RUTGERS

CS211 Computer Architecture

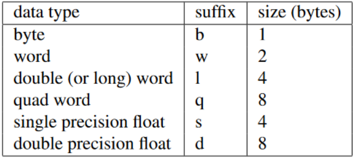
Fall 2020

Recitation 8

Reference

https://www.felixcloutier.com/x86/

Reminder:



data type	suffix	size (bytes)
byte	b	1
word	w	2
double (or long) word	l	4
quad word	q	8
single precision float	s	4
double precision float	d	8

mov vs lea

Does lea attempt to access memory? How about mov?

mov vs lea

- mov – load a value
- lea – load effective address

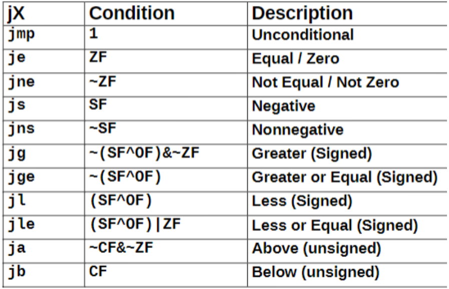
- **lea** source, destination □ won’t “dereference” address like mov; moves address itself

Example:

```
%eax = 0x20, %ebx = 0x4, 0x28 = $50
leal (%eax, %ebx, 2), %ecx *source = 0x28 □ moves 0x28 into ecx
movl (%eax, %ebx, 2), %ecx *source = 0x28 □ moves 50 into ecx
```

Jump Statements

- Recursion and loops can be implemented using **jump statements**
  - Jump to another part of code if given condition is satisfied
  - Carries out conditions give which **carry flags** are set to act as “true” or “false”



jX	Condition	Description
jmp	1	Unconditional
je	ZF	Equal / Zero
jne	~ZF	Not Equal / Not Zero
js	SF	Negative
jns	~SF	Nonnegative
jg	~(SF^OF)&~ZF	Greater (Signed)
jge	~(SF^OF)	Greater or Equal (Signed)
jl	(SF^OF)	Less (Signed)
jle	(SF^OF)|ZF	Less or Equal (Signed)
ja	~CF&~ZF	Above (unsigned)
jb	CF	Below (unsigned)

GDB/HW4 Review

Important notes

- Disassemble the assembly (duh) - **disas**
- Breakpoints are how you will prevent the bomb from exploding – **b** <function name or address>
    - Function names for each phase are “phase_1, phase_2, etc”
    - Function name that gets called to explode is “explode_bomb”
- Show register info – **i r**
- Print contents at address or register – **x <…>**
    - x $rax; x 0x5175879; x/s $rdi