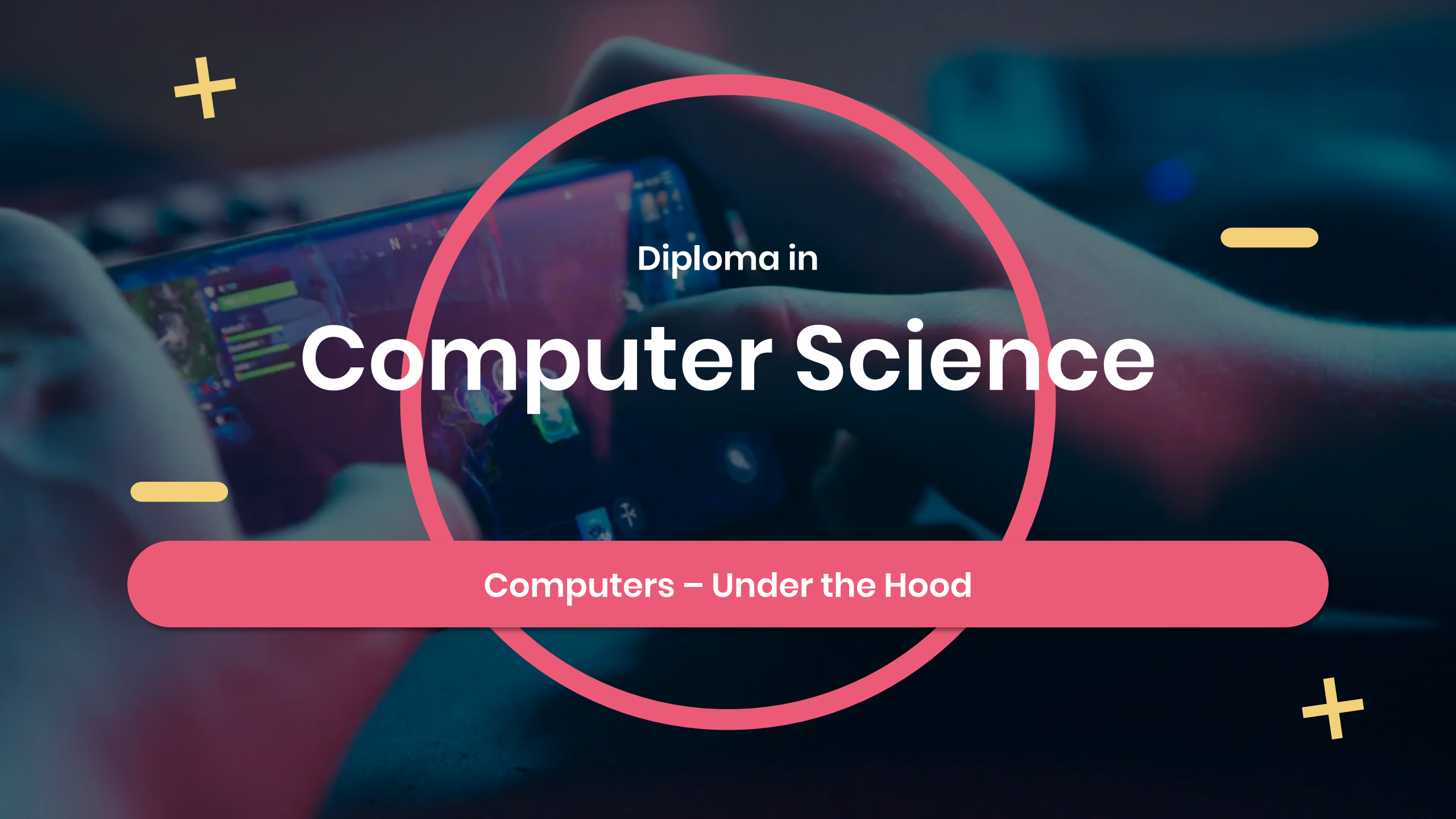


Diploma in

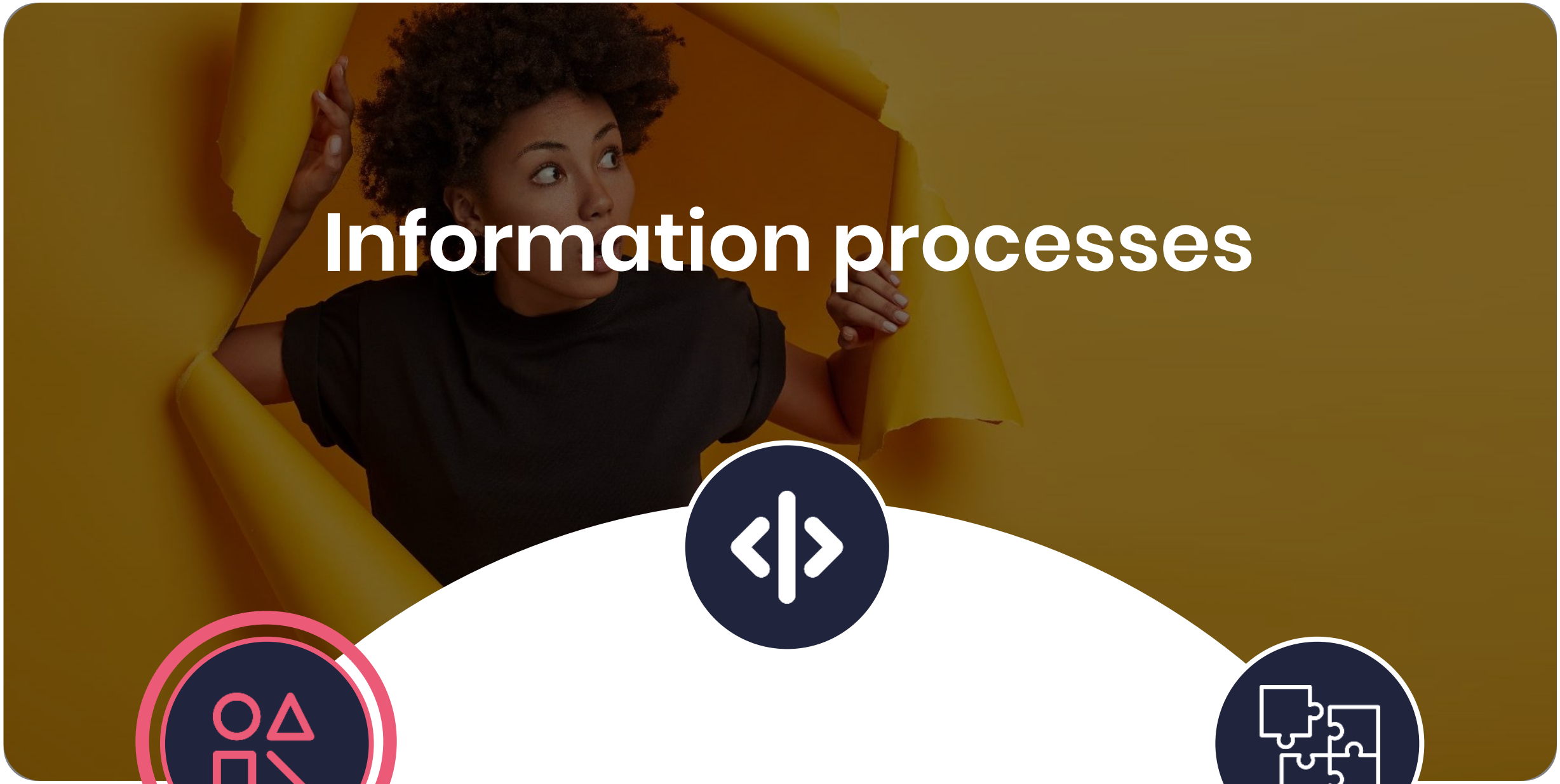
Computer Science

Computers – Under the Hood

Objectives

- Understand information processes
- Examine the structure of operating systems
- Identify types of operating systems

Timestamp
00:00
00:00
00:00



Information processes

Example: Baking a cake

- Gather all the ingredients
- Follow a recipe and mix the ingredients
- Add the mixture to the oven
- Take out the cake

Data collection

- First step in information processing
- Data is collected different sources
- Sensors gather information from the environment
- Data must be format set by program

Input example: flour

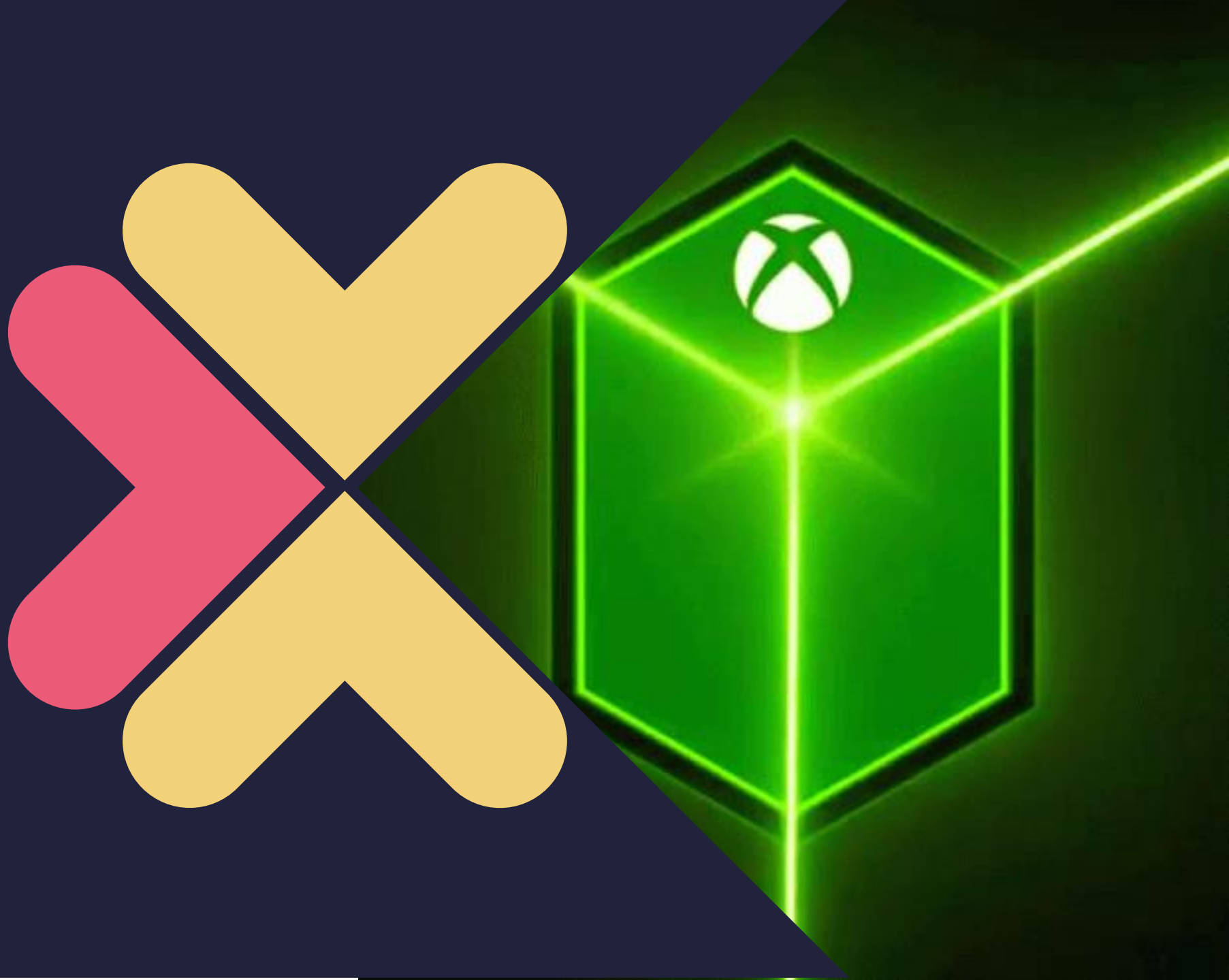
- A sheaf of wheat cannot be used as is
- Must be ‘formatted’ as flour
- Applies to all ingredients

Data

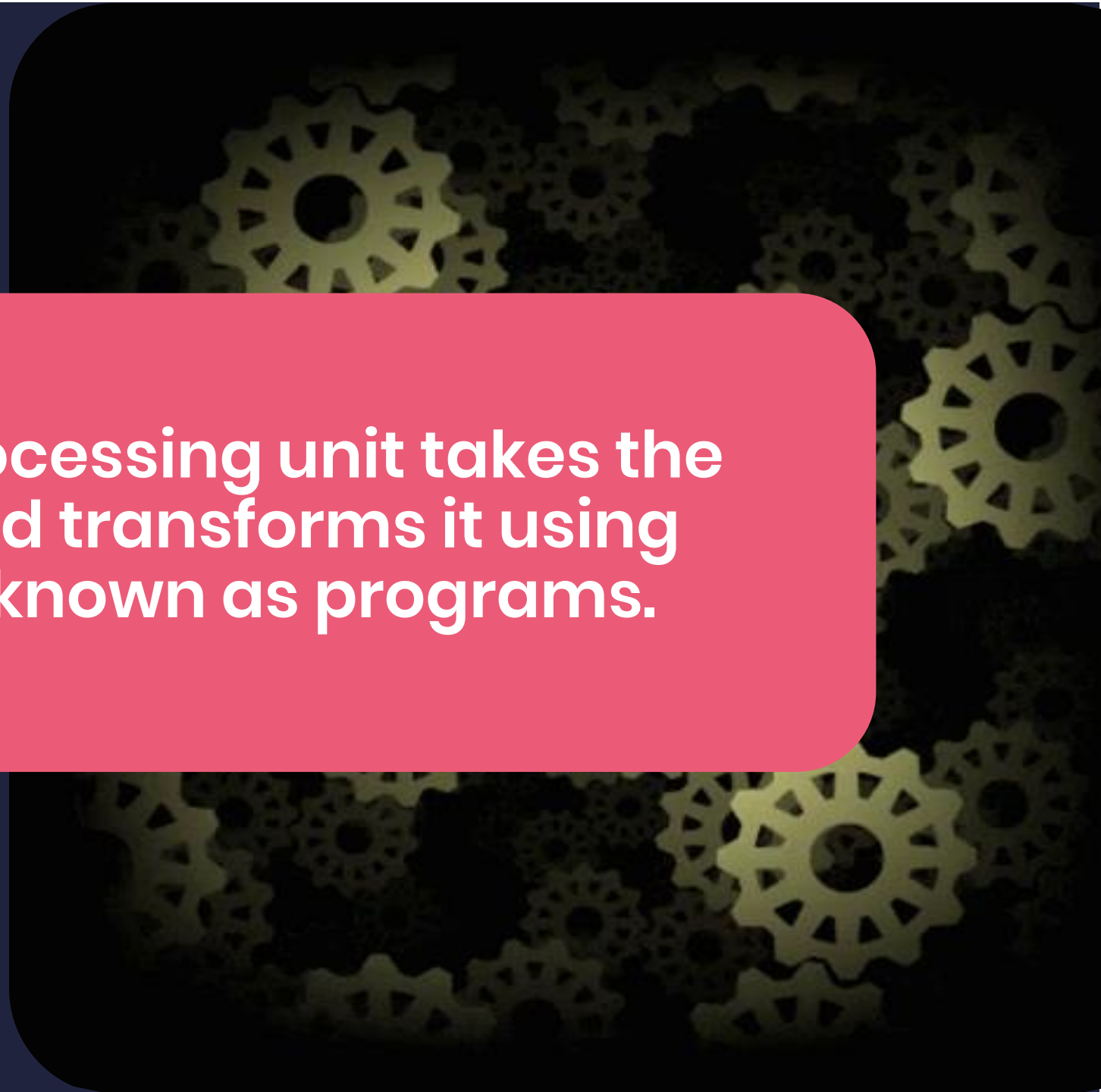
- Generally stored in a file or database
- Converted into machine-readable format by input devices, such as keyboards

DID YOU KNOW?

More than 122 years of gameplay have been streamed from Xbox One to Windows 10 devices!



The central processing unit takes the input data and transforms it using instructions known as programs.

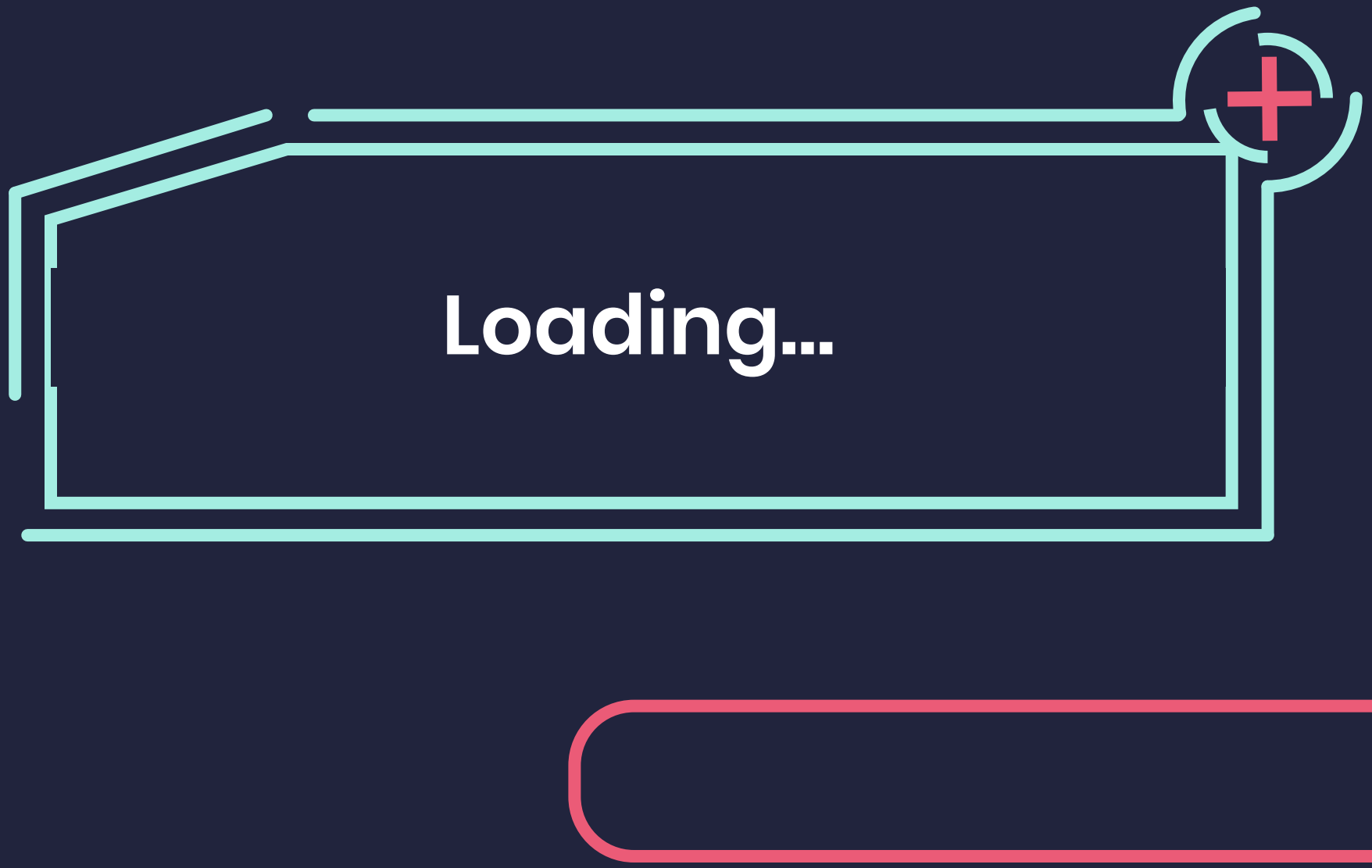


Program = recipe

The program gives the steps.

The central processing unit processes everything.

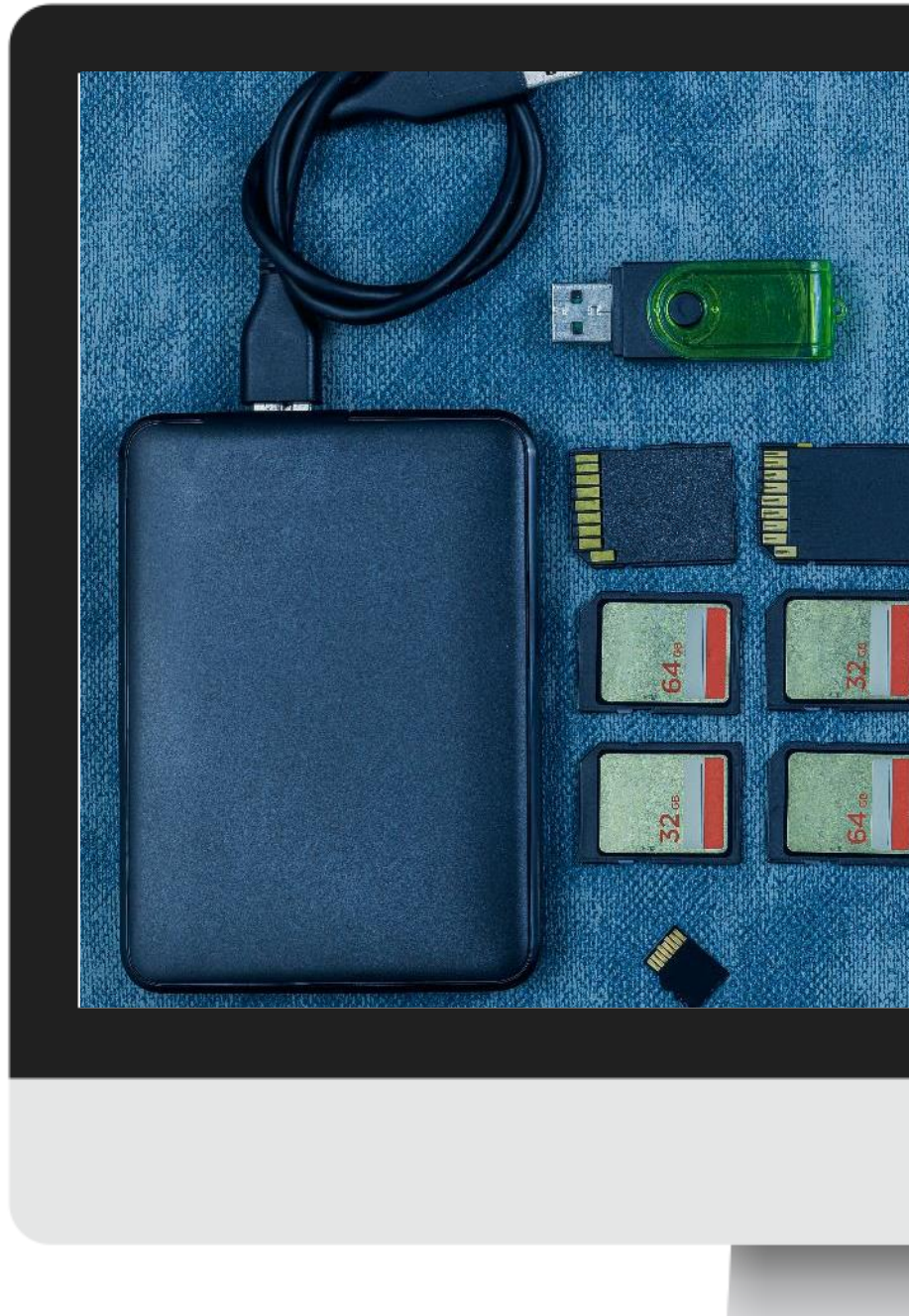
The output is information.

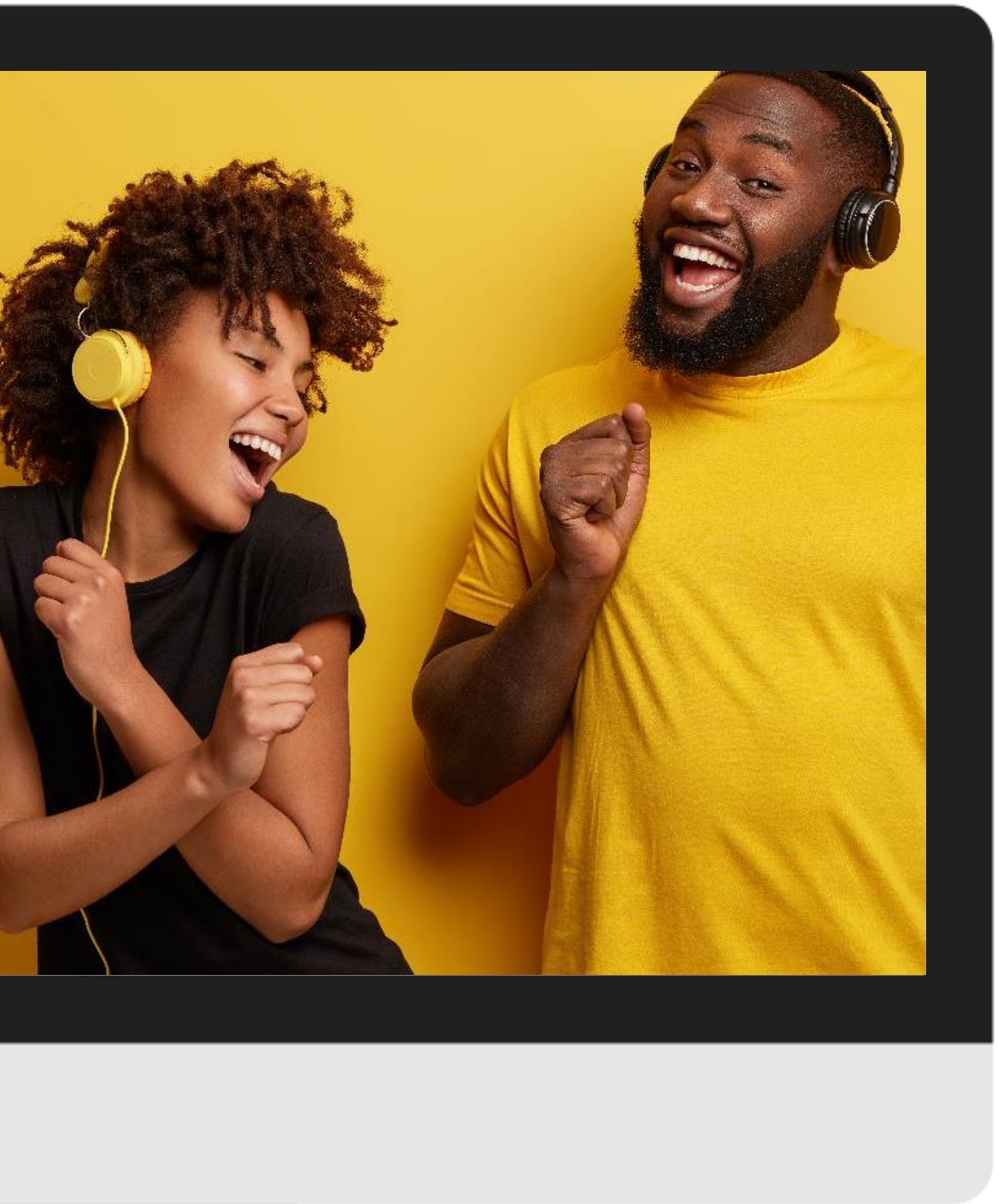

Loading...

Data storage and retrieval

- Processed data needs to be stored
- Temporary storage is called memory or storage (RAM)
- RAM loses all content when there is no power
- When information is needed, it is moved to storage

Examples: computer drive, flash drive, phone storage

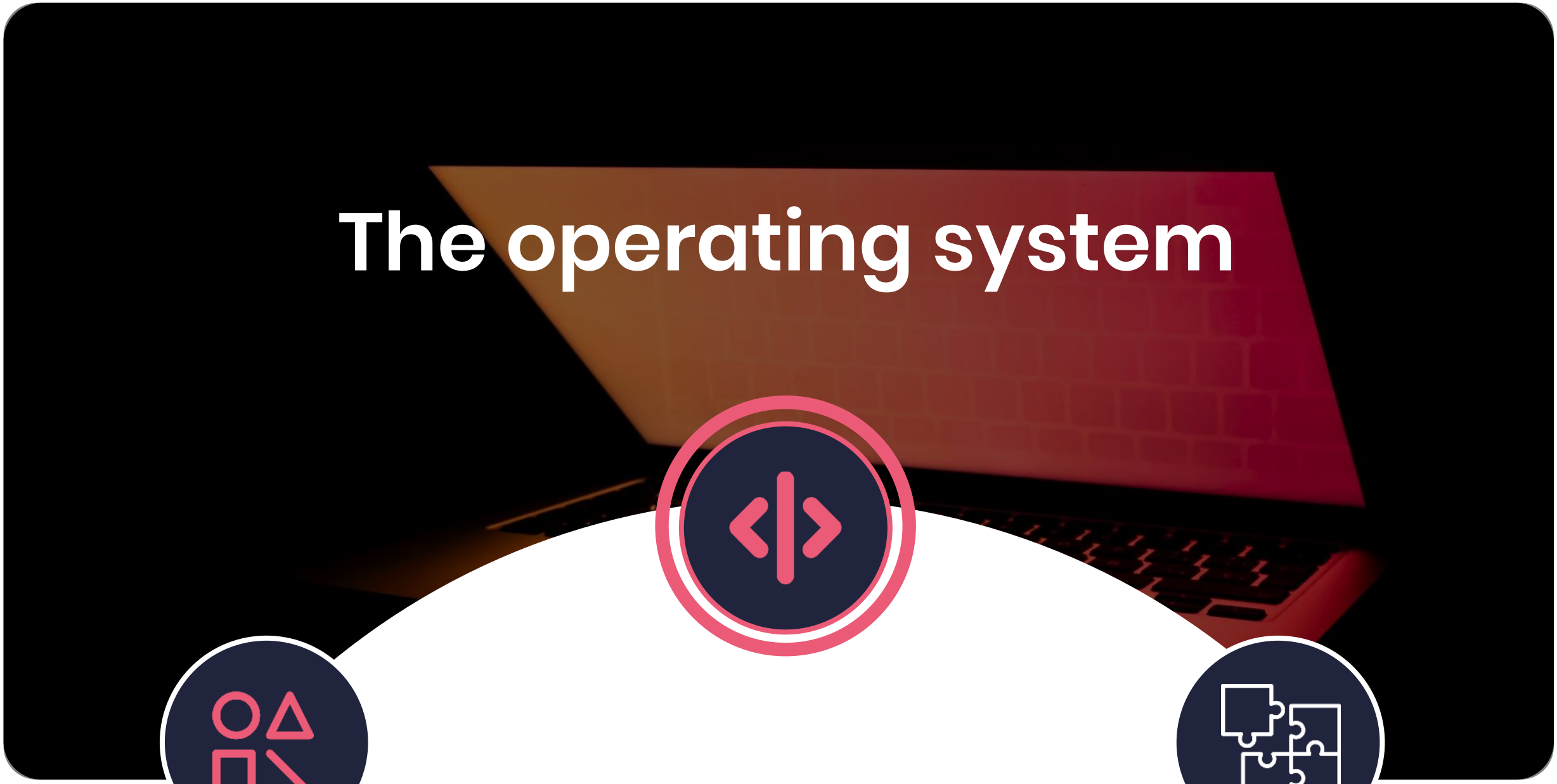




Displaying

- Most common way to present information is visual output - screens
- Visual output can also be printed
- Output can also be presented as sound or actions

The operating system



What is an operating system?

A set of manual and automatic procedures that enable a group of people to efficiently **share** a computer installation.

Operating system structure

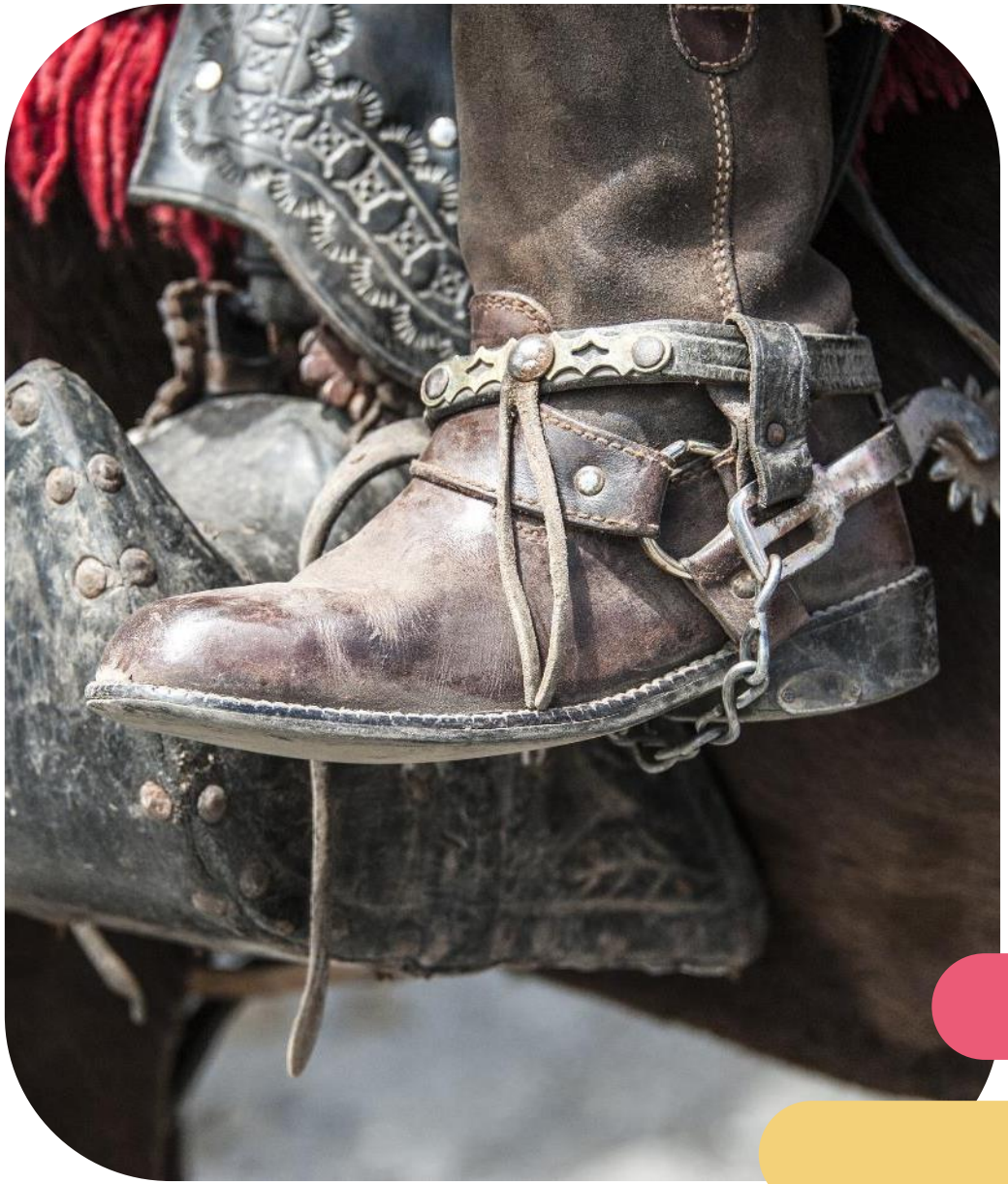
- Kernel at the core
- Has complete control over everything in the system
- Always resident in the memory
- Consists of device drivers

What do cowboys and computers have in common?

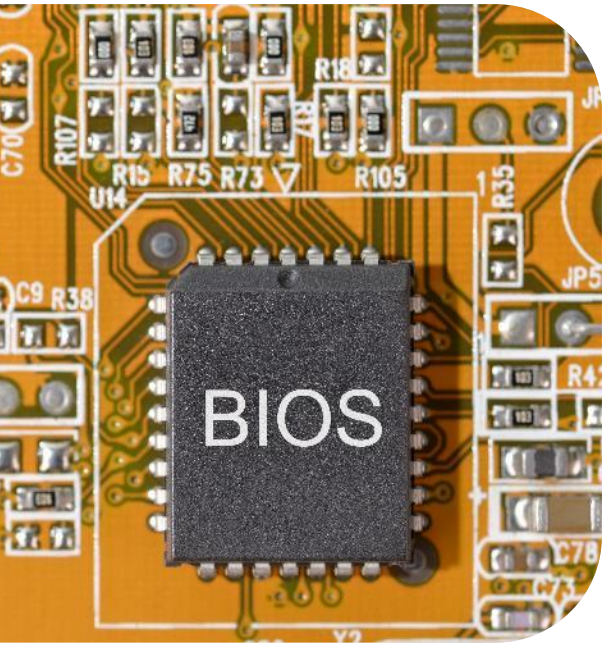
Answer:
The word ‘boot’

Booting

- Pressing the power button initiates booting
- Instructions need to be pulled from storage



Booting and BIOS



BIOS initialises hardware, performs tests, loads boot loader

Bootloader locates OS, loads kernel into RAM

Kernel takes over, loads secondary services, drivers, apps

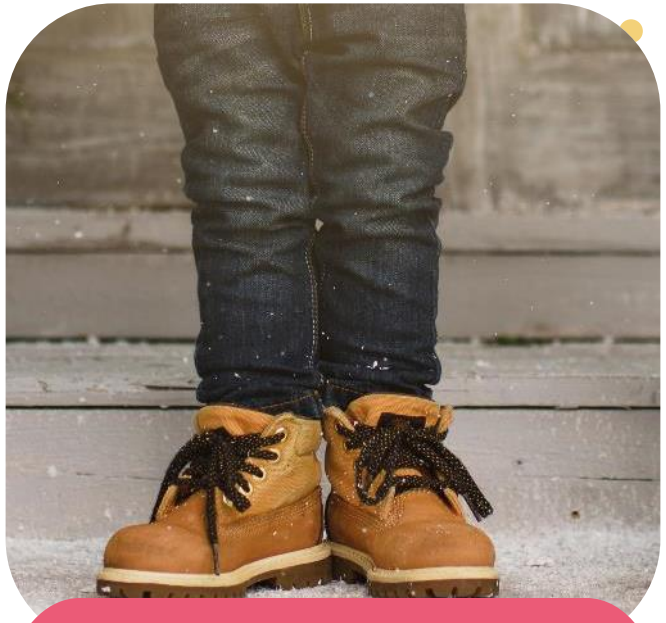
Runtime state achieved

Computers are state machines

- They go through many states to accomplish tasks
- Beginning of the boot process is the zero state
- Rebooting resets the machine to its zero state

Rebooting

Hard reboot: Power is cut off from the system

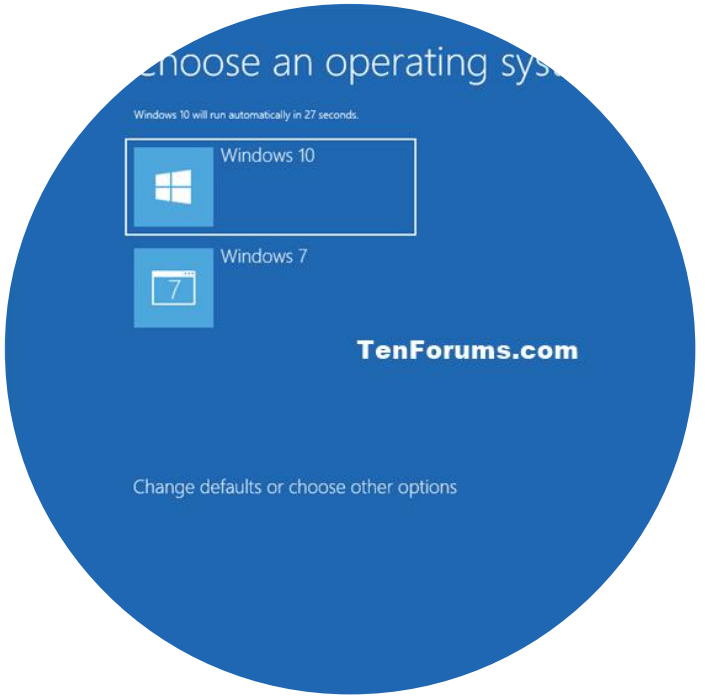


Soft reboot: Only RAM contents cleared to zero

Types of bootloaders

Android uses OEM specific bootloaders

Linux uses Grub



Windows uses Windows Boot Manager

Why do we need to know about operating systems?

OS roles

Memory management

Tells apps what can and cannot stay in RAM

Processor management

Tells the processor which tasks to run, when and for how long

OS roles

Device management

Manages devices, preventing conflict between hardware

File management

Keeps track of files and ensures they are not overwritten

OS roles

Security

You aren’t able to access anything you’re not supposed to

System performance

Prevents fighting for time with the system

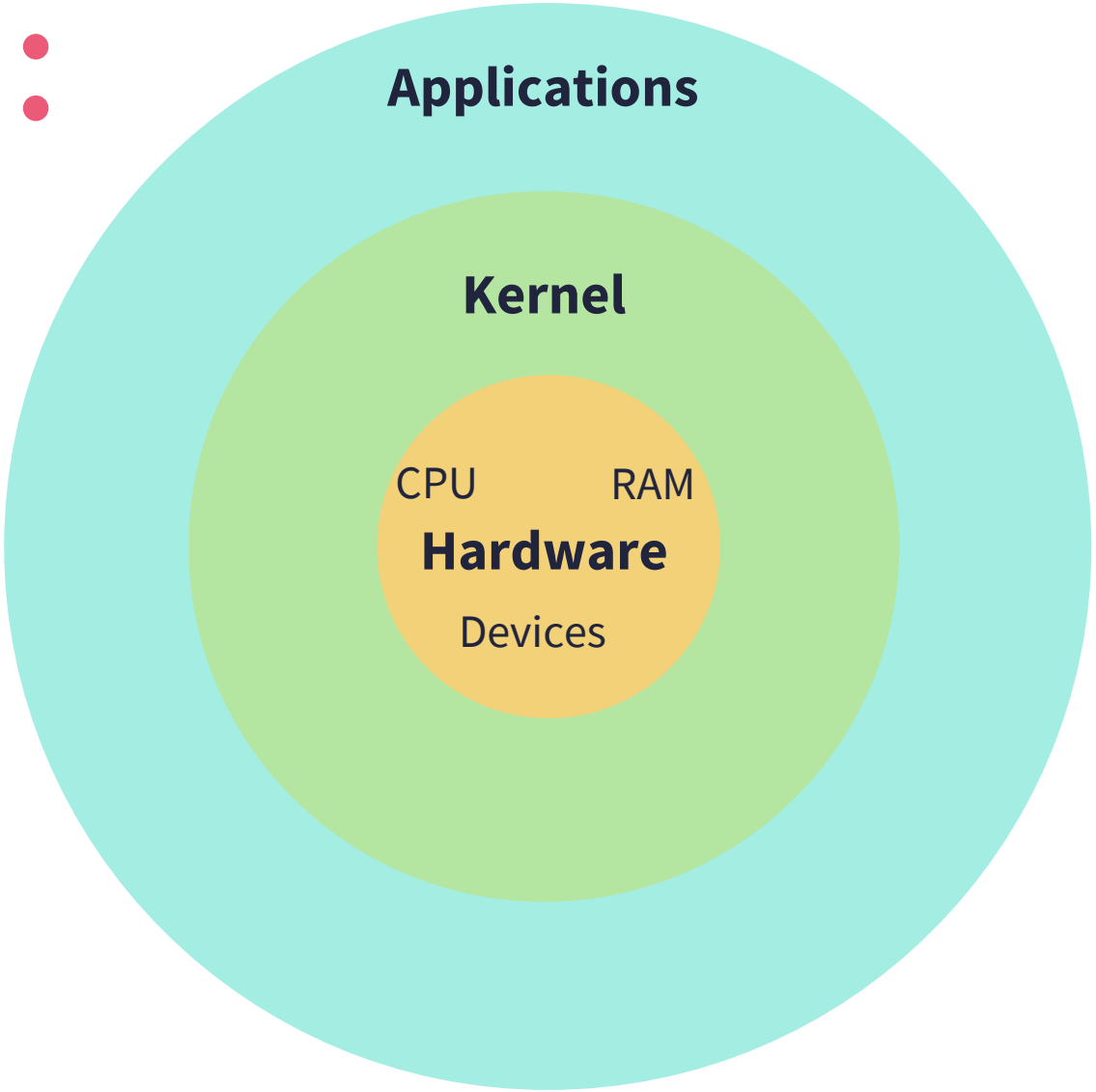
OS roles

Error handling

Handles errors internally, preventing computer crashes

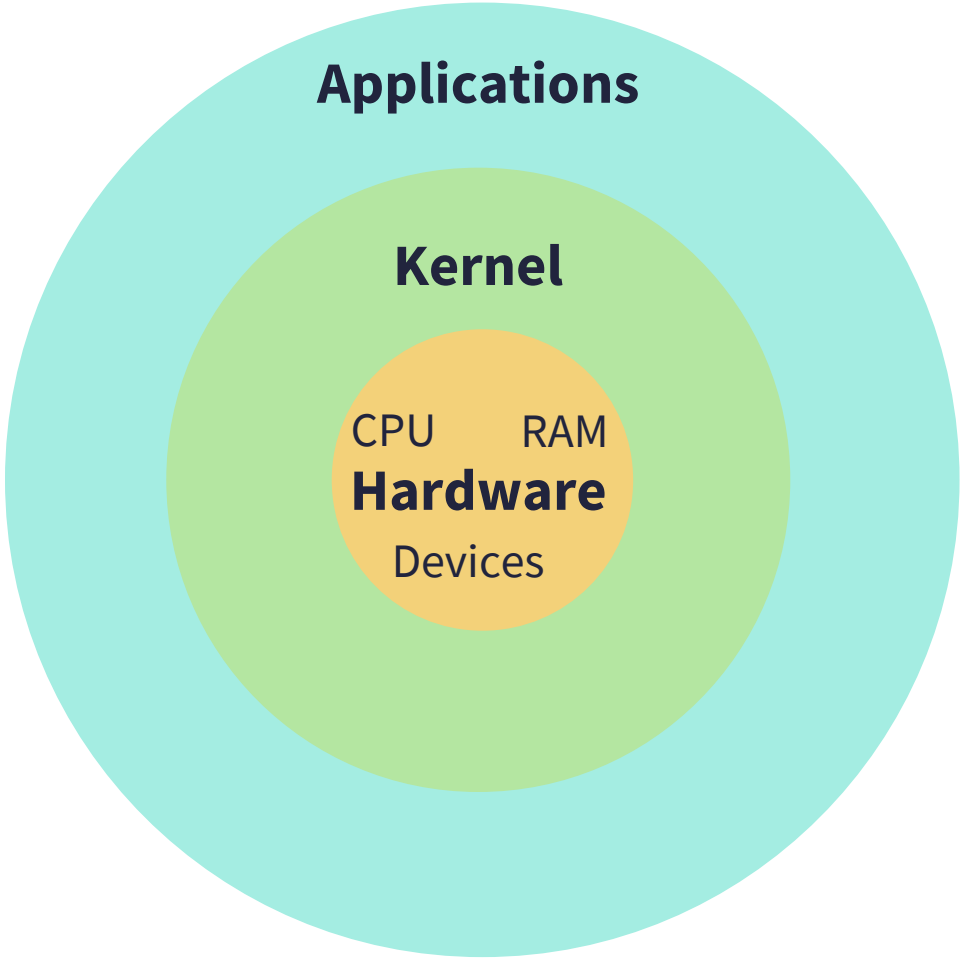
Application platform

Provides the platform on which applications can run



The operating system

Types of kernels

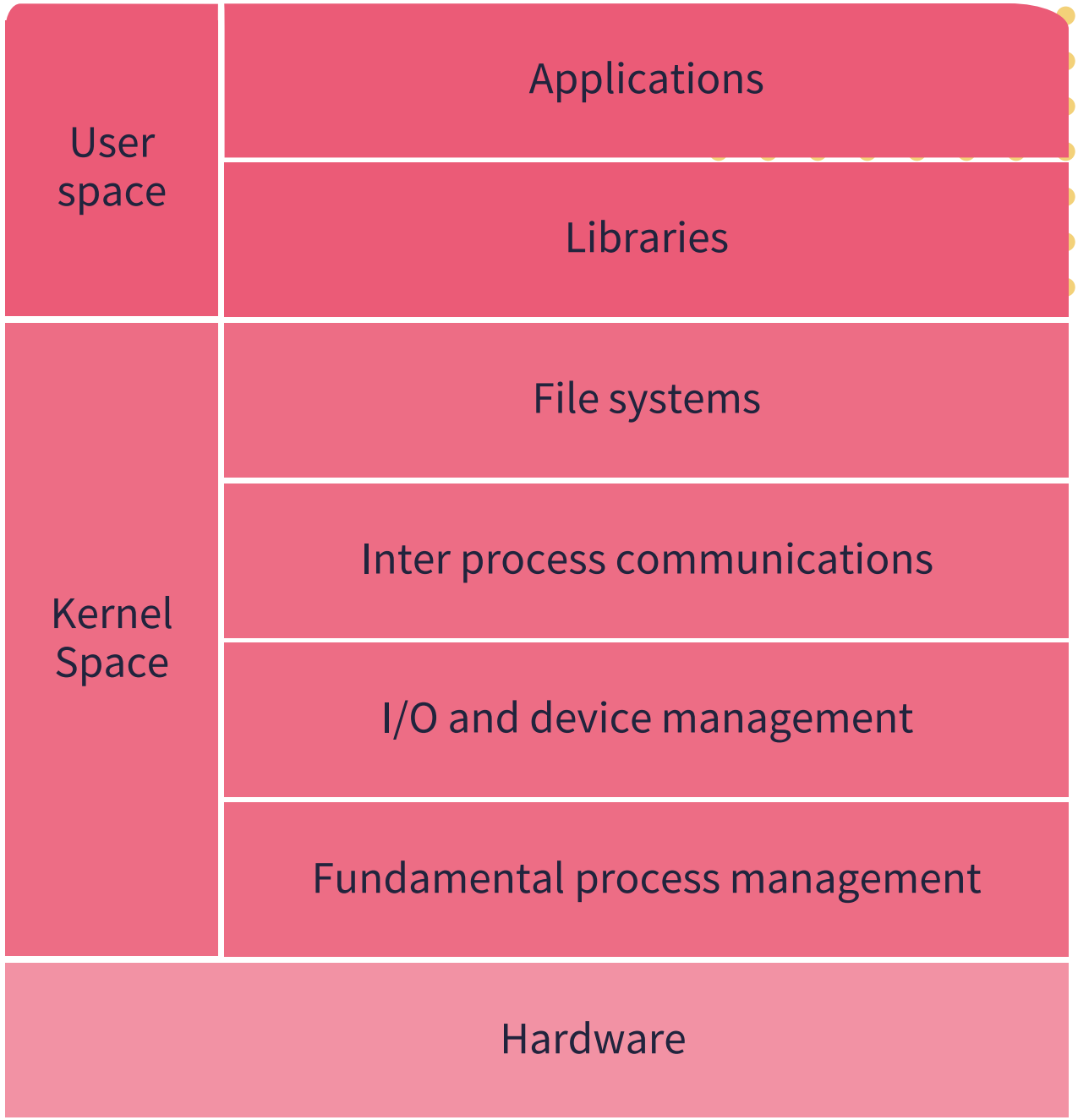


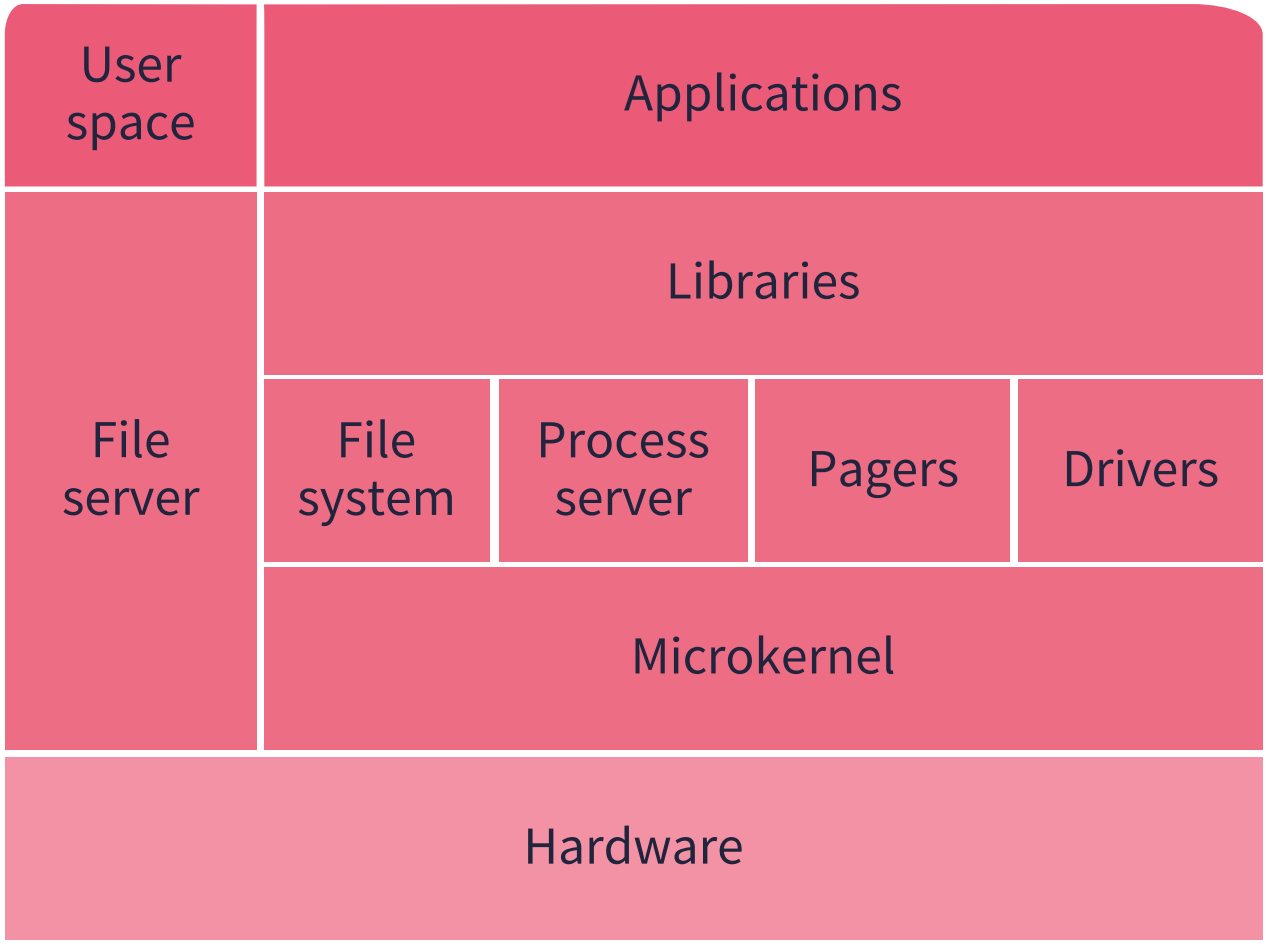
- Monolithic kernel
- Microkernel
- Hybrid kernel
- Nanokernel
- Exokernel

Monolithic kernel

- User and kernel services in the same space
- Increases size of kernel but means faster execution

Example: Linux systems





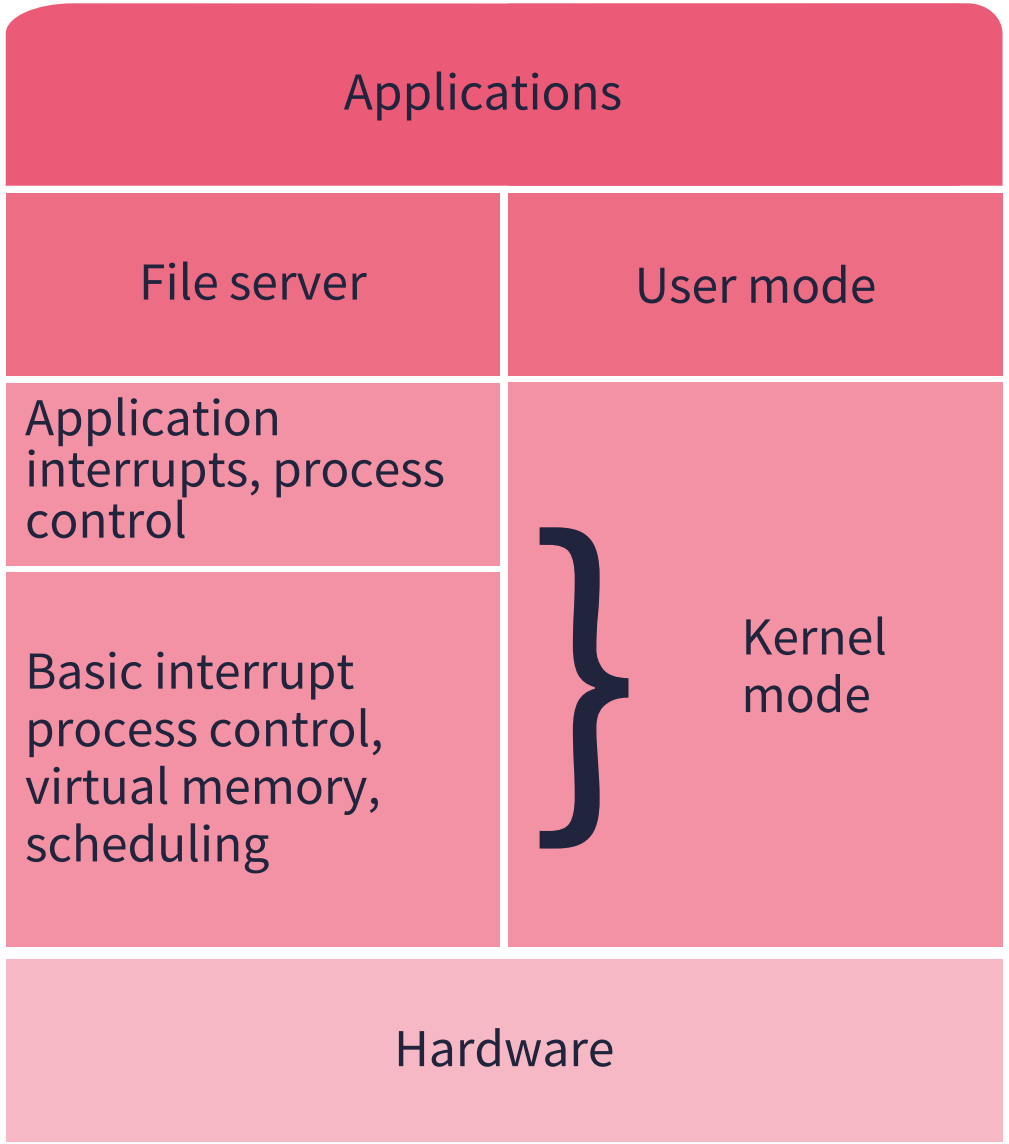
Microkernel

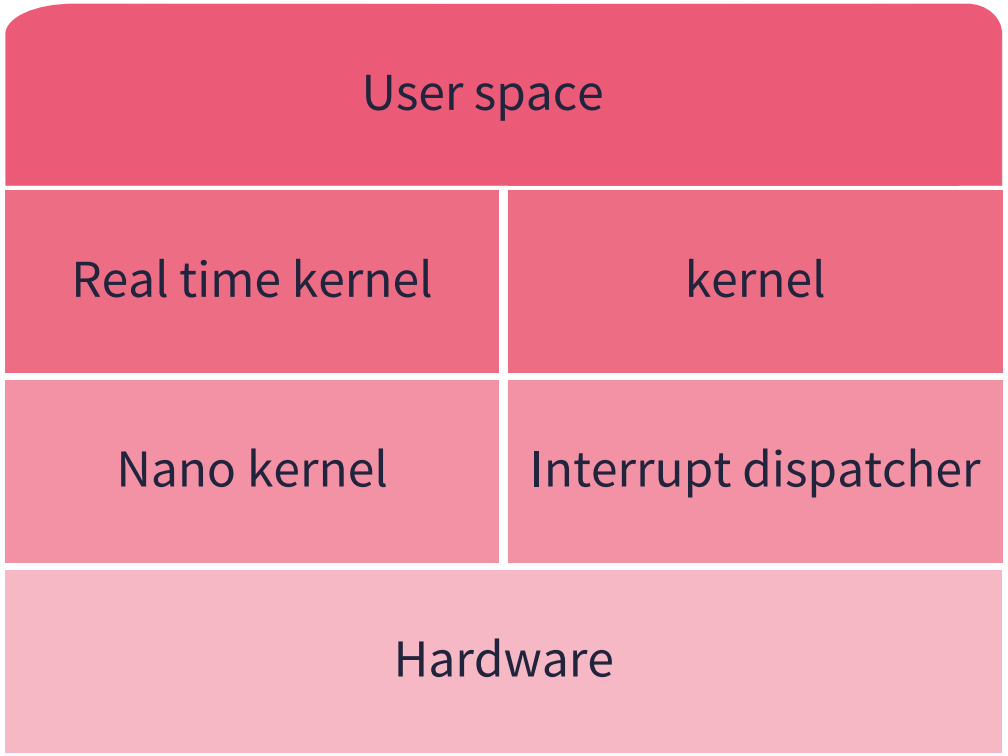
- User space and kernel services are separate
- Reduces the size of the OS
- Uses message parsing for communication between applications and services

Hybrid kernel

- Combination of the monolithic and microkernel
- Speed (monolithic) and modularity (micro)

Example: Windows



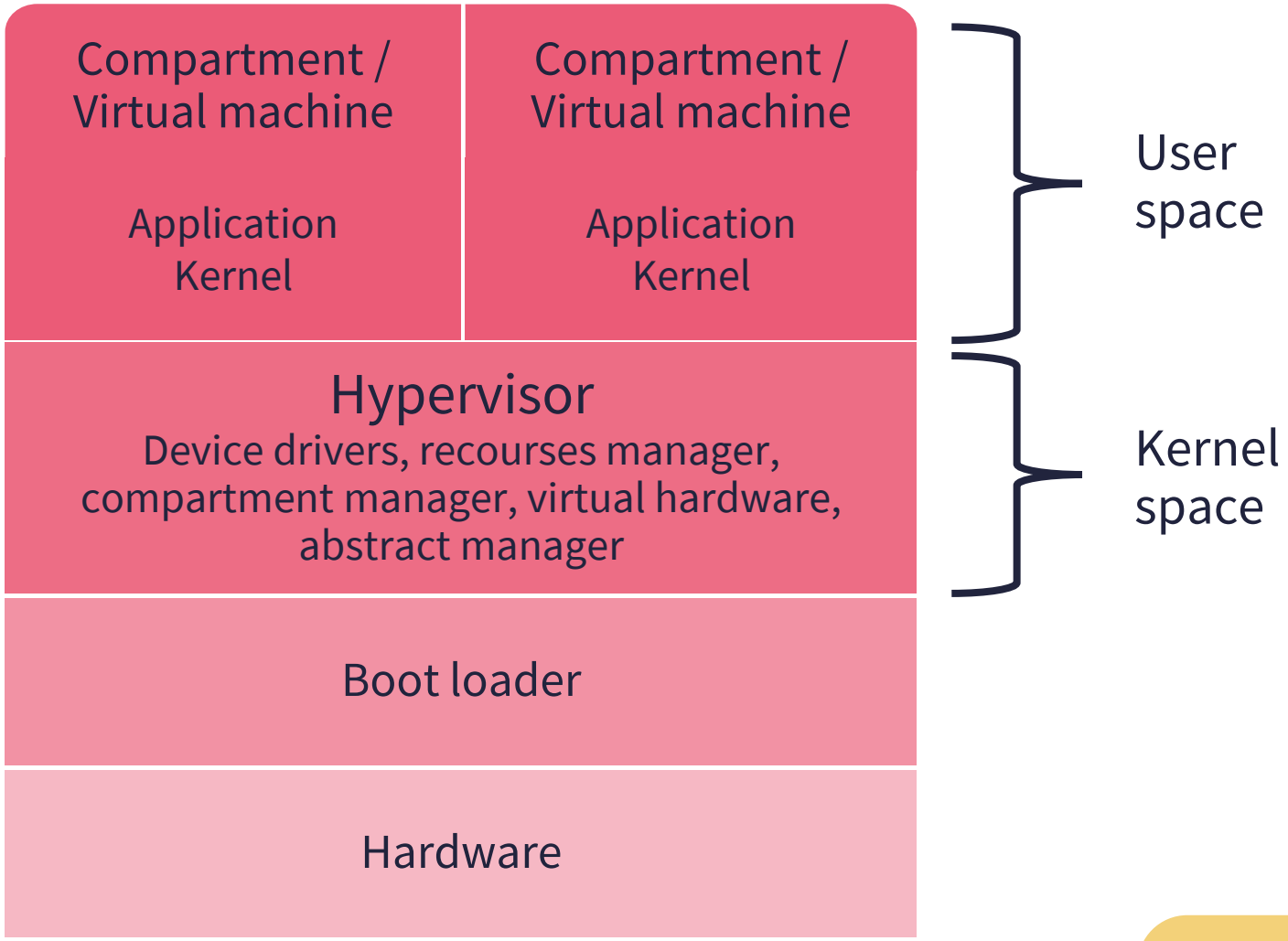


Nanokernel

- Has a very small code base
- Offers hardware abstraction but no system services

Exokernel

- Complex kernel design where as few abstractions as possible are imposed
- Allows for application-specific customisation



Separate spaces

Operating systems separate processing into kernel-space processing and user-space processing to prevent a catastrophic failure.

Smart components

- Keyboard, mouse, drives, monitor, fingerprint sensor, USB ports have control mechanisms
- Used to start and stop the device, and for computer to communicate directly with devices

Device drivers

- The software that sits between a computer and a device
- A shared library of privileged, memory-resident, low-level hardware handling routines

Device drivers

- All hardware devices abstracted and represented by files

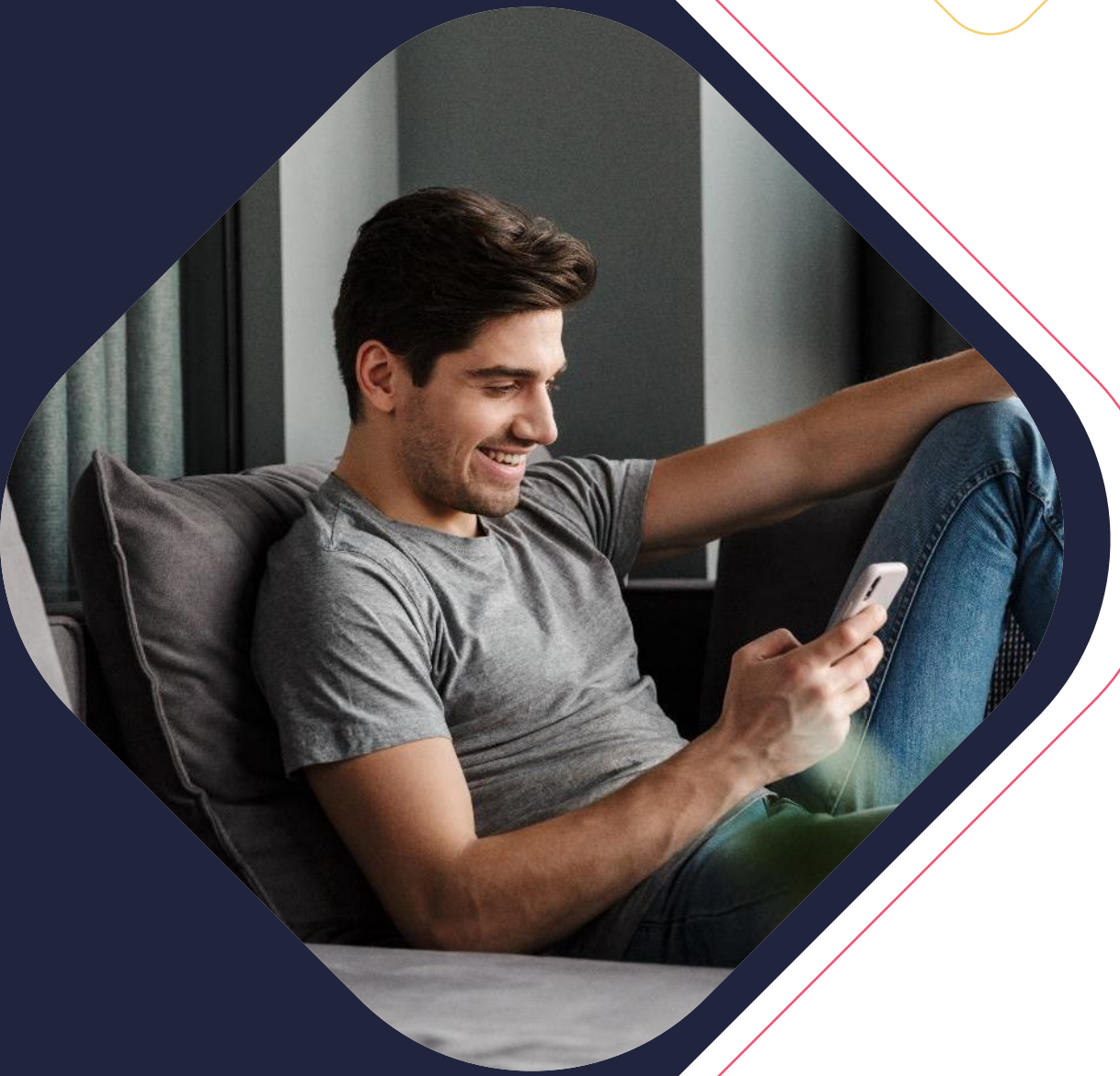
  Example: adjusting the brightness on your screen

Writing device drivers

- Drivers must be kept as efficient as possible
- Need updates

The operating system is the most important software that runs on a computer.

Types of operating systems

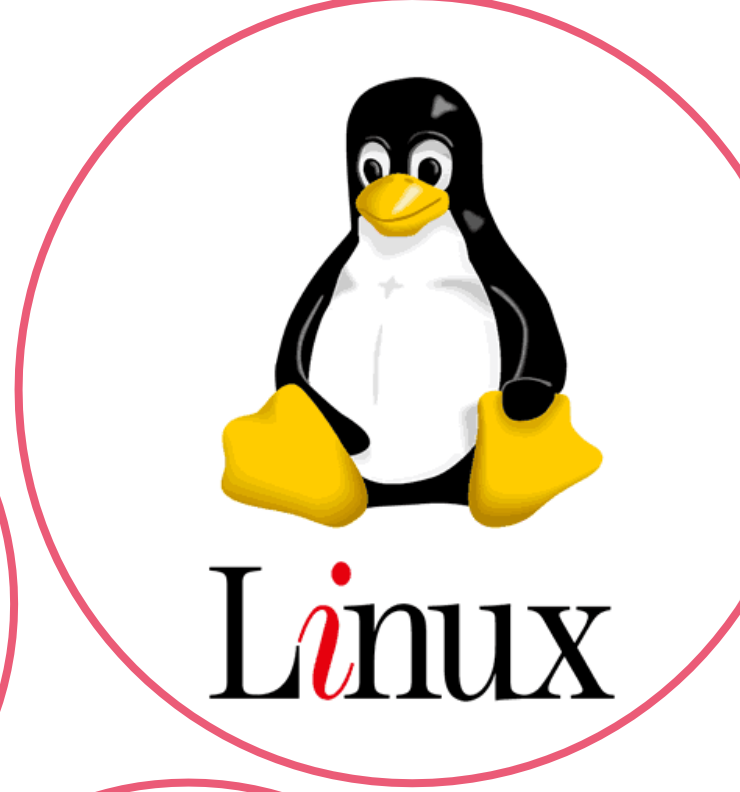


Quick challenge!

Can you identify the operating system on the device that you are using right now?

Linux and Unix-like systems

- Genetic Unix: shares the same code base as the original Unix system
- Branded Unix: deviated from the original code base
- Functional Unix: behaves similarly to Unix-certified systems



Mac operating system

- Most common
- Started as a branch from Unix system
- Still has Unix roots for security, stability and speed
- Written in C language
- Praised for lack of malware until 2006
- Makes up 13% of OS users worldwide

Who is this man and why is it significant to mention him at this point?

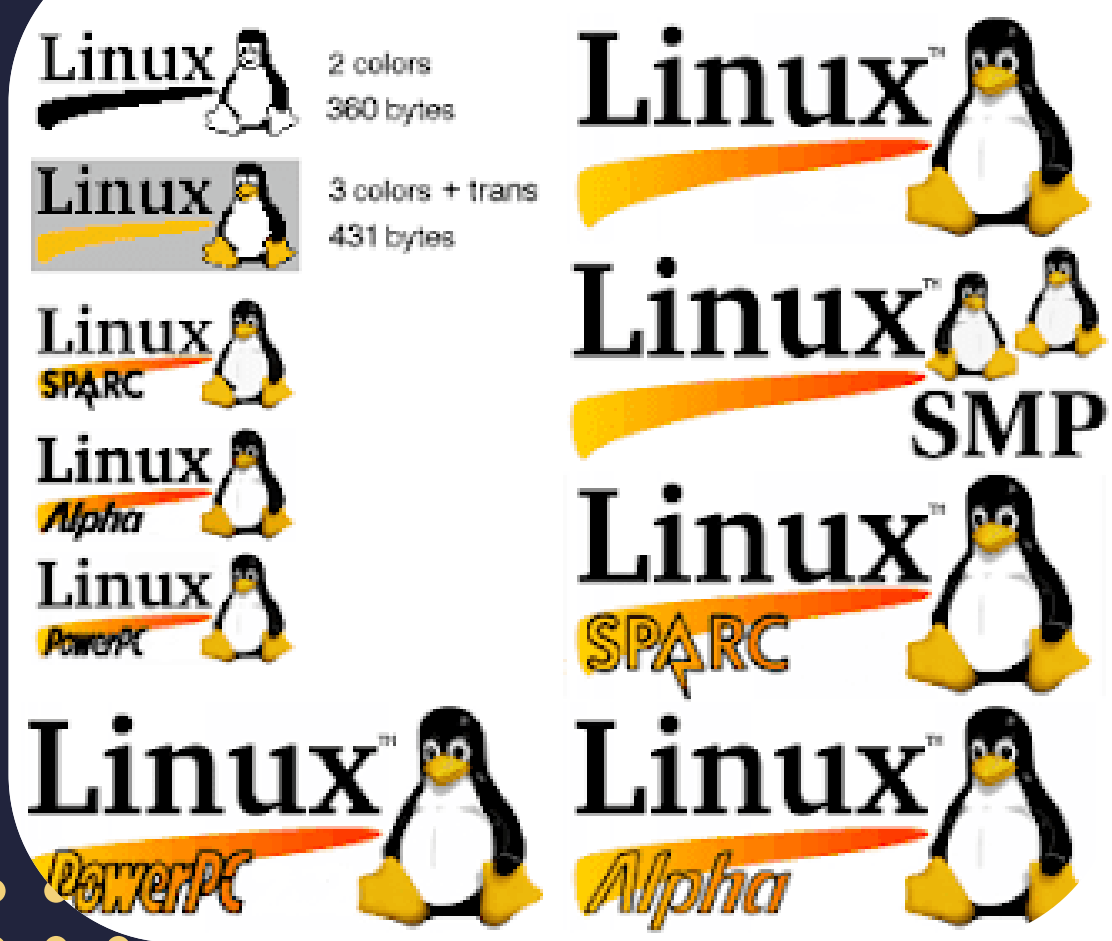
Richard Stallman

- Wanted to create free software
- Proprietary software is illegal to copy or modify
- Free software under the GNU licence enabled copying, adding and modifying
- Gave rise to Linux, an OS with many variants or distributions

Linux distributions

- A Linux variant has been tuned and tweaked to suit a particular use case

Examples: Puppy Linux and Linux mint



Linux commercial variants

- Red Hat Enterprise Linux and SUSE Enterprise
- Desktop versions: X-window
- Desktop environments: Gnome, KDE, Mate, and Xfce

Linux runs on embedded systems

- Devices where the OS is built into the firmware and is made specifically for that system
- The most popular form of Linux is Android
- Used on servers, desktops, IOT devices

DID YOU KNOW?

The Linux kernel has around 27.8 million lines of code!

Android

- Most popular system
- Free Google product distributed under Apache
- Based on a modified version of the Linux kernel
- Released in 2007
- Two billion users and 85% of smartphone market

Android

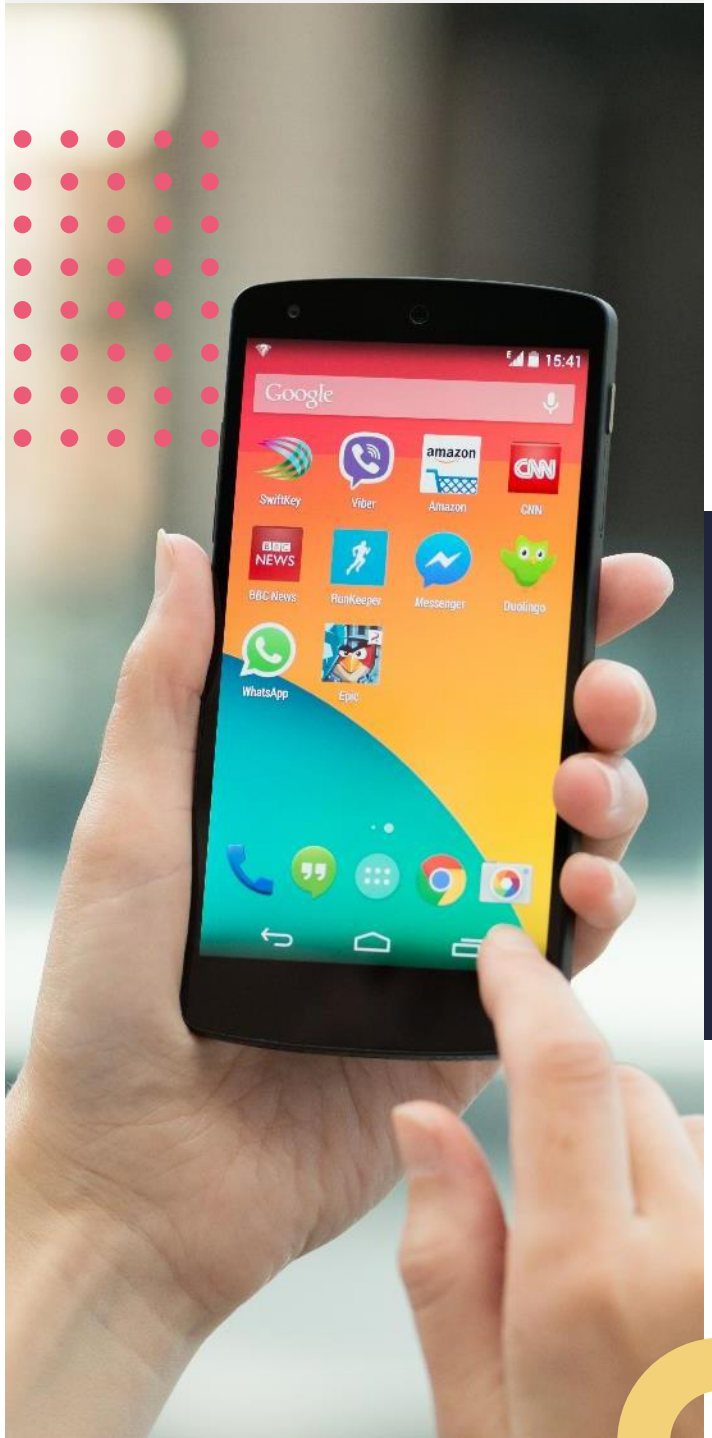
- Android is a Linux distribution
- Includes a kernel, libraries and a layer known as ART – a virtual machine
- Requires more RAM than similar devices

What's a virtual machine?

Microsoft Hyper-V

- A machine running on top of another operating system
- Host OS interacts with hardware and allocates the share of resources to virtual machine
- Provides greater compatibility but affects performance

vmware™



Android evolution

- Primarily touch-based
- Praised for being powerful and customisable
- Customised with ‘skins’

Examples: OneUI by Samsung, OxygenOS by OnePlus, EMUI by Huawei, MIUI by Xiaomi and Sony Xperia UI

Skins

- Not transferable between devices
- Firmware built for a particular device
- Android OEMs usually include Google apps like Google Playstore in the firmware

Windows

- Closed source product of Microsoft
- Used on PCs, servers, and embedded systems
- Typically pre-installed OS and primarily written in C++, with the kernel written in C
- Most popular OPS with 77% of desktop market share

Relation to programming?

- Operating systems are created by programmers
- Linux, Windows and MacOS receive annual/bi-annual updates
- Programmers develop APIs – enables connection to the OS without having to code from scratch

Relation to programming?

- Need to know which OS is best suited to solve the problem
- Some OS have limitations – knowledge of these systems helps to map the development path
- Enables the writing of efficient code