CS420: Operating Systems

OS Overview

James Moscola
Department of Engineering & Computer Science
York College of Pennsylvania

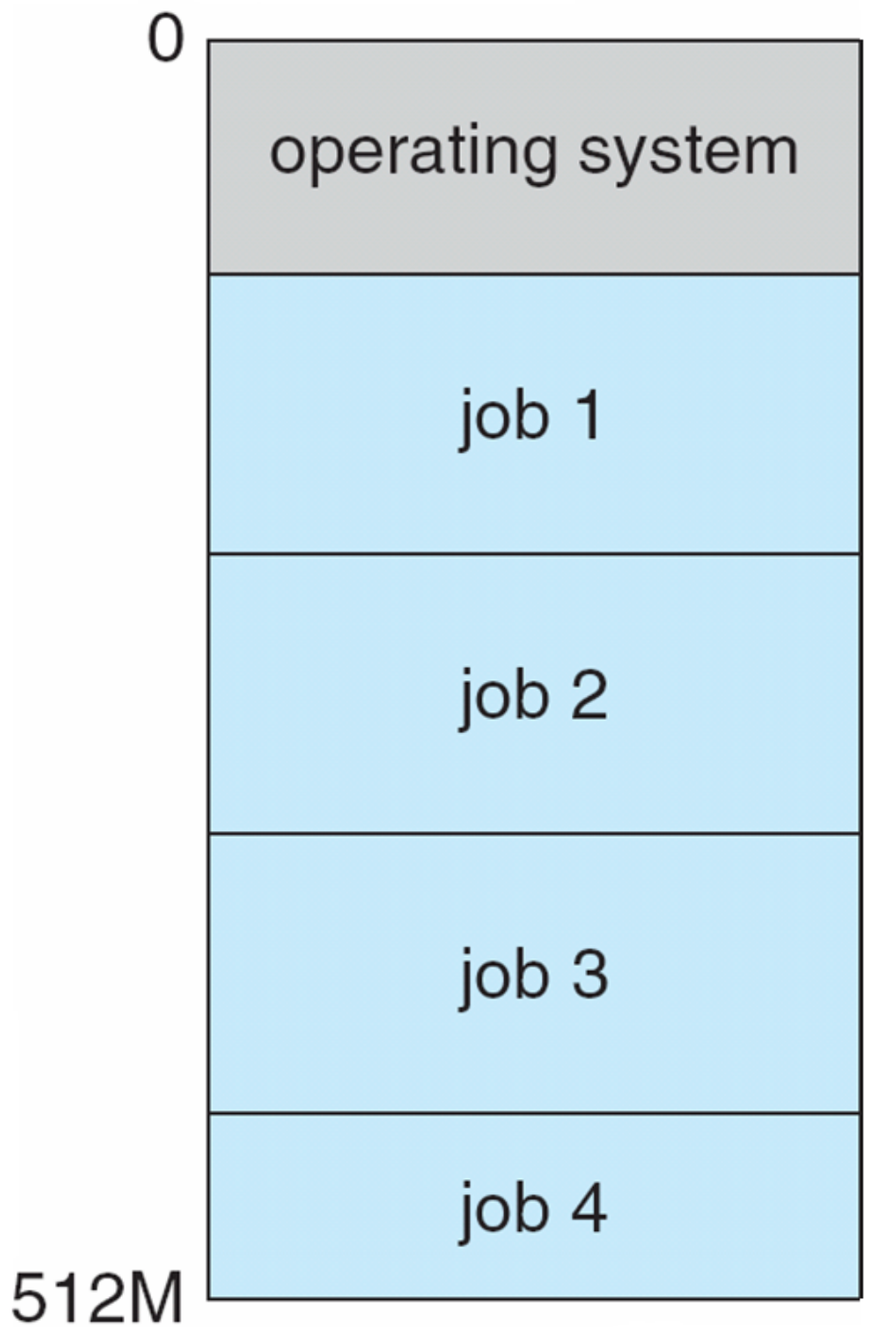
Operating System Structure

- **Multiprogramming needed for efficiency**
  - Single application cannot keep CPU and I/O devices busy at all times
  - Multiprogramming organizes jobs (code and data) so CPU always has something to execute
  - A subset of total jobs in system is kept in memory
  - One job selected and run via job scheduling
  - When it has to wait (for I/O for example), OS switches to another job

Operating System Structure

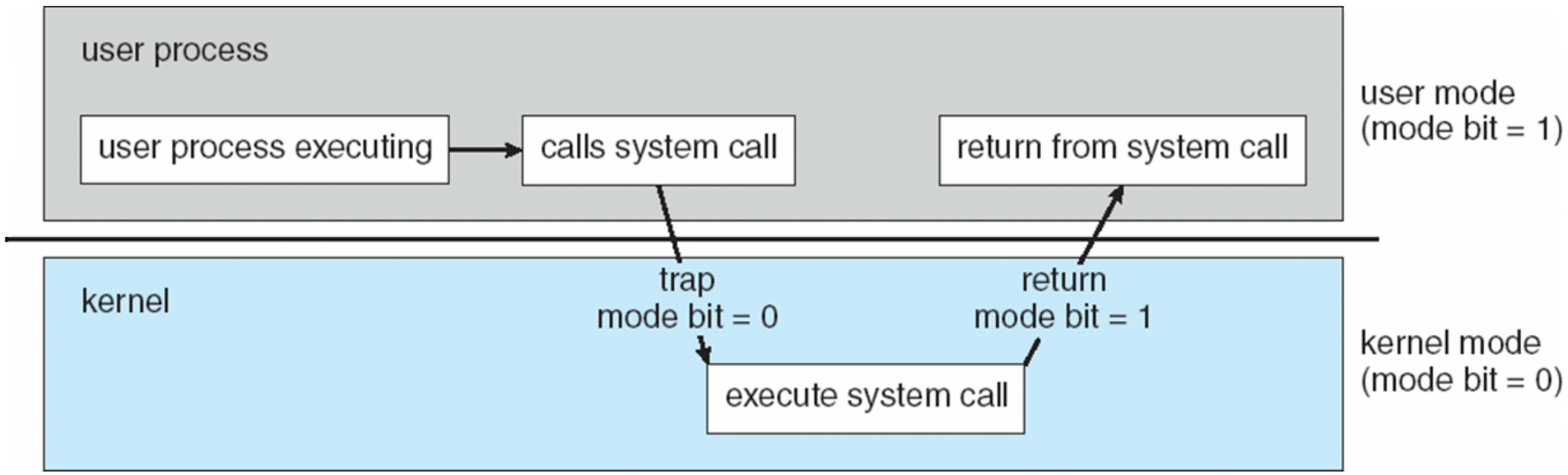
- **Timesharing (multitasking) is logical extension in which CPU switches jobs so frequently that users can interact with each job while it is running, creating interactive computing**
  - Response time should be < 1 second
  - Each user has at least one program executing in memory (a process)
  - If several jobs ready to run at the same time CPU scheduling is required
  - If processes don’t fit in memory, swapping moves them in and out to run
  - Virtual memory allows execution of processes not completely in memory

Memory Layout for Multiprogrammed System



Dual-Mode Operation (User Mode & Kernel Mode)

- **Dual-mode operation allows OS to protect itself and other system components**
  - User mode and kernel mode
  - Mode bit provided by hardware
    - Provides system ability to distinguish when running user code or kernel code
    - Some instructions designated as privileged, only executable in kernel mode
    - System call changes mode to kernel mode, return from call resets it to user



Process Management

- **A process is a program in execution**
  - It is a unit of work within the system
  - A program is a passive entity, a process is an active entity
- **A process needs resources to accomplish its task**
  - CPU cycles, memory, I/O, files, etc.
  - Initialization data
- **When process terminates, OS reclaims any reusable resources**

Process Management (Cont.)

- **Single-threaded process has one program counter specifying location of next instruction to execute**
  - Process executes instructions sequentially, one at a time, until completion

- **Multi-threaded process has one program counter per thread**

- **Typical system has many processes running concurrently on 1 (or more) CPUs**
  - Some user processes running, some operating system processes running
  - Get concurrency by multiplexing the CPUs among the processes / threads
    - The CPU rapidly switches between multiple processes

Process Management Activities

- **The operating system is responsible for the following activities in connection with process management:**
  - Creating and deleting both user and system processes
  - Scheduling processes and threads on the CPUs
  - Suspending and resuming processes
  - Providing mechanisms for process synchronization
  - Providing mechanisms for process communication

Maintaining Control of Hardware

- **OS needs to monitor process activity to handle problems that occur**
- **Software error or request can create an exception or trap**
  - e.g. division by zero, request for operating system service
    - OS needs to recover
- **Other process problems include infinite loop, processes modifying each other or the operating system**
  - Timer to prevent infinite loop / process hogging resources
    - Set interrupt after specific period
    - Operating system decrements counter
    - When counter zero generate an interrupt
    - Set up before scheduling process to regain control or terminate program that exceeds allotted time

Memory Management

- **All data is read from and stored into memory when processing**
  - Data must be in memory before it can be read by CPU
- **All instructions must be in memory in order to execute**
- **Memory management determines what is in memory and when**
  - Optimizing CPU utilization and computer response to users
- **Memory management activities**
  - Keeping track of which parts of memory are currently being used and by whom
  - Deciding which processes (or parts thereof) and data to move into and out of memory
  - Allocating and deallocating memory space as needed

Storage Management

- **OS provides uniform, logical view of information storage**
  - Abstracts physical properties to logical storage unit - (i.e., a file)
  - Each medium is controlled by device driver (i.e., disk drive, tape drive)
    - Varying properties include access speed, capacity, data-transfer rate, access method (sequential or random)

Storage Management

- **File-System management**
  - Files usually organized into directories
  - Access control on most systems to determine who can access what
  - OS activities include
    - Creating and deleting files and directories
    - Primitives to manipulate files and directories
    - Mapping files onto secondary storage
    - Backup files onto stable (non-volatile) storage media

Mass-Storage Management

- **Usually disks are used to store data that does not fit in main memory or data that must be kept for a “long” period of time**
- **Entire speed of computer operation hinges on disk subsystem and its algorithms**
- **OS activities**
  - Mounting and unmounting
  - Free-space management
  - Storage allocation
  - Disk scheduling
- **Some storage need not be fast**
  - Tertiary storage includes optical storage, magnetic tape
  - Still must be managed – by OS or applications

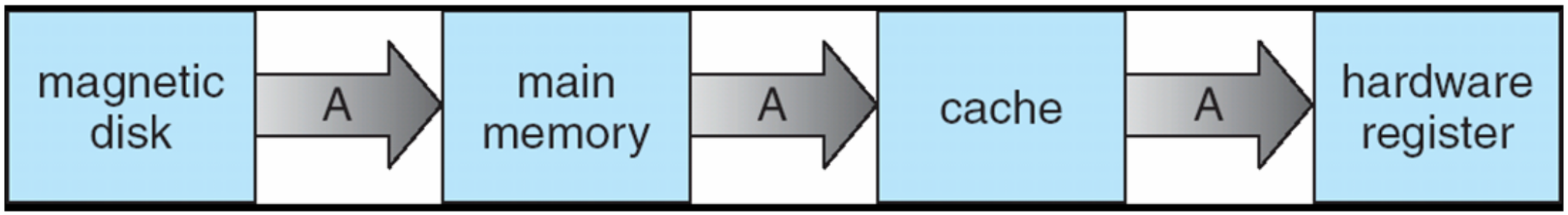
Performance of Various Levels of Storage

- **Movement between levels of storage hierarchy can be explicit or implicit**

Level	1	2	3	4	5
Name	registers	cache	main memory	solid-state disk	magnetic 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	magnetic 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

Migration of Integer A from Disk to Register

- **Multitasking environments must be careful to use most recent value, no matter where it is stored in the storage hierarchy**



- **Multiprocessor environment must provide cache coherency in hardware such that all CPUs have the most recent value in their cache**

- **Distributed environment situation even more complex**
  - Several copies of a datum can exist

I/O Subsystem

- **One purpose of OS is to hide peculiarities of hardware devices from the user**

- **I/O subsystem responsible for**
  - Memory management of I/O including:
    - Buffering (storing data temporarily while it is being transferred)
    - Caching (storing parts of data in faster storage for performance)
    - Spooling (the overlapping of output of one job with input of other jobs)
  - General device-driver interface
  - Drivers for specific hardware devices

Protection and Security

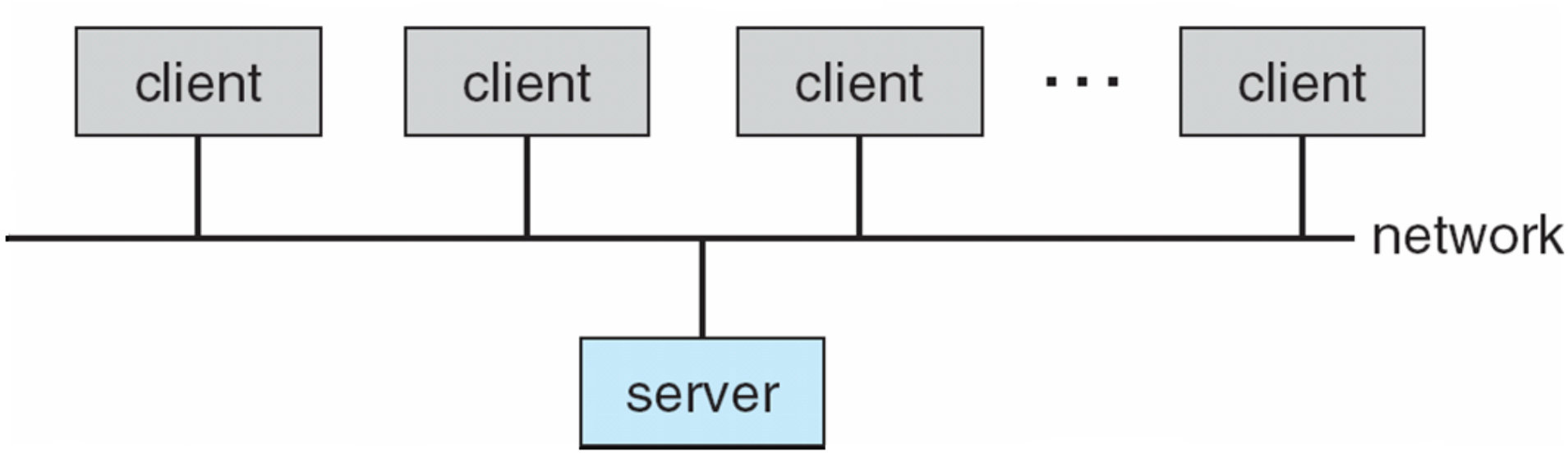
- **Protection – any mechanism for controlling access of processes or users to resources defined by the OS**
  - Systems generally first distinguish among users, to determine who can do what
    - User identities (user IDs, security IDs) include name and ID number, one per user
    - User ID is associated with all files, processes of that user to determine access control
    - Group identifier (group ID) allows set of users to be defined and controls managed, then also associated with each process, file
    - Privilege escalation allows user to change to effective ID with more rights
- **Security – defense of the system against internal and external attacks**
  - Huge range, including denial-of-service, worms, viruses, identity theft, theft of service

Computing Environments

- **Traditional computer**
  - Blurring over time
  - Office environment
    - PCs connected to a network, terminals attached to mainframe or minicomputers providing batch and timesharing
    - Now portals allowing networked and remote systems access to same resources
  - Home networks
    - Used to be single system, then modems
    - Now contain firewalls, and networked systems

Computing Environments (Cont.)

- **Client-Server Computing**
  - Dumb terminals supplanted by smart PCs
  - Many systems now **servers**, responding to requests generated by **clients**
  - **Compute-server** provides an interface to client to request services (i.e., database)
  - **File-server** provides interface for clients to store and retrieve files



Distributed Computing

- **Collection of separate, possibly heterogeneous, systems networked together**
  - Network is a communications path
    - Local Area Network (LAN)
    - Wide Area Network (WAN)
    - Metropolitan Area Network (MAN)

- **Network Operating System provides features between systems across network**
  - Communication scheme allows systems to exchange messages
  - Illusion of a single system

Peer-to-Peer Computing

- **Another model of distributed system**

- **P2P does not distinguish clients and servers**
  - Instead all nodes are considered peers
  - May each act as client, server or both
  - Node must join P2P network
    - Registers its service with central lookup service on network, or
    - Broadcast request for service and respond to requests for service via discovery protocol

Web-Based / Cloud Computing

- **Computing that delivers computing, storage, and even applications as a service across a network**
  - Software as a service (SaaS)
    - Applications available via the internet (e.g. Google Docs)
  - Platform as a service (PaaS)
    - Software stack on internet available for use by application developers (e.g. AWS Elastic Beanstalk, Google App Engine, Microsoft Azure)
  - Infrastructure as a service (IaaS)
    - Servers or storage available via the internet (e.g. AWS, Microsoft Azure VMs, DigitalOcean, Google Compute Engine)