UNIVERSITY of WASHINGTON | BOTHELL

CSS430 Introduction

Textbook Chapter 1

Instructor: Stephen G. Dame
e-mail: sdame@uw.edu

These slides were adapted from the OSC textbook slides (Silberschatz, Galvin, and Gagne), Professor Munehiro Fukuda and the instructor's class materials.

"Computer science is no more about computers than astronomy is about telescopes." - Edsger Dijkstra

Edsger Wybe Dijkstra (Dutch pronunciation: [ɛtsxər ʊibə dɛikstra]); (1930–2002) was a Dutch computer scientist. He received the 1972 Turing Award for fundamental contributions to developing programming languages, and was the Schlumberger Centennial Chair of Computer Sciences at The University of Texas at Austin from 1984 until 2000.

Course Objectives

- You will:
  - ✓ Study the fundamental concepts of operating systems
  - ✓ Practice the logical design using Java
- But not:
  - ✓ Learn how to build web sites or mobile applications
    - ❖ If this is your main objective, you should take UW Computing Training or other classes with that focus

    http://www.washington.edu/lst/workshops/fundamentals
  - ✓ Increase your C/C++ programming skills
    - ❖ Program 1 will deal with several system C/C++ calls, but for more C/C++ skill building please refer to CSS432, CSS434 or other C/C++ programming courses.

Important Timelines*

- Program 1: System calls and shell
- Program 2: Scheduler
- Midterm Exam: Process Management
- Program 3: Synchronization
- Program 4: Paging
- Final Exam: Memory/File Management
- Final Project: Unix-like file system

(* Check the syllabus for due dates)

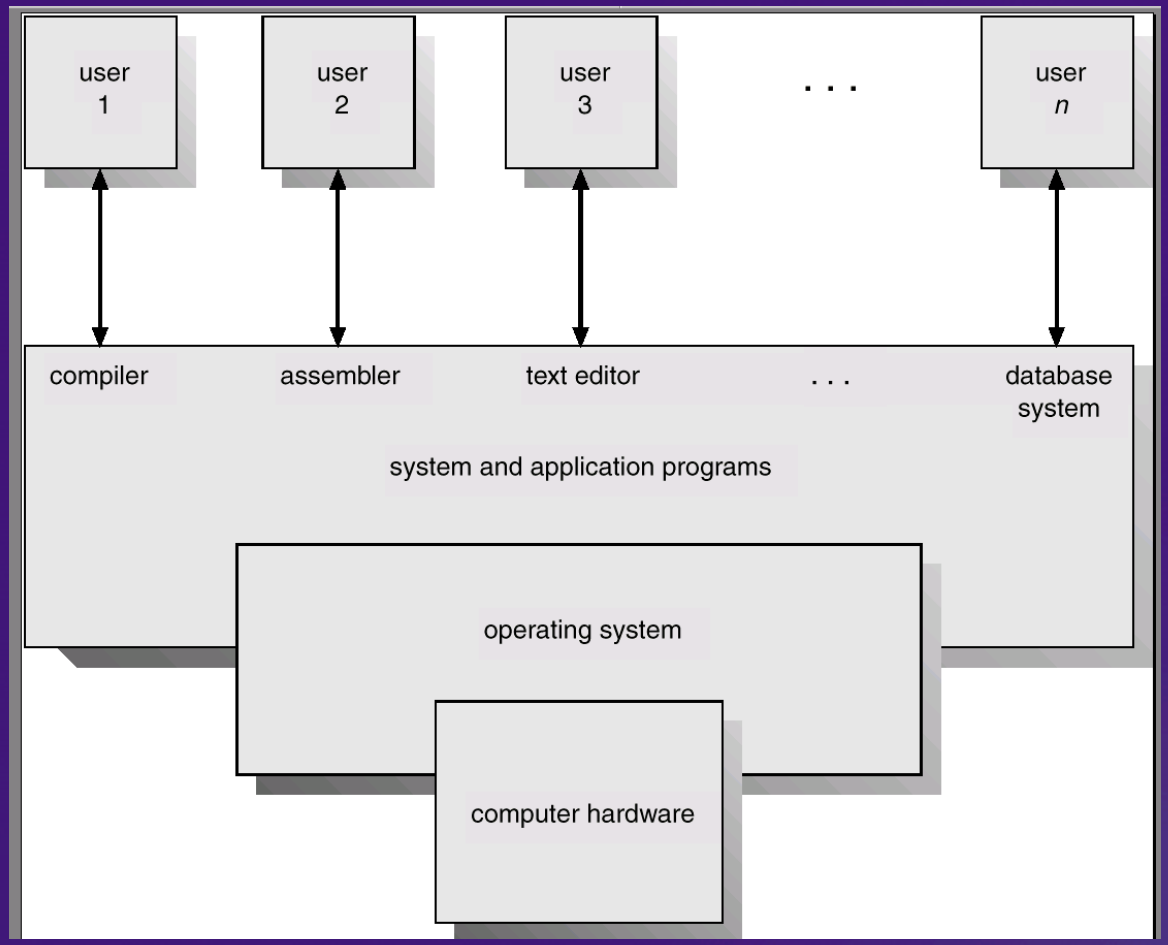
Important Links, Email & Resources

- **Our class web:**
  http://courses.washington.edu/css430
- **Textbook:**
  http://codex.cs.yale.edu/avi/os-book/OS8/os8j/
- **Java (JavaSE7):**
  http://download.oracle.com/javase/7/docs/api/
- **Instructor's email:**
  sdame@uw.edu
- **Class discussion mailing list:**
  Canvas Discussion Board for CSS430
- **Assignment submission:**
  Submit your tar-archived file to Canvas LMS
  (from your MyUW account page)

What is an Operating System?

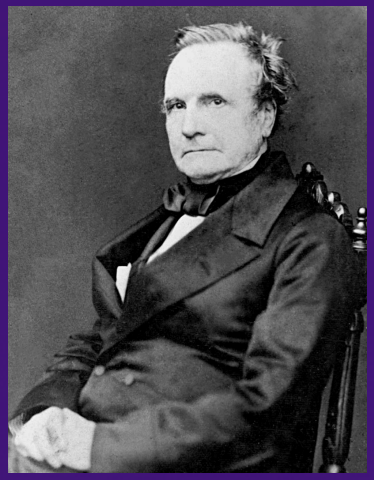
- Goals
  - Execute user programs and make solving user problems easier
  - Making the computer system convenient to use
  - Using computer hardware in an efficient manner
- Definitions
  - Resource allocator – manages and allocates resources
  - Control program – controls the execution of user programs and operations of I/O devices
  - Kernel – the one program running at all times (all else being application programs)

Computer System Components

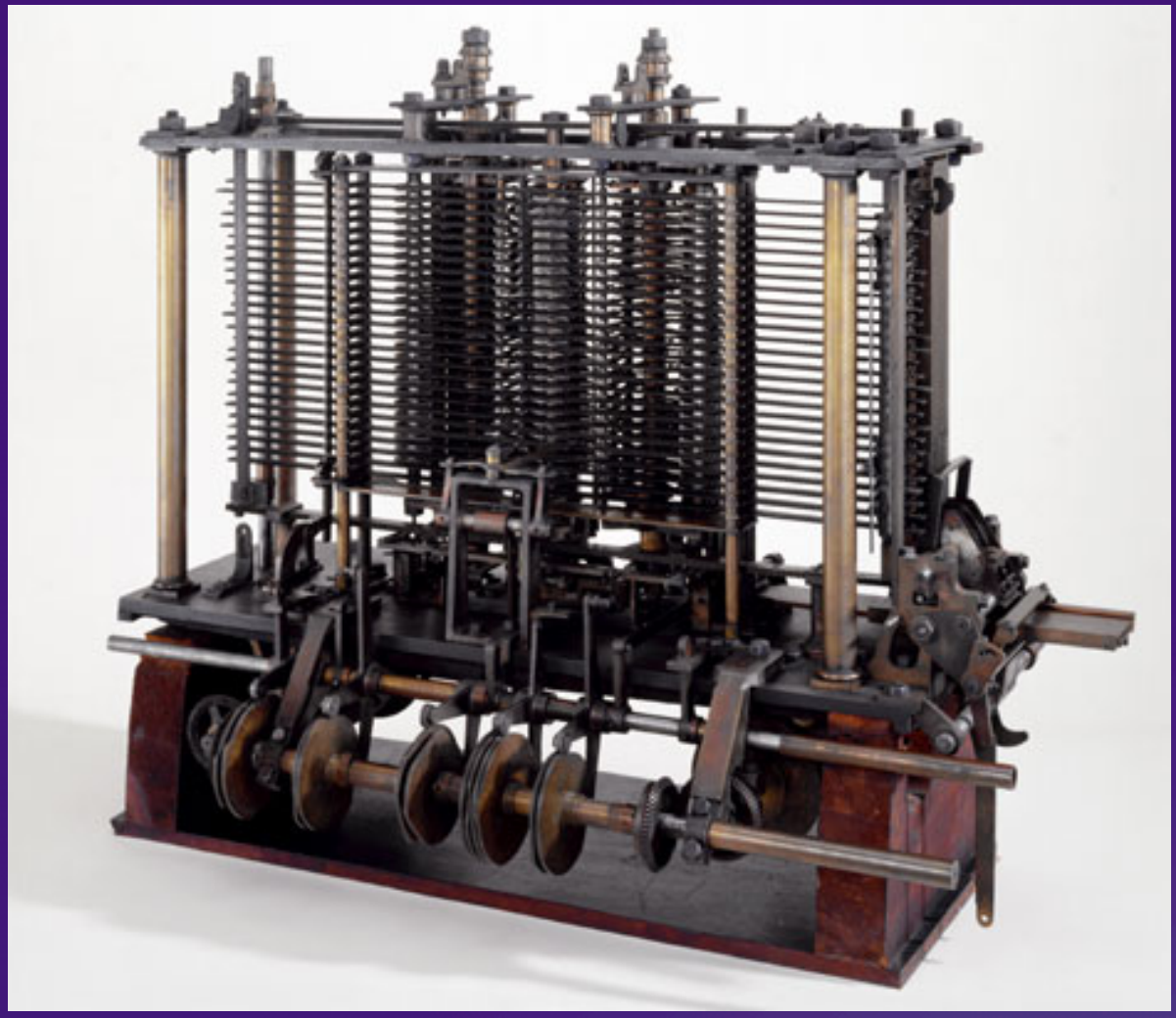


Some Operating System History….

Charles Babbage (1792-1871)



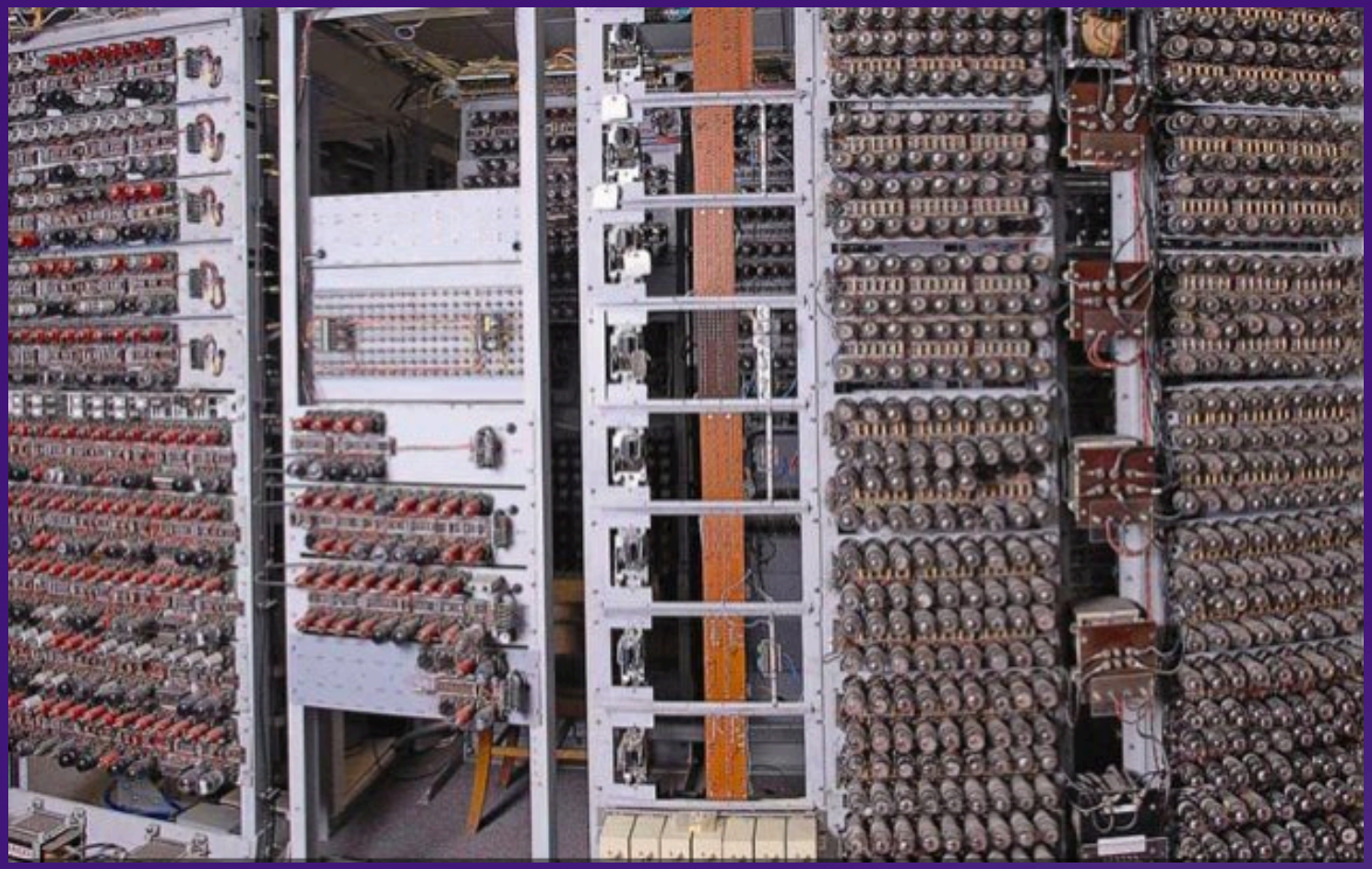
Punch Cards



Analytical Engine

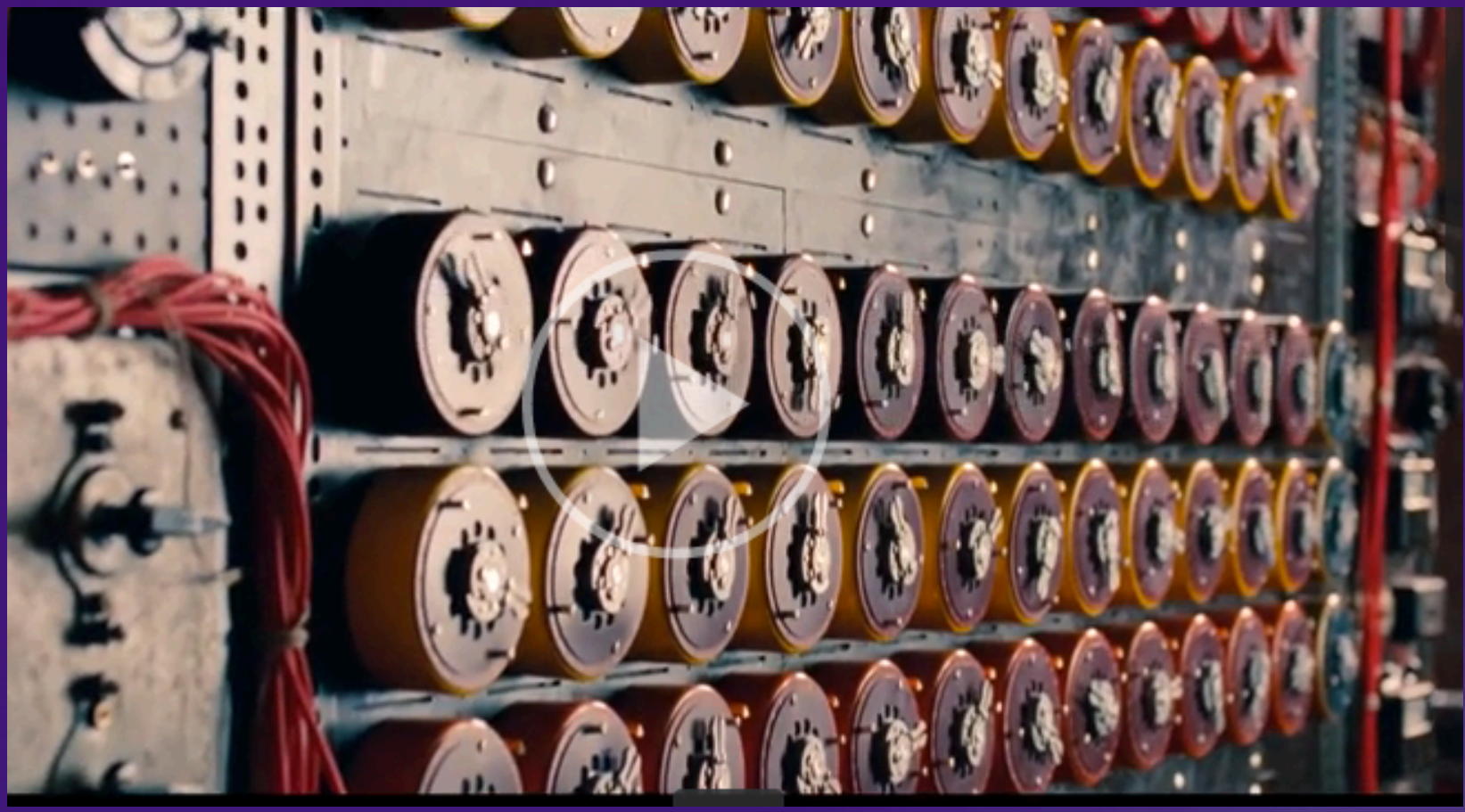
Hired Ada Lovelace as the world's first programmer!

2,500 Vacuum Tubes



1944 Colossus – Bletchley Park deciphered encrypted teleprinter messages between Hitler and his commanders in WWII.
-- Programmed by Alan Turing and others at Bletchley Park --

Alan Turing and The Imitation Game

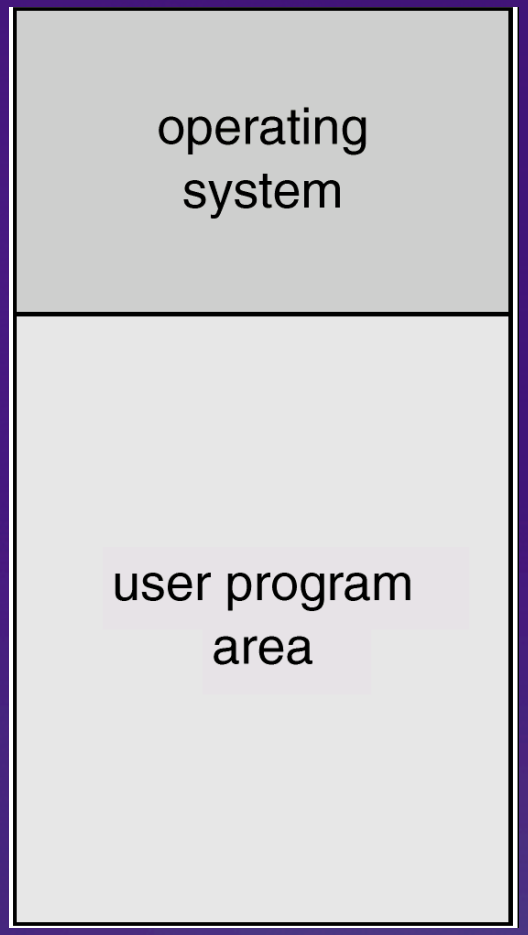


ENIAC

1946 ENIAC –
US army supported general purpose electronic computer to calculate artillery firing tables.
Presper Eckert and John Mauchley main developers.
Led to EDVAC and Univac.

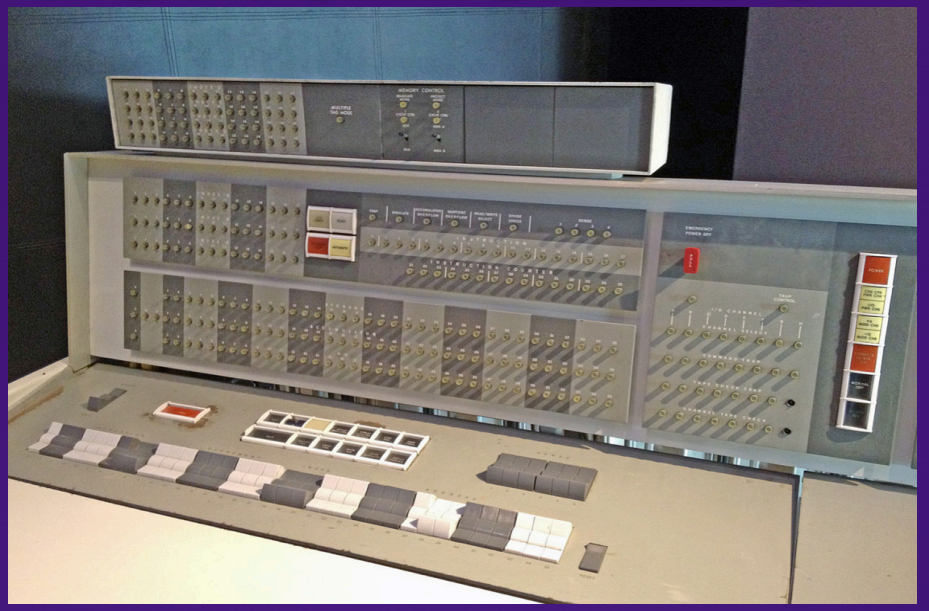
Batch Systems

- A job is assembled of the program, the data, and some control information (in control cards).
- Programmers pass their jobs to an operator.
- The operator batches together jobs.
- OS transfers control from one job to another.
- Each job output is sent back to the programmer.

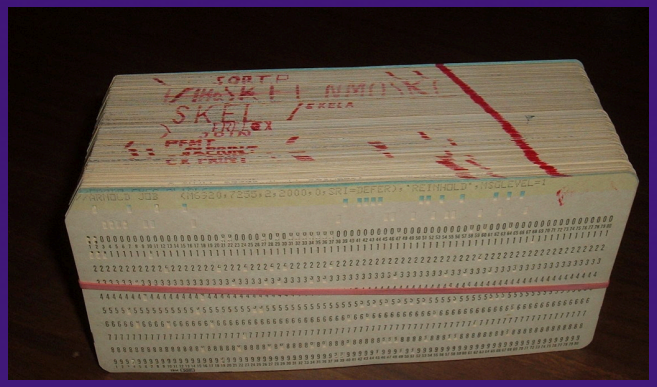


FORTRAN PUNCH CARDS

CS in the 1950/60s!



IBM 7094 – IBMSYS OS



SINGLE PROGRAM Software

Punch card reader

3rd Generation -IBM 360 – OS/360

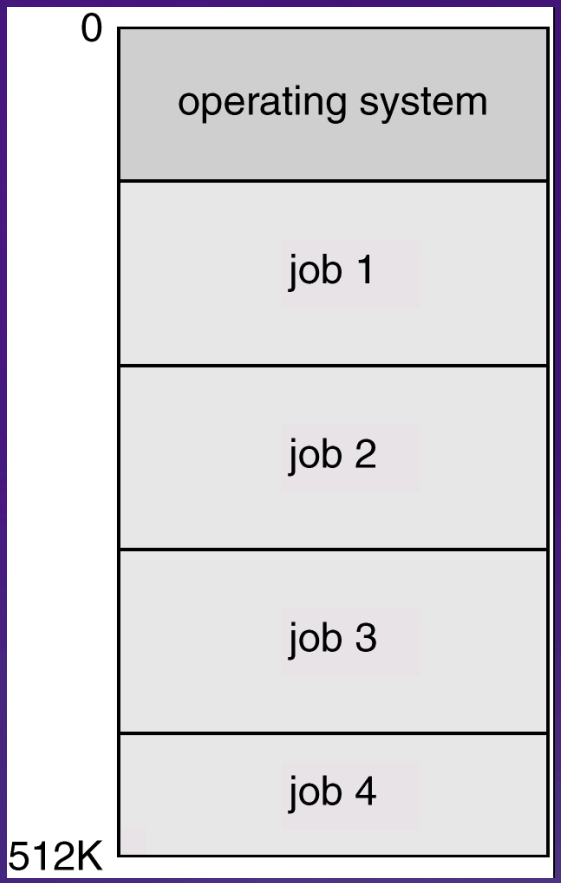
- Millions of assembly LOC
- Thousands of programmers
- Thousands of bugs

... but

- Multiprogramming!

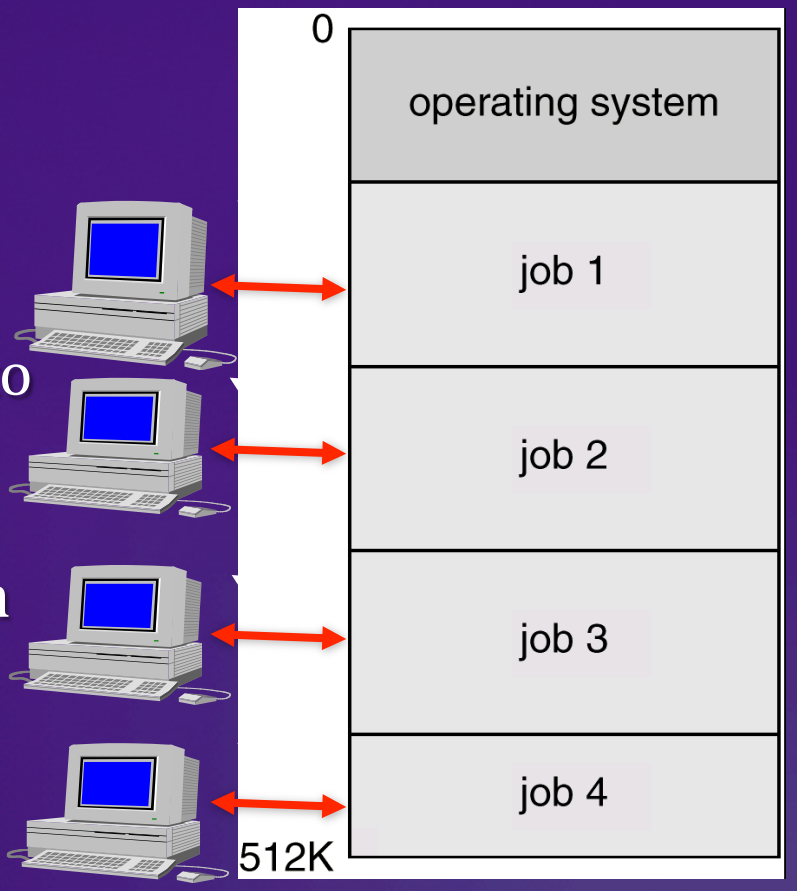
Multiprogramming

- Several jobs are kept in main memory at the same time.
- OS picks one of them to execute.
- The job may have to wait for a slow I/O operation to complete.
- OS switches to and executes another job.
- To facilitate multiprogramming, OS needs:
  - ✓ Job scheduling
  - ✓ Memory management



Time-Sharing Systems

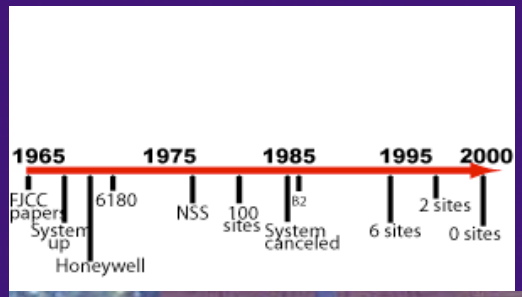
- This is a logical extension of multiprogramming.
- Each user has at least one separate program in memory.
- A program in execution is referred to as a process.
- Process switch occur so frequently that the users can interact with each program while it is running.
- File system allows users to access data and program interactively.

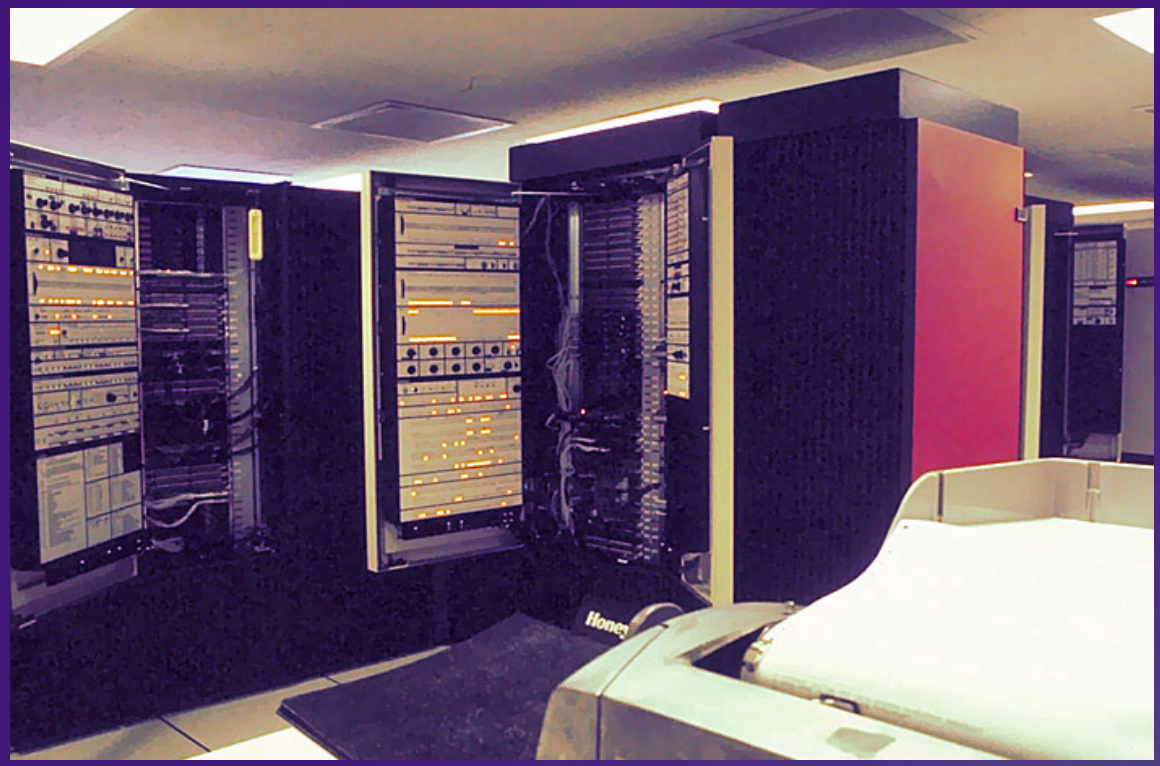


Multics → Unix

Multiplexed **I**nformation and **C**omputing **S**ervice

- Conceived as a general purpose time-sharing utility
- Joint project of MIT, GE and Bell Labs started in 1965
- Taken over by Honeywell in 1970
- MIT shut down its Multics system in 1988
- Last system was deactivated in 2000





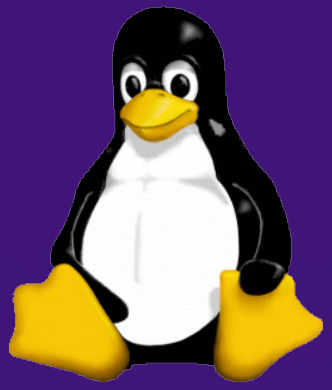
Unix

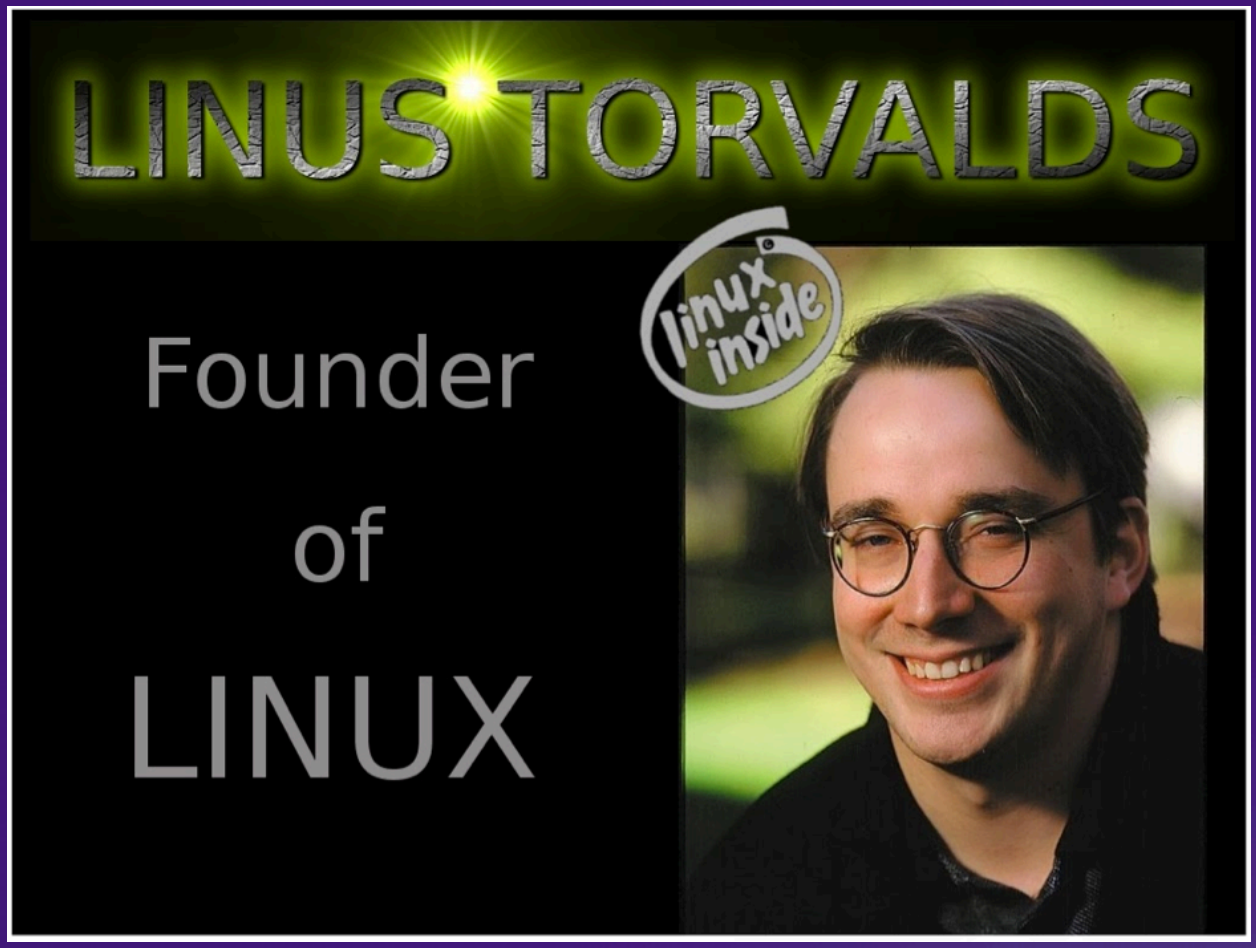
1969 Ken Thompson and Dennis Ritchie – Bell Labs on a $70,000 DEC PDP-7

Developing an alternative to Multics

Unix then BSD Unix

Linux





- Started with MINIX3 for education (Tanenbaum)
- Became the world-wide commercial blockbuster – LINUX!

Personal-Computer Systems

- Personal computers – computer system dedicated to a single user.
- User convenience and responsiveness
- Can adopt technology developed for larger operating systems' features.
- At its beginning, a single user system did not need advanced CPU utilization and protection.
- Later, file protection was necessary to avoid viruses.
- Overall, the same OS concepts are appropriate for the various different classes of computers.

4th Generation – PCs (1980 - now)

- CP/M (Control Program for Microcomputers)
- MS-DOS and the IBM PC
- MAC OS, MAC OSX
- Windows, Windows NT
- Linux and its Multiple Distros:
  - ✓ RedHat
  - ✓ SUSE
  - ✓ Debian
  - ✓ Ubuntu

5th Generation – Mobile (1990 - now)

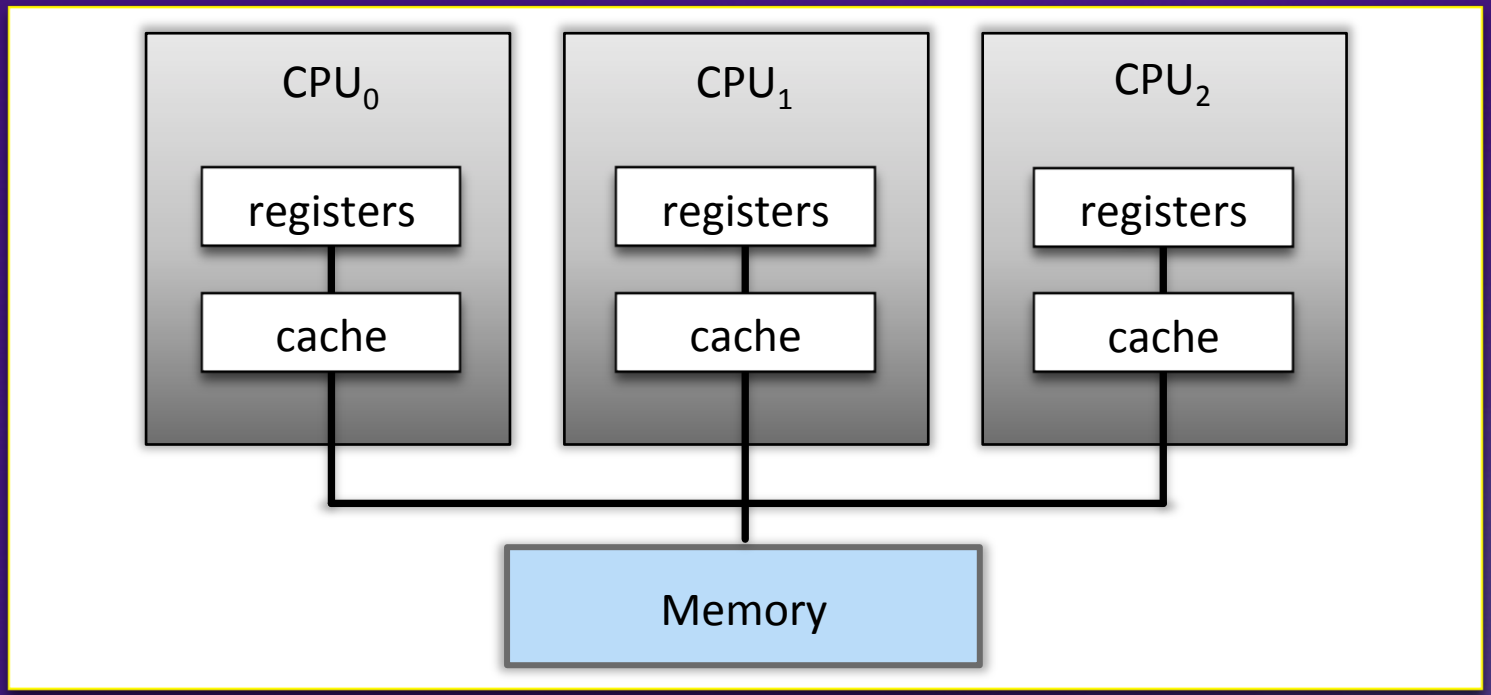
Originally termed PDAs – Personal Digital Assistants

Now SMART PHONE OSs

- Google Android (Linux Based)
- Apple iOS (iPod, iPad, iPhone and ... Apple Watch!)
- Symbian OS (Nokia, now owned by Microsoft)
- RIM's Blackberry OS
- Windows Phone

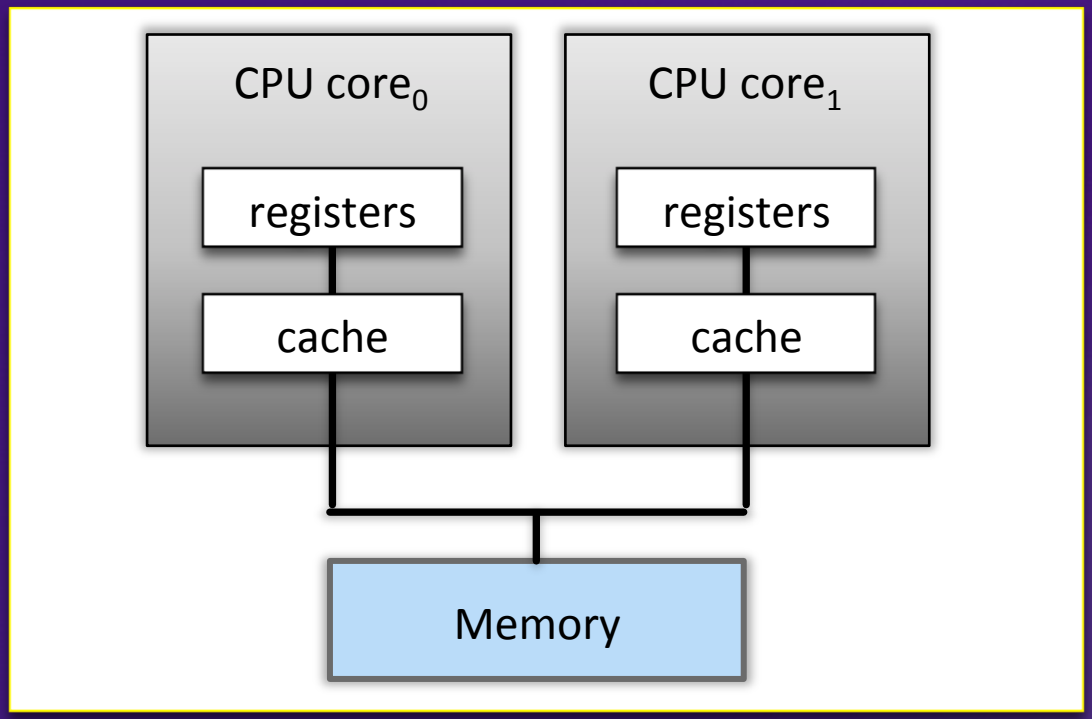
Symmetric Multiprocessing Architecture

- Multiprocessor systems with more than one CPU in close communication (in one box).
- *Tightly coupled system* – processors share memory and a clock; shared-memory-based communication.
- Advantages of parallel multiprocessor systems:
  - ✓ Increased *throughput*
  - ✓ Economical
  - ✓ Increased reliability



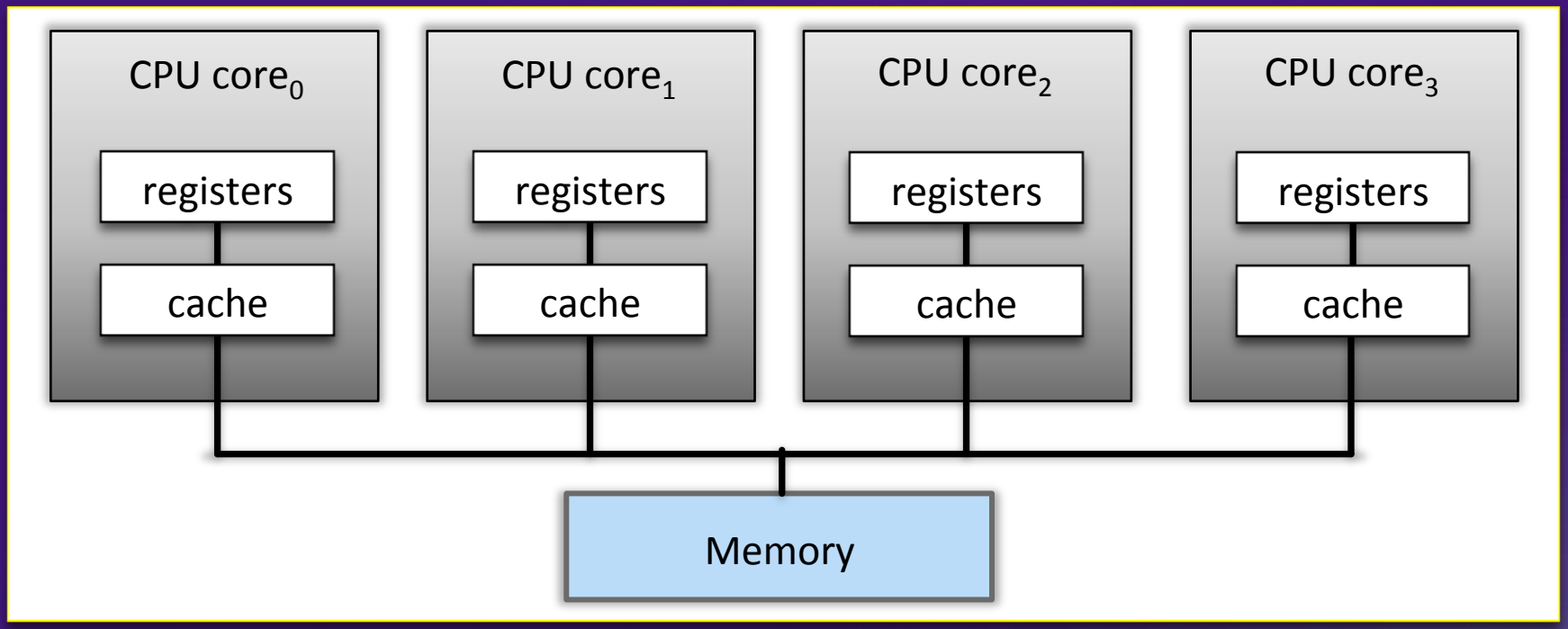
Dual-Core Design

- An MPU chip has two CPU cores, each:
  - owning its own small L1 cache,
  - sharing a large L2 cache, and
  - accessing the main memory through L2.

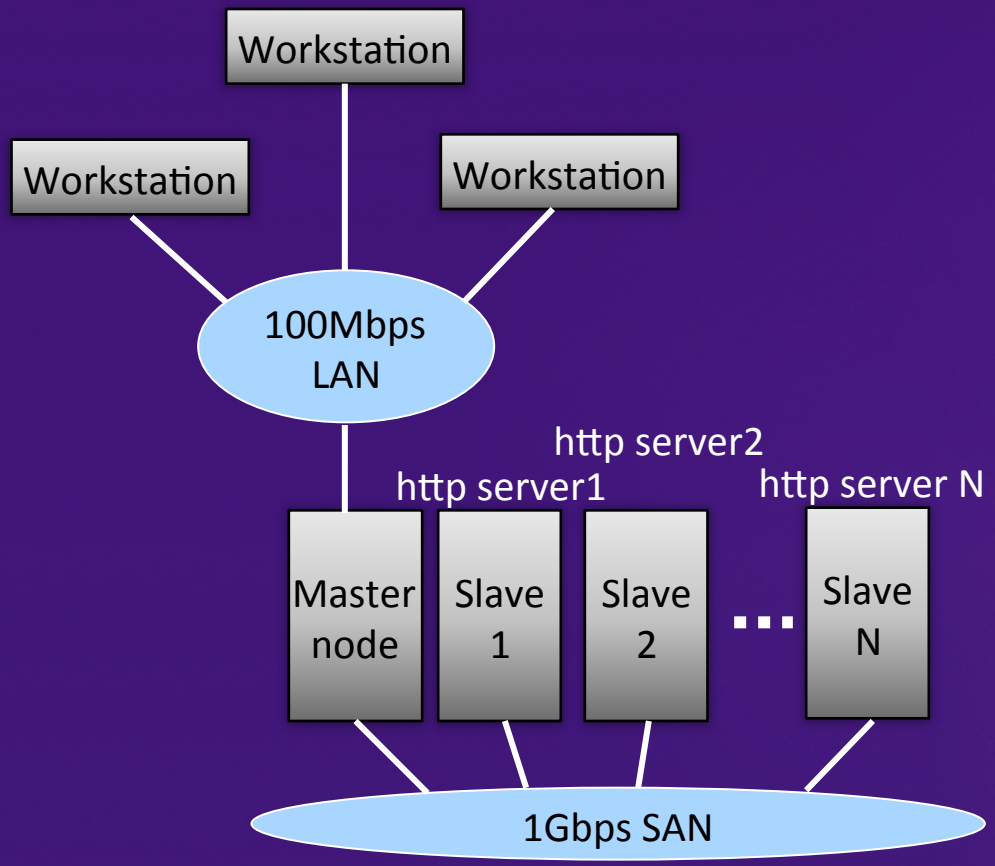


Quad-Core Design

- An MPU chip has four CPU cores, each:
  - ✓ owning its own small L1 cache,
  - ✓ sharing a large L2 cache, and
  - ✓ accessing the main memory through L2.

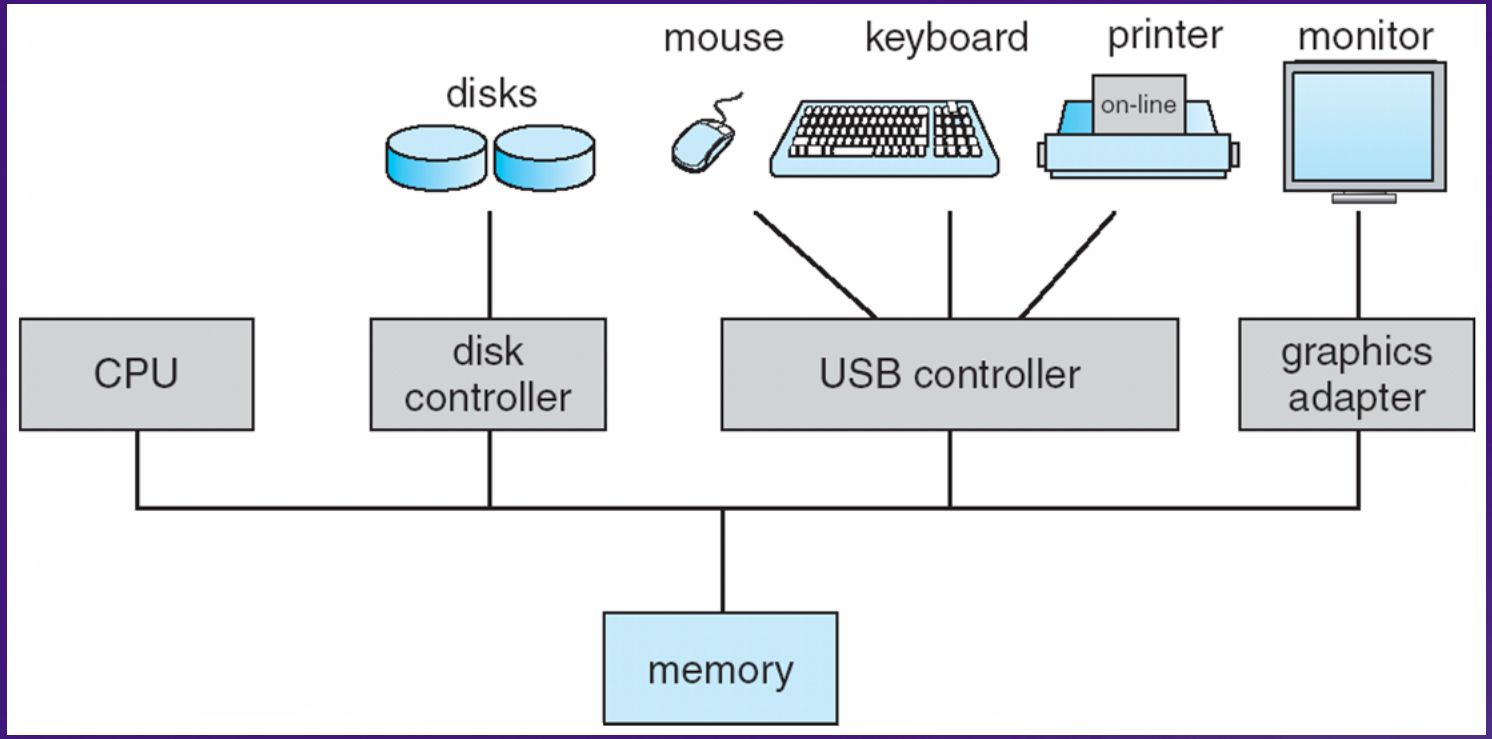


Cluster Systems

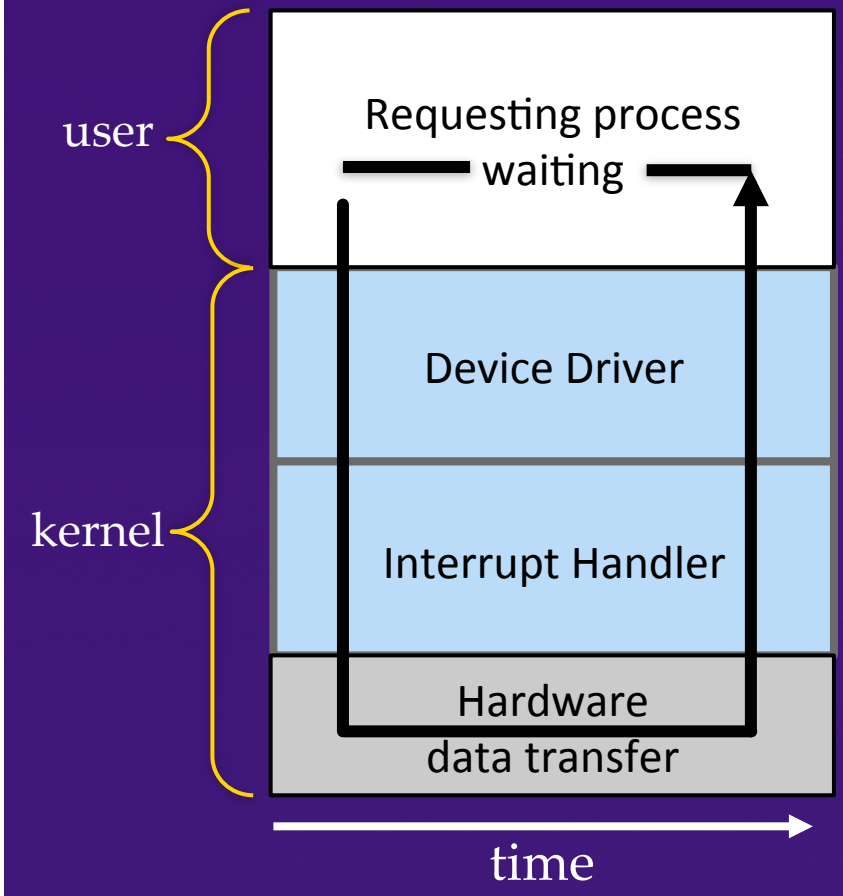


- Client
  - ✓ Takes a client-server model
- Server
  - ✓ Consists of many PC/ workstations connected to a high-speed network or a storage-area network (SAN).
  - ✓ Puts more focus on high-performance computing (HPC)
  - ✓ serves for requests in parallel.

Computer Hardware

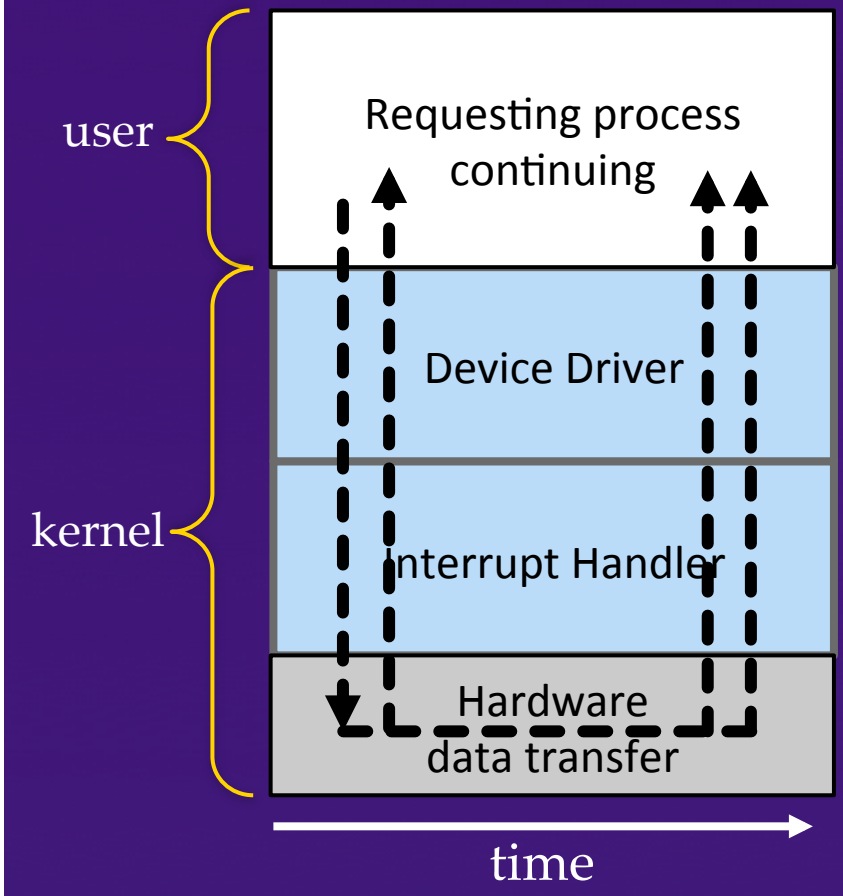


Synchronous I/O



- During execution, each program needs I/O operations to receive keyboard inputs, open files, and print out results.
- In the early computer era, a program had to wait for an I/O operation to be completed. (Synchronous I/O)
- This frequently causes CPU idle.

Async I/O and Interrupts



- Asynchronous I/O returns control to a user program without waiting for the I/O to complete.
- When the I/O is completed, an interrupt occurs to CPU that temporarily suspends the user program and handles the I/O device.

Discussion 1

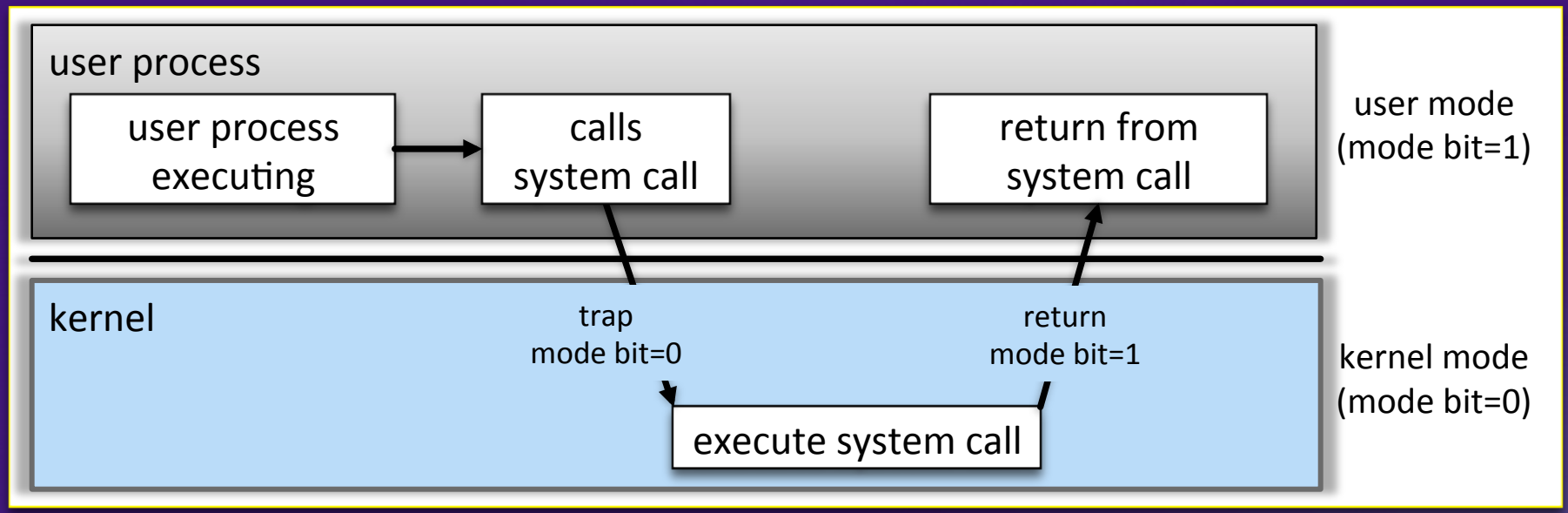
1. Timers can be used to compute the current time. How do operating systems acquire and manage to keep accurate time?
2. What is a *context switch* and what is the main challenge to an OS during a context switch?
3. How does a *real-time* OS differ from a standard OS?

Hardware Protection

- Purpose:
  - With resource sharing, many programs could be affected by a bug in one program.
  - Incorrect or malicious resource accesses cause a hardware trap to the operating system.
- Dual-Mode Operation:
  - User mode: no privileged instructions allowed.
  - Kernel mode: Privileged instructions allowed.
- I/O Protection: all privileged
- Memory Protection: A region from the base to the limit register allowed to use
- CPU Protection: CPU allowed to use until the timer gets 0.

Dual-Mode Operations

- Provides hardware support to differentiate between at least two modes of operations.
  1. ***User mode*** – execution done on behalf of a user.
  2. ***Monitor mode*** (also *supervisor mode, system mode, or* ***kernel mode***) – execution done on behalf of operating system.
- Switching between two modes
  - ✓ Device interrupts, hardware traps, system calls cause a trap to the kernel mode
  - ✓ The operating system returns to the user mode after servicing requests.



Process Management

- A program running within a process needs:
  - ✓ CPU Execution Time (allocation)
  - ✓ Memory (for instructions and data)
  - ✓ Files
  - ✓ I/O Devices (keyboards, mouse, graphics, printers, …)
- Process is the fundamental unit of work in a system.
  - ✓ Operating System Processes (system code)
  - ✓ User Processes (user code)
- A process can have 1 or more (concurrent) threads
  - ✓ Each thread has a program counter on a program sequence
  - ✓ Each thread is considered a separate execution sequence
- 5 Major OS Process Management activities:
  - ① **S**cheduling of processes and threads
  - ② **C**reating/**D**eleting processes (system and user)
  - ③ **S**uspending and **R**esuming processes
  - ④ **P**rocess **S**ynchronization
  - ⑤ **P**rocess **C**ommunication

Memory Management

- Main Memory
  - ✓ CPUs are only able to access programs and data in main memory.
  - ✓ Very large, but volatile and varies in speed
  - ✓ Contains bytes or words each with their own *address*
- Programs
  - ✓ Mapped to absolute addresses
  - ✓ Loaded into main memory
  - ✓ Accessible to multiple *threads* (via different program counters)
- Memory Management
  - ✓ Different algorithms are optimized for different systems
  - ✓ Dependent on specific hardware support
- 3 Major OS Memory Management activities:
  - ① **K**eeping track of memory used and by whom
  - ② **D**eciding which processes and data to move in/out of memory.
  - ③ **A**llocating and **D**eallocating memory as necessary.

File System Management

- File
  - ✓ The logical unified view of information storage
  - ✓ Logical storage unit abstracted from the details of the device by OS
- File System
  - ✓ Controls access from multiple users (e.g. read, write, append)
  - ✓ Organizes the structure (e.g. directories / subdirectories of files)
  - ✓ Interacts with storage device drivers (abstracting details)
- Memory Management
  - ✓ Different algorithms are optimized for different systems
  - ✓ Dependent on specific hardware support
- 5 Major OS File Management activities:
  - ① **C**reating and **D**eleting files
  - ② **C**reating and **D**eleting folders(directories) to organize files
  - ③ **P**rimitives for manipulating files and directories
  - ④ **M**apping files to secondary storage media
  - ⑤ **B**acking up files to non-volatile storage media

Open Source Operating Systems

- Open Source
  - ✓ Software Industry started in an Open Source culture (~1950s)
  - ✓ "Homebrew Clubs" and MITs Model Railroad Club source code
  - ✓ Open Source may be more secure and bug free due to large community
- Copyright vs. Copyleft
  - ✓ Binary only releases and source code copyrights (closed source)
  - ✓ Digital Rights Management (DRM) =illegal to reverse engineer
  - ✓ Free Software Foundation (FSF) – GNU project ("GNU's Not Unix")
  - ✓ GPL – General Public License – must ship source with binaries
- GNU/Linux
  - ✓ Hundreds of unique distributions from the core…
  - ✓ https://www.kernel.org/pub/linux/kernel/v2.6/
- 4 Major Open Source OS Distributions:
  - ① **R**ed Hat / Fedora/Centos – (Redhat $s, Fedora/Centos – FREE)
  - ② **S**USE/**O**penSUSE - Enterprise FREE
  - ③ **D**ebian – Popular on small Linux platforms (e.g. Raspberry PI)
  - ④ **U**buntu – (e.g. Our UW Bothell Lab OS!)

Summary

- OS is the SW that manages the computer hardware and provides the application program run-time environment.
- Programs run in main memory that is optimized according to speed, size and cost. Main memory is volatile.
- Secondary storage (e.g. disks) are used to hold non-volatile programs and data in a typical computer system
- Contemporary computer architectures today are SMP Multicore systems connected by a shared memory and cluster-based parallel and HPC systems connected by a LAN.
- For operational protection OS is divided into Kernel mode and User mode.
- Processes are the fundamental unit of work in an operating system.
- OS contain protection measures for securing access to resources by users and processes
- Open Source Linux Systems provide an easy way to dissect OS functionality in a legal free operating environment

Discussion 2

There are two main categories of operating systems available in the world --*Open Source and Closed Source.*

List several PRO/CONs of each category.

Open Source		Closed Source	
PROS	CONS	PROS	CONS

Linux Lab Orientation

Linux Commands

Little tips to make life easier in the Linux lab ☺

Basic Commands: (Brackets:<> means required, [] means *optional*)	
cd <dir name>	Change directory
mkdir <dir name> rmdir <dir name>	Make directory Remove directory (get in the habit of using this)
ls ls -l ls -al	Basic listing of the current directory Listing in list format with more details Listing in list format with all details
mv <source> <dest>	Move files and/or entire directories
rm rm -r rm -rf	Remove a file Remove a full path (VERY DANGEROUS–USE WITH CARE) FORCED recursive remove (EVEN MORE DANGEROUS!!)
cp <source> <dest> cp -r <source> <dest>	Copy files Copy files recursively (for full multilevel folder cp)
man <command name>	Unix manual for a given command
chmod <permissions> <filename>	Changes permissions on a file or directory Ex. $chmod 700 <directory> - readable only by owner $chmod -R 755 <directory> - sets contents readable and writable by anyone
~mynetid . ..	Shortcut name for your home directory Shortcut name for the current directory Shortcut name for the parent folder above

Linux Lab

Little tips to make life easier in the Linux lab ☺

TASK	DESCRIPTION
Servers	UW1-321 : 16 Ubuntu x86-64bit Machines #(00-15) 00-15 connected to 1000Mbps network
Login	$ ssh uwnetid@uw1-320-lab.uwb.edu (generic machine) $ ssh uwnetid@uw1-320-07.uwb.edu (specific machine)
Goto my home	$ cd ~mynetid
Transferring Files→ UWB	$ scp foo.tgz uwnetid@uw1-320-07.uwb.edu:
Transferring UWB→local	$ scp uwnetid@uw1-320-07.uwb.edu:bar.tgz .
sshkeys	$ cd ~/.ssh $ ssh-keygen (UNIX ONLY! - on Windows machines use putty-gen)
Fingerprint	md5sum foobar.tgz → 3fd1de80dfa62bce793b01663b70e46d

Printing	**Basic Printing**
	$ lpr -Puw1-320-p1 foo.java
	Printing with name on header (in postscript)
	$ a2ps -Puw1-320-p2 *.java

Editors/Compilers

NAME	DESCRIPTION
pico	Simple file.txt oriented editor
Vim	(or vi) the most widely used editor in the Unix world
Emacs	Hardcore programmer editor – very powerful in the right hands
Notepad++	Great editor on Windows platform
Textmate	Great editor on Mac platform, but check out …
Sublime	Cross platform beautiful code editor – many plugins
	…Just to name a few

Compiling	gcc filename.c → successful compile/link to ➔ a.out gcc filename.c –o filename → compiles to ➔ filename g++ filename.cpp → successful compile/link to ➔ a.out g++ filename.cpp –o filename → compile/link to ➔ filename javac Classname.java → Classname.class

Running	$ ./a.out $ ./filename $ java Classname

VIM TIPS WIKI

VIM

VIM CMD	EDITOR ACTION
<ESC>	switches to command mode (bottom of editor screen) ": commands"
:q	quits with no save
:w	:w - writes changes to file
:wq	:wq - writes changes and exits VIM
i	insert text (transitions to insert mode)
A	append to current cursor position
^	jump to the beginning of a line
$	jump to the end of a line
nY	copy n line(s) to clipboard
X	delete a character
R	Replace a character
dw	dw - delete the rest of a word
D	D - delete the rest of the line
dd	dd - delete a line
nD	[n]d - delete n lines
u	u - undo last action
<ctrl>-r	redo last command
nG	[n]G - go to line number n
<shift>G	<shift>G - go to the last line in the document
P	P - paste entire clipboard
J	J - joins the following line to the current line
/foo	finds and highlights all occurrences of "foo"
%s/foo/bar/g	global search and replace of "foo" to "bar"
gg=G	**Coolest command of them all!**

IDEs

NAME	DISCRIPTION
Netbeans	Free IDE from Sun/Oracle – great, easy to use Java IDE
Eclipse	Most popular Open Source IDE – takes some time adapting, terse
Visual Studio	Very powerful IDE for C/C++/C# - Windows platform only
IntelliJ IDEA	Industrial Strength Cross Platform/Cross Language IDE – large plugin ecosystem
Xcode	Mac OSX IDE (mainly) for Objective-C, Java possible
	…Just to name a few

…Next Class we'll Show IDE Use

Homework

- Exercises 1.1, 1.17, 1.25, 1.30
- Upload your answers to Canvas assignment 1
- Due by Sunday midnight.