

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

#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

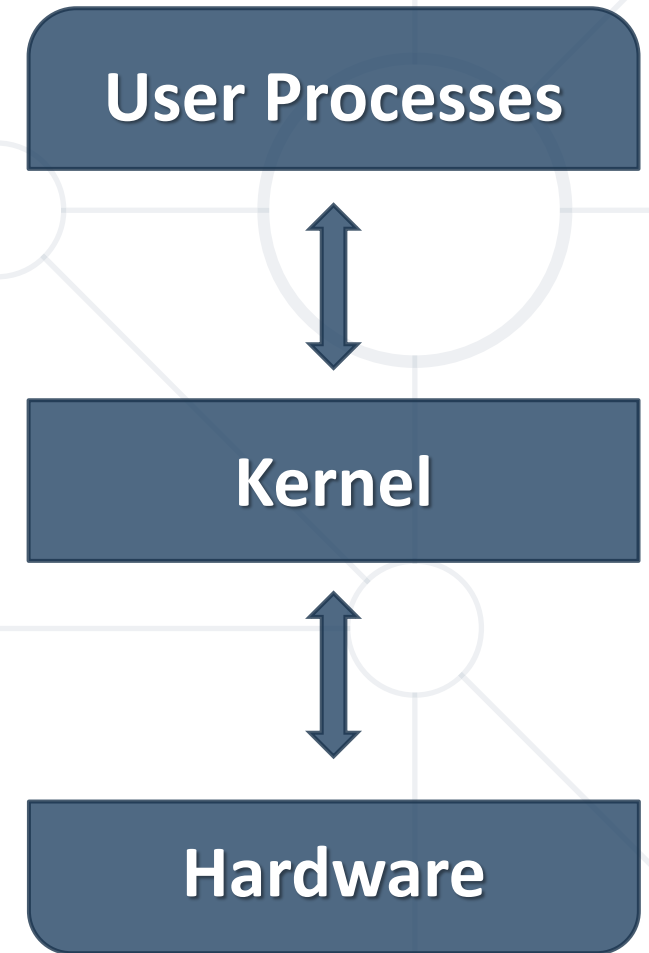


OS Main Functions (2)

- **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

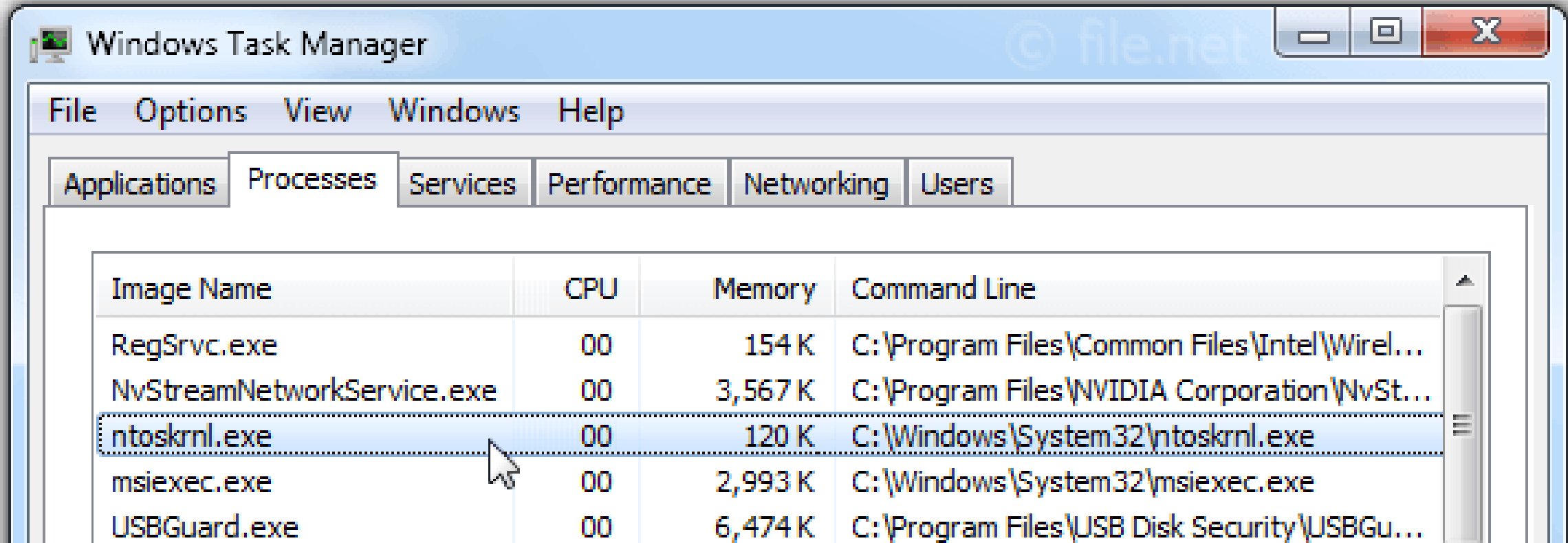
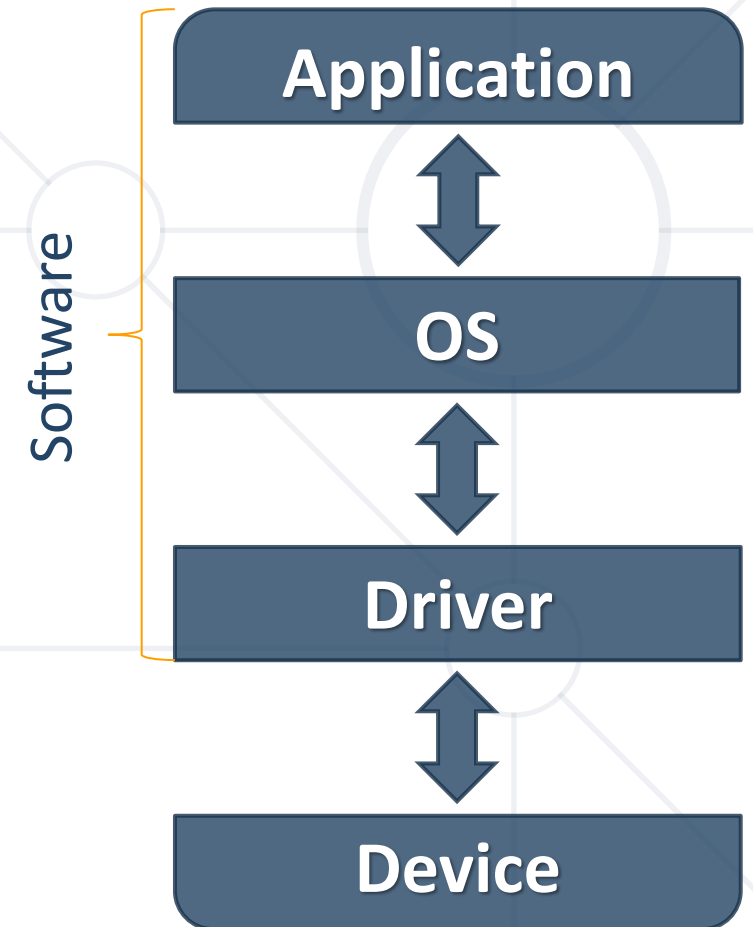
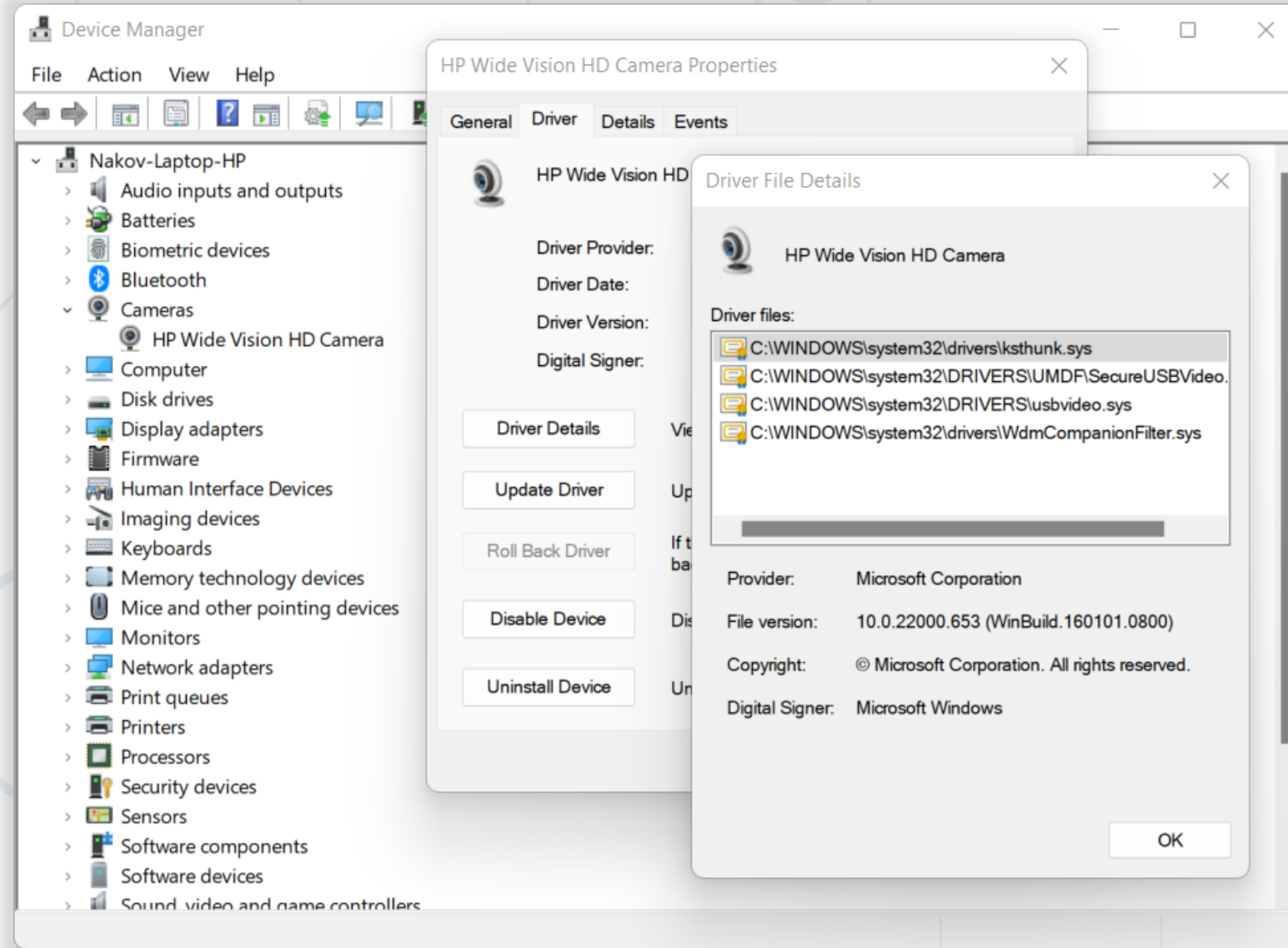


Image Name	CPU	Memory	Command Line
RegSrv.exe	00	154 K	C:\Program Files\Common Files\Intel\Wirel...
NvStreamNetworkService.exe	00	3,567 K	C:\Program Files\NVIDIA Corporation\NvSt...
ntoskrnl.exe	00	120 K	C:\Windows\System32\ntoskrnl.exe
msiexec.exe	00	2,993 K	C:\Windows\System32\msiexec.exe
USBGuard.exe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

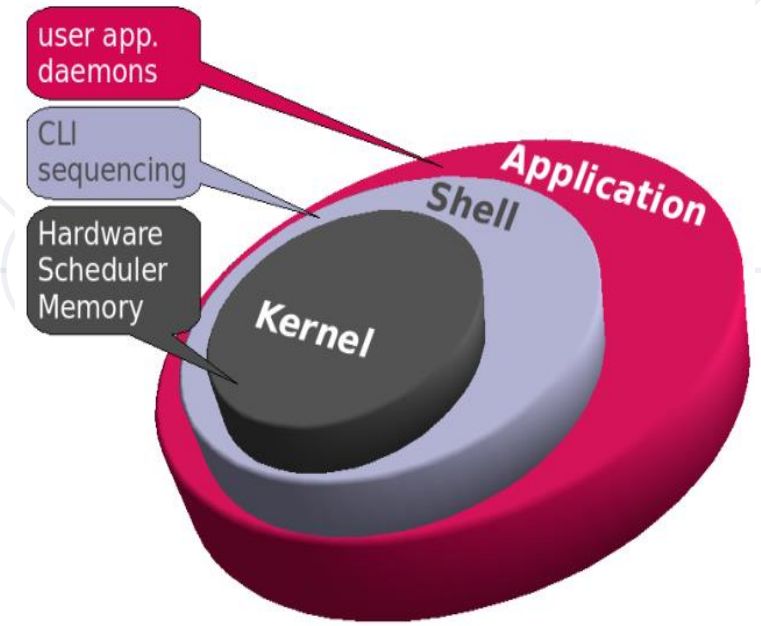
- **Drivers** == set of **system programs** that enable hardware components to function
- Drivers **connect the OS and devices**
 - Enable **hardware components** or peripherals to operate properly
- Drivers are low-level software programs **without a user interface (UI)**
- All hardware components **require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



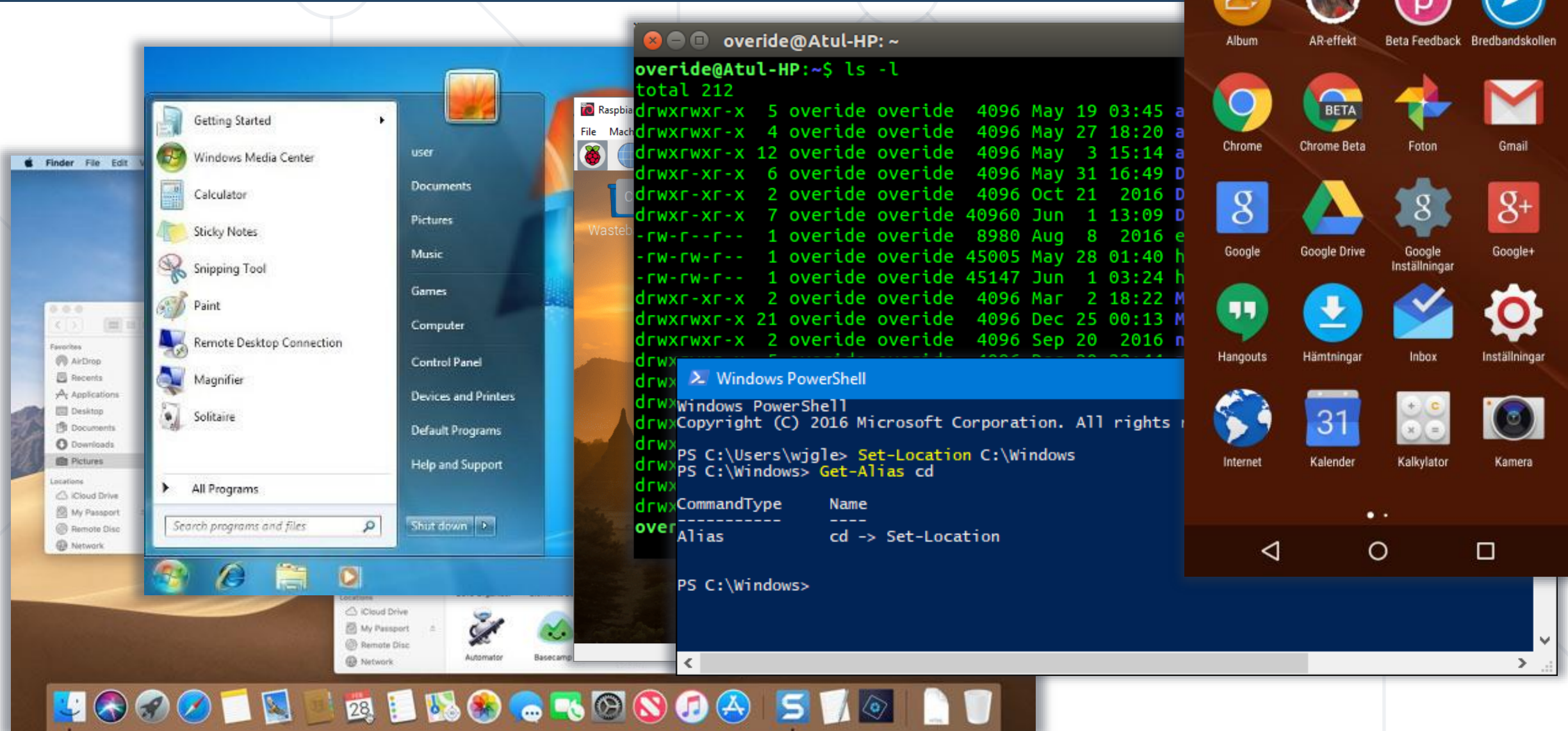
Drivers – Example



- **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

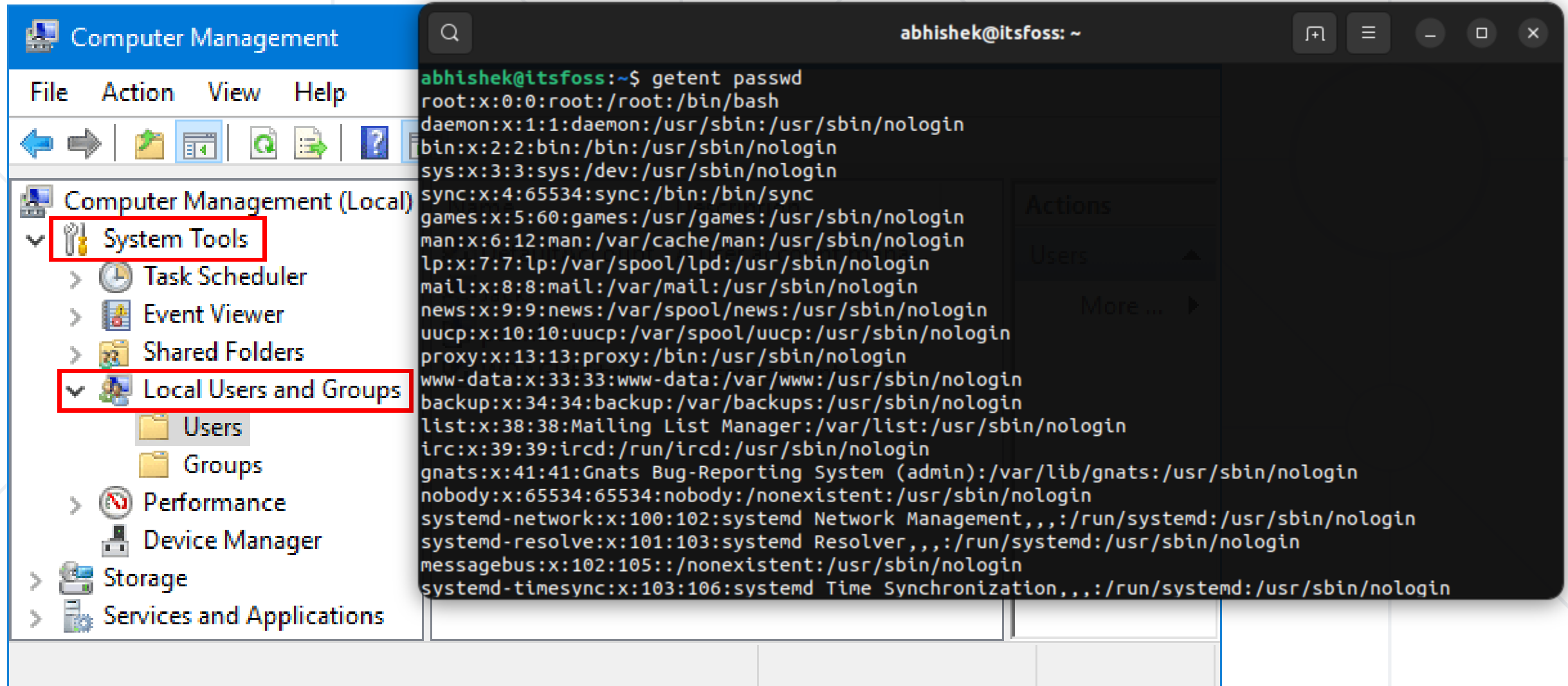


Users in Operating Systems



- **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 Computer Management

- Menu: File, Action, View, Help
- Tree View:
 - Computer Management (Local)
 - System Tools** (highlighted with a red box)
 - Task Scheduler
 - Event Viewer
 - Shared Folders
 - Local Users and Groups** (highlighted with a red box)
 - Users
 - Groups
 - Performance
 - Device Manager
 - Storage
 - Services and Applications

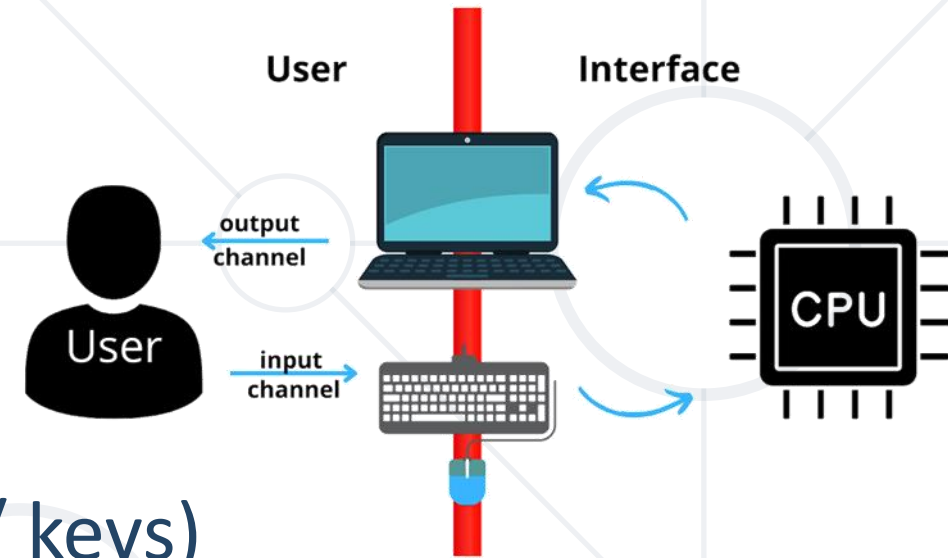
Right Window: Linux Terminal (abhishek@itsfoss: ~)

```
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
```

Actions Panel:

- Users
- More ...

- **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**



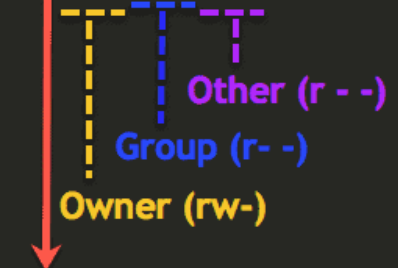
Authentication vs. Authorization

- 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
```

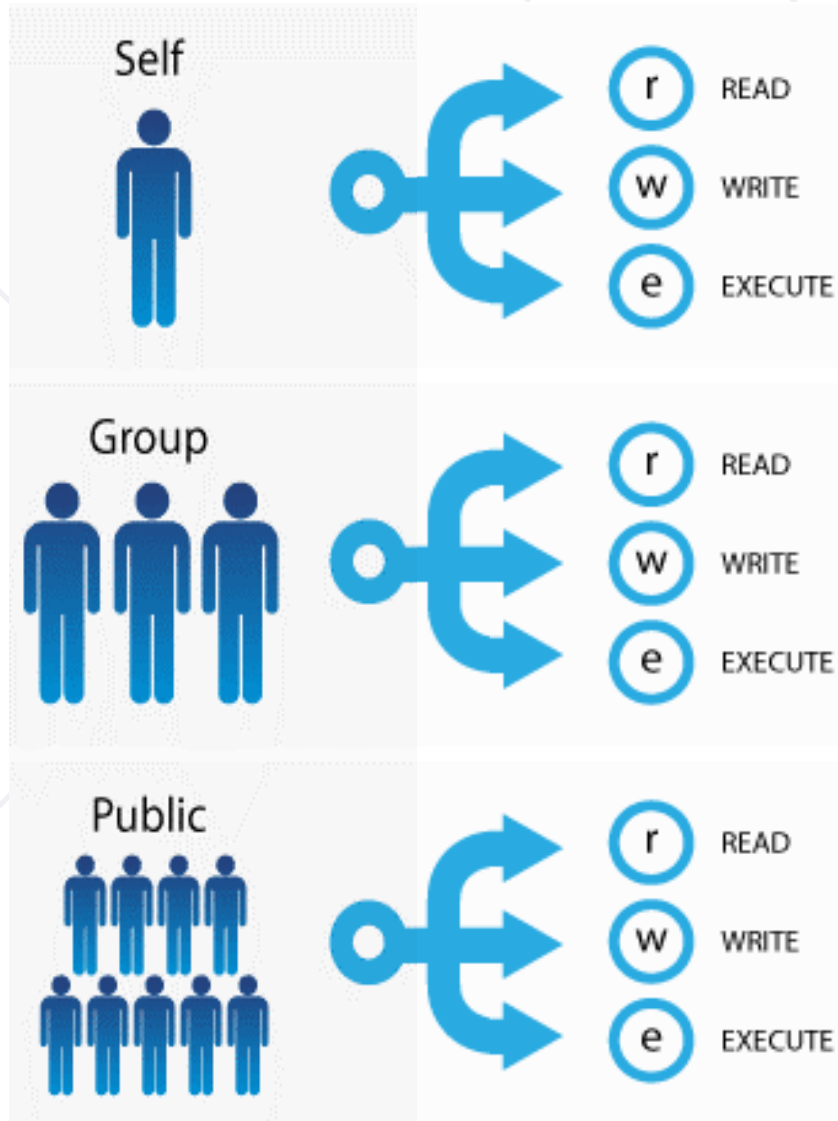


r	=	Readable
w	=	Writeable
x	=	Executable
-	=	Denied

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



Task Managers

- In OS, a **task manager** is a **system monitoring app**
 - View **processes**, **users**, consumed resources, etc.
 - View RAM, CPU, GPU, disk, network load
 - Start / terminate (kill) processes
- Examples:
 - **Windows Task Manager** in MS Windows
 - **top** and **htop** in Linux
 - **Activity Monitor** in macOS



Processes – Examples

Task Manager

Details

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

```
top - 08:53:06 up 10
Tasks: 238 total,
%Cpu(s): 5.4 us,
KiB Mem : 3742792
KiB Swap: 5631996
```

PID	USER	PR
2469	aaronki+	20
3691	aaronki+	20
1946	root	20
170	root	20
6	root	20
921	root	20
1743	shinken	20
1817	shinken	20
7	root	20
1865	shinken	20
1908	shinken	20
1953	root	20
2082	shinken	20
3684	aaronki+	20
1	root	20
2	root	20
3	root	20

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

Activity Monitor

All Processes

CPU Memory Energy Disk Network

Process Name	% CPU	CPU Time	% GPU	GPU Time	PID	User
Activity Monitor	17.5	12.93	0.0	0.00	3357	kirk
kernel_task	15.8	23:36.60	0.0	0.00	0	root
WindowServer	6.6	15:44.35	0.0	2:07.32	133	_windowserver
sysmond	4.0	4.23	0.0	0.00	447	root
bluetoothd	1.9	41.21	0.0	0.00	122	root
airportd	1.6	1:19.97	0.0	0.00	246	root
Music	1.0	4:42.70	0.0	0.01	1445	kirk
iStat Menus Status	1.0	3:54.28	0.0	0.00	856	kirk
cloudd	0.9	34.73	0.0	0.00	583	kirk
Resilio Sync	0.9	2:00.71	0.0	0.00	2514	kirk

System: 4.14%

User: 5.26%

Idle: 90.60%

CPU LOAD

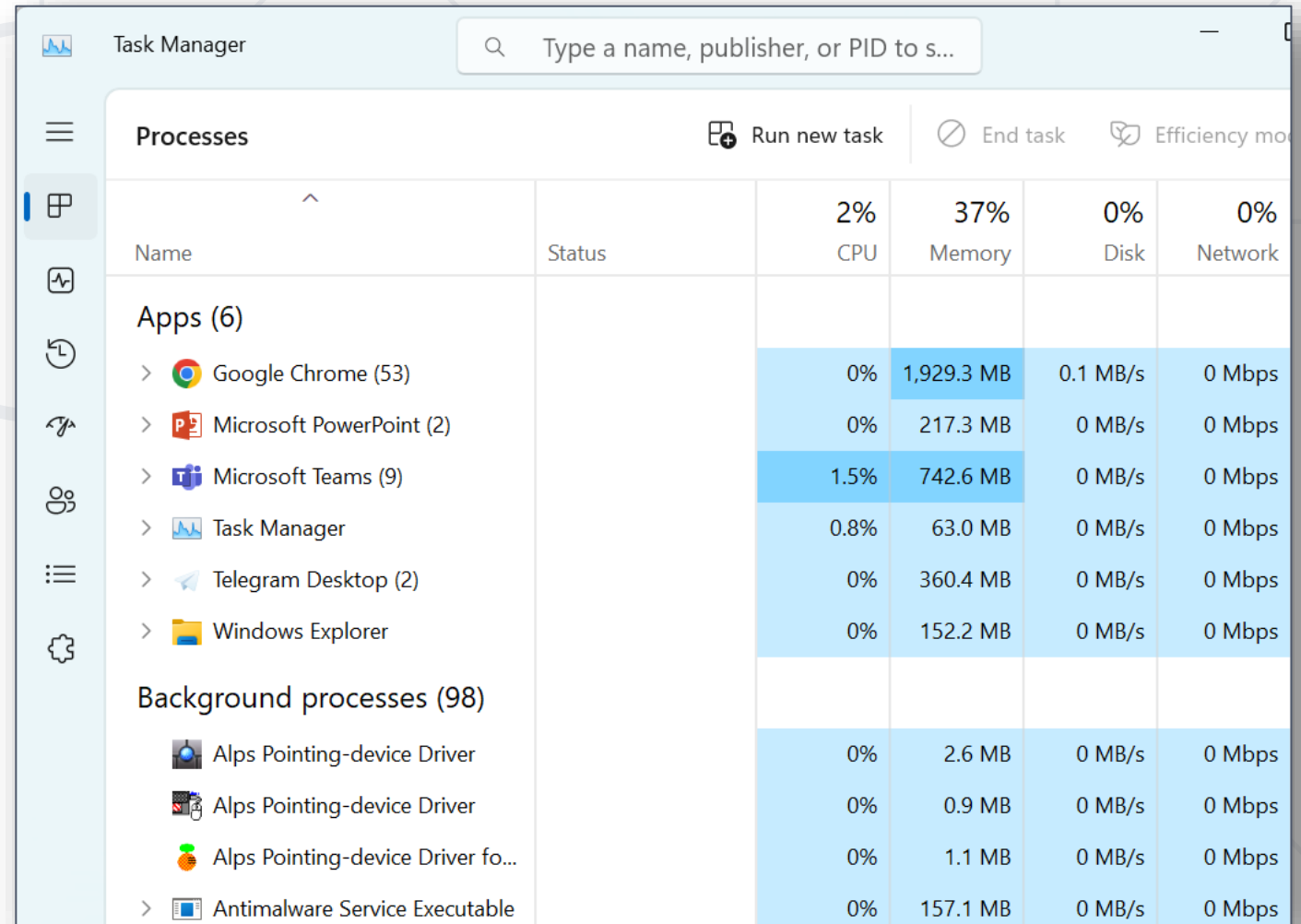
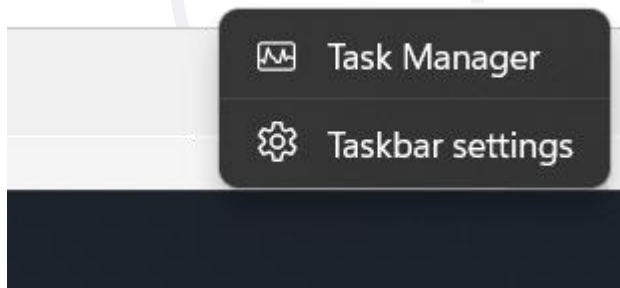
Threads: 2,152

Processes: 618

23

Windows Task Manager

- Open the **Task Manager** in MS Windows:
 - [**Ctrl + Alt + Delete**] → select [**Task Manager**] from the menu
 - Right click on the task bar → [**Task Manager**]



Task Manager

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

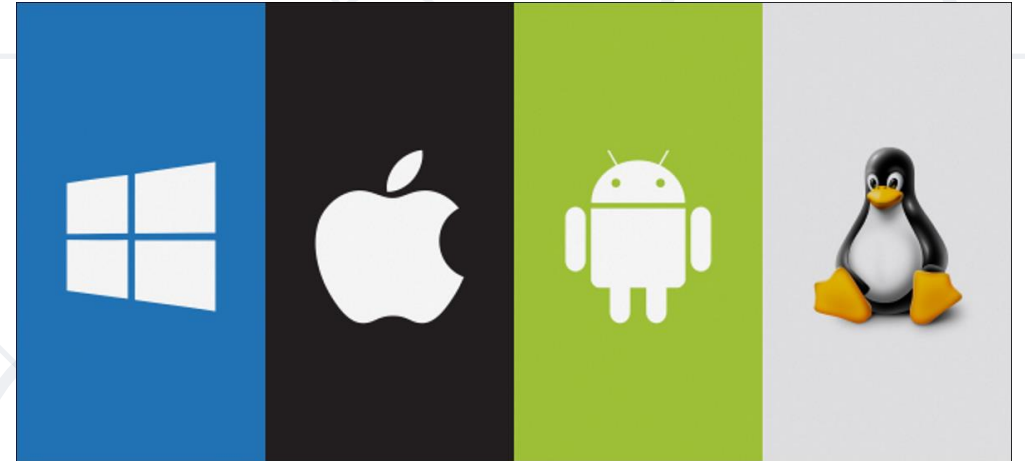


Popular Operating Systems

Windows, Linux, macOS, Android, iOS

Most Popular Operating Systems

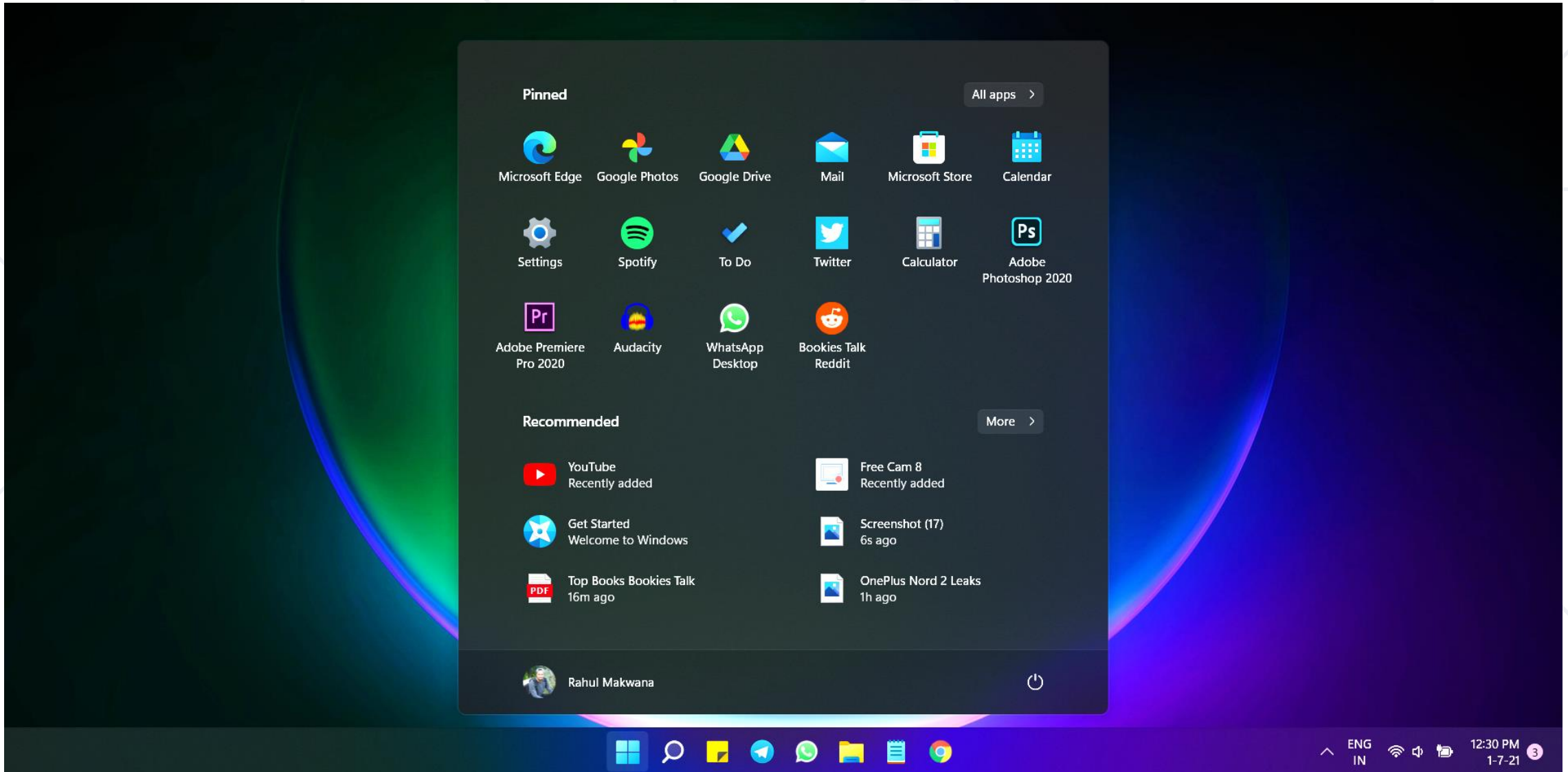
- **Five major** operating system
 - Microsoft Windows
 - Apple macOS
 - Google's Android OS
 - Apple iOS
 - Linux (open source)





- Proprietary OS, developed by **Microsoft**
- One of the **most popular** OS
 - Typically **preinstalled** on new PC
- Several versions: Windows 95 / 98 / Vista, Windows 7 / 8 / 10 / 11
 - Has been around since the 1980s
- Easy-to-use, intuitive GUI shell
 - Many apps and games

Microsoft Windows – Desktop

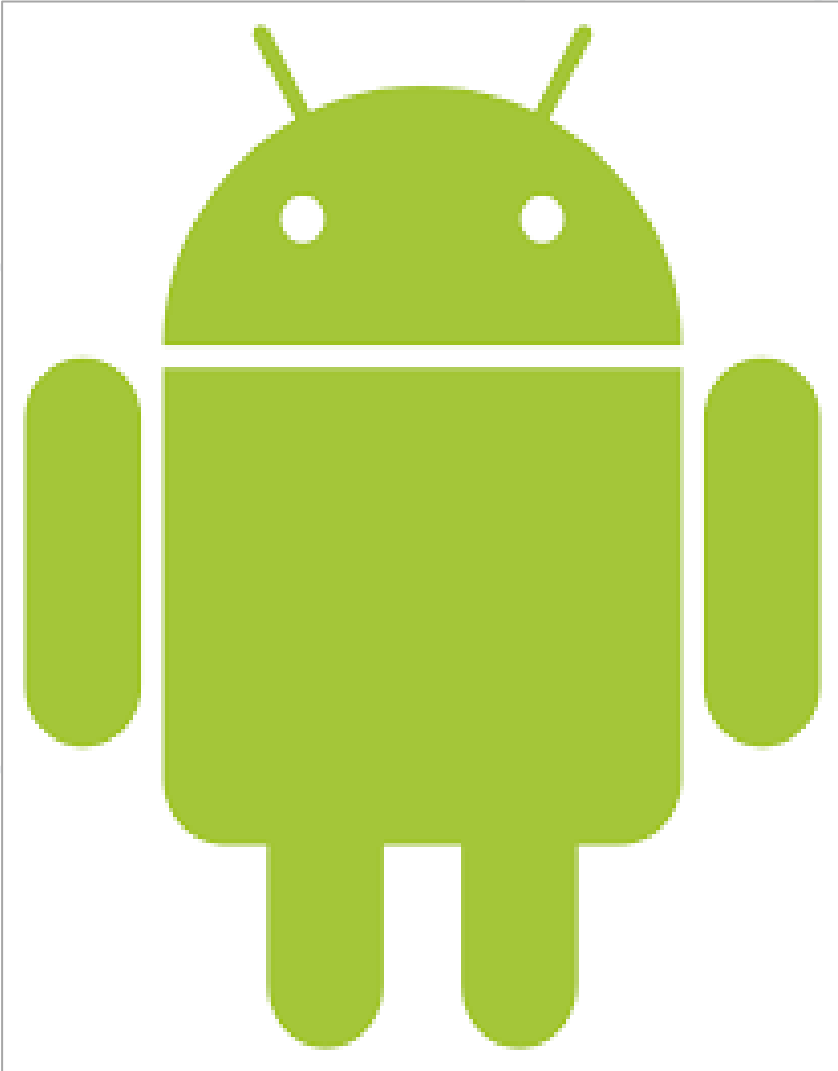




- **Apple** and **Macintosh** computers run on **macOS** and **OS X**
 - Proprietary OS developed by Apple
- **macOS** is a **Unix-based** OS
 - Released over 20 years ago
- In 2020, Apple began **transitioning** to its own 64-bit ARM-based Apple M CPU
 - Apple M1 / M2 CPU: powerful and silent

Apple macOS – Desktop





- Mobile OS, **designed** for **touchscreen** mobile devices
- **Based** on a **modified version** of the **Linux kernel** and other open-source software
- Core OS is called **Android Open-Source Project (AOSP)**
 - Free and open-source software
 - **Developed** and **maintained** by **Google**
- Many distributions (by Samsung, Xiaomi)

Android OS – Smartphone and Tablet

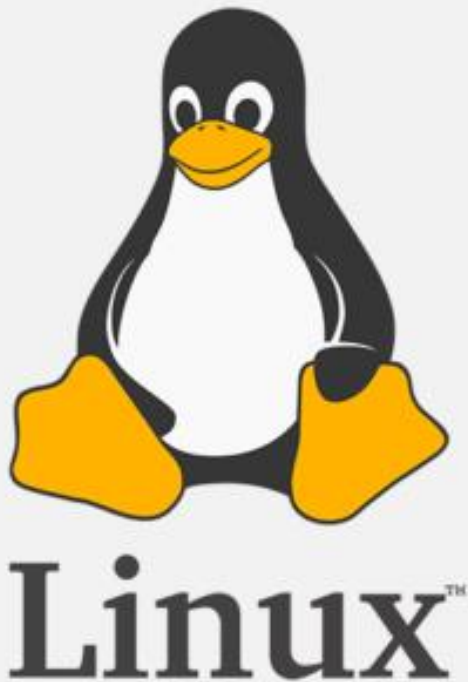




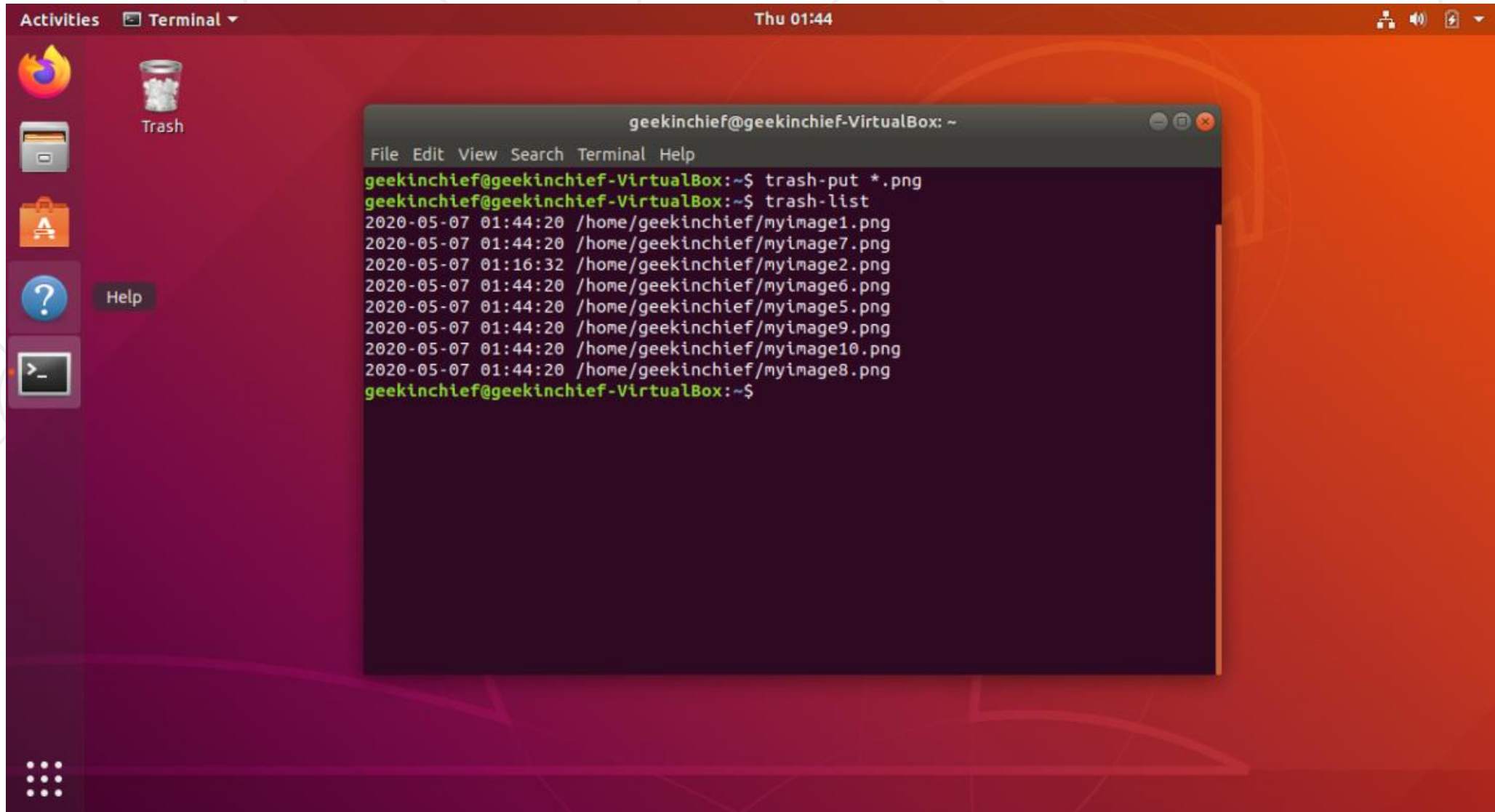
- **Mobile OS**, developed by Apple
 - Exclusively for its hardware devices: **iPhone, iPad and iPod Touch**
- Closed ecosystem, dominated by Apple
- iOS UI uses multi-touch gestures: **swipe, tap, pinch, and reverse pinch**
- iOS runs on **Apple hardware only**
 - Might run on PC emulators, but is illegal

Apple iOS – Smartphone and Tablet





- Linux is Free **and open-source family** of operating systems
 - Linux's **popularity** comes from its ease of customization and open license
- Offers **CLI shell** and many **GUI desktops**
- Many **distributions**: Ubuntu, CentOS, Debian, Mint, openSUSE, Alpine, ...
- It offers a **variety of options** for those who understand how to use it





Virtual Machines & Containers

Remote Instances & Emulators

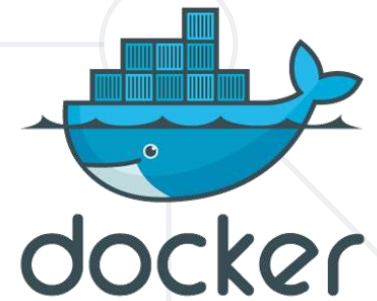
Virtual Machines (VM)

- A **virtual machine (VM)** is a software-based computer resource, used to run an OS inside another OS
 - Digital version of a physical computer that **can run programs** and **OS, store data, connect to networks**, and other computing functions
- **Virtualization** == running a **virtual machine (VM)** / virtual environment inside a physical hardware system
 - E. g. run Android VM or Linux inside a Windows host
 - Storage, networking, desktops can also be virtual

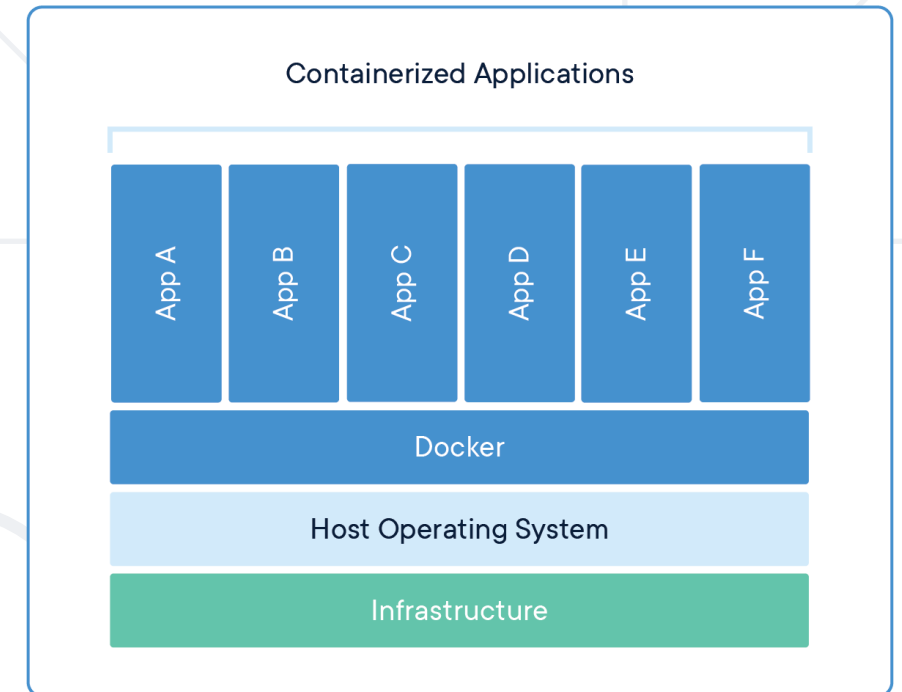


Containers and Docker

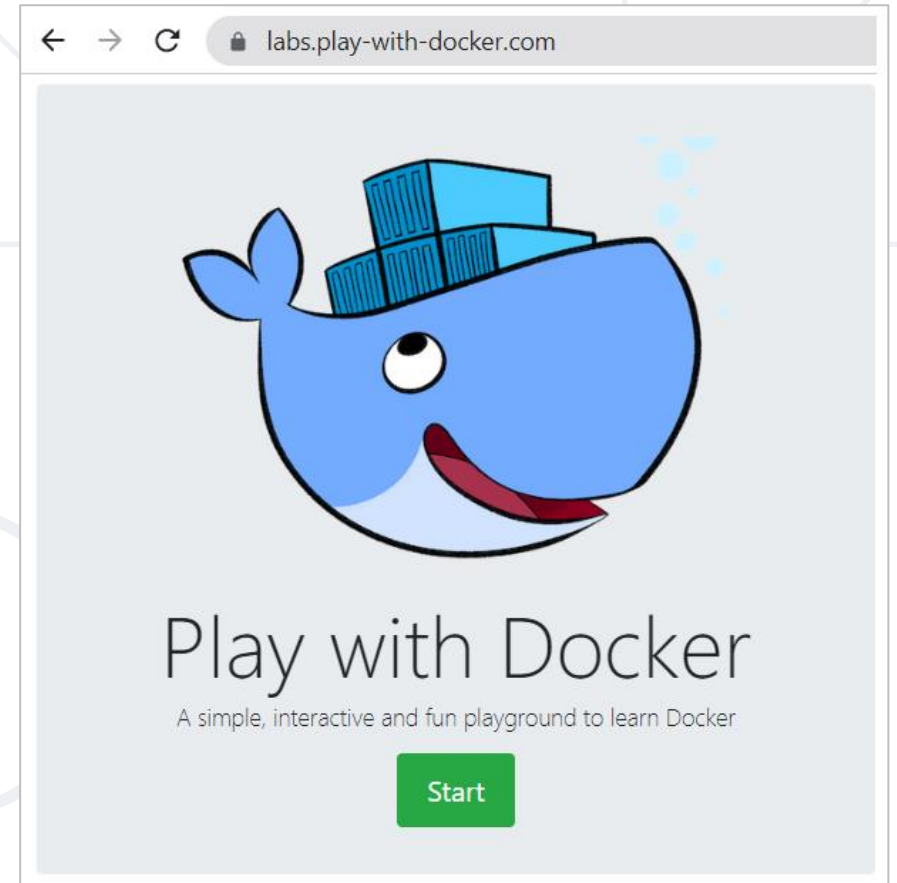
- **Container image** == software, packaged with its dependencies, designed to run in a virtual environment (like Docker)
 - E. g. WordPress instance (Linux + PHP + Apache + WordPress)
 - Simplified installation, configuration and deployment
 - **Lightweight** – containers use shared OS kernel with the host
- **Docker** is the most popular containerization platform
 - Runs **containers** from local **image** or downloaded from the **Docker Hub** online repository
 - Open-source, runs on Linux, Windows, Mac



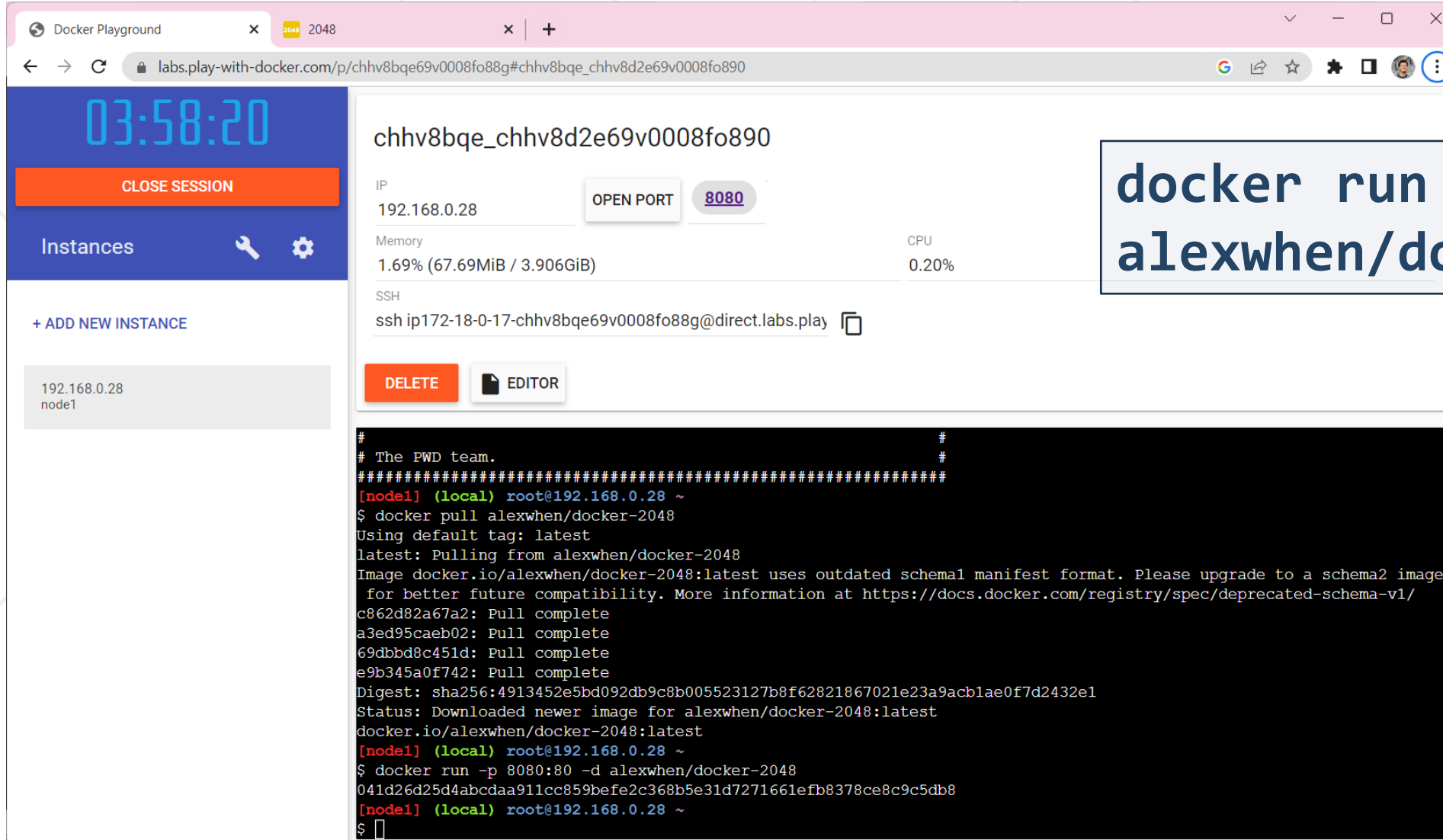
- A Docker **container image** is a lightweight, standalone executable package of software
 - Contains everything needed to run an app: code, runtime, libraries, tools, and settings
- **Container** == running Docker image
 - App, running inside the Docker Engine
- Containers provide fast and simple way to **run apps, without installing them** on the host OS
- Containers are **isolated** from the host and other containers → **security**



- Containers allow for **customizable** and **replicable instances** of an application
 - Without interfering with anything else on a user's system (no conflicts)
- **Docker Playground** is an interactive and fun way to learn Docker
 - Provides free Linux + Docker VMs
 - Accessible for 4 hours, for learning
 - <https://labs.play-with-docker.com>



Docker Playground – Live Demo



03:58:20

CLOSE SESSION

Instances

+ ADD NEW INSTANCE

192.168.0.28
node1

chhv8bqe_chhv8d2e69v0008fo890

IP
192.168.0.28

OPEN PORT 8080

Memory
1.69% (67.69MiB / 3.906GiB)

CPU
0.20%

SSH
ssh ip172-18-0-17-chhv8bqe69v0008fo88g@direct.labs.play

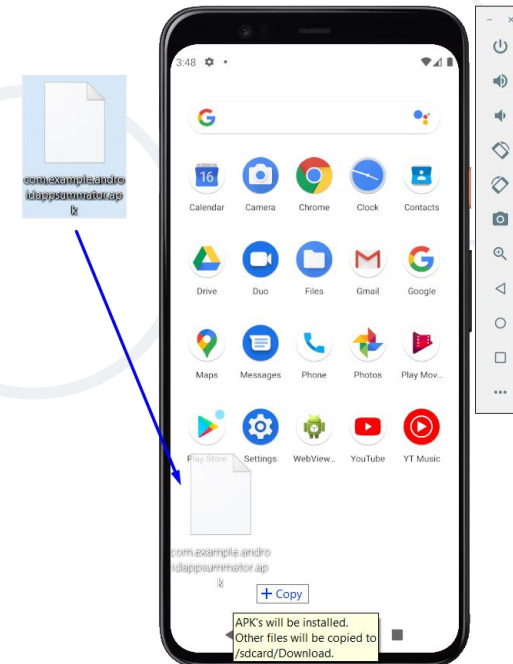
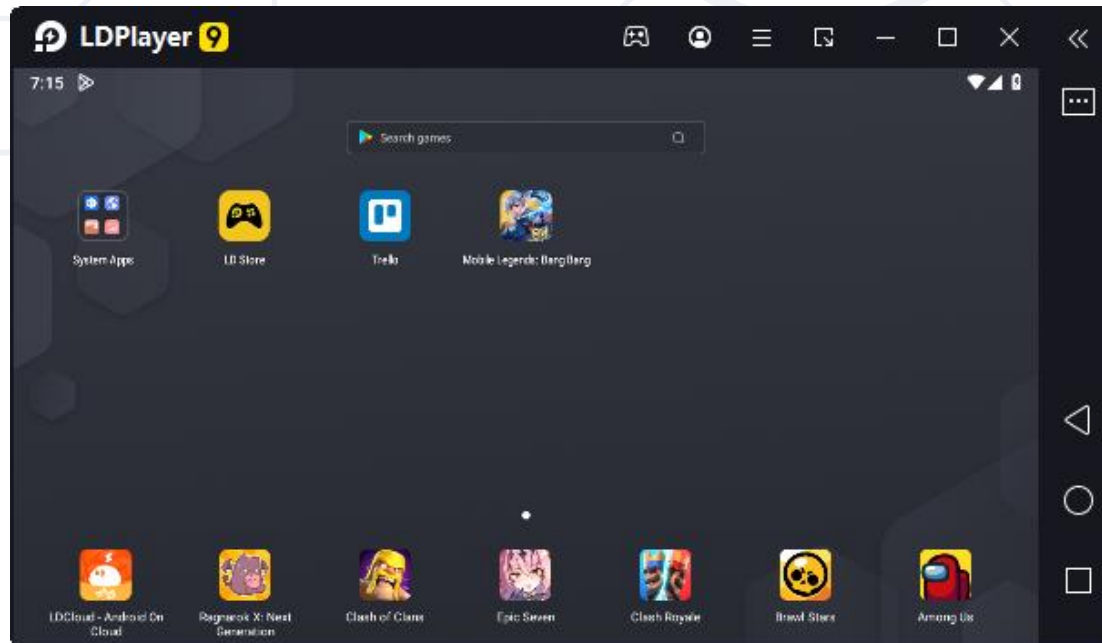
DELETE EDITOR

```
#  
# The PWD team.  
#####  
[node1] (local) root@192.168.0.28 ~  
$ docker pull alexwhen/docker-2048  
Using default tag: latest  
latest: Pulling from alexwhen/docker-2048  
Image docker.io/alexwhen/docker-2048:latest uses outdated schema1 manifest format. Please upgrade to a schema2 image  
for better future compatibility. More information at https://docs.docker.com/registry/spec/deprecated-schema-v1/  
c862d82a67a2: Pull complete  
a3ed95caeb02: Pull complete  
69dbbd8c451d: Pull complete  
e9b345a0f742: Pull complete  
Digest: sha256:4913452e5bd092db9c8b005523127b8f62821867021e23a9acb1ae0f7d2432e1  
Status: Downloaded newer image for alexwhen/docker-2048:latest  
docker.io/alexwhen/docker-2048:latest  
[node1] (local) root@192.168.0.28 ~  
$ docker run -p 8080:80 -d alexwhen/docker-2048  
041d26d25d4abcdaa911cc859befe2c368b5e31d7271661efb8378ce8c9c5db8  
[node1] (local) root@192.168.0.28 ~  
$
```

`docker run -p 8080:80
alexwhen/docker-2048`

Device Emulators

- **Device emulators** run Android / iOS / other OS in a virtual machines (VM) and simulate device functions (e. g. rotation)
- **BlueStacks, LDPlayer, Android Emulator** – run Android apps in Windows and simulate mobile devices

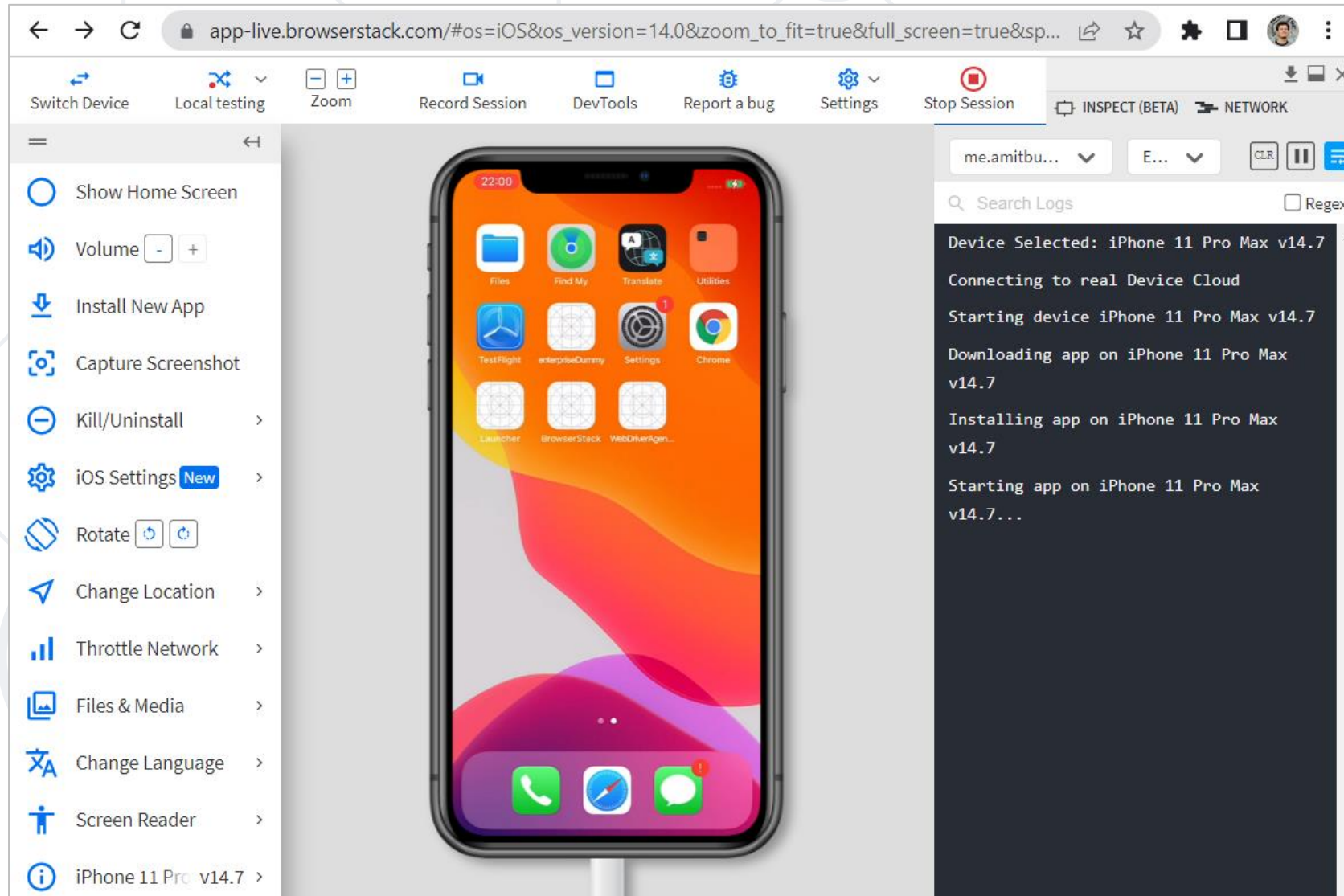


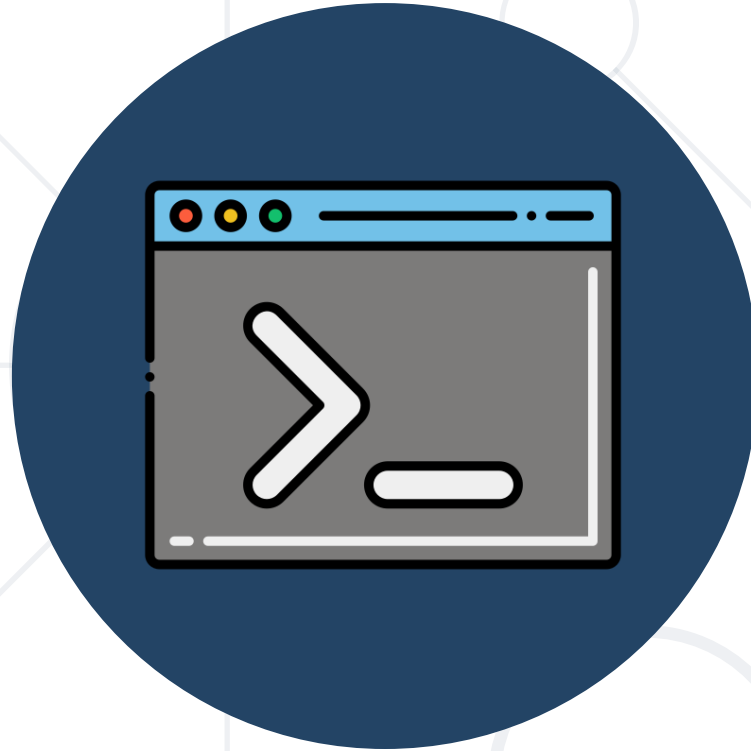
- **BrowserStack** – manual and automated online mobile testing for Web sites and mobile apps
 - Test on **remote physical devices**: iPhone, iPad, Samsung, Xiaomi, Google smartphones / tablets
 - Modern devices, modern Web browsers
 - Android, iOS, Windows, macOS
- **BrowserStack Live** offers **3000+ device-browser-OS** combinations for testing



BrowserStack

BrowserStack – Live Demo



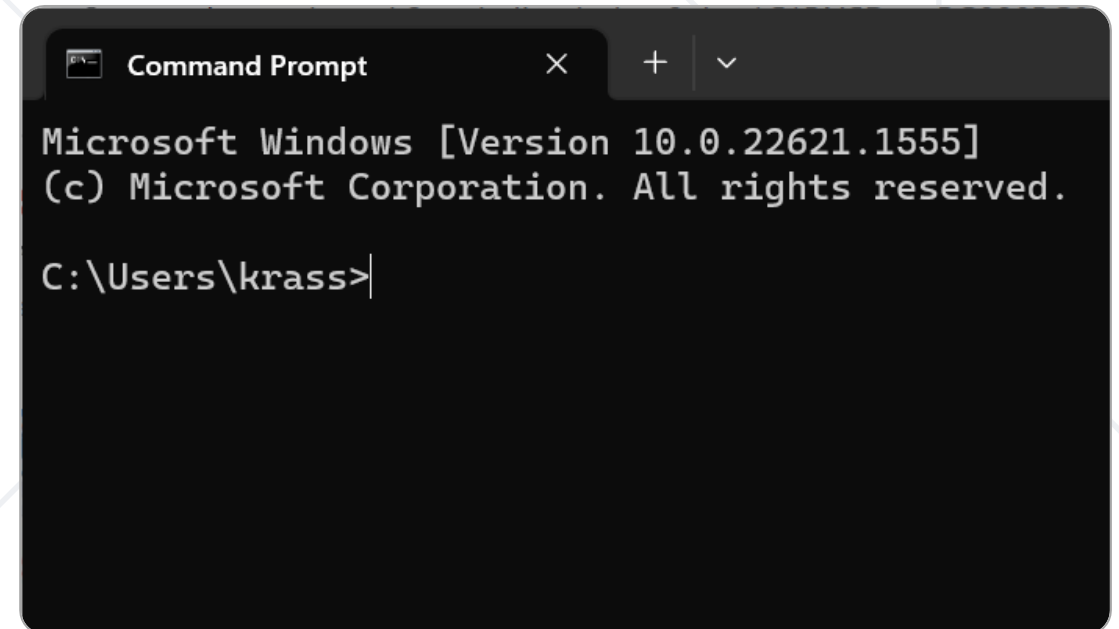
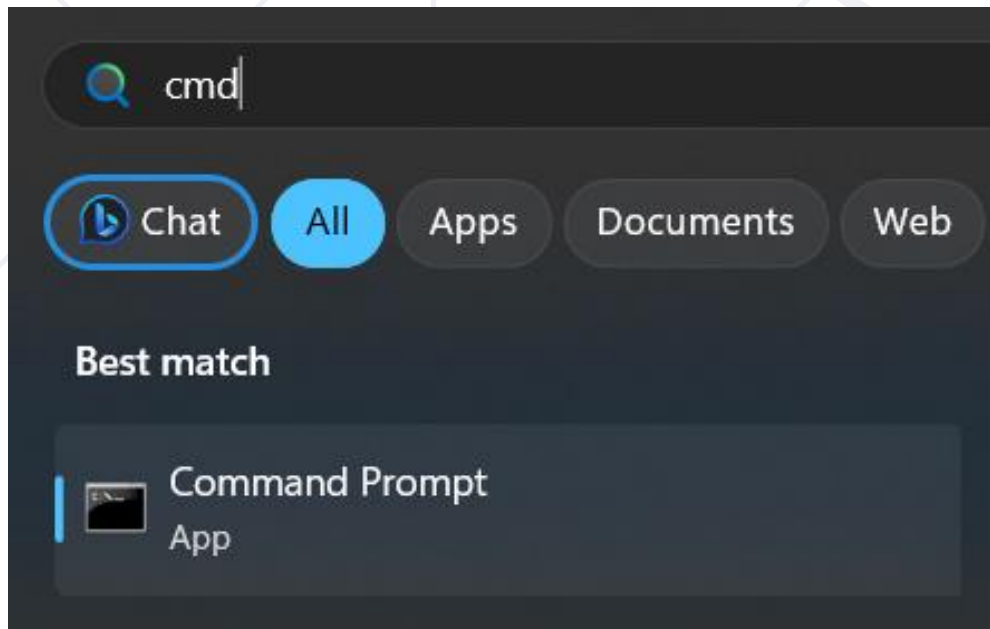


Shell & Shell Commands

Shell Command Execution on Linux and Windows

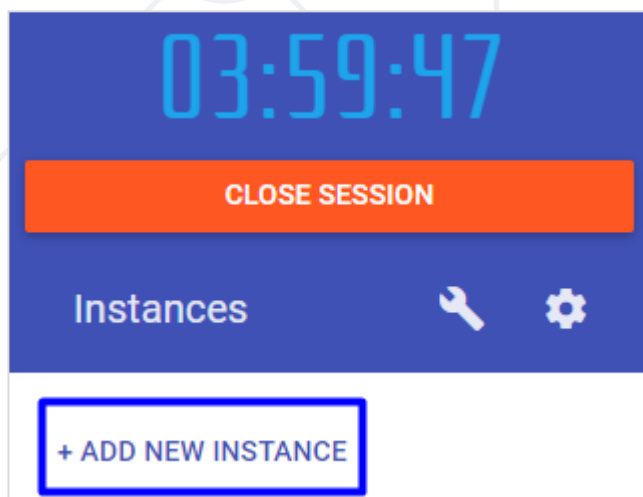
Opening the CLI Shell in MS Windows

1. Click [**Start**] -> [**Run**] or press [**Windows + R**] key
2. Type "**cmd**"
3. Click on [**Command Prompt**]



Linux Shell in Docker Playground

- Starting a **Docker Playground** session
 - Open **Docker Playground**, register and log in
 - Press **[Start]** and add a **new instance**
 - Now you have a **Linux VM + Docker environment** to experiment with



```
#####  
#                                     WARNING!!!!                               #  
# This is a sandbox environment. Using personal credentials                   #  
# is HIGHLY! discouraged. Any consequences of doing so are                   #  
# completely the user's responsibilities.                                     #  
#                                                                              #  
# The PWD team.                                                                #  
#####  
[node1] (local) root@192.168.0.13 ~  
$
```

Commands: ls & dir

- **ls** list files and directories in Linux / UNIX / macOS

```
user@host:~$ ls
```

```
user@host:~$ ls -al
```

```
nakov@Nakov-Laptop-HP:~$ ls -al
total 64
drwxr-xr-x 9 nakov nakov 4096 May 16 19:38 .
drwxr-xr-x 3 root  root  4096 Dec 11  2021 ..
-rw----- 1 nakov nakov 2520 May 17 01:04 .bash_history
-rw-r--r-- 1 nakov nakov  220 Dec 11  2021 .bash_logout
-rw-r--r-- 1 nakov nakov 3771 Dec 11  2021 .bashrc
drwx----- 3 nakov nakov 4096 Mar 27 12:56 .cache
drwx----- 5 nakov nakov 4096 Mar 27 12:56 .config
```

- **dir** lists the files and folders in Windows

```
C:\Users\nakov> dir
```

```
C:\Users\nakov>dir
Volume in drive C is Nakov's SSD
Volume Serial Number is B295-4B6D

Directory of C:\Users\nakov

09-May-23  14:32    <DIR>          .
29-Sep-22  18:44    <DIR>          ..
11-May-23  19:23    <DIR>          .android
28-Apr-23  14:58    <DIR>          .azure
16-May-23  21:35             1 112 .bash_history
07-Mar-23  21:55    <DIR>          .cache
```



Commands: cd

- **cd** changes the current working directory in Linux

```
user@host:~$ cd /home
user@host:~/home$ ls -al
```

```
nakov@Nakov-Laptop-HP:~$ cd /home
nakov@Nakov-Laptop-HP:/home$ ls -al
total 12
drwxr-xr-x  3 root  root  4096 Dec 11  2021 .
drwxr-xr-x 19 root  root  4096 May 17 11:16 ..
drwxr-xr-x  9 nakov nakov 4096 May 16 19:38 nakov
```

```
user@host:~/home$ cd ..
user@host:~/$ ls -al
```

- **cd** works the same way in Windows

```
C:\Users\nakov> cd ..
C:\Users> dir
```

```
C:\Users>dir
Volume in drive C is Nakov's SSD
Volume Serial Number is B295-4B6D

Directory of C:\Users

29-Sep-22  18:44    <DIR>          .
29-Sep-22  18:47    <DIR>          defaultuser100000
09-May-23  14:32    <DIR>          nakov
29-Sep-22  21:43    <DIR>          Public
18-Jan-22  15:13    <DIR>          svetl
```



Commands: pwd / cd

- **pwd** prints the current working directory in Linux

```
user@host:~$ pwd
```

```
nakov@Nakov-Laptop-HP:~$ pwd  
/home/nakov
```

- **cd** works the same way in Windows

```
C:\Users\nakov> cd
```

```
C:\Users\nakov>cd  
C:\Users\nakov
```



Commands: echo and cat / echo and type

- **echo '...' > filename** prints a text to a file in Linux
- **cat** displays the content of given file
- **echo ... > filename** prints a text to a file in Windows
- **type** displays the content of given file

```
echo 'Hi Linux' > hi.txt  
cat hi.txt
```

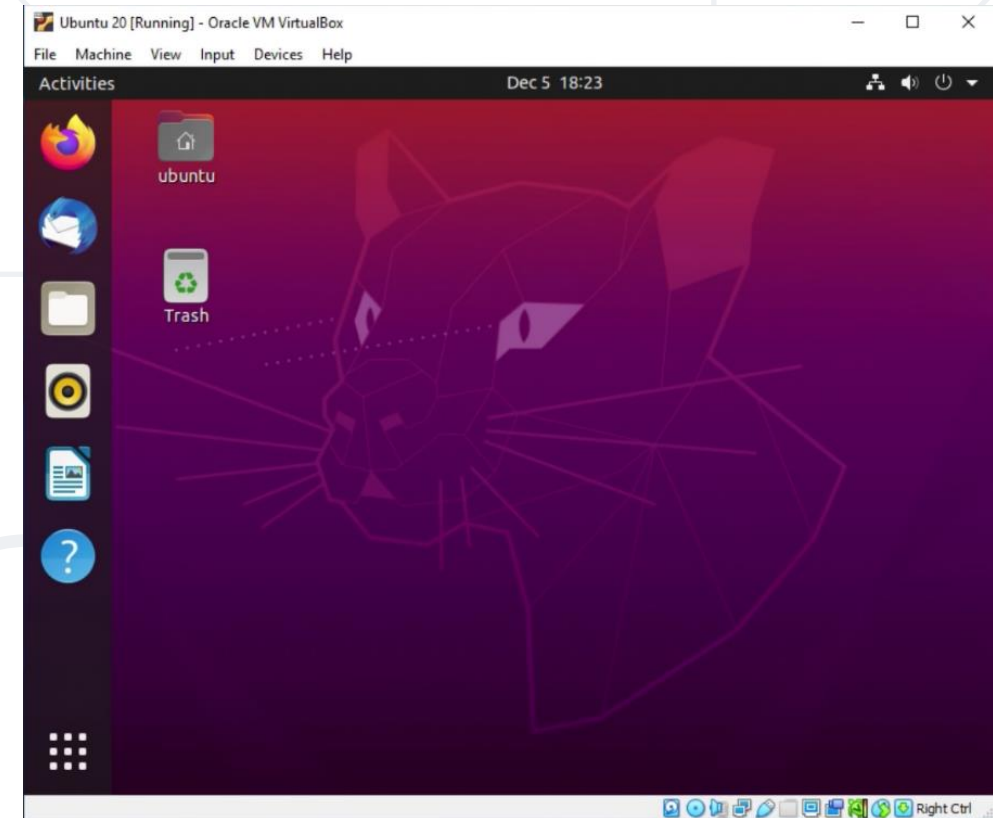
```
nakov@Nakov-Laptop-HP:~$ echo 'Hi Linux' > hi.txt  
nakov@Nakov-Laptop-HP:~$ cat hi.txt  
Hi Linux
```

```
echo Hi Windows > hi.txt  
type hi.txt
```

```
C:\Users\nakov>echo Hi Windows > hi.txt  
C:\Users\nakov>type hi.txt  
Hi Windows
```

Can I Run Linux Commands on Windows PC?

- You can run **Linux** in **Windows** through a **virtual machine**
 - E. g. Ubuntu Linux in Virtual Box
- You can run Linux in **Windows Subsystem for Linux (WSL)**





Windows Subsystem for Linux

Apps | Developer tools
Microsoft Corporation

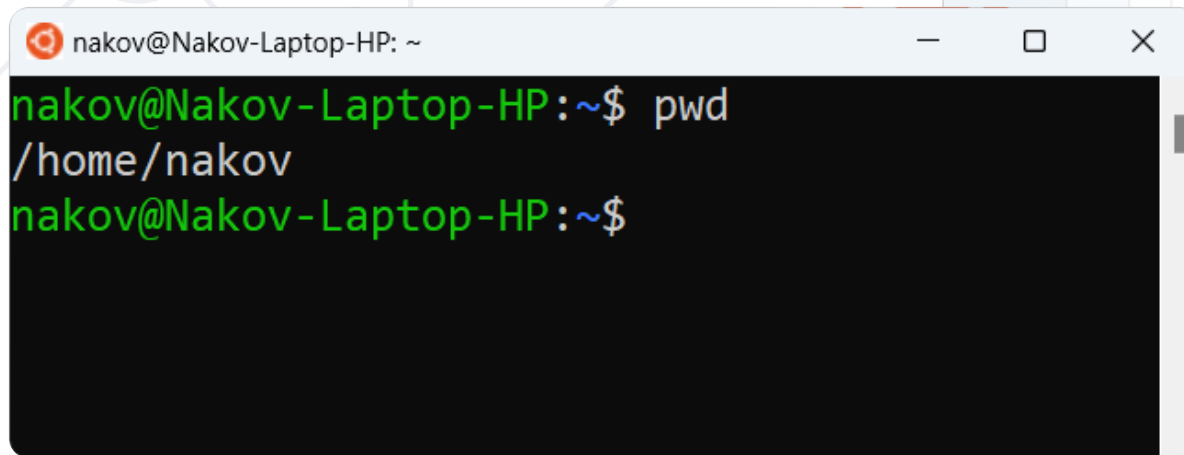
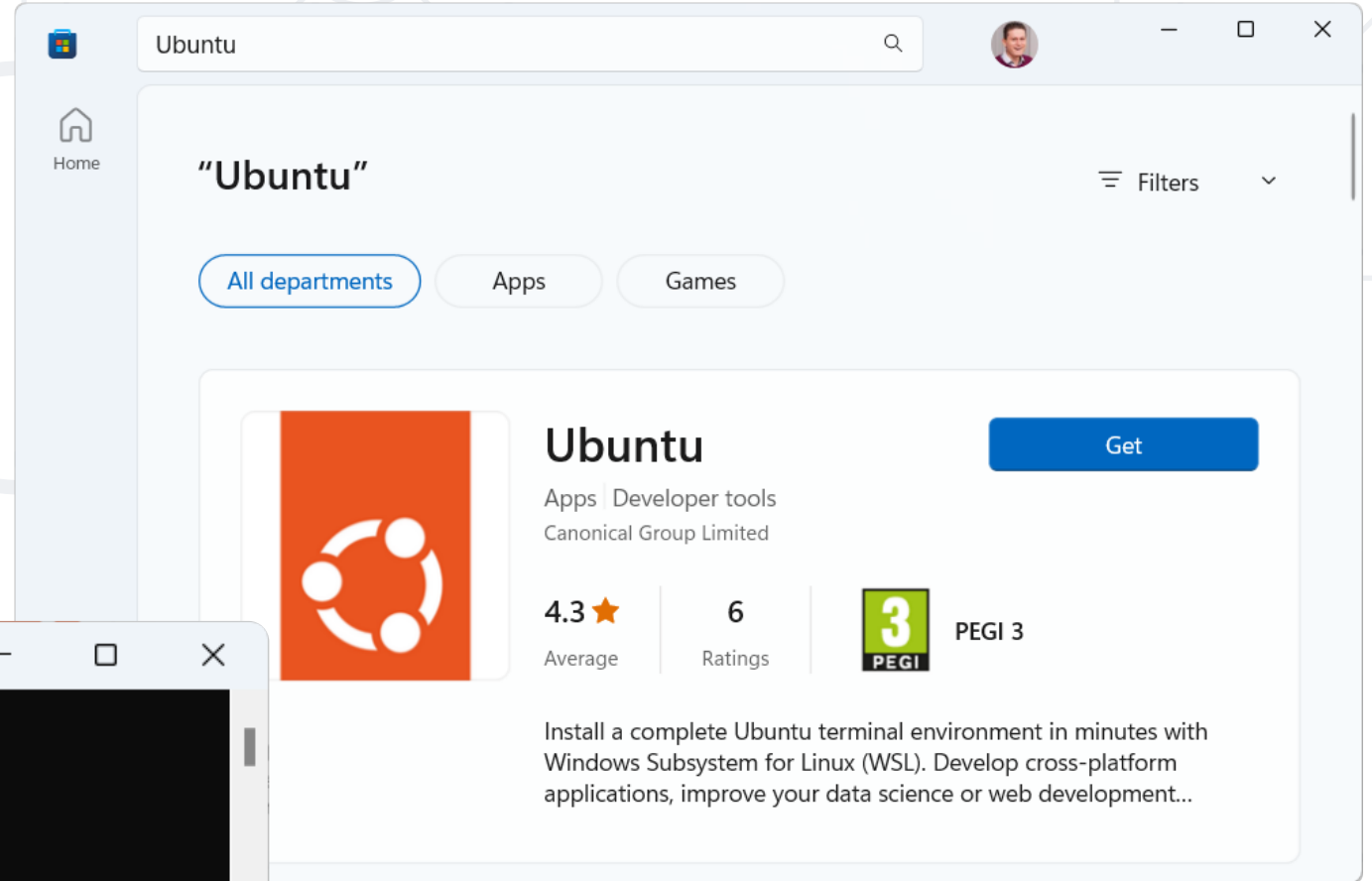
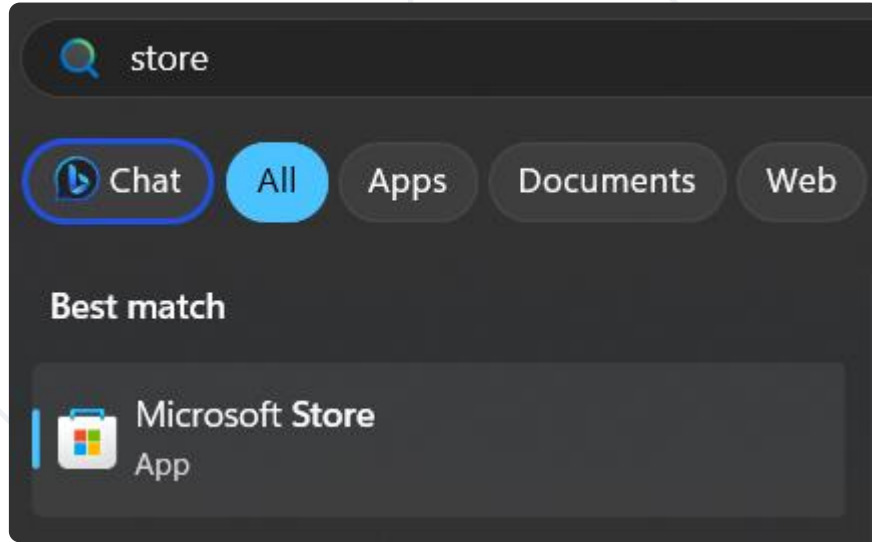
3.9 ★
Average

7
Ratings

 PEGI 3

Windows Subsystem for Linux (WSL) lets developers run a GNU/Linux environment -- including most command-line tools, utilities, and applications -- directly on Windows, unmodified, without the overhead of a traditional...

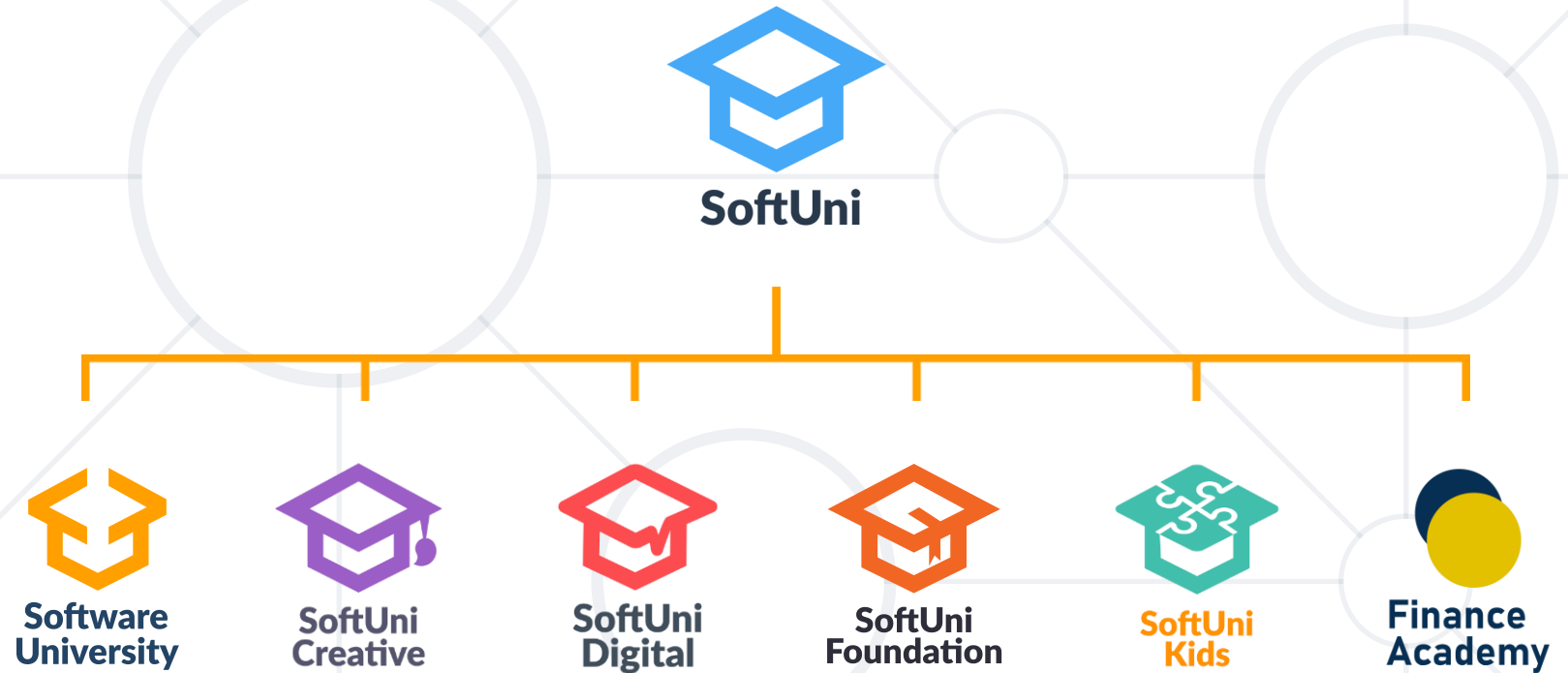
Install WSL and Ubuntu Linux in Windows 11



- **Operating Systems (OS)** manage processes, users, files and other resources
- **OS Examples:** Windows, macOS, Linux, Android, iOS
- **Virtual machine** (VM) == OS inside another OS
- **Container** == app image, running in Docker
- **Shell commands** == execute commands from the console (Linux / Windows shell)



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

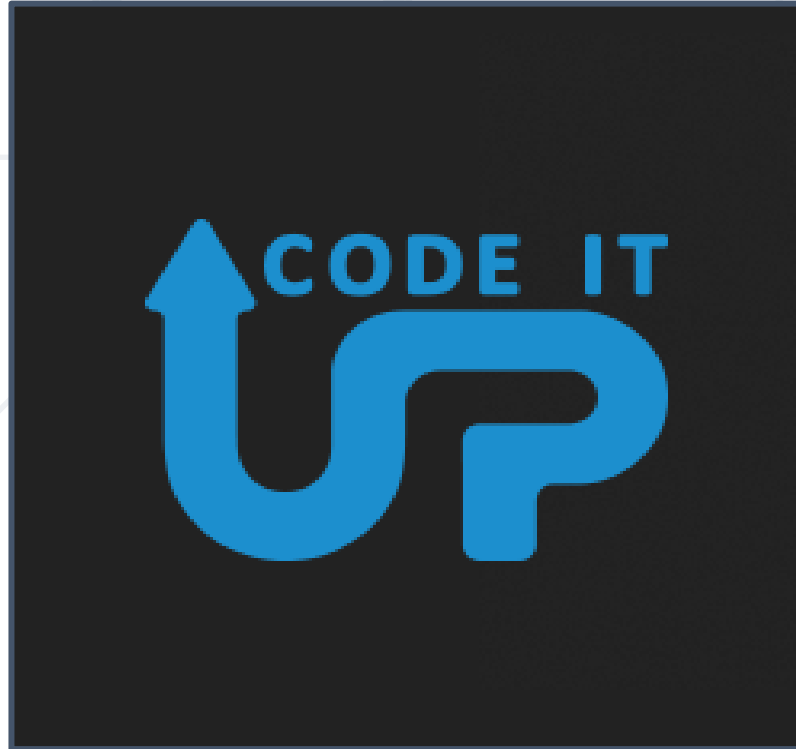


BOSCH

DXC
TECHNOLOGY



SmartIT



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

- softuni.bg, about.softuni.bg

- Software University Foundation

- softuni.foundation

- Software University @ Facebook

- facebook.com/SoftwareUniversity

- Software University Forums

- forum.softuni.bg



- 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>

