CSE 3320
Operating Systems
Computer and Operating Systems Overview

Jia Rao

Department of Computer Science and Engineering

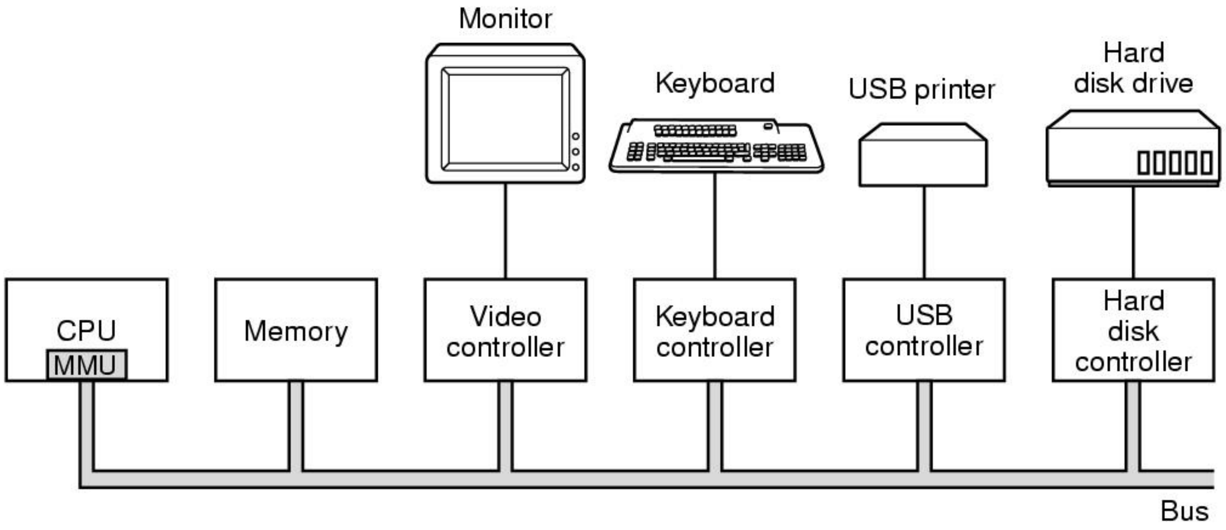
http://ranger.uta.edu/~jrao

Overview

- Recap of last class
  - What is an operating system ?
  - Functionalities of operating systems
  - Types of operating systems
- Computer hardware review
- Operating system organization

Computer Hardware Review

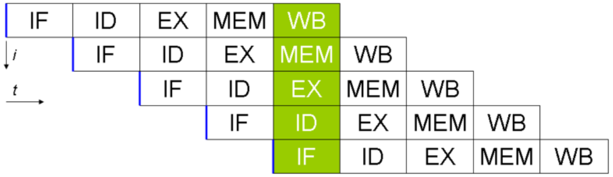
- Basic components of a simple personal computer



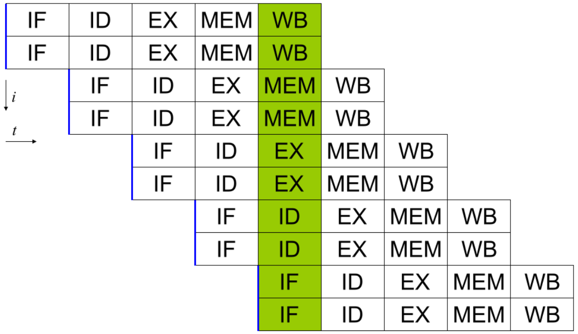
- CPU: data processing
- Memory: volatile data storage
- Disk: persistent data storage
- NIC: inter-machine communication
- Bus: intra-machine communication

Central Processing Unit (CPU)

- Components
  - Arithmetic Logic Unit (ALU)
  - Control Unit (CU)
- Clock rate
  - The speed at which a CPU is running
- Data storage
  - General-purpose registers: EAX, EBX …
  - Special-purpose registers: PC (EIP), SP, IR …
- Parallelism
  - Instruction-level parallelism
  - Thread-level parallelism
    - Hyper-threading: duplicate units that store architectural states
    - Replicated: registers. Partitioned: ROB, load buffer… Shared: reservation station, caches



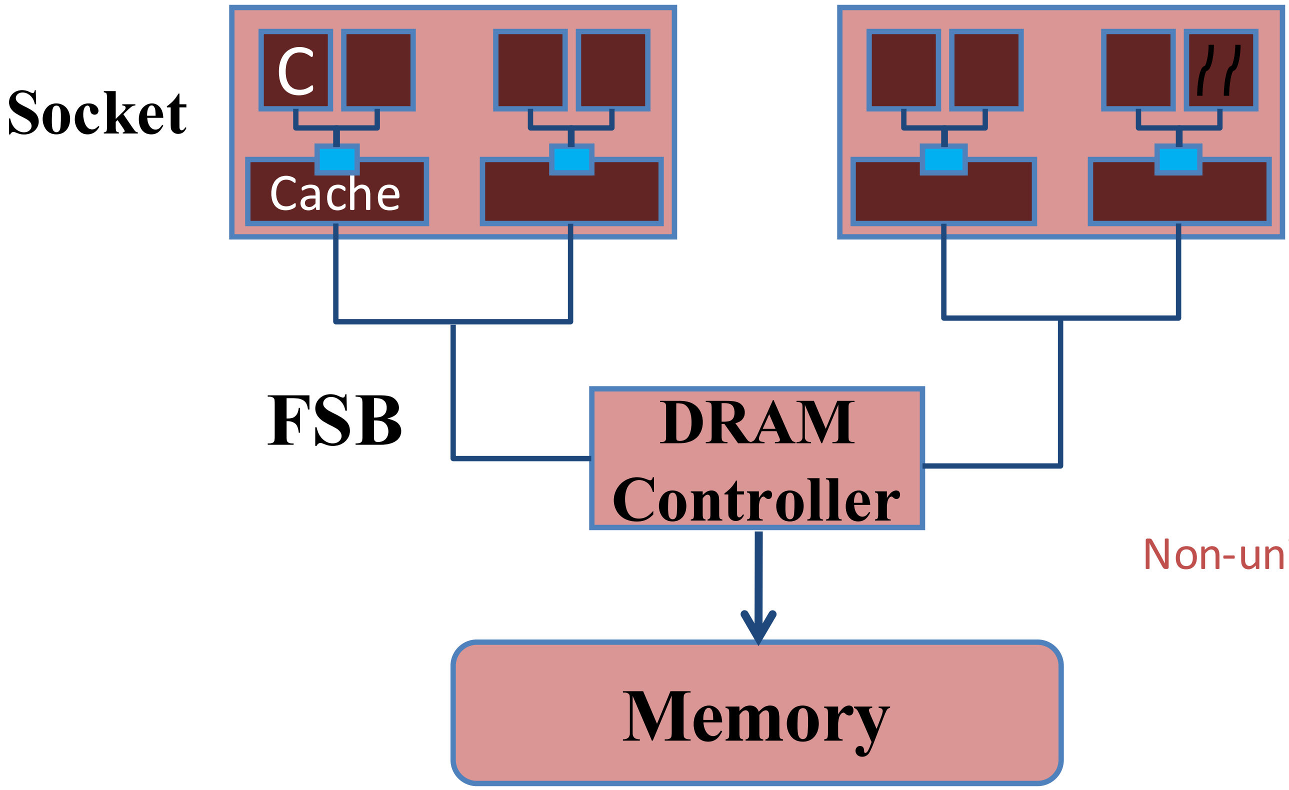
IF	ID	EX	MEM	WB				
	IF	ID	EX	MEM	WB			
		IF	ID	EX	MEM	WB		
			IF	ID	EX	MEM	WB	
				IF	ID	EX	MEM	WB



IF	ID	EX	MEM	WB				
IF	ID	EX	MEM	WB				
	IF	ID	EX	MEM	WB			
	IF	ID	EX	MEM	WB			
		IF	ID	EX	MEM	WB		
		IF	ID	EX	MEM	WB		
			IF	ID	EX	MEM	WB	
			IF	ID	EX	MEM	WB	
				IF	ID	EX	MEM	WB
				IF	ID	EX	MEM	WB

Multi-Core Processors

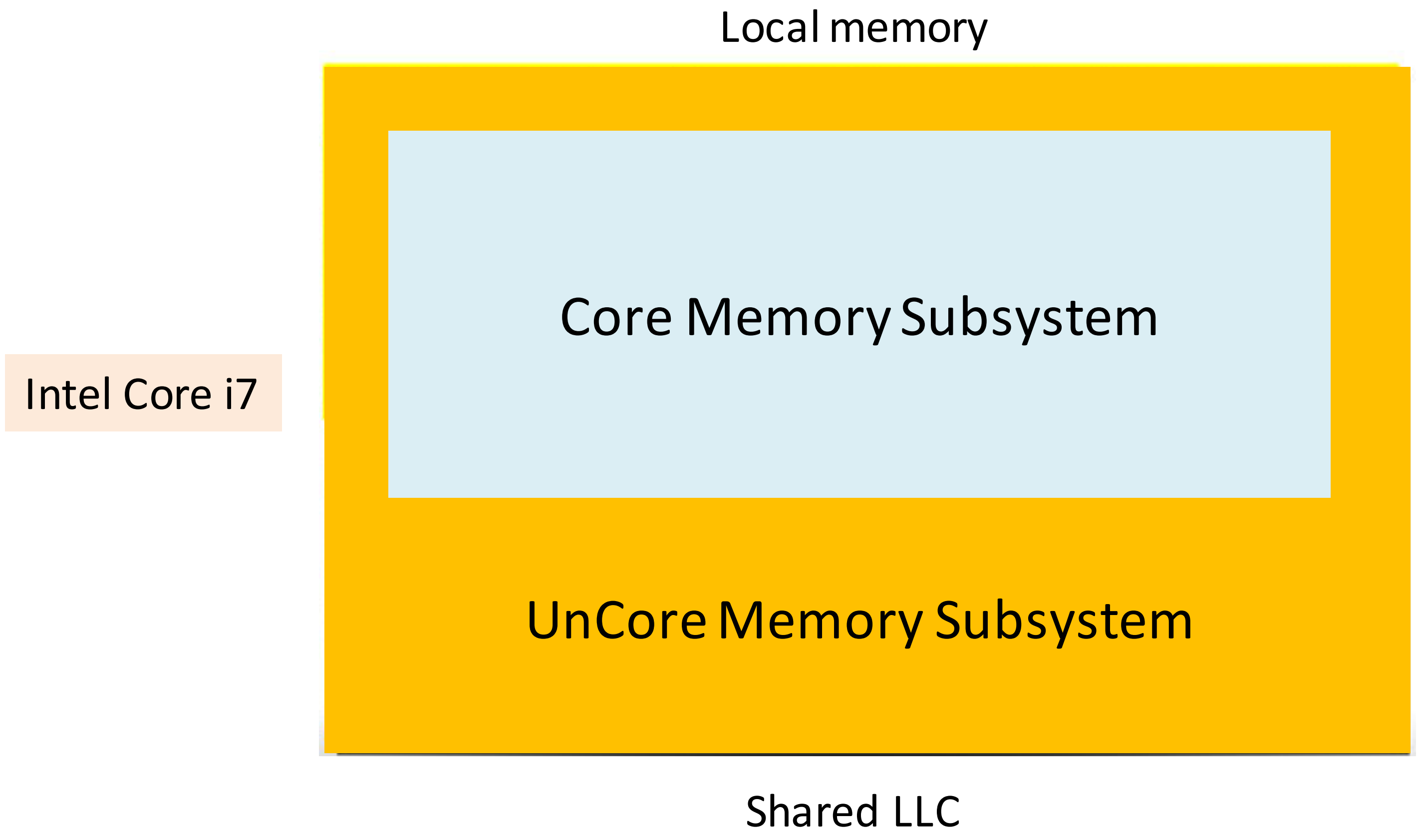
- Multiple CPUs on a single chip



Non-uniform performance

A schematic view of Intel Core 2

Multi-Core Processors: NUMA



Local memory

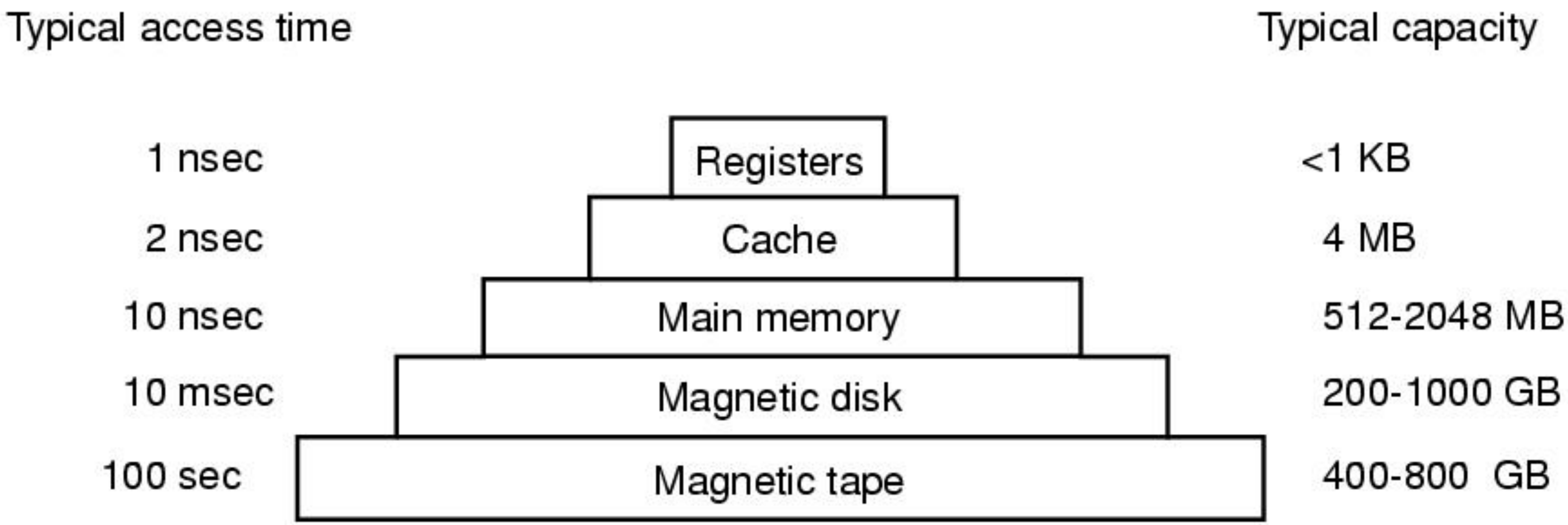
Intel Core i7

Core Memory Subsystem

UnCore Memory Subsystem

Shared LLC

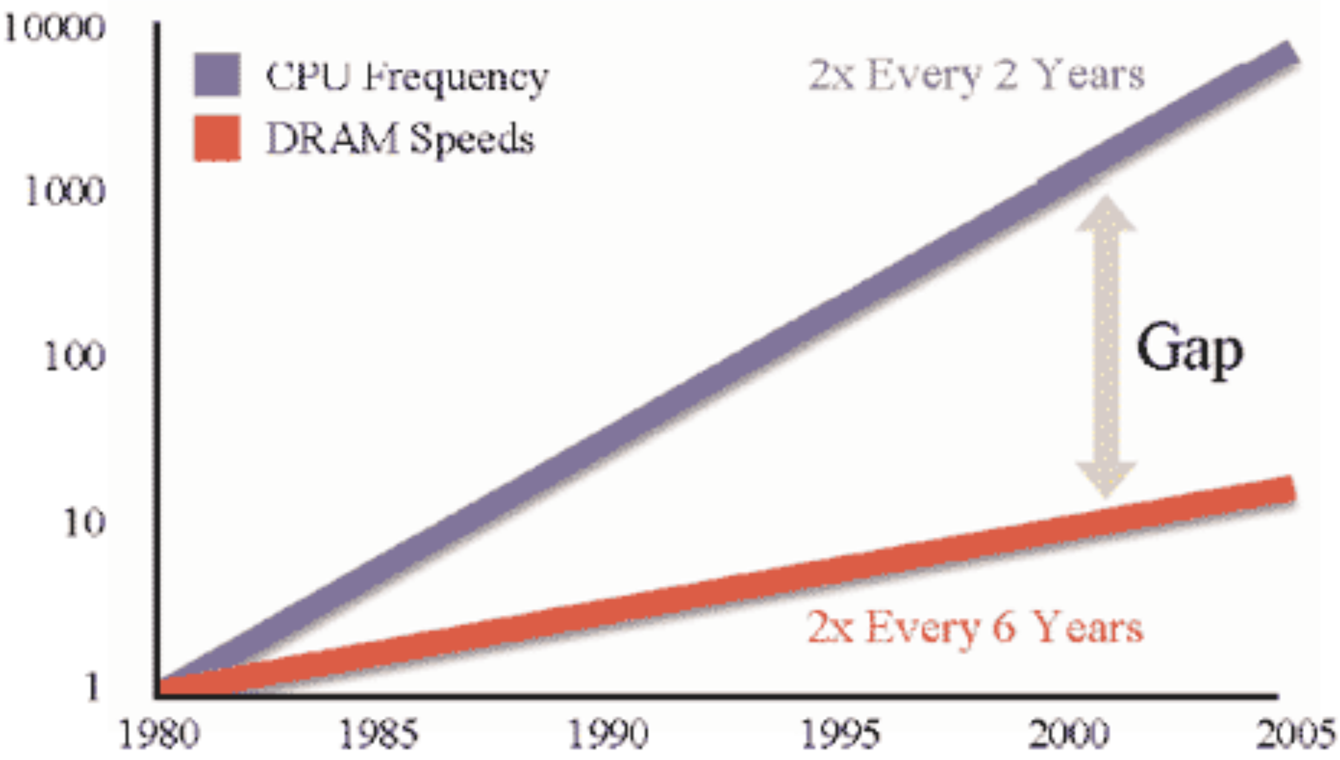
Memory



Typical access time		Typical capacity
1 nsec	Registers	<1 KB
2 nsec	Cache	4 MB
10 nsec	Main memory	512-2048 MB
10 msec	Magnetic disk	200-1000 GB
100 sec	Magnetic tape	400-800 GB

A typical memory hierarchy

Why Cache is important ?

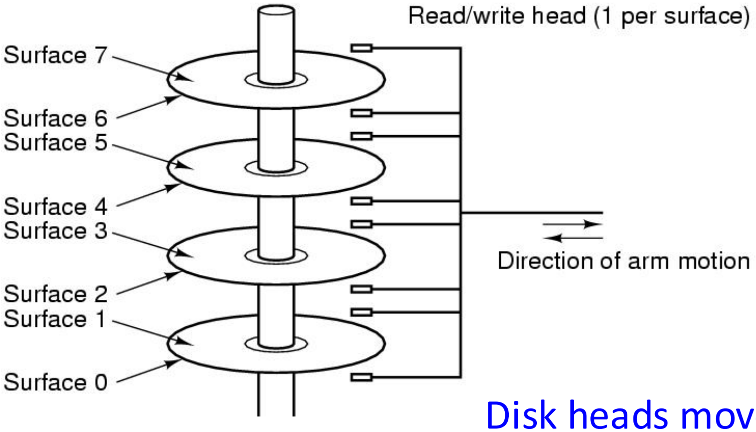


- A larger size than registers
- A much faster speed than memory
- Concurrent accesses to memory when cache misses occur

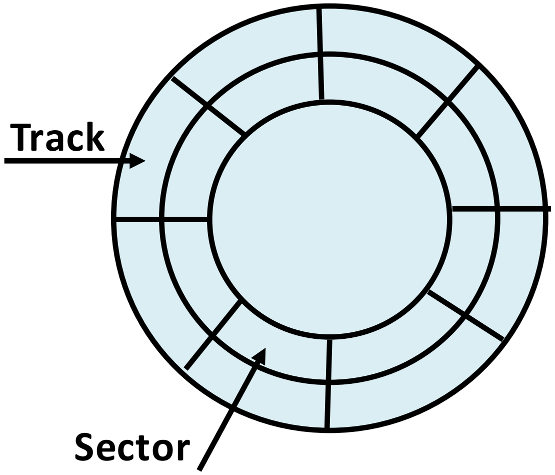
More on CPU Cache

- Cache management
  - When to put a new item into the cache.
  - Which cache line to put the new item in.
  - Which item to remove from the cache when a slot is needed.
  - Where to put a newly evicted item in the larger memory.
  - When to write dirty item back to memory

Hard Disks



Disk heads move together



- A stack of platters, a surface with a magnetic coating
- Typical numbers (depending on the disk size):
  - 500 to 2,000 tracks per surface
  - 32 to 128 sectors per track
    - A sector is the smallest unit that can be read or written
- Originally, all tracks have the same number of sectors:
  - “Constant” bit density: record more sectors on the outer tracks

Magnetic Disk Characteristics

- Disk head: each side of a platter has separate disk head
- Read/write data is a three-stage process:
  - Seek time: position the arm over the proper track
  - Rotational latency: wait for the desired sector to rotate under the read/write head
  - Transfer time: transfer a block of bits (sector) under the read-write head

- Long seek time
- Only one request at a time
- Throughput is dependent on data size

- Average seek time as reported by the industry:
  - Typically in the range of 8 ms to 15 ms
  - (Sum of the time for all possible seek) / (total # of possible seeks)
- Due to locality of disk reference
  - Actual average seek time may only be 25% to 33% of the advertised number

CPU v.s. Hard Disks

- Similarity
  - Time-sharing
- Differences (execution vehicle v.s. storage device)
  - CPU
    - Cache reuse -> temporal locality
    - Easy to switch between sharing parties -> fine grain, overhead sensitive
    - Usually multiple CPUs-> load balancing
    - Multiple execution modes -> energy saving
  - Hard disks
    - Almost no data reuse, but faster to read adjacent data -> spatial locality
    - Expensive to switch between sharing parties -> coarse grain
    - Striping or replication required if using multiple disks -> coordination

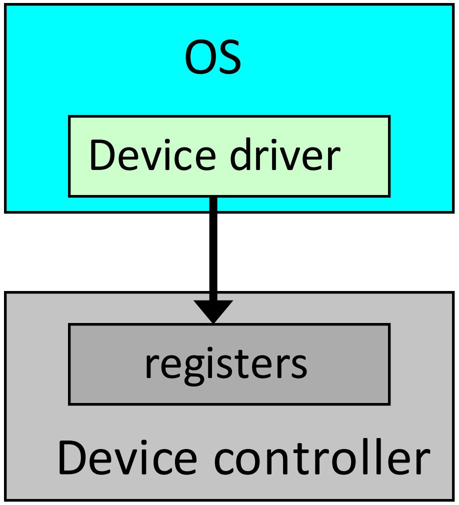
Memory Management

- Multiprogramming
  - How to *protect* the programs from one another and the kernel from them all?
  - How to handle *relocation* ?

Virtual memory space/address → Physical memory space/address

I/O Devices

- Device controller
  - To provide a simple interface of device control to OS
- Device driver
  - The software that talks to a controller, giving it commands and accepting responses



Interactions between OS and I/O Devices

- The OS gives commands to the I/O devices
- The I/O device notifies the OS when the I/O device has completed an operation or has encountered an error
- Data is transferred between memory and an I/O device

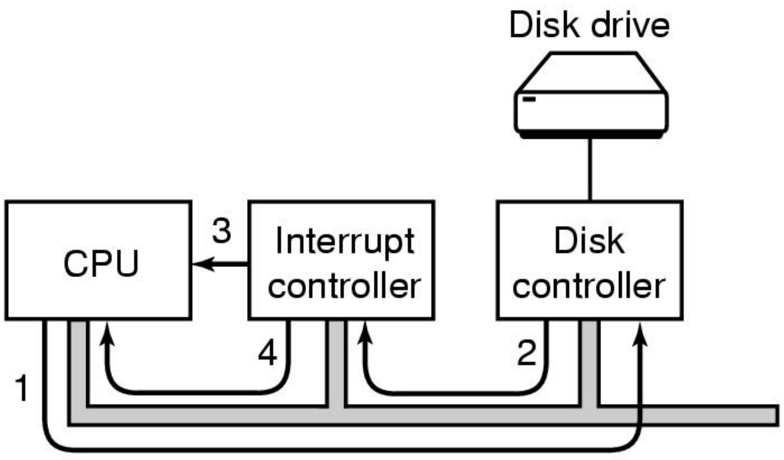
How I/O Devices Notify the OS ?

- Polling
  - The I/O device put information in a status register
  - The OS periodically check the status register
- Interrupt
  - Whenever an I/O device needs attention from the processor, it interrupts the processor from what it is currently doing
- DMA
  - Delegate I/O responsibility from CPU

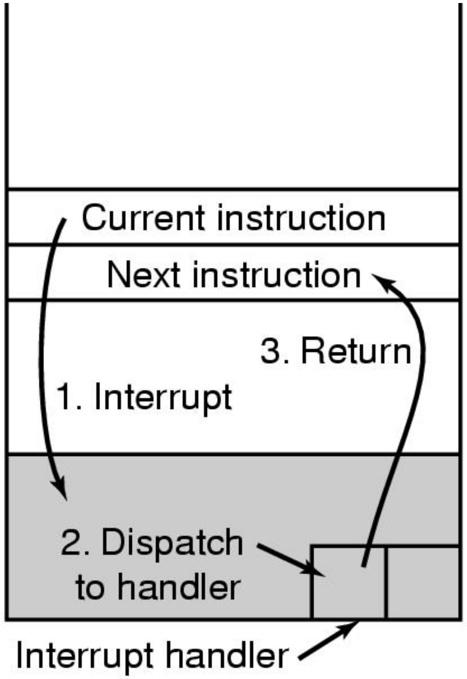
Interrupts

- Interrupts
  - An interruption of the normal sequence of execution
  - Improves processing efficiency
  - Allows the processor to execute other instructions while an I/O operation is in progress
  - A suspension of a process caused by an event external to that process and performed in such a way that the process can be resumed
- Types of interrupts
  - I/O
  - Program (exception)
    - arithmetic overflow
    - division by zero
    - reference outside user's memory space
  - Timer, Hardware failure

I/O Interrupt



(a)



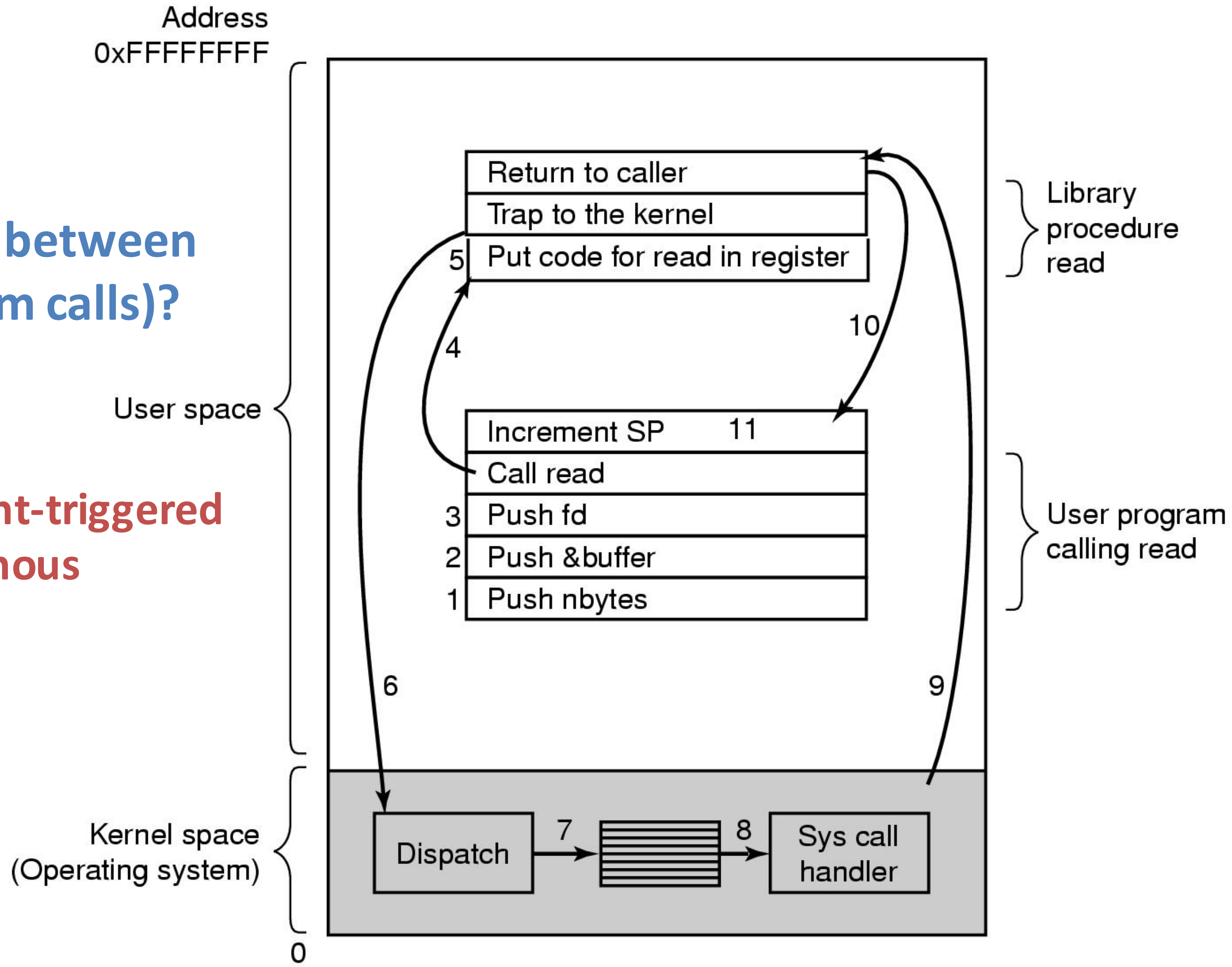
(b)

1. CPU writes cmds in to device registers
2. The device signals interrupt controller
3. Interrupt controller informs a CPU
4. The CPU accepts the interrupt and triggers the service routine

System Calls

What is the key difference between interrupts and traps (system calls)?

- **program-triggered vs. event-triggered**
- **synchronous vs. asynchronous**



There are 11 steps in making the system call read (fd, buffer, nbytes)

Operating System Components

- Process management
- Memory management
- File and storage
- Networking

Process Management

- Process: a fundamental OS concept
  - Memory address space
  - Some set of registers
  - Protection domain
  - Resource allocation unit
- OS responsibilities for process management
  - Process creation and deletion
  - Process scheduling, suspension, and resumption
  - Inter-process communication and synchronization

Memory Management

- Memory
  - A large array of addressable words and bytes
- OS responsibilities for memory management
  - Allocate and de-allocate memory space
  - Keep memory space efficiently utilized
  - Keep track of which part of memory are used and by whom

Design goals: *transparency* and *efficiency*

File and Storage Management

- A file is a collection of data (usually) stored on disk with a unique name
  - Programs
  - Data
  - Devices (UNIX & Linux)
- OS responsibilities for file management
  - Organize directories and files
  - Map files onto disk
- OS responsibilities for disk management
  - Disk space management
  - Disk scheduling

Summary

- Computer hardware
  - Time-sharing: CPU, disk
  - Space-sharing: memory, disk
- OS components
  - Process management
  - Memory management
  - File and storage management
- Additional readings and practice
  - Section 1.6 and try the following Linux commands
    - Who, uname, ls, cat, cp, rm, mv, cd, mkdir, touch, chmod
    - Use “man” to see the manual of above commends
  - Write an C program with an output to the screen
    - Strace –o trace.txt ./YOUR_PROG
    - See the system calls triggered (execve, write, …)
    - http://unix.stackexchange.com/questions/797/understanding-the-linux-kernel-source
  - Check the VMware tutorial on course website