



Owen Chase and Óscar Chávez Ortiz

Review

- Create a new text file and write about a thing (or things) we have learned so far
- Turn to someone near you and discuss what you wrote. Did they focus on the same thing or something different?
- Prepare to briefly share

Learning Objectives

At the end of the class you will be able to...

- describe the role of a high level programming language.

- explain why we use Python instead of Bash or Assembly.

- use the Python environment as a calculator.

- write and run a .py file.

Is Python installed?

In terminal type `which python`

Computer Architecture Crash Course

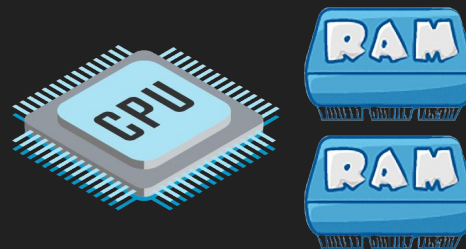
Python, Bash, etc

Assembly

x86, ARM, etc.

```
11 _start:
12     movia    sp, LAST_RAM_WORD
13     movi     r2, '\n'
14     call     PrintChar
15     movia    r2, MSG
16     call     PrintString
17 _end:
18     br       _end
19
20 PrintChar:
21     subi     sp, sp, 8
22     stw      r3, 4(sp)
23     stw      r4, 0(sp)
24     movia    r3, JTAG_UART_BASE
25 pc_loop:
26     ldwio    r4, STATUS_OFFSET(r3)
27     andhi    r4, r4, WSPACE_MASK
28     beq      r4, r0, pc_loop
29     stwio    r2, DATA_OFFSET(r3)
30     ldw      r3, 4(sp)
31     ldw      r4, 0(sp)
32     addi     sp, sp, 8
33     ret
34
35 PrintString:
36     subi     sp, sp, 12
37     stw      ra, 0(sp)
38     stw      r3, 4(sp)
39     stw      r2, 0(sp)
40     mov      r3, r2
41 ps_loop:
42     ldb      r2, 0(r3)
43     beq      r2, r0, end_ps_loop
44     call     PrintChar
45     addi     r3, r3, 1
46     br       ps_loop
47 end_ps_loop:
48     ldw      ra, 0(sp)
49     ldw      r3, 4(sp)
50     ldw      r2, 0(sp)
51     addi     sp, sp, 12
52     ret
```

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	ADD				ES PUSH		SS POP		OR				CS PUSH		TWO BYTE POP DS	
1	ADC								SBB				CS PUSH		POP DS	
2	AND				ES POP		DAA		SUB				CS POP		DAS	
3	XOR				SS POP		AAA		CMP				DS POP		AAS	
4	INC								DEC							
5	PUSH								POP							
6	PUSHAD	POPAD	BOUND	ARPL	FS	GS	OPLOCK	ST	NOLOCK	PUSH		IMUL	PUSH	IMUL	INS	OUTS
7	JO	JNO	JB	JNB	JE	JNE	JBE	JA	JS	JNS	JPE	JPO	JL	JGE	JLE	JGT
8	ADD/ADC/AND/OPR/SBB/SUB/CMPI				TEST				XCHG		MOV REG		MOV SREG		LEA	
9	NOP				XCHG EAX				CWD		CDQ		CALL		WAIT	
A	MOV EAX				MOVS				CMPS		TEST		STOS		LDS	
B	MOV								SCAS							
C	SHIFT IMM		RETN		LES		LDS		MOV IMM		ENTER		LEAVE		RET	
D	SHIFT 1		SHIFT CL		AAM		AAD		SALC		XLAT		FPU			
E	LOOPNZ		LOOPZ		JECQZ		JNC		IN IMM		OUT IMM		CALL		JMP	
F	LOCK		ICE BP		REP		REP		HLT		CMC		TESTING		IMUL	
	CLC		STC		CLD		STD		CLI		STI		CJ		INC	
	NO OPR		NO OPR		NO OPR		NO OPR		NO OPR		NO OPR		NO OPR		NO OPR	
	Arithmetic & Logic				Prefix				Stack				Control Flow & Conditional			
	Memory				System & I/O				Control Flow & Conditional				Control Flow & Conditional			



SCRATCH



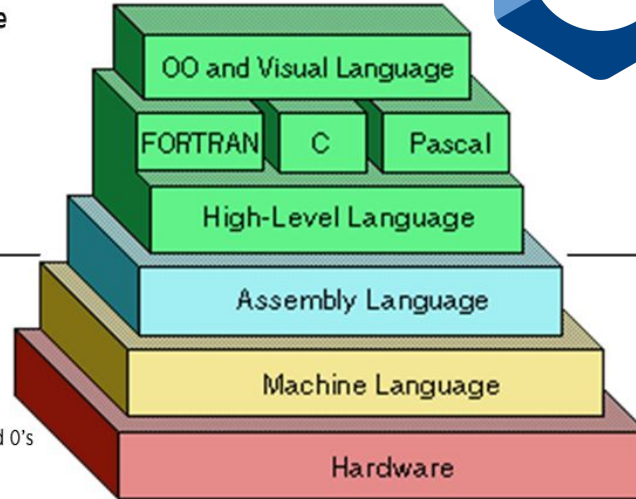
- More machine independent
- Slower

High Level Language

- Easy for Programmers to understand
- Contains English Words

Low Level Language

- The computer's own Language
- Binary numbers, in 1's and 0's



justcode.me

Python or Bash?

2+3

echo \$((2 + 3))

Open Python!

run `python` in the terminal

Basic Math

Try using the four basic math operations:

$+$, $-$, $*$, $/$

Add in some $()$ to test order of operations.

What happens if you use $[]$ or $\{\}$ instead?

Here are some other operations. What do they do?

$**$, $\%$, $//$

Do these operations work with decimals?

Challenge: What do $\&$ and $|$ do?

Variables

To assign a value to a variable, use `=`

Ex: `a = 5`

View the variable's value using its name

Try using some variables. How do they work in expressions?

What happens if we do something like `a = 2 + 3`?

What about `b = a`?

What happens if I use the same variable on both sides, e.g. `a = a + 5`

In Python, we can do `+=`, `-=`, `*=`, `/=`

What do those mean?

Basic Math +

How can we write large numbers?

1,000,000? 1×10^6 ?

In Python we'd write:

`1_000_000` or `1*10**6` or `1e6` or `1E6`

Check for equality with `==`

Try the other comparisons: `>`, `<`, `>=`, `<=`

What happens when I do `0.1 * 3 == 0.3`?

Strings

We'll talk about data types later, but for now just know that to use text data in Python, we need to indicate it using `"""` or `'`

Try typing `hi` then `"hi"`

Try doing math with numbers like `'5'`

Try using `+`, `-`, `*`, `/` on strings

Try doing `'string' * int`

Python files

We usually don't want to write and execute a program line-by-line, or we want to be able to share/rerun it without typing it all out again.

We can save our Python programs in files!
(To show output we need to use `print()` now)

Write some python code to a file and save it.
Canonically your code would be: `print("Hello, World")`

Run it with `python filename`

You should generally save Python files with a `.py` extension, but remember, none of Linux, bash, or the python executable care about extensions. The `.py` is just good practice!

One more thing

If you want to give input to your program at runtime, you can use the `input()` function to prompt the user for input.

Ex:

```
a = input("Enter a number")  
print(a)
```

Exercises

Level 1: Write a python program that adds two numbers supplied by the user

Level 2: Write a python program that converts Fahrenheit to Celsius

Level 3: Write a python program that can convert Fahrenheit to Celsius or the reverse based on user input