# First Lecture of CS202: Advanced Operating Systems

Hung-Wei Tseng

# CS202: Let’s say something!

What’s your name?

Why’re you taking CS202

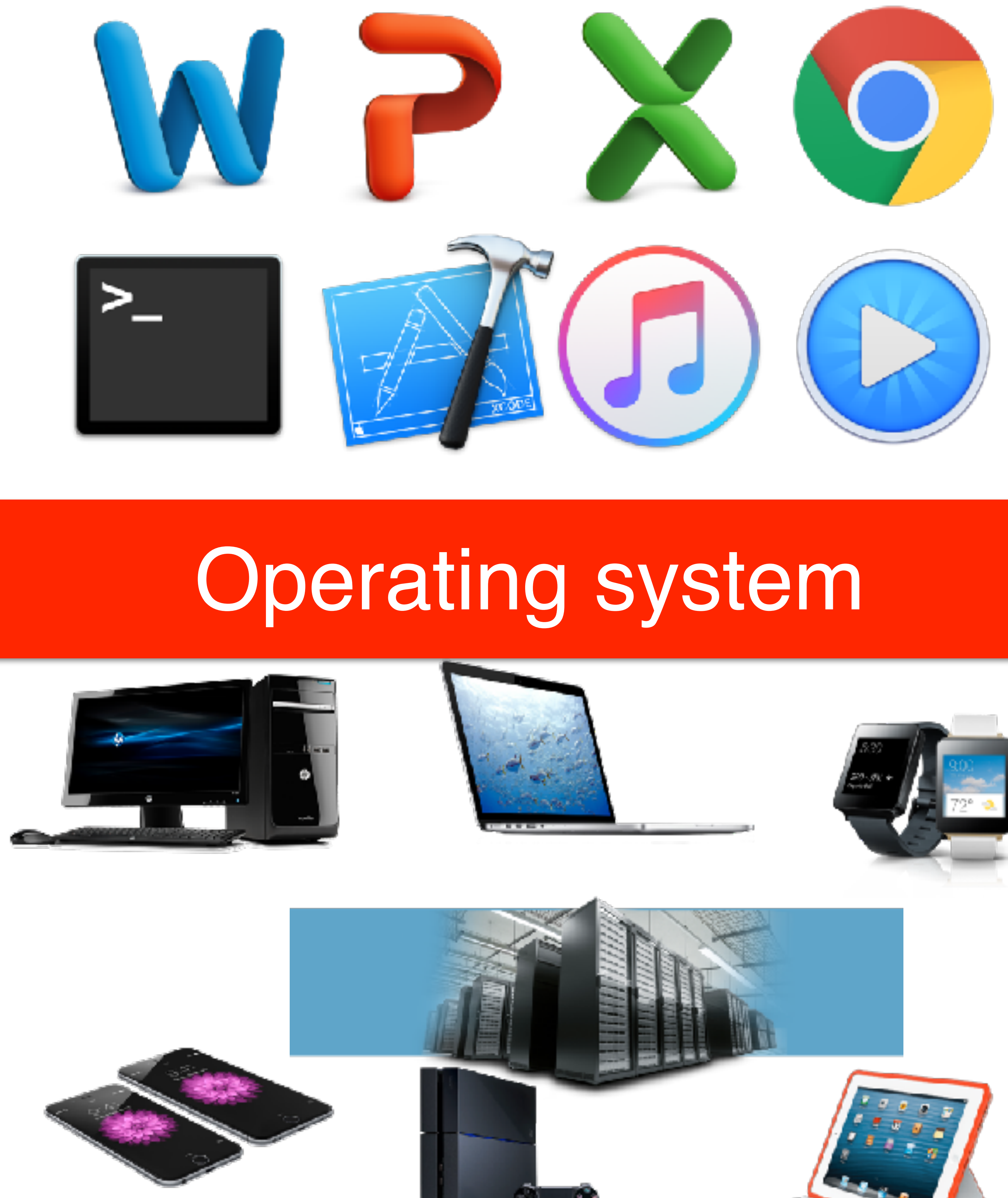
What’s the most exciting thing you did over the holiday season?

湘南
SHONAN
青のせんべい
マンゴー
紫いも
チョコレート
練乳いちご
練乳いちご
奥のクランターへ
抹茶
紫いも
抹茶
白桃
白桃
マンゴー
バニラ＆チョコレート
巨峰 メロン
丸焼きたこせんべい
1枚入　400円税込
丸焼きたこせんべい
1枚入　800円税込
100円
丸焼きたこ
1箱　450円

https://az-eandt-live-legacy.azureedge.net/news/2013/apr/images/640_edsac-web.jpg

# What releases human beings from the queues?

# Operating systems



Operating system

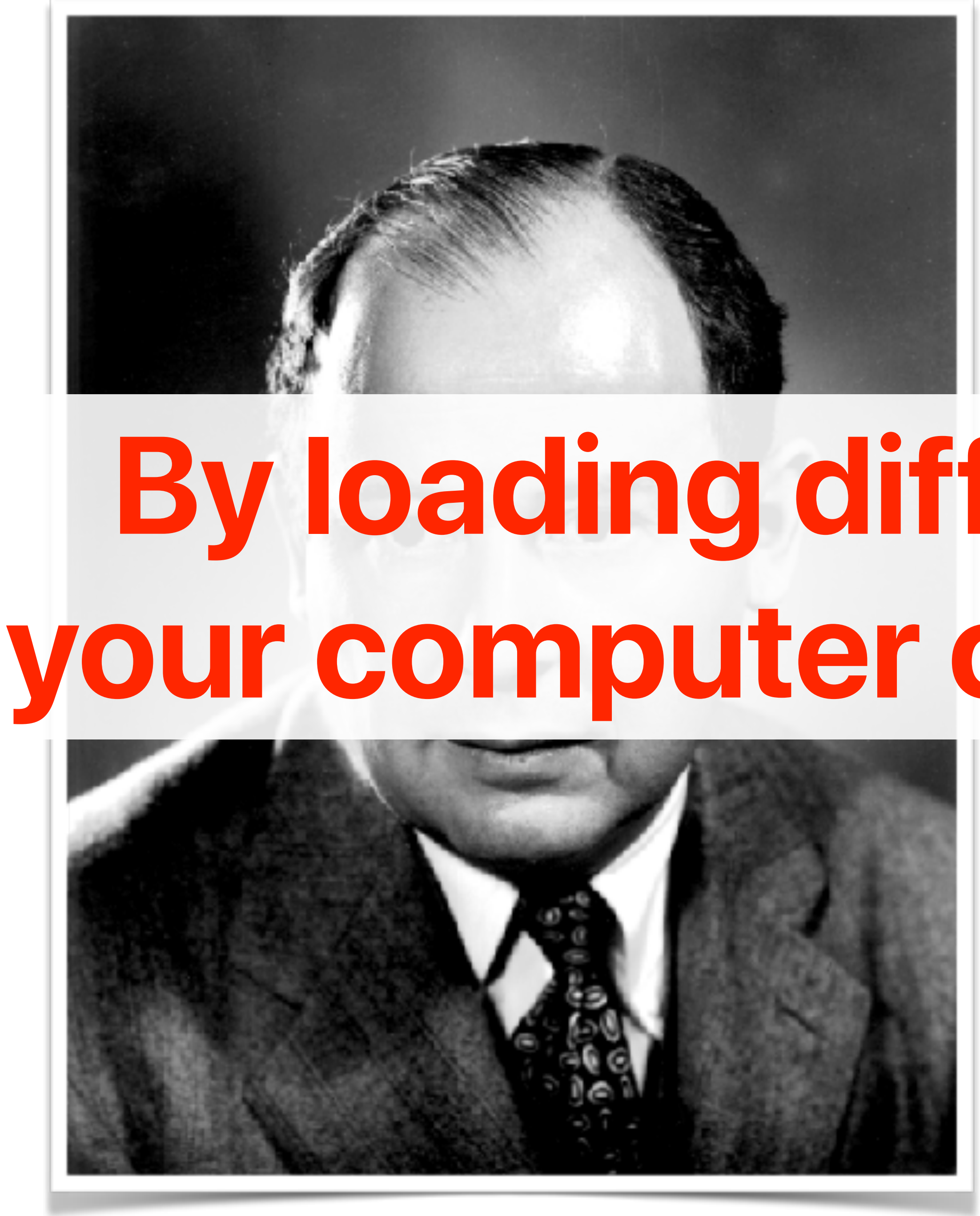
# What do we want from operating systems?

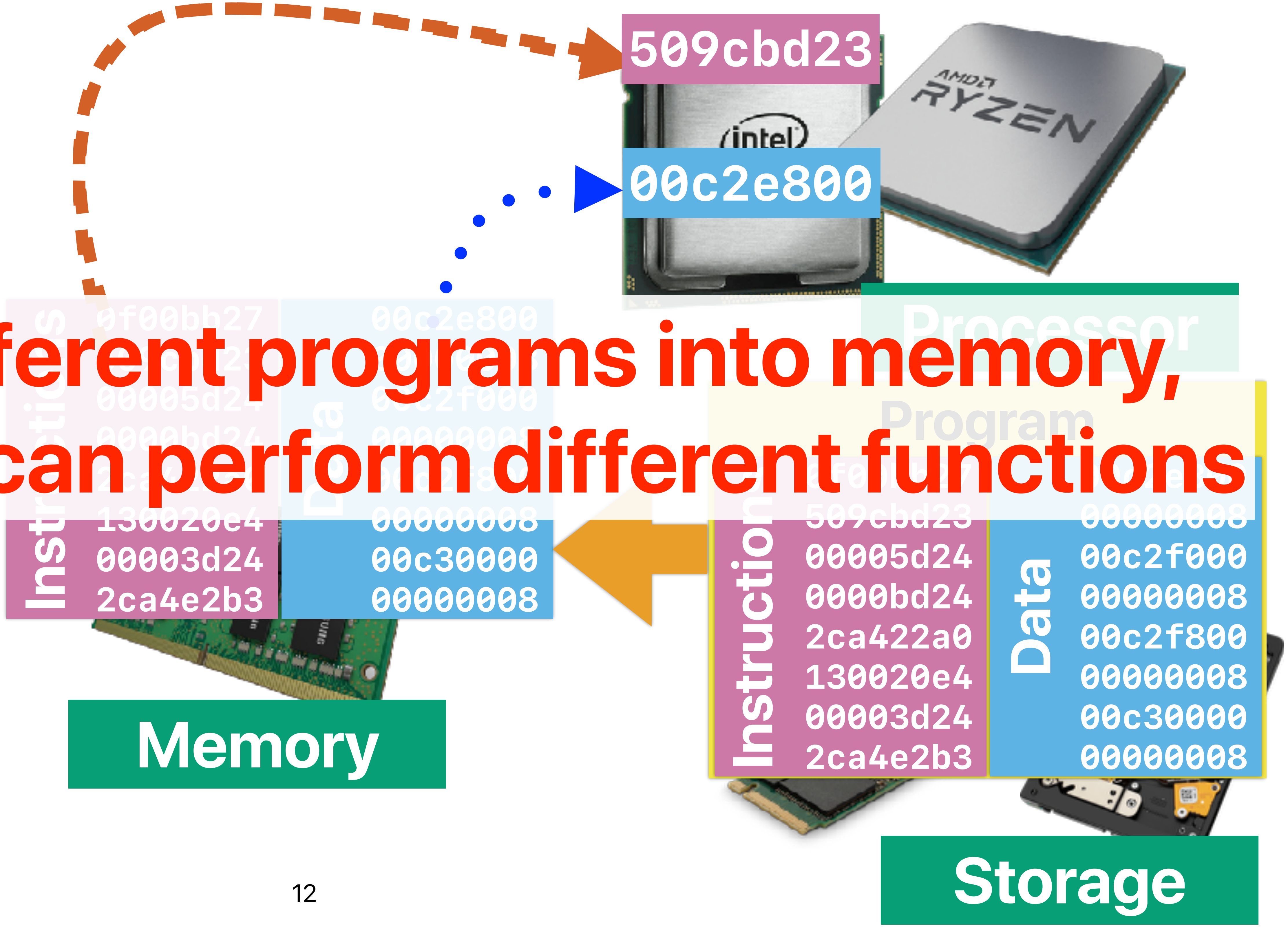
# What do we want from operating systems?

- 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

# The basic idea of execution

# The beast: von Neumman Architecture

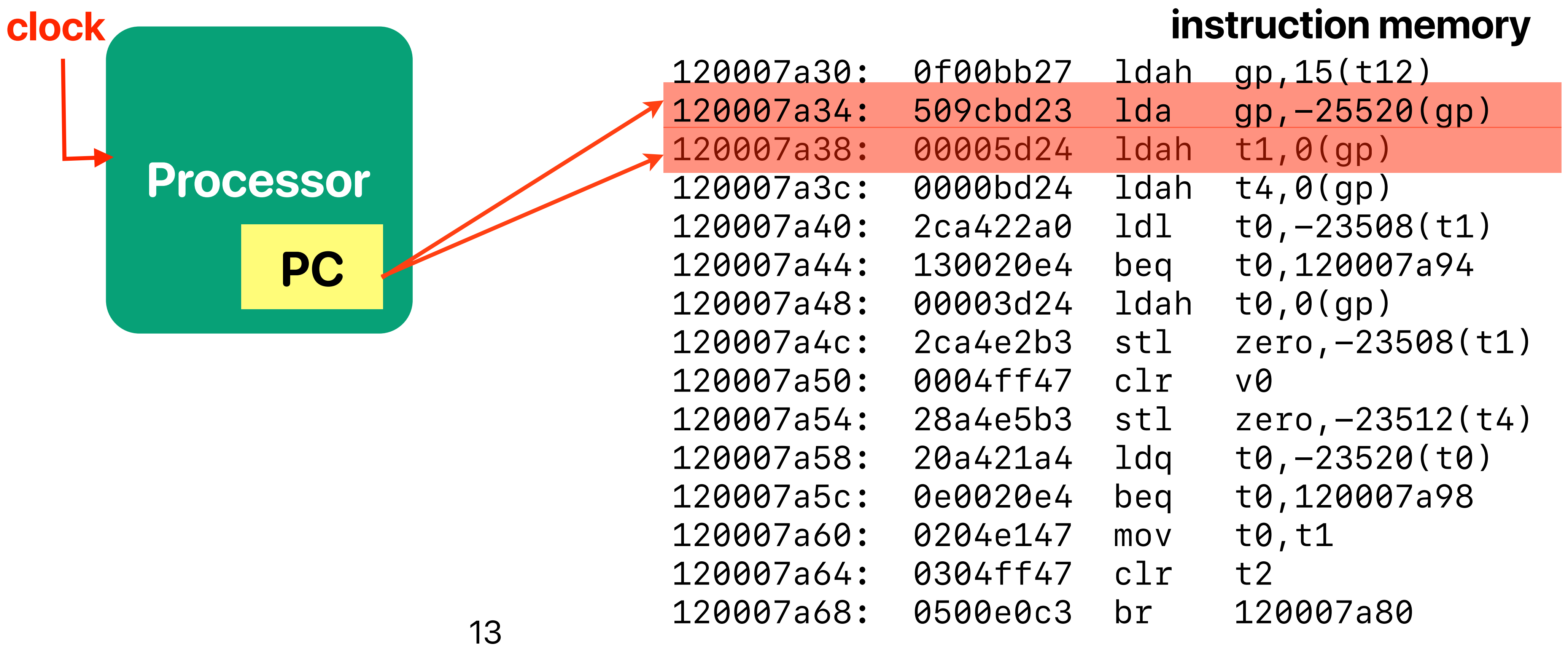




**By loading different programs into memory, your computer can perform different functions**

# How processor executes a program

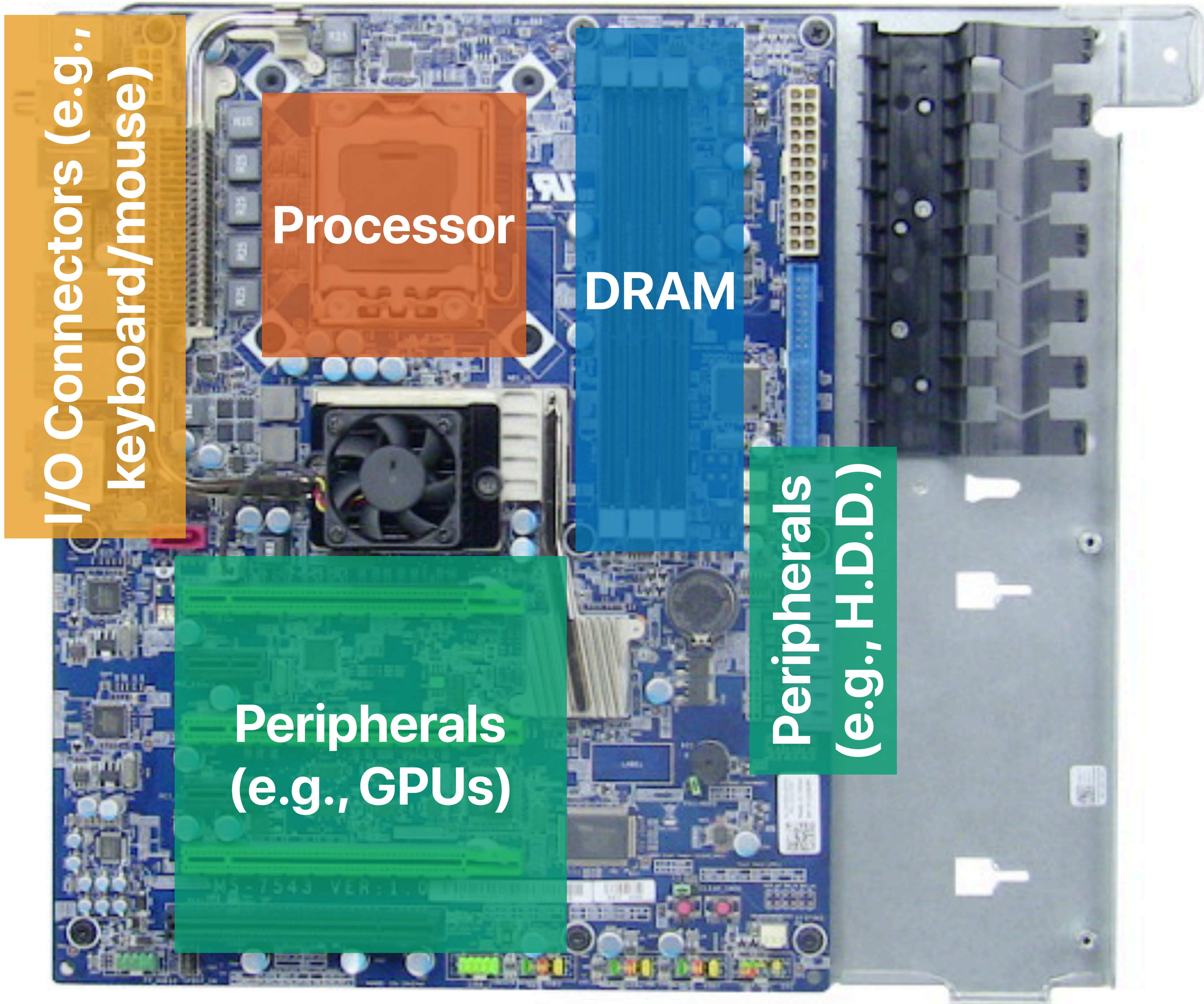
- The program counter (PC) tells where the upcoming instruction is in the memory
- Processor fetches the instruction, decode the instruction, execute the instruction, present the instruction results according to clock signals
- The processor fetches the next instruction whenever it's safe to do so



instruction memory

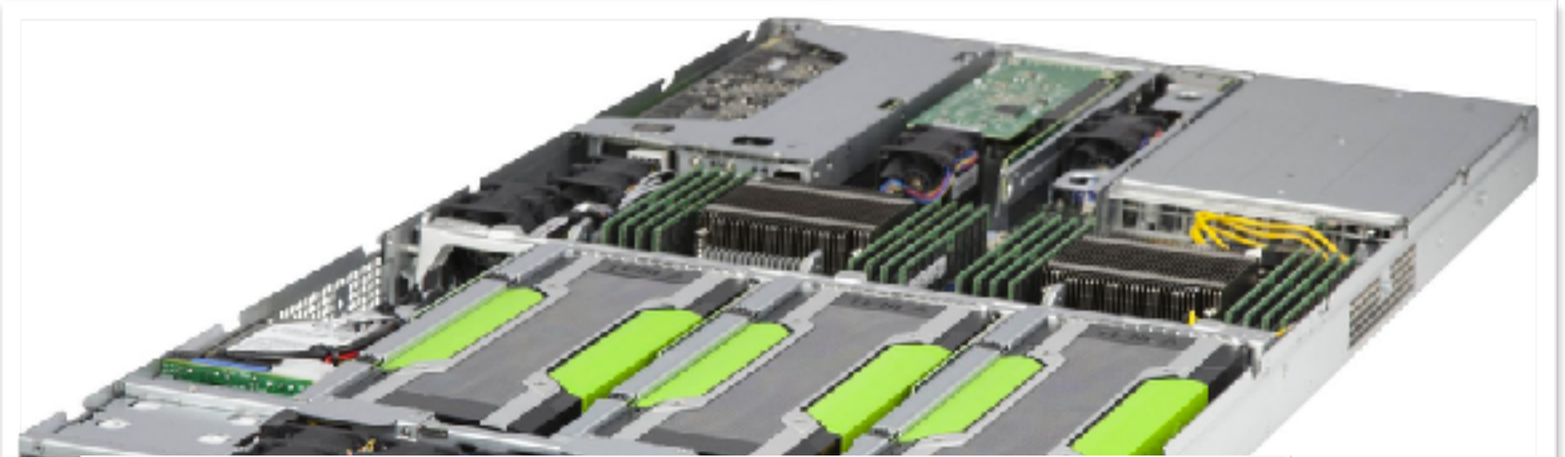
```
120007a30:  0f00bb27  ldah  gp,15(t12)
120007a34:  509cbd23  lda   gp,-25520(gp)
120007a38:  00005d24  ldah  t1,0(gp)
120007a3c:  0000bd24  ldah  t4,0(gp)
120007a40:  2ca422a0  ldl   t0,-23508(t1)
120007a44:  130020e4  beq   t0,120007a94
120007a48:  00003d24  ldah  t0,0(gp)
120007a4c:  2ca4e2b3  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
```

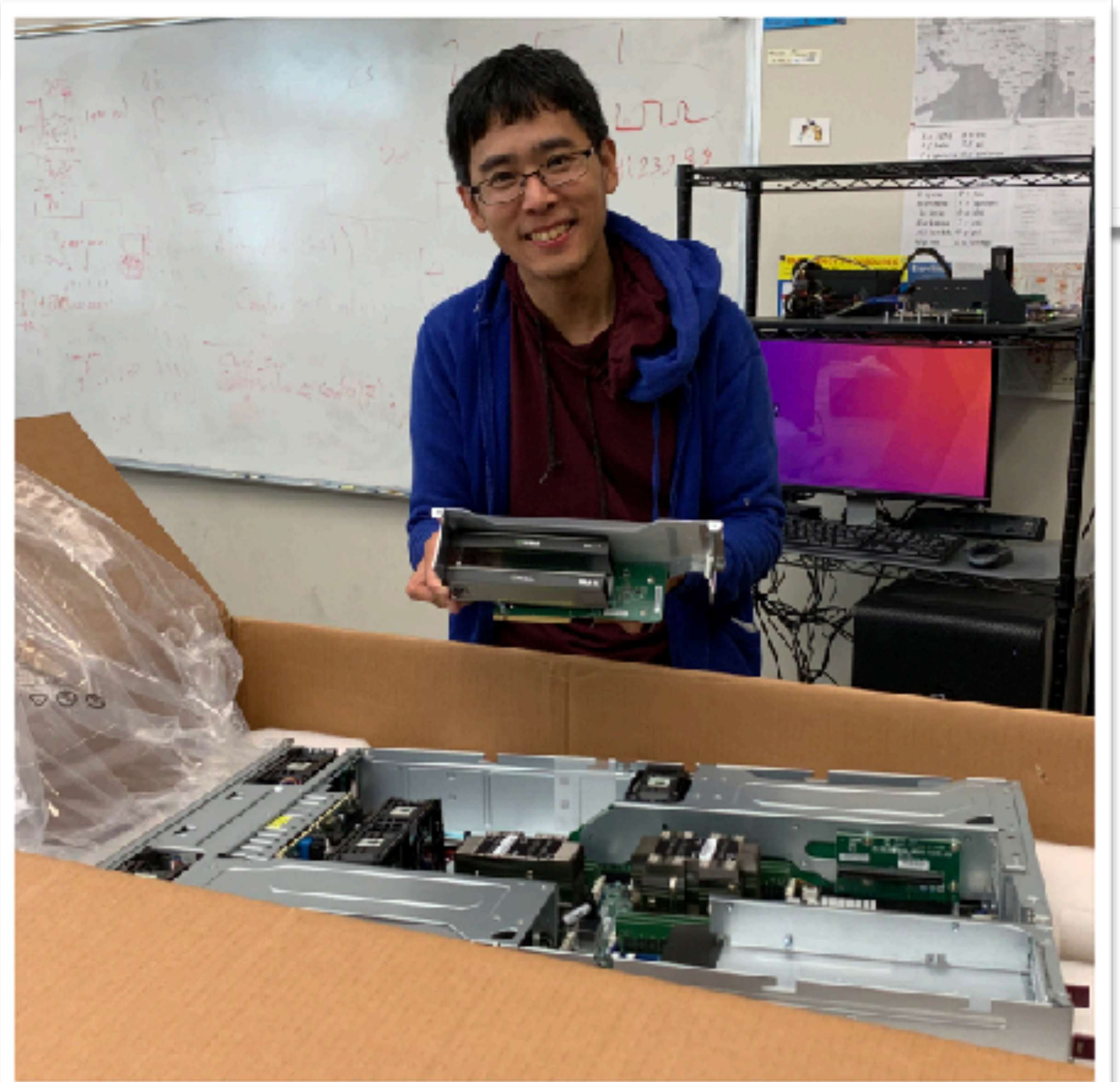
# Desktop Computer

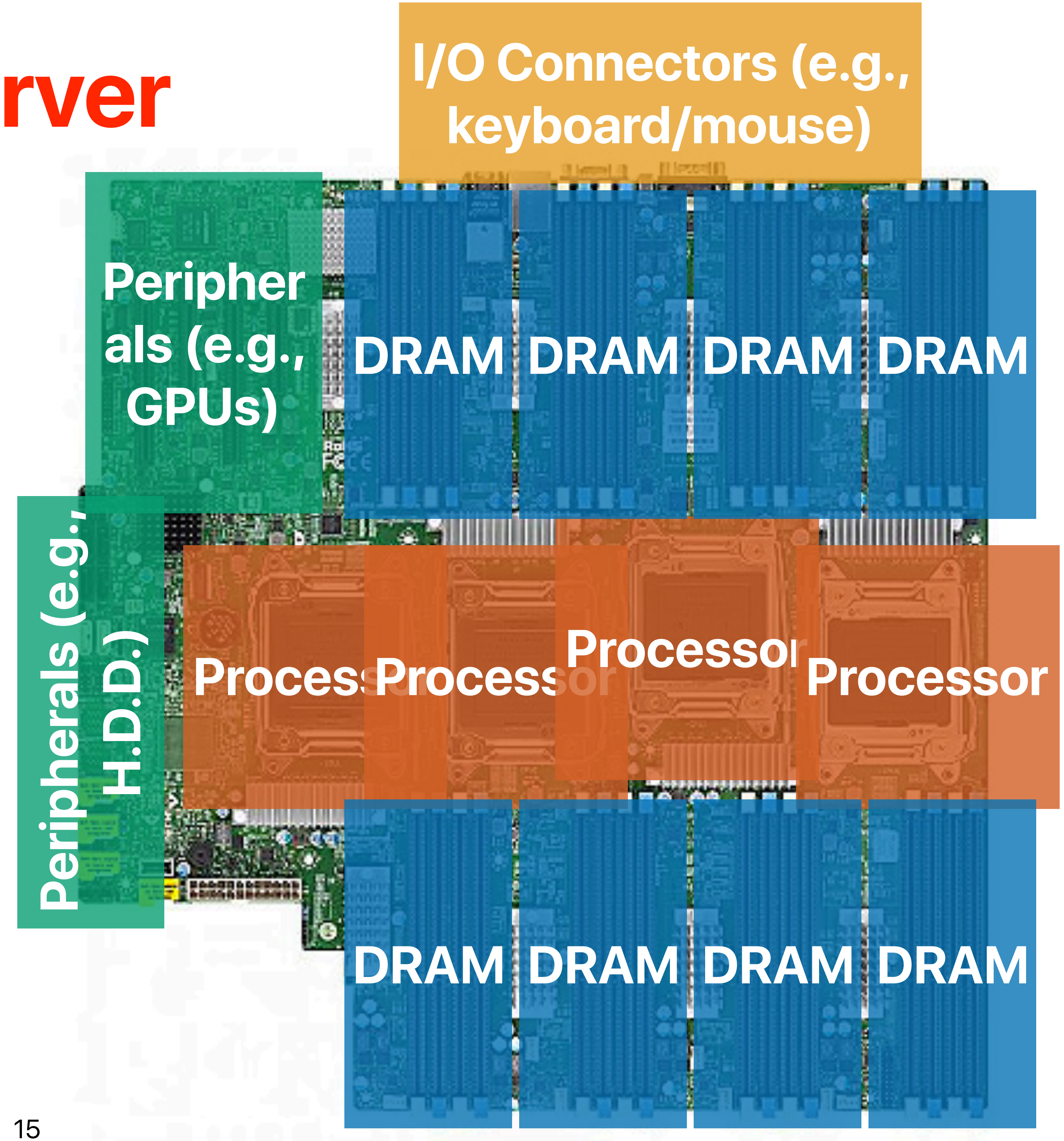


I/O Connectors (e.g., keyboard/mouse)
Processor
DRAM
Peripherals (e.g., GPUs)
Peripherals (e.g., H.D.D.)

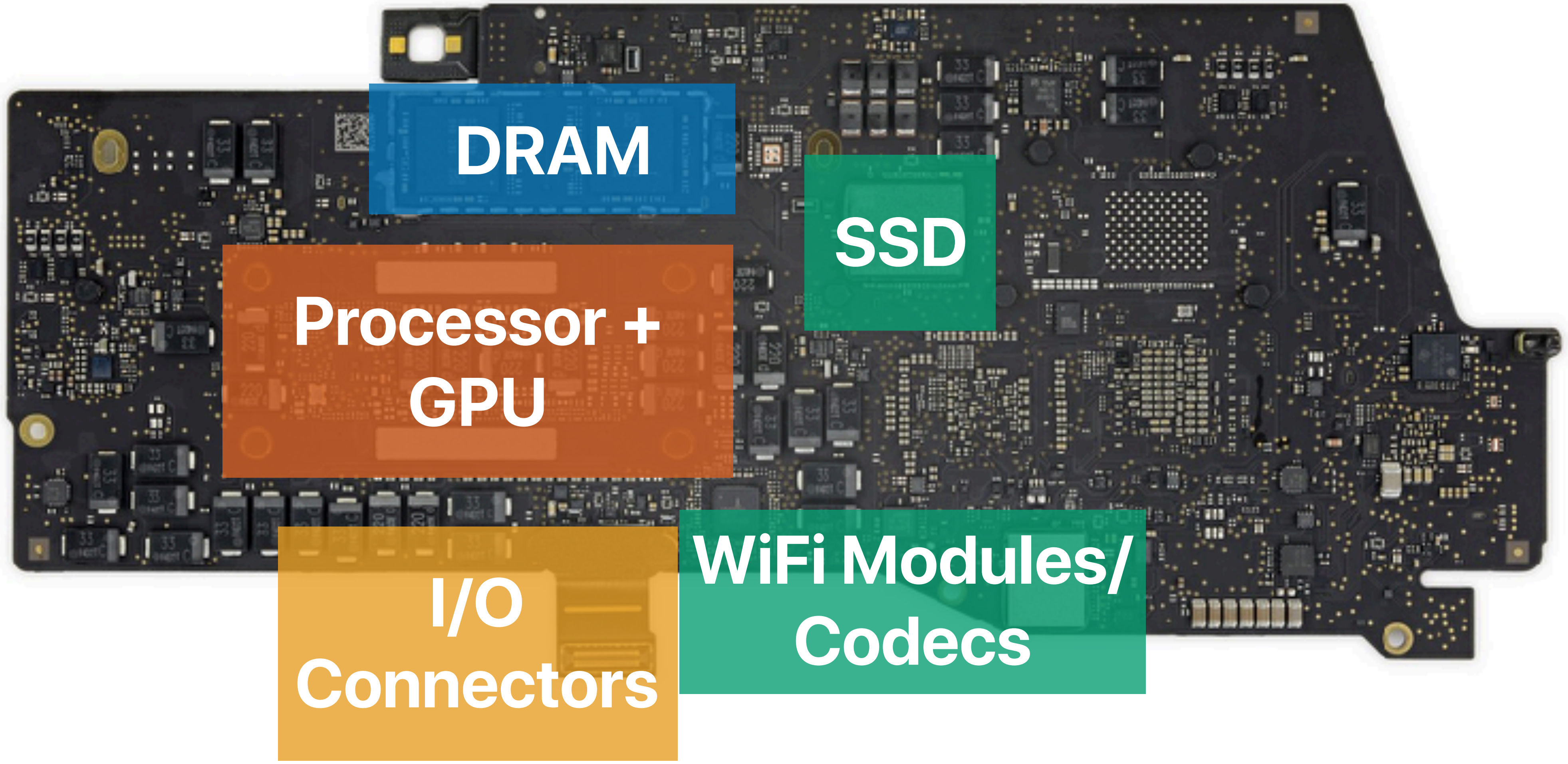
# Server



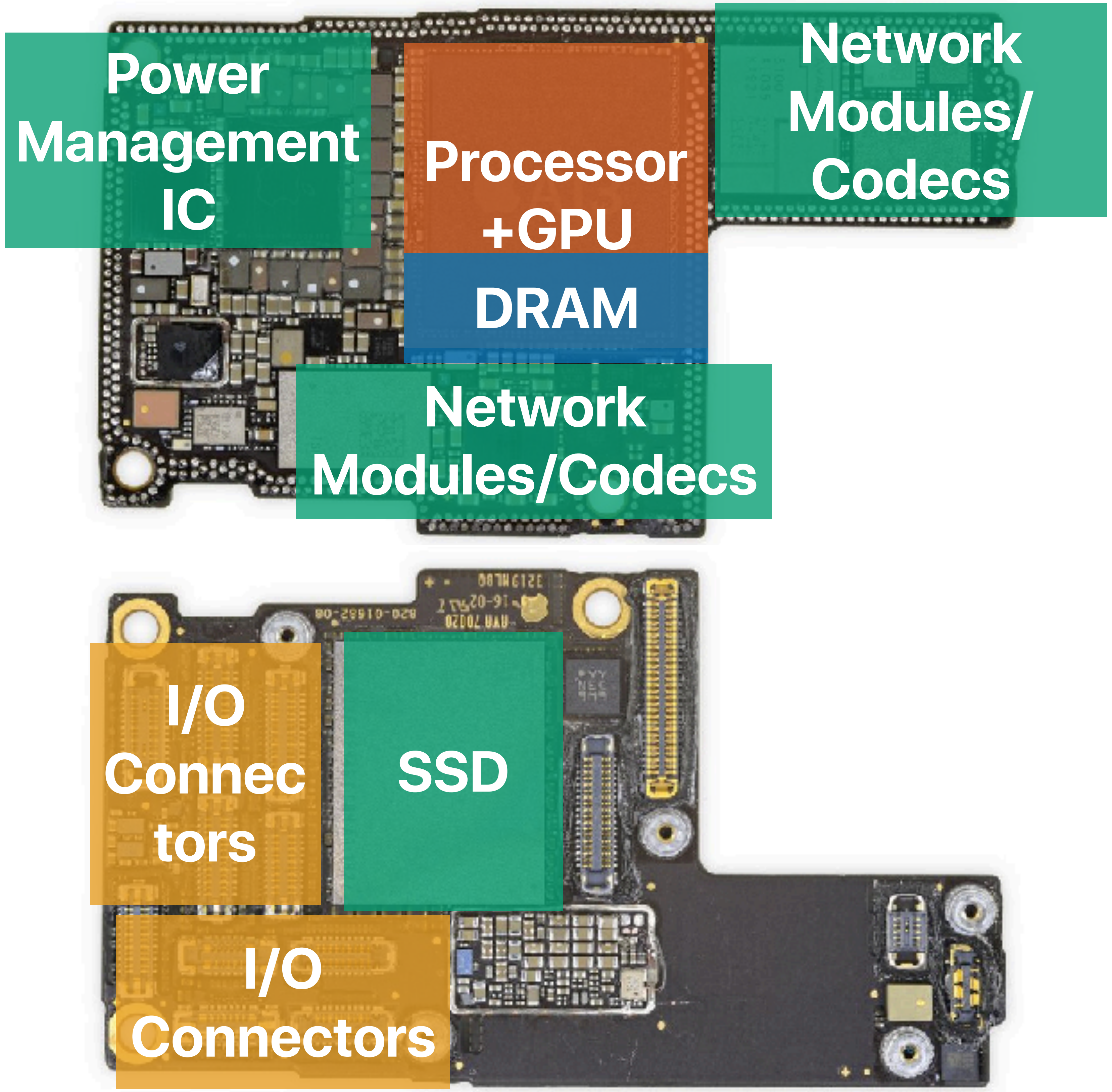



I/O Connectors (e.g., keyboard/mouse)
Peripherals (e.g., GPUs)
DRAM
DRAM
DRAM
DRAM
Peripherals (e.g., H.D.D.)
Processor
Processor
Processor
Processor
DRAM
DRAM
DRAM
DRAM

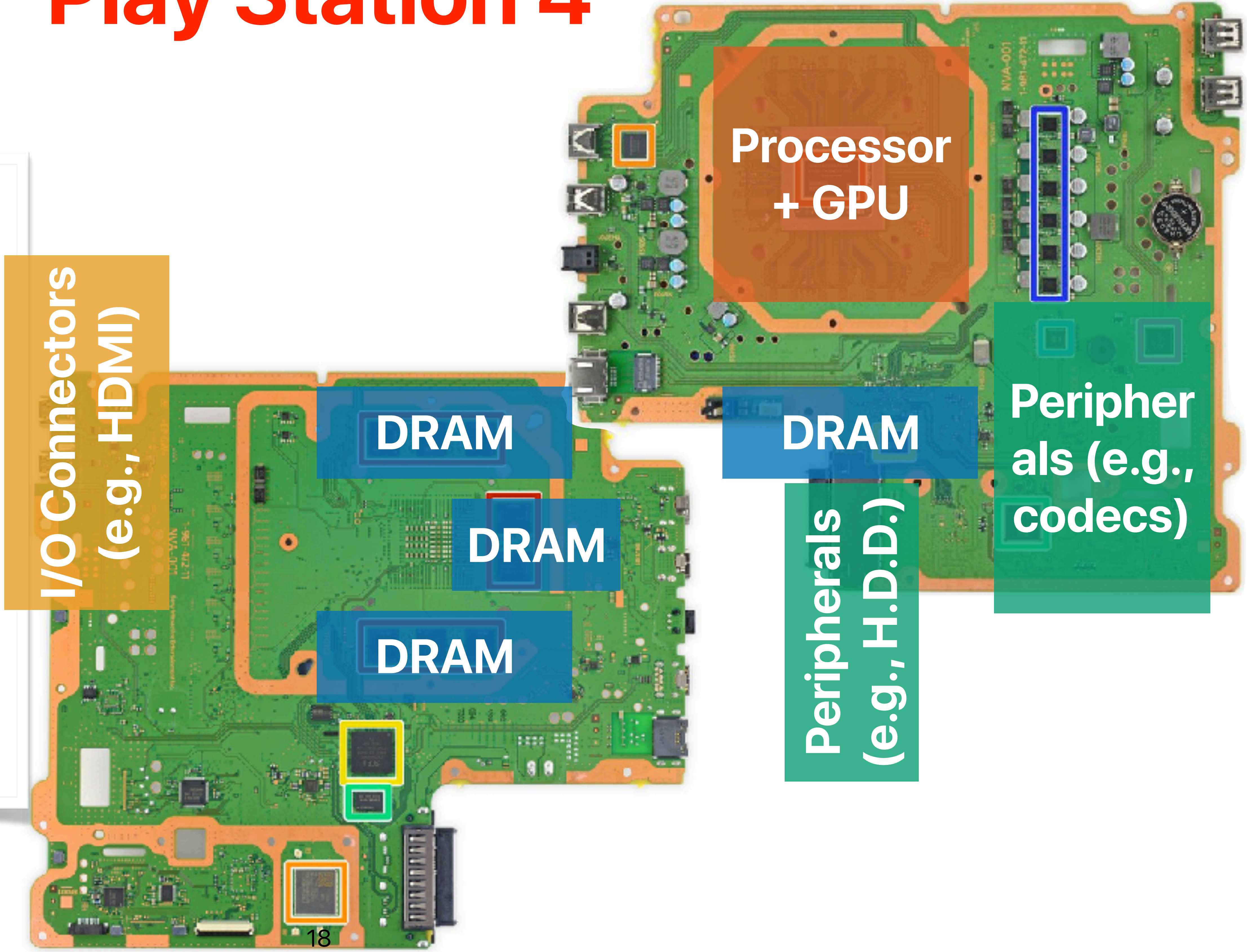
# MacBook Pro 13"



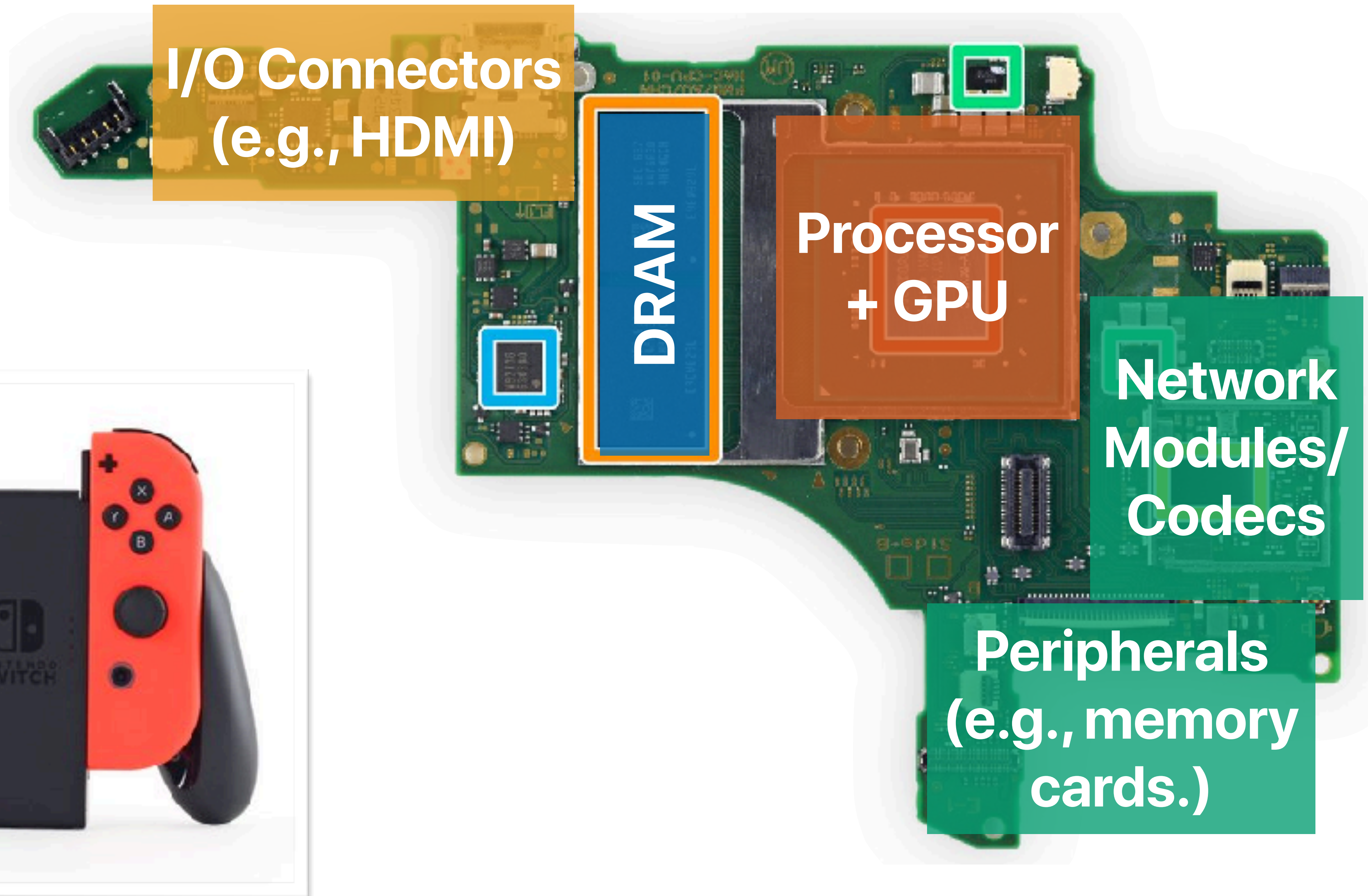
# iPhone 11


Power Management IC
Processor +GPU
Network Modules/ Codecs
DRAM
Network Modules/Codecs
I/O Connec tors
SSD
I/O Connectors

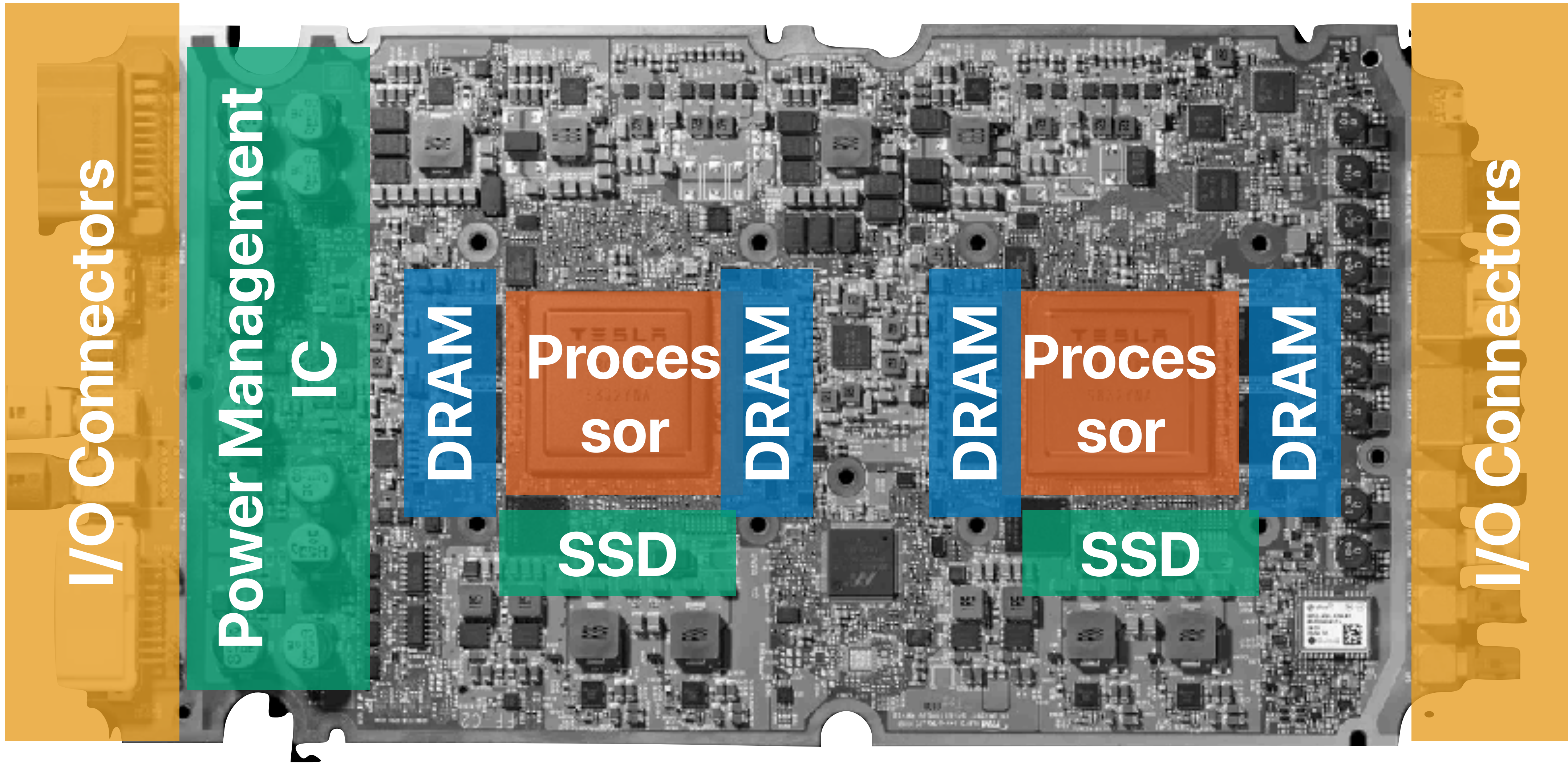
# Play Station 4


Processor + GPU
I/O Connectors (e.g., HDMI)
DRAM
DRAM
DRAM
DRAM
Peripherals (e.g., codecs)
Peripherals (e.g., H.D.D.)

# Nintendo Switch


I/O Connectors (e.g., HDMI)
DRAM
Processor + GPU
Network Modules/ Codecs
Peripherals (e.g., memory cards.)

# Tesla Model 3



# The evolution of OSs

# Batch systems: earliest type of operating systems

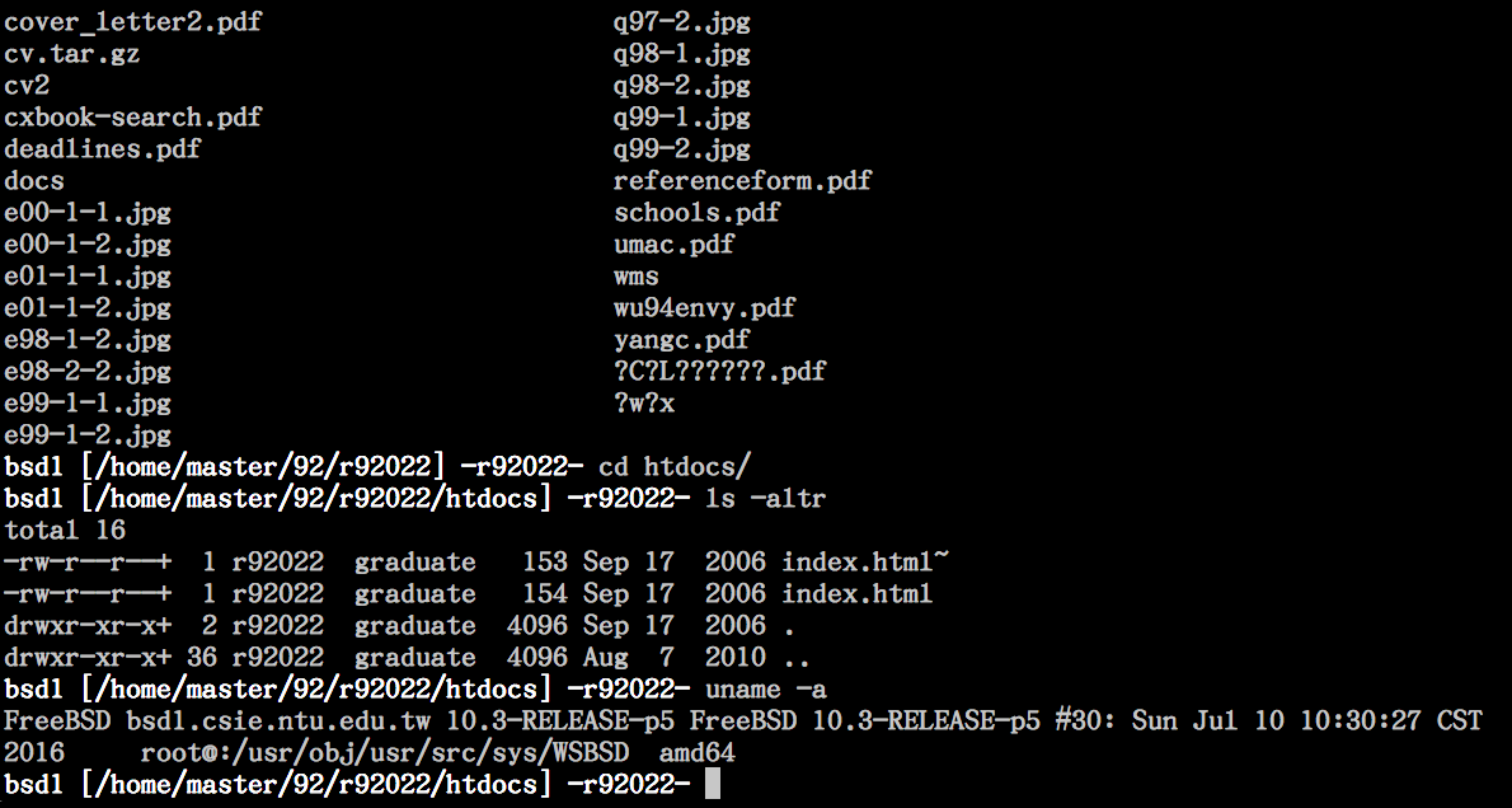
- Executes jobs in-order, one at a time
  - Provide storage (drum, card holder)
  - Load programs into the memory
  - Setup the processor to execute the job
  - Run until the program finishes and load the next in the queue

# 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
  - ……

# 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
- 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

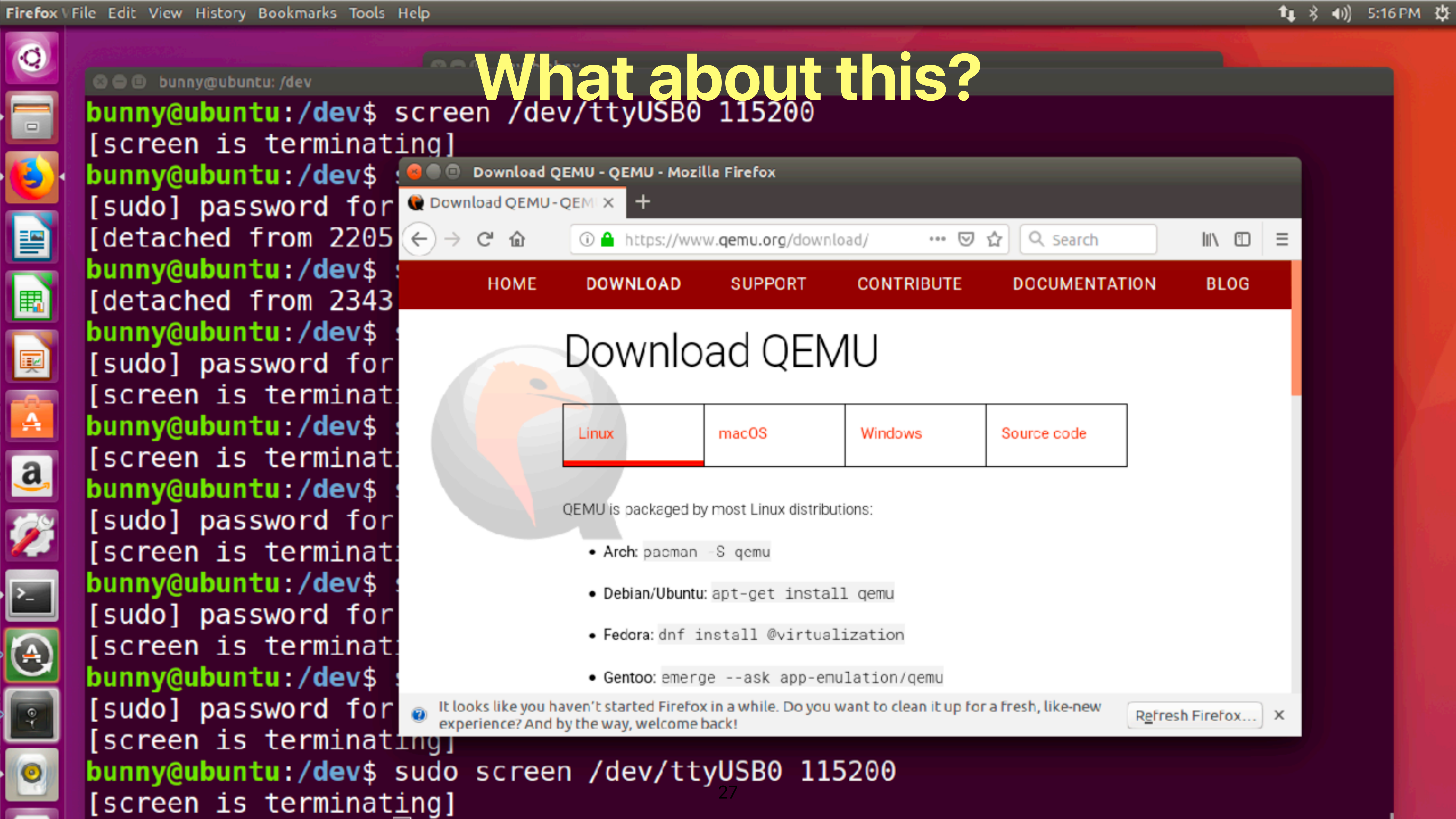


```
cover_letter2.pdf                       q97-2.jpg
cv.tar.gz                               q98-1.jpg
cv2                                     q98-2.jpg
cxbook-search.pdf                       q99-1.jpg
deadlines.pdf                           q99-2.jpg
docs                                    referenceform.pdf
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-
```

# UNIX (cont.)

- Support **multiple users**
- Support **interprocess communication**
- First **portable** operating system
- Everything is a file
  - Max filename length: 255 bytes
- No GUI

# What about this?



```
bunny@ubuntu:/dev$ screen /dev/ttyUSB0 115200
[screen is terminating]
bunny@ubuntu:/dev$ s
[sudo] password for
[detached from 2205
bunny@ubuntu:/dev$ s
[detached from 2343
bunny@ubuntu:/dev$ s
[sudo] password for
[screen is terminat
bunny@ubuntu:/dev$ s
[screen is terminat
bunny@ubuntu:/dev$ s
[sudo] password for
[screen is terminat
bunny@ubuntu:/dev$ s
[sudo] password for
[screen is terminat
bunny@ubuntu:/dev$ s
[sudo] password for
[screen is terminating]
bunny@ubuntu:/dev$ sudo screen /dev/ttyUSB0 115200
[screen is terminating]
```

## Download QEMU

QEMU is packaged by most Linux distributions:

- Arch: `pacman -S qemu`
- Debian/Ubuntu: `apt-get install qemu`
- Fedora: `dnf install @virtualization`
- Gentoo: `emerge --ask app-emulation/qemu`

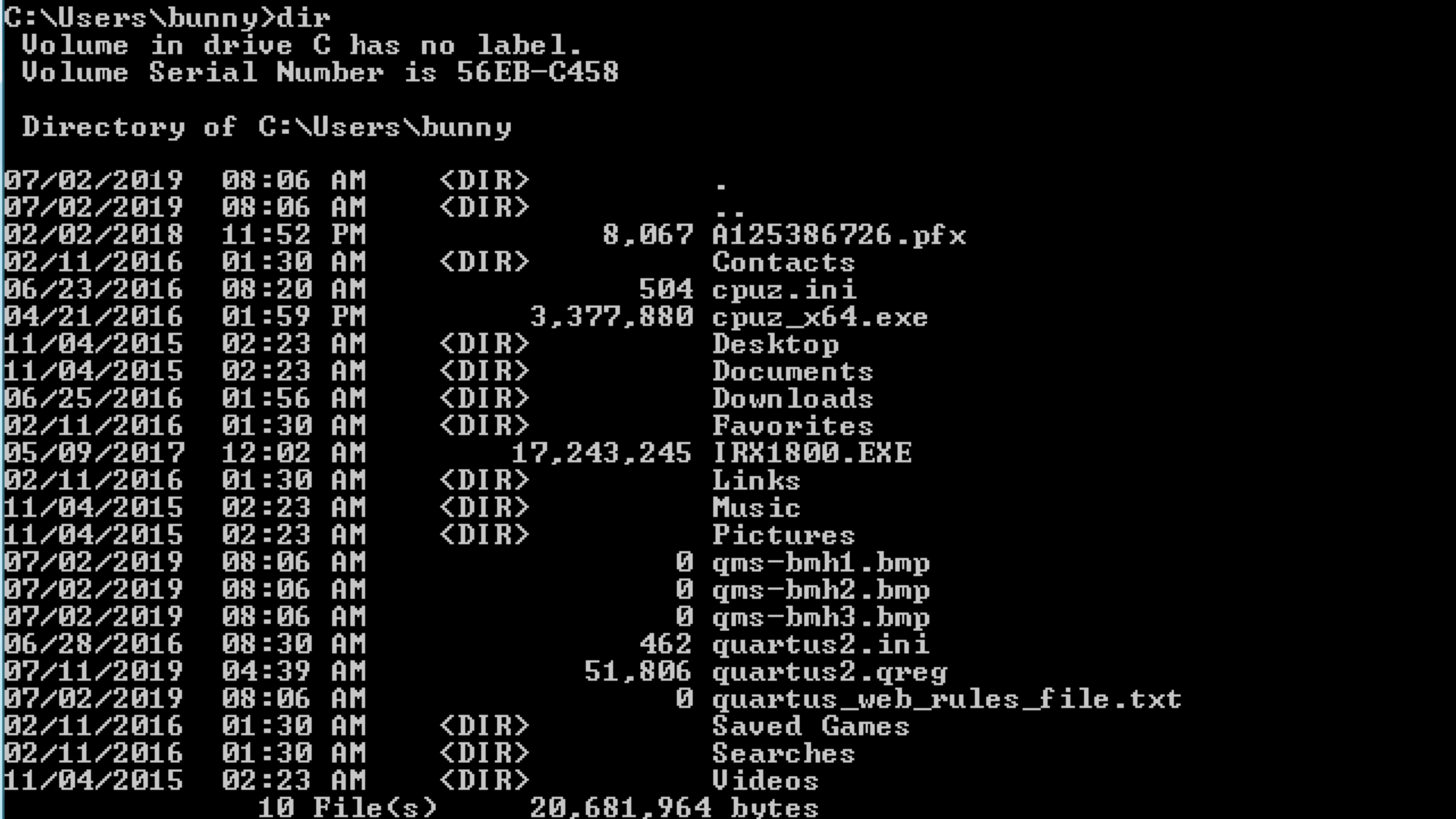
It looks like you haven't started Firefox in a while. Do you want to clean it up for a fresh, like-new experience? And by the way, welcome back!

# UNIX (cont.)

- Support **multiple users**
- Support **interprocess communication**
- First **portable** operating system
- Everything is a file
  - Max filename length: 255 bytes
- No GUI
  - X Window provides a GUI for UNIX since 1987.
  - Now X Window is replaced by X.Org — still not part of the default system

# UNIX (cont.)

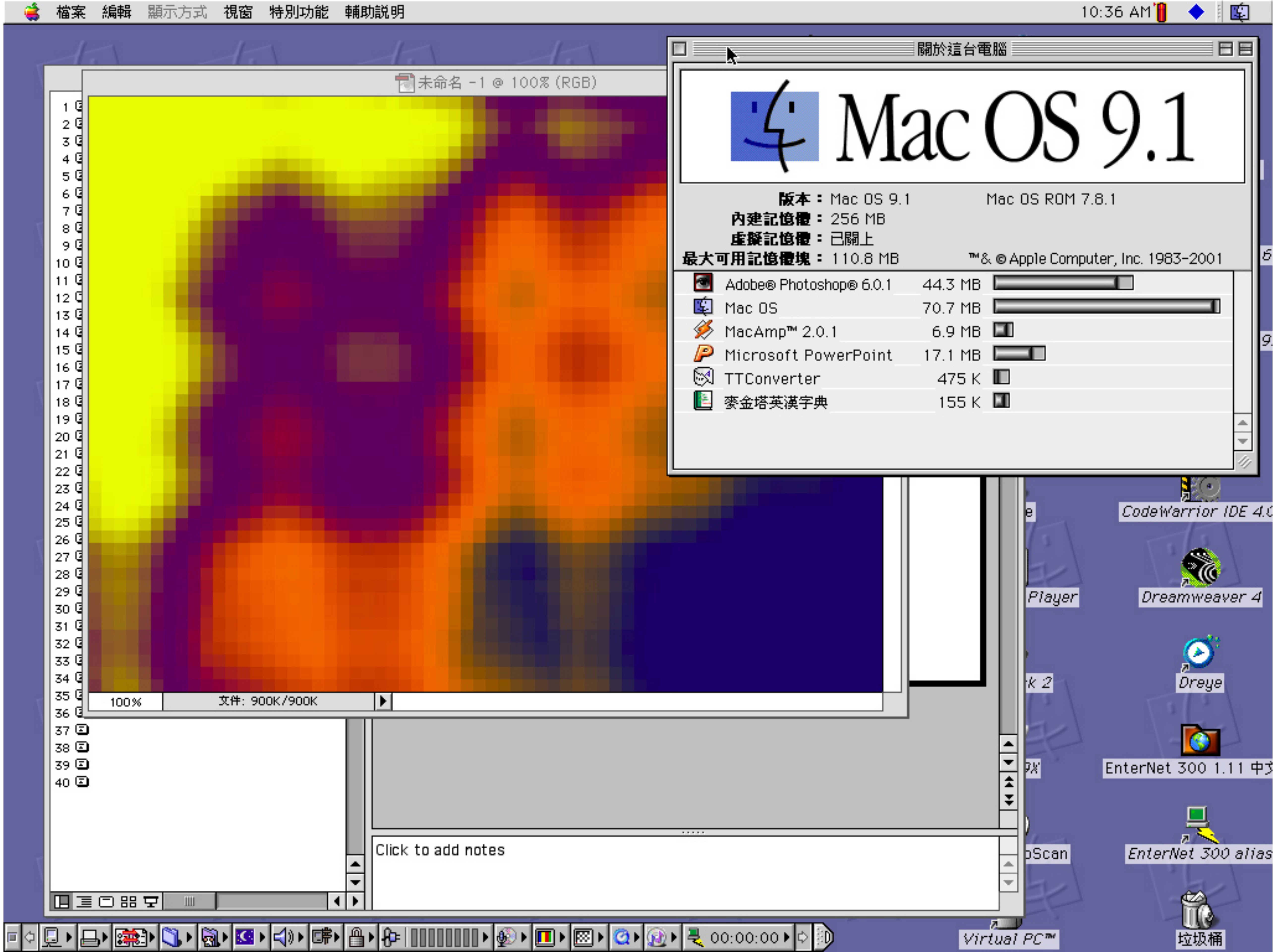
- Descendants
  - BSD (Berkeley Software Distribution)
    - FreeBSD, OpenBSD, NetBSD
    - The base of Apple’s MacOS X and iOS
  - Solaris
  - IBM AIX
- Affected
  - Linux
    - Started in 1983 by Richard Stallman
    - Linus Torvalds, principal developer of the Linux kernel



```
C:\Users\bunny>dir
 Volume in drive C has no label.
 Volume Serial Number is 56EB-C458

 Directory of C:\Users\bunny

07/02/2019  08:06 AM    <DIR>          .
07/02/2019  08:06 AM    <DIR>          ..
02/02/2018  11:52 PM             8,067 A125386726.pfx
02/11/2016  01:30 AM    <DIR>          Contacts
06/23/2016  08:20 AM               504 cpuz.ini
04/21/2016  01:59 PM         3,377,880 cpuz_x64.exe
11/04/2015  02:23 AM    <DIR>          Desktop
11/04/2015  02:23 AM    <DIR>          Documents
06/25/2016  01:56 AM    <DIR>          Downloads
02/11/2016  01:30 AM    <DIR>          Favorites
05/09/2017  12:02 AM        17,243,245 IRX1800.EXE
02/11/2016  01:30 AM    <DIR>          Links
11/04/2015  02:23 AM    <DIR>          Music
11/04/2015  02:23 AM    <DIR>          Pictures
07/02/2019  08:06 AM                 0 qms-bmh1.bmp
07/02/2019  08:06 AM                 0 qms-bmh2.bmp
07/02/2019  08:06 AM                 0 qms-bmh3.bmp
06/28/2016  08:30 AM               462 quartus2.ini
07/11/2019  04:39 AM            51,806 quartus2.qreg
07/02/2019  08:06 AM                 0 quartus_web_rules_file.txt
02/11/2016  01:30 AM    <DIR>          Saved Games
02/11/2016  01:30 AM    <DIR>          Searches
11/04/2015  02:23 AM    <DIR>          Videos
              10 File(s)     20,681,964 bytes
```

# 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



關於這台電腦

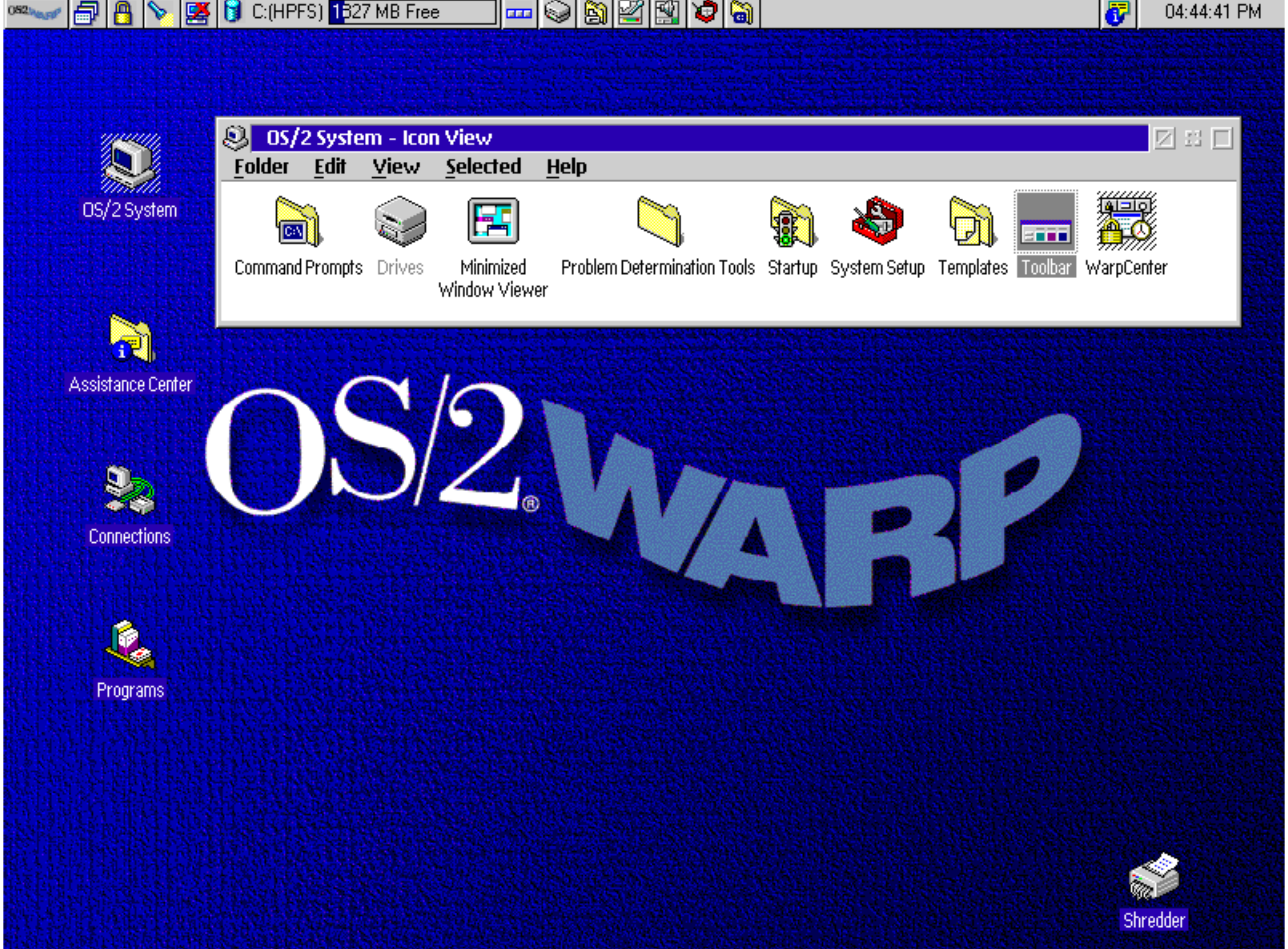
Mac OS 9.1

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 |

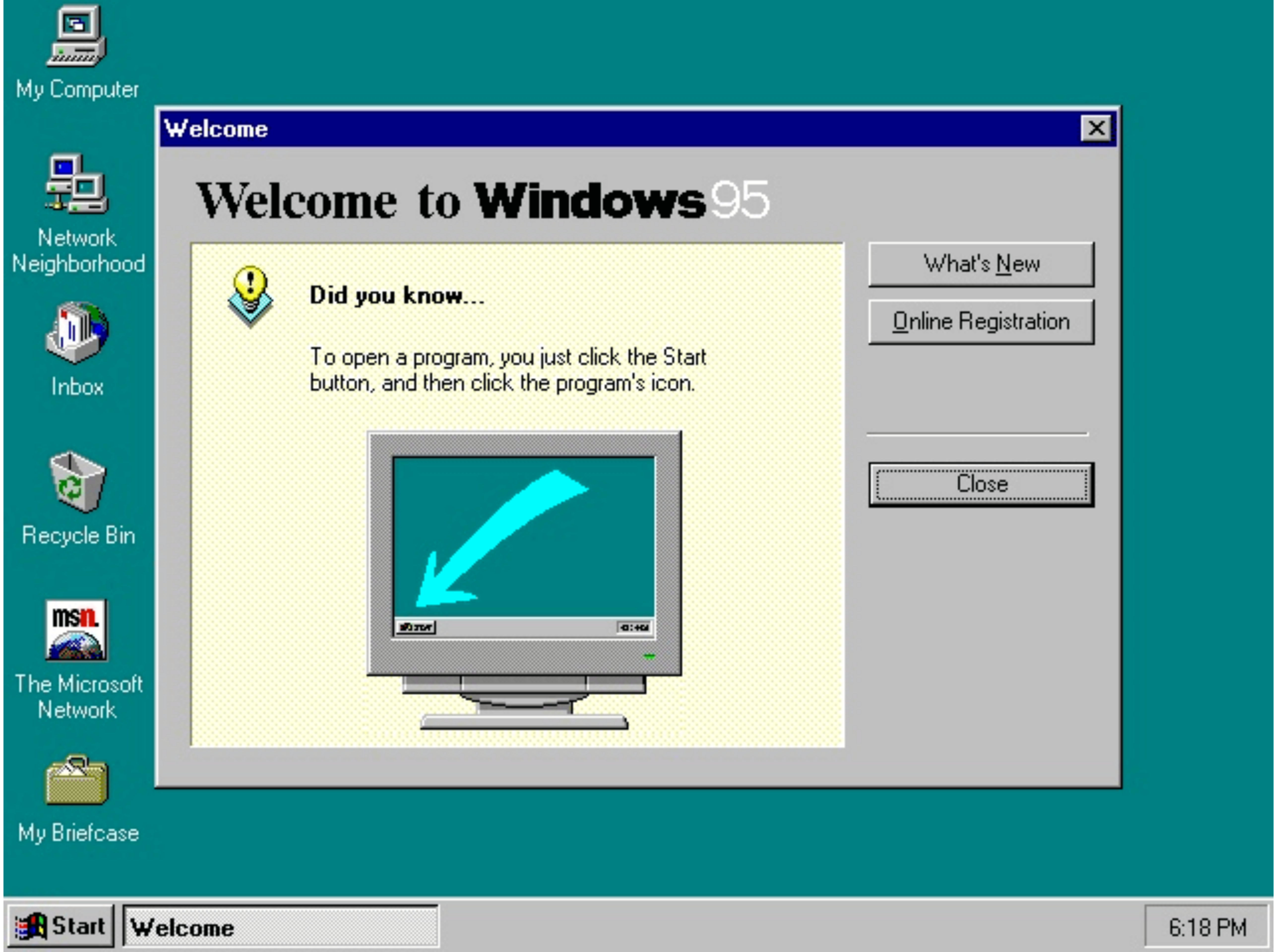
# MacOS "Classic"

- 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


C:(HPFS) 1327 MB Free
04:44:41 PM
OS/2 System - Icon View
Folder
Edit
View
Selected
Help
Command Prompts
Drives
Minimized Window Viewer
Problem Determination Tools
Startup
System Setup
Templates
Toolbar
WarpCenter
OS/2 System
Assistance Center
Connections
Programs
Shredder
OS/2®
WARP

# OS/2 Warp

- Released in 1987, discontinued in 2006
- First true 32-bit OS on x86
- Was developed together by IBM/MS to be the GUI OS alternative to DOS



Welcome

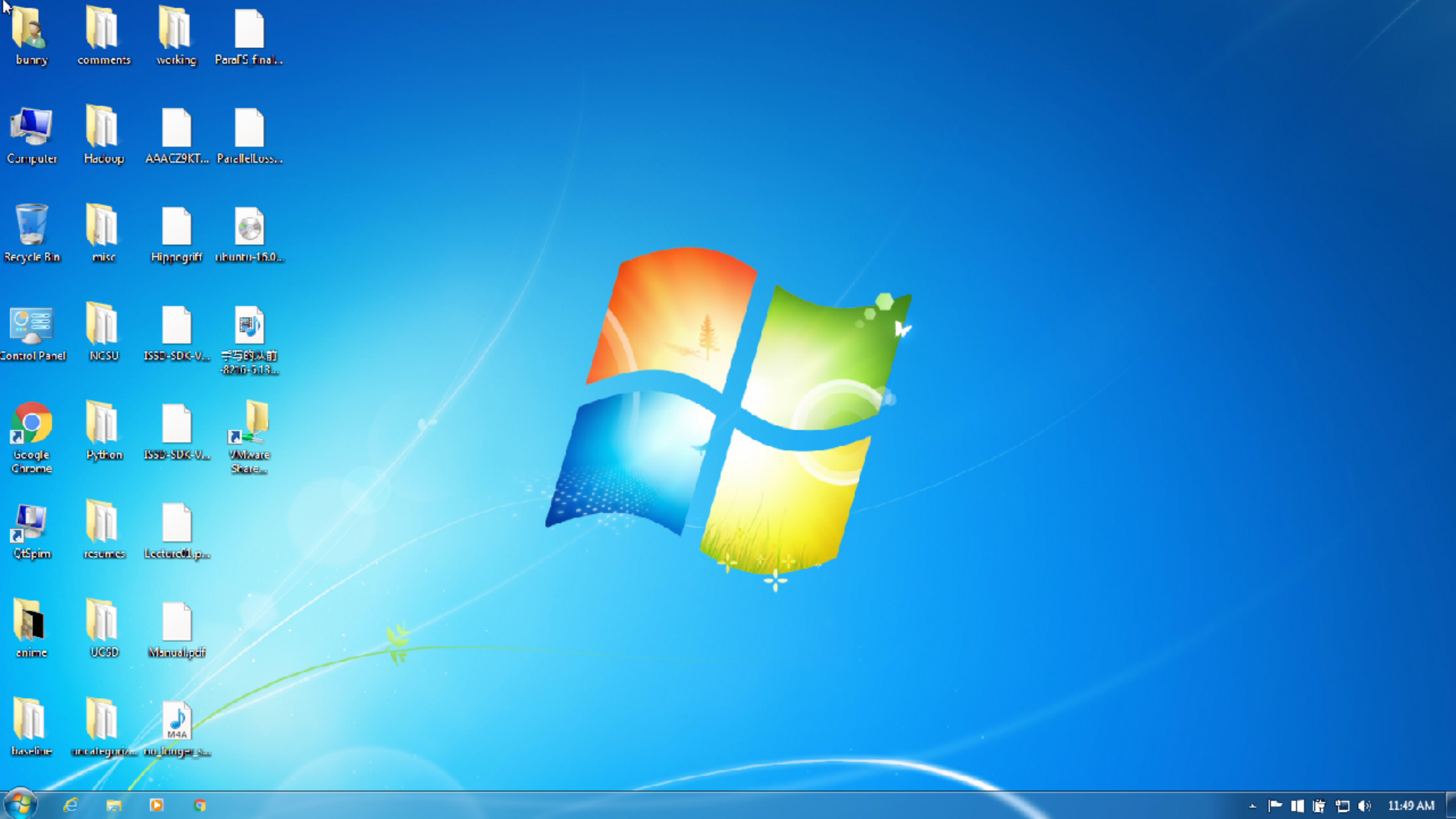
Welcome to Windows95

Did you know...

To open a program, you just click the Start button, and then click the program's icon.

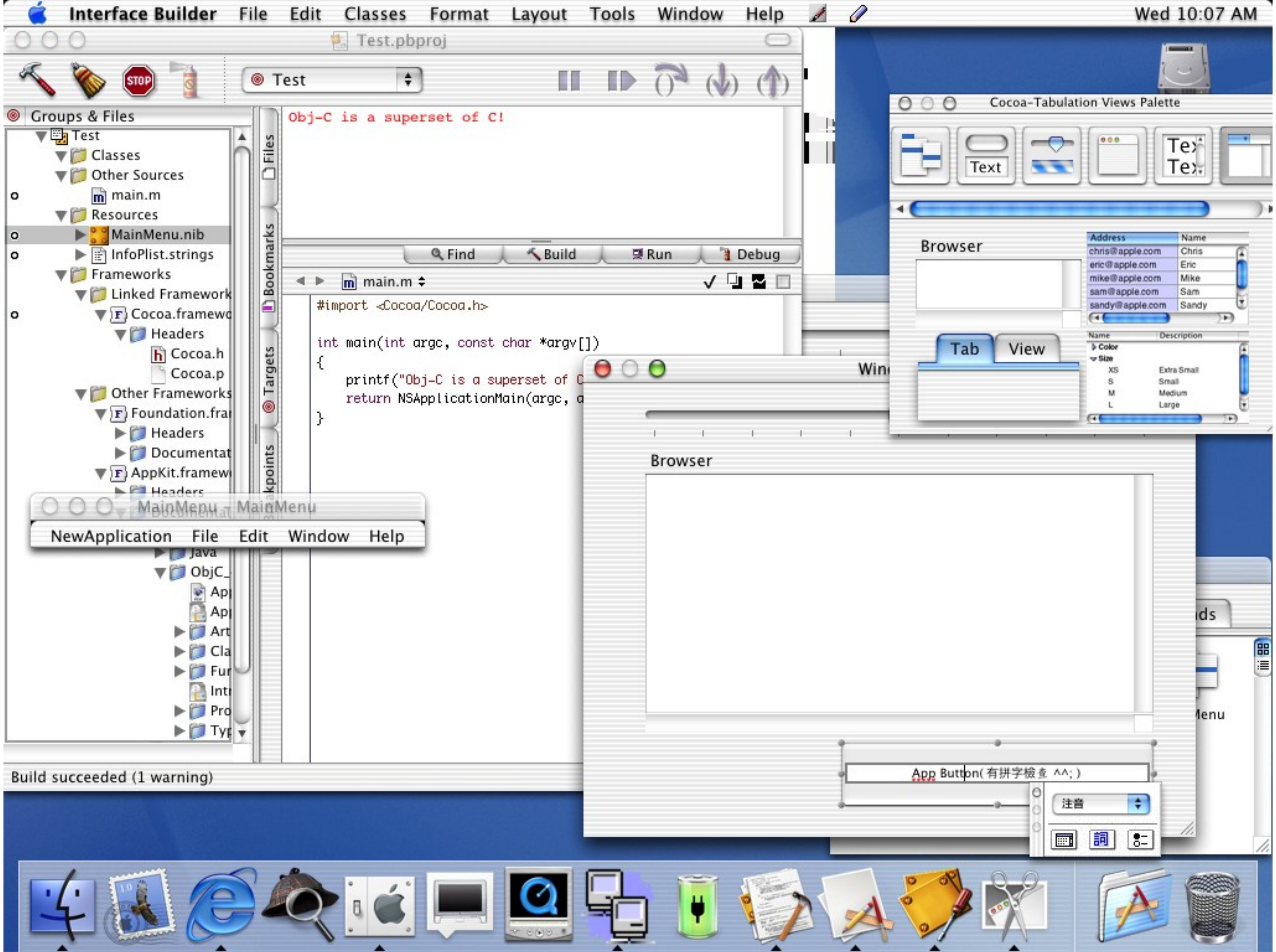
# Windows 95/98/ME

- Before Windows 95, was 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 95 as Windows 95


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 -8216-513...
Google Chrome
Python
ISSD-SDK-V...
VMware Share...
QtSpim
resumes
Lecture01.p...
anime
UCSD
Manual.pdf
baseline
uncategoriz...
M4A
no_longer_s...
11:49 AM

# Windows NT/2000/XP/Vista/7/8/10

- 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)



```
Obj-C is a superset of C!
```

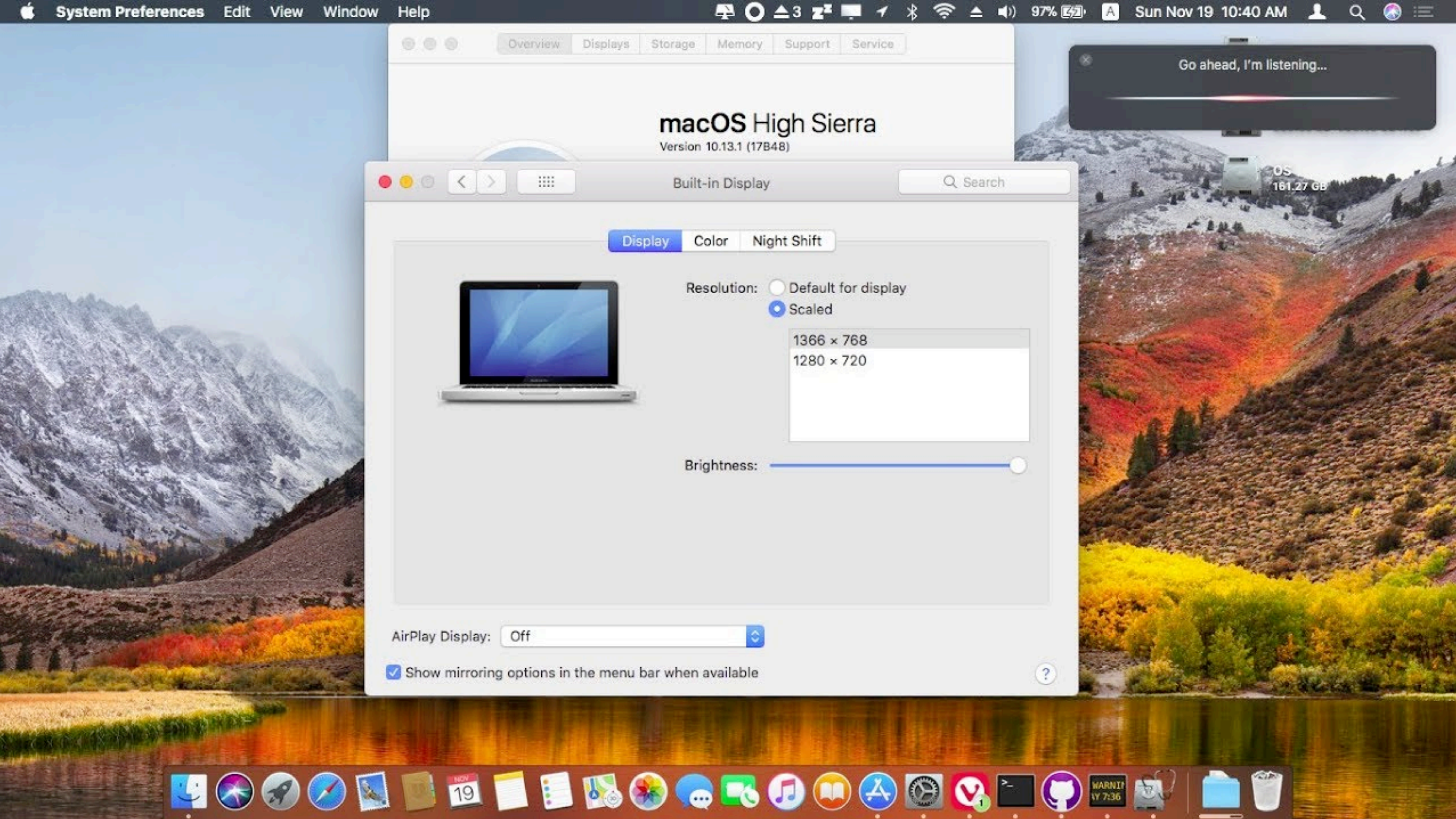
```
#import <Cocoa/Cocoa.h>

int main(int argc, const char *argv[])
{
    printf("Obj-C is a superset of C
    return NSApplicationMain(argc, a
}
```

Build succeeded (1 warning)

| Address | Name |
|---|---|
| chris@apple.com | Chris |
| eric@apple.com | Eric |
| mike@apple.com | Mike |
| sam@apple.com | Sam |
| sandy@apple.com | Sandy |

| Name | Description |
|---|---|
| ▷ Color | |
| ▽ Size | |
| XS | Extra Small |
| S | Small |
| M | Medium |
| L | Large |

App Button(有拼字檢查 ^^; )



macOS High Sierra

Version 10.13.1 (17B48)

Go ahead, I'm listening...

Built-in Display

Resolution: ○ Default for display ● Scaled

1366 × 768

1280 × 720

Brightness:

AirPlay Display: Off

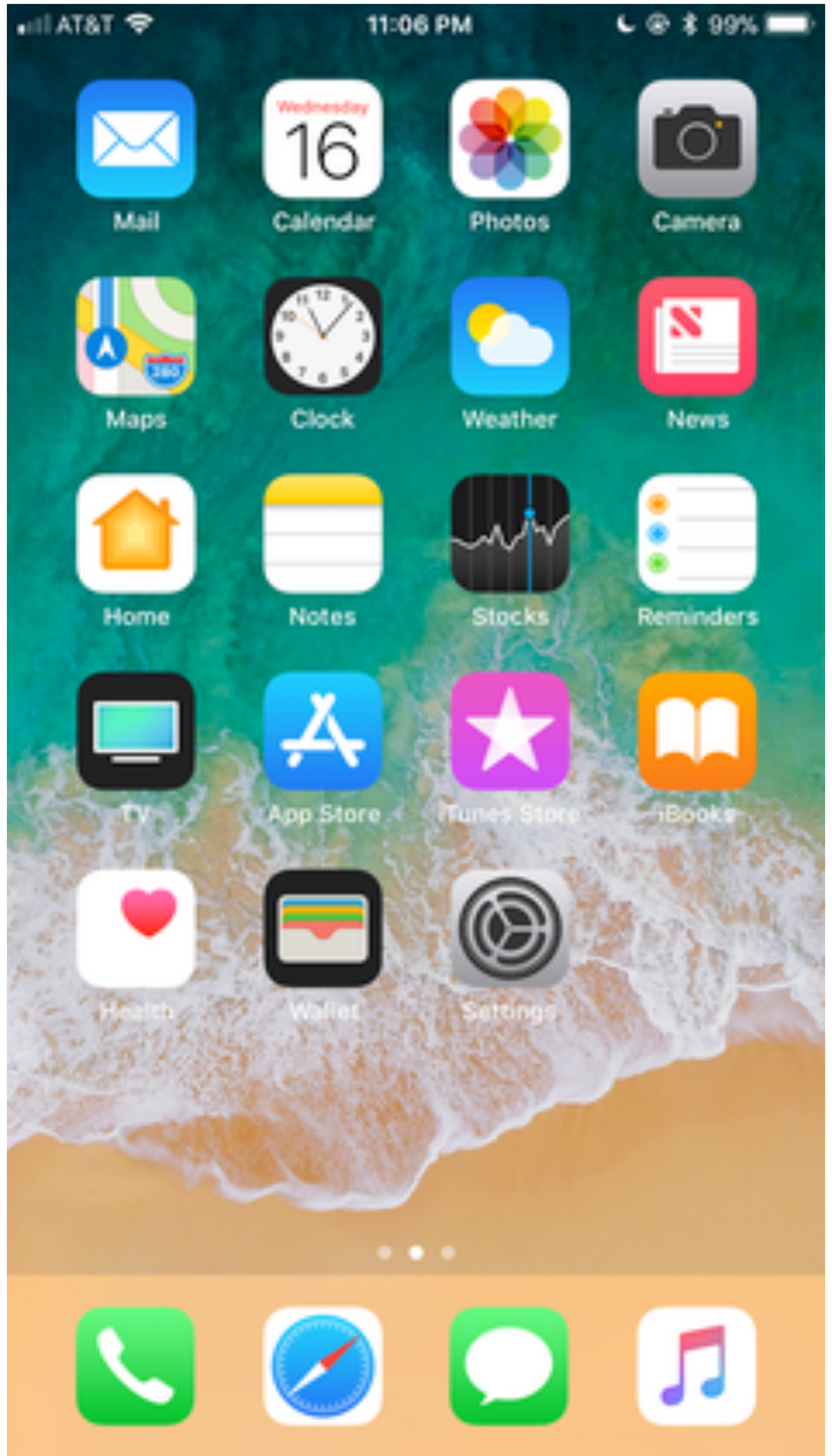
☑ Show mirroring options in the menu bar when available

# 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

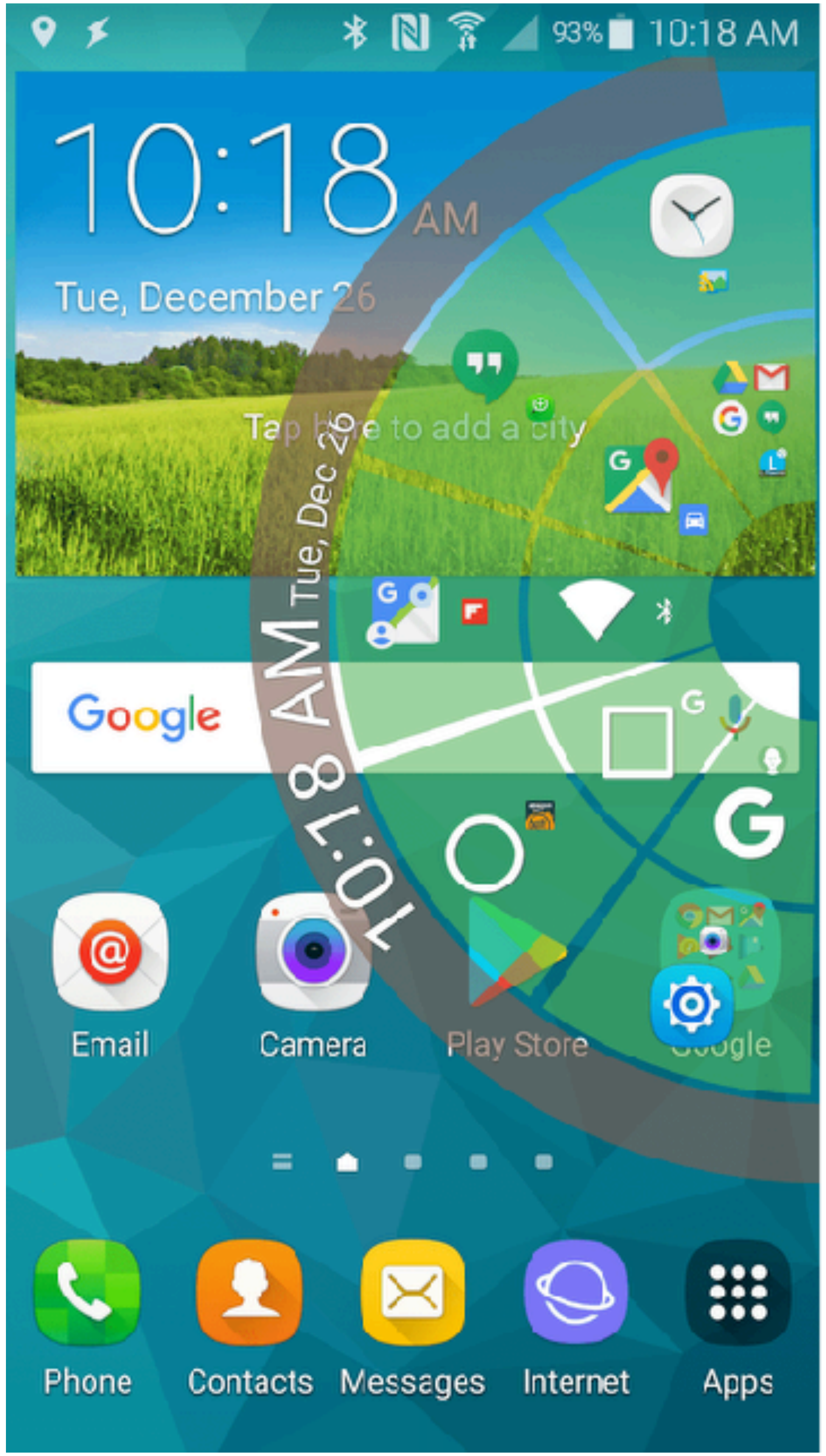
# iOS

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



# Android

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



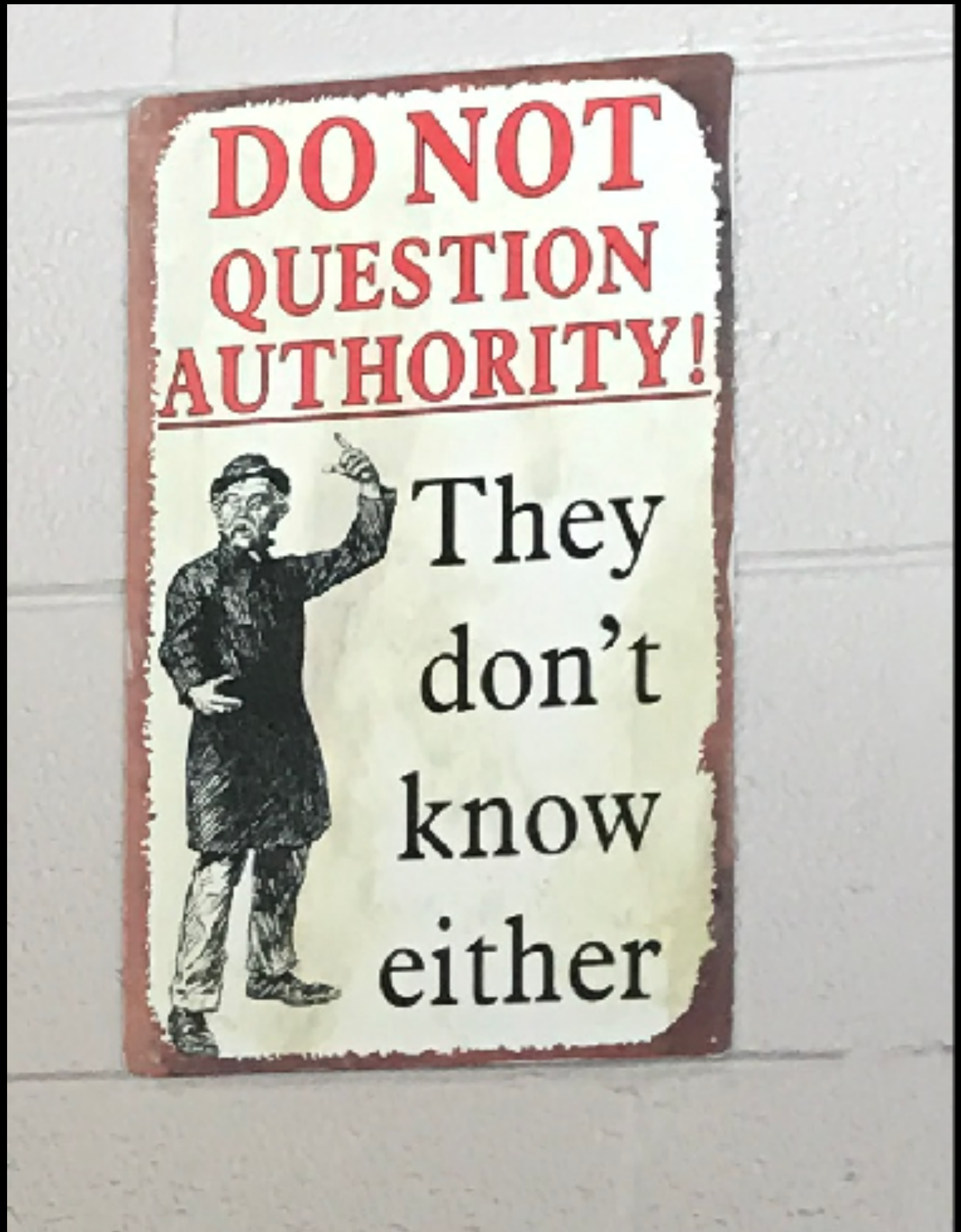
# What modern operating systems support?

- **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

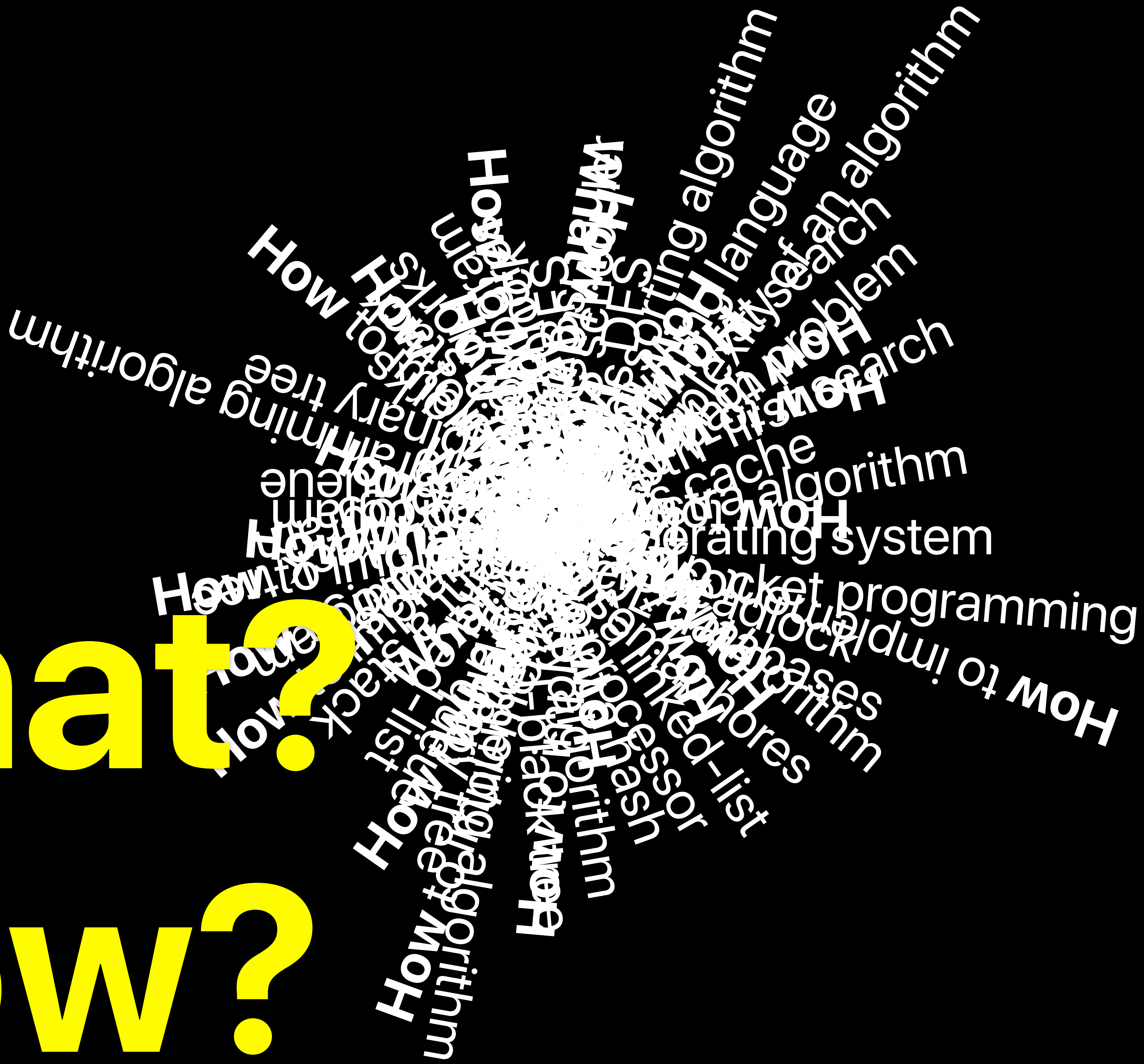
# CS202: Advanced Operating Systems

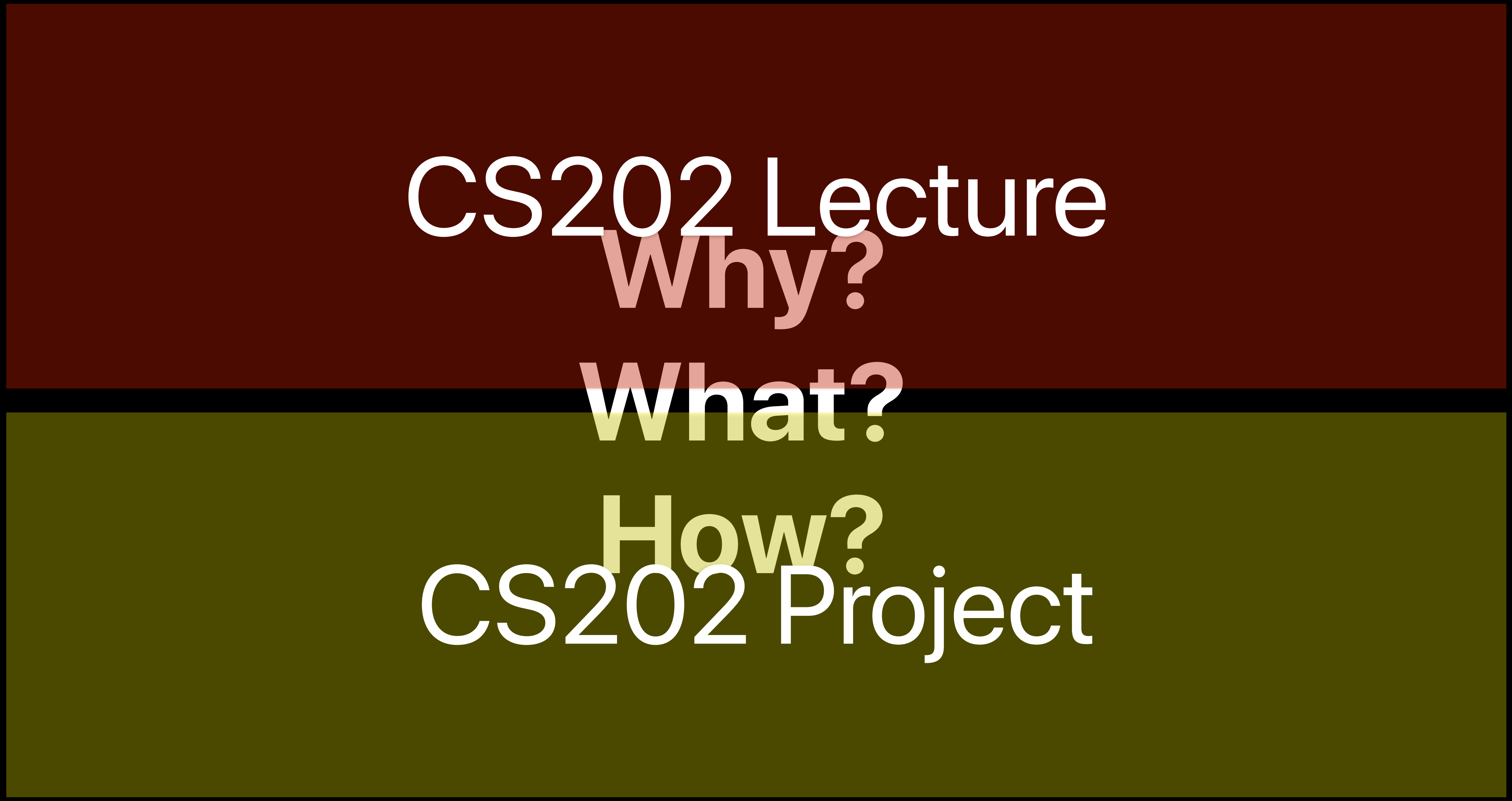
# Advanced

**Why?**
**What?**
**How?**



# What?
# How?


sorting algorithm
language
an algorithm
search
problem
binary search
cache
Dijkstra algorithm
operating system
socket programming
deadlock
databases
How to impl
algorithm
semaphores
linked-list
processor
hash
algorithm
binary tree
ming algorithm
queue
stack
- list
How
How to



CS202 Lecture

Why?

What?

How?

CS202 Project

# Logistics

# Course resource

- Lectures: TuTh 6:30p-7:50p, WCH 142
- Schedule, slides on **course webpage**: https://www.escalab.org/classes/cs202-2020wi/
- Discussion on **piazza**: https://piazza.com/class/k53188n4ht71em
- Reading quizzes, homework submissions on **iLearn**: https://ilearn.ucr.edu/
- We do **podcasting**: TBA

# Instructor — Hung-Wei Tseng

- Website: https://intra.engr.ucr.edu/~htseng/
- Office hour:
  WTh 1:00p-2:00p @ WCH 406
- E-mail: htseng@ucr.edu
- BS/MS in **Computer Science**, National Taiwan University
- PhD in **Computer Science**, University of California, San Diego
- Research Interests
  - Intelligent storage devices
  - Non-volatile memory based systems
  - Near-data processing
  - Anything could accelerate applications

# Teaching Assistant — Zhenxiao Qi

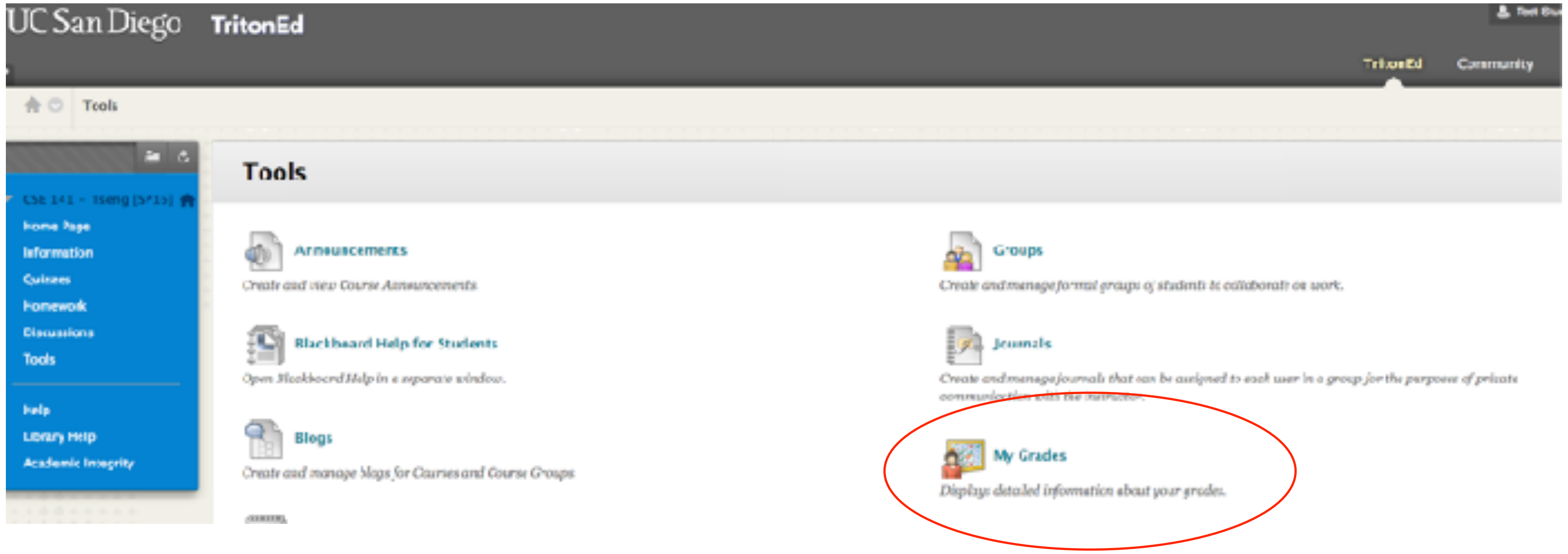
- Office hours: TuF 2p-3p @ WCH 110
- E-mail: zqi020@ucr.edu

# Your tasks

- Login/discussion in iLearn 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 iLearn (10%)
- Come to class (10%) — through Clickers
- **Project (20%)**
- **Midterm (20%)** — take home/online midterm
- **Final (40%)**

# Grading

- You can see your grades on iLearn.



Tools

Announcements
Create and view Course Announcements

Blackboard Help for Students
Open Blackboard Help in a separate window.

Blogs
Create and manage blogs for Courses and Course Groups

Groups
Create and manage formal groups of students to collaborate on work.

Journals
Create and manage journals that can be assigned to each user in a group for the purpose of private communication with the instructor.

My Grades
Displays detailed information about your grades.

- 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...

You

Me

# Peer instruction

- 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.
- You will be GET CREDIT for your efforts to learn in class
  - By answering questions with a clicker (Poll Everywhere)
  - Answer 80% of the clicker questions in class, get 10% of your final grade

# Peer instruction

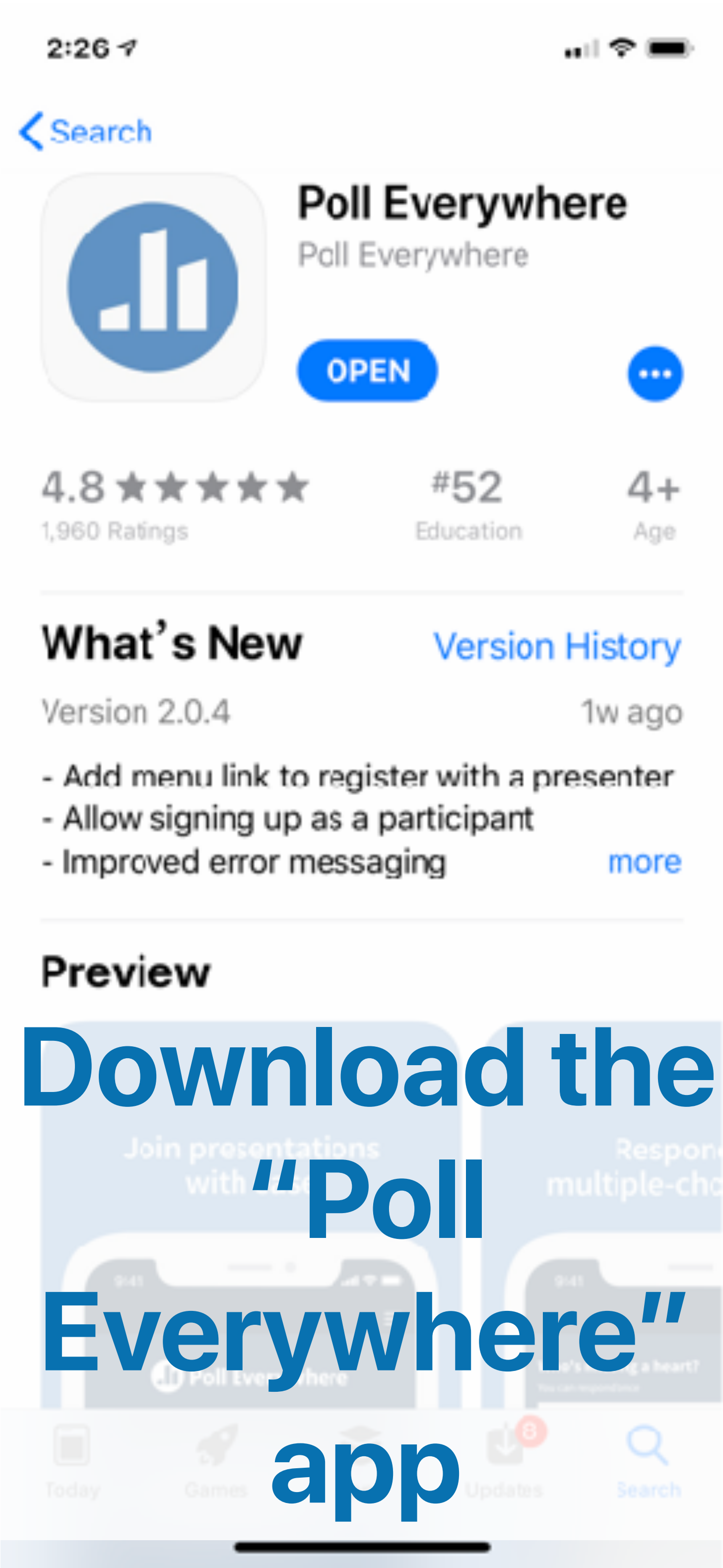
- Before the lecture — You need to complete the required **reading**
- During the lecture — I'll bring in activities to ENGAGE you in exploring your understanding of the material
  - Popup questions
  - Individual **thinking** — use your clicker to express your opinion
  - Group **discussion** — use your clicker to express your group's opinion
  - Whole-classroom **discussion** — we would like to hear from you

**Read** **Think** **Discuss**

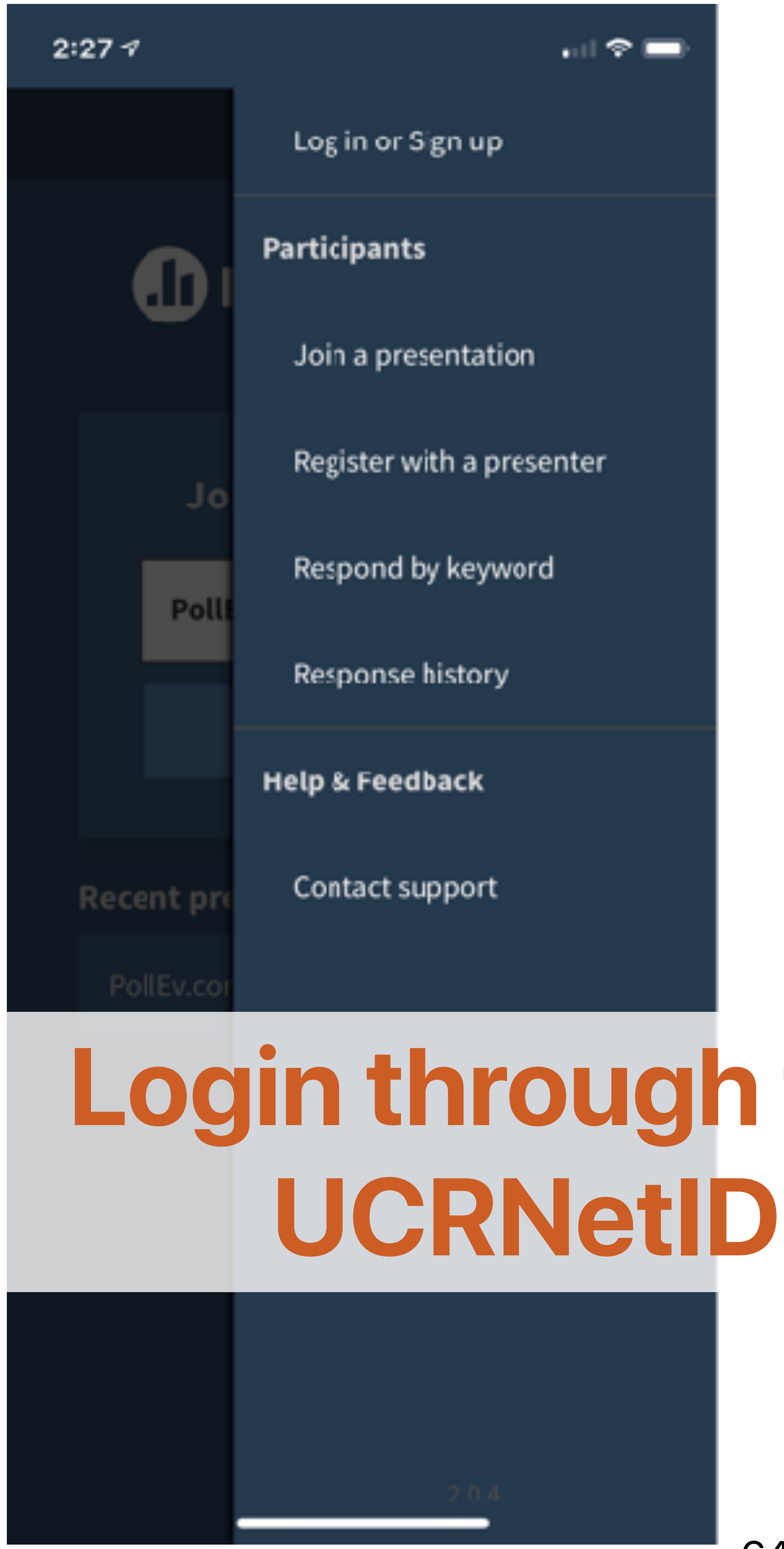
# Before lectures: reading quizzes

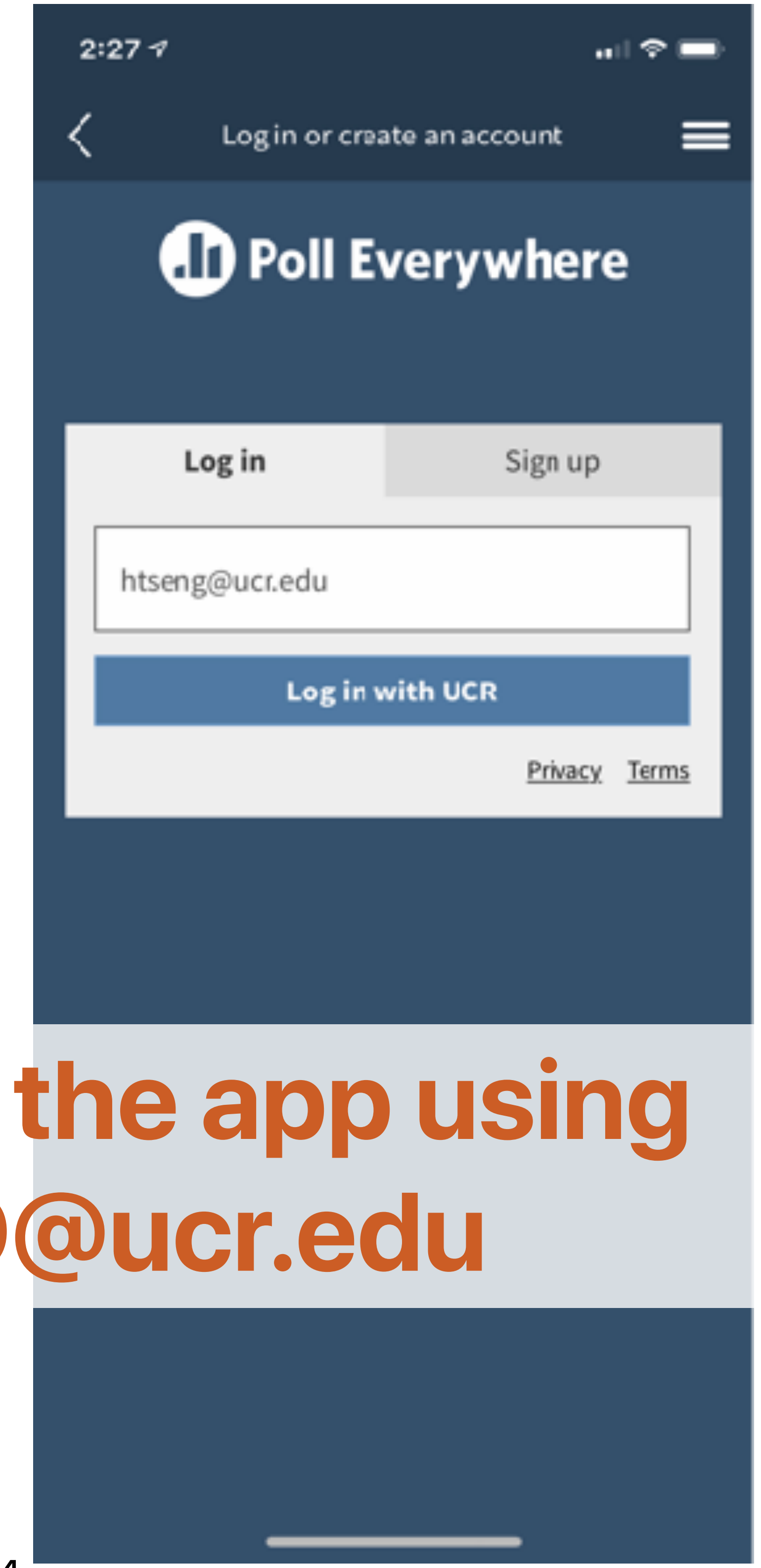
- Reading assignments from
  - Textbook: Operating Systems: Three Easy Pieces Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau
(free online http://pages.cs.wisc.edu/~remzi/OSTEP/)
  - Papers — at least get through those “focuses” listed in the schedule
- Reading quizzes:
  - On iLearn
  - Due before 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

# About the time of the Lecture — Setup polling everywhere



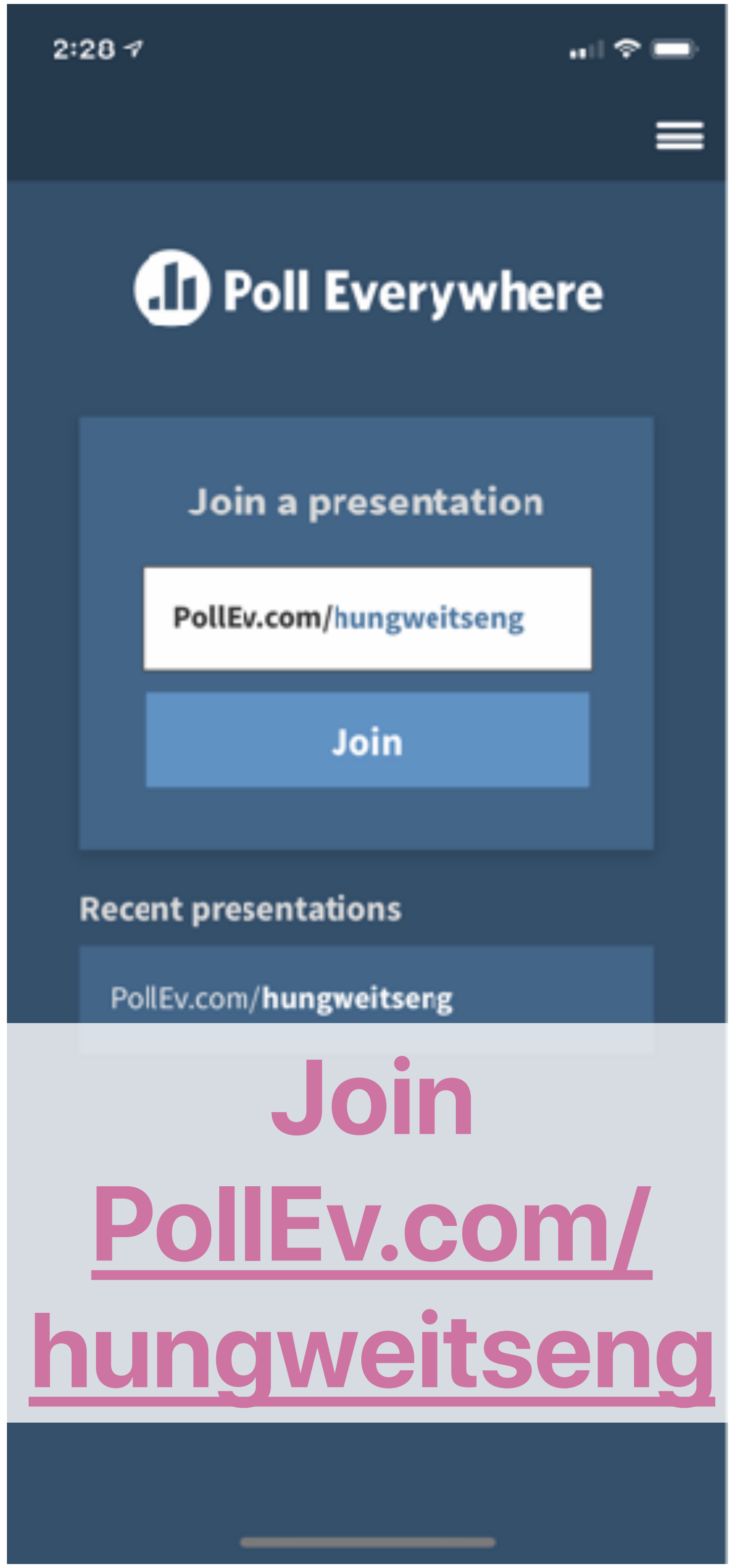
Download the “Poll Everywhere” app





htseng@ucr.edu

Login through the app using UCRNetID@ucr.edu



PollEv.com/hungweitseng

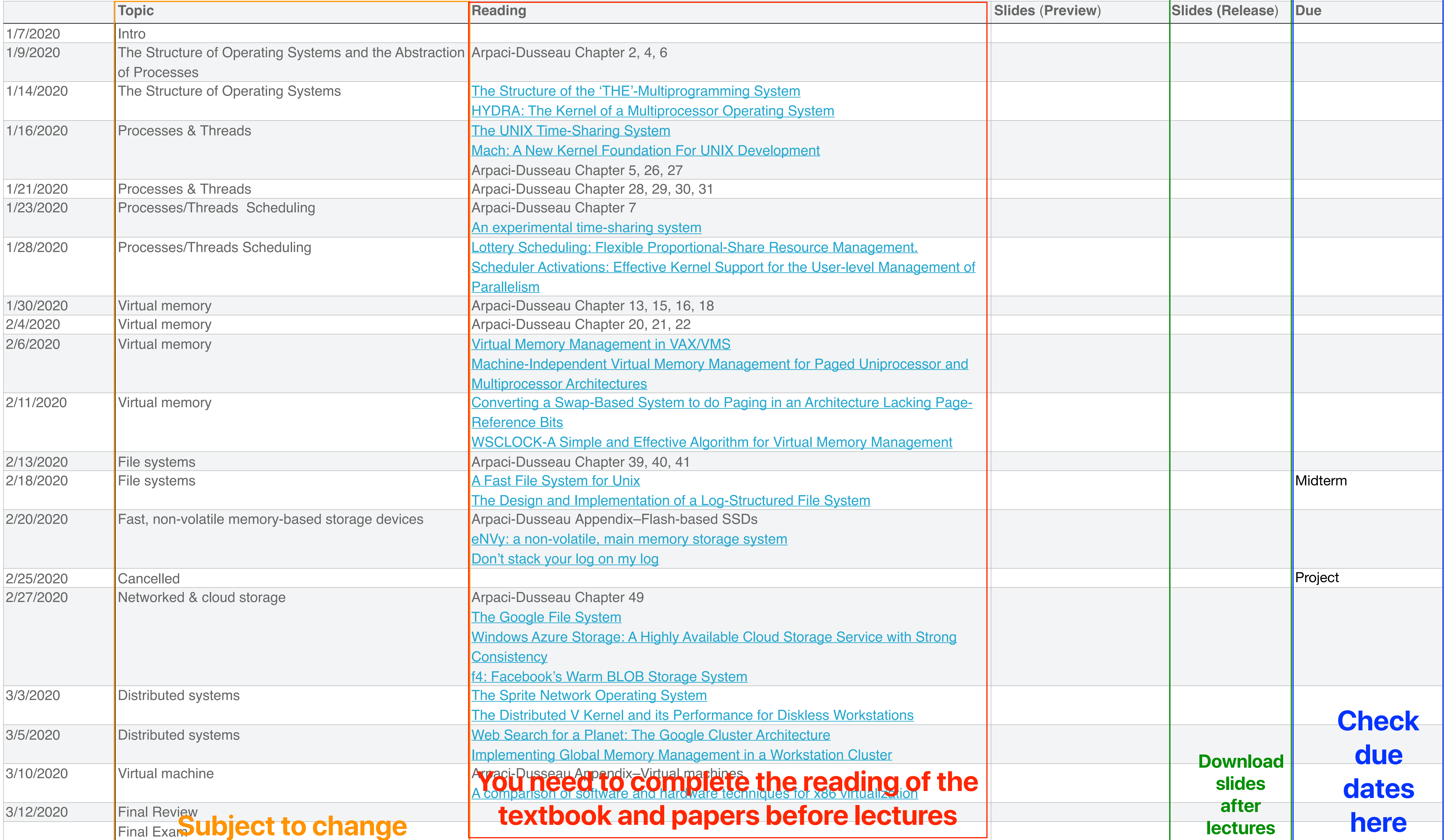
PollEv.com/hungweitseng

Join PollEv.com/hungweitseng

# During lectures: peer instruction

- 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.
- You will GET CREDITS for your efforts to learn in class
  - By answering questions with a clicker
  - Answer 80% of the clicker questions in 80% of the lectures, get 10% of your final grade

# Schedule



| | Topic | Reading | Slides (Preview) | Slides (Release) | Due |
|---|---|---|---|---|---|
| 1/7/2020 | Intro | | | | |
| 1/9/2020 | The Structure of Operating Systems and the Abstraction of Processes | Arpaci-Dusseau Chapter 2, 4, 6 | | | |
| 1/14/2020 | The Structure of Operating Systems | The Structure of the ‘THE’-Multiprogramming System<br>HYDRA: The Kernel of a Multiprocessor Operating System | | | |
| 1/16/2020 | Processes & Threads | The UNIX Time-Sharing System<br>Mach: A New Kernel Foundation For UNIX Development<br>Arpaci-Dusseau Chapter 5, 26, 27 | | | |
| 1/21/2020 | Processes & Threads | Arpaci-Dusseau Chapter 28, 29, 30, 31 | | | |
| 1/23/2020 | Processes/Threads  Scheduling | Arpaci-Dusseau Chapter 7<br>An experimental time-sharing system | | | |
| 1/28/2020 | Processes/Threads Scheduling | Lottery Scheduling: Flexible Proportional-Share Resource Management.<br>Scheduler Activations: Effective Kernel Support for the User-level Management of Parallelism | | | |
| 1/30/2020 | Virtual memory | Arpaci-Dusseau Chapter 13, 15, 16, 18 | | | |
| 2/4/2020 | Virtual memory | Arpaci-Dusseau Chapter 20, 21, 22 | | | |
| 2/6/2020 | Virtual memory | Virtual Memory Management in VAX/VMS<br>Machine-Independent Virtual Memory Management for Paged Uniprocessor and Multiprocessor Architectures | | | |
| 2/11/2020 | Virtual memory | Converting a Swap-Based System to do Paging in an Architecture Lacking Page-Reference Bits<br>WSCLOCK-A Simple and Effective Algorithm for Virtual Memory Management | | | |
| 2/13/2020 | File systems | Arpaci-Dusseau Chapter 39, 40, 41 | | | |
| 2/18/2020 | File systems | A Fast File System for Unix<br>The Design and Implementation of a Log-Structured File System | | | Midterm |
| 2/20/2020 | Fast, non-volatile memory-based storage devices | Arpaci-Dusseau Appendix–Flash-based SSDs<br>eNVy: a non-volatile, main memory storage system<br>Don’t stack your log on my log | | | |
| 2/25/2020 | Cancelled | | | | Project |
| 2/27/2020 | Networked & cloud storage | Arpaci-Dusseau Chapter 49<br>The Google File System<br>Windows Azure Storage: A Highly Available Cloud Storage Service with Strong Consistency<br>f4: Facebook’s Warm BLOB Storage System | | | |
| 3/3/2020 | Distributed systems | The Sprite Network Operating System<br>The Distributed V Kernel and its Performance for Diskless Workstations | | | |
| 3/5/2020 | Distributed systems | Web Search for a Planet: The Google Cluster Architecture<br>Implementing Global Memory Management in a Workstation Cluster | | | |
| 3/10/2020 | Virtual machine | Arpaci-Dusseau Appendix–Virtual machines<br>A comparison of software and hardware techniques for x86 virtualization | | | |
| 3/12/2020 | Final Review | | | | |
| | 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 23 this quarter!

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

# Why papers?

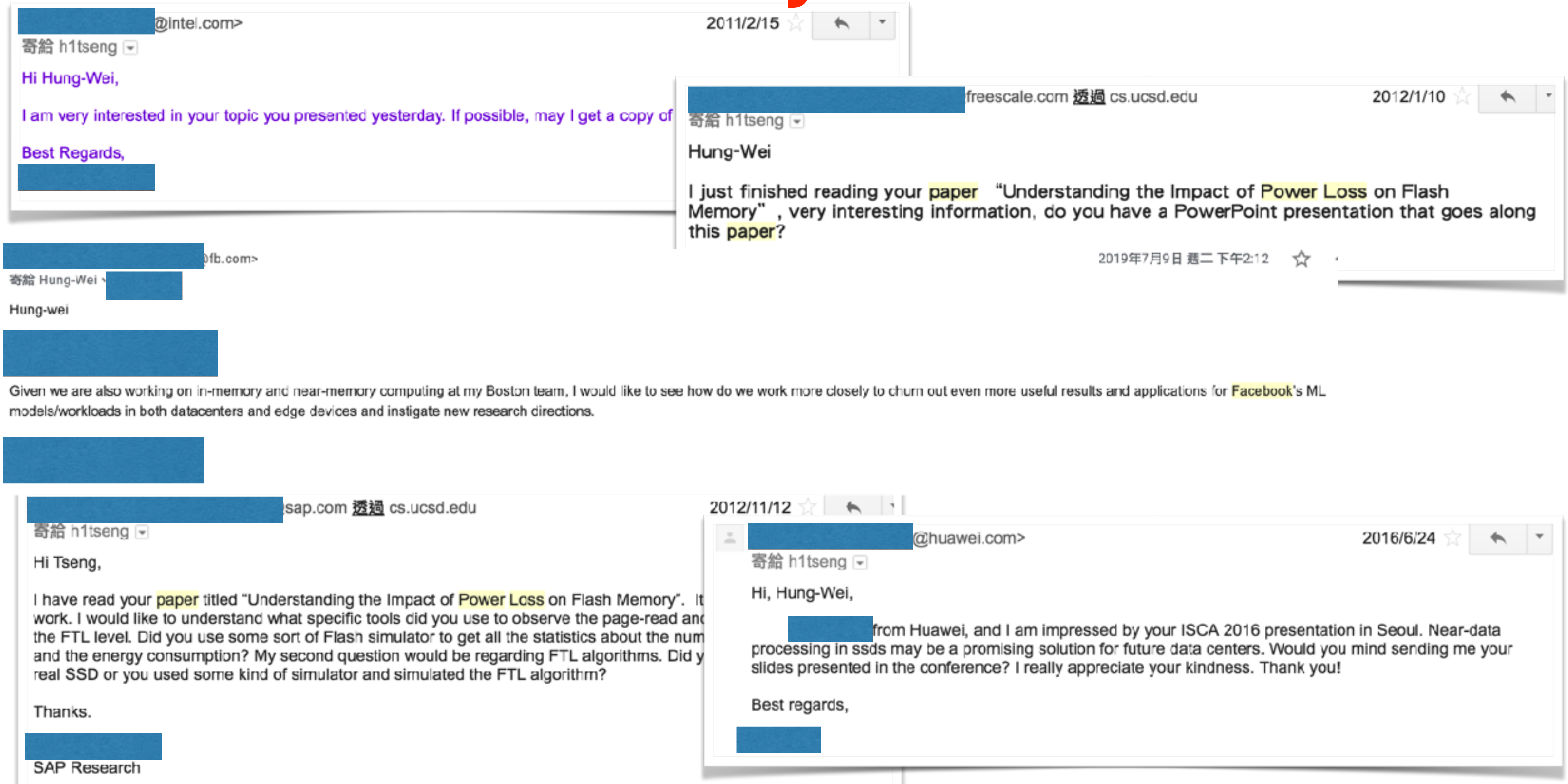
# No alternative facts

- 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,

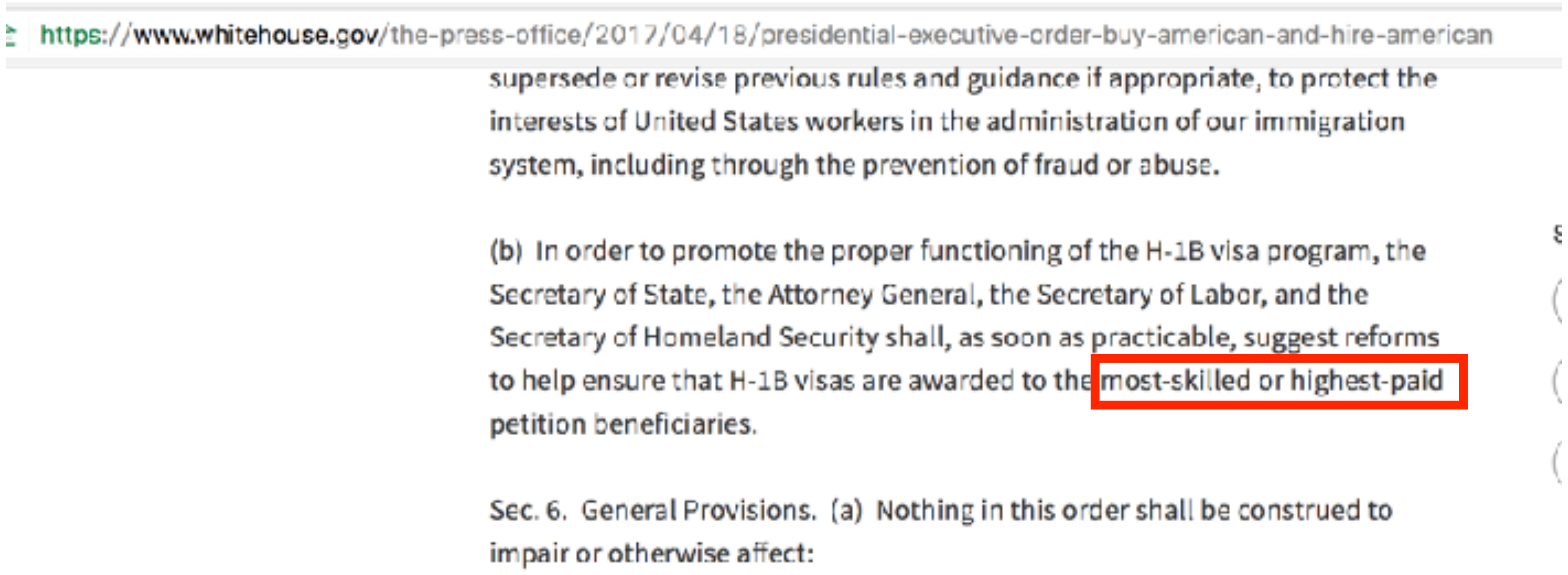
[redacted]from Huawei, and I am impressed by your ISCA 2016 presentation in Seoul. Near-data processing in ssds may be a promising solution for future data centers. Would you mind sending me your slides presented in the conference? I really appreciate your kindness. Thank you!

Best regards,

[redacted]

# 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 CSC 501-001, Fall 2018.**

2017 Spring @ NC State

2017 Fall @ NC State

2019 Summer II @ UCSD

2018 Fall @ NC State

2016 Fall @ NC State

2018 Spring @ NC State

You

2020 Winter @ UCR

2019 Spring @ NC Sta

r @ UCSD

2019 Fall @ UCR

2016 Spring @ UCSD

2019 Summer I @ UCSD

2012 Summer @ UCSD

2014 Summer @ UCSD

# Announcement

- The first reading quiz due this Thursday before class!
- Please have your "Poll everywhere" app ready and login before the next lecture