

Att komma igång

- Vad är ett program?
- Kompilerande-Interpreterande
- Programmeringsmiljö
- Hello World!
- Att programmera och ett enkelt program

Vad är ett program?

En processor kan endast utföra extremet enkla operationer såsom: flytta ett tal i minnet till en annan plats, lägga ihop ett tal med ett annat tal, dela ett tal med två osv.

Ett program som processorn kör måste därför bestå av sådana enkla instruktioner. Dessutom finns hoppinstruktioner som gör att den kan köra om sekvenser av koden (oftast efter att ha bytt plats på talen i minnet). Förutom vid hoppinstruktioner utför processorn instruktionerna i ordning en efter en.

- ett sådant enkelt program som processorn kan förstå sägs vara i **maskinkod**

Oftast vill man själv inte programmera i maskinkod då det blir extremt arbetsamt att få datorn att göra något användbart.

Istället programmerar vi i något högnivåspråk, t.ex C. Denna kod kallas **källkod** (C-kod). Denna kompileras sedan av en **kompilator** (ett program) till maskinkod (objektkod) som processorn kan förstå. Filen som vi då får kallas för exekverbar, körbar eller en exe-fil.

Kompilerande-Interpreterande

- Det ovan beskrivna sättet att arbeta beskriver hur man arbetar med kompilerande språk. Ex. C, C++, Fortran
- I interpreterande språk kompileras inte koden i förväg. Istället låter vi ett program översätta åt processorn varje gång vi kör koden. Dessa program kallas interpreterande. Ex. Basic, Lisp
- Samma språk kan ha båda möjligheterna (Java är ett mellanting).

Programmeringsmiljö

- För att kunna programmera behöver vi ett eller flera program på vår dator. Dessa kallas vår programmeringsmiljö.
- Till att börja med är det bäst att använda enklast tänkbara miljö för att verkligen förstå vad man gör. Vi ska använda en miljö som heter MinGW (mingw.org). Denna innehåller kompilator (och en del andra program såsom länkare) men ingen editor. För att skriva in kod kan man använda vilken texteditor som helst (kod är vanlig text tills den kompileras). Vi rekommenderar att ni använder [emacs](#). MinGW körs i kommandotolken så den blir också en del av vår programmeringsmiljö.
- Senare, kanske redan under denna kurs, när man börjar förstå hur saker hänger ihop kan det vara smidigare att använda en lite mer avancerad programmeringsmiljö. Den som vill kan då prova CodeBlocks eller Bloodshed Dev-C++.

Programmeringsmiljö

- Egen dator

- Installera VirtualBox (gratis)

Kör en virtuell windows xp med all programvara förinstallerad. Instruktioner hittar du i introduktionskursen [HF1010](#) på kth-social.

Du behöver då en windows image. Den finns på skolans nätverk på utb:

U:\Kurs\HF1010_Introduktion_Till_Dator teknik\HT2011\SmallWinXP2.ova

- Installera MinGW själv. Se till att välja att installera även C++ kompilator när du under installationen har det valet.

Installera emacs eller xemacs.

Du behöver sätta path variabeln, se särskild instruktion:

U:\Kurs\HI1024\InstruktionerProgrammeringSkolansDatorer.pdf

- Skolans datorer

- Kör en virtuell miljö som redan finns installerad. Följ instruktionerna på [HF1010](#)

- **Kör programmen under windows 7 direkt.** Se särskild instruktion om ni får problem

U:\Kurs\HI1024\InstruktionerProgrammeringSkolansDatorer.pdf

OBS: viktigt att ni använder skolans datorer för att redovisa laborationer. Vi ska nämligen tentera på skolans datorer och då behöver ni förtrogenhet med dem.

Kommandotolken

- `cd` – change directory – byt katalog
`cd namn` – byter till katalog `namn`
`cd ..` – hoppar upp en katalog
`h:` – byter till enheten `h:`
- `dir` – directory – katalog
listar alla filer i den katalog man står i.
- `type fil` – skriver ut texten i filen `fil`
- `H:\program>` - kallas prompt
`H:` står för eran hemkatalog, sedan följer vilka kataloger ni gått ner i – er path

Hello World!

- Är ett litet program som gör minsta möjliga för att testa att man har en fungerande programmeringsmiljö och att man förstår hur man använder den:

```
#include <stdio.h>
main()
{
    printf("Hello World!");
}
```

- Spara denna text i en fil **hello.c** i en mapp i din hemkatalog (h:).
- Öppna kommandotolken och använd *cd* för att gå till rätt mapp.
>*cd mappNamn*
Kontrollera med *dir* att filen ligger där.
Skriv kommandot : `gcc hello.c <enter>`
- Använd *dir* för att se att en fil `a.exe` skapats
- Prova att köra filen genom att skriva: `a <enter>`
- Ett litet första steg mot fantastiska program och ett steg man alltid börjar med på en ny dator eller i en ny miljö och ett steg man alltid går tillbaka till om allt krånglar.

Ett enkelt program

```
/* Beräkning av sträckan om man färdas med hastigheten 25 m/s */
```

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    int stracka,hastighet,tid;
```

```
    hastighet = 25;
```

```
    printf("Hur lang tid fardas du? ");
```

```
    scanf("%d", &tid);
```

```
    stracka = hastighet * tid;
```

```
    printf("Med hastigheten 25 m/s kommer du att fardas %d m.\n", stracka);
```

```
    printf("Lycka till pa farden");
```

```
}
```

- Hur programmerar man? skriv en rad – kompilera – provkör - osv

- Hur läser man ett program? Som en kompilator – rad efter rad

- Vad gör varje tecken? Allt betyder något

Målet är att förstå allt – ibland får ni nu i början nöja er med att acceptera

- Kompilatorn är extremt kinkig – ha tålamod – kompileringsfel är ”enkla” att fixa!

Ett enkelt program

```
/* Beräkning av sträckan om man färdas med hastigheten 25 m/s */
```

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    int stracka,hastighet,tid;
```

```
    hastighet = 25;
```

```
    printf("Hur lang tid fardas du? ");
```

```
    scanf("%d", &tid);
```

```
    stracka = hastighet * tid;
```

```
    printf("Med hastigheten 25 m/s kommer du att fardas %d m.\n", stracka);
```

```
    printf("Lycka till pa farden");
```

```
}
```

Kommandotolken:

```
H:\programmering>gcc enkeltP.c -o enkeltP.exe
```

```
H:\programmering>enkeltP
```

```
Hur lang tid fardas du? 10
```

```
Med hastigheten 25 m/s kommer du att fardas 250 m.
```

```
Lycka till pa farden
```

```
H:\programmering>
```

Ett exempel till

```
#include <stdio.h>
int main(void)
{
    float r1,r2,r3;
    printf("Resistans hos R1 (ohm) : ");
    scanf("%f",&r1);
    printf("Resistans hos R2 (ohm) : ");
    scanf("%f",&r2);
    r3=r1*r2/(r1+r2);
    printf("R3 beräknas till %6.3f ohm\n",r3);
}
```

//från boken