**Computer Software**

**Software Fundamentals:**

Software consists of **computer programs**, which are sequences of instructions for the computer. The process of writing (or *coding*) programs is called *programming*, and individuals who perform this task are called *programmers*.

**Systems software** is a set of instructions that serves primarily as an intermediary between computer hardware and application programs, and may also be directly manipulated by knowledgeable users. Systems software provides important self-regulatory functions for computer systems, such as loading itself when the computer is first turned on, managing hardware resources such as secondary storage for all applications, and providing commonly used sets of instructions for all applications to use.

**Application software** is a set of computer instructions that provide more specific functionality to a user. That functionality may be broad, such as general word processing, or narrow, such as an organization’s payroll program. An application program applies a computer to a certain need. *Application programming* is either the creation or the modification and improvement of application software.



**Fig.1 Systems software serves as intermediary between hardware and functional applications.**

**Systems Software**

Is the class of programs that control and support the computer system and its information-processing activities. Systems software also facilitates the programming, testing, and debugging of computer programs. It is more general than application software and is usually independent of any specific type of application. Systems software programs support application software by directing the basic functions of the computer.

Systems software can be grouped into two major functional categories: system control programs and system support programs.

1. **System control programs**

control the use of the hardware, software, and data resources of a computer system. The main system control program is the operating system. The **operating system** supervises the overall operation of the computer, including monitoring the computer’s status and scheduling operations, which include the input and output processes. In addition, the operating system allocates CPU time and main memory to programs running on the computer, and it also provides an interface between the user and the hardware. This interface hides the complexity of the hardware from the user. That is, you do not have to know how the hardware actually operates, just what the hardware will do and what you need to do to obtain desired results.

**Process management** means managing the program or programs (also called jobs) running on the processor at a given time. In the simplest case (a desktop operating system), the operating system loads a program into main memory and executes it. The program utilizes the computer’s resources until it relinquishes control. Some operating systems offer more sophisticated forms of process management, such as *multitasking*, *multithreading*, and *multiprocessing*. The management of two or more tasks, or programs, running on the computer system at the same time is called **multitasking**, or **multiprogramming**. The first program is executed until an interruption occurs, such as a request for input. While the input request is handled, the execution of a second program begins. Because switching among these programs occurs so rapidly, they appear to be executing at the same time. However, because there is only one processor, only one program is actually in execution mode at any one time.

**Multithreading** is a form of multitasking that focuses on running multiple tasks within a single application simultaneously. For example, a word processor application may edit one document while another document is being spell-checked.

**Time-sharing** is an extension of multiprogramming. In this mode, a number of users operate online with the same CPU, but each uses a different input/output terminal. The programs of these users are placed into partitions in primary storage. Execution of these programs rotates among all users, occurring so rapidly that it appears to each user as though he or she were the only one using the computer.