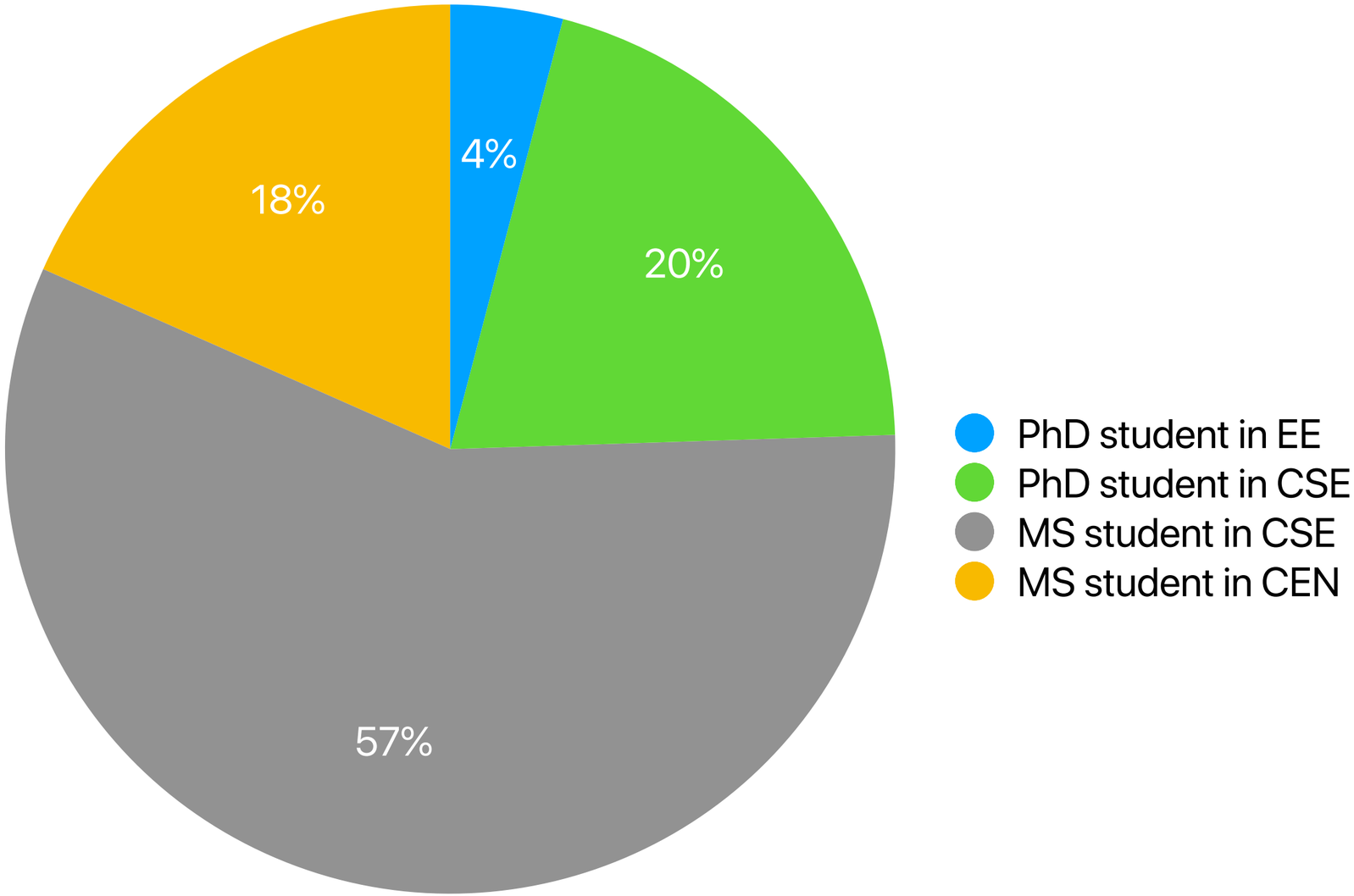
First Day of CS202, 2022 Winter

Hung-Wei Tseng

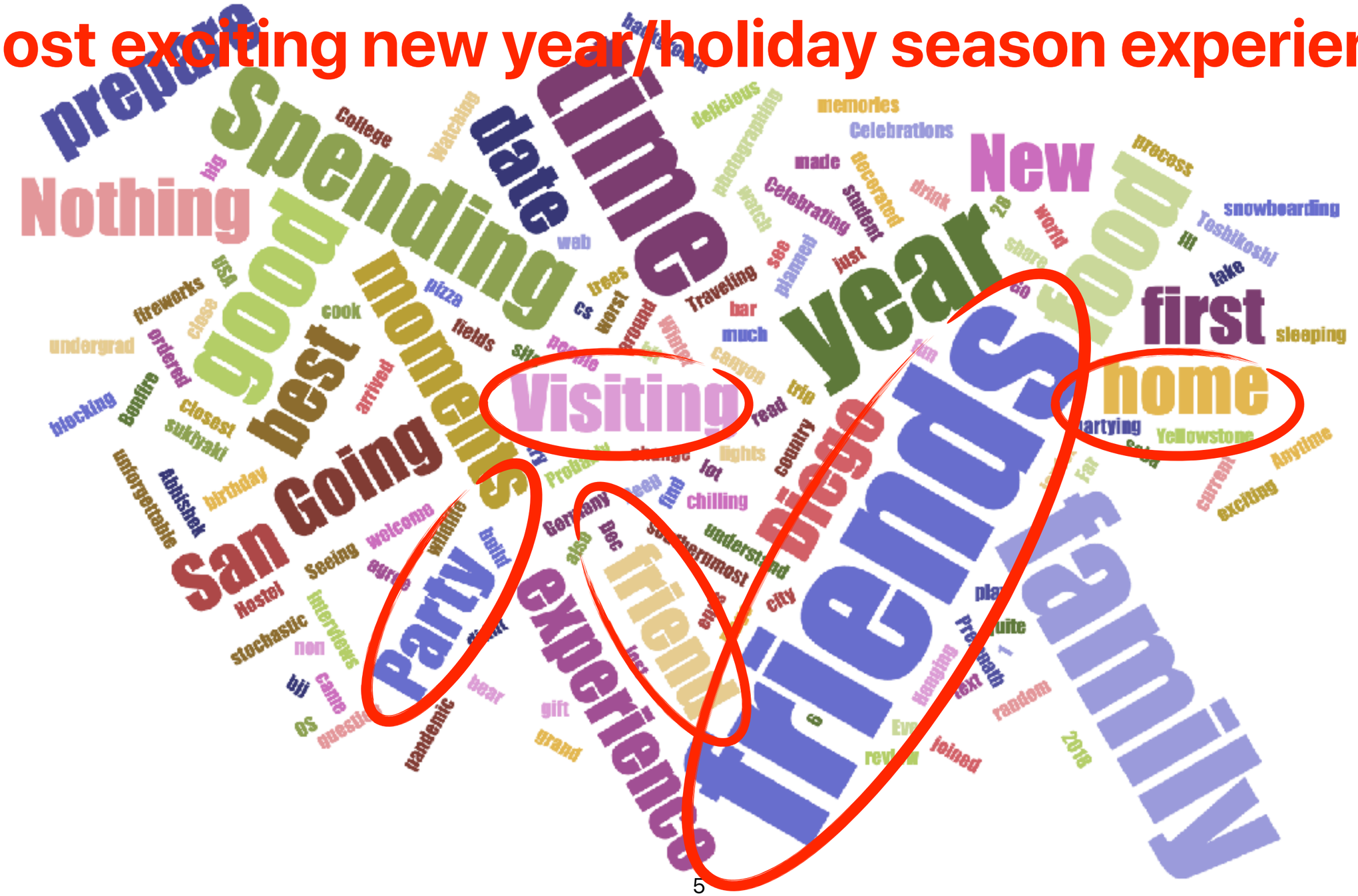
Fun facts about CS202, 2022 Winter

Who we are


4%
20%
57%
18%
PhD student in EE
PhD student in CSE
MS student in CSE
MS student in CEN


Why we are here
degree
requirement
computers
advanced
want
systems
undergrad
Requirements
operating
interested
research
complete
learn
help
course
core
os
knowledge
fulfill
love
like
professor
understanding
interest
system
understand
different
exam
MS
better
high
also
required
enjoyed
area
cs
graduate
workloads
adapt
gain
distributed
comprehensive
Design
liked
working
Technical
took
aspects
curriculum
teaching
structures
optional
coreqreq
interesting
quarter
build
questions
depth
project
kernel
performance
concepts
Recommendation
fundamental
machine
sounds
practical
looking
virtual
architecture
security

Most exciting new year/holiday season experience

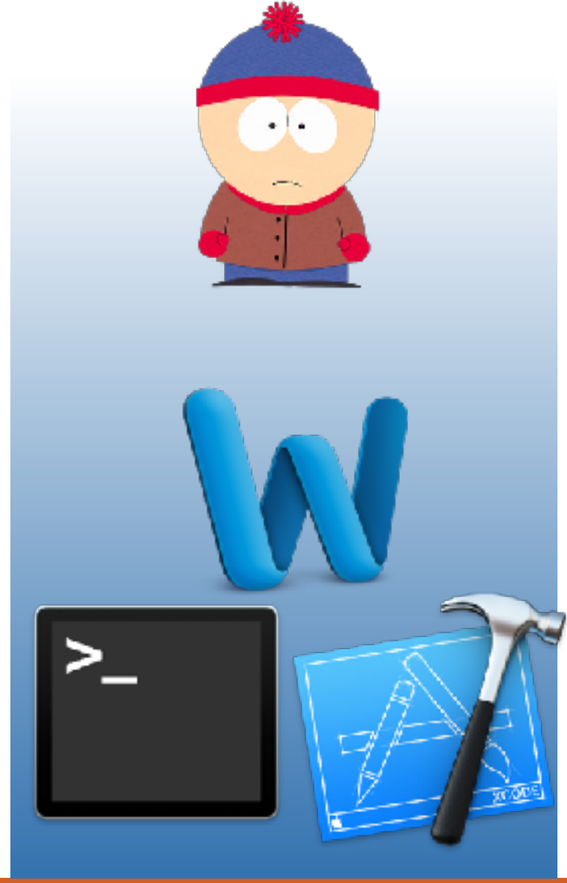
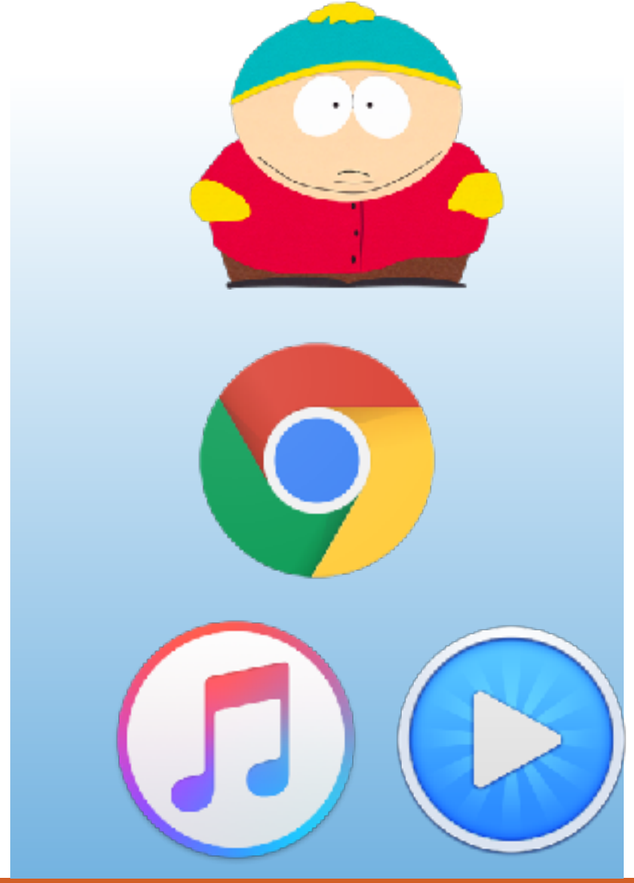
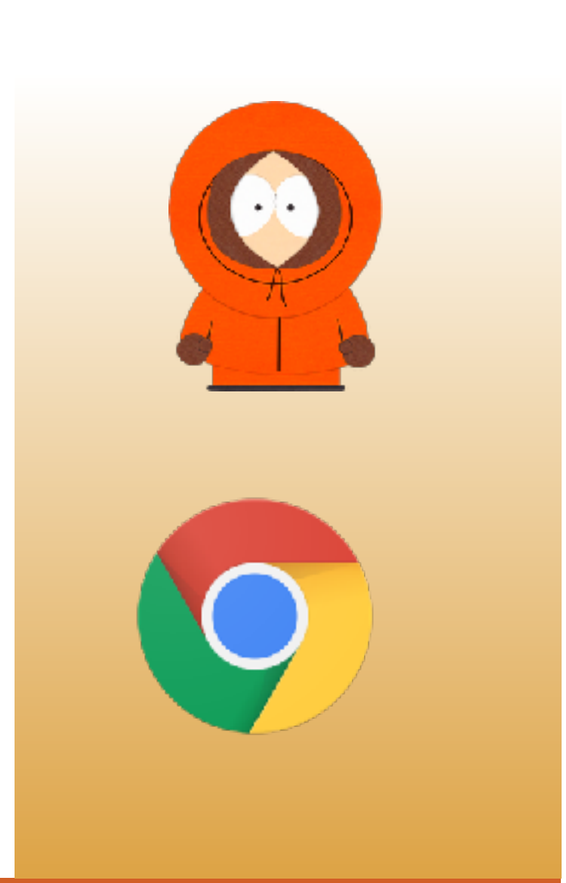


https://upload.wikimedia.org/wikipedia/commons/d/d9/EDSAC_2_1960.jpg

The usage model of computers in the past...

What releases human beings from the queues?

The usage model of computers in the past…

Operating System

Why is there an operating system?

- Make it easy to run programs
- Enable programs to interact with devices
- Allow programs to share hardware resources
- Support multithreaded programs
- Execute programs efficiently
- Low overhead
- Store data safely
- Secure

What modern operating systems support?

- **Virtualize** hardware/architectural resources
  - Easy for programs to interact with hardware resources
  - Share hardware resource among programs
  - Protect programs from each other (security)
- Execute multithreaded programs **concurrently**
  - Support multithreaded programming model
  - Execute multithreaded programs efficiently
- Store data **persistently**
  - Store data safely
  - Secure

Fun facts about operating systems

The first operating system — GMOS

- The first operating system used for real work was GM-NAA I/O, produced in 1956 by General Motors' Research division for its IBM 704.
- Functionality
  - Batch processing — The main function of GM-NAA I/O was to automatically execute a new program once the one that was being executed had finished
  - I/O virtualization — provided common access to the input/output devices

https://en.wikipedia.org/wiki/GM-NAA_I/O

Batch systems

- Benefits
  - You don’t have to be physically in the line, just drop your cards and take the result later
  - Keep the computer running
- Drawbacks
  - Head-of-line blocking
  - Cannot terminate a process in the middle
  - Cannot communicate among different machines
  - Hard to debug
  - ……

The first "portable" operating system — UNIX

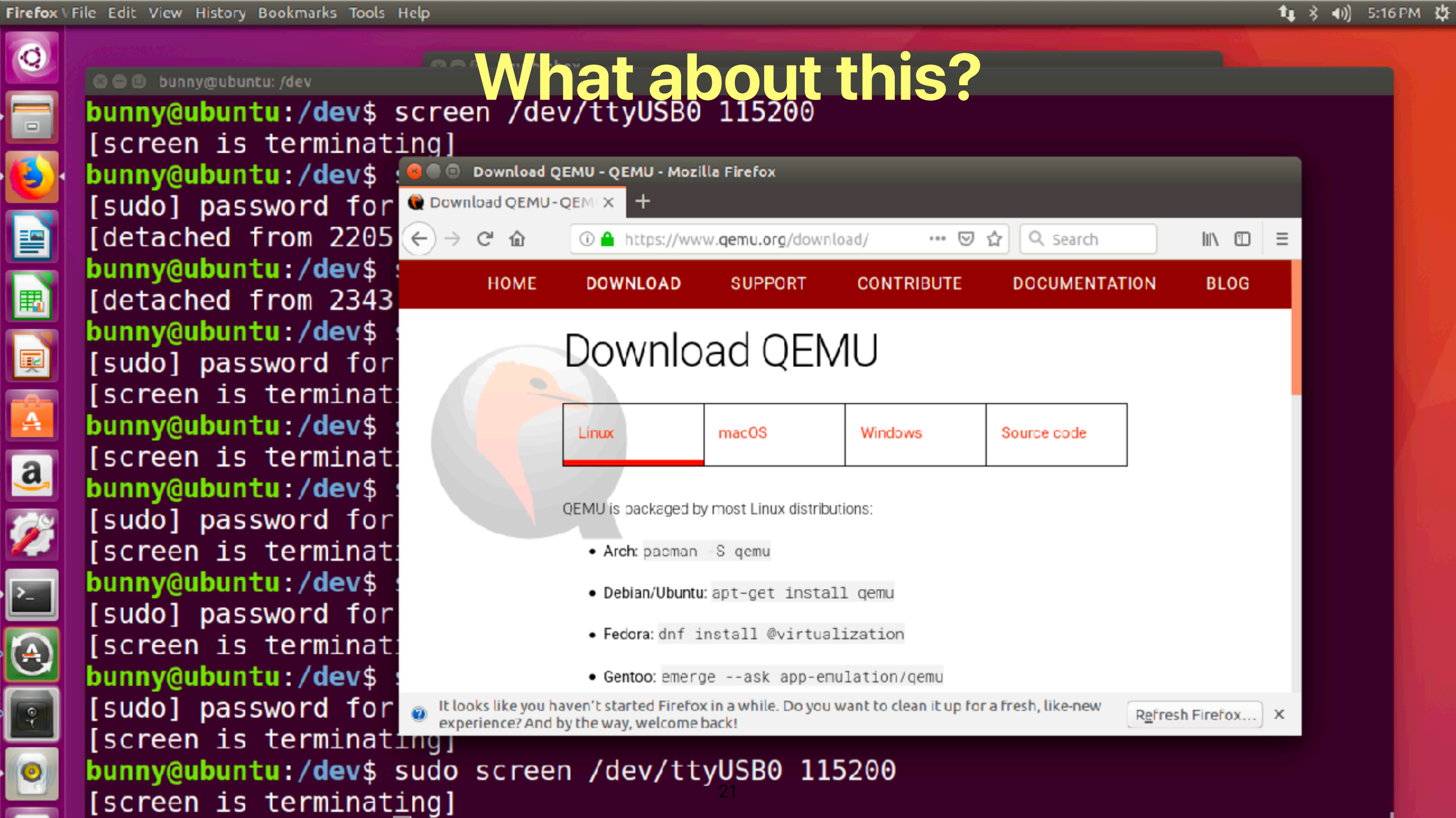
- Created in AT&T Bell Labs, a project leading by Ken Thompson and Dennis Ritchie — Started in 1969, internally public in 1971, public in 1973
- Also the first OS written in a "high-level language" — Closely tied to the development of the C programming language
  - Large portion of UNIX version 2 was written in C (version 1 was written in assembly)
  - Unix was one of the first operating system kernels implemented in a language other than assembly
  - Easier to port to many other platforms
- Support **multiple users**
- Support **interprocess communication**
- **No GUI**

UNIX (cont.)

- Descendants
  - BSD (Berkeley Software Distribution)
    - FreeBSD, OpenBSD, NetBSD
    - The base of Apple's MacOS X and iOS
  - Solaris
  - IBM AIX
- Affected
  - Linux
    - Started in 1983 by Richard Stallman
    - Linus Torvalds, principal developer of the Linux kernel

```
cover_letter2.pdf                       q97-2.jpg
cv.tar.gz                               q98-1.jpg
cv2                                     q98-2.jpg
cxbook-search.pdf                       q99-1.jpg
deadlines.pdf                           q99-2.jpg
docs                                    referenceform.pdf
e00-1-1.jpg                             schools.pdf
e00-1-2.jpg                             umac.pdf
e01-1-1.jpg                             wms
e01-1-2.jpg                             wu94envy.pdf
e98-1-2.jpg                             yangc.pdf
e98-2-2.jpg                             ?C?L??????.pdf
e99-1-1.jpg                             ?w?x
e99-1-2.jpg
bsd1 [/home/master/92/r92022] -r92022- cd htdocs/
bsd1 [/home/master/92/r92022/htdocs] -r92022- ls -altr
total 16
-rw-r--r--+  1 r92022  graduate   153 Sep 17  2006 index.html~
-rw-r--r--+  1 r92022  graduate   154 Sep 17  2006 index.html
drwxr-xr-x+  2 r92022  graduate  4096 Sep 17  2006 .
drwxr-xr-x+ 36 r92022  graduate  4096 Aug  7  2010 ..
bsd1 [/home/master/92/r92022/htdocs] -r92022- uname -a
FreeBSD bsd1.csie.ntu.edu.tw 10.3-RELEASE-p5 FreeBSD 10.3-RELEASE-p5 #30: Sun Jul 10 10:30:27 CST
2016     root@:/usr/obj/usr/src/sys/WSBSD  amd64
bsd1 [/home/master/92/r92022/htdocs] -r92022-
```

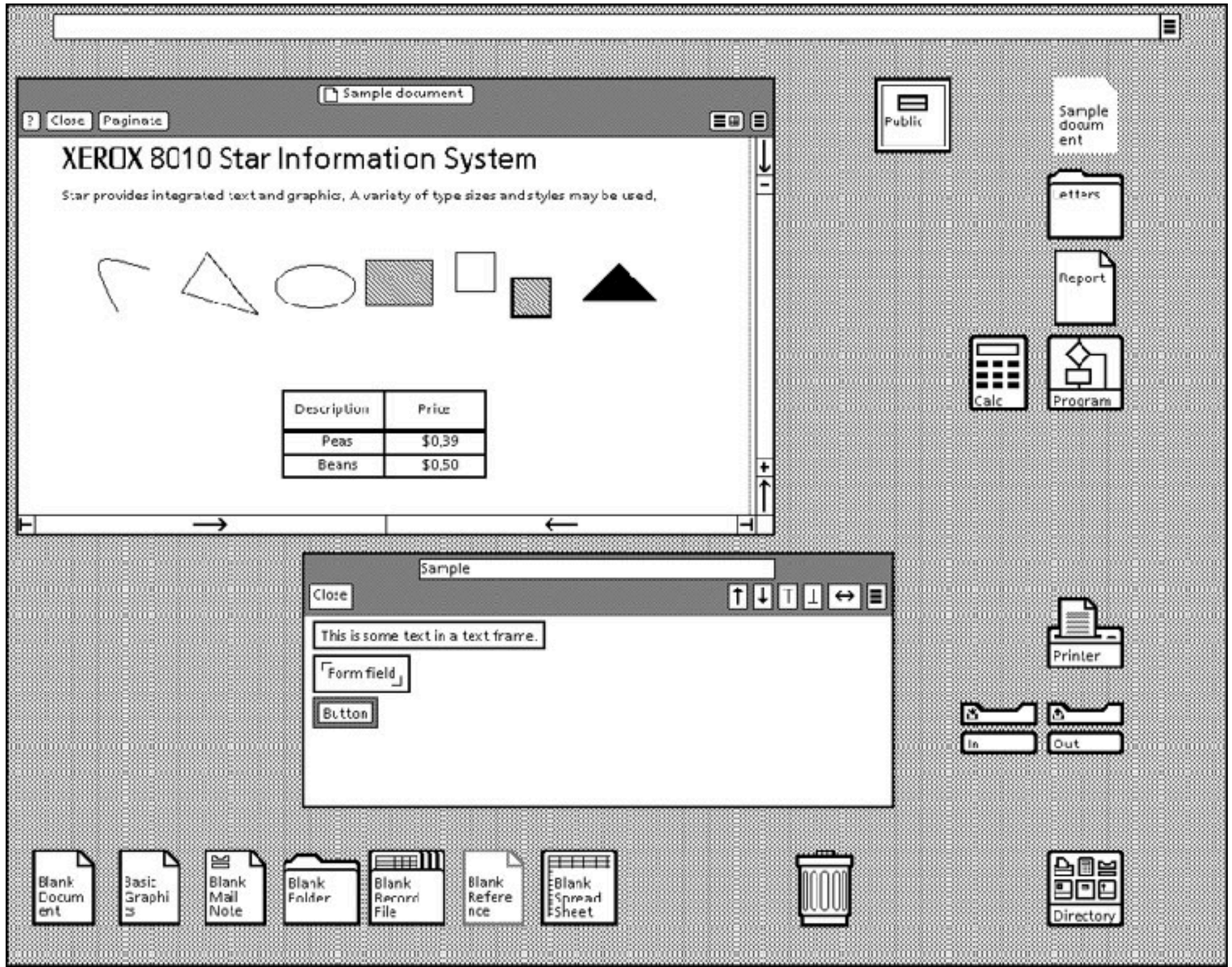
What about this?



The most popular OS in the 80s — DOS

- Disk Operating System
  - Originally Quick and Dirty Operating System
  - Introduced in 1981 for IBM PC based on 8086/8088
- Only 640KB memory available for applications
  - No virtual memory
  - Need quite a few tricks (EMS, XMS, QEMM, and etc.) to use all memory that you installed on the computer
- No multi-user, no multi-tasking, no multi-threading
- Notorious 8.3 filename restrictions
- No GUI
  - Now the command line environment of Windows
  - Windows is originally a graphic user interface running on DOS — like X-Window

The first GUI-based OS — Xerox PARC

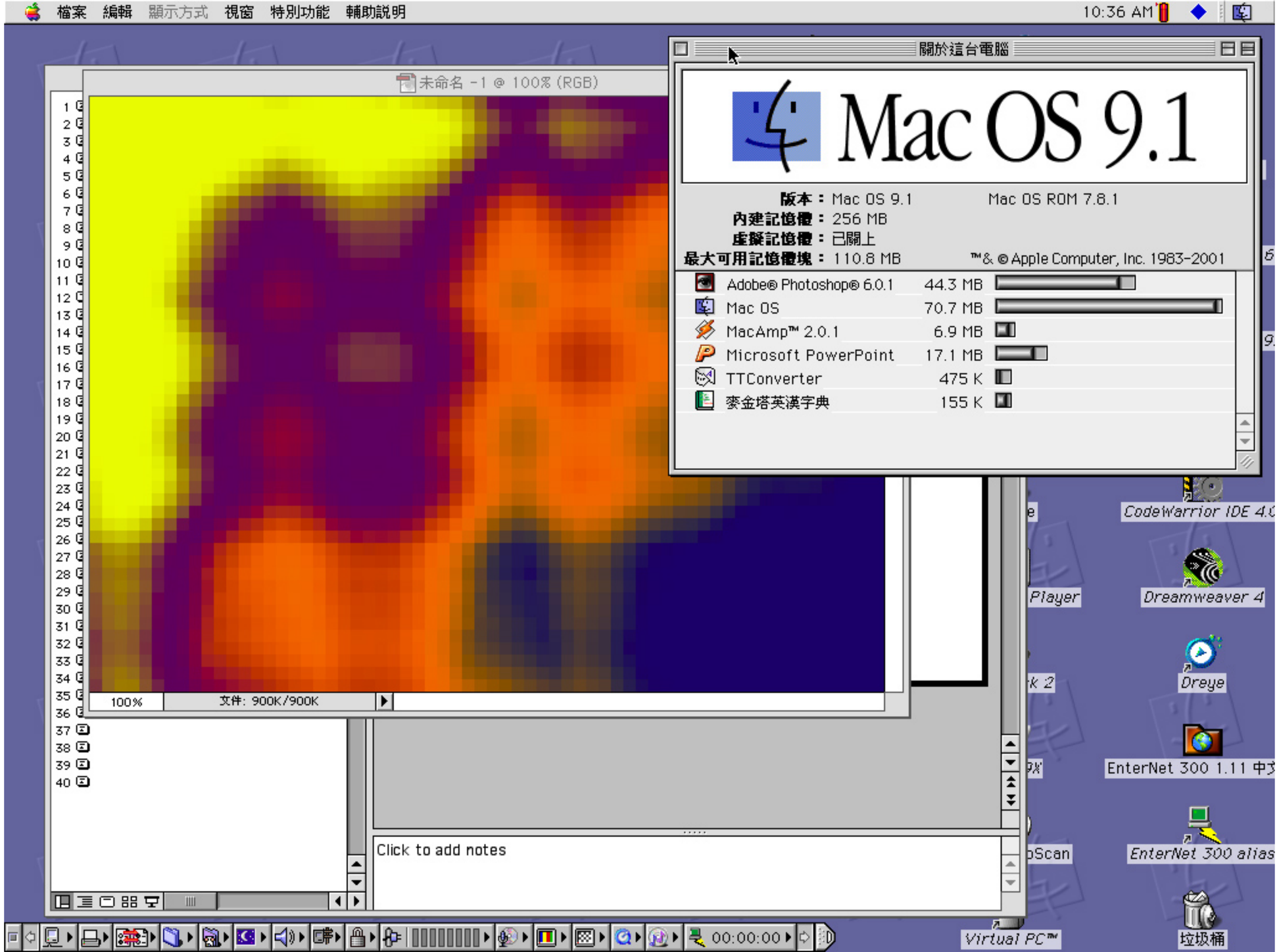


The first GUI-based OS — Xerox PARC

- Designed for Star Information system.
- GUI elements: bitmapped display, windows, folders, icons, Ethernet networking, file and print servers, and the mouse.
- Object-oriented design
- Editor: “What You See Is What You Get” (WYSIWYG)
- Only around 25,000 devices were sold

The first popularized (kind of) GUI OS — Classical Mac OS

- Released in 1984 w/ the legendary Macintosh
- Adopted GUI/mouses from Xerox PARC
- The first popularized all GUI OS
- Support multitasking
- Not a multi-user system


檔案 編輯 顯示方式 視窗 特別功能 輔助說明
10:36 AM
關於這台電腦
Mac OS 9.1
版本： Mac OS 9.1
Mac OS ROM 7.8.1
內建記憶體： 256 MB
虛擬記憶體： 已關上
最大可用記憶體塊： 110.8 MB
™& © Apple Computer, Inc. 1983-2001
Adobe® Photoshop® 6.0.1 44.3 MB
Mac OS 70.7 MB
MacAmp™ 2.0.1 6.9 MB
Microsoft PowerPoint 17.1 MB
TTConverter 475 K
麥金塔英漢字典 155 K
未命名 -1 @ 100% (RGB)
100%
文件: 900K/900K
Click to add notes
CodeWarrior IDE 4.0
Dreamweaver 4
Dreye
EnterNet 300 1.11 中文
EnterNet 300 alias
垃圾桶
Virtual PC™
00:00:00

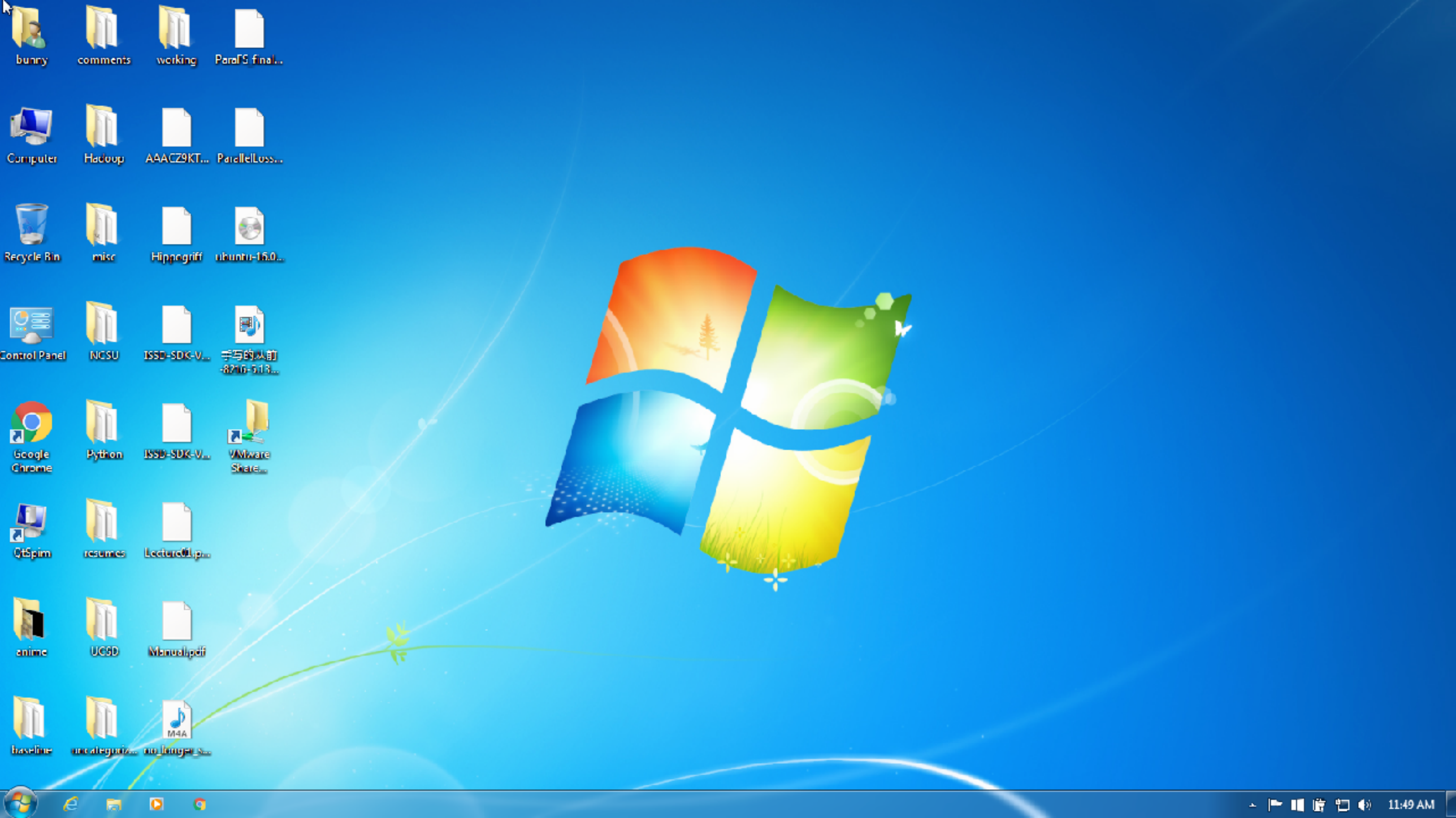
The most popular OS in late 90s — Windows 95/98/ME

- Before Windows 95, "Windows" (e.g., Windows 3.1) was just a GUI operating "environment" on DOS
  - You cannot directly boot your machine using early versions of Windows
  - Similar to X-window, Xorg in UNIX/Linux
- First full-fledged Windows OS introduced in 1995 as Windows 95


Microsoft
Microsoft
Windows 95
Microsoft Internet Explorer

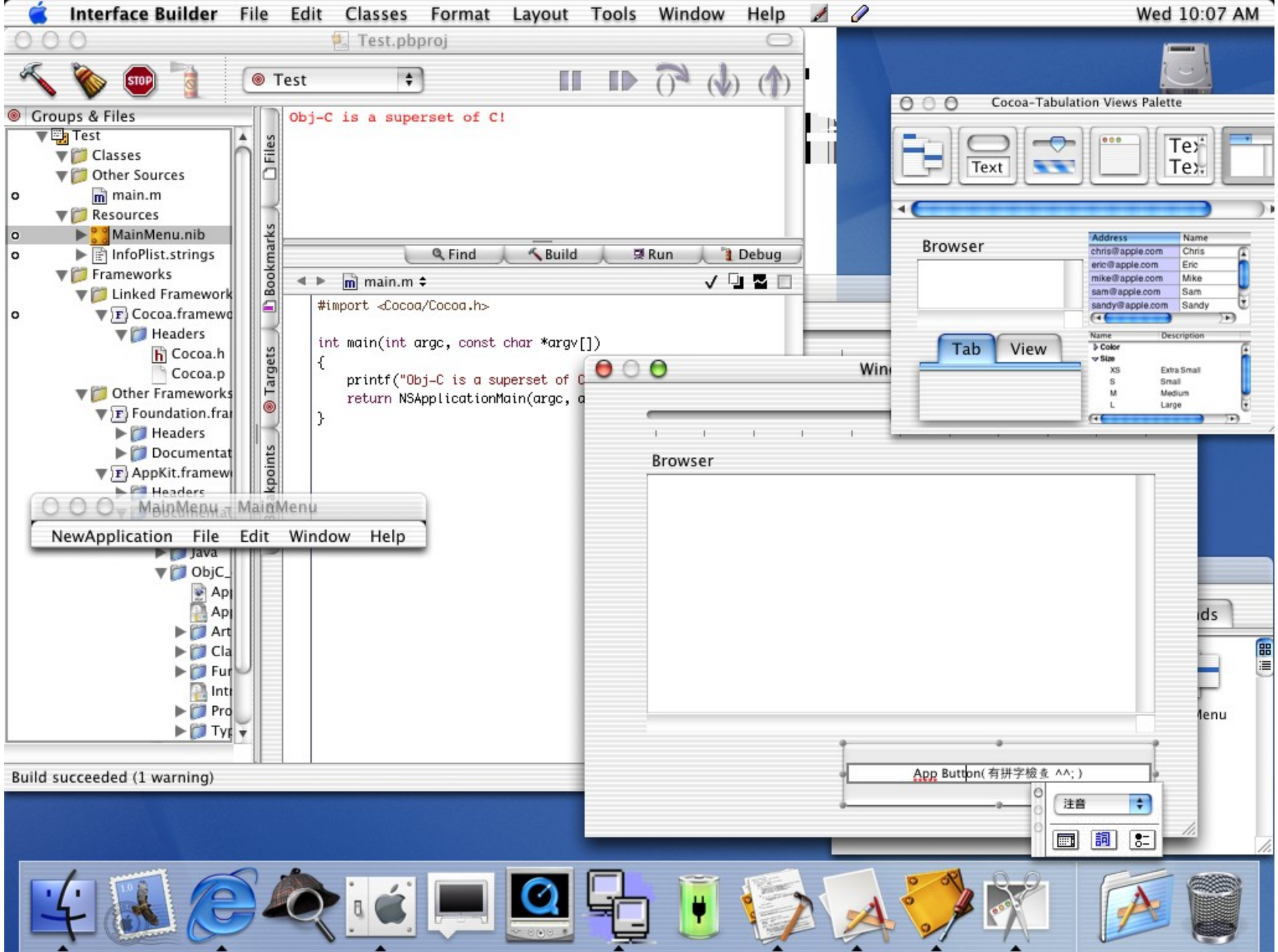
The most popular OS before 2014 — Windows NT/2000/XP/Vista/7/8

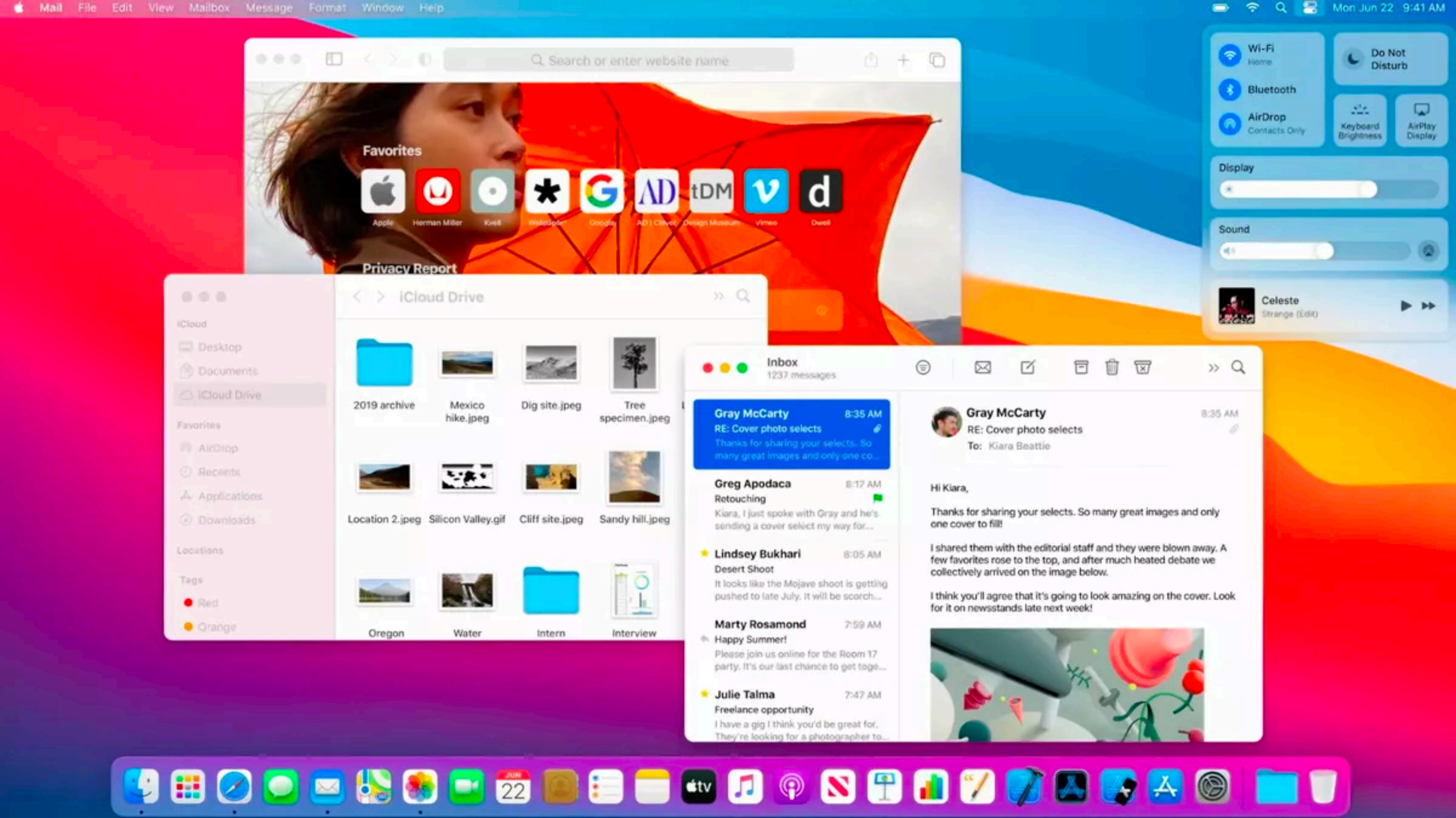
- Originally for servers, initially released in 1993
- First true 32-bit Windows OS, Windows Vista/7 started to become natively 64-bit
- Support multi-user, multi-tasking
- NTFS: more secure, modernized file system
- Different driver model than DOS/Windows 95
- Most code in C/C++, reasonably portable (IA-32, x86-64, DEC Alpha, MIPS, PowerPC, ARM, Itanium)


bunny
comments
working
ParaFS final...
Computer
Hadoop
AAACZ9KT...
ParallelLoss...
Recycle Bin
misc
Hippogriff
ubuntu-16.0...
Control Panel
NCSU
ISSD-SDK-V...
手写的从前
-8216-5138...
Google Chrome
Python
ISSD-SDK-V...
VMware Share...
QtSpim
resumes
Lecture01.p...
anime
UCSD
Manual.pdf
baseline
uncategoriz...
M4A
11:49 AM

The most popular UNIX now — MacOS X

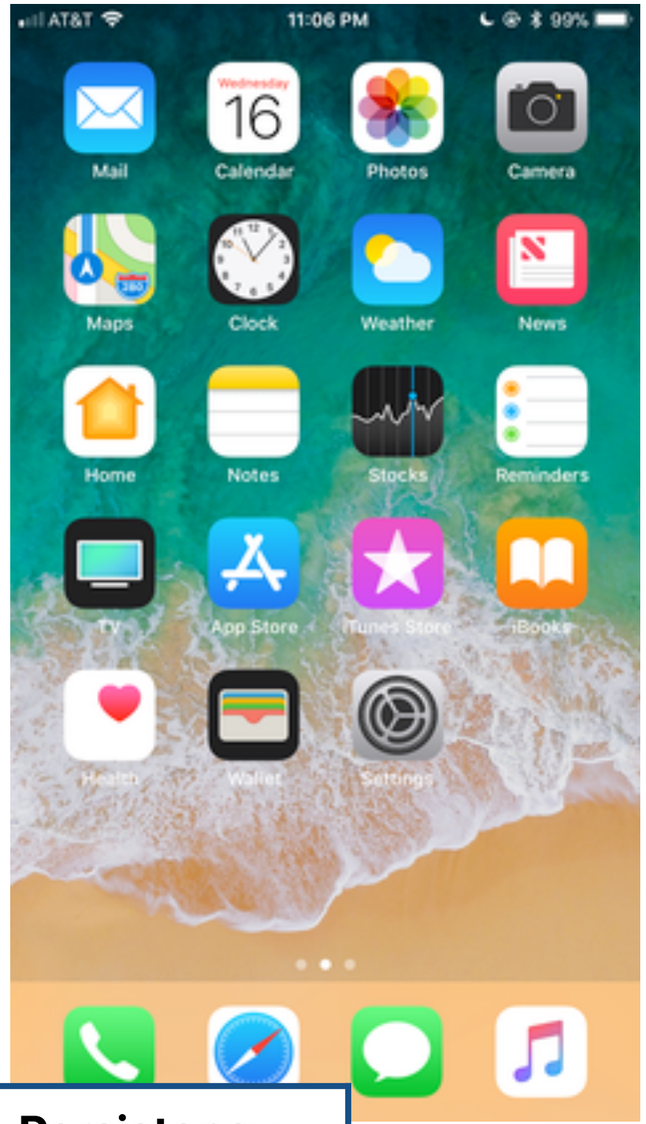
- Initially released in 2001
- Originated from NeXTSTEP, a company Steve Jobs funded after leaving from Apple in 1985
- Darwin: based on Mach and BSD kernels
  - Inherits all the good things from UNIX
  - Better integration with GUI
- Shares the same kernel with iOS


Interface Builder File Edit Classes Format Layout Tools Window Help
Wed 10:07 AM
Test.pbproj
Test
Groups & Files
Test
Classes
Other Sources
main.m
Resources
MainMenu.nib
InfoPlist.strings
Frameworks
Linked Framework
Cocoa.framewo
Headers
Cocoa.h
Cocoa.p
Other Frameworks
Foundation.fra
Headers
Documentat
AppKit.framew
Obj-C is a superset of C!
Files
Bookmarks
Targets
Find
Build
Run
Debug
main.m
#import <Cocoa/Cocoa.h>
int main(int argc, const char *argv[])
{
printf("Obj-C is a superset of C
return NSApplicationMain(argc, a
}
MainMenu – MainMenu
NewApplication File Edit Window Help
Build succeeded (1 warning)
Cocoa-Tabulation Views Palette
Text
Browser
Address
Name
chris@apple.com
Chris
eric@apple.com
Eric
mike@apple.com
Mike
sam@apple.com
Sam
sandy@apple.com
Sandy
Tab
View
Name
Description
Color
Size
XS
Extra Small
S
Small
M
Medium
L
Large
Browser
App Button(有拼字檢查 ^^;)
注音
詞


Mail File Edit View Mailbox Message Format Window Help
Mon Jun 22 9:41 AM
Search or enter website name
Favorites
Apple
Herman Miller
Knoll
Google
Design Museum
Vimeo
Dwell
Privacy Report
Wi-Fi
Home
Bluetooth
AirDrop
Contacts Only
Do Not Disturb
Keyboard Brightness
AirPlay Display
Display
Sound
Celeste
Strange (Edit)
iCloud Drive
iCloud
Desktop
Documents
iCloud Drive
Favorites
AirDrop
Recents
Applications
Downloads
Locations
Tags
Red
Orange
2019 archive
Mexico hike.jpeg
Dig site.jpeg
Tree specimen.jpeg
Location 2.jpeg
Silicon Valley.gif
Cliff site.jpeg
Sandy hill.jpeg
Oregon
Water
Intern
Interview
Inbox
1237 messages
Gray McCarty 8:35 AM
RE: Cover photo selects
Thanks for sharing your selects. So many great images and only one co...
Greg Apodaca 8:17 AM
Retouching
Kiara, I just spoke with Gray and he's sending a cover select my way for...
Lindsey Bukhari 8:05 AM
Desert Shoot
It looks like the Mojave shoot is getting pushed to late July. It will be scorch...
Marty Rosamond 7:59 AM
Happy Summer!
Please join us online for the Room 17 party. It's our last chance to get toge...
Julie Talma 7:47 AM
Freelance opportunity
I have a gig I think you'd be great for. They're looking for a photographer to...
Gray McCarty
RE: Cover photo selects
To: Kiara Beattie
8:35 AM
Hi Kiara,
Thanks for sharing your selects. So many great images and only one cover to fill!
I shared them with the editorial staff and they were blown away. A few favorites rose to the top, and after much heated debate we collectively arrived on the image below.
I think you'll agree that it's going to look amazing on the cover. Look for it on newsstands late next week!

The 2nd most popular mobile OS — iOS

- Share the same kernel foundation with MacOS X
- The 2nd most popular mobile OS



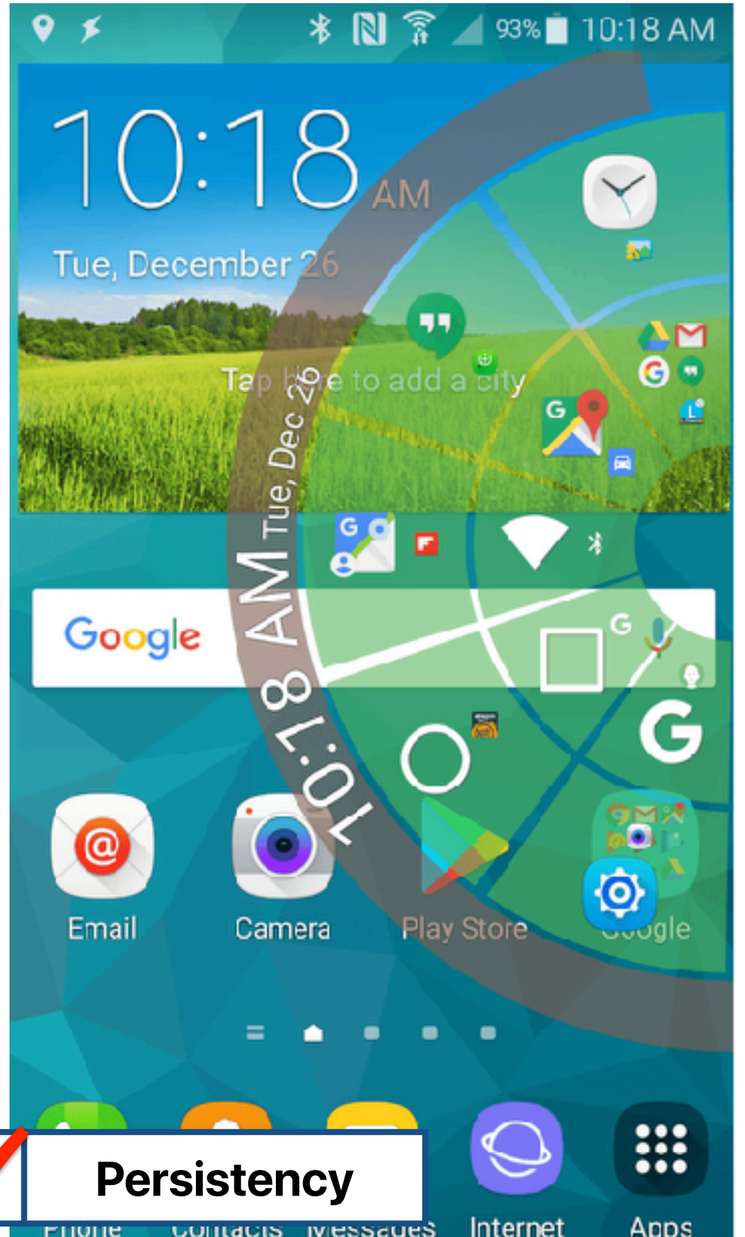
✓ Virtualization

✓ Concurrency

✓ Persistency

The most popular OS now — Android

- Based on Linux
- The most popular operating system since 2014



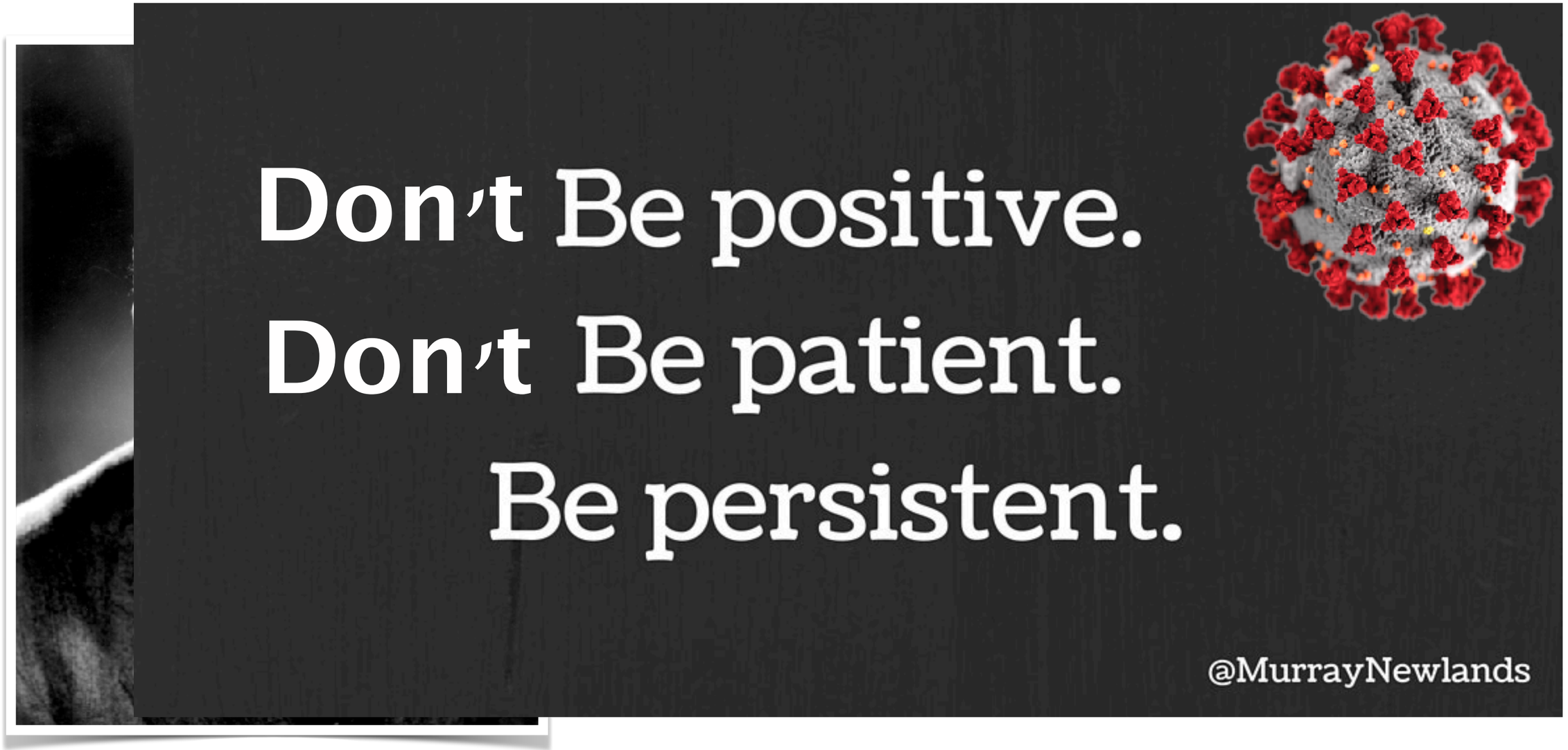
✓ Virtualization

✓ Concurrency

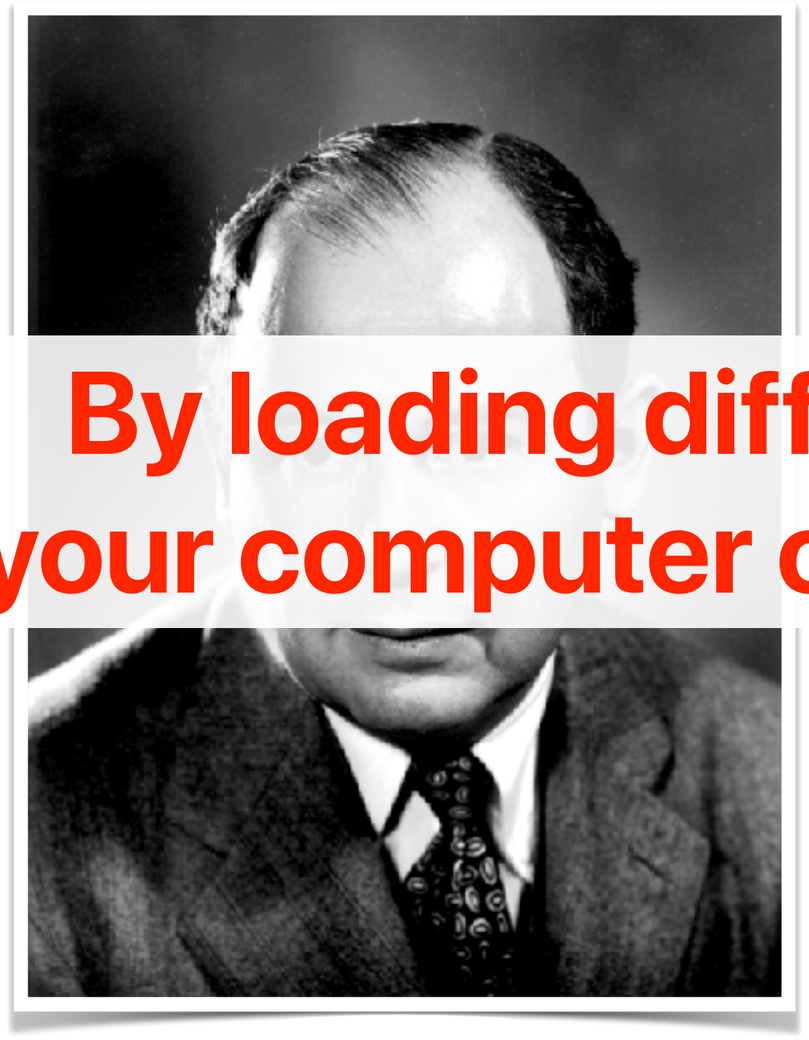
✓ Persistency

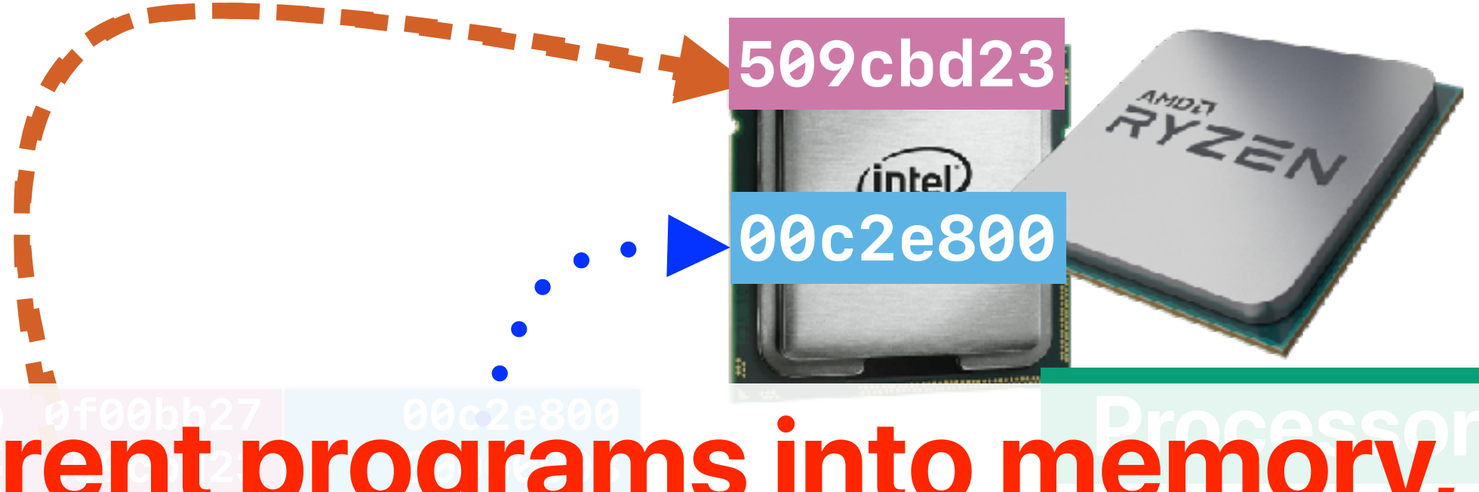
What to virtualize?

The beast: von Neumman Architecture



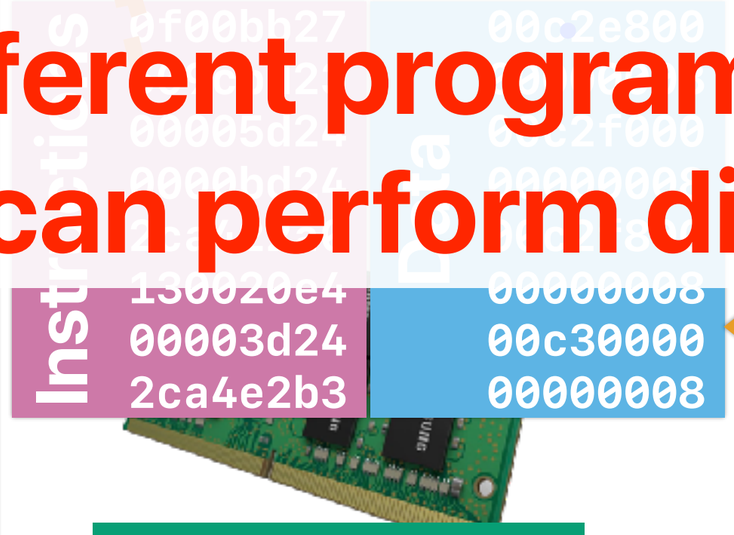
The beast: von Neumman Architecture





Processor

By loading different programs into memory, your computer can perform different functions



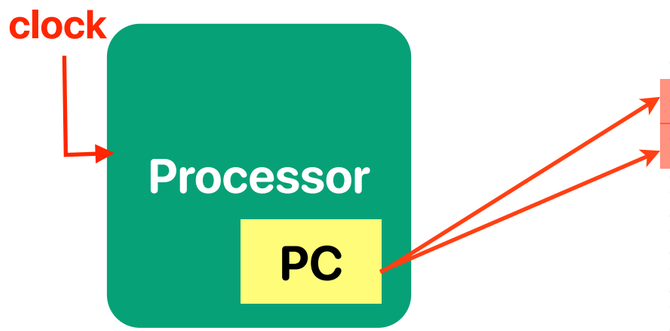
Memory

Program

Storage

How processor executes a program

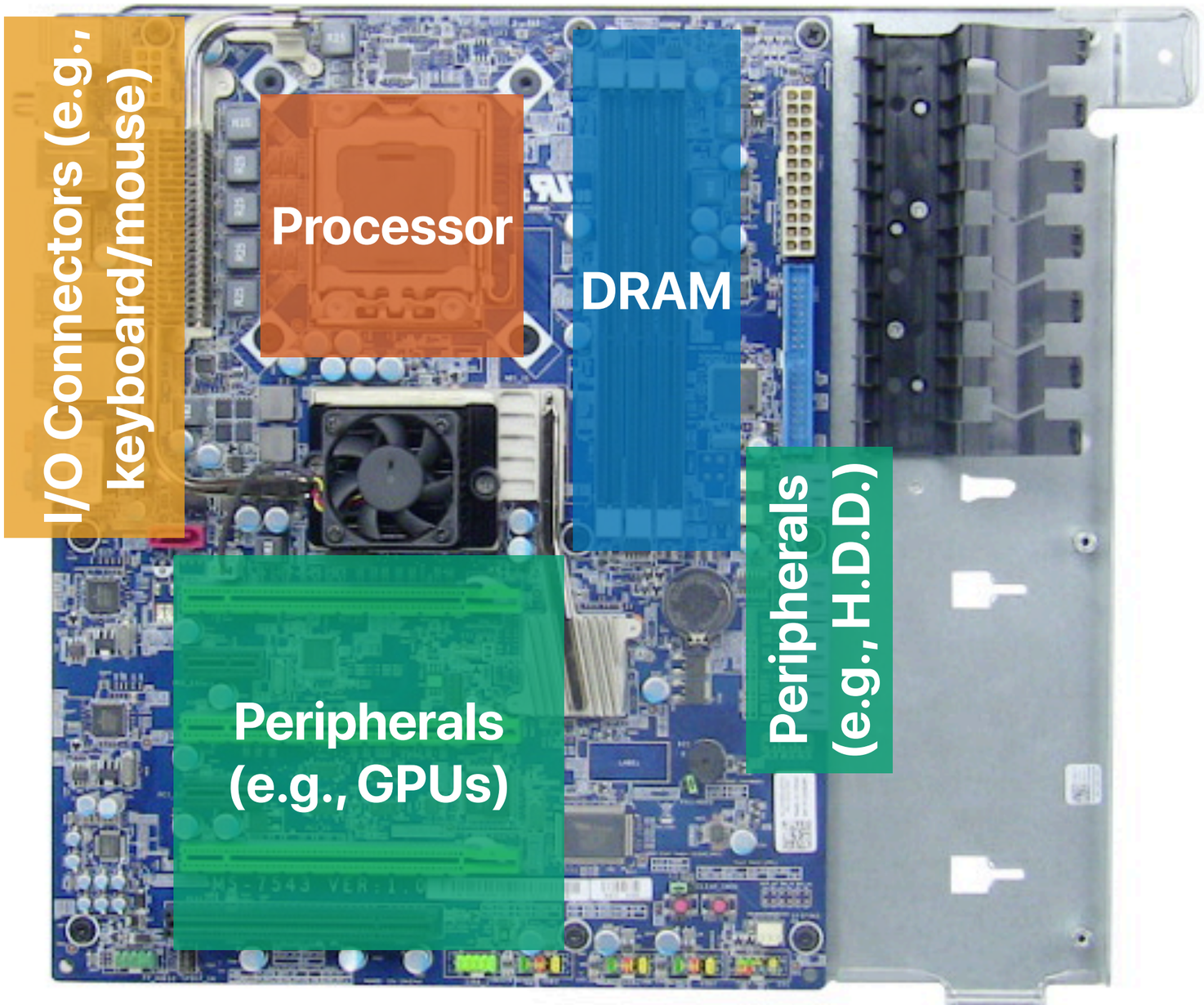
- The program counter (PC) tells where the upcoming instruction is in the memory
- Processor fetches the instruction, decode the instruction, execute the instruction, present the instruction results according to clock signals
- The processor fetches the next instruction whenever it's safe to do so



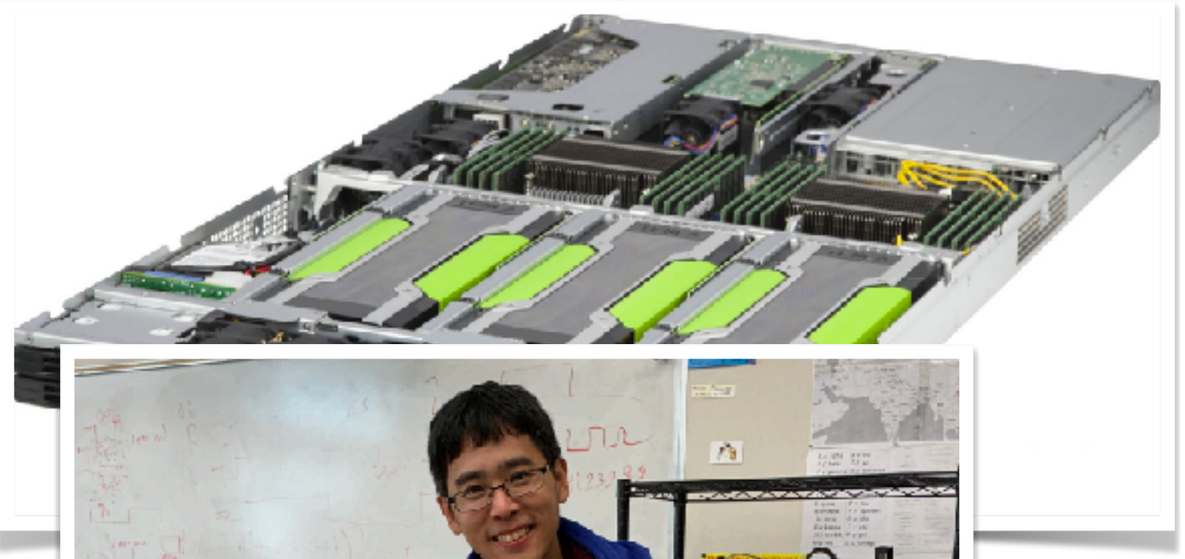
instruction memory

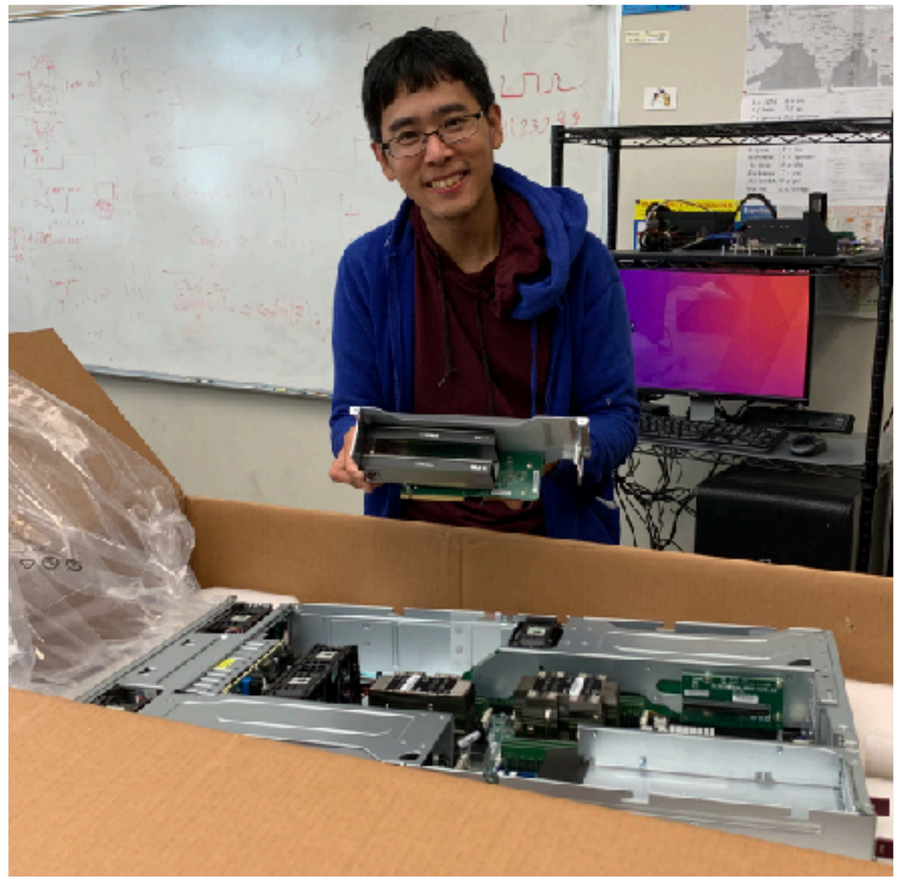
```
120007a30:   0f00bb27   ldah   gp,15(t12)
120007a34:   509cbd23   lda    gp,-25520(gp)
120007a38:   00005d24   ldah   t1,0(gp)
120007a3c:   0000bd24   ldah   t4,0(gp)
120007a40:   2ca422a0   ldl    t0,-23508(t1)
120007a44:   130020e4   beq    t0,120007a94
120007a48:   00003d24   ldah   t0,0(gp)
120007a4c:   2ca4e2b3   stl    zero,-23508(t1)
120007a50:   0004ff47   clr    v0
120007a54:   28a4e5b3   stl    zero,-23512(t4)
120007a58:   20a421a4   ldq    t0,-23520(t0)
120007a5c:   0e0020e4   beq    t0,120007a98
120007a60:   0204e147   mov    t0,t1
120007a64:   0304ff47   clr    t2
120007a68:   0500e0c3   br     120007a80
```

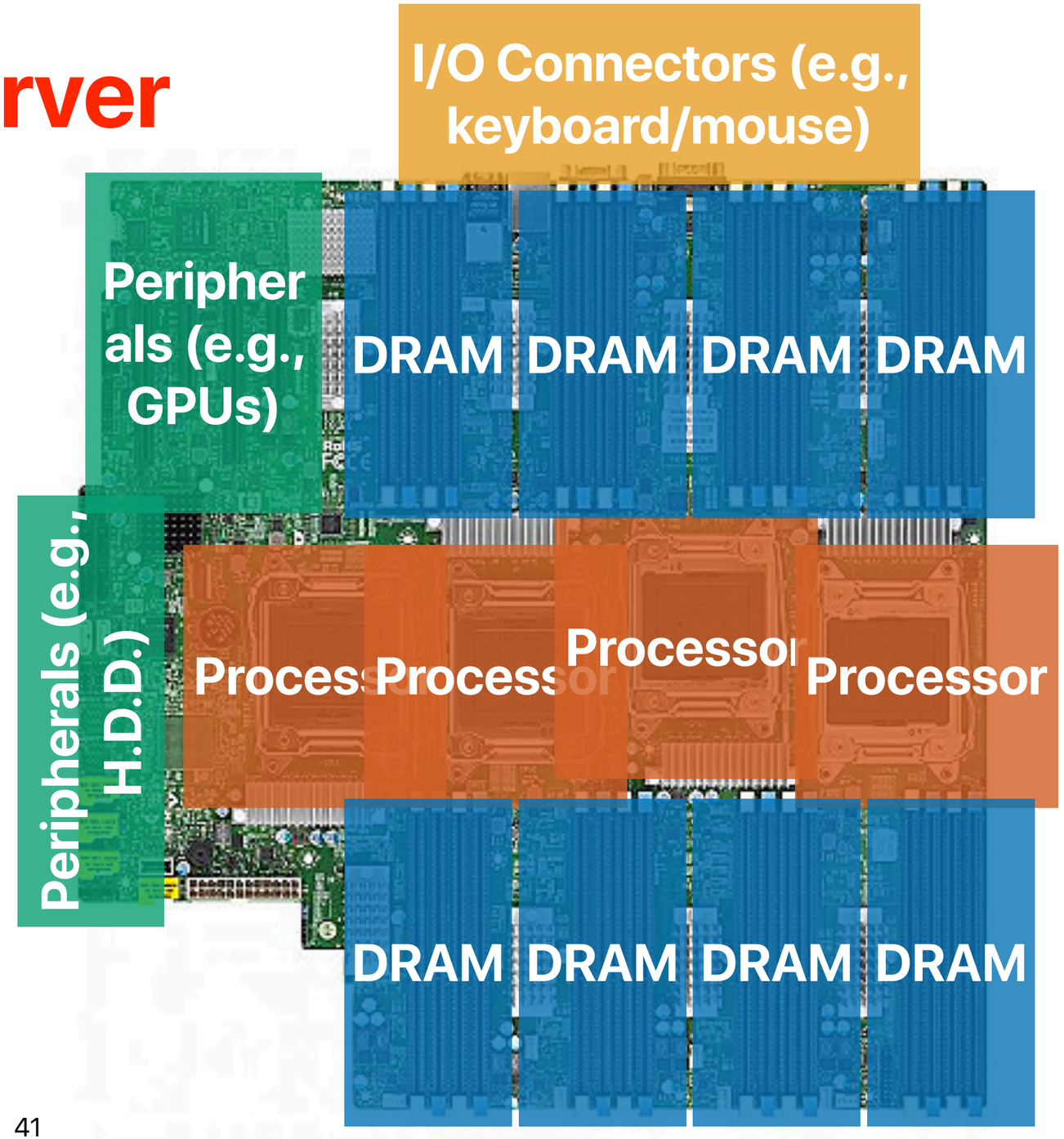
Desktop Computer



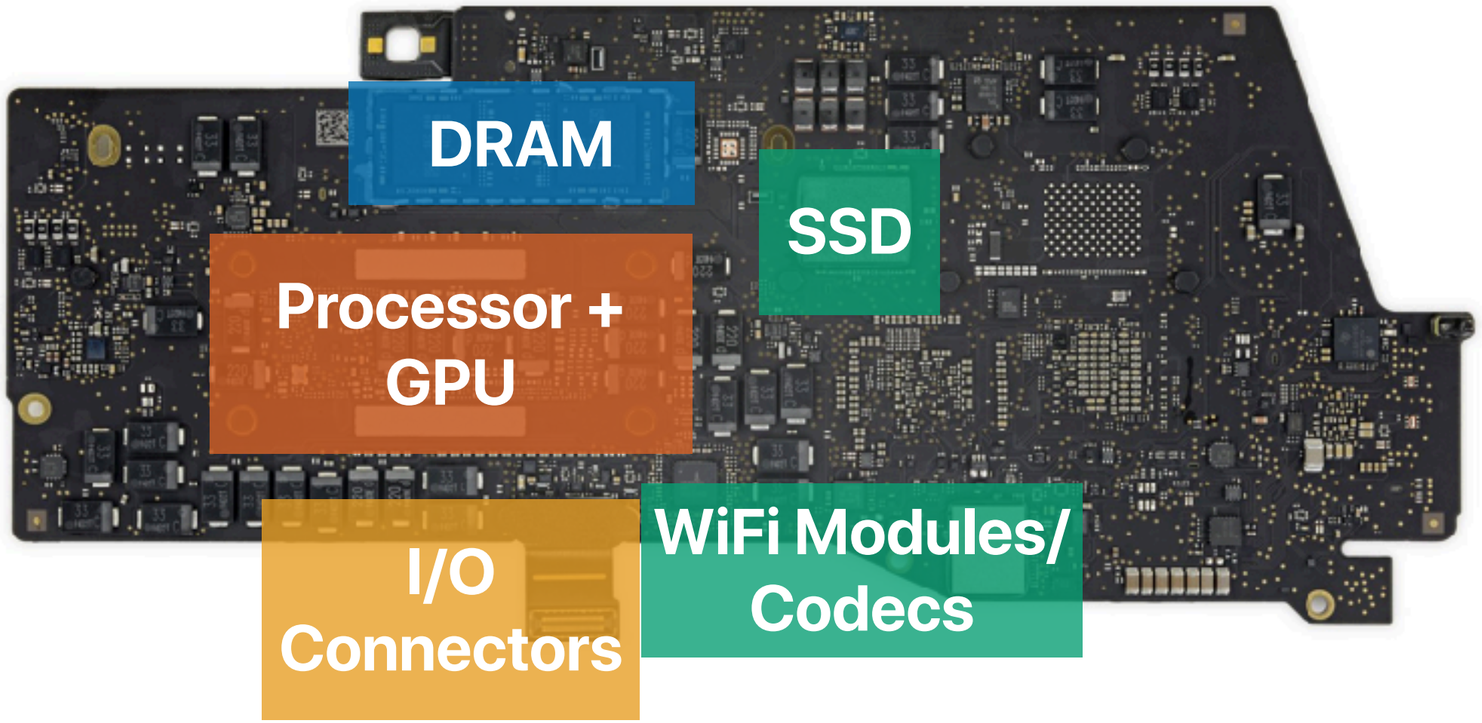
Server



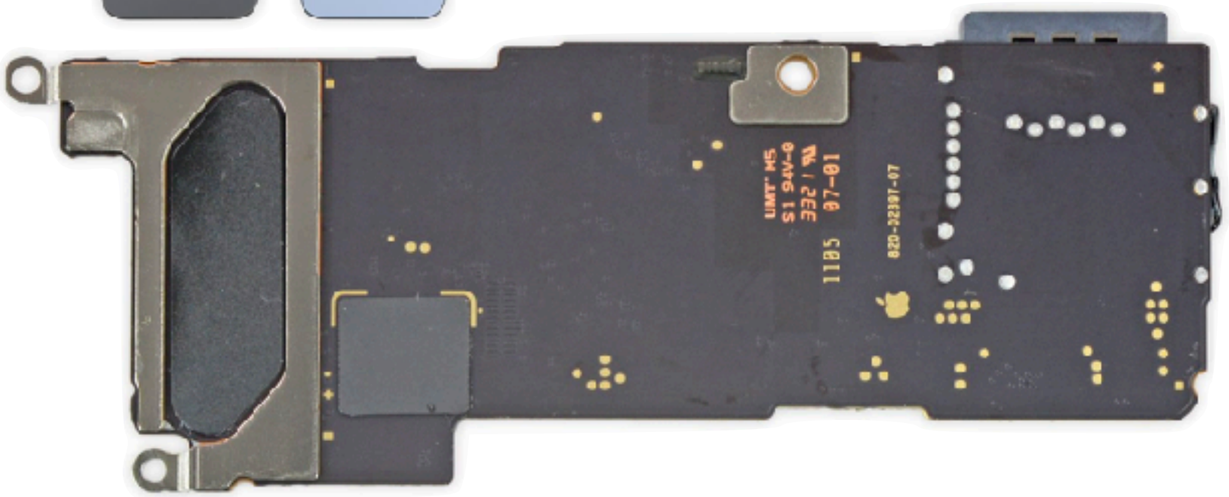


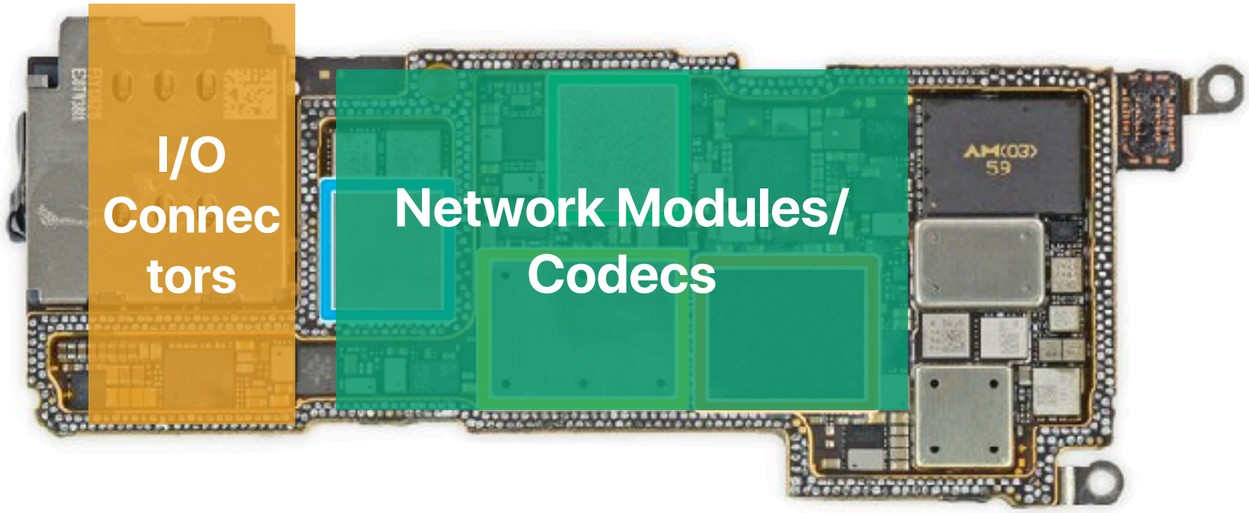

I/O Connectors (e.g., keyboard/mouse)
Peripherals (e.g., GPUs)
DRAM
DRAM
DRAM
DRAM
Peripherals (e.g., H.D.D.)
Processor
Processor
Processor
Processor
DRAM
DRAM
DRAM
DRAM

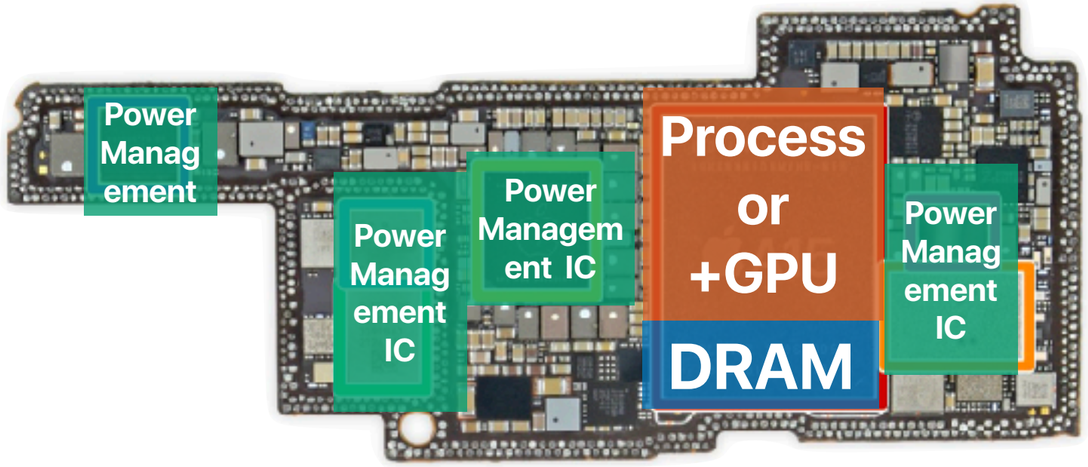
MacBook Pro 13″

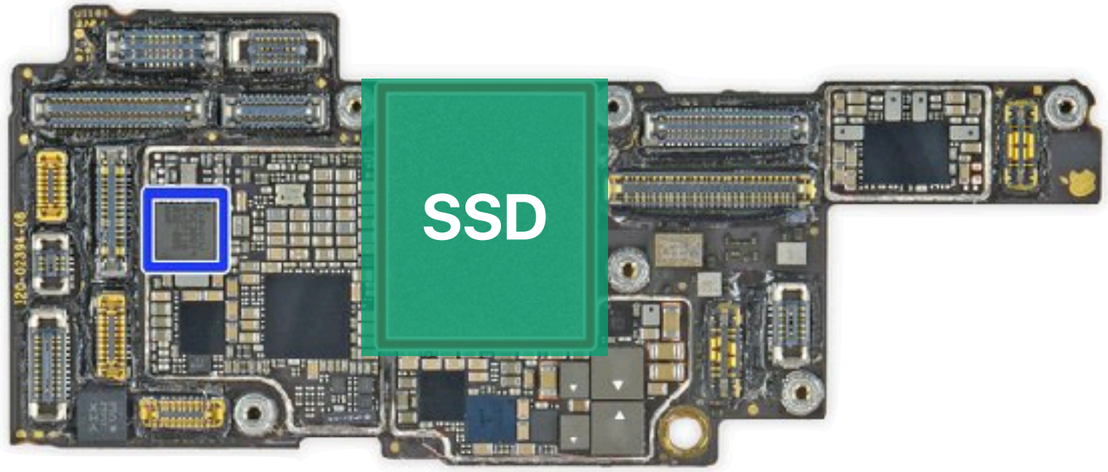


iPhone 13 Pro

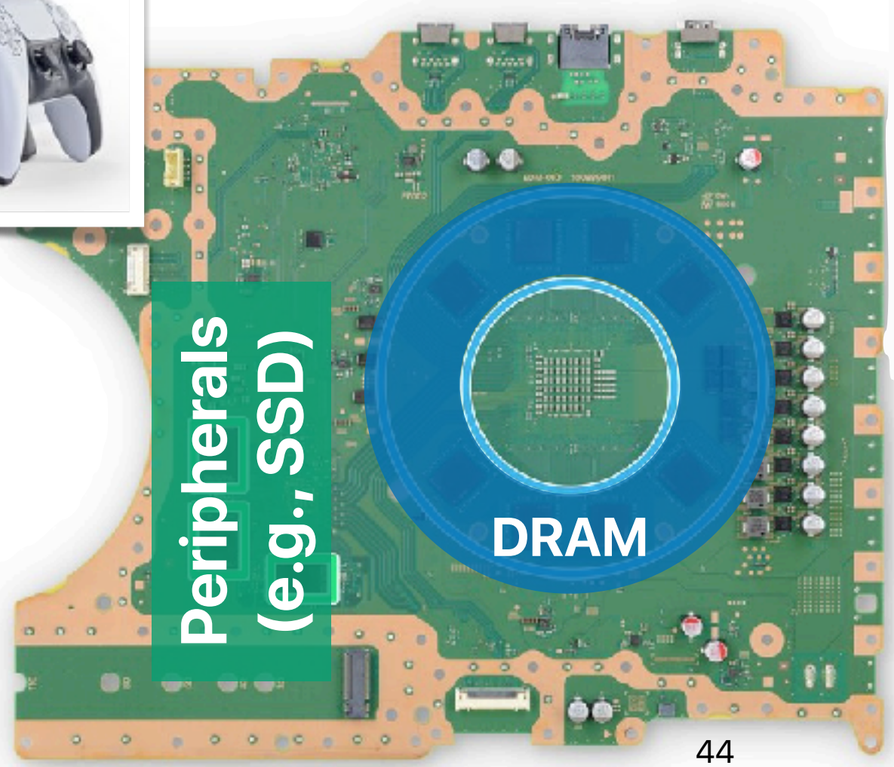


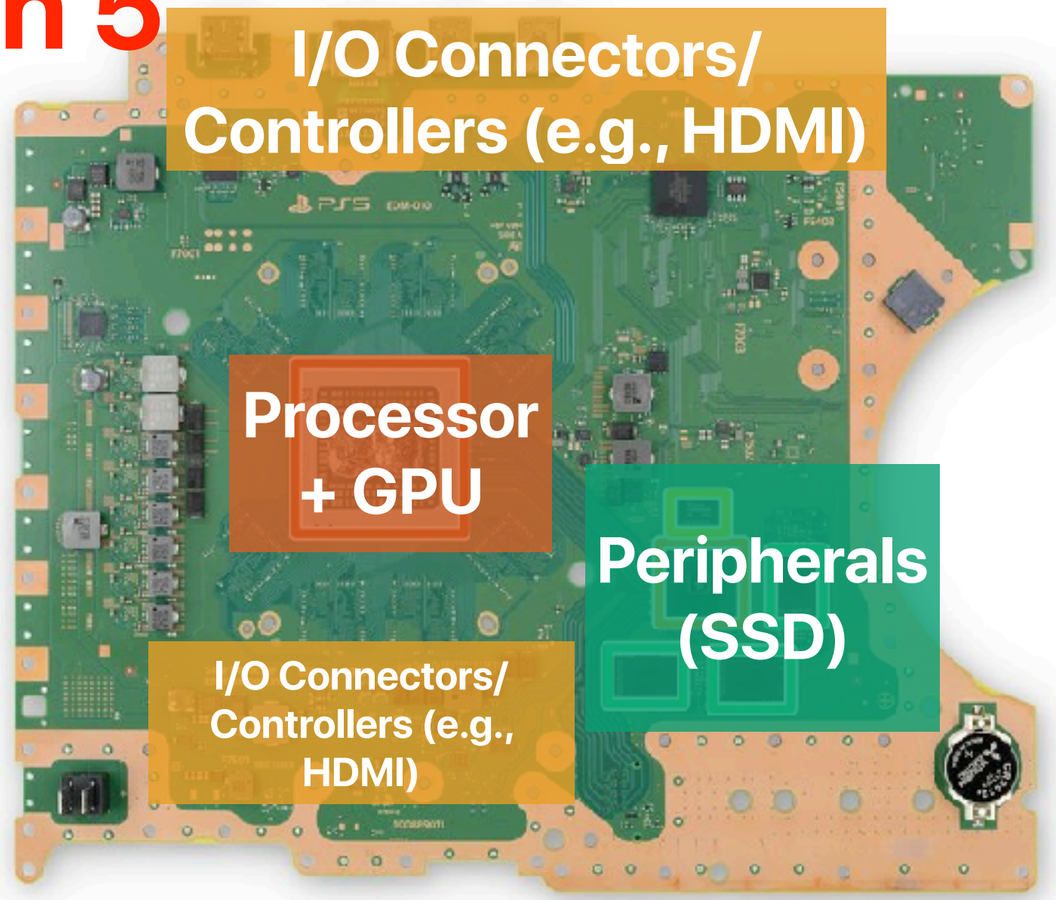




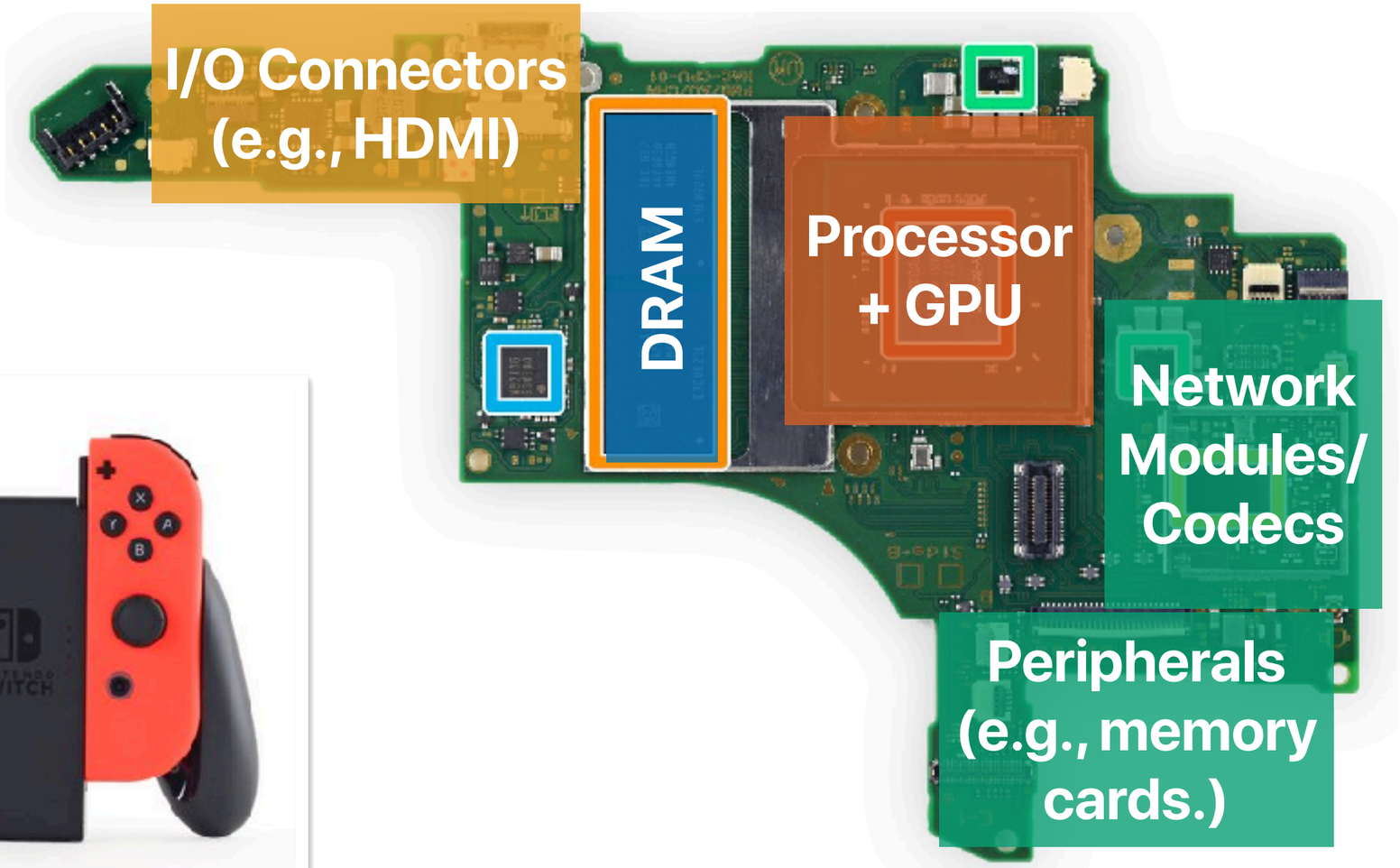


Play Station 5

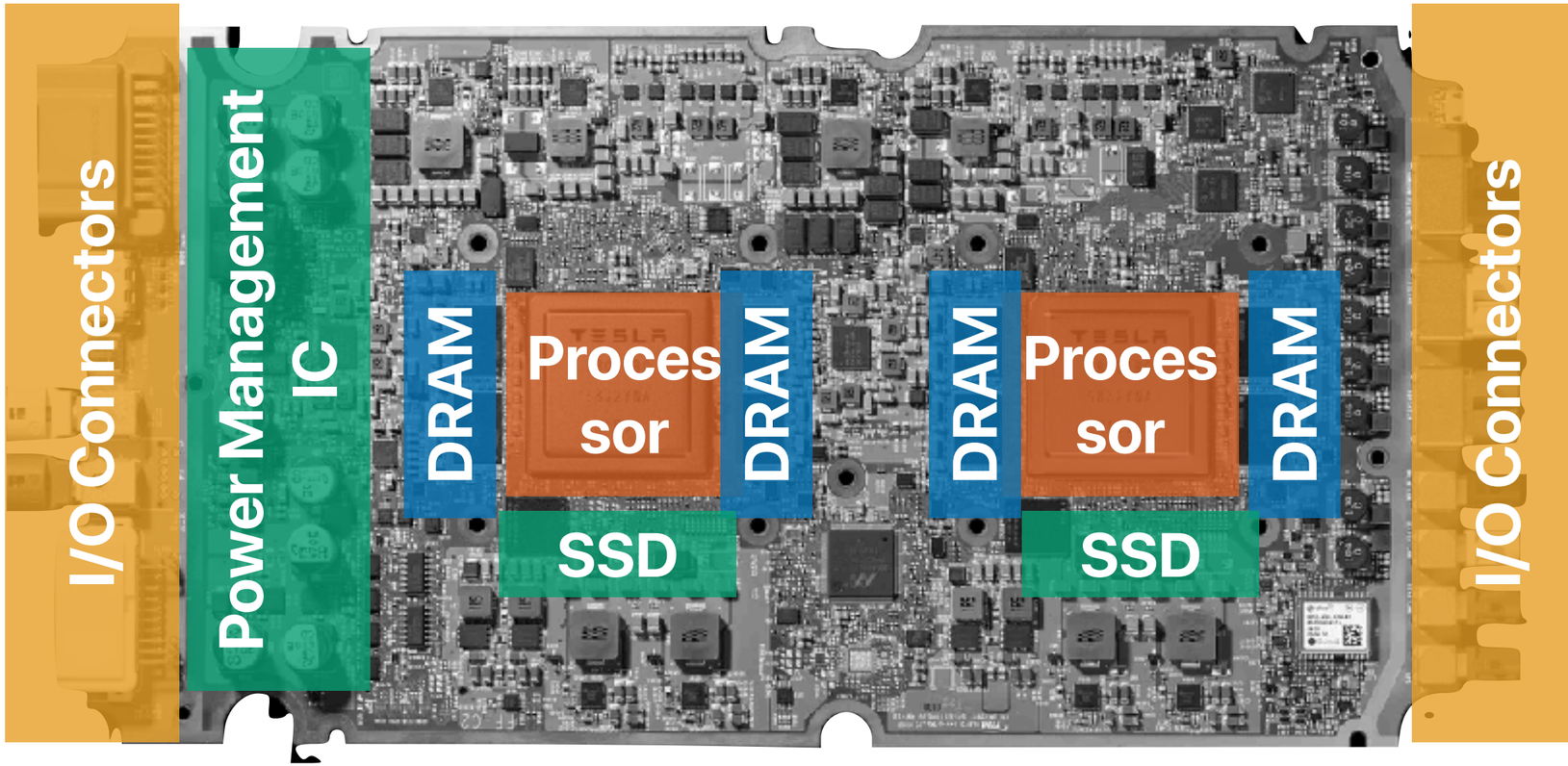


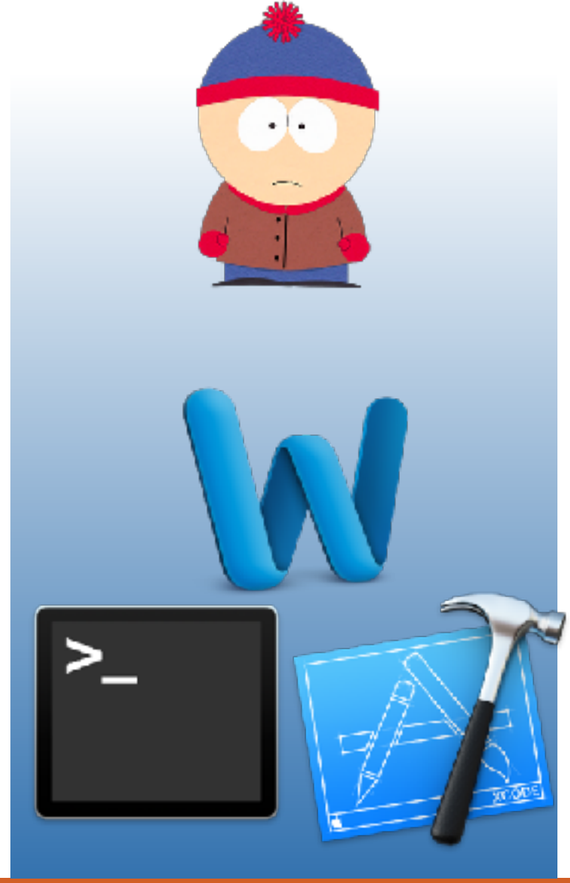
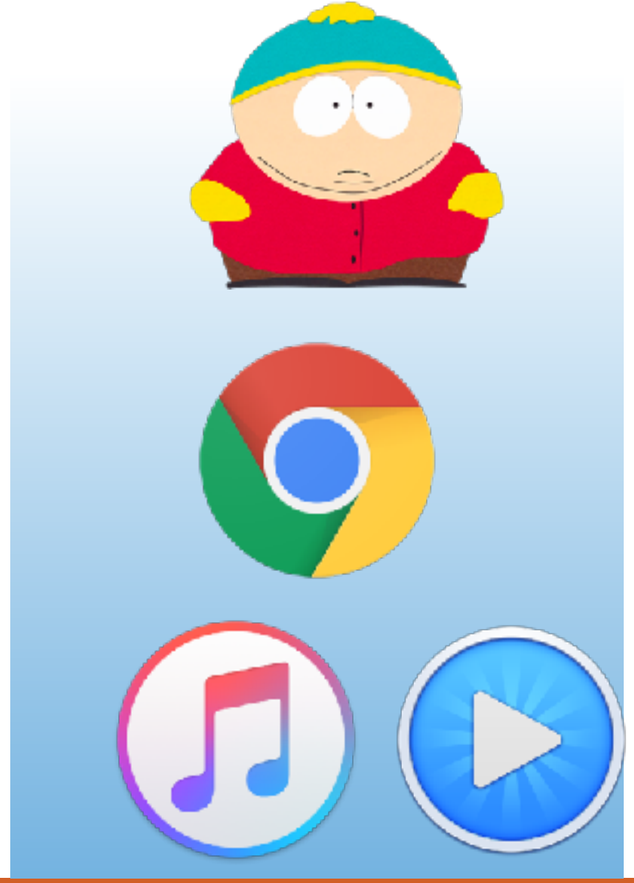
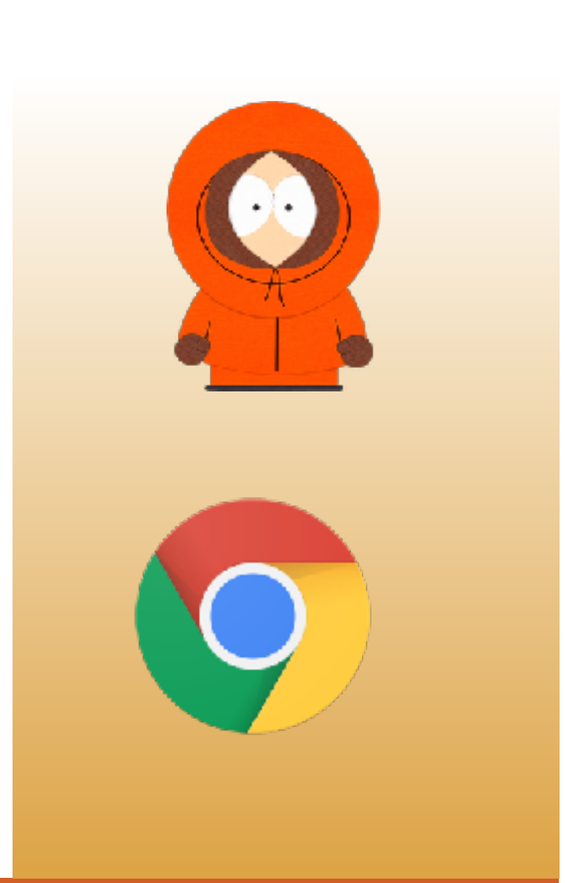


Nintendo Switch



Tesla Model 3



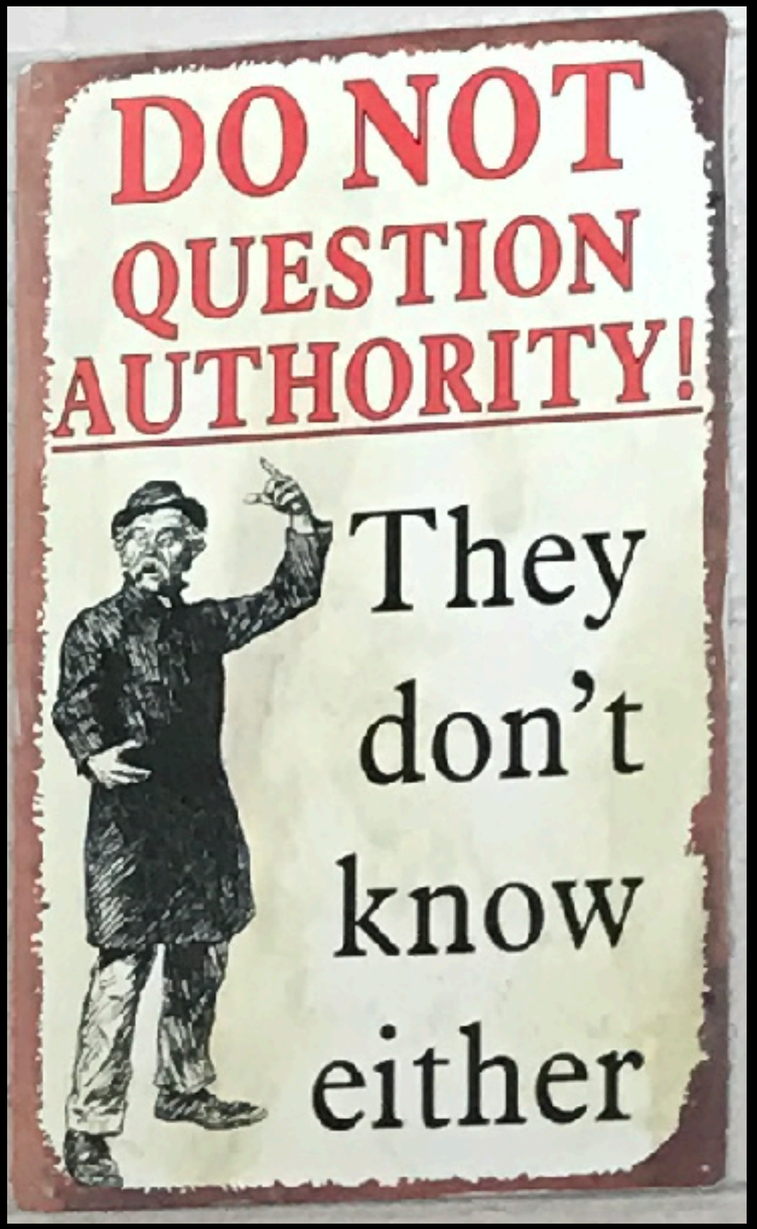
Operating System

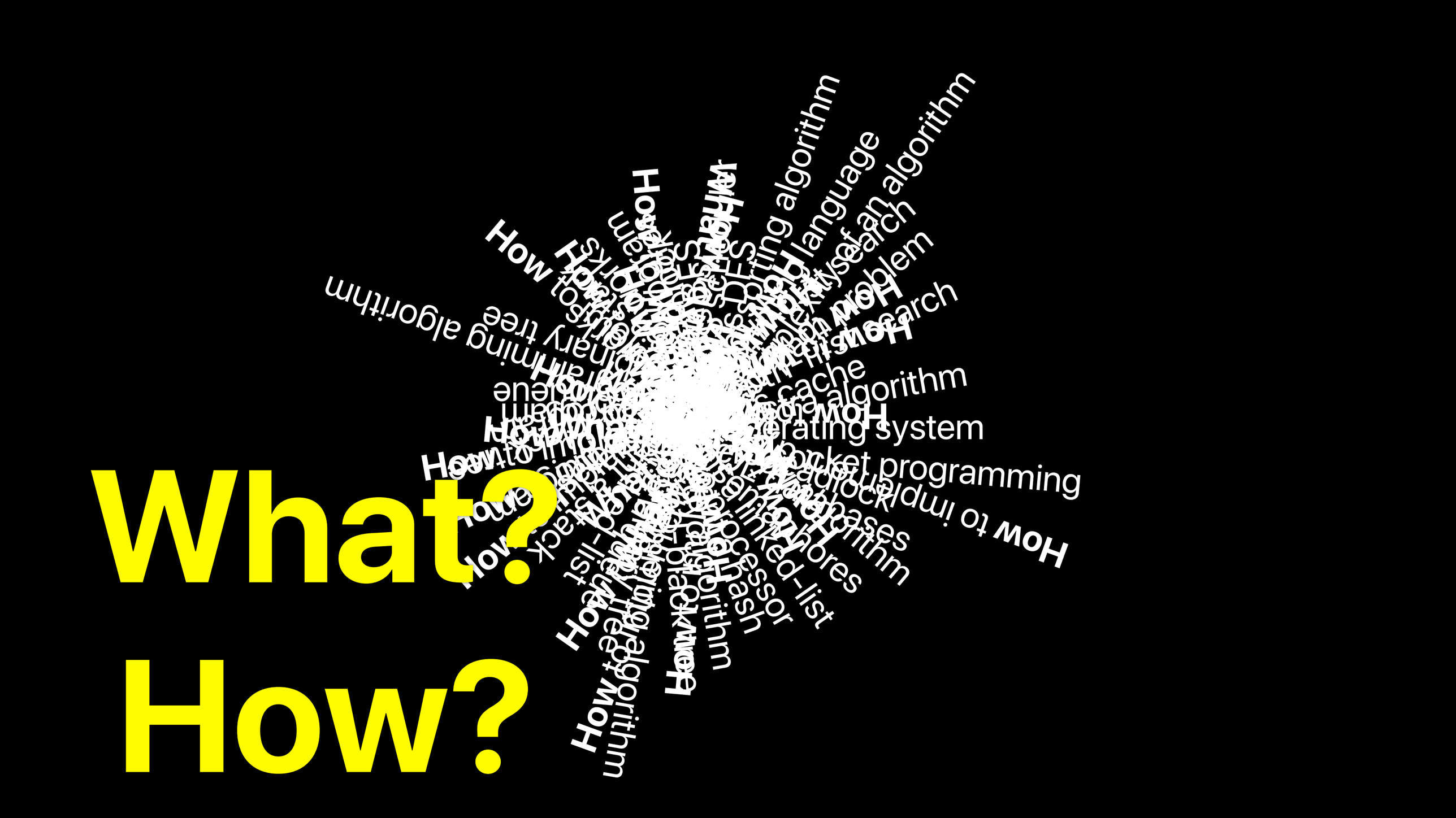



CS202: Advanced Operating Systems

Advanced

Why?
What?
How?




What?
How?

CS202 Lecture

Why?

What?

How?

CS202 Project

Logistics

Course resource

- Lectures: TuTh 2p-3:20p on Zoom (for at least the first two weeks)
- Office Hours: (Find the link through Google Calendar)
  Hung-Wei Tseng: MTu 11a-12p on Zoom
  Yu-Chia Liu: W 2p-3p, F 11a-12p on Zoom
- Schedule, slides on **course webpage**:
  https://www.escalab.org/classes/cs202-2022wi/
- Discussion on **piazza**:
  https://piazza.com/class/kxgldzml6k71g2
- Reading quizzes, homework submissions on **eLearn**:
  https://elearn.ucr.edu/courses/31822
- Youtube Channel
  https://www.youtube.com/profusagi
- Calendar
  https://calendar.google.com/calendar/u/0/r?cid=ucr.edu_b8u6dvkretn6kq6igunlc6bldg@group.calendar.google.com

Winter 2022
Home
Modules
Assignments
Piazza
Syllabus
Grades
Zoom

CS_202_001 - ADVANCED OPERATING SYSTEMS

CS202 Advanced Operating Systems (2022, Winter)

Instructor

Hung-Wei Tseng
email: htseng @ ucr.edu
Office Hours: MTu 11a-12p

Teaching Assistant

Yu-Chia Liu
e-mail: yliu719 @ ucr.edu
Office Hours: W 2p-3p F 11a-12p

Other important links

Quizzes, Assignments, Grading: eLearn
Discussion Forum on Piazza: https://piazza.com/class/kxg|dzml6k71g2
Youtube Channel: https://www.youtube.com/profusagi

Calendar

Instructor — Hung-Wei Tseng

- Website: https://intra.engr.ucr.edu/~htseng/
- Office hour: MTu 11:00a-12:00p on Zoom
- E-mail: htseng@ucr.edu
- BS/MS in **Computer Science**, National Taiwan University
- PhD in **Computer Science**, University of California, San Diego
- Research Interests
  - Accelerating applications using AI/ML accelerators
  - Intelligent storage devices
  - Non-volatile memory based systems
  - Anything could accelerate applications

Teaching Assistant — Yu-Chia Liu

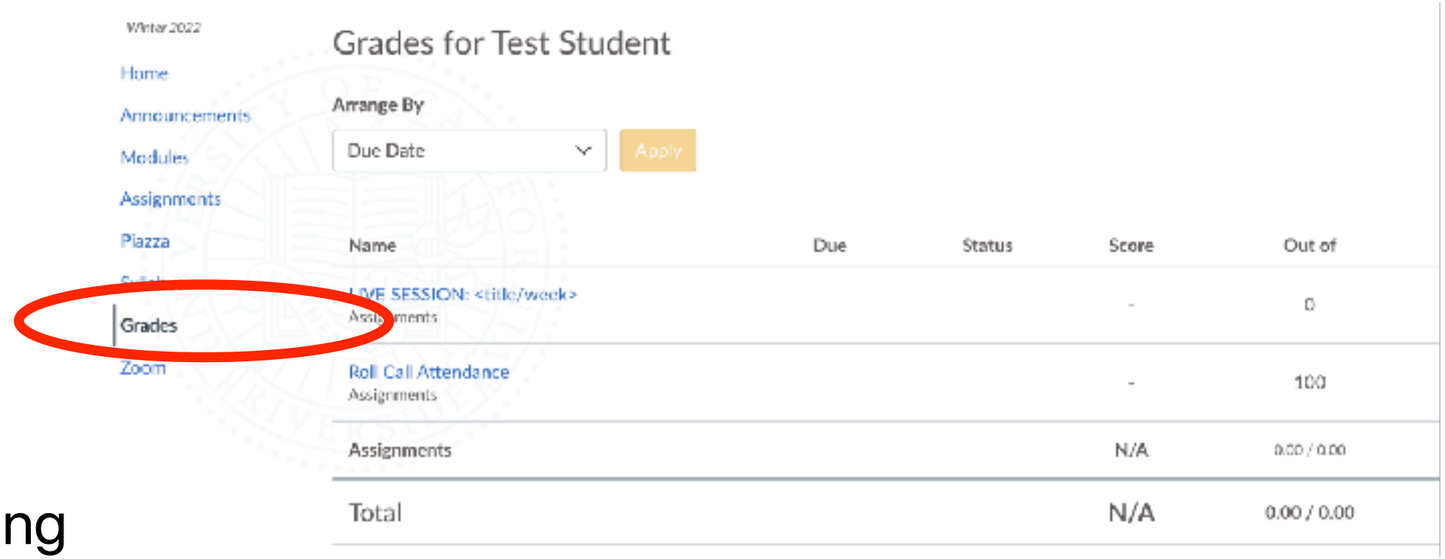
- Office hours: W 2p-3p F 11a-12p on Zoom
- E-mail: yliu719@ucr.edu

Your tasks

- Login/discussion in eLearn and piazza.
- Read the text before class!
  - Operating Systems: Three Easy Pieces Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau (free online http://pages.cs.wisc.edu/~remzi/OSTEP/)
  - I'm not going to cover everything in class, but you are responsible for all the assigned text.
  - Papers
- Reading quizzes in eLearn (15%)
  - Come to class — answering at least 50% of Zoom Polls or Everywhere Polling during 4 grading periods, counted as 4 reading quizzes
  - We will drop at least 5 of your lowest reading quizzes, so it's OK if you don't attend
- Project (25%) — intensive C programming in the system/kernel level
- Midterm (20%) — take home/online, format TBA
- Final (40%) — take home/online, format TBA

Grading

- You can see your grades on eLearn.



- Errors in grading
  - If you feel there has been an error in how an assignment or test was graded, you have one week from when the assignment is return to bring it to our attention. You must submit (via email to the instructor and the appropriate TAs) a written description of the problem. Neither I nor the TAs will discuss regrades without receiving an email from you about it first.
- For arithmetic errors (adding up points etc.)
  - you do not need to submit anything in writing, but the one week limit still applies.

Academic Honesty

- Don’t cheat.
  - Cheating on a test will get you an F in the class and no option to drop, and a visit with your college dean.
  - Cheating on homework means you don’t have to turn them in any more, but you don’t get points either. You will also take at least 25% penalty on the exam grades.
- Copying solutions of the internet or a solutions manual is cheating
  - They are incorrect sometimes
- Review the UCR student handbook
- When in doubt, ask.

Learning eXperience

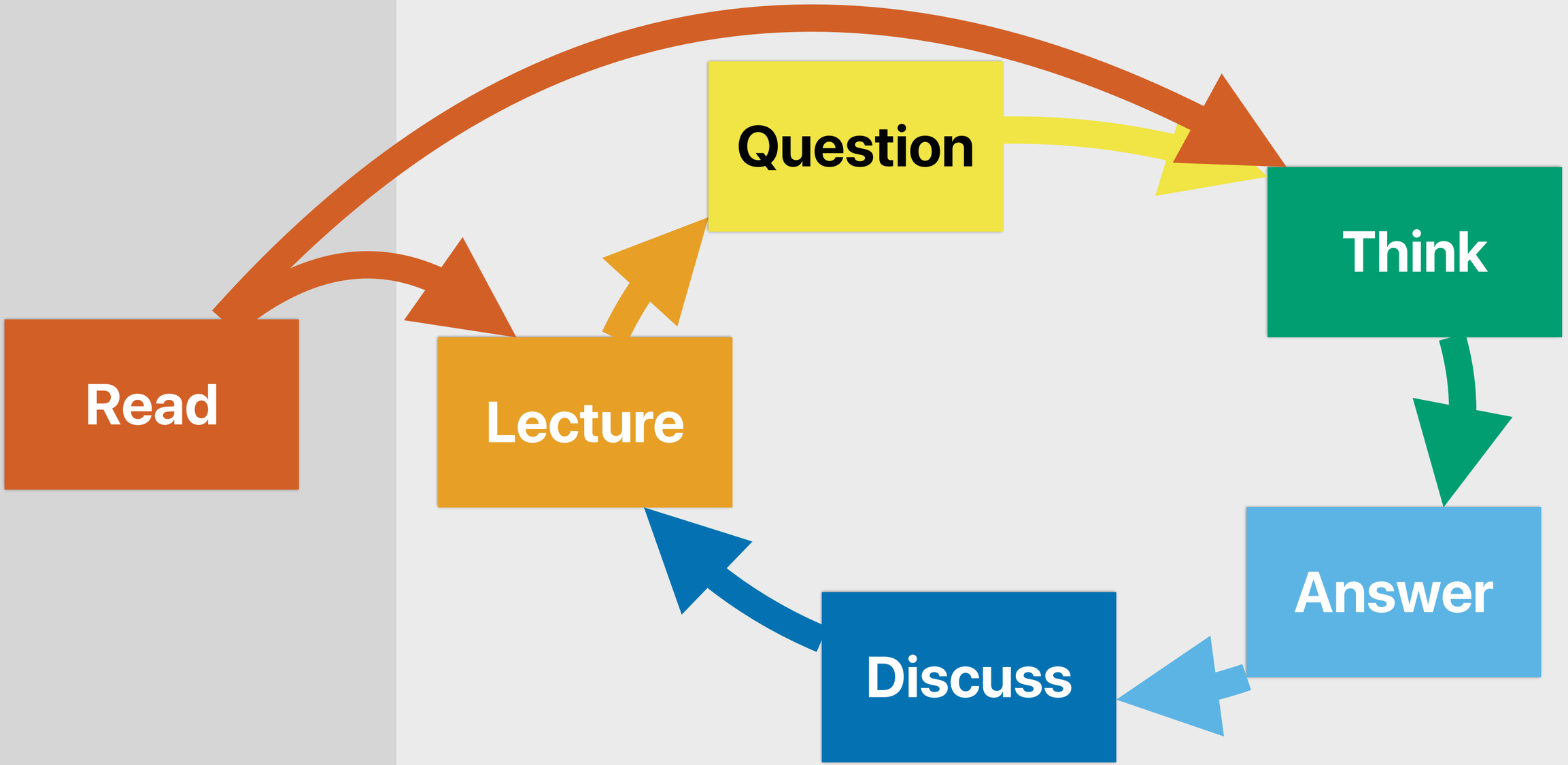
Most lectures today …

I expect the lecture to be...

Peer instruction

- An **Active Learning** teaching method proposed by Prof. Eric Mazur from Harvard University in the early 1990s
- Before the lecture — You will first try your best to go through and understand the required **reading**
- During the lecture — I'll bring in activities to ENGAGE you in exploring your understanding of the material
  - Popup questions
  - Individual **thinking** — use polls in Zoom to express your opinion
  - Group **discussion**
    - Discuss in breakout rooms
    - Use polls in Zoom or Poll Everywhere to express your group's opinion
  - Whole-classroom **discussion** — we would like to hear from you
  - I will explain and lecture on those related concepts

Peer Instruction



Before the lecture

During the lecture

Before lectures: reading quizzes

- This is a peer instruction class
  - The lecture will require you to read and try your best to understand the material first
  - We need to make sure that you read the material first to achieve the best learning outcome
- Reading assignments from
  - Textbook: Operating Systems: Three Easy Pieces Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau
    (free online http://pages.cs.wisc.edu/~remzi/OSTEP/)
  - Papers — at least get through those "focuses" listed in the schedule
- Reading quizzes:
  - On eLearn
  - Due before the lecture, usually once a week. Check the schedule on our webpage
  - You will have two chances. We take the average
  - No time limitation until the deadline
  - No make up reading quizzes — we will drop probably one or two lowest at least

Why attend live sessions and discuss?

- I'll bring in activities to ENGAGE you in exploring your understanding of the material
  - Let you practice
  - Bring out misconceptions
  - Let us LEARN from each other about difficult parts
  - It's going to be fun!
- You will be GET CREDIT for your efforts to learn in class
  - By answering questions with polls within Zoom/Polleverywhere
  - Answer **50%** of the clicker questions in class, get full credits for 4 reading quizzes
- Group Discussion
  - We will divide the class into four groups for the first two weeks (at least)

Project

- We will work on “real Linux systems” and implement a linux kernel module
- Details will come soon
- **Real human beings work on real systems!**

Schedule

	Topic	Reading	Slides (Preview)	Slides (Release)	Due
1/4/2022	Intro				
1/6/2022	The Structure of Operating Systems and the Abstraction of Processes	- Arpaci-Dusseau Chapter 2, 4, 6			
1/8/2022	The Structure of Operating Systems	- E. W. Dijkstra. The Structure of the 'THE'-Multiprogramming System. Communications of the ACM, Vol. 11, No. 5, May 1968, pp. 341-346. - P. B. Hansen. The Nucleus of a Multiprogramming System, Communications of the ACM, Vol. 13, No. 4, April 1970, pp. 238-241, 250. Focusing on:			
1/13/2022	Processes & Threads	- D. M. Ritchie and K. Thompson. The UNIX Time-Sharing System, Communications of the ACM, Vol. 17, No. 7, July 1974, pp. 365-375. - Accetta, Mike, Robert Baron, William Bolosky, David Golub, Richard Rashid, Avadis Tevanian, and Michael Young. Mach: A New Kernel Foundation For UNIX Development. Proc. USENIX Summer Conference, Atlanta, GA, 1986, pp. 93-112.			
1/15/2022	Processes & Threads	Arpaci-Dusseau Chapter 5, 26--31			
1/20/2022	Processes/Threads Scheduling	- Arpaci-Dusseau Chapter 7 - Paul E. McKenney, Dipankar Sarma, Andrea Arcangeli, Andi Kleen, Orran Krieger, and Rusty Russell. Read Copy Update. In *Proceedings of the Ottawa Linux Symposium*, June 2002, pp. 338–367.			
1/22/2022	Processes/Threads Scheduling	- Corbató, Fernando J., Marjorie Merwin-Daggett, and Robert C. Daley. An experimental time-sharing system. In *Proceedings of the May 1-3, 1962, spring joint computer conference* (pp. 335-344). - Carl A. Waldspurger and William E. Weihl. Lottery Scheduling: Flexible Proportional-Share Resource Management. The First USENIX Symposium on Operating System Design and Implementation (OSDI), November, 1994. - Thomas E. Anderson, Brian N. Bershad, Edward D. Lazowska, Henry M. Levy. Scheduler Activations: Effective Kernel Support for the User-level Management of Parallelism. Proceedings of the 13th ACM Symposium on Operating Systems Principles (SOSP), Sept. 1991, pp. 95-109.			
1/27/2022	Virtual memory	- Arpaci-Dusseau Chapter 13, 15, 16, 18			
2/1/2022	Virtual memory	- Arpaci-Dusseau Chapter 20, 21, 22			
2/3/2022	Virtual memory	- H. M. Levy and P. Lipman. Virtual Memory Management in VAX/VMS. IEEE Computer, Vol. 15, No. 3, March 1982, pp.35-41. - Richard Rashid, Avadis Tevanian, Michael Young, David Golub, Robert Baronn, David Black, William Bolosky, and Jonathan Chew. Machine-Independent Virtual Memory Management for Paged Uniprocessor and Multiprocessor Architectures. The Second International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), October 1987, pp. 31-39.			
2/8/2022	Virtual memory	- O. Babaoglu and W. Joy. Converting a Swap-Based System to do Paging in an Architecture Lacking Page-Reference Bits. Eighth ACM Symposium on Operating System Principles (SOSP), December 1981, 78-86. - R. Carr and J. Hennessy. WSCLOCK-A Simple and Effective Algorithm for Virtual Memory Management. Eighth ACM Symposium on Operating System Principles (SOSP), December 1981, 87-95.			
2/10/2022	Midterm				
2/15/2022	File systems	- Arpaci-Dusseau Chapter 39, 40, 41			
2/17/2022	File systems	- Marshall K. McKusick, William N. Joy, Samuel J. Leffler, and Robert S. Fabry. A Fast File System for Unix. ACM Transactions on Computer Systems, 2(3), August 1984, pp. 181-197. - Mendel Rosenblum and John K. Ousterhout. The Design and Implementation of a Log-Structured File System. The 13th ACM Symposium on Operating Systems Principles (SOSP), December 1991.			
2/22/2022	Fast, non-volatile memory-based storage devices	- Arpaci-Dusseau Appendix–Flash-based SSDs - Michael Wu and Willy Zwaenepoel. eNVy: a non-volatile, main memory storage system. The sixth international conference on Architectural support for programming languages and operating systems (ASPLOS). - Jingpei Yang, Ned Plasson, Greg Gillis, Nisha Talagala, and Swaminathan Sundararaman. Don't stack your log on my log. 2nd Workshop on Interactions of NVM/Flash with Operating Systems and Workloads (INFLOW 14).			
2/24/2022	Networked & cloud storage	- Arpaci-Dusseau Chapter 49 - Sanjay Ghemawat, Howard Gobioff, Shun-Tak Leung. The Google File System. Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles (SOSP), October 2003, pp. 29–43.			
3/1/2022	Distributed systems	- Brad Calder et al. Windows Azure Storage: A Highly Available Cloud Storage Service with Strong Consistency. Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles (SOSP), October 2011, pp. 143–157. - Subramanian Muralidhar et al. f4: Facebook's Warm BLOB Storage System. 11th USENIX Symposium on Operating Systems Design and Implementation (OSDI).			Project
3/3/2022	Distributed systems	- Luiz André Barroso, Jeffrey Dean, and Urs Hölzle. Web Search for a Planet: The Google Cluster Architecture. IEEE Micro, March 2003, 23(2): 22–28.			
3/8/2022	Virtual machine	- Arpaci-Dusseau Appendix--Virtual machines - Keith Adams and Ole Agesen. A comparison of software and hardware techniques for x86 virtualization. The 12th international conference on Architectural support for programming languages and operating systems (ASPLOS).			
3/10/2022	Virtual machine	- P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield. Xen and the art of virtualization. The 19th Symposium on Operating Systems Principles (SOSP), October 2003. - B. W. Lampson. Hints for computer system design. The Ninth ACM Symposium on Operating System Principles (SOSP), October 1983, pp. 33-48.			
	Final Exam				

Subject to change

You need to complete the reading of the textbook and papers before lectures

Download slides after lectures

Check due dates here

Lots of paper reading — up to 4 per week, a total of 24 this quarter!


Background music: We're Not Gonna Take It/ Songwriter(s): Dee Snider/Performed by Twisted Sister

Why papers?

No alternative facts

- Papers are written by authors who create/invent these artifacts
  - First-hand information
  - Not being cooked by media/press...
- Papers are reviewed based on originality
- Papers are reviewed by experts without conflict of interests

Papers give you insights!

- Papers contain **design principles** that are missing in your textbook or online documents
- You can apply these design principles and the skills of analyzing these principles to anywhere (e.g. you will surprisingly find how the paper you read next week affects software engineering)
- You can learn those **whys** for those proposed work

Industry cares

[redacted]@intel.com> 2011/2/15

寄給 h1tseng

Hi Hung-Wei,

I am very interested in your topic you presented yesterday. If possible, may I get a copy of

Best Regards,

[redacted]

[redacted]freescale.com 透過 cs.ucsd.edu 2012/1/10

寄給 h1tseng

Hung-Wei

I just finished reading your paper "Understanding the Impact of Power Loss on Flash Memory", very interesting information, do you have a PowerPoint presentation that goes along this paper?

[redacted]fb.com> 2019年7月9日 週二 下午2:12

寄給 Hung-Wei [redacted]

Hung-wei

[redacted]

Given we are also working on in-memory and near-memory computing at my Boston team, I would like to see how do we work more closely to churn out even more useful results and applications for Facebook's ML models/workloads in both datacenters and edge devices and instigate new research directions.

[redacted]

[redacted]sap.com 透過 cs.ucsd.edu 2012/11/12

寄給 h1tseng

Hi Tseng,

I have read your paper titled "Understanding the Impact of Power Loss on Flash Memory". It work. I would like to understand what specific tools did you use to observe the page-read and the FTL level. Did you use some sort of Flash simulator to get all the statistics about the num and the energy consumption? My second question would be regarding FTL algorithms. Did y real SSD or you used some kind of simulator and simulated the FTL algorithm?

Thanks.

[redacted]

SAP Research

[redacted]@huawei.com> 2016/6/24

寄給 h1tseng

Hi, Hung-Wei,

[redacted] from Huawei, and I am impressed by your ISCA 2016 presentation in Seoul. Near-data processing in ssds may be a promising solution for future data centers. Would you mind sending me your slides presented in the conference? I really appreciate your kindness. Thank you!

Best regards,

[redacted]

v.mit.edu/courses/electrical-engineering-and-compute

11	File System Performance and	Tweedie,

cseweb.ucsd.edu/classes/fa17/cse221-a/readings.

Historical Perspective

Tue 10/3

- E. W. Dijkstra, The Structure of the 'THE'-Multiprogramming System, Communications of the ACM, Vol. 11, No. 5, May 1968, pp. 341-346.

  (Additional historical background on semaphores in Wikipedia.)

  Q: Dijkstra explicitly states their goals for the THE operating system. How do these goals compare to, say, Microsoft's goals for the Windows operating system? Why do we no longer build operating systems with the same goals as THE?

- P. B. Hansen, The Nucleus of a Multiprogramming System, Communications of the ACM, Vol. 13, No. 4, April 1970, pp. 238-241, 250.

  Optional related paper on a deployment experience of RC 4000:

  P. B. Hansen, The RC 4000 Real-Time Control System at Pulway, BIT 7, pp. 279-288, 1967.

  Q: How does synchronization in the RC 4000 system compare with synchronization in the THE system?

Thu 10/5

- D. G. Bobrow, J. D. Burchfiel, D. L. Murphy, and R. S. Tomlinson, TENEX, a Paged Time Sharing System for the PDP-10, Communications of the ACM, Vol. 15, No. 3, March 1972, pp. 135-143.

  Q: What features in TENEX are reminiscent of features in Unix (a later system)?

- W. Wulf, E. Cohen, W. Corwin, A. Jones, R. Levin, C. Pierson, and F. Pollack, HYDRA: The Kernel of a Multiprocessor Operating System, Communications of the ACM, Vol. 17, No. 6, June 1974, pp. 337-345.

  Q: How is a Hydra procedure different from the procedures we are familiar with in a typical language and runtime environment?

Structure

- B. Lampson, Protection, Operating Systems Review, Vol. 8, No. 1, January 1974, pp. 18-24.

  Q: What are the concepts in HYDRA that correspond to Lampson's definitions of "Domain", "Object", and "Access Matrix"? What about Multics?

UC San Diego

courses.cs.washington.edu/courses/csep551/14au/

University of Washington

cs.wisc.edu/~bart/736/f2019/reading_list.html

Wisconsin

VERSITY OF WISCO
Computer Sciences

CS 736: Readi

pyright limitations,
f you want to down

gement

d Jerome H. Saltzer, /
Rings, *Communicatio*

yer, Peter Drushel and
uperpages, *5th Sympo*
December 2002.

n and Comm

ating Sequential Proc
-677.

An Operating System
r 1974, pp. 549-557

ld D. Redell, Experien
ACM, **23** 2, February 1

tin T. Clements, Yand
and Nickolai Zeldovic
sium on Operating Sy
ada, October 2010.

t, Concurrent Reading
1977

cs523-uiuc.github.io/fall19/readings.html

UIUC

Historical Perspectives

Fri 8/30

- E. W. Dijkstra, The Structure of the "THE"-Multiprogramming System, In Proceedings of the 1st ACM Symposium on Operating System Principles (SOSP '67), October 1967.
- P. B. Hansen, The Nucleus of a Multiprogramming System, Communications of the ACM, Vol. 13, No. 4, April 1970, pp. 238-241.

Wed 9/4

- D. G. Bobrow, J. D. Burchfiel, D. L. Murphy, and R. S. Tomlinson, TENEX, a Paged Time Sharing System for the PDP-10, In Proceedings of the 3rd Annual Symposium on Operating Systems Principles (SOSP '71), October 1971.
  - Additional historical background about PDP-10
- W. Wulf, E. Cohen, W. Corwin, A. Jones, R. Levin, C. Pierson, and F. Pollack, HYDRA: The Kernel of a Multiprocessor Operating System , Communications of the ACM, Vol. 17, No. 6, June 1974, pp. 337-345.
  - H. M. Levy, Chapter 8: The Hydra System, Capability-Based Computer Systems, Digital Press, 1984.
  - W. Wulf and C. G. Bell, C.mmp: a multi-mini-processor, In Proceedings of AFIPS Fall Joint Computing Conference, December 1972.

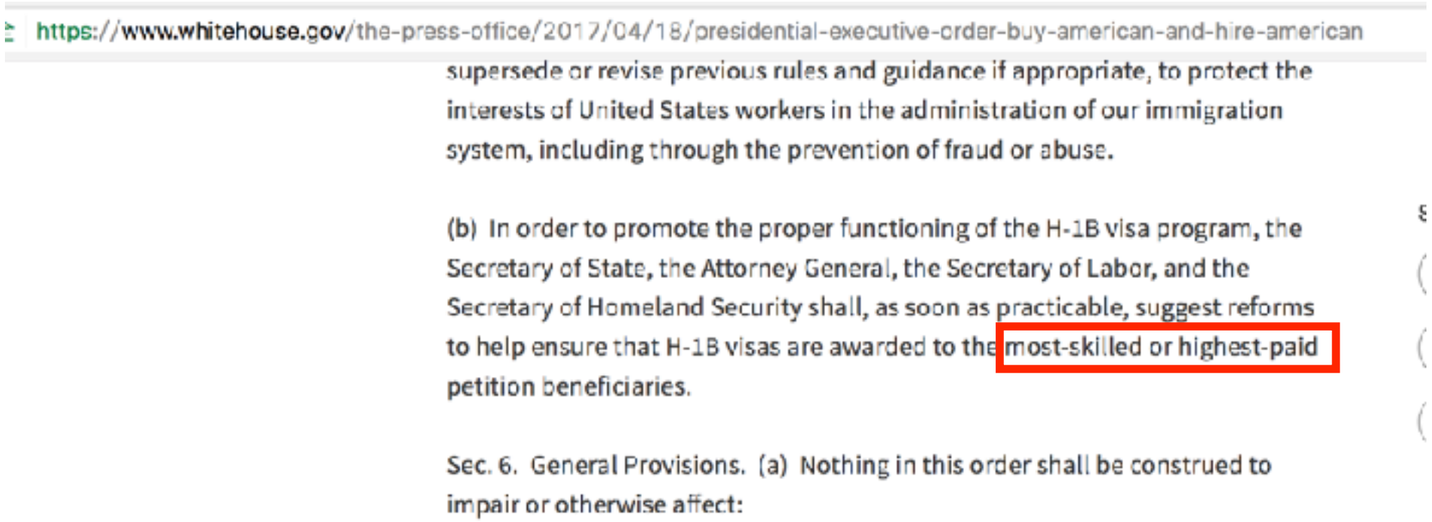
Unix and Plan 9 (and MINIX and Linux)

Fri 9/6

- D. M. Ritchie and K. Thompson, The UNIX Time-Sharing System, In Proceedings of the 4th Annual Symposium on Operating Systems Principles (SOSP '73), October 1973.
- R. Pike, D. Presotto, S. Dorward, B. Flandrena, K. Thompson, H. Trickey, and P. Winterbottom, Plan 9 From Bell Labs, USENIX Computing Systems, Vol. 8, No. 3, Summer 1995, pp. 221-254.
  - Linux's History written by Linus Torvalds

Make yourself more valuable

- Every top 20 CS MS/PhD program has their students reading papers in OS classes and every instructor at UCR teaches similar sets of materials
- You have to compete with them when you're on the market
- You need some context to prove that you're also geeky enough to be one of their colleagues


https://www.whitehouse.gov/the-press-office/2017/04/18/presidential-executive-order-buy-american-and-hire-american

supersede or revise previous rules and guidance if appropriate, to protect the interests of United States workers in the administration of our immigration system, including through the prevention of fraud or abuse.

(b) In order to promote the proper functioning of the H-1B visa program, the Secretary of State, the Attorney General, the Secretary of Labor, and the Secretary of Homeland Security shall, as soon as practicable, suggest reforms to help ensure that H-1B visas are awarded to the most-skilled or highest-paid petition beneficiaries.

Sec. 6. General Provisions. (a) Nothing in this order shall be construed to impair or otherwise affect:

Academic honesty

- Don’t cheat.
  - Cheating on a test will get you an F in the class and no option to drop, and a visit with your college dean.
  - Cheating on project means you don’t have to turn them in any more, but you don’t get points either. You will also take at least 25% penalty on the exam grades.
- Copying solutions/code of the internet or a solutions manual is cheating — we do random sampling, we do check/compare all coding projects
- When in doubt, ask.
- Final grading is based your **relative ranking** in class — **if you help people cheat, you hurt yourself**


Background music: We're Not Gonna Take It/ Songwriter(s): Dee Snider/Performed by Twisted Sister

Term of Service

- CS202 is an operating system related class for graduate students. It's not our responsibility to recap everything that should be covered by an undergraduate operating system class from a regular computer science undergraduate program.
- This class requires intensive readings in research papers and the assigned textbook.
- This class requires you to speak and discuss your opinion with your classmates as well as the instructor.
- This class requires programming projects that uses the C programming language. It is your responsibility to learn how to program in C. It is also your responsibility to design the architecture, implementation details and tests for your coding projects.
- The instructor and course staffs reserve the right to refuse to answer inappropriate questions (e.g. directly telling if an answer is right or not).
- It is your responsibility to track the latest schedule, information, grades and materials from our course website, e-mails from the course staffs and the piazza forum.
- Any cheating will be treated seriously. You will get an F and we will report to the Dean's office

By clicking this box, you are agreeing to the Terms and Conditions of CS202, Winter 2021.

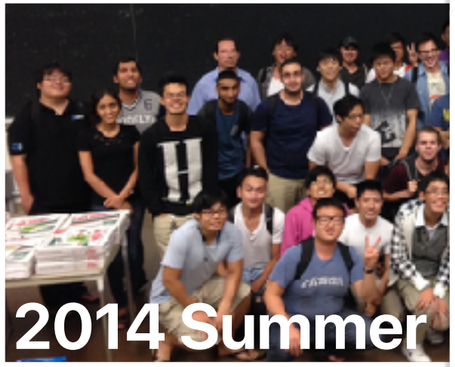
UC San Diego

NC STATE UNIVERSITY

UC RIVERSIDE


2012 Summer


2019 Summer I


2014 Summer


2019 Summer II


2016 Spring


2016 Summer


2016 Fall


2017 Spri


2017 Fall


2018 Spring


2018 Fall


2019 Spring


2021 Fall

2020 Winter

You
2022 Winte

UC zoom


2021 Winter
2020 Summer

2020 Spring


WHITE HO
WASHINGTON
Q & A
NBC NEWS
NOW

Announcement

- The first reading quiz due this Thursday before class!
  - Please find the reading quiz in eLearn!
  - Please visit the course webpage for the most accurate reading list

Computer
Science &
Engineering

202

つづく