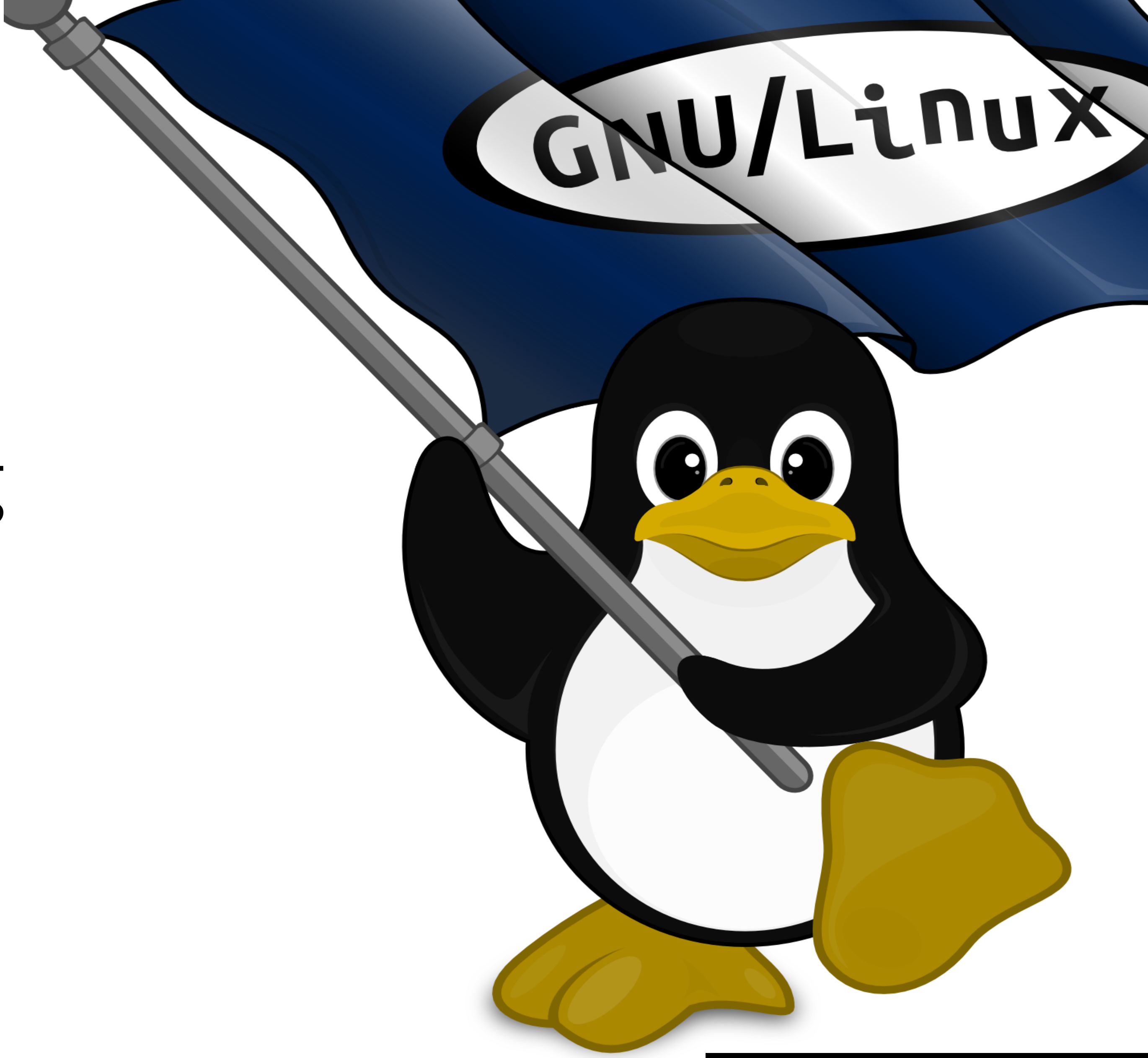
CS4500
Operating Systems

Week 2

Course Content

Unit	Module	Week	Topic
1 – Introduction	1 & 2	1	Welcome to OS & Introduction
	2	2	OS Overview
2 – Processes and Threads	3	3	Processes
	4	4	Threads
	5	5	CPU Scheduling
	6	6	Multiprocessor Scheduling
	7	7	Interprocess Communication
	8	8	Pthread
3 – Deadlocks & Memory	9	9	Deadlock
	10	10	Memory Management
	--	11	Midterm Exam
4 – Additional Topics	11	11	Page Replacement
	12	13	File Systems
	13	14	IO Devices
	14	15	Security
	--	16	Final Exam

Operating Systems Overview



What to expect today:

Overview of Operating Systems

Assigned reading

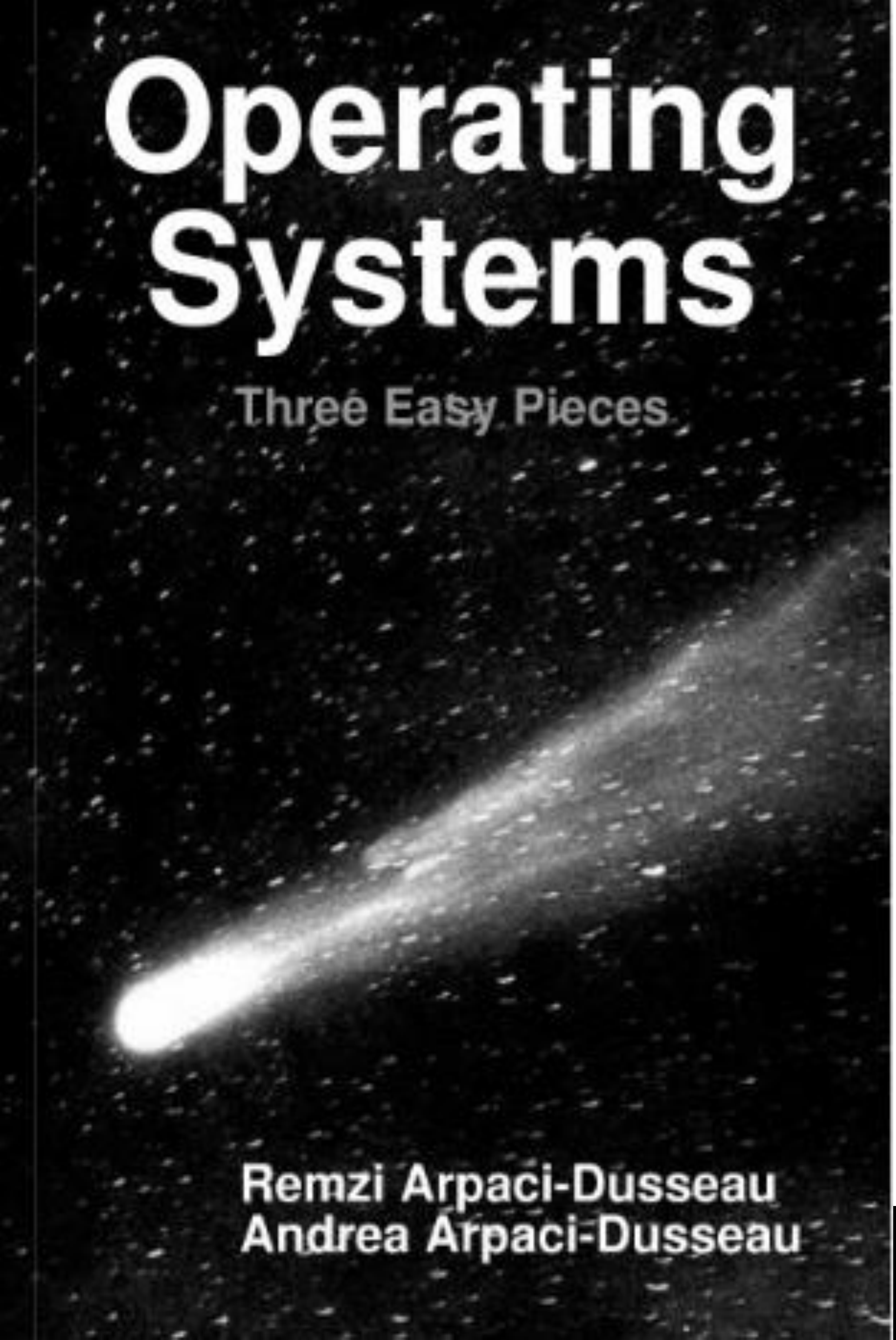
- Andrew S. Tanenbaum, Modern Operating Systems, 5th edition, 2022, Prentice Hall
- Chapter 1

Homework 1

Lecture Resources

Andrew S. Tanenbaum, Modern Operating Systems, 5th edition, 2022, Prentice Hall

Peter Jay Salzman, Michael Burian, Ori Pomerantz, Bob Mottram, Jim Huang. The Linux Kernel Module Programming Guide, 2022. https://sysprog21.github.io/lkmpg/



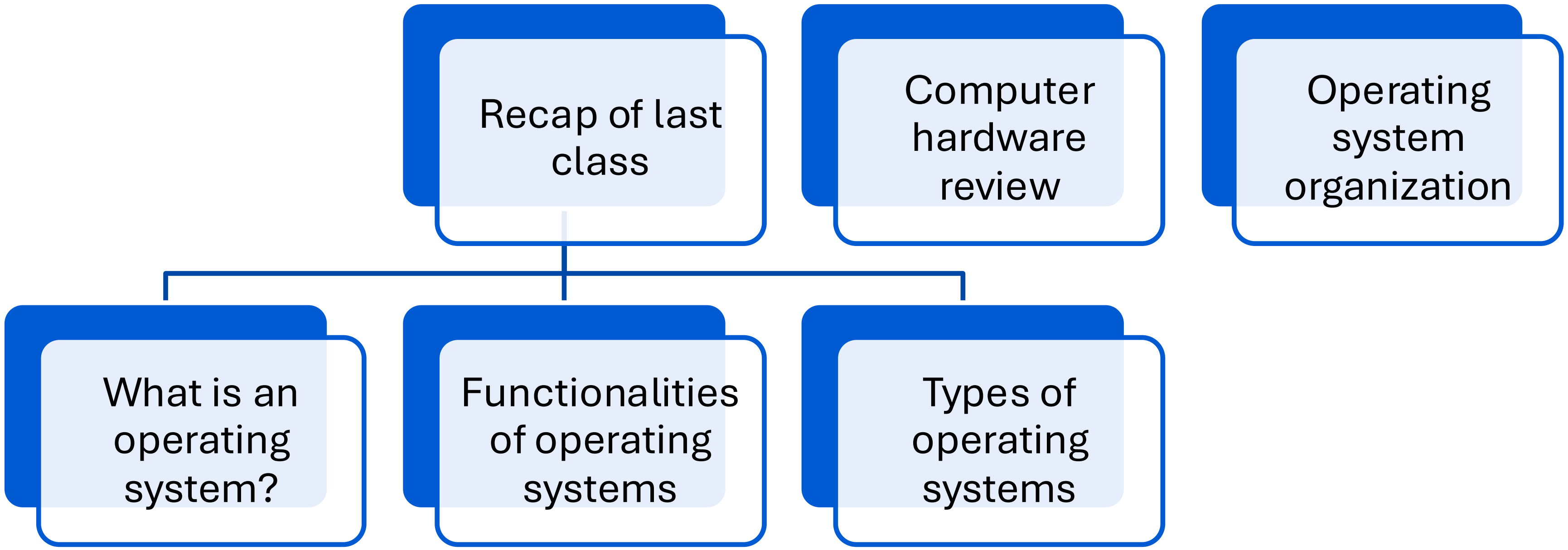
Additional Resource

Remzi H. Arpaci-Dusseau, R. H., Arpaci-Dusseau, A.C. *Operating systems: Three easy pieces: Chapter 10 and 28.* (2018). Arpaci-Dusseau Books.

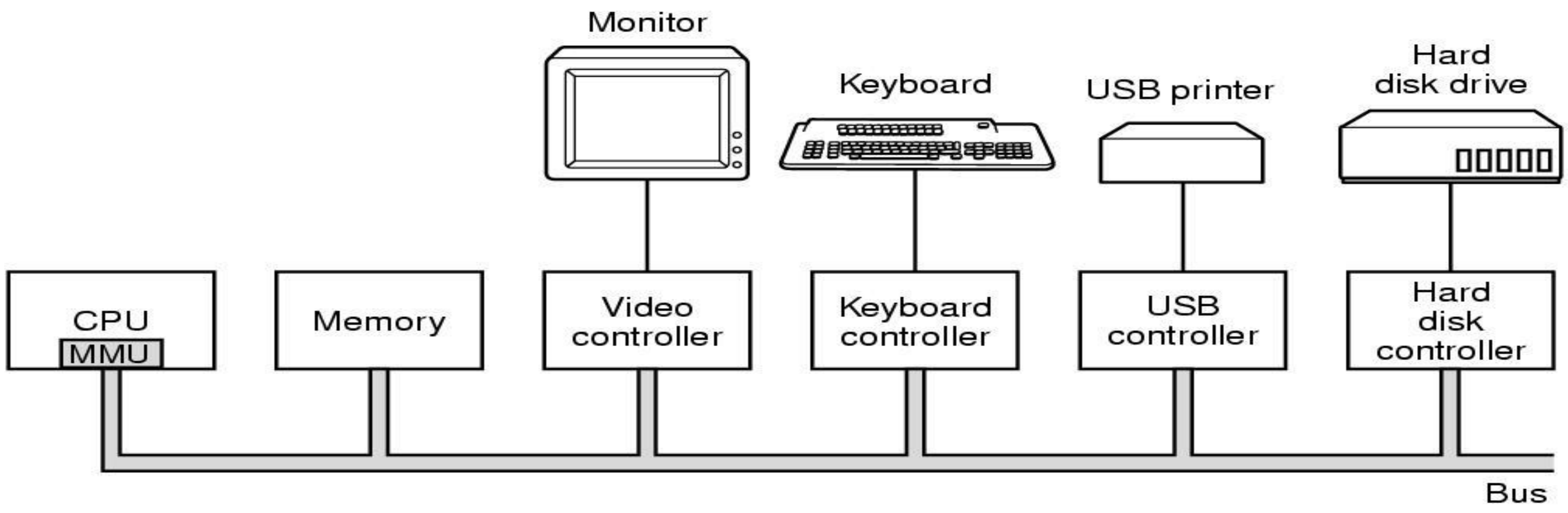
https://pages.cs.wisc.edu/~remzi/OSTEP/cpu-sched-multi.pdf

Copy of .pdf available in Canvas

University of Colorado
Boulder | Colorado Springs | Denver | Anschutz Medical Campus


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

Computer Hardware Review



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

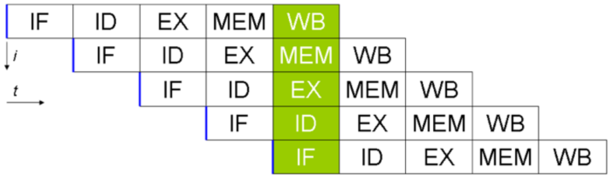
Central Processing Unit (CPU)

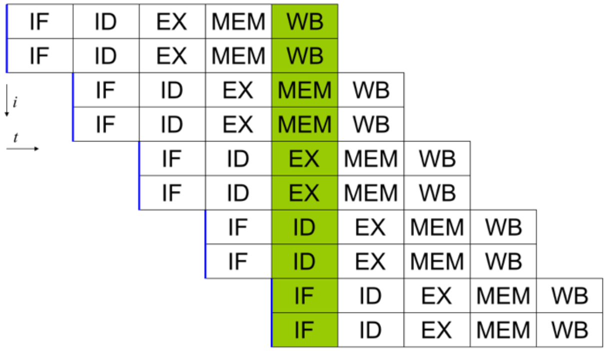
Components

- Arithmetic Logic Unit (ALU)
- Control Unit (CU)
- Fetch, decode, execute

Data storage: mem access slow

- General-purpose registers: EAX, EBX …
- Special-purpose registers: PC, SP, …

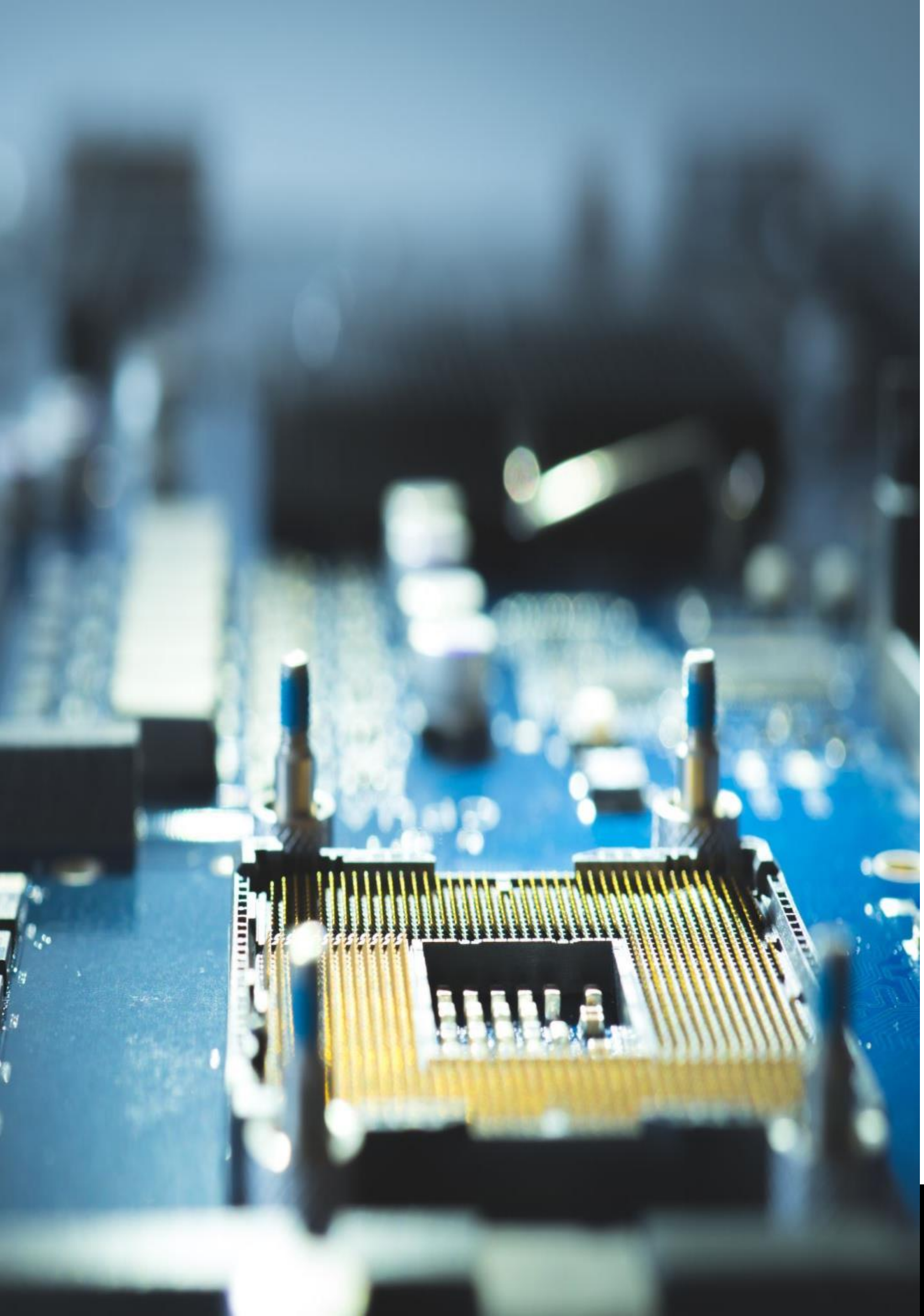




Multithreading

Each thread appears to the operating system as a separate CPU

- Ex: A system with two actual CPUs, each with two threads --> The OS sees four CPUs



Multicore

- Many CPU chips now have four, eight, or more complete processors or cores on them
- Multicore chips carry four minichips, each with its own independent CPU
- Some CPUs with core counts in the hundreds
- Multicore chips require a multiprocessor operating system

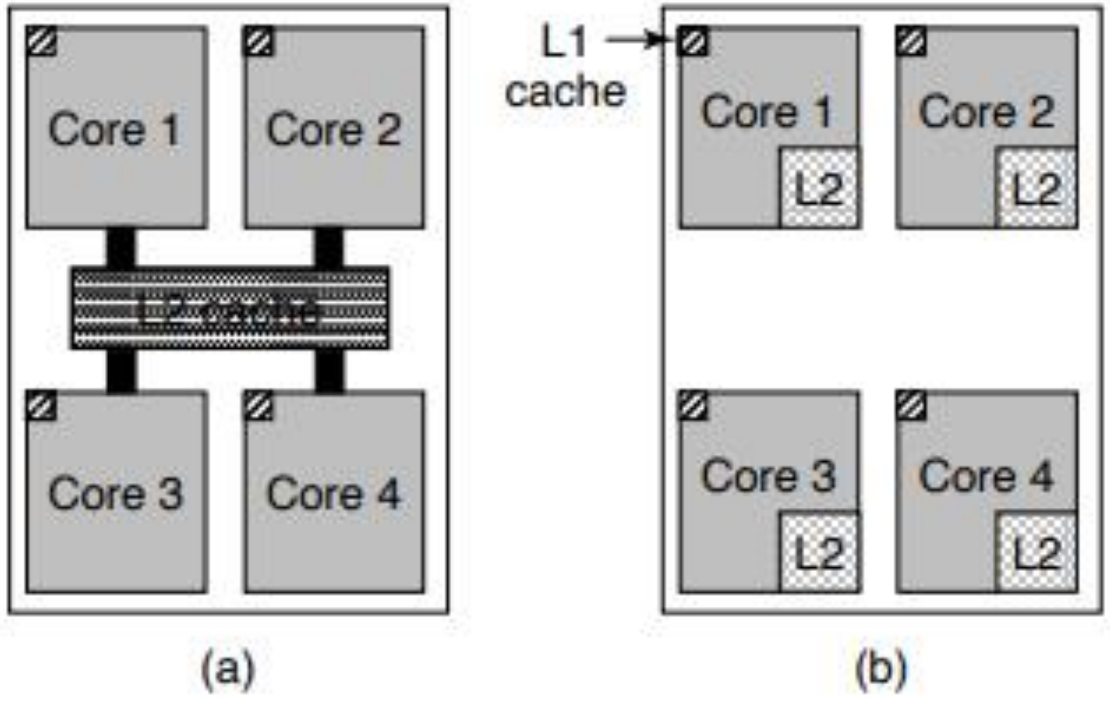


Figure 1-8. (a) A quad-core chip with a shared L2 cache. (b) A quad-core chip with separate L2 caches.

General Purpose Registers

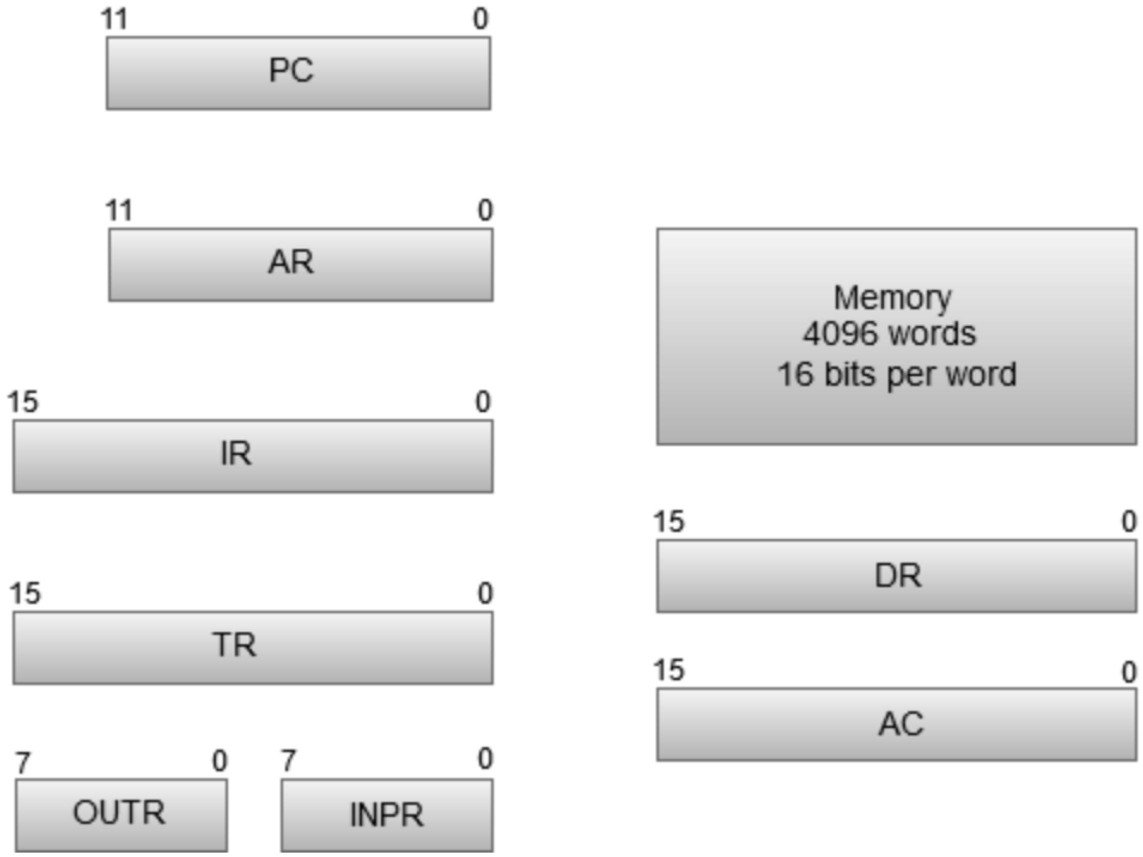
- EAX, EDX, ECX, EBX, EBP, EDI, and ESI
- 32-bit general-purpose registers, used for temporary data storage and memory access
- History of EAX blog post: https://keleshev.com/eax-x86-register-meaning-and-history/

Special Purpose Registers

- Data Register (DR) contains 16 bits
  - Holds the operand read from the memory location
- Memory Address Register (MAR) contains 12 bits
  - Holds the address for the memory location
- Program Counter (PC) contains 12 bits
  - Holds the address of the next instruction to be read from memory
- Instruction register (IR) holds instruction read from memory
- Temporary Register (TR) holds temporary data during the processing
- Input Registers (IR) holds the input characters from the user
- Output Registers (OR) holds output

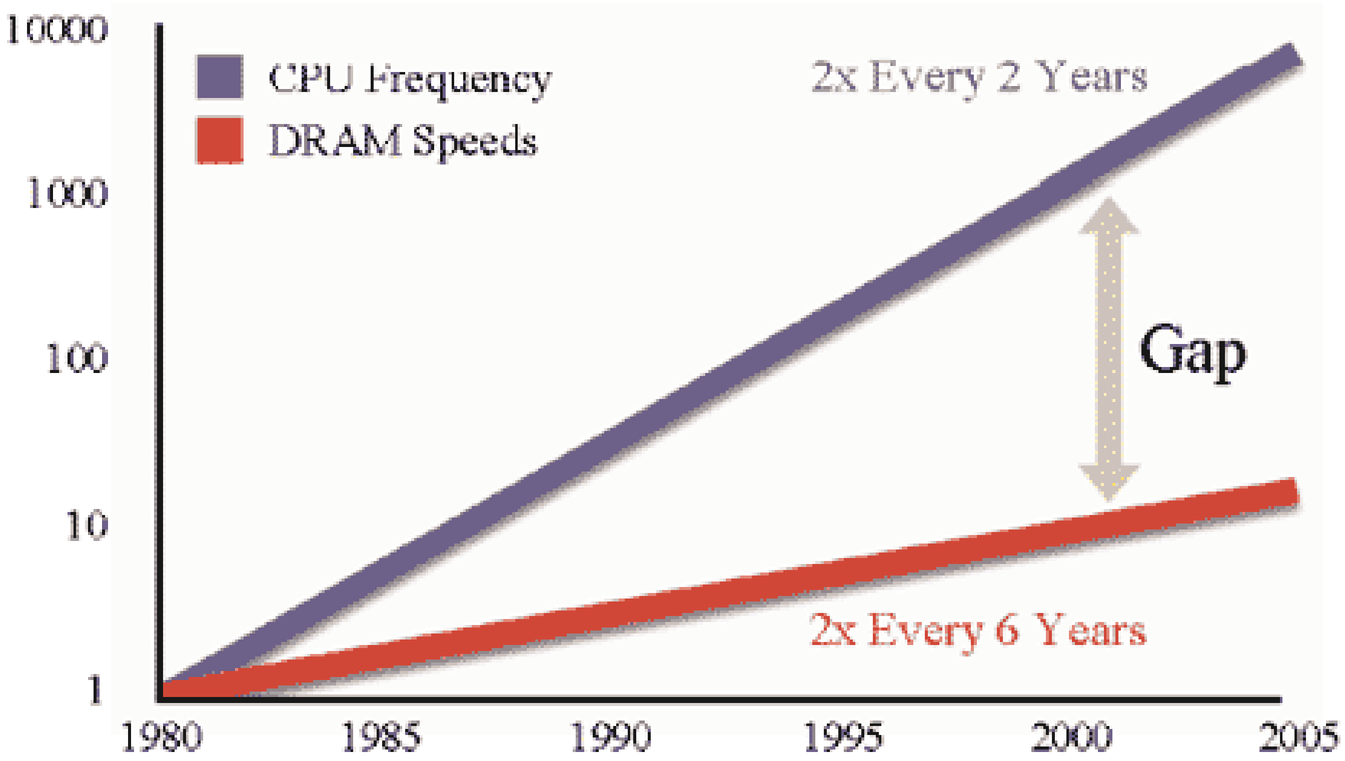
Register and Memory Configuration

Register and Memory Configuration of a basic computer:



Why Cache is important?

Whenever resource can be divided into pieces, some of which used much more heavily than others, caching is often used to improve performance.



- Larger size than registers
- Much faster than RAM

Cache

- Cache controlled by hardware
- Main memory is divided up into cache lines, typically 64 bytes, with addresses 0 to 63 in cache line 0, 64 to 127 in cache line 1, and so on
- Heavily used cache lines are kept in a high-speed cache near CPU
- Program read a memory word, the cache hardware checks to see if the line needed is in the cache
    - Cache hit, the request is satisfied from the cache and no memory request is sent over the bus to the main memory
    - Cache hits normally take only a few clock cycles.
    - Cache misses have to go to memory, with a substantial time penalty of tens to hundreds of cycles.
- Limited in size due to high cost

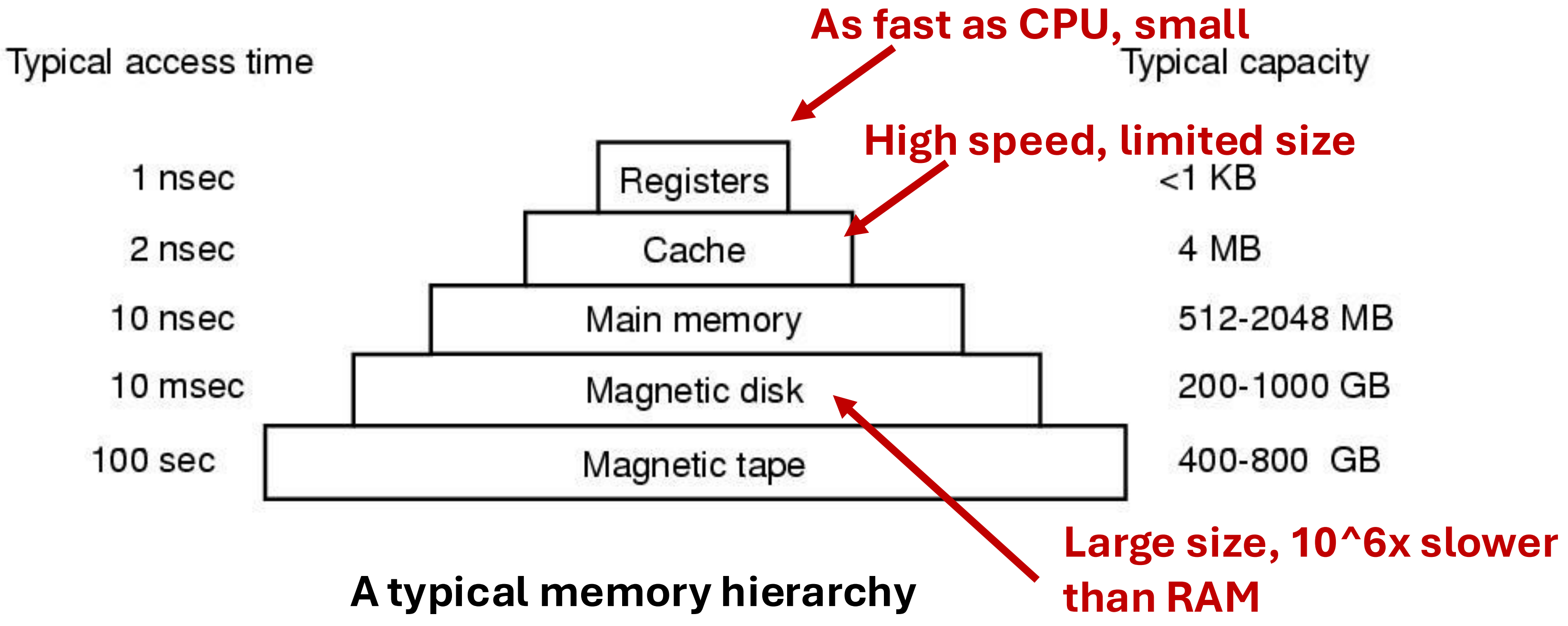
More on CPU Cache

- Different types of cache
  - L1 cache: inside CPU and feeds decoded instructions into CPU’s execution engine (no delay)
  - L2 cache: inside or outside CPU, holds megabytes of recently used memory words (1-2 clock cycles)
  - L3 cache: improves the performance of L1 & L2 cache

Cache Management

1. When to put a new item into the cache.
2. Which cache line to put the new item in.
3. Which item to remove from the cache when a slot is needed.
4. Where to put a newly evicted item in the larger memory.

Memory



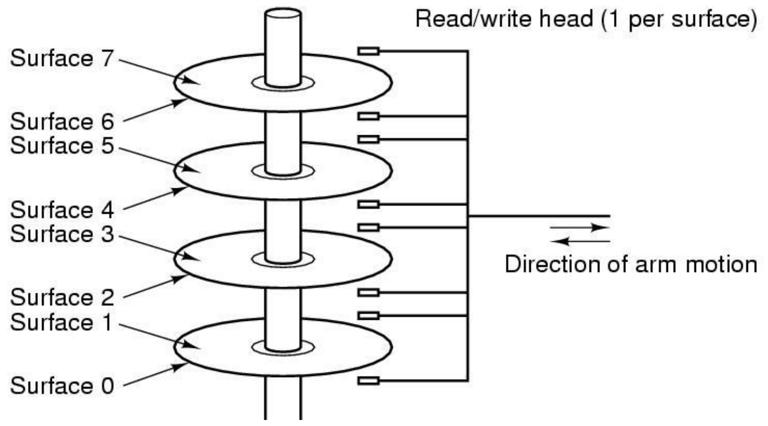
A typical memory hierarchy

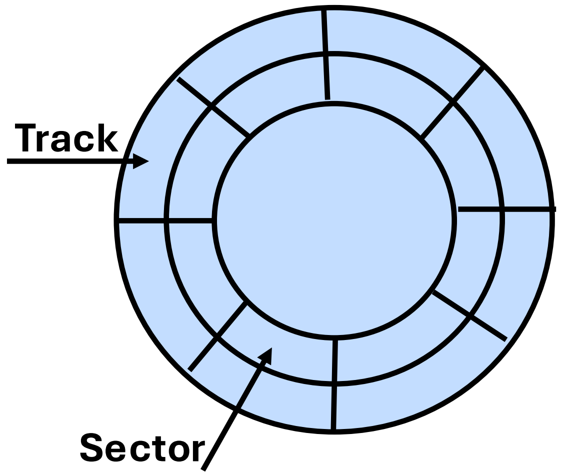
Memory Management

- Multiprogramming
  - Accommodate multiple processes in main memory
  - How to *protect* the programs from one another and the kernel from them all?
    - Processes should not reference memory locations in another process without permission
  - What if all processes' memory requirement exceeds the physical memory?

Virtual memory address ←→ Physical memory address

Hard Disks





Disk heads move together

- A stack of platters, a surface with a magnetic coating
- Disk head: each side of a platter has separate disk head
- Each surface is divided into tracks and sectors
  - 500 to 2,000 tracks per surface
  - 32 to 128 sectors per track
  - YouTube Video: https://youtu.be/3owqvmMf6No
    - *How a Hard Drive works in Slow Motion - The Slow Mo Guys*, 2012

Magnetic Disk Characteristics

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

Average seek time as reported by the industry:

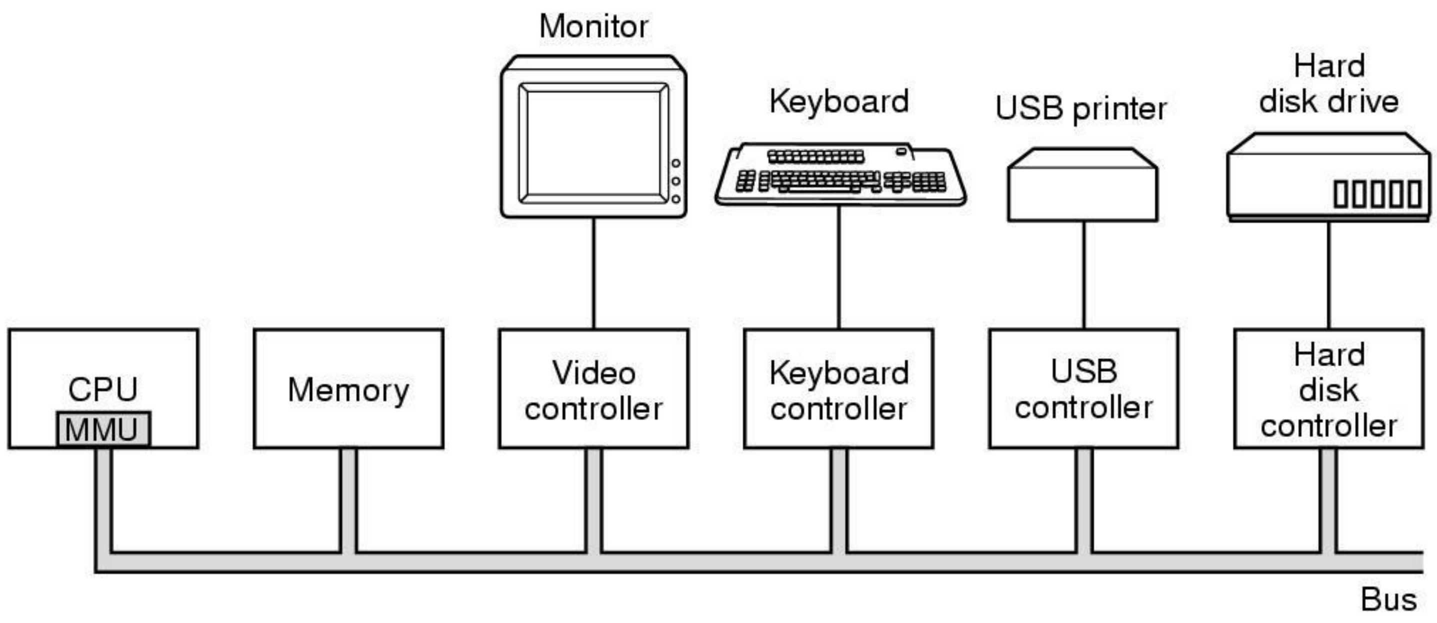
- Typically, in the range of 8 ms to 15 ms

Due to locality of disk reference

- Actual average seek time may only be 25% to 33% of the advertised number

Solid State Drives

- No moving parts
- Based on semiconductor chips and features:
  - low power consumption
  - compact size
  - shock resistance
  - high performance for random data accesses
- No seek time but constant latency
- Reading data from an SSD takes tens of microseconds instead of milliseconds as with hard disks.
- Writes are more complicated as they require a full data block to be erased first and take more time



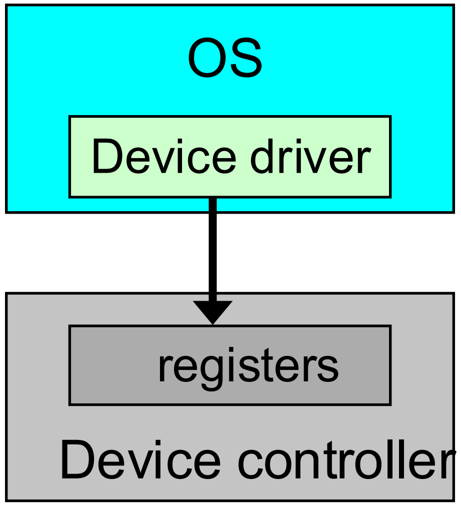
I/O Devices

- Device controller
  - To provide a simple interface of device control to OS
  - Physically controls the device: accepts commands from OS, and carries them out
- Device driver
  - The software that talks to a controller, giving it commands and accepting responses

I/O Devices

- Device controller
  - To provide a simple interface of device control to OS
  - Physically controls the device: accepts commands from OS, and carries them out
- Device driver
  - The software that talks to a controller, giving it commands and accepting responses

Every controller has a small number of registers:
A disk controller might have registers for specifying
disk address, direction (r/w), etc.



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?

Hardware device events

- Hardware devices produce events at times and in patterns not known in advance
  - Keyboard presses
  - Incoming network packets
  - Mouse movements

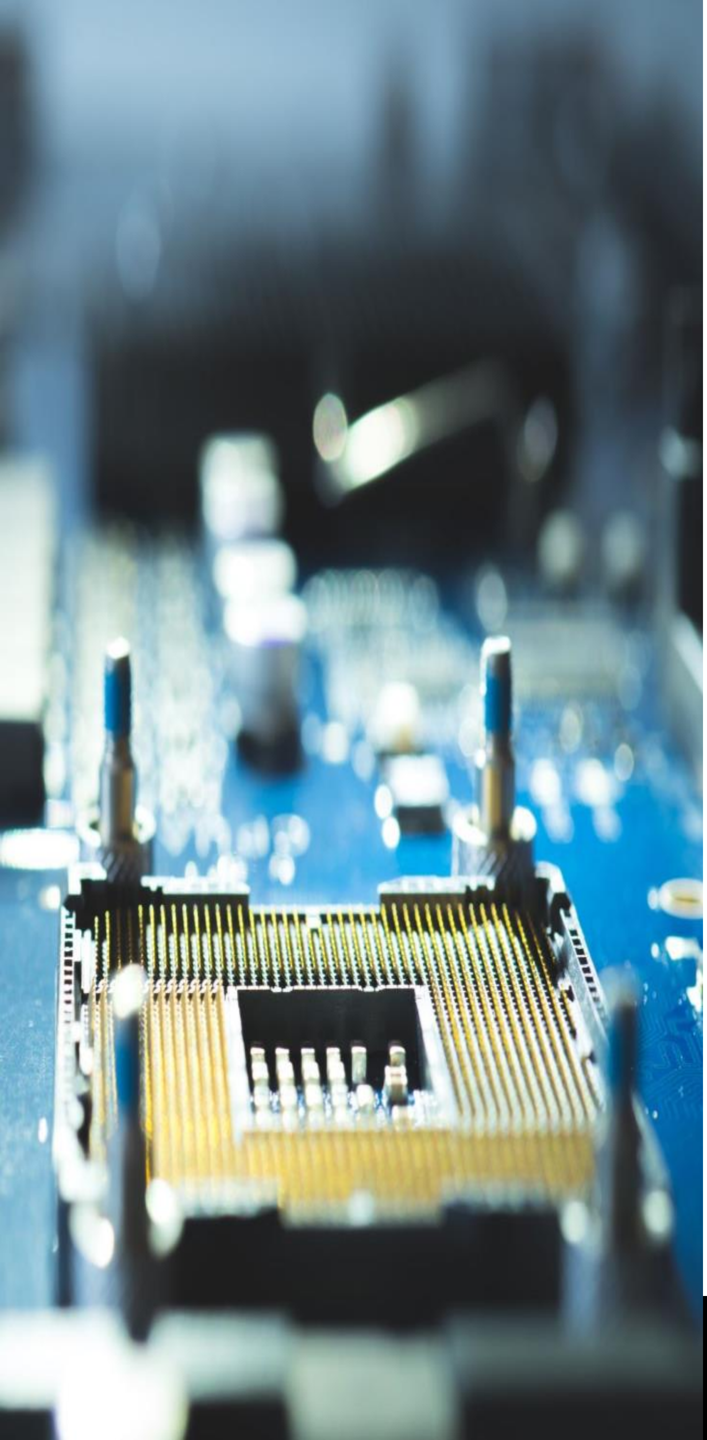
How to make information generated by these events available to running applications?

- Three methods: polling, interrupt, DMA

How I/O Devices Notify the OS?

Polling (busy waiting)

- The I/O device put information in a status register
- The OS periodically check the status register
- Reading data from an SSD takes tens of microseconds instead of milliseconds as with hard disks.
- Writes are more complicated as they require a full data block to be erased first and take more time



How I/O Devices Notify the OS?

Polling (busy waiting)

- 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

Direct Memory Access (DMA) --> Ch. 5

- A chip that can control the flow of bits between memory and some controller without constant CPU intervention

Interrupts

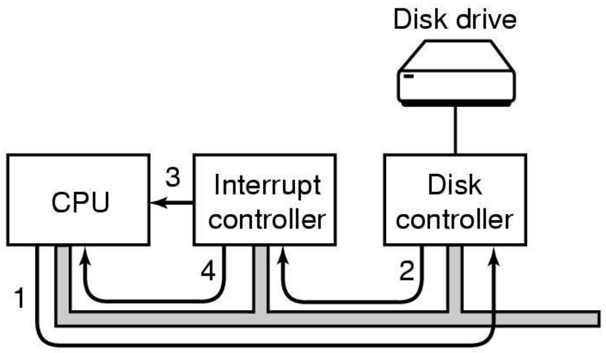
Interrupts

- An interruption of the normal sequence of execution
- Improves processing efficiency (compare to polling)
- Complex
  - A suspension of a process caused by an external event
  - Performed in a way that the process can be resumed

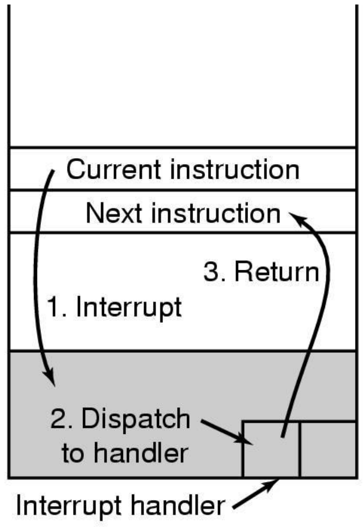
Types of interrupts

- I/O
- Timer

I/O Interrupt



(a)



(b)

1. Save current instruction info, switch to kernel mode
2. Find interrupt handler (service routine) for device
3. Interrupt handler finishes: return to user program

System Calls

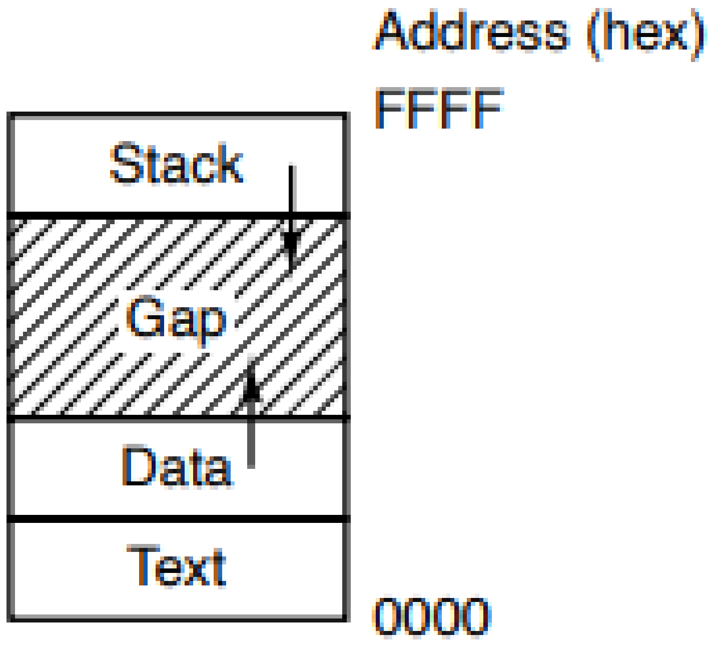
Transition between user/kernel mode

- Interrupt – HW device requests OS services
- Trap – user program requests OS services: system calls
  - A user program needs a system service (reading a file): execute a **trap** instruction to transfer control to OS
  - OS inspects parameters, carries out system call, returns control to the instruction following the system call
- Exception – error handling

Processes in Unix

Memory divided up into three segments:

- The text segment (i.e., the program code)
- The data segment (i.e., the variables)
- Stack segment



'e 1-20. Processes have three segments: tex

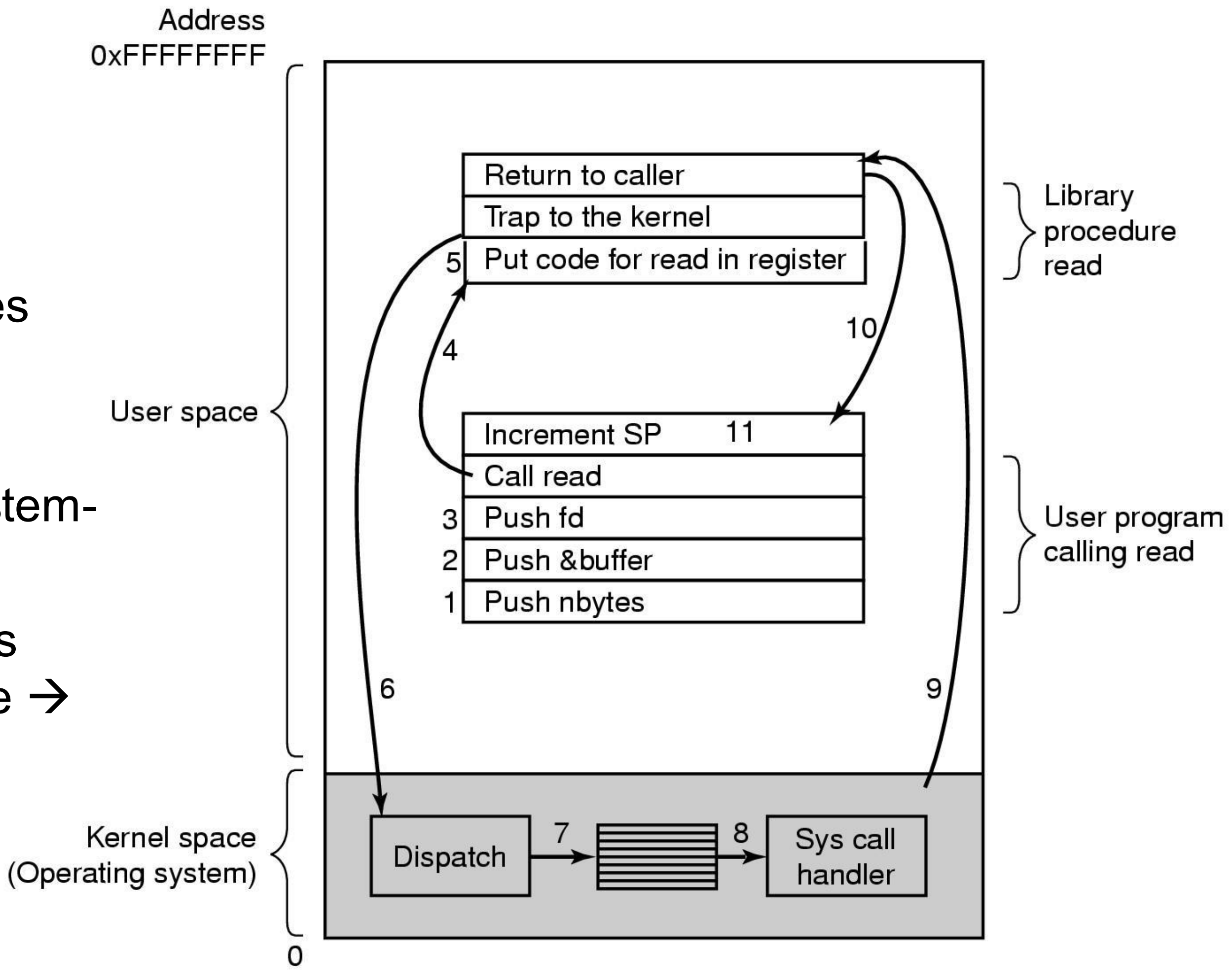
System Calls

1 – 3: calling program pushes paras on stack

4: calls the library procedure

5: library procedure puts system-call # in a register

6: library procedure executes TRAP instruction: user mode → kernel mode



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

System Calls

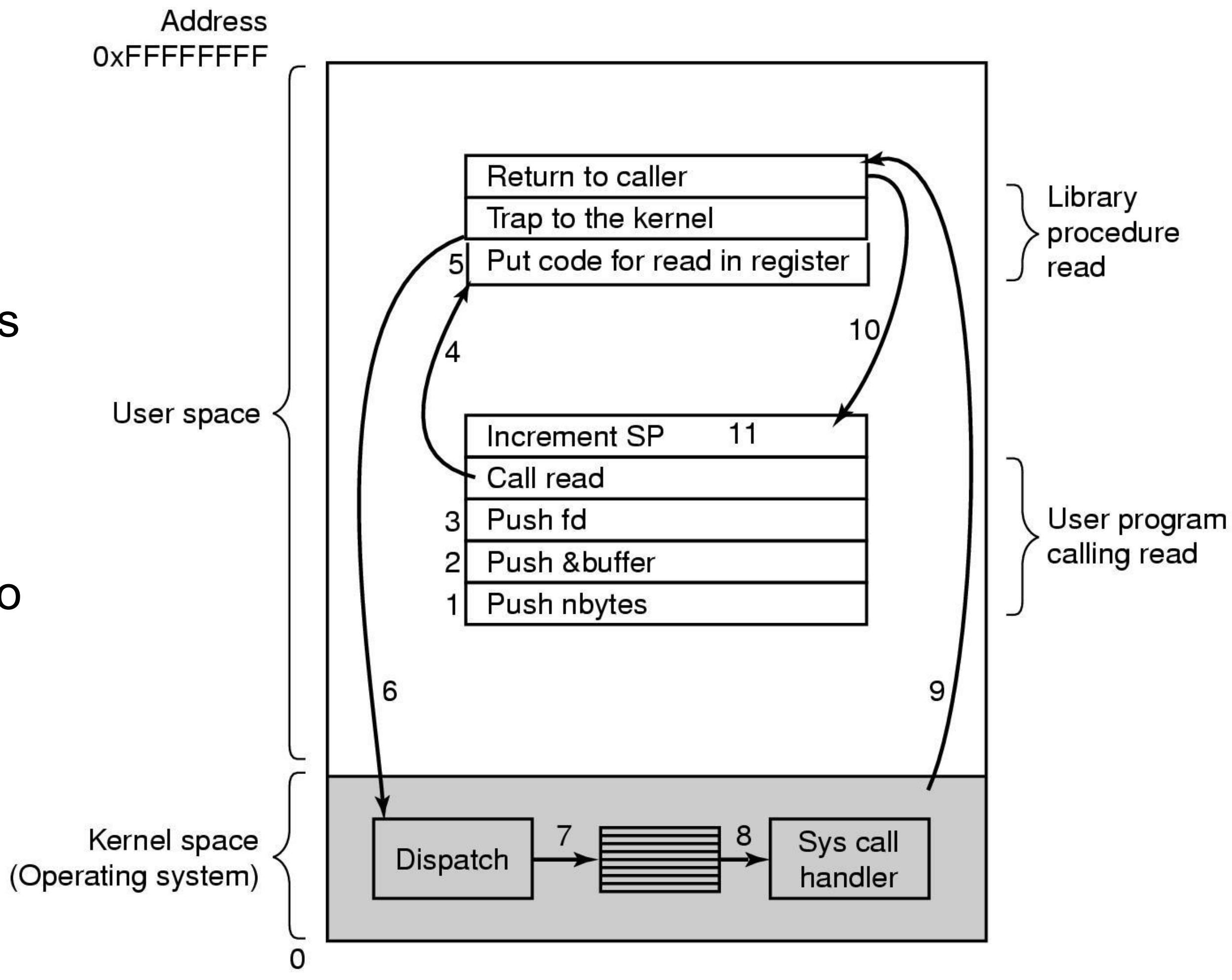
7: kernel code examines system-call #, dispatches to correct system-call handler

8: the system-call handler runs

9: control returned to user-space library procedure at the instruction following TRAP

10: library procedure returns to user program

11: user program cleans up stack

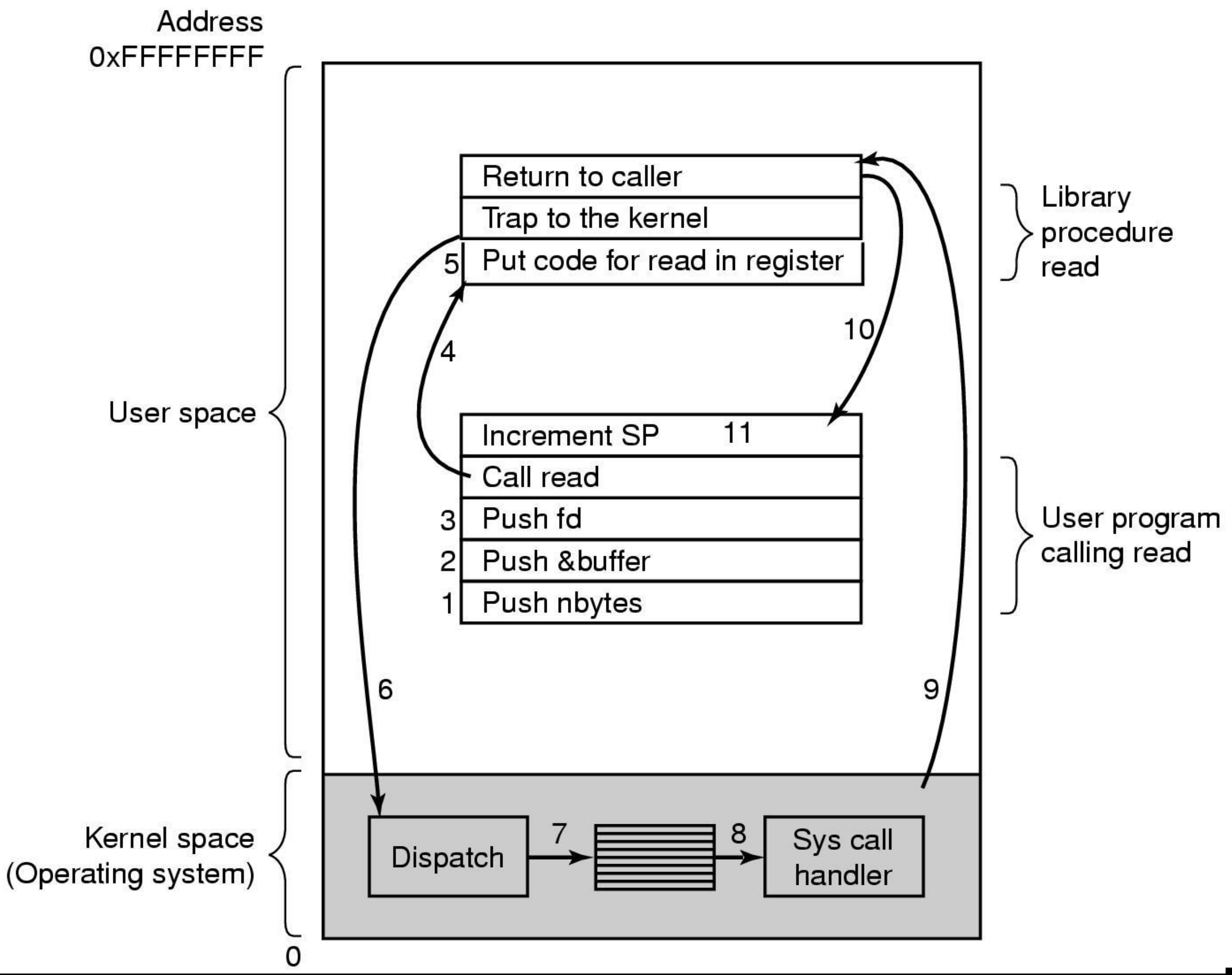


There are 11 steps in making the system call read (fd, buffer, nbytes)

System Calls

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

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



Operating System Components

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

Process Management (Chapter 2)

- Process: a fundamental OS concept
  - Memory address space
  - Some set of registers (program counter, stack pointer)

OS responsibilities for process management

- Process creation and deletion
- Process scheduling, suspension, and resumption
- Inter-process communication and synchronization

Memory Management (Chapter 3)

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 (Chapter 4)

A file is a collection of data usually stored on disk with a unique name

- Programs, data
- Devices (UNIX & Linux)

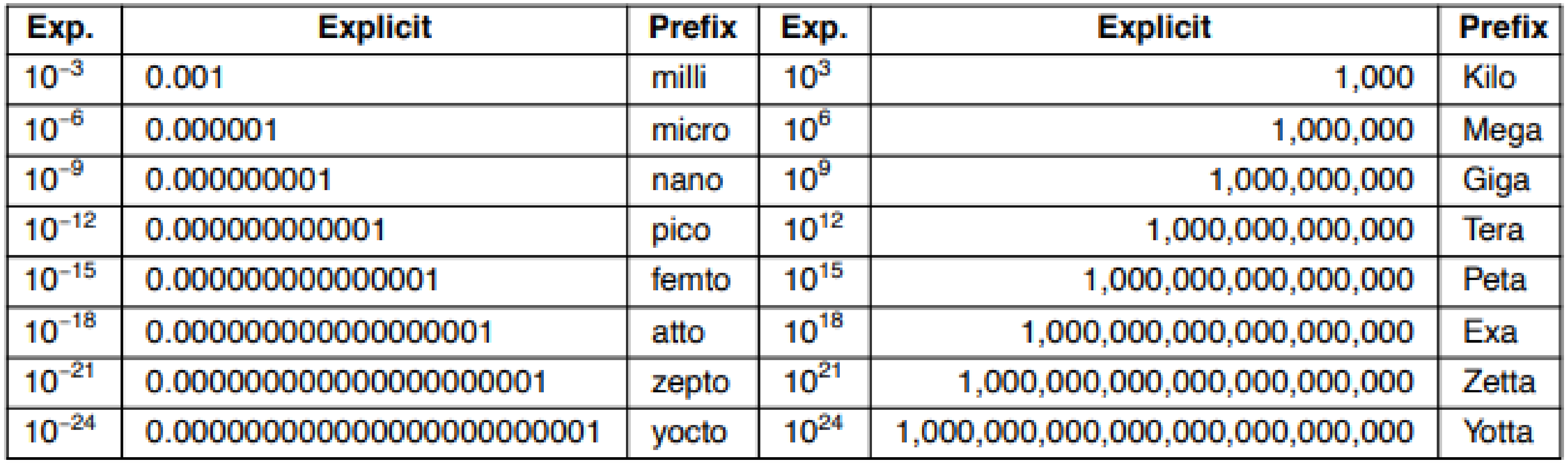
OS responsibilities for file management

- Organize directories and files
- Map files onto disk

OS responsibilities for disk management

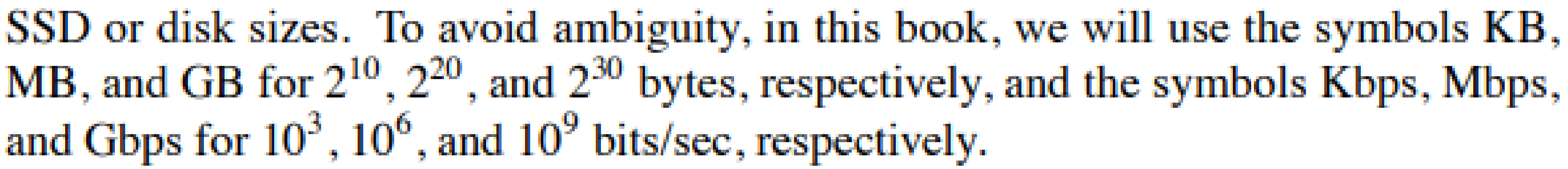
- Disk space management
- Disk scheduling

Metric Units



Exp.	Explicit	Prefix	Exp.	Explicit	Prefix
10^{-3}	0.001	milli	10^{3}	1,000	Kilo
10^{-6}	0.000001	micro	10^{6}	1,000,000	Mega
10^{-9}	0.000000001	nano	10^{9}	1,000,000,000	Giga
10^{-12}	0.000000000001	pico	10^{12}	1,000,000,000,000	Tera
10^{-15}	0.000000000000001	femto	10^{15}	1,000,000,000,000,000	Peta
10^{-18}	0.000000000000000001	atto	10^{18}	1,000,000,000,000,000,000	Exa
10^{-21}	0.000000000000000000001	zepto	10^{21}	1,000,000,000,000,000,000,000	Zetta
10^{-24}	0.000000000000000000000001	yocto	10^{24}	1,000,000,000,000,000,000,000,000	Yotta

Figure 1-31. The principal metric prefixes.



SSD or disk sizes. To avoid ambiguity, in this book, we will use the symbols KB, MB, and GB for 2^{10}, 2^{20}, and 2^{30} bytes, respectively, and the symbols Kbps, Mbps, and Gbps for 10^{3}, 10^{6}, and 10^{9} bits/sec, respectively.

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 a 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

EAS datacenter tutorial

Summary

Computer hardware

- Time-sharing: CPU
- Space-sharing: memory, disk

OS components

- Process management
- Memory management
- File and storage management

For Next Time!

- Read Chapter 2: Processes and Threads in *Modern Operating Systems*
- Continue to complete Homework 1

Course Content

Unit	Module	Week	Topic
1 – Introduction	1 & 2	1	Welcome to OS & Introduction
	2	2	OS Overview
2 – Processes and Threads	3	3	Processes
	4	4	Threads
	5	5	CPU Scheduling
	6	6	Multiprocessor Scheduling
	7	7	Interprocess Communication
	8	8	Pthread
3 – Deadlocks & Memory	9	9	Deadlock
	10	10	Memory Management
	--	11	Midterm Exam
4 – Additional Topics	11	11	Page Replacement
	12	13	File Systems
	13	14	IO Devices
	14	15	Security
	--	16	Final Exam