UMEÅ UNIVERSITET 2005-04-08
Institutionen för datavetenskap
Johan Eliasson, Jon Hollström & Rickard Lönneborg
Modifierat av Helena Lindgren, Daniel Lundmark & David Olsson & Tomas Larsson
{johane, helena, lundmark, don, tomasl}@cs.umu.se

Introduktionsmaterial

Programmeringsteknik

Innehåll

1 Logga in

I PC-labben körs Windows XP och före du kan använda systemet måste du logga in.

1. Tryck om det behövs CTRL ALT DEL för att komma till loginfönstret.
2. Skriv ditt användarnamn (ens05xxx).
3. Skriv in ditt nuvarande lösenord.
4. Klicka OK.

2 Byta lösenord

Det första du bör göra när du har loggat in är att byta lösenord:

1. Klicka på ikonen **Byt lösenord** på skrivbordet.
2. Klicka på **Yes** när fönstret **Host key** är synligt.
3. Skriv in ditt nuvarande lösenord.
4. Skriv in ditt nya lösenord två gånger.
5. Kom ihåg ditt nya lösenord.

OBS! Ditt nya lösenord är inte giltigt omedelbart. Det kan ta upp till 30 minuter innan det nya lösenordet kan användas. Om du har problem med ditt lösenord, kontakta **support** på översta våningen i MIT-huset.

3 Hemkatalog

Din hemkatalog på Unix-systemet är tillgängligt genom att använda **H:** i utforskaren. Denna katalog motsvarar ditt användarnamn i Unix-systemet **~ens03xxx**. Det finns en gräns för hur mycket du kan lagra i din hemkatalog. Normalt mellan 15 till 25 MByte.

4 Unixintroduktion

Om du arbetat med en PC förut har du kanske kommit i kontakt med Microsofts operativsystem Windows. Ett operativsystem är ett program som lägger sig som ett skal på datorns hårdvara, och låter dig starta vilka program du vill.

En annan familj av operativsystem brukar man kalla Unix. Här räknar man i dagligt tal in bland annat Solaris (som vi använder här på CS), Linux och de olika varianterna av BSD. Den stora skillnaden mellan Unix och Windows är att Unix-systemen är designade för att flera användare ska arbeta med samma dator, samtidigt. Detta betyder att det är oviktigt om vi sitter framför den dator vi vill arbeta med, eller om vi loggar in på den utifrån, med t ex ssh. Du kan välja att logga in hemifrån, via internet, eller varför inte sitta på andra sidan jorden och labba, allt du behöver är en internetuppkoppling.

Vi presenterar här de viktigaste kommandon du behöver känna till för att komma igång. En mer komplett introduktion till Unix finns tillgänglig via www på:

http://www.cs.umu.se/~jon/unixintro/

4.1 Viktiga kommandon

Manualsida för varje enskilt kommando fås med kommandot man kommando där kommando är det kommando du vill ha information om.

ls	Listar de filer som finns i den aktuella katalogen
pwd	Visar aktuell katalog
cd	Byter aktuell katalog
cp	Kopierar filer eller kataloger
mkdir	Skapar en katalog
rmdir	Tar bort en katalog
rm	Tar bort en fil
man kommando	Visar manualen för kommando
w	Visar vilka som är inloggade på datorn
gcc	C-kompilator
pine	Mail-program
passwd	Byter lösenord på UNIX-systemet

Exempel:

```
peppar:~> mkdir edu
peppar:~> cd edu
peppar:~/edu> cd ..
peppar:~>
...
```

I exemplet skapas en katalog som heter edu, med cd edu sätts edu till aktuell katalog. cd .. går till föregående katalog.

5 Läsa e-mail

Det program ni kommer att använda för detta är *pine*, vilket körs i UNIX. Logga in på peppar mha putty och skriv pine.

Användbara kommandon i pine:

m	Visar huvudmenyn
i	Visar din aktuella mailbox
l	visar tillgängliga mailboxes (sent-mail, inbox, saved-messages)
c	Skriv ett mail (när det är färdigt, skicka med Ctrl-X)

6 Dirigera om mail till annan adress

Vill ni att mail som når er användare ska skickas vidare till en annan adress skriver ni adressen er mail ska skickas till i en fil. Filen ska heta .forward och ligga i roten i er hemkatalog. Kontrollera efteråt att mail når rätt adress.

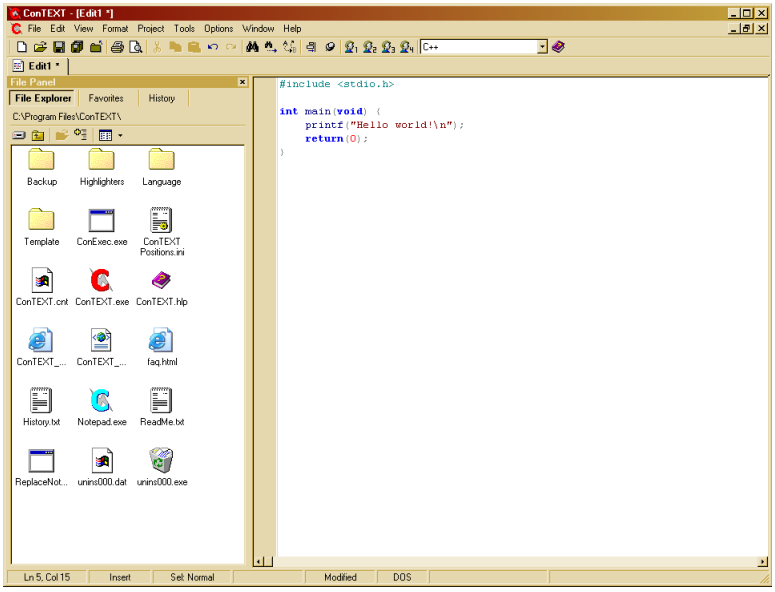
7 WWW – World Wide Web

I princip all information relaterad till kursen finns presenterad på en kurssida på WWW. Där finns t ex specifikationer till laborationer, regler, föreläsningsschema, handledningstider, osv. För att komma åt sidan använder du dig av en webb-läsare, t ex Netscape Navigator. Den startas lämpligen från Start-menyn i Windows.

Om du ska kunna komma åt sidor utanför universitetes nät måste du logga in på en så kallad *proxy-server*. När en dialogruta kommer upp för detta fyller du i *hela* din email-adress *här på cs*. *Ex: ens00xyz@cs.umu.se*. Som lösenord skriver du ditt UNIX-lösenord.

8 Editera filer

Att skapa och ändra en c-fil kan du göra med valfri text-editor, exempel på sådana är *Notepad, vim, emacs, pico* osv... *ConTEXT* är den text-editor vi kommer använda oss av på den här kursen.
För att starta *ConTEXT*,
välj **Programs→Editors→ConTEXT→ConTEXT** i startmenyn. Det fönster som kommer bör se ut såhär:

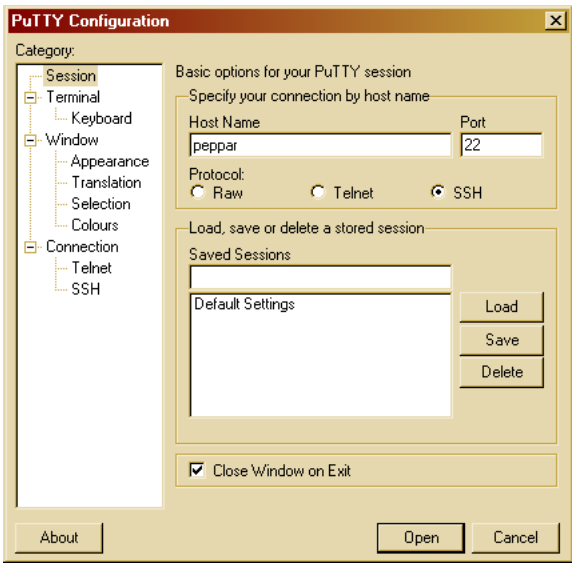


Välj **File→New** och **File→Save As** i menyn. Välj din hemkatalog som plats att spara filen och döp den till *filnamn.c* för att ConTEXT ska förstå att det är just en C-fil du ska editera. Nu kan du skriva ditt C-program och spara det innan du går vidare till kompilering. Ett enkelt C-program finns bifogat.

9 Kompilera

Kompilering ska göras på ett UNIX-system. För att göra det måste du logga in på ett UNIX-system med en *ssh-klient* eller motsvarande. Den ssh-klient vi kommer att använda heter *putty*. Den är gratis och finns att ladda hem från nätet.

För att starta putty, välj **Programs→Internet tools→Putty** i Startmenyn. Du får nu upp en dialogruta som ska se ut såhär när du fyllt i den nästan rätt (istället för **peppar**, skriv **peppar.cs.umu.se**):



Namnet på det UNIX-system du loggar in på är *peppar* och *ssh* innebär att att kryptering används (ssh är säkrare än telnet).

Tryck på **Open** och du får upp ett terminalfönster där du får fylla i ditt användarnamn och lösenord. När lösenordet accepterats har du loggat in på *peppar*. Nu kan du med hjälp av UNIX-kommandon manipulera filer i din hemkatalog. Du **bör** testa de UNIX-kommandon som finns bifogade. Kompilera gör du med kommandot gcc. Kommandot skrivs enligt:

gcc [options] files...

Exempel:

gcc -o myprog myprog.c	kompilerar myprog.c till den körbara filen myprog
gcc -Wall -o myprog myprog.c	som ovan men med varningar påslagna (Rekommenderas!)
gcc -pedantic -ansi ...	ger ännu fler varningar

10 Testkörning

När din program är kompilerat kan du testköra det, *gcc* kommer att lägga det körbara programmet i den katalog du befinner dig i. Att programmet ligger där det ska kan du kontrollera med kommandot *ls*. Om du kompilerat ditt program med *gcc -Wall -o myprog myprog.c* kommer den körbara filen heta myprog. Växeln *-o myprog* anger just att du vill att den körbara filen ska heta *myprog*.

För att testköra *myprog* om det ligger i aktuell katalog, skriv *./myprog*. Du skriver *./* före programnamnet för att ange att programmet ska köras från aktuell katalog.

Exempel:

```
peppar:~> gcc -Wall -o hello hello.c
peppar:~> ./hello
Hello world!
peppar:~>
```

11 Enkelt C-program

```
#include <stdio.h>

int main(void){
        printf("Hello World!\n");  /* Skriv ut en liten text och radmatning */
        return 0;                  /* Retunera 0 för att tala om att allt gått bra */
}
```

12 Felsökning

När man programmerar C gör man vanligtvis diverse fel i koden. Felen delas in i två kategorier, syntaxfel och sematiska fel. Kortfattat betyder syntaktiska fel att man stavat fel i C-koden och semantiska fel att man stavat rätt men kanske tänkt fel. Syntaktiska fel upptäcks vid kompilering och brukar vara enkla att åtgärda om man kan tolka vad gcc skriver. Semantiska fel får man upptäcka genom att testköra programmen.

12.1 Kompilering, länkning

Här nedan följer ett par vanliga felmeddelanden och dess innebörd.

```
fel.c: In function ‘main’:
fel.c:7: parse error before ‘}’
```

Detta är ett vanligt fel som oftast beror på att man glömt ett **;** i slutet på en rad. Gcc kommer då att rapportera att felet ligger på raden efter. I detta fall saknas ett semikolon på rad 6 när gcc säger att felet ligger på rad 7.

```
fel.c:5: ‘i’ undeclared (first use in this function)
fel.c:5: (Each undeclared identifier is reported only once
fel.c:5: for each function it appears in.)
```

I det här fallet betyder det att kompilatorn stött på en variabel **i** som inte var deklarerad.

Förutom dessa fel finns ett antal varningar som kan innebära att programmet fungerar helt, kraschar ibland eller alltid kraschar, beroende på vilken varning det är. Ett exempel på en ”ofarlig” varning är:

```
fel.c:4: warning: unused variable ‘k’
```

Nedan följer exempel på mer ”allvarliga” varningar.

```
fel.c:6: warning: implicit declaration of function ‘orintf’
```

Implicit declaration of *funktionsnamn* får du som felmeddelande om du :

- Stavat funktionsnamnet fel
- Anropat en funktion och inte inkluderat rätt .h-fil
- Om det är en egen funktion, skrivit den under main() utan att ha lagt första raden ovanför main.

Om du stavat funktionsnamnet fel resulterar det i fel vid *länkningen* som är det sista gcc utför på ditt program. Felen ser ut enligt nedan.

```
Undefined                       first referenced
 symbol                             in file
```

```
orintf                                /var/tmp/ccNsJkDb.o
ld: fatal: Symbol referencing errors. No output written to a.out
collect2: ld returned 1 exit status
```

I det här fallet är det en felstavning av funktionen ”printf”.

12.2 Testkörning

Vid testkörning bör man ha lämpliga utskrifter för att se om programmet gör det som förväntas. Om så inte är fallet kan en vettig metod för att isolera felen vara att **sätta ut många utskrifter** i programmet för att kontrollera om en funktion körts klart m.m.

I och med att pekare används i stor utsträckning i C så finns alltid en risk för att dessa pekar på något otillåtet och ställer till besvär. Typiska fel på att en pekare pekar på något otillåtet eller att man försöker avreferera en NULL-pekare är dessa:

```
Segmentation fault (core dumped)
```

och

```
Bus error (core dumped)
```

Dyker ett sånt här fel upp så bör man kolla upp om man har någon pekare som är NULL när den inte ska vara det etc. Vill man ta reda på var i koden det kraschat kan man använda *gdb* som är en debugger. Denna kommer vi titta på senare under kursen om vi får tid till det.

13 Utskrift

För att skriva ut c-kod från prompten kan man använda kommandot a2ps. Kombinerar man det med kommandot lpr kan man styra utskriften till en speciell skrivare:

a2ps -C *filnamn* | lpr -P *skrivarnamn*

Utskriften blir tvåsidig och -C ger ett visst format till utskriften som passar för C-kod.

14 Översikt av datorsystem

Här följer en bild över de delar som beskrivits i introduktionen.