

Operating Systems (Honor Track)

Lecture 4: Von Neumann Architecture

Yao Guo (郭耀)

Peking University Fall 2021

Review: OS History



- □ Generations:
 - Gen1: (1945–55) Vacuum Tubes
 - Gen2: (1955–65) Transistors and Batch Systems
 - Gen3: (1965–1980) ICs and Multiprogramming
 - Gen4: (1980–Present) Personal Computers
 - Gen5: (1990-Present) Mobile Computer
 - Gen6?

This Lecture



Von Neumann Architecture A Peek into Unix/Linux

Buzz Words



Von Neumann architecture

Memory-storage hierarchy

I/O device

Bus

Where is the OS?



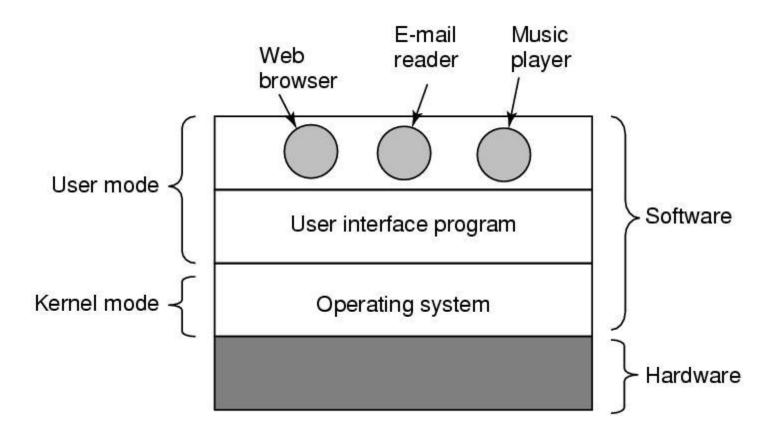


Figure 1-1. Where the operating system fits in.

Why Start With Hardware?



- Operating system functionality depends upon hardware
 - Key goals of an OS are to enforce protection and resource sharing
 - If done well, applications can be oblivious to HW details
- □ Hardware support can greatly simplify or complicate OS tasks
 - Early PC operating systems (DOS, MacOS) lacked virtual memory in part because hardware did not support it

Von Neumann Architecture



■ What is the Von Neumann Architecture?

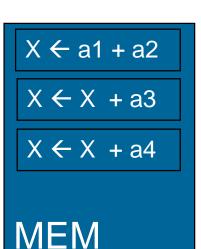


EDSAC, May 1949

$$X \leftarrow \sum_{i=1}^{N} a_i$$

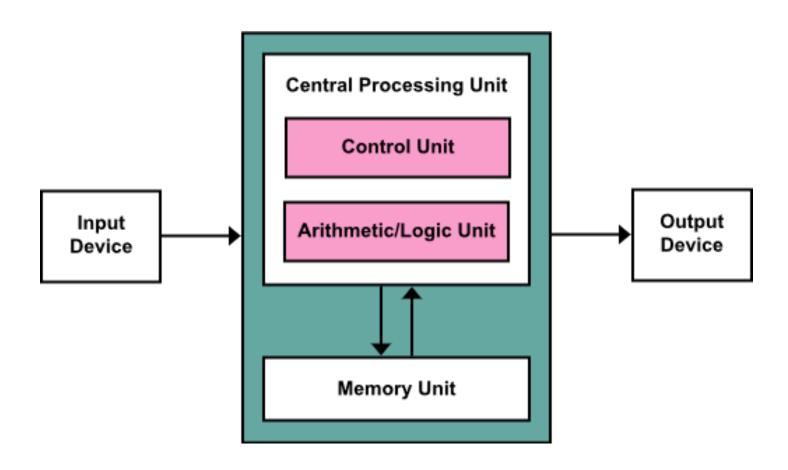
Concepts

- Instructions
- Program counter



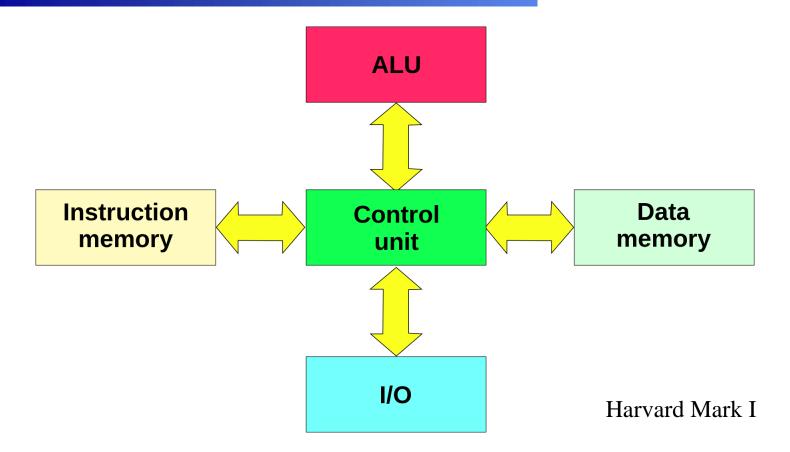
Von Neumann Architecture





vs. Harvard Architecture





Other non von Neumann architectures:

• DSP processors (systolic arrays), dataflow architecture

Von Neumann Bottleneck



□ John Backus: 1977 ACM Turing Award lecture

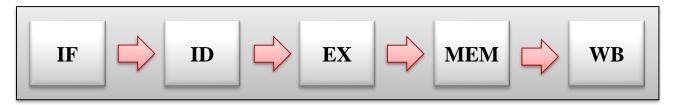
"Surely there must be a less primitive way of making big changes in the store than by pushing vast numbers of words back and forth through the **von Neumann bottleneck**.

Not only is this tube a literal bottleneck for the data traffic of a problem, but, more importantly, it is an intellectual bottleneck that has kept us tied to word-at-a-time thinking instead of encouraging us to think in terms of the larger conceptual units of the task at hand.

Thus programming is basically planning and detailing the enormous traffic of words through the von Neumann bottleneck, and much of that traffic concerns not significant data itself, but where to find it."

Inside a Von Neumann Arch. Process

- □ Instruction Set Architecture (ISA)
- General registers
- Program counter (PC)
- □ Program status word (PSW)
- □ Stages



Context switch

Memory-Storage Hierarchy



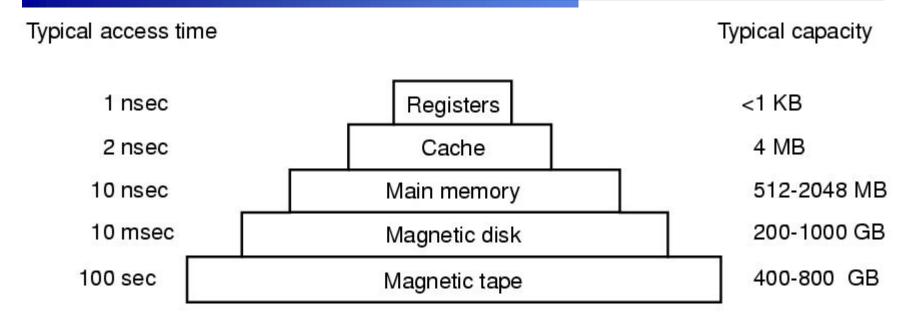


Figure 1-9. A typical memory hierarchy.

The numbers are very rough approximations.

Why such complicated hierarchy?

Multicore Chips



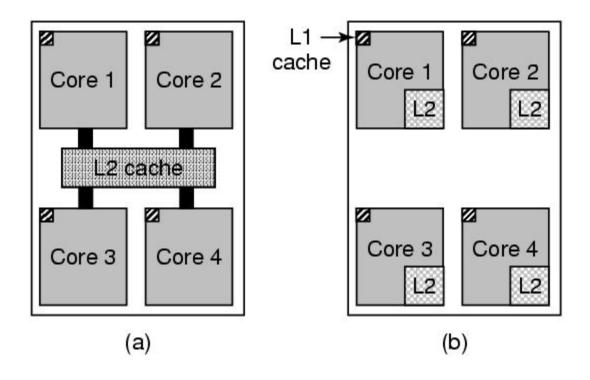


Figure 1-8. (a) A quad-core chip with a shared L2 cache. (b) A quad-core chip with separate L2 caches.

Disks



Cylinder, head, sector (CHS), track, ...

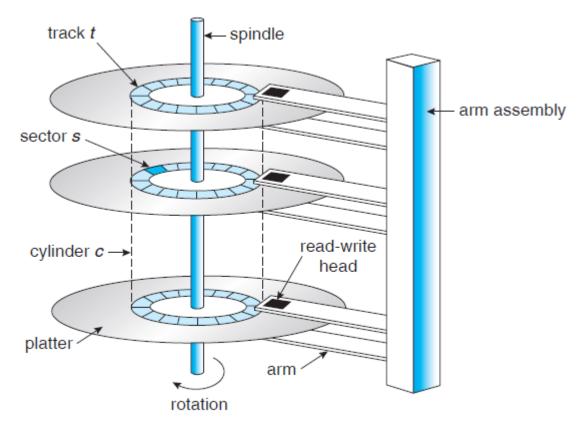


Figure 11.1 Moving-head disk mechanism.

I/O Devices



- Registers
 - Status
 - Control
 - Data-in
 - Data-out
- □ Polling vs. interrupts
- Programmed I/O (PIO) vs. Direct-memory access (DMA)
- □ How do programmers/users deal with all these details?
 - Device drivers

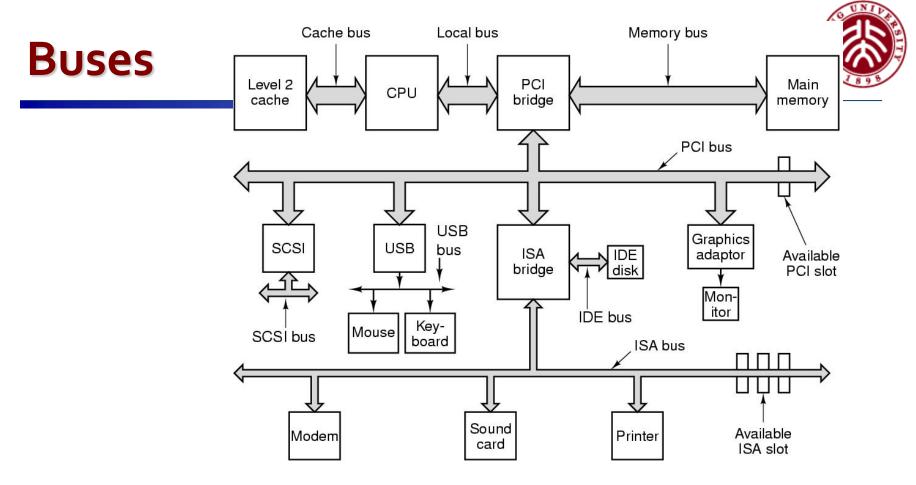


Figure 1-12. The structure of a large Pentium system

- Why do we need different buses?
- □ What are the popular buses today?



Booting the Computer



- BIOS is started
 - Checks if the hardware functions well
 - Detects all the devices attached
 - Determines the boot device/disk and loads its first sector
- The first sector is read into memory and executed
 - Determines the active partition of the device/disk
 - Loads the second boot loader from the active partition
- The second boot loader loads the OS and starts it

You will learn the details through Lab 1!

This Lecture



Von Neumann Architecture A Peek into Unix/Linux

A Short History of Unix



- Summer 1969: Unix was developed.
 - Linus Torvalds was born on December 28, 1969.
- First edition of Unix released on Nov 3, 1971.
 - The first edition of the "Unix PROGRAMMER'S MANUAL by Thompson & Ritchie. ", includes over 60 commands
 - b (compile B program); boot (reboot system); cat (concatenate files); chdir (change working directory); chmod (change access mode); chown (change owner); cp (copy file); ls (list directory contents); mv (move or rename file); roff (run off text); wc (get word count); who (who is on the system).
 - The main thing missing was pipes.

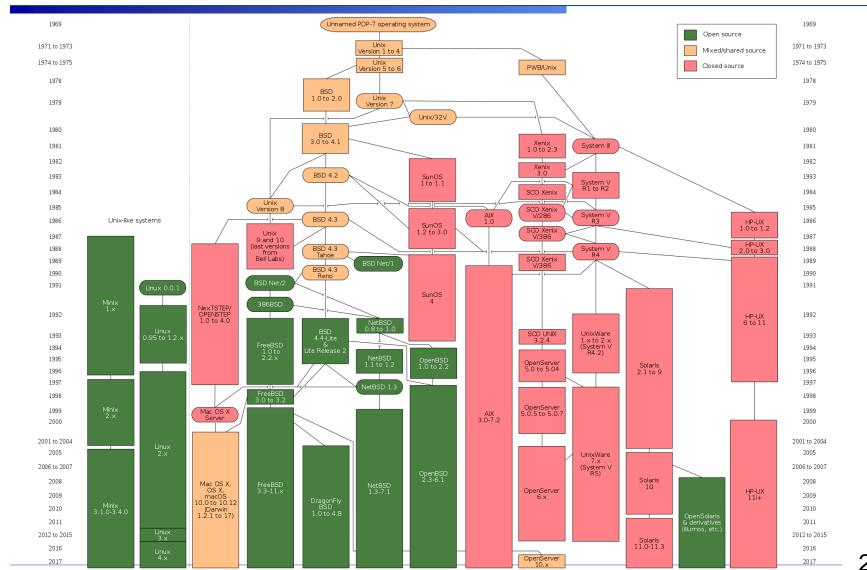
Pipeline in Unix



- The pipeline concept was invented by Douglas McIlroy, his ideas were implemented in 1973 when ("in one feverish night", wrote McIlroy)
- Ken Thompson added the pipe() system call and pipes to the shell and several utilities in Version 3 Unix.
- "The next day", McIlroy continued, "saw an unforgettable orgy of one-liners as everybody joined in the excitement of plumbing."
- □ Thompson invented the | notation (from Unix V4)

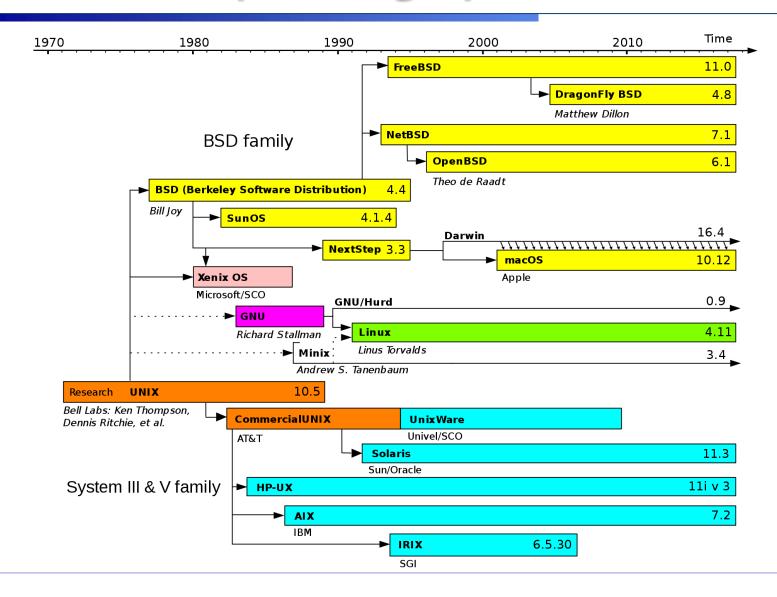
Development of Unix & Beyond





Unix-like Operating Systems





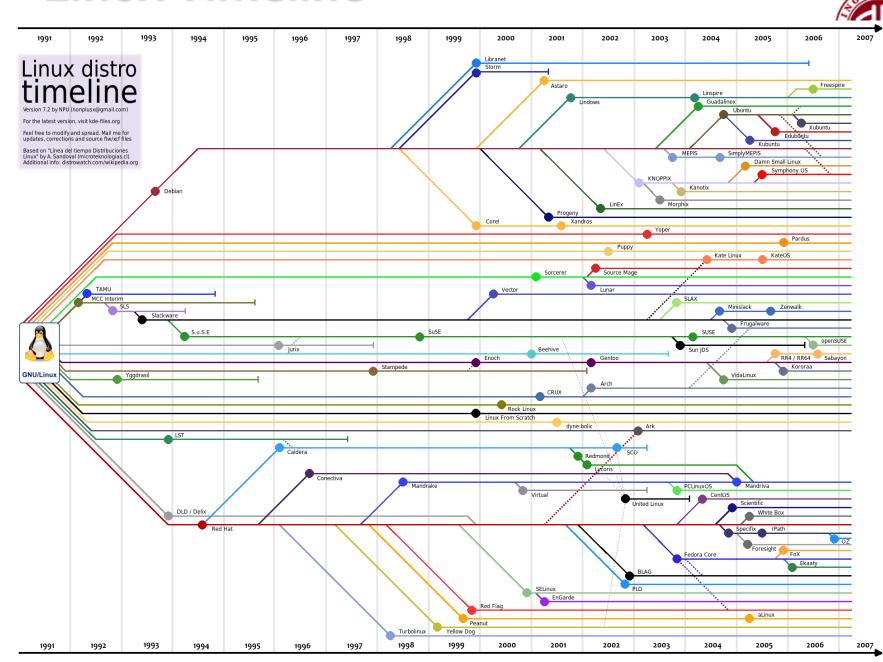
Linux is not Unix



- ☐ GNU: GNU is Not Unix
- ☐ Linux: Linux is not Unix

- ☐ Linux is "Unix-like"
 - Or "Unix-compatible"
 - But free and open-source

Linux Timeline



A Peek into Unix/Linux



Application

Libraries

User space/level

Portable OS Layer

Machine-dependent layer

Kernel space/level

Unix: Application



Application (E.g., emacs)

Libraries

Written by programmers
Compiled by programmers
Uses function calls

Portable OS Layer

Machine-dependent layer

Unix: Libraries



Application

Libraries (e.g., stdio.h)

Portable OS Layer

Machine-dependent layer

Written by elves
Provided pre-compiled
Defined in headers
Input to linker (compiler)
Invoked like functions
May be "resolved" when
program is loaded

Typical Unix OS Structure



Application

Libraries

Portable OS Layer

Machine-dependent layer

system calls (read, open..)
All "high-level" code

Typical Unix OS Structure



Application

Libraries

Portable OS Layer

Machine-dependent layer

System initialization
Interrupt and exception
I/O device driver
Memory management
Kernel/user mode
switching
Processor management

Paper Reading (for all)



- □ The Beginning
 - Dijkstra: The Structure of the "THE" Multiprogramming System
- □ Early Virtual Memory
 - Bensoussan: The Multics Virtual Memory: Concepts and Design
 - (Daley: Virtual Memory, Processes, and Sharing in MULTICS)
- OS Kernel Design
 - Engler: Exokernel: An Operating System Architecture for Application-Level Resource Management
- ☐ File System
 - Rosenblum: The Design and Implementation of a Log-Structured
 File System
- New OS Design
 - LegoOS: A Disseminated, Distributed OS for Hardware Resource Disaggregation

Paper Reading Requirements



- Submit a write-up for each paper/topic, including
 - What kind of paper is this?
 - Describe the motivation of the paper
 - List its benefits over the "state-of-the-art"
 - List its drawbacks or potential problems
 - What do you think of the paper from today's opinion?
- Paper presentation
 - A group of three students

Summary



- Von Neumann architecture
- □ A peek into Unix/Linux
 - Video 1: AT&T Archives: The UNIX Operating System, https://www.youtube.com/watch?v=tc4ROCJYbm0
 - Video 2: The mind behind Linux, https://www.youtube.com/watch/o8NPllzkFhE
- Paper reading assignment:
 - P1: "THE" Multiprogramming System (Dijkstra)
 - Report due: 10/17
- Next lecture: Hardware Features for OS