

CS 202

Advanced Operating Systems

Winter 26

Lecture 2: Historical Perspective

Instructor: Chengyu Song

What is an OS?

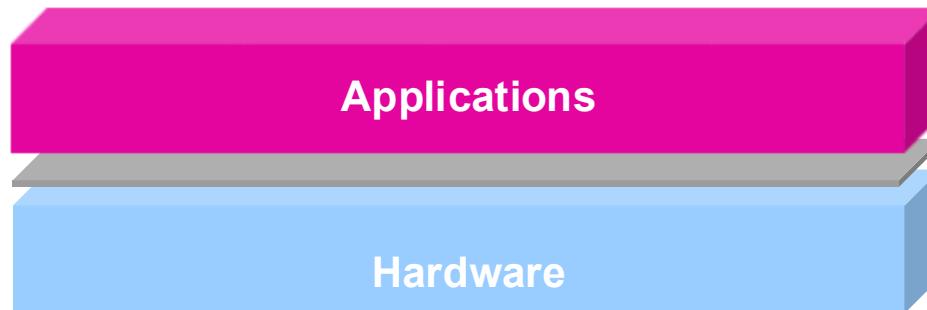
- Operating System
 - ◆ (of a person) control the functioning of (a machine, process, or system)
"a shortage of workers to operate new machines"
 - ◆ a set of things working together as parts of a mechanism or an interconnecting network
"the state railroad system"
 - ◆ a set of principles or procedures according to which something is done
"a multiparty system of government"
- The software that supports a computer's basic functions, such as scheduling tasks, executing applications, and controlling peripherals.

Topics for today

- What is an operating system?
- Why do we need operating systems?
- What does an operating system need to do?
- Looking back, looking forward.

Why having an OS?

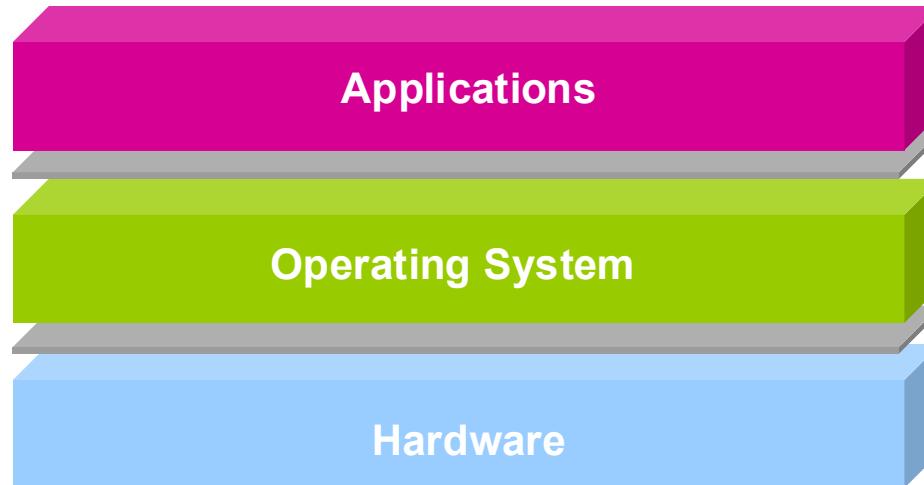
- What if applications ran directly on hardware?



- Problems:
 - ◆ Portability → OS Task 1: abstraction
 - ◆ Resource sharing → OS Task 2: multiplexing
 - ◆ Resource management → OS Task 3: scheduling

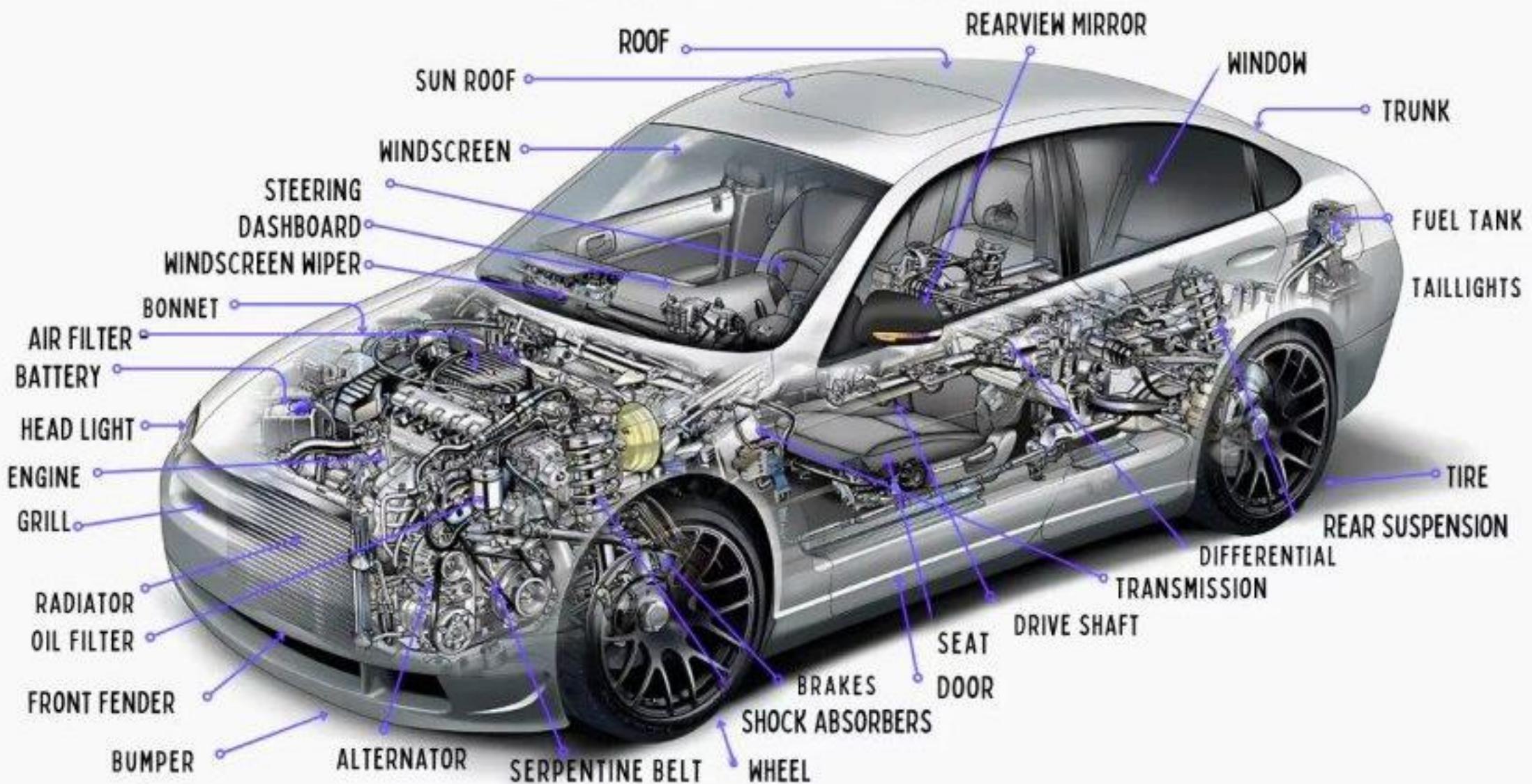
Why having an OS?

- The operating system is the software layer between user applications and the hardware (resources)



The OS is “all the code that you
didn’t have to write” to finish your
tasks





Topics for today

- What is an operating system?
- Why do we need operating systems?
- What does an operating system need to do?
- Looking back, looking forward.

Fundamental Issues

- The fundamental issues/questions in this course are:
 - ◆ **Functionality**: how to enable the users/programs to do more?
 - ◆ **Performance**: how to do better?
 - ◆ **Management**: how to allocate and schedule resources for tasks?
 - ◆ **Protections**: how to make sure things won't go wrong?
 - ◆ **Communication**: how to enable collaboration?
 - ◆ **Security**: how to create a safe environment?
 - ◆ **Reliability and fault tolerance**: how to mask failures?

Basic Roles of an OS

- **Abstraction:** defines a set of logical resources (**objects**) and well-defined operations on them (**interfaces**)
- **Virtualization:** isolates and multiplexes physical resources via spatial and temporal **sharing**
- **Control:** who, when, how
 - ◆ Scheduling (when): efficiency and fairness
 - ◆ Permissions (how): security and privacy
- **Persistence:** how to keep and share data

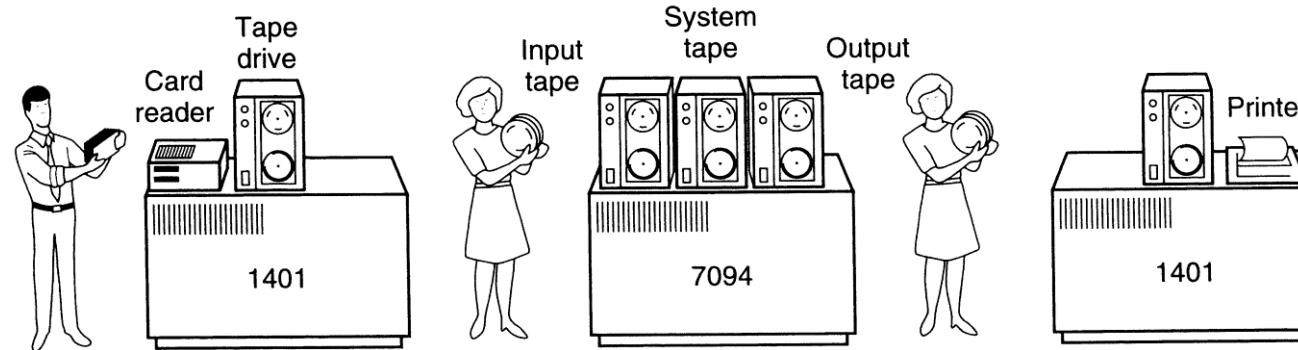
Topics for today

- What is an operating system?
- Why do we need operating systems?
- What does an operating system need to do?
- Looking back, looking forward.

Phase 0

- In the beginning, OS is just runtime libraries (routines)
 - ◆ A piece of code used/sharable by many programs
 - ◆ Abstraction: reuse magic to talk to physical devices
 - ◆ Avoid bugs
- User scheduled an exclusive time where they would use the machine
- User interface was switches and lights, eventually punched tape and cards
 - ◆ An interesting side effect: less bugs

Phase 1: batch systems (1955-1970)



- Computers expensive; people cheap
 - ◆ Use computers efficiently – move people away from machine
- OS in this period became a program loader
 - ◆ Loads a job, runs it, outputs result, then moves on to next
 - ◆ More efficient use of hardware but increasingly difficult to debug
 - » Still less bugs ☺

Advances in OS in this period

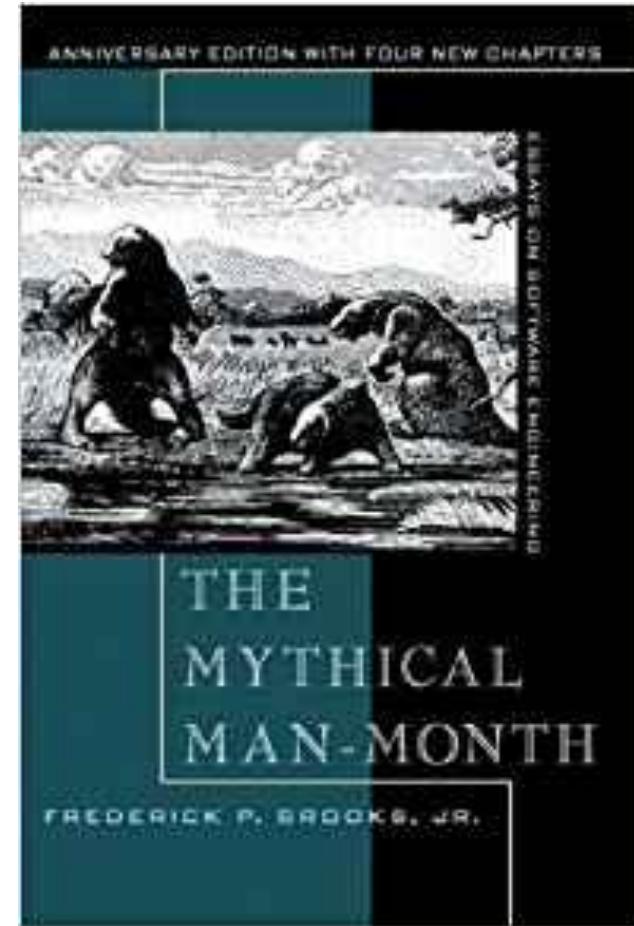
- SPOOLING/Multiprogramming
 - ◆ Simultaneous Peripheral Operations On-Line (SPOOL)
 - » Non-blocking tasks
 - » Copy document to printer buffer so printer can work while CPU moves on to something else
 - ◆ Hardware provided memory support (protection and relocation)
 - ◆ Scheduling
 - ◆ OS must manage interactions between concurrent things
- OS/360 from IBM first OS designed to run on a family of machines from small to large

Multiprogramming

- Discussion: infrastructure to support multiprogramming (THE/nucleus)
 - ◆ Execution
 - ◆ Storage
 - ◆ Synchronization
 - ◆ Correctness (provable termination)
 - ◆ Collaboration / communication

Phase 1, problems

- Utilization is low (one job at a time)
- No protection between jobs
 - ◆ But one job at a time, so what can go wrong?
- Scheduling
- Coordinating concurrent activities
- People time is still being wasted
- Operating Systems didn't really work
 - ◆ The mythical man month
 - ◆ Birth of software engineering



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
- Persistence: keep data online on fancy file systems

Unix appears

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

Unix (cont'd)

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

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

Phase 4: 1990s to now?

- Its all about connectivity
- Enables parallelism but performance is not goal
- Goal is communication/sharing/power consumption/...
 - ◆ Requires high speed communication
 - ◆ We want to share data not hardware
- Networked applications drive everything
 - ◆ Web, email, messaging, social networks, ...
 - ◆ Chromebook

New problems

- Large scale
 - ◆ Google file system, map-reduce, ...
- Parallelism on the desktop (multicores)
- Heterogeneous systems, IoT
 - ◆ GPU, FPGA, ...
 - ◆ Real-time; energy efficiency
- Security and Privacy

Phase 5

- New generation?
- Computing evolving beyond networked systems
 - ◆ Cloud computing, edge computing, IoT, wearable devices, drones, cyber-physical systems, autonomous cars, computing everywhere
 - ◆ But what is it?
 - ◆ ... and what problems will it bring?

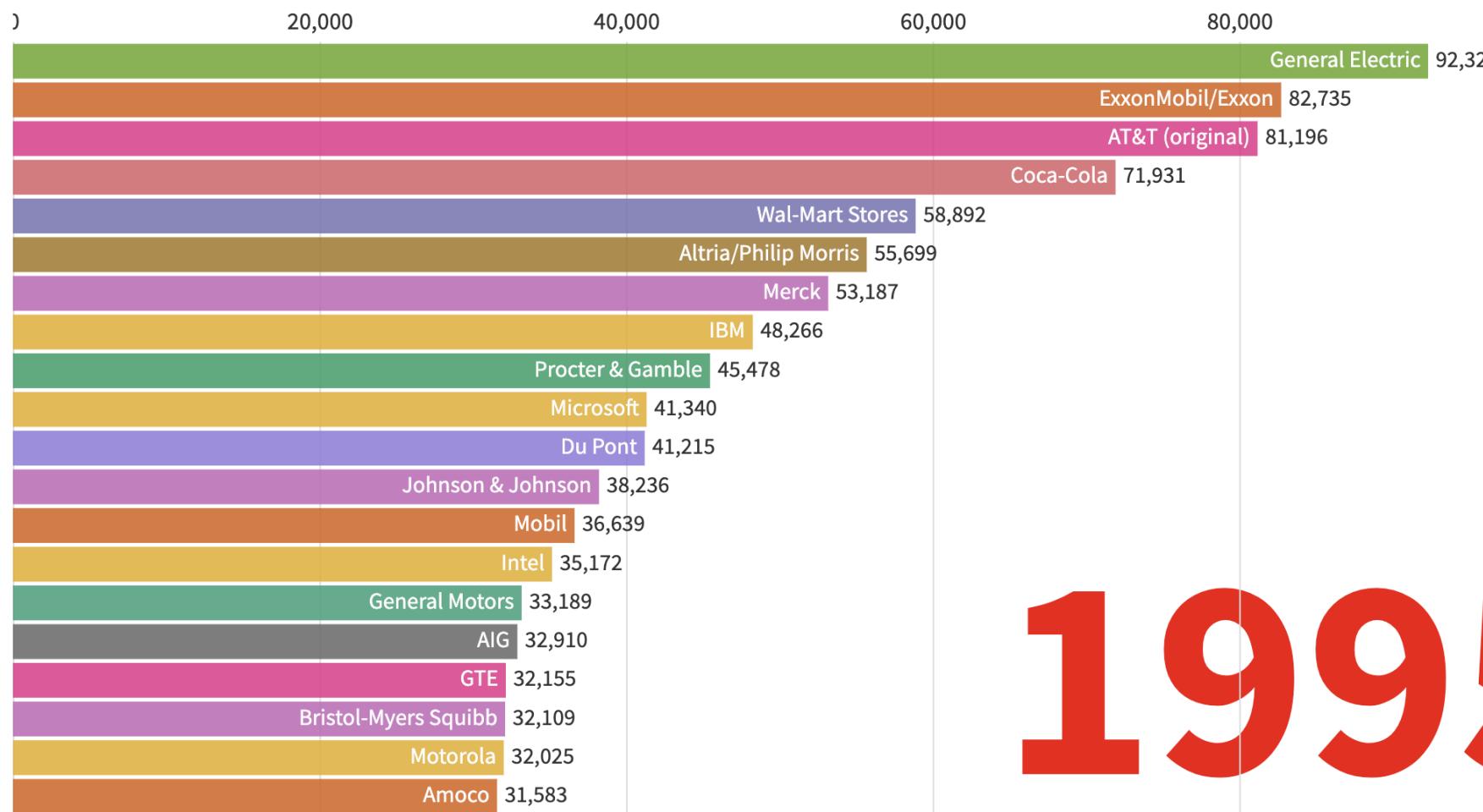
Discussion

- What are the general driving forces behind the evolution of OS?
- What are the impacts of a new (type) of OS?
 - ◆ PC (Windows, Mac)
 - ◆ Web (browsers)
 - ◆ Mobile (iOS, Android)
 - ◆ Cloud (AWS, Azure)
 - ◆ AI?

20 MOST VALUABLE FORTUNE 500 COMPANIES 1995-2020

Market Capitalization in March of each year in Millions of Dollars

Autos Energy Retail Communications Diversified Technology Tobacco Finance Chemicals Household Products
 Food & Beverage Aerospace Transportation Health Photography Metals Paper Media Machinery Foodservice Hotels
 Materials



1995

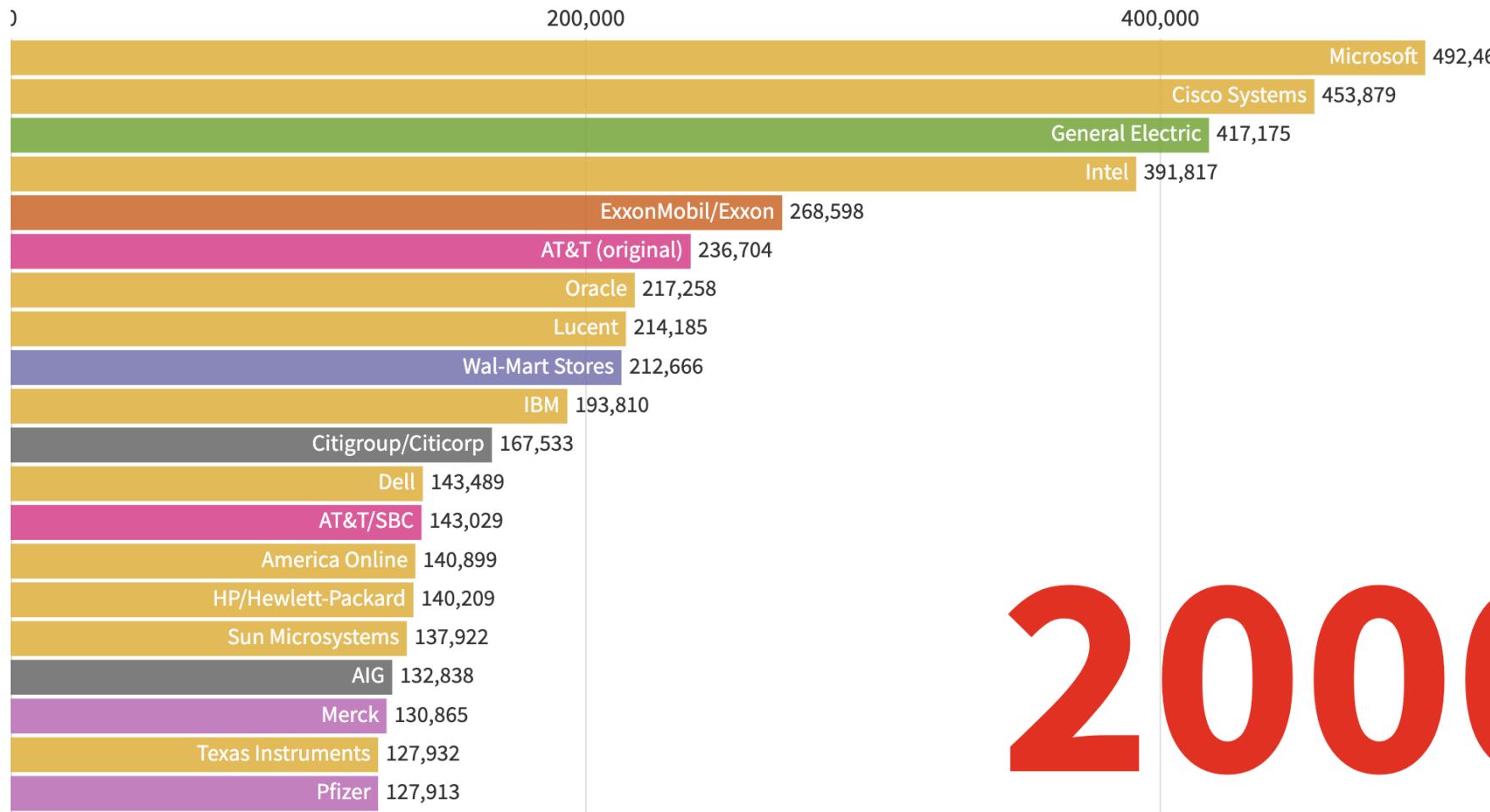


1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019

20 MOST VALUABLE FORTUNE 500 COMPANIES 1995-2020

Market Capitalization in March of each year in Millions of Dollars

Autos Energy Retail Communications Diversified Technology Tobacco Finance Chemicals Household Products
 Food & Beverage Aerospace Transportation Health Photography Metals Paper Media Machinery Foodservice Hotels
 Materials

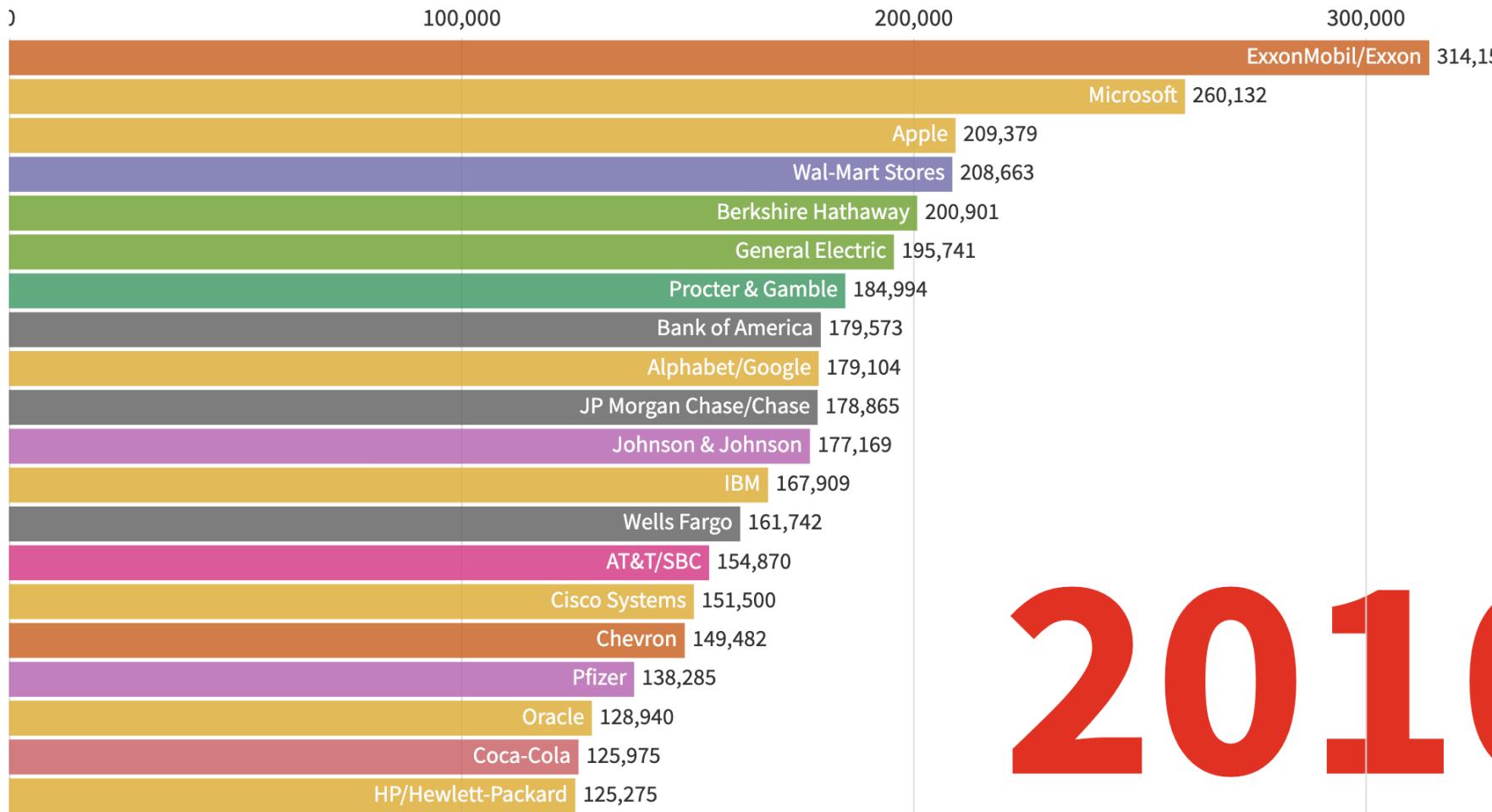


1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019

20 MOST VALUABLE FORTUNE 500 COMPANIES 1995-2020

Market Capitalization in March of each year in Millions of Dollars

Autos Energy Retail Communications Diversified Technology Tobacco Finance Chemicals Household Products
 Food & Beverage Aerospace Transportation Health Photography Metals Paper Media Machinery Foodservice Hotels
 Materials



2010

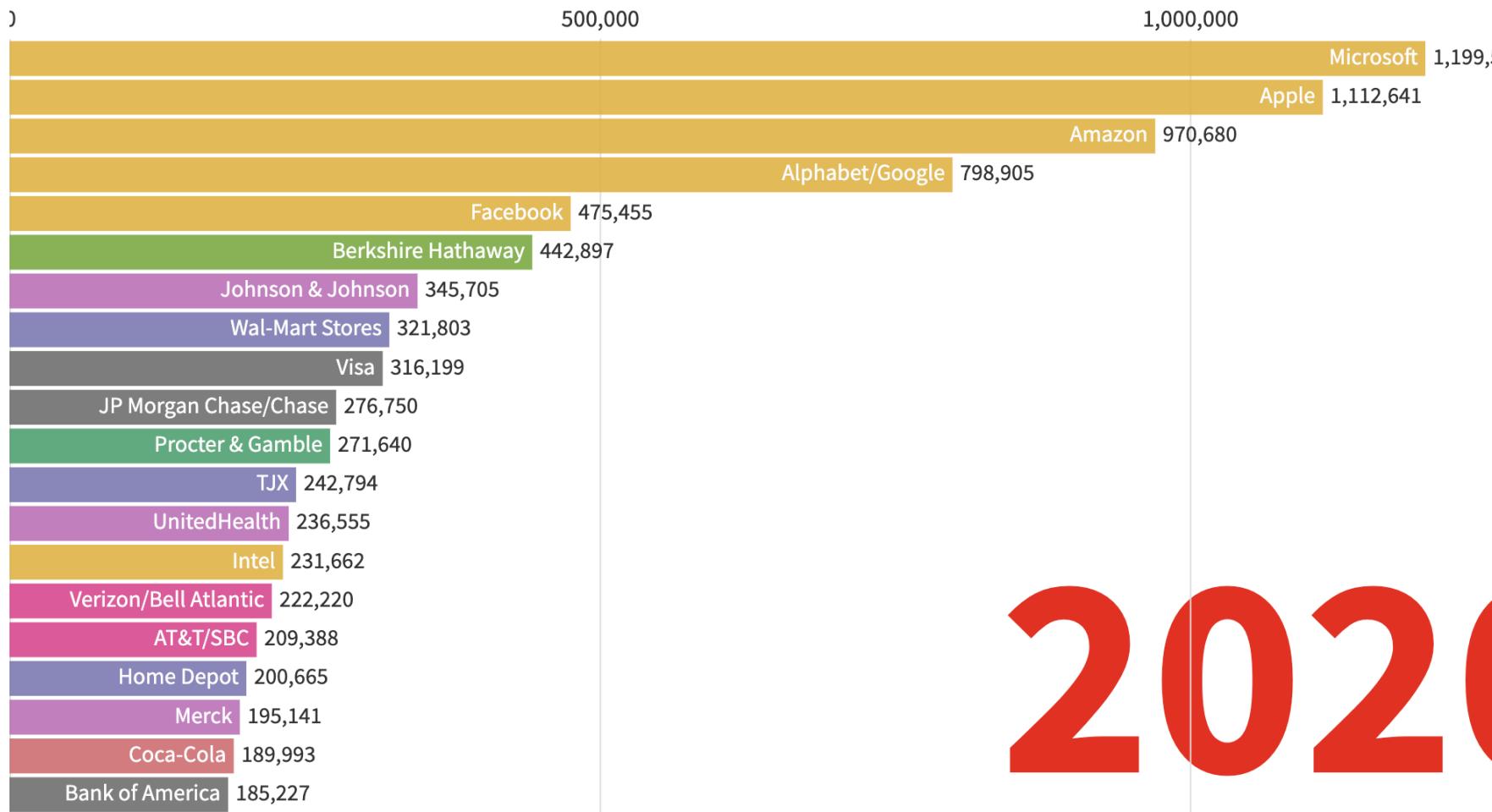


1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019

20 MOST VALUABLE FORTUNE 500 COMPANIES 1995-2020

Market Capitalization in March of each year in Millions of Dollars

Autos Energy Retail Communications Diversified Technology Tobacco Finance Chemicals Household Products
 Food & Beverage Aerospace Transportation Health Photography Metals Paper Media Machinery Foodservice Hotels
 Materials



2020



1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019