System Programming under Linux (Draft)

REDA MAHER

Agenda

#	Training Logistics
#	Preparing the environment
#	Linux basic commands
#	Build Process

Logistics

What and Why?

What will we learn?

What is an OS?

- Process management
- Device management
- Filesystem management

Why Unix?

System Programming vs Application Programming

Why do we need to learn this topic?

Create a chance for a job

Deep understanding of the computer systems

Deep understanding of the high-level languages

Linux is used every where

Syllabus

Linux History

Command line usage and Build Process

Process Management

- Process overview
- System calls and command line arguments
- Process creation
- Orphan & Zombie processes
- Bash features
- Intro to Proc fs
- IO redirection in the shell

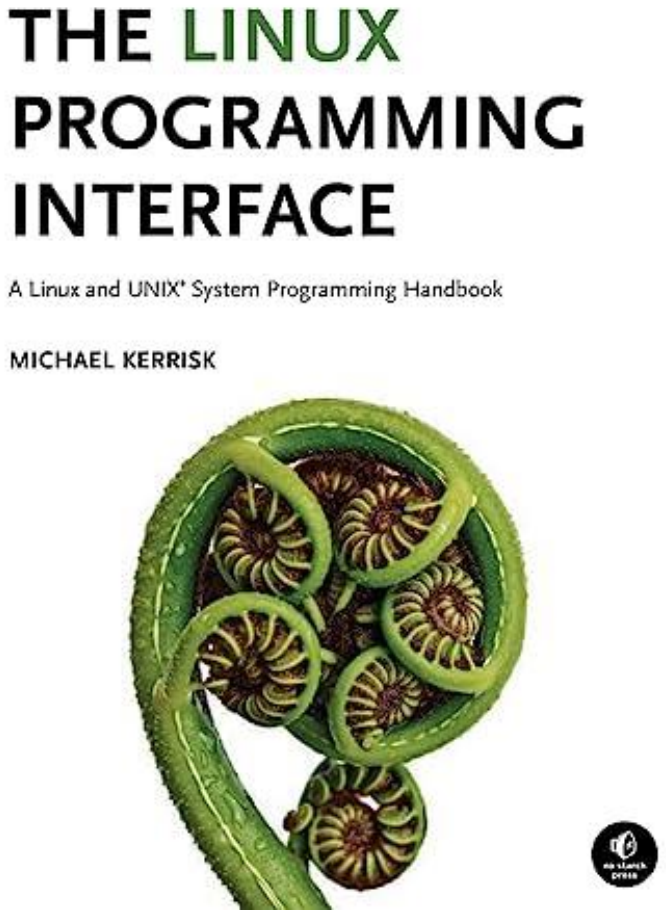
User Management

Filesystem Management

- Intro to HDD
- Parsing MBR Partition Tables
- File system formatting, mounting, and architecture
- Directories, files, links, and permissions in the file system
- Implementing “ls”

Textbook

- “The Linux Programming Interface: A Linux and UNIX System Programming Handbook”, by Michael Kerrisk



Sessions Schedule

Sunday, Tuesday, and Thursday 10:00 AM (~3 - 4hr with one break)

Training duration: 3 consecutive weeks (TBD)

Takeaways

- Interactive Training.
- Get your hands dirty.
- Unified setup (VM or Native).
- The course is an applied course.
- Take notes.
- You need patience in learning.
- This is not a Linux administration course.
- Training certificate.

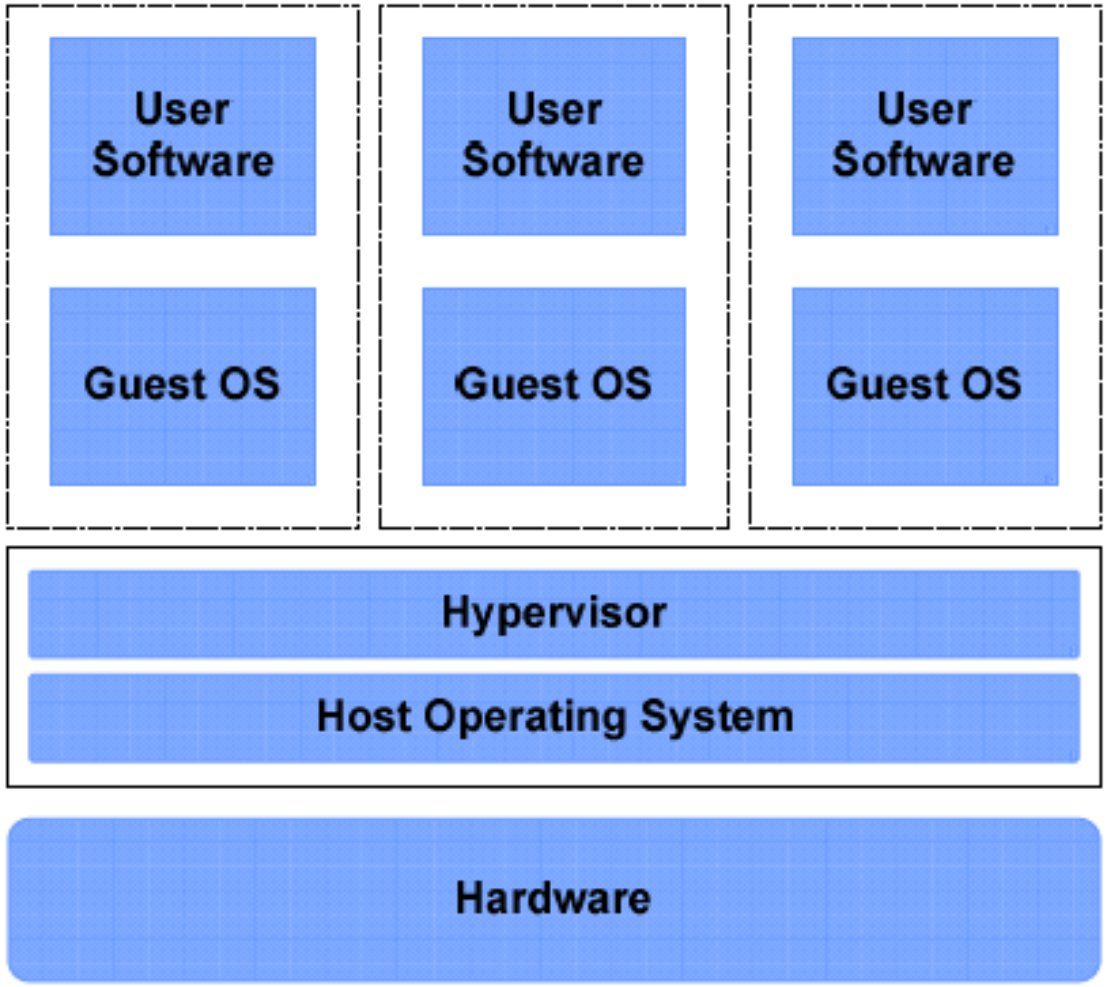
Communication Platform Setup

Environment Preparation

What is Virtualization?

- Virtualization is a framework or methodology of dividing the resources of a computer system into multiple execution environments.

- Platform virtualization is performed on a given hardware platform by host software (a control program), which creates a simulated computer environment, a virtual machine (VM), for its guest software.



Agenda

- # Training Logistics
- # Preparing the environment
- # Linux basic commands
- # Build Process

Logistics

What and Why?

What will we learn?

What is an OS?

- Process management
- Device management
- Filesystem management

Why Unix?

System Programming vs Application Programming

Why do we need to learn this topic?

Create a chance for a job

Deep understanding of the computer systems

Deep understanding of the high-level languages

Linux is used every where

Syllabus

Linux History

Command line usage and Build Process

Process Management

- Process overview
- System calls and command line arguments
- Process creation
- Orphan & Zombie processes
- Bash features
- Intro to Proc fs
- IO redirection in the shell

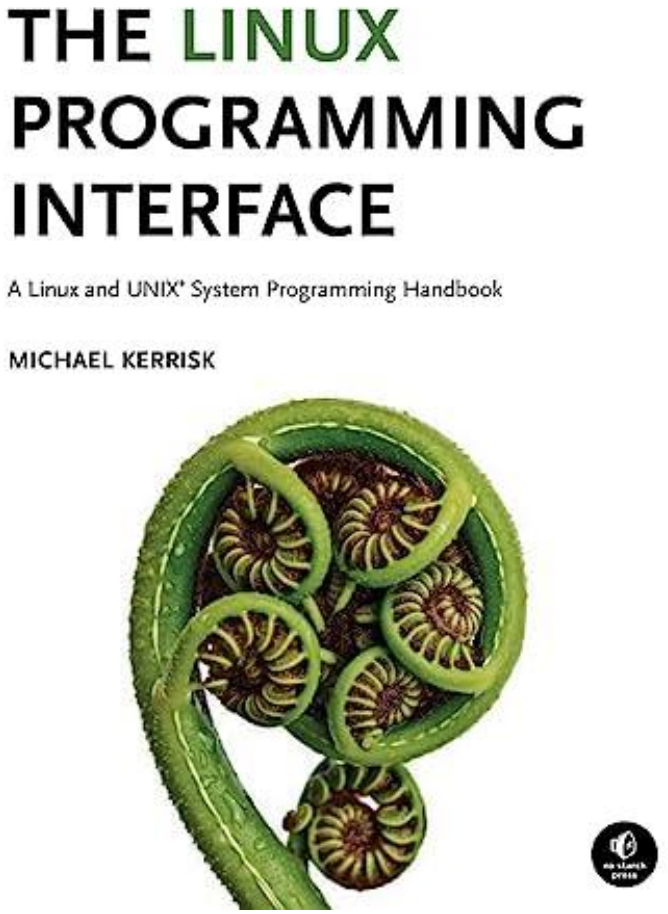
User Management

Filesystem Management

- Intro to HDD
- Parsing MBR Partition Tables
- File system formatting, mounting, and architecture
- Directories, files, links, and permissions in the file system
- Implementing “ls”

Textbook

- "The Linux Programming Interface: A Linux and UNIX System Programming Handbook", by Michael Kerrisk



Sessions Schedule

Sunday, Tuesday, and Thursday 10:00 AM (~3 - 4hr with one break)

Training duration: 3 consecutive weeks (TBD)

Takeaways

- Interactive Training.
- Get your hands dirty.
- Unified setup (VM or Native).
- The course is an applied course.
- Take notes.
- You need patience in learning.
- This is not a Linux administration course.
- Training certificate.

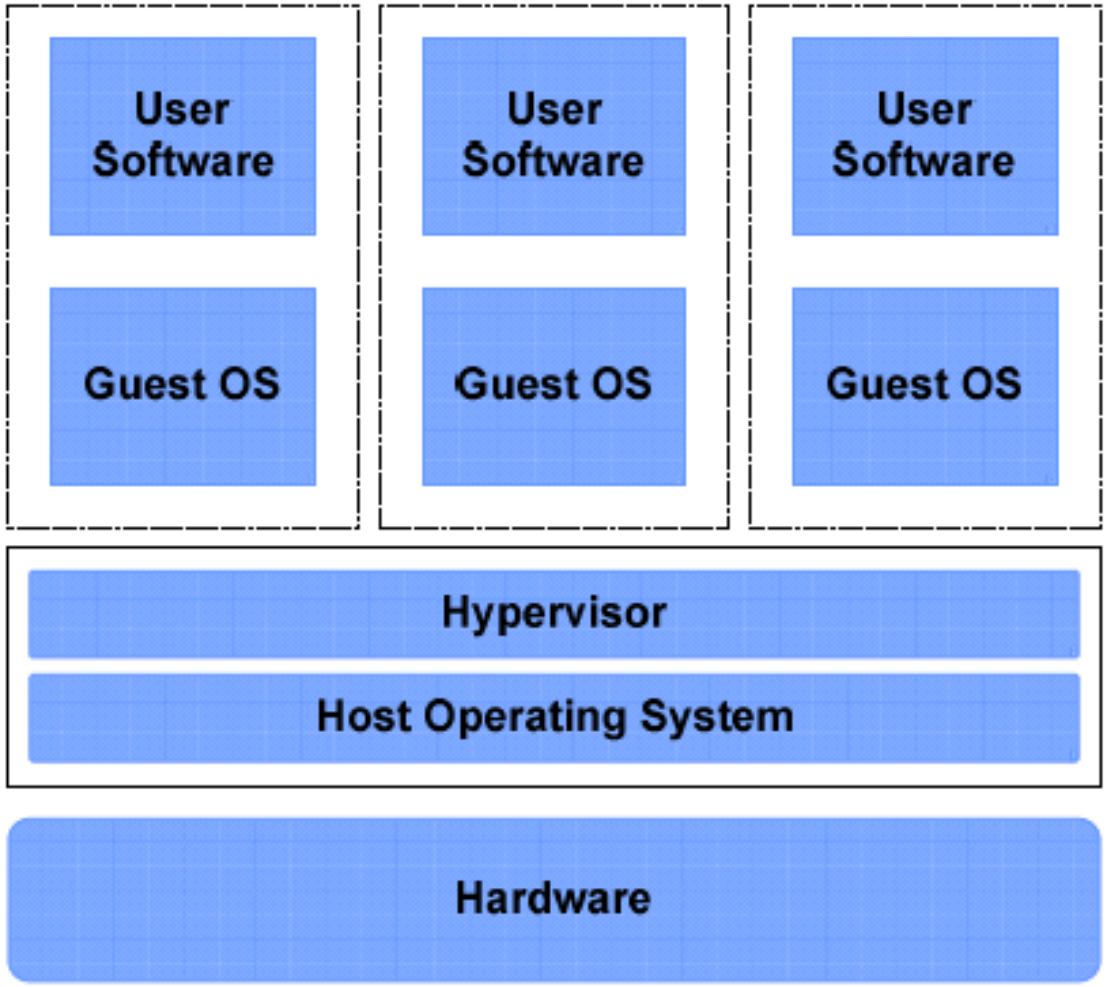
Communication Platform Setup

Environment Preparation

What is Virtualization?

- Virtualization is a framework or methodology of dividing the resources of a computer system into multiple execution environments.

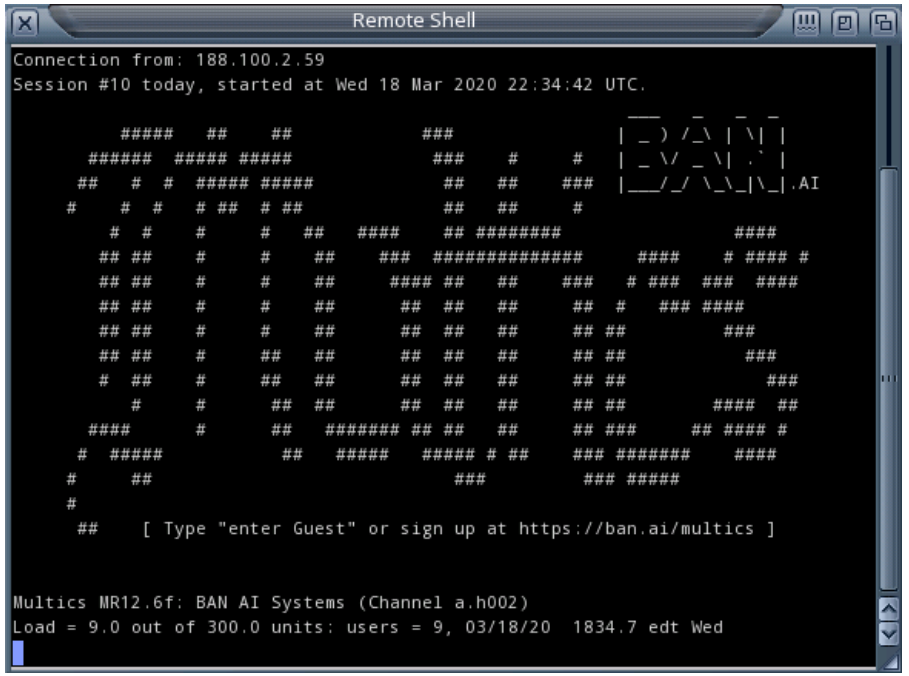
- Platform virtualization is performed on a given hardware platform by host software (a control program), which creates a simulated computer environment, a virtual machine (VM), for its guest software.



Linux History

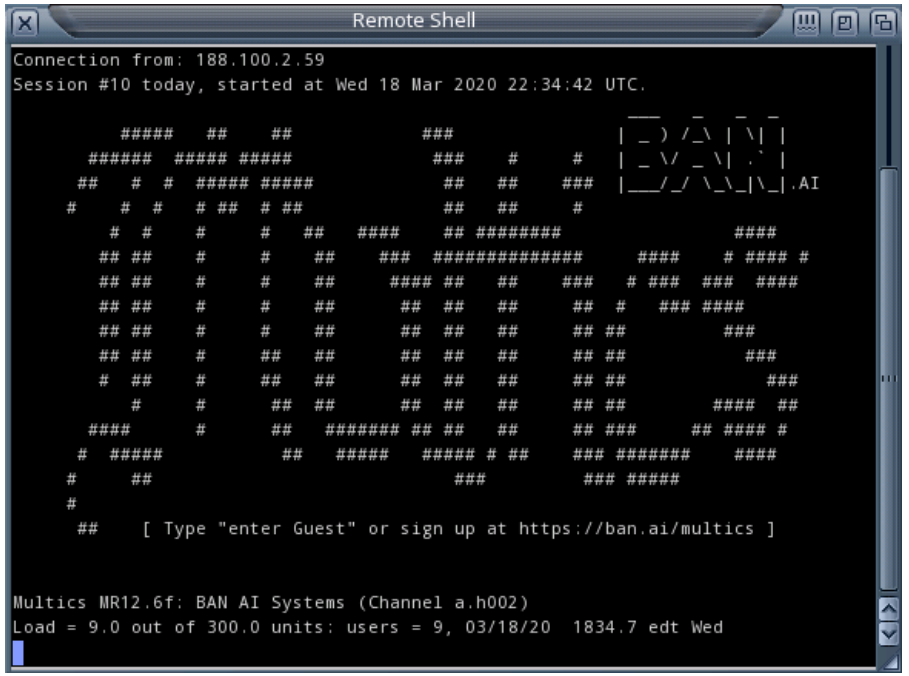
Multics

- Multics ("MULTiplexed Information and Computing Service") is an influential early time-sharing operating system based on the concept of a single-level memory.
- In 1964, Multics was developed as a cooperative project led by MIT along with General Electric and Bell Labs.



Multics Failure

- In 1969, Bell withdrew from the project as it became clear it would not deliver a working system in the short term.
- In 1970, GE decided to exit the computer industry entirely and sold the division to Honeywell.
- Nathan Gregory writes that “*Multics has influenced all modern operating systems since, from microcomputers to mainframes*”.
- Novel Ideas:
  - Dynamic linking.
  - Hierarchical file system.
  - Single-level store for data access.
  - …



Bell Labs

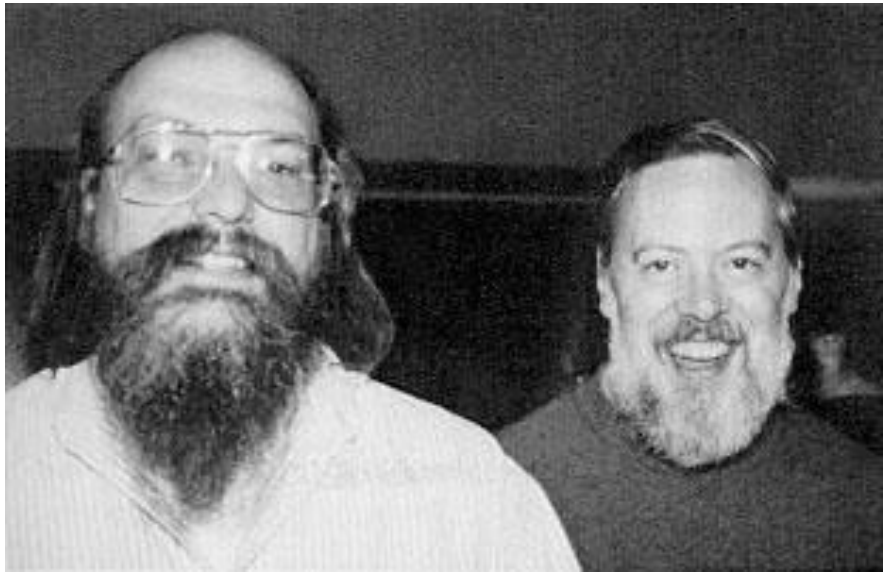
- In 1880, when the French government awarded Alexander Graham Bell the Volta Prize of 50,000 francs for the invention of the telephone. He used the award to fund the Volta Laboratory ("Alexander Graham Bell Laboratory").
- In 1889, *American Telephone & Telegraph Company (AT&T)* and its own subsidiary company took control of American Bell and the Bell System.
- Innovations at Bell labs:
  - Transistor.
  - Laser.
  - Information theory.
  - C, C++, AWK, and others.
  - Unix.
- Nine Nobel Prizes have been awarded for work completed at Bell Laboratories.

Bell Laboratories

AT&T Bell Laboratories

Unix Creation

- Ken Thompson and Dennis Ritchie developed Unix in Bell Labs on DEC PDP-7 machine.

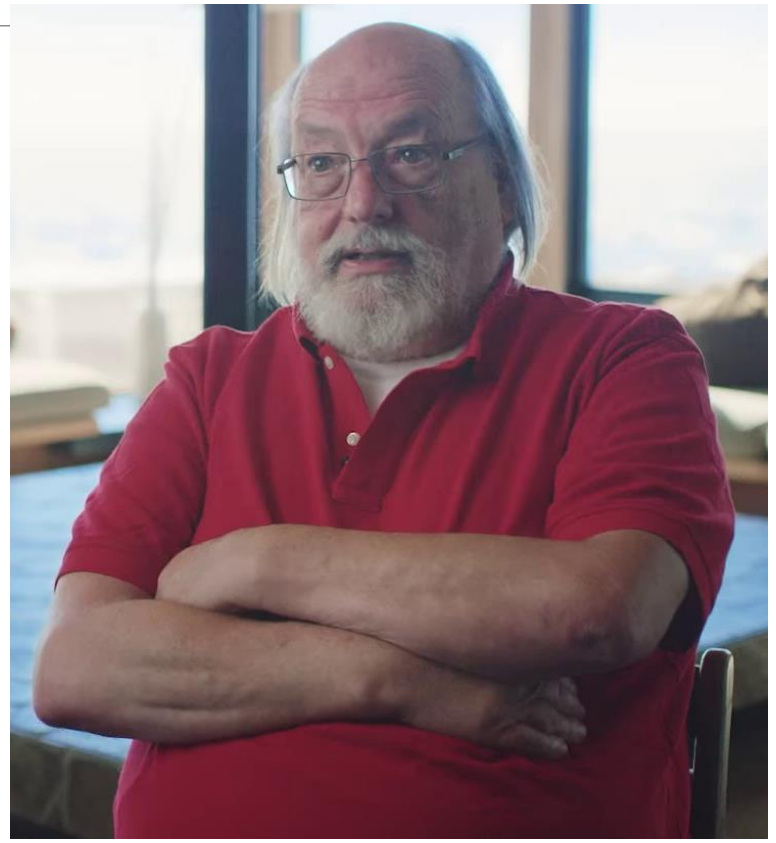


DEC PDP-7

- The PDP-7 was an 18-bit minicomputer produced by Digital Equipment Corporation (DEC) as part of the PDP series (1965).
- Price: US$72,000 (equivalent to $668,604 in 2022).
- Weight: 500 KG.
- Memory: 4K words (9.2 KB).
- Display: Printer.
- Input: Keyboard.

Unix First Version

- Ken Thompson wrote UNIX in 3 weeks in his wife vacation ☺.
- He wrote:
  - Editor.
  - Assembler.
  - Kernel.

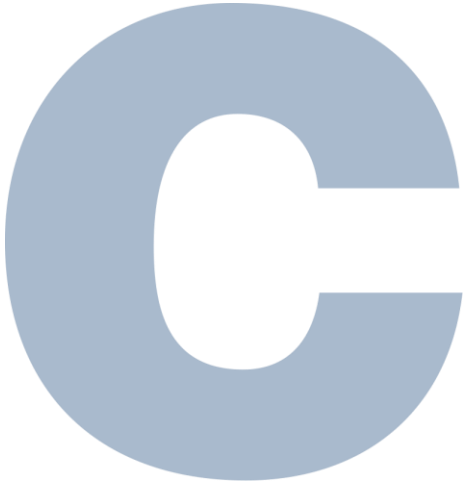


Rewriting Unix on PDP-11

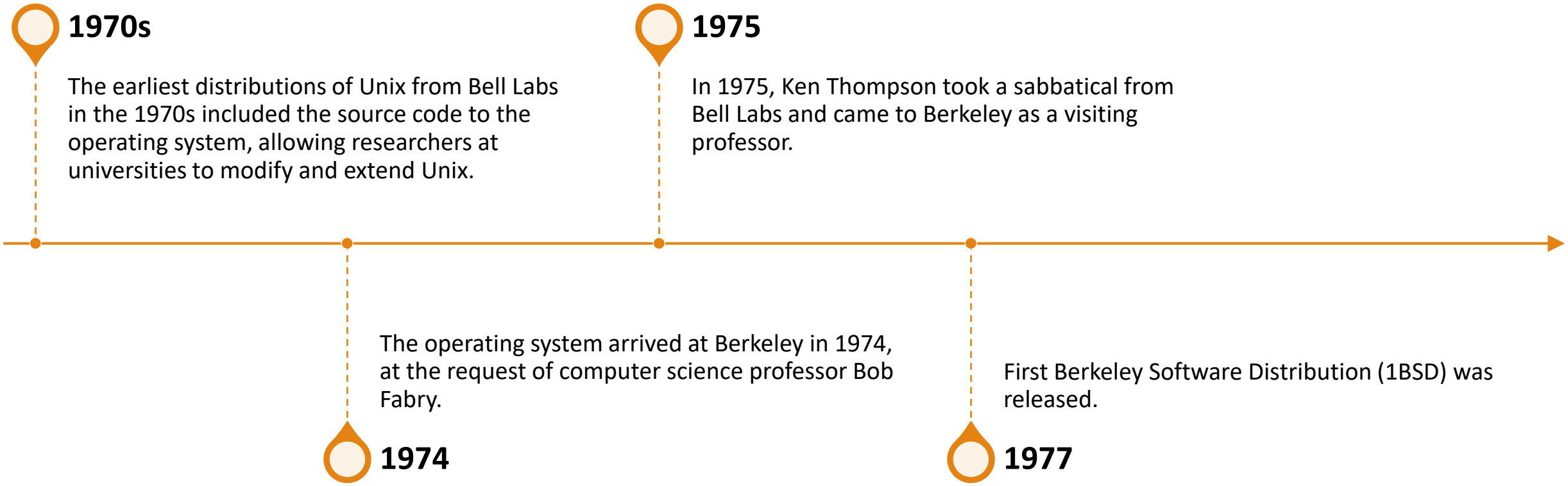
- As UNIX was written in assembly, Ken Thompson needs to rewrite it again on the PDP-11.

Inventing C language

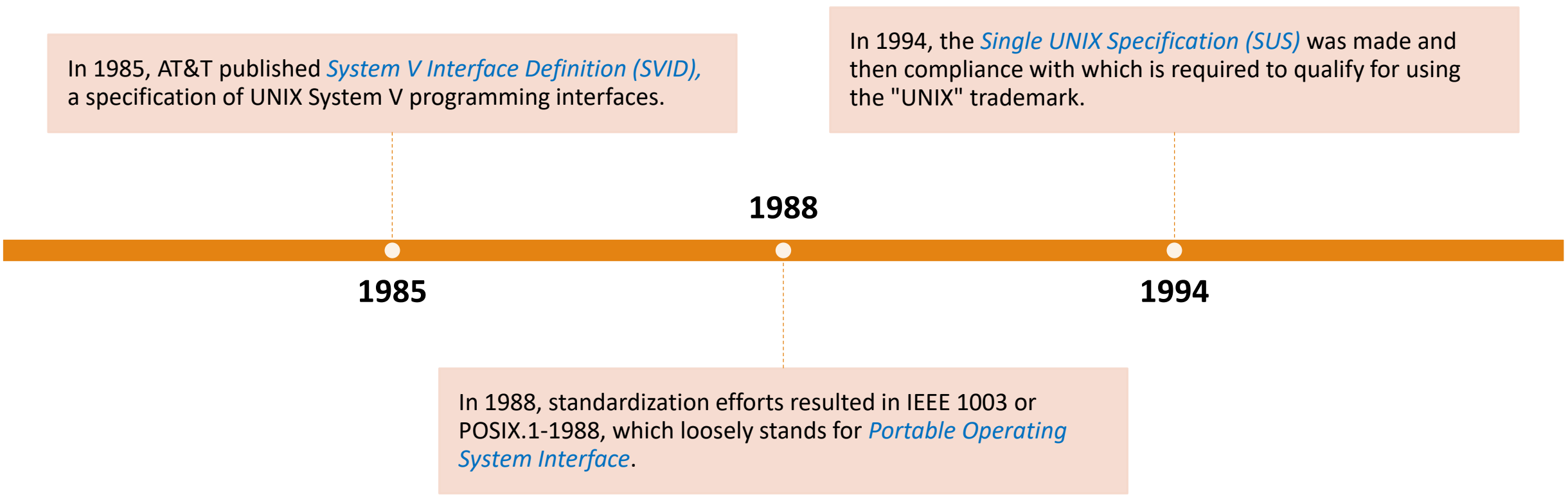
- Dennis Ritchie developed C language as a successor to B language (created by Ken Thompson).
- Then, Unix was ported to the C language.



Unix in UC Berkeley

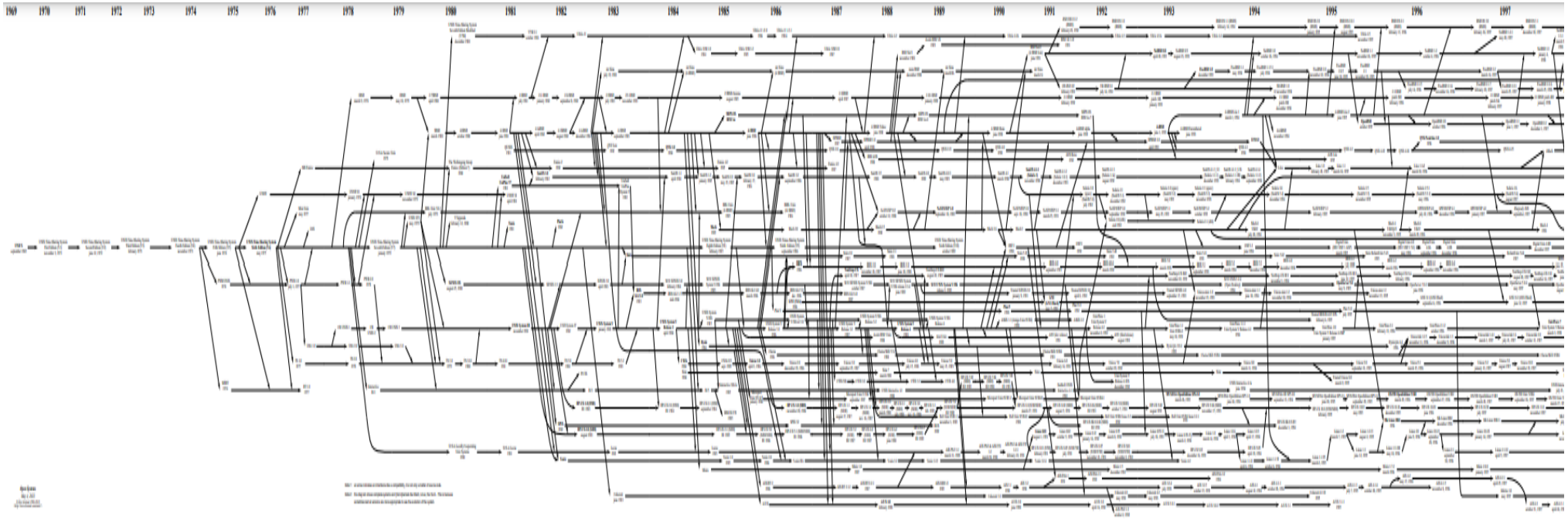


Unix standardization efforts



Unix Evolution

https://www.levenez.com/unix/



GNU

The GNU Project is a free software, mass collaboration project announced by *Richard Stallman* on September 27, 1983.

A recursive acronym meaning ***"GNU's not Unix!“***

The GNU General Public License (GNU GPL or simply GPL) is a series of widely used free software licenses that guarantee end users the four freedoms on the software:

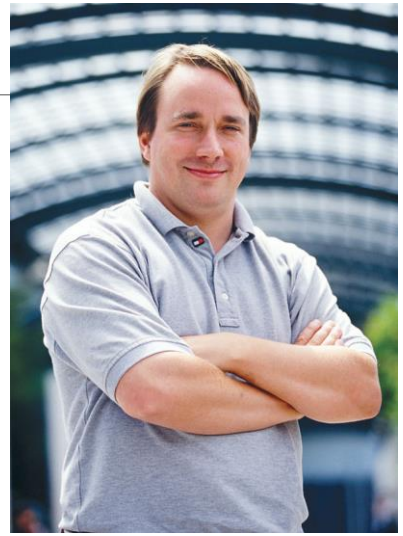
- Run.
- Study.
- Share.
- Modify.

The Creation of Linux

Surrounding conditions:

- Hardware evolved and Intel created X86.
- Internet evolved and mailing lists were popular.
- In 1987, MINIX, a Unix-like system intended for academic use, was released by Andrew S. Tanenbaum.

Linus Torvalds wrote the first kernel version in 1991.



Linux Statistics

47% of professional developers use Linux-based operating systems. (Statista)

Linux powers 39.2% of websites whose operating system is known. (W3Techs)

Linux powers 85% of smartphones. (Hayden James)

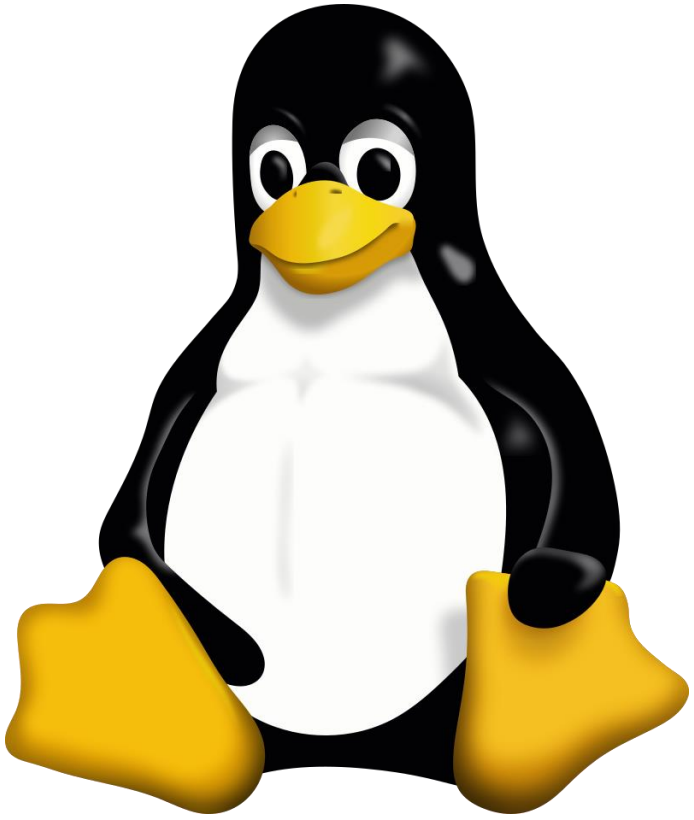
Linux, the third most popular desktop OS, has a market share of 2.09%. (Statista)

The Linux market size worldwide will reach $15.64 billion by 2027. (Fortune Business Insights)

The world's top 500 fastest supercomputers all run on Linux. (Blackdown)

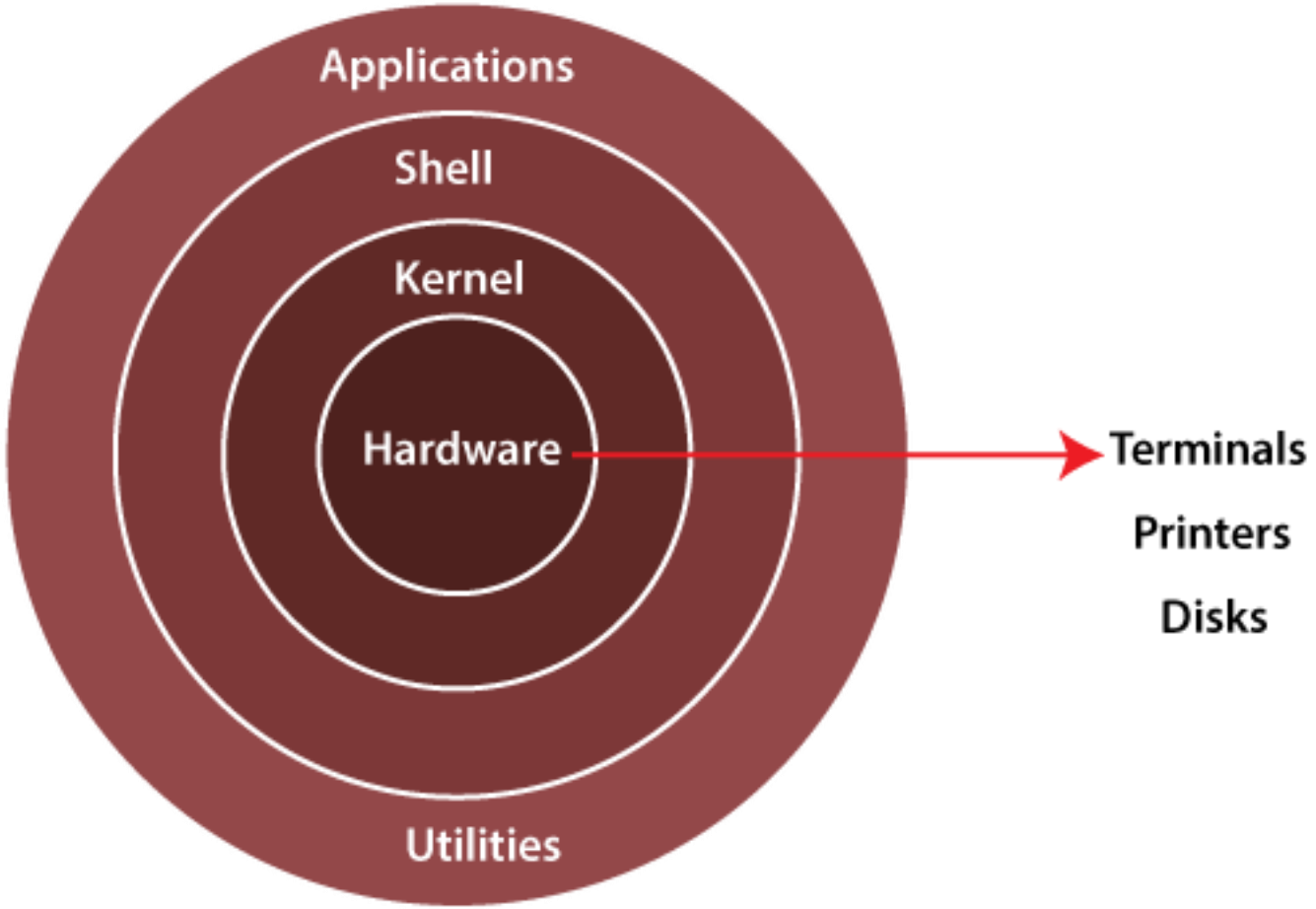
96.3% of the top one million web servers are running Linux. (ZDNet)

Today, there are over 600 active Linux distros. (Tecmint)

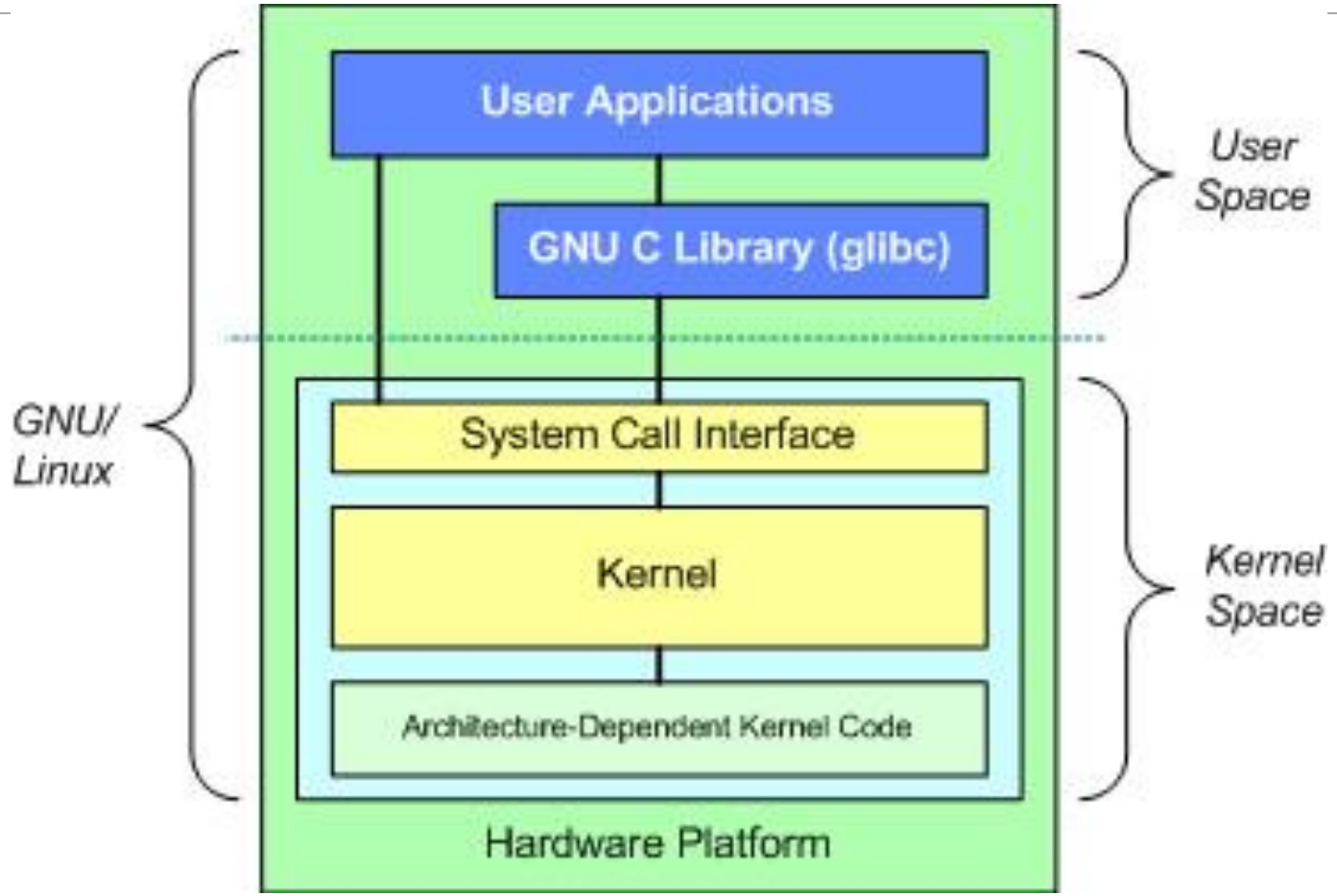


Linux Architecture

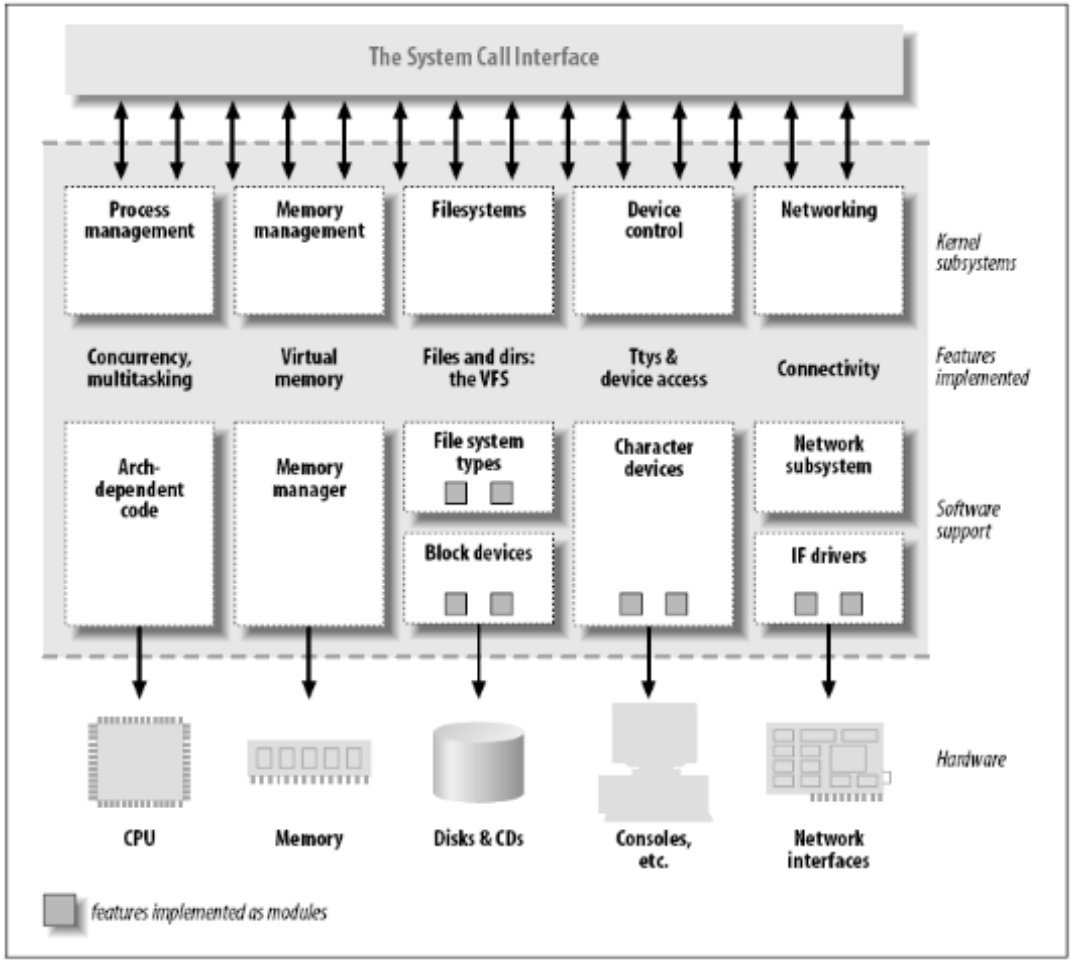
Linux OS


Applications
Shell
Kernel
Hardware
Utilities
Terminals
Printers
Disks

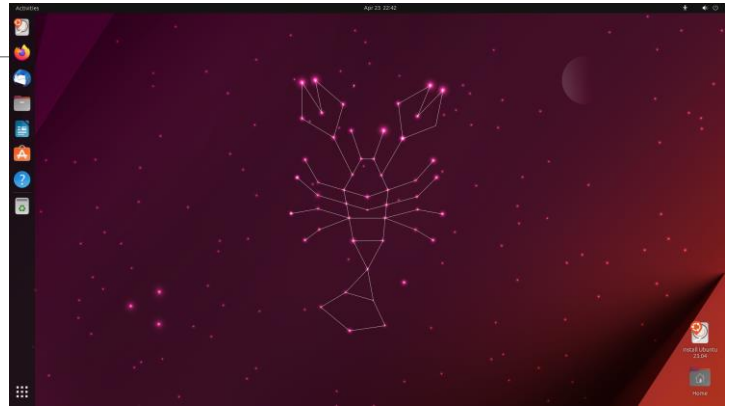
Kernel Space vs User Space



Linux Kernel



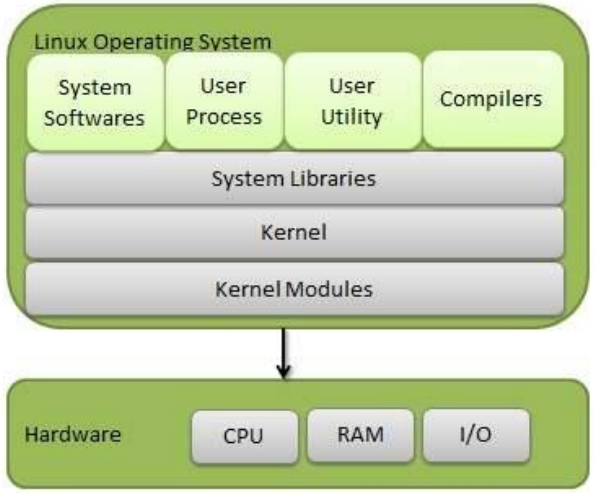
Linux Distributions



- A Linux distribution an OS made through a software collection that contains a Linux kernel, GNU libraries and tools, other software, a window system, documentation, a desktop environment, and a window manager.

Examples:

- Ubuntu
- Linux Mint
- Debian
- Red Hat Enterprise / CentOS
- Fedora
- Arch Linux
- Yocto



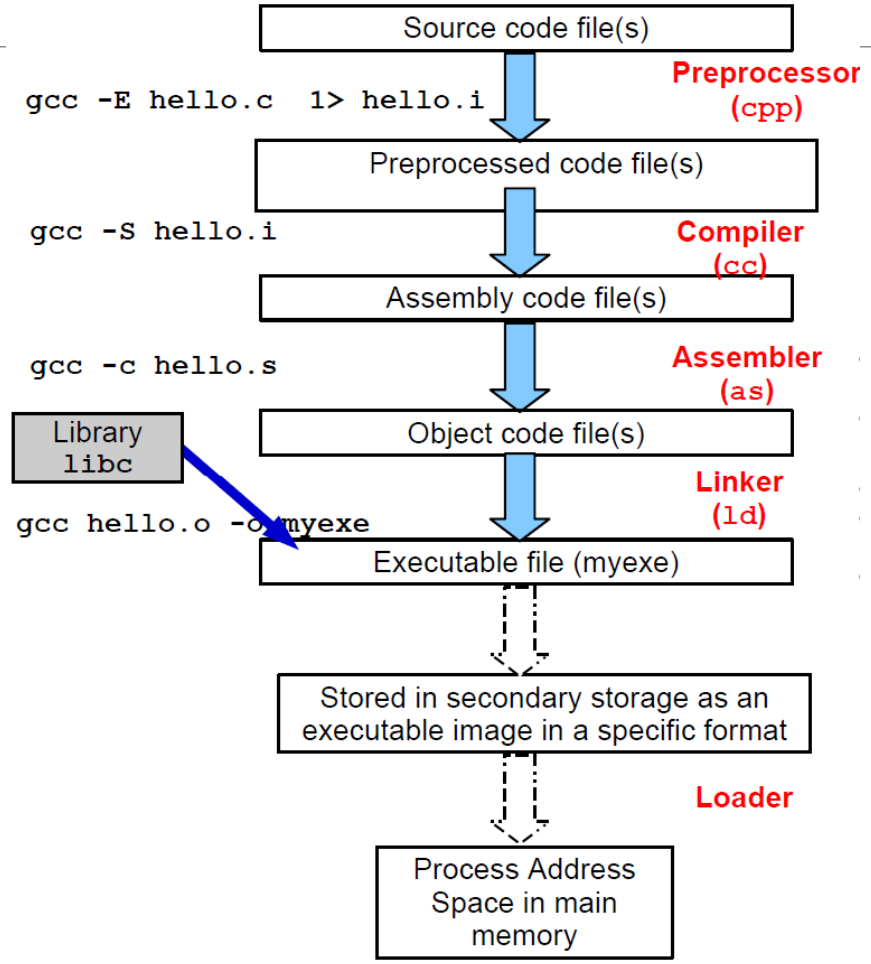
Linux Basic Commands

Basic Commands

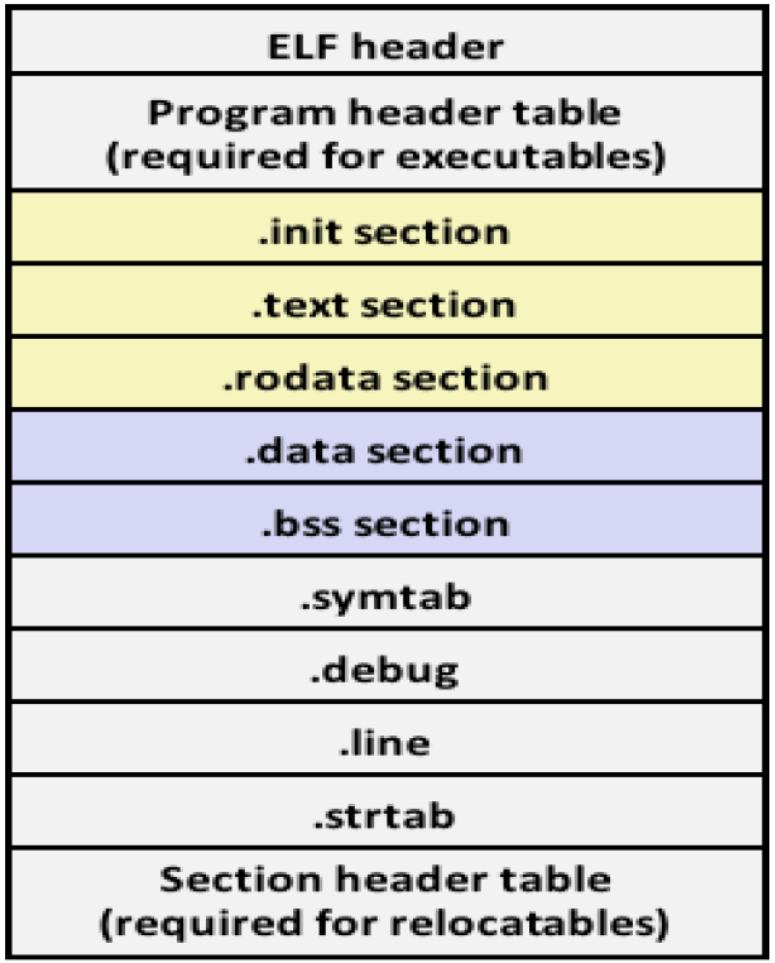
- Navigation (pwd, cd, mkdir, ls, ..).
- Getting help (help, man, ..).
- Copy and rename files.
- Creating directories.
- Viewing files.
- Editing files.
- I/O redirection.
- Pipes.
- History.

Build Process

Build Process



ELF Format


ELF header
Program header table
(required for executables)
.init section
.text section
.rodata section
.data section
.bss section
.symtab
.debug
.line
.strtab
Section header table
(required for relocatables)

Execution View