Informationsteknologi, Del 3, IT1, VT 2003
Introduktion till C-programmering

24 mars 2003

1 Lärare

Namn	E-post	Rum	Postfack
Mikael Pettersson	mikpe@csd.uu.se	PB 1318	PB, hus 1, våning 4, nr 86

2 Kursinnehåll

Uttryck: variabler, konstanter, operatorer. Styrande satser (`if`, `return`, `while`, `for`, `do`, `switch`). Deklarationer av variabler och funktioner. Programstruktur, separatkompilering och Makefiler. Standardbibliotek, textuell I/O, strängar. Kommandoradsparametrar. Data: bastyper och sammansatta typer (arrayer och poster). Pekare. Dynamiska datastrukturer. `malloc` och `free`. Minneshantering. Unionstyper. Fallanalys.

3 Kurslitteratur

Valfri lärobok i C-programmering. Harbison & Steele, "C: A Reference Manual", eller Kernighan & Ritchie, "The C Programming Language", är de som rekommenderas för seriöst bruk, men de är inte läroböcker i allmän programmering.

På `http://www.csd.uu.se/~mikpe/teaching/it-cintro/` finns denna kursinformation samt kopior av föreläsningsanteckningar.

Schema finns på `www.uth.uu.se`.

4 Examination

Kursen examineras genom tre laborativa inlämningsuppgifter, i vilka enklare programmeringsproblem skall lösas i C. Inlämningsuppgifterna skall lösas i grupper om två, och lösningarna skall lämnas in skriftligt i kursledarens postfack senast vid angivna deadlines.

Filerna för varje inlämningsuppgift skall finnas tillgängliga i en separat katalog (en katalog per uppgift). Denna katalog skall vara lässkyddad för andra studenter men tillåta läsning för kursledaren. Detta åstadkommes med:

```
chmod go-rwx katalog
setfacl -m u:mikpe:r-x katalog
setfacl -m m:r-x katalog
```

Dessutom skall alla filer i katalogen ha samma skydd som katalogen: upprepa `chmod`- och `setfacl`-kommandona för filerna i katalogen.

Den skriftliga redovisningen skall innehålla en programlistning och ett försättsblad med kursnamn, numret på inlämningsuppgiften, samt studenternas namn och personnummer (utan personnummer kan resultat inte registreras!). Dessutom skall den ange namnet på den katalog där filerna för uppgiften finns.

5 Undervisning

Undervisningen ges som föreläsningar, laborationer och handledning. All undervisning är frivillig (utom uppropet).

Kursledaren har handledningstid Fredagar kl 14.00-15.00 veckorna 13, 14 och 21, men det går också bra att få handledning via e-post. Observera att handledning inte är till för att avlusa er kod, utan för att hjälpa er med att lösa specifika problem med t ex verktygen och problemstrukturering.

6 Preliminär kursplan

Vecka 13		
Mån	Fö1	Upprop. C-vs-SML. Skalära datatyper, variabler, uttryck, tilldelningar, satser, funktioner, programstruktur. `gcc`. `make`.
Tis	Fö2	Arrayer, strängar, pekare, loopar, standardbibliotek, pre–processorn, bit–operatorer. Introduktion av inlämningsuppgift 1.
Ons	Lab	
Fre	Le1	? strykes ?
Vecka 14		
Tis	Fö3	Poster, dynamiska datastrukturer och minneshantering.
Ons		Pekararitmetik. Deadline för inlämningsuppgift 1 kl. 13.00.
Ons	Fö4	Fortsättning. Genomgång av inlämningsuppgift 1. Introduktion av inlämningsuppgift 2.
Tor	Lab	
Vecka 21		
Mån		(ingen föreläsning, schemafel)
Tis		(ingen föreläsning, schemafel)
Ons		Deadline för inlämningsuppgift 2 kl 10.00
Ons	Fö5	"Avancerad" C. Unionstyper. Fallanalys. Genomgång av inlämningsuppgift 2. Introduktion av inlämningsuppgift 3.
Tor	Lab	
Fre	Le2	? strykes ?
Vecka 22		
Fre		Deadline för inlämningsuppgift 3 kl 17.00

7 Inlämningsuppgift 1

7.1 Deluppgift A

Implementera en procedur som multiplicerar en $k \times m$ matris A och en $m \times n$ matris B, och skriver resultatet till en $k \times n$ matris C vars element beräknas enligt

$$c_{ij} = \sum_{p=1}^{m} (a_{ip} \times b_{pj})$$

för $i = 1, 2, \ldots, k$ och $j = 1, 2, \ldots, n$.

Matriselementen skall vara flyttal i dubbel precision. Dimensionerna k, m och n är konstanter, men de ska enkelt kunna ändras i källkoden.

Implementera sedan ett huvudprogram `main()` som initialiserar två matriser A och B med valfria värden, anropar proceduren för matrismultiplikation, och sedan skriver ut resultatmatrisen C på `stdout`. Observera att programspråket C indexerar vektorer och matriser från 0 och inte 1.

7.2 Deluppgift B

Implementera ett program som tar tre parametrar på kommandoraden. De första två parametrarna är flyttal, den tredje är en sträng `"add"` eller `"mul"`. Beroende på utseendet på den tredje strängen skall de två talen antingen adderas eller multipliceras. Resultatet skall skrivas ut till `stdout`.

För att lösa uppgiften skall ni använda funktionen `strtod()` från `<stdlib.h>` och funktionen `strcmp()` från `<string.h>`. Använd `man`-kommandot för att hitta dokumentationen för dessa funktioner, eller titta i er lärobok.

8 Inlämningsuppgift 2

Skriv ett program som sorterar heltal. Indata är en textfil innehållandes ett antal heltal, ett per rad. Utdata är en textfil med samma mängd heltal, ett per rad, i stigande ordning. Dubletter skall tas bort. Indata skall läsas från `stdin`. Utdata skall skrivas till `stdout`.

För att sortera talen skall ett binärt sökträd användas, vars storlek anpassas under körningen. Dynamisk minnesallokering krävs följdaktligen.

9 Inlämningsuppgift 3

En enkel interpretator för aritmetiska uttryck skall implementeras. De uttryck som skall hanteras är: heltalskonstanter, negation, addition och multiplikation. Ex: $1 + 2 * -3$.

Steg 1: Definiera en datatyp för representation av dessa uttryck i trädliknande form. Implementera hjälpfunktioner `mk_int_expr(int)`, `mk_neg_expr(e)`, `mk_add_expr(e,e)` och `mk_mul_expr(e,e)` som skapar träd för de givna typerna av uttryck.

Steg 2: Implementera en procedur `void print(e)` som tar ett uttrycksträd `e` och skriver ut motsvarande uttryck på en rad. Parenteser skall användas när de behövs, men ej annars: ex.vis. skall $(1 + 2) * 3$ skrivas ut på det sättet, men $(1 * 2) + 3$ skall skrivas ut som $1 * 2 + 3$.

Steg 3: Implementera en funktion `int eval(e)` som tar ett uttrycksträd `e` och beräknar dess värde.

Steg 4: Implementera ett huvudprogram `main()` som först skapar ett uttrycksträd genom anrop till funktionerna från Steg 1, skriver ut det med `print()`, evaluerar det med `eval()`, och till sist skriver ut dess värde. Ni får själva välja vilket uttryck trädet skall representera, men det måste innehålla samtliga former av uttyck som skall hanteras, och det måste demonstrera att er `print()`-procedur hanterar parenteser korrekt.