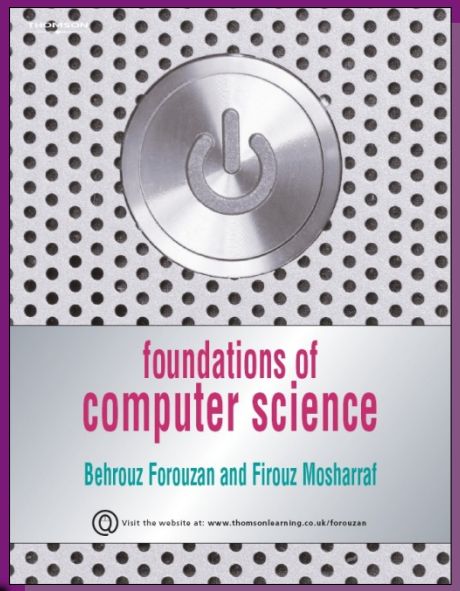
7 Operating Systems



Objectives

After studying this chapter, the student should be able to:

- ❑ Understand the role of the operating system.
- ❑ Understand the process of bootstrapping to load the operating system into memory.
- ❑ List the components of an operating system.
- ❑ Discuss the role of the memory manager.
- ❑ Discuss the role of the process manager.
- ❑ Discuss the role of the device manager.
- ❑ Discuss the role of the file manager in an operating system.
- ❑ Understand the main features of three common operating systems: UNIX, Linux and Windows NT.

A computer is a system composed of two major components: hardware and software. Computer hardware is the physical equipment. Software is the collection of programs that allows the hardware to do its job. Computer software is divided into two broad categories: the **operating system** and **application programs** (Figure 7.1). Application programs use the computer hardware to solve users' problems. The operating system, on the other hand, controls the access to hardware by users.

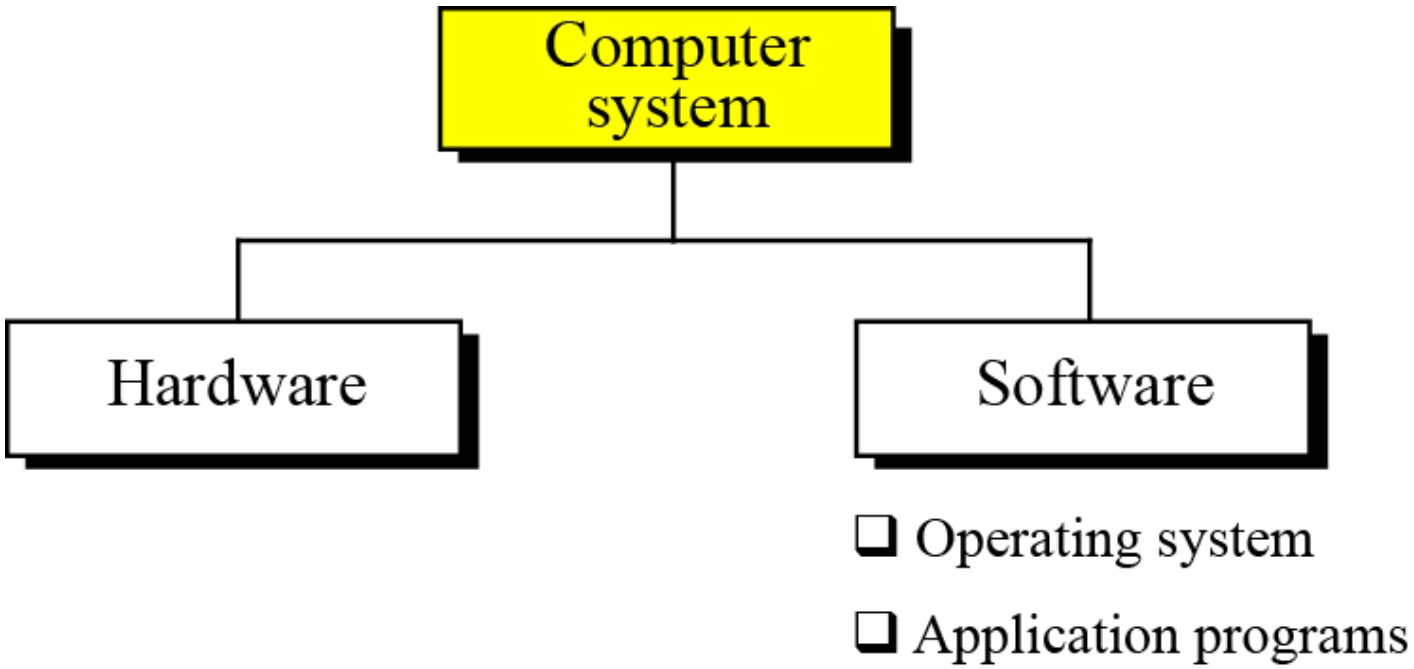


Figure 7.1 **A computer system**

7-1 INTRODUCTION

An **operating system** is complex, so it is difficult to give a simple universal definition. Instead, here are some common definitions:

- **An operating system is an interface between the hardware of a computer and the user (programs or humans).**
- **An operating system is a program (or a set of programs) that facilitates the execution of other programs.**
- **An operating system acts as a general manager supervising the activity of each component in the computer system.**

An operating system is an interface between the hardware of a computer and the user (programs or humans) that facilitates the execution of other programs and the access to hardware and software resources.

Two major design goals of an operating system are:

- **Efficient use of hardware.**
- **Ease of use of resources.**

Bootstrap process

The operating system, based on the above definitions, provides supports for other programs. For example, it is responsible for loading other programs into memory for execution. However, the operating system itself is a program that needs to be loaded into the memory and be run. How is this dilemma solved?

The solution is a two-stage process. A very small section of memory is made of ROM and holds a small program called the **bootstrap program**. When the computer is turned on, the CPU counter is set to the first instruction of this bootstrap program and executes the instructions in this program. When loading is done, the program counter is set to the first instruction of the operating system in RAM.

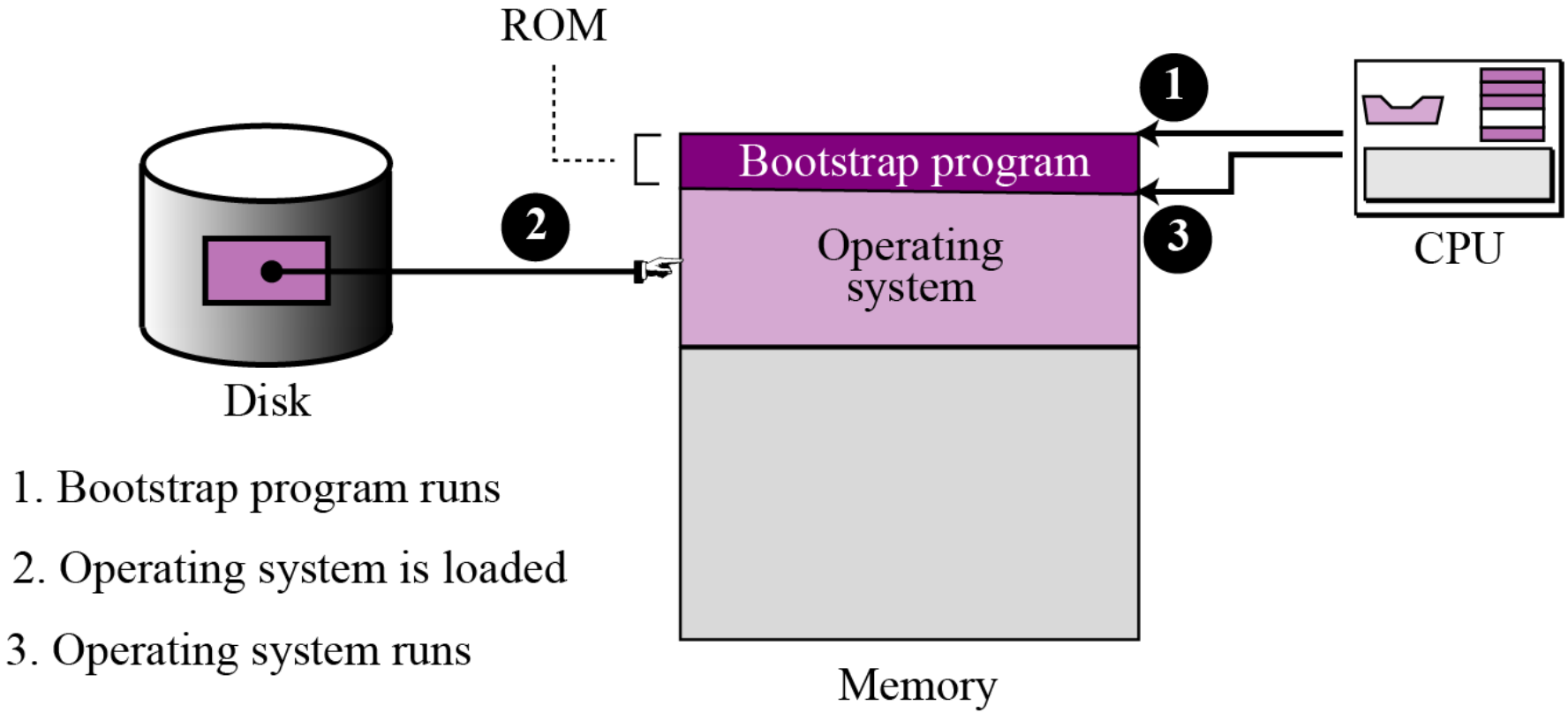


Figure 7.2 **The bootstrap process**

7-2 EVOLUTION

Operating systems have gone through a long history of evolution, which we summarize here.

Batch systems

Batch operating systems were designed in the 1950s to control mainframe computers. At that time, computers were large machines that used punched cards for input, line printers for output and tape drives for secondary storage media. Each program to be executed was called a job. A programmer who wished to execute a job sends a request to the operating system.

Time-sharing systems

To use computer system resources efficiently, *multiprogramming* was introduced. The idea is to hold several jobs in memory at a time, and only assign a resource to a job that needs it on the condition that the resource is available.

Multiprogramming brought the idea of **time sharing**: resources could be shared between different jobs, with each job being allocated a portion of time to use a resource. Because a computer is much faster than a human, time sharing is hidden from the user—each user has the impression that the whole system is serving them exclusively.

Personal systems

When personal computers were introduced, there was a need for an operating system for this new type of computer. During this era, single-user operating systems such as **DOS (Disk Operating System)** were introduced.

Parallel systems

The need for more speed and efficiency led to the design of **parallel systems**: multiple CPUs on the same machine. Each CPU can be used to serve one program or a part of a program, which means that many tasks can be accomplished in parallel instead of serially. The operating systems required for this are more complex than those that support single CPUs.

Distributed systems

Networking and internetworking, as we saw in Chapter 6, have created a new dimension in operating systems. A job that was previously done on one computer can now be shared between computers that may be thousands of miles apart. Distributed systems combine features of the previous generation with new duties such as controlling security.

Real-time systems

A real-time system is expected to do a task within a specific time constraint. They are used with real-time applications, which monitor, respond to or control external processes or environments.

7-4 A SURVEY OF OPERATING SYSTEMS

In this section we introduce some popular operating systems and encourage you to study them further. We have chosen three operating systems that are familiar to most computer users: UNIX, Linux and Windows.

UNIX

UNIX was originally developed in 1969 by Thomson and Ritchie of the Computer Science Research Group at Bell Laboratories. UNIX has gone through many versions since then. It has been a popular operating system among computer programmers and computer scientists.

UNIX is a multiuser, multiprocessing, portable operating system.
It is designed to facilitate programming, text processing and communication.

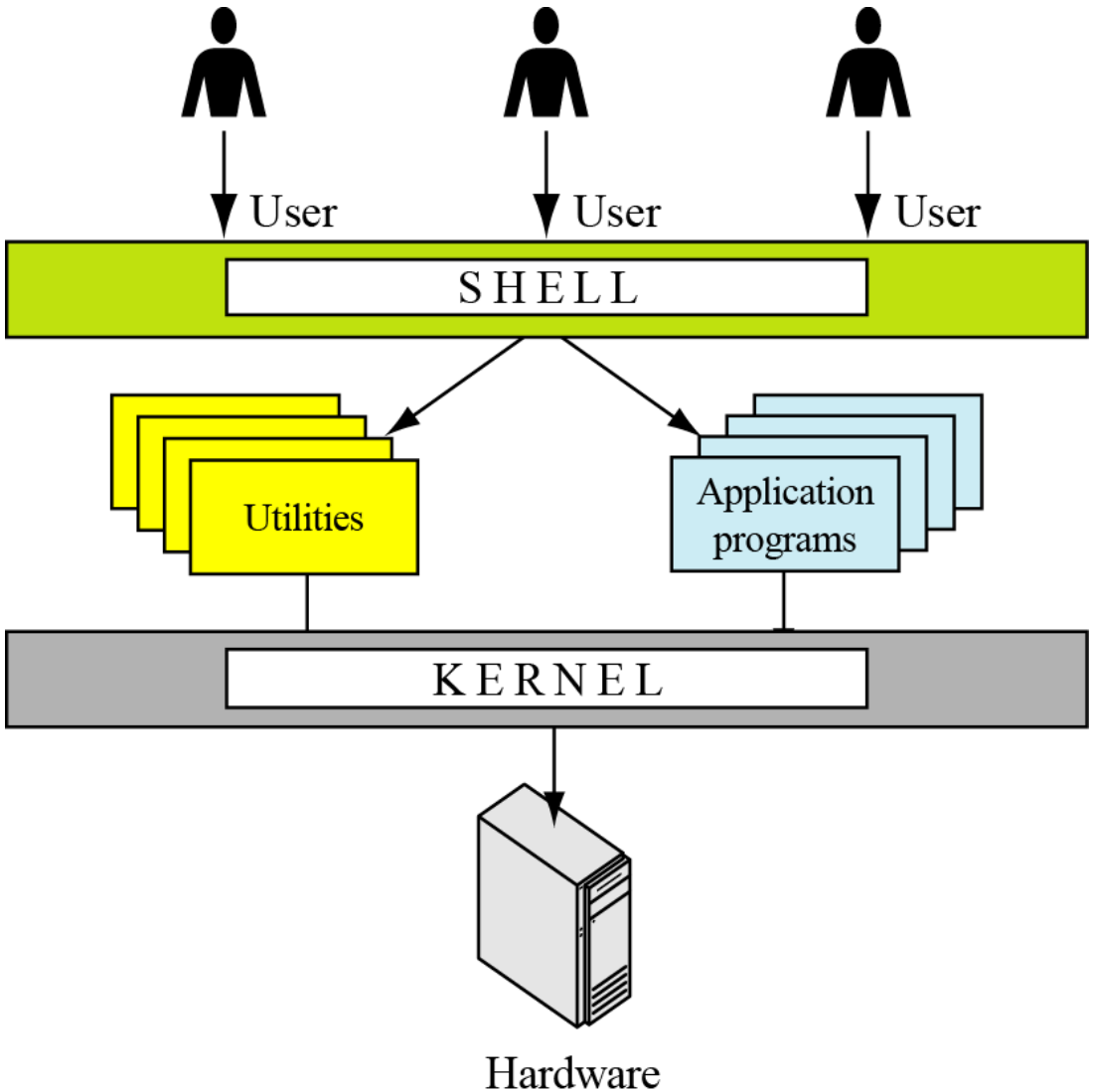


Figure 7.20 **Components of the UNIX operating system**

Linux

In 1991, Linus Torvalds, a Finish student at the University of Helsinki at the time, developed a new operating system that is known today as Linux. The initial kernel, which was similar to a small subset of UNIX, has grown into a full-scale operating system today. The Linux 2.0 kernel, released in 1997, was accepted as a commercial operating system: it has all features traditionally attributed to UNIX.

Windows NT/2000/XP

In the late 1980s Microsoft, under the leadership of Dave Cutler, started development of a new single-user operating system to replace MS-DOS (Microsoft Disk Operating System). Windows NT (NT standing for New Technology) was the result. Several versions of Windows NT followed and the name was changed to Windows 2000. Windows XP (XP stands for eXPerience) was released in 2001. We refer to all of these versions as Windows NT or just NT.

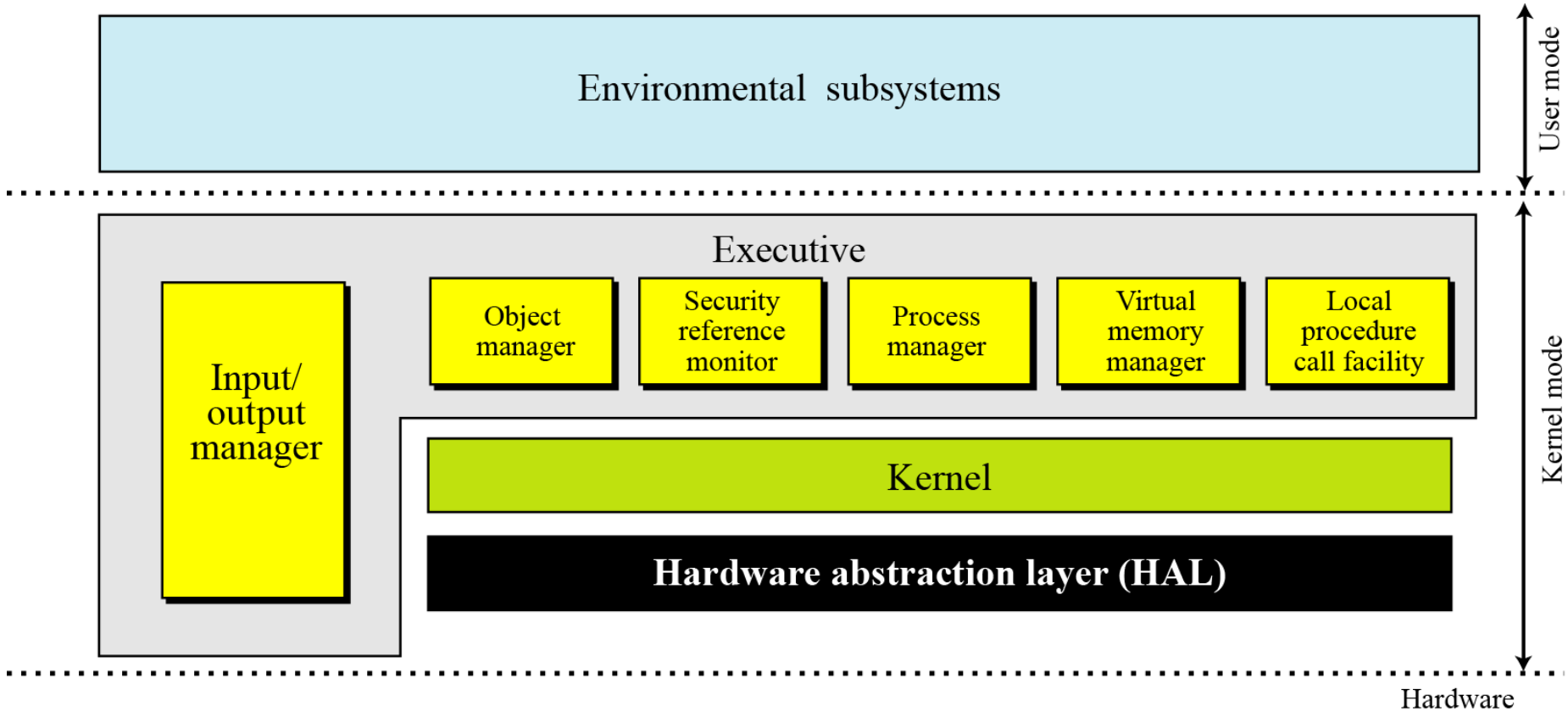


Figure 7.21 **The architecture of Windows NT**