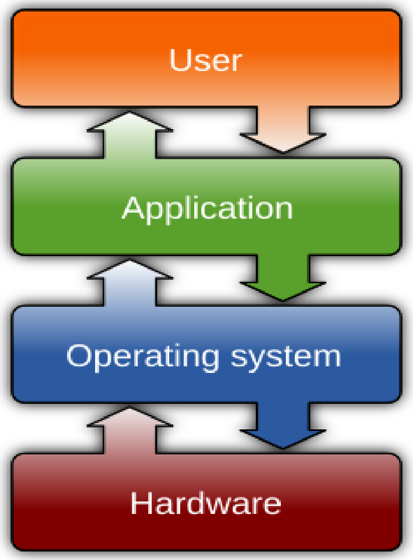
COMP 7003: Introduction to Information and Network Security

Linux Basics

Ashkan Jangodaz
British Columbia Institute of Technology

Operating System

- An **operating system (OS)** is system **software** that acts as an **intermediary** between computer **hardware** and user **applications**
  - Managing hardware resources (CPU, memory, storage, I/O)
  - Enforcing protection and isolation
  - Providing common services so programs can execute efficiently and safely.



Major Operating Systems

- **Unix Family**
  - UNIX, 1969
  - BSD, 1977
  - Linux, 1991
- **Personal Computers**
  - MS-DOS, 1981
  - Windows 1.0, 1985
  - Windows NT, 1993
  - macOS (Mac OS X), 2001

Major Operating Systems

- **Mobile Operating Systems**
  - Symbian, 1997
  - iOS, 2007
  - Android, 2008

Kernel

- The **kernel** is the **core of the operating system**.
  - Talks directly to **hardware**
  - Manages **CPU, memory, devices**
  - Enforces **security & isolation**
  - Provides **system calls** to programs
- **Responsibilities:**
  - Process scheduling
  - Memory management
  - File systems
  - Networking
  - Device drivers

Kernel

- Linux by itself:
  - Cannot run commands
  - Has no shell
  - Has no compiler
  - Has no utilities
- It just sits there waiting for programs.

Apps → system calls → KERNEL → hardware

User space

- User space is where ordinary programs run.
  - Shells, Services, Scripts, and Tools
- It includes:
  - Service managers (commonly systemd) that start and supervise long-lived processes
  - Schedulers (cron, timers) that trigger execution
  - Libraries and runtimes that applications depend on
  - The programs that implement the organization’s actual workload

Linux

- **Origins:**
  - Linux was created by **Linus Torvalds**
  - Originally a **hobby kernel** for x86 PCs
  - Inspired by **UNIX** design principles
  - An **open inspectable kernel**
- The dominant operating system for **servers**, **cloud infrastructure**, **supercomputers**, and **embedded devices**
- Important distinction:
  - **Linux** = kernel
  - **GNU/Linux** = kernel + userland tools

GNU

- **GNU** is a **collection of user-space tools and libraries**.
  - Created by the **GNU Project** (1983, Richard Stallman).
- GNU provides:
  - Shells (bash)
  - Core utilities (ls, cp, grep)
  - C standard library (glibc)
  - Compilers (gcc)
  - Build tools (make)
- Without GNU, Linux is mostly unusable for humans.

GNU/Linux

- GNU/Linux = Linux kernel + GNU userland
- This combination gives you:
  - A usable command line
  - POSIX tools
  - Development environment
  - A complete OS
- GNU is **not an operating system kernel** and **not a distribution** by itself.
- **“GNU’s Not Unix”** (a classic hacker joke)

Linux Distributions

- A **Linux distribution (distro)** is a **complete operating system package** built around the **Linux kernel**, combined with:
  - System libraries (e.g., GNU C Library)
  - Core user-space utilities (GNU tools)
  - A package manager (apt)
  - An init system (e.g., systemd)
  - …
- Linux itself is just the kernel.
  - A distribution turns the kernel into a usable operating system.

Linux Distributions

- Major Linux Distributions
  - **Ubuntu:** General-Purpose / Desktop-Friendly
  - **Debian:** Servers, security-critical systems
  - **Fedora:** Developers, security learning, modern systems
  - **Kali Linux:** Penetration testing, red teaming, security labs
  - **Raspberry Pi OS:** Embedded systems, IoT

Linux Distributions

redhat
MEPIS
turbolinux
LUNAR
EvilEntity
debian
Vine Linux
cAos/CentOS
MiniKazit
UTUTO
archlinux
m0n0wall
jamd
Knoppix STD
gentoo linux
DeLi Linux
Hiweed
amlug
slackware
yellow dog linux
Fedora
LPG
PLD
SLAX
COREL LINUX
Progeny
GEEXBOX
BIGLINUX
FREEDUC
Lycoris
EnGarde
Mandrakelinux
BeatrIX
Linspire
SuSE
中文延伸套件
YOPER
BearOps
ASPLINUX
kalango
Slackintosh
Frugalware
Foresight
Mint
PCLinuxOS
Haydar Linux
sabayon
ubuntu
JULEX
blag

Linux

- Linux is **Unix-like**, but it is **not UNIX**.
- **POSIX** is the reason UNIX and Linux feel the same.
  - **POSIX** (Portable Operating System Interface) is a **set of standards** that defines how an operating system should behave at the **API and shell level**. POSIX is defined and maintained by the IEEE.
  - POSIX defines **what must exist and how it must behave**.
  - **Examples of system calls and standard APIs:**
    - fork(), exec(), wait()
    - open(), read(), write()
    - signals, pipes, shared memory
    - threads (pthread)

Linux

- **Linux powers:**
  - The majority of **web servers**
  - Almost all **cloud platforms** (AWS, GCP, Azure internals)
  - **Routers, firewalls, IDS/IPS systems**
  - **Android** (Linux kernel)
  - …

Linux Command Line

- **The terminal**
  - A terminal is a text-based interface that allows users to interact with the operating system by entering commands and receiving output.
- **The shell**
  - A shell is a program that interprets user commands and executes them by communicating with the operating system.
  - A program running inside the terminal that reads what you type, runs commands, and prints results.

Filesystem

- Linux uses a single, unified filesystem tree starting at the root directory (/).
- There are no drive letters.
- Extra disks and partitions appear by being **mounted** somewhere inside this tree.
  - **Mounting** means making a filesystem available at a specific directory so the operating system can access its files.

Path

- A path tells Linux where something is in the filesystem.

Symbol	Meaning	Practical use
/	Root	Jump to a system folder: cd /etc
~	Home	Go to your home directory: cd ~
.	Current directory	Refer to “right here”
..	Parent directory	One level up: cd ..
-	Previous directory	Toggle back: cd -

Current working directory

- Your shell is always “standing in” one directory at a time. That directory is your current working directory (cwd).
  - It is the path in the shell that all relative paths are based on.
- If a command cannot find a file, first verify:
  - Where you are (pwd)
  - What exists there (ls)

PATH environment variable

- **PATH variables** (usually called the **PATH environment variable**) tell the operating system **where to look for executable programs** when you type a command.

PATH environment variable

- When you type a command like **ls** or **top**, the shell has to find the program to run.
  - It does this by searching the directories listed in the PATH environment variable, in order.
- **Practical implications:**
  - If the "command not found" message appears, the program may not be installed or may not be on your PATH.
  - If a command runs the “wrong” program, a different directory earlier in PATH may be providing a different executable.

PATH environment variable

- Useful commands:
  - echo $PATH: show the search path
  - which <command>: show which executable the shell will run
    - E.g., which python3

Some Commands (reference)

- **Help and discovery:**
  - man <command>
- **Diagnostics:**
  - uptime
  - top
  - ps aux
  - free -h
- **Filesystem capacity:**
  - df -h
  - du -sh <path>
  - du -h <path>

Some Commands (reference)

- **Where you are and how you move:**
  - pwd
  - cd, cd .., cd ~, cd –
- **See what exists:**
  - ls, ls -l, ls -a, ls -lh
- **Find things when you don’t know the path:**
  - find <path> -name <pattern>
- **Read safely:**
  - less <file>
  - more <file>
  - cat <file>

Some Commands (reference)

- **Search inside text:**
  - grep <text> <file>
- **Change files deliberately:**
  - cp <src> <dst>
  - mv <src> <dst>
  - rm <file>
  - rm -r <dir>

Read more in here: 03-linux-command-line-survival-guide.pdf

Some Other Commands (reference)

- **Network:** ssh, scp, ping, telnet, nslookup, wget
- **Shells:** BASH, TCSH, alias, watch, clear, history, chsh, echo, set, setenv, xargs
- **System Information:** w, whoami, man, info, which, free, echo, date, cal, df, free, man, info
- **Command Information:** man, info
- **Symbols:** |, >, >>, <, &, >&, 2>&1, ;, ~, ., .., $!, !:<n>, !<n>
- **Filters:** grep, egrep, more, less, head, tail
- **Hotkeys:** <ctrl><c>, <ctrl><d>
- **File System:** ls, mkdir, cd, pwd, mv, ln, touch, cat, file, find, diff, cmp, /net/<hostname>/<path>, mount, du, df, chmod, find
- **Line Editors:** awk, sed
- **File Editors:** vim, gvim, emacs –nw, emacs