Architektury systemów komputerowych

Lista zadań nr 5

Na zajęcia 26–27 marca 2018

UWAGA! W zadaniu 1 można używać wyłącznie poniższych instrukcji:

- transferu danych: `mov` `cbw/cwde/cdqe` `cwd/cdq/cqo` `movzx` `movsx`,
- arytmetycznych: `add` `adc` `sub` `sbb` `imul` `mul` `idiv` `div` `idiv` `inc` `dec` `neg` `cmp`,
- logicznych: `and` `or` `xor` `not` `sar` `sarx` `shr` `shrx` `shl` `shlx` `ror` `rol` `test`,
- innych: `lea` `ret`.

Przy tłumaczeniu kodu w asemblerze `x86-64` do języka `C` należy trzymać się następujących wytycznych:

- Używaj złożonych wyrażeń minimalizując liczbę zmiennych tymczasowych.
- Nazwy wprowadzonych zmiennych muszą opisywać ich zastosowanie, np. `result` zamiast `rax`.
- Instrukcja `goto` jest zabroniona. Należy używać instrukcji sterowania `if`, `for`, `while` i `switch`.
- Należy dążyć do przetłumaczenia pętli `while` na pętle `for`.

Zadanie 1. Zaimplementuj funkcję zdefiniowaną poniżej w asemblerze `x86-64`. Taka procedura w języku `C` miałaby sygnaturę «`long cmp(uint64_t x, uint64_t y)`».

$$cmp(x, y) = \begin{cases} -1 & \text{gdy } x < y \\ 1 & \text{gdy } x > y \\ 0 & \text{gdy } x = y \end{cases}$$

Wskazówka: Rozwiązanie wzorcowe ma cztery wiersze (bez `ret`). Użyj instrukcji `adc`, `sbb` i `neg`.

Zadanie 2. Poniżej zamieszczono kod procedury o sygnaturze «`long puzzle2(char *s, char *d)`». Wyznacz **bloki podstawowe** oraz narysuj **graf kontroli przepływu**. Przetłumacz tę procedurę na język `C`, a następnie jednym zdaniem powiedz co ona robi.

```
1  puzzle2:
2        movq  %rdi, %rax
3  .L3:  movb  (%rax), %r9b
4        leaq  1(%rax), %r8
5        movq  %rsi, %rdx
6  .L2:  movb  (%rdx), %cl
7        incq  %rdx
8        testb %cl, %cl
9        je    .L4
10       cmpb  %cl, %r9b
11       jne   .L2
12       movq  %r8, %rax
13       jmp   .L3
14 .L4:  subq  %rdi, %rax
15       ret
```

Zadanie 3. Poniżej widnieje kod funkcji o sygnaturze «`uint32_t puzzle3(uint32_t n, uint32_t d)`». Wyznacz bloki podstawowe oraz narysuj graf kontroli przepływu, po czym przetłumacz tę funkcję na język `C`. Na podstawie ustępu „Mixing C and Assembly Language” strony GNU Assembler Examples[1] napisz program, który pomoże Ci powiedzieć co ta funkcja robi.

[1] `http://cs.lmu.edu/~ray/notes/gasexamples/`

```
puzzle3:
        movl  %edi, %edi
        salq  $32, %rsi
        movl  $32, %edx
        movl  $0x80000000, %ecx
        xorl  %eax, %eax
.L3:    addq  %rdi, %rdi
        movq  %rdi, %r8
        subq  %rsi, %r8
        js    .L2
        orl   %ecx, %eax
        movq  %r8, %rdi
.L2:    shrl  %ecx
        decl  %edx
        jne   .L3
        ret
```

Zadanie 4. Poniżej zamieszczono kod rekurencyjnej procedury o sygnaturze «`int puzzle4(long *a, long v, uint64_t s, uint64_t e)`». Wyznacz bloki podstawowe oraz narysuj graf kontroli przepływu. Przetłumacz tę procedurę na język `C`, a następnie jednym zdaniem powiedz co ona robi.

```
puzzle4:
        movq  %rcx, %rax
        subq  %rdx, %rax
        shrq  %rax
        addq  %rdx, %rax
        cmpq  %rdx, %rcx
        jb    .L5
        movq  (%rdi,%rax,8), %r8
        cmpq  %rsi, %r8
        je    .L10
        cmpq  %rsi, %r8
        jg    .L11
        leaq  1(%rax), %rdx
        call  puzzle4
.L10:   ret
.L11:   leaq  -1(%rax), %rcx
        call  puzzle4
        ret
.L5:    movl  $-1, %eax
        ret
```

Wskazówka: Z reguły procedurę «puzzle4» woła się następująco: «`i = puzzle4(a, v, 0, n - 1)`».

Zadanie 5. Poniżej widnieje kod procedury o sygnaturze «`long puzzle5(void)`». Podaj rozmiar i składowe rekordu aktywacji procedury «puzzle5». Jaką sygnaturę ma procedura «`readlong`»? Przetłumacz procedurę «puzzle5» na język `C` i wytłumacz jednym zdaniem co ona robi.

```
puzzle5:
        subq   $24, %rsp
        movq   %rsp, %rdi
        call   readlong
        leaq   8(%rsp), %rdi
        call   readlong
        movq   (%rsp), %rax
        cqto
        idivq  8(%rsp)
        xorl   %eax, %eax
        testq  %rdx, %rdx
        sete   %al
        addq   $24, %rsp
        ret
```

Zadanie 6. Poniższy kod w asemblerze otrzymano w wyniku deasemblacji funkcji zadeklarowanej jako «`long switch_prob(long x, long n)`». Zapisz w języku `C` kod odpowiadający tej funkcji.

```
400590 <switch_prob>:
400590: 48 83                   subq  $0x3c,%rsi
400594: 48 83 fe 05             cmpq  $0x5,%rsi
400598: 77 29                   ja    *0x4005c3
40059a: ff 24 f5 f8 06 40 00    jmpq  *0x4006f8(,%rsi,8)
4005a1: 48 8d 04 fd 00 00 00 00 lea   0x0(,%rdi,8),%rax
4005a9: c3                      retq
4005aa: 48 89 f8                movq  %rdi,%rax
4005ad: 48 c1 f8 03             sarq  $0x3,%rax
4005b1: c3                      retq
4005b2: 48 89 f8                movq  %rdi,%rax
4005b5: 48 c1 e0 04             shlq  $0x4,%rax
4005b9: 48 29 f8                subq  %rdi,%rax
4005bc: 48 89 c7                movq  %rax,%rdi
4005bf: 48 0f af ff             imulq %rdi,%rdi
4005c3: 48 8d 47 4b             leaq  0x4b(%rdi),%rax
4005c7: c3                      retq
```

Zrzut pamięci przechowującej tablicę skoków:

```
(gdb) x/6gx 0x4006f8
0x4006f8: 0x4005a1
0x400700: 0x4005a1
0x400708: 0x4005b2
0x400710: 0x4005c3
0x400718: 0x4005aa
0x400720: 0x4005bf
```

Zadanie 7. Procedurę ze zmienną liczbą parametrów używającą pliku nagłówkowego `stdarg.h`[2] skompilowano z opcjami «`-Og -mno-sse`». Po jej deasemblacji otrzymano następujący wydruk. Co robi ta procedura i jaka jest jej sygnatura? Jakie dane są przechowywane w rekordzie aktywacji tej procedury?

```
1  puzzle7:
2        movq  %rsi, -40(%rsp)
3        movq  %rdx, -32(%rsp)
4        movq  %rcx, -24(%rsp)
5        movq  %r8, -16(%rsp)
6        movq  %r9, -8(%rsp)
7        movl  $8, -72(%rsp)
8        leaq  8(%rsp), %rax
9        movq  %rax, -64(%rsp)
10       leaq  -48(%rsp), %rax
11       movq  %rax, -56(%rsp)
12       movl  $0, %eax
13       jmp .L2
14 .L3:  movq  -64(%rsp), %rdx
15       leaq  8(%rdx), %rcx
16       movq  %rcx, -64(%rsp)
17 .L4:  addq  (%rdx), %rax
18 .L2:  subq  $1, %rdi
19       js  .L6
20       cmpl  $47, -72(%rsp)
21       ja  .L3
22       movl  -72(%rsp), %edx
23       addq  -56(%rsp), %rdx
24       addl  $8, -72(%rsp)
25       jmp .L4
26 .L6:  ret
```

Wskazówka: Przeczytaj rozdział §3.5.7 dokumentu opisującego `ABI` dostępnego na stronie przedmiotu.

Zadanie 8. Poniżej zamieszczono kod procedury o sygnaturze «`struct T puzzle8(long *a, long n)`». Na jego podstawie podaj definicję typu «`struct T`». Przetłumacz tę procedurę na język `C`, po czym jednym zdaniem powiedz co ona robi.

```
1  puzzle8:
2        movq  %rdx, %r11
3        xorl  %r10d, %r10d
4        xorl  %eax, %eax
5        movq  $LONG_MIN, %r8
6        movq  $LONG_MAX, %r9
7  .L2:  cmpq  %r11, %r10
8        jge .L5
9        movq  (%rsi,%r10,8), %rcx
10       cmpq  %rcx, %r9
11       cmovg %rcx, %r9
12       cmpq  %rcx, %r8
13       cmovl %rcx, %r8
14       addq  %rcx, %rax
15       incq  %r10
16       jmp .L2
17 .L5:  cqto
18       movq  %r9, (%rdi)
19       idivq %r11
20       movq  %r8, 8(%rdi)
21       movq  %rax, 16(%rdi)
22       movq  %rdi, %rax
23       ret
```

Wskazówka: Zauważ, że wynik procedury nie mieści się w rejestrach `%rax` i `%rdx`.

[2] `https://en.wikipedia.org/wiki/Stdarg.h`