Smith College

Computer Science

# CSC231—Assembly

Week #10 — Spring 2017

Dominique Thiébaut
dthiebaut@smith.edu

# Conditional Jumps

# Example with Jnz

```
_Start:
        mov     ecx, 10
for:    . . .


        dec     ecx         ;ecx <– ecx - 1
        jnz     for         ;if previous op didn't result in 0 in ALU
                            ; then jump
        . . .               ; else continue here…
```

# Family of Conditional Jumps

- **JE, JZ**
- **JNE, JNZ**
- **JG, JGE, JNL**
- **JL, JLE, JNG**

# Family of Conditional Jumps

EAX: 0xFFFF FFFF
EBX: 0x0000 0001
} Which is greater?

- **JE, JZ**
- **JNE, JNZ**
- **JG, JGE, JNL**
- **JL, JLE, JNG**

# Family of Conditional Jumps

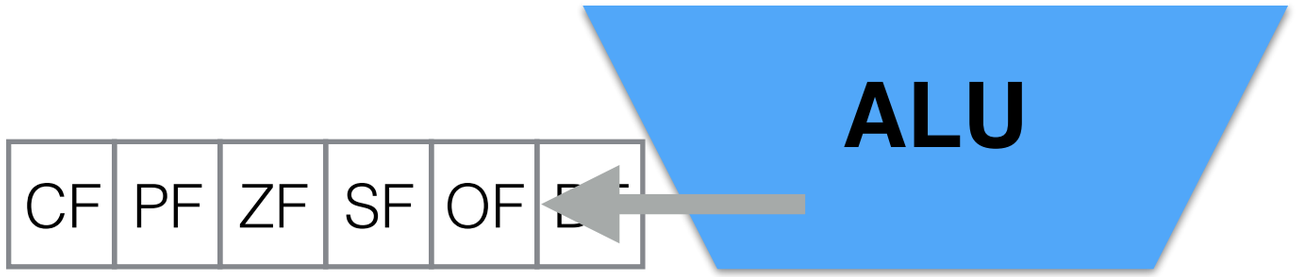
EAX: 0xFFFF FFFF
EBX: 0x0000 0001
} Which is greater?

- **JE, JZ**
- **JNE, JNZ**
- **JG, JGE, JNL**
- **JL, JLE, JNG**

- **JE, JZ**
- **JNE, JNZ**
- **JA, JAE, JNB**
- **JB, JBE, JNA**

# How do we compare two quantities?

```
; if (a==b)
;      c = 3
; else
;      c = -1

        mov     eax, dword[a]           ;eax <- a
        sub     eax, dword[b]           ;eax <- a - b, set flags
        jnz     else                    ;if ZF flag not set, go to else

then:   mov     dword[c], 3             ;otherwise, a==b, set c to 3
        jmp     done                    ;and skip else part

else:   mov     dword[c], -1            ;a!=b, set c to -1

done:   . . .
```




```
; if (a==b)
;      c = 3
; else
;      c = -1

        mov     eax, dword[a]           ;eax <- a
        sub     eax, dword[b]           ;eax <- a - b, set flags
        jnz     else                    ;if ZF flag not set, go to else

then:   mov     dword[c], 3             ;otherwise, a==b, set c to 3
        jmp     done                    ;and skip else part

else:   mov     dword[c], -1            ;a!=b, set c to -1

done:   . . .
```

**sub** is ok, but it
modifies the dest operand

```
; if (a==b)
;      c = 3
; else
;      c = -1
```

```
        mov     eax, dword[a]       ;eax <- a
        sub     eax, dword[b]       ;eax <- a - b, set flags
        jne     else                ;if ZF flag not set, go to else

then:   mov     dword[c], 3         ;otherwise, a==b, set c to 3
        jmp     done                ;and skip else part

else:   mov     dword[c], -1        ;a!=b, set c to -1

done:   . . .
```

**cmp** is better, it subtracts b from eax, ***but does not modify eax***

```
; if (a==b)
;      c = 3
; else
;      c = -1

        mov     eax, dword[a]       ;eax <– a
        cmp     eax, dword[b]       ;compute a - b, set flags
        jne     else                ;if ZF flag not set, go to else

then:   mov     dword[c], 3         ;otherwise, a==b, set c to 3
        jmp     done                ;and skip else part

else:   mov     dword[c], -1        ;a!=b, set c to -1

done:   . . .
```

# Another example with cmp

```
; int a, c   // signed ints
; if (a < 10)
;      c = 3
; else
;      c = -1

        cmp     dword[a], 10        ;compute a - 10, set flags
        jnl     else                ;if not less than 10, go to else

then:   mov     dword[c], 3         ;a<10, set c to 3
        jmp     done                ;and skip else part

else:   mov     dword[c], -1        ;a >= 10, set c to -1

done:   . . .
```

# Another example with cmp

or...

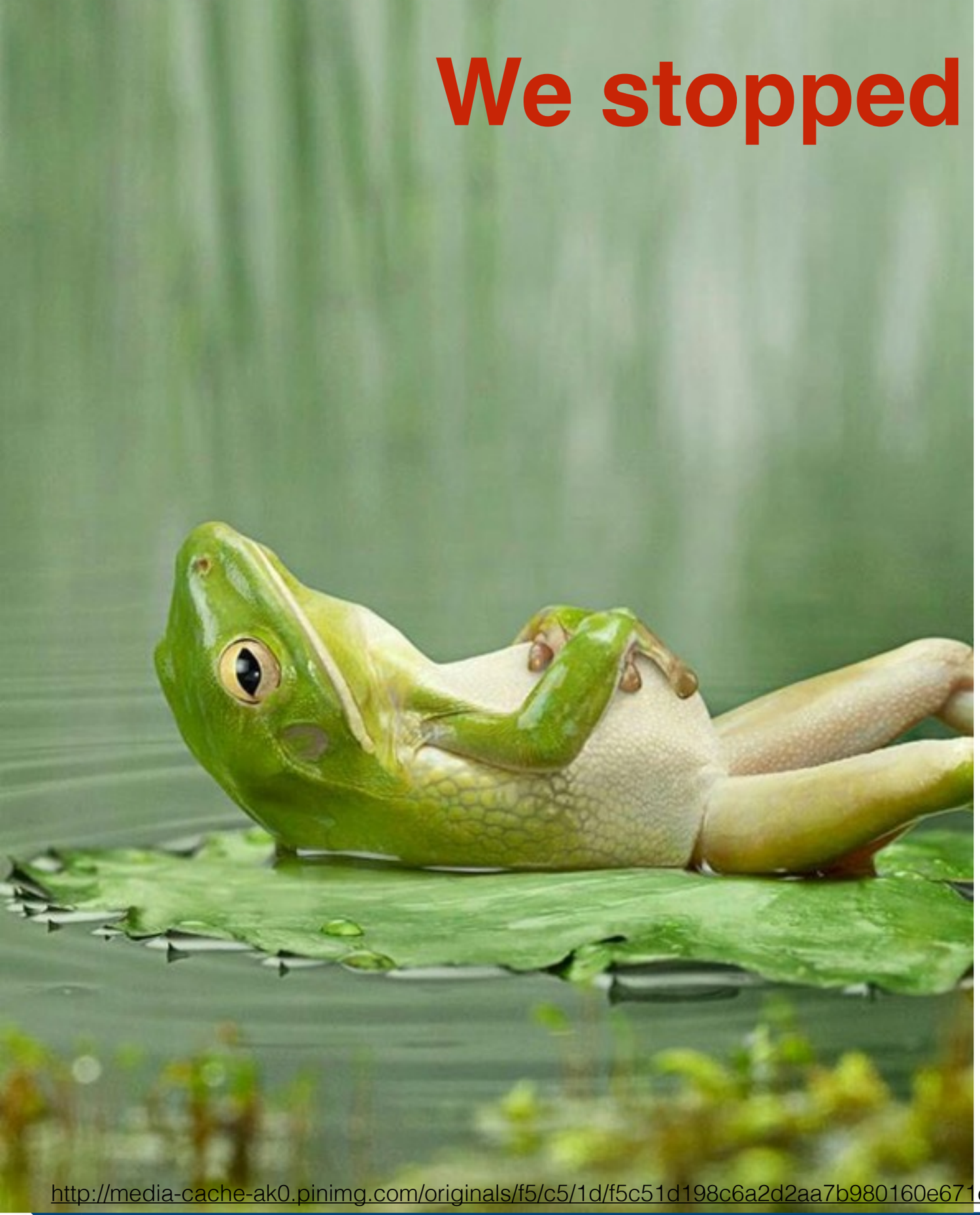
```
; int a, c   // signed ints
; if (a < 10)
;      c = 3
; else
;      c = -1
```

```
        cmp     dword[a], 10        ;compute a - 10, set flags
        jl      then                ;if a<10 go to then
else:   mov     dword[c], -1        ;otherwise, a>=10, set c to -1
        jmp     done                ;and skip then part

then:   mov     dword[c], 3         ;a < 10, set c to 3

done:   . . .
```

# Exercise 1


```
; int sum = 0
; for (int i=0; i<20; i+=2 ) {
;     sum += i;
; }
```

# We stopped here last time…



http://media-cache-ak0.pinimg.com/originals/f5/c5/1d/f5c51d198c6a2d2aa7b980160e671efc.jpg

# A Quick Detour: ddd

DDD: Registers

Registers

| | | |
|---|---|---|
| eax | 0x8 | 8 |
| ecx | 0x0 | 0 |
| edx | 0x0 | 0 |
| ebx | 0xd | 13 |
| esp | 0xffffda40 | 0xffffda40 |
| ebp | 0x0 | 0x0 |
| esi | 0x0 | 0 |
| edi | 0x0 | 0 |
| eip | 0x80480a4 | 0x80480a4 <_start+36 |
| eflags | 0x202 | [ IF ] |
| cs | 0x23 | 35 |
| ss | 0x2b | 43 |
| ds | 0x2b | 43 |

Integer registers  All registers

Close  Help

DDD: /Users/classes/231b/handout/demoDDD.asm

File Edit View Program Commands Status Source Data Help

0: `x /3xw &table`

Lookup Find» Break Watch Print Display Plot Hide Rotate Set Undisp

```
0x80490b0 <a>:  0x00000003
0x80490b4 <b>:  0x00000005
0x80490b8 <sum>:  0x00000008
0x80490bc <table>:  0x00000006  0x00000007  0x0000000d
```

```
28              nop                     ; we put nops to
29              nop                     ; set a breakpoint first
30
31              mov     eax, dword[a]
32              add     eax, dword[b]
33              mov     dword[sum], eax
34
35              mov     ebx, dword[table]
36              add     ebx, dword[table+4]
37              mov     dword[table+8], ebx
38
39 ;;; exit
40              mov     ebx, 0
41              mov     eax, 1
42              int     0x80
```

Run Interrupt Step Stepi Next Nexti Until Finish Cont Kill Up Down Undo Redo Edit Make

```
0x080480a4 <+36>:    mov    $0x0,%ebx
0x080480a9 <+41>:    mov    $0x1,%eax
0x080480ae <+46>:    int    $0x80
End of assembler dump.
```

```
(gdb) stepi
0x0804809e in _start ()
(gdb) stepi
0x080480a4 in _start ()
(gdb)
```

Display -5: `x /3xw &table` (enabled)

# Exercise 2

Translate this for-loop into assembly

How long will it take a 2GHz Pentium to run the loop?

```
; unsigned int i, sum = 0;
; for (i=0; i<4000000000; i+=2 ) {
;     sum += i;
; }
```

# Exercise 3


```
; unsigned int i, sum = 0
; i = 0
; while ( sum < 1000 ) {
;     sum += i++;
;     if ( i>=100 ) break;
; }
```

# Exercise 4


```
; unsigned int i, sum = 0
; i = 0
; while ( sum < 1000 ) {
;     i++;
;     if ( i%2 == 0 )
;         continue;
;     sum += i;
; }
```