

## Zad 2.

sobota, 13 maja 2023 14:28

**Zadanie 2.** Poniżej zamieszczono uproszczony wynik kompilacji plików «start.c», «odd.c» i «even.c» do assemblera. W wygenerowanych plikach wskaż miejsca występowania definicji symboli i referencji do symboli. Czemu assembler nie może wygenerować ostatecznego ciągu bajtów reprezentujących instrukcje «call» i «jmp»? Zweryfikuj to wyświetlając zdeasembrowany kod przy pomocy «objdump -d». Jakie informacje assembler zostawia w plikach relokowalnych, żeby konsolidator mógł uzupełnić te instrukcje? Przypomnij uczestnikom zajęć jakie są główne zadania pełnione przez konsolidator – posłuż się w tym celu **mapą konsolidacji** z pliku «start.map». Pokaż, że konsolidator uzupełnił wymienione instrukcje w pliku wykonywalnym «start».

```
1 .text
2 .globl start
3 .type start, @function
4 _start:
5 pushq %rax
6 movl $42, %edi
7 call is_even
8 movl %eax, %edi
9 movl $60, %eax
10 syscall
11 popq %rdx
12 ret
13 .size start, .-start
```

start

```
1 .text
2 .globl is_odd
3 .type is_odd, @function
4 is_odd:
5 testq %rdi, %rdi
6 je .L2
7 decq %rdi
8 jmp is_even
9 .L2:
10 xorl %eax, %eax
11 ret
12 .size is_odd, .-is_odd
```

odd

```
1 .text
2 .globl is_even
3 .type is_even, @function
4 is_even:
5 testq %rdi, %rdi
6 je .L2
7 decq %rdi
8 jmp is_odd
9 .L2:
10 movl $1, %eax
11 ret
12 .size is_even, .-is_even
```

even

**start** - def

**is\_even** - ref

- 1) nie może, bo nie zna adresu, linker to uzupełnia
- 2) zostawia rekordy relokacji

```
młuczynski@młuczynski:~/Desktop/studia/ask/Lista 9/lista_9$ objdump -d start.o
start.o:      file format elf64-x86-64

Disassembly of section .text:

0000000000000000 <_start>:
 0: 50                push    %rax
 1: bf 2a 00 00 00    mov     $0x2a,%edi
 6: e8 00 00 00 00    call   b <_start+0xb>
 b: 89 c7            mov     %eax,%edi
 d: b8 3c 00 00 00    mov     $0x3c,%eax
12: 0f 05            syscall
14: 5a              pop     %rdx
15: c3              ret
```

```

mluczynski@mluczynski:~/Desktop/studia/ask/Lista 9/lista_9$ objdump -d even.o
even.o:      file format elf64-x86-64

Disassembly of section .text:

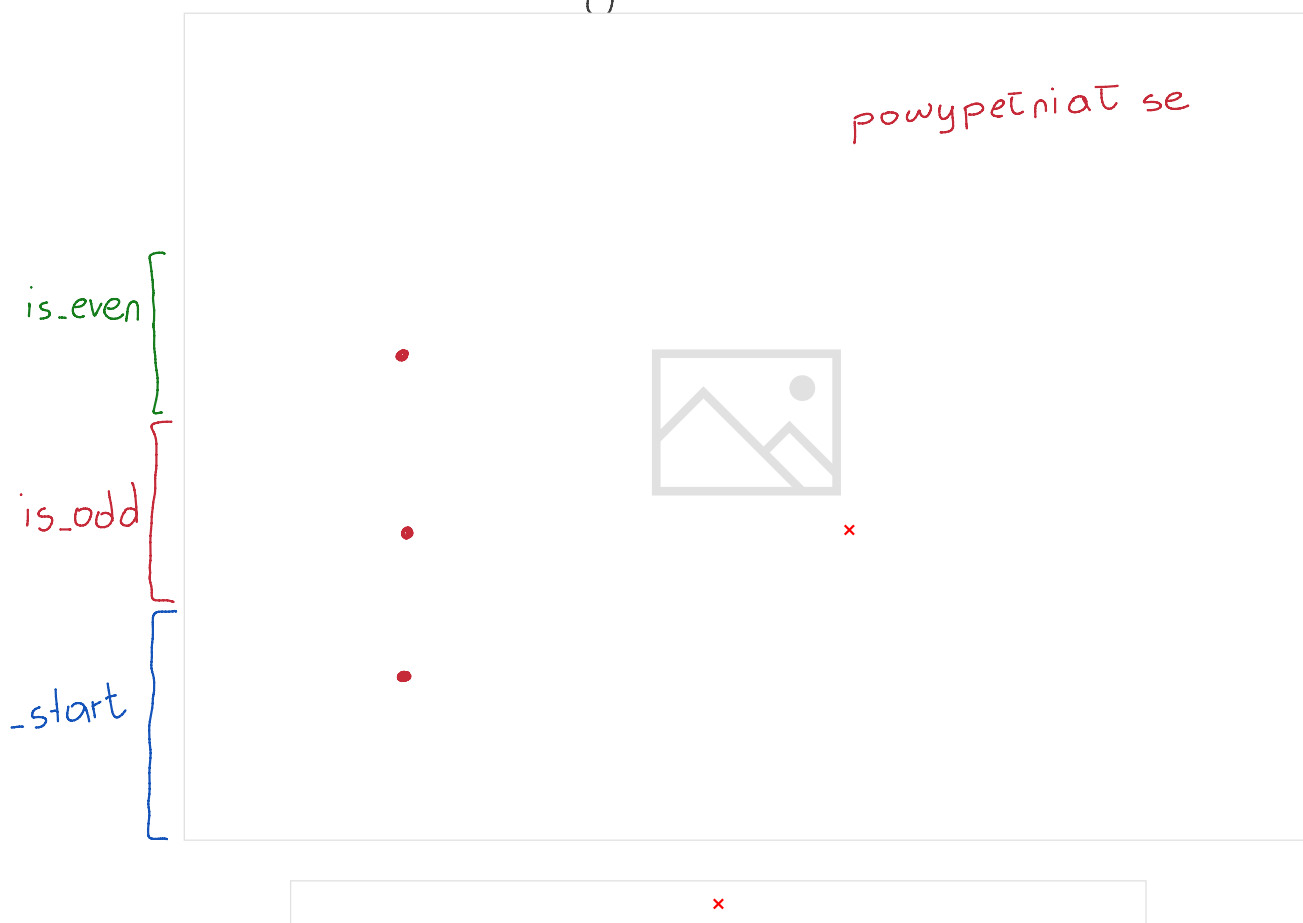
0000000000000000 <is_even>:
 0:  48 85 ff          test    %rdi,%rdi
 3:  74 08             je     d <is_even+0xd>
 5:  48 ff cf          dec     %rdi
 8:  e9 00 00 00 00    jmp     d <is_even+0xd>
 d:  b8 01 00 00 00    mov     $0x1,%eax
12:  c3               ret

```

3) mapa relokacji - plik wiążący adresy (offsety) w pliku wykonywalnym do danych i funkcji

4) główne zadania konsolidatora

- fuzja plików relokowalnych
- rozwiązywanie odwołań do symboli
- relokacja
- łączenie sekcji





entry pointy funkcji w .map