UC RIVERSIDE
UNIVERSITY OF CALIFORNIA

Advanced Operating Systems (CS 202)

OS Evolution and Organization

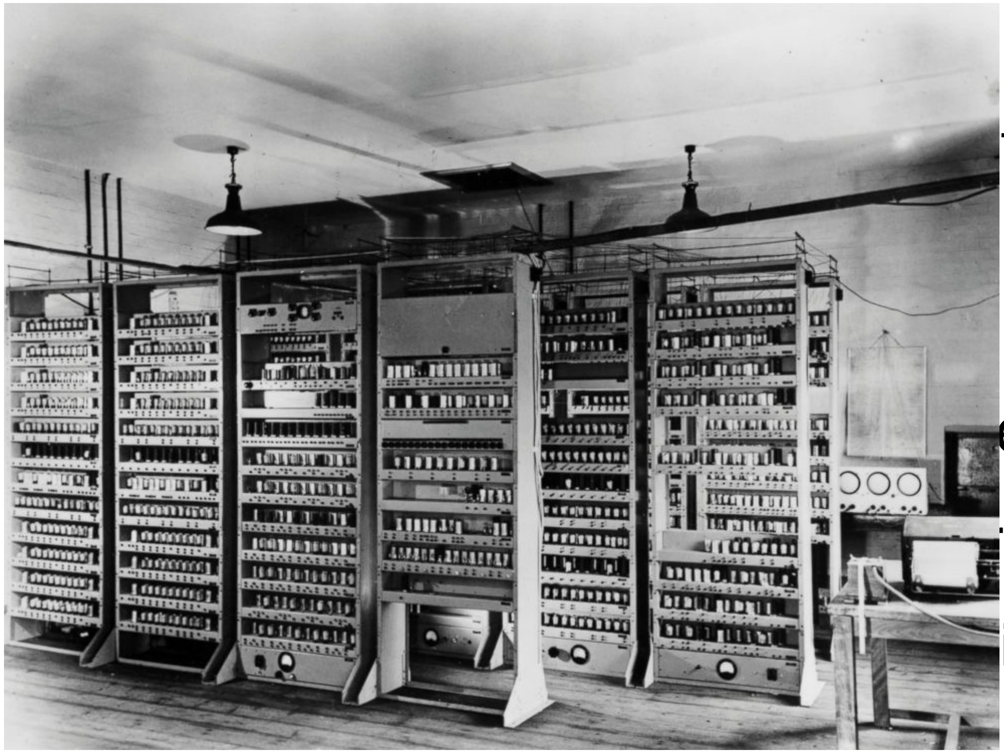
Operating Systems Models

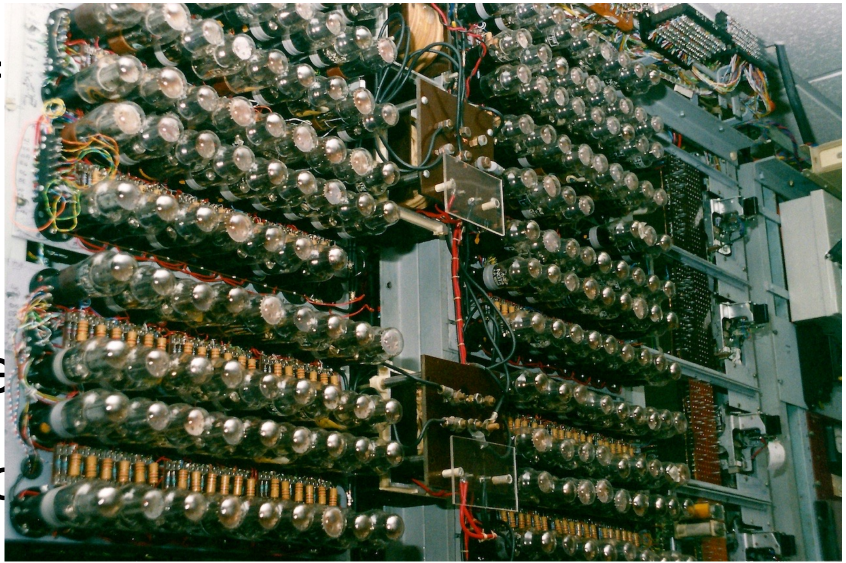
- Our first topic will explore OS models
  - Why do Operating Systems look the way they do?
  - What drives the decisions? What else is possible?

- To set the table
  - Today, we will do a walk through historical evolution of OS
  - Next (next time), we will overview traditional/modern OS organization (monolithic kernel)

Dawn of computing

- Pre 1950 : the very first electronic computers





f

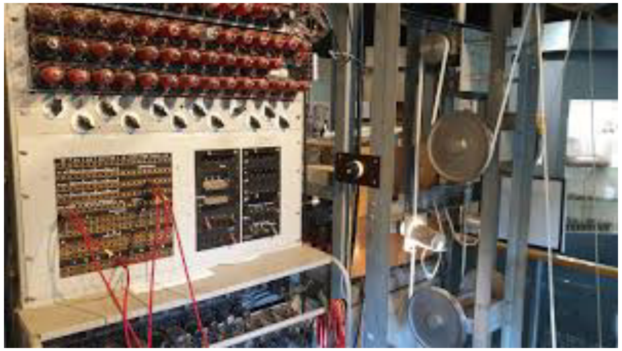
e

rc

ice Wilkes receives Turing

Phase 0 of OS Evolution (40s to 1955)

- No OS
  - Computers are exotic, expensive, large, slow experimental equipment
  - Program in machine language and using plugboards
  - User sits at console: no overlap between computation, I/O, user thinking, etc..
    - Program manually by plugging wires in
    - Goal: number crunching for missile computations
- Imagine programming that way
  - Painful and slow



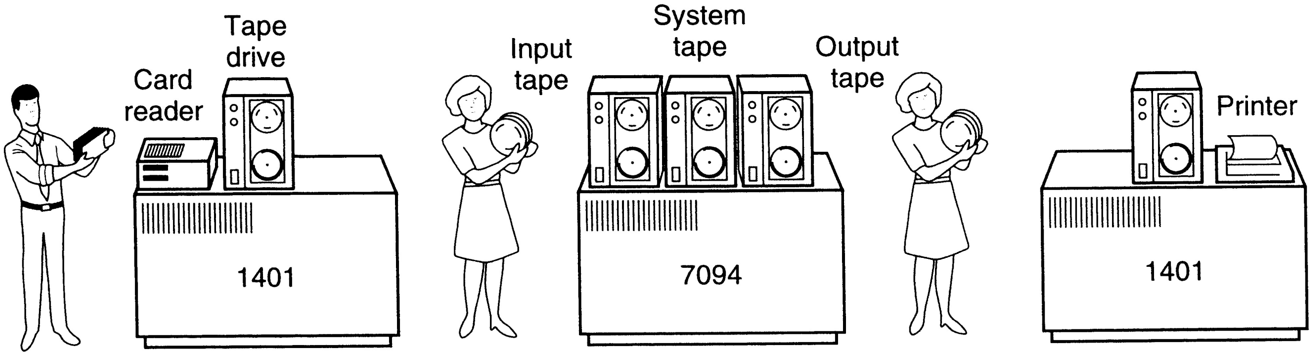
OS progress in this period

- Libraries of routines that are common
  - Including those to talk to I/O devices
  - Punch cards (enabling copying/exchange of these libraries) a big advance!
  - Pre-cursor to OS

Phase 1: 1955-1970

- Computers expensive; people cheap
  - Use computers efficiently – move people away from machine
  - OS becomes a batch monitor
    - Loads a job, runs it, then moves on to next
    - If a program fails, OS records memory contents somewhere
    - More efficient use of hardware but increasingly difficult to debug

- Batch systems on *mainframe* computers
- collections of jobs made up into a *batch*
- example: IBM 1401/7094
  - card decks spooled onto magnetic tape and from tape to printer



- example: English Electric Leo KDF9
  - 32K 48-bit words, 2μsec cycle time
  - punched paper-tape input 'walk-up' service or spooling via mag tape

IBM 7094, thought to be first computer singing (1961)
https://youtu.be/yIwhx3NQSLg

Advances in technology in this stage

- Data channels and interrupts
  - Allow overlap of I/O and computing
  - Buffering and interrupt handling done by OS
  - Spool (buffer) jobs onto “high speed” drums

Phase 1, problems

- Utilization is low (one job at a time)
- No protection between jobs
- Short jobs wait behind long jobs
  - So, we can only run one job at a time
- Coordinating concurrent activities
- Still painful and slow (but less so?)

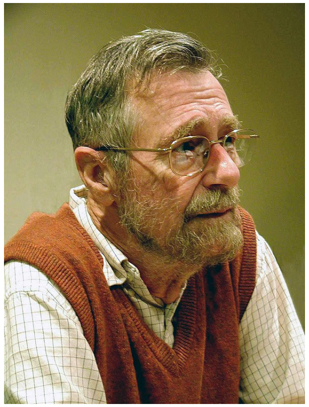
Advances in OS in this period

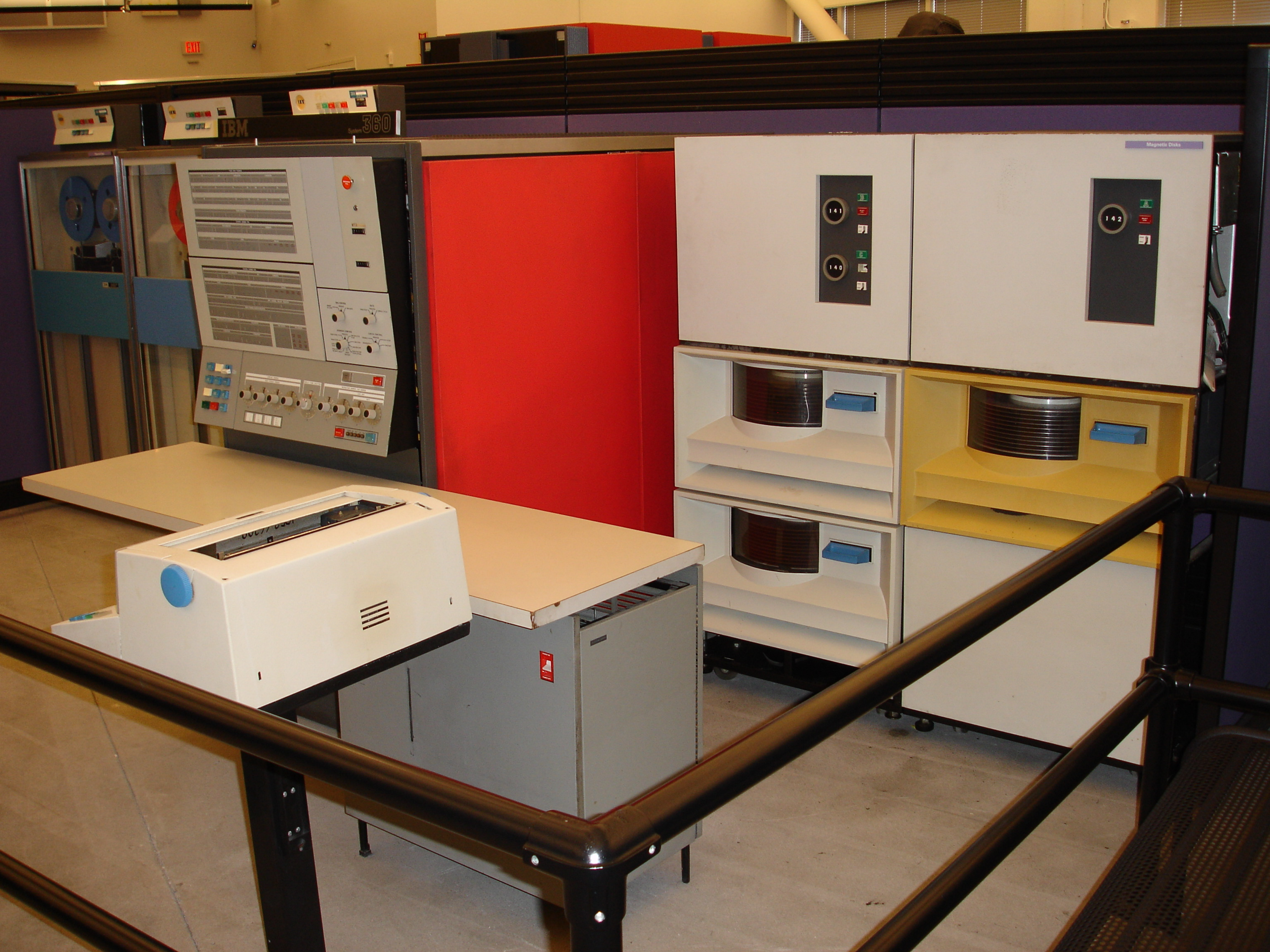
- Hardware provided memory support (protection and relocation)
- Multiprogramming (not to be confused with time sharing)
- Scheduling: let short jobs run first
- OS must manage interactions between concurrent things
  - Starts emerging as a field/science
- OS/360 from IBM first OS designed to run on a family of machines from small to large

Some important projects

- Atlas computer/OS from Manchester U. (late 50s/early 60s)
  - First recognizable OS
  - Separate address space for kernel
  - Early virtual memory

- THE Multiprogramming system (early 60s)
  - Introduced semaphores
  - Attempt at proving systems correct; interesting software engineering insights
  - Project lead by Dijkstra (Turing award winner)




EXIT
IBM
System 360
Magnetic Disks
141
140
142

Phase 2: 1970s

- Computers and people are expensive
  - Help people be more productive
  - Interactive time sharing: let many people use the same machine at the same time
  - Emergence of minicomputers
    - Terminals are cheap
  - Keep data online on fancy file systems
  - Attempt to provide reasonable response times (Avoid thrashing)

Important advances and systems

- Compatible Time-Sharing System (CTSS)
  - MIT project (demonstrated in 1961)
  - One of the first time sharing systems
  - Corbato won Turing award in 1990
  - Pioneered much of the work in scheduling
  - Motivated MULTICS

MULTICS

- Jointly developed by MIT, Bell Labs and GE
- Envisioned one main computer to support everyone
  - People use computing like a utility like electricity – sound familiar? Ideas get recycled
- Many many fundamental ideas: protection rings, hierarchical file systems, devices as files, …
- Building it was more difficult than expected
- Technology caught up

Unix appears

- Ken Thompson, who worked on MULTICS, wanted to use an old PDP-7 laying around in Bell labs
- He and Dennis Ritchie built a system designed by programmers for programmers
- Originally in assembly. Rewritten in C
  - If you notice for the paper, they are defending this decision
  - However, this is a new and important advance: portable operating systems!
- Shared code with everyone (particularly universities)

Ken Thompson

Dennis M. Ritchie

1983 Turing Award for unix

Unix (cont’d)

- Berkeley added support for virtual memory for the VAX
- DARPA selected Unix as its networking platform in arpanet
- Unix became commercial
  - …which eventually lead Linus Torvald to develop Linux

Some important ideas in Unix

- OS written in a high level language
- OS portable across hardware platforms
  - Computing is no longer a pipe stove/vertical system
- Pipes
  - E.g., grep foo file.txt | wc -l
- Mountable file systems
- Many more (we'll talk about unix later)

Phase 3: 1980s

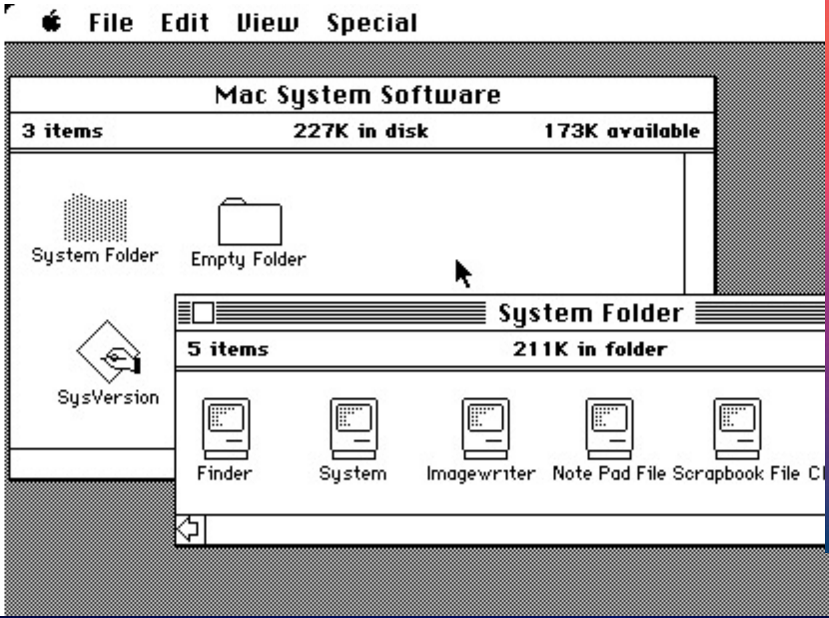
- Computers are cheap, people expensive
  - Put a computer in each terminal
  - CP/M from DEC first personal computer OS (for 8080/85) processors
  - IBM needed software for their PCs, but CP/M was behind schedule
  - Approached Bill Gates to see if he can build one
  - Gates approached Seattle computer products, bought 86-DOS and created MS-DOS
  - Goal: finish quickly and run existing CP/M software
  - OS becomes subroutine library and command executive

Disk Operating System (DOS)

- 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

New advances in OS

- PC OS was a regression for OS
  - Stepped back to primitive phase 1 style OS leaving the cool developments that occurred in phase 2
- Academia was still active, and some developments still occurred in mainframe and workstation space
- Eventually, Windows, Linux, MacOS, took over
  - Phase 2 OS' making it to PCs
  - GUIs!

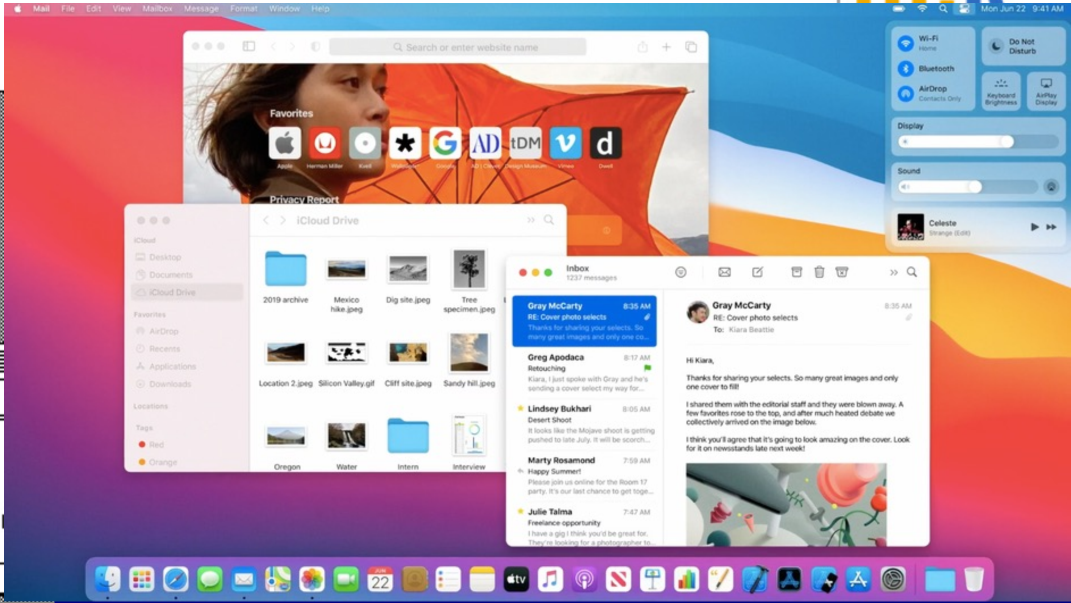


Mac System Software

3 items 227K in disk 173K available

System Folder

5 items 211K in folder



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...

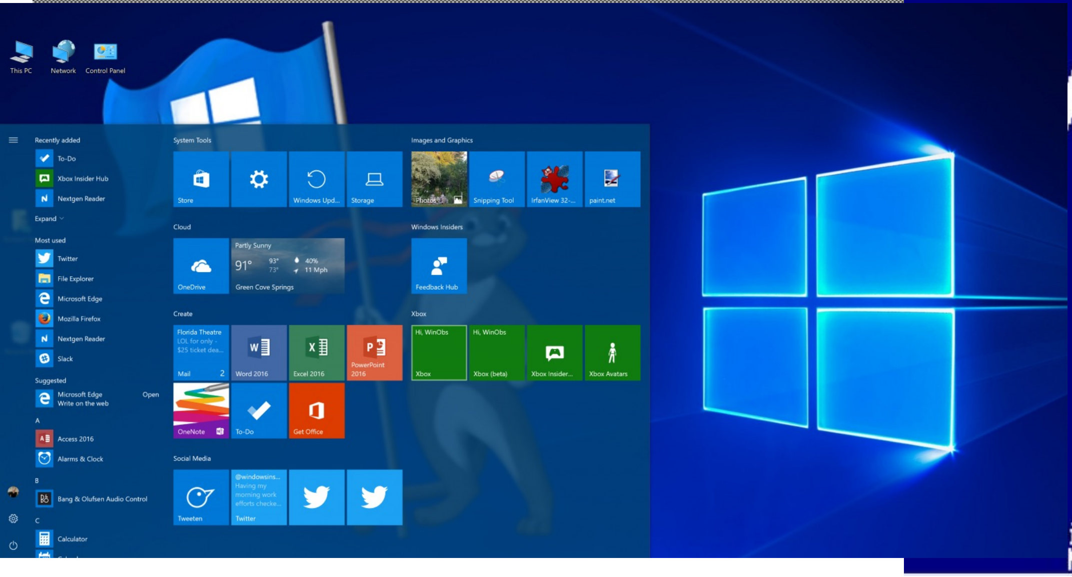
Gray McCarty 8:35 AM
RE: Cover photo selects
To: Kiara Beattie

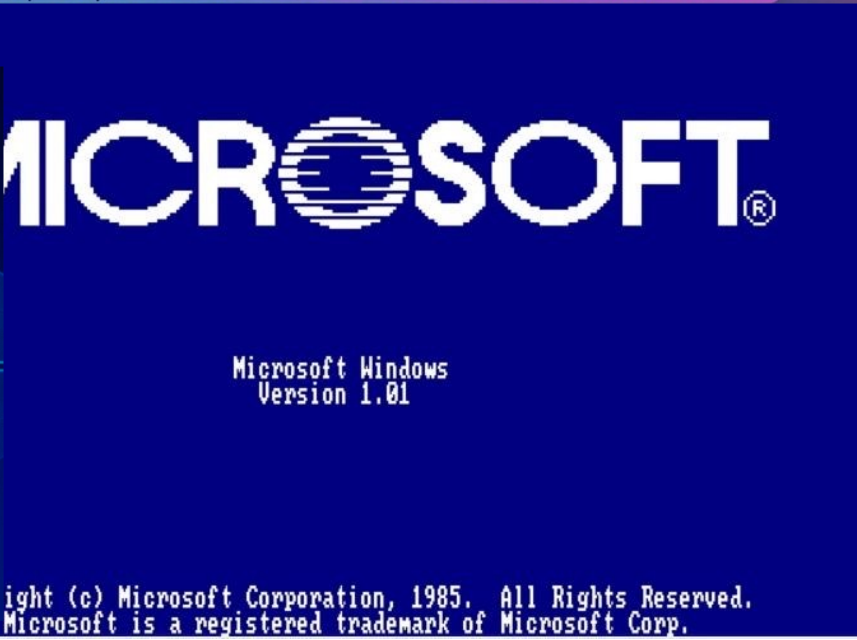
Hi Kiara,

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!





Microsoft Windows
Version 1.01

ight (c) Microsoft Corporation, 1985. All Rights Reserved.
Microsoft is a registered trademark of Microsoft Corp.

Phase 4: Networked systems 1990s to 2010s

- Machines can talk to each other
  - its all about connectivity
- We want to share data not hardware
- Networked applications drive everything
  - Web, email, messaging, social networks, …
- Protection and multiprogramming less important for personal machines
  - But more important for servers

Phase 4, continued

- Market place continued horizontal stratification
  - ISPs (service between OS and applications)
  - Information is a commodity
  - Advertising a new marketplace

- New network based architectures
  - Client server
  - Clusters
  - Grids
  - Distributed operating systems
  - Cloud computing (or is that phase 5?)

New problems

- Large scale
  - Google file system, mapreduce, …
- Concurrency at large scale
  - ACID (Atomicity, Consistency, Isolation and Durability) in Internet Scale systems
    - Very large delays
    - Partitioning
- Security and Privacy

Phase 5: 2010s -- ??

- New generation?
- Mobile devices that are powerful
- Sensing: location, motion, …
- Cyberphysical systems
- Machine learning everywhere
- Computing evolving beyond networked systems
  - But OS for them looks largely the same
  - Is that a good idea?