

Mikroprosessorit, demo 3

30.9 ja 2.10.2009 A2065

Ensimmäisten mikroluokkaharjoitusten tarkoituksena on tutustua käytettävään simulaattoriin ja tehdä sillä muutama lyhyt ohjelma.

Simulointitehtävät voi tehdä myös kotona, mutta ne täytyy demota harjoitusten vetäjälle. Kaikkien simulaatiodemojen yhteinen deadline on 16.10. Tämän kerran tehtäviä voi siis jatkaa ensi viikolla.

Tenttioikeuden saamiseksi simulaatioharjoituksista pitää tehdä 50%.

Tehtävä 1.

- Lataa MARS-simulaattori osoitteesta
<http://courses.missouristate.edu/KenVollmar/MARS/>
ja talleta mars.jar-tiedosto työpöydälle.
- Käynnistä ohjelma tuplaklikkaamalla.
- Kirjoita demojen 1 tehtävässä 2 oleva ohjelma Edit-välilehdelle.
- Talleta tiedosto ja käännä ohjelmasi. Mars Messages -välilehdellä alhaalla näet virheilmoitukset. Tuotettu koodi näkyy Execute-välilehdellä.
- Kirjoita rekisteriin \$a0 Registers-välilehdellä arvoksi 0x10010000. Kirjoita Execute-välilehden Data Segment -kohdassa muistipaikkoihin 0x10010000-0x10010008 luvut 5, 6 ja 7.
- Suorita ohjelma (simuloi) run-nappulalla.
- Mikä on rekisterin \$v0 arvo suorituksen jälkeen ?

Tehtävä 2.

Kirjoita MIPS-ohjelma joka toteuttaa seuraavan pseudokoodin:

```
while (memory[a]<10) {  
    memory[a]++  
    memory[b]*=2  
}
```

Memory[a] tarkoittaa tässä muistiosoitetta 0x10010000 ja memory[b] muistiosoitetta 0x10010004. Käytä kertolaskuun (*=) rotaatiota. Kirjoita muistipaikkaan 0x10010004 luku 1. Testaa ohjelmaasi. Mitä muistissa on ohjelman suorituksen jälkeen ?

Tehtävä 3.

Tee ohjelma joka lukee lukuja muistista ja tallettaa ne pinoon. Lukeminen päättyy kun luetaan nolla. Suorituksen jälkeen rekisterissä \$v0 on lukujonon kolmanneksi viimeinen luku. Voidaan olettaa että lukuja on ainakin neljä.

Laita ohjelmakoodiisi muistin alustusta varten:

```
.text  
  
la $s0, omajono  
  
#ohjelma tähän  
  
.data  
omajono:  
-1, 0, 1, 2, 4, 6, 8, 10
```

Tehtävä 4.

Tutustu aliohjelmien käyttöön. Täydennä puuttuvat rivit ja käskyt.
Mitä ohjelma tekee ?

```
.text

        la $s0, omatluvut      # _____
        _____           # ladataan $s1:een...
        _____           # ...arvo 0x80000000 (32 bittiä)
loop:    ____ $a0, ($s0)        # luetaan muistista $a0:aan
        beq $a0, $zero, finish # jos $a0 nolla, lopetetaan
        slt $t0, $s1, $a0      # jos 0x80000000 pienempi kuin $a0...
        beq $t0, $zero, skip   # ...ohitetaan aliohjelmakutsu
        ____ proc              # aliohjelmakutsu, parametri $a0
        ____ $v0, ($s0)        # talletetaan aliohjelman tulos ($v0)
skip:    addi $s0, $s0, 4       # muistiosoitetta kasvatetaan
        j loop                 # uusi loop

proc:    ____ $v0, $a0, $zero   # aliohjelma kääntää bitit...
        ____ $v0, $v0, 1       # ja lisää ykkösen
        ____ $ra

finish:

.data
omatluvut:
.word -2, -1, 2
```

Pohdittavaa: Voisiko aliohjelmat toteuttaa pelkästään hyppykäskyillä ?