CS 452 Operating Systems

Introduction

Dr. Denton Bobeldyk

Slide Overview

Source: https://www.travelandleisure.com/trip-ideas/beach-vacations/best-beaches-in-hawaii

What is an Operating System?

- Manages the computer's hardware
- Intermediary between the computer user and the computer hardware
- Basis for application programs to interface with

Where are computers used?

Where are computers used?

- Toasters
- Cars
- Ships
- Spacecraft
- Homes
- Businesses

What is an Operating System?

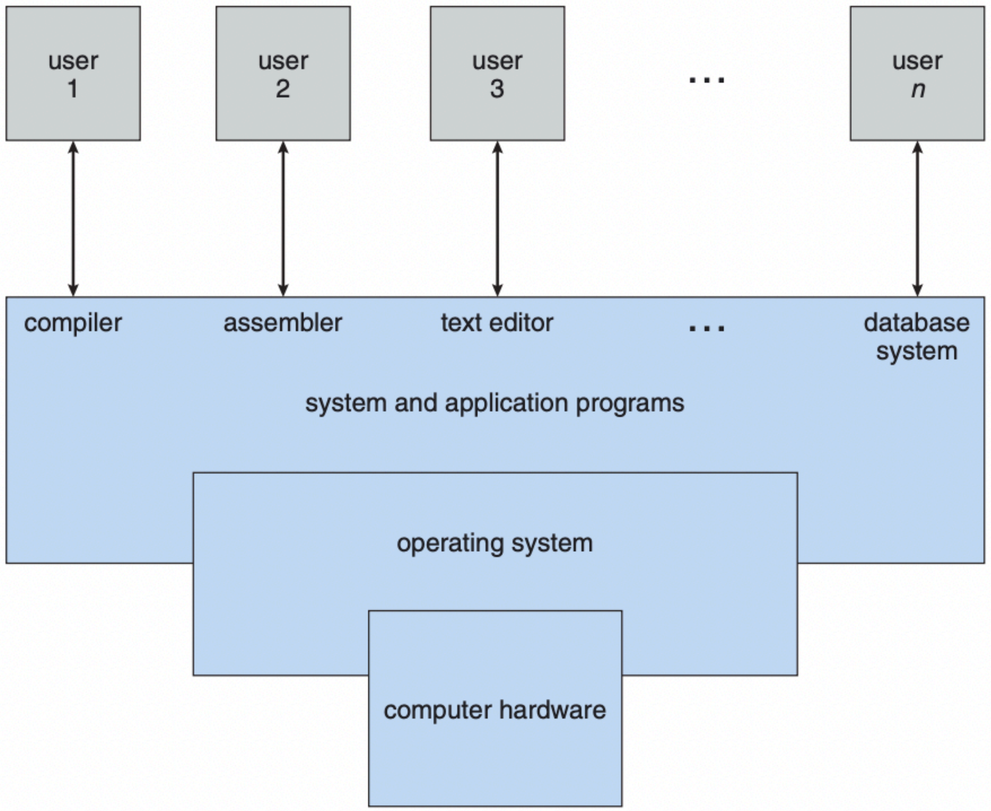

Figure 1.1 Abstract view of the components of a computer system.

Operating system purposes

- Mainframe - designed primarily to optimize utilization of hardware
- PC - support complex games, business applications, everything in between
- Mobile - provide an environment for easy user interface

What is an operating system?

- Operating systems are designed to be:
  - Convenient
  - Efficient
  - Combination of the above

What does an operating system do?

- A computer system can be divided into four components:
  - Hardware
  - Operating system
  - Application Programs
  - Users

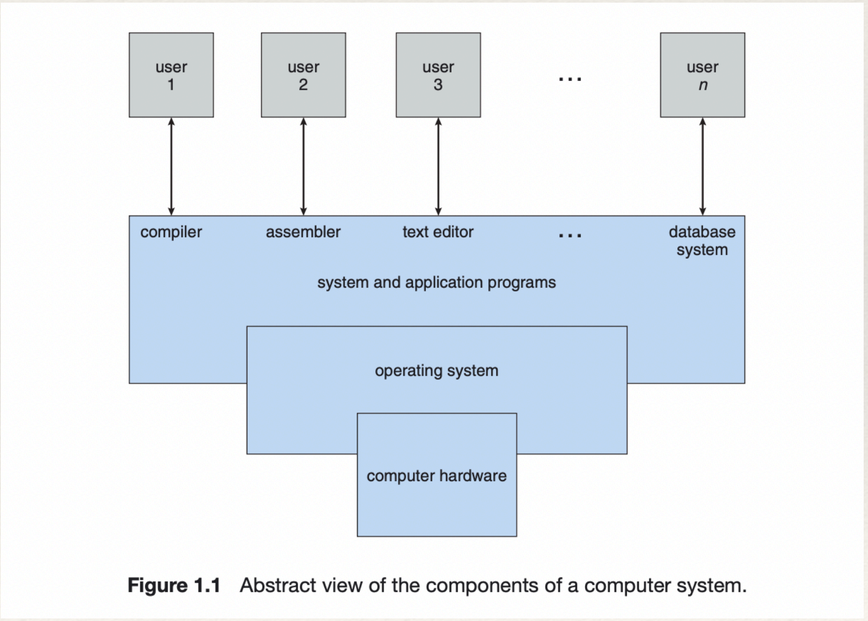

Figure 1.1 Abstract view of the components of a computer system.

What is an Operating System?

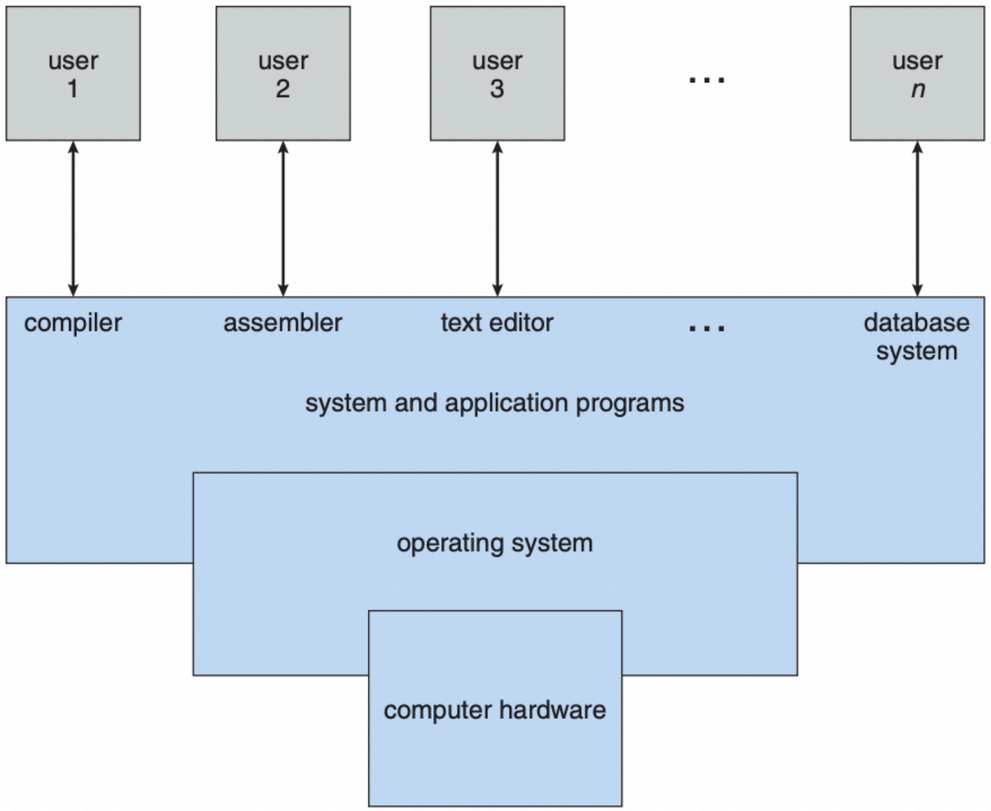

Figure 1.1 Abstract view of the components of a computer system.

User's View

- Answer the question for each of the following {PC/Mobile Device/EOS machine}
  - How is this device used?
    - Single User/Multiple User?
    - Games/Productivity
    - User Interface, command line, touch, console, GUI?

System View

- Operating System can be viewed as a resource allocator:
  - CPU time
  - Memory space
  - File-storage space
  - I/O devices

Class Exercise

- Split into small groups
- Determine how to solve the following:
  - Each student in the class must write their name and favorite movie on the white board
  - There are only two markers, one that writes very clearly, one that writes clearly but very lightly
  - How would you handle resource (the markers) allocation?

System View

- Operating System can be viewed as a control program:
  - Manage execution of user programs to prevent
    - errors
    - improper use
  - Concerned with the operation and control of I/O devices

Defining an Operating System

- No clear definition
- Kernel ‘program’ - YES
- System ‘programs’ - YES
- Application programs - NO

Defining an Operating System

- In 1998, the US States Department of Justice filed a law suit against Microsoft for what??

Defining an Operating System

- In 1998, the US States Department of Justice filed a law suit against Microsoft that Microsoft included too much functionality in its operating system limiting competition from application vendors (e.g., their web browser). They were found guilty of using its operating system monopoly to limit competition

Modern computer system

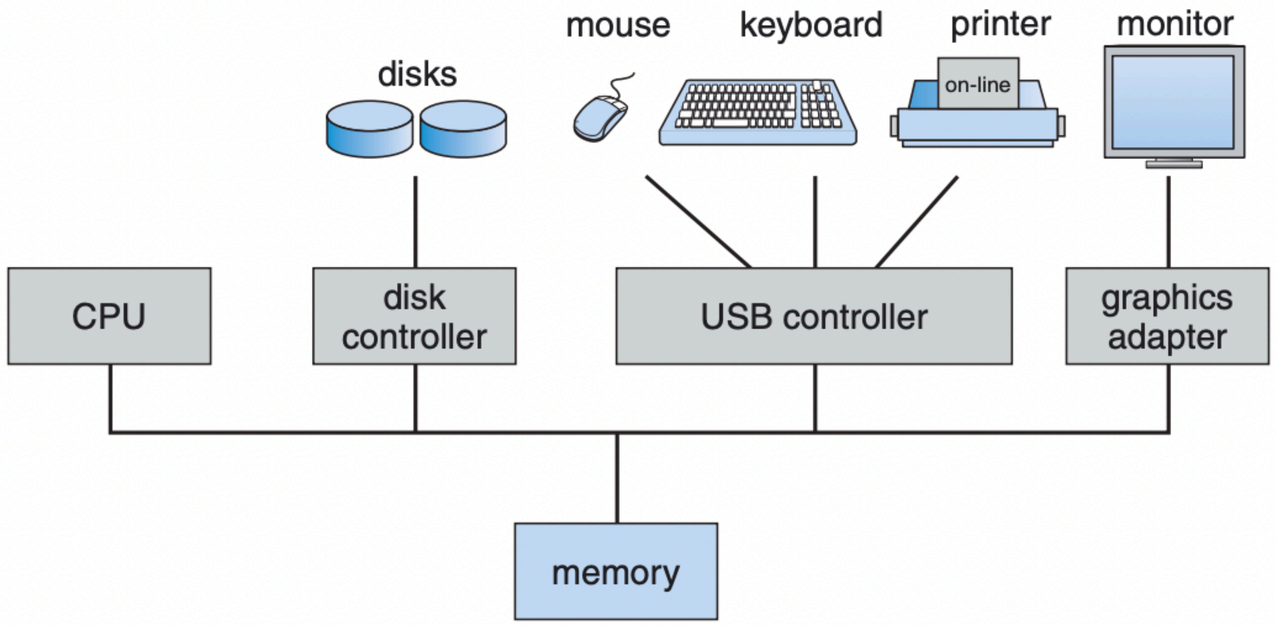

Figure 1.2 A modern computer system.

Computer Startup

- Initial program, bootstrap program loads
  - Stored in firmware
    - ROM (read-only memory)
    - EEPROM (Electrically erasable programmable read-only memory)
  - Initializes all aspects of the system
  - Load the operating system kernel into memory

Computer Startup

- System processes/daemons
  - Programs loaded into memory at boot time that run the entire time the kernel is running
- Interrupt
  - Signals an event that has occurred.
  - Hardware may trigger them at any time, usually by the way of a system bus
  - Software may trigger them by using a ‘system call’ (can also be referred to as a monitor call)
  - CPU transfers execution to a fixed location (starting address for a service routine for that interrupt)

Interrupts - Review

- Hardware generated
- Software generated
  - Triggered through a system/monitor call

Storage Structure

- CPU - loads instructions from memory
- RAM - random-access memory
  - DRAM - dynamic random-access memory
  - ROM
  - EEPROM (Electrically erasable programmable read-only memory)

Main memory

- Ideal for all programs to reside in memory permanently
- This is not possible:
  - Main memory is too small to store all needed programs
  - Main memory is ‘volatile’, which means it loses its contents when power is turned off, or otherwise lost

Secondary storage

- Extension of main memory
- Hold large quantities of data permanently
- Hard Disk Drive (HDD) common secondary-storage device

Secondary storage

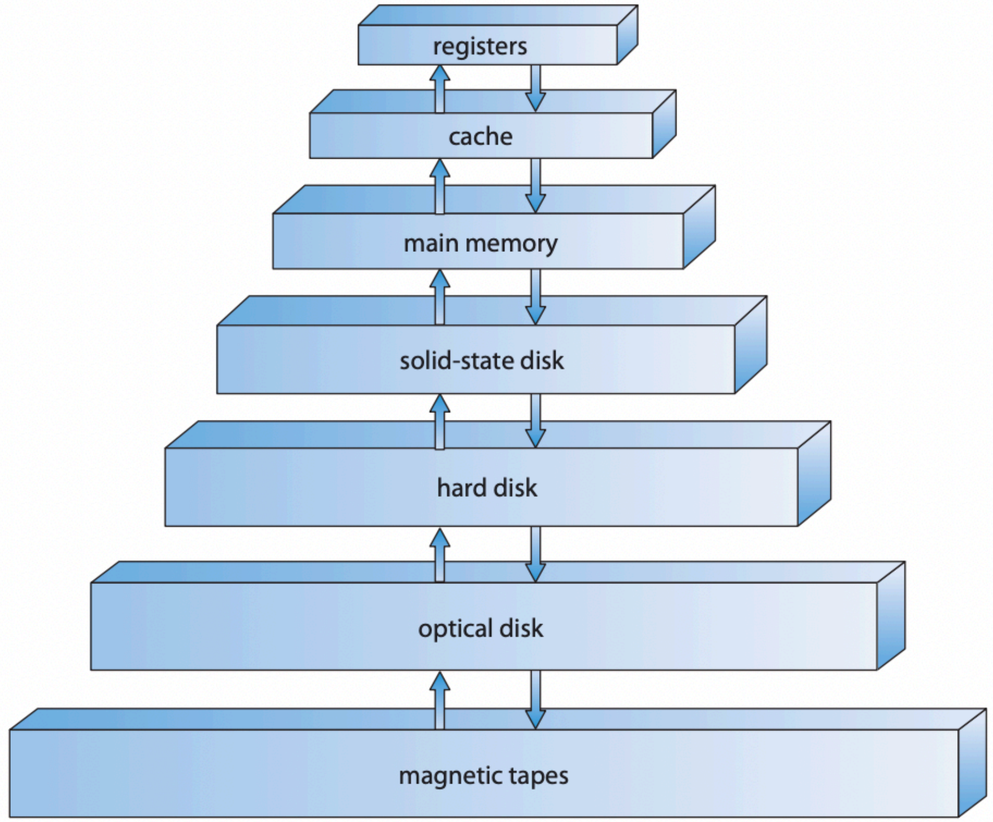

Figure 1.4 Storage-device hierarchy.

Secondary storage

Cost / Access speed

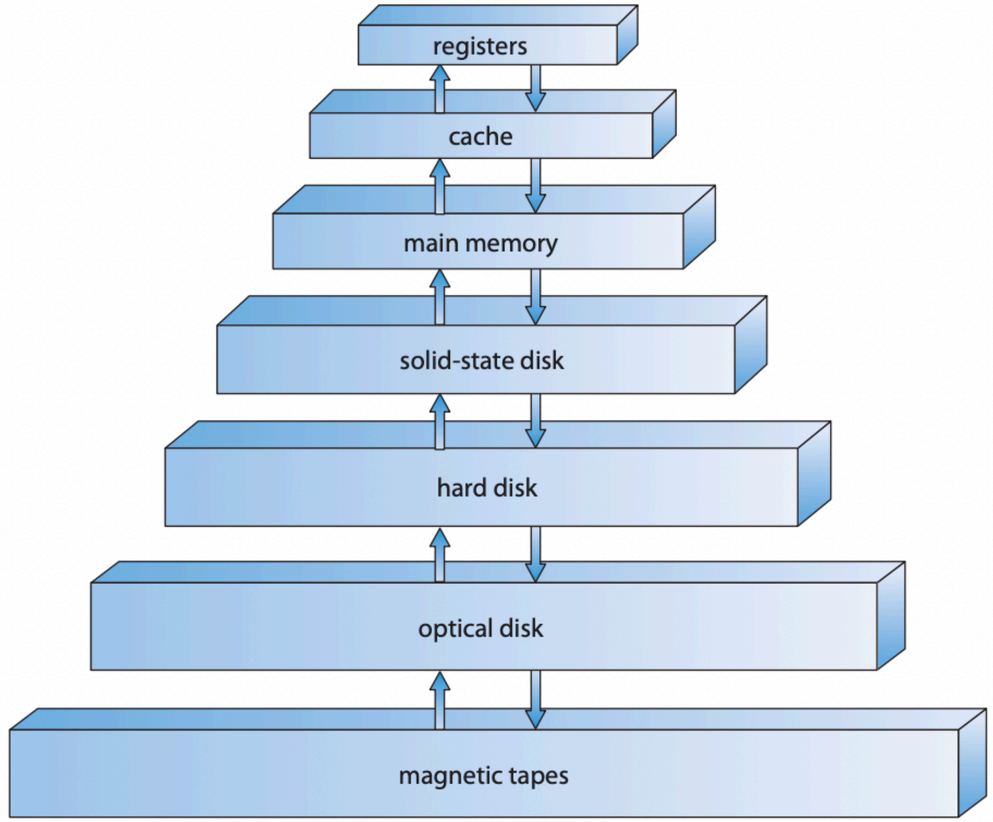

Figure 1.4 Storage-device hierarchy.

Secondary storage

- Solid-state disks - faster than hard disks, nonvolatile
- NVRAM - nonvolatile storage. DRAM with battery backup power. Fast as DRAM and nonvolatile

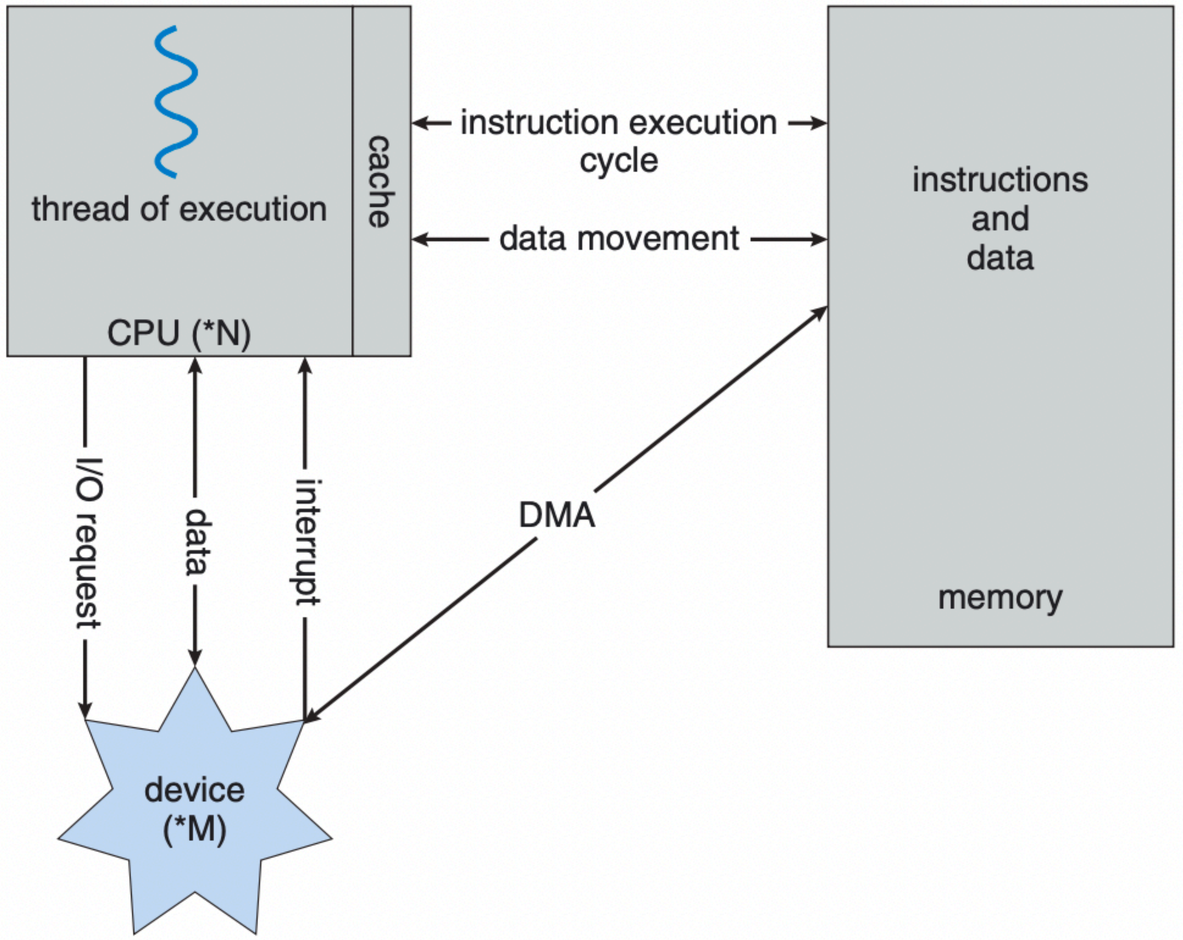

Figure 1.5 How a modern computer system works.

Computer Overview

Multiprocessor Systems

- System with 2 or more processors
- Shared computer bus, clock, memory, peripheral devices
  - Varying shared components, not always consistent
- Initially only available in servers, but have made their way to desktops, laptops and mobile devices

Multiprocessor Systems

- Increased throughput
- Economy of scale
  - Shared multi cpu access to a single storage device as opposed to multiple cpus individually housed and multiple storage devices
- Increased reliability

Multiprocessor Systems

- Asymmetric multiprocessing
  - One boss, assigns work to the worker processors
- Symmetric multiprocessing (SMP)
  - Each processor performs all tasks, all peers, no boss

Multicore Systems

- Multicore - Multiple computing cores on a single chip

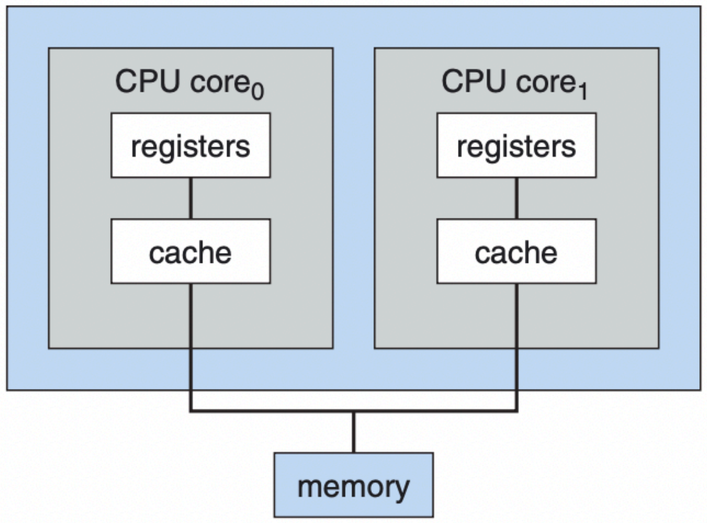

Figure 1.7 A dual-core design with two cores placed on the same chip.

Multiprogramming

- Job pool
  - Resides on disk
  - Small number of jobs reside in memory
  - CPU chooses a job and begins to work on it

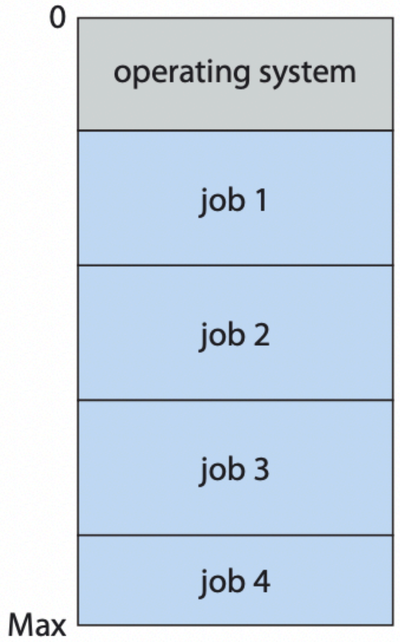

Figure 1.9 Memory layout for a multiprogramming system.

Multiprogramming

- Job pool
  - Resides on disk
  - Small number of jobs reside in memory
  - CPU chooses a job and begins to work on it

- What "real-world" scenarios does this mimic?

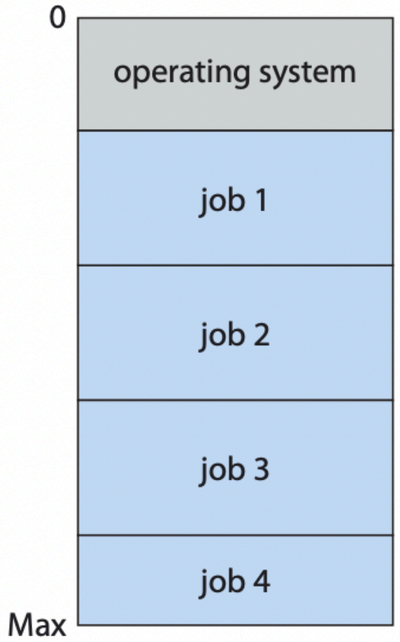

Figure 1.9 Memory layout for a multiprogramming system.

Dual-mode Operation

- User Mode
  - User application processes
- Kernel Mode
  - Operating system processes

Dual-mode Operation

- Why?
  - Provides system protection
  - Prevent user processes from:
    - Infinite loops
    - “Hogging” resources
    - Affecting other users

Dual-mode Operation

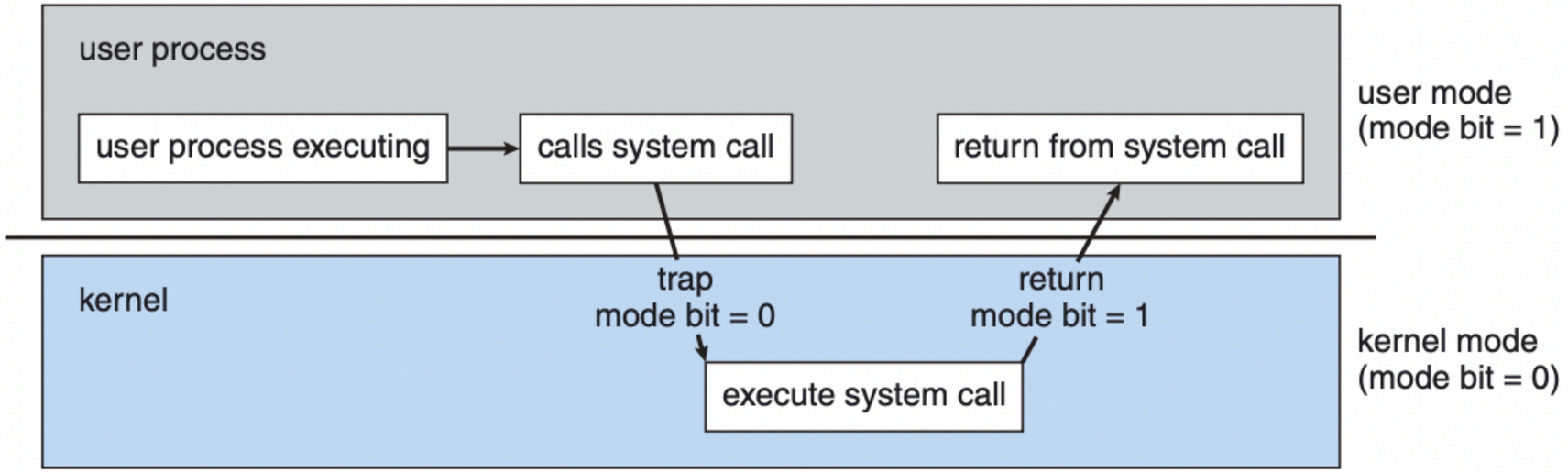

Figure 1.10 Transition from user to kernel mode.

Timer

- Set to interrupt the computer after a fixed period of time
- Ensures a process doesn't get stuck in an infinite loop and starve out the other processes.

Process Management

- A program does nothing unless its instructions are executed by a CPU
- A program in execution is a process
  - Also referred to as a ‘job’
  - Active entity as opposed to a program which is passive
- Single threaded process has one program counter which specifies the next instruction to execute

Process Management

- Operating system is responsible for:
  - Scheduling processes and threads on the CPU
  - Creating/Deleting user and system processes
  - Suspending and resuming processes
  - Providing mechanisms for process synchronization
  - Providing mechanisms for process communication

Memory Management

- Main memory
  - Large array of bytes
  - Each byte has its own address
  - Quickly accessible data shared by the CPU and I/O devices
- Memory Management discussed in Chapters 7 and 8

Storage Management

- Storage Management
  - Creating and deleting directories to organize files
  - Supporting primitives for manipulating files and directories
  - Mapping files onto secondary storage
  - Backing up files on stable (nonvolatile) storage media
- Memory Management discussed in Chapters 10 and 11

Cache

- Information normally kept in some storage system (e.g., main memory)
- As it is used, copied to a faster storage system - the cache

Cache Types

Level	1	2	3	4	5
Name	registers	cache	main memory	solid state disk	hard disk
Typical size	< 1 KB	< 16MB	< 64GB	< 1 TB	< 10 TB
Implementation technology	custom memory with multiple ports CMOS	on-chip or off-chip CMOS SRAM	CMOS SRAM	flash memory	hard disk
Access time (ns)	0.25 - 0.5	0.5 - 25	80 - 250	25,000 - 50,000	5,000,000
Bandwidth (MB/sec)	20,000 - 100,000	5,000 - 10,000	1,000 - 5,000	500	20 - 150
Managed by	compiler	hardware	operating system	operating system	operating system
Backed by	cache	main memory	disk	disk	disk or tape

Cache Management

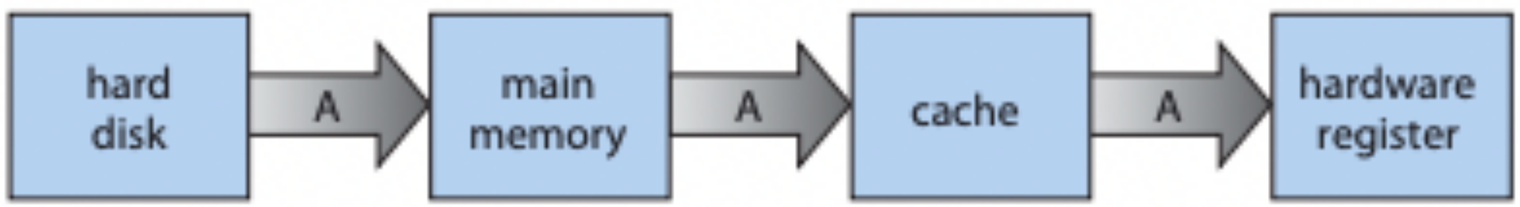

Figure 1.12 Migration of integer A from disk to register.