Outline

- Introduction - history
- Command line basics – getting help
- File system
- Working with files and directories
- More file handling
- The shell revisited
- Monitoring resources

1

Outline detail

- Background info
- OS kernel shell
- Command Line Interface (CLI)

2

Some background

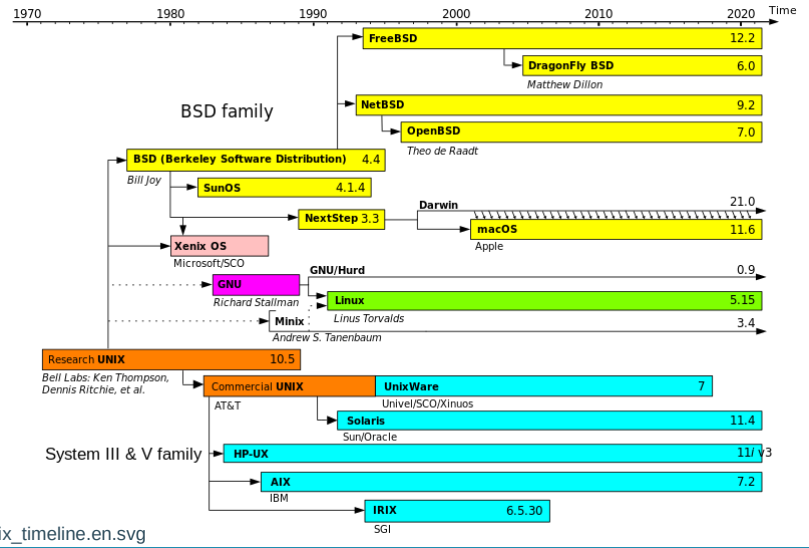
3

Operating system

- An operating system (OS) is system software that manages computer hardware, software resources, and provides common services for computer programs. (https://en.wikipedia.org/wiki/Operating_system)
  - Windows
    - Desktop systems
    - https://gs.statcounter.com/os-market-share/desktop/worldwide/
  - Linux
    - Server and supercomputing
  - MacOS
- Linux and Windows are based on foundations developed in the mid-1970s
- DOS, MacOS and UNIX are proprietary, i.e., the source code of their kernel is protected

4

*nix time line



https://en.m.wikipedia.org/wiki/File:Unix_timeline.en.svg

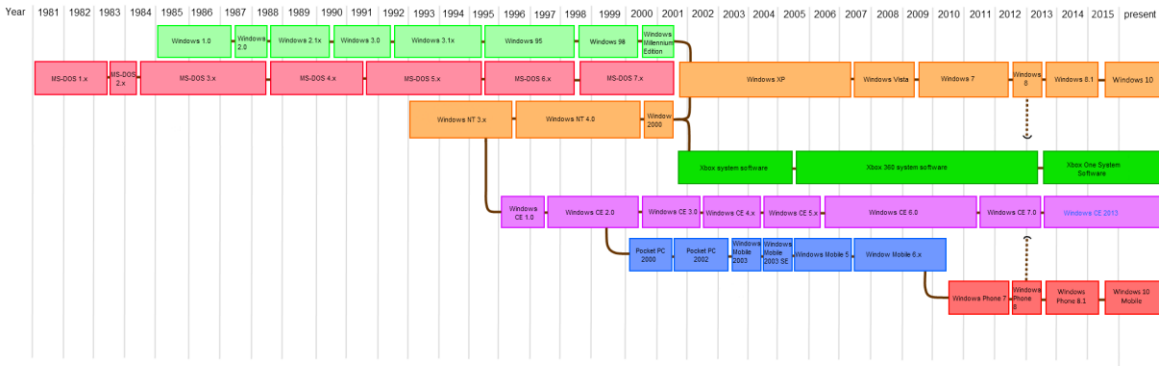
5

Some history

- 1970s: UNIX: roots in Bell Labs (AT&T) (https://www.bell-labs.com/about/history/innovation-stories/50-years-unix/#gref)
- 1985: Professor Andy Tanenbaum creates a UNIX like operating system based on System V Unix for the IBM PC & PC/AT computers: Minix.
- 1991: Linus Torvalds started developing Linux to create a system similar to Minix; released version 0.02
- Late 1980s: Richard Stallman and the Free Software Movement (FSF) made efforts to create an open-source UNIX-like operating system called GNU. In contrast to Linus Torvalds, Stallman started by creating utilities for the operating system first.
- These utilities were then added to the Linux kernel to create a complete system called GNU/Linux (or just Linux).

6

Windows time line



https://en.wikipedia.org/wiki/List_of_Microsoft_operating_systems#/media/File:Microsoft_timeline_of_operating_systems_2.png

7

OS, kernel, shell, etc.

14

Linux OS

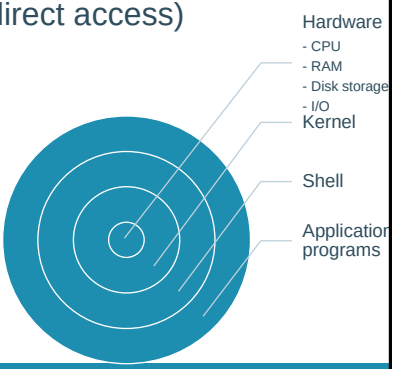
- Linux is not a single operating system, but rather a large family of free and open source operating systems based on the Linux kernel.
- Different variants within this family are referred to as **Linux Distributions** (https://distrowatch.com/) CentOS, Ubuntu, etc.
- There are two major components of Linux:
  - The **kernel** is the core of the Linux operating system:
    - Schedules processes and interfaces directly with the hardware.
    - It manages system and user I/O, processes, devices, files, and memory.
  - The **shell** is a text-only interface to the kernel.
    - Users input commands through the shell, and the kernel receives the tasks from the shell and performs them.
    - The shell works interactively (REPL- Read Evaluate Print Loop).
    - Users *do not* interact with the kernel directly, it is done through the shell or a desktop environment.

(https://cvw.cac.cornell.edu/Linux/introduction)

15

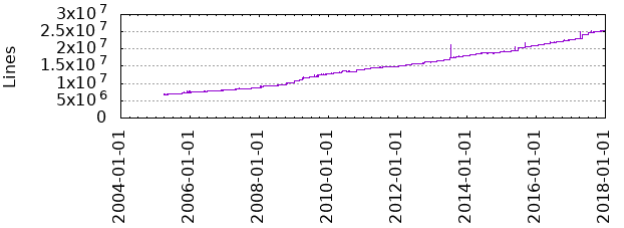
Linux kernel

- aka: executive, system monitor.
- Provides a layer between the computer hardware and user applications.
- Provides an interface for software to use hardware (no direct access)
- Planning and assigning:
  - Memory, CPU, disk, etc.
  - Security aspects
  - Fulfill user requests (system calls).
  - Filesystem, networking, ...



16

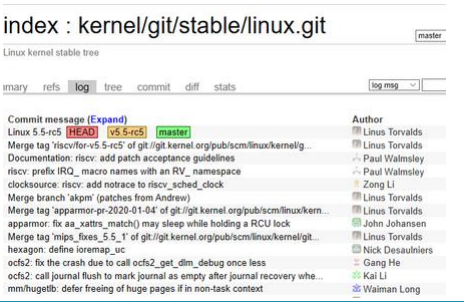
Linux kernel



- Lines of code

(https://www.phoronix.com/scan.php?page=news_item&px=Linux-Kernel-Commits-2017)

- Linux in 2020: 27.8 million lines of code in the kernel
https://www.theregister.co.uk/2020/01/06/linux_2020_kernel_systemd_code/



17

Linux kernel

- File Management
  - Controls the creation, removal of files and provide directory maintenance
  - For a multiuser system, every user should have its own right to access files and directories
- Process Management
  - Multitask system: multiple programs can be executed simultaneously
  - When a program starts to execute, it becomes a *process.* The same program executing at 2 different times will become 2 different processes
  - Processes are managed by the kernel: creating, suspending, terminating
  - A process is protected from other processes and can communicate with the others

19

Linux kernel

- Memory management
  - Memory in a computer is divided into main memory (RAM) and secondary storage (usually refer to hard disk)
  - Memory is small in capacity but fast in speed, and hard disk is vice versa
  - Data that are not currently used should be saved to hard disk first, while data that are urgently needed should be retrieved and stored in RAM
- Device drivers
  - Interfaces between the kernel and the BIOS (*basic input/output system*)
  - Different device has different driver

20

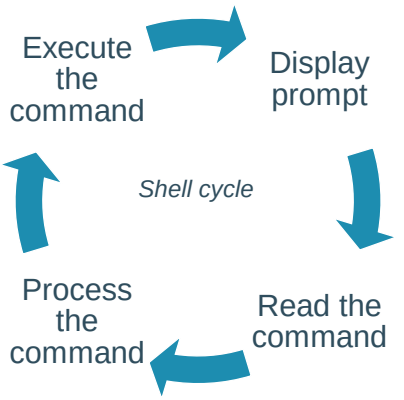
Linux kernel

- Get the kernel version
  `uname -r`
- `uname -a` provides the information in the order
  - KERNEL-NAME NODENAME KERNEL-RELEASE KERNEL-VERSION
  - MACHINE PROCESSOR HARDWARE-PLATFORM OPERATING-SYSTEM

21

Linux shell

- A shell is an interface for accessing the underlying commands of an operating system.
- When you sign in at the command line or launch a terminal window on Linux, the system launches the shell program.
- Is the command line interpreter: a program that accepts input from a user (e.g. a command) and performs the requested task. (REPL - *read–eval–print loop*)
- The shell's **prompt** identifies the type of shell being used



24

Most popular shells

- There are several types of shells for Linux.
- Check it with
  `$ echo $SHELL`
- Change the shell with
  `chsh`

File:check_shell.sh

Shell	Prompt	Name	Note
sh	$	Bourne Shell	Default on some Unix systems
bash	$	Bourne Again Shell	Enhanced replacement for the Bourne shell Default on most Linux and Mac OS X systems
csh	%	C Shell	Default on many BSD systems
tcsh	>	TC Shell	Enhanced replacement for the C shell
ksh	$	Korn Shell	Default on AIX systems

25

Shell

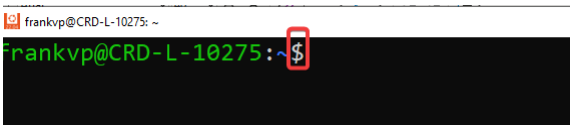
What does it?

- Command line editing and tools (auto complete)
- History
- Job control
- Configuration management (startup scripts)
- Automation / Scripting language

26

The prompt

- $: prompt, which shows us that the shell is waiting for input
- The shell may use a different character as a prompt and likely adds information before the prompt.



27

CLI

28

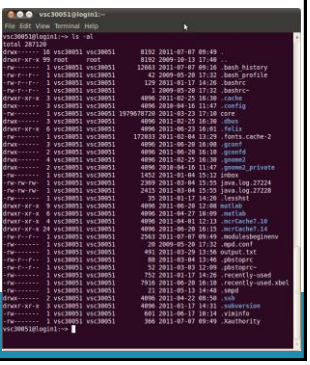
Linux User Interface

- Traditional *nux (Linux, Unix) systems use **command line interface** (**cli** or text-based interface)
- User needs to type lines of command to instruct the computer to work, similar to Windows DOS command window
- Advantages:
  - Fast: few resources required
  - Reliable: some of these commands have been around for years. Commands change less frequently than the GUI counterparts. GUI counterparts often use these same commands under the hood.
  - Productivity: use the keyboard instead of navigating through a bunch of menus and screens with the mouse and tabbing back and forth between different applications
- Disadvantages:
  - Memorize the commands
  - Typing a command can be error prone

29

Command Line Interface: Beware!

- Common use on servers and HPC system
- Implicitly assumes that you know what you are doing. Don't be scared!
- Often there are no warnings with commands, no *undo*
- No recycle bin!
- If anything goes wrong, you can stop the command with **`ctrl+C`**



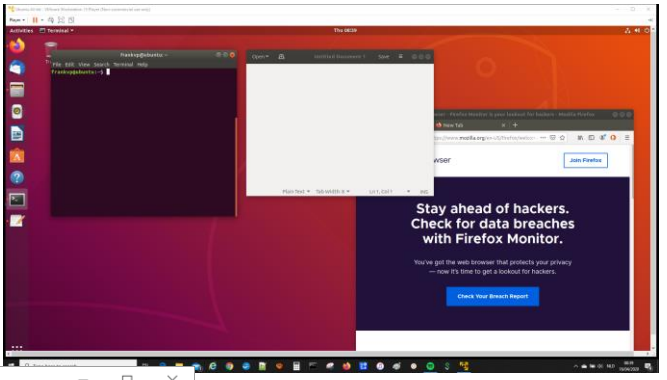
30

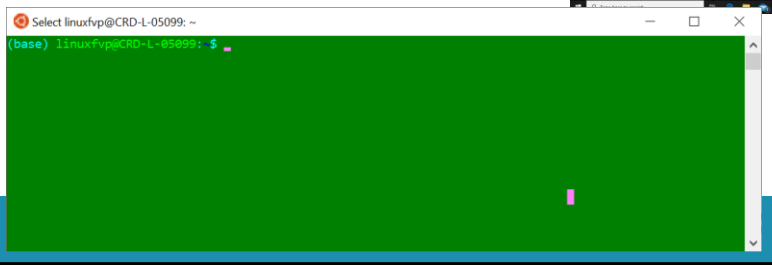
Linux User Interface

- By adopting the X-Window technology, graphical user interface (GUI) is available for Linux.
- Uses pointing devices (e.g. mouse) to control the system, similar to Microsoft's Windows
- Provide menu-driven and/or icon-driven interfaces
  - menu-driven: user is provided with a menu of choices. Each choice refers to a particular task
  - icon-driven: tasks are represented by pictures (icon) and shown to user. Click on an icon invokes one task
- Advantages:
  - No need to memorize commands. Always select task from menus or icons
- Disadvantages:
  - Slow and requires resources for its implementation
  - Suitable for general users and systems, such as PC use

31

Linux User Interface





32

Someone is doing something somewhere

- Linux is a multiuser OS
  - Allows multiple users to use the resource of a computer at the same time
  - Before you can do anything, you have to log in.
    - This can be either a graphical or a nongraphical prompt.
      - Linux desktop: you are likely to see a graphical environment.
      - A server: you'll just see a shell login prompt.
- Check who you are on the computer: `whoami`
- List the users logged on with: `who`
- What is the machine you are working on: `hostname`

34

Useful commands

- What is the directory you are working in: `pwd`
- What date/time is it: `date`
- Clear the contents of the current screen
  `$ clear`
- `$ logout`
  - The logout command logs your account out of the system (in a text mode).
  - This will end the terminal session and return to the login screen.
  - Some systems may have a file called `.logout` or `.bash_logout` in each user's home directory.
- `$ exit`
  - Exit the current shell. The `exit` command is similar to the `logout` command with the exception that it does not run the logout script located in the user's home directory.

35

Hands-on

37