

ICS'13 homework 4

1. Addressing:

We have following value address binding table:

Address	value	Resgister	value
0x004	0x10	%eax	0x004
0x30C	0x20	%ebx	0x200
0x300	0x30	%ecx	0x100
0x204	0x40	%edx	0x300
0x308	0x50		

Please fill in the blank

Operand	Value
%eax	
(%eax)	
(%ebx,%ecx)	
8(%edx)	
4(%edx,%eax,2)	

2. Stack operation:

Suppose the initial value of %esp is 0x0000FFF4, initial value of %ebp is 0xBFFFFFFC4. The value stored in address 0x0000FFF0 is 0x10, value stored in address 0x0000FFF4 is 0x30, the value stored in address 0xBFFFFFFC4 is 0x20.

We have following x86 assembly code executed sequentially:

```

pushl %ebp      (instruction 1)
movl %esp,%ebp  (instruction 2)
popl %ebp       (instruction 3)
    
```

Question: After each instruction executed, what is the value of %esp and %ebp

(1) Instruction 1:

(2) Instruction 2:

(3) Instruction 3:

3. Arithmetic operation:

Give out each instruction's destination and corresponding value according to the initial value of register or memory

Address	value	Resgister	value
0x100	0x2	%eax	0x100
0x30C	0x20	%ebx	0x200
0x300	0x30	%ecx	0xCF
0xCF	0x41	%edx	0x4
0x108	0x00		

Instruction	Destination	Value
addl %eax,%ebx		
subl %edx,(%eax)		
decl (%ecx)		
leal 6(%eax),%ecx		
addl 8(%eax),%ecx		

4. Open Question:

c	c3	ret
	c9	leave
	00	
	00	
8	00	
	03	
	fc	
	45	
4	c7	movl \$0x3,-0x4(%ebp)
	14	
	ec	
	83	sub \$0x14,%esp
0	e5	
	89	mov %esp,%ebp
	55	push %ebp

The above picture is captured from the slides(No.15) of Lecture 9, maybe you have noticed that different instruction has different size in X86. In the picture, you can compute the size of movl sub and mov by hand. This feature is CISC style, there is also another Instruction Set Architecture called RISC, its instruction has fixed size. Search and read introduction to RISC, (you can just google CISC vs RISC) which one do you think is better? CISC or RISC , tell me your opinion on the advantage or disadvantage. And at the end , Enjoy your holiday~