# Lab 7: ELF

## Task0a

1. ELF header, 0x80482e0
2. 33 / 34 (does NULL count?)
3. 18b
4. 080482e0 (\ text section (12)?)
5. 08048388 (\ text section (12)?)
6. ?

* typeof main is func
* text offset is 02e0, and its' size is 1b8

## Task1c

**Note that, depending on the chosen unit size, the printed hexadecimal values may differ in order when compared with the output of *hexedit*. Why is that?**

In our program we print out a number, and in hexedit it is printed according to little endian

## Task2a

2. _start
3. 8048080. entry point is at 804808a

```
  10: 08049166     0 NOTYPE  LOCAL  DEFAULT    2 d1
  11: 0804916c     0 NOTYPE  LOCAL  DEFAULT    2 d2
  12: 08049175     0 NOTYPE  LOCAL  DEFAULT    2 dr
  13: 0804917c     0 NOTYPE  LOCAL  DEFAULT    2 d3
  14: 08049180     0 NOTYPE  LOCAL  DEFAULT    2 happy
  15: 0804809c     0 NOTYPE  LOCAL  DEFAULT    1 loop1
  16: 080480c5     0 NOTYPE  LOCAL  DEFAULT    1 loop1.continue
  17: 080480de     0 NOTYPE  LOCAL  DEFAULT    1 loop1.end
  18: 08048080     0 NOTYPE  GLOBAL DEFAULT    1 _start
  19: 08049185     0 NOTYPE  GLOBAL DEFAULT  ABS __bss_start
  20: 08049185     0 NOTYPE  GLOBAL DEFAULT  ABS _edata
  21: 08049198     0 NOTYPE  GLOBAL DEFAULT  ABS _end
```

Figure 1-3: ELF Header

```
#define EI_NIDENT       16

typedef struct {
        unsigned char   e_ident[EI_NIDENT];
        Elf32_Half      e_type;
        Elf32_Half      e_machine;
        Elf32_Word      e_version;
        Elf32_Addr      e_entry;
        Elf32_Off       e_phoff;
        Elf32_Off       e_shoff;
        Elf32_Word      e_flags;
        Elf32_Half      e_ehsize;
        Elf32_Half      e_phentsize;
        Elf32_Half      e_phnum;
        Elf32_Half      e_shentsize;
        Elf32_Half      e_shnum;
        Elf32_Half      e_shstrndx;
} Elf32_Ehdr;
```

## Task2b

***What are the values of location/length? How do you know that?***

24 (decimal) = 18 (hexa)location = 18

according to the struct
ident starts at 0 (16 bytes)
type starts at 16 (2 bytes)
machine starts at 18 (2 bytes)
version starts at 20 (4 bytes)
=> entry starts at 24 (18 hexa) and it is an address so it is 4 bytes.

**Figure 1-2: 32-Bit Data Types**

| Name | Size | Alignment | Purpose |
|---|---|---|---|
| Elf32_Addr | 4 | 4 | Unsigned program address |
| Elf32_Half | 2 | 2 | Unsigned medium integer |
| Elf32_Off | 4 | 4 | Unsigned file offset |
| Elf32_Sword | 4 | 4 | Signed large integer |
| Elf32_Word | 4 | 4 | Unsigned large integer |
| unsigned char | 1 | 1 | Unsigned small integer |

```
8-Quit
7
Please enter <location> <val>
18 08048080
Location: 18, Val: 8048080
Unit size: 4, File name: chezi, Mem count: 0
Choose action:
0-Toggle Debug Mode
1-Set File Name
2-Set Unit Size
3-Load Into Memory
4-Toggle Display Mode
5-Memory Display
6-Save Into File
7-Memory Modify
8-Quit
6
Please enter <source-address> <target-location> <length>
18 18 1
Unit size: 4, File name: chezi, Mem count: 0
Choose action:
0-Toggle Debug Mode
1-Set File Name
2-Set Unit Size
3-Load Into Memory
4-Toggle Display Mode
5-Memory Display
6-Save Into File
7-Memory Modify
8-Quit
8
quitting
shira@shira-Inspiron-5379:~/archi/labs/lab7/task2/task2b$ chmod +x chezi
shira@shira-Inspiron-5379:~/archi/labs/lab7/task2/task2b$ ./chezi
Answer to Life, the Universe, and Everything
42
shira@shira-Inspiron-5379:~/archi/labs/lab7/task2/task2b$ readelf -a chezi| less
```

length = 1 (assumint unit_size = 4)Task3a
Entry point for main: 08048464
Size: 175 (decimal)
.text offset: 3b0
.text size: 20c
.text address: 080483b0

main offset within the .text: 08048464-080483b0 = b4 (180 decimal)
main offset within the file: 08048464-080483b0+3b0 = 464 (1162 decimal)

Task3b
https://c9x.me/x86/html/file_module_x86_id_270.html

```
gcc -m32 -g -Wall -o hexeditplus hexeditplus.o
shira@shira-Inspiron-5379:~/archi/labs/lab7/task3/task3a$ objdump -d abc

abc:     file format elf32-i386


Disassembly of section .init:

0804830c <_init>:
 804830c:       55                      push   %ebp
 804830d:       89 e5                   mov    %esp,%ebp
 804830f:       53                      push   %ebx
 8048310:       83 ec 04                sub    $0x4,%esp
 8048313:       e8 00 00 00 00          call   8048318 <_init+0xc>
 8048318:       5b                      pop    %ebx
 8048319:       81 c3 dc 1c 00 00       add    $0x1cdc,%ebx
 804831f:       8b 93 fc ff ff ff       mov    -0x4(%ebx),%edx
 8048325:       85 d2                   test   %edx,%edx
 8048327:       74 05                   je     804832e <_init+0x22>
 8048329:       e8 1e 00 00 00          call   804834c <__gmon_start__@plt>
 804832e:       e8 0d 01 00 00          call   8048440 <frame_dummy>
 8048333:       e8 58 02 00 00          call   8048590 <__do_global_ctors_aux>
 8048338:       58                      pop    %eax
 8048339:       5b                      pop    %ebx
 804833a:       c9                      leave
 804833b:       c3                      ret

Disassembly of section .plt:

0804833c <.plt>:
 804833c:       ff 35 f8 9f 04 08       pushl  0x8049ff8
 8048342:       ff 25 fc 9f 04 08       jmp    *0x8049ffc
 8048348:       00 00                   add    %al,(%eax)
        ...

0804834c <__gmon_start__@plt>:
 804834c:       ff 25 00 a0 04 08       jmp    *0x804a000
 8048352:       68 00 00 00 00          push   $0x0
 8048357:       e9 e0 ff ff ff          jmp    804833c <.plt>

0804835c <__libc_start_main@plt>:
 804835c:       ff 25 04 a0 04 08       jmp    *0x804a004
 8048362:       68 08 00 00 00          push   $0x8
 8048367:       e9 d0 ff ff ff          jmp    804833c <.plt>

0804836c <fopen@plt>:
 804836c:       ff 25 08 a0 04 08       jmp    *0x804a008
 8048372:       68 10 00 00 00          push   $0x10
 8048377:       e9 c0 ff ff ff          jmp    804833c <.plt>

0804837c <fgetc@plt>:
 804837c:       ff 25 0c a0 04 08       jmp    *0x804a00c
```

## Task 4

The problem with ntsc is that it only counts digits 1-8 (0 and 9 are not counted)

## ntsc

Entry point address: 0x410

| NUM | Value | Size | Type | Bind | Vis | Ndx | Name |
|---|---|---|---|---|---|---|---|
| 68: | 00000577 | 1136 | FUNC | GLOBAL | DEFAULT | 14 (.text) | digit_cnt |
| 69: | 000004ed | 67 | FUNC | GLOBAL | DEFAULT | 14 | digit_cnt |

| [Nr] | Name | Type | Addr | Off | Size | ES | Flg | Lk | Inf | Al |
|---|---|---|---|---|---|---|---|---|---|---|
| [14] | .text | PROGBITS | 00000410 | 000410 | 0006b2 | 00 | AX | 0 | 0 | 16 |
| [14] | .text | PROGBITS | 000003b0 | 0003b0 | 000212 | 00 | AX | 0 | 0 | 16q |

digit_cnt offset within the .text: 410- 00000410 = 0
digit_cnt offset within the file: 000410

task4 code

* The limitation is that our code zise must be <= than 1136

```
Choose action:...
0
Debug flag now on
Unit size: 1, File name: , Mem count: 0
Choose action:...
8-Quit
1
Please enter the file name: task4
Unit size: 1, File name: task4, Mem count: 0
Choose action:...
3
Please enter <location> <length>
4ed 67
File name: task4
Location: 4ED
Length: 67
Loaded 67 units into memory
Unit size: 1, File name: task4, Mem count: 0
Choose action:...
1
Please enter the file name: ntsc
Unit size: 1, File name: ntsc, Mem count: 0
Choose action:...
6
Please enter <source-address> <target-location> <length>
0 577 67
Unit size: 1, File name: ntsc, Mem count: 0
Choose action:...
8
quitting

shira@shira-Inspiron-5379:~/archi/labs/lab7/task4$ ./ntsc 09
The number of digits in the string is: 2
```