Tölvutækni og Forritun Heimadæmi 10

Ragnar Björn Ingvarsson, rbi3

2. nóvember 2024

1

a) Við byrjum með `0x2051` og framkvæmum svo `0x2051 | 12` (or-að við 12) sem gefur, ef við breytum yfir í tvíundakerfið, `0010000001010001 | 1100` svo við fáum `0010000001011101` sem, ef við svo framkvæmum `>> 3` (hliðrum um 3 til vinstri) á það fæst `10000001011` eða `0x40b`.

b) i. Við sjáum að $13 = 1101_2$ og þá rennum við einu sinni í gegn og þá er `v = 1100`, rennum svo aftur og `v = 1000` og svo loks rennum við einu sinni enn og fæst `v = 0000` og við hættum. Svo í lokin er `v = 0` og `c = 3`.

ii. Almennt sjáum við að `v = 0` í lokin þar sem lykkjan hættir bara þegar `v` gefur falsgildið 0. Svo verður `c` bara fjöldi ása í tvíundartölunni fyrir `v`.

2

Sjáum að `i` er í `rdi`, `j` er í `rsi` og `k` er í `rdx`. Fyrst er `k` fært í skilagildi og við byrjum á samanburði á `i` og `j` og framkvæmum línu `8` ef sannur, annars hoppum við beint í annan samanburð sem aftur framkvæmir bara eina línu, `10`, ef sannur. Svo kóðinn er svona

```
int bla(int i, int j, int k) {
        if (i >= j) {
            j = i;
        }
        if (j >= k) {
            return j;
        }
        return k;
}
```

3

a) Byrjum á `main.c`, `fun` er veikt, `b` er veikt vegna `extern`, `c` er sterkt og `main` er sterkt.

Í `fun.c` er `a` sterkt, `b` líka sterkt, `c` veikt vegna `extern` og `fun` er sterkt.

b) Það prentast út að a sé 0, b sé 3 og c sé 5. Þetta er líklega vegna þess að þegar við setjum `b = 3` þá er það að virka sem `int` í stað `short int` svo breytan í `fun.c` fær neðri partinn, en efri breytir óvart `short int a` í staðinn og `a` verður þá 0.

4

a) Kyrrleg tenging getur verið betri þar sem óþarfi er að hafa öll söfnin sem forrit notar á tölvunni, þ.e. allt sem þarf fyrir forritið kemur með keyrsluskránni. Einnig veldur þetta því að kyrrleg tenging er fljótari þar sem ekki þarf að tengja söfn þegar forritinu er hlaðið í minni eða í miðri keyrslu, allt er nú þegar komið þegar forritið er þýtt.

b) Kvik tenging er oft betri vegna þess t.d. að keyrsluskrárnar verða miklu minni vegna þess að ekki þarf að pakka öllum notkunum í söfnum saman við forritið. Einnig er auðveldara að uppfæra söfn þar sem ekki þarf að endurþýða öll forrit sem nota það í hvert skipti sem það er uppfært. Svo geta líka mörg forrit notað sama minni fyrir safnið í einu.

5

a) Við sjáum að hér kemur villa þar sem eftir kallið á `f()` þá breytist `x` ekki í `a1.c` og helst sem 2, í stað þess að breytast í 4. Þetta er vegna þess að þó við skilgreinum `extern int x` sem tengist við `x`-ið í `a2.c` þá er þessi skilgreining veik og þess vegna notum við frekar `x`-ið skilgreint í `int x = 2` þar sem það er sterk skilgreining sem tekur forgang yfir hina skilgreininguna.

b) Hér er engin villa þar sem eina skilgreiningin á `y` í `b1.c` er `int y = 10` sem er sterk og við vísum í hana frá `b2.c` með `extern int y`. Þess vegna er engin tvískilgreining og allt er hreint og skýrt og virkar eins og við búumst við, `y` byrjar sem 10 og prentast út, verður svo 12 og prentast aftur út.