Operating Systems

OS Knowledge, Shell, VM and Containers

SoftUni Team

Technical Trainers

Software University

SoftUni

Software University

https://about.softuni.bg/

Table of Content

Have a Question?

Software University

sli.do

#qa-fund

Operating Systems Overview

Purpose and Structure

What is Operating System?

- An **operating system (OS)**, manages other application programs in a device
- The OS is loaded into a device through a process called **booting**
- It enables applications to **interact** with the device's **hardware**
- Applications make requests for services through a defined interface called an **application program interface (API)**
- At **least one OS** must be **installed** in a device to run basic programs, e.g. web browsers

OS Main Functions

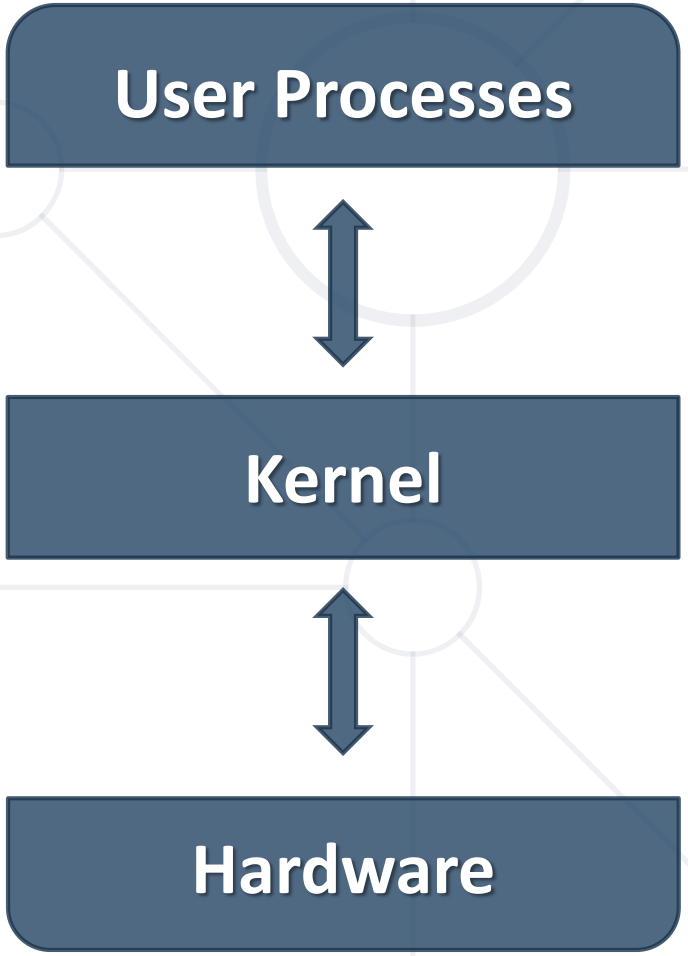
- **Booting** - the process of turning on the device and powering up the system
- **Memory management** - controls and coordinates the computer applications while allocating space for programs
- **Loading and execution** – load / start up, a program and execute it, so that it opens and runs
- **Data security** - OS includes features that keep data and computer programs secure.
- **Disk management** - manages all the drives installed, including hard drives, optical disk drives, and flash drives. Also used to divide disks, format drives, etc.

OS Main Functions

- **Process management** - Allocates resources to different computer processes, allows processes to share information, protects, and synchronizes them
- **Device controlling** – Controls access to devices like removable devices, CD/DVDs, USBs, and more
- **Printing controlling** - takes control of printers connected and manages the printing process
- **User interface (UI)** - allows users to interact with the computer by entering and receiving information through typed commands, code, and other formats

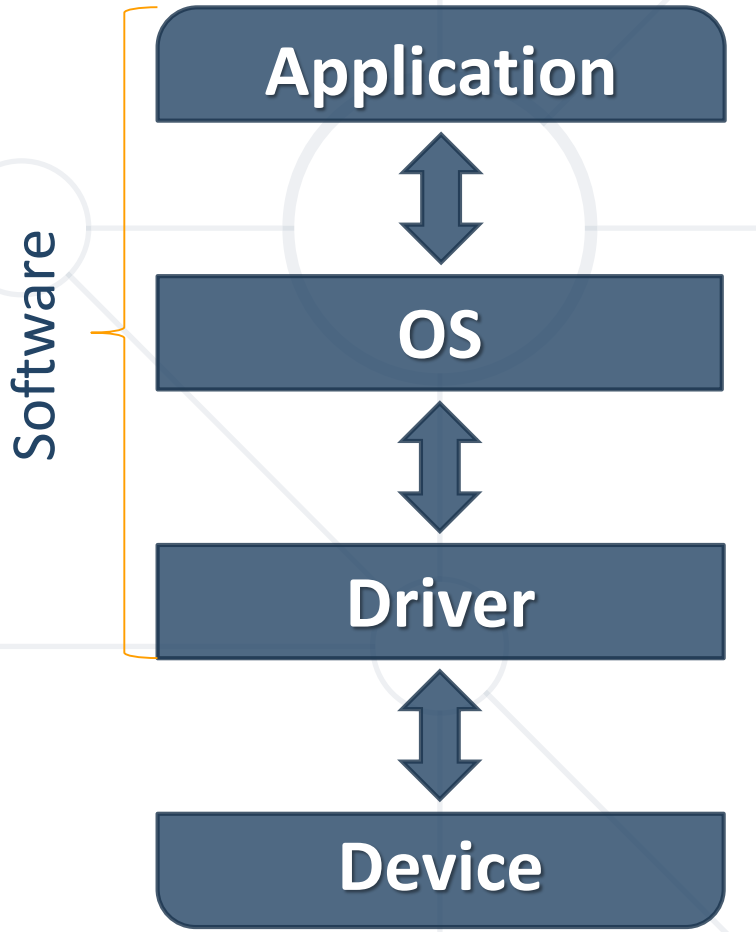
Kernel

- **Core component** of an operating system
- **Bridges hardware** and **software** components
- **Translates** user queries into machine language
- **Facilitates communication** between different system components
- **Provides complete control** over the system
- **Always resident** in memory
- **Essential** for running any operating system



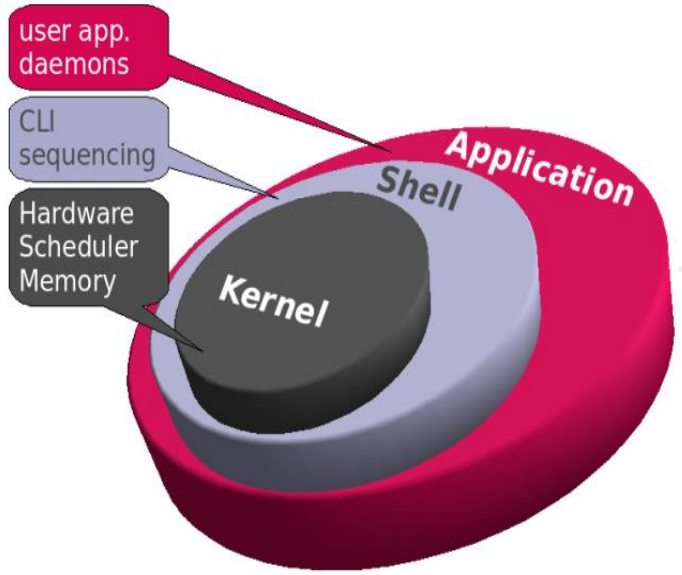
Drivers

- **Set of files** that enables hardware components to function
- It **communicates** with the computer's operating system to **manage** and **enable hardware components** or peripherals to operate properly
- Drivers are software programs **without a user interface (UI)**
- All hardware components **require a driver**



Shell

- **Interface** to the operating system
- Outermost layer of the OS, located **between the kernel and the user**
- Incorporates a **programming language** to control processes and files
- **Two types** of shells:
  - **Command-line shells -** knowledge of commands, calling syntax, and concepts about the shell-specific scripting language (e.g., bash)
  - **Graphical shells** - easy to use
- Most GUI-enabled operating systems provide **CLI shells** for **advanced tasks**

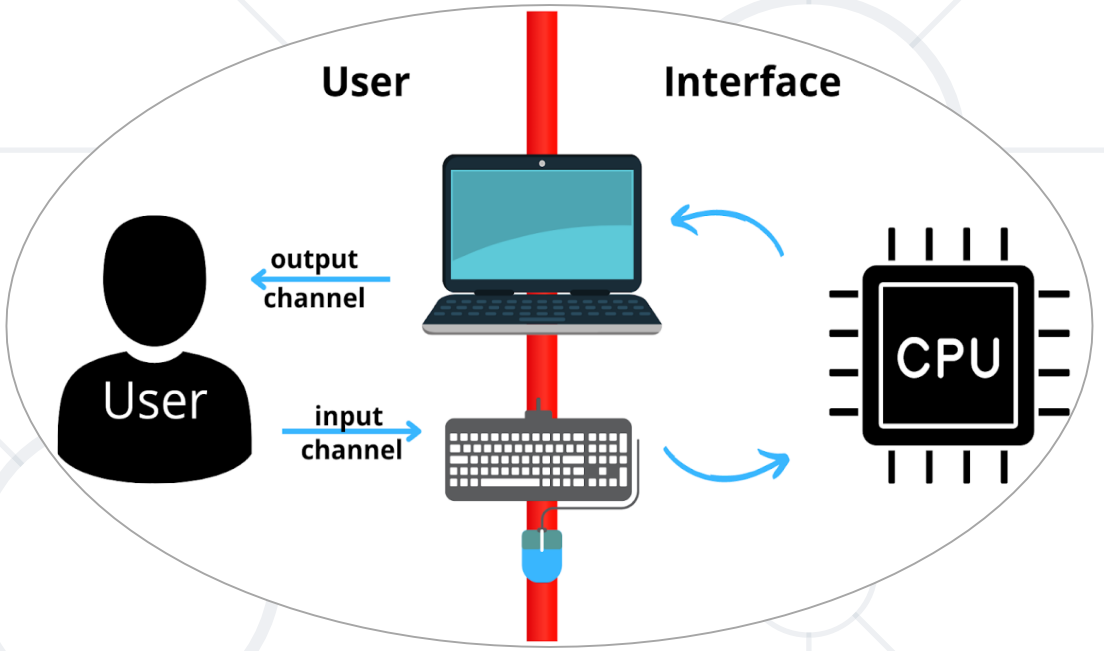


Users in Operating Systems

- Software User is a **single user** of the Software on a **single computer** or within a directory environment
- An operating system is a **construct** that allows the user's application **programs to interact** with the system's hardware
- A user often has a **user account** and is identified to the system by a **username.**

Users in OS

- **User accounts** allow access to a system's resources.
- **Authentication** is the process of verifying a user's identity through credentials like passwords.
- **Authorization** determines what resources a user can access based on their authenticated identity.
- User accounts are **important** for **accounting**, **security**, **logging**, and **resource management**.



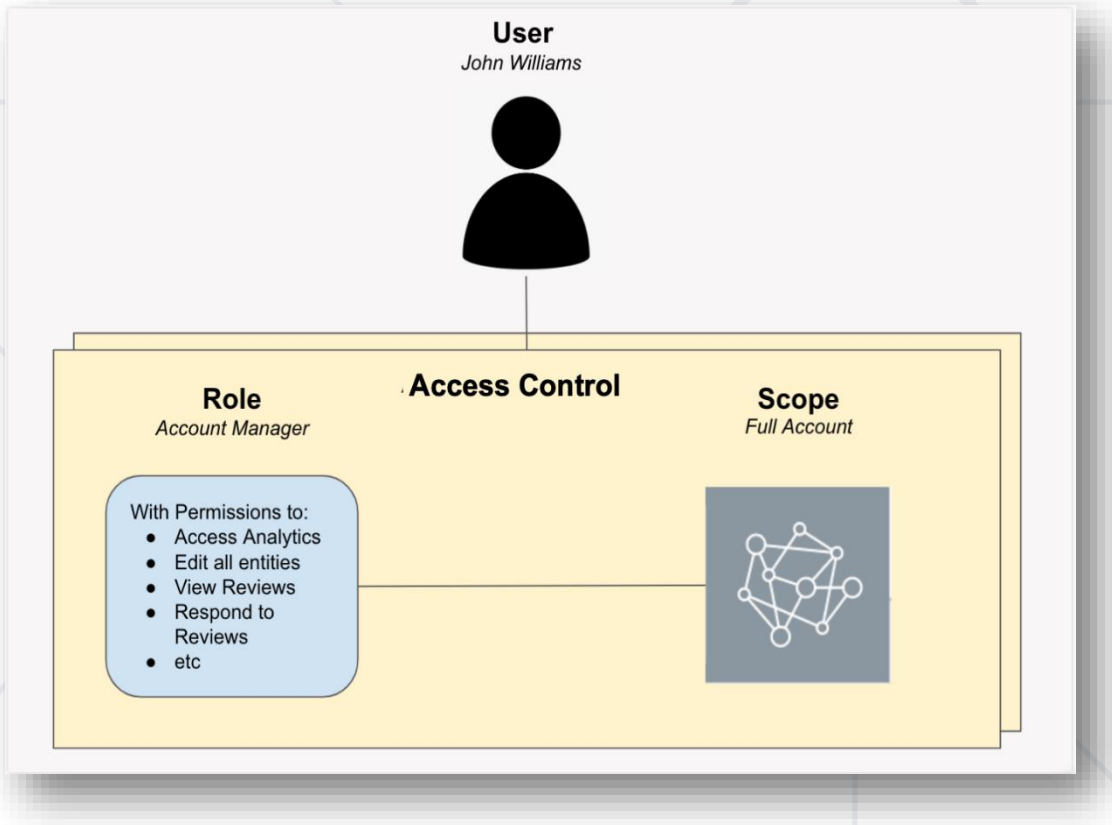
Authentication vs. Authorization

- Authentication **verifies the identity** of a user or service
- Authentication answers the question:
  - **Who are you?**

- Authorization determines the **user's access rights**
- Authorization answers the question:
  - **What are you allowed to do?**

User Permissions

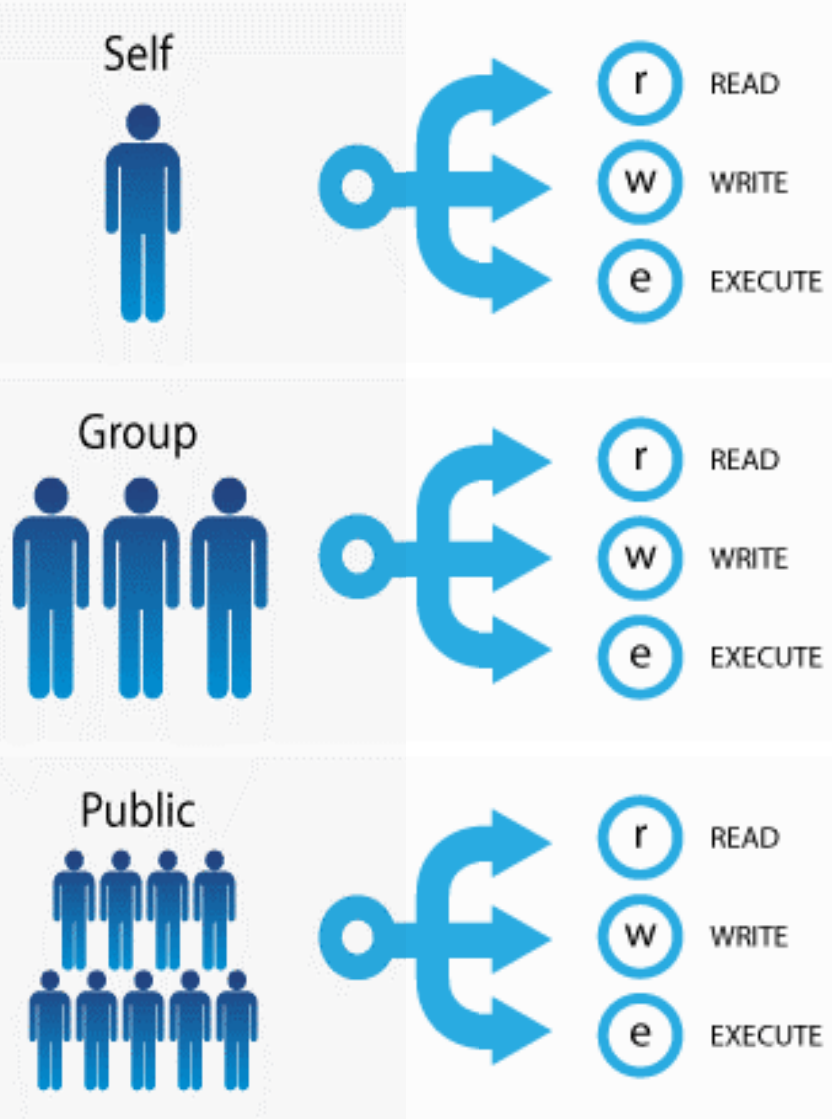
- OS can **control the use of system and network resources** through the interrelated mechanisms of **authentication** and **authorization**
- The OS **determines** if an **authenticated user** has the **correct permissions** to access a resource using built-in authorization and access control technologies



User Roles

- User Roles are **permission sets** that control access to areas and features within the Professional Archive Platform.
- Each User account requires a Role assignment.
- The set of roles are **Administrator**, **Publisher**, **Editor**, **Designer**, and **Viewer**.
- Data & Insights users can also be assigned permissions for individual datasets.

Access Permissions in OS

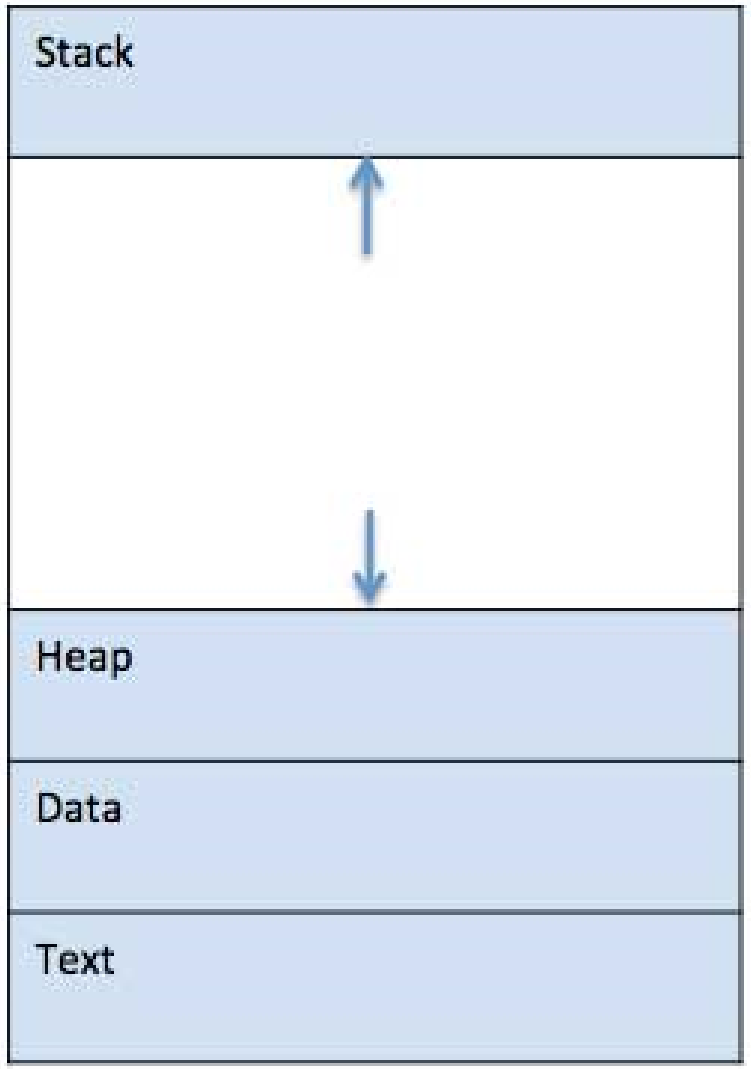


- **Access permissions** determine a user's ability to perform a specific action, or access a feature or object
- Set access permissions to specify which **users**, **groups**, or **roles** can **access** your **content**
- The most common permissions are **read**, **write**, **delete**, and **execute**

Processes in OS

- A **process** is a **program in action**
- It's the basic unit of work in a system
- Unlike a program, which is **passive**, a process is an **active entity**
- For example, when you open a browser to search the web, that's a process

Processes in OS



- When a **program** is **loaded** into the **memory** and it **becomes a process**, it can be divided into four sections – **stack**, **heap**, **text** and **data**
- The image shows a simplified layout of a process inside the main memory.

Processes in OS – Component & Description

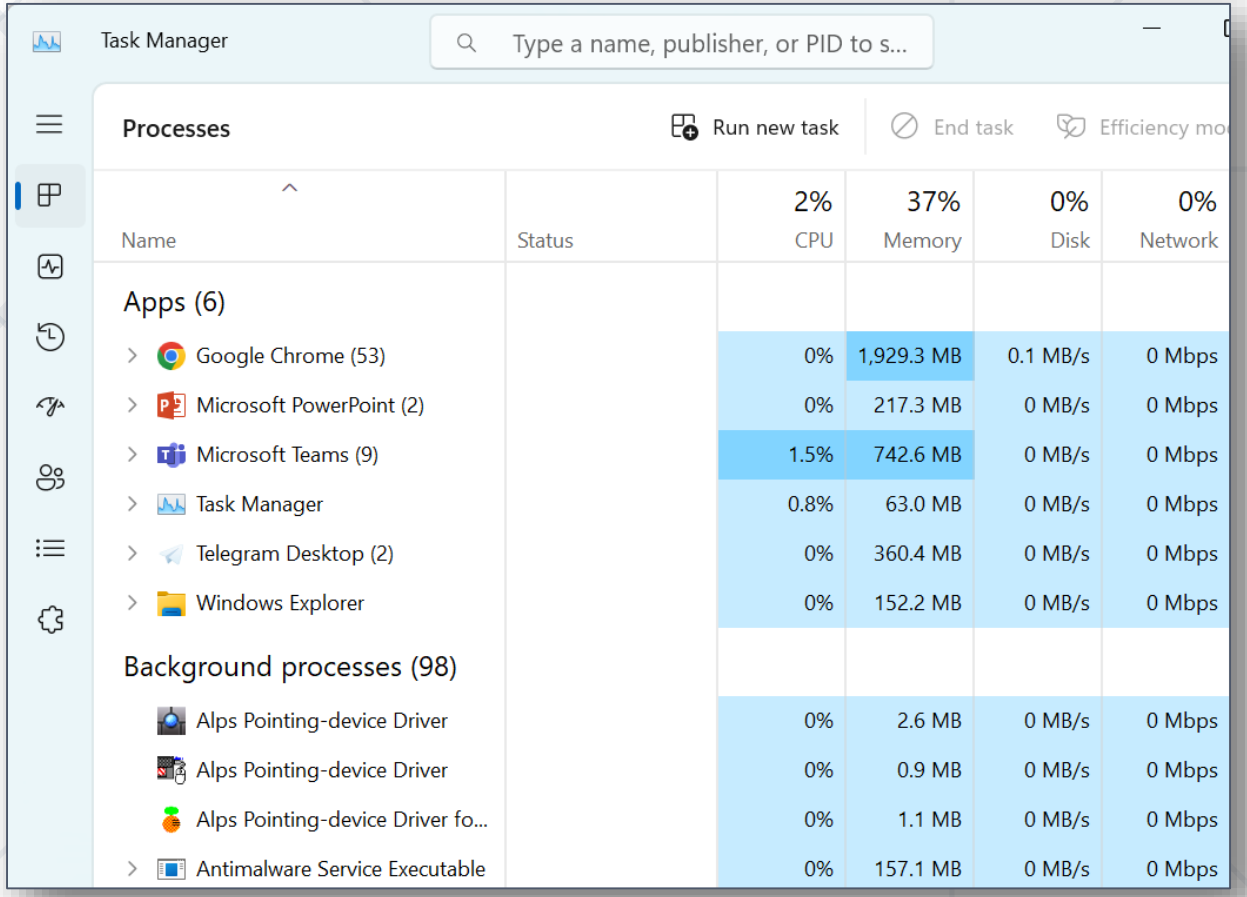
- **Stack -** contains the temporary data such as method/function parameters, return address and local variables
- **Heap -** dynamically allocated memory to a process during its run time.
- **Text -** the current activity represented by the value of Program Counter and the contents of the processor's registers
- **Data -** contains the global and static variables

Task Manager

- In OS, a task manager is a **system monitor program** used to provide information about the **processes and applications running on a device**, and the general status of the device
- Some implementations can also be used to **terminate processes** and applications, and change the processes' **scheduling priority**
- Task managers can display **running services** (processes) and those that were **stopped**

Task Manager Overview

- Open **Task manager** in Windows:
  - **CTRL + Alt + Delete**
  - Select Task Manager from the menu
- Task Manager allows the system to be **shut down** or **restarted**, when it is otherwise **busy** or **unresponsive**



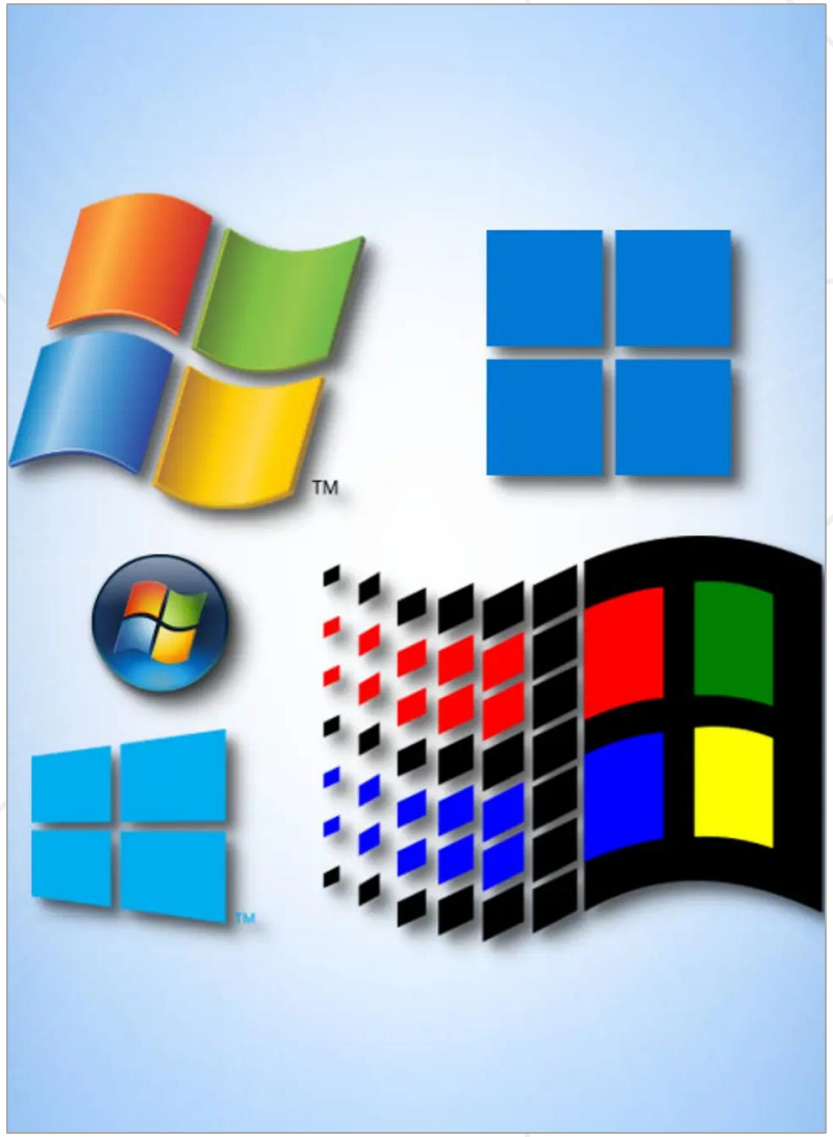
Operating Systems

Different Types

Most Popular Operating Systems

- **Five main types** of operating system
  - Microsoft Windows
  - Apple macOS
  - Google's Android OS
  - Apple iOS
  - Linux Operating System

Microsoft Windows

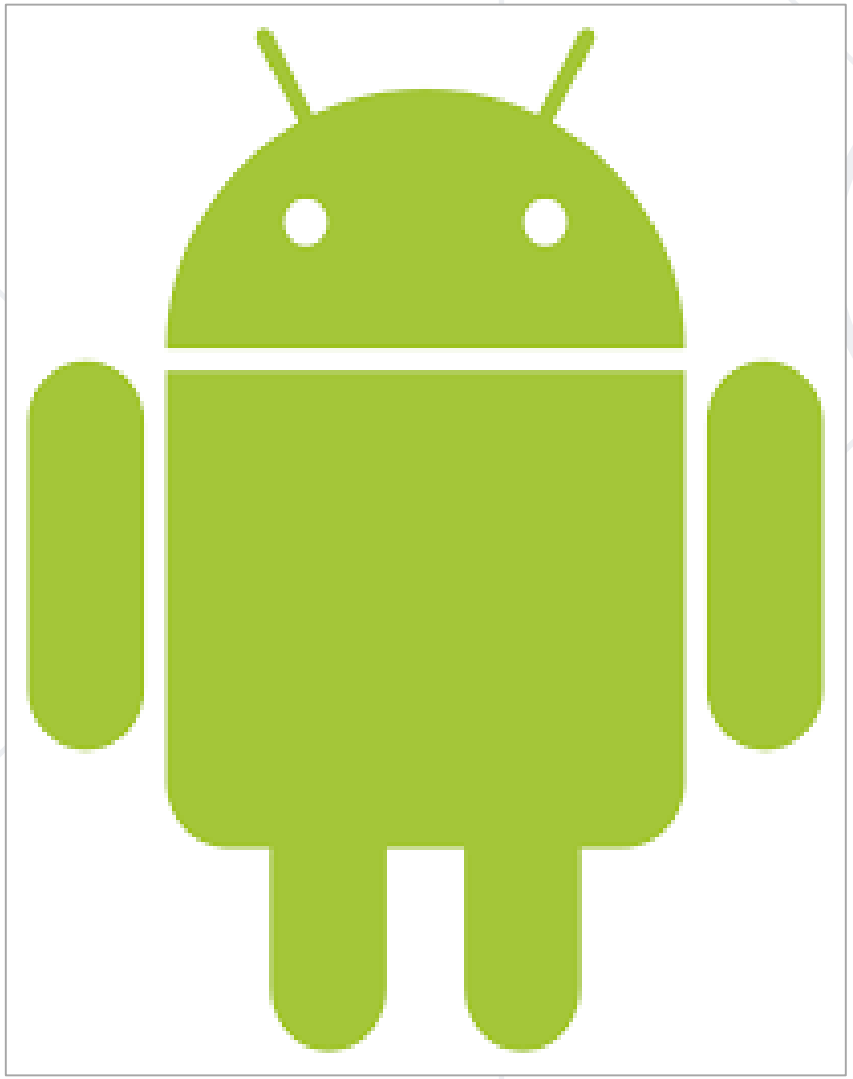


- Operating system developed by **Microsoft** that has been around since the 1980s
- Several versions and updates of Windows, including Windows 95, Windows Vista, Windows 7/8/10/11, and more
- One of the **most popular** operating system types and is typically **preloaded** on new PC hardware

Apple macOS

- **Apple** and **Macintosh** computers run on **macOS** and **OS X**, which are proprietary operating systems developed and marketed by Apple Inc
- **macOS** is a **Unix-based** operating system that was first released over 20 years ago
- In 2020, Apple began **transitioning** to its own 64-bit ARM-based Apple M series processors on its latest Macintosh computers

Android OS

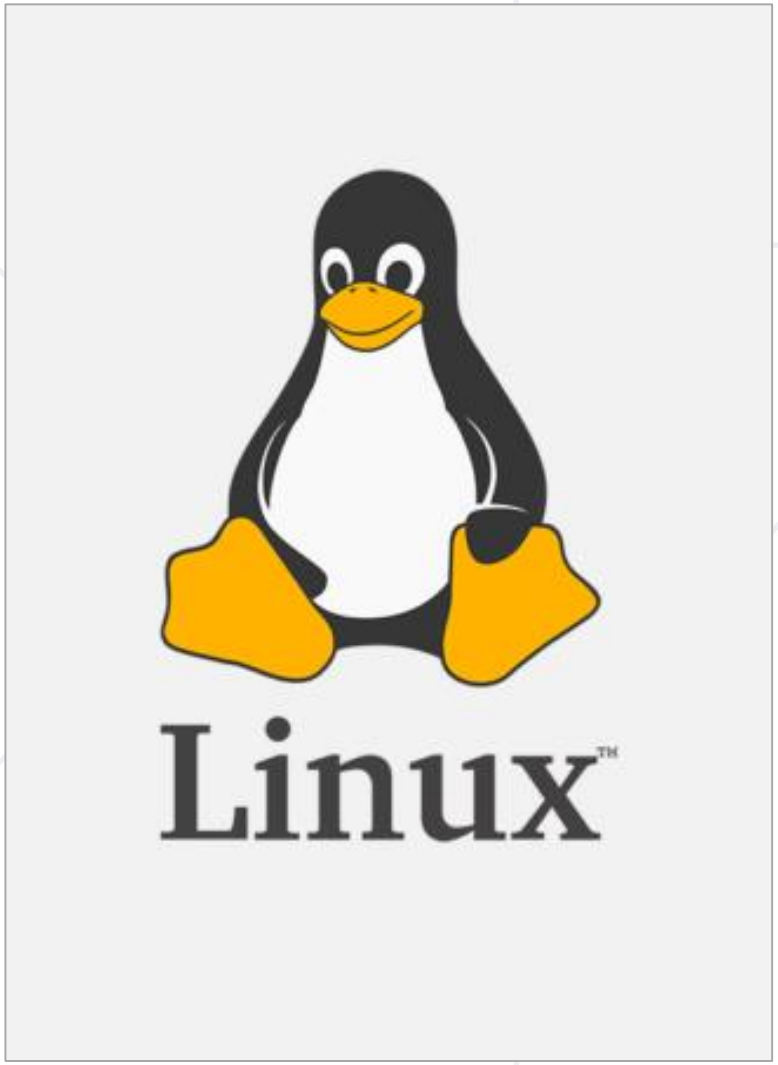


- Mobile operating system **designed** for **touchscreen** mobile devices
- **Based** on a **modified version** of the **Linux kernel** and other open-source software
- Core operating system is called **Android Open Source Project (AOSP),** free and open-source software, primarily licensed under the Apache License
- **Developed** and **maintained** by **Google**

Apple iOS

- **Mobile operating system** developed by Apple Inc. exclusively for its hardware devices, including the **iPhone**, **iPad**, and **iPod Touch**
- User interface is based on **direct manipulation** and uses multi-touch gestures like **swipe**, **tap**, **pinch**, and **reverse pinch** to interact with the system
- **Interface control** elements include **sliders**, **switches**, and **buttons**, used to control various **settings** and **features** on the device

Linux



- **Open source family** of operating systems
- It is **not proprietary software**, which means anyone can modify and distribute it
- Linux's **popularity** comes from its ease of customization
- It offers a **variety of options** for those who understand how to use it

Virtual Machines & Containers

Remote Instances & Emulators

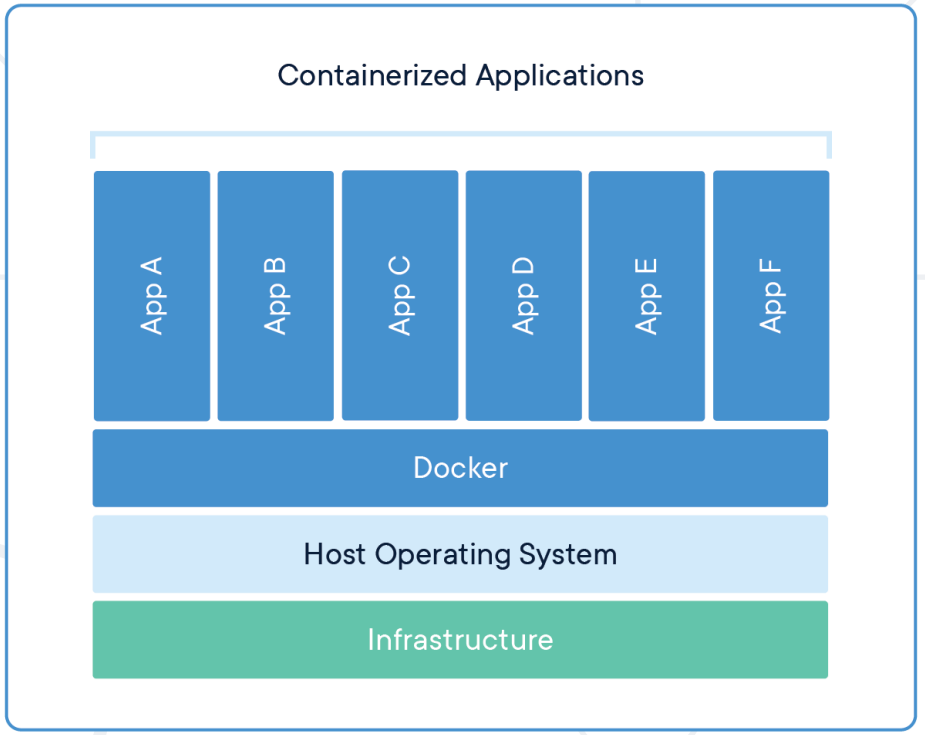
Virtual Machines

- A **virtual machine (VM)** is a software-based computer resource
- Digital versions of physical computers that **can run programs** and **operating systems**, **store data**, **connect to networks**, and other computing functions
- **Require maintenance**, such as updates and system monitoring
- Useful for **running multiple operating systems** on a single physical computer or for testing software in a controlled environment

Containers

- **Packages of software** that can run in **any environment**
- They **virtualize the operating system** and can run on private data centers, public clouds, or personal laptops
- Provide a solution to the problem of **running software** reliably **across different computing environments**
- **Isolate software** from its environment, ensuring consistent performance across different environments
- **Work for** both **Linux** and **Windows**-based applications, making containerized software platform-agnostic

OS inside another OS – e.g. Docker Containers

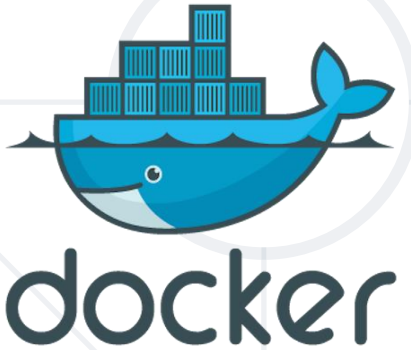


- A Docker container image is a lightweight, standalone, executable package of software
- It contains everything needed to run an application, including code, runtime, system tools, system libraries, and settings
- When a Docker container image is run on the **Docker Engine**, it becomes a container.
- Containers are isolated from the host system and from other containers, making them an efficient and secure way to run applications.

Docker Overview

Software University

- An open platform for **developing**, **shipping**, and **running** applications.
- **Separates applications** from **infrastructure** for faster delivery.
- **Manages infrastructure** in the same way as applications for easier deployment and scaling



- Docker's methodologies **reduce the delay** between writing code and running it in production for faster innovation
- Provides a **range of tools and services** for streamlined containerized application development and deployment

Remote VM Instances – e.g. Docker Playground

- Containers allow for **customizable** and **replicable instances** of an application without interfering with anything else on a user's system
- Play with Docker / Docker Playground is an interactive and fun way to learn Docker
- It lets you run **multiple versions** of Docker on the same machine to test different software versions without rebuilding

OS Emulators

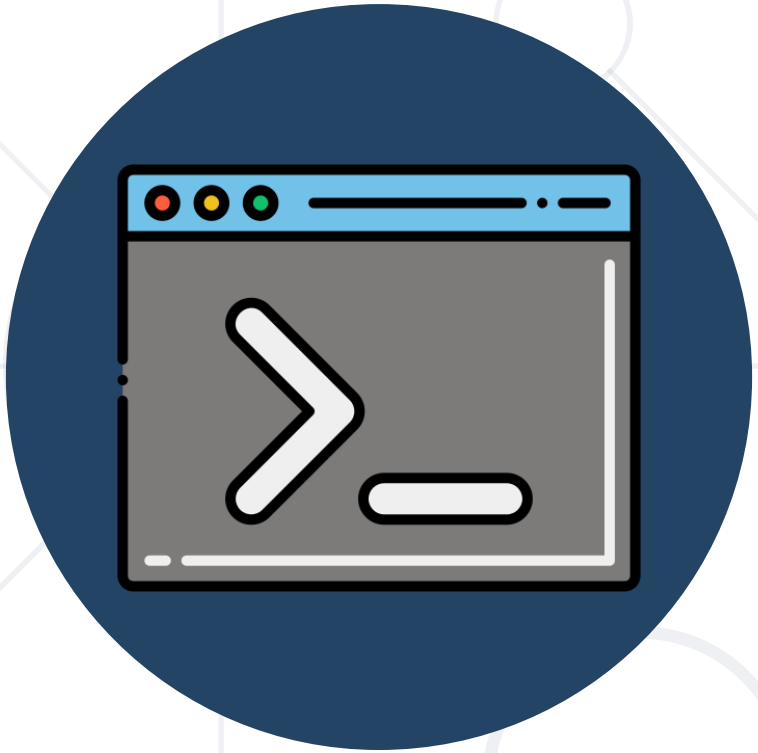
- **Emulation** is the use of one program or device to **imitate another** program or device's behavior
- It can be used to **run** an operating system on hardware it **wasn't originally designed for**
- In **server virtualization**, emulation is similar to a virtual environment, which can be called a partition, guest, instance, or container
- **Hardware emulation** is the use of hardware to imitate another hardware device's function, usually for connecting devices together

BrowserStack – App & Browser Testing

Software University

- BrowserStack provides **manual** and **automated** online mobile testing for websites and apps
- BrowserStack Live offers **3000+ device-browser-OS** combinations for testing
- QA can choose from a **wide range of devices** to run their website during testing
- BrowserStack Automate supports **automation frameworks** and tools like Cypress, Selenium, Puppeteer, Appium, and Playwright
- It also supports **popular programming languages** like C#, Python, Java, and JavaScript

BrowserStack



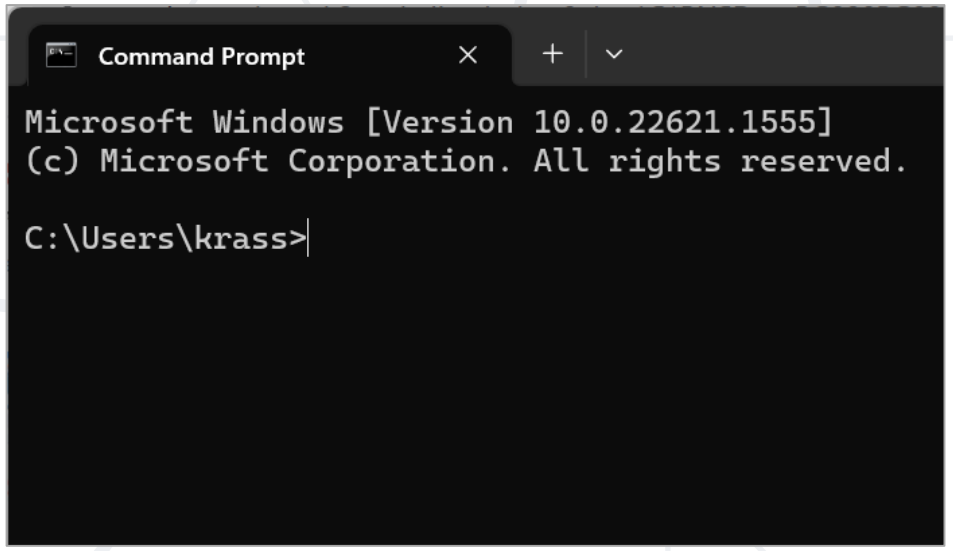
Shell & Shell Commands

Command Execution

Shell - Navigating Files & Directories

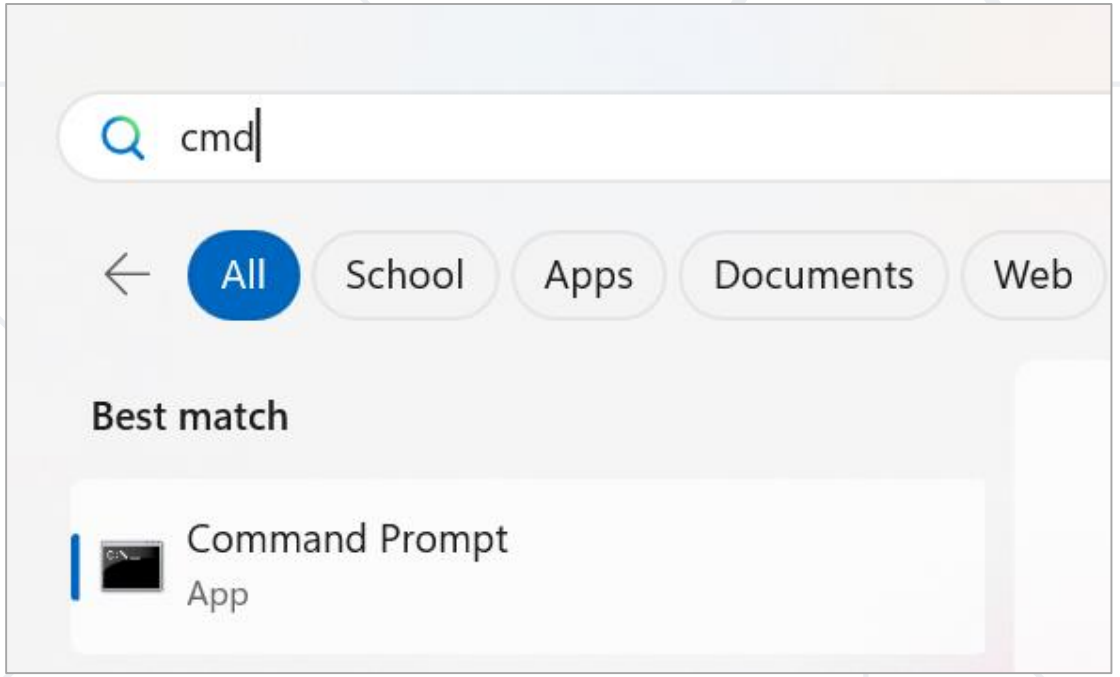
Software University

- **The file system** is the part of the operating system responsible for managing files and directories
- It **organizes data** into files, and directories (also known as folders), which hold files or other directories
- Various commands are used to **create**, **inspect**, **rename**, and **delete** files and directories
- To explore these commands, use an **open shell window** or **terminal**



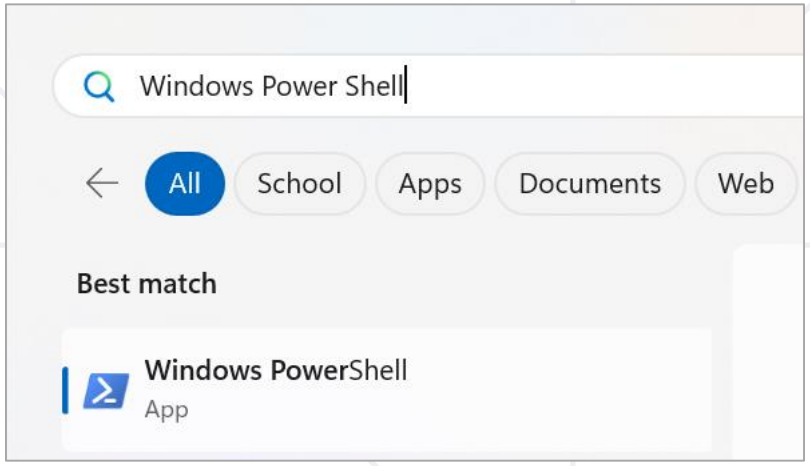
Opening Command or Shell Prompt

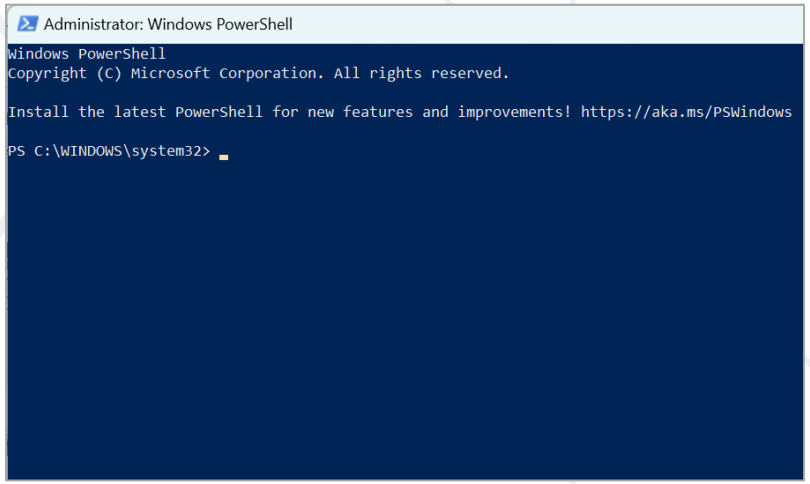
- Click Start > Run or press Windows + R key.
- Type cmd
- Click on Command Prompt



Windows Power Shell

- **PowerShell** is an **automation engine** and **scripting language** developed by Microsoft for IT professionals to configure systems and automate tasks
- Commonly used to **improve efficiency**, **reduce manual errors**, and **build**, **test**, and **deploy solutions** in CI/CD environments





Commands: ls & dir

- ***ls*** is the traditional UNIX method of viewing the files in a directory
- In Linux, the ***ls*** command stands for list files
- ***dir*** is the windows command prompt equivalent
- in Windows, the ***dir*** command means produce a directory listing

Commands: cd & pwd

- ***cd*** changes the current working directory to the specified drive
- The ***cd*** command can be used to:
  - change into a subdirectory
  - move back into the parent directory
  - move all the way back to the root directory
  - move to any given directory

- ***pwd*** prints the directory you are currently in, it does nothing else
- ***pwd*** does not take any arguments
- ***pwd*** is equivalent to typing ***cd*** without arguments; both display the name of the current working directory

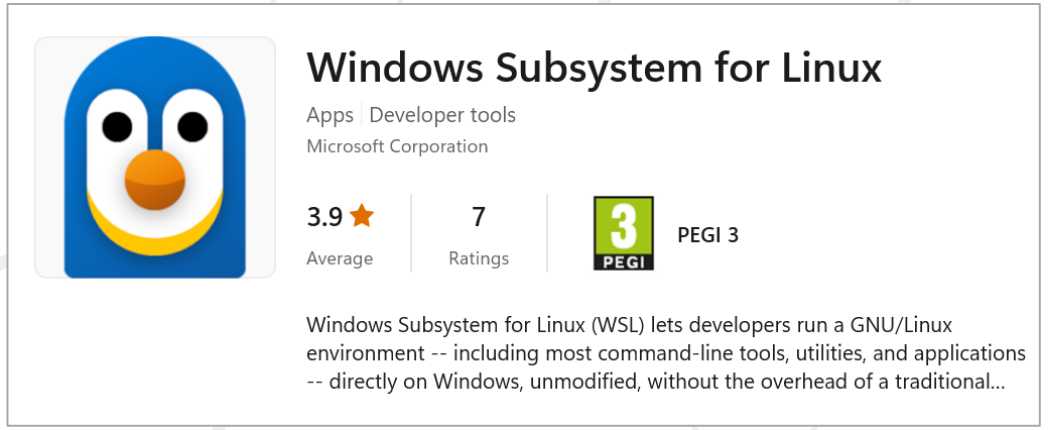
Commands: cat

- If you have worked in **Linux**, you surely have seen a code snippet that uses the ***cat*** command, the most **universal** and **powerful** tool
- It is considered to be one of the most frequently used commands
- It can be used to:
  - display the content of a file
  - copy content from one file to another
  - concatenate the contents of multiple files
  - display the line number
  - display $ at the end of the line, etc.

Can I Run Linux Commands on Windows PC

Software University

- You can run Linux commands in Windows 10 and 11 without setting up a virtual machine
- You don't have to configure a virtual machine using VMWare to use Linux
- It is possible through the Windows Subsystem for Linux (WSL)



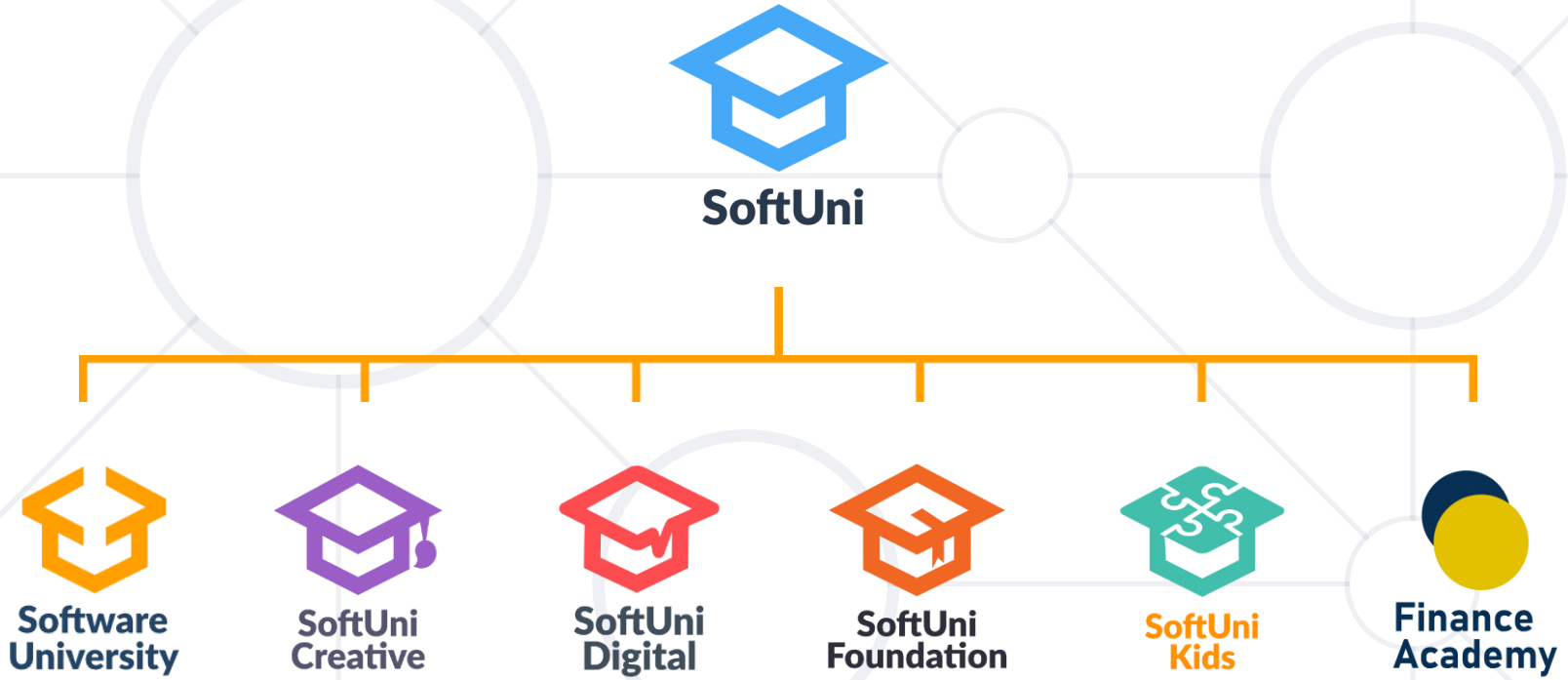
Run Linux distributions with WSL

- WSL supports running as many different Linux distributions
- This can include choosing distributions from the Microsoft Store
- From Windows Command Prompt or PowerShell, open your default Linux distribution inside your current command line, by entering: ***wsl.exe***
- WSL also allows to run Linux command-line tools and apps alongside Windows command-line, desktop and store apps, and to access Windows files from within Linux

Summary

- **Operating Systems** Overview
- **OS Examples**
- **Virtual Machines & Containers**
  - OS inside another OS
  - Remote VM instances
  - OS Emulators
- **Shell & Shell Commands**

Questions?



SoftUni Diamond Partners

Software University

SUPER HOSTING .BG

Coca-Cola HBC Bulgaria

POKERSTARS
POKER | CASINO | SPORTS
a Flutter International brand

INDEAVR
Serving the high achievers

AMBITIONED

DRAFT KINGS

SOFTWARE GROUP

createX

Postbank
Решения за твоето утре

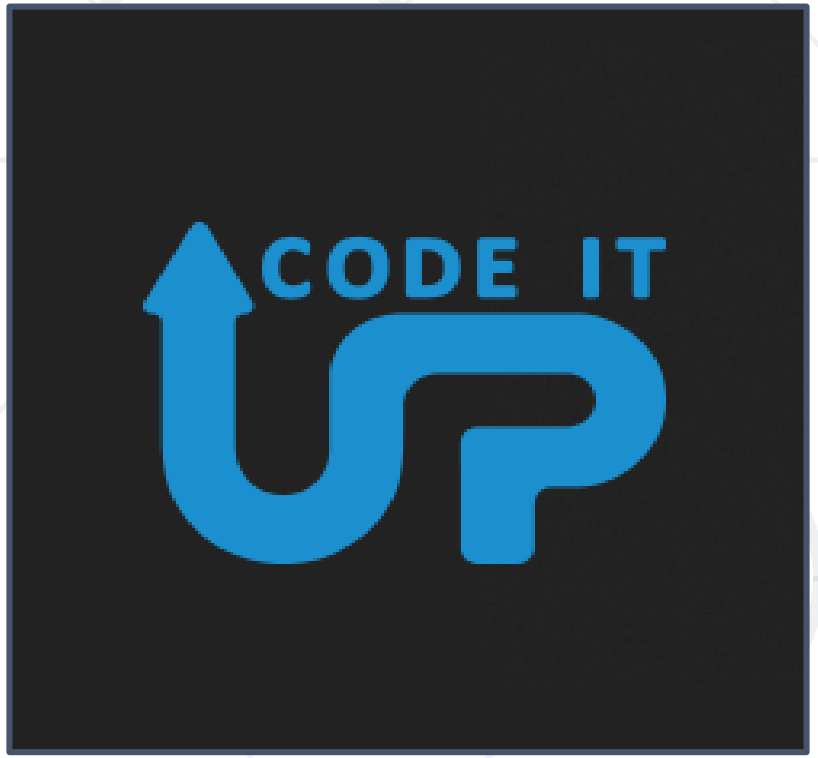
BOSCH

DXC TECHNOLOGY

SmartIT

Educational Partners

Software University



Trainings @ Software University (SoftUni)

- Software University – High-Quality Education, Profession and Job for Software Developers
  - softuni.bg, about.softuni.bg
- Software University Foundation
  - softuni.foundation
- Software University @ Facebook
  - facebook.com/SoftwareUniversity
- Software University Forums
  - forum.softuni.bg

License

- This course (slides, examples, demos, exercises, homework, documents, videos and other assets) is **copyrighted content**
- Unauthorized copy, reproduction or use is illegal
- © SoftUni – https://about.softuni.bg/
- © Software University – https://softuni.bg