

```
dherve@cslinux:~/cscd340/Lab 3$ readelf -h
./a.out
```

ELF Header:

Magic: 7f 45 4c 46 02 01 01 00 00 00 00 00 00 00 00 00 00 00 00 00

Class: ELF64

Data: 2's complement, little  
endian

Version: 1 (current)

OS/ABI: UNIX - System V

ABI Version: 0

Type: EXEC (Executable file)

Machine: Advanced Micro  
Devices X86-64

Version: 0x1

Entry point address: 0x400580

Start of program headers: 64 (bytes into  
file)

Start of section headers: 4504 (bytes into  
file)

Flags: 0x0

Size of this header: 64 (bytes)

Size of program headers: 56 (bytes)

Number of program headers: 9

Size of section headers: 64 (bytes)

Number of section headers: 30

Section header string table index: 27

← The header portion of readelf gives info on  
the size and number of headers

```
dherve@cslinux:~/cscd340/Lab 3$ readelf -l
./a.out
```

Elf file type is EXEC (Executable file)

Entry point 0x400580

There are 9 program headers, starting at offset 64

Program Headers:

Type	Offset	VirtAddr	PhysAddr	FileSiz	MemSiz	Flags	Align
PHDR	0x0000000000000040	0x0000000000000040	0x0000000000000040	0x00000000000001f8	0x00000000000001f8	R E 8	
INTERP	0x0000000000000238	0x0000000000000238	0x0000000000000238	0x000000000000001c	0x000000000000001c	R 1	
[Requesting program interpreter: /lib64/ld-linux-x86-64.so.2]							
LOAD	0x0000000000000000	0x0000000000000000	0x0000000000000000	0x000000000000093c	0x000000000000093c	R E 200000	
LOAD	0x0000000000000e28	0x0000000000000e28	0x0000000000000e28	0x0000000000000218	0x0000000000000230	RW 200000	
DYNAMIC	0x0000000000000e50	0x0000000000000e50	0x0000000000000e50	0x0000000000000190	0x0000000000000190	RW 8	

← The FileSiz and MemSiz give info on how big the file is and how much memory it's taking up

NOTE 0x0000000000000254  
0x00000000000400254 0x00000000000400254

0x0000000000000044  
0x0000000000000044 R 4

GNU\_EH\_FRAME 0x0000000000000868  
0x00000000000400868 0x00000000000400868

0x000000000000002c  
0x000000000000002c R 4

GNU\_STACK 0x0000000000000000  
0x0000000000000000 0x0000000000000000

0x0000000000000000  
0x0000000000000000 RW 8

GNU\_RELRO 0x0000000000000e28  
0x00000000000600e28 0x00000000000600e28

0x00000000000001d8  
0x00000000000001d8 R 1

#### Section to Segment mapping:

Segment Sections...

00

01 .interp

02 .interp .note.ABI-tag .note.gnu.build-id  
.gnu.hash .dynsym .dynstr .gnu.version  
.gnu.version\_r .rela.dyn .rela.plt .init .plt .text  
.fini .rodata .eh\_frame\_hdr .eh\_frame

03 .ctors .dtors .jcr .dynamic .got .got.plt  
.data .bss

04 .dynamic

05 .note.ABI-tag .note.gnu.build-id

06 .eh\_frame\_hdr

07

08 .ctors .dtors .jcr .dynamic .got