Assignment 7

Q1

This week we are starting a new chapter.

- True
- ~~False~~

This is true, the assembly chapter is the chapter for week 7

Q2

The program counter keeps track of what instruction to run next.

- True
- ~~False~~

Yes, it holds the memory address of the next instruction that would be executed.

Q3

The Von Neuman model for computer architecture uses the unified memory model.

- ~~True~~
- False

Q4

The design of the x86 processor family use the Reduced Instruction Set (RISC) ideology.

- ~~True~~
- False

No, is is completly different family.

Q5

The frequency of a CPU is the only measure needed to know its true performance.

- ~~True~~
- False

Q6

Cache is just extra memory that the CPU can use to store temporary data during complex calculations.

- True
- ~~False~~

Q7

A program compiled on one Instruction Set Architecture (ISA) can be copied and run, as is, on a different ISA.

- ~~True~~
- False

Q8

The code we write in the programming language is exactly the same as the instructions run on the CPU.

- ~~True~~
- False

Q9

ARM is a CPU designer but not a CPU manufacture.

- True
- ~~False~~

Q10

float is a 16 bit data type.

- ~~True~~
- False

No, float is a 32-bit data type.

Q11

The old IA32 architecture has 8 general purpose registers.

- True
- ~~False~~

Q12

%rax, %eax, %ax, %ah and %al are, in fact, all references the same register.

- True
- ~~False~~

Yes, they point to different parts of the same register.

%rax is the full 64 bit register (bits 0-63) %eax is the lower 32 bits of %rax (bits 0-31) %ax is the lower 16 bits of %rax (bits 0-15) %ah is the high 8 bits of %ax (bits 8-15 of %rax) %al is the low 8 bits of %ax (bits 0-7 of %rax)

Q13

Assembly has neither variable names nor datatypes as such.

- ~~True~~
- False

Assembly has variables and has some form of data types.

Q14

The move instruction always cleans up after it self by putting all 0's in the source.

- ~~True~~
- False

No, it doesn't do that.

Q15

In the ABCD rule, D can be any value.

- ~~True~~
- False

No, the D is a multiplier for the index and is always: 1, 2, 4 or 8.

Q16

In the ABCD rule: when it is used we will always see all 4 values given in the Assembly code.

- ~~True~~
- False

A: A constant displacement(a fixed number like 0x1000) B: A base register (typically EBP or ESP in 32-bit) C: An index register (any general-purpose register like EAX, ECX) D: A scale factor(a multiplier for the index: 1,2,3 or 8)

[Base + Index + Scale + Displacement]

Q17

Whenever we see (some-register) in assembly we know the register is a pointer and the action is to de-reference that pointer.

- ~~True~~
- False

False, when a register is in brackets [] == Dereference. But without brackets it's being used as a regular value, not as a pointer.

Q18

There is a register dedicated to keeping track of where the "top" stack is at all times

- True
- ~~False~~

True, it's called %RSP.

Q19

If we need more space on the stack and it grows, the address pointing to the "top" of the stack increases (i.e., it gets a higher value).

- ~~True~~
- False

No, the address decreases downvards.

Q20

Assuming both %rbx and %rcx contain memory addresses to valid memory; The following command is legal:

movq (%rbx), (%rcx)

- ~~True~~
- False

No, it's not possible to move memory to memory directly. Both of these commands are dereferencing memory.