CPSC 346 Operating Systems

Instructor: Yanping Zhang

Office hours: M: 10~11am, 2:10~3:10pm; W: 2:10 pm~4:10 pm

Email: zhangy@gonzaga.edu

Today

- Course policy
- Overview

About this course

- Reading
- Homework and Projects
  - Homework and projects (mostly individual work)
  - At least a week
  - Late turn-in will be a penalty: with 15% deduction each day delayed
  - Electronic copies are required
    - Electronic copy: Blackboard

About this course

- Grading breakdown
  - Homework: 20%
  - Projects: 30%
  - Midterms: 30%
  - Final: 20%
- Questions about your grades: come to me no later than one week after grades are returned to you

About this course

- Midterm and Final
  - No make ups
  - Three midterms

About this course

- Attendance:
  - You are expected to attend all classes
  - Contact me if you have to miss a class
  - 6 absence: V

About this course

- Electronic Devices
  - Laptops, cell phones, ipads, etc.
  - Inappropriate use of electronic devices are not permitted in class
  - You must have prior approval from me to use an electronic device in class
  - Always keep your cell phone “silent” in class

About this course

- Disability
  - Phone: 5093134134
  - Foley Center Library, Room 208
- Religious Accommodations
  - Reasonably accommodate students who, due to the observance of religious holidays
  - Let the instructor know as early as possible

About this course

- Last but not least: **ACADEMIC INTEGRITY**
- All members of the Gonzaga community are expected to adhere to principles of honesty and integrity in their academic endeavors.
- https://www.gonzaga.edu/academics/academic-calendar-resources/center-for-student-academic-success/academic-integrity

Acknowledgements

- Some pictures used in this class were obtained from the Internet
- The instructor used some slides from textbooks and reference books

Textbooks

- Operating System Concepts Essentials, 2nd Edition, A. Silberschatz, P. Galvin and G. Gagne.
- Computer Systems: A Programmer's Perspective, 3rd Edition, Bryant and O'Hallaron.

Topics

- Operating System Organization
- Architecture Support (x86 architecture)
- Concurrency: Processes and Threads, CPU Scheduling
- Memory Management
- I/O, File Systems
- Network/Security

Introduction

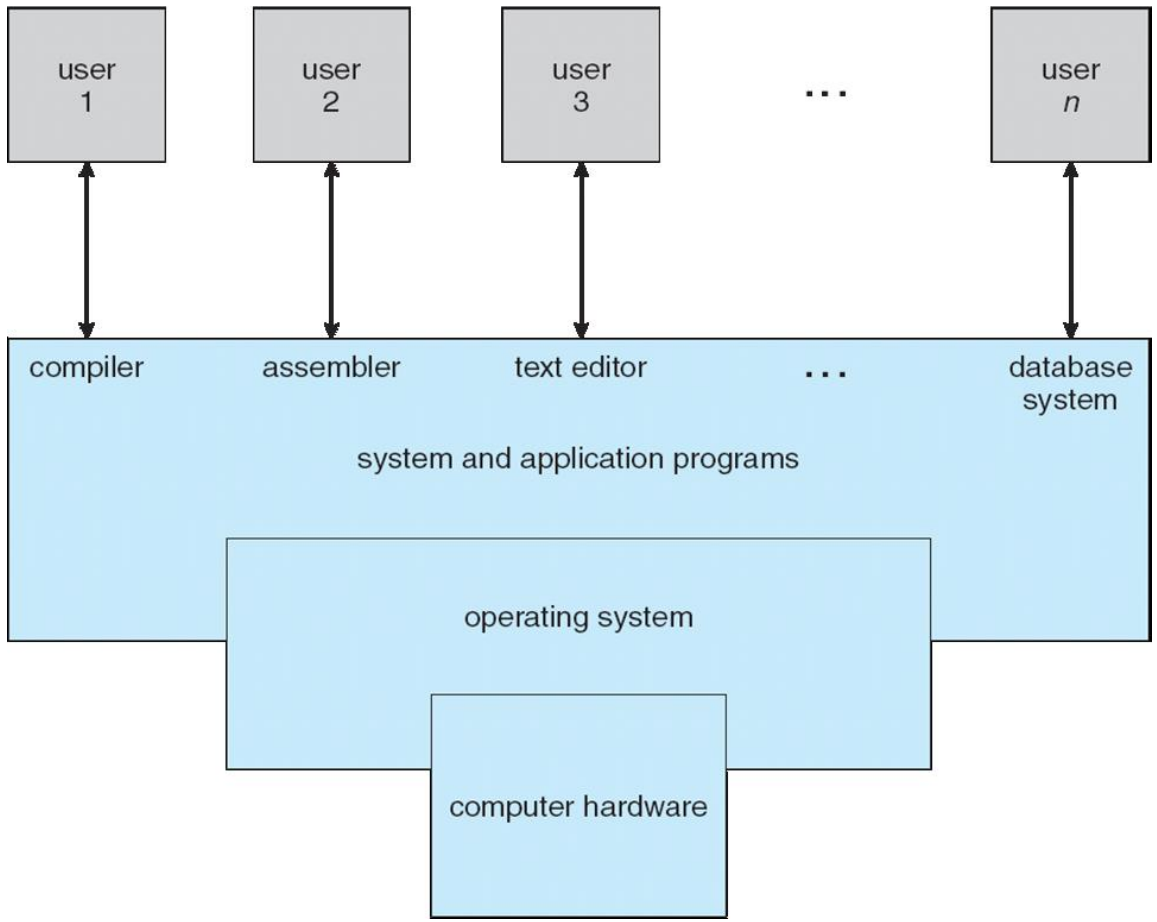
- Reading
  - Operating System Concepts Essentials: Page 3-30
  - C: http://heather.cs.ucdavis.edu/~matloff/unix.html
- What is an Operating System?
  - The code that Microsoft, Apple, Linux, Google provides
  - The code that you didn't write
  - The code that makes things work
  - The code that makes things crash
  - Etc.

Operating Systems

- Providing an appropriate interface for applications
- A program that acts as an intermediary between a user of a computer and the computer hardware
- Make the computer system convenient to use
- Use the hardware in an efficient manner

Operating System Structure

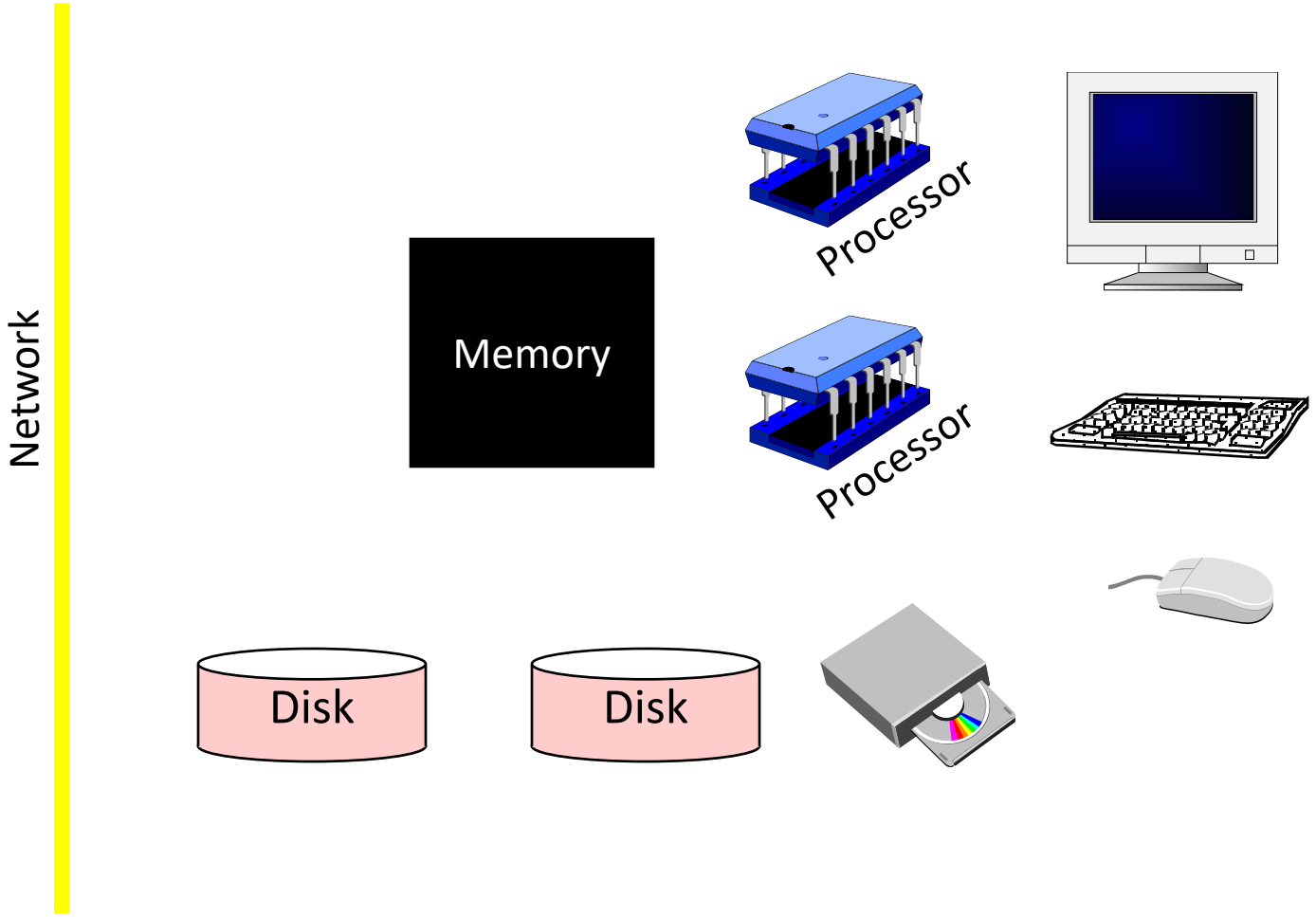
- Hardware
- Operating System
- Applications
- Users


user 1
user 2
user 3
...
user n
compiler
assembler
text editor
...
database system
system and application programs
operating system
computer hardware

Two Modes

- OS : kernel mode
  - Privileged mode
  - Access to all of hardware
- All other software: user mode
  - When needs to deal with hardware, OS will take care of it.
- OS is trusted; user is not
- OS has super-privileges; user does not

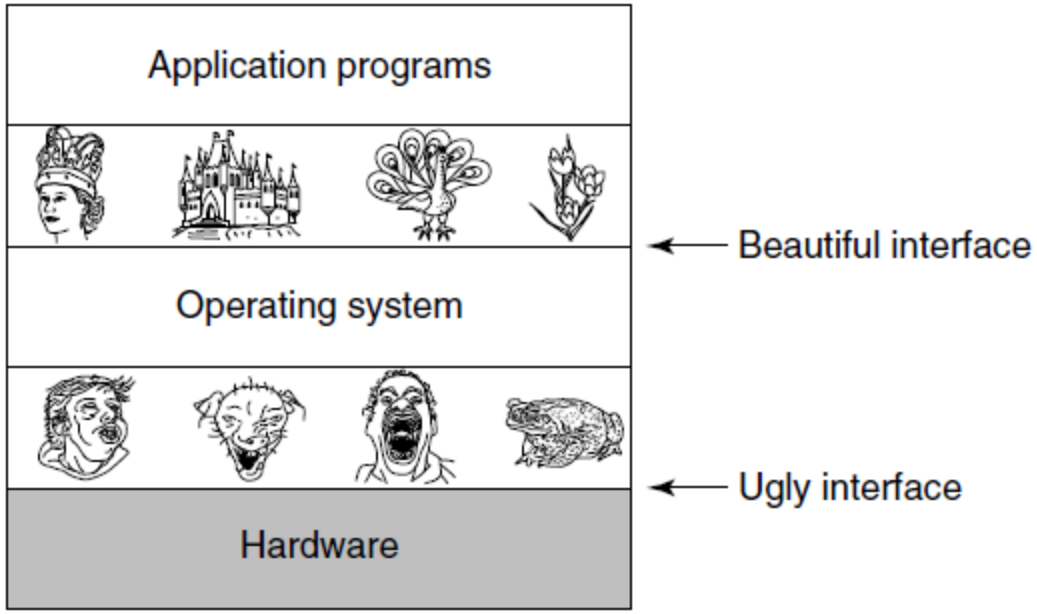
Hardware



Two Tasks of O/S

- Shield the user from the complexity of the underlying hardware
- Allocate resources to competing users

Task 1: Shield the User



Operating systems turn ugly hardware into beautiful abstractions.

An Example

- Read/Write in C
  - open(), close(), scanf(), printf(), etc.
- But (to use an old example) NEC PD765 controller for floppy disk drive has
  - 16 commands to read, write, move disk arm, format tracks
  - read/write has 13 parameters that specify sectors/track, etc.
  - commands to monitor the status of the motor
  - after each operation, chip returns 23 status and error fields

Task 2: Resource Allocator

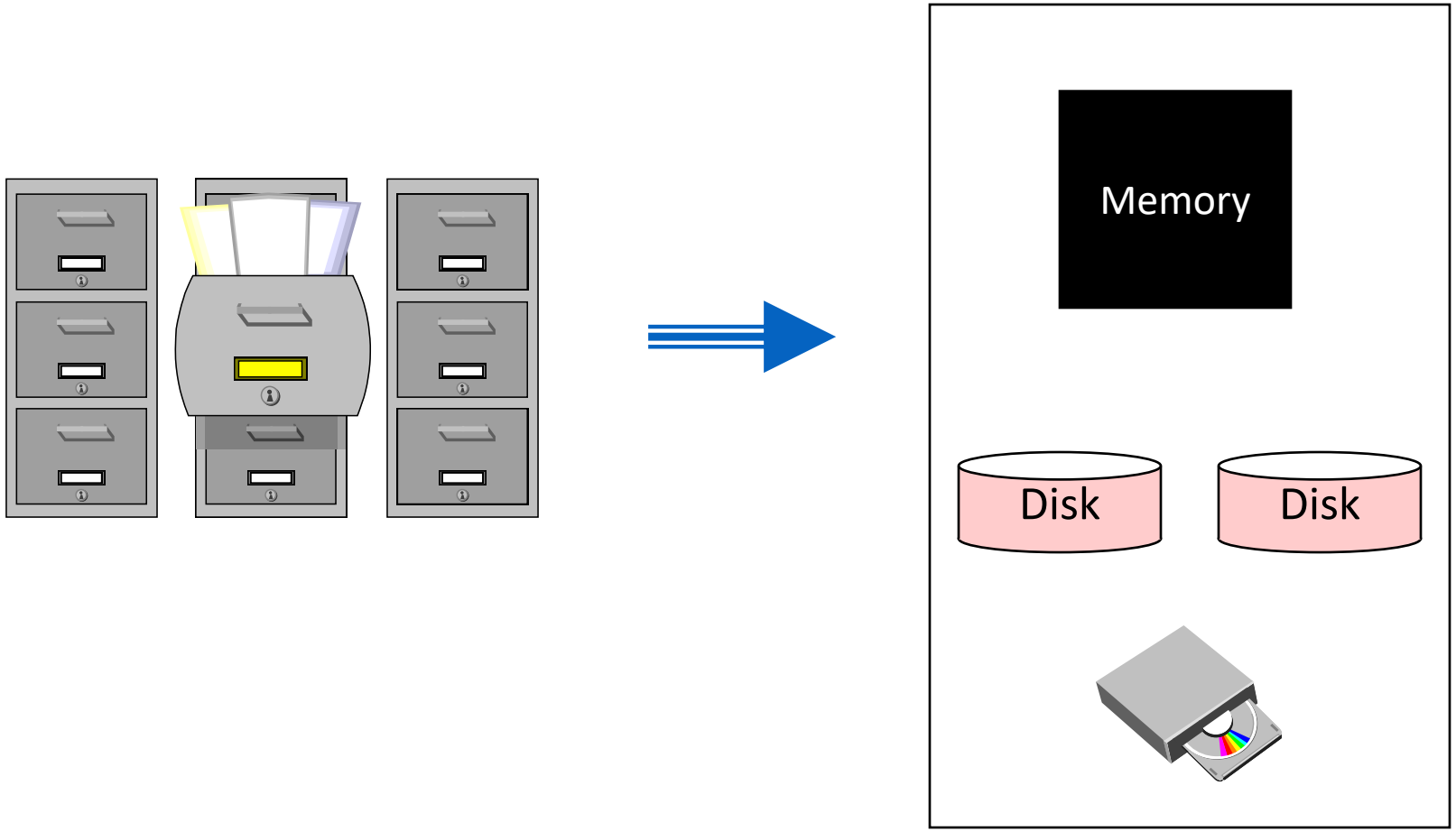
- O/S allocates
  - cpu
  - printer
  - memory
  - etc.
  - between competing users/processes

- Evolution of O/S traces these two functions:
  - shield the user
  - resource allocator

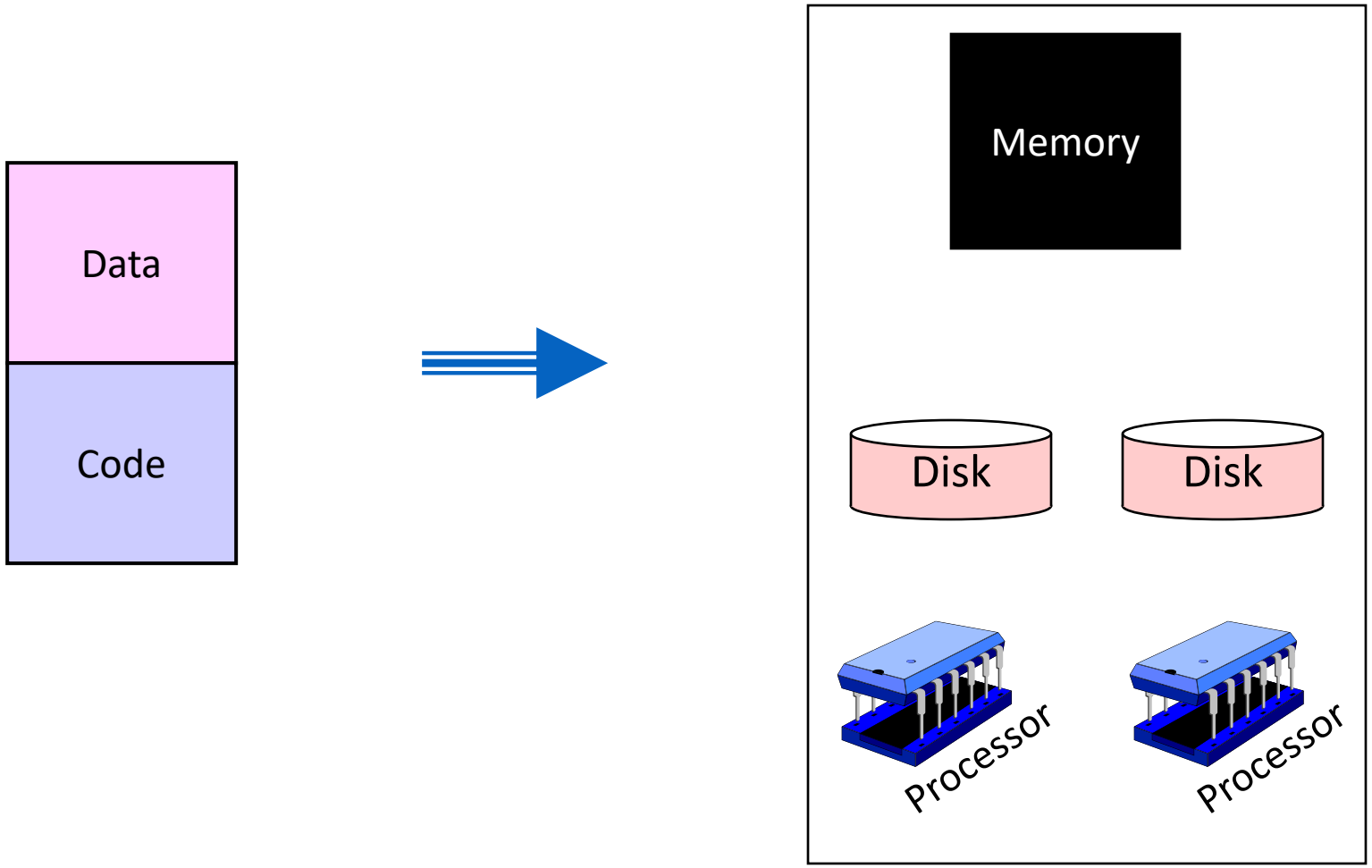
Five Primary Tasks

- Process Management
- Memory Management
- Storage Management
- I/O Subsystem
- Security and Protection

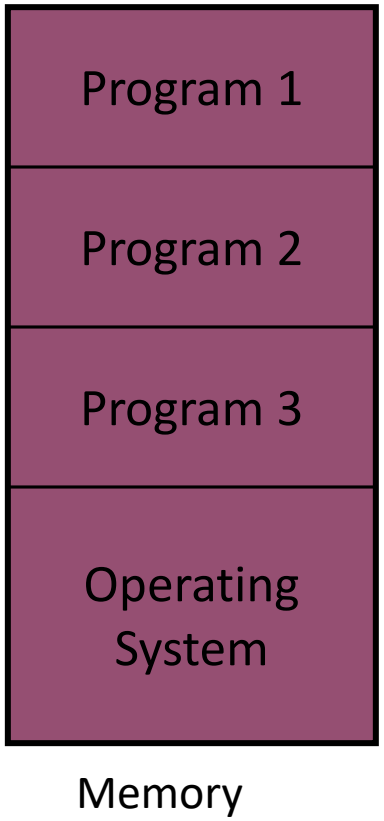
Files



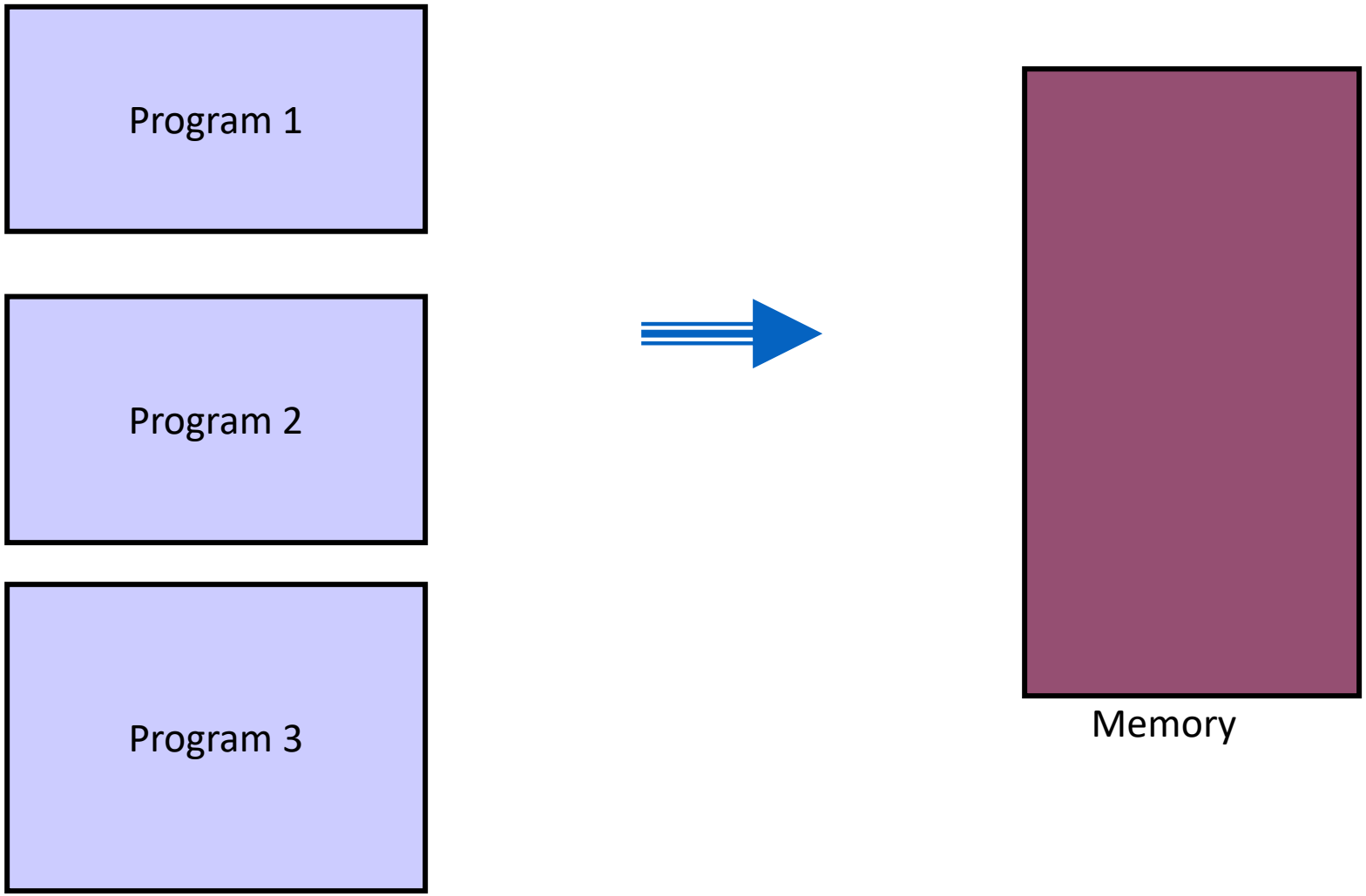
Programs



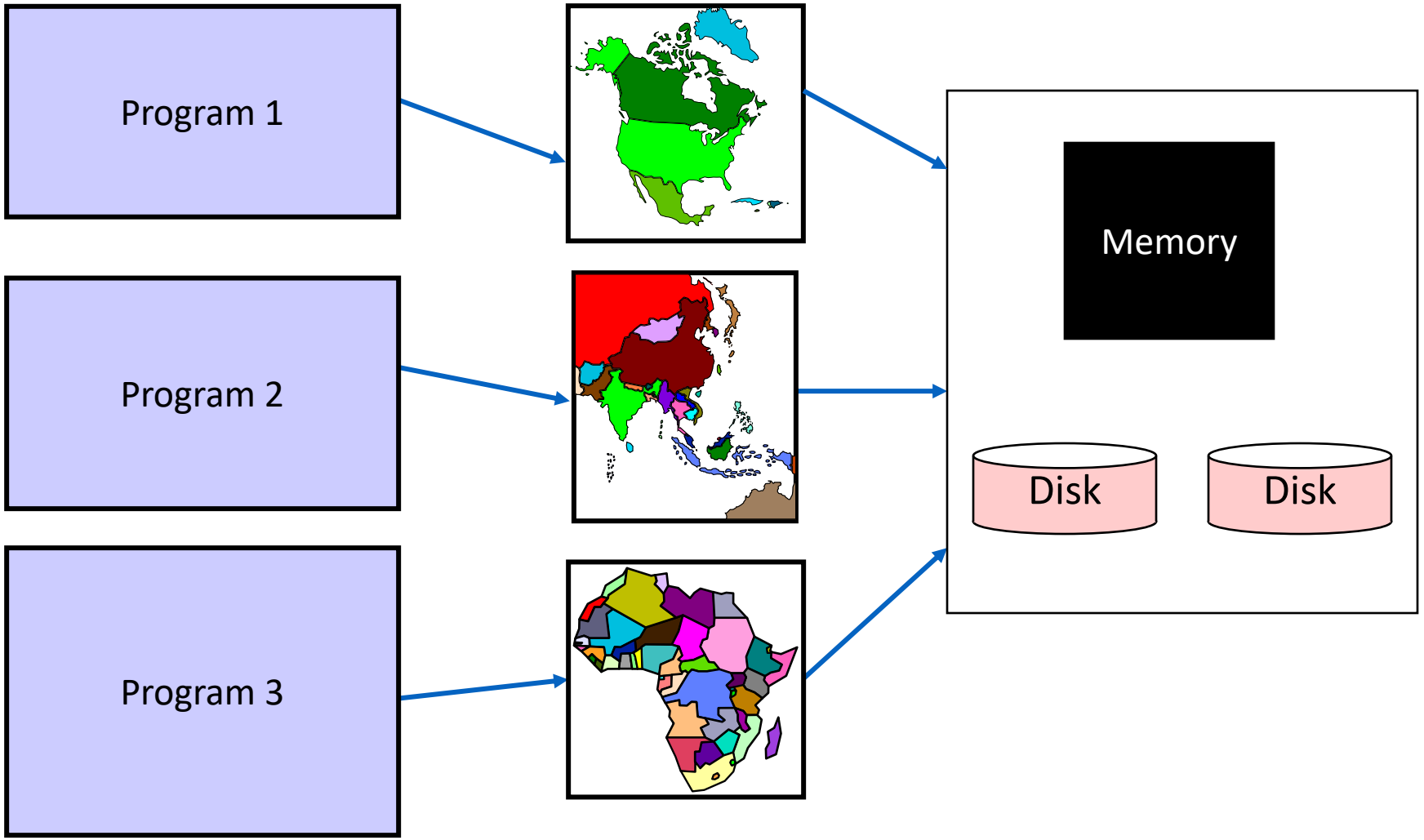
Memory Sharing (1)



MemorySharing (2)



Virtual Memory



In the Beginning …

- There was hardware
  - processor
  - storage
  - card reader
  - tape drive
- And not much else
  - no operating system
  - no libraries
  - no compilers

IBM 701

OS:

Initially, none

IBM 650

OS:

none


IBM

Programming Without an OS

- Assemble all software into a deck of punched cards
- Get 15-minute computer slot
  1) pay $75 ($611 in 2010 dollars)
  2) mount tapes containing data
  3) read cards into computer
  4) run program
     - it probably crashes
  5) output goes to printer
- Steps 1, 2, 3, and 5 take 10 minutes
  - leaving 5 minutes for step 4!

Unix

Late 80s/Early 90s

- 1988: Most major Unix vendors get together and form OSF to produce a common Unix: OSF/1, based on IBM's AIX
- 1989: Microsoft begins work on NT
- 1990: OSF abandons AIX, restarts with Mach
- 1991: OSF releases OSF/1
- 1992: Sun releases Solaris 2
  - many SunOS (Solaris 1) programs are broken
- 1993: All major players but DEC have abandoned OSF/1
- 1993: Microsoft releases Windows NT 3.1
- 1994: Linux 1.0 released

Late 90s

- IBM has three different versions of Unix, all called “AIX”
- 1996: DEC renames its OSF/1 “Digital Unix”
- 1996: Microsoft releases Windows NT 4
- 1996: Linux 2.0 released
- 1998: DEC is purchased by Compaq; “Digital Unix” is renamed “Tru64 Unix”
- 1999: Sun’s follow-on to Solaris 2.6 is called Solaris 7

The ’00s Part 1

- 2000: Microsoft releases Windows 2000 and Windows Me
- 2000: Linux 2.2 is released
- 2000: IBM “commits” to Linux (on servers)
- ~2000: Apple releases OS X, based on Unix (in particular, OSF/1)
- 2001: Linux 2.4 is released
- 2001: Microsoft releases Windows XP
- 2002: Compaq is purchased by HP
- 2003: SCO claims their code is in Linux, sues IBM; IBM countersues
  - August 10, 2007: judge rules that SCO is not the rightful owner of the Unix copyright, Novell is
  - Novell says there is no Unix in Linux
  - September 2007: SCO files for Chapter 11 bankruptcy protection

The ’00s Part 2

- 2004: Linux 2.6 is released
- 2005: IBM sells PC business to Lenovo
- July 2005: Microsoft announces Windows Vista
- January 2007: Microsoft releases Windows Vista
- later in 2007: Microsoft starts hinting at Windows 7
- April 2009: Oracle announces purchase of Sun Microsystems
- July 2009: Google announces Chrome OS
- October 2009: Microsoft releases Windows 7

History of C

- Early 1960s: CPL (Combined Programming Language)
  - developed at Cambridge University and University of London
- 1966: BCPL (Basic CPL): simplified CPL
  - intended for systems programming
- 1969: B: simplified BCPL (stripped down so its compiler would run on minicomputer)
  - used to implement earliest Unix
- Early 1970s: C: expanded from B
  - motivation: they wanted to play “Space Travel” on minicomputer
  - used to implement all subsequent Unix OSes