

## Heimadæmi 7

### TÖL309 Tölvutækni og forritun

Hjörvar Sigurðsson

1. .

hoho:

`leaq 31(%rdi), %rax` Lagt saman 31 og fyrsta viðfangfalls, köllum það  $x$ , og niðurstæða sett í viðfang `%rax`, köllum það  $r$ .  
 $r = 31 + x$ .  $r$  er skilagildi fallins.

`testq %rdi, %rdi` Setur  $CF=0$ ,  
 $OF=0$ ,  $ZF=1$  ef  $a \& b == 0$ ,  
 $SF=1$  ef  $a \& b < 0$ .

`cmovns %rdi, %rax` Ef  $\sim SF$ , þ.e. ef fyrri samanburðurinn leiddi í ljós að  $a \& b < 0$ , þá er gildi `%rdi`,  $x$ , sett í skilagildi fallins, `%rax`, eða  $r$ .

`sarq $5, %rax` Hliðrar `%rax`, eða  $r$ , til hægri um 5.

`ret` skilar `%rax`, eða  $r$ .

C kóði:

```
long hoho (long x) {
    long r = 31 + x;
    if ((x & x) >= 0) {
        r = x;
    }
    r = r >> 5;
    return r;
}
```

2.

whi:

movl \$0, %eax    %eax er 32-bit  
heiltölun útgáfa af skilagildi fallstíðs,  
kölum það r.  
r = 0.

jmp .L2    Stóðkoma í .L2.

.L3:

leaq -1(%rdi), %rdx    %rdi er fyrsta  
viðfang fallstíðs, köllum það x.  
Niðurstaða úr  $x + (-1)$  er sett í  
%rdx, sem er tímabundin  
gildi.

andq %rdx, %rdi    Samsvætur  
 $\%rdi = \%rdi \& \%rdx$ .

xorq \$1, %rax    Samsvætur  
 $\%rax = \%rax \wedge 1$ .

.L2:

testq %rdi, %rdi    Setur CF=0,  
OF=0, ZF=1 ef  $x \& x == 0$ ,  
SF=1 ef  $x \& x < 0$ .

jne .L3    Ef  $\sim ZF$ , þá stóðkoma  
í .L3.

ret    Skila skilagildi r.

C kóði:

```
int whi (long x) {
    int r = 0;                // Samsvarar movl $0, %eax.

    do {                      // Samsvarar .L3.
        x = x & (x-1);        // Samsvarar andq %rdx, %rdi.
                                // %rdx (eða x-1 í sviganum)
                                // samsvarar svo
                                // leaq -1(%rdi), (rdx).
        r = r | 1;            // Samsvarar xorq $1, %rax.
    }
    while ((x & x) != 0);      // Samsvarar jne .L3.

    return r;                  // Samsvarar ret.
}
```

3. `.L3:`  
`movl $1, %eax`  $\%eax = 1$   
`ret` Skilar  $\%eax$

$\%rdx = 3.$  viðfang  $1.$  ang  $\%rsi = 2.$  viðfang  
 $\%rdi =$  fyrsta viðfang  $= x$   
 $\%edi =$   $-1$  heiltalan  $-1$

func:

`cmpl $1, %edi` Set flags eftir  $\%edi = 1$ .

`jle .L3` Ef  $1 \leq x$ , þá stæðle  
`.L3:`

`pushq %rbx` Geymir  $\%rbx$  með þín að  
 setja það á staflann.

`leal (%rdi,%rdi,2), %ebx` Setur  
 $x + 2x$   $\%ebx$  sem  
 viðfang  $(x + 2x)$ .

`leal 3(%rdi), %edx` Setur  $\%edx$  sem  
 $x + 3$  viðfang  $(x + 3)$

`testl %edi, %edi`

`cmovns %edi, %edx` Ef  $(x \& x) \geq 0$ , þá  
 er  $\%edx = x$ . Þ.e.  $x$  er sett í  $\%edx$ .

`scall $2, %edx` 3. viðfang,  $\%edx$ , er  
 hlíðrand til margri um 3.

$\%edx = \%edx \gg 3$ .

$\%edx = \frac{\%edx}{2^3}$ .

`movl %edx, %edi`  $\%edi = \%edx$ ,  
 Þ.e.  $\%edx$  er sett  
 í  $\%edi$ .

`call func`

`addl %ebx, %eax`  $\%ebx$  geymir  
 gildi yfir fallsköfl.

$\%eax$  er skilafaldi  
 falls.

`popq %rbx`

$\%rbx$  er return  
 address.

`ret`

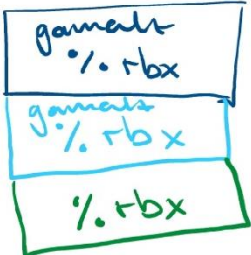


a)

C kóði:

```
int func(int n) {  
    if (n <= 1) {  
        return 1;  
    }  
  
    return (n + func(n >> 2));  
}
```

// Samsvarar `cmpl $1, %edi` og `jle .L3`.  
// Innihald if-segðarinnar samsvarar `.L3`.  
// Samsvarar `movl $1, %eax` og `ret`.  
  
// n samsvarar `cmovns %edi, %edx`.  
// n >> 2 samsvarar `sarl $2, %edx`.

b)

Fyrsta kalli.  pushq %rbx í fyrsta kalli.  
2. kalli  pushq %rbx í öðru kalli.  
3. kalli  pushq %rbx í þriðja kalli.

Síðan er poppað öllum þessum %rbx.

4.

a.

fact:

```
cmp    rdi, 1
```

Ber saman %rdi, eða fyrsta viðfang fallins, n, og 1. Stillir flags í kjölfarið.

```
jg     .L8
```

Stekkur í .L8 ef %rdi > 1.

```
mov    eax, 1
```

Setur 1 inn í %eax. Semsagt, %eax = 1.

```
Ret
```

Skilar skilagildi.

.L8:

```
push   rbx
```

Pushar %rbx á staflann til þess að geta farið aftur þangað þegar kalli er lokið.

```
mov    rbx, rdi
```

Færir gildi %rdi inn í %rbx.

```
lea    rdi, [rdi-1]
```

Reiknar vístfang (%rdi-1) og færir það inn í %rdi.

```
call   fact
```

Kallar á fallið fact.

<code>imul</code>	<code>rax, rbx</code>	Margfaldar %rax og %rbx, en %rax er útkoma kallsins „fact(n-1)“.
<code>pop</code>	<code>rbx</code>	Poppar rbx þar sem kallið á fallið er lokið, og við þurfum ekki að fara þangað aftur.
<code>ret</code>		Skilar skilagildi og fallskalli lýkur.
b.		
<code>fact:</code>		
<code>mov</code>	<code>eax, 1</code>	Færir 1 inn í %eax. Þ.e. %eax = 1.
<code>cmp</code>	<code>rdi, 1</code>	Ber saman %rdi og 1, og stillir flags í kjölfarið.
<code>jle</code>	<code>.L1</code>	Ef %eax <= 1, þá er stokkið í .L1.
<code>.L2:</code>		
<code>mov</code>	<code>rdx, rdi</code>	%rdi er sett inn í %rdx.
<code>sub</code>	<code>rdi, 1</code>	Dregið er 1 frá %rdi og útkoman sett í %rdi. Þ.e. %rdi = %rdi - 1.
<code>imul</code>	<code>rax, rdx</code>	%rax og %rdx eru margfölduð saman.
<code>cmp</code>	<code>rdi, 1</code>	Ber saman %rdi og 1, og stillir flags í kjölfarið.
<code>jne</code>	<code>.L2</code>	Stekkur í .L2 ef ZF=0, en ZF var stillt út frá samanburði %rdi og 1.
<code>.L1:</code>		
<code>ret</code>		Skilar skilagildi og fallskalli lýkur.

Seinni útgáfan er líklega fljótari þar sem ekki þarf að pusha og poppa á staflann.