

# Introduction to Operating Systems

Jo, Heeseung

# Today's topics

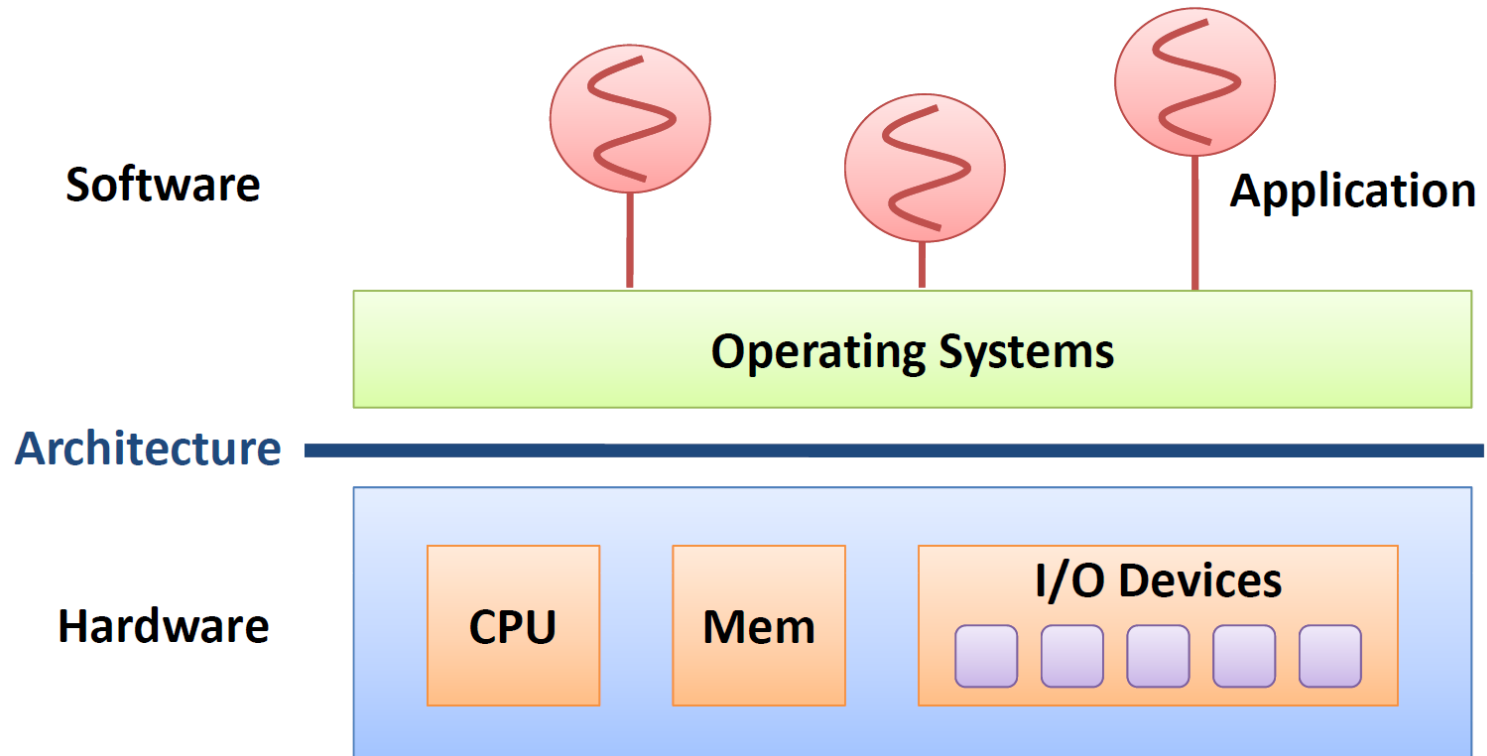
---

What is OS?

History of OS

# Operating system?

Computer systems internals



# Why do we learn OS?

---

To graduate?

To make a better OS or system

- Functionality
- Performance/cost
- Reliability
- Energy efficiency

To make a new hardware up and running

To design OS-aware hardware

To understand computer systems better

Just for fun

# What is OS? (1)

---

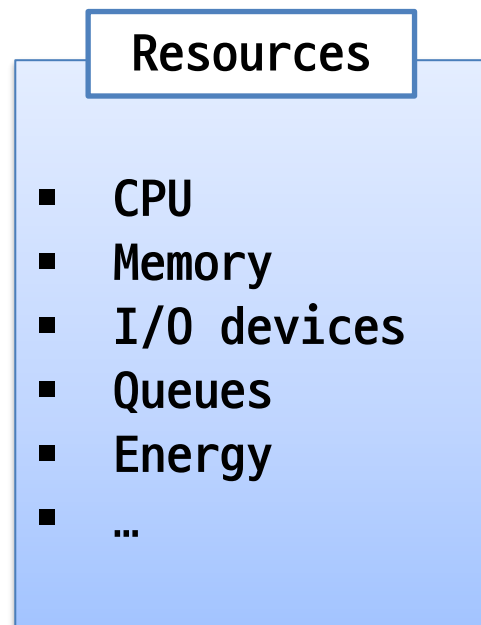
## Application view

- Provides an execution environment for running programs
- Provides an abstract view of the underlying computer system
  - Processors → Processes, Threads
  - Memory → Address spaces (virtual memory)
  - Storage → Volumes, Directories, Files
  - I/O Devices → Files (ioctl)
  - Networks → Files (sockets, pipes, ...)
  - ...

# What is OS? (2)

## System view

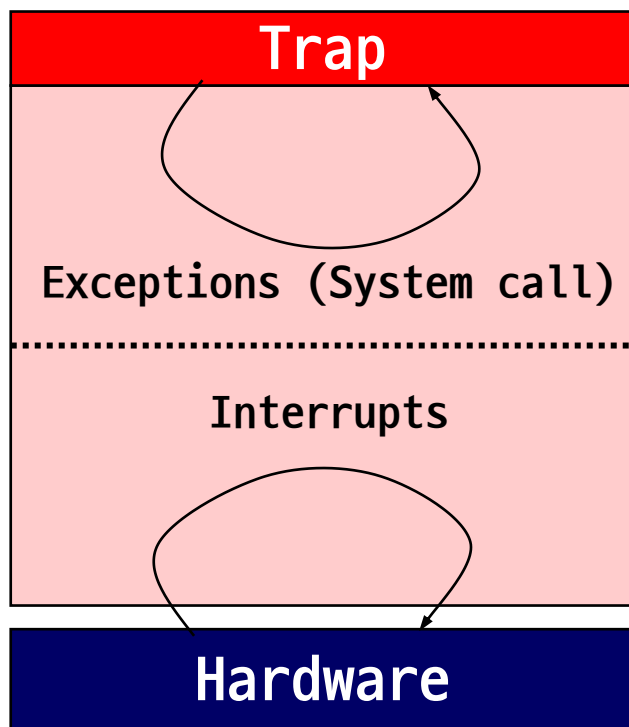
- **Manages various resources** of a computer system
- Sharing
- Protection
- Fairness
- Efficiency
- ...



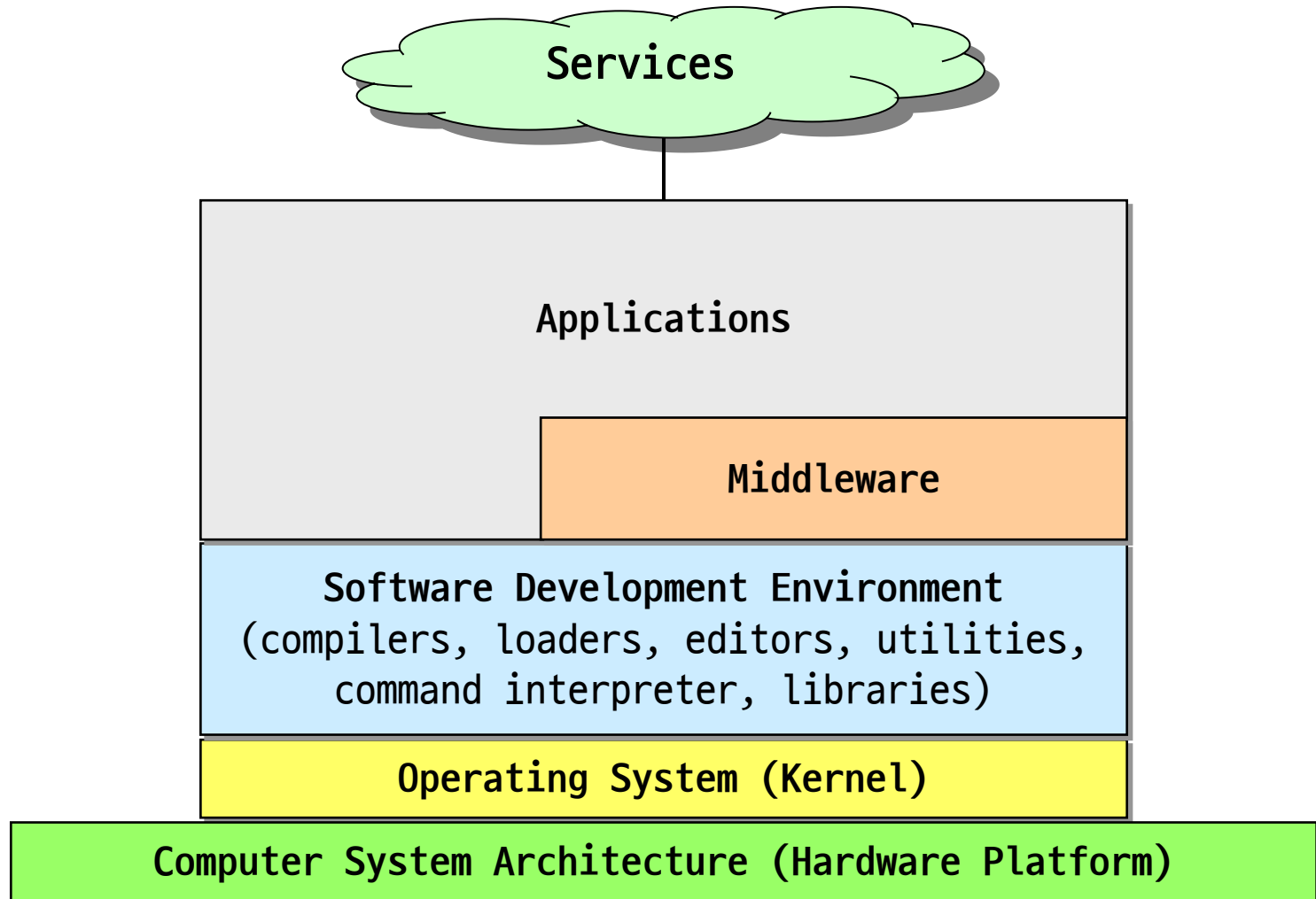
# What is OS? (3)

## Implementation view

- Highly-concurrent, event-driven software



# Computer systems

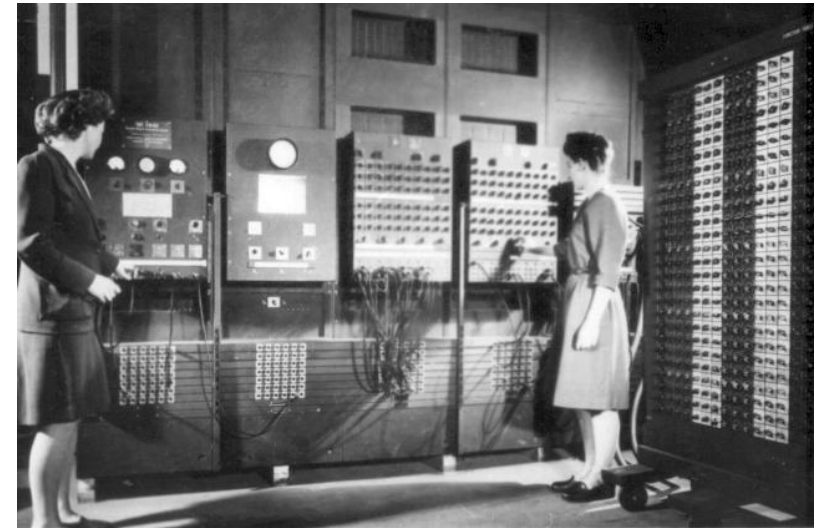
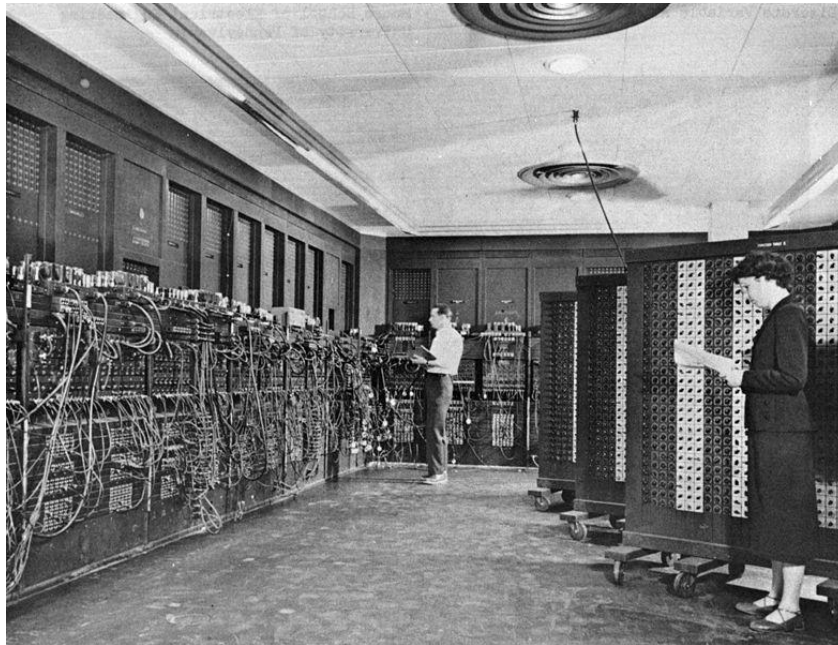


# 1st Generation (1945-55) - Vacuum tubes and plugboards

No OS

No programming languages

No assembly languages

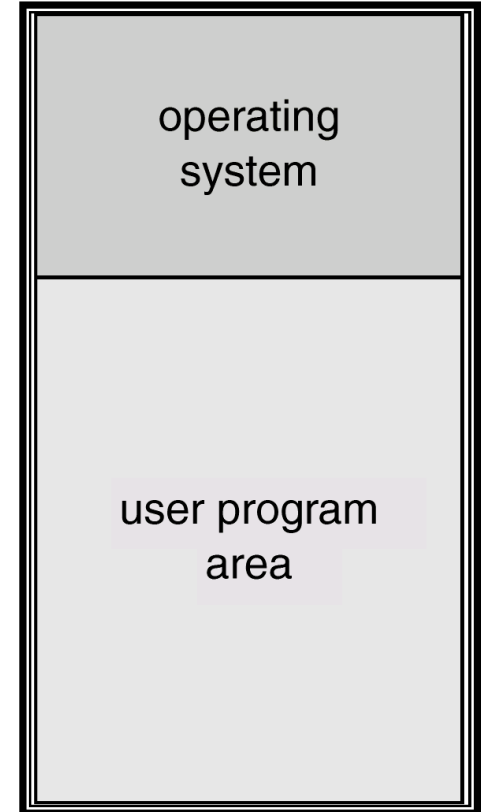


**ENIAC** (Electronic Numerical Integrator And Computer), 1946

## 2nd Generation (1955-65) - Transistors and mainframes

### Batch systems

- One job at a time
- Card readers, tape drives, line printers
- OS is always resident in memory and merely transfers a control
- CPU is underutilized due to the bottleneck in I/O



# 3rd Generation (1965-80) - Integrated circuits (ICs)

---

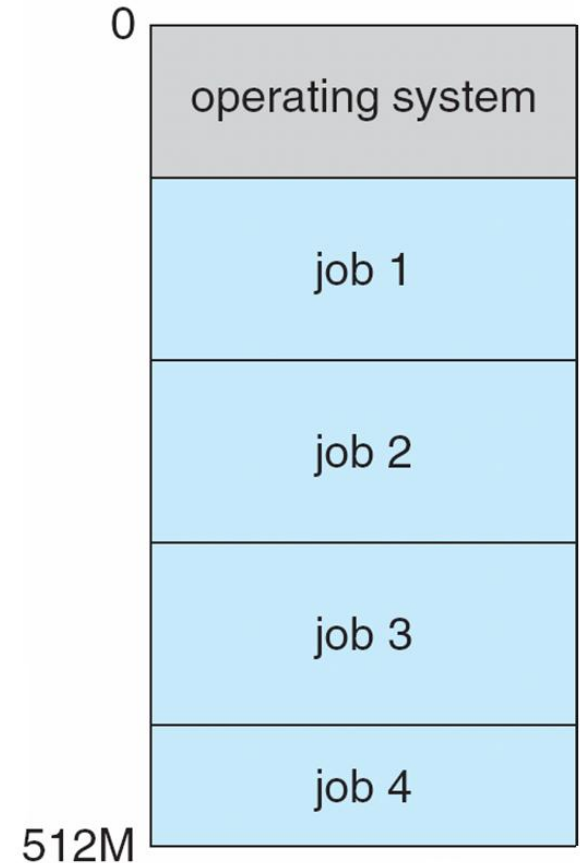
## Architectural advances

- Using ICs: better performance/price
- Disk drives
- On-line terminals
  
- The notion of "Computer Architecture":
  - IBM System/360 family

# 3rd Generation (1965–80) - Integrated circuits (ICs)

## Multiprogrammed systems

- Increase CPU utilization
- OS features
  - Job scheduling
  - Memory management
  - CPU scheduling
  - Protection
- Spooling (Simultaneous Peripheral Operation On-Line)



# 3rd Generation (1965–80) - Integrated circuits (ICs)

---

## Time-sharing systems

- Improve response time
- OS features
  - Swapping
  - Virtual memory
  - File system
  - Sophisticated CPU scheduling
  - Synchronization
  - Interprocess communication
  - Interactive shell
  - More protection, ...

# 4th Generation (1980-) - LSIs & VLSIs

---

## Architectural advances

- Microprocessors: smaller and faster
- Storage: larger and faster
- Personal computers
- CPU work is offloaded to I/O devices

## Modern OS features

- GUI (Graphical User Interface)
- Multimedia
- Internet & Web
- Networked / Distributed, etc.

# The computer revolution

---

Progress in computer technology

- Underpinned by Moore's Law

Makes novel applications feasible

- Computers in automobiles
- Cell phones
- Human genome project
- World Wide Web
- Search Engines
- AI computation

Computers are pervasive and mobile

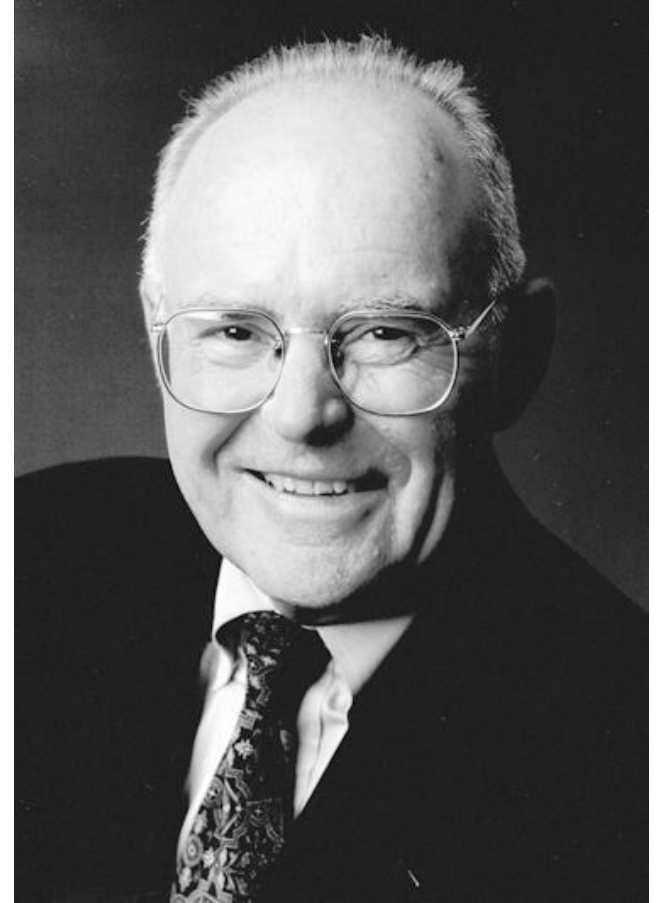
# Gordon Moore, 1929 -

Cofounded Intel in 1968 with Robert Noyce

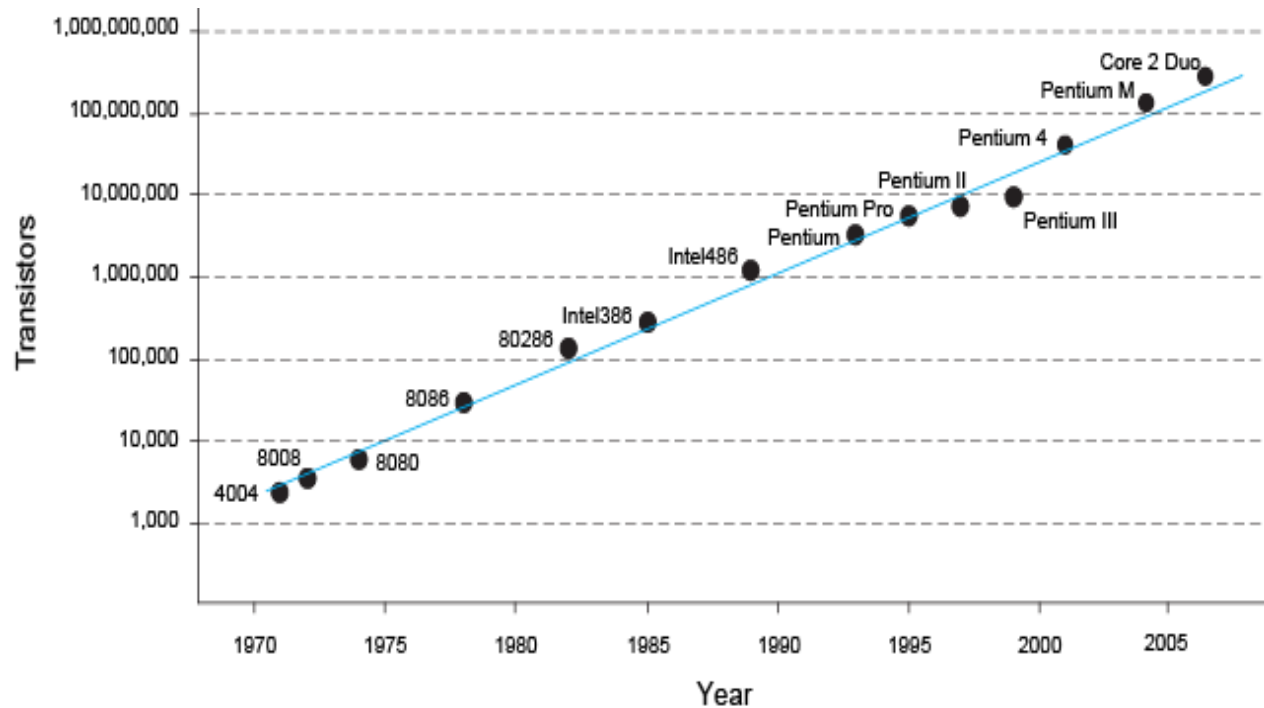
Moore's Law

- The number of transistors on a computer chip doubles every year (observed in 1965)

Since 1975, transistor counts have doubled every two years



# Moore's Law



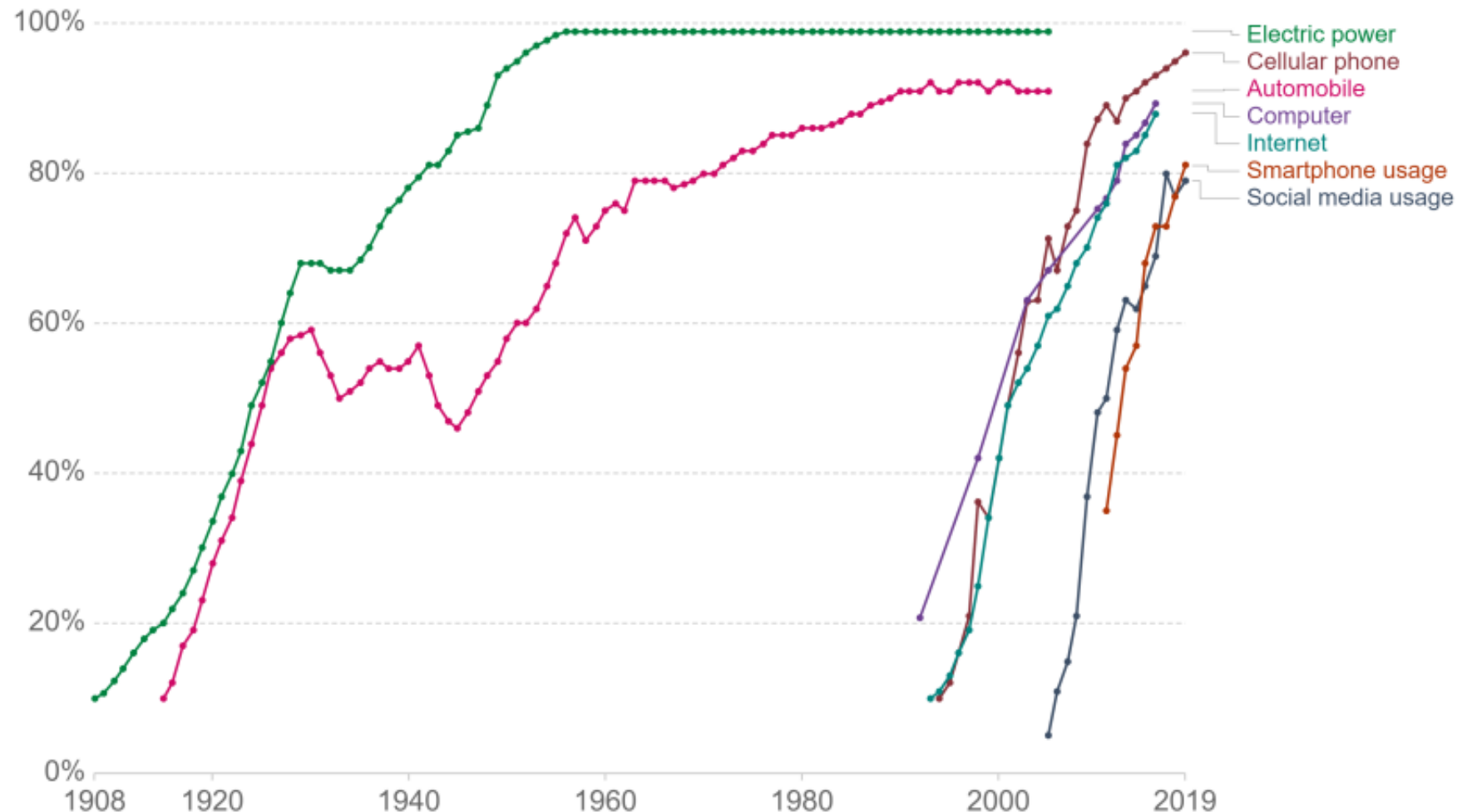
*“If the automobile had followed the same development cycle as the computer, a Rolls-Royce would today cost \$100, get one million miles to the gallon, and explode once a year . . .”*

*– Robert Cringley*

# Adoption of technology

## Technology adoption in US households, 1908 to 2019

Technology adoption rates, measured as the percentage of households in the United States using a particular technology.



Source: Comin and Hobijn (2004) and others

OurWorldInData.org/technology-adoption/ • CC BY

Note: See the sources tab for definitions of household adoption, or adoption rates, by technology type.

# OS history

**CTSS (1961, MIT)**  
**(Compatible Time Sharing System)**

**OS/360 (1964, IBM)**

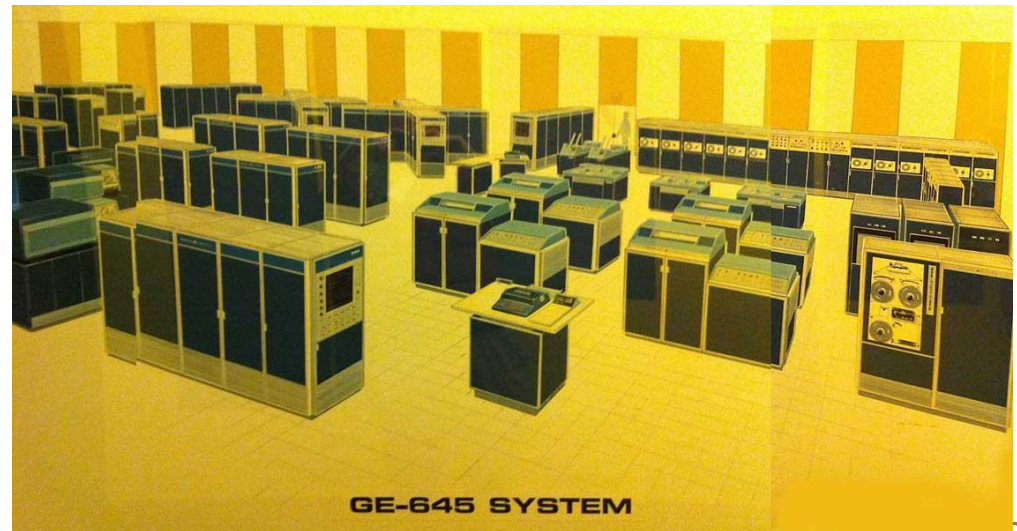
**MULTICS (1965, MIT, Bell Labs, GE)**  
**(MULTiplexed Information and Computing Service)**

**Unix (1969, Bell Labs)**

# Multics (1)

## Multics

- **Multiplexed Information and Computing Service**
- A time-shared, multi-processor mainframe "computing utility"
- Originally started by MIT, GE, and Bell Labs for GE-645, a 36-bit system, in 1965
  - Bell Labs quit in 1969 and built Unix
  - GE's computer business, including Multics, was taken over by Honeywell in 1970
  - Last system shutdown on 10/31/2000
- <http://www.multicians.org>

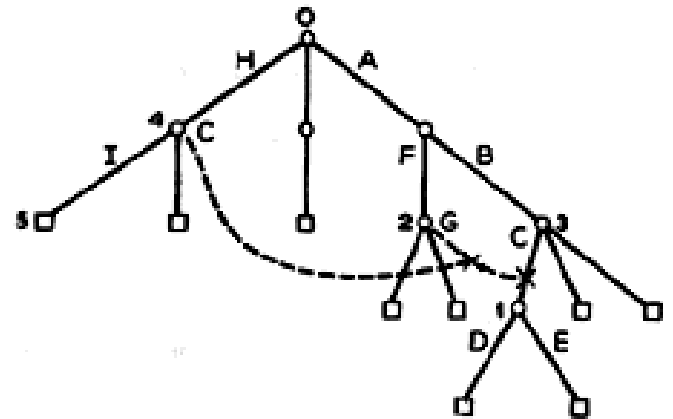


# Multics (2)

## Multics innovations

- Hierarchical file system

- File / directory / path name / working directory
- Access Control Lists (ACLs)
- Long names on entries
- Multiple names on entries
- Symbolic links
- Storage quotas
- Removable devices
- The backup procedures



- Lots of developments in management of virtual memory including segmentation and paging

# Multics (3)

---

## Multics innovations (cont'd)

- Separating the command `shell` from the OS kernel
- `Dynamic linking`
- Implementation of an OS in a `high level language` (PL/1)
- Management of `shared memory`
- Mapping of logical disk volumes onto physical volumes
- Many developments in the area of secure computer systems
  - Multics was rated B2 by the NCSC in 1985
  - A subsequent system (based on the Multics experience) built by Honeywell was the first computer system ever rated A1

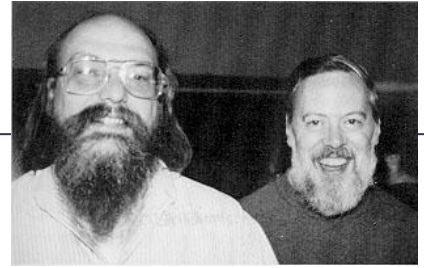
# Multics (4)

---

## Multics innovations (cont'd)

- **Multics Relational Data Store (MRDS)** in 1976
  - The first commercial relational DBMS
  - The MRDS query language was similar to early SQL
  - Concurrent access to a database by multiple processes was supported
  - The database could be backed up in its entirety
- **Spreadsheets** were developed on the Multics platform
- Multics supports BCPL, BASIC, APL, FORTRAN, LISP, C, COBOL, ALGOL 68, and Pascal
- Many optimizations for the LISP language through work on the Multics MACLISP compiler

# Unix (1)



Ken Thomson & Dennis Ritchie

"... When BTL (Bell Telephone Laboratories) withdrew from the Multics project, they needed to rewrite an operating system in order to **play space war** on another smaller machine (a DEC PDP-7 with 4K memory for user programs). The result was a system which a punning colleague called **UNICS (UNiplexed Information and Computing Services)** – an 'emasculated Multics'; no one recalls whose idea the change to UNIX was."

*-- Peter H. Salus, A Quarter Century of Unix, Addison-Wesley, 1994.*

"... It was the summer of '69. In fact, **my wife went on vacation** to my family's place in California.... I allocated **a week** each to the operating system, the shell, the editor, and the assembler, to reproduce itself, and during the month she was gone, it was totally rewritten in a form that looked like an operating system, with tools that were sort of known, you know, assembler, editor, and shell .... **Yeh, essentially one person for a month.**"

*-- Ken Thompson*

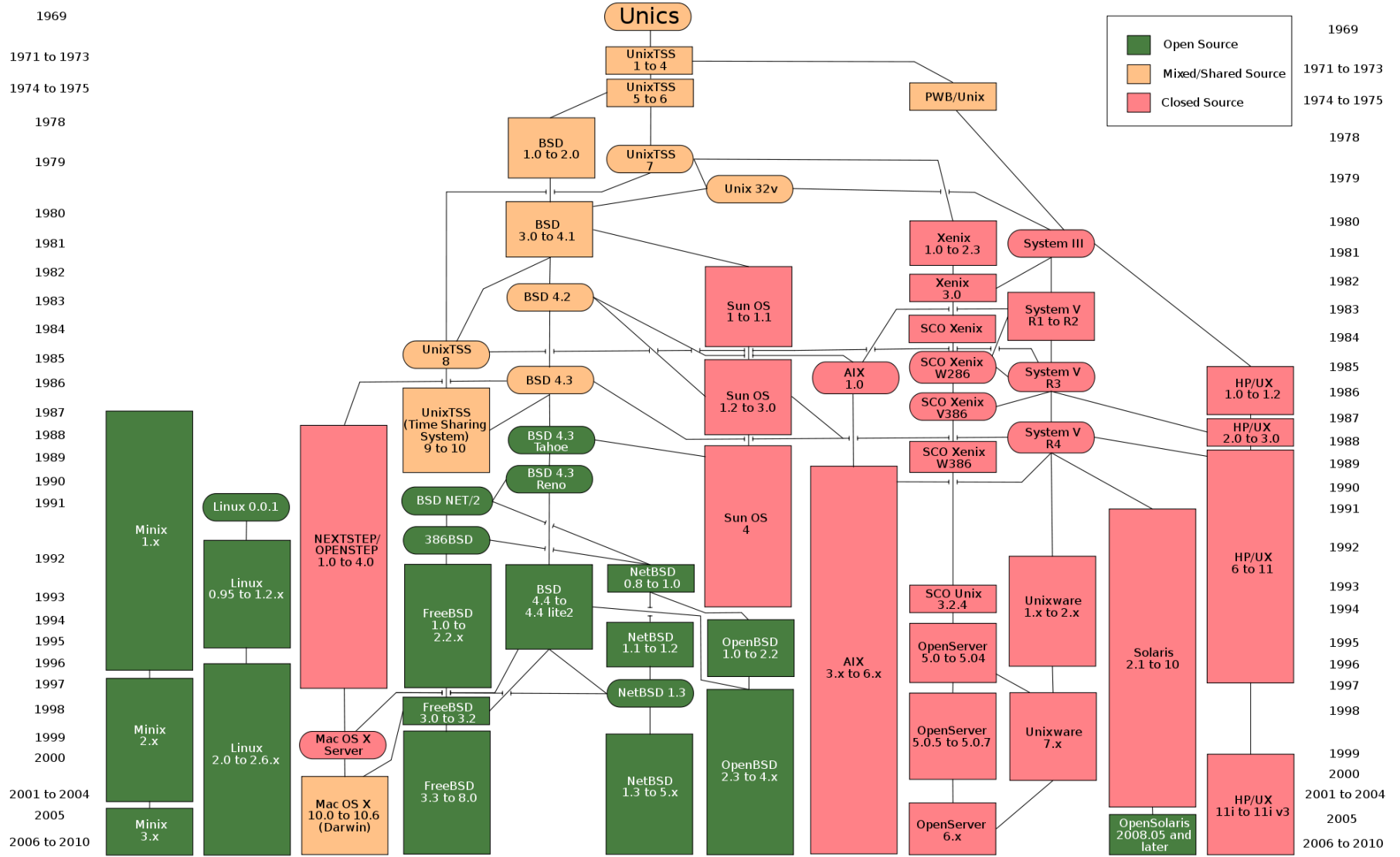
# Unix (2)

---

## Unix Features

- **Hierarchical file systems**
  - Special files: uniform I/O, naming, and protection
  - Removable file systems via mount/umount
  - i-node
- **Process control**
  - `fork()`, `exec()`, `wait()`, `exit()`
  - Pipes for inter-process communication
- **Shells**
  - Standard I/O and I/O redirection
  - Filters
  - Command separators
  - Shell scripts
- **Signals**

# Unix (3)

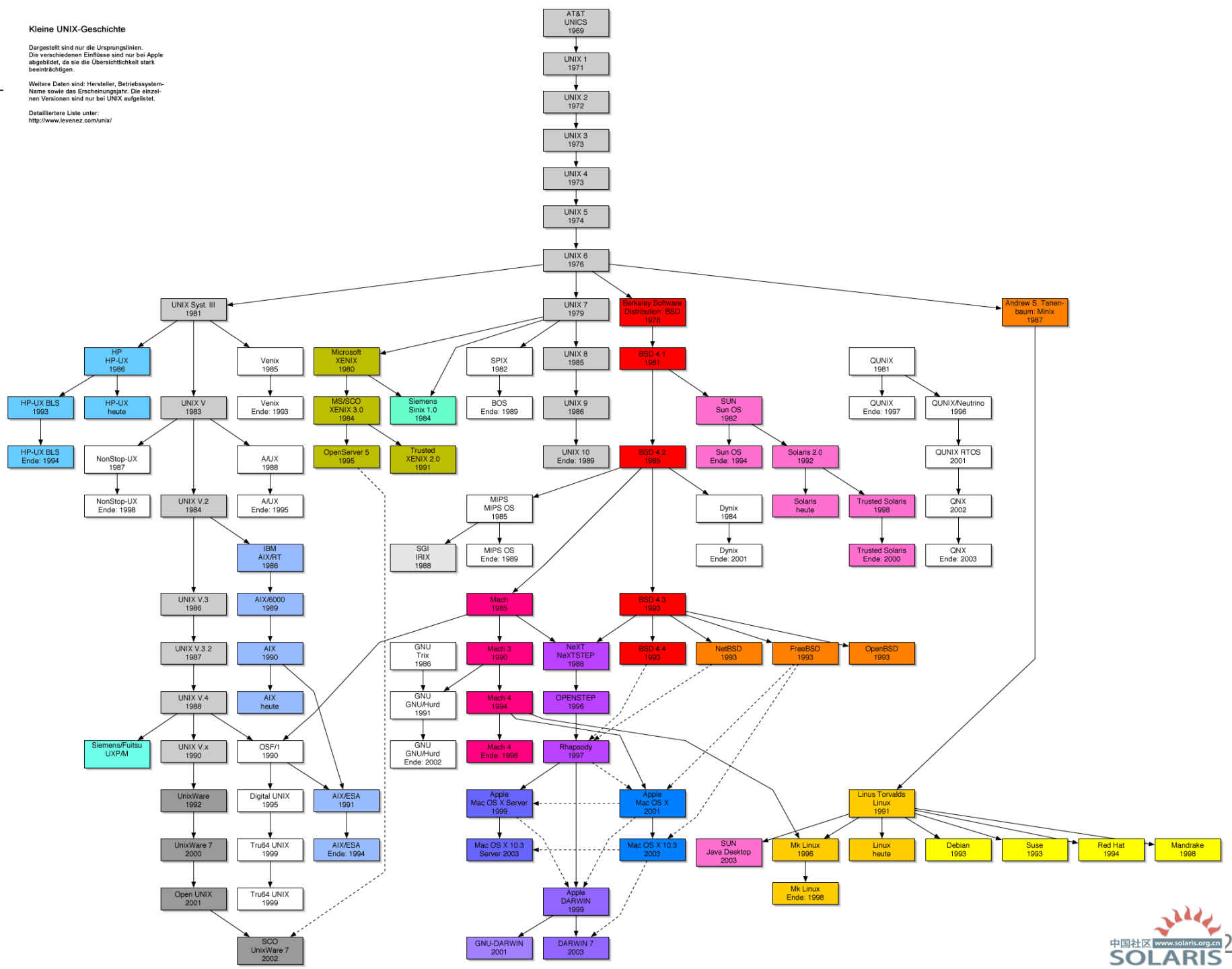


# Kleine UNIX-Geschichte

Dargestellt sind nur die Ursprungslinien. Die verschiedenen Einflüsse sind nur bei Apple abgebildet, da sie die Übersichtlichkeit stark beeinträchtigen.

Weitere Daten sind: Hersteller, Betriebssystem-Name sowie das Erscheinungsjahr. Die einzelnen Versionen sind nur bei UNIX aufgelistet.

Detailliertere Liste unter:  
<http://www.levenez.com/unix/>



# Unix (4)

---

Sun Solaris

HP HP-UX

IBM AIX

Compaq (Digital) Tru64

SGI Irix

SCO Unixware

FreeBSD, NetBSD, OpenBSD

... ..

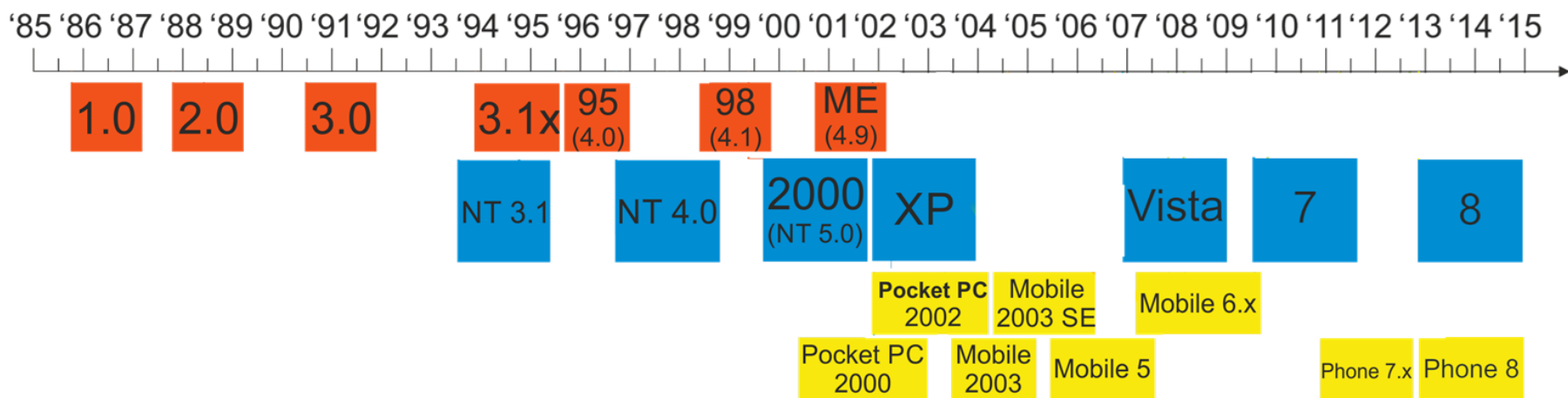
# Multics vs. Unix

---

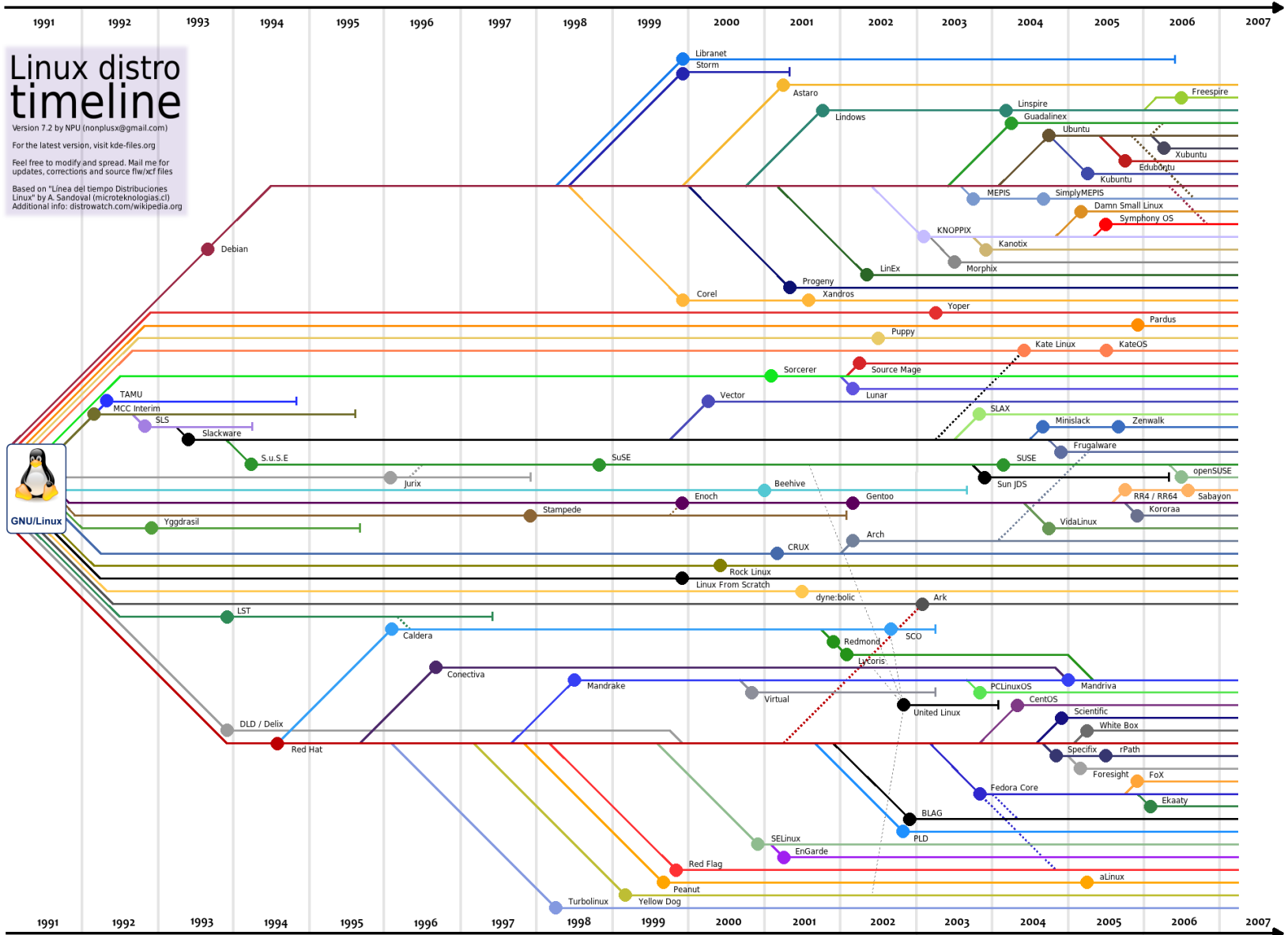
## Comparison

- Multics:
  - Top-down approach
  - 150 Man-Years for design and system programming
  - Another 50 Man-Years for improvements
  - Too complicated, too costly hardware
  - Many novel ideas had a great impact
- Unix:
  - Bottom-up approach
  - Simplicity and ease of use
  - Low cost hardware, university adoption
  - 2 Man-Years
  - The root of the modern operating systems

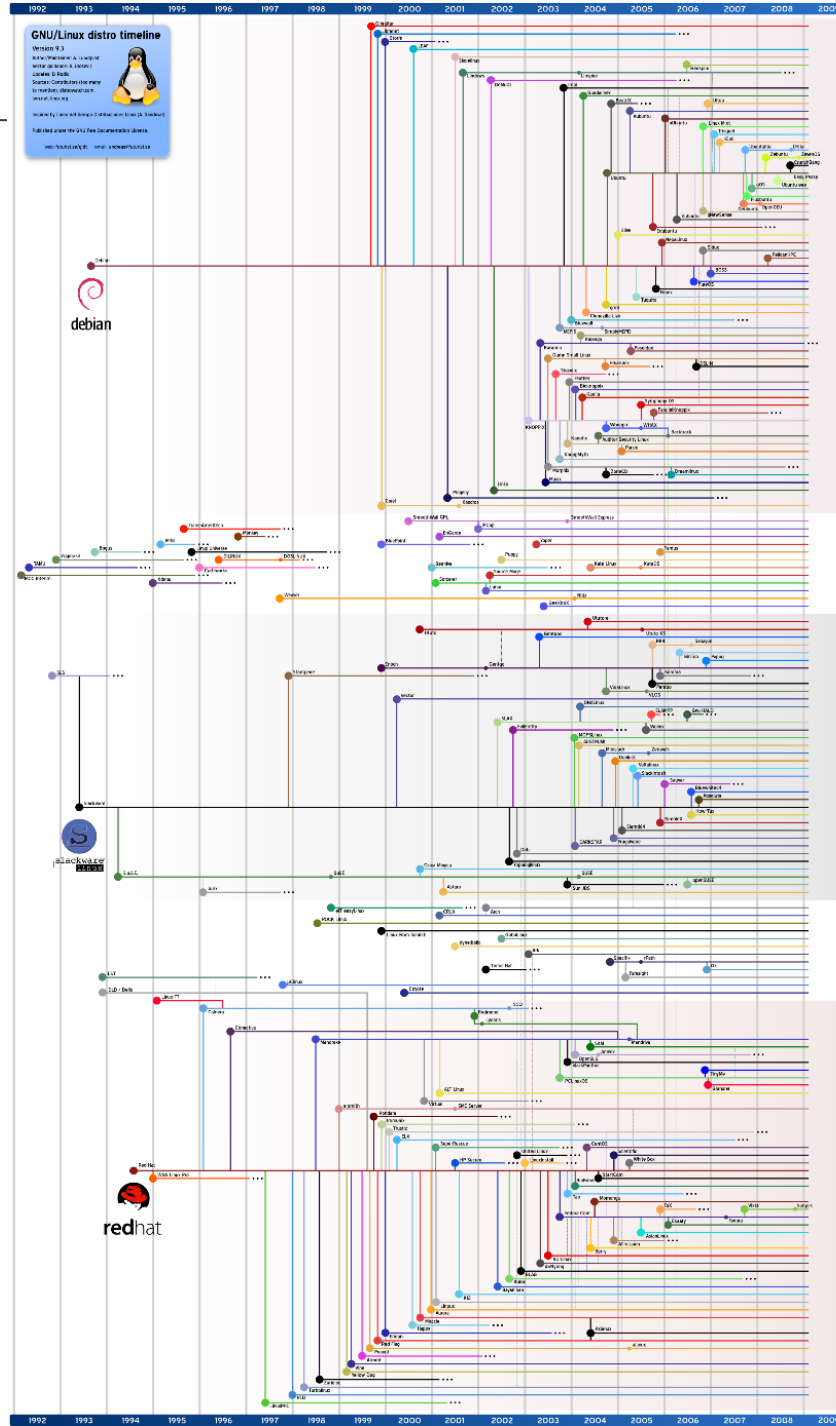
# Windows history



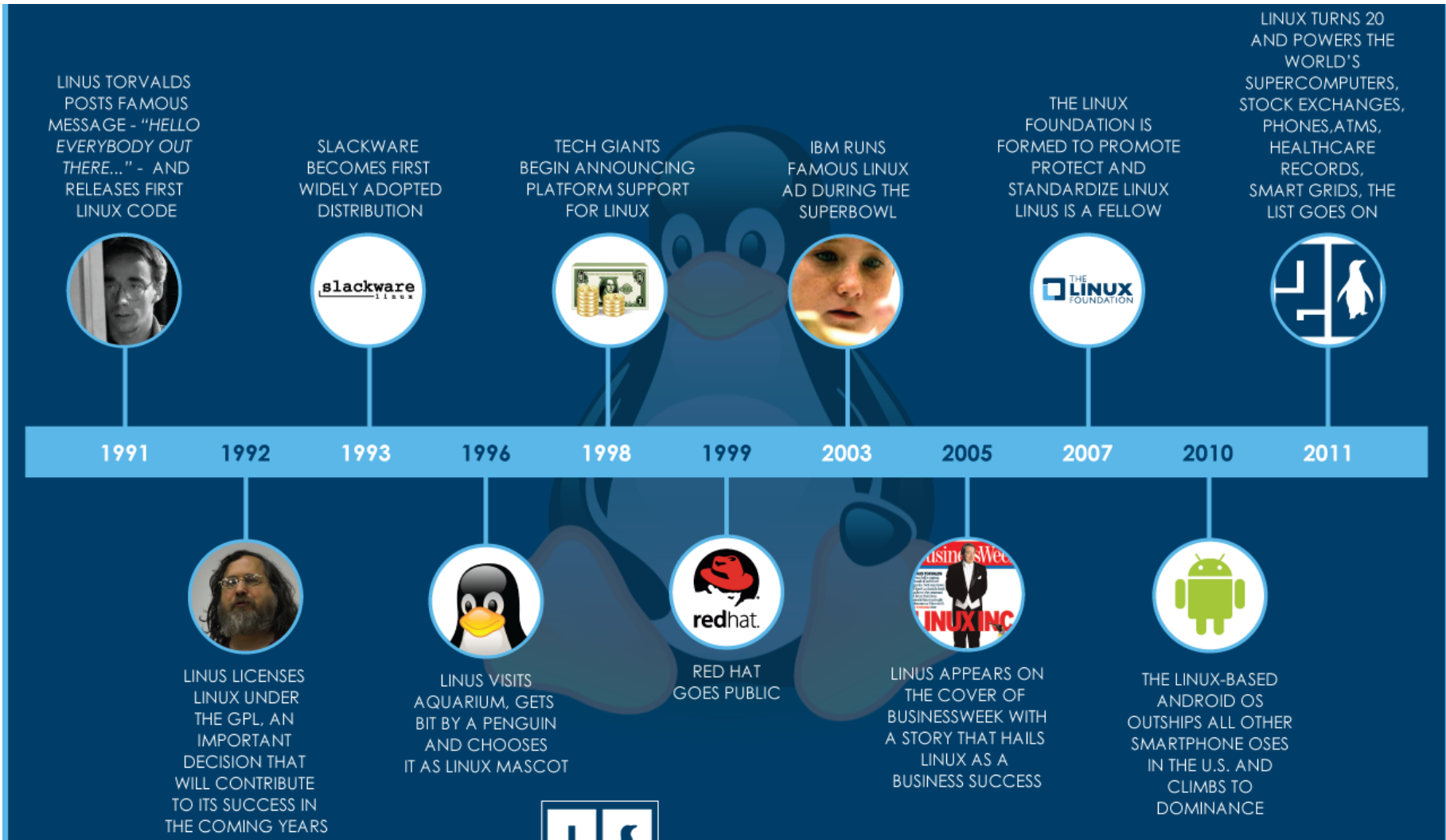
# Linux history



# Linux history



# Linux history



Torvalds wrote the Linux based on the Unix

Linux's power is "Open source" and "Compatibility"