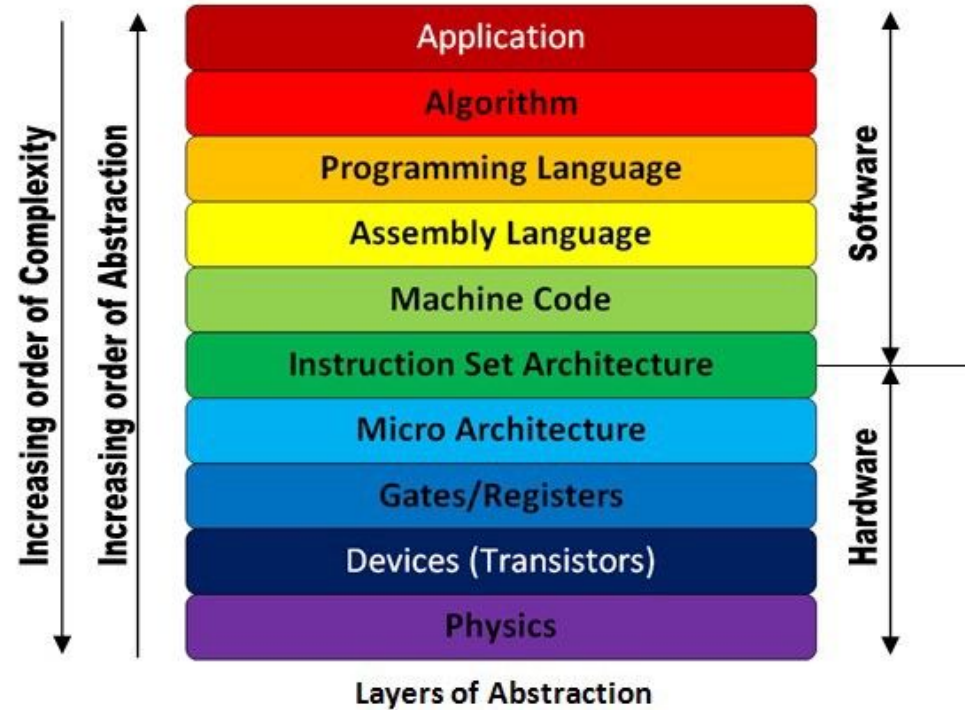


# Low Level System Programming

A dark blue diagonal gradient bar that starts from the bottom left and extends towards the top right, covering the lower half of the slide.

# Computers

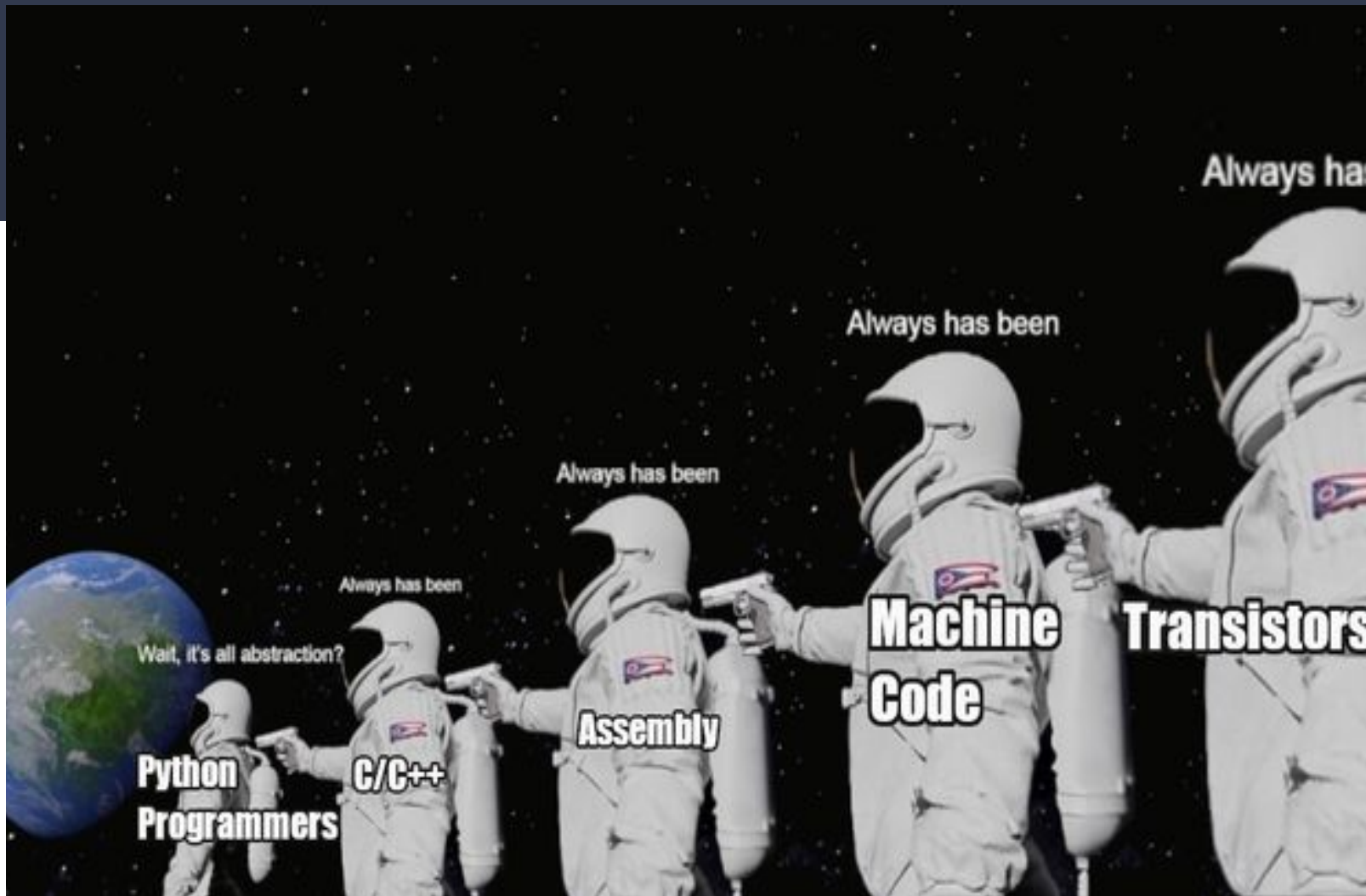


# Low Level

- Provides little abstraction over hardware
- The more we move outward from hardware, the more abstraction
- Ultimately, everything needs to run on some hardware

# Why Low Level?

- Absolute performance
- Maximum use of hardware
- To avoid bottleneck
- Maximum control/ highest flexibility



**Python  
Programmers**

**C/C++**

**Assembly**

**Machine  
Code**

**Transistors**

Wait, it's all abstraction?

Always has been

Always has been

Always has been

Always has

# CPU Programming

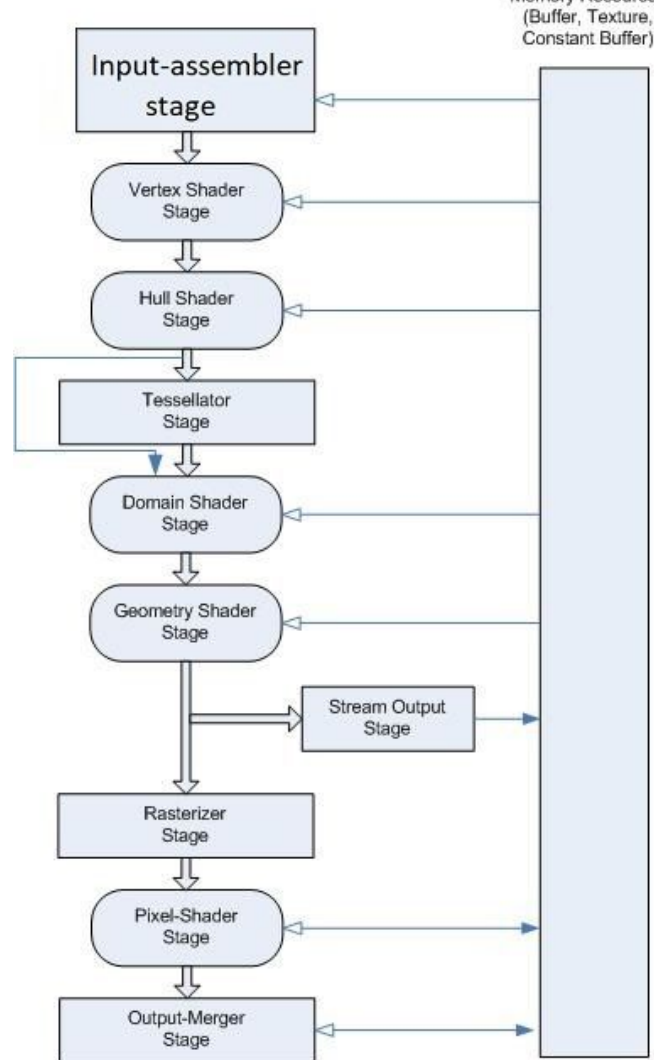
- Programming driven by robustness, correctness, and performance
- Must know the hardware to some degree
- Not necessarily embedded system programming
- Maximum use of hardware

# GPU Programming

- The lowest we need to go for programming graphics is Rendering APIs

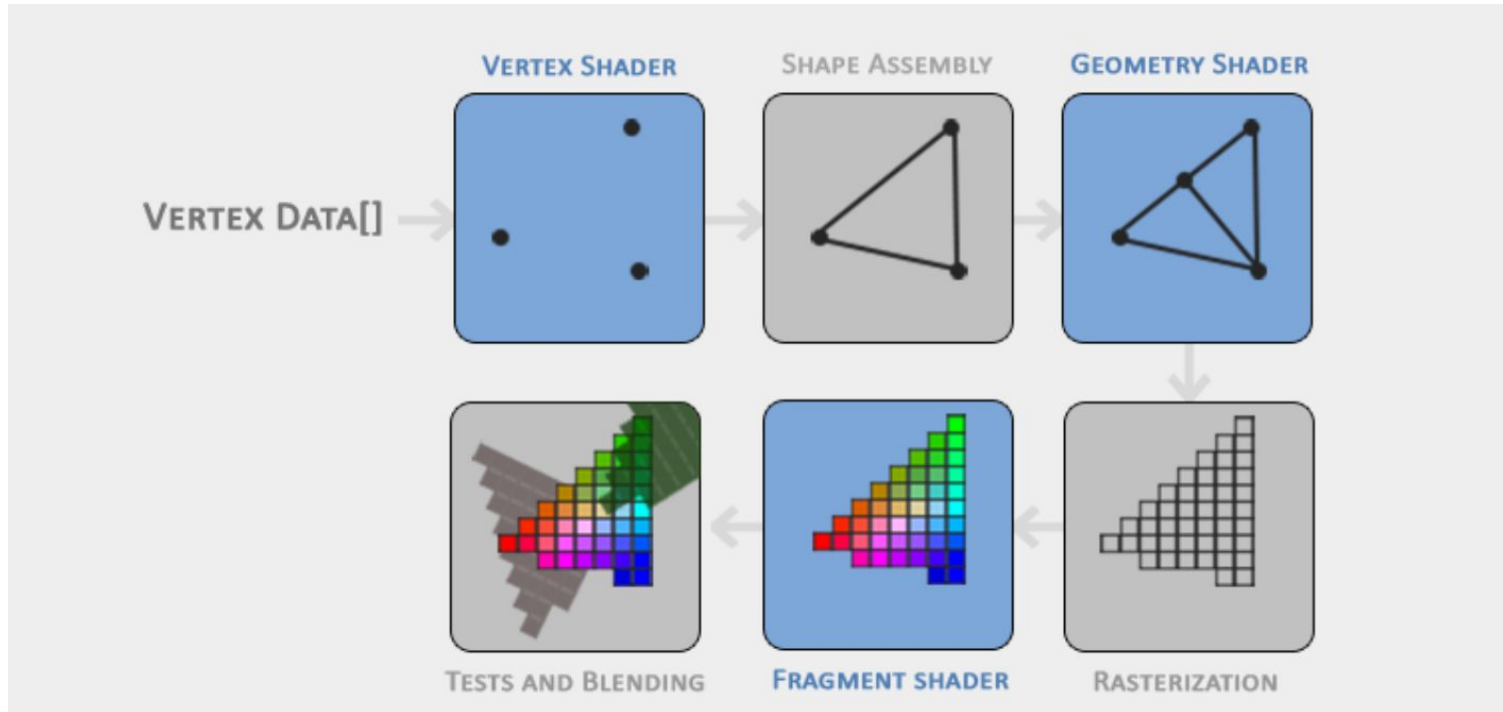


# Graphics Rendering Pipeline





# Rendering Pipeline



# OS API

Windows :

Win32 API

Linux :

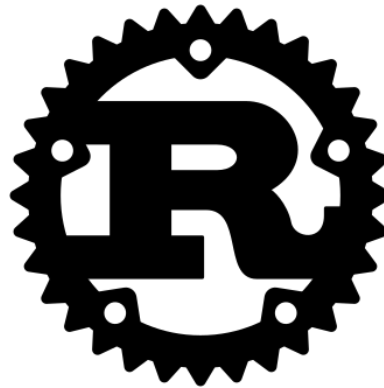
X -> The graphical server

Wayland

Core Kernel API

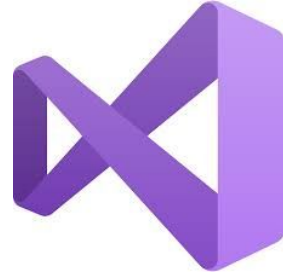
Can directly consumed by 'The C Programming Language'

# ‘Low Level’ Languages



# Development Environment

On Windows :



Linux :



# Tools



GNU Debugger



# Profiler

Output Zoom In Reset Zoom Clear Selection

Diagnostics session: 24.095 seconds

CPU (% of all processors)



Memory Usage CPU Usage

## Function Name

Alternative::ThreadPool::worker\_threads\_func

Total CPU [unit, %]

100200 (98.71%)

Self CPU [unit, %] 🔥

35204 (34.68%)

Parallel::ParallelShadowMapper

28999 (28.57%)

20295 (19.99%)

Parallel::ParallelRenderableDraw

35671 (35.14%)

19371 (19.08%)

Parallel::Rasteriser

10930 (10.77%)

8161 (8.04%)

ShadowMapper::ClipSpace2D

7368 (7.26%)

3362 (3.31%)

## Hot Path

### Function Name

Render (PID: 9804)

Total CPU [unit, %]

101510 (100.00%)

Self CPU [unit, %]

7 (0.01%)

[System Code] ntdll.dll!0x00007ffb5fd02651

101474 (99.96%)

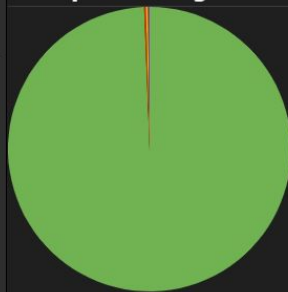
5 (0.00%)

Open details...

Categories

Filter

## Top Five Categories



Kernel : 99.4% (100873)

Graphics : 0.2% (226)

Runtime : 0.2% (226)

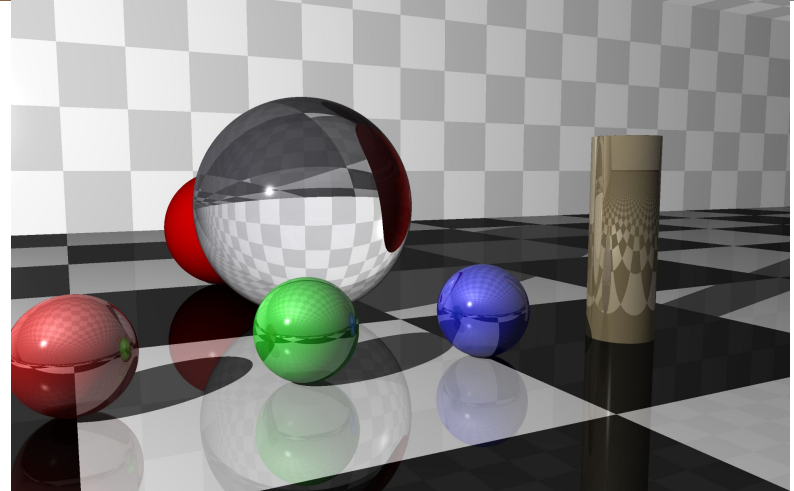
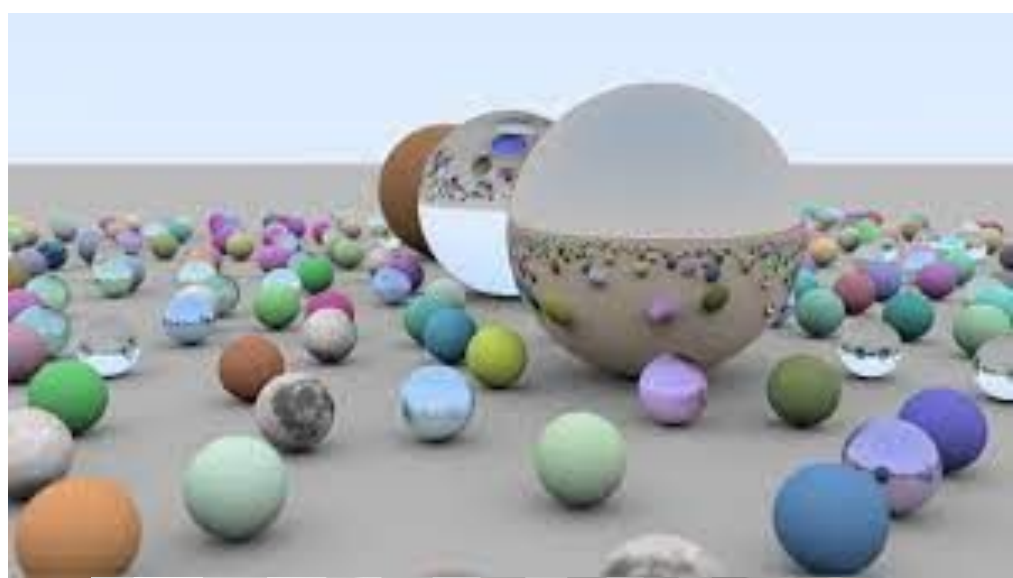
UI : 0.2% (172)

Other : < 0.1% (13)

# Project Ideas

- A ray tracer
- A rasteriser
- Emulator :
  - Chip 8
  - NES Emulator
- A simple interpreted language, maybe just a calculator
- A video decoder
- A HTTP Server
- Parsers (json, csv, xml) -> LL(k), LR(k) and LALR
- Finite State Machines
- Programmable Turing Machine

# Ray Tracer





# References

Intel Developer Manual

<https://www.intel.com/content/www/us/en/developer/articles/technical/intel-sdm.html>

AMD Software Developer Manual

<https://developer.amd.com/resources/developer-guides-manuals/>

Online x86/x64 reference

<https://www.felixcloutier.com/x86/>