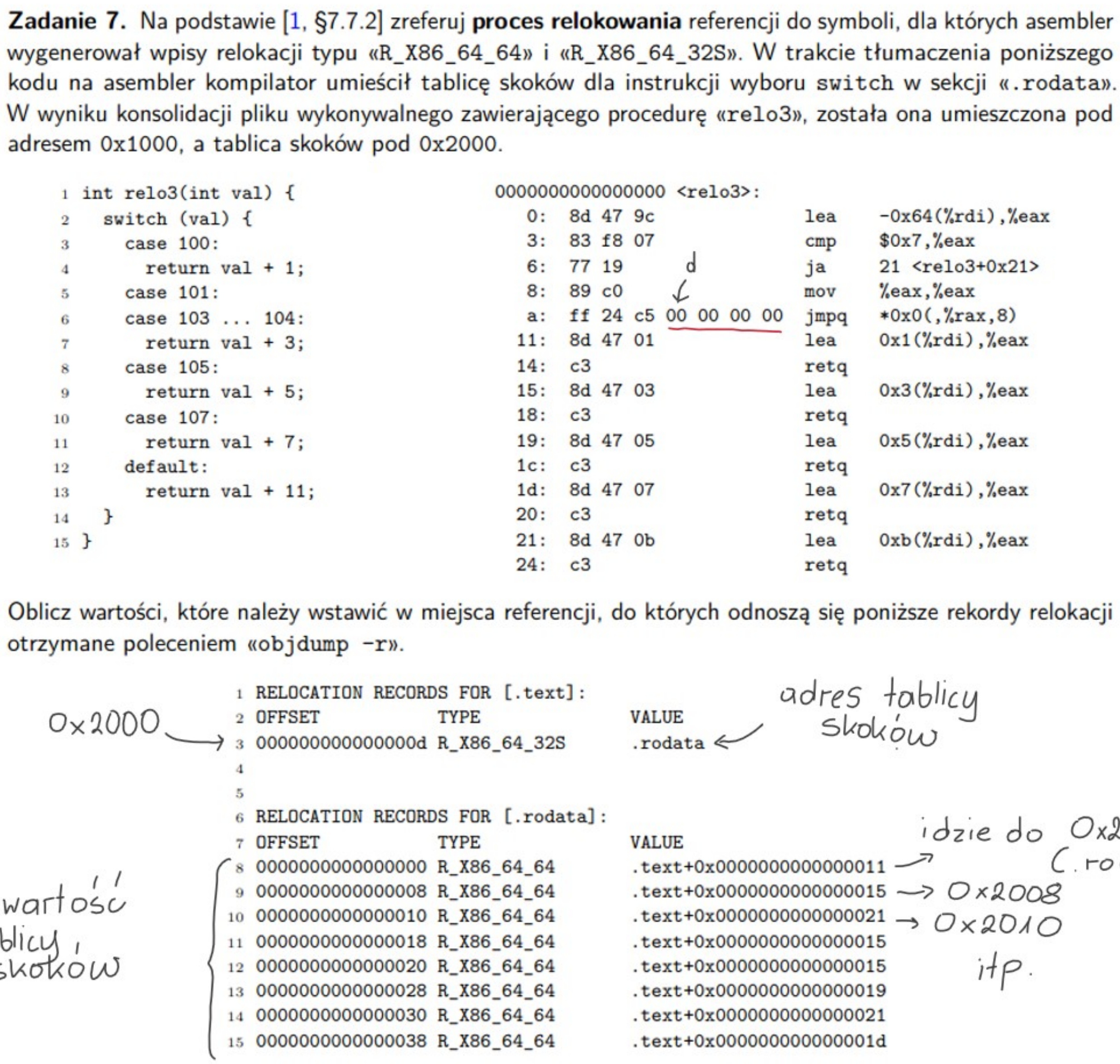
Zad 7.

sobota, 6 maja 2023 14:21



Zadanie 7. Na podstawie [1, §7.7.2] zreferuj **proces relokowania** referencji do symboli, dla których asembler wygenerował wpisy relokacji typu «R_X86_64_64» i «R_X86_64_32S». W trakcie tłumaczenia poniższego kodu na asembler kompilator umieścił tablicę skoków dla instrukcji wyboru switch w sekcji «.rodata». W wyniku konsolidacji pliku wykonywalnego zawierającego procedurę «relo3», została ona umieszczona pod adresem 0x1000, a tablica skoków pod 0x2000.

```
int relo3(int val) {
  switch (val) {
    case 100:
      return val + 1;
    case 101:
    case 103 ... 104:
      return val + 3;
    case 105:
      return val + 5;
    case 107:
      return val + 7;
    default:
      return val + 11;
  }
}
```

```
0000000000000000 <relo3>:
   0:  8d 47 9c              lea    -0x64(%rdi),%eax
   3:  83 f8 07              cmp    $0x7,%eax
   6:  77 19                 ja     21 <relo3+0x21>
   8:  89 c0                 mov    %eax,%eax
   a:  ff 24 c5 00 00 00 00  jmpq   *0x0(,%rax,8)
  11:  8d 47 01              lea    0x1(%rdi),%eax
  14:  c3                    retq
  15:  8d 47 03              lea    0x3(%rdi),%eax
  18:  c3                    retq
  19:  8d 47 05              lea    0x5(%rdi),%eax
  1c:  c3                    retq
  1d:  8d 47 07              lea    0x7(%rdi),%eax
  20:  c3                    retq
  21:  8d 47 0b              lea    0xb(%rdi),%eax
  24:  c3                    retq
```

d

Oblicz wartości, które należy wstawić w miejsca referencji, do których odnoszą się poniższe rekordy relokacji otrzymane poleceniem «objdump -r».

```
RELOCATION RECORDS FOR [.text]:
OFFSET           TYPE              VALUE
000000000000000d R_X86_64_32S      .rodata


RELOCATION RECORDS FOR [.rodata]:
OFFSET           TYPE              VALUE
0000000000000000 R_X86_64_64       .text+0x0000000000000011
0000000000000008 R_X86_64_64       .text+0x0000000000000015
0000000000000010 R_X86_64_64       .text+0x0000000000000021
0000000000000018 R_X86_64_64       .text+0x0000000000000015
0000000000000020 R_X86_64_64       .text+0x0000000000000015
0000000000000028 R_X86_64_64       .text+0x0000000000000019
0000000000000030 R_X86_64_64       .text+0x0000000000000021
0000000000000038 R_X86_64_64       .text+0x000000000000001d
```

0x2000 →

adres tablicy skoków

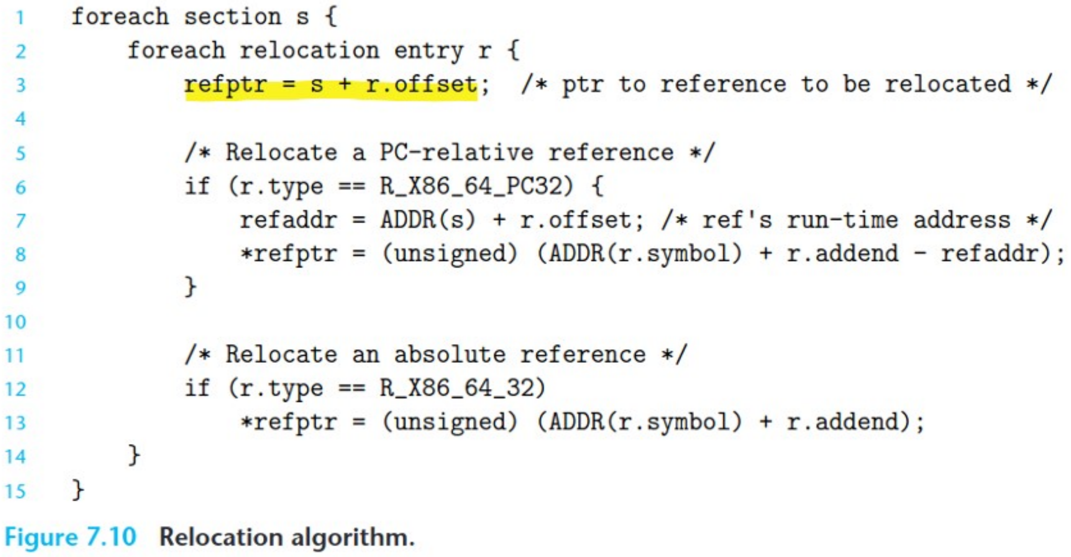
idzie do 0x2000 (.rodata)

→ 0x2008

→ 0x2010

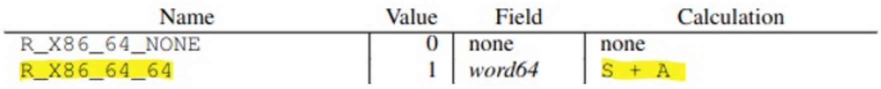
itp.

zawartość tablicy skoków

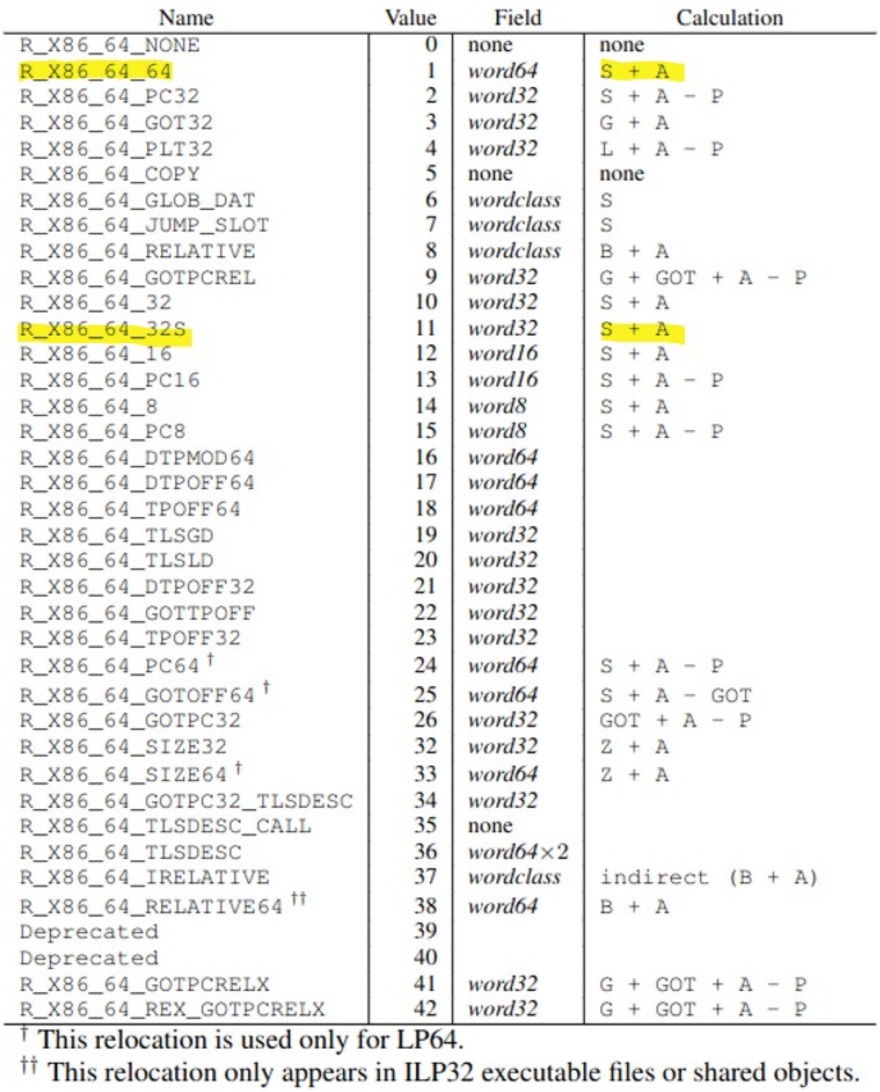


```
foreach section s {
    foreach relocation entry r {
        refptr = s + r.offset;  /* ptr to reference to be relocated */

        /* Relocate a PC-relative reference */
        if (r.type == R_X86_64_PC32) {
            refaddr = ADDR(s) + r.offset; /* ref's run-time address */
            *refptr = (unsigned) (ADDR(r.symbol) + r.addend - refaddr);
        }

        /* Relocate an absolute reference */
        if (r.type == R_X86_64_32)
            *refptr = (unsigned) (ADDR(r.symbol) + r.addend);
    }
}
```

Figure 7.10 Relocation algorithm.



Name	Value	Field	Calculation
R_X86_64_NONE	0	none	none
R_X86_64_64	1	*word64*	S + A



Name	Value	Field	Calculation
R_X86_64_NONE	0	none	none
R_X86_64_64	1	*word64*	S + A
R_X86_64_PC32	2	*word32*	S + A - P
R_X86_64_GOT32	3	*word32*	G + A
R_X86_64_PLT32	4	*word32*	L + A - P
R_X86_64_COPY	5	none	none
R_X86_64_GLOB_DAT	6	*wordclass*	S
R_X86_64_JUMP_SLOT	7	*wordclass*	S
R_X86_64_RELATIVE	8	*wordclass*	B + A
R_X86_64_GOTPCREL	9	*word32*	G + GOT + A - P
R_X86_64_32	10	*word32*	S + A
R_X86_64_32S	11	*word32*	S + A
R_X86_64_16	12	*word16*	S + A
R_X86_64_PC16	13	*word16*	S + A - P
R_X86_64_8	14	*word8*	S + A
R_X86_64_PC8	15	*word8*	S + A - P
R_X86_64_DTPMOD64	16	*word64*	
R_X86_64_DTPOFF64	17	*word64*	
R_X86_64_TPOFF64	18	*word64*	
R_X86_64_TLSGD	19	*word32*	
R_X86_64_TLSLD	20	*word32*	
R_X86_64_DTPOFF32	21	*word32*	
R_X86_64_GOTTPOFF	22	*word32*	
R_X86_64_TPOFF32	23	*word32*	
R_X86_64_PC64 †	24	*word64*	S + A - P
R_X86_64_GOTOFF64 †	25	*word64*	S + A - GOT
R_X86_64_GOTPC32	26	*word32*	GOT + A - P
R_X86_64_SIZE32	32	*word32*	Z + A
R_X86_64_SIZE64 †	33	*word64*	Z + A
R_X86_64_GOTPC32_TLSDESC	34	*word32*	
R_X86_64_TLSDESC_CALL	35	none	
R_X86_64_TLSDESC	36	*word64*×2	
R_X86_64_IRELATIVE	37	*wordclass*	indirect (B + A)
R_X86_64_RELATIVE64 ††	38	*word64*	B + A
Deprecated	39		
Deprecated	40		
R_X86_64_GOTPCRELX	41	*word32*	G + GOT + A - P
R_X86_64_REX_GOTPCRELX	42	*word32*	G + GOT + A - P

† This relocation is used only for LP64.

†† This relocation only appears in ILP32 executable files or shared objects.

S Represents the value of the symbol whose index resides in the relocation entry.

A Represents the addend used to compute the value of the relocatable field.