

09.71.35 Stýrikerfi I

Lokapróf

18. desember 1997

kl. 9⁰⁰-13⁰⁰

Öll *skrifleg* hjálpargögn leyfileg.

1. [10%] Tiltekinn harður diskur hefur 10 plötur, þar sem skrifað er á allar hliðar. Það eru 1648 spor, hver með 128 geira, sem hver hefur 512 bæti. Snúningshraði disksins er 7200 sn/mín og það tekur 1.5 msek að færa leskaus á milli samliggjandi spora. Einnig gefur framleiðandinn upp að meðalsóknartími (seek time) sé 7 msek.

- Hver er stærð disksins í MB?
- Hver er meðalaðgangstími (access time) disksins?
- Áætlið tímann sem tekur að lesa 8MB skrá sem er samliggjandi á disknum.
- Hver er hámarksleshraði í stuttan tíma (burst rate)?

2. [10%] a) Lýsið muninum á *hliðrun* (shift, SHL og SHR í 8086) og *reiknihliðrun* (arithmetic shift, SAL og SAR í 8086). Notið báðar aðferðirnar til að hliðra 8-bitu 2's complement tölunni -5 til hægri um tvö sæti. Er niðurstaðan jöfn -5 deilt með 4? Útskýrið!

b) Sýnið dæmi með 32-bitu fleytitölum um að fleytitölusamlagning er ekki tengin (associative), þ.e. $x + (y + z) \neq (x + y) + z$.

3. [10%] Fleytitölur á IEEE formi nota tvær aðferðir til að tákna neikvæð gildi, formerkisbita fyrir brothluta og excess-aðferð fyrir veldisvísi. Við viljum hanna nýja gerð fleytitalna sem notar 2's complement aðferðina fyrir bæði veldi og brothluta, en er að öðru leyti eins og IEEE-fleytitölurnar.

- Hverju myndi þetta breyta um hvaða tölur væri hægt að tákna sem 32-bitu fleytitölur?
- Sýnið töluna -0.25 sem svona 32-bitu fleytitölu.

4. [20%] Tvíleitartré (binary search tree) eru tvíundartré með þann eiginleika að fyrir hvern hnút v er gildi í vinstra barni v minna en gildið í v og gildi í hægri barni er ekki minna en gildið í v . Hér að neðan er endurkvæm útgáfa af fallinu `TviLeit`, sem leitar að gildinu k í tvíleitartré, sem kallað er á með bendi á rót trésins. Skrifðu þessa útgáfu af `TviLeit` sem 8086-fall, kallalegt úr Borland C++. Gerið ráð fyrir að allir bendar séu 16 bita (near pointers) og allar heiltölur líka 16 bita. Athugið að skilgreiningin á `TreHnut` er aðeins til útskýringar, hún býr ekki til neinn forritskóða.

```

struct TreHnut {
    int gildi;
    TreHnut* vinstri;
    TreHnut* haegri;
};

TreHnut* TviLeit(TreHnut* v, int k) {
    if ((v == NULL) || (k == v->gildi))
        return v;
    if (k < v->gildi)
        return TviLeit(v->vinstri, k);
    else
        return TviLeit(v->haegri, k);
}

```

5. [10%] Hér að neðan eru 3 forritsbútar í 8086-smalamáli. Fyrir hvern þeirra segið hvað þeir gera, þ.e. almennt hvernig lokagildi gistanna eru háð upphafsgildum þeirra. Í *i*) er inntakið í gistinu `AX` og sömuleiðis úttakið. Í *ii*) er inntakið í gistunum `AX` og `BX`, og sömuleiðis úttakið. Í *iii*) er inntakið í `AX`, en úttakið í `BX`.

(Skipunin `CWD` framlengir 16-bitu 2's complement töluna í `AX` yfir í `DX`, `AX`, svokölluð "sign extension")

<i>i)</i>	<code>CWD</code>	
	<code>XOR</code>	<code>AX, DX</code>
	<code>SUB</code>	<code>AX, DX</code>
<i>ii)</i>	<code>XOR</code>	<code>AX, BX</code>
	<code>XOR</code>	<code>BX, AX</code>
	<code>XOR</code>	<code>AX, BX</code>
<i>iii)</i>	<code>MOV</code>	<code>BX, AX</code>
	<code>SHR</code>	<code>AX, 1</code>
<code>L1:</code>	<code>SUB</code>	<code>BX, AX</code>
	<code>SHR</code>	<code>AX, 1</code>
	<code>JNZ</code>	<code>L1</code>

6. [10%] Fallið hér að neðan ákvarðar hvort gefin heiltala sé frumtala (prímtala). Skrifðu það sem 8086 smalamálsfall, sem kallanlegt er úr Borland C++.

```
unsigned int FrumTala (unsigned int n) {
    if (n%2 == 0) return 0;
    for (unsigned int i=3; i*i<=n; i+=2)
        if (n%i == 0) return 0;
    return 1;
}
```

Lýsið (í smalamálskóðanum eða í orðum) hvernig þið getið komist hjá því að framkvæma margföldunina $i*i$ í hverri ítrun.

7. [15%] Hér að neðan er bútur úr C-forriti. Breyturnar n og k eru heiltölubreytur. Þýðið þennan bútt yfir í PA-RISC smalamál. Þið megið nota gistun $\%n$ og $\%k$ fyrir breyturnar tvær.

```
n = k;
while ( n >= k ) {
    if ( n%2 != 0)
        n = 3*n + 1;
    n = n/2;
}
```

Athugið að eitt af skilyrðunum sem hægt er að nota með skilyrtum hopp skipunum í PA-RISC er OD, sem er satt ef niðurstaðan er oddatala (t.d. hoppa í merki ef $\%t1 - \%t2$ er oddatala: comb, OD $\%t1, \%t2, merki$). Á sama hátt gildir skilyrðið EV ef niðurstaðan er jöfn tala.

8. [15%] Pentium með MMX hefur gagna- og skipanaskyndiminni, sem hvort um sig er 16KB. Þau eru mengistenging með 4 stökum í mengi og línustærð 32 bæti. Þegar velja á línu til að henda út úr mengi er notuð svokölluð gervi-LRU (pseudo-LRU). Þrír bitar eru notaðir fyrir hvert mengi, B0, B1 og B2. Biti B0 er settur 0 ef lína í sæti 2 eða 3 var notað síðast, en 1 ef lína í sæti 0 eða 1 var síðast notuð. Biti B1 er síðan settur 0 ef sæti 1 var notuð síðast, en 1 ef sæti 0 var notað síðast. Biti B2 er notaður á sama hátt fyrir sæti 2 og 3. Þegar velja á hvaða línu á að henda út er biti B0 skoðaður. Ef hann er 0 þá er biti B1 notaður til að velja á milli línanna í sætum 0 og 1, en ef B0 er 1, þá er biti B2 notaður til að velja á milli lína í sætum 2 og 3. Þessi aðferð er eftirlíking af LRU (Least Recently Used).

- Sýnið dæmi um minnisaðgang í eitt tiltekið mengi þar sem gervi-LRU hendir ekki út elstu línunni (þ.e. þeirri sem lengst er síðan var notuð) og hegðar sér því ekki eins og LRU.
- Sýnið forritsbútt í æðra forritunarmáli (C, C++, Pascal, ...) sem er slæmur fyrir þessa útskiptaðferð og útskýrið af hverju.