Lets start with a little game

lime.fly.dev

Internet & a lot of Things

and all the little stuff we overlooked

soo a bit about me…

soo a bit about me…

Hiiiiiii 👋

2024 Passout

Currently work at 

Lets get back to the game…

Internet & how it works

Internet &
how it works

understanding
cpu & memory

1. The Miracle of DSL

1. The Miracle of DSL

This was the very time internet reached a million people. And was growing at a staggering rate.

1. The Miracle of DSL

This was the very time internet reached a million people. And was growing at a staggering rate.

Affordable Wifi

1. The Miracle of DSL

This was the very time internet reached a million people. And was growing at a staggering rate.

Affordable Wifi

How do we share the internet????

2. The LoopBack Interface

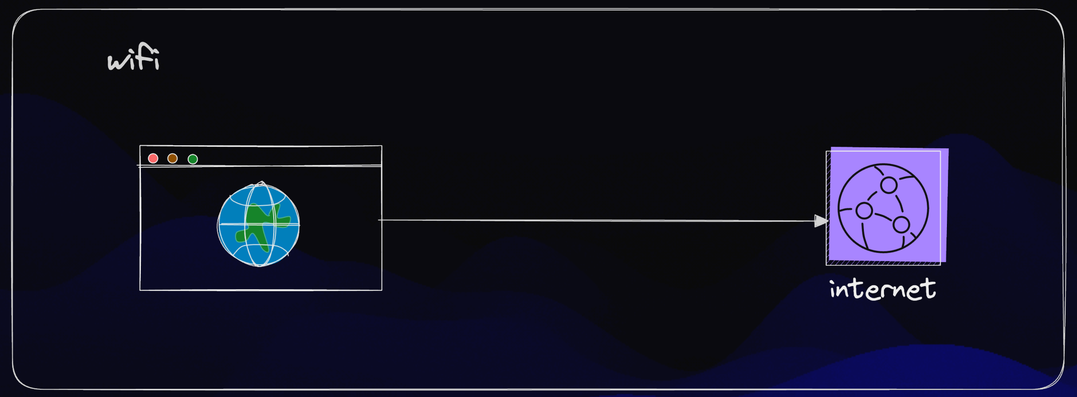
2. The LoopBack Interface

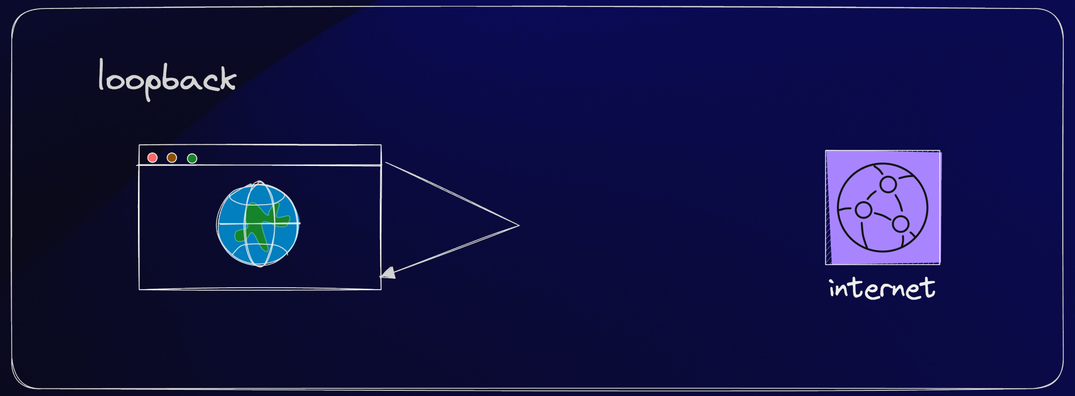
http://127.0.0.1:3000

http://localhost:3000

🙄 ????

2. The LoopBack Interface





2. The LoopBack Interface

127.0.0.1 / 8

16.78 million addresses

127.0.0.0 to 127.255.255.255

2. The LoopBack Interface

RFC - 5735

127.0.0.0/8 - This block is assigned for use as the Internet host loopback address.

A datagram sent by a higher-level protocol to an address anywhere within this block loops back inside the host.

2. The LoopBack Interface

3. The DNS

3. The DNS

RFC - 1034

The RFC is an introduction to the Domain Name System (DNS), and omits many details which can be found in a companion RFC, "Domain Names - Implementation and Specification" [RFC-1035]

4. IPv6 is easy

4. IPv6 is easy

4. IPv6 is easy

IPv4 has 4 billion address

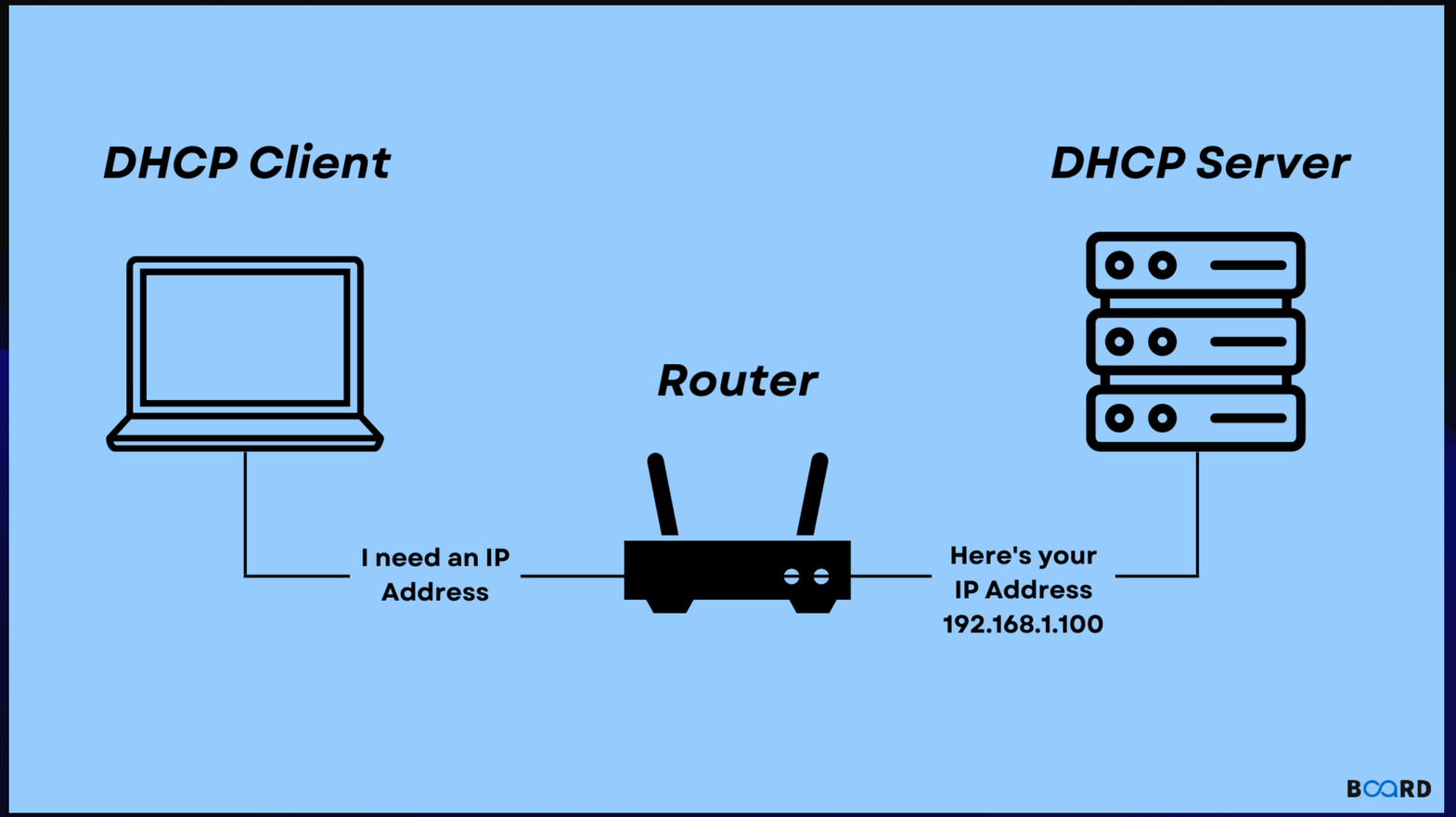
4. IPv6 is easy

IPv4 has 4 billion address

IPv6 has 340 billion billion billion address

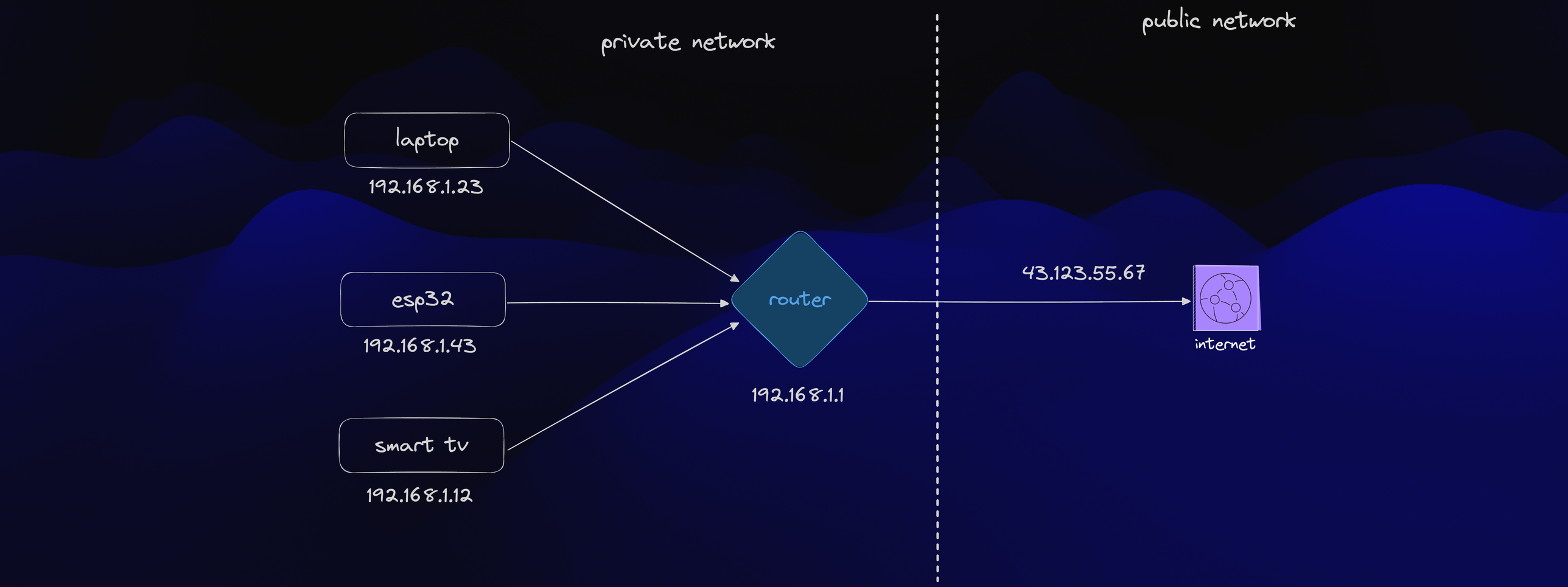
5. DHCP

5. DHCP

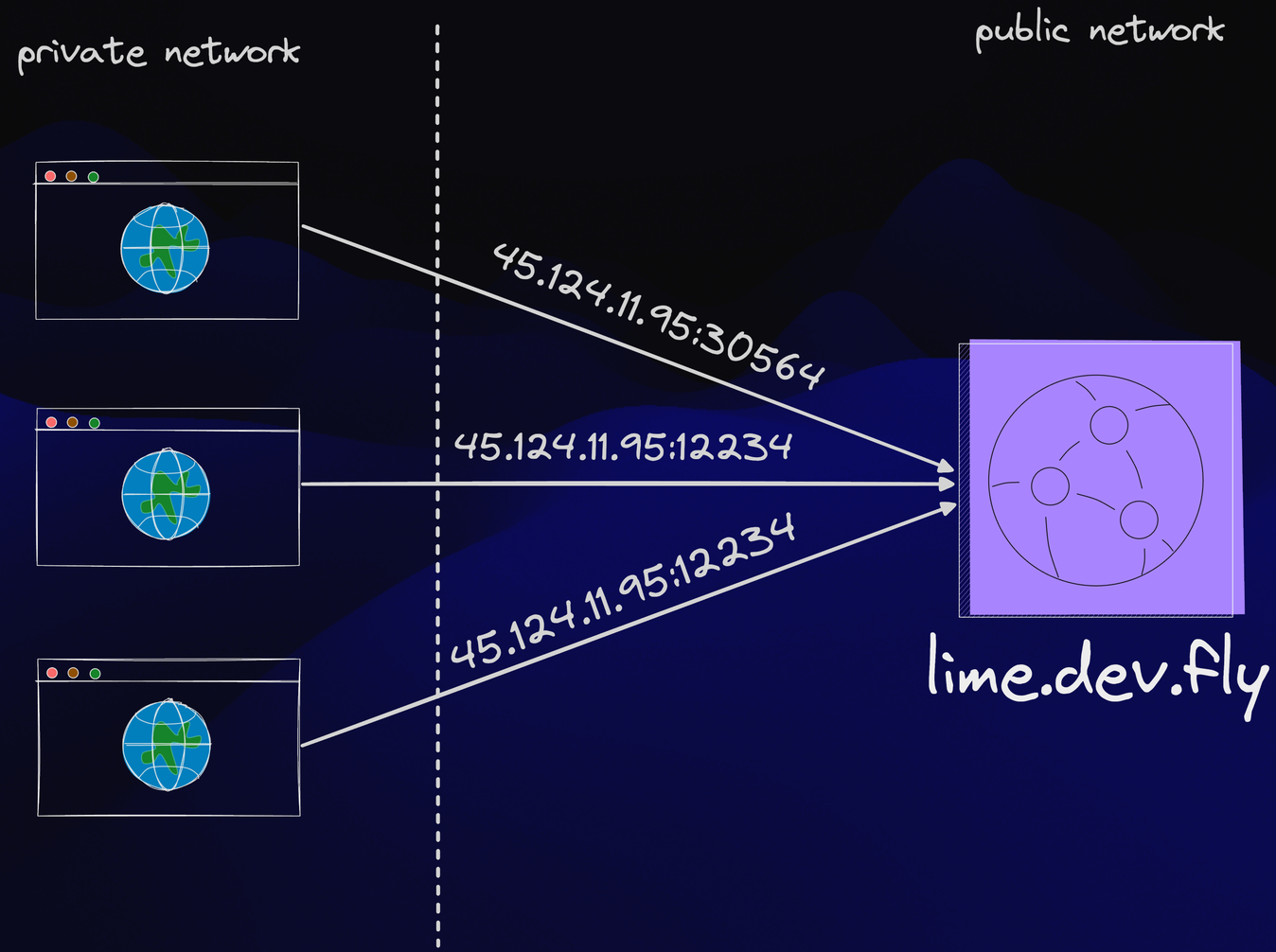

DHCP Client
DHCP Server
Router
I need an IP Address
Here's your IP Address 192.168.1.100
BOARD

6. NAT

6. NAT


private network
public network
laptop
192.168.1.23
esp32
192.168.1.43
smart tv
192.168.1.12
router
192.168.1.1
43.123.55.67
internet

6. NAT

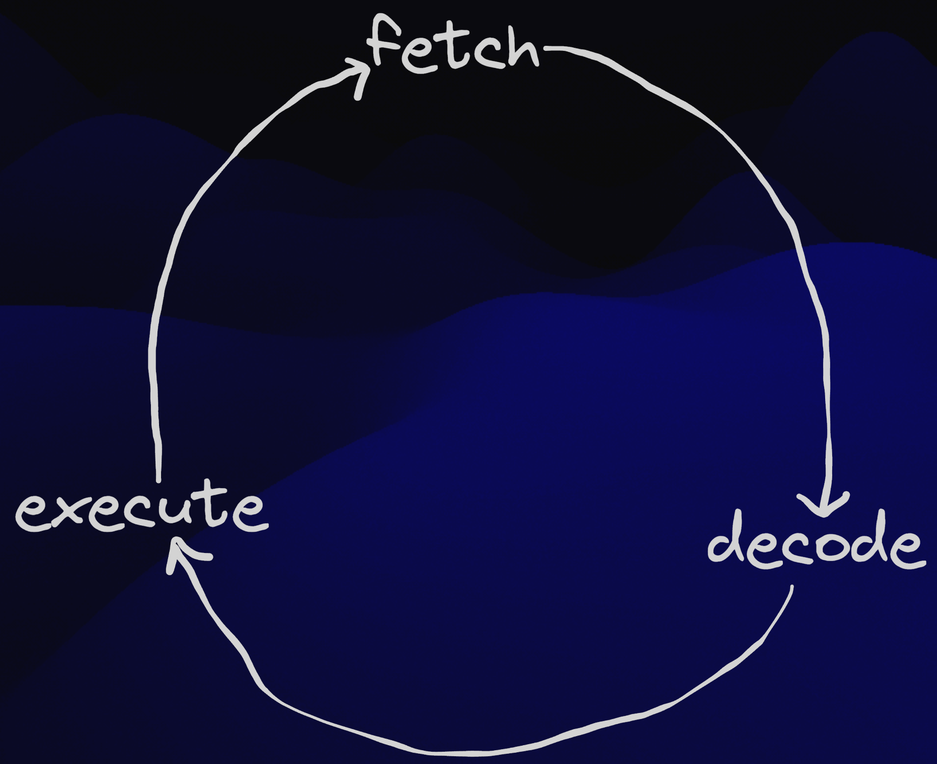

private network
public network
45.124.11.95:30564
45.124.11.95:12234
45.124.11.95:12234
lime.dev.fly

As long as there were no machines, programming was no problem at all; when we had a few weak computers, programming became a mild problem, and now that we have massive computers, programming has become a massive problem.

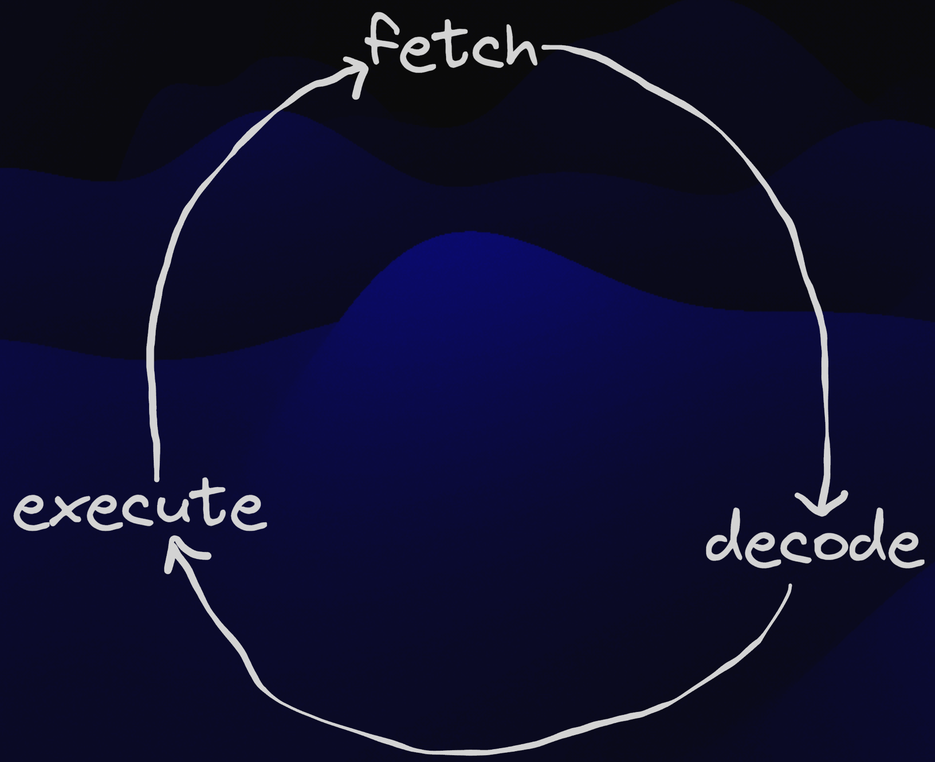
- Edsger Dijkstra

Understanding the CPU

Understanding the CPU

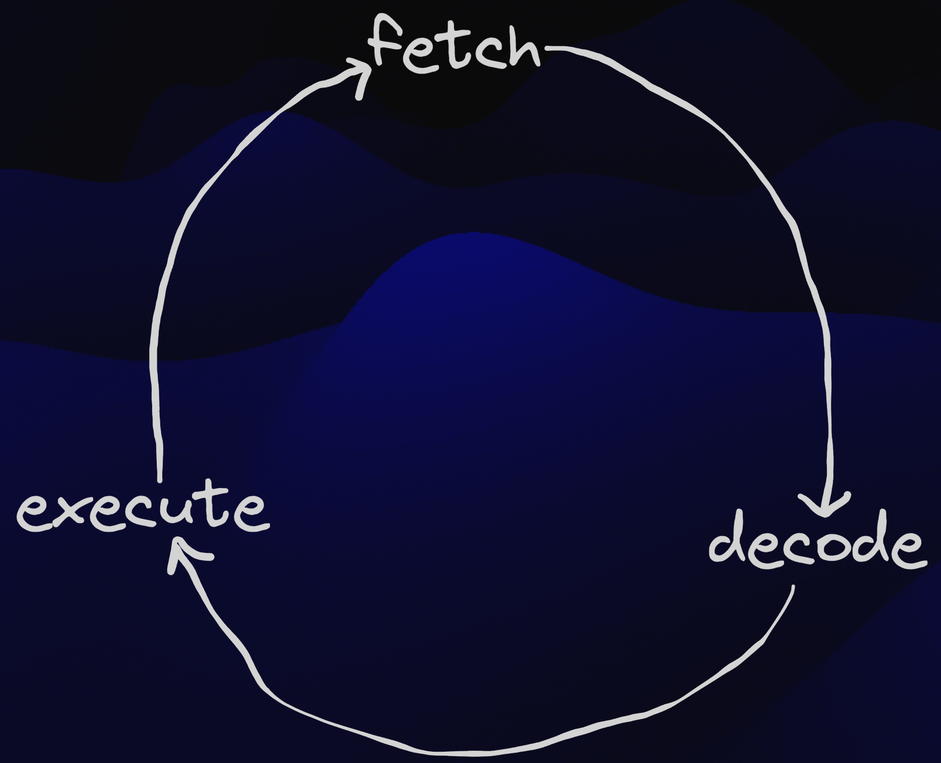


Understanding the CPU



The instruction is fetched from main memory and stored in the processor.

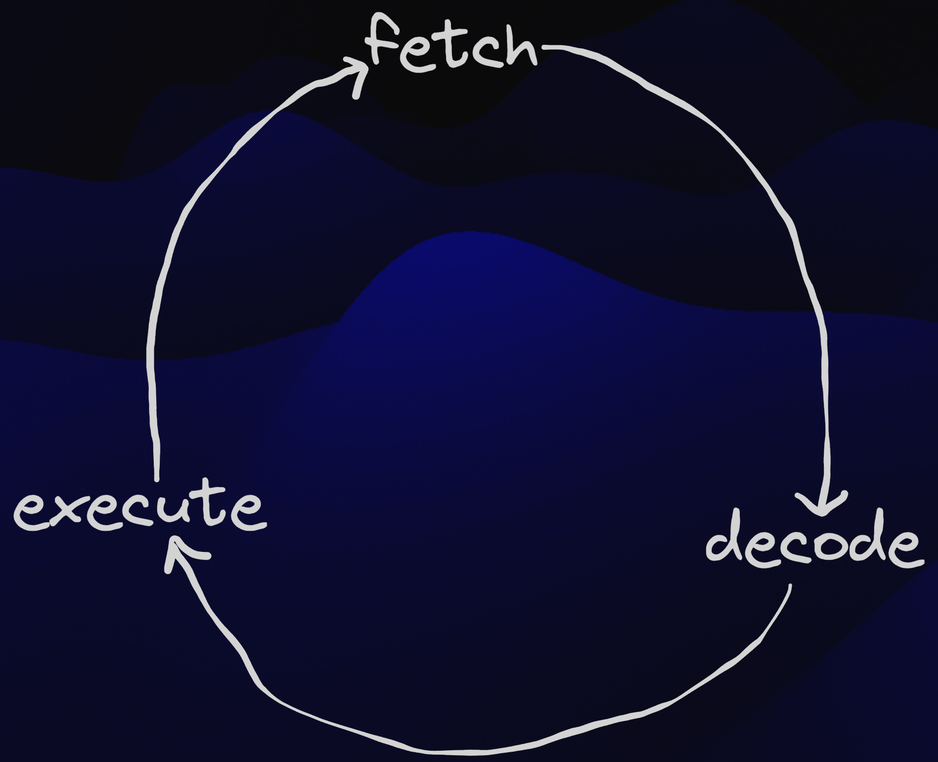
Understanding the CPU



The instruction is fetched from main memory and stored in the processor.

The instructions needs to be decoded before it can be run.

Understanding the CPU

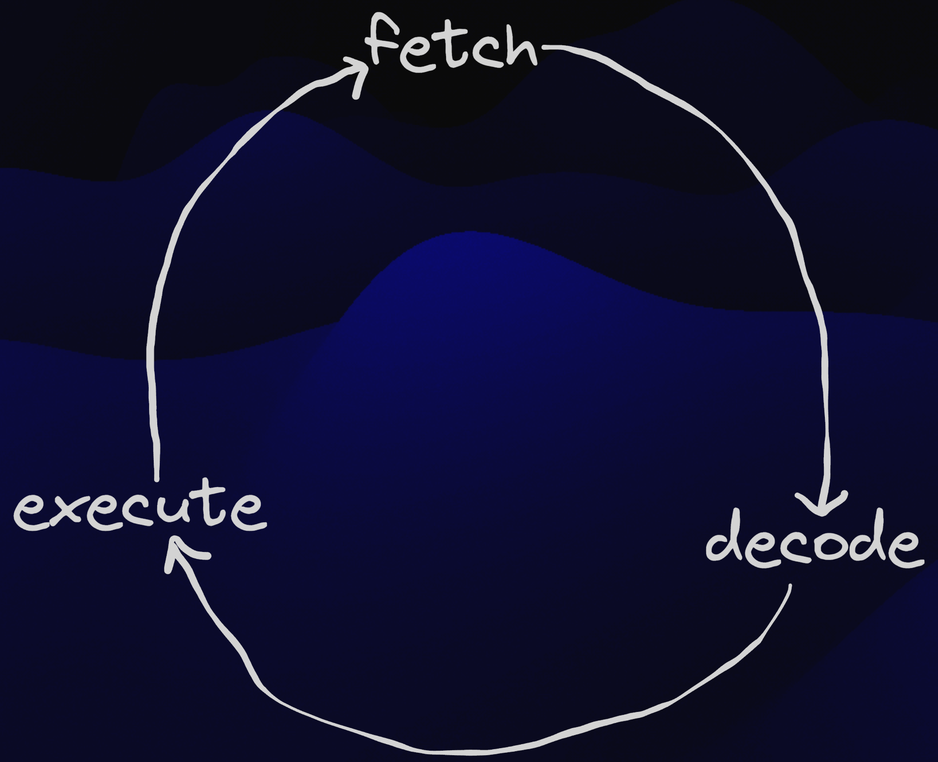


The instruction is fetched from main memory and stored in the processor.

The instructions needs to be decoded before it can be run.

The CPU performs the operations specified by the instruction.

Understanding the CPU



The instruction is fetched from main memory and stored in the processor.

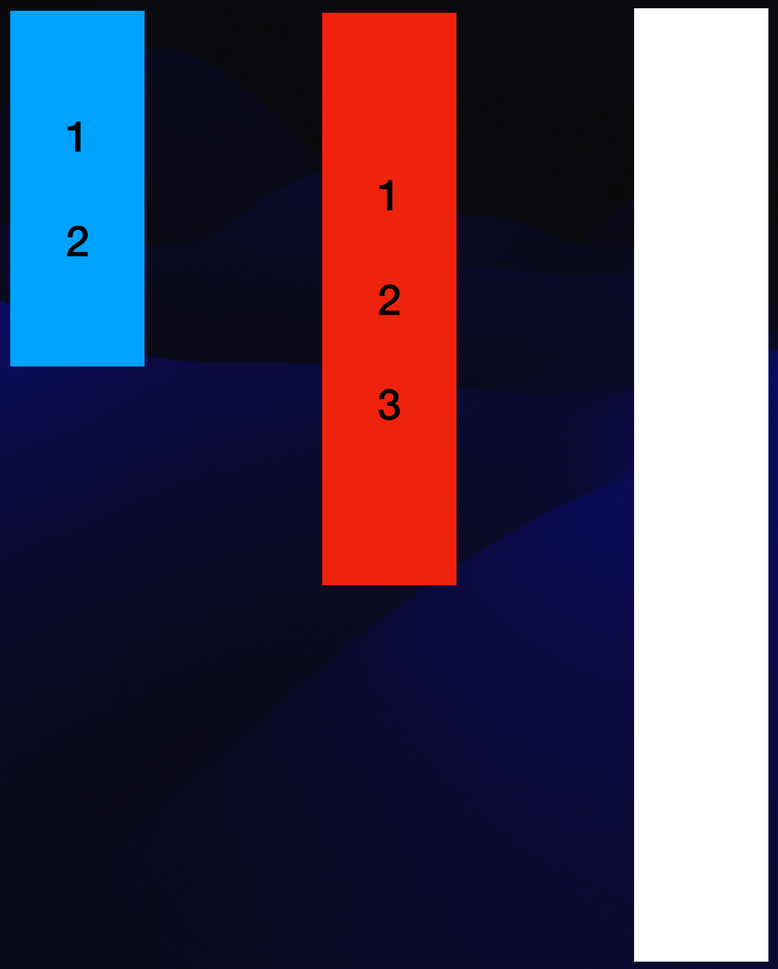
The instructions needs to be decoded before it can be run.

The CPU performs the operations specified by the instruction.

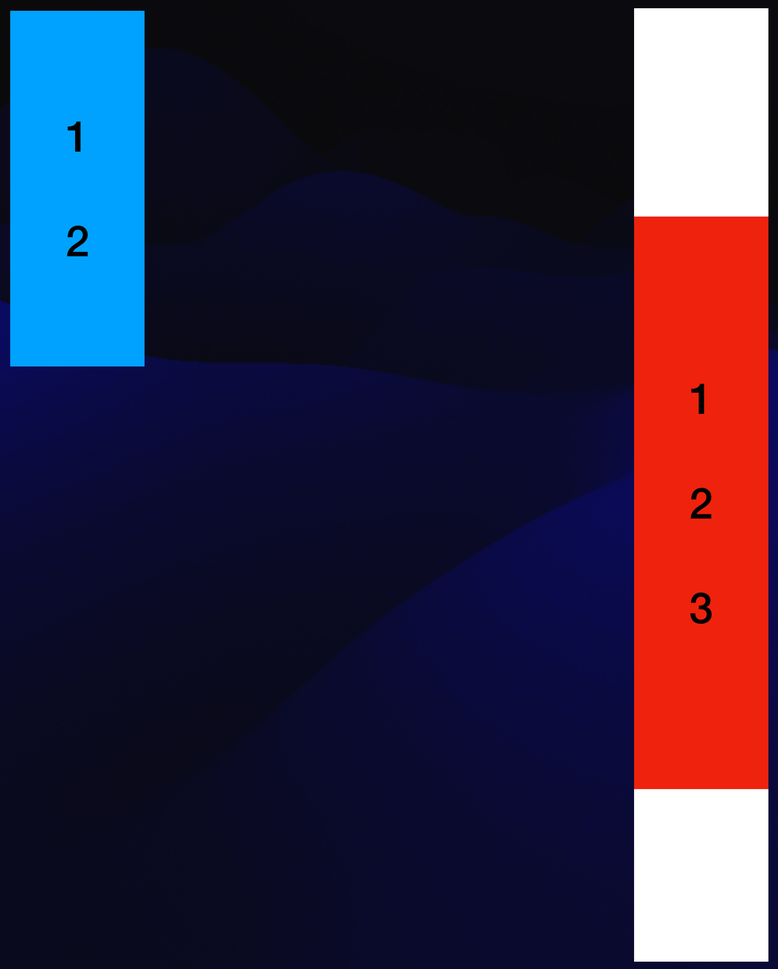
Repeat…

More Memory!!!

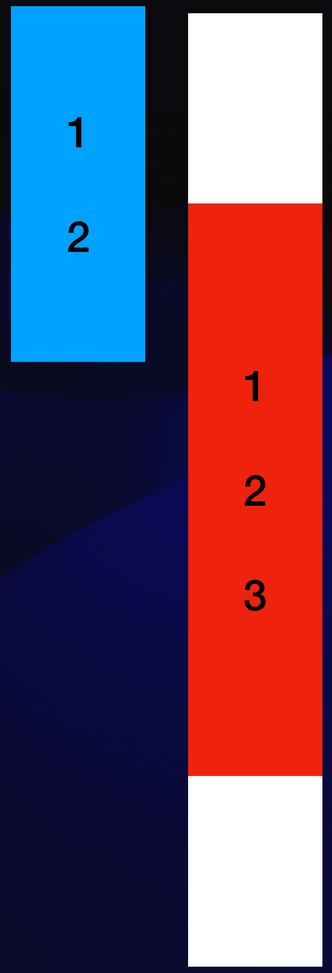
More Memory!!!



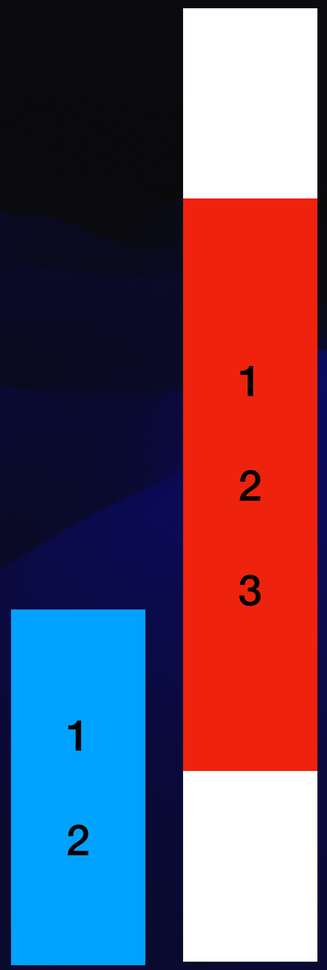
More Memory!!!



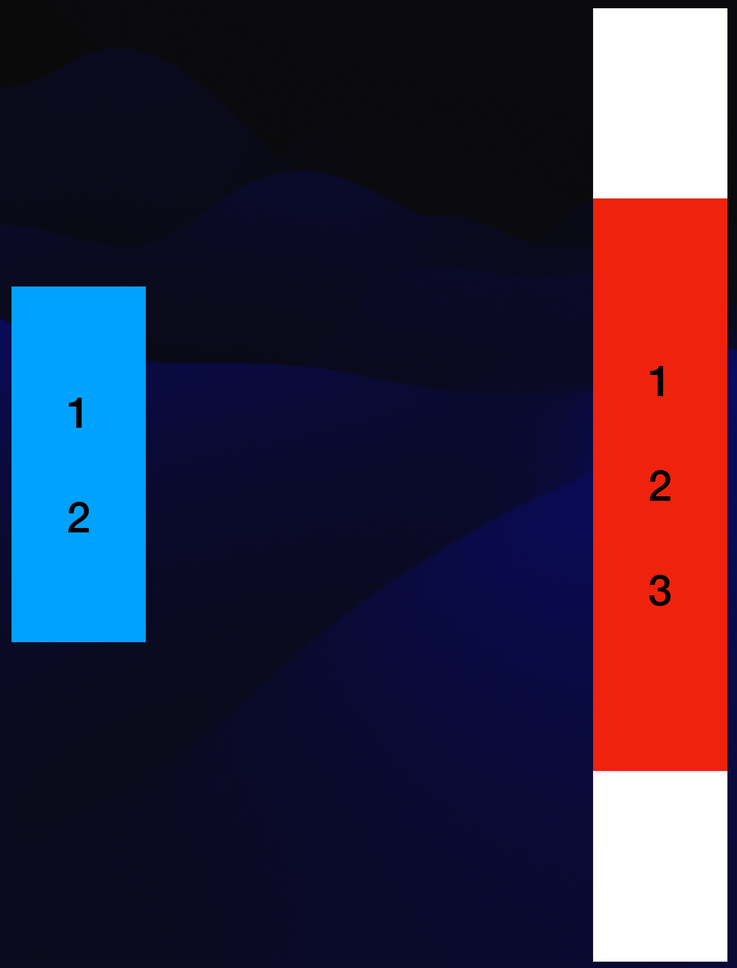
More Memory!!!



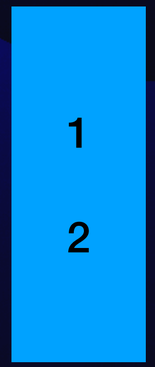
More Memory!!!

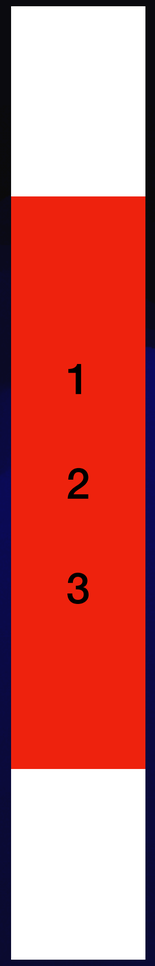


More Memory!!!



More Memory!!!



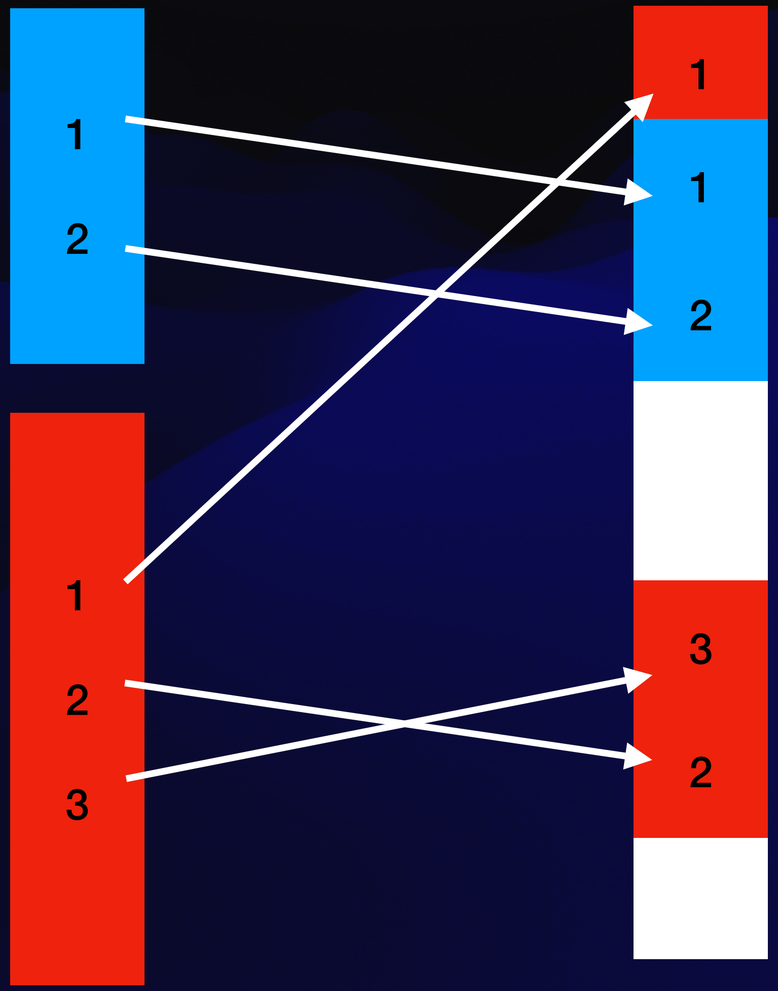


Problem of Physical

Memory Addressing

Virtual Memory Addressing

Virtual Memory Addressing



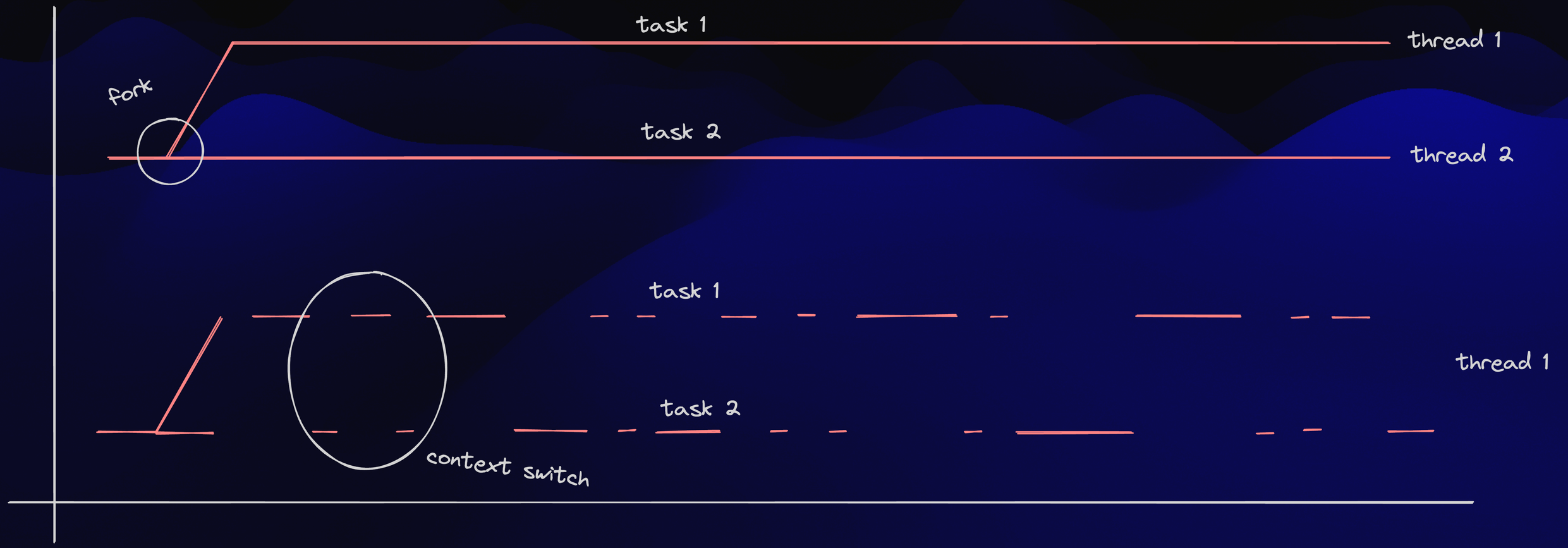
Concurrency vs Parallelism

Concurrency vs Parallelism

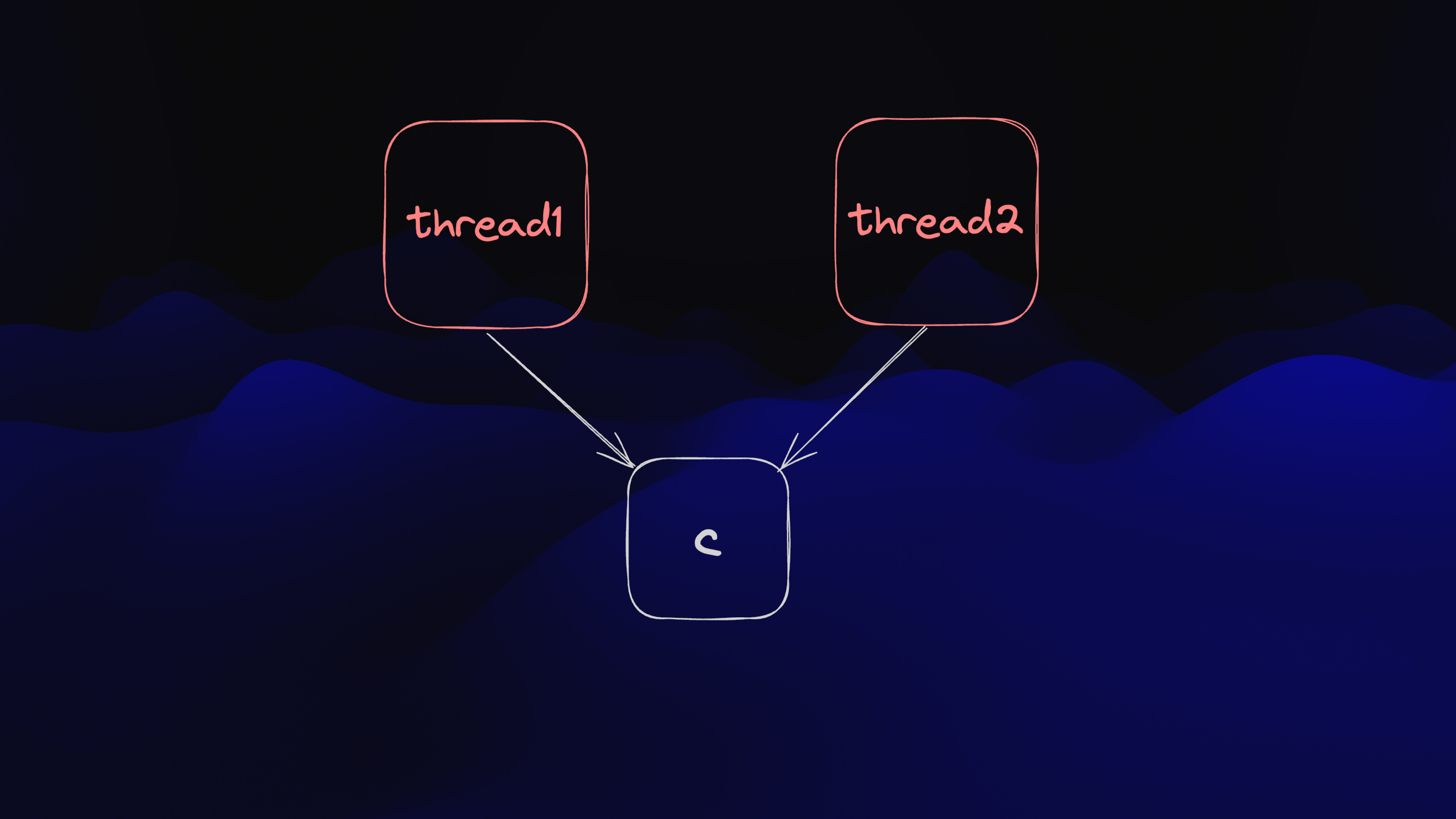
Concurrency is about **dealing with lots of things** at once.

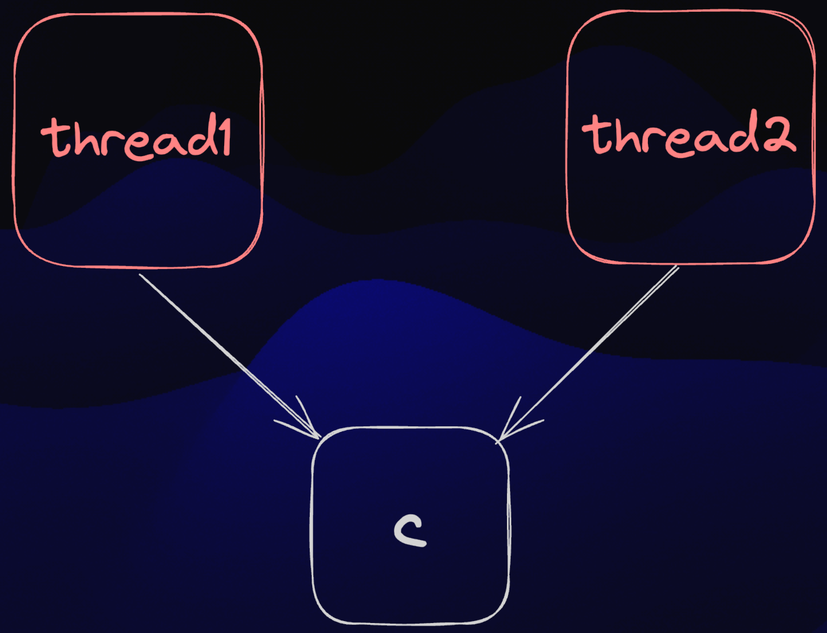
Parallelism is about **doing lots of things at once**.

Concurrency vs Parallelism



Race Conditions


thread1
thread2
c




```
#include <iostream>
#include <thread>

int c;
int t = 1000000;

void run() {
    for (int i = 0; i < t; i++) {
        ++c;
    }
}

int main() {
    std::thread thread1(run);
    std::thread thread2(run);

    thread1.join();
    thread2.join();

    std::cout << c << "\n";
    return 0;
}
```

That’s a lot…

github.com/lmnzx/iotricity