Operating Systems (Honor Track)

Lecture 4: Von Neumann Architecture

Yao Guo (郭耀)

Peking University

Fall 2024

Acknowledgements: Prof. Xiangqun Chen & Tao Wang at PKU and Prof. Yuanyuan Zhou at UCSD

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?

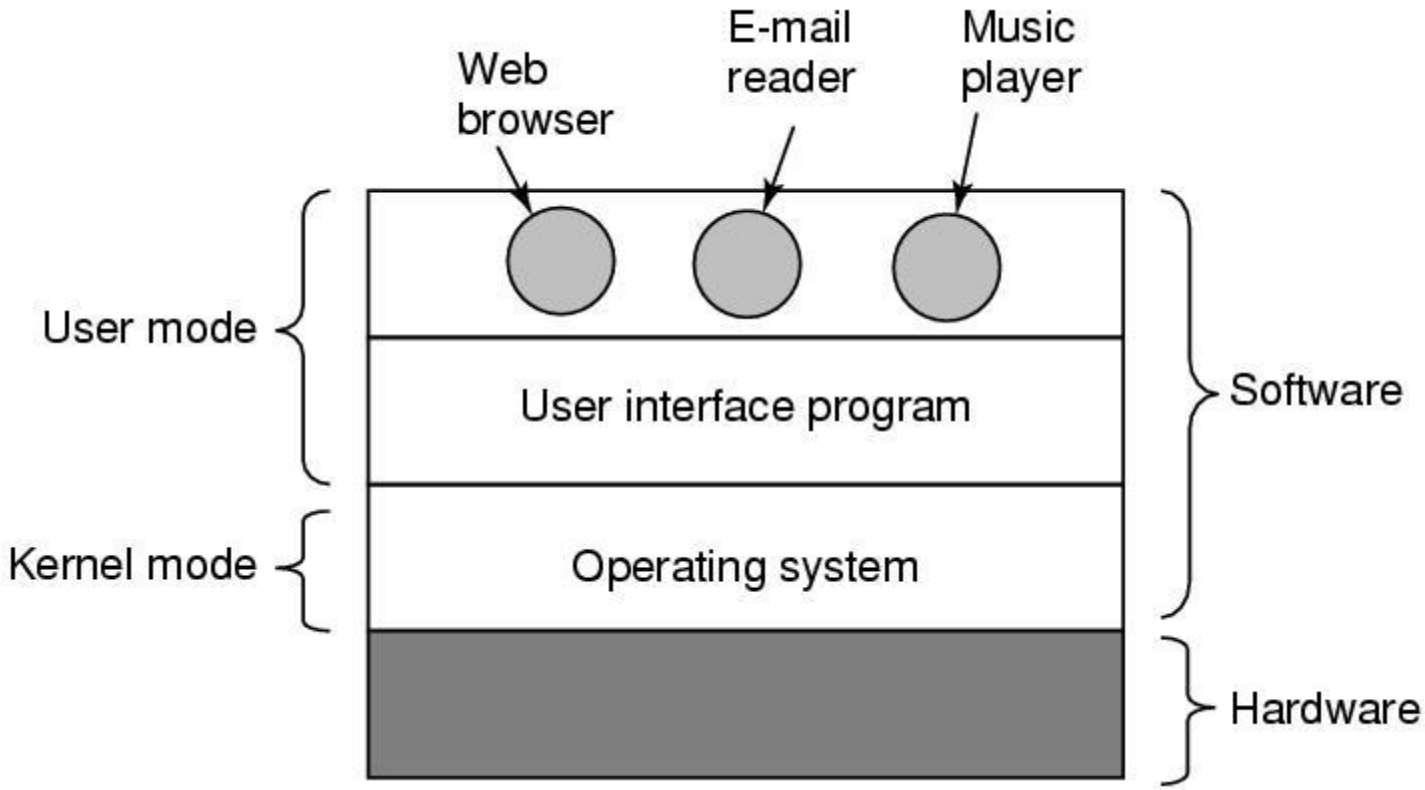


Figure 1-1. Where the operating system fits in.

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

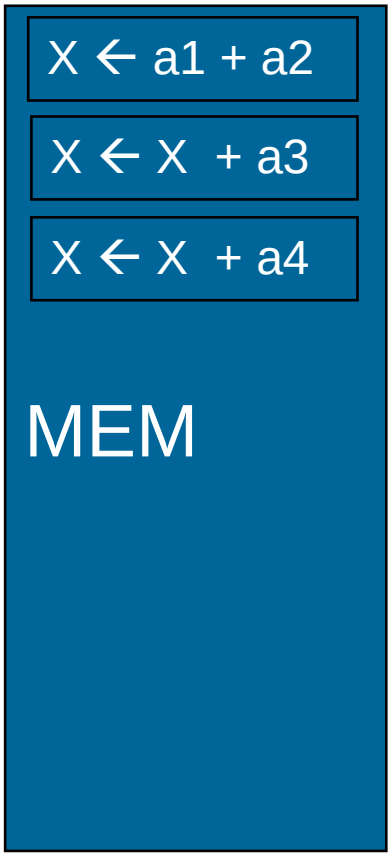
- What is the Von Neumann Architecture?

EDSAC, May 1949

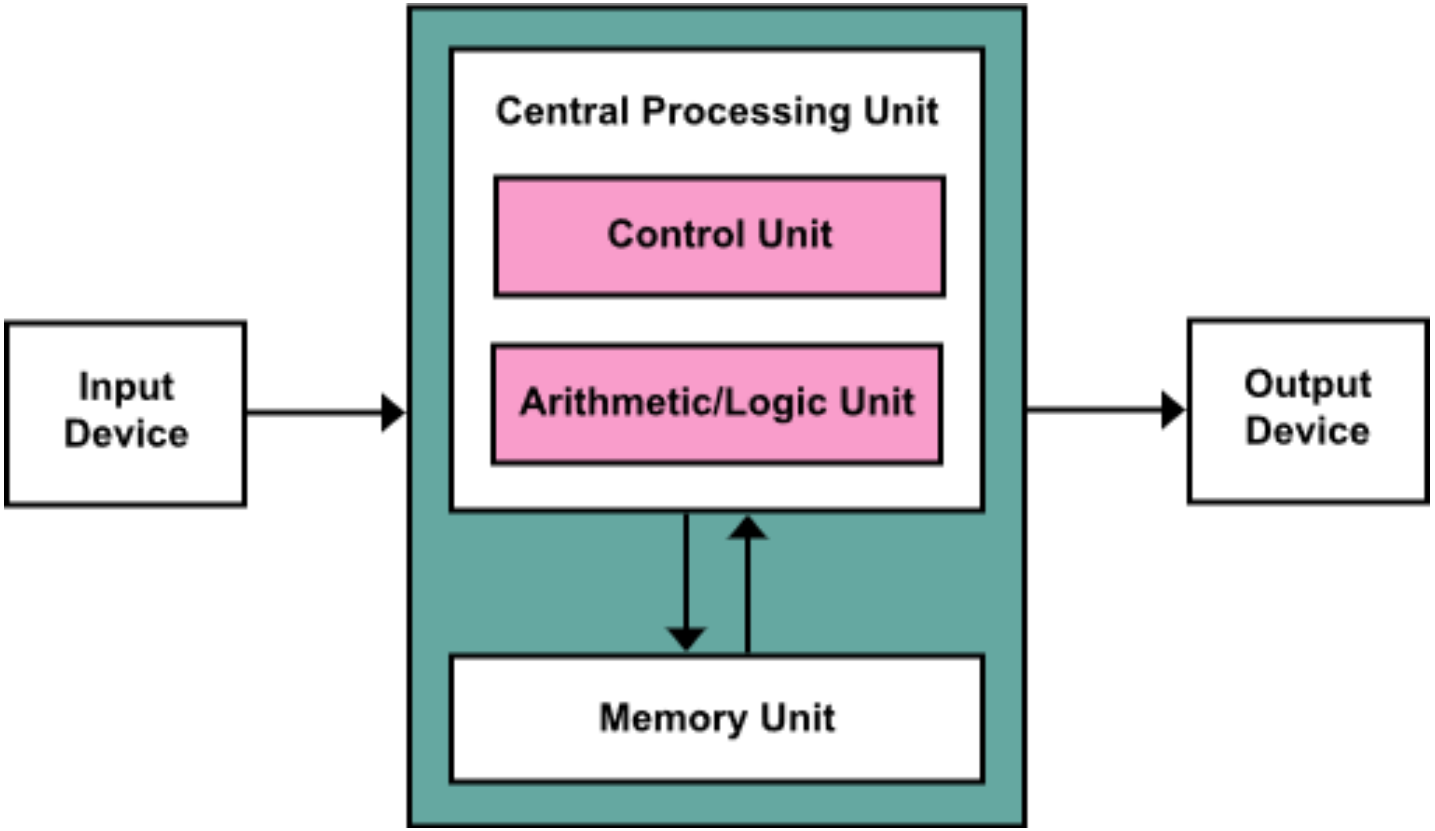
$$X \leftarrow \sum_{i=1}^{N} a_i$$

Concepts

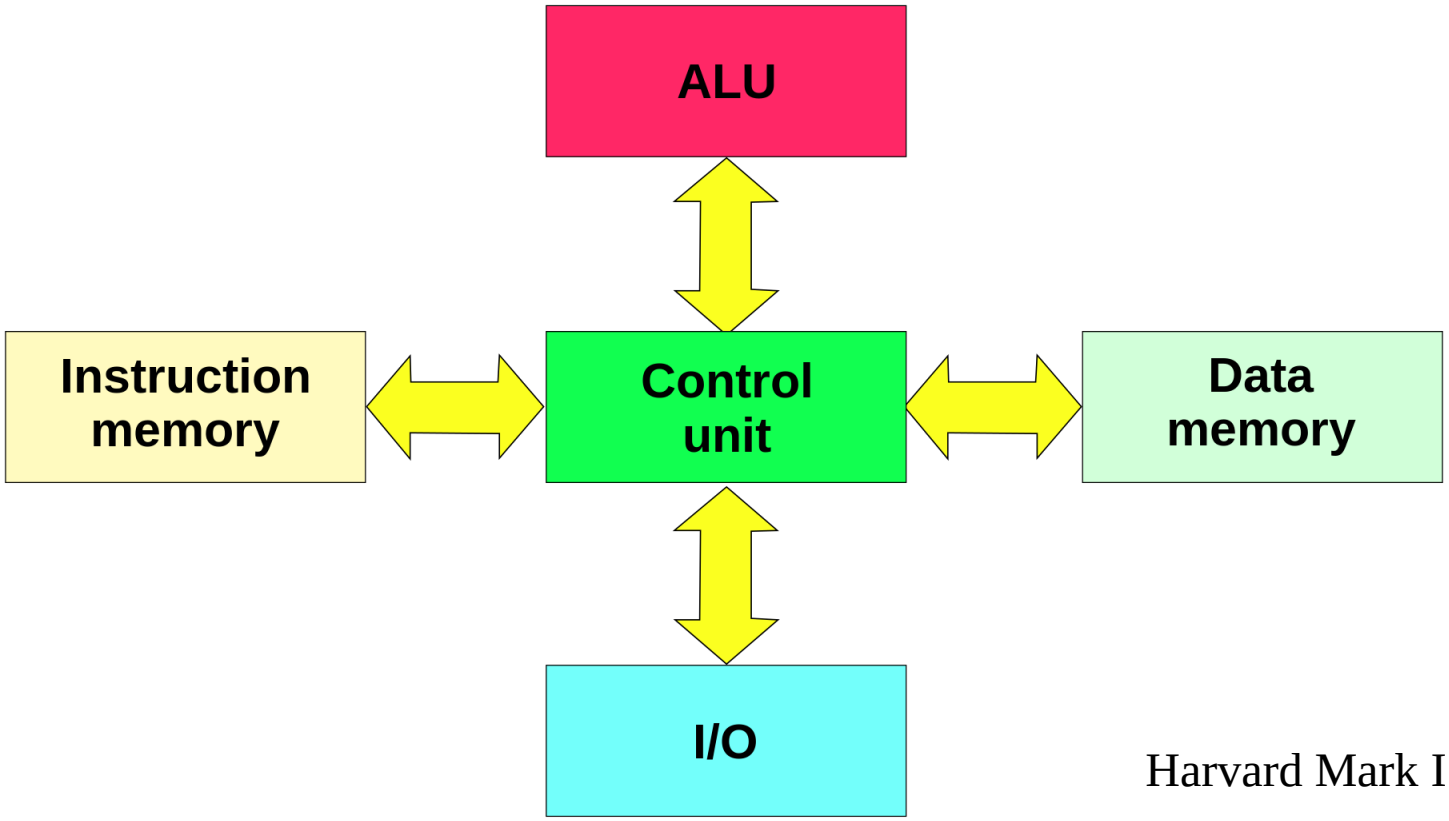
- Instructions
- Program counter



Von Neumann Architecture



vs. Harvard Architecture



Harvard Mark I

Other non von Neumann architectures:

- DSP processors (systolic arrays), dataflow architecture

Von Neumann Bottleneck

- **John Backus: 1977 ACM Turing Award lecture**

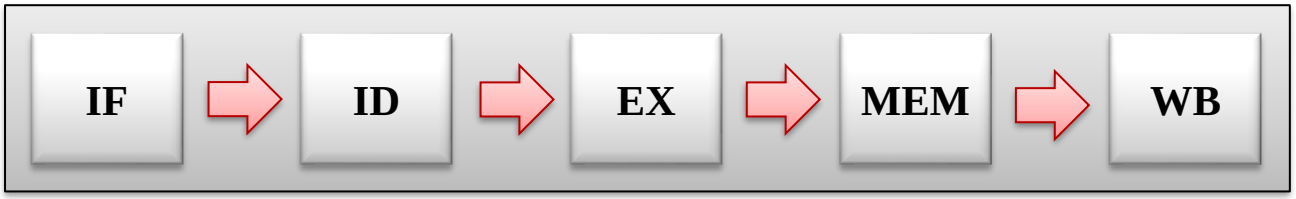
"Surely there must be a less primitive way of making big changes in the store than by pushing vast numbers of words back and forth through the **von Neumann bottleneck**.

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. Processor

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



- Context switch

Memory-Storage Hierarchy

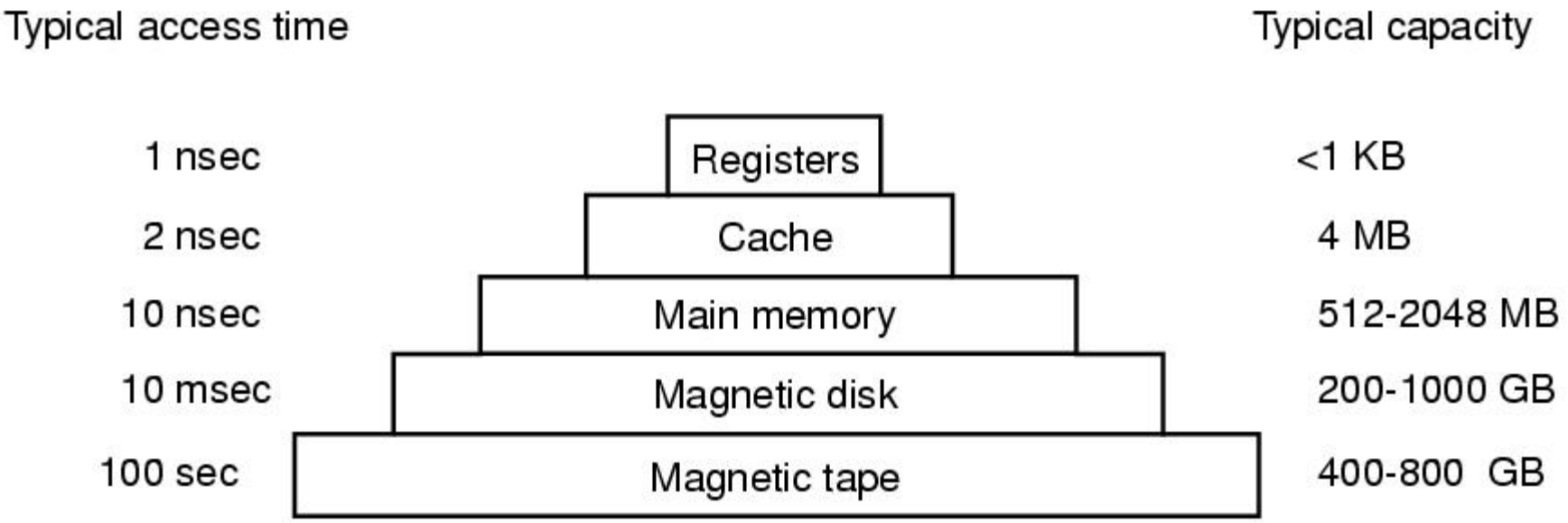


Figure 1-9. A typical memory hierarchy.
The numbers are very rough approximations.

Why such complicated hierarchy?

Multicore Chips

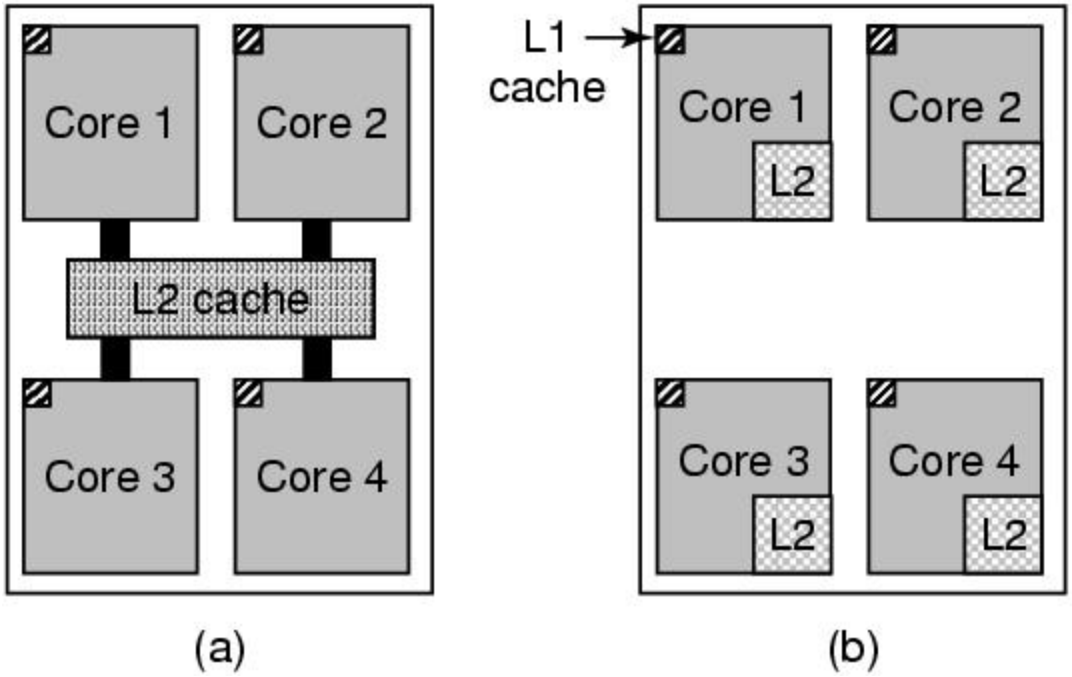


Figure 1-8. (a) A quad-core chip with a shared L2 cache. (b) A quad-core chip with separate L2 caches.

Disks

- Cylinder, head, sector (CHS), track, …

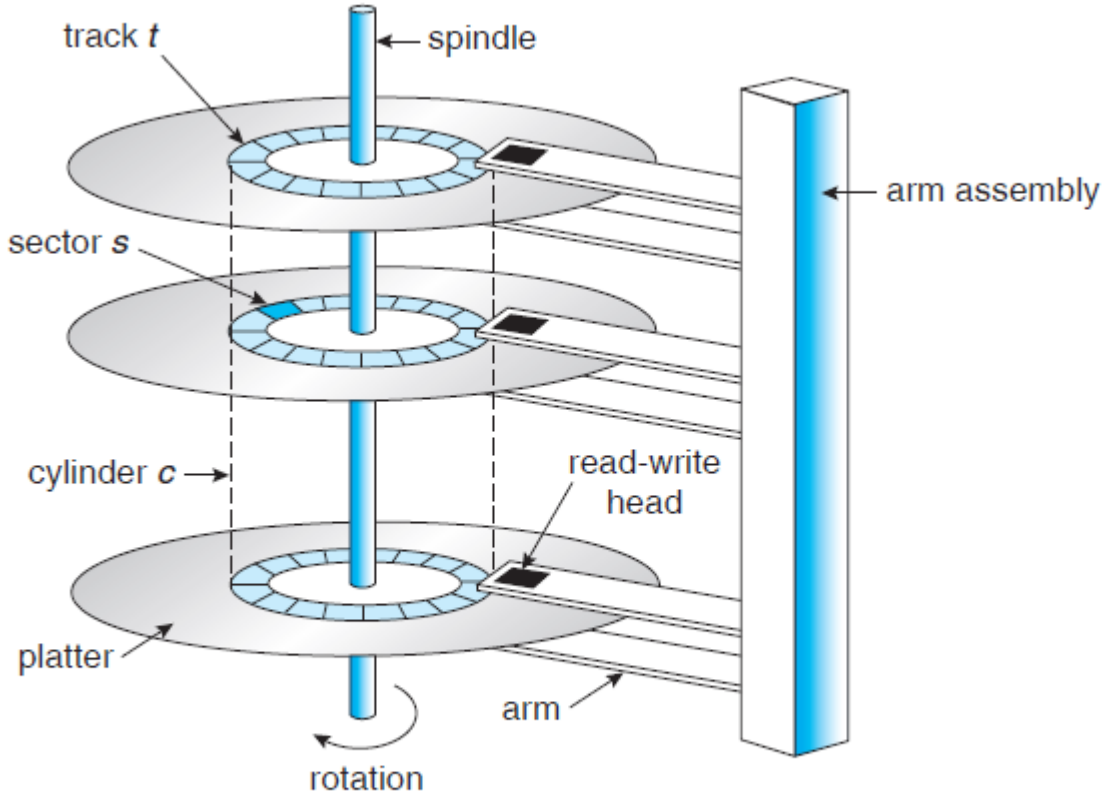


Figure 11.1 Moving-head disk mechanism.

Figure from "Operating System Concepts Essentials", A. Silberschatz et at., John Wiley & Sons. Inc

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

Buses

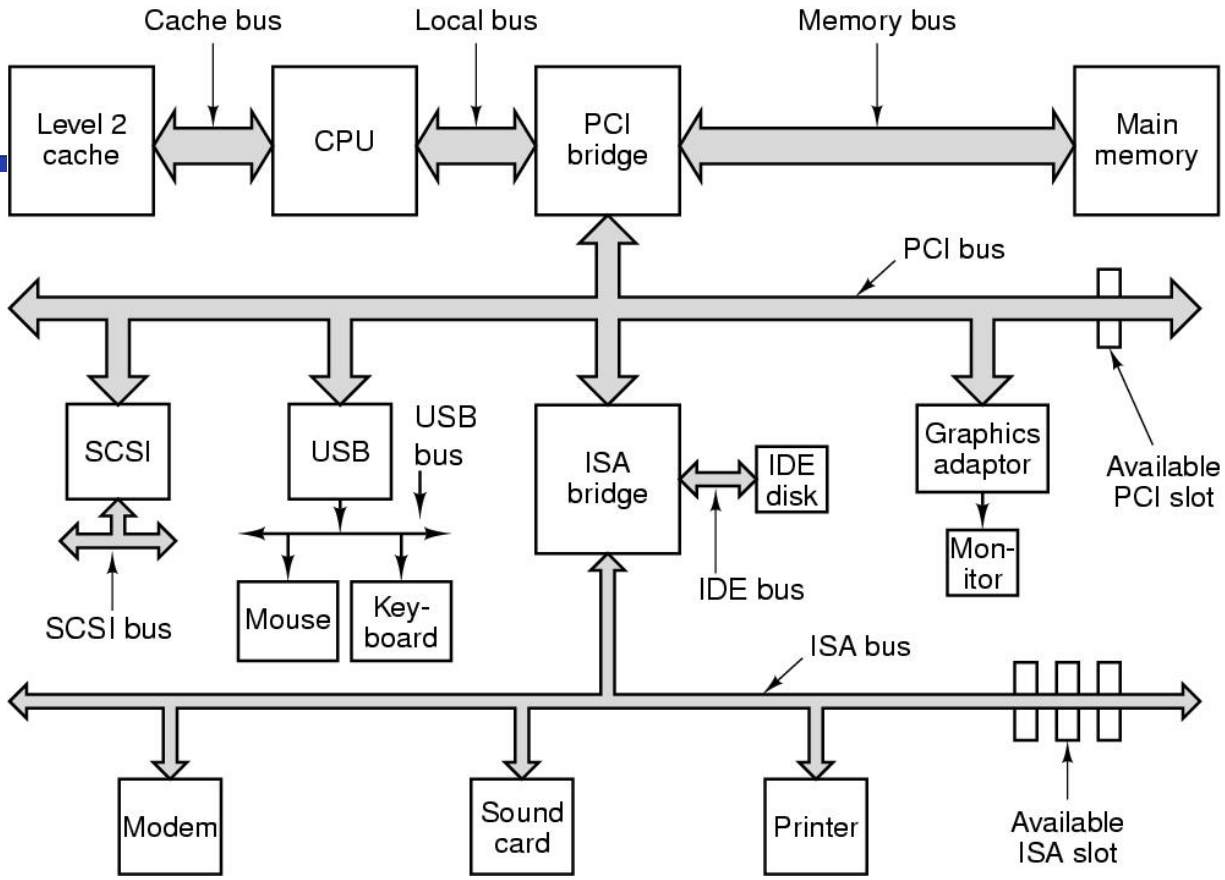


Figure 1-12. The structure of a large Pentium system

- 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

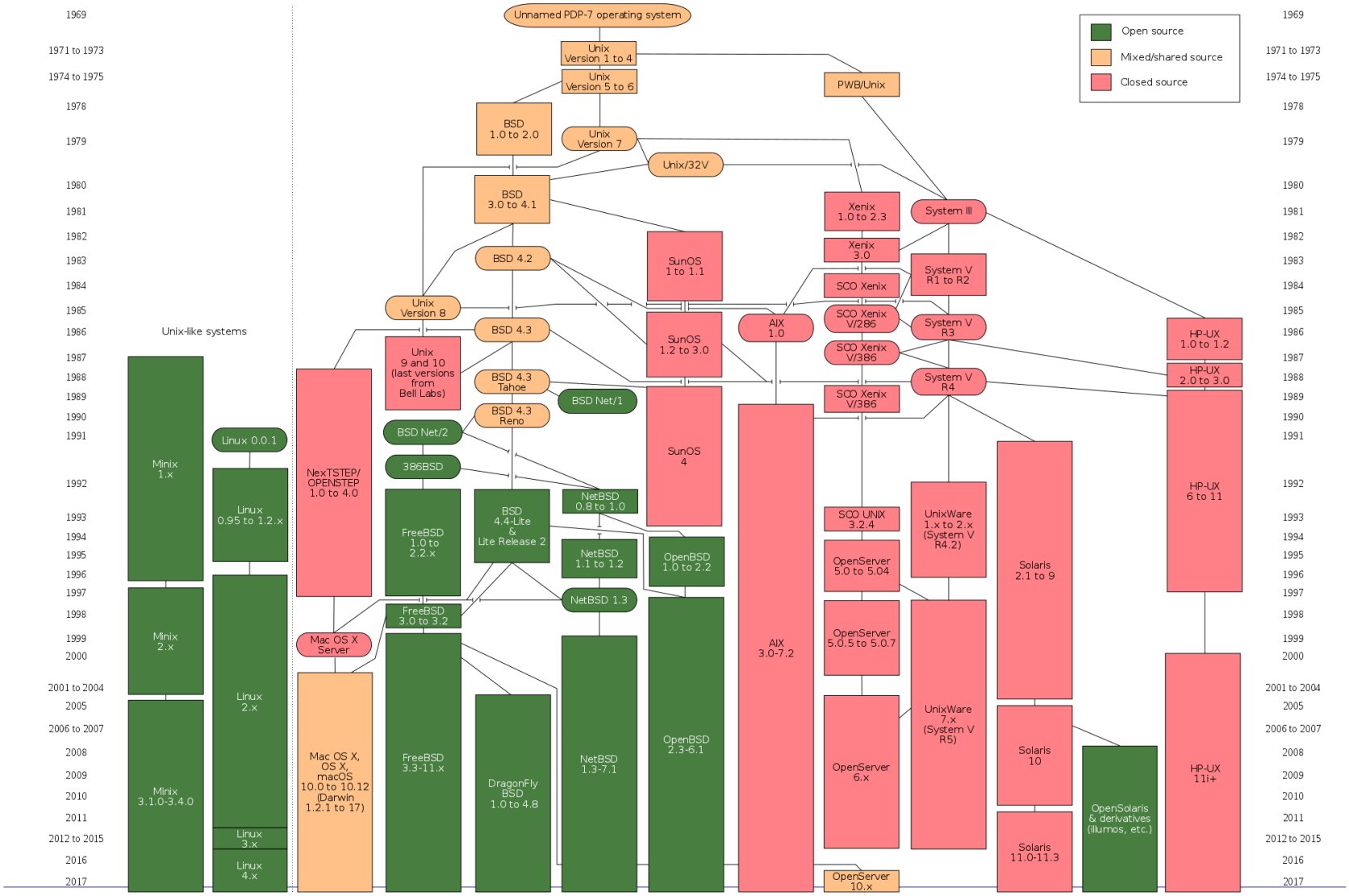
A Short History of Unix

- 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.

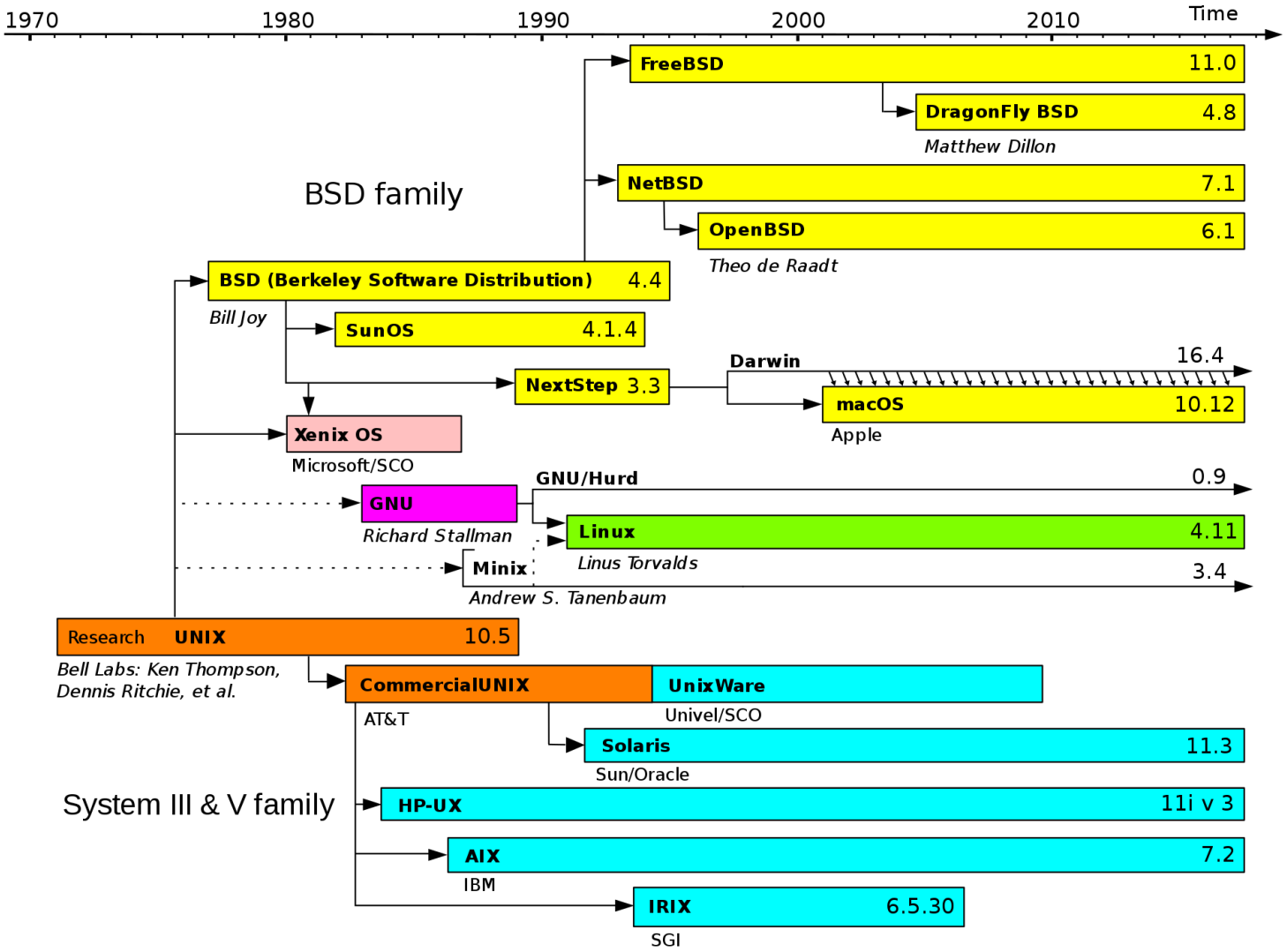
Pipeline in Unix

- 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

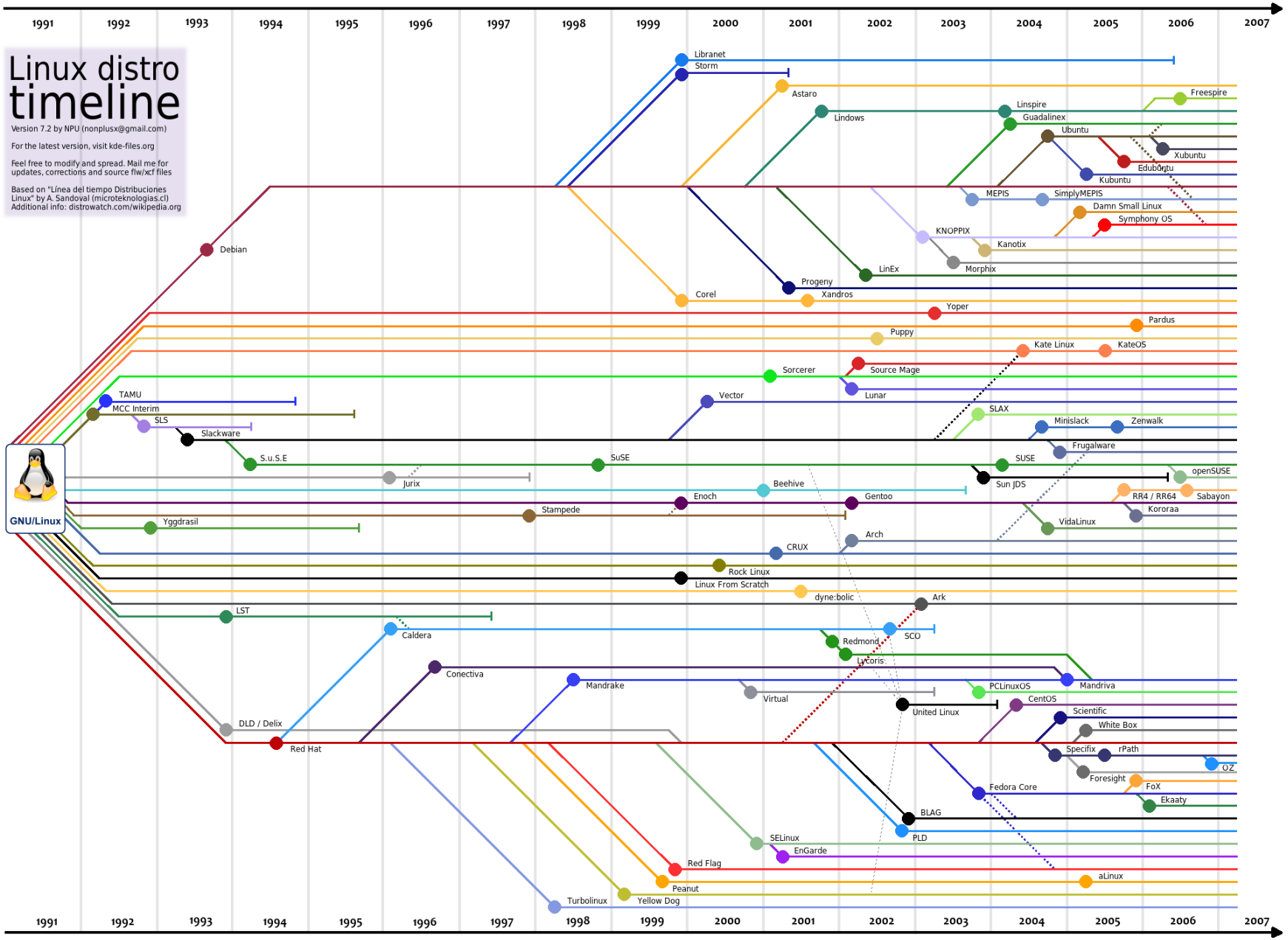


Linux is not Unix

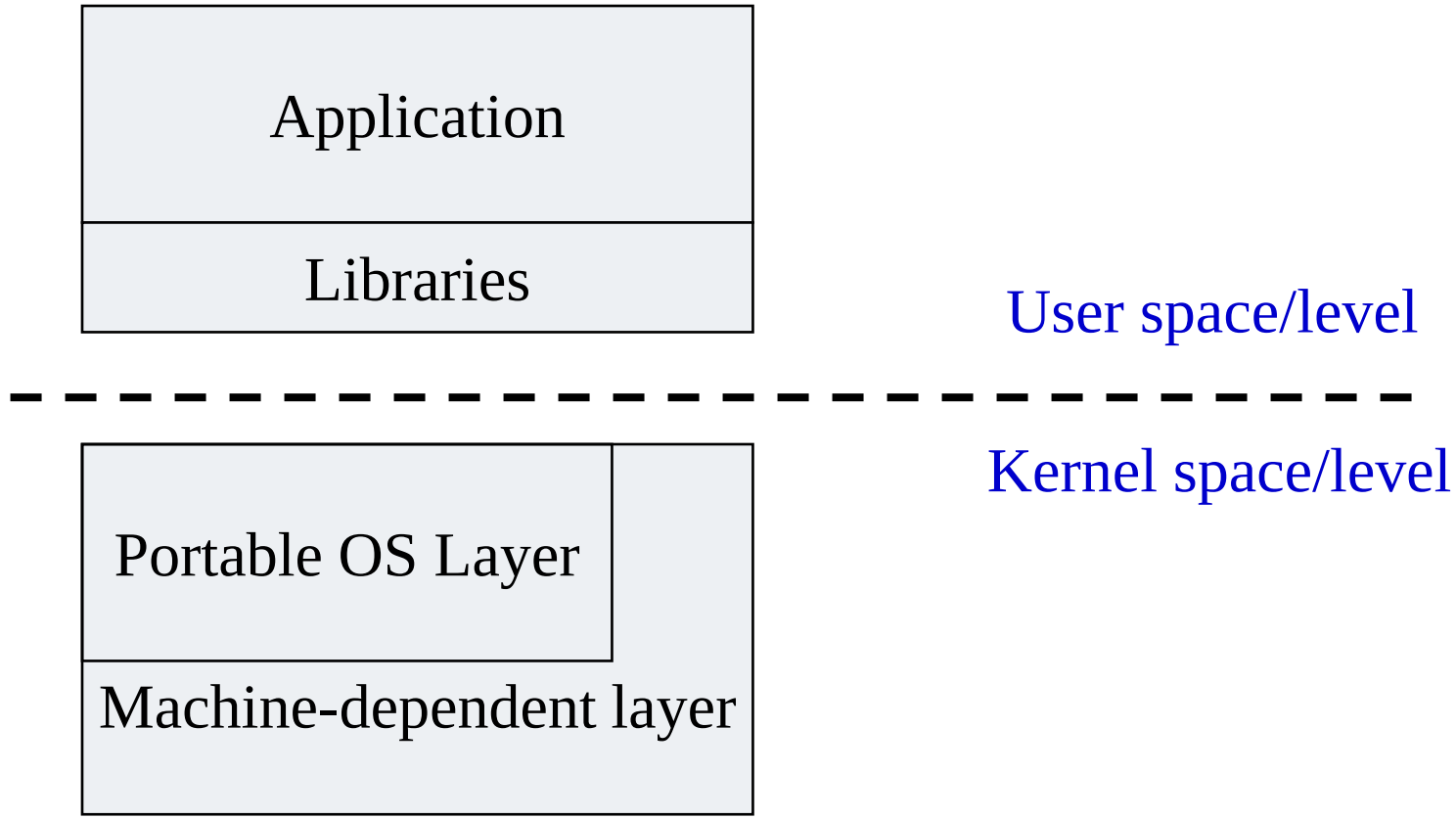
- GNU: GNU is Not Unix
- Linux: Linux is not Unix

- Linux is “Unix-like”
  - Or “Unix-compatible”
  - But free and open-source

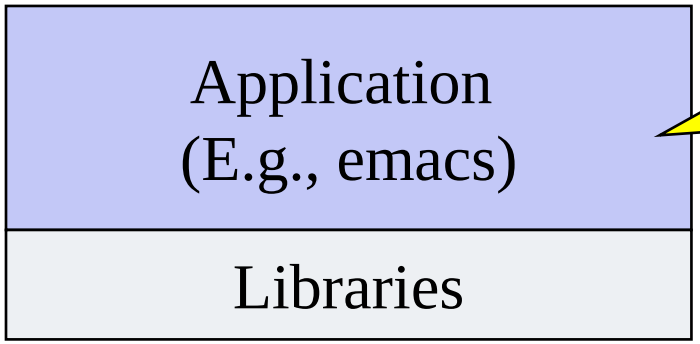
Linux Timeline



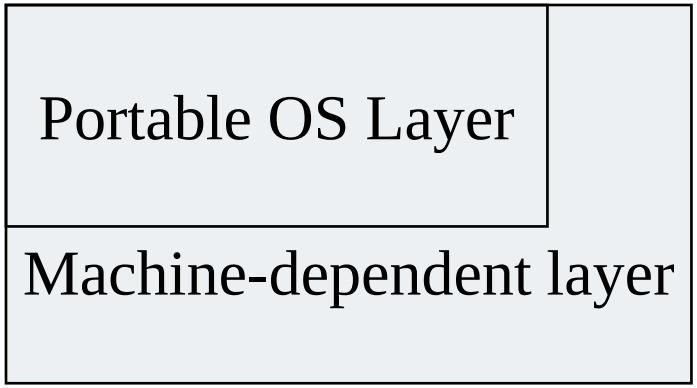
A Peek into Unix/Linux



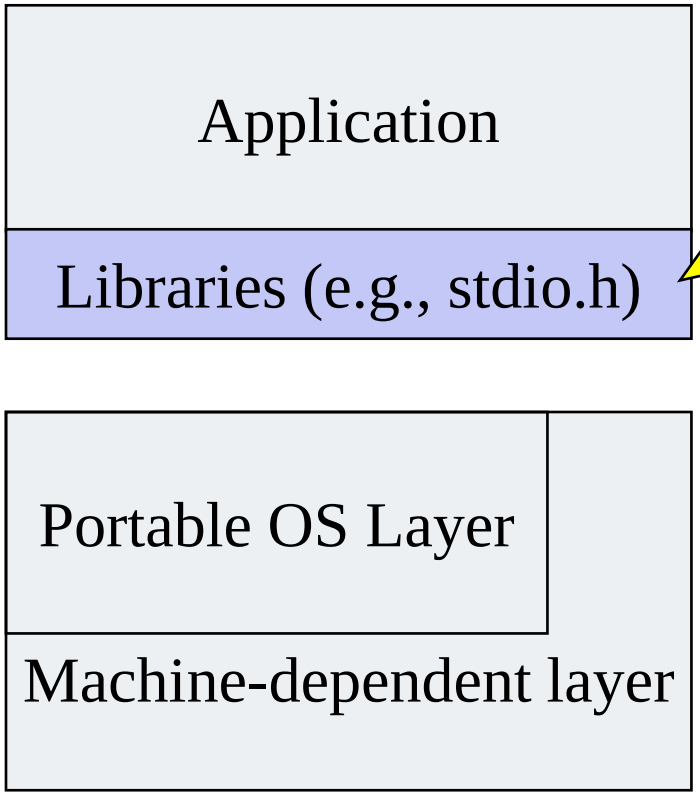
Unix: Application



Written by programmers
Compiled by programmers
Uses function calls

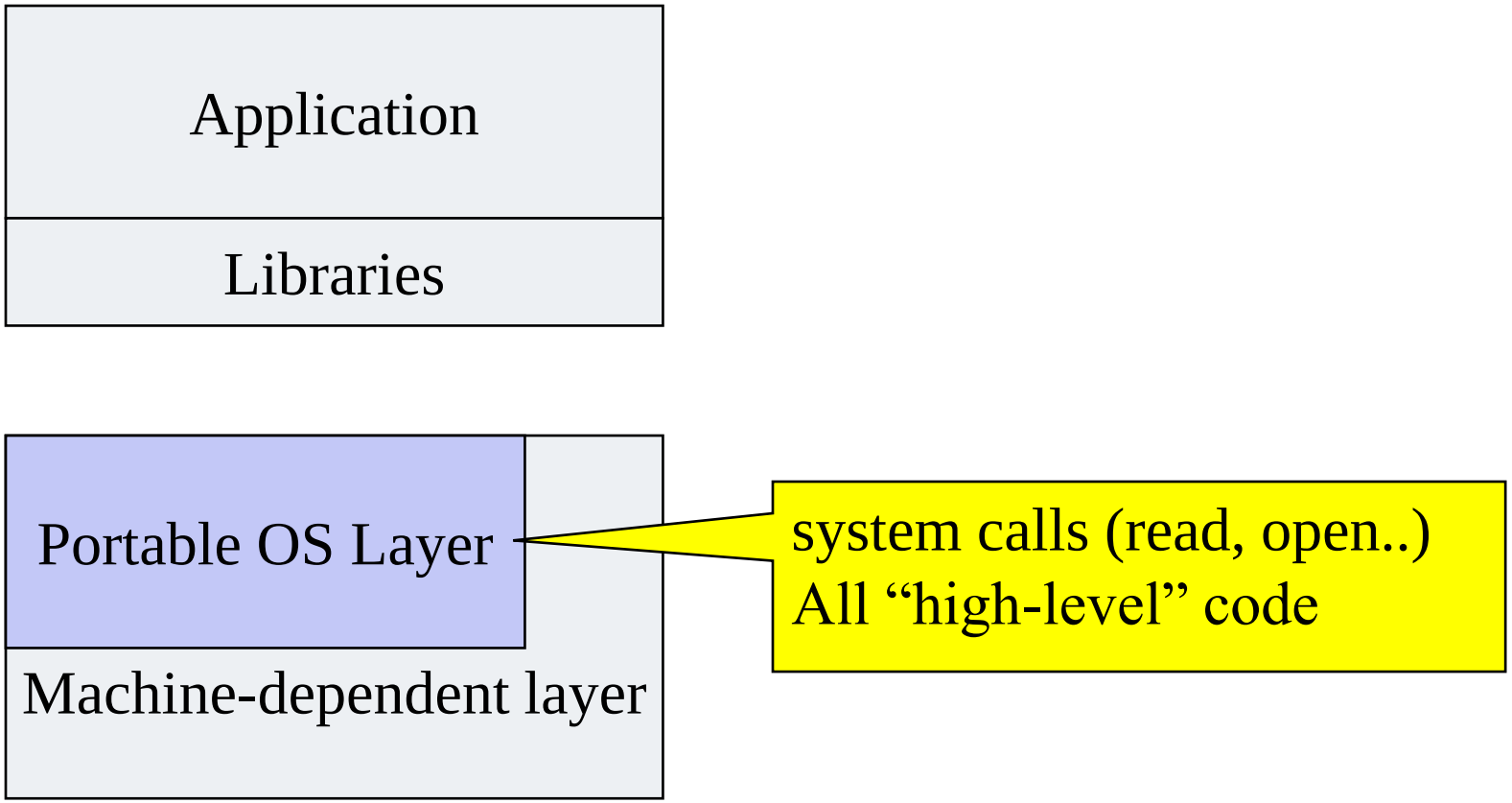


Unix: Libraries

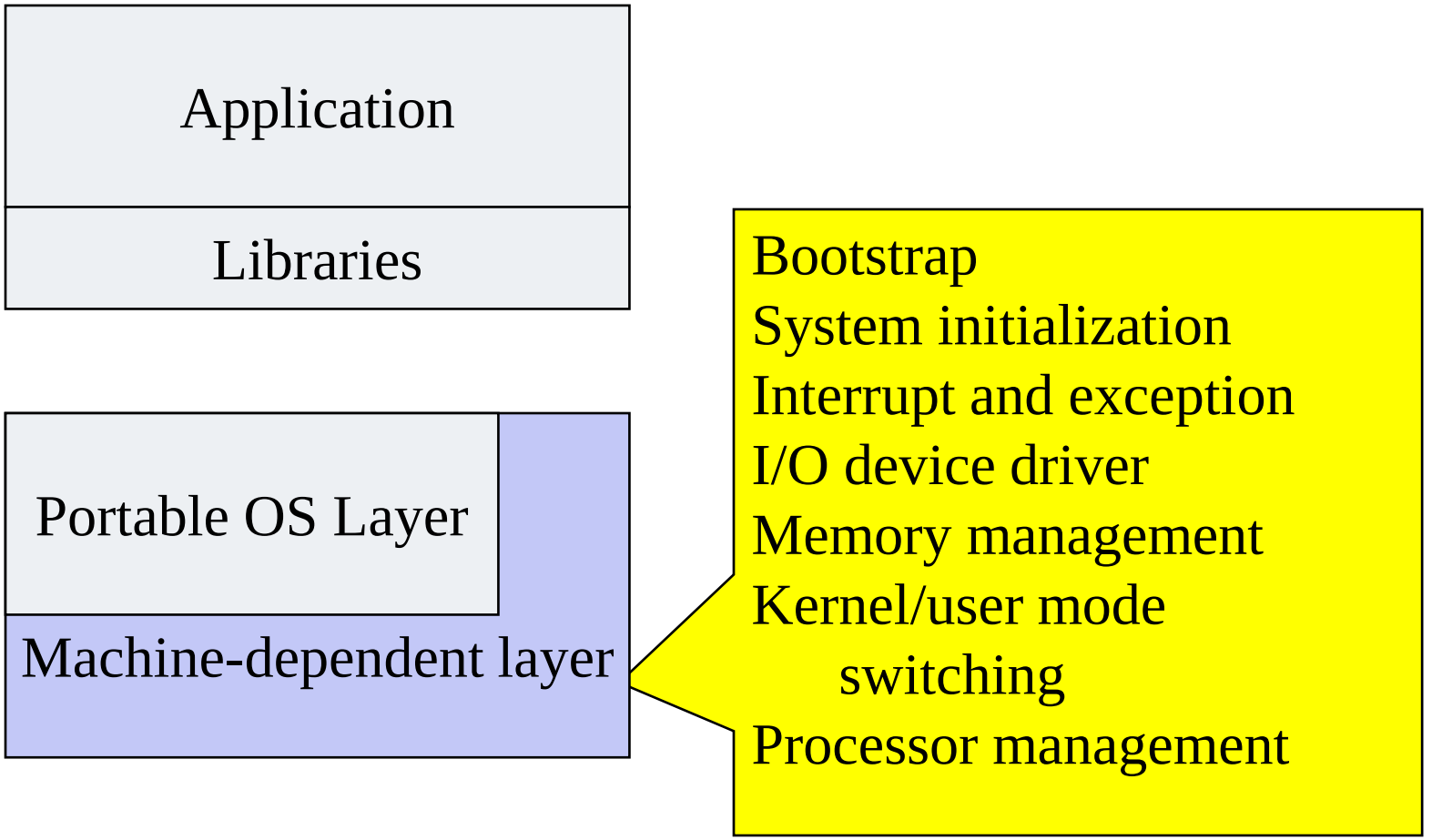


Written by elves
Provided pre-compiled
Defined in headers
Input to linker (compiler)
Invoked like functions
May be “resolved” when program is loaded

Typical Unix OS Structure



Typical Unix OS Structure



OS Survey & Paper Reading

- Paper Reading
  - I will provide a list of papers (~20).
  - You are expected to read them through, and
  - Write reviews for 5 Papers.
- Seminar: Paper Presentation
  - Each paper will be presented by 2 students
  - Around 4 papers each class (20 min each paper)
- Seminar: OS Survey
  - Each group will conduct survey on a set of old/new OS, and
  - Present them in class.
  - About 15 minutes for each group of 2 students

OS Surveys

- Find the information (paper, product, website, or propose one yourself) of a new/old OS.
  - Write a short presentation (< 10 pages)
    - What are its main functionalities?
    - What are the new (or special) features?
    - Comparing it with traditional OSes such as Windows/Linux
      - What are the main differences?
      - What are the similarities?
    - Why did it succeed/fail?

Paper Reading List (1)

- The Beginning
  - Dijkstra: The Structure of the "THE" Multiprogramming System
  - Corbató: An Experimental Time-Sharing System
- Virtual Memory
  - Kilburn: One Level Storage System
  - Bensoussan: The Multics Virtual Memory: Concepts and Design
- Virtualization
  - Barham: Xen and the Art of Virtualization
  - Kivity: OSv—Optimizing the Operating System for Virtual Machines
  - Soltesz: Container-based Operating System Virtualization
  - Agache: Firecracker: Lightweight Virtualization for Serverless Applications

Paper Reading List (2)

- OS Kernel Design
  - Engler: Exokernel: An Operating System Architecture for Application-Level Resource Management
  - Baumann: The Multikernel: A New OS Architecture for Scalable Multicore Systems
  - Porter: Rethinking the Library OS from the Top Down
  - Hand: Are Virtual-Machine Monitors Microkernels Done Right?
  - (Heiser: Are Virtual-Machine Monitors Microkernels Done Right?)

Paper Reading List (2)

- File System
  - Rosenblum: The Design and Implementation of a Log-Structured File System
  - McKusick: A Fast File System for UNIX
  - Sandberg: Design and Implementation of the Sun Network Filesystem
  - Howard: Scale and Performance in a Distributed File System
  - Bittman: Twizzler: a Data-Centric OS for Non-Volatile Memory
- OS Verification
  - seL4: Formal Verification of an OS Kernel
  - Nelson: Hyperkernel: Push-Button Verification of an OS Kernel
- New OS Design
  - LegoOS: A Disseminated, Distributed OS for Hardware Resource Disaggregation
  - Boos: Theseus: an Experiment in Operating System Structure and State Management
  - Wang: Tinkertoy: Build your own operating systems for IoT devices

Paper Reading Requirements

- Submit a write-up for the required papers, 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 two 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
- Next lecture: Hardware Features for OS