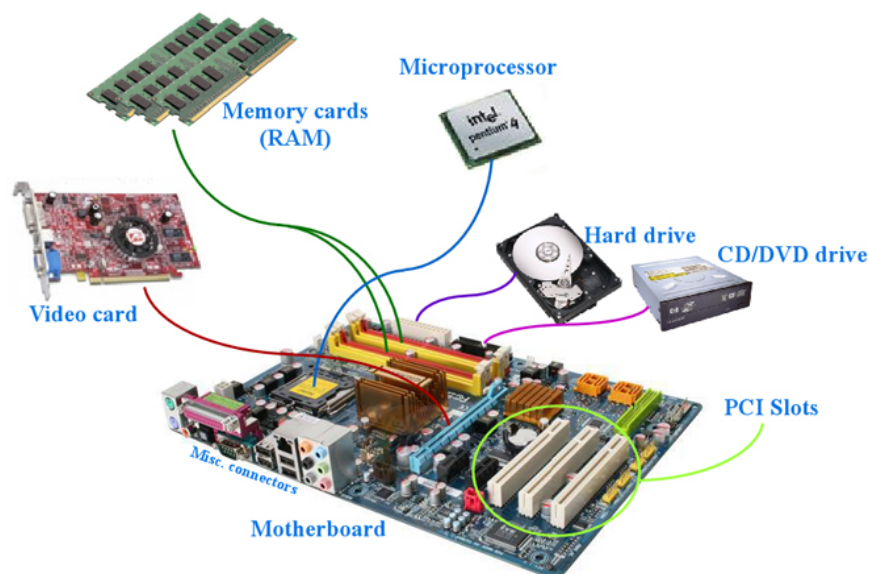


# Introduction/Background Knowledge

## What will you get out of this?

- Be able to talk confidently about algorithms, data structures, and memory
- Show off at job interviews
- Approach leetcode problems with a formal understanding of what they are asking
- Understand the restrictions of a computer and how to wrangle it to do magic
- Learning stuff is fun

## Parts of a computer



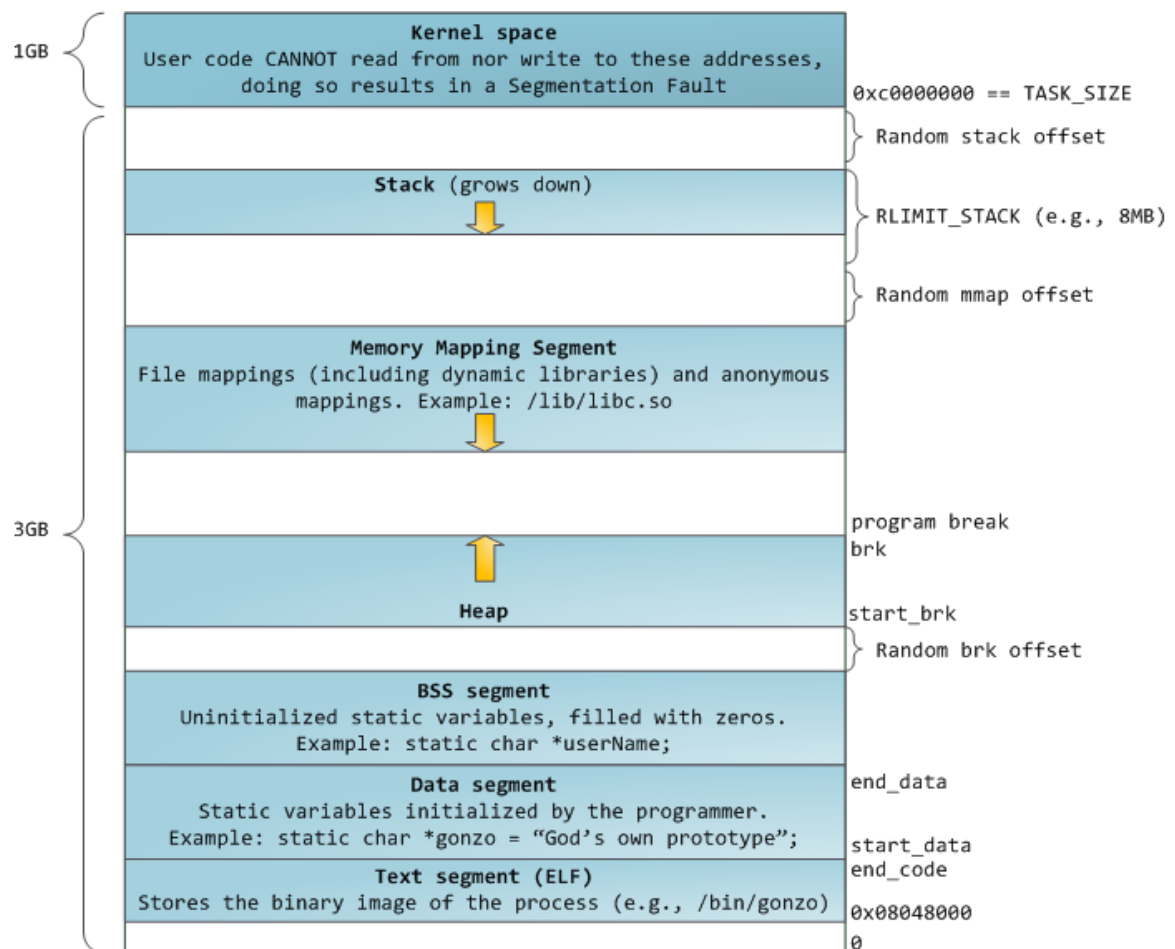
## Where does code run?

### What is the stack?

The stack keeps track of function calls and the variable names/locations associated with those function calls.

### What is the heap?

The heap keep track of the values of variables.



<https://manybutfinite.com/post/anatomy-of-a-program-in-memory/>

```
const foo = () => {
  let x = 0;
  bar();
}

const bar = () => {
  let x = 1;
}

foo();
```

- **What happens when your code runs?**

1. It gets compiled into bytecode or instructions (compiling is sometimes a separate step)

```

DATA SEGMENT
    MESSAGE DB "HELLO WORLD!!!"
ENDS
CODE SEGMENT
    ASSUME DS:DATA CS:CODE
START:
    MOV AX,DATA
    MOV DS,AX
    LEA DX,MESSAGE
    MOV AH,9
    INT 21H
    MOV AH,4CH
    INT 21H
ENDS
END START

```

2. It gets loaded (copy/pasted kinda?) into memory
3. It gets executed

- **What is a stack overflow?**
- **What does an array look like in memory?**