

TÖL309G Tölvutækni og forritun

Heimadæmi 10

Hjörvar Sigurðsson

1. Náði ekki alveg í einni línu.

a.

```
int AnyBit1(int x) {  
    for (int i = 0; i < sizeof(x); i++) {  
        if ((x >> i) & 0x01) {  
            return 1;  
        }  
    }  
    return 0;  
}
```

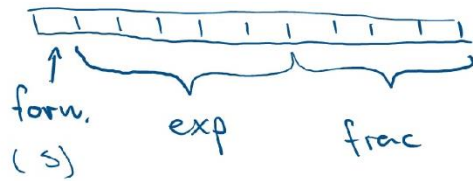
b.

```
int AnyBit0(int x) {  
    for (int i = 0; i < sizeof(x); i++) {  
        if ((x >> i) & 0x00) {  
            return 1;  
        }  
    }  
    return 0;  
}
```

c.

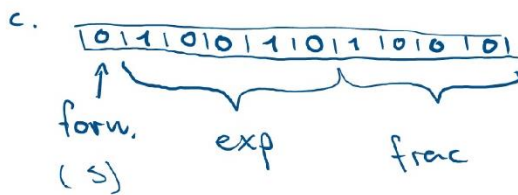
```
int ThirdBit1(int x) {  
    if ((x >> 3) & 1) {  
        return 1;  
    }  
    return 0;  
}
```

2.



a. $\text{Bias} = 2^{k-1} - 1$ | k er fj. bitar í exp.
 $= 2^{5-1} - 1 = 16 - 1 = \underline{\underline{15}}$

b.
 $v = (-1)^s (M) (2^{\text{exp-bias}})$ | M vísar til Frac hlutans,
 t.d. $M = 1 + \frac{1}{2} + \frac{1}{8}$ ef
 Frac hlutinn er 101.

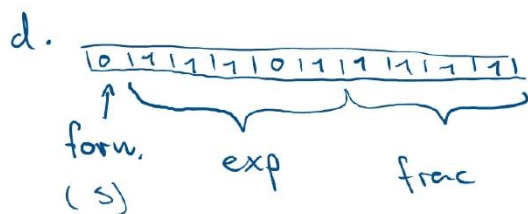


$\text{Bias} = 16$

$V = (-1)^s (M) (2^{\text{exp-bias}})$

$= (-1)^0 (1 + \frac{1}{2}) (2^{18-16})$

$= 1.5 (2^2) = 1.5 (4) = \underline{\underline{6}}$



Talan sem verið er að hármerken
má lýsa með eftirfarandi
jöfnu:

$$(-1)^s (M) (2^{\text{exp-bias}})$$

Þetta segir okkur að s skal
vera 0, og M og exp skulu
vera eins stór og hægt er.

Fyrsti bitinn (s -bitinn) þarf
að vera 0 til þess að talan
sé jákvæð.

Exp bitarnir skulu allir vera
1 til þess að fá sem
hastu tölu $2^{\text{exp-bias}}$,
nema síðasti (minnsti) exp
bitinn, þar sem að ef
allir exp bitarnir eru 1, þá
tákna talan inf eða NaN
(hvort það er far eftir frac).

Frac bitarnir skulu líka
allir vera 1 til þess að
fá sem hæst M .

3.

a. Breytan í gisti %rdi hefur gagnatagið long, en við vitum það þar sem q-ið í addq skipuninni í línu 6 segir okkur að um sé að ræða fjórorð (8 bæti).

b. .

Skipunin er „for (%edx = 0; %edx < %esi; %edx = %edx + 2)“

%edx er upphafsstillt sem 0 í línu 1.

Samanburðurinn í línu 8 og stökk-skipunin í línu 9 segir okkur að lykkjan er endurtekin ef %esi > %edx.

Lagt er 2 við %edx í línu 7.

c. Fallið tekur inn tvær breytur, a og b; ítrar frá 0 til b; og bætir a+8(i) við skilagildið í hverri ítrun. Að lokum skilar það svo skilagildinu.

Sauðakóði:

```
huh (%rdi, %rsi)
```

```
{
```

```
%edx = 0;
```

```
%eax = 0;
```

```
%rcx = %edx;
```

```
%rax = %rax + %rdi + 8(%rcx);
```

```
%edx = %edx + 2;
```

Eða

```
huh(a, b) {
```

```
    sum = 0;
```

```
    for (i = 0; i < b; i = i + 2){
```

```
        sum = sum + a + 8(i);
```

```
    }
```

```
}
```

4.

- a. main.c:
b: veikt.
main(): sterkt.

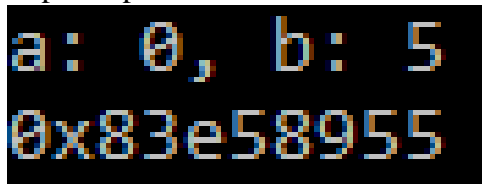
fall.c:
a: sterkt.
b: sterkt.
main: veikt.
fall(): sterkt.

- b. Hún prentar
a: 0, b: 5

Fyrir a prentar hún 0 þar sem að %d gefur í skyn að a sé int, en a-gildið í fall.c er short int. Því prentar hún útkomu main() fallsins.

Fyrir b prentar hún 5 þar sem %d gefur í skyn að b sé int, en b-gildið í fall.c er short int. Því prentar hún b-gildið úr main-fallinu í main.c.

- c. .
Skipunin prentar



Í objdump skjalinu má sjá eftirfarandi í main fallinu:

```
00401460 <_main>:
401460: 55                push    %ebp
401461: 89 e5             mov     %esp,%ebp
401463: 83 e4 f0          and     $0xffffffff0,%esp
401466: e8 95 05 00 00    call   401a00 <__main>
40146b: c7 05 04 40 40 00 05 movl    $0x5,0x404004
401472: 00 00 00
401475: e8 0a 00 00 00    call   401484 <_fall>
40147a: b8 00 00 00 00    mov     $0x0,%eax
40147f: c9                leave
401480: c3                ret
401481: 90                nop
401482: 90                nop
401483: 90                nop
```

Skipunin prentar semsagt út fyrstu fjögur bætín í vélarkóða main-fallsins.

5.

a. main.c

void fall();	ytri breyta geymd í .bss svæði
static int a = 1;	staðvær breyta geymd í .data svæði
int b = 2;	víðvær breyta geymd í .data svæði
extern int c;	víðvær breyta geymd í .bss svæði
extern int d;	víðvær breyta geymd í .bss svæði
int main() {	víðvær breyta geymd í .bss svæði
int c = 3;	staðvær breyta geymd á hlaða
}	

fall.c

int a;	víðvær breyta geymd í .bss svæði
extern int b;	víðvær breyta geymd í .bss svæði
int c = 4;	víðvær breyta geymd í .data svæði
int d = 5;	víðvær breyta geymd í .data svæði

void fall() {	víðvær breyta geymd í .bss svæði
static int d;	staðvær breyta geymd í .bss svæði

a = 6;	víðvær breyta geymd á hlaða
b = 7;	víðvær breyta geymd á hlaða
c = 8;	víðvær breyta geymd á hlaða
d = 9;	víðvær breyta geymd á hlaða
}	

b. main.c

int b = 2;	sterkt
extern int c;	veikt
extern int d;	veikt
int main() {	sterkt

fall.c

int a;	veikt
extern int b;	veikt
int c = 4;	sterkt
int d = 5;	sterkt

void fall() {	sterkt
a = 6;	sterkt
b = 7;	sterkt
c = 8;	sterkt
d = 9;	sterkt
}	

c. Úttak:

```
a:1, b:7  
c:3, d:5
```

a – það prentar út static int a breytuna í main.c.

b – það prentar út extern int b breytuna úr fall.c þar sem fall() var lesið á undan main(), en því var ekki hægt að segja „int b = 2“ þar sem b var nú þegar skilgreind breyta með gildið 7.

c – það prentar út int c = 3 gildið í main() fallinu, þar sem hún er í fallinu sjálfu.

d – það prentar út int d = 5 gildið í fall.c þar sem það er skilgreint sem extern í main.c.