

# Operating Systems

OS Overview, Linux Shell, VM and Containers



SoftUni Team  
Technical Trainers



Software University

<https://about.softuni.bg/>

# Table of Content

## 1. Operating Systems (OS) – Overview

## 2. OS – Examples

- Windows, Linux, macOS, Android, iOS

## 3. Virtual Machines and Containers

- Virtualization: OS inside another OS
- Remote VM Instances
- OS Emulators

## 4. Linux Shell and Shell Commands



# Have a Question?

[sli.do](https://sli.do)

**#qa-fund**



# **Operating Systems (OS) – Overview**

OS Purpose and Structure

# What is Operating System (OS)?



- The **operating system (OS)** manages applications (processes), users, file system and resources in a device
- The OS is loaded into a device through a process called **booting**
- OS enables applications to **interact** with the device's **hardware** and software **resources**
- 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 browser, file explorer, video player

# OS Main Functions

- **Booting** – turning on the device and loading the OS
- **App loading and execution** – load and run programs (processes, apps), start / view / pause / terminate apps
- **Process management** – allocates resources to OS processes, share data between processes, protects, and synchronizes them
- **Memory management** – controls and coordinates the memory allocation for the applications running in the OS
- **Disk management** – manages storage (hard drives, SSD disks, optical disk drives, flash drives) and file systems



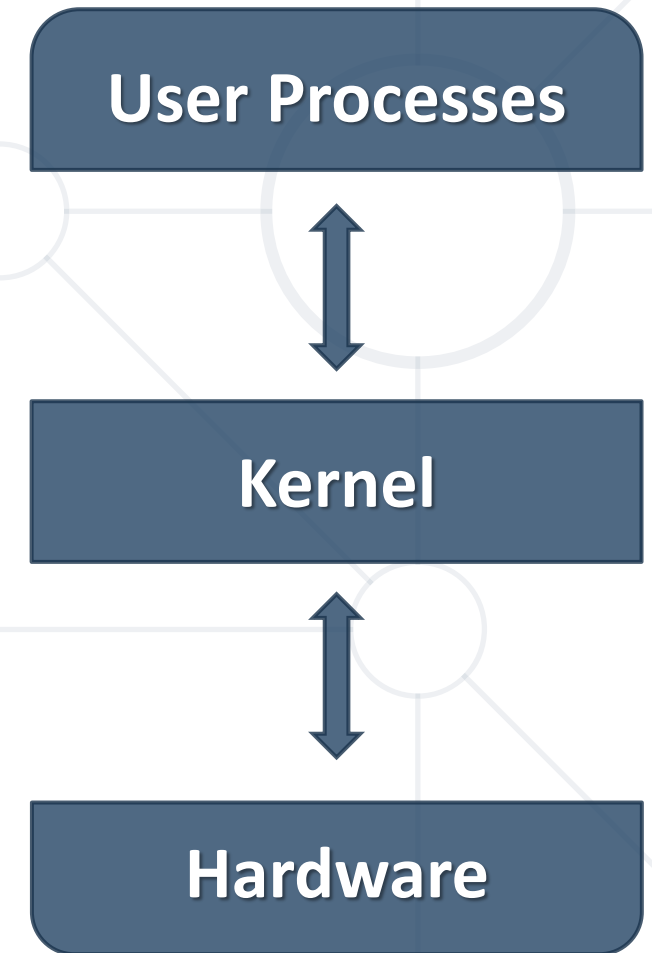
# OS Main Functions (2)

- **Device controlling** – controls the access to **physical devices** (like disk drives, CD/DVDs, USBs) and **virtual devices** (like random)
- **Networking** – communication over the network and Internet
- **Printing controlling** – takes control of **printers** connected and manages the printing process
- **User interface (UI)** – provides UI for the users to interact with the computer by commands or visual UI elements
- **Data security** – isolate apps, users and files to keep data secure (e. g. using file system / resource permissions)





- **Kernel** == core component of the OS
- The OS "heart" – **bridges hardware** and **software** components
- **Facilitates communication** between different system components
- **Provides complete control** over the system
- **Always stays resident** in memory
- **Essential** for running any operating system





# OS Kernel – Example

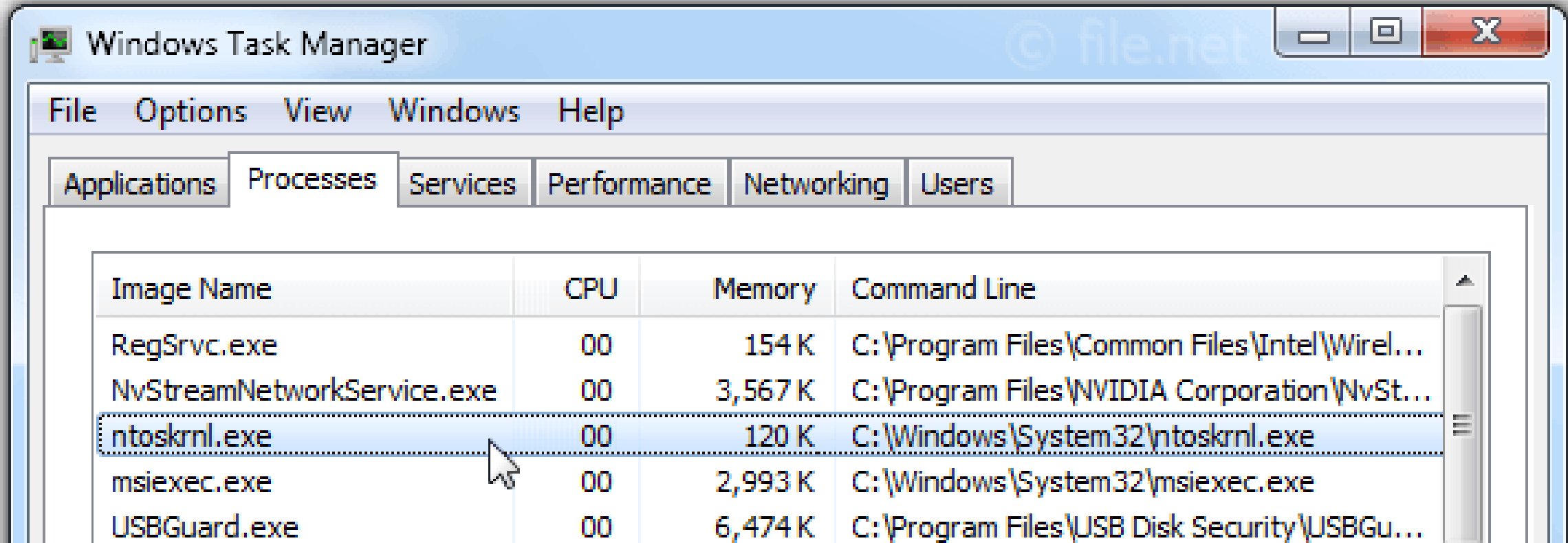
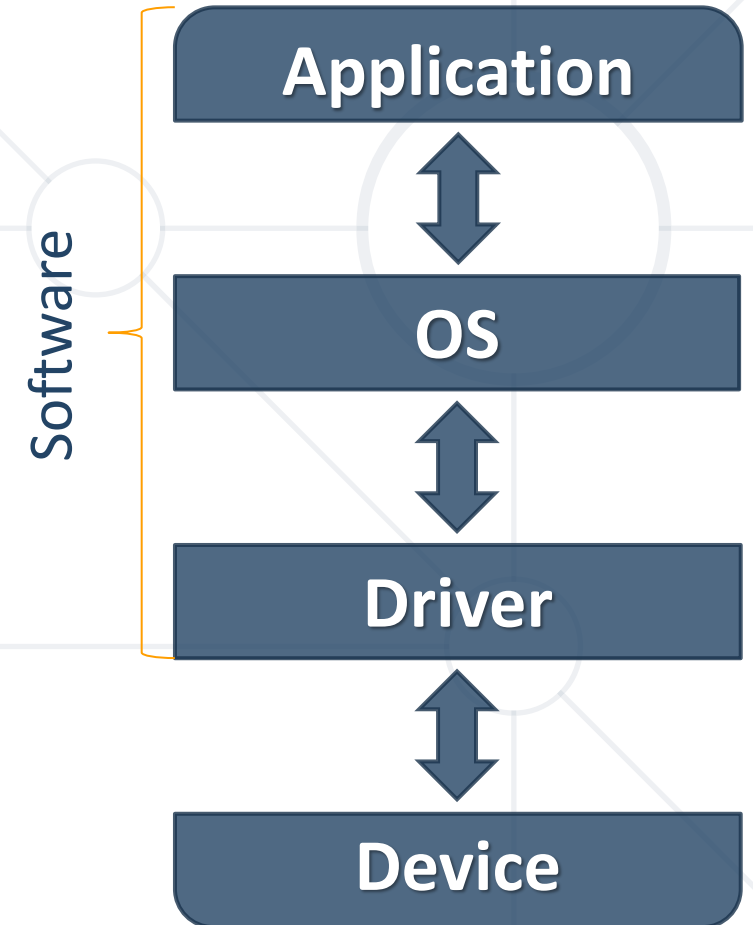
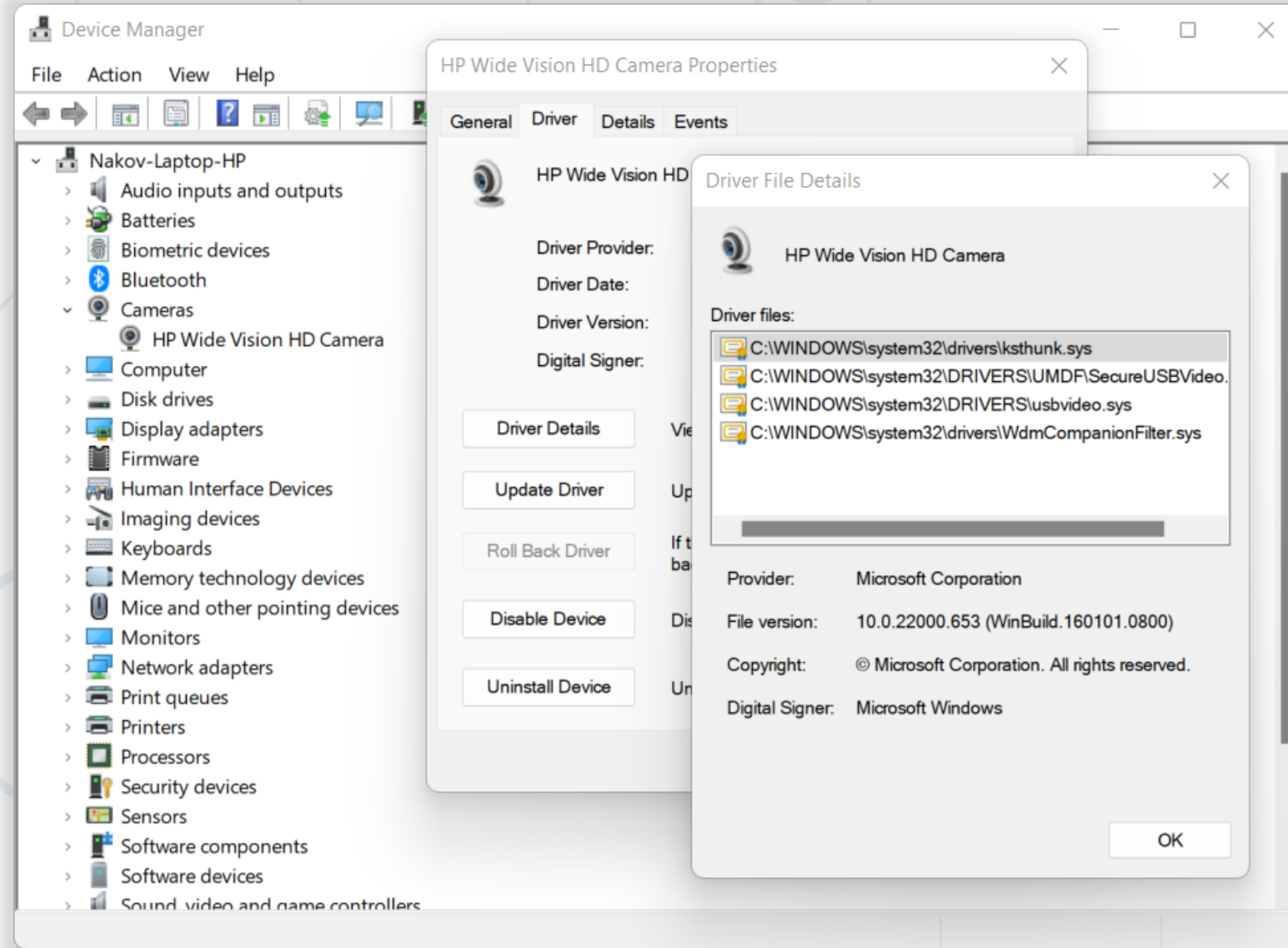


Image Name	CPU	Memory	Command Line
RegSvc.exe	00	154 K	C:\Program Files\Common Files\Intel\Wirel...
NvStreamNetworkService.exe	00	3,567 K	C:\Program Files\NVIDIA Corporation\NvSt...
ntoskrnl.exe	00	120 K	C:\Windows\System32\ntoskrnl.exe
msiexec.exe	00	2,993 K	C:\Windows\System32\msiexec.exe
USBGuard.exe	00	6,474 K	C:\Program Files\USB Disk Security\USBGu...

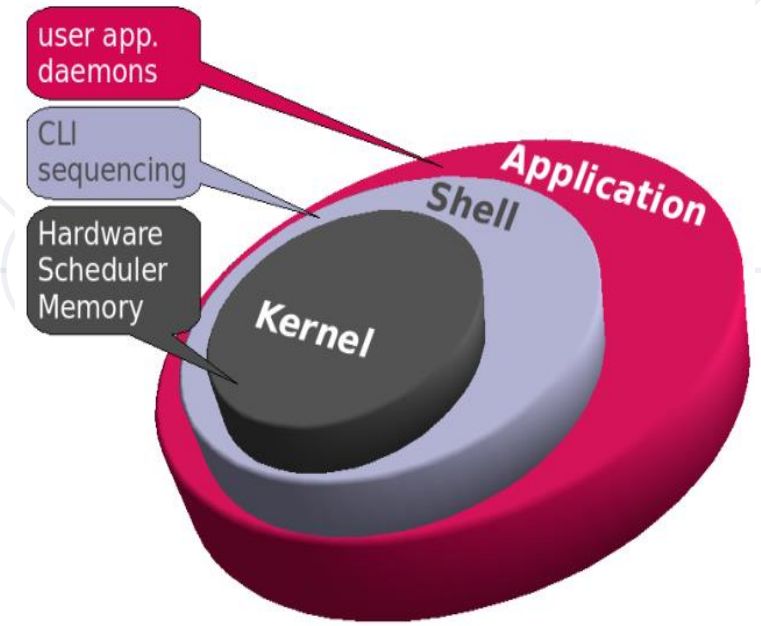
- **Drivers** == set of **system programs** that enable hardware components to function
- Drivers **connect the OS and devices**
  - Enable **hardware components** or peripherals to operate properly
- Drivers are low-level software programs **without a user interface (UI)**
- All hardware components **require a driver** (e. g. disk drives, printers, keyboards)



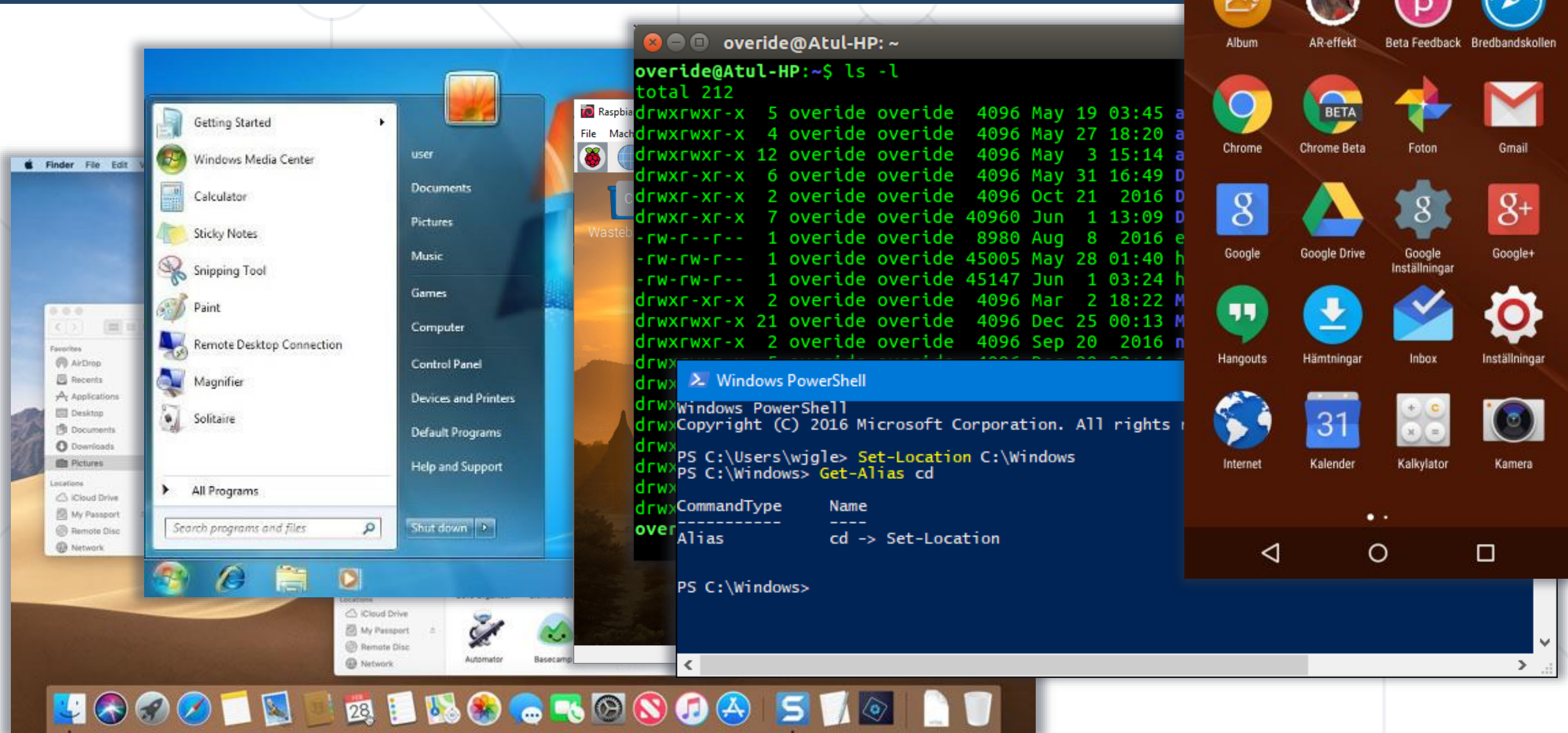
# Drivers – Example



- **Shell** == user interface (UI) to the OS
  - Outermost layer of the operating system, located **between the kernel and the apps**
  - Provides a **UI and tools** to control processes, files, installed software, users, etc.
- Two types of shells:
  - **Command-line (CLI) shells** – require knowledge of commands, syntax, and concepts about the shell-specific scripting language (e. g. bash)
  - **Graphical (GUI) shells** – intuitive, easy to use (e. g. Windows Desktop)
- Most GUI-enabled OS provide also **CLI shells** for advanced users



# OS Shells – Examples



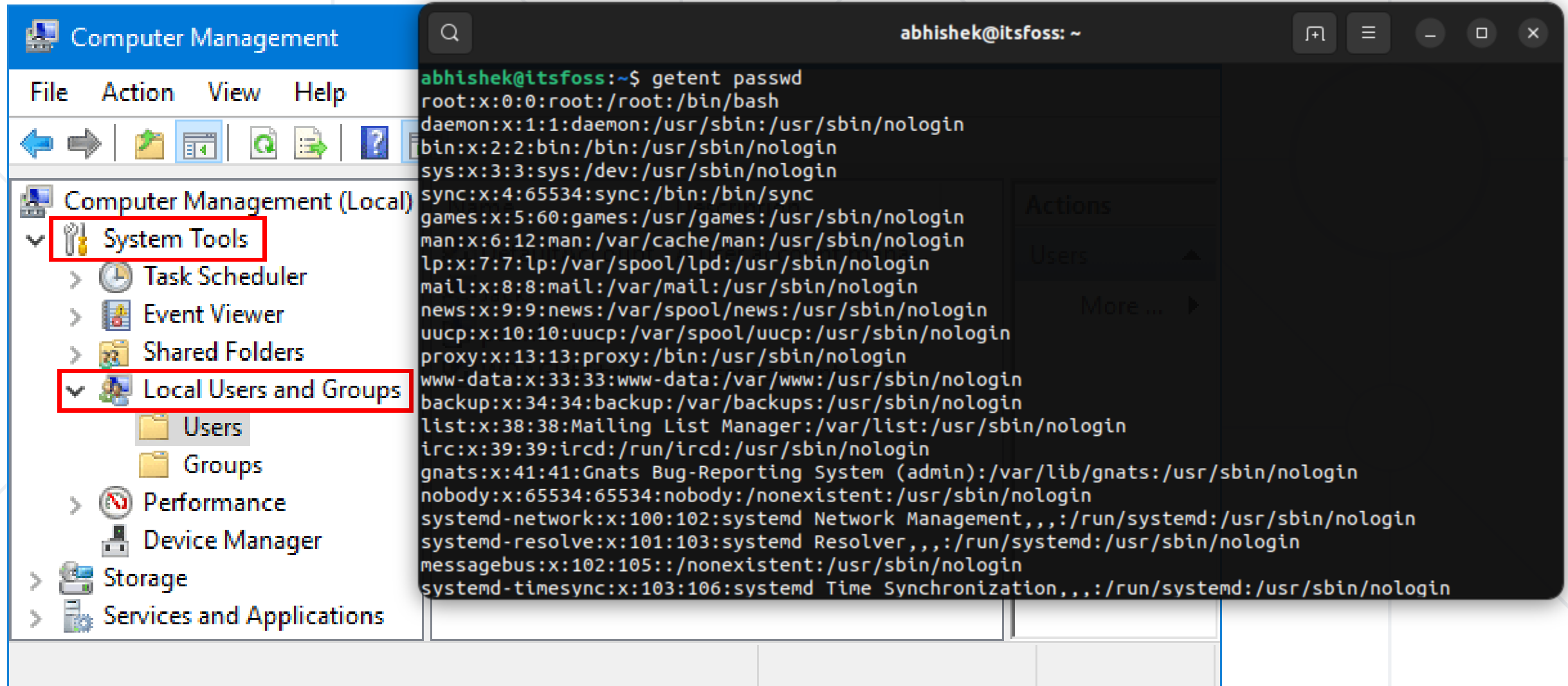
# Users in Operating Systems

- **Users** in the OS == individuals or entities who interact with the system by logging in and performing tasks
- A user often has a **user account** and is identified to the system by a **username**
- Users may have **privileges** over processes, folders and files, devices, services, network and other resources
  - Users are typically **isolated** from each other
- OS can be **single-user** (e. g. DOS) or **multi-user** (e. g. Linux, macOS, Windows)





# Users in the OS – Examples



The image displays two side-by-side windows illustrating user management in different operating systems.

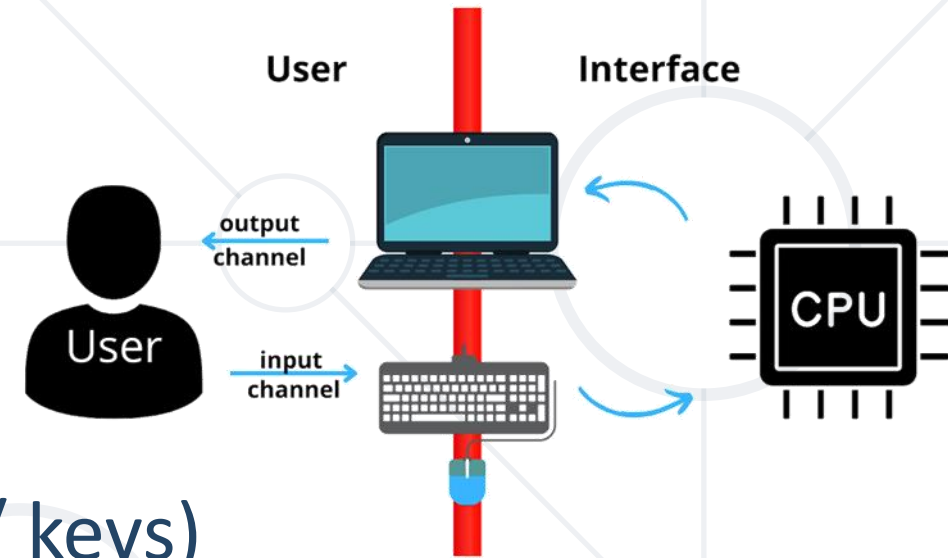
**Left Window (Windows):** The "Computer Management" console is shown. The "System Tools" section is expanded, and "Local Users and Groups" is selected. The "Users" folder is visible under "Local Users and Groups".

**Right Window (Linux Terminal):** A terminal window titled "abhishek@itsfoss: ~" shows the output of the command `getent passwd`. The output lists system and regular users with their respective home directories and shells.

```
abhishek@itsfoss:~$ getent passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:102:105:/:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:103:106:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
```



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



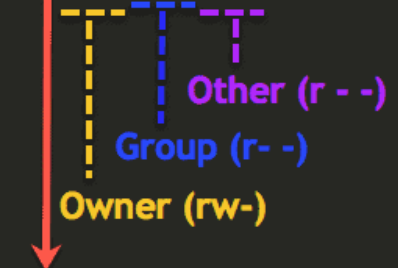
# Authentication vs. Authorization

- 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?**



- OS **controls the use of system and network resources**
  - Through **authentication and authorization**
    - Based on user **permissions** over resources (e. g. file permissions)
- The OS **determines** if an **authenticated user** has the **correct permissions** to access a resource
  - Using built-in authorization and access control technologies

```
# ls -l file
-rw-r--r-- 1 root root 0 Nov 19 23:49 file
```

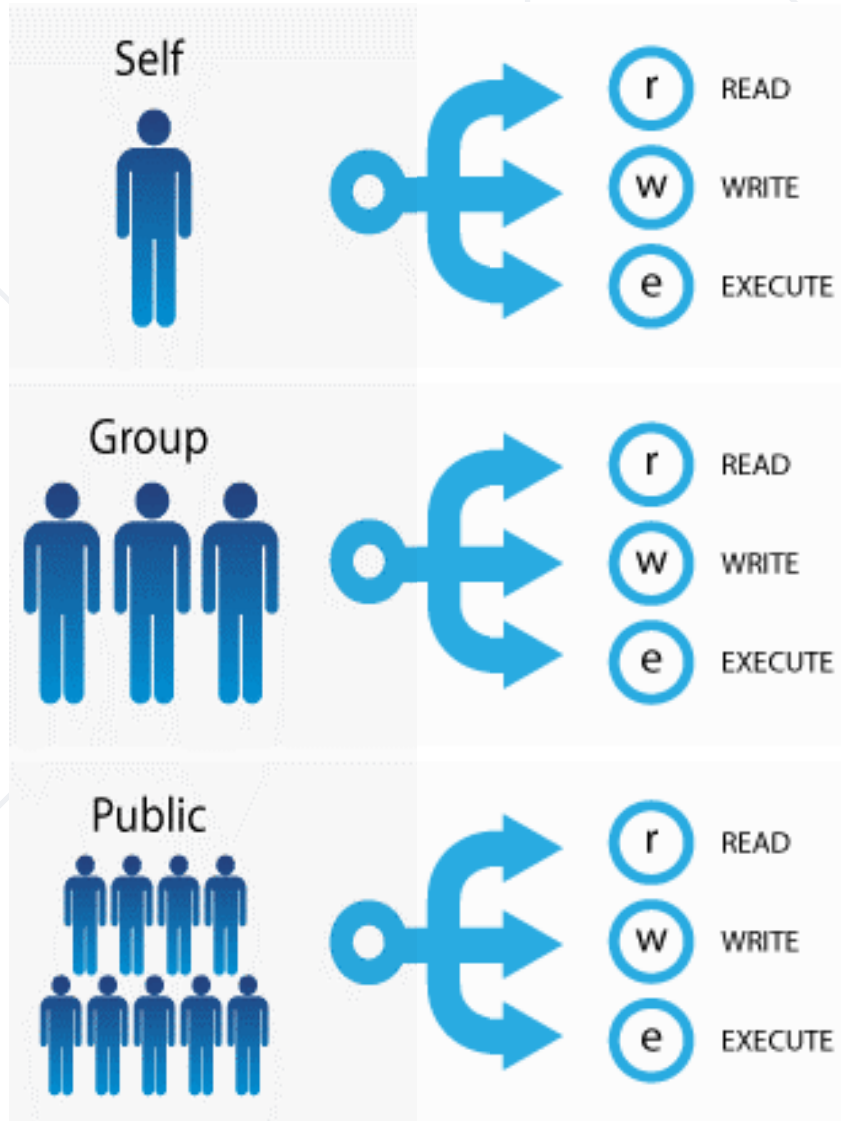


r	=	Readable
w	=	Writeable
x	=	Executable
-	=	Denied

# User Roles (Groups)

- **User roles (groups)** are **permission sets** that control access to resources (files, folders, processes, services)
  - Simplify permission assignments, e. g. in a hosting company, all customers may use the group "**web**"
- Each user account may have multiple **roles**
- Examples of user roles in MS Windows:  
**Administrator, User, Power User, Guest**
- Examples of user groups in Linux: **root, user, nobody**





- **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 and execute**

# Processes in OS

- A **process** is a **program in action** (a running app)
  - Consume CPU time, RAM memory, file handles and other OS resources
- It's the basic unit of work in the operating system
- Unlike files, which are **passive**, processes are an **active entity**
- For example, when you open a browser to search the web, that's a process



# Processes – Examples

Task Manager

Details

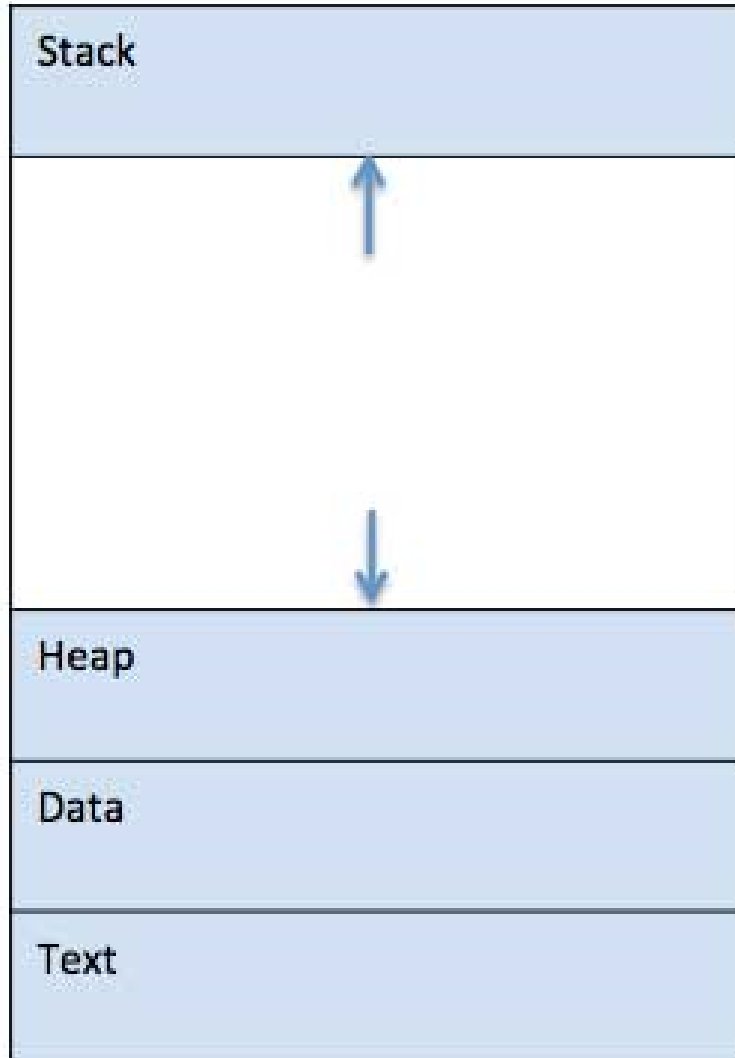
Name	PID	Status
Taskmgr.exe	24460	Running
Skype.exe	15260	Running
System interrupts	-	Running
dwm.exe	2256	Running
explorer.exe	9568	Running
System	4	Running
msedge.exe	9832	Running
chrome.exe	14908	Running
svchost.exe	2748	Running
chrome.exe	15596	Running
chrome.exe	13344	Running
csrss.exe	1092	Running
Skype.exe	15000	Running
POWERPNT.EXE	6356	Running
TGitCache.exe	13172	Running
svchost.exe	7492	Running
Skype.exe	14424	Running
chrome.exe	2092	Running

```
top - 08:53:06 up 16 min, 1 user, load average: 0.20, 0.28, 0.35
Tasks: 238 total, 1 running, 237 sleeping, 0 stopped, 0 zombie
%Cpu(s): 5.4 us, 1.0 sy, 0.0 ni, 93.0 id, 0.6 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 3742792 total, 1144416 free, 1236544 used, 1361832 buff/cache
KiB Swap: 5631996 total, 5631996 free, 0 used. 2249948 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2469	aaronki+	20	0	1757760	168424	56580	S	11.6	4.5	0:37.61	cinnamon
3691	aaronki+	20	0	479484	35208	26268	S	6.3	0.9	0:00.42	gnome-scre+
1946	root	20	0	463000	96932	85920	S	4.3	2.6	0:21.98	Xorg
170	root	20	0	0	0	0	S	1.3	0.0	0:00.69	kworker/u1+
6	root	20	0	0	0	0	S	1.0	0.0	0:00.85	kworker/u1+
921	root	20	0	449740	19428	14048	S	0.7	0.5	0:01.05	NetworkMan+
1743	shinken	20	0	1557824	31040	6504	S	0.7	0.8	0:06.95	shinken-sc+
1817	shinken	20	0	1631460	32172	7856	S	0.7	0.9	0:04.32	shinken-re+
7	root	20	0	0	0	0	S	0.3	0.0	0:01.17	rcu_sched
1865	shinken	20	0	1632116	32604	7284	S	0.3	0.9	0:05.92	shinken-br+
1908	shinken	20	0	1557024	30232	6556	S	0.3	0.8	0:03.24	shinken-re+
1953	root	20	0	1633896	34552	5712	S	0.3	0.9	0:04.19	shinken-ar+
2082	shinken	20	0	1631232	29112	4728	S	0.3	0.8	0:00.20	shinken-po+
3684	aaronki+	20	0	41908	3808	3104	R	0.3	0.1	0:00.04	top
1	root	20	0	119696	5924	4040	S	0.0	0.2	0:01.45	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	20	0	0	0	0	S	0.0	0.0	0:00.01	ksoftirqd/0

svetl	00	9 604 K	x64	Host Process for Windows Services		
svetl	00	141 512 K	x86	Skype		
svetl	00	122 536 K	x64	Google Chrome		





- When a **program** is **loaded** into the **memory**, 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

- **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

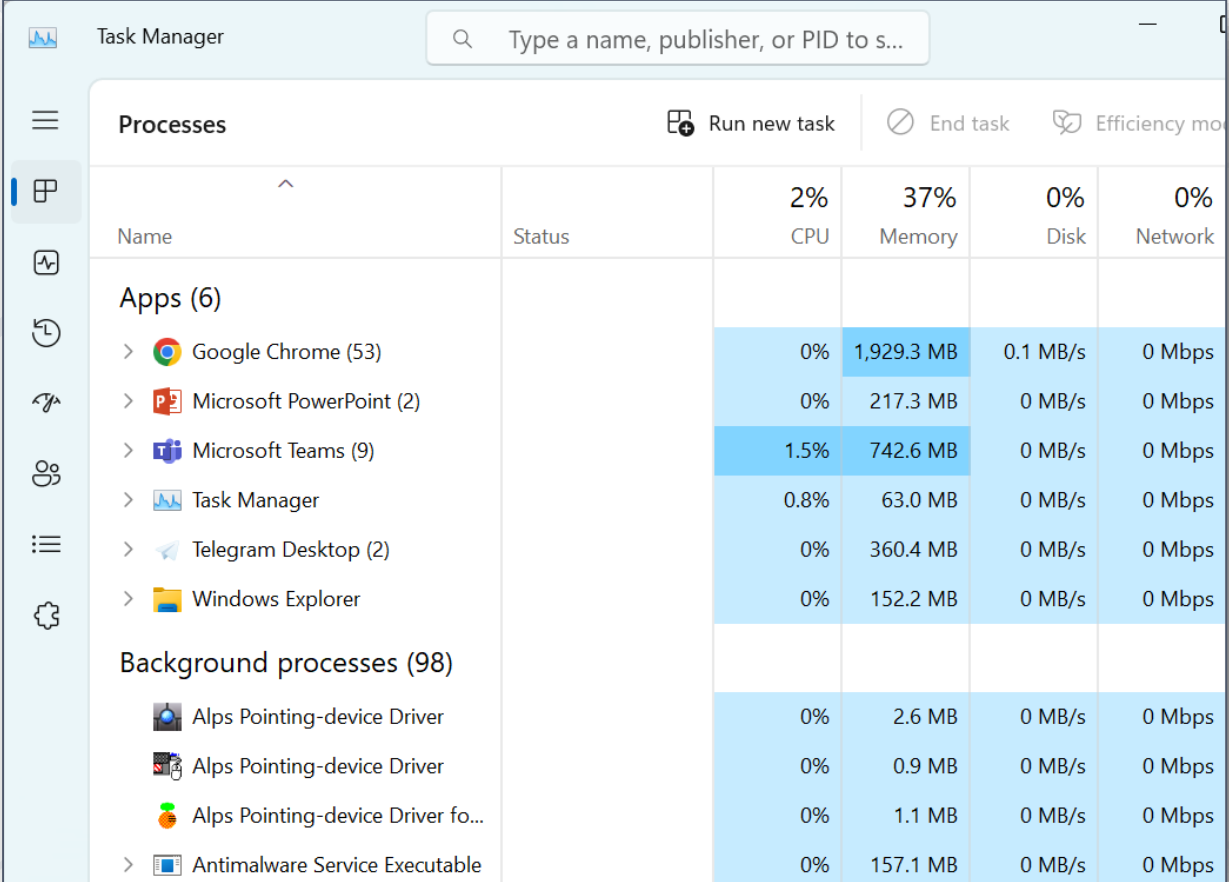
# Task Manager

- 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 the **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**



Type a name, publisher, or PID to s...					
Processes					
Run new task End task Efficiency mo					
Name	Status	2% CPU	37% Memory	0% Disk	0% Network
Apps (6)					
> Google Chrome (53)		0%	1,929.3 MB	0.1 MB/s	0 Mbps
> Microsoft PowerPoint (2)		0%	217.3 MB	0 MB/s	0 Mbps
> Microsoft Teams (9)		1.5%	742.6 MB	0 MB/s	0 Mbps
> Task Manager		0.8%	63.0 MB	0 MB/s	0 Mbps
> Telegram Desktop (2)		0%	360.4 MB	0 MB/s	0 Mbps
> Windows Explorer		0%	152.2 MB	0 MB/s	0 Mbps
Background processes (98)					
Alps Pointing-device Driver		0%	2.6 MB	0 MB/s	0 Mbps
Alps Pointing-device Driver		0%	0.9 MB	0 MB/s	0 Mbps
Alps Pointing-device Driver fo...		0%	1.1 MB	0 MB/s	0 Mbps
> Antimalware Service Executable		0%	157.1 MB	0 MB/s	0 Mbps



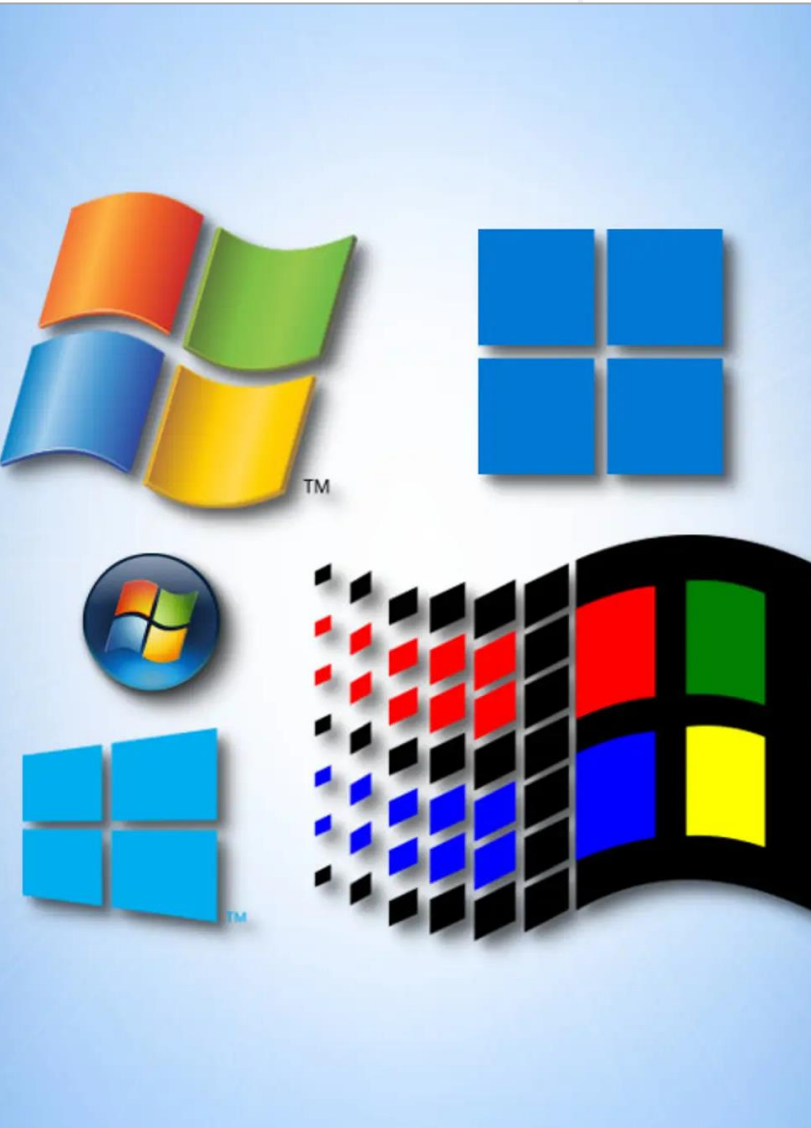
# Operating Systems

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





- 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





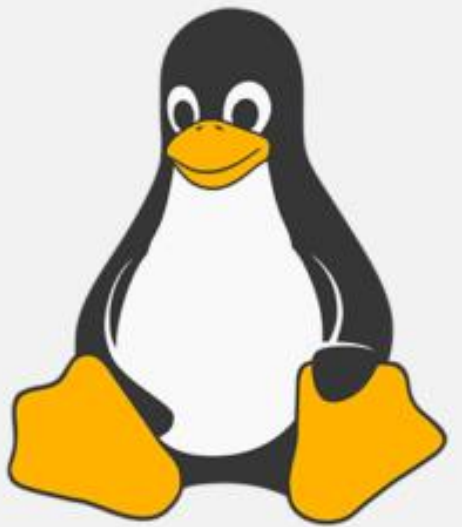
- **Apple** and **Macintosh** computers run on **macOS** and **OS X**, which are proprietary operating systems developed and marketed by Apple Inc
- **macOS** is a **Unix-based** operating system that was first released over 20 years ago
- In 2020, Apple began **transitioning** to its own 64-bit ARM-based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



- 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**



- **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

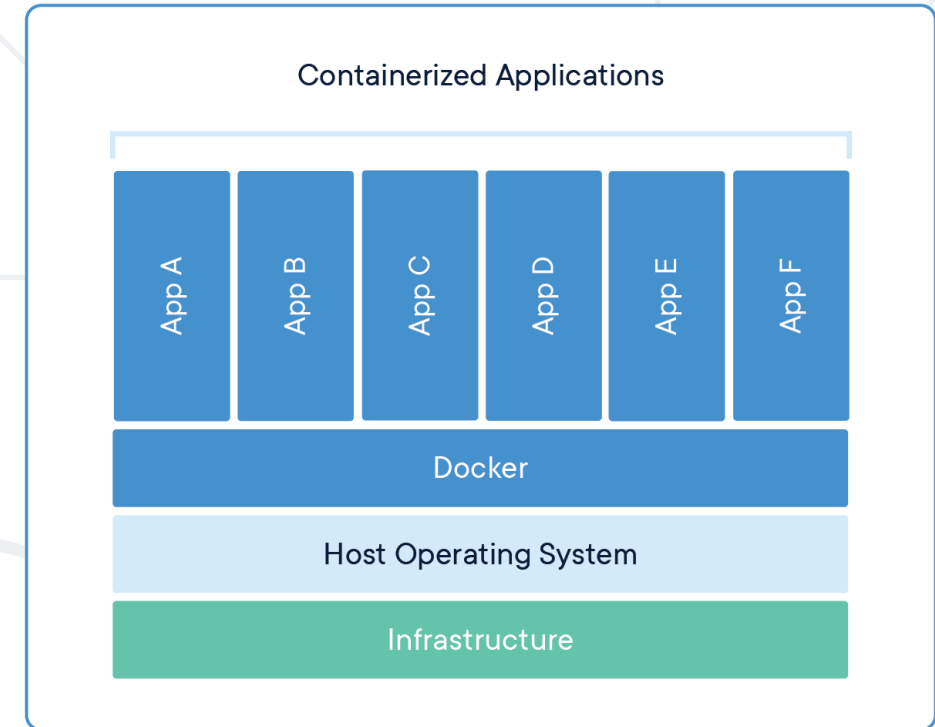


- **Packages of software** that can run in **any environment**
- They **virtualize the operating system** and can run on private data centers, public clouds, or personal laptops
- Provide a solution to the problem of **running software** reliably **across different computing environments**
- **Isolate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

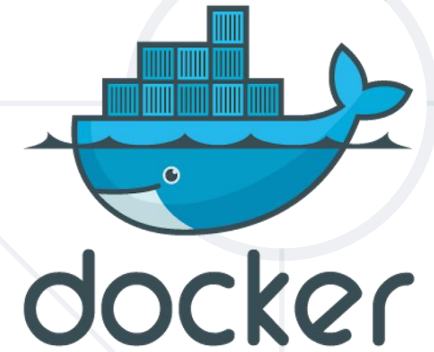


# OS inside another OS – e.g. Docker Containers

- 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.



- 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



- 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



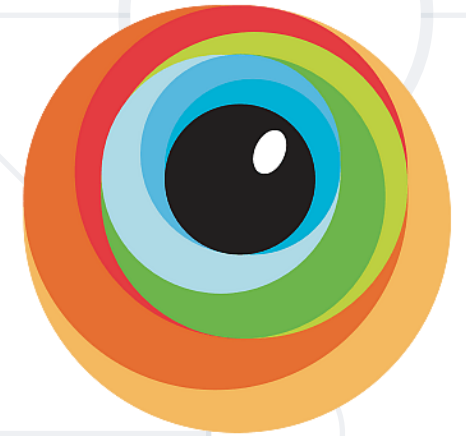
# OS Emulators

- **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

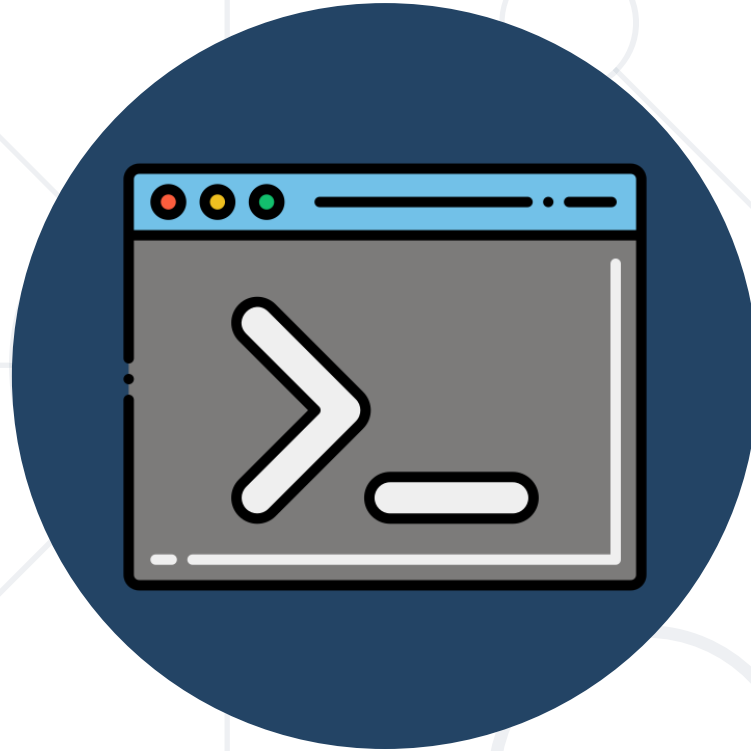


# BrowserStack – App & Browser Testing

- 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**

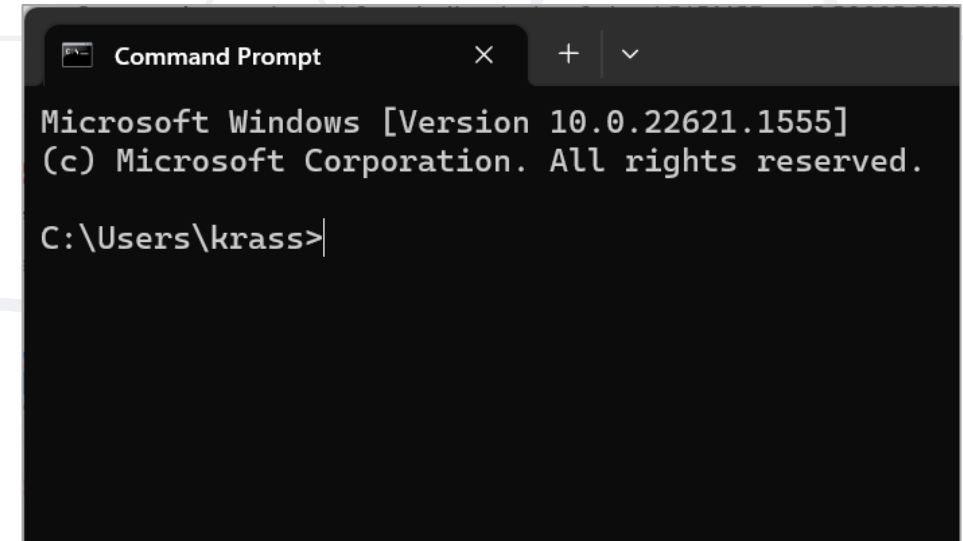


# Shell & Shell Commands

## Command Execution

# Shell - Navigating Files & Directories

- **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**

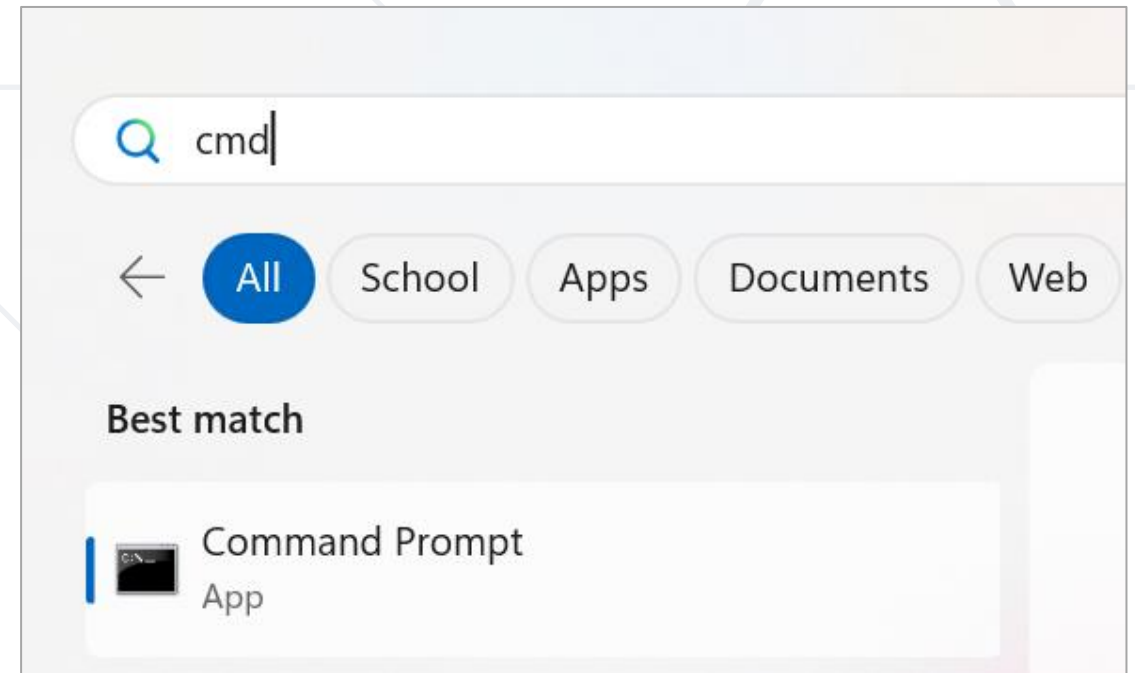
A screenshot of a Windows Command Prompt window. The title bar reads "Command Prompt". The window content shows the Microsoft Windows version (10.0.22621.1555) and copyright information for Microsoft Corporation. The current directory is C:\Users\krass>.

```
Command Prompt
Microsoft Windows [Version 10.0.22621.1555]
(c) Microsoft Corporation. All rights reserved.

C:\Users\krass>
```

# Opening Command or Shell Prompt

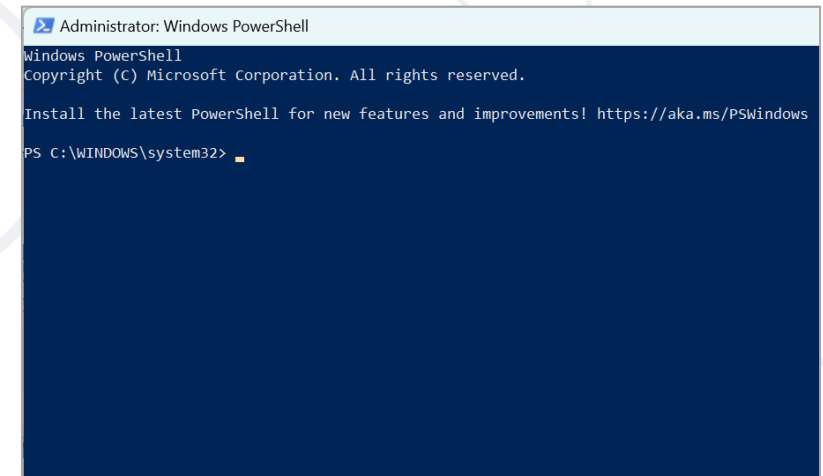
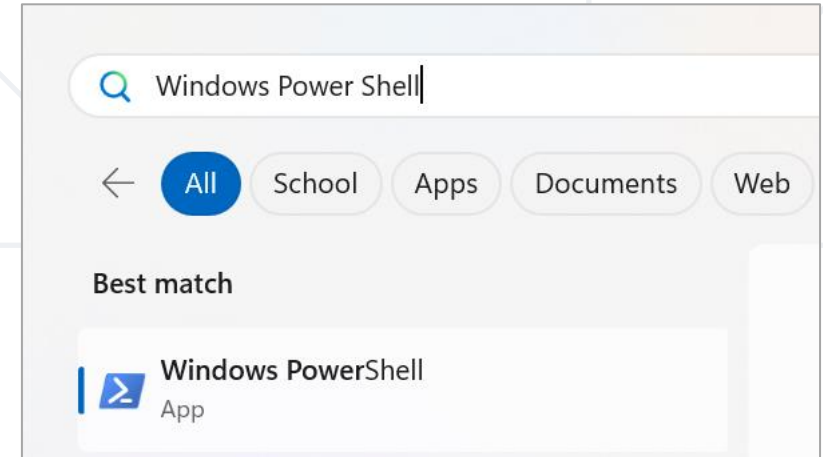
- Click Start > Run or press Windows + R key.
- Type cmd
- Click on Command Prompt





# 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`

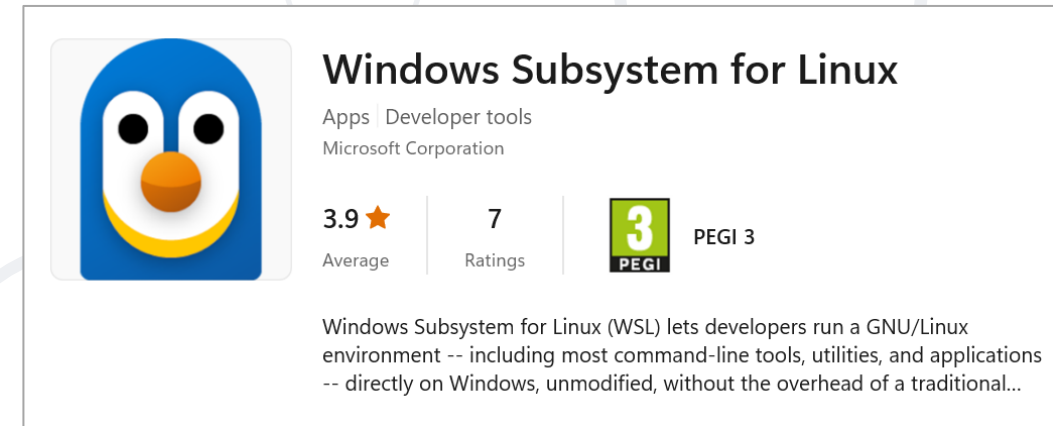
- **`cd`** changes the current working directory to the specified drive
- The **`cd`** command can be used to:
  - change into a subdirectory
  - move back into the parent directory
  - move all the way back to the root directory
  - move to any given directory
- **`pwd`** prints the directory you are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



- 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

- 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

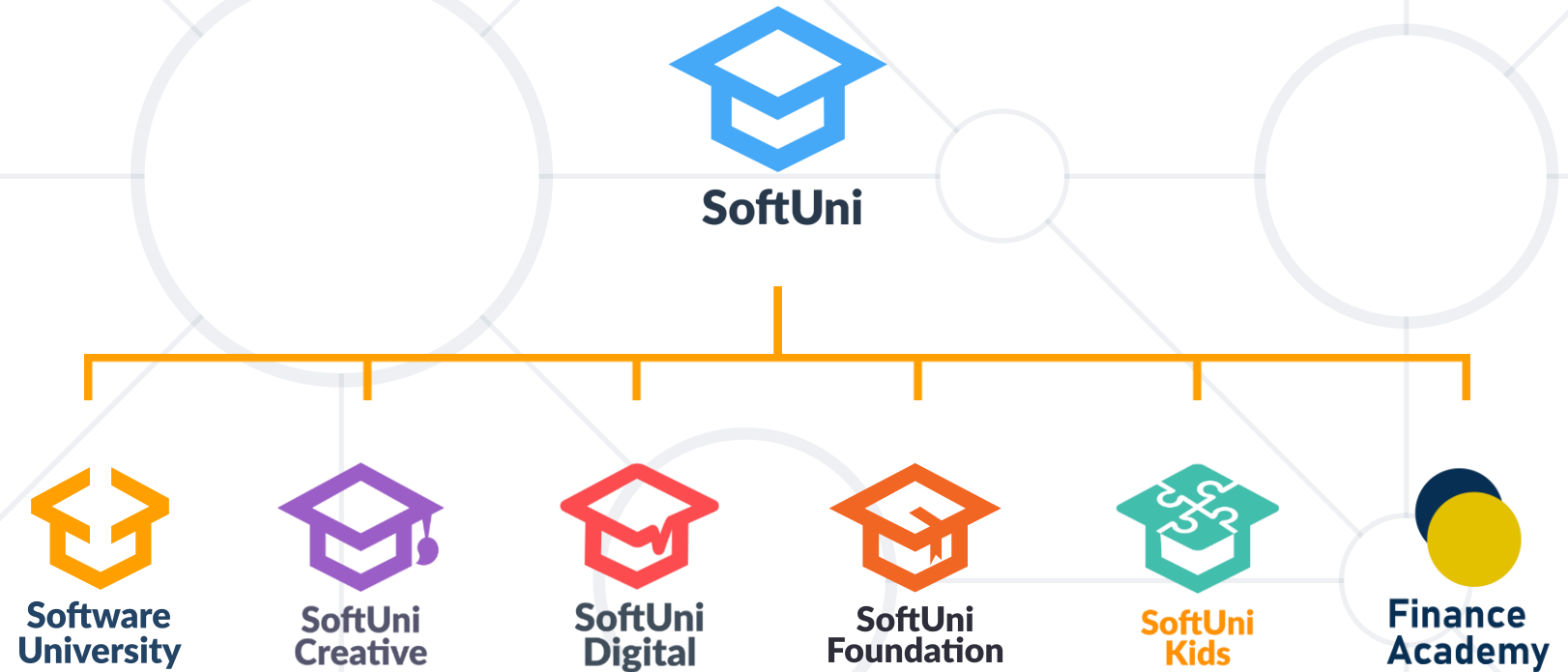


- **Operating Systems** Overview
- **OS Examples**
- **Virtual Machines & Containers**
  - OS inside another OS
  - Remote VM instances
  - OS Emulators
- **Shell & Shell Commands**





# Questions?



# SoftUni Diamond Partners

**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

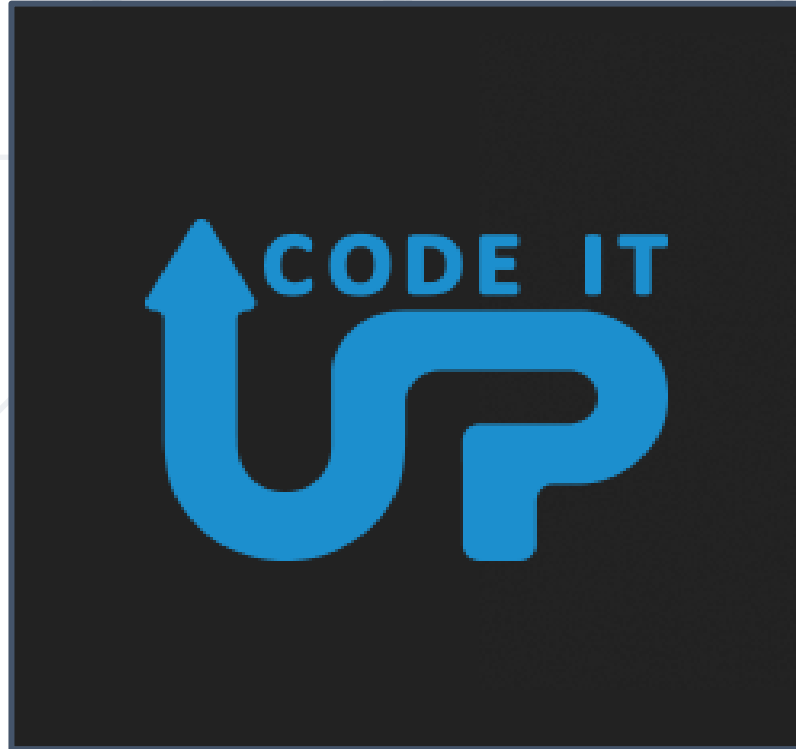


**BOSCH**

**DXC**  
TECHNOLOGY



**SmartIT**



- Software University – High-Quality Education, Profession and Job for Software Developers

- [softuni.bg](http://softuni.bg), [about.softuni.bg](http://about.softuni.bg)

- Software University Foundation

- [softuni.foundation](http://softuni.foundation)

- Software University @ Facebook

- [facebook.com/SoftwareUniversity](https://facebook.com/SoftwareUniversity)

- Software University Forums

- [forum.softuni.bg](http://forum.softuni.bg)



Software University



- 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>

