JUST ENOUGH C.S TO BE DANGEROUS

2019, College of Engineering Chengannur

Aswin Mohan

A TALK ON (MOSTLY) EVERYTHING YOU

NEED TO KNOW ABOUT COMPUTING



the world in (0, 1)

BOOLEAN LOGIC

operations on bits

- AND
- OR
- NOT
- NAND
- NOR

Eight bits make a $\underline{\text{byte}}$

- KiloByte = 1024 bytes
- MegaByte = 1024 KB
- GigaByte = 1024 MB
- TeraByte = 1024 GB

LOGIC CIRCUITS

You take some bits and some operations and boom you got a Logic Circuit.

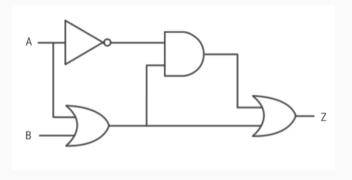


Figure 1: Logic Circuit

CPU - Central Processing Unit

Reprogrammable Logic Gates

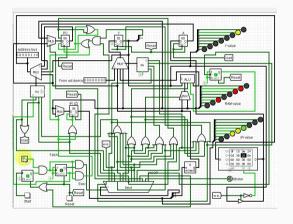


Figure 2: CPU Logic Gates

CODE CODEEE

MACHINE CODE

- Executed directly by the processor
- Corresponds to Logic Circuit in CPU
- Nobody Does it

Program that does something I got from the Internet

```
[ op | rs | rt | rd |shamt| funct]
    0    1    2    6    0    32    decimal
    000000 00001 00010 00110 00000 100000   binary
```

ASSEMBLY LANGUAGE

- Machine Code 10110000 01100001
- Hexadecimal Code B0 61
- Easy to remember names for reading and writing

MOV AL, 61h ; Load AL with 97 decimal (61 hex)

• Assembler is needed to convert Assembly to Machine Code

LOW-LEVEL LANGUAGE

- Assembly is still hard
- \bullet So we code in C and a Compiler converts to Assembler

```
% otool -tv fib
fib:
                                                                       % cat fib.c
(__TEXT,__text) section
                                                                       #include <stdio.h>
main:
                            %rbp
000000100000f20
                    pusha
                                                                        int main(void) {
                            %rsp, %rbp
                    mova
999999199999f21
                                                                            int x, y, z;
                            $0x20, %rsp
000000100000f24
                    suba
                            $0x0, -0x4(%rbp)
                    mov1
0000000100000f28
                                                                            while (1) {
                    mov1
                            $0x0, -0x8(%rbp)
0000000100000f2f
                                                                                x = 0;
                            $0x1, -0xc(%rbp)
                     bv1
9999999199999f36
                            0x56(%rip), %rdi
                                                                                v = 1:
999999199999f3d
                                                                                do {
                            -0x8(%rbp), %esi
0000000100000f44
                                                                                    printf("%d\n", x);
                            $axa
9999999199999f47
000000100000f49
                                        %esi
                                                                                    z = x + y;
9999999199999£4
                                        %esi
                                                                                    x = y;
00000001000
                                        %rbp)
                                                                                    v = z;
9999999199
                                                                                  while (x < 255);
                                         esi
0000000100
99999991996
00000001000
00000001000
999999919996
000000010000
0000000100000
9999999199999
```

Figure 3: C to Assembler

HIGH-LEVEL LANGUAGE

- Low-Level language is fast, but expressive
- So we have High Level Languages
- Not Compiled but Interpreted
- Python, Ruby, Elixir

```
#!/usr/bin/python
                                   #include <stdio.h>
  print "Hello, World!";
                                   int main()
                                      printf("Hello, World! \n");
                                 6
7
                                       return 0;
"Hello, World!" program
                                    "Hello, World!" program
      in Python
                                              in C
```

Figure 4: C to Python



WEB LET'S GET ONLINE

- Desktop Apps are bad and hard to distribute
- \bullet So we have Browsers which have their own languages

WEB LANGUAGES

- HTML, CSS and JAVASCRIPT
- HTML Structure
- CSS Styling
- JAVASCRIPT Interactivity

NEW THINGS

- HTML5
- CSS3
- ES6, ES7 JS

FRAMEWORKS

Don't Repeat Yourself

- The Big Names
- JQUERY
- ANGULAR
- REACT
- VUE



ANDROID

- JAVA aane sir evante main
- XML -> Like HTML and CSS
- \bullet ANDROID STUDIO The place where you type the Code

- Uses Swift
- Needs MAC



REACT NATIVE

- REACT on Mobile and Web
- From Facebook

FLUTTER

- From Google
- Uses Dart



MACHINE LEARNING

MACHINE LEARNING

- Give a bunch of Data, the model learns itself
- No need for Explicit Programming

TENSORFLOW

• The Framework for Machine Learning from Google



BLOCKCHAIN

• Distributed Ledgers

IOT

• Small Small Computers everywhere



LEARNING

- How to learn things
- Use Youtube
- Use Google

GOOGLE FRIEND

GOOGLE FRIEND

 \bullet Check best tutorials to learn Framework

YOUTUBE

YOUTUBE

• Follow Tutorials that build projects



QUANTUM COMPUTING

- QBITS instead of bits
- QBITS are awesome



4 VARSHAM IND VERUTHE ADAKELE

ERINJ KALAYALLE

THANK YOU ;)