OPERATING SYSTEMS

Chapter 1
Introduction

What is an Operating System?

An **operating system** is a program that manages a computer's hardware.

It also provides a basis for application programs and acts as an intermediary between the computer user and the computer hardware.

What Is an Operating System?

As a manager, the operating system has two basic functions:

OS oversees all hardware resources and allocates them to user and applications as needed

Performs many low-level tasks on behalf of users and application programs

OS provides services for two reasons:

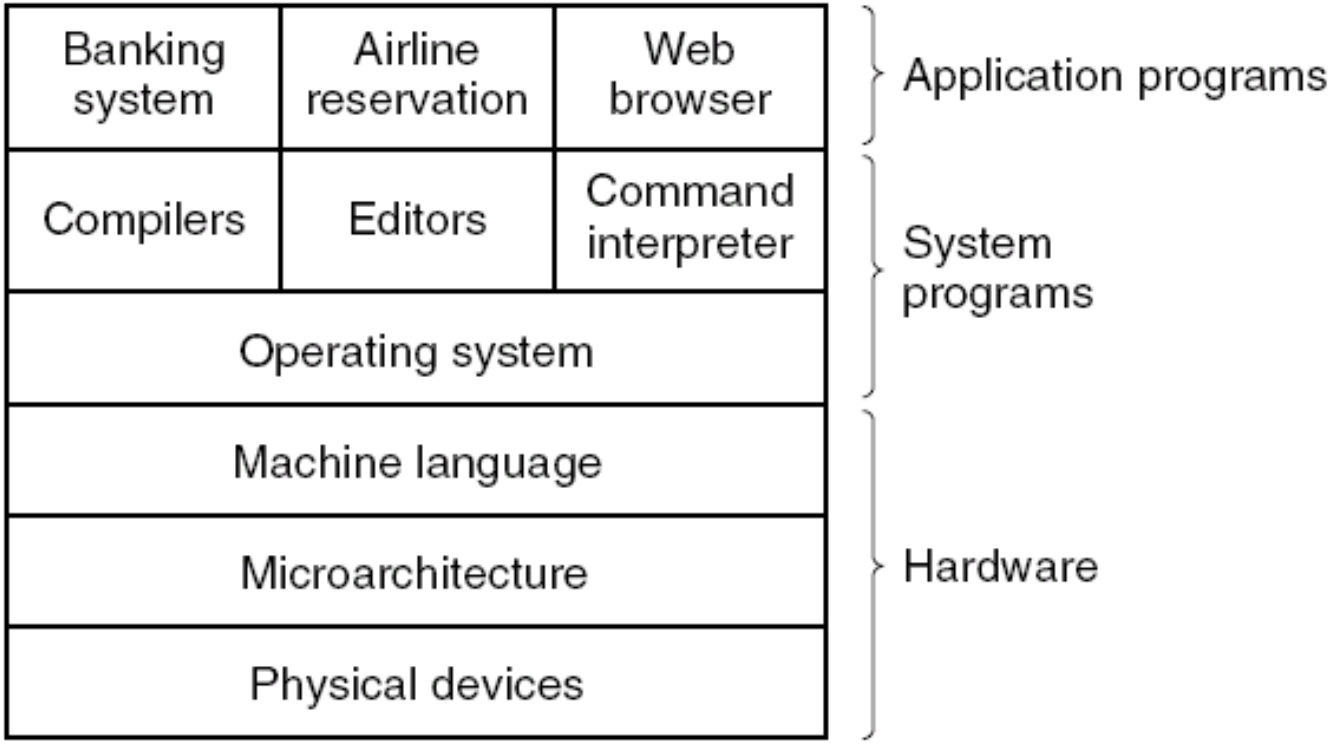
It is efficient. If many APs perform similar tasks, it makes sense to create services accessible to all applications. OS services are a form of code reuse.

Providing many low-level functions enables OS to maintain absolute control over hardware resources

OS Types

- Mainframe OS
- Server OS
- Multiprocessor OS
- PC OS
- Embedded OS
- Real-Time OS
- Smart Card OS

The Modern Computer System

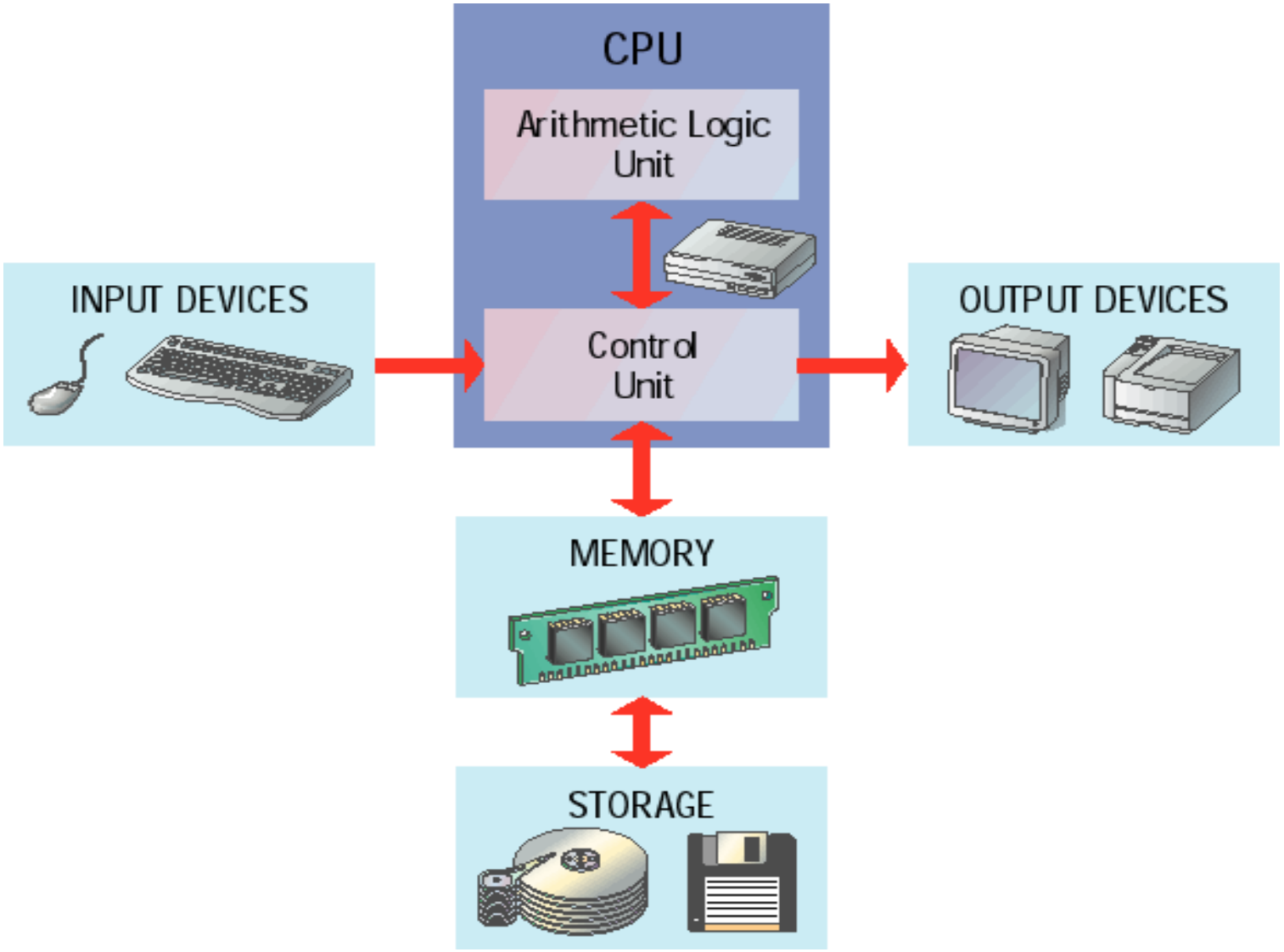


A computer system consists of hardware, system programs, and application programs.

Computer System Components

1. Hardware – provides basic computing resources (CPU, memory, I/O devices).
2. Operating system – controls and coordinates the use of the hardware among the various application programs for the various users.
3. Applications programs – define the ways in which the system resources are used to solve the computing problems of the users (compilers, database systems, video games, business programs).
4. Users (people, machines, other computers).

Computer Hardware



- I/O devices and the CPU can execute concurrently
- A number of device controllers connected through a common bus that provides access to shared memory
- Each device controller is in charge of a particular device type
- Each device controller has a local buffer
- CPU moves data from/to main memory to/from local buffers
- Device controller communicates with CPU by using an interrupt

Computer-System Operation

For a computer to start running—for instance, when it is powered up or rebooted—it needs to have an initial program to run.

This initial program, or **bootstrap program**, is stored within the computer hardware in read-only memory (ROM) or electrically erasable programmable read-only memory (EEPROM), known by the general term firmware.

It initializes all aspects of the system, from CPU registers to device controllers to memory contents.

Computer-System Operation

The bootstrap program must locate the operating-system kernel and load it into memory.

Once the kernel is loaded and executing, it can start providing services to the system and its users.

Once this phase is complete, the system is fully booted, and the system waits for some event to occur.

Interrupts

- If there are no processes to execute, no I/O devices to service, and no users to whom to respond, an operating system will sit quietly, waiting for something to happen.
- Events are almost always signaled by the occurrence of an interrupt or a trap.
- A trap (or an exception) is a software-generated interrupt caused either by an error (for example, division by zero or invalid memory access) or by a specific request from a user program that an operating-system service be performed.

Interrupts

The occurrence of an event is usually signaled by an **interrupt** from either the hardware or the software.

Hardware may trigger an interrupt at any time by sending a signal to the CPU, usually by way of the system bus.

Software may trigger an interrupt by executing a special operation called a system call

Interrupts

When the CPU is interrupted, it stops what it is doing and immediately transfers execution to a fixed location. The fixed location (**interrupt vector)** usually contains the starting address where the **service routine** for the interrupt is located.

The interrupt service routine executes; on completion, the CPU resumes the interrupted computation.

Interrupt architecture must save the address of the interrupted Instruction.

Thus, operating system is interrupt driven!

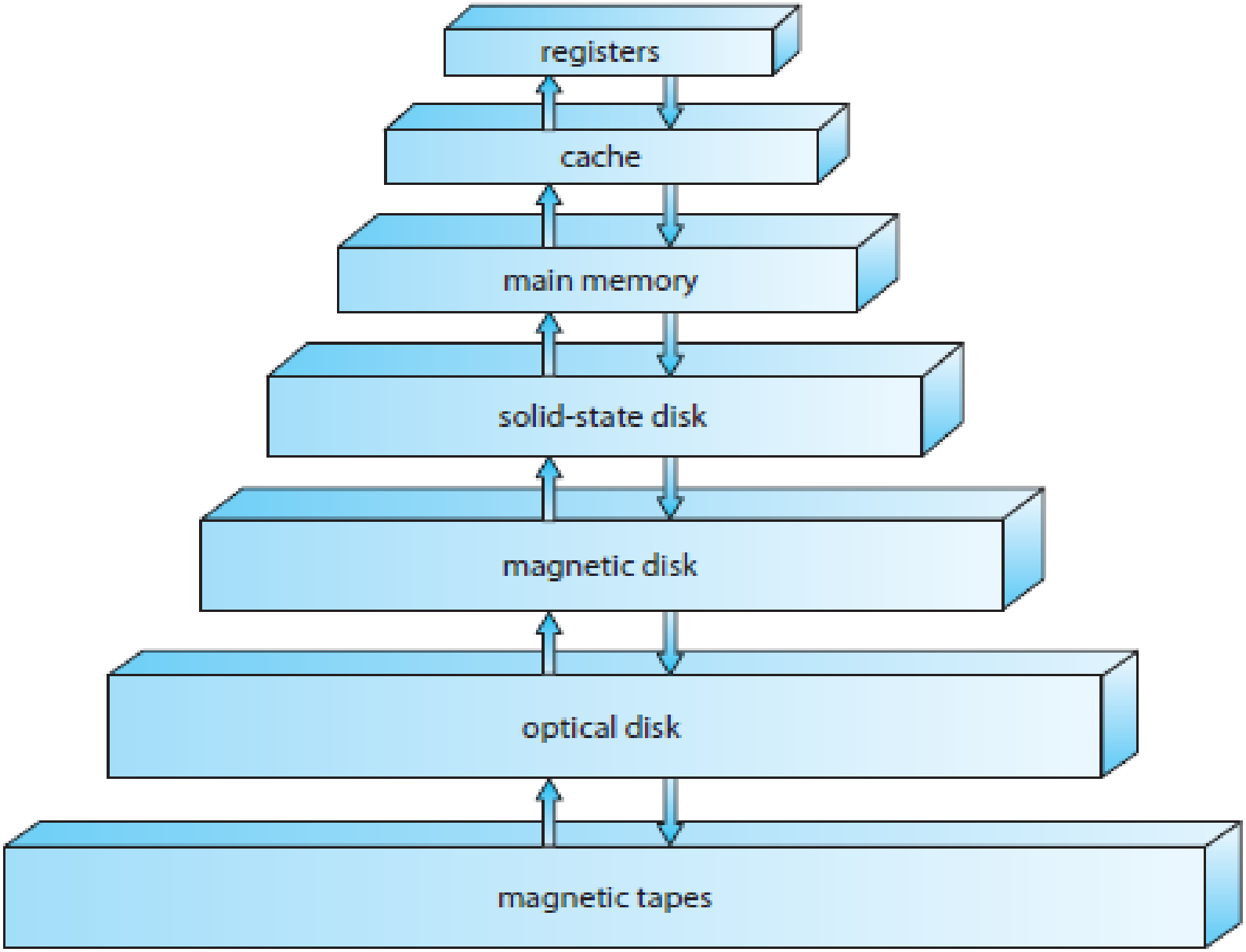
Storage Structure

- **Main memory** – only large storage media that the CPU can access directly. It is **Random access** and **volatile!**
- **Secondary storage** – extension of main memory that provides large **nonvolatile** storage capacity.
- Disk surface is logically divided into **tracks**, which are subdivided into **sectors!**
- The **disk controller** determines the logical interaction between the device and the computer

Storage Hierarchy

- Storage systems organized in hierarchy

- Speed
- Cost
- Volatility


registers
cache
main memory
solid-state disk
magnetic disk
optical disk
magnetic tapes

Caching

Information in use copied from slower to faster storage temporarily.

Faster storage (cache) checked first to determine if information is there.

If it is, information used directly from the cache (fast).

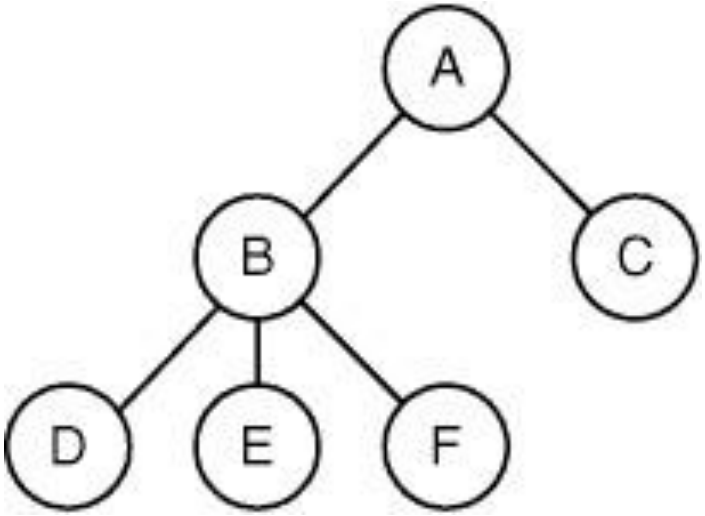
If not, data copied to cache and used there.

Cache smaller than storage being cached.

OS Concepts

Process is a running program, for each process there is **address space**, which is a list of memory locations. Also, associated with each process is a set of **resources**.

Processes

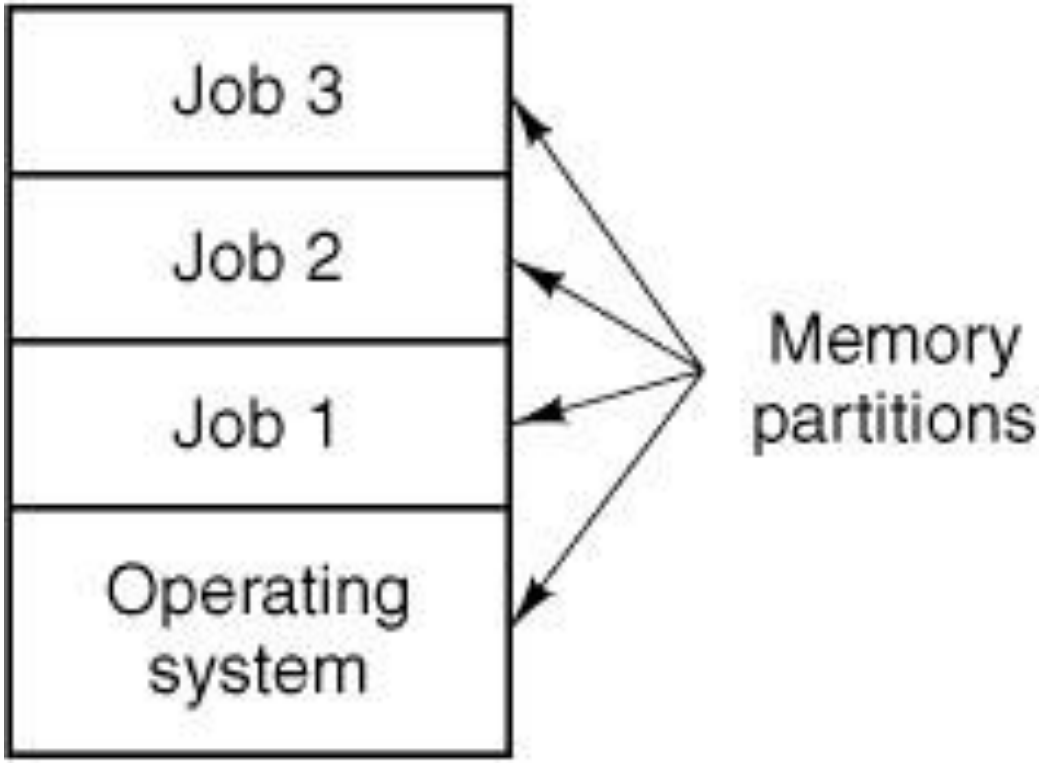


A process tree. Process A created two child processes, B and C. Process B created three child processes, D, E, and F.

Address Space

- Each computer has RAM to store executing programs.
- In a very simple OS, only one program at a time is in memory.
- More sophisticated OS allow multiple programs to be in RAM,

RAM for Multiprogramming



A multiprogramming system with three jobs in memory.

Operating System Structure

Multiprogramming needed for efficiency, Single user cannot keep CPU and I/O devices busy at all times.

Multiprogramming organizes jobs (code and data) so CPU always has one to execute.

One job selected and run via **job scheduling!**

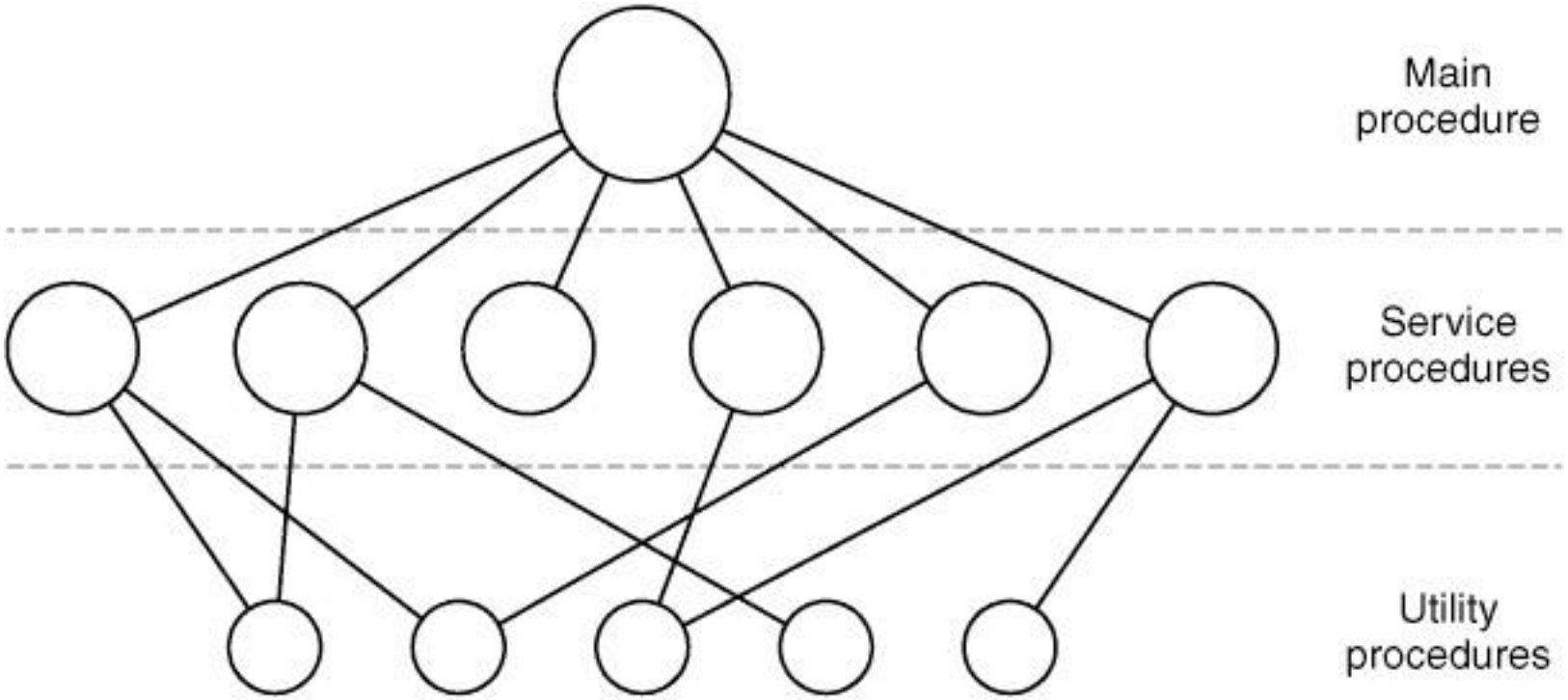
When it has to wait (for I/O for example), OS switches to another job.

Timesharing (multitasking) is logical extension in which CPU switches jobs so frequently that users can interact with each job while it is running, creating **interactive** computing.

Basic Structure for Operating System

1. A main program that invokes the requested service procedure
2. A set of service procedures that carry out the system calls
3. A set of utility procedures that help the service procedures

Layered Systems



A simple structuring model