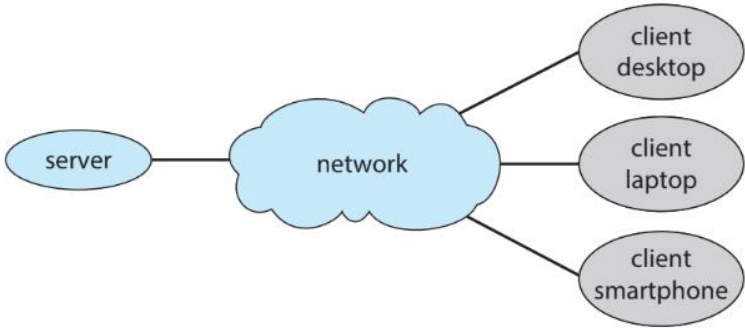
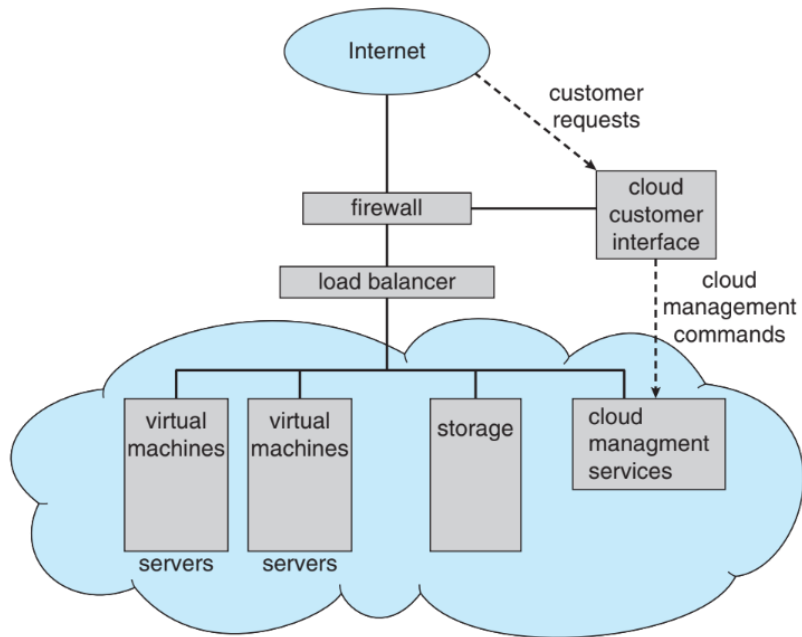


**Figure 1.16** A computer running (a) a single operating system and (b) three virtual machines.

send a request to perform an action (for example, read data). In response, the server executes the action and sends the results to the client. A server

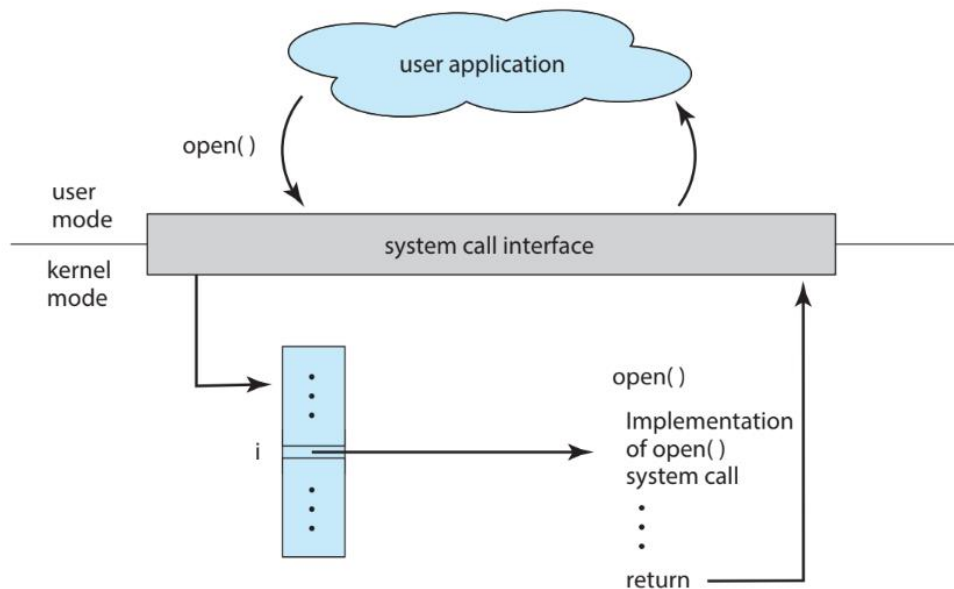


**Figure 1.22** General structure of a client–server system.



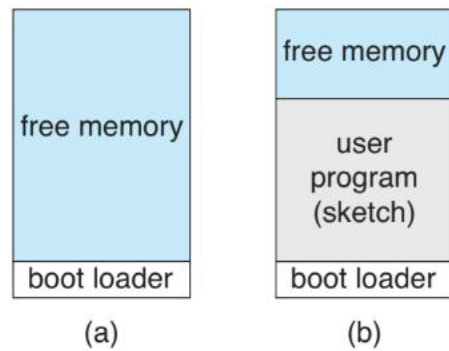
**Figure 1.24** Cloud computing.

\chapter 02



**Figure 2.6** The handling of a user application invoking the `open()` system call.

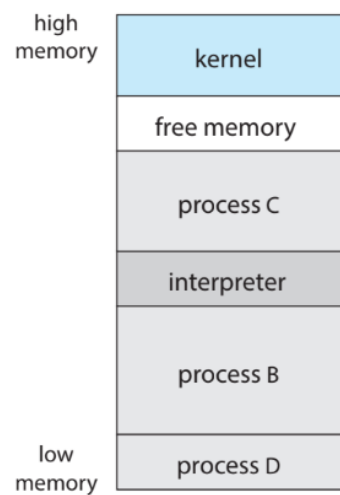
## er 2 Operating-System Structures



**Figure 2.9** Arduino execution. (a) At system startup. (b) Running a sketch.

## 2.3 System Calls

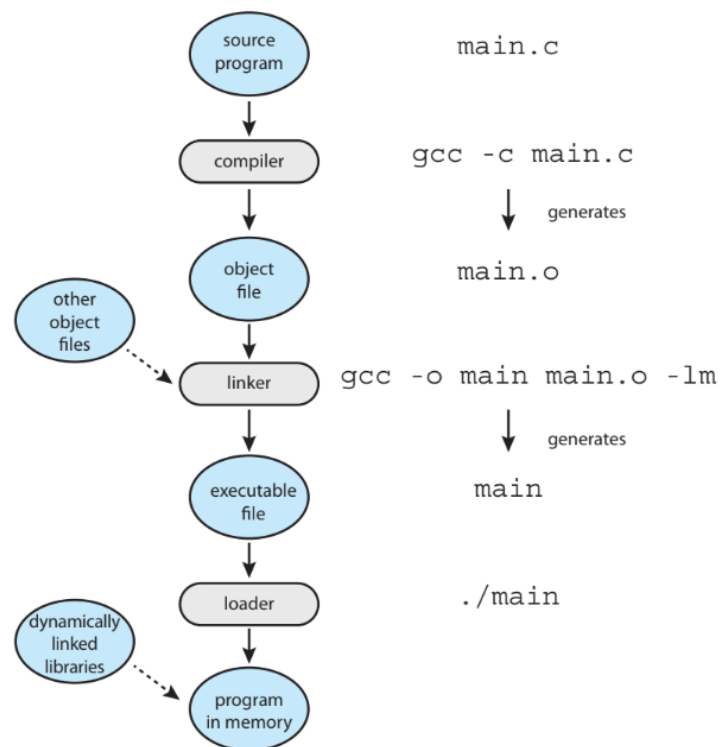
71



**Figure 2.10** FreeBSD running multiple programs.

system call to terminate, returning to the invoking process a status code of 0 or

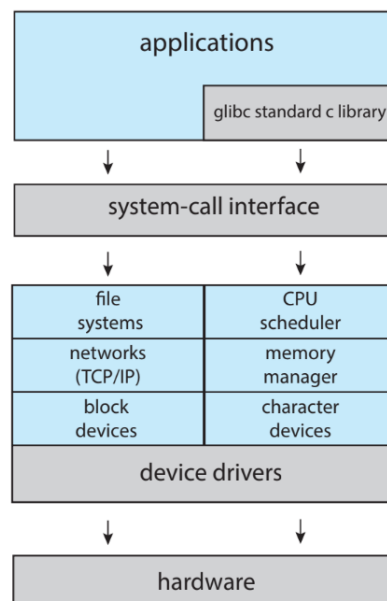
## Chapter 2 Operating-System Structures



**Figure 2.11** The role of the linker and loader.

### 2.8 Operating-System Structure

83



**Figure 2.13** Linux system structure.

Monolithic structure

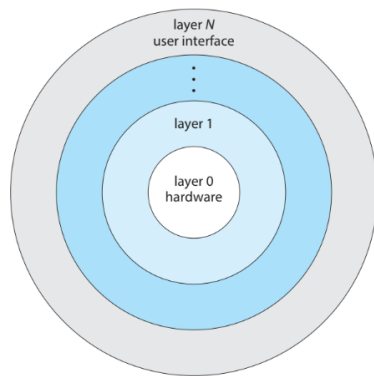


Figure 2.14 A layered operating system.

## 2.8 Operating-System Structure

85

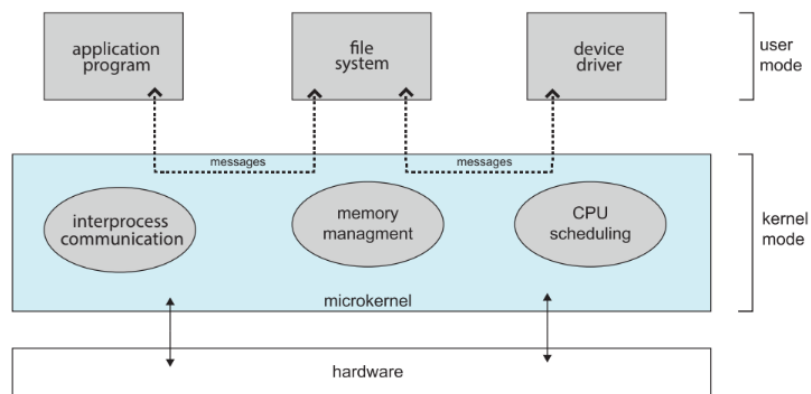
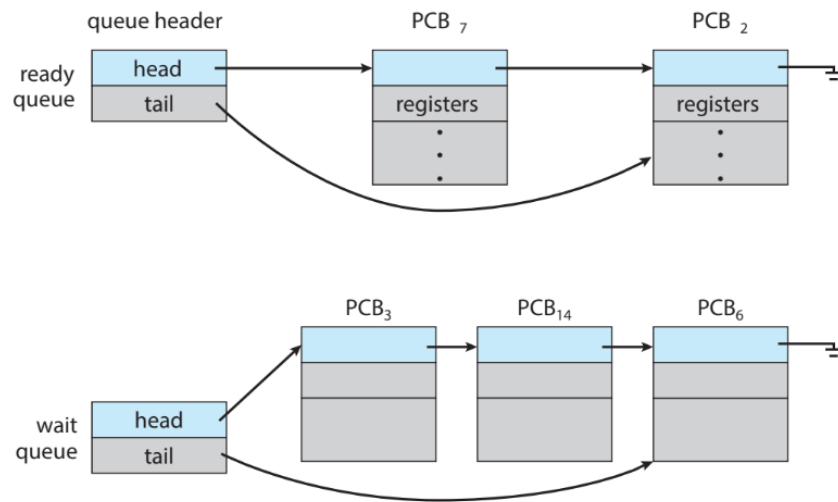


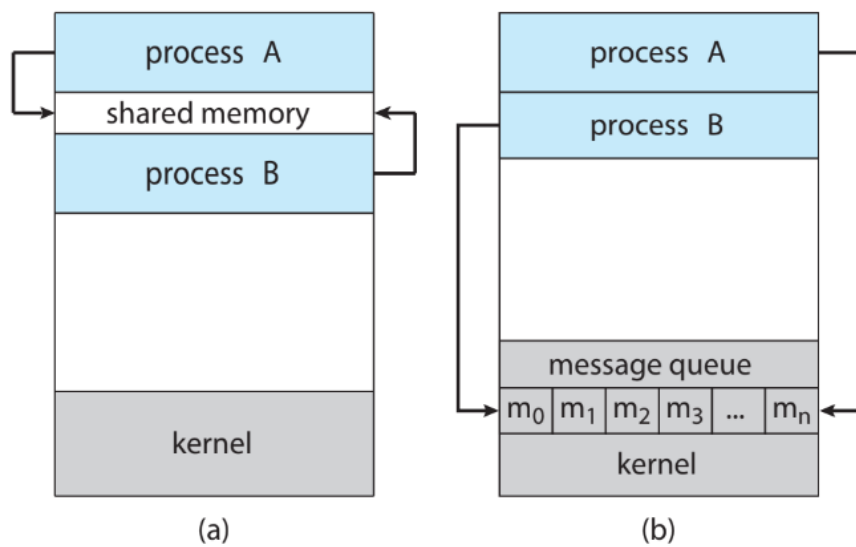
Figure 2.15 Architecture of a typical microkernel.



**Figure 3.4** The ready queue and wait queues.

### 3.5 IPC in Shared-Memory Systems

125



**Figure 3.11** Communications models. (a) Shared memory. (b) Message passing.