machine language program called an *object program*. The object program is often then linked with other supporting library code before the object can be executed on the machine.

Figure A.4 shows the process of compiling, linking, and running a program.

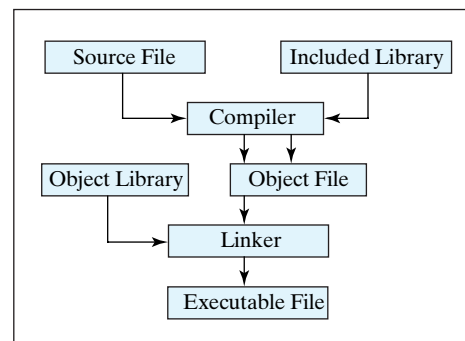


Figure A.4 Compiling, linking, and running a program.

You can run a source program on any machine with appropriate compilers. The source program must be recompiled, however, because the object program can only run on a specific machine. Nowadays computers are networked to work together. Java was designed to run object programs on any platform. With Java, you write the program once and compile the source program into a special type of object code. The object code can then run on any machine that can interpret it.

A.4 Operating Systems

An introduction to computers would be incomplete if it did not mention the operating system (OS). You are probably using Windows 98, NT, 2000, XP, or ME. Windows is currently the most popular PC operating system. The *operating system* is a piece of software that manages and controls a computer's activities. On today's computers, application programs cannot run without an operating system. The inter relationship of hardware, operating system, application software, and user is shown in Figure A.5.

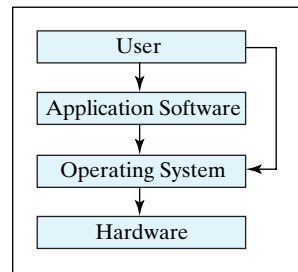


Figure A.5 *The operating system is the software that controls and manages the system.*

The major tasks of the operating systems are:

- Allocating and assigning system resources.
- Scheduling operations.
- Monitoring system activities.

A.4.1 Allocating and Assigning System Resources

The OS is responsible for determining what computer resources (CPU, memory, disks, input and output devices) a program needs and for allocating and assigning them to run the program.

A.4.2 Scheduling Operations

The OS is responsible for scheduling programs to use the system resources efficiently. Many of today's operating systems support such techniques as *multiprogramming*, *multithreading*, or *multiprocessing* to increase system performance.

Multiprogramming allows multiple programs to run simultaneously through sharing of the CPU. The CPU is much faster than the other components. As a result, it is idle most of the time; for example, while waiting for data to be transferred from the disk or from other sources. A multiprogramming OS takes advantage of this by allowing multiple programs to use the CPU when it would otherwise be idle. For example, you may use a word processor to edit a file while the Web browser is downloading a file at the same time.

Multithreading allows concurrency within a program, so that its subunits can run at the same time. For example, a word-processing program allows users to edit text and save it to a file at the same time. In this example, editing and saving are two tasks within the same application.

Multiprocessing, or parallel processing, uses two or more processors together to perform a task. It is like a surgical operation where several doctors work together on one patient.