

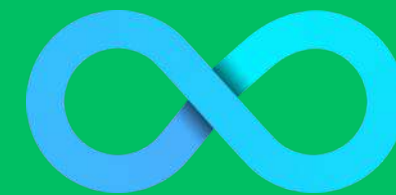
Operating System



The plan

Introduction to Operating Systems

Core Concepts of Operating Systems

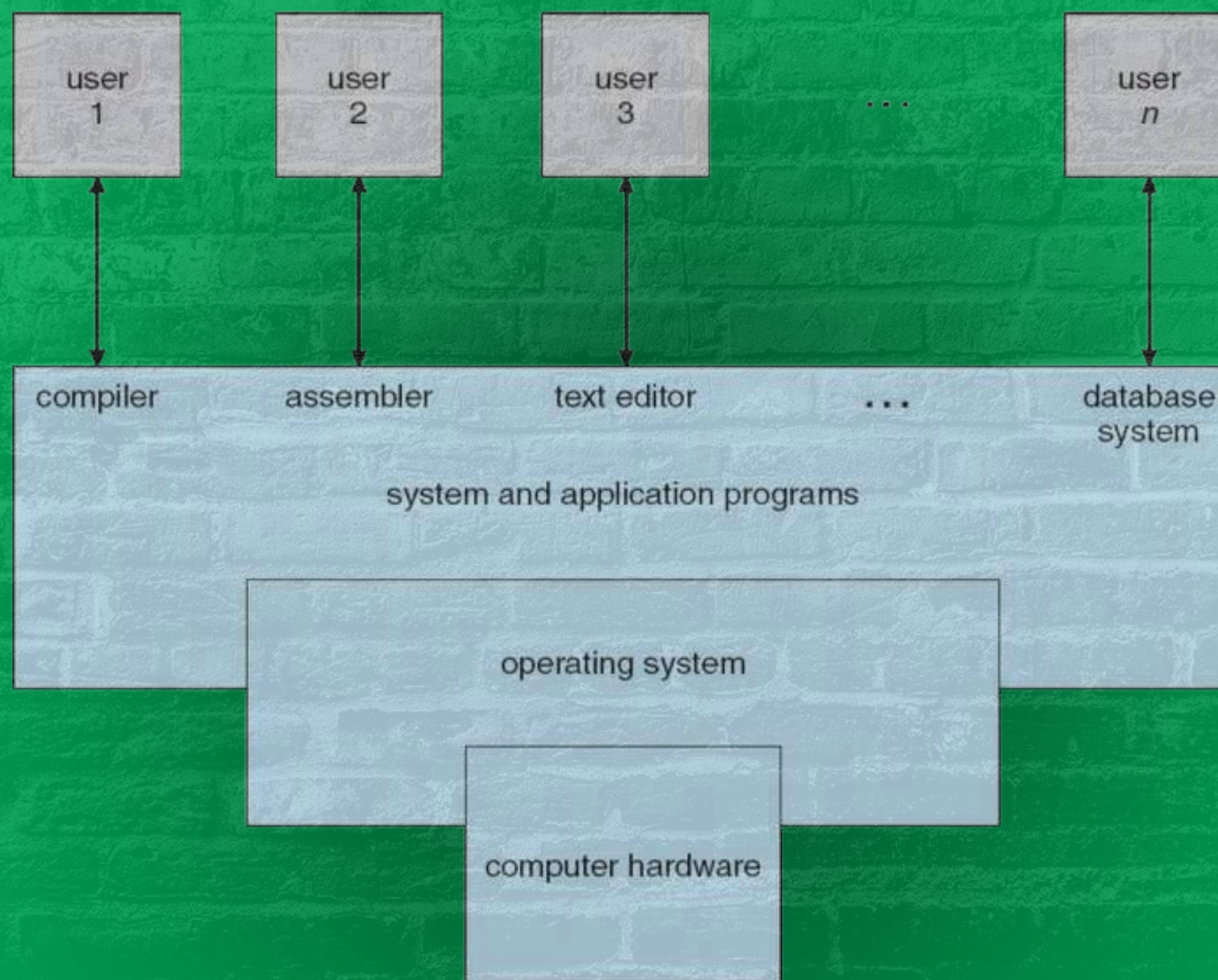


Introduction

**What's an Operating
System?**

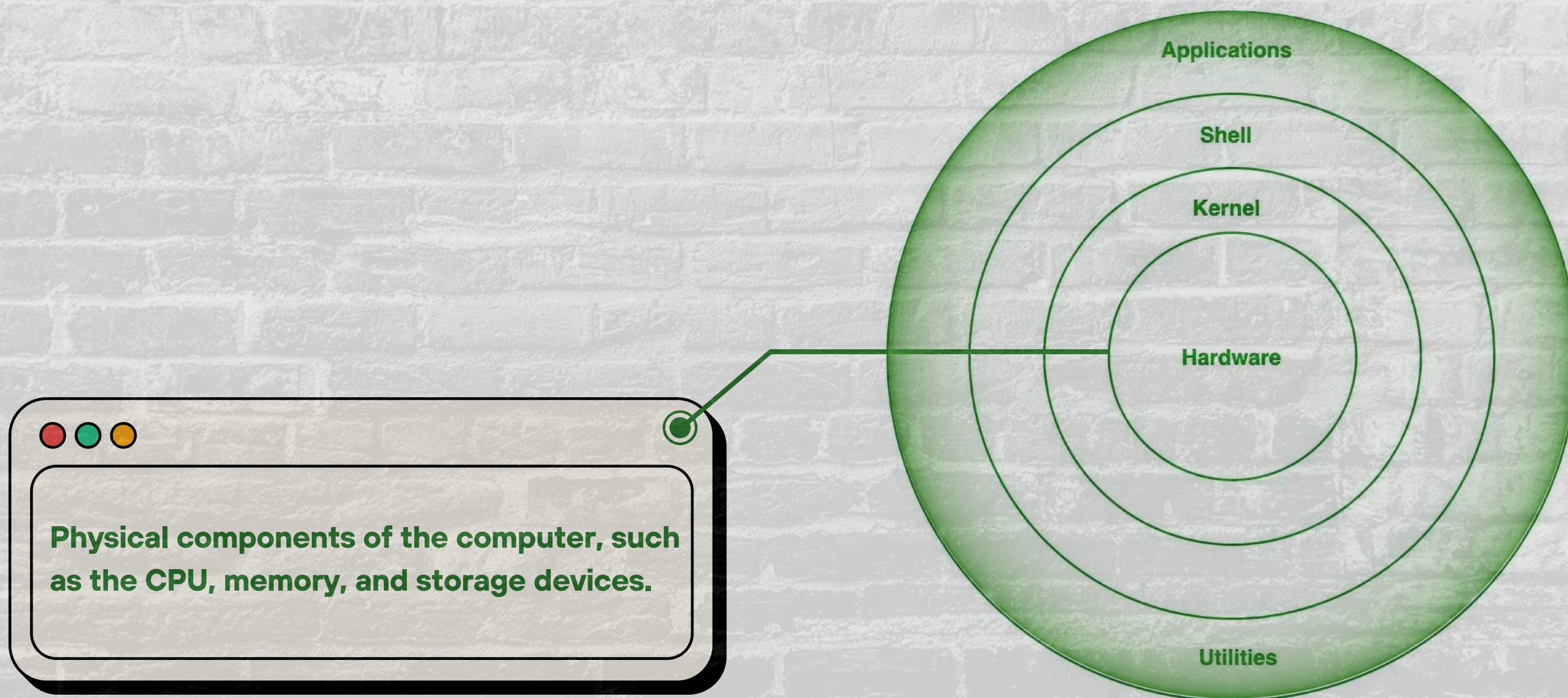


"An operating system is the software layer that hides the ugliness of hardware and makes it beautiful for the user." — Anonymous



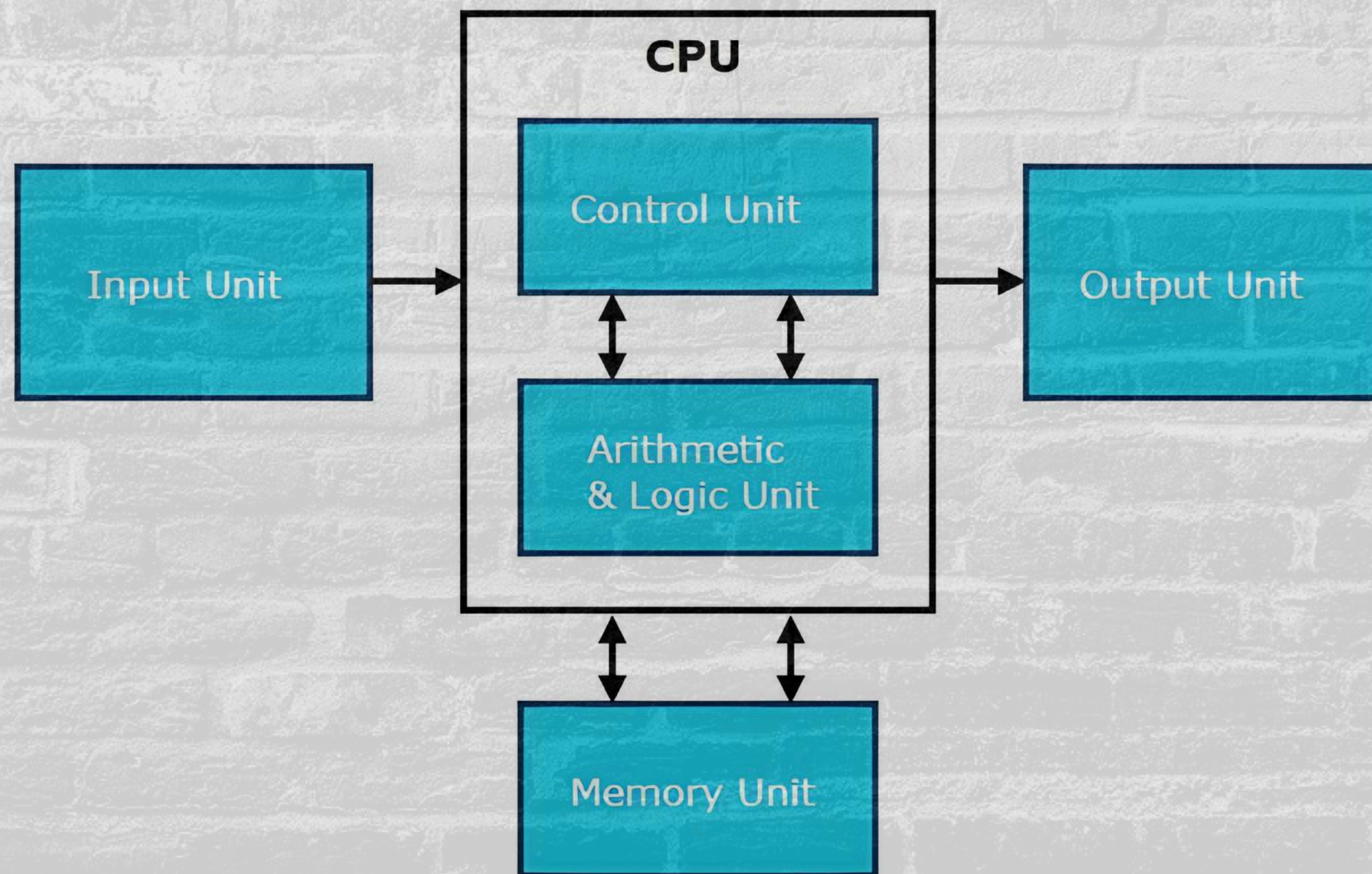


Layers of a computer system





What is a CPU?





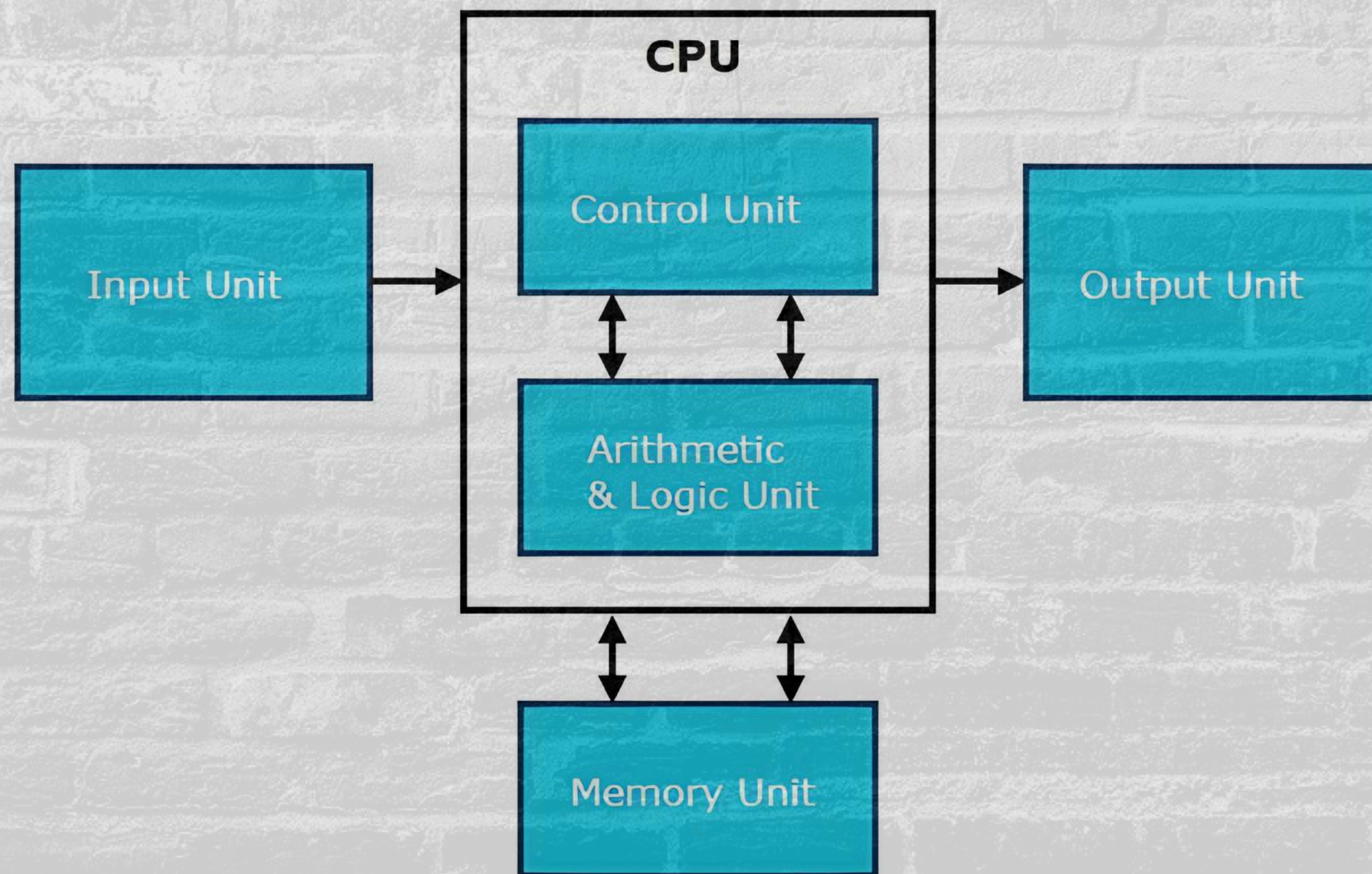
How does it work?

You can imagine the CPU's work as a loop with 3 steps:

- **Fetch:** It grabs an instruction (like, "Add two numbers").
- **Decode:** It figures out what the instruction means.
- **Execute:** It does the task, like adding the numbers.

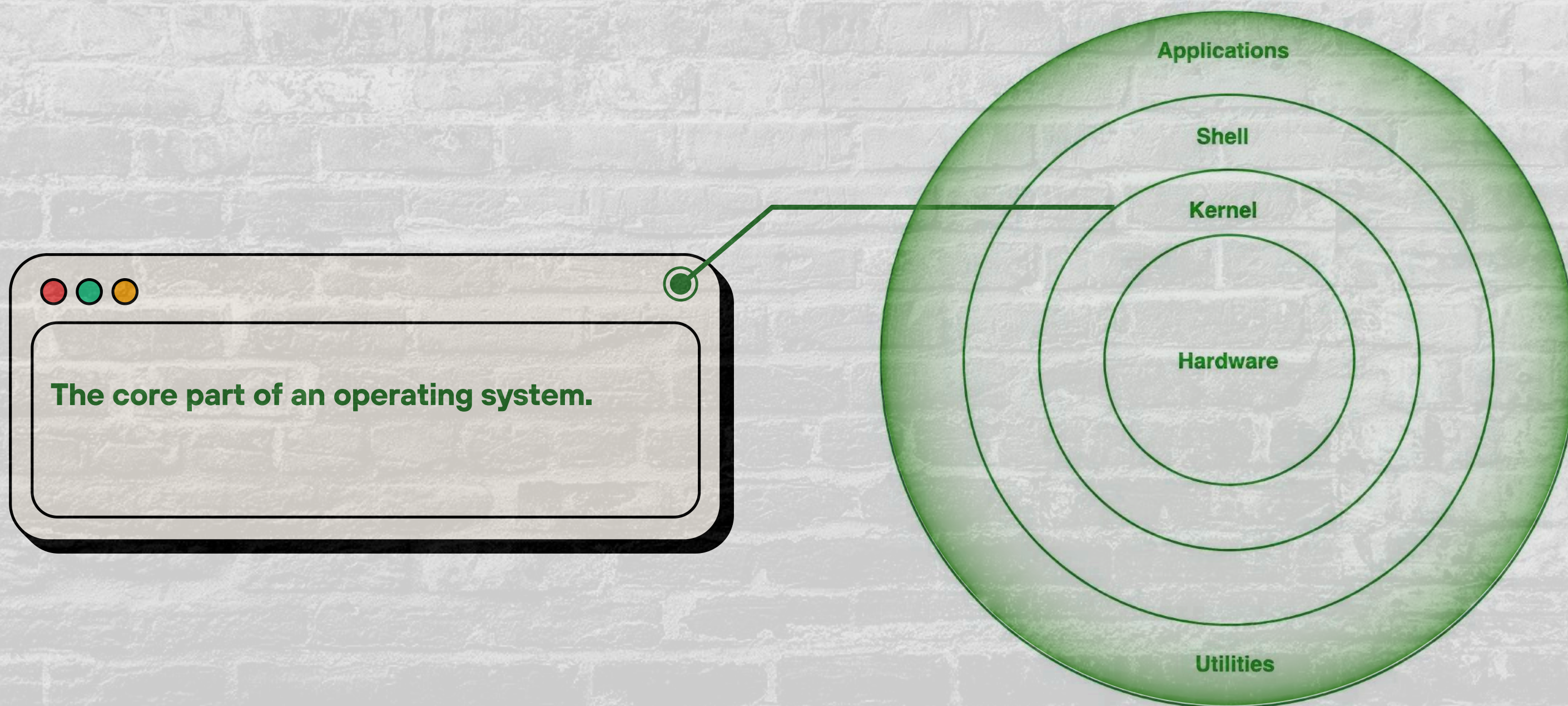


What is a CPU?





Layers of a computer system

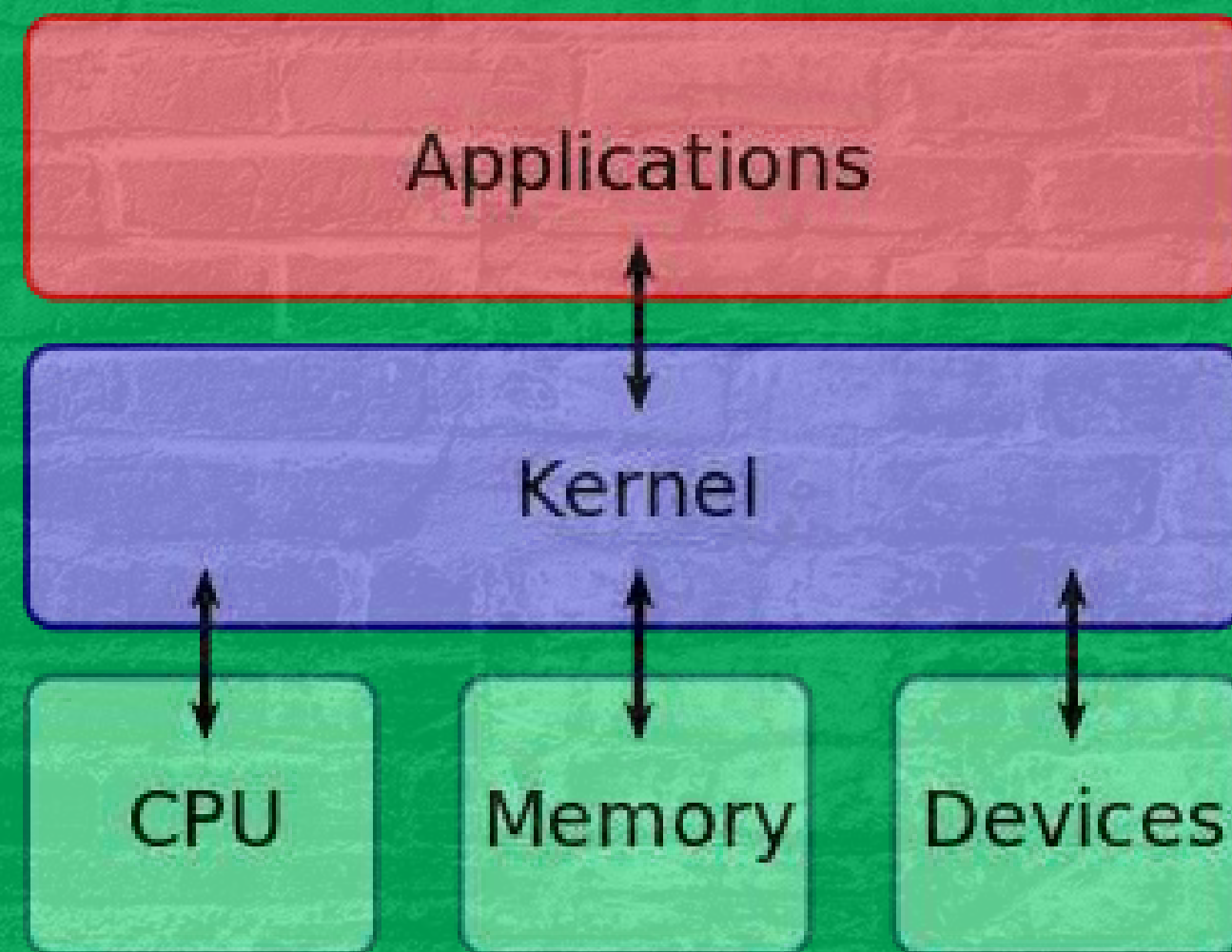




Kernel

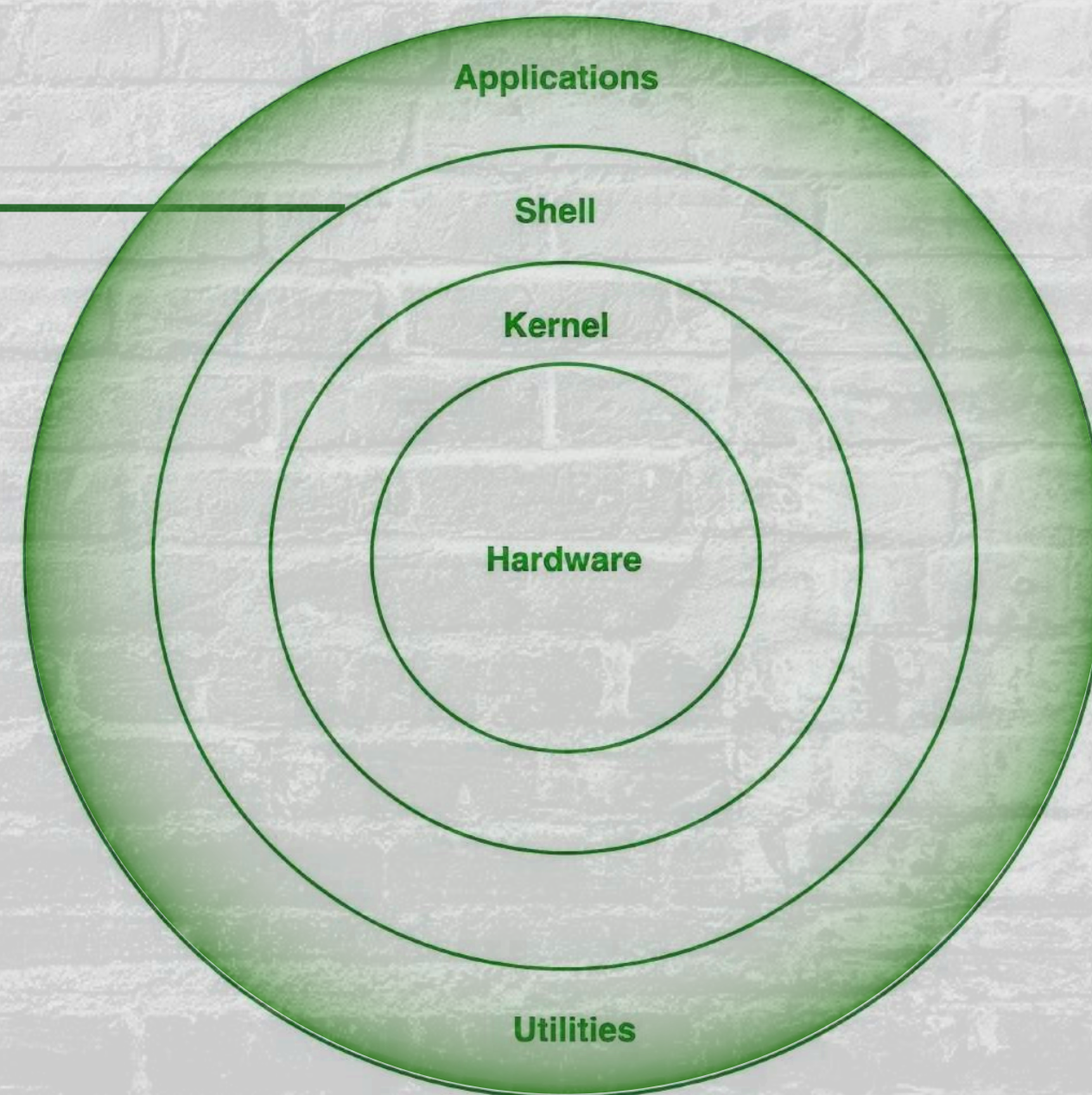
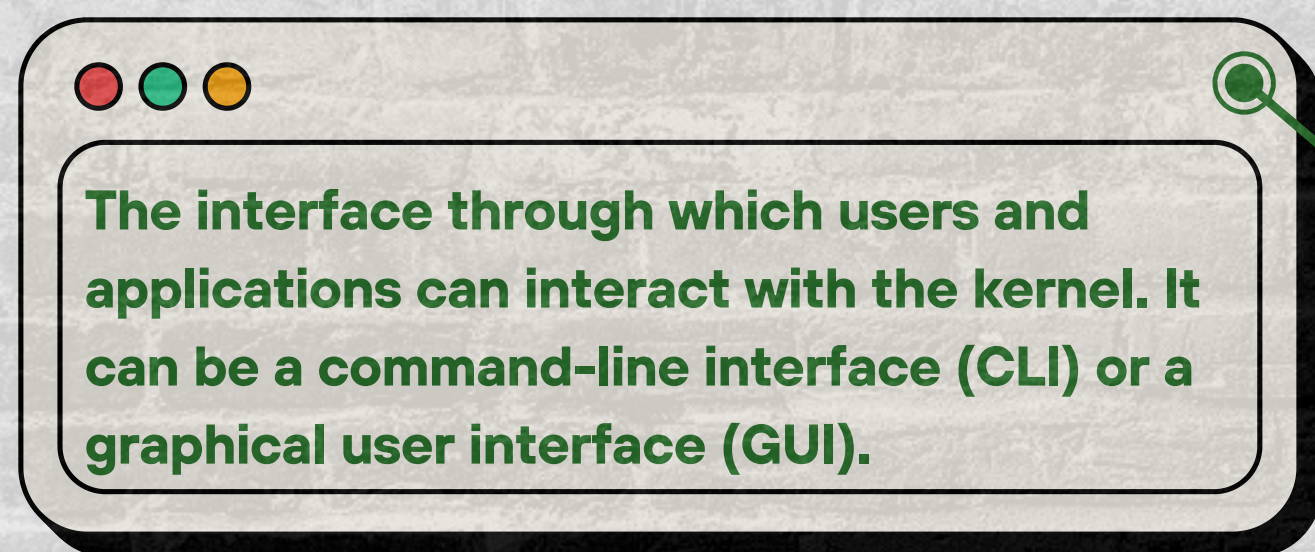
A kernel is the core part of an operating system. It acts as a bridge between software applications and the hardware of a computer.

The kernel controls the CPU, memory, and devices, making sure everything runs smoothly. It manages tasks like running programs, accessing files, and connecting to devices.



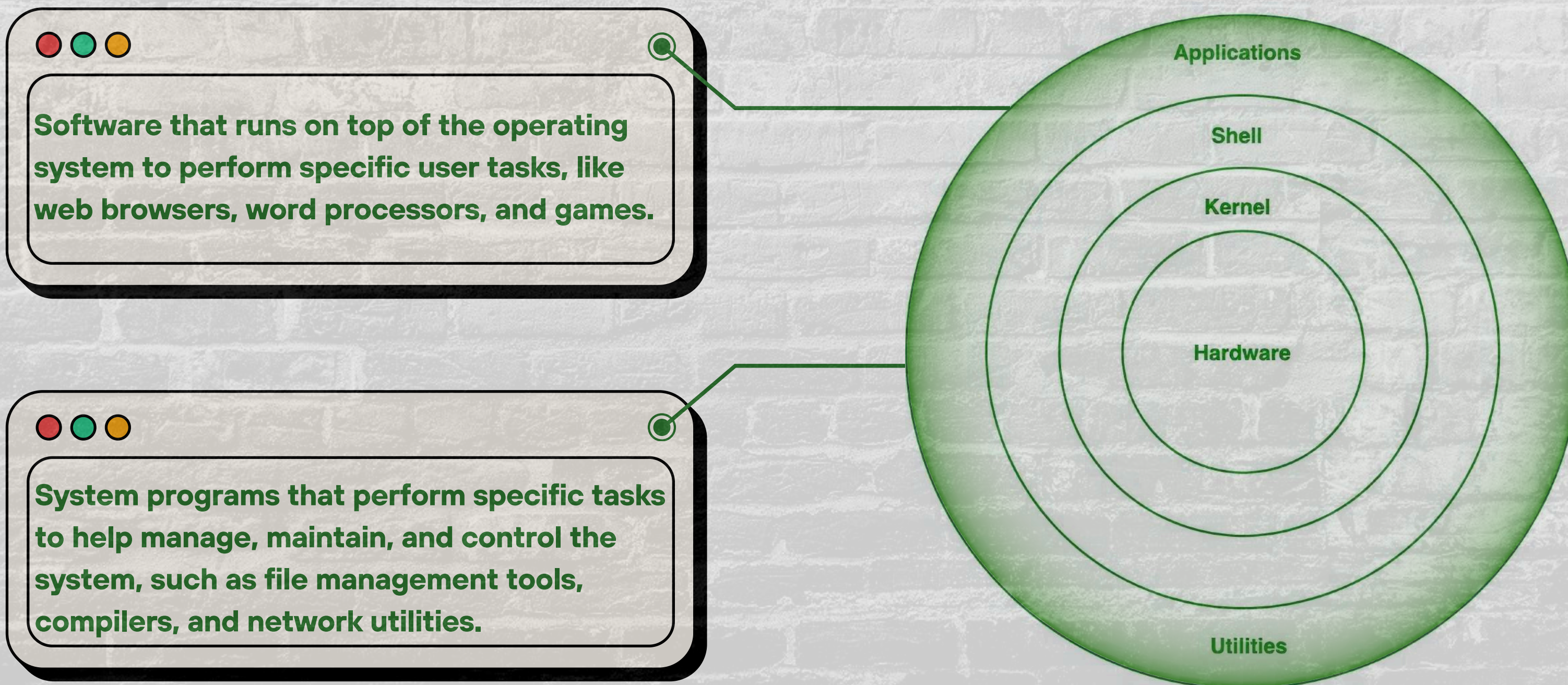


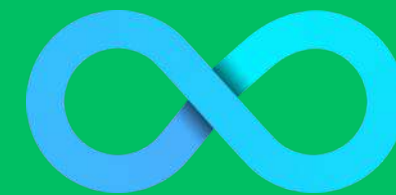
Layers of a computer system





Layers of a computer system



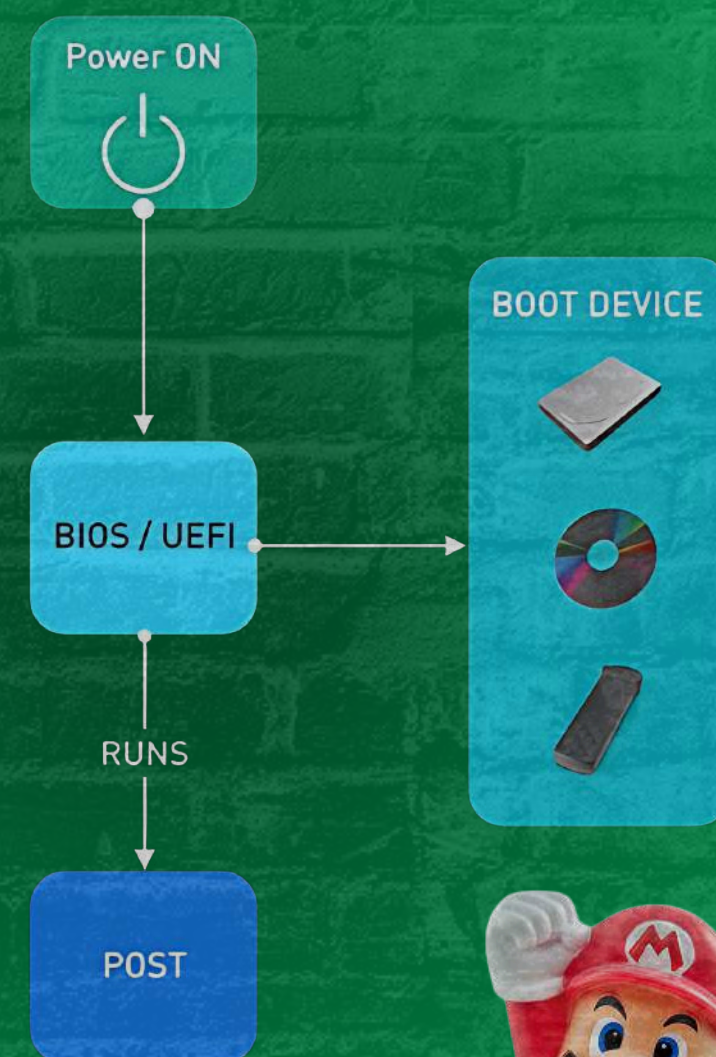
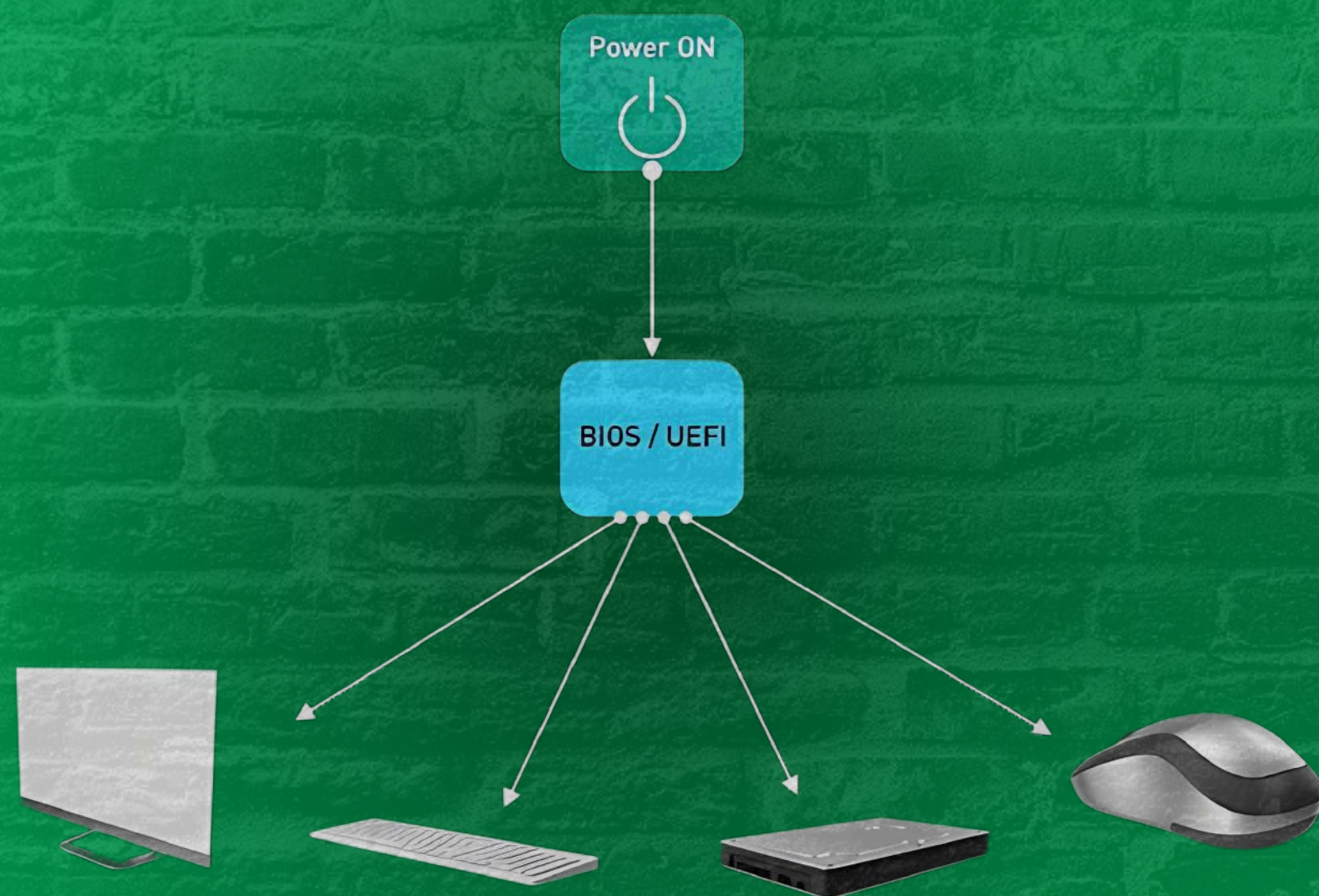


Introduction

Computer Start-up Process



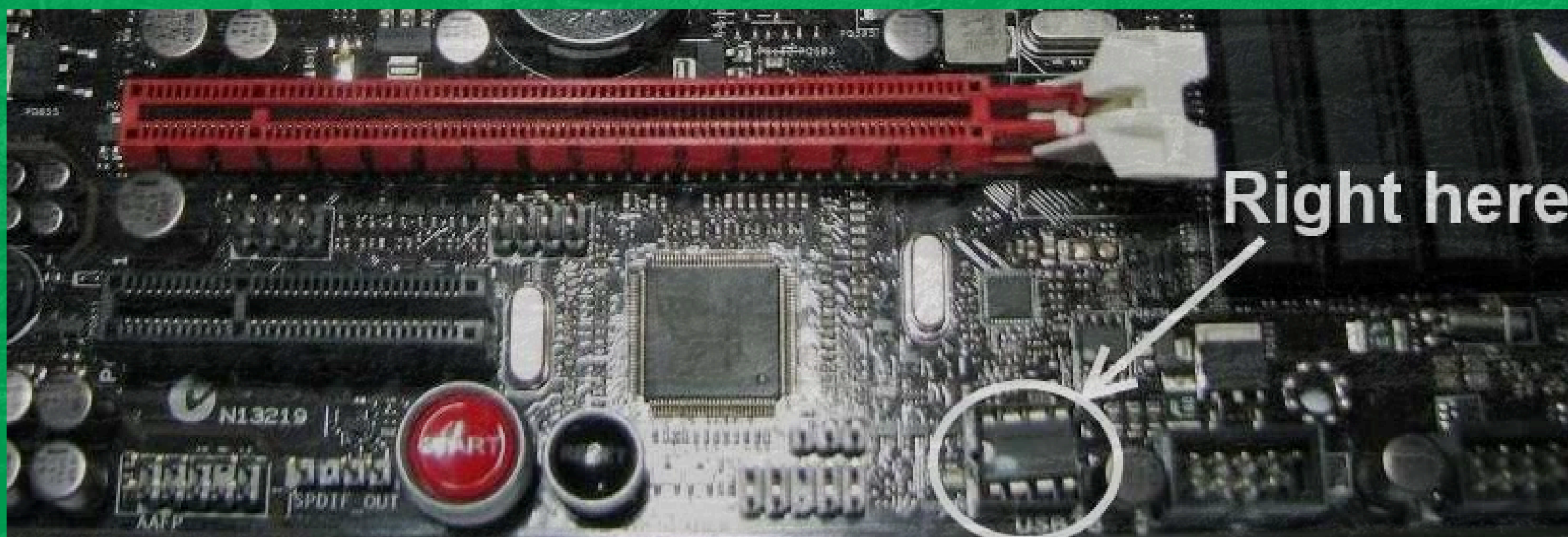
Computer Start-up Process





BIOS (Basic Input/Output System)

A **Software** stored on a small memory chip, also known as **firmware**. BIOS is found on the **motherboard**, and it is the very **first software** to run after a computer starts.

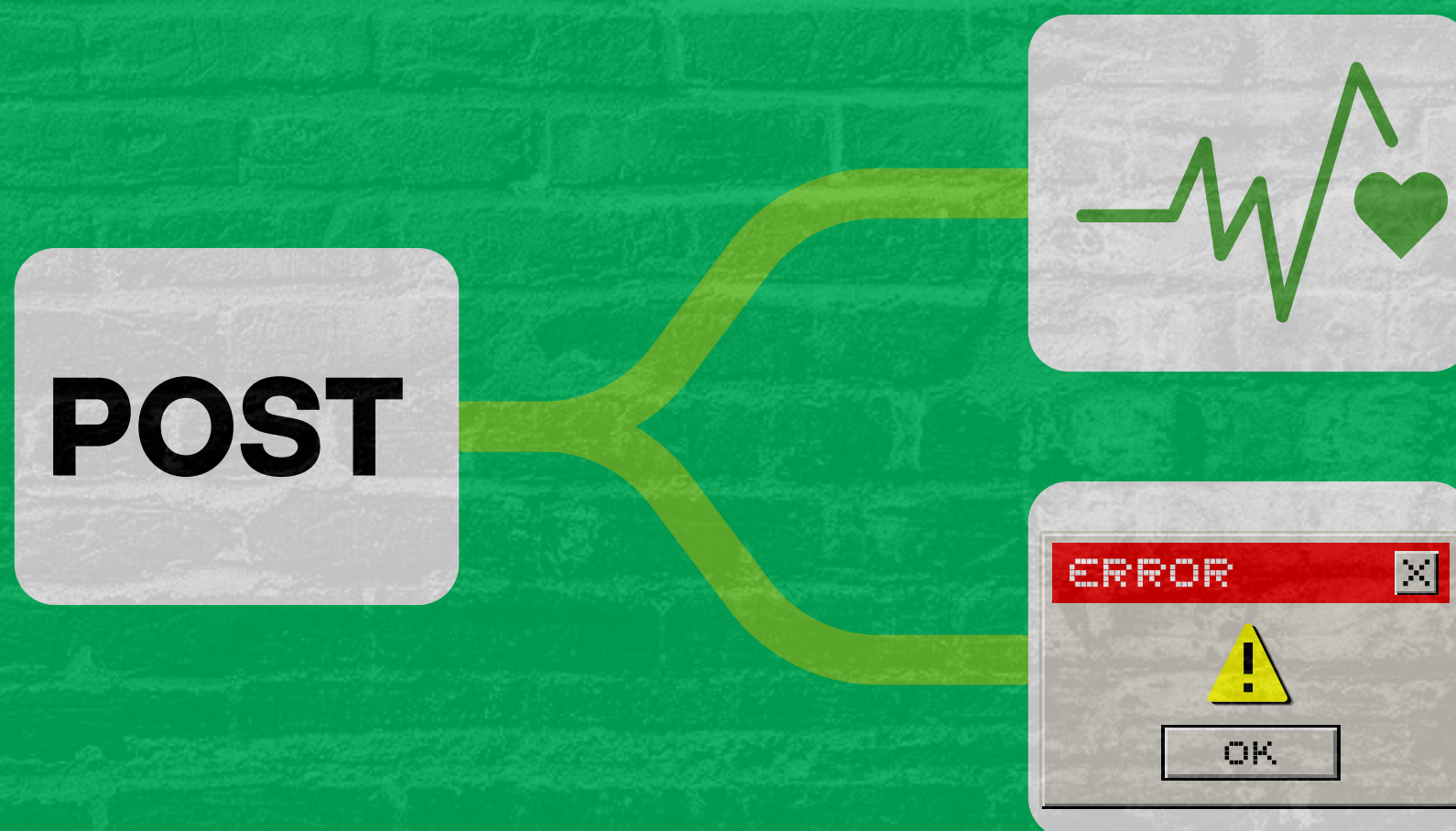




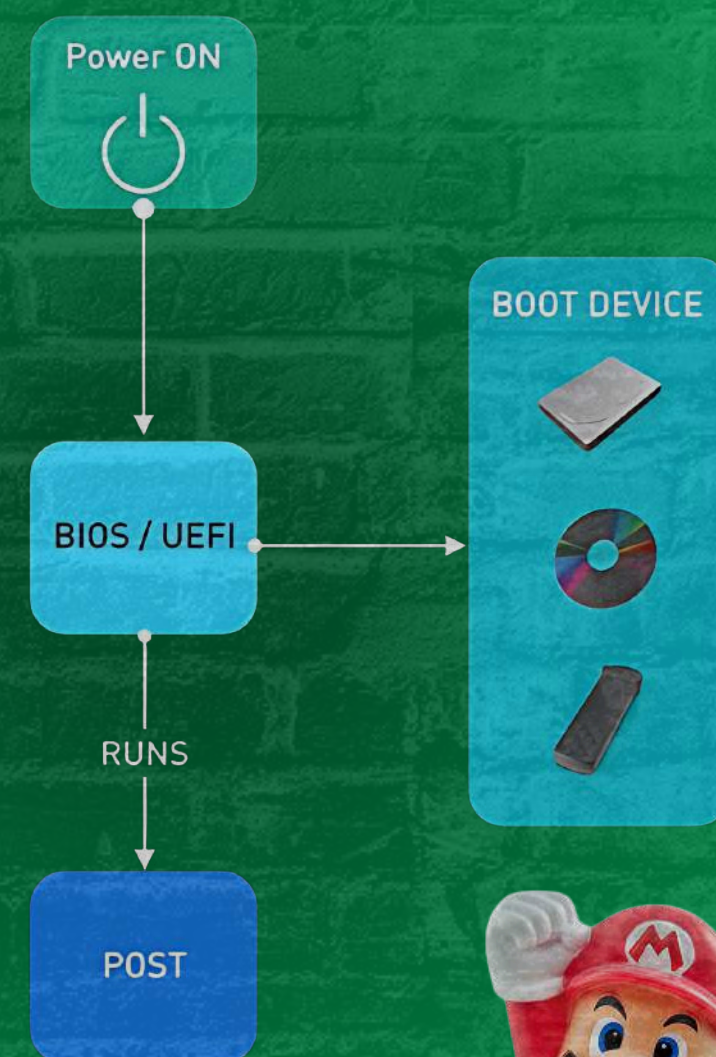
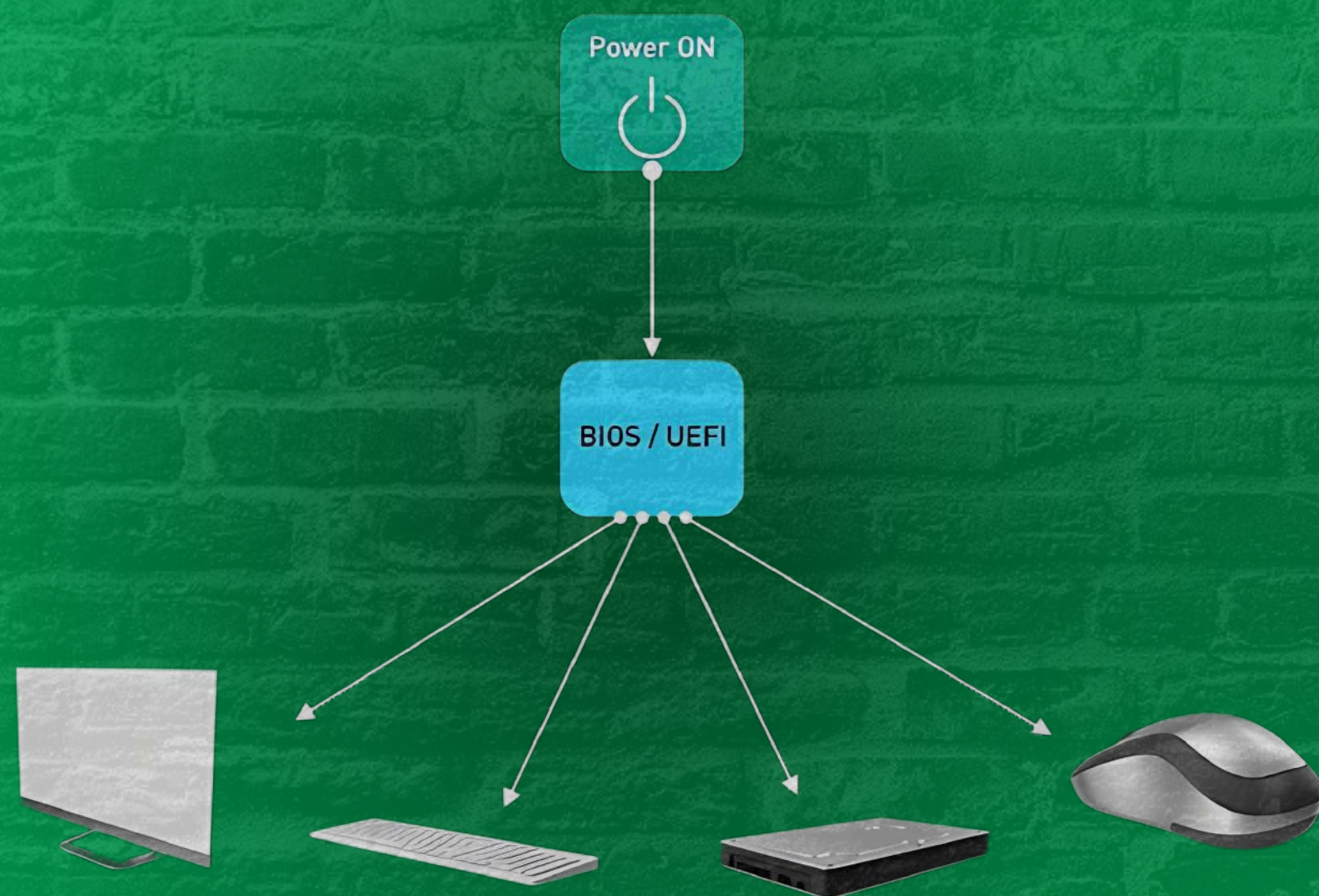
POST (Power-On Self-Test)

An operation initiated by a computer after it has been turned on but before it boots up the OS

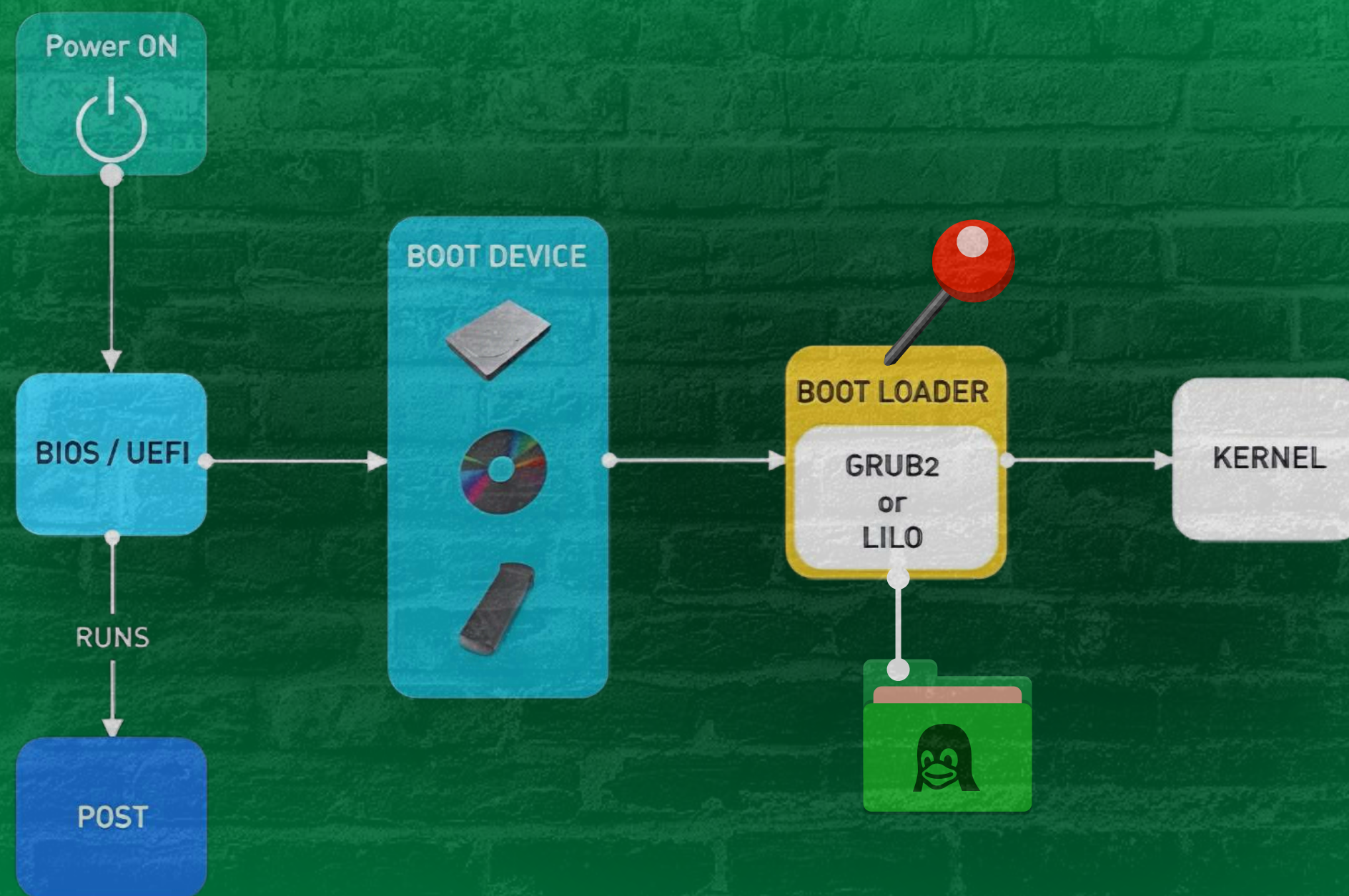
The BIOS runs a diagnostic testing sequence to determine if the computer's essential hardware is working properly.



Computer Start-up Process



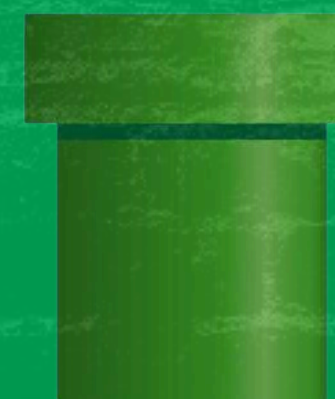
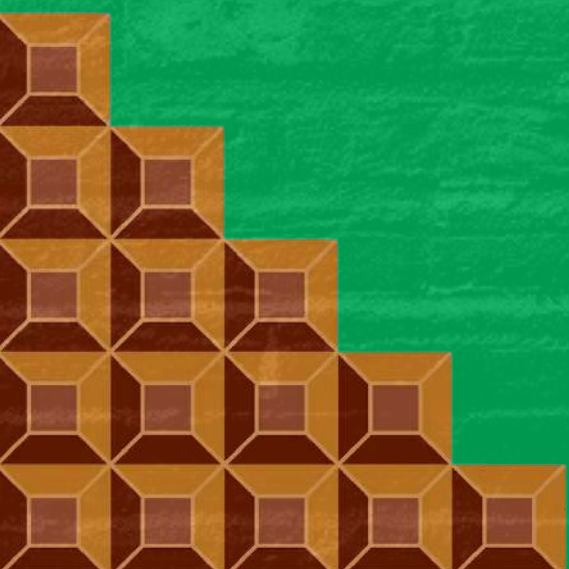
Computer Start-up Process





Boot Loader

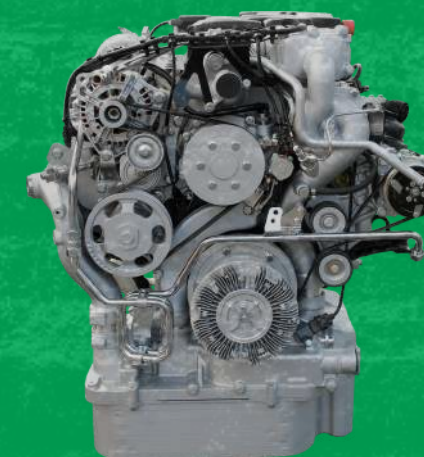
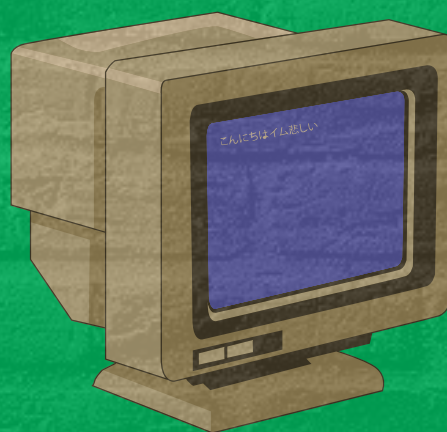
It's a small program stored in the **boot device** that **actually locates**, **then loads the OS kernel into memory** and **finally starts running the kernel code**.





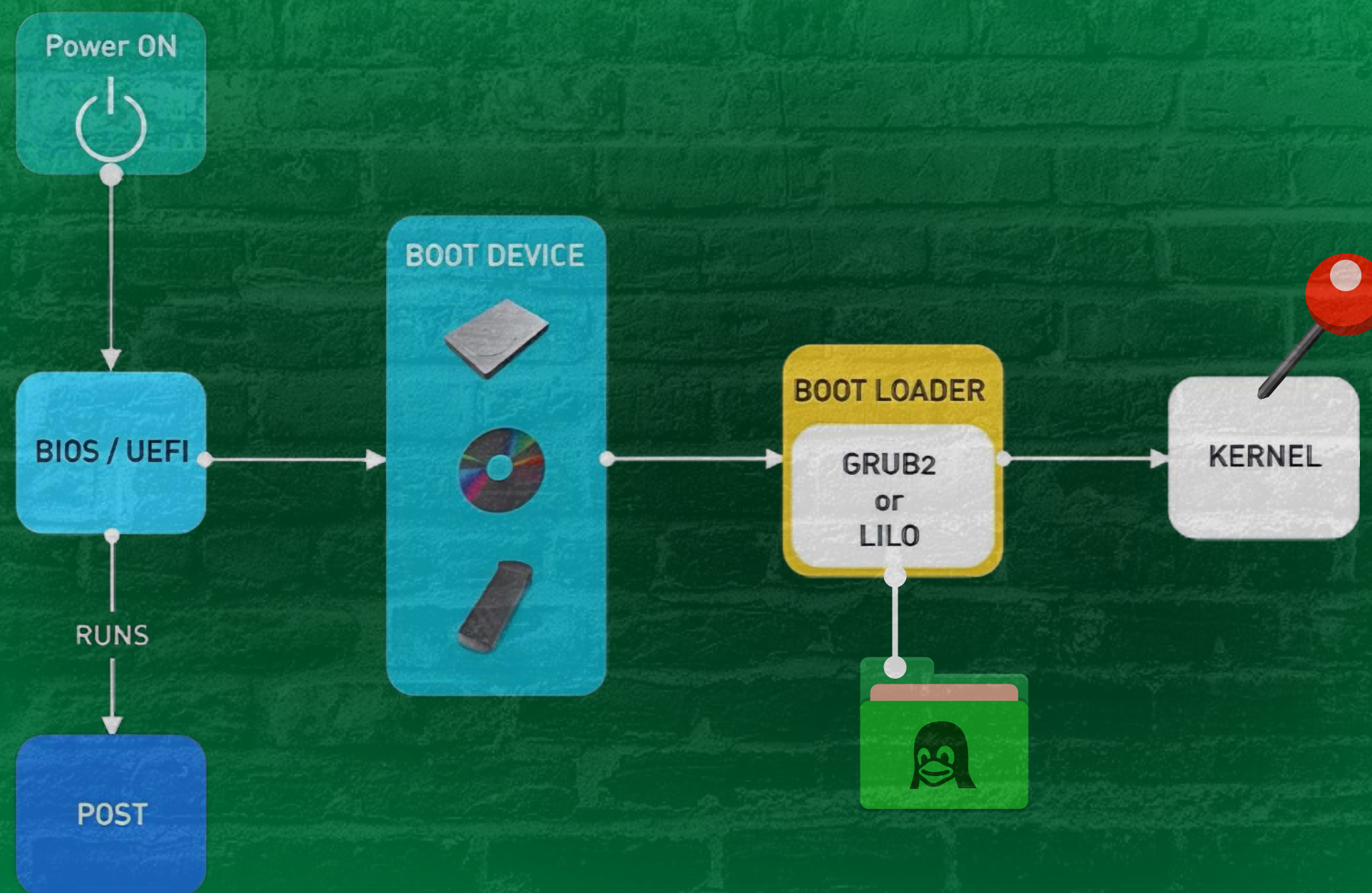
When you turn on your car (powering on your computer), you need the key (the bootloader) to start the engine (the operating system).

Once the engine is running, the car can drive (the computer can run programs and do tasks).





Computer Start-up Process

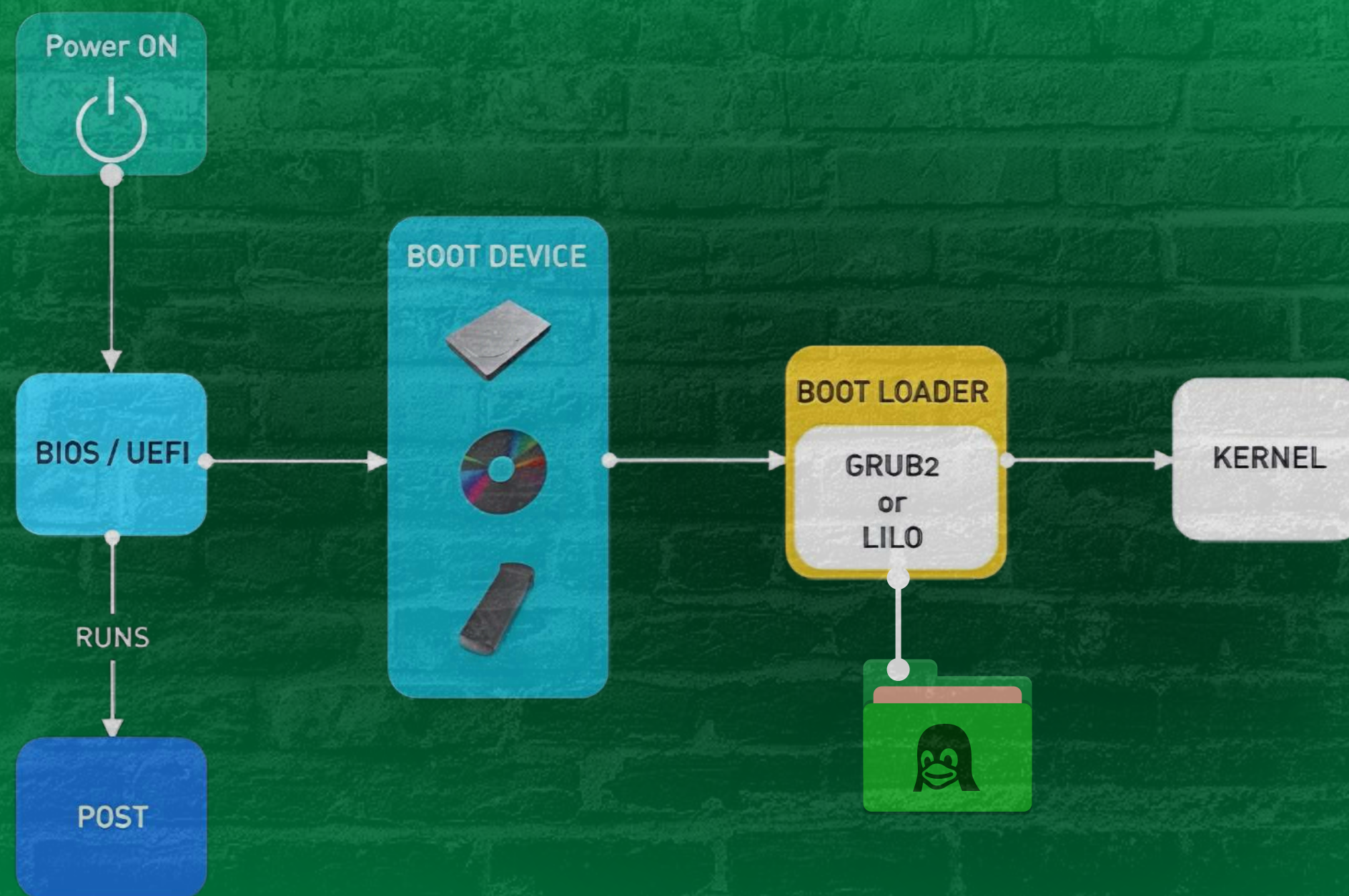


Kernel Initialization
Loading Kernel Models
Starting Background Services

•
•
•
•
•

Initial Process Creation
Running init(Systemd)
User-space Execution

Computer Start-up Process





Core Concepts

**Processes and
Threads**

**Memory
Management**



Program Vs. Process

- A program is defined as a **passive** entity, which means it is just a file stored on the disk requiring resources like **CPU and memory** to execute, typically as an executable file.
- A process is the **active** entity that results when this executable file is loaded into memory and executed by the operating system.
- The transition occurs when the program (executable file) is **loaded into memory**, becoming a process.

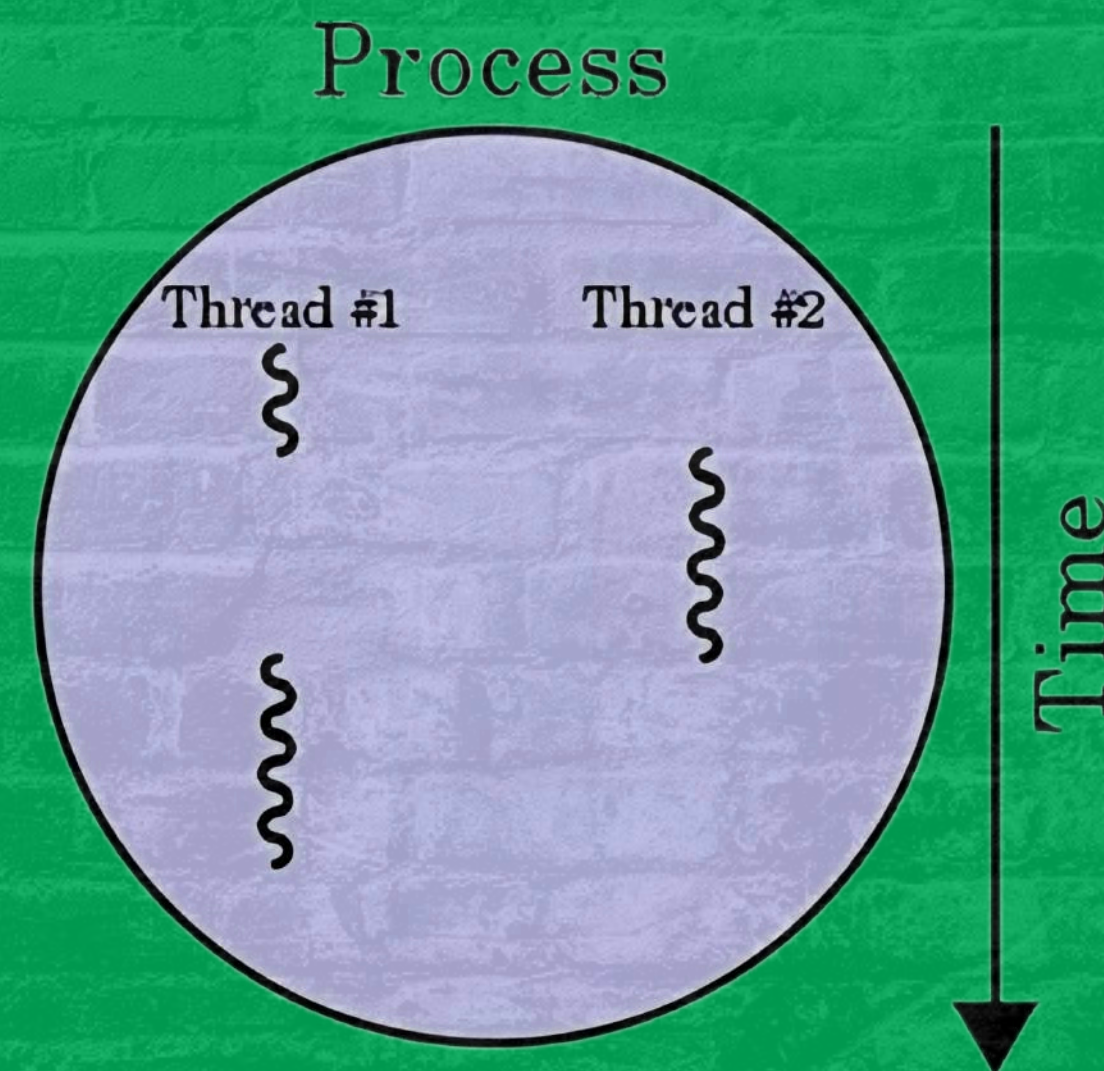
Threads

- **A thread is a single sequence of instructions within a process that can be scheduled and executed independently by the operating system.**
- **Multiple threads can exist within a single process, sharing the process's resources such as memory, but each thread executes independently.**



Threads and Processes

- **A process is an instance of a program in execution, containing its own memory space and resources.**
- **A thread is a lightweight sub-process, sharing the same memory and resources of the process but having its own execution context (e.g., program counter, registers, stack).**





Example: Watching a Video



The process is the “media player”

- A process is essentially the container that holds everything needed to run the program, like memory, resources, and the code.
- Within this video player, there are multiple threads, each handling a specific task.



There are 3 threads here...

- **One thread decodes the video:** reading the video file and turning it into frames you can watch.
- **Another one handles audio:** so that the sound is synced with what you see on the screen.
- **Another thread checks for user inputs:** This thread waits for you to press play, pause, or adjust the volume.



Process Management

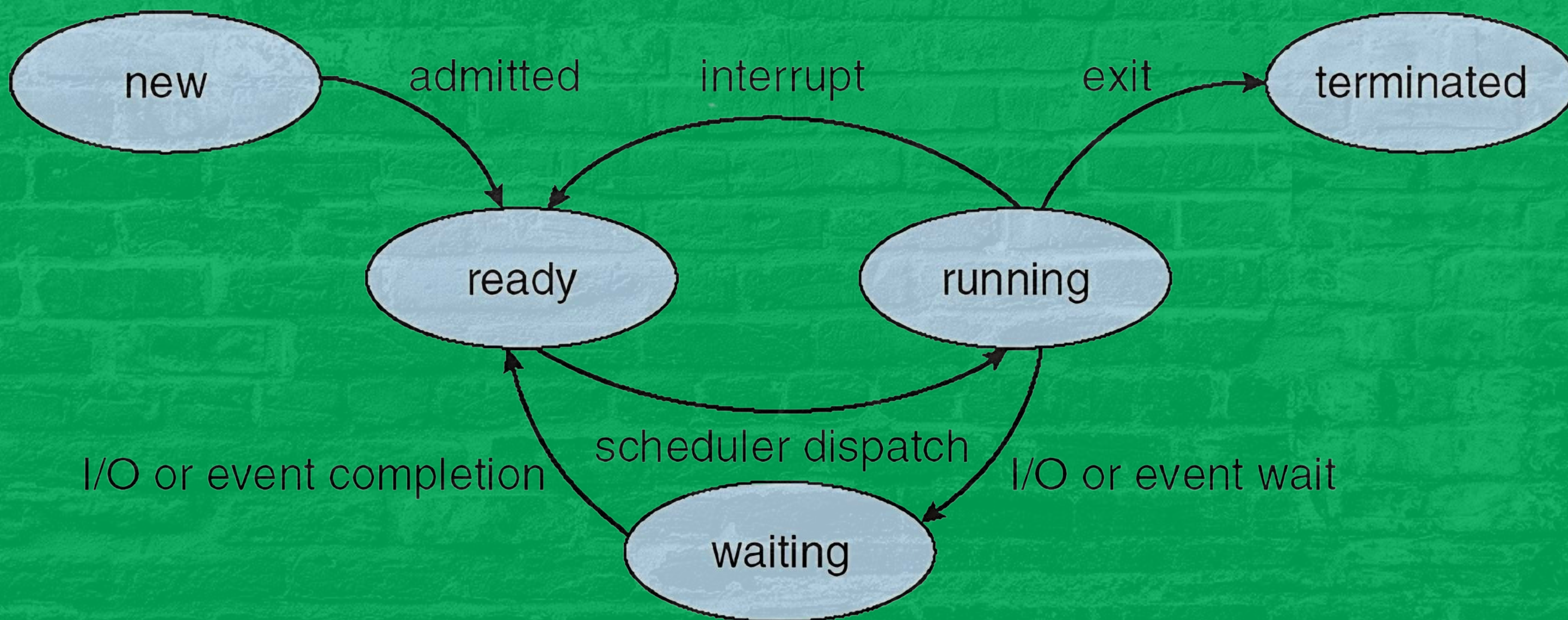


Process Control Block

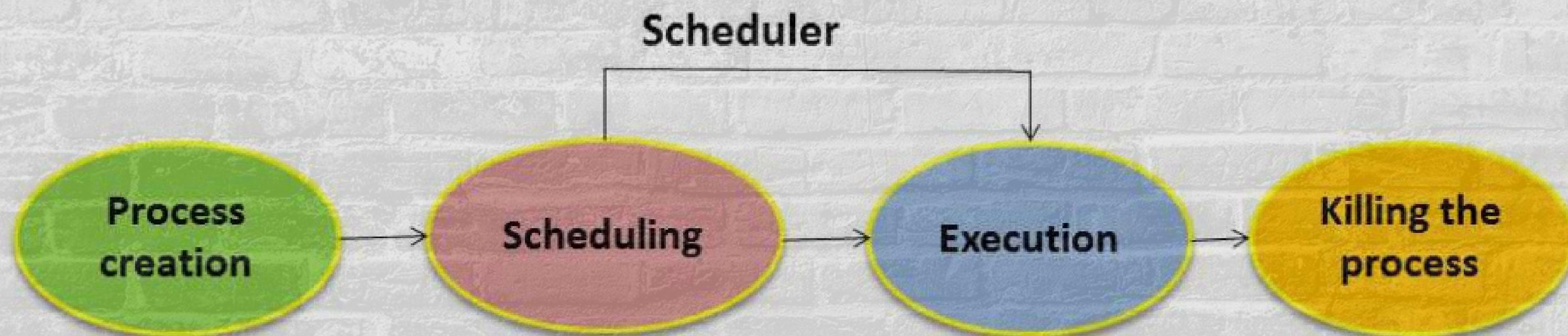
Attributes of process	Description
Process State	The process could be in any state, so the current state of the process (ready, running, waiting, terminated)
Process privileges	This is required for allowing/disallowing access to the system resource.
Process ID	This specifies Unique identification for all processes in the operating system.
Program Counter	Contains the address of the next instruction, which will be executed in the process.
CPU registers	This specifies the registers that are used by the process. They may include general-purpose registers, index registers accumulators, stack pointers, etc.
Memory management information	This includes the memory information like information related to the Segment table, memory limits, and page table.
Accounting information	This includes the information about the amount of CPU used for process execution, execution time, time limits, etc.
IO status information	There are many Input/Output devices attached to the process, and the list of I/O devices allocated to the process is maintained.



Process states



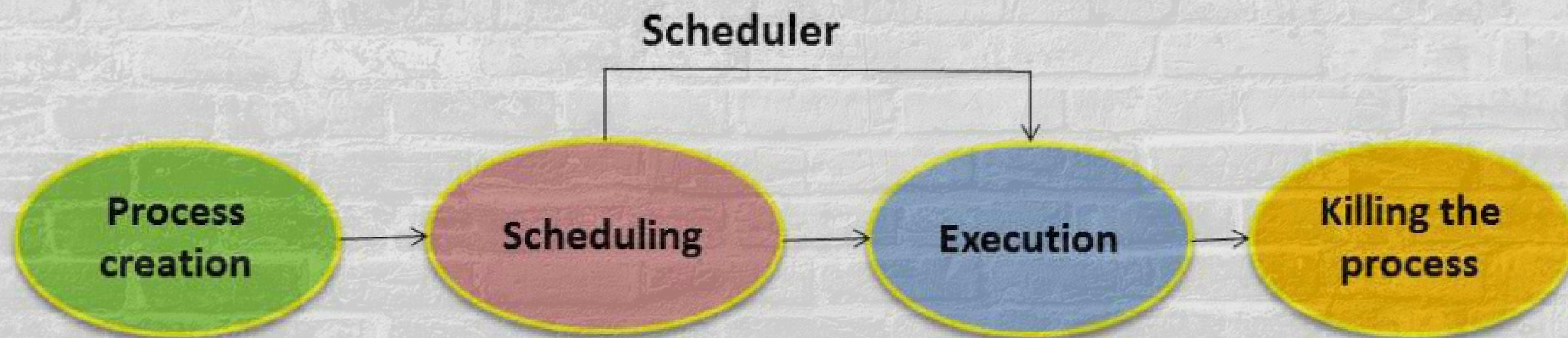
Process Operations



The first step is process creation. Process creation could be from a user request(using fork()), a system call by a running process, or system initialization.



Process Operations



Scheduling = Time sharing



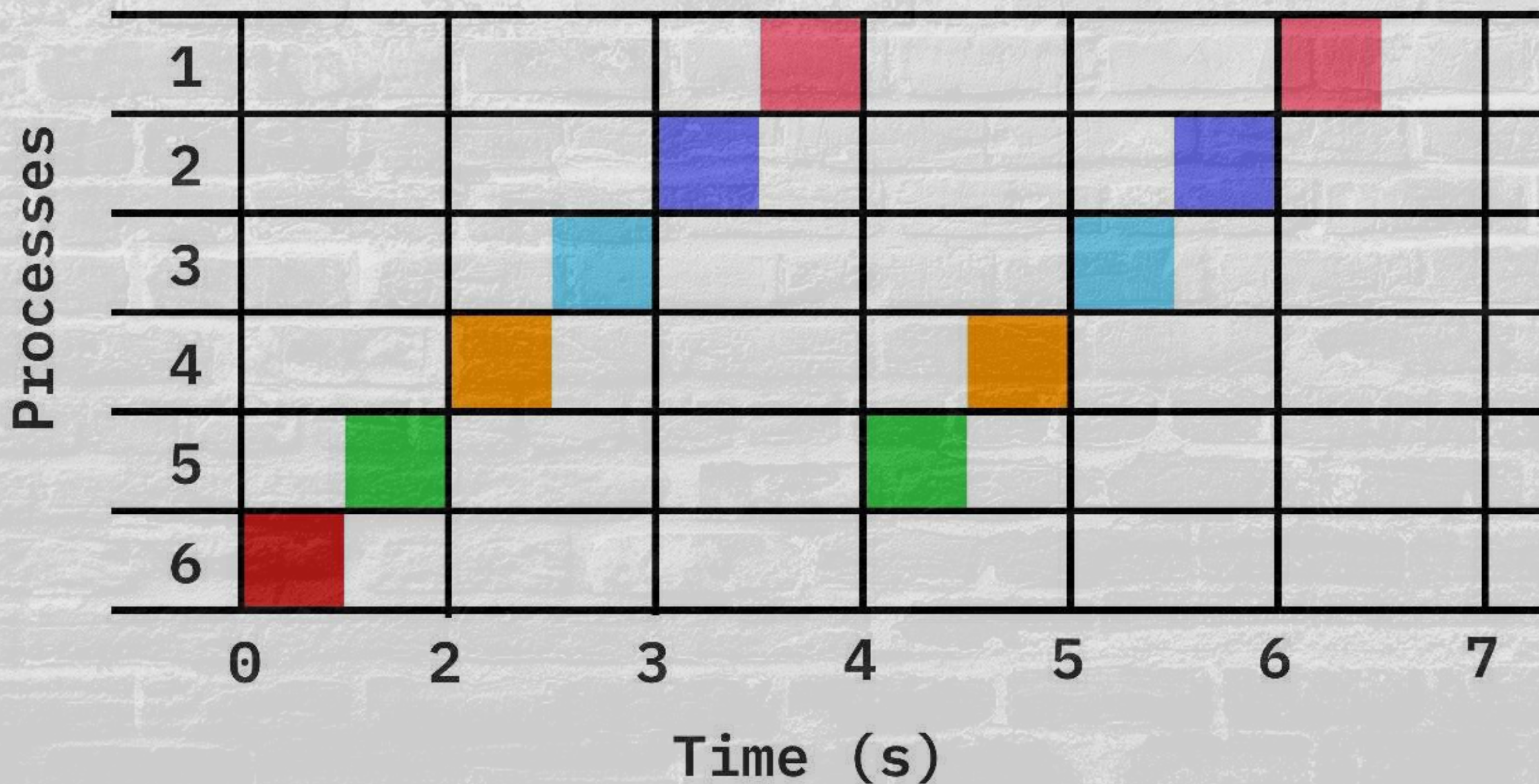
Process Operations



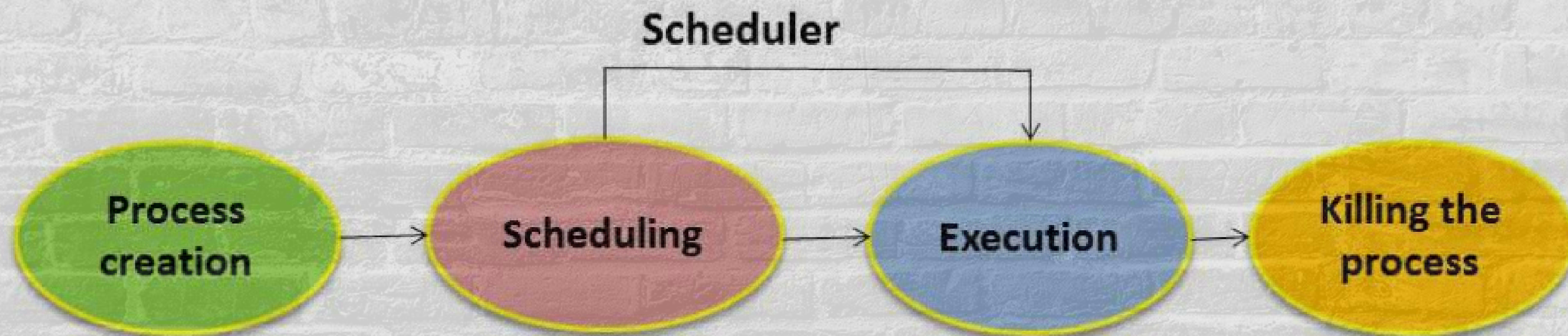


Process Operations

Round Robin Scheduling



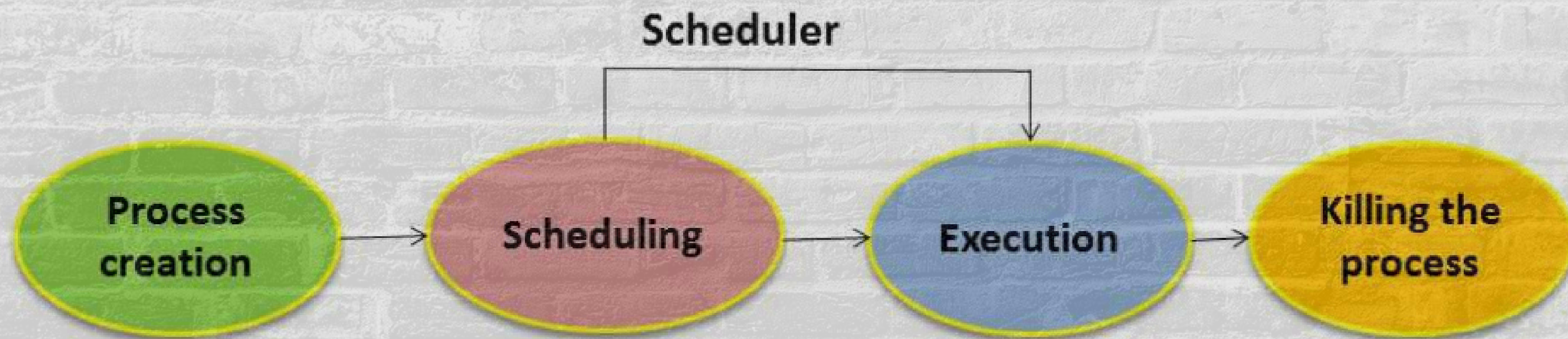
Process Operations



The execution means the CPU is assigned to the process. Once the process has started executing, it can go into a waiting queue or blocked state. Maybe the process wants to make an I/O request, or some high-priority process comes in.



Process Operations



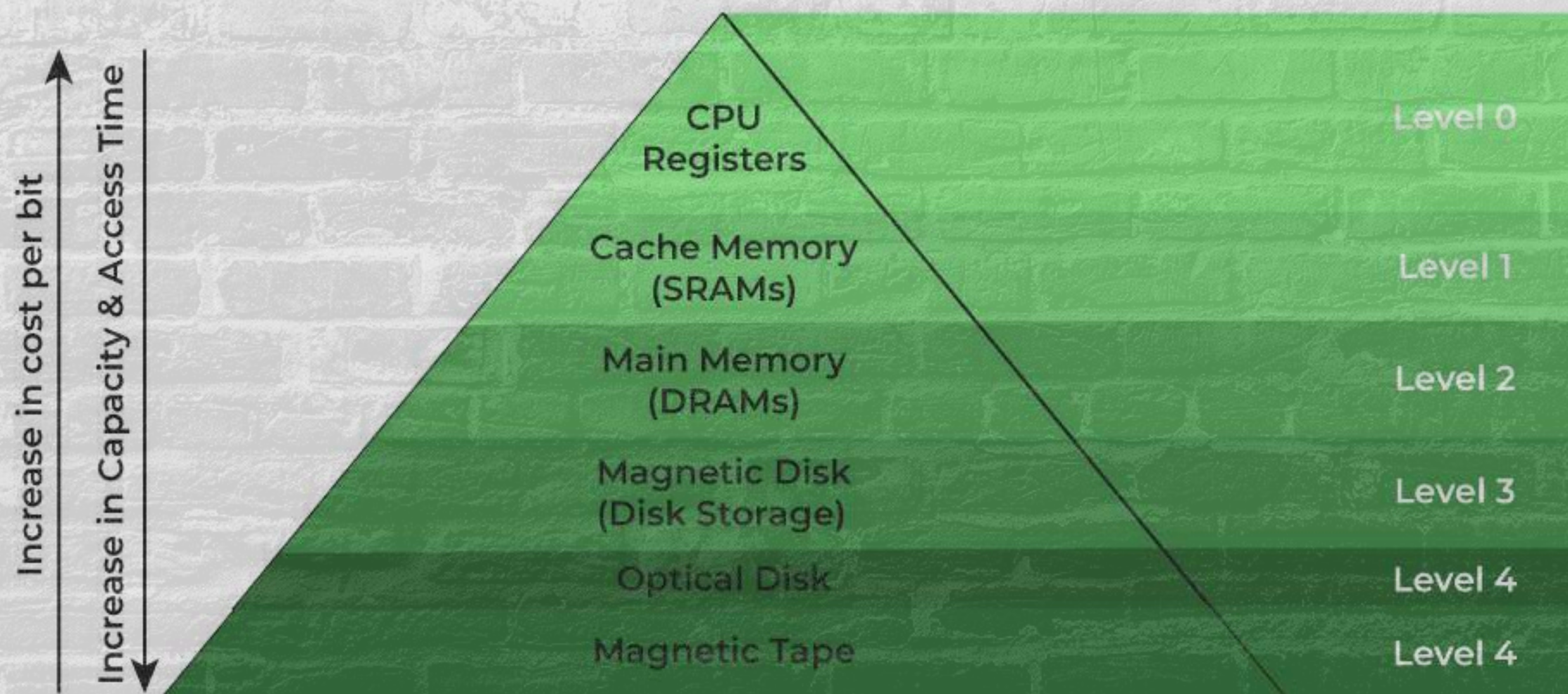
After process execution, the operating system terminates the Process control block(PCB).



Memory Management



Memory Hierarchy



Memory Hierarchy Design



But why?

- **It helps in removing some destruction, and managing the memory in a better way.**
- **It helps in spreading the data all over the computer system.**
- **It saves the consumer's price and time.**

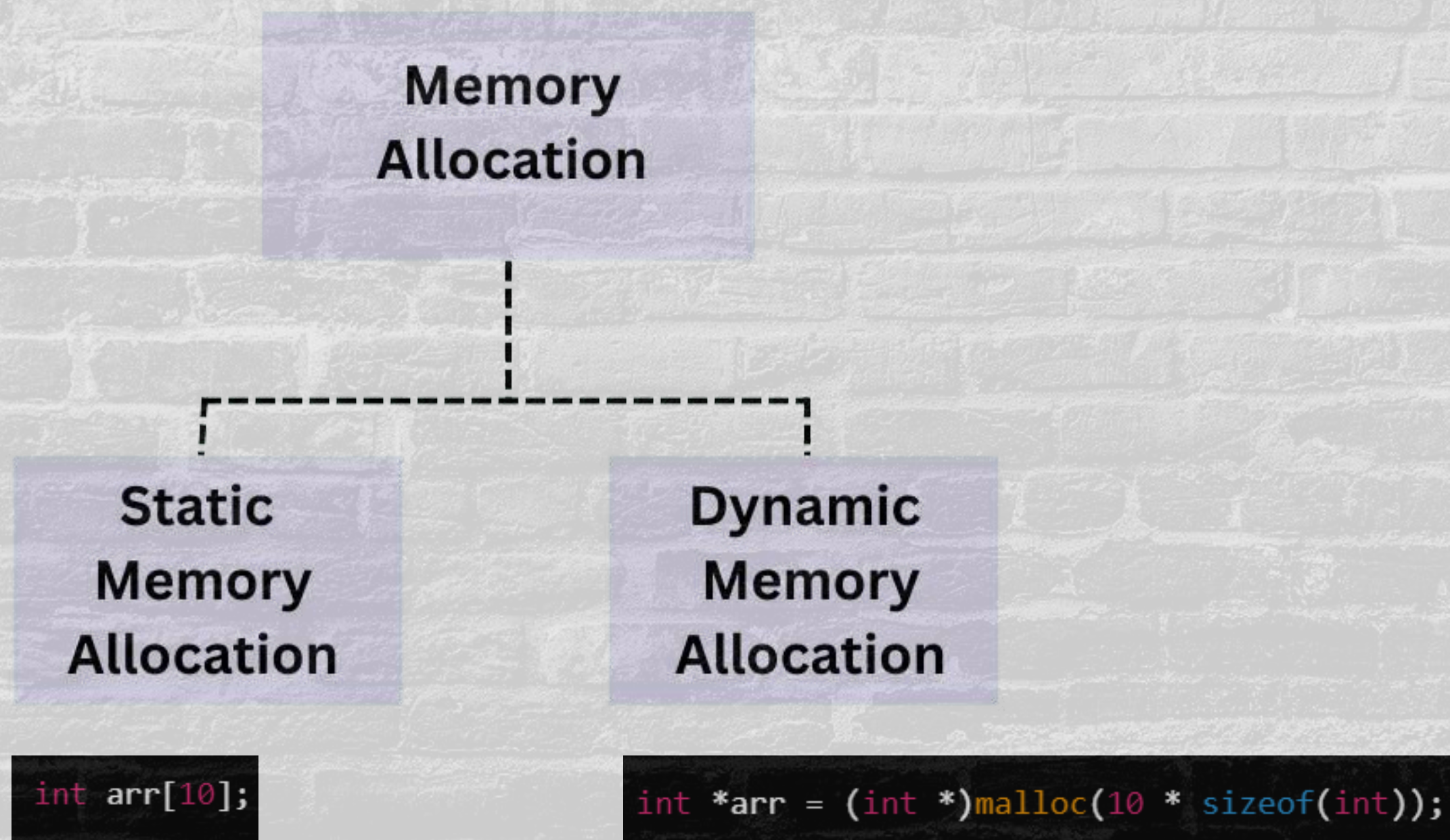


Memory Allocation

- **Memory allocation is a computer hardware operation managed through the operating system.**
- **Through the process of memory allocation, a portion of the computer memory is set aside for running programs and processes. How much memory is allocated depends on the program's requirements.**
- **When the program or application has finished its operation, the memory is released and assigned to another program.**



Types of memory allocation



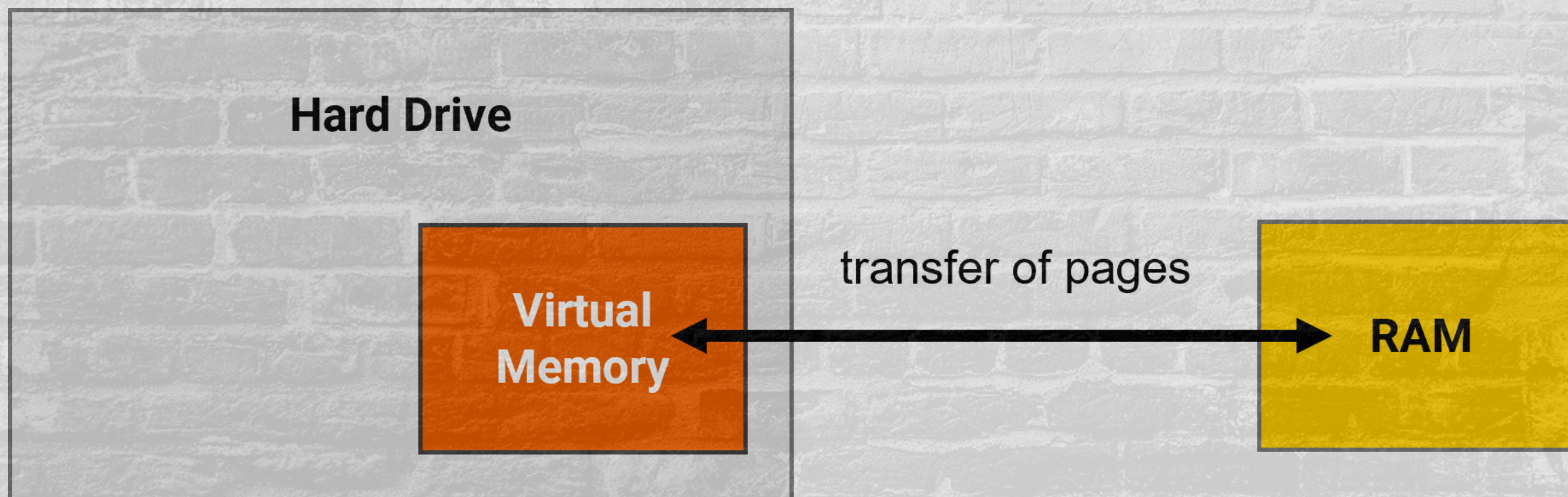


Virtual Memory

When your RAM is full, the operating system uses part of your hard drive or SSD as "virtual memory." It temporarily moves some data from RAM to this virtual memory, so you can keep running more programs even when your physical memory is limited. When you need the data again, the operating system swaps it back into RAM from the virtual memory.



Virtual Memory



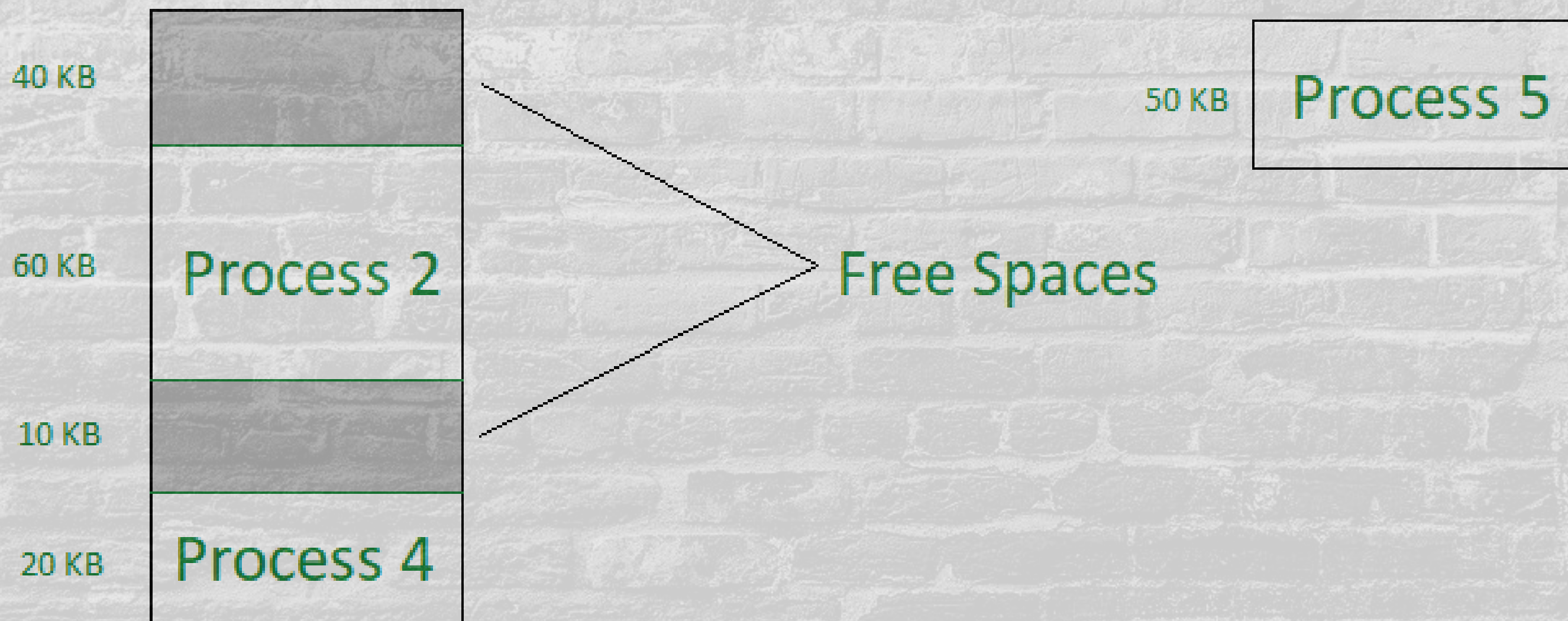


What is Paging?

Paging allows your computer to manage memory efficiently by bringing in only what's needed and swapping out what's not, just like how you manage pages of a book on a small desk. This helps your computer run bigger programs even with limited RAM.



What is Fragmentation?





MEETING'S ALMOST OVER

**THEN SOMEONE ASKS A
QUESTION**



Thank You

شکریہ

