Zad 6.

sobota, 6 maja 2023 13:37

Zadanie 6. Wydrukuj tablice **rekordów relokacji** z sekcji «.rel.text» i «.rel.data» pliku «swap.o» przy pomocy polecenia «readelf -r». Na podstawie [1, §7.7.1] wytłumacz uczestnikom zajęć składowe **rekordów relokacji** «Elf64_Rela» [2, 4-23]. Wyjaśnij jak na podstawie *tablicy rekordów relokacji* polecenie «objdump -d -r swap.o» identyfikuje w zdeasemblowanym kodzie miejsca, które konsolidator będzie musiał uzupełnić w trakcie generowania pliku wykonywalnego. Czy możliwe jest by asembler utworzył sekcję «.rel.bss»?

```
mluczynski@mluczynski:~/Desktop/studia/ask/Lista 8/lista_8$ readelf -r swap.o

Relocation section '.rela.text' at offset 0x300 contains 8 entries:
  Offset          Info           Type           Sym. Value    Sym. Name + Addend
000000000006  000100000002 R_X86_64_PC32     0000000000000000 .bss - 5
00000000000f  000500000002 R_X86_64_PC32     0000000000000000 .LC0 - 4
000000000017  000100000002 R_X86_64_PC32     0000000000000000 .bss + 4
00000000001f  000100000002 R_X86_64_PC32     0000000000000000 .bss + 4
000000000026  000200000002 R_X86_64_PC32     0000000000000000 .LC1 - 4
000000000030  000f00000004 R_X86_64_PLT32    0000000000000000 printf - 4
000000000049  001100000002 R_X86_64_PC32     0000000000000000 bufp0 - 4
000000000053  001200000002 R_X86_64_PC32     0000000000000000 buf - 4

Relocation section '.rela.data.rel' at offset 0x3c0 contains 1 entry:
  Offset          Info           Type           Sym. Value    Sym. Name + Addend
000000000000  001200000001 R_X86_64_64       0000000000000000 buf + 0
```

rekordy relokacji zawierają informacje niezbędne do powiązania referencji symbolu z jego adresem w ostatecznym pliku

czyli instrukcje dotyczące odpowiednich modyfikacji swoich sekcji

```
typedef struct {
        Elf64_Addr      r_offset;
        Elf64_Xword     r_info;
        Elf64_Sxword    r_addend;
} Elf64_Rela;
```

lokalizacja miejsca, w którym potrzebna jest relokacja (względem początku sekcji)

indeks symbolu, którego dotyczy relokacja + typ relokacji

stała potrzebna do wyliczenia wartości, która ma być wstawiona w pole, którego dotyczy relokacja

objdump identyfikuje te miejsca na podstawie pól r_offset

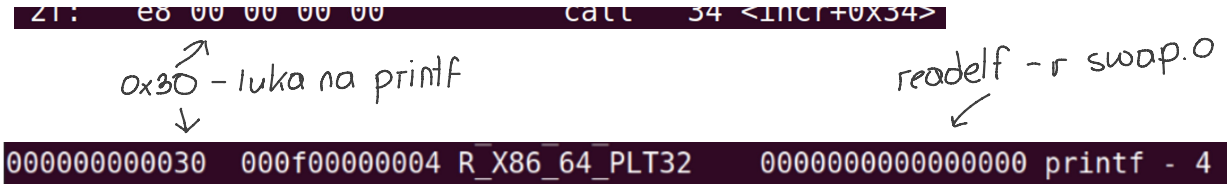
tutaj to fajnie widać

objdump -d swap.o

```
2f:   e8 00 00 00 00          call   34 <incr+0x34>
```

0x30 - luka na printf

readelf -r swap.o



assembler może wygenerować sekcje .rel.bss

https://stackoverflow.com/questions/37055896/what-does-an-elf-relocation-in-bss-but-relative-to-bss-mean

Step 2. Relocation. Compilers and assemblers generate code and data sections that start at address 0. The linker *relocates* these sections by associating a memory location with each symbol definition, and then modifying all of the references to those symbols so that they point to this memory location. The linker blindly performs these relocations using detailed instructions, generated by the assembler, called *relocation entries*.