Հայաստանի Ազգային Պոլիտեխնիկական Համալսարան

Առարկա*:* Ծրագրային անվտանգություն

Խումբ: SS-*355 (Լաբ.-1)*

Ուսանող: Մելիքյան Վերոնիկա

Outlook: veronikamelikyan.tt355@polytechnic.am

Լաբորատոր աշխատանք 9

Թեմա

1. Բինար կատարողական ֆայլի կառուցվածքը *Windows* և *Linux* օպերացիոն համակարգերում
2. *C++* լեզվով գրված պարզագույն *Hello World* ծրագրի աշխատող կոդի և տվյալների մասերը
3. Մեքենայական կոդից սկզբնական լեզվով կոդի վերականգնում

Օպերացիոն համակարգ՝ ***macOS (Apple Silicon — ARM64)***

Երկուական ֆայլի ձևաչափ՝ ***Mach-O (***նման է ***ELF***-ին ***Linux***-ում, ***PE***-ին ***Windows***-ում***)***

Կոմպիլյատոր՝ ***clang++***

Քայլ *1* - Ստեղծեք աշխատանքային թղթապանակ և աղբյուրի ֆայլ

```
1. mkdir ~/lab9 && cd ~/lab9

2. cat > hello.cpp << 'EOF'
#include <iostream>
int main() {
    std::cout << "Hello, World!";
    return 0;
}
EOF
```

```
(base) melikyan_v@MacBook-Air-Veronika lab9 % cat > hello.cpp << 'EOF'
heredoc> #include <iostream>
int main() {
    std::cout << "Hello, World!";
    return 0;
}
EOF
```

Ստեղծեցինք lab9 թղթապանակը և գրեցինք C++ լեզվով պարզ ծրագիր, որը արտածում է «Hello, World!» տողը:

*Քայլ **2** - Ծրագրի կոմպիլյացիա*

```
1. clang++ -o hello hello.cpp
2. Run: ./hello
```

```
(base) melikyan_v@MacBook-Air-Veronika lab9 % clang++ -o hello hello.cpp
(base) melikyan_v@MacBook-Air-Veronika lab9 % ./hello
Hello, World!%
```

*Քայլ **3** - Դիտել երկուական ֆայլի կառուցվածքը*

```
1. objdump -h hello
```

```
(base) melikyan_v@MacBook-Air-Veronika lab9 % objdump -h hello

hello:  file format mach-o arm64

Sections:
Idx Name              Size     VMA              Type
  0 __text            00000a78 00000001000004e8 TEXT
  1 __stubs           000000a8 0000000100000f60 TEXT
  2 __gcc_except_tab  0000006c 0000000100001008 DATA
  3 __cstring         0000000e 0000000100001074 DATA
  4 __unwind_info     000000f0 0000000100001084 DATA
  5 __got             00000088 0000000100004000 DATA
```

Սեկցիա	Չափ	Նշանակություն
__text	0xa78	Պրոգրամայի մեքենային կոդ (main ֆունկցիան և բոլոր կանչերը)
__cstring	0x0e = 14 բայթ	"Hello, World!" տող (13 նիշ + 1 զրո բայթ)
__stubs	0xa8	Արտաքին ֆունկցիաների կանչերի համար նախատեսված заглушки (cout)
__got	0x88	Արտաքին ֆունկցիաների հասցեների աղյուսակ
__gcc_except_tab	0x6c	C++ բացառությունների աղյուսակ
__unwind_info	0xf0	Ստեկի փաթաթման տեղեկատվություն

*Եզրակացություն. Կոդը և տվյալները պահվում են տարբեր բաժիններում: «Բարև, Աշխարհ!» տողը գտնվում է **__cstring**-ում՝ **0x1074** հասցեում, այլ ոչ թե կոդի բաժնում:*

Քայլ 4 – Երկուական ֆայլում տող գտնելը

***1** – Դուրս բերել բոլոր կարդացվող տողերը*

strings hello | grep "Hello"

```
(base) melikyan_v@MacBook-Air-Veronika lab9 % strings hello | grep "Hello"

Hello, World!
```

***2** – Գտնել տողը **hex** ներկայացման մեջ:*

xxd hello | grep -i "Hello"

```
[(base) melikyan_v@MacBook-Air-Veronika lab9 % xxd hello | grep -i "Hello" ]
00001070: 443c 0000 4865 6c6c 6f2c 2057 6f72 6c64  D<..Hello, World
```

*Տողը գտնվել է **0x1074 offset**-ում **(00001070 + 4** բայթ**)**:Սա ճշգրտորեն համընկնում է նախորդ ֆայլում **__cstring** բաժնի հասցեի հետ: **xxd**-ն ֆայլը ցուցադրում է տասնվեցական ֆորմատով՝ յուրաքանչյուր **2** նիշը հավասար է **1** բայթի:*

Քայլ 5 – Ապամոնտաժում (մեքենայի կոդ → ասեմբլեր)

objdump -d hello

```
(base) melikyan_v@MacBook-Air-Veronika lab9 % objdump -d hello

hello:  file format mach-o arm64

Disassembly of section __TEXT,__text:

00000001000004e8 <_main>:
1000004e8: d10083ff     sub     sp, sp, #0x20
1000004ec: a9017bfd     stp     x29, x30, [sp, #0x10]
1000004f0: 910043fd     add     x29, sp, #0x10
1000004f4: 52800008     mov     w8, #0x0                ; =0
1000004f8: b9000be8     str     w8, [sp, #0x8]
1000004fc: b81fc3bf     stur    wzr, [x29, #-0x4]
100000500: 90000020     adrp    x0, 0x100004000 <_strlen+0x100004000>
100000504: f9401800     ldr     x0, [x0, #0x30]
100000508: b0000001     adrp    x1, 0x100001000 <_strlen+0x100001000>
10000050c: 9101d021     add     x1, x1, #0x74
100000510: 94000005     bl      0x100000524 <__ZNSt3__1lsB9nqe210106INS_11
char_traitsIcEEEERNS_13basic_ostreamIcT_EES6_PKc>
100000514: b9400be0     ldr     w0, [sp, #0x8]
100000518: a9417bfd     ldp     x29, x30, [sp, #0x10]
10000051c: 910083ff     add     sp, sp, #0x20
100000520: d65f03c0     ret

0000000100000524 <__ZNSt3__1lsB9nqe210106INS_11char_traitsIcEEEERNS_13basi
c_ostreamIcT_EES6_PKc>:
100000524: d100c3ff     sub     sp, sp, #0x30
100000528: a9027bfd     stp     x29, x30, [sp, #0x20]
10000052c: 910083fd     add     x29, sp, #0x20
100000530: f81f83a0     stur    x0, [x29, #-0x8]
100000534: f9000be1     str     x1, [sp, #0x10]
100000538: f85f83a8     ldur    x8, [x29, #-0x8]
10000053c: f90007e8     str     x8, [sp, #0x8]
100000540: f9400be8     ldr     x8, [sp, #0x10]
100000544: f90003e8     str     x8, [sp]
100000548: f9400be0     ldr     x0, [sp, #0x10]
10000054c: 94000081     bl      0x100000750 <__ZNSt3__111char_traitsIcE6le
ngthB9nqe210106EPKc>
100000550: f94003e1     ldr     x1, [sp]
100000554: aa0003e2     mov     x2, x0
100000558: f94007e0     ldr     x0, [sp, #0x8]
10000055c: 94000004     bl      0x10000056c <__ZNSt3__124__put_character_s
equenceB9nqe210106IcNS_11char_traitsIcEEEERNS_13basic_ostreamIT_T0_EES7_PK
S4_m>
100000560: a9427bfd     ldp     x29, x30, [sp, #0x20]
100000564: 9100c3ff     add     sp, sp, #0x30
100000568: d65f03c0     ret
```

sub sp, sp, #0x20 ---- Ստեկի վրա տեղ հատկացնում է

add x1, x1, #0x74 ---- Տողի հասցեն 0x1074-ը բեռնում է x1 ռեգիստրում

bl 0x100000524 ---- Կանչում է cout << ֆունկցիան

ret ---- Վերադառնում է main-ից

*եզրակացություն. **add x1, x1, #0x74** հրահանգում **0x74** հասցեն հենց մեր տողի **offset**-ն է: Կպամոնտաժումը հաստատում է կոդի և տվյալների միջև կապը:*

Քայլ 6 - Դիտել սիմվոլների աղյուսակը

nm hello

```
0000000100000524 t __ZNSt3__1lsB9nqe210106INS_11char_traitsIcEEEERNS_13bas
ic_ostreamIcT_EES6_PKc
                 U __ZSt9terminatev
0000000100000aa0 t ___clang_call_terminate
                 U ___cxa_begin_catch
                 U ___cxa_end_catch
                 U ___gxx_personality_v0
0000000100000000 T __mh_execute_header
00000001000004e8 T _main
                 U _strlen
```

1. *T — հասանելի ֆունկցիա կոդի սեկցիայում, մեր `_main` ֆունկցիան է:*
2. *t — լոկալ ֆունկցիա (տեսանելի միայն իր մոդուլի ներսում):*
3. *U — Undefined (որոշված չէ) — ֆունկցիան արտաքին գրադարանից է, կապվում է կատարման ընթացքում (runtime-ում):*

***C++** անվանման խեղաթյուրում.*

`__ZNSt3__1lsB9nqe210106INS_11char_traitsIcEEEERNS_13basic_ostreamIcT_EES6_PKc`-ն իրականում std::cout << օպերատոր է: C++ կոմպիլյացիայի ժամանակ բոլոր ֆունկցիաների անունները «խեղաթյուրվում» են. սա կոչվում է անվանման խեղաթյուրում: Սկզբնական անունների հակադարձումը դժվարացնում է դրանց վերականգնումը:

Քայլ 7 – Երկուական խմբագրում (Կտրված գծեր)

***1.** Ստեղծեք պահուստային պատճեն **(резервную копию)**`*

cp hello hello_backup

```
[(base) melikyan_v@MacBook-Air-Veronika lab9 % cp hello hello_backup        ]
 (base) melikyan_v@MacBook-Air-Veronika lab9 % python3 << 'EOF'
 with open("hello", "rb") as f:
     data = bytearray(f.read())
 old = b"Hello, World!"
 new = b"Hi,   Planet!"
 idx = data.find(old)
 print(f"Found at offset: {hex(idx)}")
 data[idx:idx+len(old)] = new
 with open("hello", "wb") as f:
     f.write(data)
 print("Patched successfully!")
[EOF                                                                        ]
 Found at offset: 0x1074
 Patched successfully!
```

2.** Բինարային ֆայլի վերստորագրում **(macOS** պահանջ**):

codesign -s - hello

```
[(base) melikyan_v@MacBook-Air-Veronika lab9 % codesign -s - hello          ]
[(base) melikyan_v@MacBook-Air-Veronika lab9 % ./hello                      ]
 Hi,   Planet!%
```

macOS-ն ստուգում է կատարվող ֆայլերի թվային ստորագրությունը: Բայթերը փոխելուց հետո սկզբնական ստորագրությունն անվավեր է դառնում, և համակարգը արգելափակում է կատարումը *(killed): Команда codesign -s -* ստեղծում է նոր տեղական ստորագրություն:

"Hi, Planet!" տողը պարունակում է *13* սիմվոլ, ինչպես նաև սկզբնական *"Hello, World!"* տողը: Եթե նոր տողը ավելի երկար լիներ, այն կփոխարիներ հարակից բայթերը, և ծրագիրը կարող էր վարակում ստանալ:

*Քայլ 8 – Ստուգեք թարմացումը **xxd**-ի միջոցով*

xxd hello | grep -i "Planet"

```
(base) melikyan_v@MacBook-Air-Veronika lab9 % xxd hello | grep -i "Planet"

00001070: 443c 0000 4869 2c20 2020 506c 616e 6574  D<..Hi,   Planet
(base) melikyan_v@MacBook-Air-Veronika lab9 %
```

փոխվել է նույն հասցեում՝ 0x1074

Քայլ 9 – Վերականգնել բնօրինակը

cp hello_backup hello

codesign -s - hello

./hello

```
(base) melikyan_v@MacBook-Air-Veronika lab9 % cp hello_backup hello
codesign -s - hello
./hello
Hello, World!%
```

եզրահանգումներ

1. Բինար ֆայլի կառուցվածքը.
   macOS-ում օգտագործվում է ***Mach-O*** ձևաչափը (օգնակիցն է ***ELF***-ի ***Linux***-ում և ***PE***-ի ***Windows***-ում): Ֆայլը բաժանված է սեկցիաների՝ ***__text***-ը կոդի, ***__cstring***-ը տողերի, ***__data***-ն էլ տվյալների համար:
2. Կոդի և տվյալների անջատումը.
   "Hello, World!" տողը պահվում է ***__cstring*** սեկցիայում ***0x1074*** հասցեի վրա՝ առանձին ***__text***-ում գտնվող մեքենային կոդից: Դա հաստատվում է երեք անկախ գործիքներով՝ ***objdump -h, xxd*** և դիզասեմբլերով:
3. Ռեվերս ինժինիրինգ ***(reverse-engineering).***
   - Դիզասեմբլացիան ***(objdump -d)*** թույլ է տալիս մեքենային կոդը տեսնել որպես ասսեմբլերային հրահանգներ:
   - Սիմվոլների աղյուսակը ***(nm)*** ցույց է տալիս ֆունկցիաներ, սակայն ***C++***-ի անունները աղանդավորված են ***(name mangling):***
   - Բինար խմբագրումը թույլ է տալիս փոխել ծրագրի վարքագիծը առանց հասանալու սկզբնական կոդին՝ պարզապես փոխելով տվյալների սեկցիայում գտնվող բայթերը: