Link Example 1.B Static Linking - Listings

Young W. Lim

2019-01-21 Mon

Outline

1. Based on
2. relocation section listings
   - `nothing.o` relocation section listing
   - `nmain.o` relocation section listing
   - `nmain.out` relocation section listing
3. function call binary listings
   - `doNothingStatic` function listing
   - `doNothing` function listing
   - `doAlmostNothing` function listing
4. `nothing.o` binary file listings
   - using `objdump -d nothing.o`
   - using `objdump -dr nothing.o`
   - using `objdump -D nothing.o`
5. `nmain.o` binary file listings
   - using `objdump -d nmain.o`
   - using `objdump -dr nmain.o`
   - using `objdump -D nmain.o`

Based on

1. https://stac47.github.io/c/relocation/elf/tutorial/2018/03/01/understanding-relocation-elf.html

Compling 32-bit program on 64-bit gcc

- gcc -v
- gcc -m32 t.c
- sudo apt-get install gcc-multilib
- sudo apt-get install g++-multilib
- gcc-multilib
- g++-multilib
- gcc -m32
- objdump -m i386

readelf -r nothing.o (1)

```
young@USys3:~/nmain$ readelf -r nothing.o

Relocation section '.rel.text' at offset 0x45c contains 7 entries:
 Offset     Info    Type            Sym.Value  Sym. Name
00000004  00001002 R_386_PC32        00000000   __x86.get_pc_thunk.ax
00000009  0000110a R_386_GOTPC       00000000   _GLOBAL_OFFSET_TABLE_
00000014  00001002 R_386_PC32        00000000   __x86.get_pc_thunk.ax
00000019  0000110a R_386_GOTPC       00000000   _GLOBAL_OFFSET_TABLE_
00000024  00001002 R_386_PC32        00000000   __x86.get_pc_thunk.ax
00000029  0000110a R_386_GOTPC       00000000   _GLOBAL_OFFSET_TABLE_
00000033  00001202 R_386_PC32        00000010   doNothing
```

readelf -r nothing.o (2)

```
Relocation section '.rel.debug_info' at offset 0x494 contains 12 entries:
 Offset     Info    Type            Sym.Value  Sym. Name
00000006  00000801 R_386_32          00000000   .debug_abbrev
0000000c  00000b01 R_386_32          00000000   .debug_str
00000011  00000b01 R_386_32          00000000   .debug_str
00000015  00000b01 R_386_32          00000000   .debug_str
00000019  00000201 R_386_32          00000000   .text
00000021  00000a01 R_386_32          00000000   .debug_line
00000026  00000b01 R_386_32          00000000   .debug_str
0000002c  00000201 R_386_32          00000000   .text
00000037  00000b01 R_386_32          00000000   .debug_str
0000003d  00000201 R_386_32          00000000   .text
00000048  00000b01 R_386_32          00000000   .debug_str
0000004e  00000201 R_386_32          00000000   .text
```

readelf -r nothing.o (3)

```
Relocation section '.rel.debug_aranges' at offset 0x4f4 contains 2 entries:
 Offset     Info    Type            Sym.Value  Sym. Name
00000006  00000701 R_386_32          00000000   .debug_info
00000010  00000201 R_386_32          00000000   .text

Relocation section '.rel.debug_line' at offset 0x504 contains 1 entry:
 Offset     Info    Type            Sym.Value  Sym. Name
0000002d  00000201 R_386_32          00000000   .text

Relocation section '.rel.eh_frame' at offset 0x50c contains 4 entries:
 Offset     Info    Type            Sym.Value  Sym. Name
00000020  00000202 R_386_PC32        00000000   .text
00000040  00000202 R_386_PC32        00000000   .text
00000060  00000202 R_386_PC32        00000000   .text
00000080  00000602 R_386_PC32        00000000   .text.__x86.get_pc_thu
```

readelf -r nmain.o (1)

```
young@USys3:~/nmain$ readelf -r nmain.o

Relocation section '.rel.text' at offset 0x410 contains 3 entries:
 Offset     Info    Type            Sym.Value  Sym. Name
00000010  00001002 R_386_PC32        00000000   __x86.get_pc_thunk.ax
00000015  0000110a R_386_GOTPC       00000000   _GLOBAL_OFFSET_TABLE_
0000001c  00001204 R_386_PLT32       00000000   doAlmostNothing
```

readelf -r nmain.o (2)

```
Relocation section '.rel.debug_info' at offset 0x428 contains 11 entries:
 Offset     Info    Type            Sym.Value  Sym. Name
00000006  00000701 R_386_32          00000000   .debug_abbrev
0000000c  00000a01 R_386_32          00000000   .debug_str
00000011  00000a01 R_386_32          00000000   .debug_str
00000015  00000a01 R_386_32          00000000   .debug_str
00000019  00000201 R_386_32          00000000   .text
00000021  00000901 R_386_32          00000000   .debug_line
00000026  00000a01 R_386_32          00000000   .debug_str
00000030  00000201 R_386_32          00000000   .text
0000003f  00000a01 R_386_32          00000000   .debug_str
0000004d  00000a01 R_386_32          00000000   .debug_str
00000071  00000a01 R_386_32          00000000   .debug_str
```

readelf -r nmain.o (3)

```
Relocation section '.rel.debug_aranges' at offset 0x480 contains 2 entries:
 Offset     Info    Type            Sym.Value  Sym. Name
00000006  00000601 R_386_32          00000000   .debug_info
00000010  00000201 R_386_32          00000000   .text

Relocation section '.rel.debug_line' at offset 0x490 contains 1 entry:
 Offset     Info    Type            Sym.Value  Sym. Name
0000002b  00000201 R_386_32          00000000   .text

Relocation section '.rel.eh_frame' at offset 0x498 contains 2 entries:
 Offset     Info    Type            Sym.Value  Sym. Name
00000020  00000202 R_386_PC32        00000000   .text
00000054  00000502 R_386_PC32        00000000   .text.__x86.get_pc_thu
```

readelf -r nmain.out

```
young@USys3:~/nmain$ readelf -r nmain.out

Relocation section '.rel.dyn' at offset 0x310 contains 8 entries:
 Offset     Info    Type            Sym.Value  Sym. Name
00001edc  00000008 R_386_RELATIVE
00001ee0  00000008 R_386_RELATIVE
00001ff8  00000008 R_386_RELATIVE
00002004  00000008 R_386_RELATIVE
00001fec  00000106 R_386_GLOB_DAT    00000000   _ITM_deregisterTMClone
00001ff0  00000206 R_386_GLOB_DAT    00000000   __cxa_finalize@GLIBC_2.1.3
00001ff4  00000306 R_386_GLOB_DAT    00000000   __gmon_start__
00001ffc  00000506 R_386_GLOB_DAT    00000000   _ITM_registerTMCloneTa

Relocation section '.rel.plt' at offset 0x350 contains 1 entry:
 Offset     Info    Type            Sym.Value  Sym. Name
00001fe8  00000407 R_386_JUMP_SLOT   00000000   __libc_start_main@GLIBC_2.0
```

1. doNothingStatic (A) objdump -d nothing.o

```
00000000 <doNothingStatic>:
   0:   55                      push   %ebp
   1:   89 e5                   mov    %esp,%ebp
   3:   e8 fc ff ff ff          call   4 <doNothingStatic+0x4>
   8:   05 01 00 00 00          add    $0x1,%eax
   d:   90                      nop
   e:   5d                      pop    %ebp
   f:   c3                      ret
```

1. doNothingStatic (B) objdump -dr nothing.o

```
00000000 <doNothingStatic>:
   0:   55                      push   %ebp
   1:   89 e5                   mov    %esp,%ebp
   3:   e8 fc ff ff ff          call   4 <doNothingStatic+0x4>
                        4: R_386_PC32   __x86.get_pc_thunk.ax
   8:   05 01 00 00 00          add    $0x1,%eax
                        9: R_386_GOTPC  _GLOBAL_OFFSET_TABLE_
   d:   90                      nop
   e:   5d                      pop    %ebp
   f:   c3                      ret
```

1. doNothingStatic (C) objdump -d nmain.out

```
0000051d <doNothingStatic>:
 51d:   55                      push   %ebp
 51e:   89 e5                   mov    %esp,%ebp
 520:   e8 f4 ff ff ff          call   519 <__x86.get_pc_thunk.ax>
 525:   05 b7 1a 00 00          add    $0x1ab7,%eax
 52a:   90                      nop
 52b:   5d                      pop    %ebp
 52c:   c3                      ret
```

2. doNothing (A) objdump -d nothing.o

```
00000010 <doNothing>:
  10:   55                      push   %ebp
  11:   89 e5                   mov    %esp,%ebp
  13:   e8 fc ff ff ff          call   14 <doNothing+0x4>
  18:   05 01 00 00 00          add    $0x1,%eax
  1d:   90                      nop
  1e:   5d                      pop    %ebp
  1f:   c3                      ret
```

2. doNothing (B) objdump -dr nothing.o

```
00000010 <doNothing>:
  10:   55                      push   %ebp
  11:   89 e5                   mov    %esp,%ebp
  13:   e8 fc ff ff ff          call   14 <doNothing+0x4>
                        14: R_386_PC32  __x86.get_pc_thunk.ax
  18:   05 01 00 00 00          add    $0x1,%eax
                        19: R_386_GOTPC _GLOBAL_OFFSET_TABLE_
  1d:   90                      nop
  1e:   5d                      pop    %ebp
  1f:   c3                      ret
```

2. doNothing (C) objdump -d nmain.out

```
0000052d <doNothing>:
 52d:   55                      push   %ebp
 52e:   89 e5                   mov    %esp,%ebp
 530:   e8 e4 ff ff ff          call   519 <__x86.get_pc_thunk.ax>
 535:   05 a7 1a 00 00          add    $0x1aa7,%eax
 53a:   90                      nop
 53b:   5d                      pop    %ebp
 53c:   c3                      ret
```

3. doAlmostNothing (A) objdump -d nothing.o

```
00000020 <doAlmostNothing>:
  20:   55                      push   %ebp
  21:   89 e5                   mov    %esp,%ebp
  23:   e8 fc ff ff ff          call   24 <doAlmostNothing+0x4>
  28:   05 01 00 00 00          add    $0x1,%eax
  2d:   e8 ce ff ff ff          call   0 <doNothingStatic>
  32:   e8 fc ff ff ff          call   33 <doAlmostNothing+0x13>
  37:   90                      nop
  38:   5d                      pop    %ebp
  39:   c3                      ret
```

3. doAlmostNothing (B) objdump -dr nothing.o

```
00000020 <doAlmostNothing>:
  20:   55                      push   %ebp
  21:   89 e5                   mov    %esp,%ebp
  23:   e8 fc ff ff ff          call   24 <doAlmostNothing+0x4>
                        24: R_386_PC32  __x86.get_pc_thunk.ax
  28:   05 01 00 00 00          add    $0x1,%eax
                        29: R_386_GOTPC _GLOBAL_OFFSET_TABLE_
  2d:   e8 ce ff ff ff          call   0 <doNothingStatic>
  32:   e8 fc ff ff ff          call   33 <doAlmostNothing+0x13>
                        33: R_386_PC32  doNothing
  37:   90                      nop
  38:   5d                      pop    %ebp
  39:   c3                      ret
```

3. doAlmostNothing (C) objdump -d nmain.out

```
0000053d <doAlmostNothing>:
 53d:   55                      push   %ebp
 53e:   89 e5                   mov    %esp,%ebp
 540:   e8 d4 ff ff ff          call   519 <__x86.get_pc_thunk.ax>
 545:   05 97 1a 00 00          add    $0x1a97,%eax
 54a:   e8 ce ff ff ff          call   51d <doNothingStatic>
 54f:   e8 d9 ff ff ff          call   52d <doNothing>
 554:   90                      nop
 555:   5d                      pop    %ebp
 556:   c3                      ret
 557:   66 90                   xchg   %ax,%ax
 559:   66 90                   xchg   %ax,%ax
 55b:   66 90                   xchg   %ax,%ax
 55d:   66 90                   xchg   %ax,%ax
 55f:   90                      nop
```

1. objdump -d nothing.o (A) doNothingStatic

```
young@USys1:~$ gcc -nostdlib -c -m32 nothing.c
young@USys1:~$ objdump -d nothing.o

nothing.o:     file format elf32-i386


Disassembly of section .text:

00000000 <doNothingStatic>:
   0:   55                      push   %ebp
   1:   89 e5                   mov    %esp,%ebp
   3:   e8 fc ff ff ff          call   4 <doNothingStatic+0x4>
   8:   05 01 00 00 00          add    $0x1,%eax
   d:   90                      nop
   e:   5d                      pop    %ebp
   f:   c3                      ret
```

1. objdump -d nothing.o (B) doNothing

```
00000010 <doNothing>:
  10:   55                      push   %ebp
  11:   89 e5                   mov    %esp,%ebp
  13:   e8 fc ff ff ff          call   14 <doNothing+0x4>
  18:   05 01 00 00 00          add    $0x1,%eax
  1d:   90                      nop
  1e:   5d                      pop    %ebp
  1f:   c3                      ret
```

1. objdump -d nothing.o (C) doAlmostNothing

```
00000020 <doAlmostNothing>:
  20:   55                      push   %ebp
  21:   89 e5                   mov    %esp,%ebp
  23:   e8 fc ff ff ff          call   24 <doAlmostNothing+0x4>
  28:   05 01 00 00 00          add    $0x1,%eax
  2d:   e8 ce ff ff ff          call   0 <doNothingStatic>
  32:   e8 fc ff ff ff          call   33 <doAlmostNothing+0x13>
  37:   90                      nop
  38:   5d                      pop    %ebp
  39:   c3                      ret
```

1. objdump -d nothing.o (D) __x86.get_pc_thunk.ax

```
Disassembly of section .text.__x86.get_pc_thunk.ax:

00000000 <__x86.get_pc_thunk.ax>:
   0:   8b 04 24                mov    (%esp),%eax
   3:   c3                      ret
```

2. objdump -dr nothing.o (A) doNothingStatic

```
Disassembly of section .text:

00000000 <doNothingStatic>:
   0:   55                      push   %ebp
   1:   89 e5                   mov    %esp,%ebp
   3:   e8 fc ff ff ff          call   4 <doNothingStatic+0x4>
                        4: R_386_PC32   __x86.get_pc_thunk.ax
   8:   05 01 00 00 00          add    $0x1,%eax
                        9: R_386_GOTPC  _GLOBAL_OFFSET_TABLE_
   d:   90                      nop
   e:   5d                      pop    %ebp
   f:   c3                      ret
```

2. objdump -dr nothing.o (B) doNothing

```
00000010 <doNothing>:
  10:   55                      push   %ebp
  11:   89 e5                   mov    %esp,%ebp
  13:   e8 fc ff ff ff          call   14 <doNothing+0x4>
                        14: R_386_PC32  __x86.get_pc_thunk.ax
  18:   05 01 00 00 00          add    $0x1,%eax
                        19: R_386_GOTPC _GLOBAL_OFFSET_TABLE_
  1d:   90                      nop
  1e:   5d                      pop    %ebp
  1f:   c3                      ret
```

2. objdump -dr nothing.o (C) doAlmostNothing

```
00000020 <doAlmostNothing>:
  20:   55                      push   %ebp
  21:   89 e5                   mov    %esp,%ebp
  23:   e8 fc ff ff ff          call   24 <doAlmostNothing+0x4>
                        24: R_386_PC32  __x86.get_pc_thunk.ax
  28:   05 01 00 00 00          add    $0x1,%eax
                        29: R_386_GOTPC _GLOBAL_OFFSET_TABLE_
  2d:   e8 ce ff ff ff          call   0 <doNothingStatic>
  32:   e8 fc ff ff ff          call   33 <doAlmostNothing+0x13>
                        33: R_386_PC32  doNothing
  37:   90                      nop
  38:   5d                      pop    %ebp
  39:   c3                      ret
```

2. objdump -dr nothing.o (D) __x86.get_pc_thunk.ax

```
Disassembly of section .text.__x86.get_pc_thunk.ax:

00000000 <__x86.get_pc_thunk.ax>:
   0:   8b 04 24                mov    (%esp),%eax
   3:   c3                      ret
```

3. objdump -D nothing.o (A) doNothingStatic

```
young@USys1:~$ objdump -D nothing.o

nothing.o:     file format elf32-i386


Disassembly of section .group:

00000000 <.group>:
   0:   01 00                   add    %eax,(%eax)
   2:   00 00                   add    %al,(%eax)
   4:   06                      push   %es
   5:   00 00                   add    %al,(%eax)
        ...
```

3. objdump -D nothing.o (B) doNothingStatic

```
Disassembly of section .text:

00000000 <doNothingStatic>:
   0:   55                      push   %ebp
   1:   89 e5                   mov    %esp,%ebp
   3:   e8 fc ff ff ff          call   4 <doNothingStatic+0x4>
   8:   05 01 00 00 00          add    $0x1,%eax
   d:   90                      nop
   e:   5d                      pop    %ebp
   f:   c3                      ret
```

3. objdump -D nothing.o (C) doNothing

```
00000010 <doNothing>:
  10:   55                      push   %ebp
  11:   89 e5                   mov    %esp,%ebp
  13:   e8 fc ff ff ff          call   14 <doNothing+0x4>
  18:   05 01 00 00 00          add    $0x1,%eax
  1d:   90                      nop
  1e:   5d                      pop    %ebp
  1f:   c3                      ret

00000020 <doAlmostNothing>:
  20:   55                      push   %ebp
  21:   89 e5                   mov    %esp,%ebp
  23:   e8 fc ff ff ff          call   24 <doAlmostNothing+0x4>
  28:   05 01 00 00 00          add    $0x1,%eax
  2d:   e8 ce ff ff ff          call   0 <doNothingStatic>
  32:   e8 fc ff ff ff          call   33 <doAlmostNothing+0x13>
  37:   90                      nop
  38:   5d                      pop    %ebp
  39:   c3                      ret

Disassembly of section .text.__x86.get_pc_thunk.ax:

00000000 <__x86.get_pc_thunk.ax>:
   0:   8b 04 24                mov    (%esp),%eax
```

3. objdump -D nothing.o (D) doAlmostNothing

```
00000010 <doNothing>:
  10:   55                      push   %ebp
  11:   89 e5                   mov    %esp,%ebp
  13:   e8 fc ff ff ff          call   14 <doNothing+0x4>
  18:   05 01 00 00 00          add    $0x1,%eax
  1d:   90                      nop
  1e:   5d                      pop    %ebp
  1f:   c3                      ret

00000020 <doAlmostNothing>:
  20:   55                      push   %ebp
  21:   89 e5                   mov    %esp,%ebp
  23:   e8 fc ff ff ff          call   24 <doAlmostNothing+0x4>
  28:   05 01 00 00 00          add    $0x1,%eax
  2d:   e8 ce ff ff ff          call   0 <doNothingStatic>
  32:   e8 fc ff ff ff          call   33 <doAlmostNothing+0x13>
  37:   90                      nop
  38:   5d                      pop    %ebp
  39:   c3                      ret

Disassembly of section .text.__x86.get_pc_thunk.ax:

00000000 <__x86.get_pc_thunk.ax>:
   0:   8b 04 24                mov    (%esp),%eax
```

3. objdump -D nothing.o (E) .comment

```
Disassembly of section .comment:

00000000 <.comment>:
   0:   00 47 43                add    %al,0x43(%edi)
   3:   43                      inc    %ebx
   4:   3a 20                   cmp    (%eax),%ah
   6:   28 55 62                sub    %dl,0x62(%ebp)
   9:   75 6e                   jne    79 <doAlmostNothing+0x59>
   b:   74 75                   je     82 <doAlmostNothing+0x62>
   d:   20 37                   and    %dh,(%edi)
   f:   2e 33 2e                xor    %cs:(%esi),%ebp
  12:   30 2d 32 37 75 62       xor    %ch,0x62753732
  18:   75 6e                   jne    88 <doAlmostNothing+0x68>
  1a:   74 75                   je     91 <doAlmostNothing+0x71>
  1c:   31 7e 31                xor    %edi,0x31(%esi)
  1f:   38 2e                   cmp    %ch,(%esi)
  21:   30 34 29                xor    %dh,(%ecx,%ebp,1)
  24:   20 37                   and    %dh,(%edi)
  26:   2e 33 2e                xor    %cs:(%esi),%ebp
  29:   30 00                   xor    %al,(%eax)
```

3. objdump -D nothing.o (F) .eh_frame (a)

```
Disassembly of section .eh_frame:

00000000 <.eh_frame>:
   0:   14 00                   adc    $0x0,%al
   2:   00 00                   add    %al,(%eax)
   4:   00 00                   add    %al,(%eax)
   6:   00 00                   add    %al,(%eax)
   8:   01 7a 52                add    %edi,0x52(%edx)
   b:   00 01                   add    %al,(%ecx)
   d:   7c 08                   jl     17 <.eh_frame+0x17>
   f:   01 1b                   add    %ebx,(%ebx)
  11:   0c 04                   or     $0x4,%al
  13:   04 88                   add    $0x88,%al
  15:   01 00                   add    %eax,(%eax)
  17:   00 1c 00                add    %bl,(%eax,%eax,1)
  1a:   00 00                   add    %al,(%eax)
  1c:   1c 00                   sbb    $0x0,%al
  1e:   00 00                   add    %al,(%eax)
  20:   00 00                   add    %al,(%eax)
  22:   00 00                   add    %al,(%eax)
  24:   10 00                   adc    %al,(%eax)
  26:   00 00                   add    %al,(%eax)
  28:   00 41 0e                add    %al,0xe(%ecx)
  2b:   08 85 02 42 0d 05       or     %al,0x50d4202(%ebp)
```

3. objdump -D nothing.o (G) .eh_frame (b)

```
31:   4c                      dec    %esp
32:   c5 0c 04                lds    (%esp,%eax,1),%ecx
35:   04 00                   add    $0x0,%al
37:   00 1c 00                add    %bl,(%eax,%eax,1)
3a:   00 00                   add    %al,(%eax)
3c:   3c 00                   cmp    $0x0,%al
3e:   00 00                   add    %al,(%eax)
40:   10 00                   adc    %al,(%eax)
42:   00 00                   add    %al,(%eax)
44:   10 00                   adc    %al,(%eax)
46:   00 00                   add    %al,(%eax)
48:   00 41 0e                add    %al,0xe(%ecx)
4b:   08 85 02 42 0d 05       or     %al,0x50d4202(%ebp)
51:   4c                      dec    %esp
52:   c5 0c 04                lds    (%esp,%eax,1),%ecx
55:   04 00                   add    $0x0,%al
57:   00 1c 00                add    %bl,(%eax,%eax,1)
5a:   00 00                   add    %al,(%eax)
5c:   5c                      pop    %esp
5d:   00 00                   add    %al,(%eax)
5f:   00 20                   add    %ah,(%eax)
```

3. objdump -D nothing.o (H) .eh_frame (c)

```
61:   00 00                   add    %al,(%eax)
63:   00 1a                   add    %bl,(%edx)
65:   00 00                   add    %al,(%eax)
67:   00 00                   add    %al,(%eax)
69:   41                      inc    %ecx
6a:   0e                      push   %cs
6b:   08 85 02 42 0d 05       or     %al,0x50d4202(%ebp)
71:   56                      push   %esi
72:   c5 0c 04                lds    (%esp,%eax,1),%ecx
75:   04 00                   add    $0x0,%al
77:   00 10                   add    %dl,(%eax)
79:   00 00                   add    %al,(%eax)
7b:   00 7c 00 00             add    %bh,0x0(%eax,%eax,1)
7f:   00 00                   add    %al,(%eax)
81:   00 00                   add    %al,(%eax)
83:   00 04 00                add    %al,(%eax,%eax,1)
86:   00 00                   add    %al,(%eax)
88:   00 00                   add    %al,(%eax)
```

1. objdump -d nmain.o (A)

```
Disassembly of section .text:

00000000 <main>:
   0:   8d 4c 24 04             lea    0x4(%esp),%ecx
   4:   83 e4 f0                and    $0xfffffff0,%esp
   7:   ff 71 fc                pushl  -0x4(%ecx)
   a:   55                      push   %ebp
   b:   89 e5                   mov    %esp,%ebp
   d:   53                      push   %ebx
   e:   51                      push   %ecx
   f:   e8 fc ff ff ff          call   10 <main+0x10>
  14:   05 01 00 00 00          add    $0x1,%eax
  19:   89 c3                   mov    %eax,%ebx
  1b:   e8 fc ff ff ff          call   1c <main+0x1c>
  20:   b8 00 00 00 00          mov    $0x0,%eax
  25:   59                      pop    %ecx
  26:   5b                      pop    %ebx
  27:   5d                      pop    %ebp
  28:   8d 61 fc                lea    -0x4(%ecx),%esp
  2b:   c3                      ret
```

1. objdump -d nmain.o (B)

```
Disassembly of section .text.__x86.get_pc_thunk.ax:

00000000 <__x86.get_pc_thunk.ax>:
   0:   8b 04 24                mov    (%esp),%eax
   3:   c3                      ret
```

2. objdump -dr nothing.o (A)

```
Disassembly of section .text:

00000000 <main>:
   0:   8d 4c 24 04             lea    0x4(%esp),%ecx
   4:   83 e4 f0                and    $0xfffffff0,%esp
   7:   ff 71 fc                pushl  -0x4(%ecx)
   a:   55                      push   %ebp
   b:   89 e5                   mov    %esp,%ebp
   d:   53                      push   %ebx
   e:   51                      push   %ecx
   f:   e8 fc ff ff ff          call   10 <main+0x10>
                        10: R_386_PC32  __x86.get_pc_thunk.ax
  14:   05 01 00 00 00          add    $0x1,%eax
                        15: R_386_GOTPC _GLOBAL_OFFSET_TABLE_
  19:   89 c3                   mov    %eax,%ebx
  1b:   e8 fc ff ff ff          call   1c <main+0x1c>
                        1c: R_386_PLT32 doAlmostNothing
```

2. objdump -dr nothing.o (B)

```
  20:   b8 00 00 00 00          mov    $0x0,%eax
  25:   59                      pop    %ecx
  26:   5b                      pop    %ebx
  27:   5d                      pop    %ebp
  28:   8d 61 fc                lea    -0x4(%ecx),%esp
  2b:   c3                      ret

Disassembly of section .text.__x86.get_pc_thunk.ax:

00000000 <__x86.get_pc_thunk.ax>:
   0:   8b 04 24                mov    (%esp),%eax
   3:   c3                      ret
```

3. objdump -D nothing.o (A)

```
Disassembly of section .text:

00000000 <main>:
   0:   8d 4c 24 04             lea    0x4(%esp),%ecx
   4:   83 e4 f0                and    $0xfffffff0,%esp
   7:   ff 71 fc                pushl  -0x4(%ecx)
   a:   55                      push   %ebp
   b:   89 e5                   mov    %esp,%ebp
   d:   53                      push   %ebx
   e:   51                      push   %ecx
   f:   e8 fc ff ff ff          call   10 <main+0x10>
  14:   05 01 00 00 00          add    $0x1,%eax
  19:   89 c3                   mov    %eax,%ebx
  1b:   e8 fc ff ff ff          call   1c <main+0x1c>
  20:   b8 00 00 00 00          mov    $0x0,%eax
  25:   59                      pop    %ecx
  26:   5b                      pop    %ebx
  27:   5d                      pop    %ebp
  28:   8d 61 fc                lea    -0x4(%ecx),%esp
  2b:   c3                      ret
```

3. objdump -D nothing.o (B)

```
Disassembly of section .text.__x86.get_pc_thunk.ax:

00000000 <__x86.get_pc_thunk.ax>:
   0:   8b 04 24                mov    (%esp),%eax
   3:   c3                      ret
```

1. objdump -d nmain.out (1) .init

```
nmain.out:     formato del fichero elf32-i386


Desensamblado de la sección .init:

00000358 <_init>:
 358:   53                      push   %ebx
 359:   83 ec 08                sub    $0x8,%esp
 35c:   e8 8f 00 00 00          call   3f0 <__x86.get_pc_thunk.bx>
 361:   81 c3 7b 1c 00 00       add    $0x1c7b,%ebx
 367:   8b 83 18 00 00 00       mov    0x18(%ebx),%eax
 36d:   85 c0                   test   %eax,%eax
 36f:   74 05                   je     376 <_init+0x1e>
 371:   e8 32 00 00 00          call   3a8 <__gmon_start__@plt>
 376:   83 c4 08                add    $0x8,%esp
 379:   5b                      pop    %ebx
 37a:   c3                      ret
```

1. objdump -d nmain.out (2) .plt

```
Desensamblado de la sección .plt:

00000380 <.plt>:
 380:   ff b3 04 00 00 00       pushl  0x4(%ebx)
 386:   ff a3 08 00 00 00       jmp    *0x8(%ebx)
 38c:   00 00                   add    %al,(%eax)
        ...

00000390 <__libc_start_main@plt>:
 390:   ff a3 0c 00 00 00       jmp    *0xc(%ebx)
 396:   68 00 00 00 00          push   $0x0
 39b:   e9 e0 ff ff ff          jmp    380 <.plt>
```

1. objdump -d nmain.out (3) .plt.got

```
Desensamblado de la sección .plt.got:

000003a0 <__cxa_finalize@plt>:
 3a0:   ff a3 14 00 00 00       jmp    *0x14(%ebx)
 3a6:   66 90                   xchg   %ax,%ax

000003a8 <__gmon_start__@plt>:
 3a8:   ff a3 18 00 00 00       jmp    *0x18(%ebx)
 3ae:   66 90                   xchg   %ax,%ax
```

1. objdump -d nmain.out (4) .text

```
000003b0 <_start>:
000003f0 <__x86.get_pc_thunk.bx>:
00000400 <deregister_tm_clones>:
00000400 <deregister_tm_clones>:
00000440 <register_tm_clones>:
00000490 <__do_global_dtors_aux>:
000004e0 <frame_dummy>:
000004e9 <__x86.get_pc_thunk.dx>:
000004ed <main>:
00000519 <__x86.get_pc_thunk.ax>:
0000051d <doNothingStatic>:
0000052d <doNothing>:
0000053d <doAlmostNothing>:
00000560 <__libc_csu_init>:
000005c0 <__libc_csu_fini>:
```

1. objdump -d nmain.out (5) .fini

```
Disassembly of section .fini:

000005c4 <_fini>:
 5c4:   53                      push   %ebx
 5c5:   83 ec 08                sub    $0x8,%esp
 5c8:   e8 23 fe ff ff          call   3f0 <__x86.get_pc_thunk.bx>
 5cd:   81 c3 0f 1a 00 00       add    $0x1a0f,%ebx
 5d3:   83 c4 08                add    $0x8,%esp
 5d6:   5b                      pop    %ebx
 5d7:   c3                      ret
y
```

2. objdump -d nmain.out (1) _start (a)

```
000003b0 <_start>:
 3b0:   31 ed                    xor    %ebp,%ebp
 3b2:   5e                       pop    %esi
 3b3:   89 e1                    mov    %esp,%ecx
 3b5:   83 e4 f0                 and    $0xfffffff0,%esp
 3b8:   50                       push   %eax
 3b9:   54                       push   %esp
 3ba:   52                       push   %edx
 3bb:   e8 22 00 00 00           call   3e2 <_start+0x32>
 3c0:   81 c3 1c 1c 00 00        add    $0x1c1c,%ebx
 3c6:   8d 83 e4 e5 ff ff        lea    -0x1a1c(%ebx),%eax
 3cc:   50                       push   %eax
 3cd:   8d 83 84 e5 ff ff        lea    -0x1a7c(%ebx),%eax
 3d3:   50                       push   %eax
 3d4:   51                       push   %ecx
 3d5:   56                       push   %esi
 3d6:   ff b3 1c 00 00 00        pushl  0x1c(%ebx)
```

2. objdump -d nmain.out (2) _start (b)

```
3dc:   e8 af ff ff ff          call   390 <__libc_start_main@plt>
3e1:   f4                      hlt
3e2:   8b 1c 24                mov    (%esp),%ebx
3e5:   c3                      ret
3e6:   66 90                   xchg   %ax,%ax
3e8:   66 90                   xchg   %ax,%ax
3ea:   66 90                   xchg   %ax,%ax
3ec:   66 90                   xchg   %ax,%ax
3ee:   66 90                   xchg   %ax,%ax
```

2. objdump -d nmain.out (3) __x86.get_pc_thunk.bs

```
000003f0 <__x86.get_pc_thunk.bx>:
 3f0:   8b 1c 24                mov    (%esp),%ebx
 3f3:   c3                      ret
 3f4:   66 90                   xchg   %ax,%ax
 3f6:   66 90                   xchg   %ax,%ax
 3f8:   66 90                   xchg   %ax,%ax
 3fa:   66 90                   xchg   %ax,%ax
 3fc:   66 90                   xchg   %ax,%ax
 3fe:   66 90                   xchg   %ax,%ax
```

2. objdump -d nmain.out (4) deregister_tm_clones

```
00000400 <deregister_tm_clones>:
 400:   e8 e4 00 00 00          call   4e9 <__x86.get_pc_thunk.dx>
 405:   81 c2 d7 1b 00 00       add    $0x1bd7,%edx
 40b:   8d 8a 2c 00 00 00       lea    0x2c(%edx),%ecx
 411:   8d 82 2c 00 00 00       lea    0x2c(%edx),%eax
 417:   39 c8                   cmp    %ecx,%eax
 419:   74 1d                   je     438 <deregister_tm_clones+0x38>
 41b:   8b 82 10 00 00 00       mov    0x10(%edx),%eax
 421:   85 c0                   test   %eax,%eax
 423:   74 13                   je     438 <deregister_tm_clones+0x38>
 425:   55                      push   %ebp
 426:   89 e5                   mov    %esp,%ebp
 428:   83 ec 14                sub    $0x14,%esp
 42b:   51                      push   %ecx
 42c:   ff d0                   call   *%eax
 42e:   83 c4 10                add    $0x10,%esp
 431:   c9                      leave
 432:   c3                      ret
 433:   90                      nop
 434:   8d 74 26 00             lea    0x0(%esi,%eiz,1),%esi
 438:   f3 c3                   repz ret
 43a:   8d b6 00 00 00 00       lea    0x0(%esi),%esi
```

2. objdump -d nmain.out (5) register_tm_clones (a)

```
00000440 <register_tm_clones>:
 440:   e8 a4 00 00 00          call   4e9 <__x86.get_pc_thunk.dx>
 445:   81 c2 97 1b 00 00       add    $0x1b97,%edx
 44b:   55                      push   %ebp
 44c:   8d 8a 2c 00 00 00       lea    0x2c(%edx),%ecx
 452:   8d 82 2c 00 00 00       lea    0x2c(%edx),%eax
 458:   29 c8                   sub    %ecx,%eax
 45a:   89 e5                   mov    %esp,%ebp
 45c:   53                      push   %ebx
 45d:   c1 f8 02                sar    $0x2,%eax
 460:   89 c3                   mov    %eax,%ebx
 462:   83 ec 04                sub    $0x4,%esp
 465:   c1 eb 1f                shr    $0x1f,%ebx
 468:   01 d8                   add    %ebx,%eax
 46a:   d1 f8                   sar    %eax
```

2. objdump -d nmain.out (6) register_tm_clones (b)

```
46c:   74 14                   je     482 <register_tm_clones+0x42>
46e:   8b 92 20 00 00 00       mov    0x20(%edx),%edx
474:   85 d2                   test   %edx,%edx
476:   74 0a                   je     482 <register_tm_clones+0x42>
478:   83 ec 08                sub    $0x8,%esp
47b:   50                      push   %eax
47c:   51                      push   %ecx
47d:   ff d2                   call   *%edx
47f:   83 c4 10                add    $0x10,%esp
482:   8b 5d fc                mov    -0x4(%ebp),%ebx
485:   c9                      leave
486:   c3                      ret
487:   89 f6                   mov    %esi,%esi
489:   8d bc 27 00 00 00 00    lea    0x0(%edi,%eiz,1),%edi
```

2. objdump -d nmain.out (7) __do_global_dtors_aux

```
00000490 <__do_global_dtors_aux>:
 490:   55                      push   %ebp
 491:   89 e5                   mov    %esp,%ebp
 493:   53                      push   %ebx
 494:   e8 57 ff ff ff         call   3f0 <__x86.get_pc_thunk.bx>
 499:   81 c3 43 1b 00 00      add    $0x1b43,%ebx
 49f:   83 ec 04               sub    $0x4,%esp
 4a2:   80 bb 2c 00 00 00 00   cmpb   $0x0,0x2c(%ebx)
 4a9:   75 27                  jne    4d2 <__do_global_dtors_aux+0x42>
 4ab:   8b 83 14 00 00 00      mov    0x14(%ebx),%eax
 4b1:   85 c0                  test   %eax,%eax
 4b3:   74 11                  je     4c6 <__do_global_dtors_aux+0x36>
 4b5:   83 ec 0c               sub    $0xc,%esp
 4b8:   ff b3 28 00 00 00      pushl  0x28(%ebx)
 4be:   e8 dd fe ff ff         call   3a0 <__cxa_finalize@plt>
 4c3:   83 c4 10               add    $0x10,%esp
 4c6:   e8 35 ff ff ff         call   400 <deregister_tm_clones>
 4cb:   c6 83 2c 00 00 00 01   movb   $0x1,0x2c(%ebx)
 4d2:   8b 5d fc               mov    -0x4(%ebp),%ebx
 4d5:   c9                     leave
 4d6:   c3                     ret
 4d7:   89 f6                  mov    %esi,%esi
 4d9:   8d bc 27 00 00 00 00   lea    0x0(%edi,%eiz,1),%edi
```

2. objdump -d nmain.out (8) frame_dummy

```
000004e0 <frame_dummy>:
 4e0:   55                      push   %ebp
 4e1:   89 e5                   mov    %esp,%ebp
 4e3:   5d                      pop    %ebp
 4e4:   e9 57 ff ff ff          jmp    440 <register_tm_clones>
```

2. objdump -d nmain.out (9) __x86_get_pc_thunk.dx

```
000004e9 <__x86.get_pc_thunk.dx>:
 4e9:   8b 14 24                mov    (%esp),%edx
 4ec:   c3                      ret
```

2. objdump -d nmain.out (10) main

```
000004ed <main>:
 4ed:   8d 4c 24 04             lea    0x4(%esp),%ecx
 4f1:   83 e4 f0                and    $0xfffffff0,%esp
 4f4:   ff 71 fc                pushl  -0x4(%ecx)
 4f7:   55                      push   %ebp
 4f8:   89 e5                   mov    %esp,%ebp
 4fa:   53                      push   %ebx
 4fb:   51                      push   %ecx
 4fc:   e8 18 00 00 00          call   519 <__x86.get_pc_thunk.ax>
 501:   05 db 1a 00 00          add    $0x1adb,%eax
 506:   89 c3                   mov    %eax,%ebx
 508:   e8 30 00 00 00          call   53d <doAlmostNothing>
 50d:   b8 00 00 00 00          mov    $0x0,%eax
 512:   59                      pop    %ecx
 513:   5b                      pop    %ebx
 514:   5d                      pop    %ebp
 515:   8d 61 fc                lea    -0x4(%ecx),%esp
 518:   c3                      ret
```

2. objdump -d nmain.out (11)
__x86_get_pc_thunk.ax

```
00000519 <__x86.get_pc_thunk.ax>:
 519:   8b 04 24                mov    (%esp),%eax
 51c:   c3                      ret
```

2. objdump -d nmain.out (12) doNothingStatic

```
0000051d <doNothingStatic>:
 51d:   55                      push   %ebp
 51e:   89 e5                   mov    %esp,%ebp
 520:   e8 f4 ff ff ff          call   519 <__x86.get_pc_thunk.ax>
 525:   05 b7 1a 00 00          add    $0x1ab7,%eax
 52a:   90                      nop
 52b:   5d                      pop    %ebp
 52c:   c3                      ret
```

2. objdump -d nmain.out (13) doNothing

```
0000052d <doNothing>:
 52d:   55                      push   %ebp
 52e:   89 e5                   mov    %esp,%ebp
 530:   e8 e4 ff ff ff          call   519 <__x86.get_pc_thunk.ax>
 535:   05 a7 1a 00 00          add    $0x1aa7,%eax
 53a:   90                      nop
 53b:   5d                      pop    %ebp
 53c:   c3                      ret
```

2. objdump -d nmain.out (14) doAlmostNothing

```
0000053d <doAlmostNothing>:
 53d:   55                      push   %ebp
 53e:   89 e5                   mov    %esp,%ebp
 540:   e8 d4 ff ff ff          call   519 <__x86.get_pc_thunk.ax>
 545:   05 97 1a 00 00          add    $0x1a97,%eax
 54a:   e8 ce ff ff ff          call   51d <doNothingStatic>
 54f:   e8 d9 ff ff ff          call   52d <doNothing>
 554:   90                      nop
 555:   5d                      pop    %ebp
 556:   c3                      ret
 557:   66 90                   xchg   %ax,%ax
 559:   66 90                   xchg   %ax,%ax
 55b:   66 90                   xchg   %ax,%ax
 55d:   66 90                   xchg   %ax,%ax
 55f:   90                      nop
```

2. objdump -d nmain.out (15) __libc_csu_init (a)

```
00000560 <__libc_csu_init>:
 560:   55                      push   %ebp
 561:   57                      push   %edi
 562:   56                      push   %esi
 563:   53                      push   %ebx
 564:   e8 87 fe ff ff          call   3f0 <__x86.get_pc_thunk.bx>
 569:   81 c3 73 1a 00 00       add    $0x1a73,%ebx
 56f:   83 ec 0c                sub    $0xc,%esp
 572:   8b 6c 24 28             mov    0x28(%esp),%ebp
 576:   8d b3 04 ff ff ff       lea    -0xfc(%ebx),%esi
 57c:   e8 d7 fd ff ff          call   358 <_init>
 581:   8d 83 00 ff ff ff       lea    -0x100(%ebx),%eax
 587:   29 c6                   sub    %eax,%esi
 589:   c1 fe 02                sar    $0x2,%esi
 58c:   85 f6                   test   %esi,%esi
 58e:   74 25                   je     5b5 <__libc_csu_init+0x55>
```

2. objdump -d nmain.out (16) __libc_csu_init (b)

```
590:   31 ff                    xor    %edi,%edi
592:   8d b6 00 00 00 00        lea    0x0(%esi),%esi
598:   83 ec 04                 sub    $0x4,%esp
59b:   55                       push   %ebp
59c:   ff 74 24 2c              pushl  0x2c(%esp)
5a0:   ff 74 24 2c              pushl  0x2c(%esp)
5a4:   ff 94 bb 00 ff ff ff     call   *-0x100(%ebx,%edi,4)
5ab:   83 c7 01                 add    $0x1,%edi
5ae:   83 c4 10                 add    $0x10,%esp
5b1:   39 fe                    cmp    %edi,%esi
5b3:   75 e3                    jne    598 <__libc_csu_init+0x38>
5b5:   83 c4 0c                 add    $0xc,%esp
5b8:   5b                       pop    %ebx
5b9:   5e                       pop    %esi
5ba:   5f                       pop    %edi
5bb:   5d                       pop    %ebp
5bc:   c3                       ret
5bd:   8d 76 00                 lea    0x0(%esi),%esi
```

2. objdump -d nmain.out (13) __libc_csu_init (a)

```
00000560 <__libc_csu_init>:
 560:   55                      push   %ebp
 561:   57                      push   %edi
 562:   56                      push   %esi
 563:   53                      push   %ebx
 564:   e8 87 fe ff ff          call   3f0 <__x86.get_pc_thunk.bx>
 569:   81 c3 73 1a 00 00       add    $0x1a73,%ebx
 56f:   83 ec 0c                sub    $0xc,%esp
 572:   8b 6c 24 28             mov    0x28(%esp),%ebp
 576:   8d b3 04 ff ff ff       lea    -0xfc(%ebx),%esi
 57c:   e8 d7 fd ff ff          call   358 <_init>
 581:   8d 83 00 ff ff ff       lea    -0x100(%ebx),%eax
 587:   29 c6                   sub    %eax,%esi
 589:   c1 fe 02                sar    $0x2,%esi
 58c:   85 f6                   test   %esi,%esi
 58e:   74 25                   je     5b5 <__libc_csu_init+0x55>
 590:   31 ff                   xor    %edi,%edi
 592:   8d b6 00 00 00 00       lea    0x0(%esi),%esi
 598:   83 ec 04                sub    $0x4,%esp
 59b:   55                      push   %ebp
 59c:   ff 74 24 2c             pushl  0x2c(%esp)
 5a0:   ff 74 24 2c             pushl  0x2c(%esp)
 5a4:   ff 94 bb 00 ff ff ff    call   *-0x100(%ebx,%edi,4)
 5ab:   83 c7 01                add    $0x1,%edi
```