JS Concurrency

What is Concurrency

Concurrency

Multitasking for Programs

What is Concurrency?

- Concurrency is the ability for different parts of a program to execute out-of-order without affecting the outcome of the program.
- This allows those concurrent parts of the program to be executed in parallel, which can greatly speed up execution of the overall program.
- Almost every computer system and programming language implements this idea differently.

What is concurrency used for?

PROCESS SCHEDULING

GRAPHICS PROCESSING

NETWORK I/O

EVENT SYSTEMS

Hardware vs. Software concurrency

Hardware

- Limited to the number of physical processor cores
- Code that can be executed in parallel is limited by the execution set of the processor
- Capable of true parallelism

Software

- Can support arbitrarily many program “threads”
- Threads are software abstractions that can encapsulate any logic the programming language can express
- Simulates parallelism through context-switching between threads

Is JavaScript Concurrent?

Programs in JavaScript are single-threaded

But JavaScript still supports concurrency!

The event Loop

Concurrency in JavaScript

What is an Event Loop?

- When a JavaScript engine (like Node or a browser) runs a JavaScript program it executes it line-by-line, in a single thread.
- At the same time, it allows the program to schedule functions to execute in response to specific events that are managed by the engine.
- The engine then inserts those function calls into the execution call stack of the main program when they’re ready to be called.
- This process of looping through the execution of a program and scheduling interleaved event handlers is called an event loop.

What events can we listen for?

TIMEOUTS AND INTERVALS

USER INPUT: CLICKS, KEYSTROKES, AND MORE

STREAMING DATA (NETWORK OR FILE SYSTEM)

ANYTHING ELSE SUPPORTED BY THE ENGINE