Tölvutækni og Forritun Heimadæmi 4

brj46

September 2024

1

Gefinn er eftirfarandi kóði:

```
int x = 5, y;
x *= y = 2;
x = y == 2;
x == (y = 3);
x = (unsigned)-3 > 0 ? y << 1 : y >> 1;
```

förum í gegnum kóðan og útskýrum hvernig gildið á **x** breytist:

1. `int x = 5, y;`

   - x er frumstillt með gildinu 5.
   - y er skilgreind en ekki frumstillt.

   `x = 5`

2. `x *= y = 2;`

   - Fyrst er y sett sem 2 `y = 2`.
   - Síðan er x margfaldað með y: `x = x * y` sem er $5 \times 2 = 10$.

   `x = 10`

3. `x = y == 2;`

   - Hér er verið að skoða hvort y sé jafnt og 2`y == 2`.
   - Þar sem y er 2, er skilyrðið satt.
   - satt er táknað sem 1, þannig að x fær gildið 1.

   `x = 1`

4. `x == (y = 3);`

   - Fyrst er y sett sem 3 með `y = 3`.
   - Síðan er boolean, `x == y` þar sem það er bara skoðað hvort x sé jafny og y þá helst x óbreytt.

   `x = 1`

5.

```
x = (unsigned)-3 > 0 ?  y < < 1 :  y > > 1;
```

- Fyrst er `-3` breytt í unsigned heiltölu
- Síðan er athugað hvort sú tala sé stærri en 0
- Það er satt svo þá er `y < < 1` valin.
- `y < < 1` þýðir að bitarnir í `y` eru færðir eitt skref til vinstri sem er bara sinnum 2.
- Þar sem `y = 3` fáum við `3 < < 1 = 6`.
- `x` fær gildið 6.

```
x = 6
```

$$\begin{aligned} \mathtt{x} &= 5 \quad \text{(eftir línu 1)} \\ \mathtt{x} &= 10 \quad \text{(eftir línu 2)} \\ \mathtt{x} &= 1 \quad \text{(eftir línu 3)} \\ \mathtt{x} &= 1 \quad \text{(eftir línu 4)} \\ \mathtt{x} &= 6 \quad \text{(eftir línu 5)} \end{aligned}$$

2

a)

Hver er stærsta talan sem hægt er að tákna án formerkis þegar unnið er með 6 bita heiltölur:
Svar: Það er $2^6 - 1$ eða 63.

b)

Hverjar eru stærstu og minnstu tvíandhverfu heiltölur sem hægt er að tákna (þ.e. $TMax_6$ og $TMin_6$)?

Svar $Tmax_6$**:** er þá $2^{6-1} - 1$ eða 31
Svar $Tmin_6$**:** er þá -2^{6-1} eða -32

c)

Framkvæmið eftirfarandi samlagningu sem 6-bita tvíandhverfu samlagningu og sýnið niðurstöðuna sem tugatölu: 0x17 + 0x21.

Svar: Reiknum $0x17 + 0x21 = 0x38 = 56$ talan 56 er ekki inn í bilinu 31 til -32 svo það er yfirflæði. Reiknum það með $56 - 64 = -8$ svo lokaniðurstaðan er -8

d)

Við ætlum að stýfa (truncate) 6-bita tvíandhverfutöluna -9 niður í 4 bita. Sýnið bitaútgáfu af stýfingunni. Breytist gildi tölunnar?

Svar: Táknum -9 sem 6-bita tvíandhverfu: $-9 = 110111$ þegar við stýfum það niður í 4 bita fáum við 0111 ef við færum það yfir í tugatölu þá fáum við 7 svo já þá breytist talan þar sem hún veldur einskonar yfirflæði.

3

Við erum með kóðabútinn

```
int hmm(unsigned int n) {
    return (~n) & (~n << 1);
}
```

a)

Rekjum okkur í gegnum fallið með gildinu

i) $n = 01010101$
$\sim n = 10101010$
$\sim n << 1 = 01010100$
$(\sim n)\&(\sim n << 1) = 10101010\&01010100 = 00000000$
Við fáum út 0 þar sem engir samliggjandi 0-bitar eru til staðar:

ii) $n = 10010010$
$\sim n = 01101101$
$\sim n << 1 = 11011010$ $(\sim n)\&(\sim n << 1) = 0110110111011010 = 01001000$ þar sem fallið skilar ekki 0 það þýðir að það fundust samligjandi 0-bitar

b)

Fallið notar neitun og vinstri hliðrun og AND-ar þeim síðan saman.

1. neitar og breytir öllum 0 í 1 og öfugt
2. hliðrar þessu gildi eitt sæti til vinstri
3. Nú er AND-að þessu saman og ber þá saman hvort tölurnar eru eins og skilar 1 ef þær eru eins annars 0 og ef allar tölurnar eru 0 þá eru engir samliggjandi -0 bitar.

c)

Skrifum fallið sem leitar eftir þrem samliggjandi 0-bitum

```
int hmm2(unsigned int n) {
    return (~ n) & (~n << 1 ) & (~n << 2);
}
```

d)

skrifum fall sem ákvarðar hvort það séu tveir samliggjandi 1-bitar

```
int 1hmm ( unsigned int n) {
    return n & (n << 1);
}
```

Í þessu dæmi þá hliðrum við bara n um 1 og AND-um það saman við n og þá sjáum við að ef það eru tveir 1- bitar samliggjandi þá mun fallið ekki skila núlli.

4

a)

Skrifum einnar línu fall sem skilar sammengi tveggja mengja:

```
unsigned long sammengi(unsigned long a, unsigned long b) {
    return a | b;
}
```

b)

Skrifum einnar línu fall fall sem skilar mengjamun mengjanna a og b

```
unsigned long munur(unsigned long a, unsigned long b) {
    return a & ~b;
}
```

c)

Hvaða mengi táknar segðin $1ul << i$? segðin $1ul << i$ táknar mengi þar aðeins stak i er í menginu

$1ul$ er 64 bita heiltalan með 1 á lægsta bitanum. þegar við hliðrum $<< i$ um i sæti til vinstri fáum við bitastreng sem biti i er 1 og allir aðrir bitar eru "0"

dæmi: $(1ul << 3)$ táknar mengi 3 því bitastrengurinn verður 000000...1000

d)

Skrifum einnar línu fall sem skilar því hvort stak i er í menginu a:

```
int erStakI(unsigned long a, int i){
    return (a & (1ul <<i)) != 0;
}
```

ef þetta fall skilar ekki núll þá er i í mengi a.

5

a)

sínum hvernig hliðrun -69 um 3 sæti (deiling með 8) sýni ekki rétta niðurstöðu og sínum síðan hvernig skal gera það rétt:

táknum -69 á two's complement bitaformi:

$$-69 = 10111011$$

Hliðrum -69 um 3

$$10111011- > 11110111 = -9$$

við sjáum að við fáum út -9 en við myndum vilja að talan yrði rúnuð upp að 0

leiðréttum með $2^k - 1$ þar sem k er fjöldi sæta sem við hliðrum um

$$2^3 - 1 = 7$$

$$-69 + 7 = -62$$

Táknum -62 á two's complement og hliðrum um 3:

$$11000010 >> 3 = 11111000 = -8$$

við sjáum að hér fengum við rétta niðurstöðu.

b)

Sýnum núna að það gengur ekki að bara bæta bjöguninni við allar tölur áður en þeim er hliðrað.

Tökum til dæmis 69:

$$69 = 01000101$$

bætum við bjöguninni:

$$69 + 7 = 76 = 01001100$$

Hliðrum um 3:

$$01001100 >> 3 = 00001001 = 9$$

en þar sem $69/8 er 8.65$ og við viljum að tölur rúnnist að 0 þá er þetta vitlaust hér hefðum við frekar viljað fá 8 sem rétta svar.