

Heimadæmi 6

Arnar Sigurðsson

1. a) 8 reikni- og gagnaflutningsaðgerðir

```
6   div2k:
7       .seh_endprologue
8       movl    %ecx, %eax
9       movl    %edx, %ecx
10      cltd
11      movl    $1, %r8d
12      sall    %cl, %r8d
13      subl    $1, %r8d
14      andl    %r8d, %edx
15      addl    %edx, %eax
16      sarl    %cl, %eax
17      ret
18      .seh_endproc
19      .def     __main; .scl    2; .type    32; .endef
20      .globl   main
21      .def     main; .scl    2; .type    32; .endef
22      .seh_proc main
```

- b) 15 reikni- og gagnaflutningsaðgerðir án hoppa. Svo eru hoppin L4 með 8 skipanir og L1 með 3 skipanir.

```
6   div2kif:
7       pushq   %rbx
8       .seh_pushreg %rbx
9       subq    $48, %rsp
10      .seh_stackalloc 48
11      movaps   %xmm6, 32(%rsp)
12      .seh_savexmm %xmm6, 32
13      .seh_endprologue
14      movl    %ecx, %ebx
15      pxor    %xmm6, %xmm6
16      cvtsi2sdl %edx, %xmm6
17      movapd   %xmm6, %xmm1
18      movsd    .LC0(%rip), %xmm0
19      call     pow
20      subsd    .LC1(%rip), %xmm0
21      cvtsd2ss %xmm0, %xmm0
22      testl   %ebx, %ebx
23      jg      .L4
24      pxor    %xmm1, %xmm1
25      cvtsi2ssl %ebx, %xmm1
26      divss    %xmm1, %xmm0
27      cvtss2sil %xmm0, %eax
28      .L1:
29      movaps   32(%rsp), %xmm6
30      addq    $48, %rsp
31      popq    %rbx
32      ret
33      .L4:
34      movapd   %xmm6, %xmm1
35      movsd    .LC0(%rip), %xmm0
36      call     pow
37      pxor    %xmm1, %xmm1
38      cvtsi2sdl %ebx, %xmm1
39      divsd    %xmm1, %xmm0
40      cvtsd2ss %xmm0, %xmm0
41      cvtss2sil %xmm0, %eax
42      jmp     .L1
43      .seh_endproc
44      .section .rdata,"dr"
45      .align 8
46      .LC0:
47      .long    0
48      .long    1073741824
49      .align 8
50      .LC1:
51      .long    0
52      .long    1072693248
53      .ident    "GCC: (GNU) 11.2.0"
54      .def     pow; .scl    2; .type    32; .endef
```

c)

2. a) %rax = 0x0123456789ABCD0A en 0A=10

b) Ólöglegt, %bx kemst ekki fyrir í %al.

c) %rax = 00000000000008765

d) Ólöglegt, er ekkert %bh.

e) %rax = 000000000000CDEF

3. incq -8(%rbx) -> hækkar um 1 í minni 0x208 og verður 1. ($0x210 - 8 \text{ dec} = 0x208$)

subq %rax, (%rbx) -> lækkar um 2 í minni 0x210 og verður E. ($0x10 - 2 \text{ dec} = 0xE$)

imulq %rax, 8(%rbx) -> margfaldar 1 í minni 0x218 með 2 í %rax. ($0x210 + 8 \text{ dec} = 0x218$)

addq -16(%rbx, %rax, 4), %rax -> Bætir við 1 sem er í minni 208 við 2 sem er í %rax og verður 3.
($0x216 + (0x2 + 4) - 16 = 0x208$)

við lok:

%rax = 0x2, %rbx = 0x210, 0x200 = 0x20, 0x208 = 0x3, 0x210 = 0xE, 0x218 = 0x2

4. not %edi -> not tekur bitwise andhverfuna af inntakinu, $n = \sim n$.

lea (%rdi, %rdi, 1) -> lea bætir (%rdi +%rdi) við %eax, eða tvöfaldar andhverfu inntaks.

and %edi, %eax -> and og-ar tvöföldun á andhverfunni við andhverfuna.

5. a) leaq (%rsi, %rsi, 2) %rax = %rax fær gildið $b + 2 * b = 3b$ svo %rax = 3b

b) addq %rax, %rdi = %rdi fær gildið úr %rax + gildið úr %rdi = $3b + a$

c) leaq (%rdi, %rdi, 4), %rdx = %rax fær gildið úr %rdx tvisvar = $30b + 10a$ svo %rax hefur gildið $30b + 10a$

d) ret skilar gildinu í rax sem er $30b + 10a$.

Mynd af C forriti:

```
C: > cygwin64 > home > ad > C comp.c
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  long comp(long a, long b) {
5      return 30*b + 10*a;
6  }
```