CH 1 Introduction

Ubiquity of Computing Devices




Complexity is Our Enemy

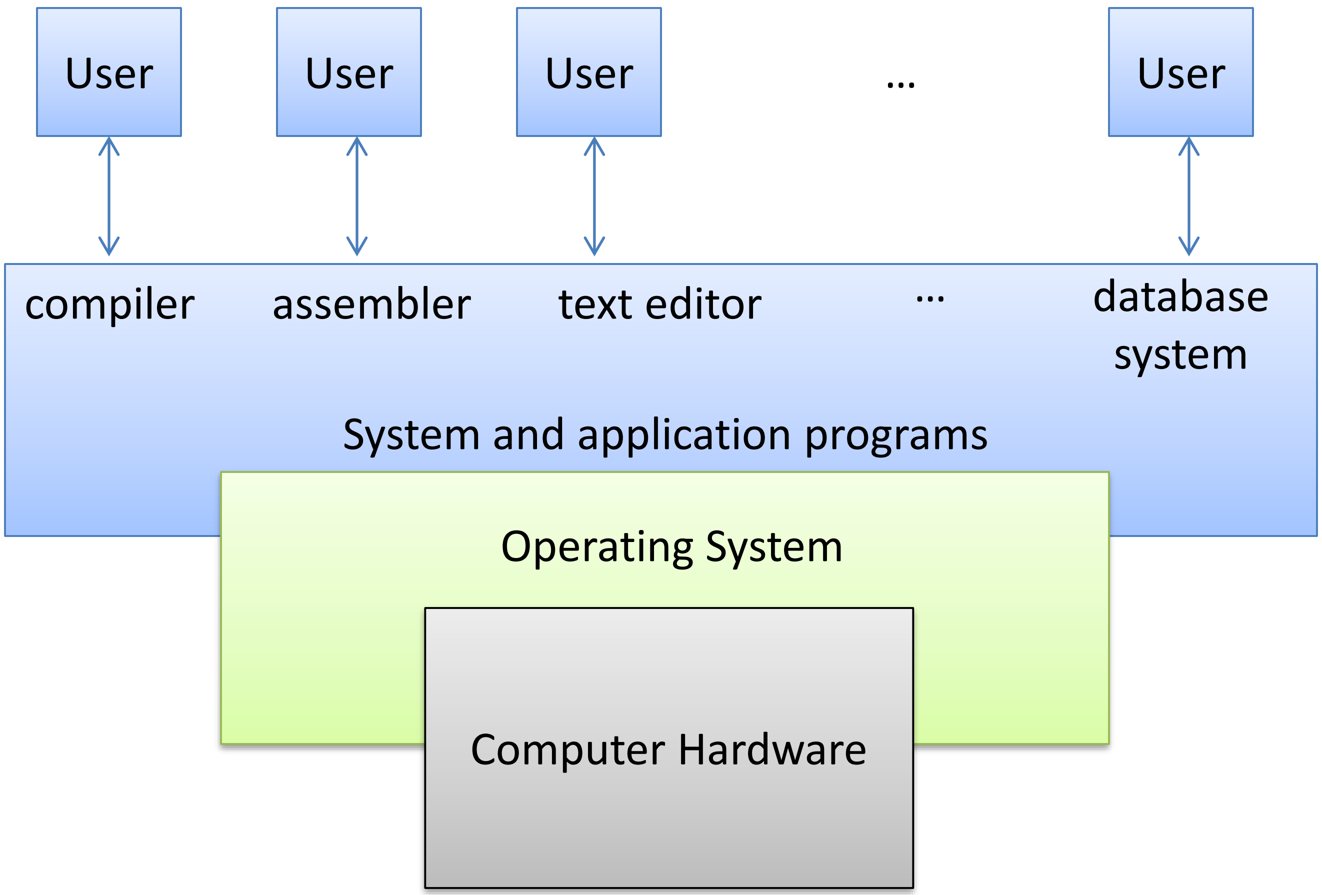
- Diversity of Hardware (and Standards)
  - CPU: ISA, cache size, clock speed
  - Memory size, type, disk size, type
  - I/O devices – keyboard, mouse, touchpad, camera, Bluetooth speaker, printer …
  - Network – wifi(6,5,802-11agnb …), wired, 5G
  - Interfaces: USB2,3,type-c, PCI, Bluetooth, zigbee
- Diversity of Software to run
  - Text editor, Calculator, game, web browser, scientific programs, database, router, image processing …
- Even worse: They grow and change rapidly.

What is an Operating System?

- A program that manages hardware.
- A program that acts as an intermediary between a user of a computer and the computer hardware.
- Operating System goals:
  - Execute user programs and make solving user problems easier.
  - Make the computer system convenient to use.
  - Use the computer hardware in an efficient manner.

Abstract View of Computer System

- Computer System Components
  - Hardware, Operating System, Applications, Users



Operating Systems View Points

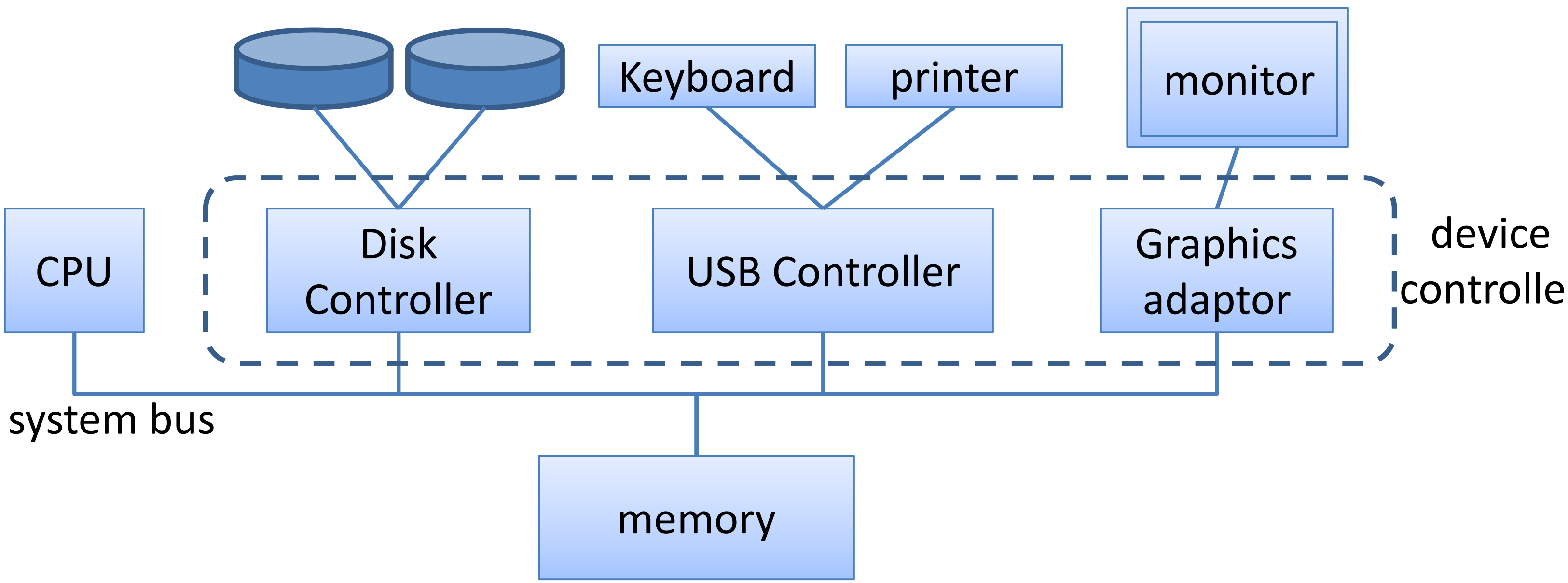
- User View
  - Home PC user
    - Primary goal is the ease of use
    - Resource utilization is of no concern
  - Mainframe/minicomputer
  - Workstation
  - Mobile devices
- System View
  - Resource allocator
    - Resource: CPU, memory, storage, I/O devices …
    - Handle conflicting requests
  - Control program
    - To prevent errors and improper use of computer

Definition of Operating System

- OS is loosely defined
  - No clear-cut definition
  - Why? Diverse H/W, various purpose
- “Everything a vendor ships”
- Entire package consisting of the central S/W that manages resources and the accompanying S/W tools
  - kernel + system programs
- Ex) Microsoft’s bundling of the browser

Computer System Organization

- CPU
  - Perform computation
- Memory
  - Programs and data
- I/O devices
  - Disk, monitor, printers
- System bus
  - Communication channel



- CPU & device controllers execute in parallel
- Memory controller synchronizes the access

Need for OS

- Several users are running programs.
  - They all want to use printer.
  - They all want more memory.
  - They want to peep into other's data.
- A programmer wants to write
  - A program that runs on any laptop.
  - A program that manages many files on the disk.
  - A program that communicate with other computer over the Internet.
- The owner wants to
  - Not waste CPU and memory.
  - Add new disk.
  - Switch from wired network to wifi.

Computer Operations

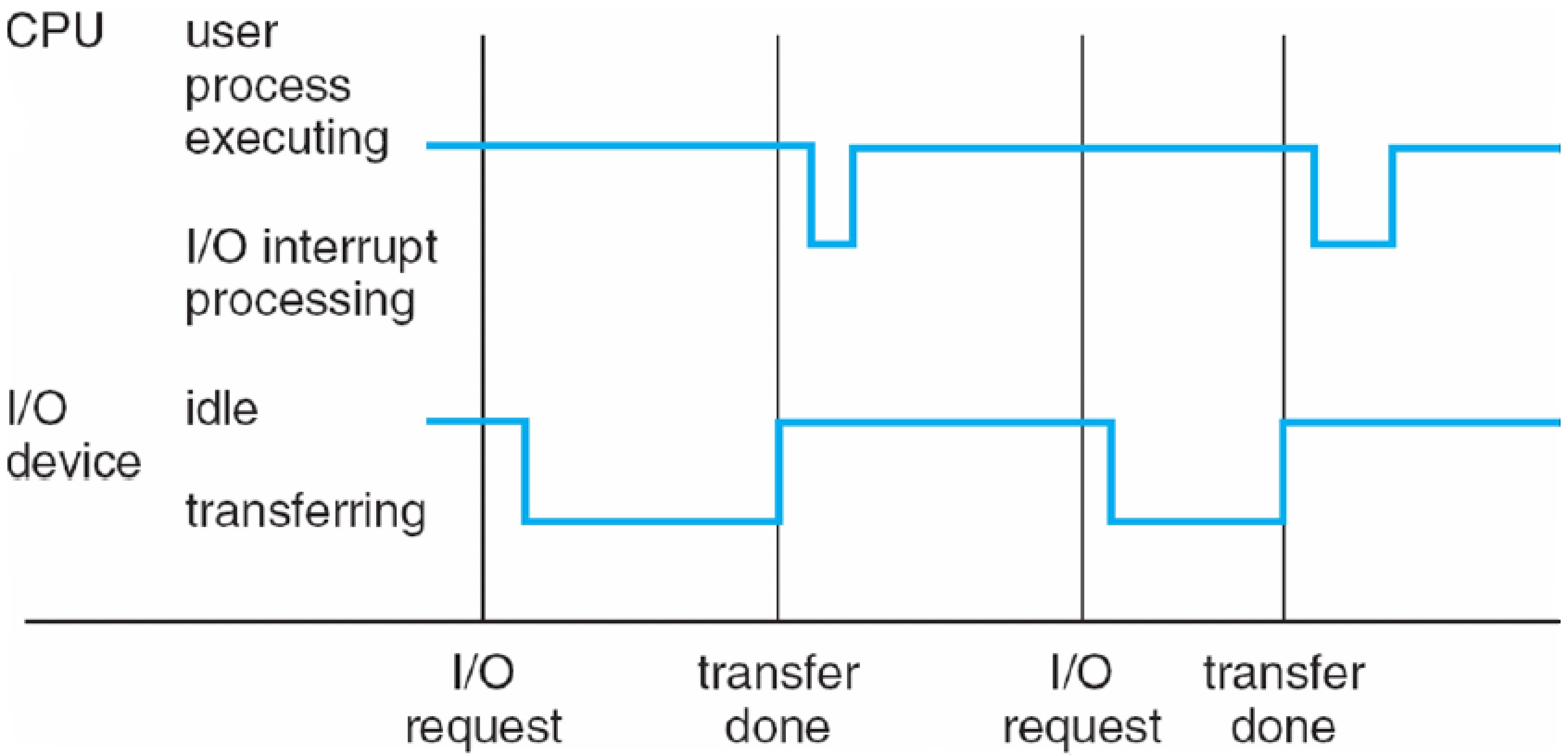
- Start-up
  - Bootstrap program: the first program that runs when computer is powered up
    - stored in ROM or EEPROM (firmware)
    - initializes CPU registers, device controllers, memory
    - loads OS kernel and starts it
  - System daemons
    - runs the entire time the kernel is running in the background
    - "init" is the first process, it starts many other deamons

Computer Operations

- Interrupt
  - An event that stops CPU to run the interrupt handler
  - H/W interrupts
    - timer, keyboard, mouse, DMA
    - delivered through system bus
  - S/W interrupts = system call
  - Interrupt handling is one of the key function of CPU
  - Interrupt vector: list of addresses for interrupt service routines
  - Interrupt handling must be quick
  - Need to save the address of the interrupted instruction, and save the register state

Interrupt Timeline

- While I/O is working, CPU can work on other jobs.
- I/O completion generates interrupts.



Storage Structure

- Programs must be loaded to memory to run
  - Memory: RAM (Random Access Memory)
  - load/store instruction
- Von Neumann architecture
  - Instruction and data are stored in the same memory for which there is a single link to the CPU
  - CPU can have multiple functional units
  - Memory access can be enhanced by use of cache
- Secondary storage
  - Usually it means HDD
  - Secondary storage is needed because main memory is volatile and too small

Storage Structure

- Storage device hierarchy
  - Cost-per-bit
  - Speed (access time)
  - Volatility

register
cache
main memory
solid-state disk (SSD)
magnetic disk
optical disk
magnetic tapes

- Non-volatile storage
  - Solid-state disks (SSD)
    - faster than HDD and non-volatile
  - flash memory
  - NVRAM: DRAM with battery backup power

I/O Structure

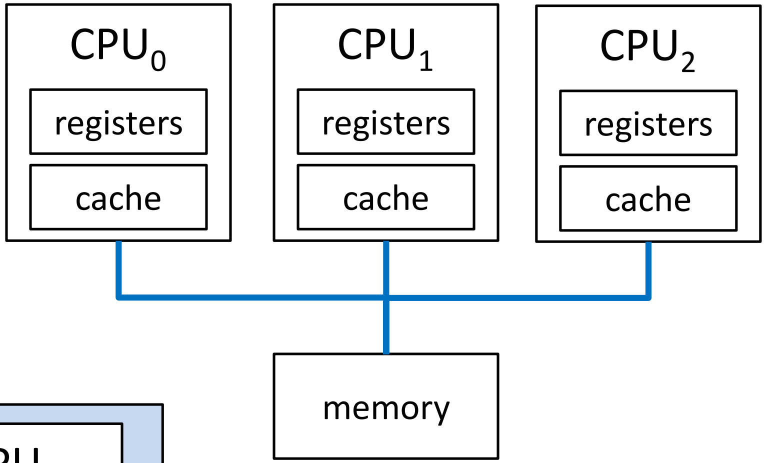
- Device controller
  - CPU and device controllers connected through a common bus
    - E.g. Small computer-systems interface (SCSI) controller
  - Local buffer, special-purpose registers
  - Responsible for moving data in/out
  - OS has device driver that knows how to interact with the device controller
- I/O operations
  1. Device driver loads the registers of device controller
  2. Device controller examine the register contents
  3. Device controller transfers data
  4. Device controller raises interrupt when done
  5. Device driver returns control to the OS
- DMA (Direct memory access)
  - CPU is released from the critical path

Computer System Architecture

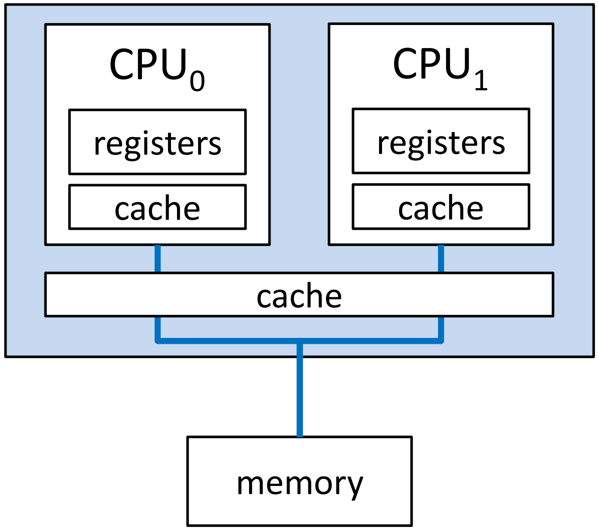
- Single-processor Systems
  - One general-purpose processor + many special-purpose processors
    - May be controlled by main CPU
    - In other case, built into H/W, autonomous
- Multiprocessor Systems
  - Why multi? Unable to improve the performance of single processor
  - Advantages of Multiprocessor System
    - Increased throughput, but *N* CPU != *N* times perf, why?
    - Economy of scale (from sharing)
    - Increased reliability (?)
      - graceful degradation, fault tolerant

Computer System Architecture

- Asymmetric multiprocessing
  - Master(boss) and slave
- Symmetric multiprocessing (SMP)
  - All processors are peers
  - All share physical memory
    - UMA vs. NUMA



- Multicore system

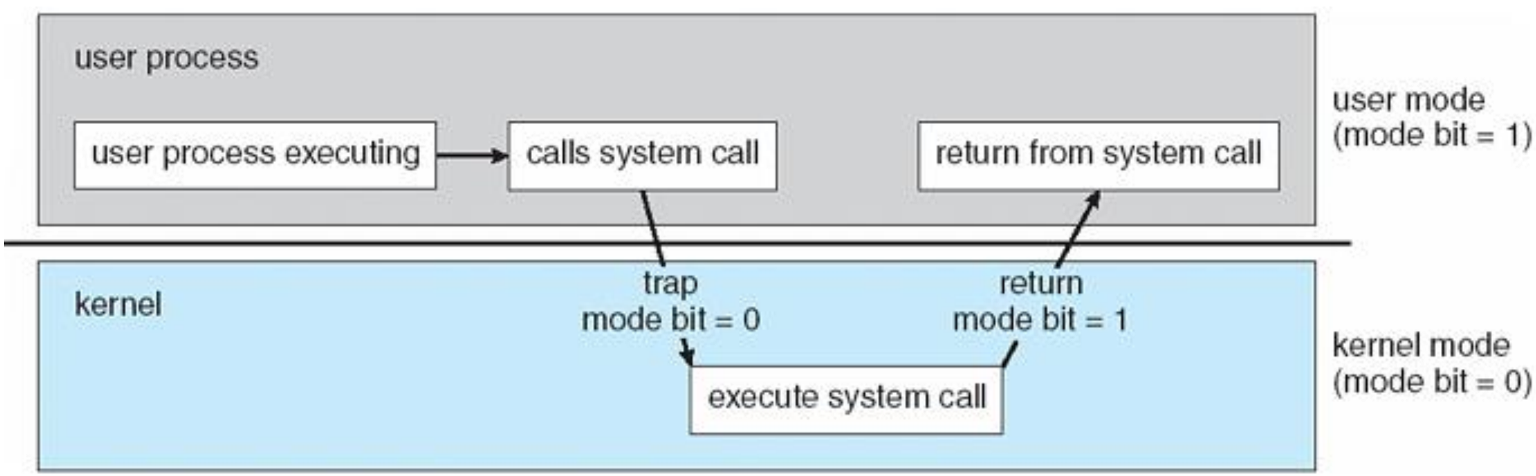


Operating System Structure

- OS provides:
  An environment within which programs are executed
- Key features of OS
  - Multi-programming: keep multiple jobs in memory
    - Why? To increase the CPU utilization
    - Job pool and job switching
  - Time sharing (multitasking)
    - Rapid switch between jobs → interactivity, short response time
    - Allow many users to share the computer
  - Process: A running instance of a program loaded into memory and in execution
    - Alternates between Computation and I/O
  - Job scheduling from job queue (pool), run queue → CH6
  - Running several programs require the Memory management → CH 8,9
    - Virtual memory (CH9): Physical memory vs. Logical memory

Operating System Operations

- OS is Interrupt driven
- Trap (exception)
  - Software interrupt
    - Division by zero, invalid address, Ctrl-c, Child process ended
  - Interrupt service routine
- Kernel mode vs. User mode
  - kernel mode = supervisor mode, system mode, privileged mode
    - CPU provides the mode bit to indicate the current mode
  - User process requests a service to kernel
    - Transition from user mode to kernel mode



Operating System Operations

- Privileged instructions
  - instructions that could harm the system when used carelessly
  - can execute them only in *kernel mode*
  - Attempting to run privileged inst. in user mode → **trap**
- System calls
  - Means for a program to request OS service
    - Read/write to file, network … etc.
  - invoked by special instruction (int or syscall)
  - trap to a specific location in the interrupt vector
  - System call is a software interrupt
  - Control → interrupt vector → interrupt service routine
    - mode bit set to kernel mode
  - Kernel examines what system call is invoked (read?write?send?recv?)
  - Kernel verifies the parameters, executes, returns.

Mode bits

- What if mode bit is not supported in CPU?
  - Application can damage the operating system
  - Multiple applications write to a device at the same time …
- What to do in case of the mode violation?
  - Violations
    - Attempt to execute privileged instruction
    - Attempt to access forbidden memory address
  - Hardware traps to OS
  - Operating system handles such errors
    - usually terminate the process
    - memory is dumped

Process Management

- Process
  - An instance of an executing program
  - Needs resources
    - Kernel loads the program into virtual memory, allocate spaces for variables, and setup metadata for pid, user ID, termination status, group ID … etc.
  - Execution of a process
    - One **program counter** per one thread of a process
      - program counter: next instruction to execute
    - Execution is sequential, one instruction at a time
    - Two process from the same program
      - Two separate execution sequences
- OS responsibilities
  - Scheduling of processes and threads
  - Create/delete/suspend/resume processes
  - provide mechanisms for process synchronization
  - provide mechanisms for process communication

Memory Management

- Memory: large array of bytes, each addressable
- CPU reads instruction and data from main memory through single channel → von Neumann architecture
  - Main memory is the only storage CPU can directly access
- Program is loaded into memory and generates memory address to access data and instructions
- OS needs to keep several programs in memory to increase the memory (as well as CPU) utilization
  - memory management is needed
- Memory management algorithm requires hardware support
- OS responsibilities (in memory management)
  - Keep track of which part of memory is in use or not
  - Decide which process/data to move in/out of memory
  - Allocate/deallocate memory

Storage Management (1/3)

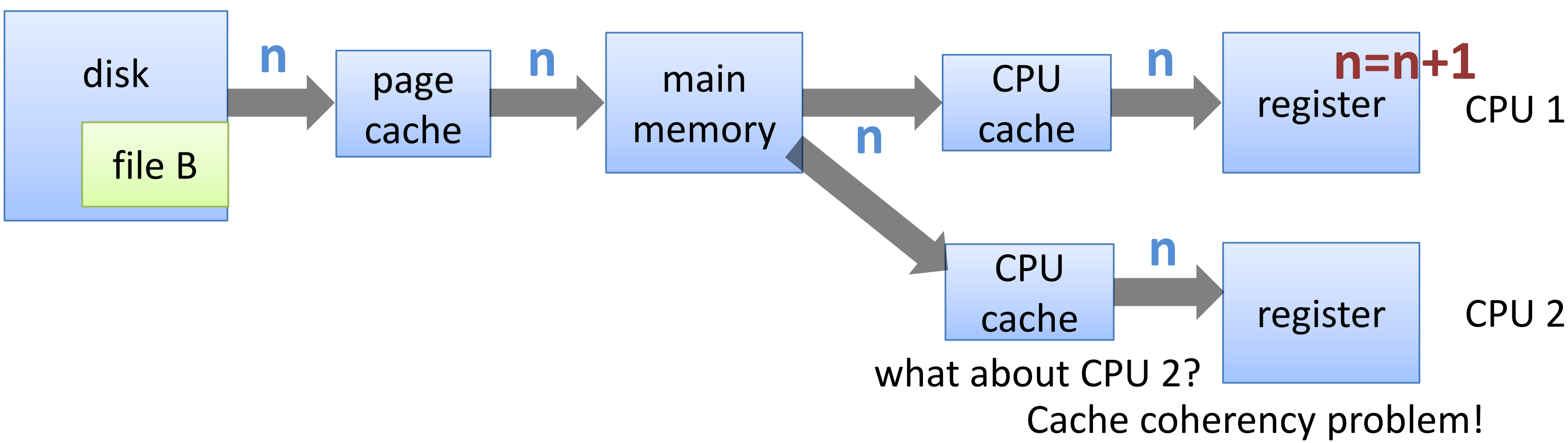
- File system management
  - file: a collection of related information defined by user
  - file may contain program or data, in free format, in any type
  - files are organized using directory structure
    - directory itself is a file
  - file accesses are controlled using permission
- OS responsibilities
  - Create/delete directories to organize files
  - Supporting primitives for manipulating files and directories
  - Mapping files onto secondary storage
    - file system: how to map storage space to files
  - Backing up files on persistent storage media
    - persist files

Storage Management (2/3)

- Cache
  - Faster storage in front of a slow storage device to speed up the access
  - Smaller and more expensive than the slow storage device
  - Transparency
- When accessing a data
  - Check if data exists in the cache
    - if yes, quickly return the data
    - if no, access the slow storage and copy it to the cache and return the data
  - locality assumption must hold
- Caching applies to different levels
  - Between registers and RAM → controlled by H/W
  - Between RAM and Disk → controlled by OS

Storage Management (3/3)

- Cache management
  - cache size selection and replacement policy
  - results in big performance difference
- Cache controlled by H/W and S/W
  - Instruction cache and data cache in CPU: controlled by H/W
    - Hardware cache is not discussed in OS course
  - Data transfer from disk to memory: controlled by S/W (i.e. OS)
- Replication of data due to caching



I/O Systems

- OS hides details of hardware devices and provide uniform interface to the application
- I/O subsystem consists of
  - Memory management component for buffering, caching and spooling
  - A general device-driver interface
  - Device drivers