

Laboration 3

HENRIK BÄCK
850611-6253

Karlstads Universitet
2005-02-14

Handledare: Hans Hedbom
Nils Dåverhög

Innehållsförteckning

1. ANTAGANDEN.....	3
2. ÖVERSIKT	3
3. PROBLEM.....	3
4. REFLEKTION.....	3
5. SAMMANFATTNING.....	4
6. REFERENSLISTA.....	4
7. BILAGOR	4

1. Antaganden

Under projektet antogs att de strängar med tal som skulle skrivas ut kunde formateras genom att skapa en global variabel av typen `.asciiz`. För övrigt gjordes inga antaganden.

2. Översikt

Programmet är skrivet för att ta emot en vektor innehållandes tal. Denna vektor kommer först att skrivas ut till konsolen i sin originalordning. Därefter kommer den att sortera den i stigande ordning. När detta är gjort skrivs vektorn till konsolen igen.

3. Problem

Under projektets gång uppstod några mindre slarvfel i programkod. Under tiden som programmeringen hade skett hade vissa fel smugit sig in i koden. Dessa bestod i av att looparna var felformaterade på vissa ställen. Detta kunde exempelvis vara så att det använts `a<b` istället för `a>b`.

När rutinen för utskrift till konsol skedde hade instruktionerna för `sll` och `slt` blandats ihop. Genom att `slt` användes istället för `sll` uppstod många konstiga fel.

4. Reflektion

Efter att ha låtit laborationsmiljön översätta den c-kod som var given såg man mycket olikheter med den kod som under projektet framställdes.

Den största skillnaden var att den programkod som skrivits under projektets gång fungerade i jämförelse med den som översätts av laborationsmiljön. Detta berodde främst på att laborationsmiljön inte hade stöd för de funktioner som koden använde sig av.

En annan stor skillnad var att den kod som skrivits under projektet var mycket kortare och förhoppningsvis effektivare än den direktöversatta. Detta beror också mycket på att den kod som översatts av laborationsmiljön är gjord för att kunna köras vid multitasking. Detta då för att programmen av och till har tillgång till processorn och all data skall helst sparas till stacken.

Man kan också se att programkoden som genererats av laborationsmiljön har många fler hopp än den som skrivits under projektet. Detta kan tyckas medföra att den kod med färre hopp kommer att köras och avslutas

snabbare. Dock kunde detta inte ses på något sätt då programkoden som genererats av laborationsmiljön inte gick att köra.

5. Sammanfattning

Projektet löpte väl ut och slutade med att den översatta programkoden kunde köras och dess resultat kunde läsas av. Lärdom som dragits av detta projekt är en större förståelse för de instruktioner som finns i MIPS-arkitekturen. Samtidigt som mer kunskap om hur en processor arbetar har kunnat dras från detta projekt.

Eftersom man inte kunde prova programkoden förrän samtliga rutiner var färdigskrivna så uppstod lite problem när allt länkades ihop, detta kunde dock lösas efter lite effektiv felsökning.

6. Referenslista

Inget utöver kurslitteratur har använts.

7. Bilagor

Bilaga 1 – Projektets assemblerkod.

Bilaga 2 – Laborationsmiljöns assemblerkod.