Δομημένος Προγραμματισμός

## Φοιτητής: Βασιλείου Παύλος

## ΑΕΜ: 8783

**Άσκηση 2**

**Ερώτημα 1**

**objdump:** Δείχνει πληροφορίες ενός ή παραπάνω object files. Επίσης μπορεί να λειτουργήσει και ως disassembler δηλαδή να δείξει το κώδικα ενός εκτελέσιμου (executable).

**readelf:** Διαβάζει πληροφορίες των object files τύπου ELF(Executable Linkable Format). Δείχνει παρόμοιες πληροφορίες με την objdump αλλά με περισσότερες λεπτομέρειες. Επίσης, σε αντίθεση με το objdump δεν χρησιμοποιεί την βιβλιοθήκη BFD(Binary File Descriptor) οπότε και τα όποια λάθη της δεν την επηρεάζουν.

**Ερώτημα 2**

Με την προϋπόθεση ότι έχω κάνει navigate στον φάκελο όπου βρίσκεται το target file (hello), με χρήση της εντολής cd στο terminal εκτελώ τις παρακάτω εντολές με τα αποτελέσματα τις καθεμιάς να φαίνεται παρακάτω:

**1.**

**Εντολή:** readelf -h hello

**Αποτέλεσμα:** UNIX - System V

**2.**

**Εντολή:** readelf --sections hello

**Αποτέλεσμα:**

There are 30 section headers, starting at offset 0x1198:

Section Headers:

[Nr] Name Type Address Offset

Size EntSize Flags Link Info Align

[ 0] NULL 0000000000000000 00000000

0000000000000000 0000000000000000 0 0 0

[ 1] .interp PROGBITS 0000000000400238 00000238

000000000000001c 0000000000000000 A 0 0 1

[ 2] .note.ABI-tag NOTE 0000000000400254 00000254

0000000000000020 0000000000000000 A 0 0 4

[ 3] .note.gnu.build-i NOTE 0000000000400274 00000274

0000000000000024 0000000000000000 A 0 0 4

[ 4] .gnu.hash GNU\_HASH 0000000000400298 00000298

000000000000001c 0000000000000000 A 5 0 8

[ 5] .dynsym DYNSYM 00000000004002b8 000002b8

0000000000000060 0000000000000018 A 6 1 8

[ 6] .dynstr STRTAB 0000000000400318 00000318

000000000000003d 0000000000000000 A 0 0 1

[ 7] .gnu.version VERSYM 0000000000400356 00000356

0000000000000008 0000000000000002 A 5 0 2

[ 8] .gnu.version\_r VERNEED 0000000000400360 00000360

0000000000000020 0000000000000000 A 6 1 8

[ 9] .rela.dyn RELA 0000000000400380 00000380

0000000000000018 0000000000000018 A 5 0 8

[10] .rela.plt RELA 0000000000400398 00000398

0000000000000048 0000000000000018 A 5 12 8

[11] .init PROGBITS 00000000004003e0 000003e0

000000000000001a 0000000000000000 AX 0 0 4

[12] .plt PROGBITS 0000000000400400 00000400

0000000000000040 0000000000000010 AX 0 0 16

[13] .text PROGBITS 0000000000400440 00000440

0000000000000172 0000000000000000 AX 0 0 16

[14] .fini PROGBITS 00000000004005b4 000005b4

0000000000000009 0000000000000000 AX 0 0 4

[15] .rodata PROGBITS 00000000004005c0 000005c0

0000000000000010 0000000000000000 A 0 0 4

[16] .eh\_frame\_hdr PROGBITS 00000000004005d0 000005d0

0000000000000034 0000000000000000 A 0 0 4

[17] .eh\_frame PROGBITS 0000000000400608 00000608

00000000000000f4 0000000000000000 A 0 0 8

[18] .init\_array INIT\_ARRAY 0000000000600e10 00000e10

0000000000000008 0000000000000000 WA 0 0 8

[19] .fini\_array FINI\_ARRAY 0000000000600e18 00000e18

0000000000000008 0000000000000000 WA 0 0 8

[20] .jcr PROGBITS 0000000000600e20 00000e20

0000000000000008 0000000000000000 WA 0 0 8

[21] .dynamic DYNAMIC 0000000000600e28 00000e28

00000000000001d0 0000000000000010 WA 6 0 8

[22] .got PROGBITS 0000000000600ff8 00000ff8

0000000000000008 0000000000000008 WA 0 0 8

[23] .got.plt PROGBITS 0000000000601000 00001000

0000000000000030 0000000000000008 WA 0 0 8

[24] .data PROGBITS 0000000000601030 00001030

0000000000000010 0000000000000000 WA 0 0 8

[25] .bss NOBITS 0000000000601040 00001040

0000000000000008 0000000000000000 WA 0 0 1

[26] .comment PROGBITS 0000000000000000 00001040

000000000000004d 0000000000000001 MS 0 0 1

[27] .shstrtab STRTAB 0000000000000000 0000108d

0000000000000108 0000000000000000 0 0 1

[28] .symtab SYMTAB 0000000000000000 00001918

0000000000000618 0000000000000018 29 45 8

[29] .strtab STRTAB 0000000000000000 00001f30

0000000000000237 0000000000000000 0 0 1

**3.**

**Εντολή:** objdump -d hello

**Αποτέλεσμα:**

file format elf64-x86-64

Disassembly of section .init:

00000000004003e0 <\_init>:

4003e0: 48 83 ec 08 sub $0x8,%rsp

4003e4: 48 8b 05 0d 0c 20 00 mov 0x200c0d(%rip),%rax # 600ff8 <\_DYNAMIC+0x1d0>

4003eb: 48 85 c0 test %rax,%rax

4003ee: 74 05 je 4003f5 <\_init+0x15>

4003f0: e8 3b 00 00 00 callq 400430 <\_\_gmon\_start\_\_@plt>

4003f5: 48 83 c4 08 add $0x8,%rsp

4003f9: c3 retq

Disassembly of section .plt:

0000000000400400 <puts@plt-0x10>:

400400: ff 35 02 0c 20 00 pushq 0x200c02(%rip) # 601008 <\_GLOBAL\_OFFSET\_TABLE\_+0x8>

400406: ff 25 04 0c 20 00 jmpq \*0x200c04(%rip) # 601010 <\_GLOBAL\_OFFSET\_TABLE\_+0x10>

40040c: 0f 1f 40 00 nopl 0x0(%rax)

0000000000400410 <puts@plt>:

400410: ff 25 02 0c 20 00 jmpq \*0x200c02(%rip) # 601018 <\_GLOBAL\_OFFSET\_TABLE\_+0x18>

400416: 68 00 00 00 00 pushq $0x0

40041b: e9 e0 ff ff ff jmpq 400400 <\_init+0x20>

0000000000400420 <\_\_libc\_start\_main@plt>:

400420: ff 25 fa 0b 20 00 jmpq \*0x200bfa(%rip) # 601020 <\_GLOBAL\_OFFSET\_TABLE\_+0x20>

400426: 68 01 00 00 00 pushq $0x1

40042b: e9 d0 ff ff ff jmpq 400400 <\_init+0x20>

0000000000400430 <\_\_gmon\_start\_\_@plt>:

400430: ff 25 f2 0b 20 00 jmpq \*0x200bf2(%rip) # 601028 <\_GLOBAL\_OFFSET\_TABLE\_+0x28>

400436: 68 02 00 00 00 pushq $0x2

40043b: e9 c0 ff ff ff jmpq 400400 <\_init+0x20>

Disassembly of section .text:

0000000000400440 <\_start>:

400440: 31 ed xor %ebp,%ebp

400442: 49 89 d1 mov %rdx,%r9

400445: 5e pop %rsi

400446: 48 89 e2 mov %rsp,%rdx

400449: 48 83 e4 f0 and $0xfffffffffffffff0,%rsp

40044d: 50 push %rax

40044e: 54 push %rsp

40044f: 49 c7 c0 b0 05 40 00 mov $0x4005b0,%r8

400456: 48 c7 c1 40 05 40 00 mov $0x400540,%rcx

40045d: 48 c7 c7 2d 05 40 00 mov $0x40052d,%rdi

400464: e8 b7 ff ff ff callq 400420 <\_\_libc\_start\_main@plt>

400469: f4 hlt

40046a: 66 0f 1f 44 00 00 nopw 0x0(%rax,%rax,1)

0000000000400470 <deregister\_tm\_clones>:

400470: b8 47 10 60 00 mov $0x601047,%eax

400475: 55 push %rbp

400476: 48 2d 40 10 60 00 sub $0x601040,%rax

40047c: 48 83 f8 0e cmp $0xe,%rax

400480: 48 89 e5 mov %rsp,%rbp

400483: 77 02 ja 400487 <deregister\_tm\_clones+0x17>

400485: 5d pop %rbp

400486: c3 retq

400487: b8 00 00 00 00 mov $0x0,%eax

40048c: 48 85 c0 test %rax,%rax

40048f: 74 f4 je 400485 <deregister\_tm\_clones+0x15>

400491: 5d pop %rbp

400492: bf 40 10 60 00 mov $0x601040,%edi

400497: ff e0 jmpq \*%rax

400499: 0f 1f 80 00 00 00 00 nopl 0x0(%rax)

00000000004004a0 <register\_tm\_clones>:

4004a0: b8 40 10 60 00 mov $0x601040,%eax

4004a5: 55 push %rbp

4004a6: 48 2d 40 10 60 00 sub $0x601040,%rax

4004ac: 48 c1 f8 03 sar $0x3,%rax

4004b0: 48 89 e5 mov %rsp,%rbp

4004b3: 48 89 c2 mov %rax,%rdx

4004b6: 48 c1 ea 3f shr $0x3f,%rdx

4004ba: 48 01 d0 add %rdx,%rax

4004bd: 48 d1 f8 sar %rax

4004c0: 75 02 jne 4004c4 <register\_tm\_clones+0x24>

4004c2: 5d pop %rbp

4004c3: c3 retq

4004c4: ba 00 00 00 00 mov $0x0,%edx

4004c9: 48 85 d2 test %rdx,%rdx

4004cc: 74 f4 je 4004c2 <register\_tm\_clones+0x22>

4004ce: 5d pop %rbp

4004cf: 48 89 c6 mov %rax,%rsi

4004d2: bf 40 10 60 00 mov $0x601040,%edi

4004d7: ff e2 jmpq \*%rdx

4004d9: 0f 1f 80 00 00 00 00 nopl 0x0(%rax)

00000000004004e0 <\_\_do\_global\_dtors\_aux>:

4004e0: 80 3d 59 0b 20 00 00 cmpb $0x0,0x200b59(%rip) # 601040 <\_\_TMC\_END\_\_>

4004e7: 75 11 jne 4004fa <\_\_do\_global\_dtors\_aux+0x1a>

4004e9: 55 push %rbp

4004ea: 48 89 e5 mov %rsp,%rbp

4004ed: e8 7e ff ff ff callq 400470 <deregister\_tm\_clones>

4004f2: 5d pop %rbp

4004f3: c6 05 46 0b 20 00 01 movb $0x1,0x200b46(%rip) # 601040 <\_\_TMC\_END\_\_>

4004fa: f3 c3 repz retq

4004fc: 0f 1f 40 00 nopl 0x0(%rax)

0000000000400500 <frame\_dummy>:

400500: 48 83 3d 18 09 20 00 cmpq $0x0,0x200918(%rip) # 600e20 <\_\_JCR\_END\_\_>

400507: 00

400508: 74 1e je 400528 <frame\_dummy+0x28>

40050a: b8 00 00 00 00 mov $0x0,%eax

40050f: 48 85 c0 test %rax,%rax

400512: 74 14 je 400528 <frame\_dummy+0x28>

400514: 55 push %rbp

400515: bf 20 0e 60 00 mov $0x600e20,%edi

40051a: 48 89 e5 mov %rsp,%rbp

40051d: ff d0 callq \*%rax

40051f: 5d pop %rbp

400520: e9 7b ff ff ff jmpq 4004a0 <register\_tm\_clones>

400525: 0f 1f 00 nopl (%rax)

400528: e9 73 ff ff ff jmpq 4004a0 <register\_tm\_clones>

000000000040052d <main>:

40052d: 55 push %rbp

40052e: 48 89 e5 mov %rsp,%rbp

400531: bf c4 05 40 00 mov $0x4005c4,%edi

400536: e8 d5 fe ff ff callq 400410 <puts@plt>

40053b: 5d pop %rbp

40053c: c3 retq

40053d: 0f 1f 00 nopl (%rax)

0000000000400540 <\_\_libc\_csu\_init>:

400540: 41 57 push %r15

400542: 41 89 ff mov %edi,%r15d

400545: 41 56 push %r14

400547: 49 89 f6 mov %rsi,%r14

40054a: 41 55 push %r13

40054c: 49 89 d5 mov %rdx,%r13

40054f: 41 54 push %r12

400551: 4c 8d 25 b8 08 20 00 lea 0x2008b8(%rip),%r12 # 600e10 <\_\_frame\_dummy\_init\_array\_entry>

400558: 55 push %rbp

400559: 48 8d 2d b8 08 20 00 lea 0x2008b8(%rip),%rbp # 600e18 <\_\_init\_array\_end>

400560: 53 push %rbx

400561: 4c 29 e5 sub %r12,%rbp

400564: 31 db xor %ebx,%ebx

400566: 48 c1 fd 03 sar $0x3,%rbp

40056a: 48 83 ec 08 sub $0x8,%rsp

40056e: e8 6d fe ff ff callq 4003e0 <\_init>

400573: 48 85 ed test %rbp,%rbp

400576: 74 1e je 400596 <\_\_libc\_csu\_init+0x56>

400578: 0f 1f 84 00 00 00 00 nopl 0x0(%rax,%rax,1)

40057f: 00

400580: 4c 89 ea mov %r13,%rdx

400583: 4c 89 f6 mov %r14,%rsi

400586: 44 89 ff mov %r15d,%edi

400589: 41 ff 14 dc callq \*(%r12,%rbx,8)

40058d: 48 83 c3 01 add $0x1,%rbx

400591: 48 39 eb cmp %rbp,%rbx

400594: 75 ea jne 400580 <\_\_libc\_csu\_init+0x40>

400596: 48 83 c4 08 add $0x8,%rsp

40059a: 5b pop %rbx

40059b: 5d pop %rbp

40059c: 41 5c pop %r12

40059e: 41 5d pop %r13

4005a0: 41 5e pop %r14

4005a2: 41 5f pop %r15

4005a4: c3 retq

4005a5: 66 66 2e 0f 1f 84 00 data16 nopw %cs:0x0(%rax,%rax,1)

4005ac: 00 00 00 00

00000000004005b0 <\_\_libc\_csu\_fini>:

4005b0: f3 c3 repz retq

Disassembly of section .fini:

00000000004005b4 <\_fini>:

4005b4: 48 83 ec 08 sub $0x8,%rsp

4005b8: 48 83 c4 08 add $0x8,%rsp

4005bc: c3 retq

**4.**

**Εντολή:** objdump -s -j .data hello

**Αποτέλεσμα:**

Contents of section .data:

601030 00000000 00000000 00000000 00000000 ................

**5.**

**Εντολή:** nm –a hello

**Αποτέλεσμα:**

0000000000000000 a

0000000000601040 b .bss

0000000000601040 B \_\_bss\_start

0000000000000000 n .comment

0000000000601040 b completed.6973

0000000000000000 a crtstuff.c

0000000000000000 a crtstuff.c

0000000000601030 d .data

0000000000601030 D \_\_data\_start

0000000000601030 W data\_start

0000000000400470 t deregister\_tm\_clones

00000000004004e0 t \_\_do\_global\_dtors\_aux

0000000000600e18 t \_\_do\_global\_dtors\_aux\_fini\_array\_entry

0000000000601038 D \_\_dso\_handle

0000000000600e28 d .dynamic

0000000000600e28 d \_DYNAMIC

0000000000400318 r .dynstr

00000000004002b8 r .dynsym

0000000000601040 D \_edata

0000000000400608 r .eh\_frame

00000000004005d0 r .eh\_frame\_hdr

0000000000601048 B \_end

00000000004005b4 T \_fini

00000000004005b4 t .fini

0000000000600e18 t .fini\_array

0000000000400500 t frame\_dummy

0000000000600e10 t \_\_frame\_dummy\_init\_array\_entry

00000000004006f8 r \_\_FRAME\_END\_\_

0000000000601000 d \_GLOBAL\_OFFSET\_TABLE\_

w \_\_gmon\_start\_\_

0000000000400298 r .gnu.hash

0000000000400356 r .gnu.version

0000000000400360 r .gnu.version\_r

0000000000600ff8 d .got

0000000000601000 d .got.plt

0000000000000000 a hello.c

00000000004003e0 T \_init

00000000004003e0 t .init

0000000000600e10 t .init\_array

0000000000600e18 t \_\_init\_array\_end

0000000000600e10 t \_\_init\_array\_start

0000000000400238 r .interp

00000000004005c0 R \_IO\_stdin\_used

w \_ITM\_deregisterTMCloneTable

w \_ITM\_registerTMCloneTable

0000000000600e20 d .jcr

0000000000600e20 d \_\_JCR\_END\_\_

0000000000600e20 d \_\_JCR\_LIST\_\_

w \_Jv\_RegisterClasses

00000000004005b0 T \_\_libc\_csu\_fini

0000000000400540 T \_\_libc\_csu\_init

U \_\_libc\_start\_main@@GLIBC\_2.2.5

000000000040052d T main

0000000000400254 r .note.ABI-tag

0000000000400274 r .note.gnu.build-id

0000000000400400 t .plt

U puts@@GLIBC\_2.2.5

00000000004004a0 t register\_tm\_clones

0000000000400380 r .rela.dyn

0000000000400398 r .rela.plt

00000000004005c0 r .rodata

0000000000400440 T \_start

0000000000400440 t .text

0000000000601040 D \_\_TMC\_END\_\_