

# Architektury systemów komputerowych 2016

## Lista zadań nr 5

Na zajęcia 28 marca – 31 marca 2016

**Zadanie 1.** Zapisz w języku C funkcję zadeklarowaną jako `long loop(long x, int n)`, której kod w asemblerze jest przedstawiony na poniższym listingu:

```
1 loop:
2   movl  %esi,%ecx
3   xorq  %rax,%rax
4   leaq  1(%rax),%rdx
5   jmp   .L2
6 .L3:
7   movq  %rdi,%r8
8   andq  %rdx,%r8
9   orq   %r8,%rax
10  salq  %cl,%rdx
11 .L2:
12  testq %rdx,%rdx
13  jne   .L3
14  rep; retq
```

**Zadanie 2.** Przeczytaj poniższy kod w języku C i odpowiadający mu kod w asemblerze, a następnie wywnioskuj jakie są wartości stałych A i B.

1 typedef struct {	12 void set_val(str1 *p, str2 *q) {
2   int x[A][B];	13   long v1 = q->t;
3   long y;	14   long v2 = q->u;
4 } str1;	15   p->y = v1 + v2;
5	16 }
6 typedef struct {	17
7   char array[B];	18 set_val:
8   int t;	19   movslq 8(%rsi),%rax
9   short s[A];	20   addq   32(%rsi),%rax
10  long u;	21   movq   %rax,184(%rdi)
11 } str2;	22   ret

**Zadanie 3.** Przeczytaj poniższy kod w języku C i odpowiadający mu kod w asemblerze, a następnie wywnioskuj jakie są wartości stałych R, S i T.

1 long A[R][S][T];	9 store_elem:
2	10 leaq (%rsi,%rsi,2),%rax
3 long store_elem(long i, long j,	11 leaq (%rsi,%rax,4),%rax
4           long k, long *dest)	12 movq %rdi,%rsi
5 {	13 salq \$6,%rsi
6   *dest = A[i][j][k];	14 addq %rsi,%rdi
7   return sizeof(A);	15 addq %rax,%rdi
8 }	16 addq %rdi,%rdx
	17 movq A(,%rdx,8),%rax
	18 movq %rax,(%rcx)
	19 movq \$3640,%rax
	20 ret

**Zadanie 4.** Przeczytaj poniższy kod w języku C i odpowiadający mu kod w asemblerze, a następnie wywnioskuj jaka jest wartość stałej CNT i jak wygląda definicja struktury a\_struct.

```

1 typedef struct {
2     int first;
3     a_struct a[CNT];
4     int last;
5 } b_struct;
6
7 void test (long i, b_struct *bp) {
8     int n = bp->first + bp->last;
9     a_struct *ap = &bp->a[i];
10    ap->x[ap->idx] = n;
11 }
12 test:
13     movl    0x120(%rsi),%ecx
14     addl    (%rsi),%ecx
15     leaq    (%rdi,%rdi,4),%rax
16     leaq    (%rsi,%rax,8),%rax
17     movq    0x8(%rax),%rdx
18     movslq  %ecx,%rcx
19     movq    %rcx,0x10(%rax,%rdx,8)
20     retq

```

**Zadanie 5.** Przeczytaj poniższą definicję unii elem oraz kod procedury proc i odpowiedz na poniższe pytania.

```

1 union elem {
2     struct {
3         long *p;
4         long y;
5     } e1;
6     struct {
7         long x;
8         union elem *next;
9     } e2;
10 };
11 proc:
12     movq    8(%rdi),%rax
13     movq    (%rax),%rdx
14     movq    (%rdx),%rdx
15     subq    8(%rax),%rdx
16     movq    %rdx, (%rdi)
17     ret

```

- Jaki jest rozmiar unii elem w bajtach?
- Napisz funkcję w języku C, która odpowiada procedurze proc.

**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.

```

1 400590 <switch_prob>:
2 400590: 48 83                subq    $0x3c,%rsi
3 400594: 48 83 fe 05          cmpq    $0x5,%rsi
4 400598: 77 29                ja      *0x4005c3 <switch_prob+0x33>
5 40059a: ff 24 f5 f8 06 40 00 jmpq    *0x4006f8(,%rsi,8)
6 4005a1: 48 8d 04 fd 00 00 00 lea      0x0(,%rdi,8),%rax
7 4005a9: c3                  retq
8 4005aa: 48 89 f8             movq    %rdi,%rax
9 4005ad: 48 c1 f8 03          sarq    $0x3,%rax
10 4005b1: c3                  retq
11 4005b2: 48 89 f8             movq    %rdi,%rax
12 4005b5: 48 c1 e0 04          shlq    $0x4,%rax
13 4005b9: 48 29 f8             subq    %rdi,%rax
14 4005bc: 48 89 c7             movq    %rax,%rdi
15 4005bf: 48 0f af ff          imulq   %rdi,%rdi
16 4005c3: 48 8d 47 4b          leaq    0x4b(%rdi),%rax
17 4005c7: c3                  retq

```

```

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

```

**Zadanie 7.** Przeczytaj definicje struktur `strA` i `strB`, a następnie przeanalizuj kod w asemblerze funkcji zadeklarowanych jako: `strB process(strA s)` i `long eval(long x, long y, long z)`.

```

1 typedef struct {
2     long a[2];
3     long *p;
4 } strA;
5
6 typedef struct {
7     long u[2];
8     long q;
9 } strB;
10 process:
11     movq %rdi,%rax
12     movq 24(%rsp),%rdx
13     movq (%rdx),%rdx
14     movq 16(%rsp),%rcx
15     movq %rcx,(%rdi)
16     movq 8(%rsp),%rcx
17     movq %rcx,8(%rdi)
18     movq %rdx,16(%rdi)
19     retq
20 eval:
21     subq $104,%rsp
22     movq %rdx,24(%rsp)
23     leaq 24(%rsp),%rax
24     movq %rdi,(%rsp)
25     movq %rsi,8(%rsp)
26     movq %rax,16(%rsp)
27     leaq 64(%rsp),%rdi
28     call process
29     movq 72(%rsp),%rax
30     addq 64(%rsp),%rax
31     addq 80(%rsp),%rax
32     addq $104,%rsp
33     retq

```

Zapisz w języku C kod odpowiadający funkcjom `process` i `eval`. Narysuj diagram przedstawiający zawartość ramki stosu funkcji `eval`.