Today's Plan

Upcoming:

- Quiz 1
- Assignment 1

Last time:

- Understanding the internals of computer systems

Today's topics:

- Operating System Structure
- Process Management
- Main Memory Management
- File/Secondary Storage Management

Operating System Structure

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

Operating System Structure

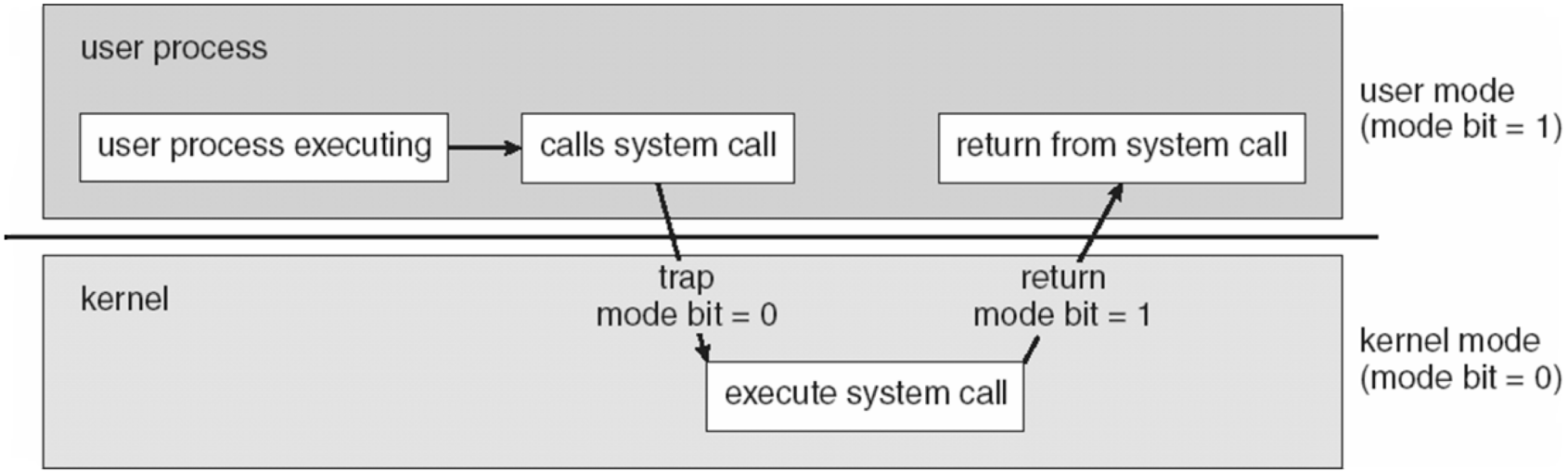
- **Timesharing (multitasking)** is a logical extension in which CPU switches jobs so frequently that users can interact with each job while it is running, creating interactive computing
  - Response time should be < 1 second
  - The user programs executing in memory ⇨ **processes**
  - If several jobs are ready to run at the same time ⇨ **CPU scheduling**
  - 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

Hardware Protection – Dual Mode

- Sharing system resources requires OS to ensure that an incorrect program cannot cause other programs to execute incorrectly
- **Dual-Mode operation**: Provide hardware support to differentiate between at least two modes of operations:
  1. *User mode:* Execution done on behalf of the user
  2. *Monitor mode* (also *kernel mode* or *system mode*): Execution done on behalf of the operating system
  - Instruction set is restricted in user mode
  - A program, running in user mode, attempting to execute a privileged instruction will cause a trap

System Calls

- **System calls** are used to request services from the OS
  - E.g. give me the current date/time, open a file for reading, etc.
- Executing a system call changes mode to kernel, return from call resets it to user

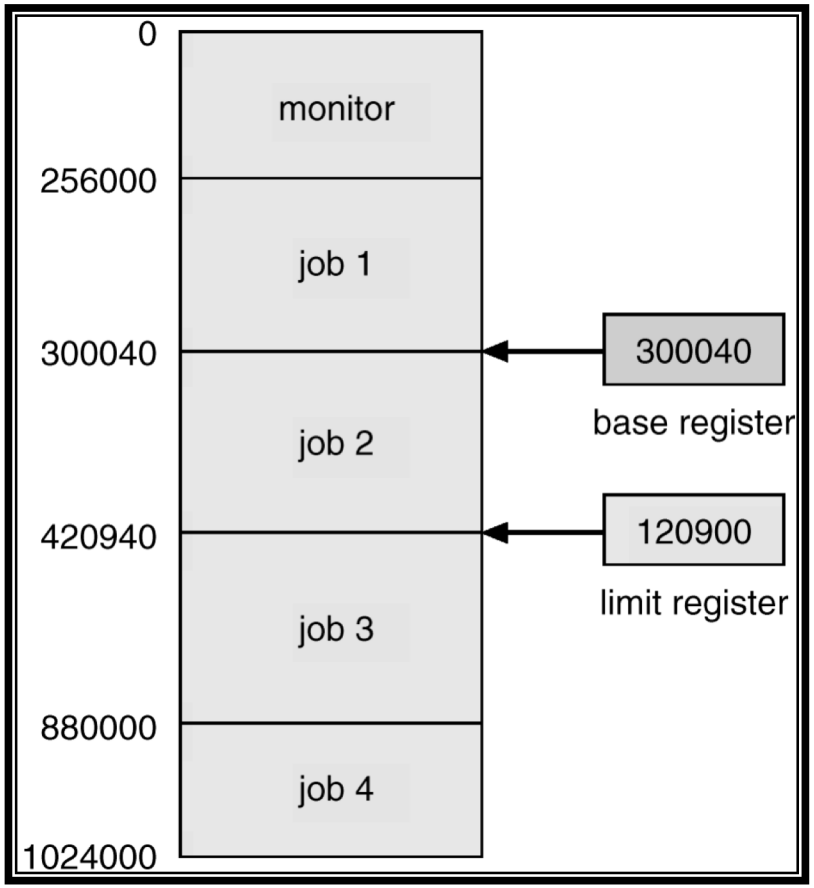


Hardware Protection – I/O & Memory Protection

- We must ensure that I/O devices are protected as well, to have *I/O protection* we ensure:
  - All I/O instructions are privileged instructions
- We must also ensure that processes (jobs) are not able to access each other's memory space
  - User jobs must also not be able to access the interrupt handlers or interrupt vectors

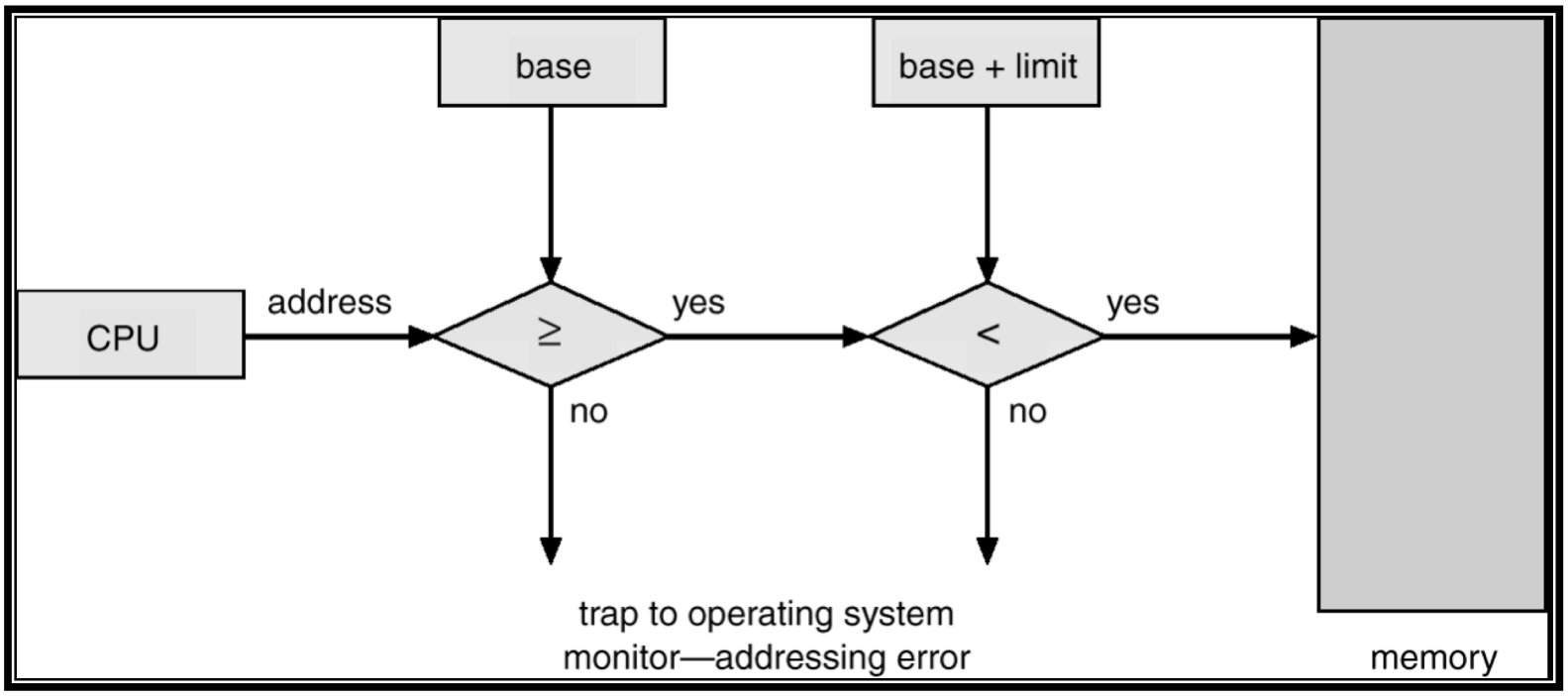
Hardware Protection – Memory Protection

- In order to have *memory protection*, add two registers that determine the range of legal addresses a program may access:
  - **Base Register:** holds the smallest legal physical memory address
  - **Limit Register:** contains the size of the range



Hardware Protection – Memory Protection

- Checking memory addresses in hardware:



Hardware Protection – CPU Protection

- We must also ensure that no one job is using up all the CPU cycles
- *Timer* – interrupts computer after specified period to ensure operating system maintains control
  - Timer is decremented every clock tick
  - When the timer reaches value 0, an interrupt occurs
- The timer is commonly used to implement *time sharing*

System Components – 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
  - E.g. CPU, memory, I/O devices, files
- Process termination requires reclaiming of any reusable resources
- A *single-threaded process* has one program counter specifying location of next instruction to execute
  - A process generally executes instructions sequentially until completion
- A *multi-threaded process* has one program counter per thread
- Typically a system has many processes, some user, some operating system running concurrently on one or more CPUs

System Components – Process Management

- The operating system is responsible for the following activities in connection with process management
  - Process creation and deletion
  - Process suspension and resumption
  - Provision of mechanisms for:
    - Process synchronization
    - Process communication
    - Deadlock handling

System Components – Main Memory Management

- Memory is a large array of words or bytes, each with its own address
- Main memory is a *volatile* storage device
- The operating system is responsible for the following activities in connection with memory management:
  - Keep track of which parts of memory are currently being used and by whom
  - Decide which processes to load when memory space becomes available
  - Allocate and deallocate memory space as needed

System Components – File Management

- A file is a collection of related information defined by its creator.
- The operating system is responsible for the following activities in connection with file management:
  - File/directory creation and deletion
  - Support of primitives for manipulating files and directories
  - Access control available on most systems
  - Mapping files onto secondary storage
  - File backup on stable (non-volatile) storage media

System Components – Secondary Storage Management

- Since main memory (*primary storage*) is volatile and too small to accommodate all data and programs permanently, the computer system must provide *secondary storage* to back up main memory.
- Most modern computer systems use drives as the principle storage medium, for both programs and data.
- The operating system is responsible for the following activities in connection with disk management:
  - Free space management
  - Storage allocation
  - Disk scheduling

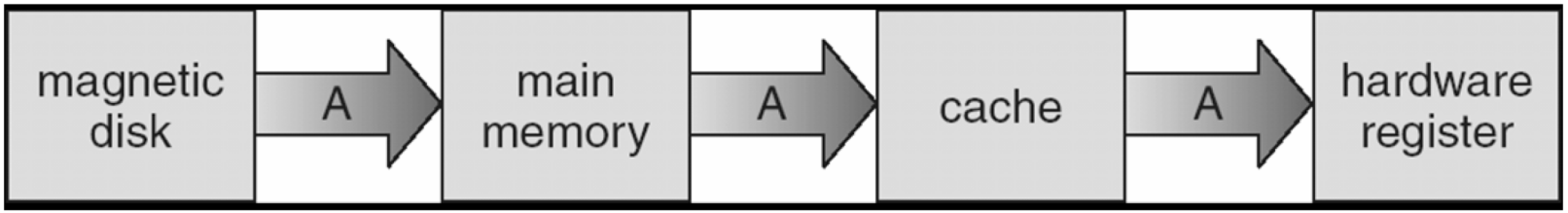
Performance of Various Levels of Storage

- 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