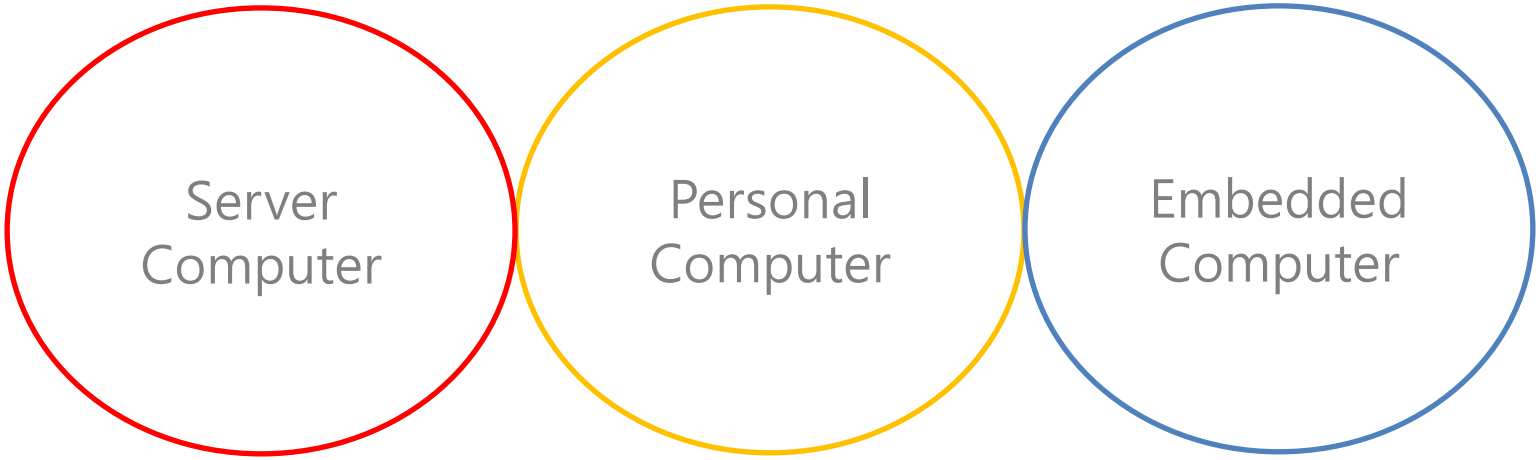
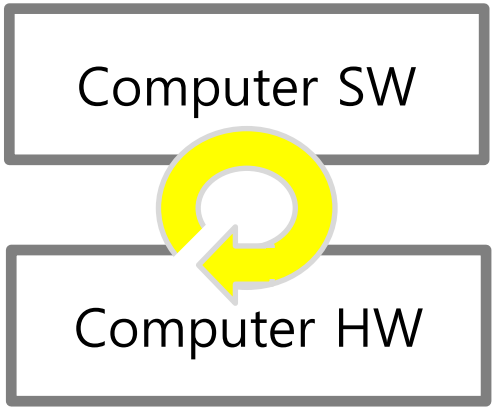
Operating System

Ch01: Introduction

BeomSeok Kim
Department of Computer Engineering
KyungHee University
passion0822@khu.ac.kr

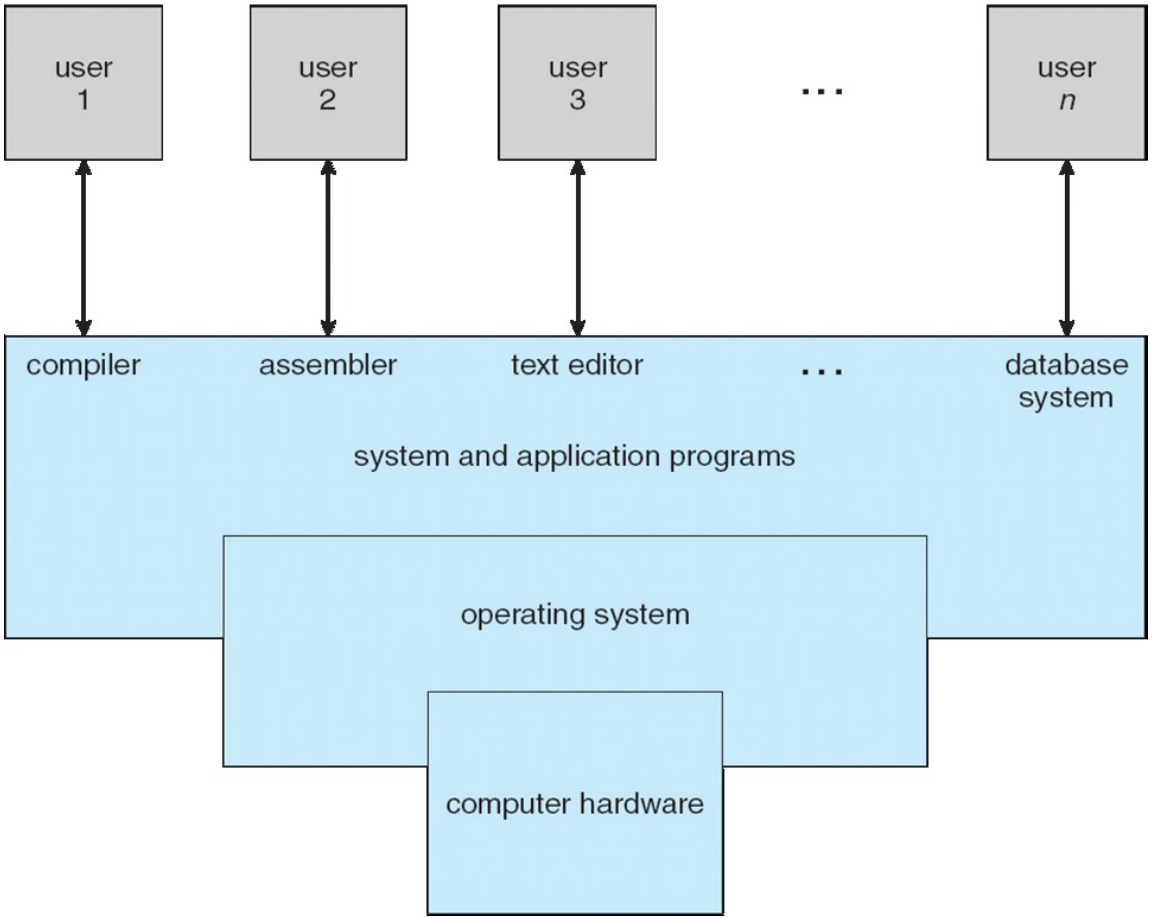
Computer?


Server Computer
Personal Computer
Embedded Computer


Computer SW
Computer HW

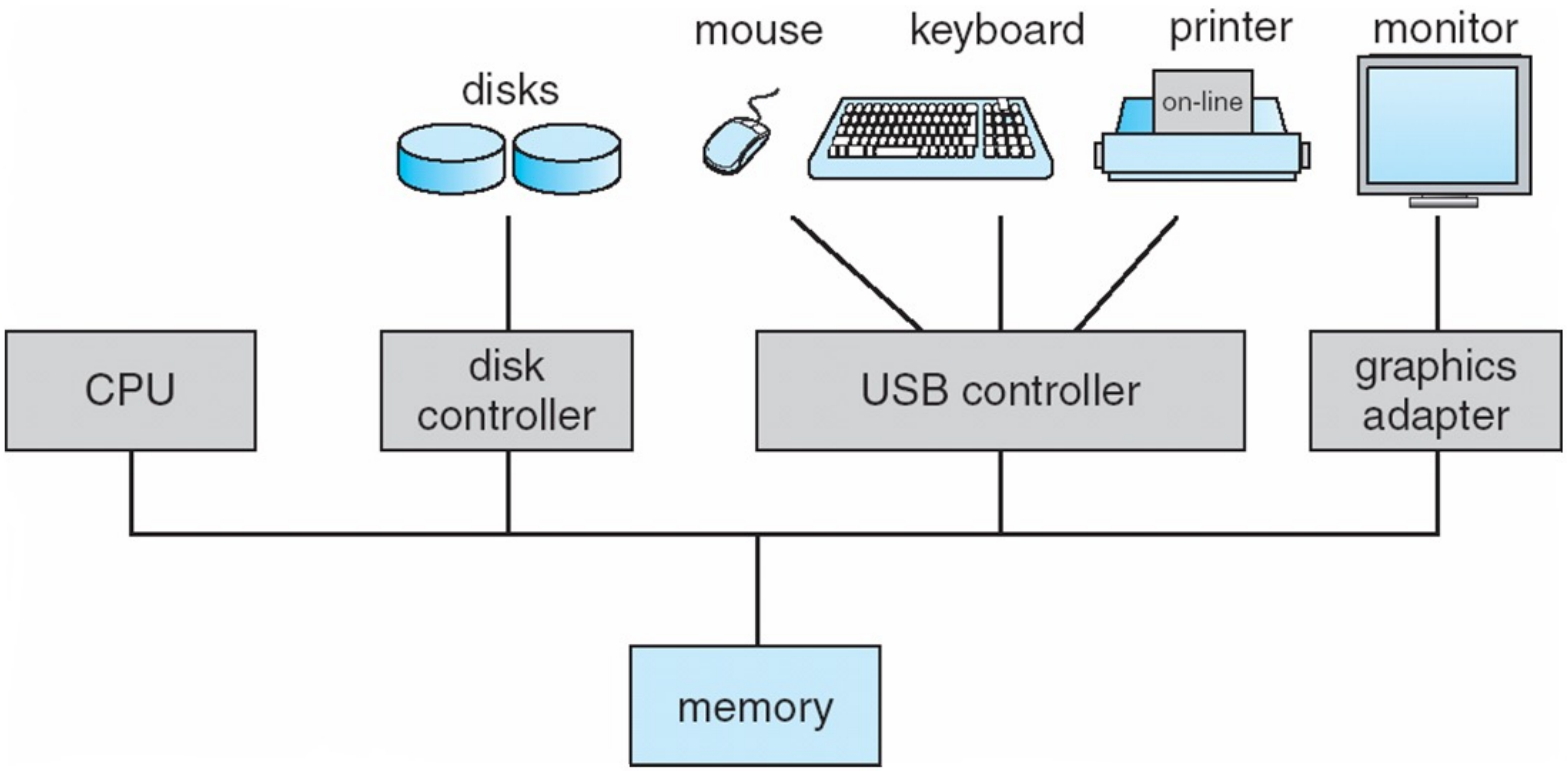
Computer?

- Computer system components (Abstract view)



Computer?

- Computer system organization

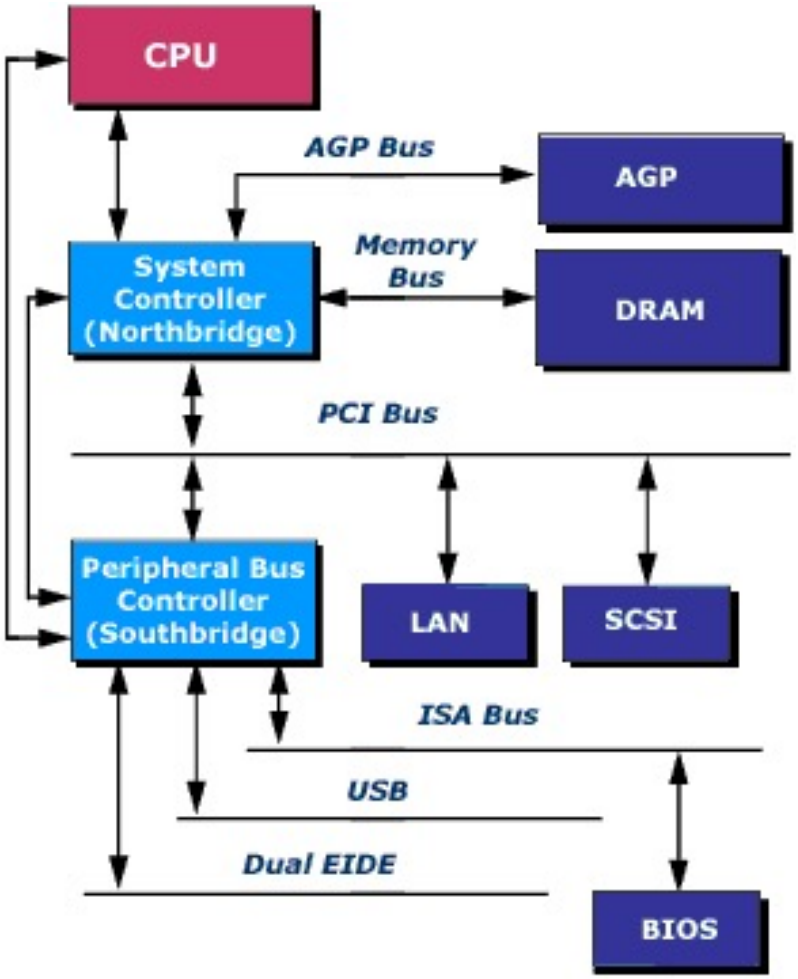


Computer?

- Modern PC architecture

Memory Controller Hub (MCH)

I/O Controller Hub (ICH)



Operating Systems?

- OS is a resource manager
  - ✓ Abstraction
  - ✓ Sharing
    - ➢ Time multiplexing
    - ➢ Space multiplexing
  - ✓ Protection
  - ✓ Fairness
  - ✓ Performance

- Definition in textbook
  - ✓ Resource allocator
  - ✓ Control program
  - ✓ Kernel

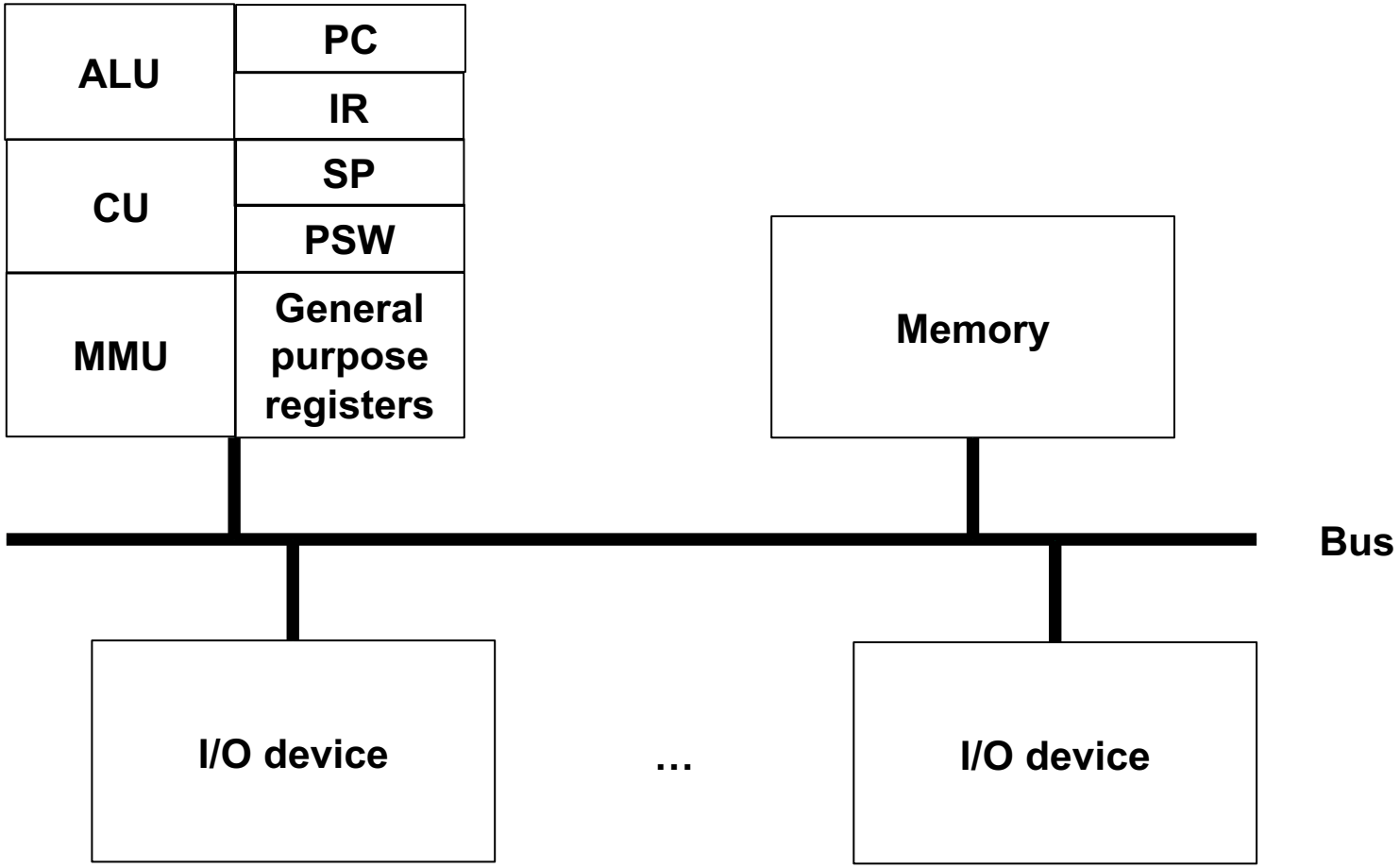
Resources

- CPU
- Memory
- I/O devices
- …

SW which manages
'Computer HW' resources for
1) convenience
2) efficiency

Computer System Operation

- Computer HW architecture (Abstract view)



Computer System Operation

- CPU operation
  - ✓ Von Neumann architecture
  - ✓ Instructions
    - ➢ Arithmetic instructions: add, subtract, multiply, divide, …
    - ➢ Logical instructions: and, or, xor, not, shift, …
    - ➢ Control flow instructions: goto, if, call, return, …
    - ➢ Data instructions: load, store, move, input, output, …

Computer System Operation

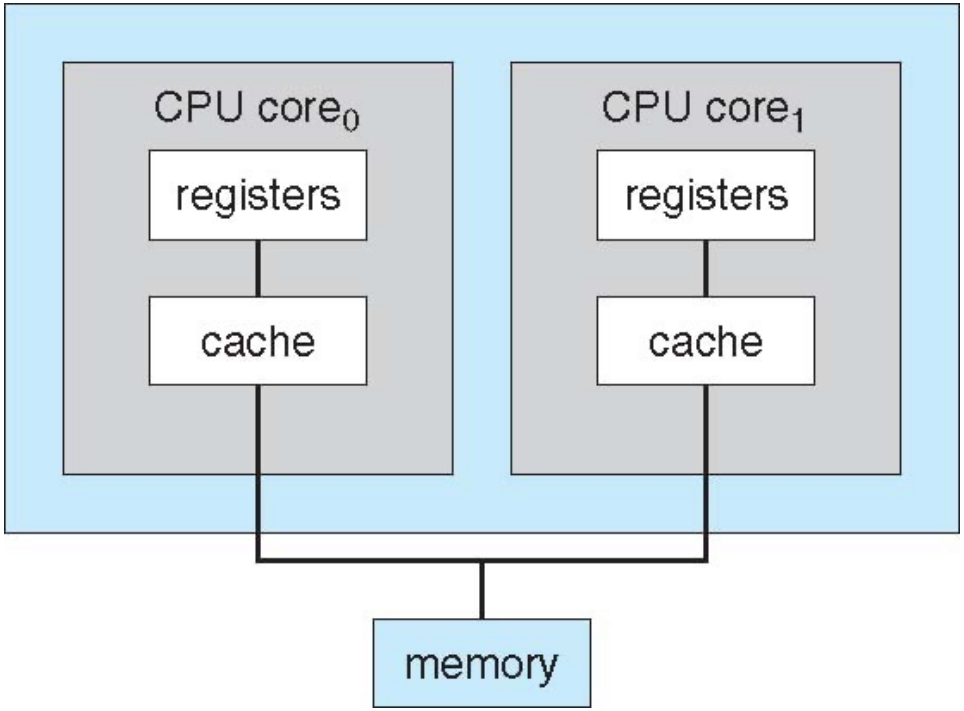
- Bootstrapping in Linux
  - ✓ The CPU initializes itself and then execute an instruction at a fixed location (0xffffff0).
  - ✓ This instruction jumps into the BIOS/UEFI.
    - ➢ BIOS (Basic Input/Output System)
    - ➢ UEFI (Unified Extensible Firmware Interface)
  - ✓ The BIOS/UEFI finds a boot device and fetches Boot Loader(LILO/GRUB).
    - ➢ MBR (Master Boot Record) in BIOS
    - ➢ ESP (Efi System Partition) in UEFI
  - ✓ The BIOS/UEFI loads and transfers control to LILO/GRUB.
  - ✓ LILO/GRUB loads the compressed kernel.
  - ✓ The compressed kernel decompresses itself and transfers control to the uncompressed kernel.

Computer System Operation

- CPU
  - ✓ Instruction Set Architecture (ISA)
    - ➢ CISC vs. RISC
    - ➢ Intel, SPARC, MIPS, PowerPC, ARM, Alpha, …
  - ✓ Pipelining
    - ➢ Fetch, Decode, Execute, Write Back, etc.
  - ✓ Instruction-Level Parallelism (ILP)
    - ➢ Superscalar vs. VLIW
    - ➢ Simultaneous multithreading
    - ➢ Multi-core

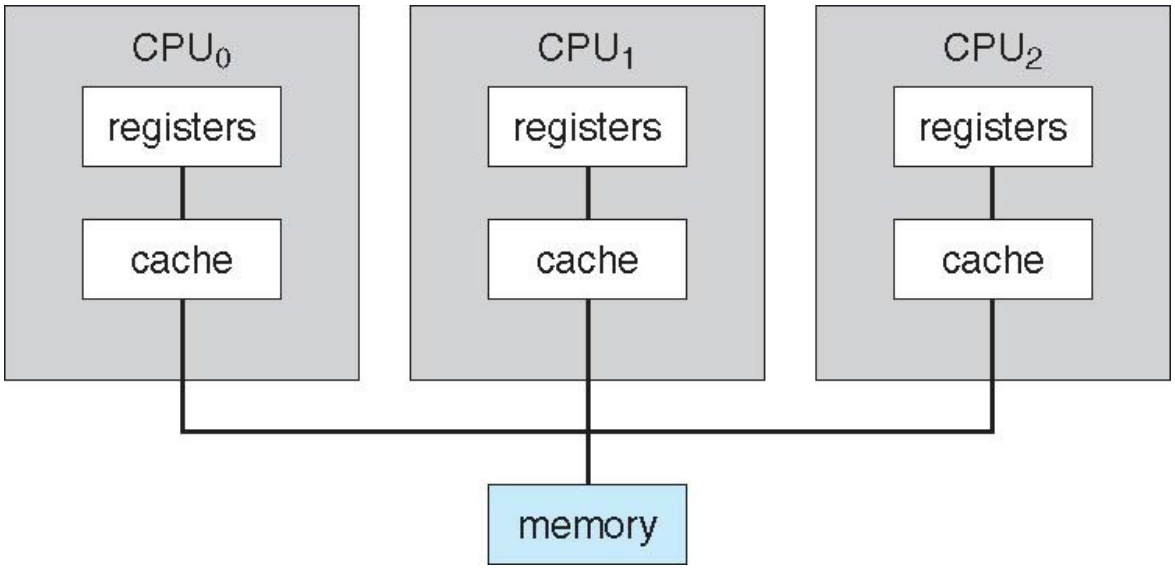
Computer System Operation

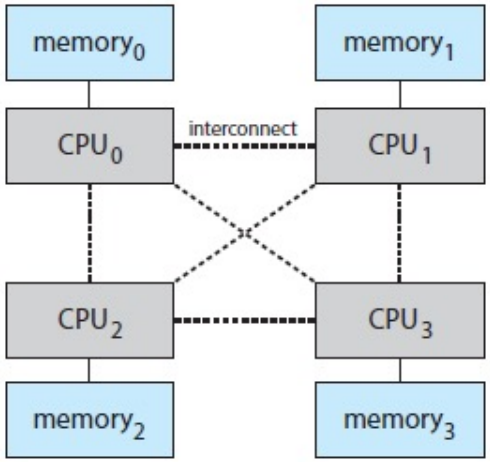
- Multi-core architecture



Computer System Operation

- Symmetric multiprocessing architecture

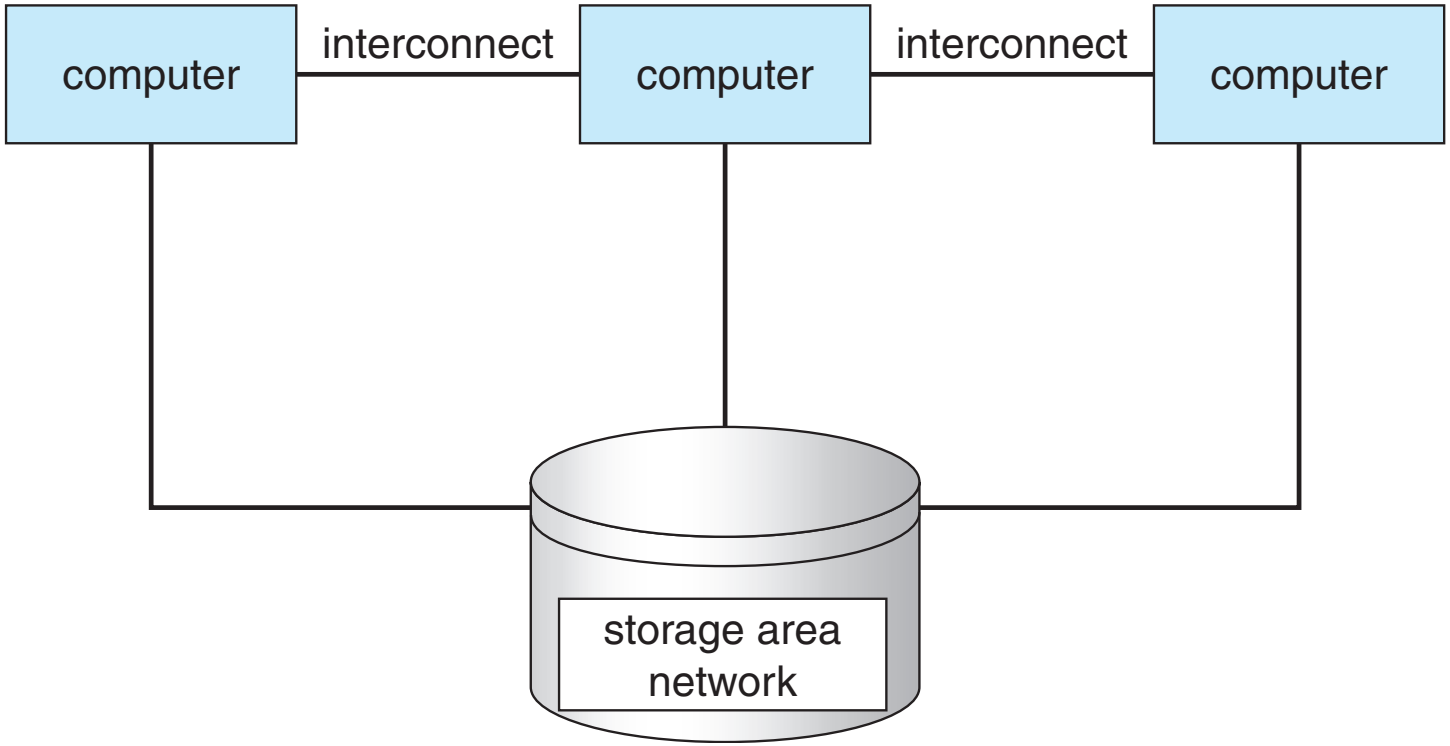




✓ Cf) NUMA (Non-Uniform Memory Access) multiprocessing architecture

Computer System Operation

- Clustered system architecture
  - ✓ Cf) Parallel vs. Distributed system

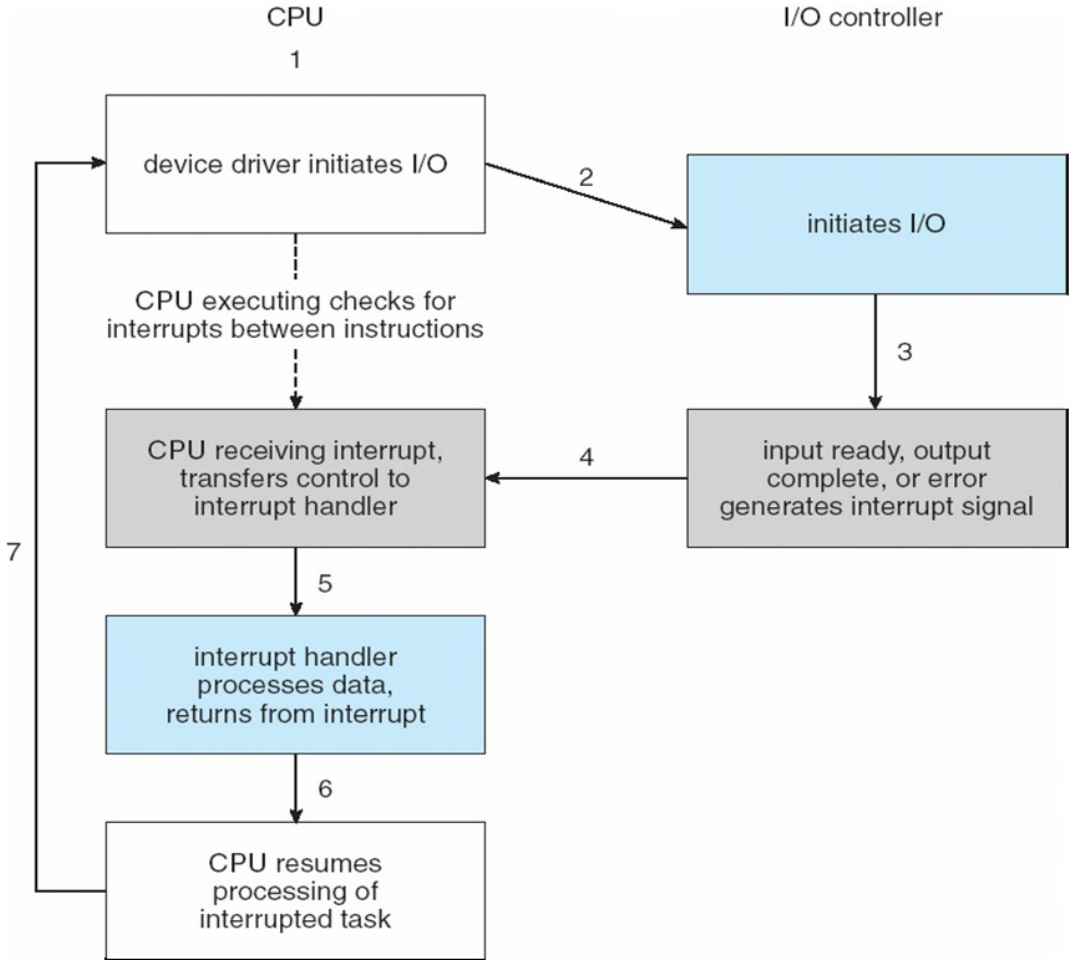


Computer System Operation

- I/O operation
  - ✓ I/O request via I/O instruction
    - ➢ Direct I/O vs. Memory-mapped I/O
    - ➢ Communicates with registers in I/O controller
    - ➢ Typically, IR (Instruction Register) & DR (Data Register)
  - ✓ I/O method
    - ➢ Programmed I/O
    - ➢ Interrupt
    - ➢ DMA (Direct Memory Access)

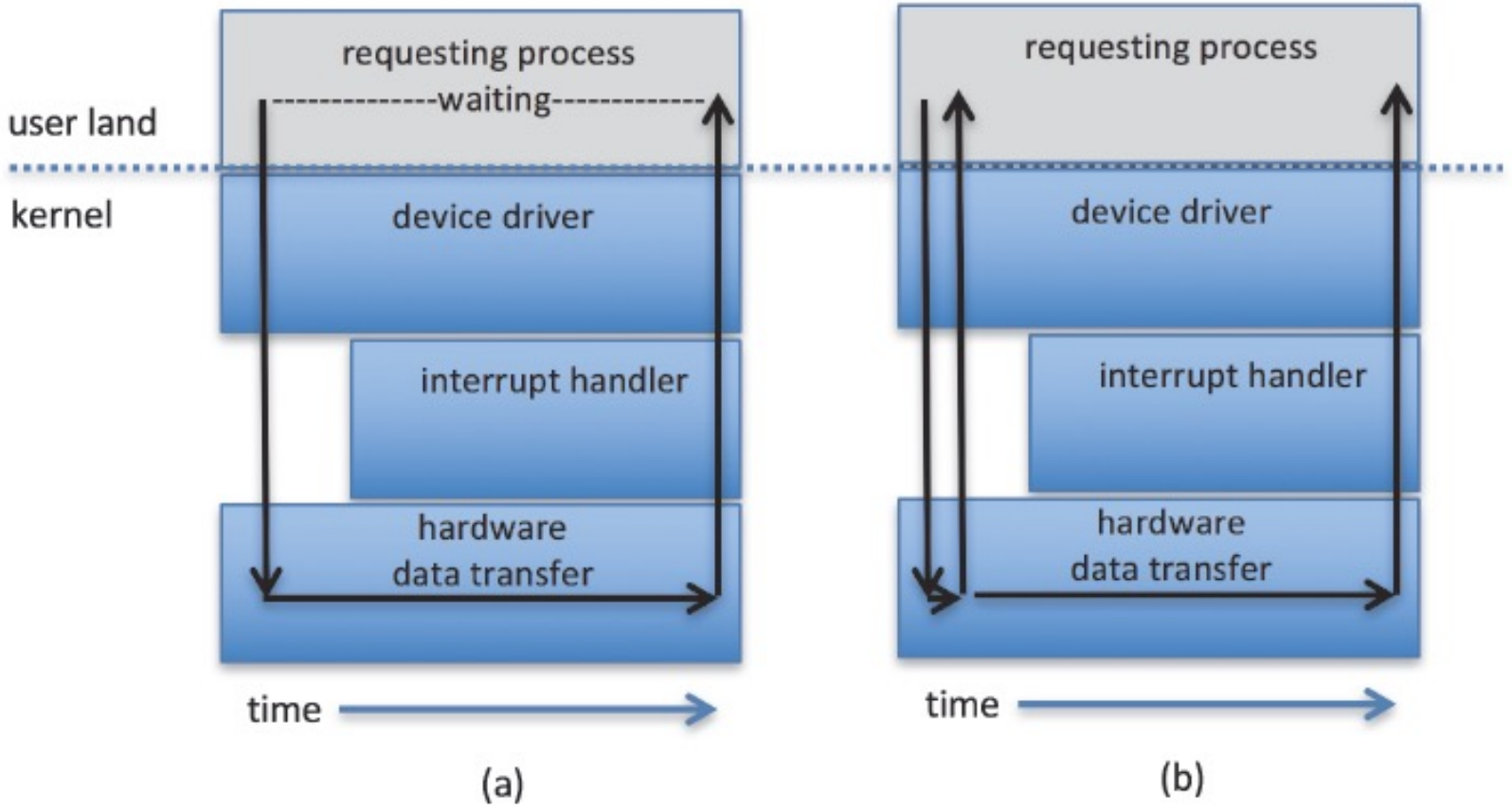
Computer System Operation

- Interrupt-driven I/O cycle



Computer System Operation

- I/O mode from the perspective of application processes

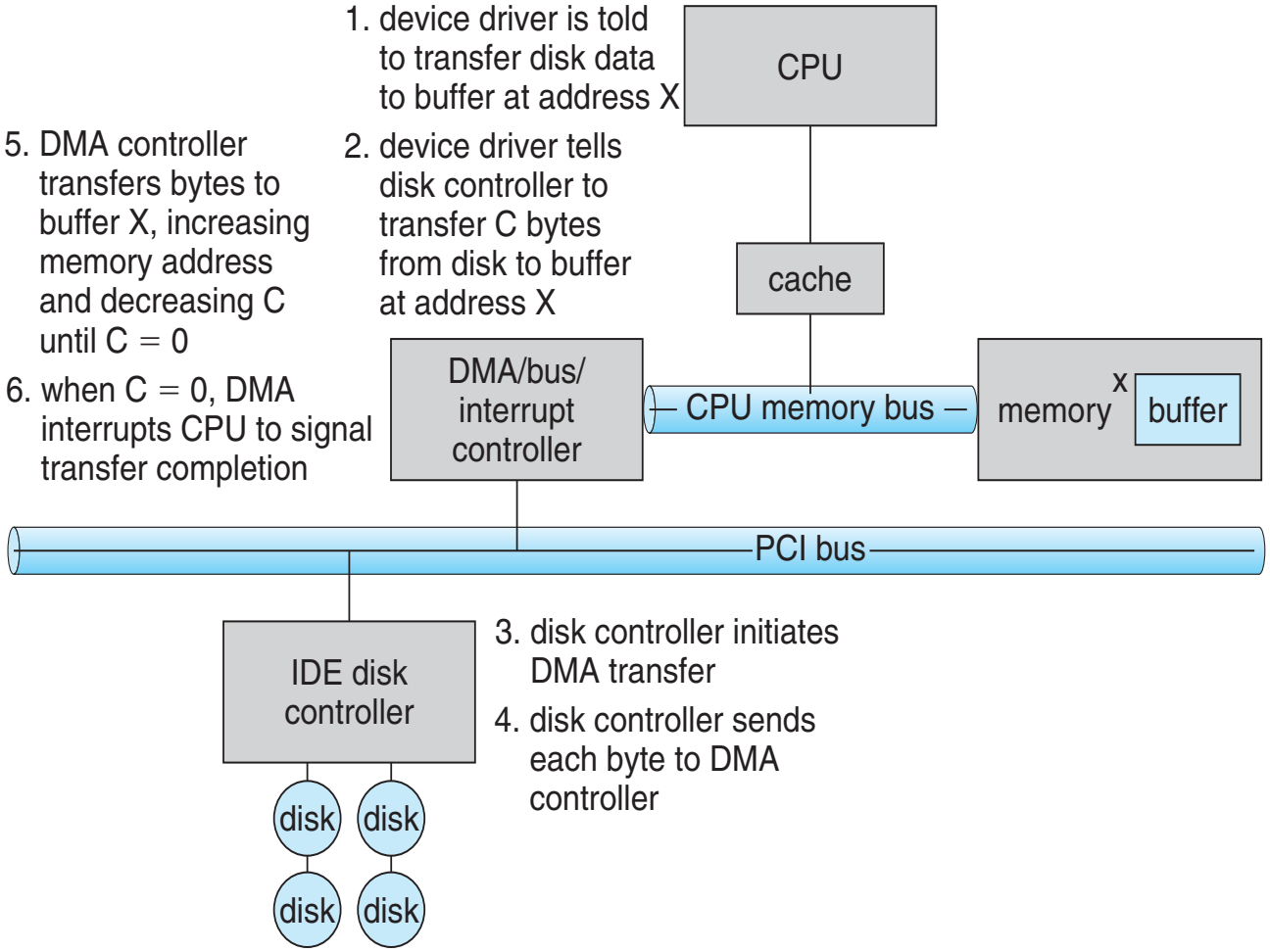


Computer System Operation

- Interrupt
  - ✓ Generated by hardware devices (asynchronous)
  - ✓ E.g.) timer interrupt, keyboard interrupt, etc.
- Trap
  - ✓ Generated by application processes (synchronous)
  - ✓ E.g.) system calls
- Fault (Exception)
  - ✓ Generated by CPU instructions (synchronous)
  - ✓ Divide-by-zero, page fault, protection fault, etc.
- Hardware interrupt vs. Software interrupt

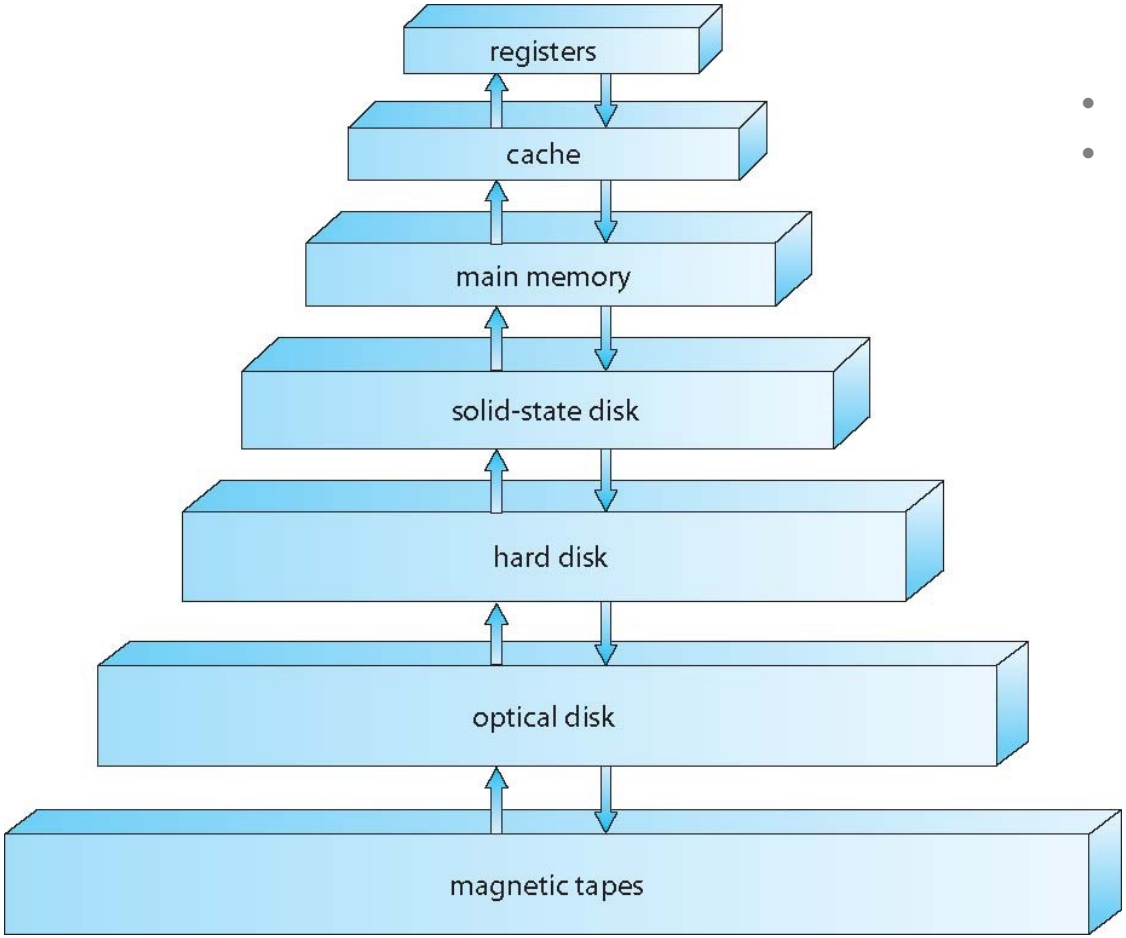
Computer System Operation

- Six steps process to perform DMA transfer



Computer System Operation

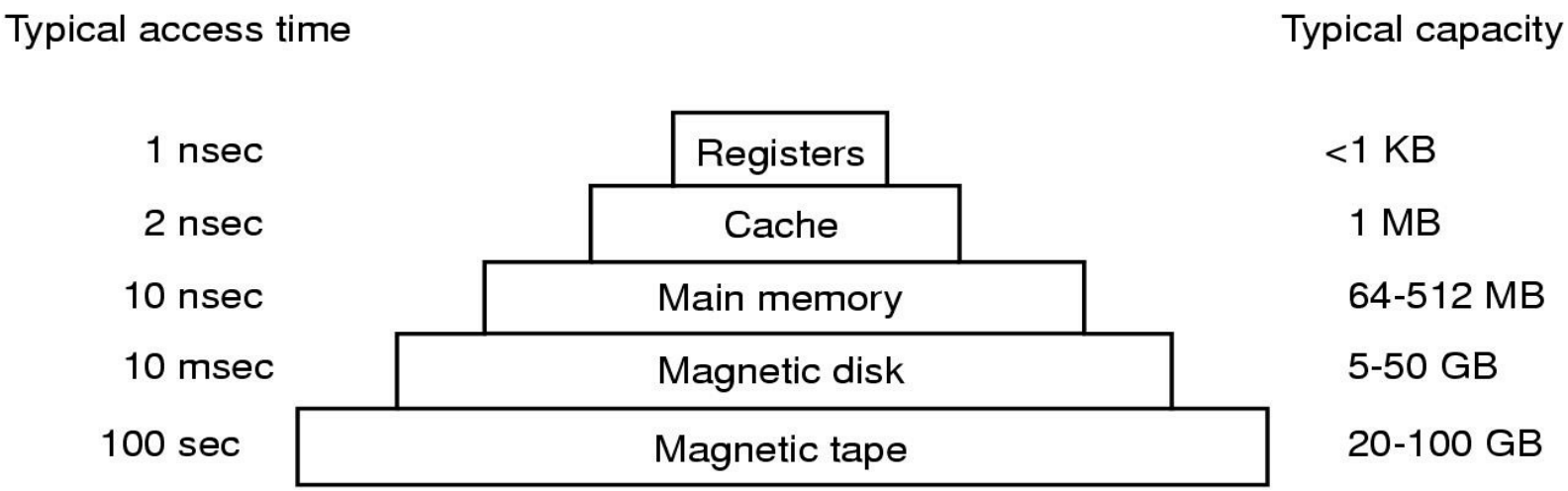
- Storage hierarchy



- **ROM vs. RAM**
- **SRAM vs. DRAM**

Computer System Operation

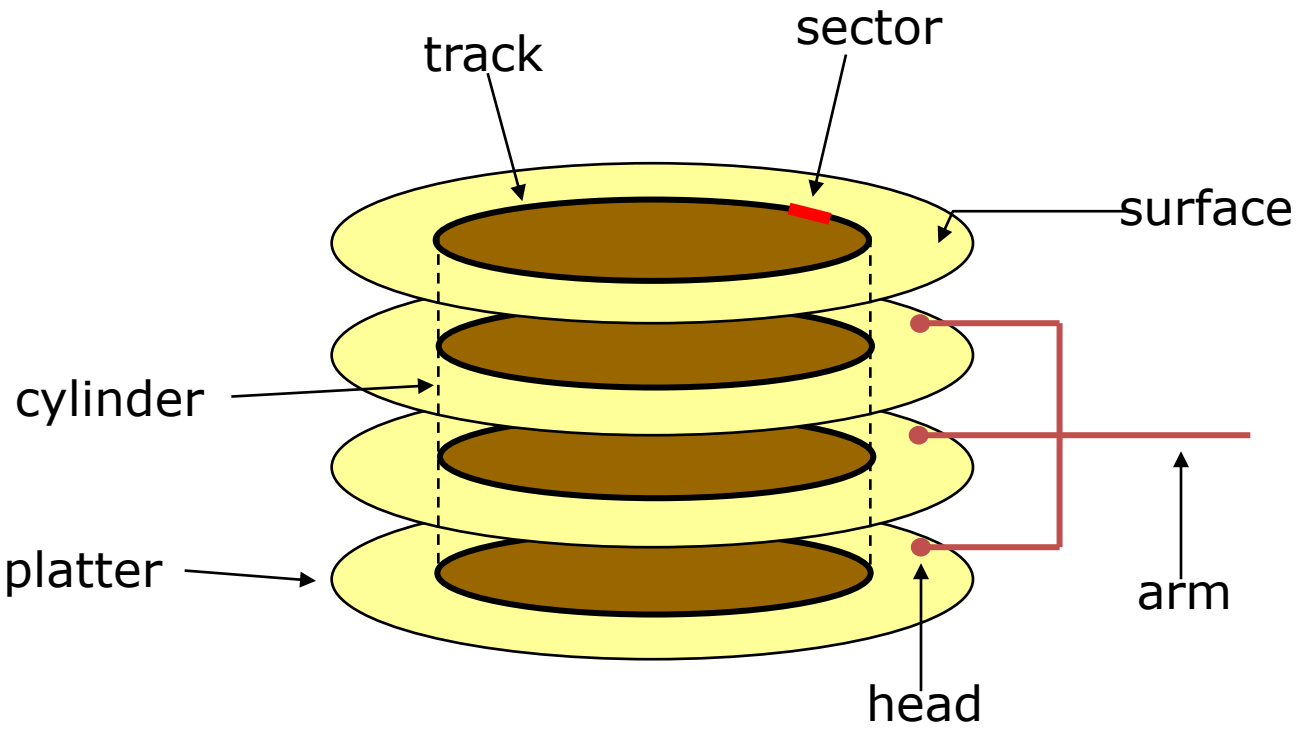
- Memory hierarchy



- Caching
  - ✓ Cache management policy
    - ➢ Write-through vs. Write-back
  - ✓ Cache coherency

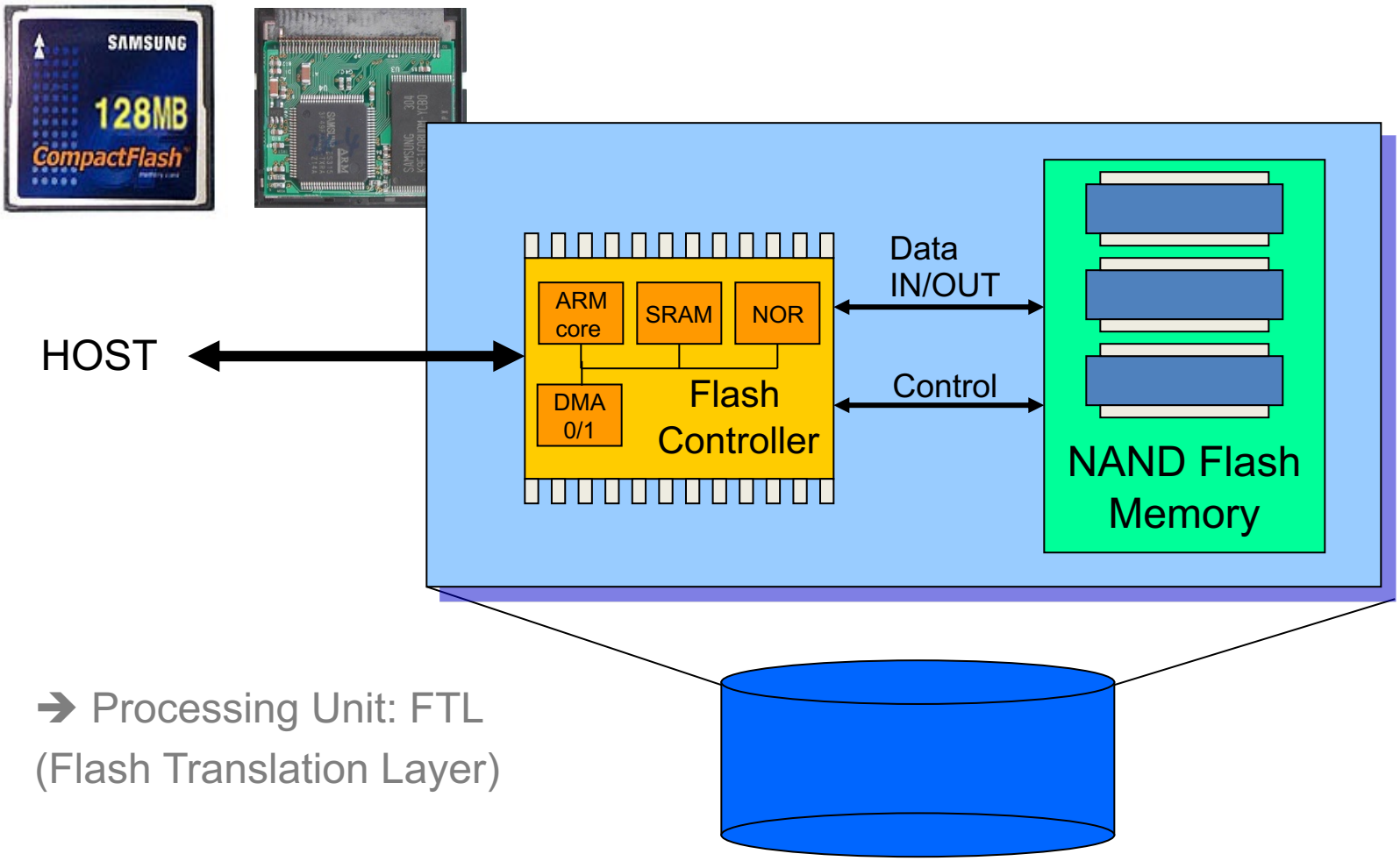
Computer System Operation

- Physical hard disk structure (HDD)
  - ✓ platters
  - ✓ surfaces
  - ✓ tracks
  - ✓ sectors
  - ✓ cylinders
  - ✓ arm
  - ✓ heads



Computer System Operation

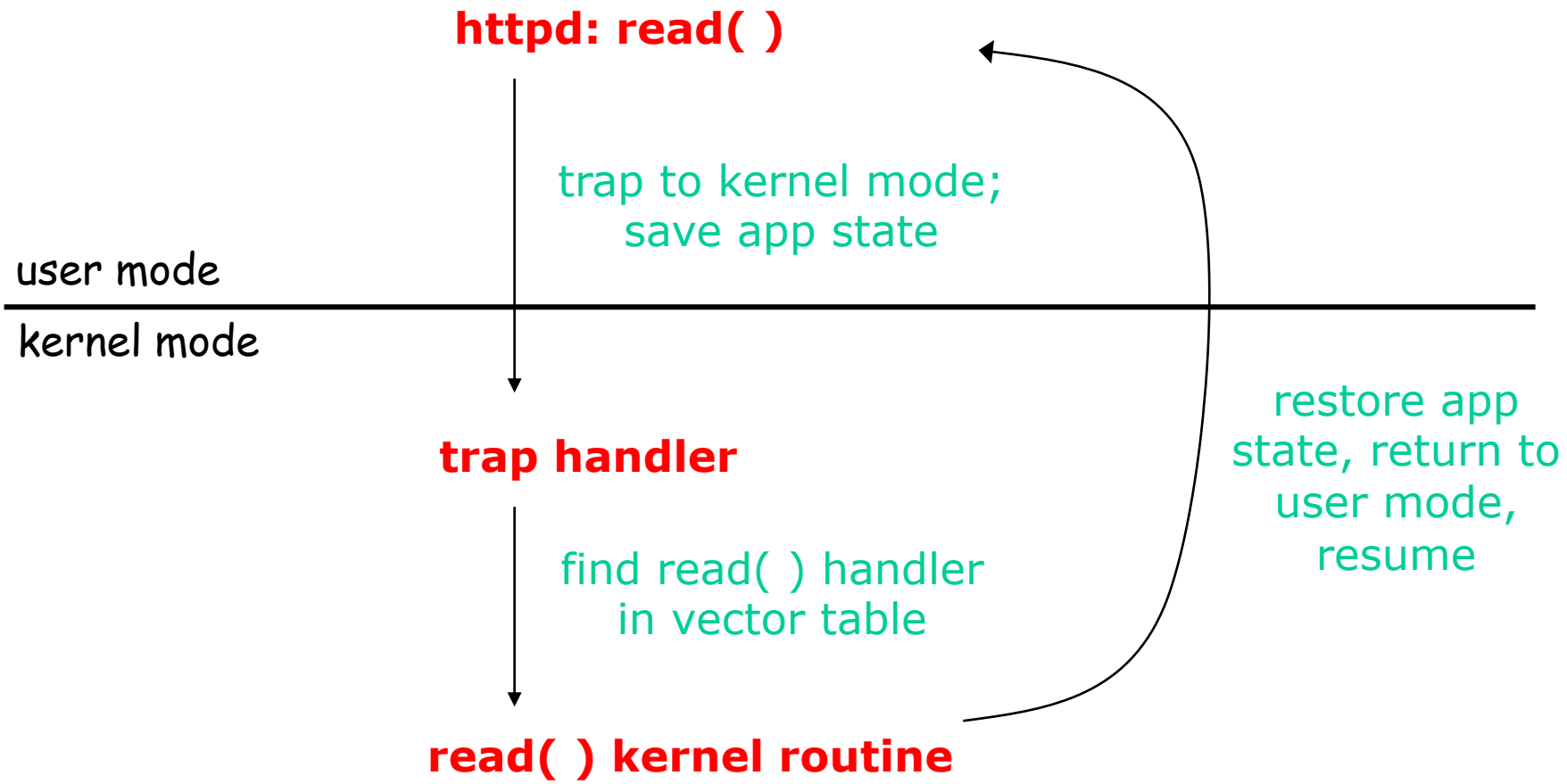
- Compact Flash card internals (SSD)



➔ Processing Unit: FTL
(Flash Translation Layer)

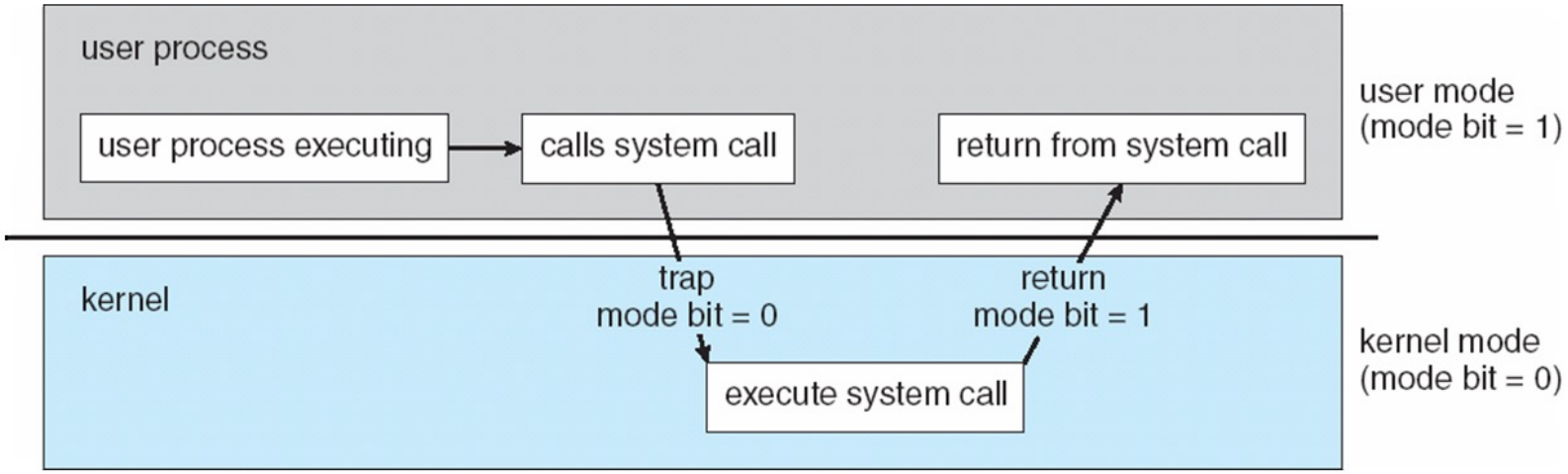
Computer System Operation

- System call
  - ✓ Cf) Function call



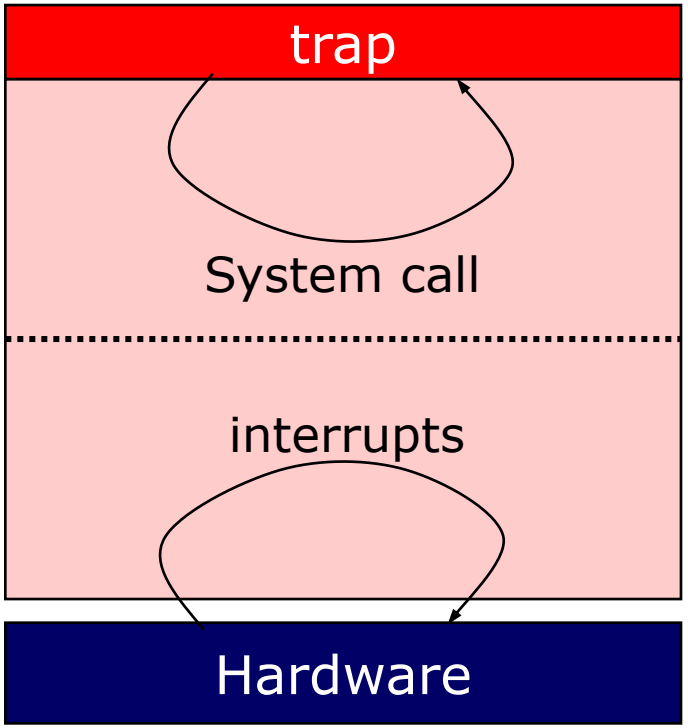
Computer System Operation

- Hardware protection
  - ✓ CPU protection
    - ➢ Timer (periodic interrupt, 10ms in Linux)
  - ✓ Memory protection
    - ➢ Protection fault (E.g. segmentation violation in Linux)
  - ✓ I/O protection
    - ➢ Dual mode operation in CPU
    - ➢ Privileged instructions only in kernel mode



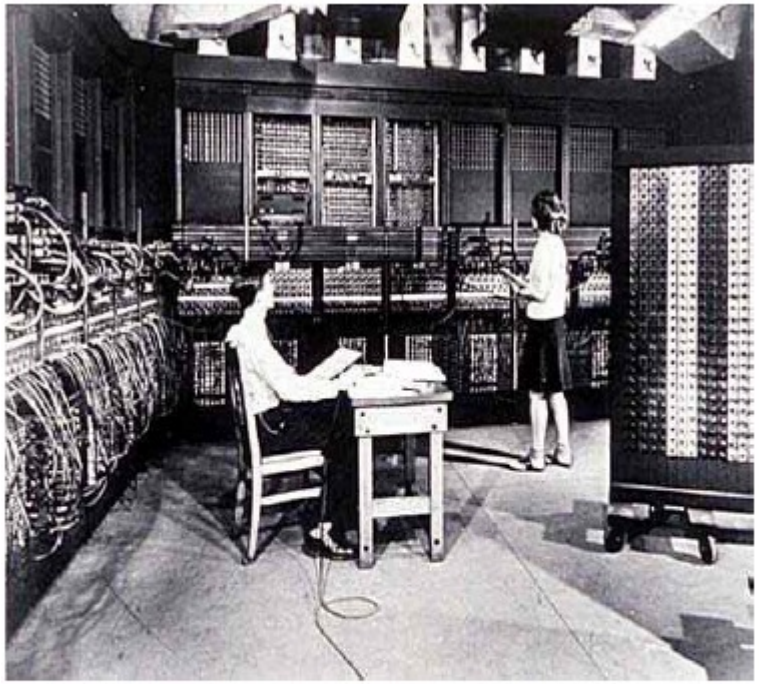
Computer System Operation

- OS takes control of the system
  - ✓ Bootstrapping
  - ✓ System calls
  - ✓ Interrupts



Computer History

- 1st Generation (1945-55)
  - ✓ *Vacuum Tubes and Plugboards*
  - ✓ No OS
  - ✓ No Programming Languages
  - ✓ No Assembly Languages

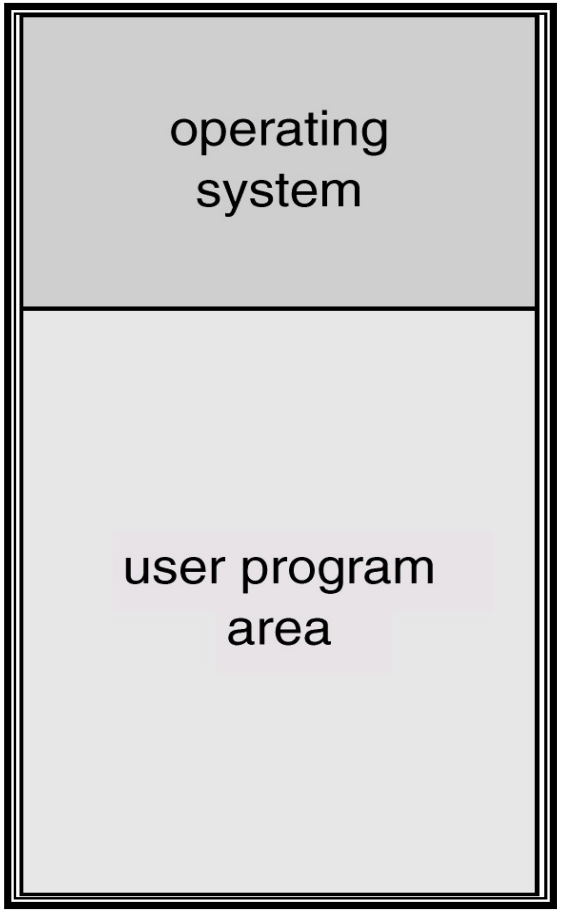


Eniac, 1940's

John Von Newman

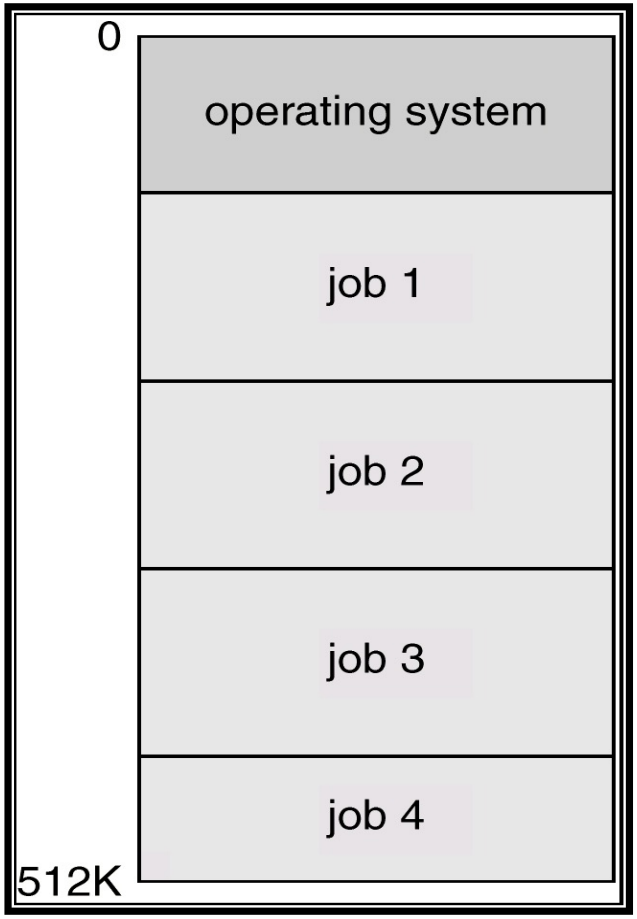
Computer History

- 2nd Generation (1955-65)
  - ✓ *Transistors and Mainframes*
  - ✓ Batch systems
    - ➢ OS is called “Resident Monitor”
    - ➢ CPU is underutilized due to the bottleneck in I/O



Computer History

- 3rd Generation (1965-80)
  - ✓ *Integrated Circuits (ICs)*
  - ✓ Architectural advances
    - ➢ The notion of “Computer Architecture”
    - ➢ IBM System/360 family
  - ✓ Multiprogramming systems
    - ➢ Increased CPU utilization
  - ✓ Time-sharing systems
    - ➢ Improved response time
    - ➢ Traditional OS features



Computer History

- 4th Generation (1980-)
  - ✓ *LSIs & VLSIs*
  - ✓ Architectural advances
    - ➢ Microprocessors: smaller and faster
    - ➢ Storages: larger and faster
    - ➢ CPU work is offloaded to I/O devices
    - ➢ Personal computers (Desktop system)
  - ✓ Modern OS Features
    - ➢ GUI (Graphical User Interface)
    - ➢ Multimedia
    - ➢ Internet & Web
    - ➢ Networked / Distributed, etc.

Computing Environments

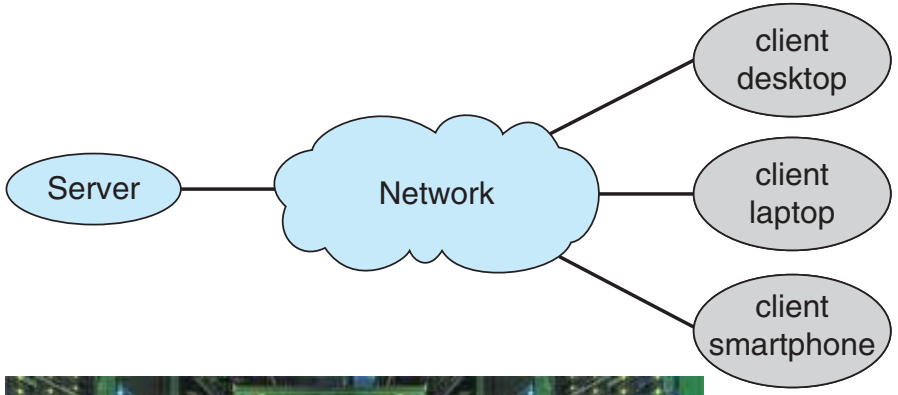
- Traditional computing
  - ✓ Mainframe system
    - ➢ Batch system
    - ➢ Multiprogramming system
    - ➢ Time-sharing system
  - ✓ Desktop system

- Mobile computing
  - ✓ Hand-held system
    - ➢ Limited memory
    - ➢ Slow processors
    - ➢ Small display screens

- Real-time embedded computing
  - ✓ Real-time systems
    - ➢ Hard real-time vs. Soft real-time
  - ✓ Embedded systems

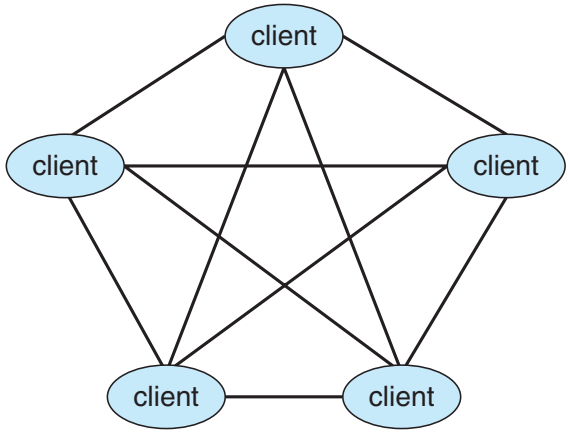
Computing Environments

- Client-server computing
  - ✓ Large-scale server in data center
  - ✓ Clustered server
    - ➢ Parallel + Distributed + Storage

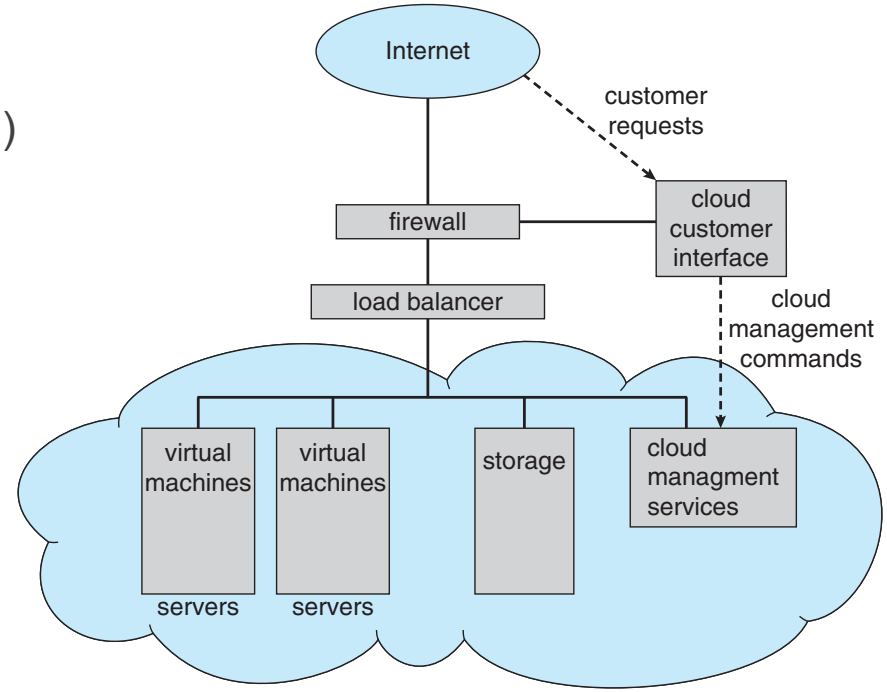


Computing Environments

- Peer-to-peer computing
  - ✓ Discovery protocol
  - ✓ Napster, Gnutella, VoIP, etc.

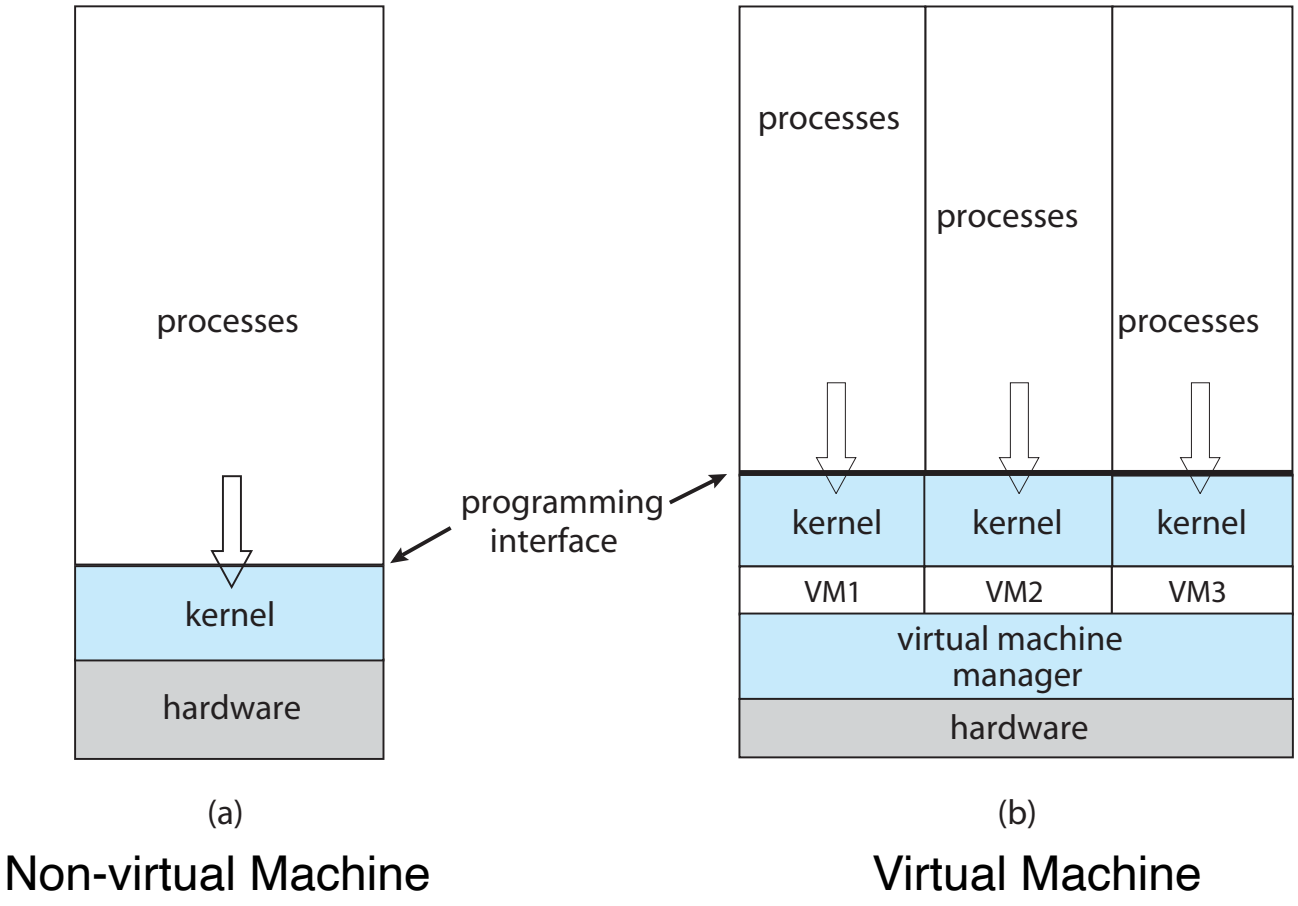


- Cloud computing
  - ✓ Infrastructure as a Service (IaaS)
  - ✓ Platform as a Service (PaaS)
  - ✓ Software as a Service (SaaS)



Computing Environments

- Virtualization
  - ✓ VMM or Hypervisor



Computing Environments

- Virtual machine examples

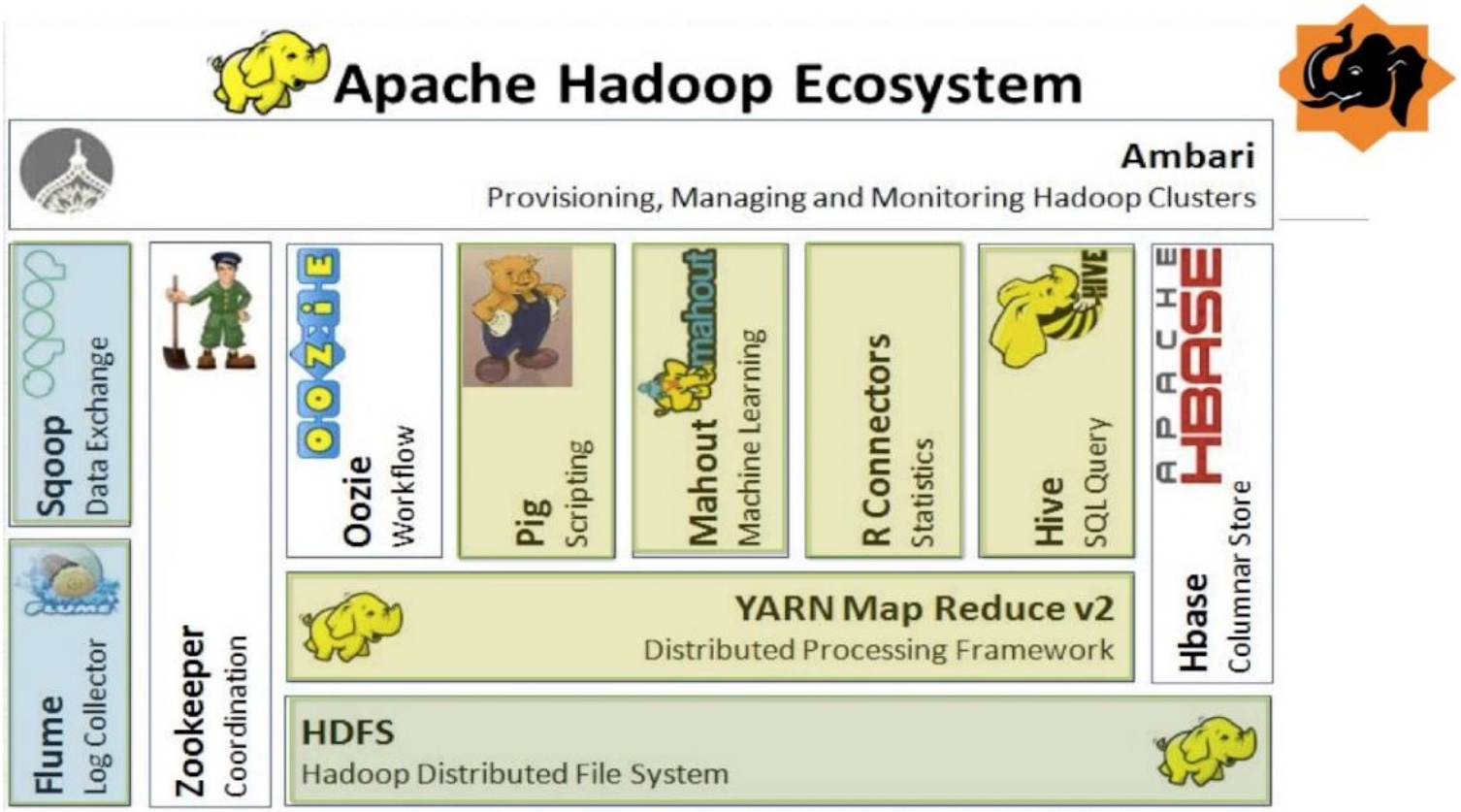
<table>
<tr><td rowspan="3">Windows Processes</td><td>Linux Processes</td><td>Java Threads</td></tr>
<tr><td>Linux</td><td rowspan="2">JVM</td></tr>
<tr><td>VMWare</td></tr>
<tr><td colspan="3">Windows</td></tr>
<tr><td colspan="3">Hardware</td></tr>
</table>

<table>
<tr><td rowspan="3">Mac Processes</td><td>Windows Processes</td><td>Linux Processes</td><td>Java Threads</td></tr>
<tr><td>Windows</td><td>Linux</td><td rowspan="2">JVM</td></tr>
<tr><td>Parallels</td><td>Parallels</td></tr>
<tr><td colspan="3">Mac OS X</td></tr>
<tr><td colspan="3">Hardware</td></tr>
</table>

- Java
  - ✓ JVM executes “platform-neutral byte-codes”
  - ✓ JIT (Just-In-Time) compilers increase performance

Computing Environments

- Distributed computing
  - ✓ Collection of separate, possibly heterogeneous, systems networked together
  - ✓ Network operating systems provides illusion of a single system
  - ✓ E.g.) Hadoop



OS History

A long time ago,
in a galaxy far, far away, …

- IBM OS/360: Multiprogramming
- MIT CTSS (Compatible Time-Sharing System)
- MIT, Bell Labs, GE, MULTICS
  (MULTiplexed Information and Computing Service)

And Unix was born in 1969

OS History: Unix (1969-85)

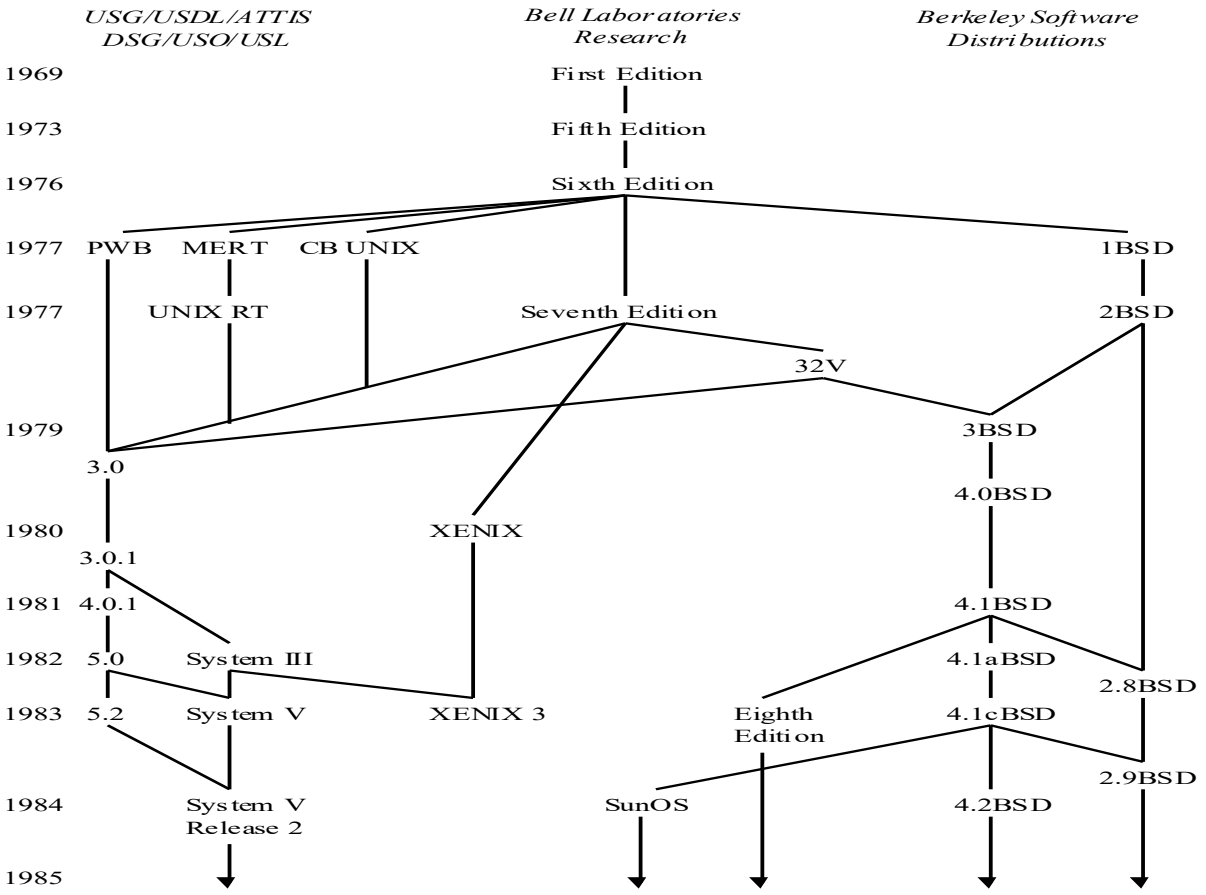


Figure 1.1 The UNIX system family tree, 1969-1985

OS History: Unix (1985-96)

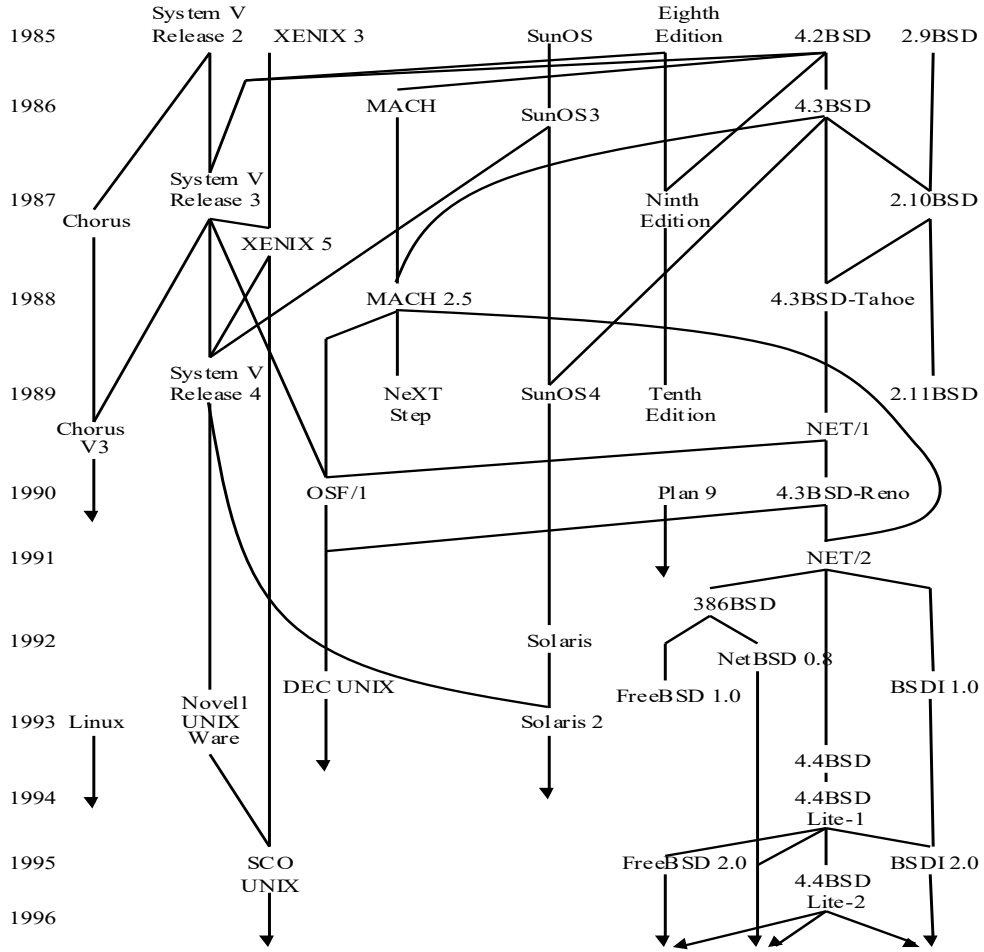


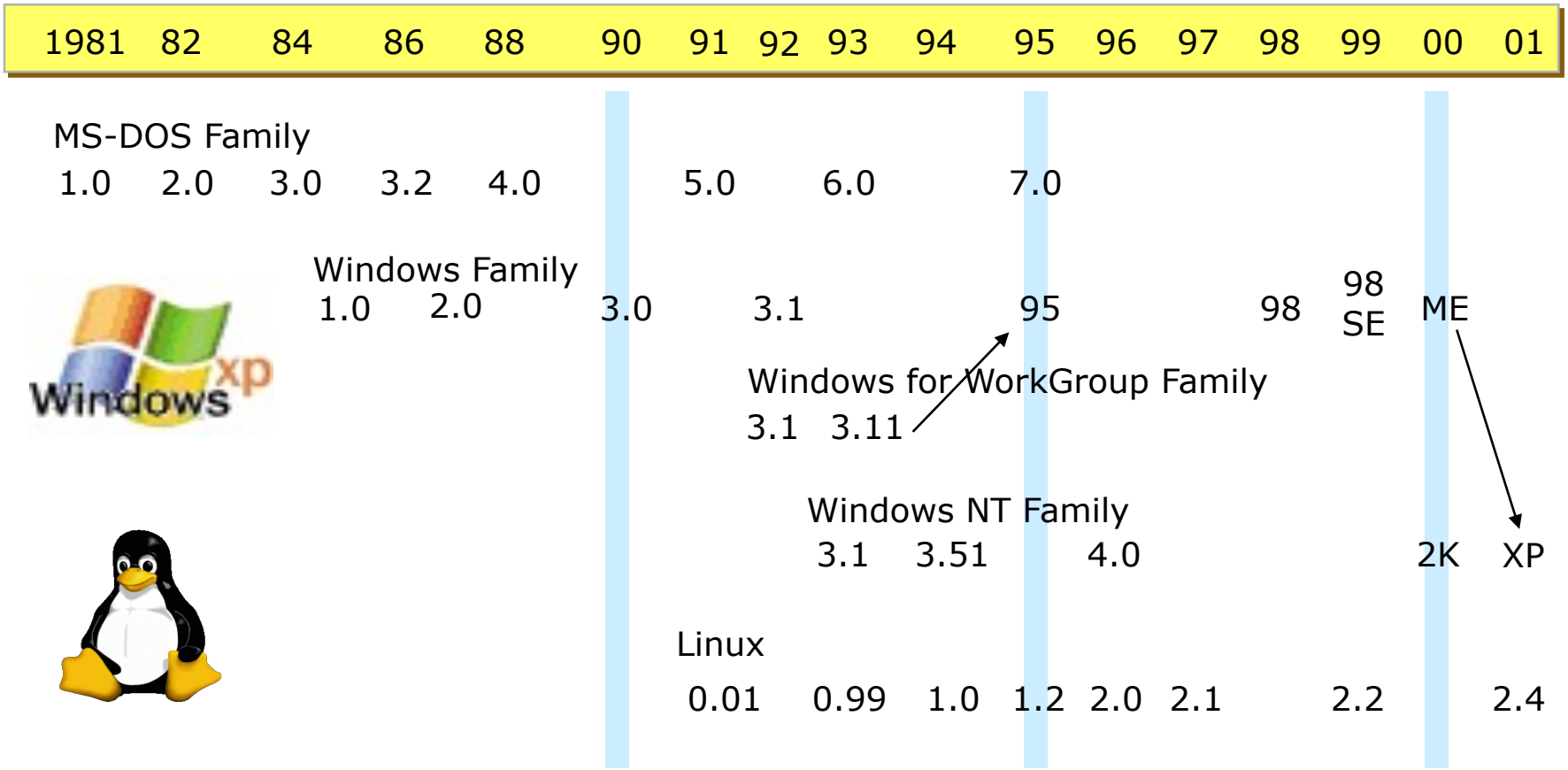
Figure 1.2 The UNIX system family tree, 1986-1996

OS History: Unix (1997-)

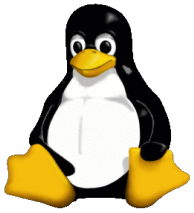
- Sun Solaris
- HP HP-UX
- IBM AIX
- Caldera (SCO) Unixware
- Compaq (Digital) Tru64
- SGI Irix
- Linux, FreeBSD, NetBSD
- Apple Mac OS X, etc.

- Cf) POSIX

OS History: Windows & Linux



OS History: Windows & Linux



04	07	09	10	12	15
2.6.0	A.B.C A: Kernel version B: Major revision C: Minor revision		2.6.33	3.2.0	4.4.0

Windows Vista™

Windows 7

Windows 8

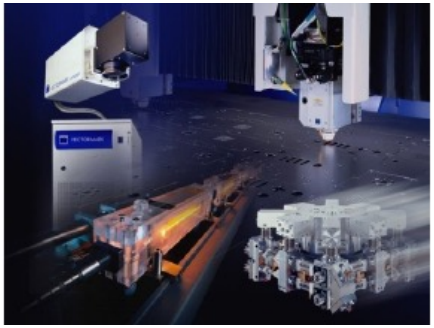
Windows 10

OS History: Linux

- 1983
  - ✓ Richard Stallman, GNU project and free software concept
  - ✓ gcc, gdb, glibc, and other tools
- 1991
  - ✓ Linus Tovalds, Linux kernel project
  - ✓ Completely free operating system: Linux/GNU
- 1995
  - ✓ Linux is more and more popular on server systems
- 2000
  - ✓ Linux is more and more popular on embedded systems
- 2008
  - ✓ Linux is more and more popular on mobile devices
- 2010
  - ✓ Linux is more and more popular on phones

OS History: Linux

- Television
- Personal router
- PoS (Point of Sales) terminal
- Laser cutting machine
- Viticulture machine



OS History: Taxonomy

- Mainframe systems
  - ✓ CTS, MULTICS, IBM MVS, VM
- Desktop systems
  - ✓ DOS, Windows, MacOS, Unix/Linux
- Distributed systems
  - ✓ Amoeba(Vrije Univ.), Locus(UCLA), Grapevine(Xerox), V(Stanford), Eden(U. of Washington), Chorus/Nucleus(Inria)
- Embedded systems
  - ✓ Vertex, pSOS, VxWorks, OSE, Windows-CE, Embedded Linux
  - ✓ Company-proprietary OS (Cisco, Qualcomm, Palm, Cellvic)
- Real-time systems
  - ✓ Real-Time Linux, Spring(U. of Massachusetts), HARTS(U. of Michigan), MARUTI(U. of Maryland)

Thank You!
Q&A