Smith College

Computer Science

CSC231-Assembly

Fall 2017 — Week #2

Dominique Thiébaut
dthiebaut@smith.edu

Outline

- Labs: emacs, assembly
- Assembly + Linking Process
- Object files
- DB directive: *Everything is a byte*!
- HexDump: we need to learn hexadecimal!
- Ascii Table

Emacs

Lab #1: Emacs

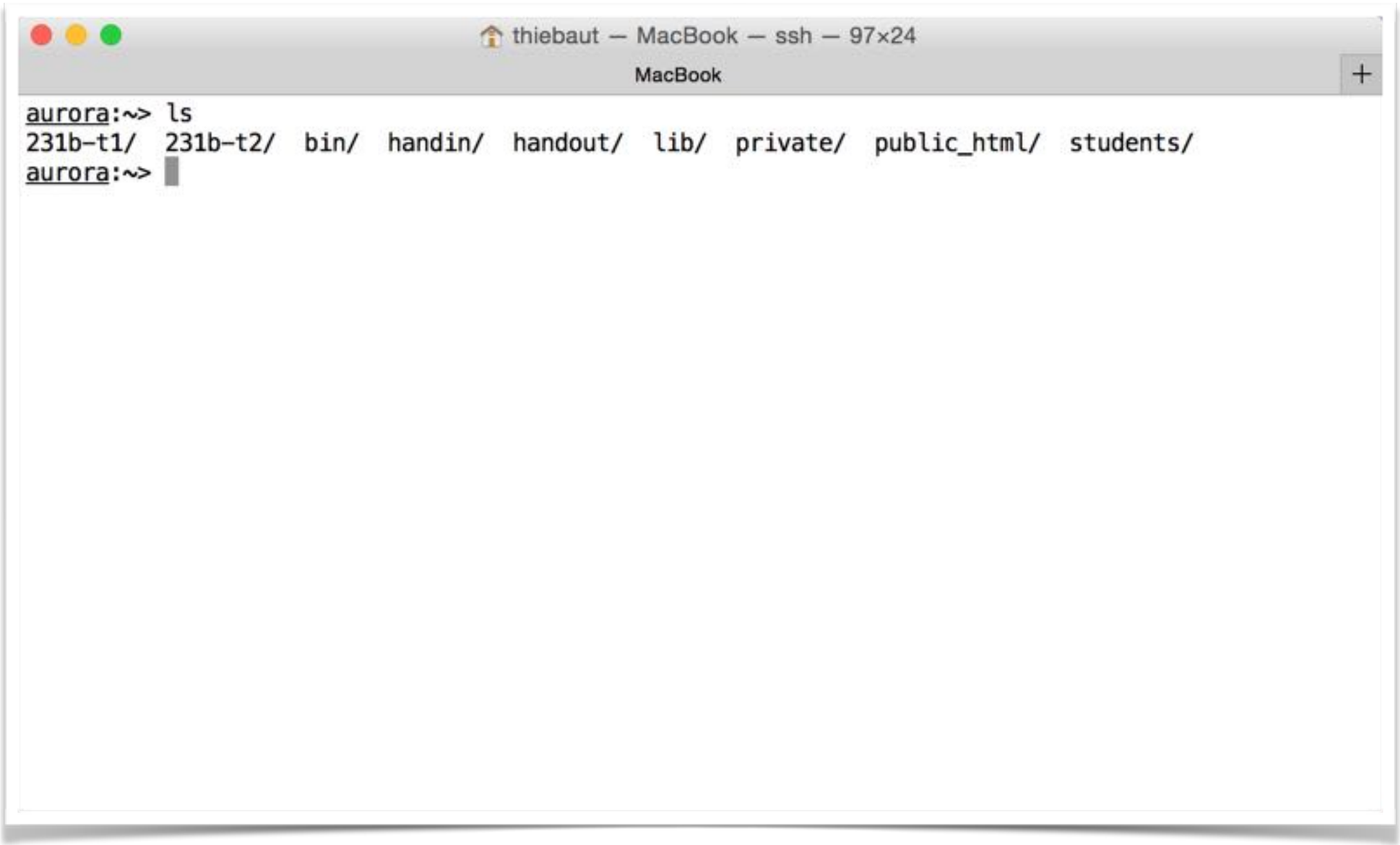
- ssh to aurora

ssh -Y cs231a-xx@aurora.smith.edu

- Emacs demo
- Basic commands

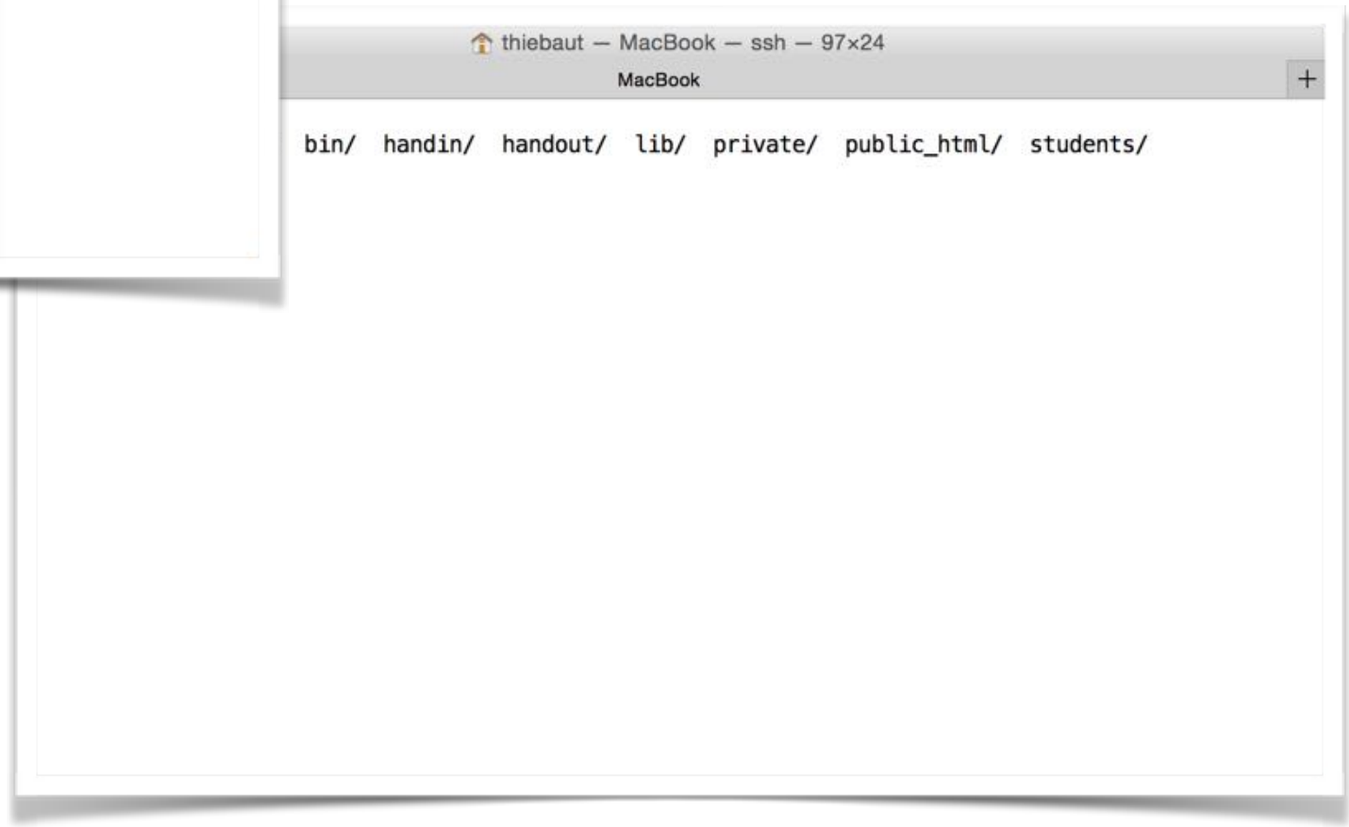
^X^C
^G
^D
^K
^Y
cursor keys

Shell


thiebaut — MacBook — ssh — 97×24
MacBook
aurora:~> ls
231b-t1/ 231b-t2/ bin/ handin/ handout/ lib/ private/ public_html/ students/
aurora:~>

Shell


/Applications/Utilities
Search
Favorites
thiebaut
Desktop
iCloud Drive
Dropbox
Applications
Utilities
Documents
Downloads
Box Sync
Devices
Remote Disc
Shared
raspberrypi
Tags
Exchange.app
Update.app
Built-in Keyboard Firmware Update.app
ColorSync Utility.app
Console.app
Digital Color Meter.app
Disk Utility.app
EyeCam.app


thiebaut — MacBook — ssh — 97×24
MacBook
bin/ handin/ handout/ lib/ private/ public_html/ students/

Setup Shell

```
aurora:~> setup231.sh
You must logout and log back in to get bash as your default shell
aurora:~> logout
Good Bye!
Connection to aurora.smith.edu closed.
```

Setup Shell

aurora:~> setup231.sh
You must logout and log back in to get bash as your default shell
aurora:~> logout
Good Bye!
Connection to aurora.smith.edu closed.

```
ssh -Y cs231a-by@aurora.smith.edu

cs231a-by@aurora.smith.edu's password:
Welcome to Linux Mint 17 Qiana (GNU/Linux 3.13.0-24-generic
x86_64)

Welcome to Linux Mint
 * Documentation:  http://www.linuxmint.com
Last login: Mon Sep 11 10:09:53 2017 from 131.229.104.254
cs231a-by@aurora ~ $
```

Setup Shell

```
ssh -Y cs231a-by@aurora.smith.edu
```

aurora:~> setup231.sh
You must logout and log back in to get bash as your default shell
aurora:~> logout
Good Bye!
Connection to aurora.smith.edu closed.

```
ssh -Y cs231a-by@aurora.smith.edu

cs231a-by@aurora.smith.edu's password:
Welcome to Linux Mint 17 Qiana (GNU/Linux 3.13.0-24-generic
x86_64)

Welcome to Linux Mint
 * Documentation:  http://www.linuxmint.com
Last login: Mon Sep 11 10:09:53 2017 from 131.229.104.254
cs231a-by@aurora ~ $
```

3 Useful Linux Commands

- **ls** (ell ess) list all files in a folder/**directory**
- **cp** copy file
- **cd** change directory

Linux Commands

command -switches file-name(s)

Linux Commands

command -switches file-name(s)

```
ls -l
ls *.asm
 ls *.o
ls hello*
  ls -l

 man ls
```

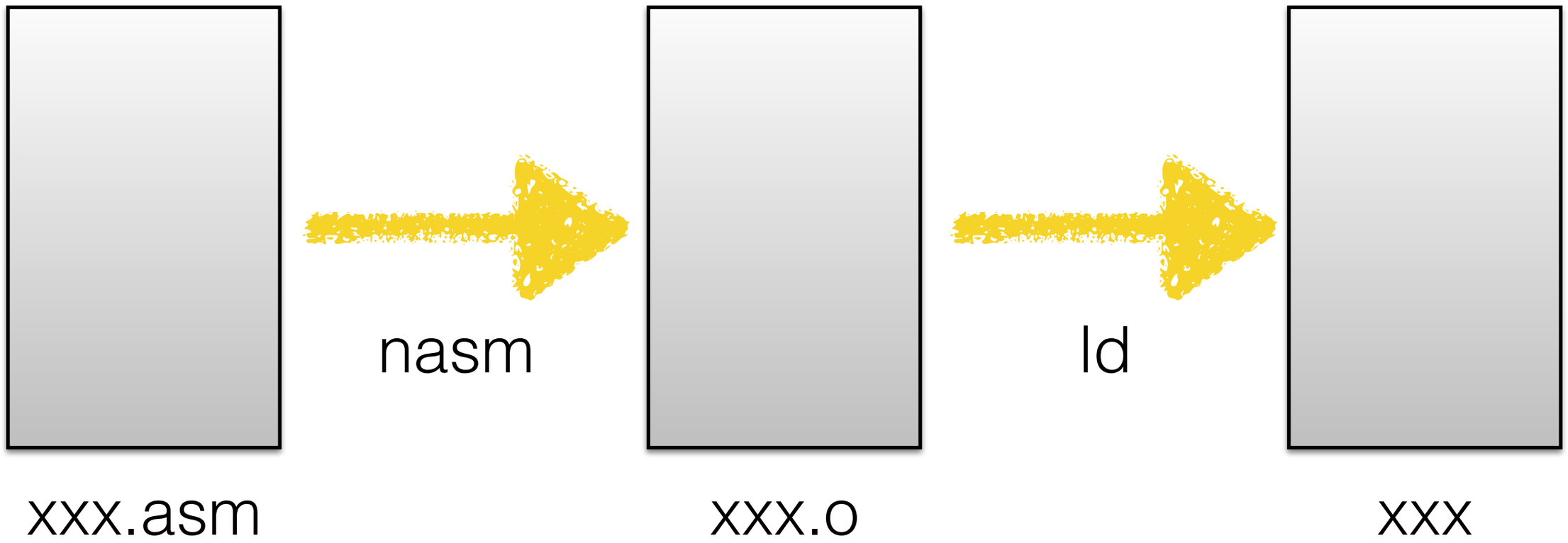
Skeleton

```
;;; program_name.asm
;;; your name
;;;
;;; a description of the program
;;;
;;; to assemble and run:
;;;
;;;     nasm -f elf -F  stabs program.asm
;;;     ld -melf_i386 -o program program.o
;;;     ./program
;;; ---------------------------------------------------------------------

;%include files here...



        ;; ---------------------------------------------------------------
        ;; data areas
        ;; ---------------------------------------------------------------

        section .data


        ;; ---------------------------------------------------------------
        ;; code area
        ;; ---------------------------------------------------------------

        section .text
        global  _start

_start:


        ;; (add your code here!!!!)



        ;; exit()

        mov     eax,1
        mov     ebx,0
        int     0x80            ; final system call
```

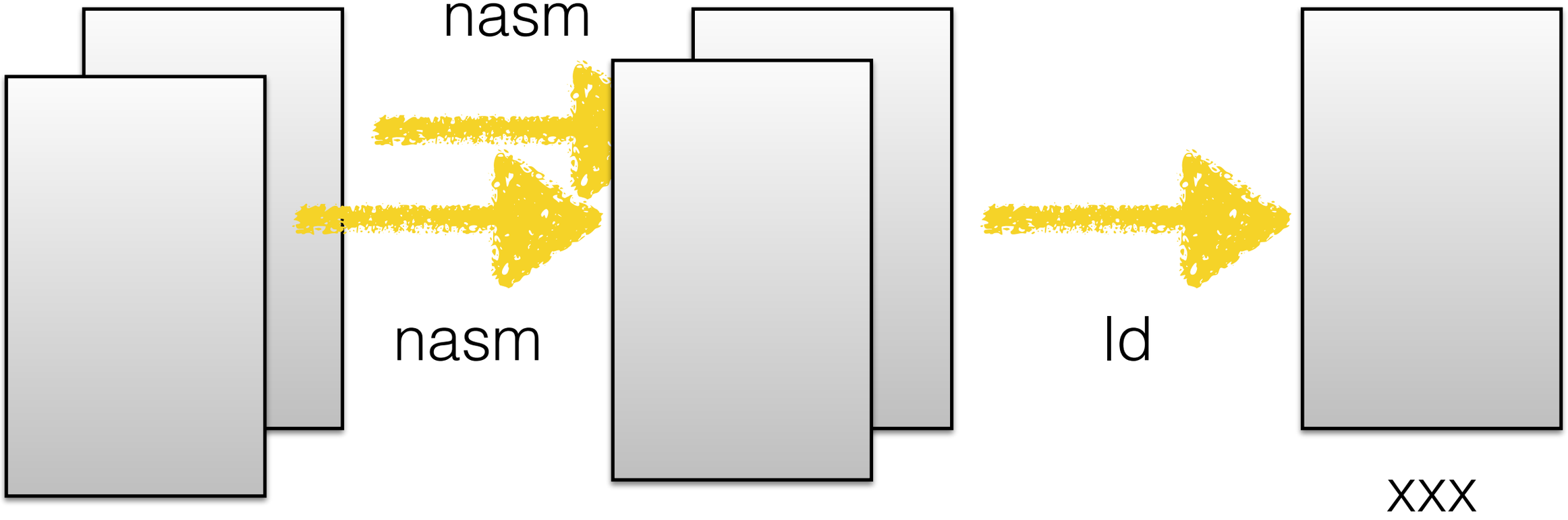
Hello World!

```
;;; helloWorld.asm
;;; D. Thiebaut
;;;
;;; Display "Hello there!" on the screen
;;;
;;; To assemble, link, and run:
;;;     nasm -f elf  helloWorld.asm
;;;     ld -melf_i386 -o helloWorld helloWorld.o
;;;     ./helloWorld
;;;

                section .data
Hello           db      "Hello there!"
HelloLen        equ     $-Hello

                section .text
                global  _start
_start:

;;; print message

                mov     eax, 4          ; write
                mov     ebx, 1          ; stdout
                mov     ecx, Hello      ; address of message to print
                mov     edx, HelloLen   ; # of chars to print
                int     0x80

;;; exit

                mov     ebx, 0
                mov     eax, 1
                int     0x80
```

Assembly+Linking



Assembly+Linking



DB: byte storage

```
section .data
Hello       db      "Hello there!"
HelloLen    equ     $-Hello
```

- A byte is…
- DB: "**D**efine **B**yte of storage"
  stores information in a byte format
  (could be more than 1 byte)
- **The X86 memory is a memory of bytes**

ASCII Table

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Source: www.LookupTables.com

Data Section vs Memory

```
message     db      "hello",10
```

Data Section vs Memory

```
db      "hello"
   db      10
```

Data Section vs Memory

```
db       "hel"
 db       "lo"
  db       10
```

Data Section vs Memory

```
db      104,"el"
  db      "lo"
   db      10
```

We stopped here last time…

Assembly  Python

```
            section .data
message      db      10, "hello world!"
             db      10, "************", 10, 10
msgLen       equ      $-message

            section .text
            global  _start

_start:
      mov eax, 4          ; print
      mov ebx, 1          ; to stdout
      mov ecx, message ; string
      mov edx, msgLen   ; # of chars

      int  0x80           ; ask Linux to print



;;;  exit()

      mov     eax,1
      mov     ebx,0
      int     0x80 ; final system call
```

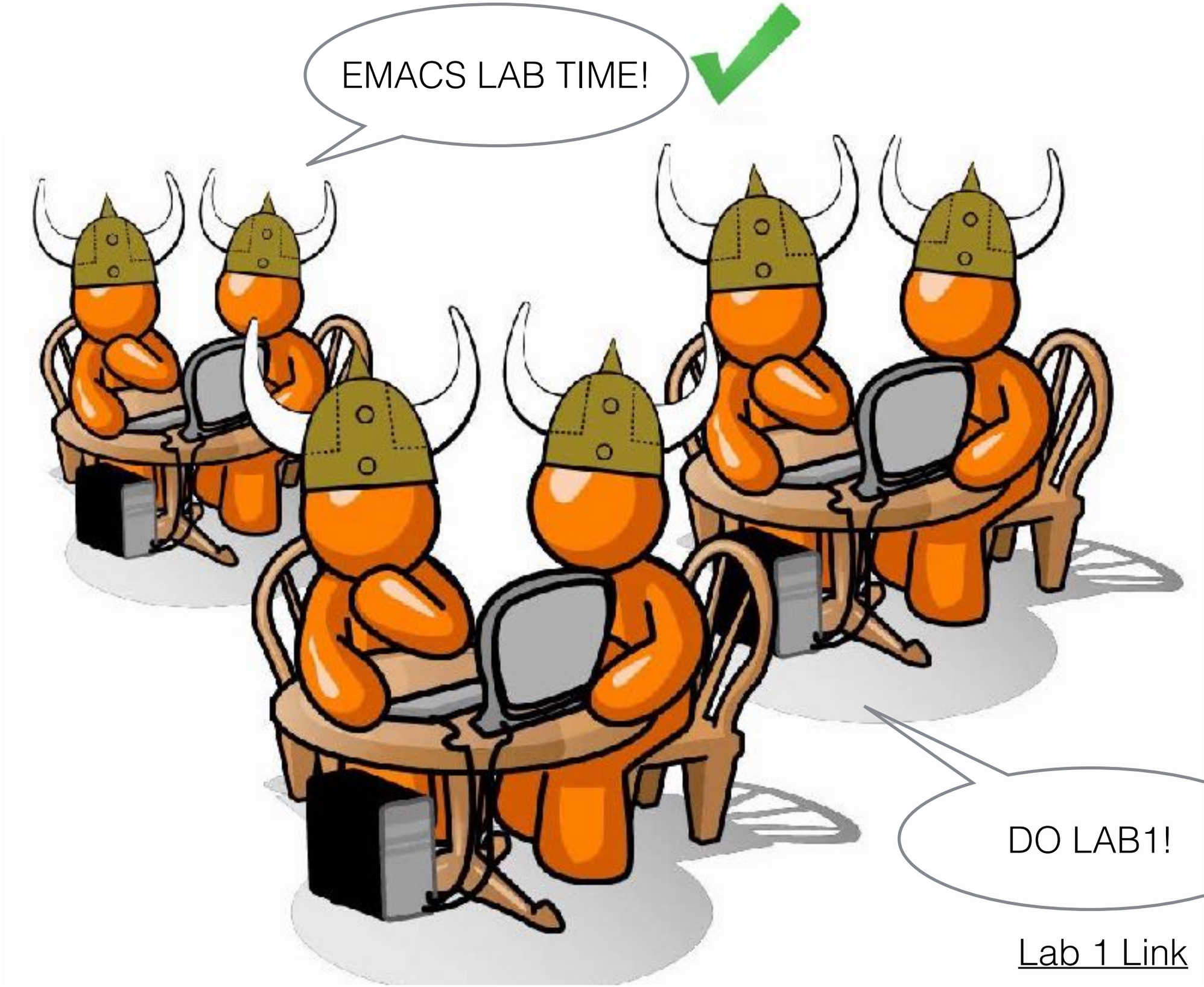
```
message = "\nhello world!\n************\n\n"




def main():
    print( message )



main()
```

Exercise

```
 section .data
db       "lo "
db       "hel"
     "world!", 10
  equ$-msg3
```

```
_start:
     mov  eax, 4      ; print
     mov  ebx, 1      ; to stdout
     mov  ecx, msg2   ; string
     mov  edx, 3
     int  0x80

     mov  eax, 4      ; print
     mov  ebx, 1      ; to stdout
     mov  ecx, msg1   ; string
     mov  edx, 3      ; # of chars
     int  0x80

     mov  eax, 4      ; print
     mov  ebx, 1      ; to stdout
     mov  ecx, msg3   ; string
     mov  edx, msgLen; # of chars
     int  0x80   ; ask Linux to print
```

What gets printed?



Lab 1 Link

https://i.ytimg.com/vi/ZV2HjJ1glLM/maxresdefault.jpg

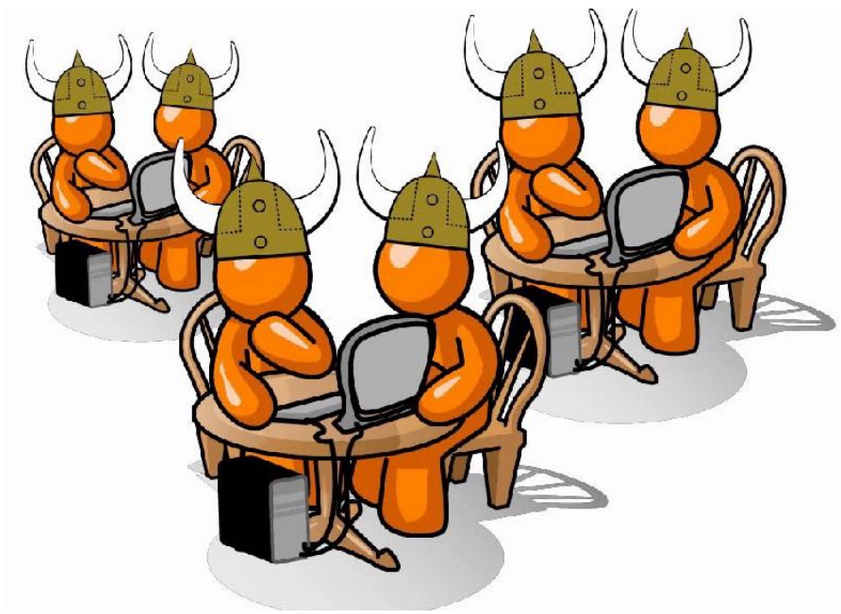
We stopped here last time…

Plan for Today

- Review Last Lab
- Hexdump
- Lab1b — Bash

Explore Solution Programs for Lab #1

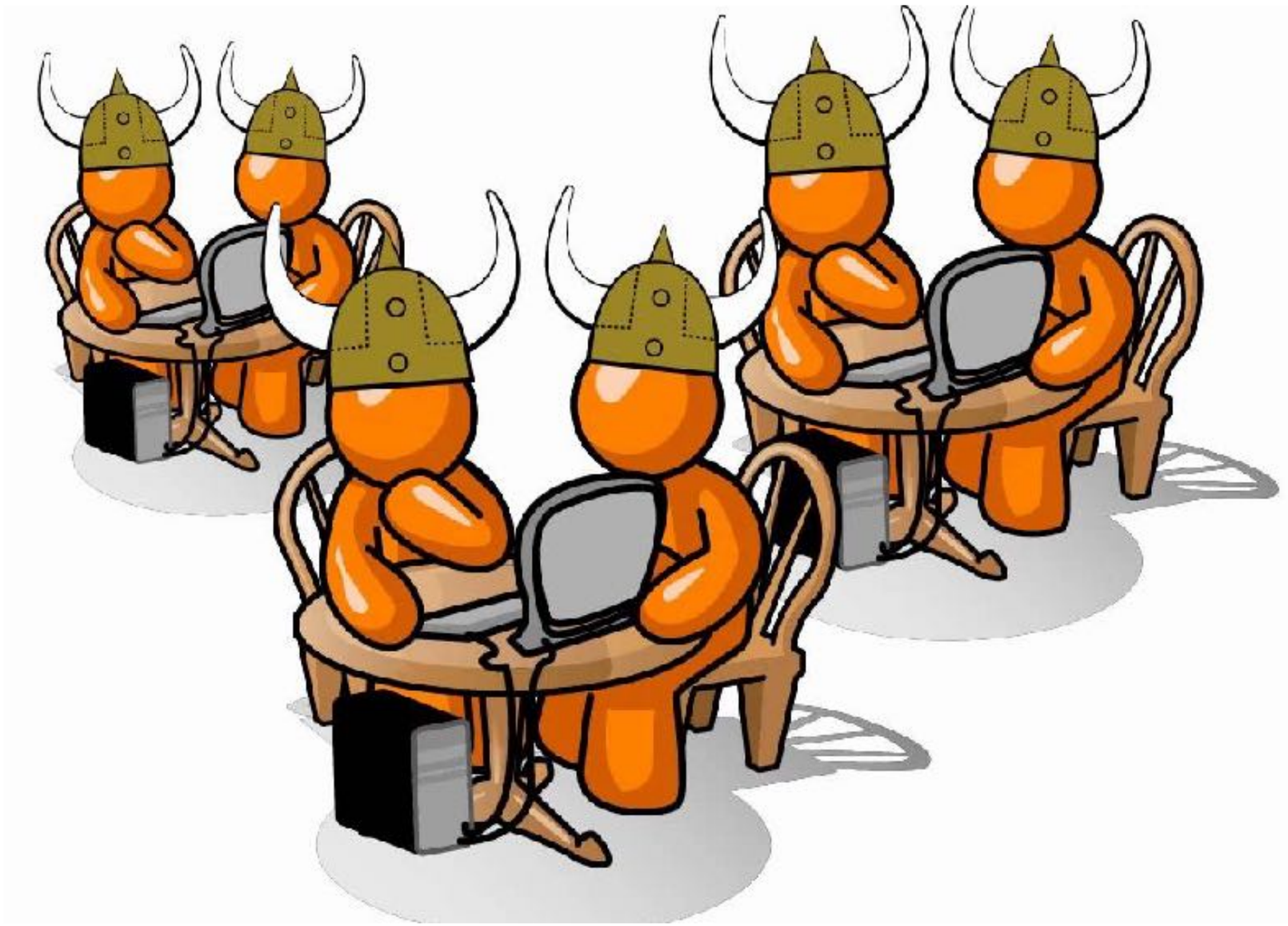
Lab 1 Link



Hexdump

```
hexdump -v -C helloWorld2

00000000  7f 45 4c 46 01 01 01 00  00 00 00 00 00 00 00 00  |.ELF............|
00000010  02 00 03 00 01 00 00 00  80 80 04 08 34 00 00 00  |............4...|
00000020  dc 00 00 00 00 00 00 00  34 00 20 00 02 00 28 00  |........4. ...(.|
00000030  06 00 03 00 01 00 00 00  00 00 00 00 00 80 04 08  |................|
00000040  00 80 04 08 a2 00 00 00  a2 00 00 00 05 00 00 00  |................|
00000050  00 10 00 00 01 00 00 00  a4 00 00 00 a4 90 04 08  |................|
00000060  a4 90 04 08 0e 00 00 00  0e 00 00 00 06 00 00 00  |................|
00000070  00 10 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000080  b8 04 00 00 00 bb 01 00  00 00 ba 0e 00 00 00 b9  |................|
00000090  a4 90 04 08 cd 80 b8 01  00 00 00 bb 00 00 00 00  |................|
000000a0  cd 80 00 00 48 65 6c 6c  6f 2c 20 57 6f 72 6c 64  |....Hello, World|
000000b0  21 0a 00 2e 73 79 6d 74  61 62 00 2e 73 74 72 74  |!...symtab..strt|
000000c0  61 62 00 2e 73 68 73 74  72 74 61 62 00 2e 74 65  |ab..shstrtab..te|
000000d0  78 74 00 2e 64 61 74 61  00 00 00 00 00 00 00 00  |xt..data........|
000000e0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
000000f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000100  00 00 00 00 1b 00 00 00  01 00 00 00 06 00 00 00  |................|
00000110  80 80 04 08 80 00 00 00  22 00 00 00 00 00 00 00  |........".......|
00000120  00 00 00 00 10 00 00 00  00 00 00 00 21 00 00 00  |............!...|
00000130  01 00 00 00 03 00 00 00  a4 90 04 08 a4 00 00 00  |................|
00000140  0e 00 00 00 00 00 00 00  00 00 00 00 04 00 00 00  |................|
00000150  00 00 00 00 11 00 00 00  03 00 00 00 00 00 00 00  |................|
00000160  00 00 00 00 b2 00 00 00  27 00 00 00 00 00 00 00  |........'.......|
00000170  00 00 00 00 01 00 00 00  00 00 00 00 01 00 00 00  |................|
00000180  02 00 00 00 00 00 00 00  00 00 00 00 cc 01 00 00  |................|
00000190  b0 00 00 00 05 00 00 00  07 00 00 00 04 00 00 00  |................|
000001a0  10 00 00 00 09 00 00 00  03 00 00 00 00 00 00 00  |................|
000001b0  00 00 00 00 7c 02 00 00  3c 00 00 00 00 00 00 00  |....|...<.......|
000001c0  00 00 00 00 01 00 00 00  00 00 00 00 00 00 00 00  |................|
000001d0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
000001e0  80 80 04 08 00 00 00 00  03 00 01 00 00 00 00 00  |................|
000001f0  a4 90 04 08 00 00 00 00  03 00 02 00 01 00 00 00  |................|
00000200  00 00 00 00 00 00 00 00  04 00 f1 ff 11 00 00 00  |................|
00000210  a4 90 04 08 00 00 00 00  00 00 02 00 16 00 00 00  |................|
00000220  0e 00 00 00 00 00 00 00  00 00 f1 ff 00 00 00 00  |................|
00000230  00 00 00 00 00 00 00 00  04 00 f1 ff 1d 00 00 00  |................|
00000240  80 80 04 08 00 00 00 00  10 00 01 00 24 00 00 00  |............$...|
00000250  b2 90 04 08 00 00 00 00  10 00 02 00 30 00 00 00  |............0...|
00000260  b2 90 04 08 00 00 00 00  10 00 02 00 37 00 00 00  |............7...|
00000270  b4 90 04 08 00 00 00 00  10 00 02 00 00 68 65 6c  |.............hel|
00000280  6c 6f 57 6f 72 6c 64 32  2e 61 73 6d 00 6d 65 73  |loWorld2.asm.mes|
00000290  67 00 4d 53 47 4c 45 4e  00 5f 73 74 61 72 74 00  |g.MSGLEN._start.|
000002a0  5f 5f 62 73 73 5f 73 74  61 72 74 00 5f 65 64 61  |__bss_start._eda|
000002b0  74 61 00 5f 65 6e 64 00                           |ta._end.|
000002b8
```

See link in Weekly Schedule page



Do Lab1b!