

# launch \_code

John Woolbright, Technical Curriculum Developer

June, 15, 2022

# Linux Terms

- Operating System
- User space applications
- Kernel
- GNU
- Shell
- Terminal
- File System



# Operating System

- An operating system contains two major components:
  - a **kernel**
  - collection of **userspace applications**.



# End User

- Any person using a computer is considered an end user.
- An end user uses a computer to achieve some goal which varies depending on their specific needs and wants.



# OS Servers the End User

- The OS is responsible for allowing the end user to interface with a computer by handling two responsibilities:
- Managing the physical components of the computer (kernel)
- Providing interfaces to the computer that can be used by the end user (user space applications)
- End users **only** interact with end user applications.



# User Space Applications

- User space applications are all of the programs on a computer that the end user interacts with directly.
- Every action you have ever completed on a computer is considered an user space application.
- Creating a word document, Writing/Compiling/Executing code, watching movies, editing photos, browsing the internet, sending emails are all user space applications.
- Even the Graphical User Interface (GUI) you use to select your documents and applications is itself a user space application!



# OS Provides the User Interface

These applications have been created with people in mind to achieve a specific goal.

In essence one of the primary responsibilities of an OS is to provide a collection of programs that allow interfacing with a computer easy and intuitive.



# Under the Hood

Additionally, under the hood the OS is also responsible for:

- Providing to a monitor
- Managing the mouse location
- Determining mouse and keyboard click events
- Opening programs by allocating memory (RAM)
- Writing to the disk when files are created or modified
- Managing CPU access to all running processes.

All of these under the hood actions are responsibilities of the kernel.





# Kernel

Software that manages computer operations including all of the physical hardware of a machine.



# Kernel Example

Consider all the actions needed to open a word document. From a user perspective we simply double click the file in our file manager or on our desktop.

The computer has to determine which file was clicked based on the mouse position, find that file from hard disk storage, open the contents of the file in memory (RAM), create and send the display of the file to the monitor.

All of these under the hood tasks would burden the user and are therefore handled by the kernel.



# Linux Kernel

The Linux kernel is an open source and actively developed monolithic kernel. It was originally created by Linus Torvalds, but has since received commits from 1000s of contributors.

GitHub repo: <https://github.com/torvalds/linux>

The Linux Kernel Documentation:  
<https://www.kernel.org/doc/html/latest/>



# GNU

GNU is a recursive acronym that stands for **G**NU is **N**ot **U**nix.

GNU offers a huge collection of free software, that is commonly bundled with the Linux kernel to create a full fledged operating system.

Many of the tools we will use in this class are created and maintained by GNU.

GNU Homepage: <https://www.gnu.org/>

GNU Package Listing: <https://www.gnu.org/software/software.html>



# Shell

A **shell** is software that is end user interactive.

It's purpose is to serve as the interface between an end user and the kernel.

There are both text-based shells (Bash shell) and graphical shells (gnome-shell).



# Terminal

A terminal is the graphical application that powers a text-based shell.

Before graphical shells (like the Xerox Neptune Directory Editor, Windows 1, Apple Lisa & Macintosh, etc) there were only text-based shells.



# Terminal Emulators

Terminals which were the monitors connected to these computers only provided a text-based interface.

In our modern era of computing we mostly use **terminal emulators**. They emulate what the original text-based only terminals felt like, but provide additional features, and are run as an application within a graphical shell.



# Terminal Emulator Examples

If you have used **git-bash**, the **bash shell**, **powershell**, or the **Windows cmd line** you have used a terminal emulator.

We will be spending the majority of our time in this class using a terminal emulator displaying the **bash shell**.





# File System

The filesystem of a computer is defined and managed by the kernel.

The Linux filesystem contains a **root** directory (“/”) which holds all of the files and directories on the computer.



# Root Directory

Many of the subdirectories inside of the **root** directory contain configurations, data, and tools that are used directly by the operating system.

The end user can see, and possibly edit or delete these files which are crucial to running the operating system.

To mitigate the risk of having end users mess around with kernel level data Linux has also designated **userspace**.



# Userspace

**Userspace** is a term that describes the areas of the Linux filesystem that contain end user files and directories.

Userspace is usually found in a couple of different locations and the location varies by Linux distribution. In Ubuntu, which is the Linux distribution we will be using, it is predominantly found in the `/usr` and `/home` directories.





launchcode.org

