History of Linux and the command line

Zubin

January 25, 2026

Contents

1 Operating Systems: Definition and Services

1.1 What is an Operating System?

An **operating system (OS)** is an intermediary between computer hardware (memory, processor, network cards, etc.) and the applications that users interact with.

- Users interact with applications.
- Applications request services from the operating system.
- The operating system manages and exploits hardware resources to provide services to the applications.

1.2 Examples of Operating Systems

- Microsoft Windows
- Apple macOS and iOS
- Google Android
- Unix and Unix-like systems such as Linux

1.3 Unix and Linux

- Unix has been around longer than Linux.
- Linux is a **Unix-like, open-source operating system**. The Linux kernel, created by **Linus Torvalds** and expanded upon by thousands of programmers, is available to the world for free.

1.4 Core Services Provided by an Operating System

1.4.1 File Management

- Managing the logical tree structure of files and their physical layout on storage devices (hard drives).

1.4.2 Memory Management

- Allocation, deallocation, and sharing of memory among multiple running processes.

1.4.3 Process Management

- Creation, execution, and termination of running applications (processes).

1.4.4 Input/Output Management

- Managing hardware like network interfaces, sound cards, video cards, printers, and other peripherals.

2 Genesis of Operating Systems

2.1 Early Computers (Mid-1940s)

- The first computers were built using **vacuum tubes** (evacuated glass containers that control electric current).
- These were huge machines that filled entire rooms but performed more slowly than a modern hand-held calculator.

2.2 Programming and Operation

- Programming was done manually by rearranging hardware components.
- Input/output capabilities were very limited.
- A single individual often acted as the designer, builder, programmer, and operator.

2.3 Invention of the Transistor

- The invention of the transistor led to smaller, more reliable computers.
- This innovation marked the beginning of operating systems through the appearance of **punch cards**.

2.4 Punch Cards and Role Separation

- Punch cards are cards with holes in specific locations to encode computer programs and data.
- This led to a separation of roles: programmers prepared the punch cards, and operators physically loaded them into the computer and handled the output.

2.5 Birth of Operating Systems

- Operating systems were invented to manage memory, processes (running programs), and input/output operations like reading punch cards.
- We can date the invention of operating systems to the **mid-1960s**.

3 UNIX Genesis

3.1 Technological Context

- The era of modern computers emerged with the appearance of **integrated circuits** and magnetic disks.
- This period also saw the development of compatible computer families, such as the **IBM System/360** (1964), which made a clear distinction between architecture and implementation.

3.2 Project MAC at MIT

- It all started with **Project MAC** (Mathematics and Computation), founded at MIT.
- It was funded by the US military's research agency (ARPA) and the National Science Foundation.
- The main goal was to develop a **timesharing system** that would allow a large community of users to access a single computer from multiple locations simultaneously.

3.3 MULTICS

- Developed by MIT, Bell Labs, and General Electric
- Stands for **Multiplexed Information and Computing Service**
- It evolved beyond timesharing to incorporate features like file sharing, file management, and system security.

3.4 Challenges of MULTICS

- The project proved much more difficult than expected.
- The system became operational in 1969 on the GE-645 computer, but its performance was far below the original targets.
- As a result, Bell Labs withdrew from the project in 1969.

3.5 Birth of UNIX

- Following the withdrawal, Bell Labs engineers **Ken Thompson** and **Dennis Ritchie** decided to create a simpler, minimal system.
- Using a little-used DEC PDP-7 machine, they began developing a single-user operating system.
- As a pun on the complexity of MULTICS, they called their system **UNICS**.

3.6 Evolution of UNIX

- In 1970, the system was enhanced to support multiple users, and its name morphed to **Unix**.
- At the time, the system was written in the **B programming language**, which was invented by Ken Thompson.

3.7 The C Programming Language

- In 1971, Dennis Ritchie improved upon B and called it **New B**.
- By 1972, the changes were so significant that Ritchie renamed his new language the **C programming language**.
- Ken Thompson then rewrote the entire Unix operating system in C.

3.8 Spread of UNIX

- The C source code for Unix was distributed to universities and research centers for educational purposes.
- From 1975 onward, a very active community emerged around Unix and C.
- Other notable developers included:
  - Douglas McIlroy (McElroy)
  - Joseph Ossanna
  - Rudd Canaday

3.9 Key Milestones

- 1978: Brian Kernighan and Dennis Ritchie published the book *The C Programming Language*.
- 1983: Thompson and Ritchie received the **Turing Award**, the highest distinction in computer science, for their invention.

3.10 Legacy of UNIX

- The concepts introduced by Unix are ubiquitous today and form the foundation for many modern operating systems.
- Derivatives of Unix include:
  - macOS
  - iOS
  - Android

– Linux, which is installed on the vast majority of today’s servers and connected objects.