

# 09.71.35 Stýrikerfi I

Lokapróf

16. desember 1998

kl. 9<sup>00</sup>-13<sup>00</sup>

Öll *skrifleg* hjálpargögn og reiknivél leyfileg.

**1. [10%]** Flestir nútíma diskar eru ekki með sama fjölda geira á öllum sporum disksins, heldur er sporunum skipt upp í nokkur svæði og innan hvers svæðis er sami fjöldi geira á öllum sporum.

a) Hvernig breytist fjöldi geira eftir svæðum disksins (þ.e. á hvaða svæðum eru flestir geirar og á hverjum eru færstir)? Rökstyðjið svar ykkar. (Ath. ekki er ætlast til að þið vitið svarið fyrirfram. Einkunin mun byggja á hversu sannfærandi rökstuðningur ykkar er).

b) Ef við gerum ráð fyrir að gagnamagn í hverjum geira sé fast (t.d. 512 bæti), hvaða áhrif hefur þessi breytilegi fjöldi þá á *i*) sóknartöf (e. seek time), *ii*) flutningstíma (e. transfer time) og *iii*) leitartíma (e. access time)?

**2. [10%]** a) Einn af kostunum við 2's complement framsetninguna er að hægt er að leggja saman tvær slíkar tölur án tillits til þess hvort þær eru jákvæðar eða neikvæðar og fá rétta niðurstöðu. En hvað með margföldun 2's complement talna? Rökstyðjið svar ykkar.

b) Sýnið að hægt er að skilgreina 2's complement andhverfu  $N$ -bita tölunnar  $x$  sem  $2^N - x$ .

**3. [10%]** Í þessu dæmi ætlum við að nota 16 bita undir fleytitölur á IEEE formi. 16-bitu IEEE fleytitölur eru ekki til, svo við þurfum sjálf að ákveða hve marga bita á að nota undir brothluta og veldishluta (formerkishluti er alltaf einn bit!).

a) Hver þarf skiptingin bitanna á milli veldishluta og brothluta að vera til að hægt sé að tákna töluna 2001.0 nákvæmlega?

b) Hversu margar skiptingar virka, þannig að hægt sé að tákna töluna 2001.0 nákvæmlega?

c) Er til skipting þannig að hægt sé að tákna töluna 100000.0 nákvæmlega? Útskýrið.

**4. [25%]** Fallið hér að neðan stokkar  $n$ -staka heiltöluvektorinn  $A$ . Þið eigið að skrifa fallið sem 8086-smalamálsfall, sem kallanlegt er úr Borland C++.

```
void stokka( int *A, int n) {
    randomize();
    for(int i=n-1; i>0; i--)
        vixla( A[i], A[slm(i+1)] );
}
```

Vektorinn  $A$  inniheldur 16-bitu heiltölugildi. Stefið `randomize` finnur upphafssæði fyrir slembitölugjafa, yfirleitt út frá klukku tölvunnar. Þið megið ná í 16-bitu gildi í minnishólfinu með nafnið `klukka` og nota það sem upphafssæði. Fallið `slm(n)` skilar 16 bitu slembitölu

á bilinu 0 til  $n-1$ . Þið getið gert ráð fyrir að hafa smalamálsfallið `slembi`, sem tekur inn sæðisgildi í `AX` og skilar næstu slembitölu í `AX` (hún er síðan notuð sem næsta sæðisgildi). Þið þurfið hins vegar að skrifa `vixla` sjálf. Það þarf ekki að vera sérstakt stef, heldur megið þið skrifa það beint inn í fallið `stokka`.

**5. [10%]** a) Hér að neðan er forritsbútur til að afrita `CX` bæti frá `DS:SI` til `ES:DI`. Útskýrið nákvæmlega tilgang hvernar skipunnar

```
shr      cx, 1
rep movsw
adc      cx, cx
rep movsb
```

b) Útskýrið hvað eftirfarandi forritsbútur gerir. (Vísbending: Heildarstærð skipananna frá `L1` niður að enda bútsins er 20 bæti)

```
L1:      lea    di, L1
mov      cx, 10
mov      bx, di
L2:      mov   ax, [cs:di]
mov      [cs:di+100], ax
add      di, 2
loop    L2
add      di, 80
jmp     bx
```

**6. [10%]** Skrifið eftirfarandi fall sem 8086-smalamálsfall, kallanlegt úr Borland C++ og notið Intel fleytitöluskipanir. Skilagildi falls sem er fleytitala er komið fyrir í efsta staki fleytitölustaflans (þ.e. `st(0)`).

```
float norm(float x, float y, float z) {
    float t = sqrt(x*x + y*y + z*z);
    if (t < EPSILON)
        return 0.0;
    else
        return t;
}
```

Gerið ráð fyrir að gildin `EPSILON` og `0.0` séu víðværar breytur.

**7. [10%]** Þýðið eftirfarandi C-forritsbút yfir í PA-RISC smalamál. Þið megið nota breytunöfnin sem nöfn á gistum (t.d. `%a` fyrir `a`).

```
if( (a>b) && (a == (c==d)) )
    b = 0;
else
    c = 1;
```

**8. [15%]** Tölva hefur 48 bita vistföng, 16KB innra skyndiminni sem er 4-vítt mengistengið (e. set associative) með línustærð 32 bæti.

a) Lýsið þessu skyndiminni nánar (fjöldi lína, fjöldi mengja, hvaða bitar í vistfangi gegna hvaða hlutverki, hvaða línur í minni fara í hvaða mengi í skyndiminni, ...).

b) Tölvun hefur líka ytra skyndiminni, sem er 1MB að stærð. Það notar beina vörpun (e. direct mapping) og línustærð 128 bæti. Lýsið því hvað gerist þegar örgjörvinn biður um bætið í 48-bitu vistfanginu (00 02 A1 7C 10 45), sem er hvorki í innra né ytra skyndiminni þegar beðið er um það. (Gerið ráð fyrir því að bæti þurfi að vera í ytra skyndiminni til að hægt sé að hlaða því í innra skyndiminni).

c) Sýnið annað vistfang sem varpast í sama sæti ytra skyndiminnis og sama mengi innra skyndiminnis og vistfangið í b)-lið, án þess þó að vera í sömu línu og það.