Smith College

Computer Science

CSC231—Assembly

Week #7—Spring 2017

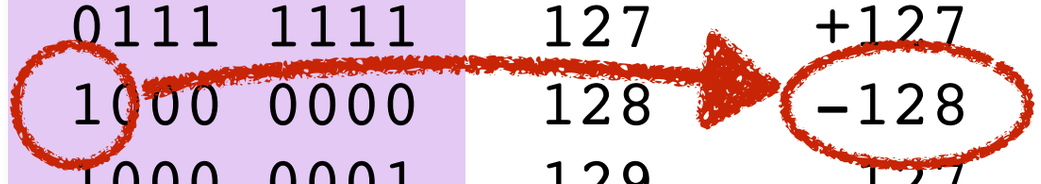
Dominique Thiébaut
dthiebaut@smith.edu

Review

Type	Minimum	Maximum	# Bytes
unsigned byte	0	255	1
signed byte	-128	127	1
unsigned short	0	65,535	2
signed short	-32,768	32,767	2
unsigned int	0	4,294,967,295	4
signed int	-2,147,483,648	2,147,483,647	4
signed long	-9,223,372,036,854,775,808	9,223,372,036,854,775,807	8

Binary	unsigned	signed
0000 0000	0	+0
0000 0001	1	+1
0000 0010	2	+2
…		
…		
0111 1110	126	+126
0111 1111	127	+127
1000 0000	128	-128
1000 0001	129	-127
1000 0010	130	-126
…		
…		
1111 1110	254	-2
1111 1111	255	-1

Binary	unsigned	signed
0000 0000	0	+0
0000 0001	1	+1
0000 0010	2	+2
…		
…		
0111 1110	126	+126
0111 1111	127	+127
1000 0000	128	-128
1000 0001	129	-127
1000 0010	130	-126
…		
…		
1111 1110	254	-2
1111 1111	255	-1



Binary	unsigned	signed	
0000 0000	0	+0	
0000 0001	1	+1	
0000 0010	2	+2	
…			
…			
0111 1110	126	+126	
0111 1111	127	+127	
1000 0000	128	-128	$= -2^7$
1000 0001	129	-127	
1000 0010	130	-126	
…			
…			
1111 1110	254	-2	
1111 1111	255	-1	

Exercises on Signed Numbers

http://www.science.smith.edu/dftwiki/index.php/CSC231_Exercises_on_Signed_Numbers

The **LOOP** Instruction

loop

loop label

```
loop      label
```

```
x       dd      1
sum     dd      0

        mov     ecx, 10
addUp:  mov     eax, dword[x]
        add     dword[sum], eax
        inc     dword[x]
        loop    addUp                   ;ecx<—ecx-1
                                        ;if ecx!=0,
                                        ; goto addUp
```

loop

loop label

```
loop      label
```

```
x       dd      1
sum     dd      0
        mov     ecx, 10
addUp:  mov     eax, dword[x]
        add     dword[sum], eax
        inc     dword[x]
        loop    addUp                   ;ecx<—ecx-1
                                        ;if ecx!=0,
                                        ; goto addUp
```

Label

loop

loop label

```
loop      label
```

```
x       dd      1
sum     dd      0
```

Label

```
        mov     ecx, 10
addUp:  mov     eax, dword[x]
        add     dword[sum], eax
        inc     dword[x]
        loop    addUp                   ;ecx<—ecx-1
                                        ;if ecx!=0,
                                        ; goto addUp
```

Labels

```
_start:     mov     eax, 4

            mov     ecx, 10
for1:       …
            …
            loop    for1

for2:       …
            …
            loop    for2
```

- Start with a **letter**
- End with a **colon** (when declared)
- Represent an **address** in the code section
- Must be **unique** in program

Tracing

One Example

eax

ecx

```
        mov     ecx, 3
        mov     eax, 1
for:    call    _printDec
        inc     eax
        loop    for                 ;ecx<—ecx-1
                                    ;if ecx!=0,
                                    ; goto for
```

eax	**?**
ecx	**3**

```
        mov     ecx, 3
        mov     eax, 1
for:    call    _printDec
        inc     eax
        loop    for                 ;ecx<—ecx-1
                                    ;if ecx!=0,
                                    ; goto for
```

eax 1

ecx 3

```
        mov     ecx, 3
        mov     eax, 1
for:    call    _printDec
        inc     eax
        loop    for                 ;ecx<—ecx-1
                                    ;if ecx!=0,
                                    ; goto for
```

1

eax	**1**
ecx	**3**

```
        mov     ecx, 3
        mov     eax, 1
for:    call    _printDec
        inc     eax
        loop    for             ;ecx<—ecx-1
                                ;if ecx!=0,
                                ; goto for
```

1

eax	~~1~~ 2
ecx	3

```
        mov     ecx, 3
        mov     eax, 1
for:    call    _printDec
        inc     eax
        loop    for                 ;ecx<—ecx-1
                                    ;if ecx!=0,
                                    ; goto for
```

1

eax ~~1~~ 2

ecx ~~3~~ 2

```
        mov     ecx, 3
        mov     eax, 1
for:    call    _printDec
        inc     eax
        loop    for
```

```
;ecx<—ecx-1
;if ecx!=0,
; goto for
```

12

eax	~~1~~ 2
ecx	~~3~~ 2

```
        mov     ecx, 3
        mov     eax, 1
for:    call    _printDec
        inc     eax
        loop    for                 ;ecx<—ecx-1
                                    ;if ecx!=0,
                                    ; goto for
```

12

eax	~~1~~ 2
ecx	~~3~~ 2

```
        mov     ecx, 3
        mov     eax, 1
for:    call    _printDec
        inc     eax
        loop    for                 ;ecx<—ecx-1
                                    ;if ecx!=0,
                                    ; goto for
```

12

eax ~~1~~ ~~2~~ 3

ecx ~~3~~ 2

```
        mov     ecx, 3
        mov     eax, 1
for:    call    _printDec
        inc     eax
        loop    for                 ;ecx<—ecx-1
                                    ;if ecx!=0,
                                    ; goto for
```

12

eax ~~1~~ ~~2~~ 3

ecx ~~3~~ ~~2~~ 1

```
        mov     ecx, 3
        mov     eax, 1
for:    call    _printDec
        inc     eax
        loop    for
```

```
;ecx<—ecx-1
;if ecx!=0,
; goto for
```

123

eax ~~1~~ ~~2~~ 3

ecx ~~3~~ ~~2~~ 1

```
        mov     ecx, 3
        mov     eax, 1
for:    call    _printDec
        inc     eax
        loop    for                ;ecx<—ecx-1
                                   ;if ecx!=0,
                                   ; goto for
```

123

eax ~~1~~ ~~2~~ ~~3~~ 4

ecx ~~3~~ ~~2~~ 1

```
        mov     ecx, 3
        mov     eax, 1
for:    call    _printDec
        inc     eax
        loop    for                 ;ecx<—ecx-1
                                    ;if ecx!=0,
                                    ; goto for
```

123

eax ~~1~~ ~~2~~ ~~3~~ 4

ecx ~~3~~ ~~2~~ ~~1~~ 0

```
        mov     ecx, 3
        mov     eax, 1
for:    call    _printDec
        inc     eax
        loop    for
```

```
;ecx<—ecx-1
;if ecx!=0,
; goto for
```

123

eax ~~1~~ ~~2~~ ~~3~~ 4

ecx ~~3~~ ~~2~~ ~~1~~ 0

```
        mov     ecx, 3
        mov     eax, 1
for:    call    _printDec
        inc     eax
        loop    for                 ;ecx<—ecx-1
        ????                        ;if ecx!=0,
                                    ; goto for
```

Example 1

Sum of 1..10

```
; computes sum(1,2, …10)
x       dd      1
sum     dd      0

        mov     ecx, 1
        mov     eax, dword[x]
addUP:  add     dword[sum], eax
        inc     eax
        loop    addUp                   ;ecx<—ecx-1
                                        ;if ecx!=0,
                                        ; goto addUp

        mov     dword[x], eax
```


```
; computes sum(1,2, …10)
x       dd      1
sum     dd      0

        mov     ecx, 10
        mov     eax, dword[x]
addUP:  add     dword[sum], eax
        inc     eax
        loop    addUp                   ;ecx<—ecx-1
                                        ;if ecx!=0,
                                        ; goto addUp

        mov     dword[x], eax
```

Example 2
Fibonaccis

http://i.dailymail.co.uk/i/pix/2016/07/17/13/1A32203E000005DC-3694326-image-m-17_1468760305397.jpg

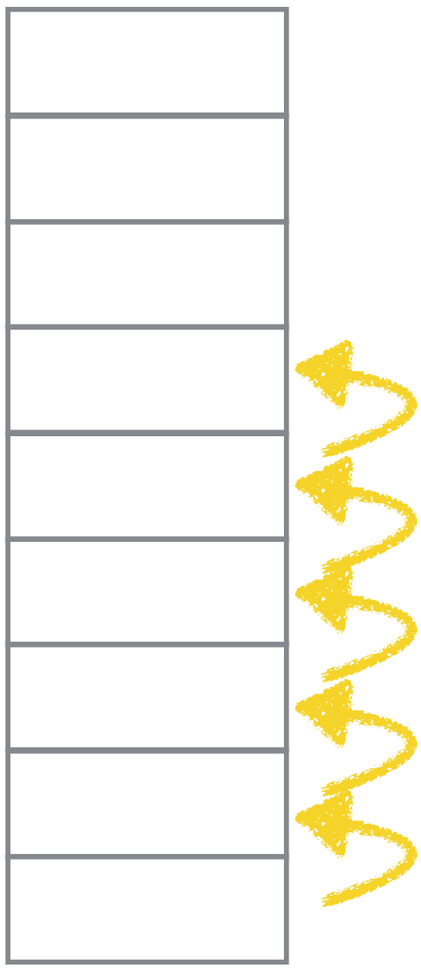
```
_start:
        mov   eax, 1    ; fibn
        mov   ebx, 1    ; fibn-1
        call _printDec
        call _println

        mov   ecx, 10-1 ; we printed 1, 9 more to go

for:    mov   edx, ebx
        mov   ebx, eax
        add   eax, edx
        call _printDec
        call _println
        loop for
```

getcopy fib.asm

Looping Through Arrays

LOOP
INSTRUCTION

Looping Through Arrays



INDIRECT
ADDRESSING
MODE