

Föreläsning 1

Programmering är ett *hantverk*. Det betyder att man inte kan läsa sig till den förmågan, man måste träna och man tränar genom att skriva mer och mer avancerade program. Programmering förutsätter ett programmeringsspråk och det finns tusentals språk. Några av de mest betydelsefulla är *C*, *C++*, *Java*, *C#*, *Python*, *Perl*, *Ruby*, *PHP*.

En stor svårighet med programmering som vi försöker komma åt i den här kursen är att man inte riktigt kommer igång och programmerar ordentligt. I den här kursen ska alla vara igång och programmera från och med idag. Från och med här och nu.

Men först måste vi förklara vad programmering är. Att programmera är, rent praktiskt, att sätta samman instruktioner (ett program) som först och främst människor kan förstå. När det är klart ska man översätta programmet till någonting som en maskin kan förstå. Det är grunden. Man brukar säga att man programmerar för en viss maskin och vi kommer i stor utsträckning att programmera för datorer. Det är där vi börjar. Rent praktiskt går programmering för datorer till så här:

1. Man skriver ett program i ett programspråk, till exempel *C*. Detta program lagras i en textfil med ändelsen `.c`, det kanske heter `mittprogram.c`.
2. Programmet `mittprogram.c` är en textfil som nu skickas till en översättare, en så kallad kompilator, som översätter programkoden i filen `mittprogram.c` till kod som en dator kan köra. Denna kod lagras i en så kallad körbar fil, i *Windows* har dessa filer ofta ändelsen `.exe`.
3. Nu kan man köra programmet, i *Windows* dubbelklickar man på ikonen som bildas då `.exe`-filen skapas. Vi ska inte göra det så mycket i den här kursen.

Vi ska inte bry oss så mycket om ikoner och saker runtomkring i den här kursen, den här kursen är designad för att vi snabbt ska få en försmak av vad programmering egentligen innebär i grunden. Och i grunden handlar programmering om att skola sitt tänkande och att tänka rätt. Vi behöver uttrycka oss väldigt precist om vi ska få en maskin att lyda våra order. En maskin är ju väldigt korkad och måste få konkreta instruktioner om exakt vad som ska utföras. Det är ju inte fallet med människor.

För att komma i kontakt med programmeringsproblematiken ska vi använda oss av en artificiell miljö som heter *Guido van Robot*. Den maskin vi där ska programmera blir en liten robot som heter *Guido*. (Det är ett startpaket för att lära sig programmera i *Python*, och språket *Python*s skapare heter Guido van Rossum, men det går lika bra att använda för att läsa sig *C* som vi ska gå vidare till i grundkursen.

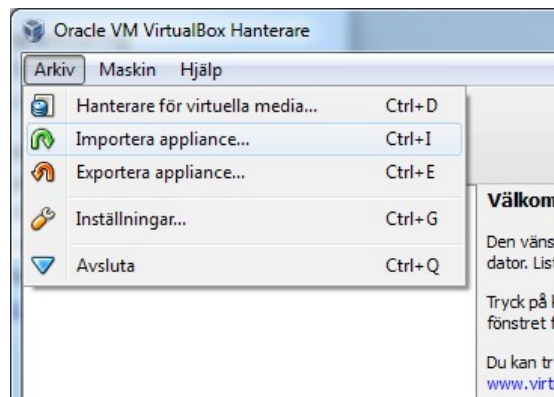
Miljön *Guido van Robot* kör tyvärr inte i de senare versionerna av *Windows* (vad jag vet) så vi ska passa på och illustrera ytterligare en mycket intressant och viktig teknik: *Virtualisering*. Vi ska köra operativsystemet *Windows XP* men vi ska köra det *virtuellt*. Vad betyder det? Låt oss ta det från början.

I grund och botten är ett datorsystem uppbyggt så här: vi har hårdvaran, själva datorn, skärmen, tangentbordet osv. Sedan finns mjukvaran, alltså de program som datorn kör. Den viktigaste mjukvaran i datorn är det så kallade *operativsystemet*. Det är ett slags grundprogram som hanterar alla resurser i datorsystemet. Utan operativsystemet kan inga andra program köras. Operativsystem

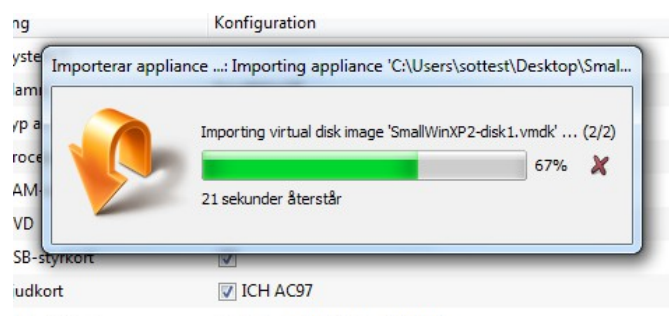
kommer i olika smaker och former, ett av de mest kända operativsystemen är *Windows* som just nu är uppe i version 7 och snart kommer 8. På skolan använder vi version 7. Den tidigare versionen hette *Windows XP*. (*Windows Vista* räknar jag som ett uttryck för inkompetens och jag räknar inte med det som ett operativsystem.)

På alla datorer i skolan finns ett program som heter *VirtualBox*, och det programmet tillåter oss att *emulera* körning av andra operativsystem. Ni kommer att använda VB i flera kurser framöver, så det är bra att vi börjar med det redan nu. Troligtvis kommer virtualisering att vara en ohyggligt viktig teknik framöver. Många av er kanske hört talas om *Cloud Computing*, virtualisering ligger till grund för det begreppet.

Starta nu VirtualBox. (Det finns en ikon på skrivbordet som du bara dubbelklickar på.) Det första du ska göra är att se till att de virtuella maskinerna som du ska arbeta med hamnar under C:\VMS\. Det gör man genom att gå till Arkiv->Inställningar och välja C:\VMS\ som standardmapp för maskiner. Denna mapp, C:\VMS, har alla skrivrättigheter i. Sedan kan du göra "Import Appliance", så här:

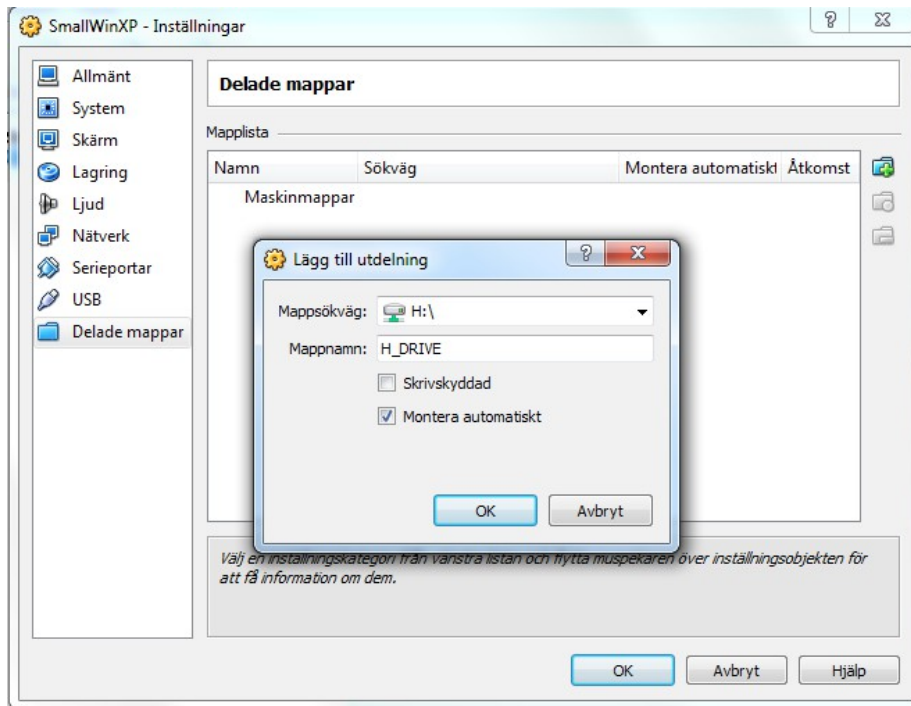


och sedan välja SmallWinXP2.OVA som finns tillgänglig precis bredvid det här dokumentet som du läser nu. Det tar en stund, man får se en mätare som visar hur långt det är kvar:

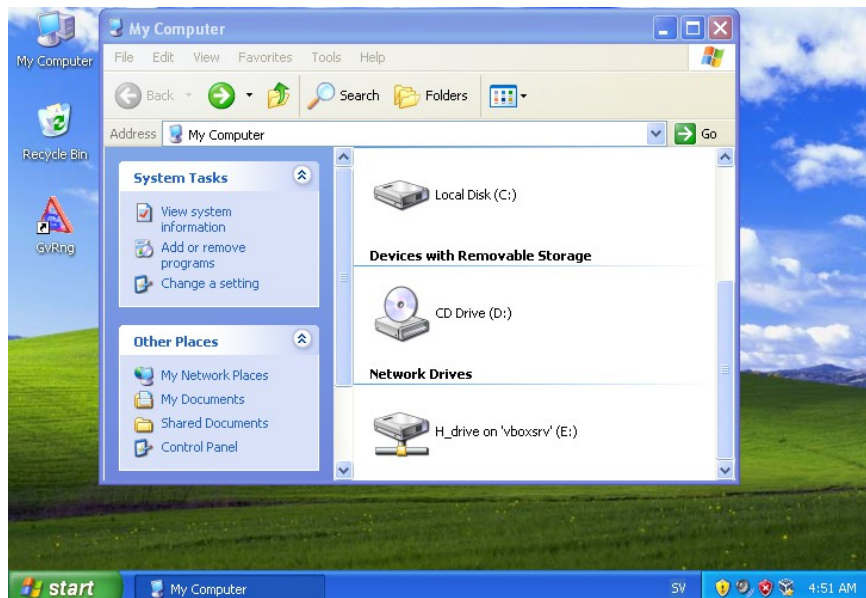


När den är klar kan man starta maskinen. Dock, innan du startar maskinen, markera maskinen och gå in på maskin->Inställningar. Här behöver du klicka bort USB 2.0 under inställningar för USB. Välj nu *Delade mappar / Shared Folders*. Skapa en ny delad mapp och välj att dela ut H: till den virtuella maskinen. Det betyder att när du kör i skolan kommer din virtuella maskin att kunna komma åt H:. Om du inte kör i skolan kan det vara bra att dela ut någon annan mapp som ligger lokalt på din dator som du sitter vid. Men det är ändå viktigt att du lär dig hantera just H: eftersom det är den mappen vi ska använda i examinationssammanhang i kursen i grundläggande

programmering, enkelt uttryckt, när du ska tenta måste du känna till och kunna använda H:. Om du arbetar med delade mappar kan du faktiskt ha de dokument och program som du skapar skilda från själva den virtuella maskin som du arbetar på. Det är då mycket enkelt att göra säkerhetskopior, bara kopiera dokumenten. Det ser ut så här när man skapar en delad mapp: (Det ser lite annorlunda ut om du sitter vid en egen dator, då finns troligtvis inte H:, då går det bra med annan mapp.)

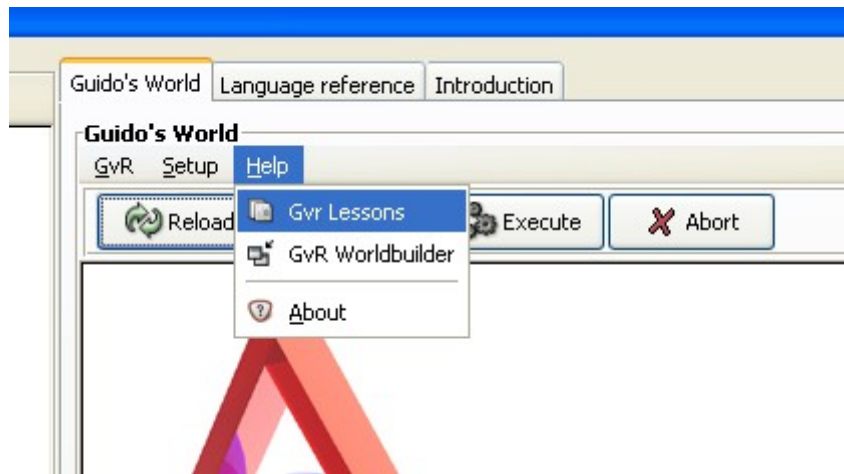


Nu kan du starta maskinen. Om allt gått som det ska har du en vy som ser ut ungefär så här:



Jag har ändrat lite på fönsterstorleken och öppnat "My Computer". E: är då den utdelade mappen. Lägg alltid alla dina filer här, lägg ingenting på skrivbordet eller C:. Ta som vana att aldrig lägga någonting på skrivbordet, skrivbordet är del av den lokala datorn och det raderas med jämna

mellanrum. Lägg alltid alla dokument på H:. Den viktigaste delen i kursen består i att gå igenom alla lektioner i GvRng. Miljön GvRng är ganska självinstruerande och lektionerna hittar du här:



Det är bara att sätta igång. Gör alla lektioner. De lektioner som ska redovisas för godkänt betyg i denna kurs är 7, 8, 11, 12, 13, 14, 15, 16. I uppgift 7, se till att du kan skriva alla möjliga kombinationer av ettor och nollor!

Del II: Att komma igång med C-programmering

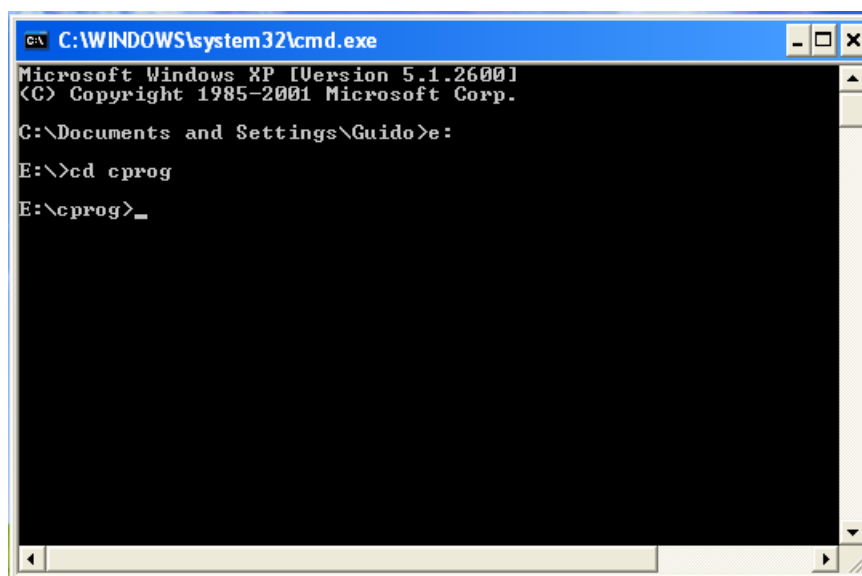
Guido van Robot är en jättebra start, det är hyfsat enkelt att jobba igenom lektionerna och man får en överblick av många aspekter av programmering. Men vi behöver komma vidare, det finns många vägar, många programmeringsspråk. I nästföljande kurs ska vi inrikta oss på programspråket C så vi ger en liten start till hur man kommer igång med C-programmering här.

Det är bra att kunna arbeta vid en så kallad kommandoprompt och en sådan startar man i *Windows XP* genom att skriva cmd i körrutan (Välj Start->Run), så här:



Då får man fram en svart ruta som man kan skriva in kommandon vid. Skapa en underkatalog på E: som heter "cprog". Man kan man göra detta grafiskt genom att högerklicka med musen i E: och

välja New->Folder. Nu ska du gå till den katalogen med kommandoprompten. Det ser ut så här:



```
CA C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
C:\Documents and Settings\Guido>e:
E:\>cd cprog
E:\cprog>_
```

Nu kan du börja ge kommandon och påverka innehållet i katalogen E:\cprog. Det vi vill göra här är att skapa en textfil som innehåller ett C-program, kompilera och köra dem. Starta *Emacs* genom att välja *runemacs.exe* på startmenyn. Gå igenom någon enkel *Emacs*-tutorial) så att du känner dig hemma med att skapa och redigera textfiler. (Det finns många på nätet (googla) och även en inbyggd i själva *Emacs*. En sammanfattning av emacskommandon finns också på kursens webbsida Lägg sedan följande textfil under E:\cprog:

```
#include <stdio.h>
main()
{
    printf("This is a C-program!\n My first!\n");
}
```

Döp filen till *firstprogram.c*. OBS: Kopiera inte detta program direct från detta document, *skriv av det*, tecken för tecken. Varje tecken måste bli exakt rätt och det blir inte riktigt rätt om man använder copy-paste. Nu har vi skrivit ett C-program. För att se att programmet verkligen finns på plats kan man använda sig av kommandot *dir*, som listar kataloginnehåll. Har du gjort rätt ska det se ut ungefär så här:

```

c:\ Command Prompt
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Guido>e:
E:\>cd cprog
E:\cprog>dir
Volume in drive E is UBOX_H_DRIVE
Volume Serial Number is 5C50-4552

Directory of E:\cprog

06/30/2011  07:17 AM    <DIR>          .
06/30/2011  07:17 AM    <DIR>          ..
06/30/2011  07:27 AM                85 firstprogram.c
               1 File(s)                85 bytes
               2 Dir(s)  1,046,056,128,512 bytes free

E:\cprog>_
    
```

Vi ser då som sagt en fil som heter `firstprogram.c`. Denna fil är en textfil som människor kan läsa, men datorn kan inte läsa denna fil. För att datorn ska förstå vad som står i filen måste filen översättas till en körbar fil i så kallat binärt format. Det gör man med hjälp av en så kallad "kompilator" och en så kallad "länkare" och dessa två saker finns inbyggda i verktyget `gcc`. För att skapa ett riktigt körbart program så skriver `gcc firstprogram.c` vid kommandoprompten och om allt gått rätt till har nu ett körbart program skapats. Det heter `a.exe` och vi kan köra det genom att skriva `a` vid prompten. Båda dessa steg, framgångsrikt genomförda, ser ut så här:

```

c:\ Command Prompt
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Guido>e:
E:\>cd cprog
E:\cprog>gcc firstprogram.c
E:\cprog>dir
Volume in drive E is UBOX_H_DRIVE
Volume Serial Number is 5C50-4552

Directory of E:\cprog

06/30/2011  07:17 AM    <DIR>          .
06/30/2011  07:17 AM    <DIR>          ..
06/30/2011  07:36 AM                22,526 a.exe
06/30/2011  07:27 AM                85 firstprogram.c
               2 File(s)                22,611 bytes
               2 Dir(s)  1,046,056,108,032 bytes free

E:\cprog>a
This is a C-program!
My first!

E:\cprog>
    
```

Vi ser ovan anropet till `gcc`, "`gcc firstprogram.c`". Om man vill skriva ett annat program som gör något annat får man lägga det programmet i en annan fil och när man vill kompilera och länka det programmet måste man ange det filnamnet till `gcc`. Vi ser sedan ytterligare ett anrop till `dir` som

listar de filer som är tillgängliga. Vi ser nu att det dykt upp ytterligare en fil som heter *a.exe*. I bakgrunden till kommandorutan ovan ser vi hur det ser ut grafiskt, textfilen *firstprogram.c*, tillsammans med den nya filen *a.exe*. Den nya filen är på binärt format och är ett körbart program. Vi kör programmet vid prompten genom att bara skriva *a*. Det är det vi ser allra sist och här ser vi också resultatet av körningen, en enkel utskrift av texten "This is a C-program!" och på nästa rad "My first!"

Här kan vi göra vår första viktiga observation: Varje C-program har två delar, textfilen som människan skriver och den körbara filen som kompilatorn skapar som datorn kan köra. De brukar kallas källkodsfil respektive objektkodsfil. Människan skriver C-kod och lägger den i källkodsfiler. Sedan anropar människan en kompilator som skapar en objektkodsfil, alltså en körbar fil.

Detn här uppgiften som du löst i och med att du fått igång ett C-program är den andra uppgiften som du ska redovisa i denna kurs. Det första som du skulle redovisa var de där uppgifterna i Guido-van-Robot. När de uppgifterna är redovisade tillsammans med den här uppgiften som innebär att du visar ett körande C-program så är den här lilla kursen fullbordad. Hurra!

Om du vill kan du nu göra någon enkel tutorial på till exempel www.cprogramming.com. Men snart börjar ju huvudkursen, HI1024, Programmering, grundkurs.