

Datorkonto

- Samma användarnamn som för Lukas-kontot (e-post m.m.)
- Samma lösenord (så länge man inte byter lokalt)
- Kåavgift → Registrering → Lukas-konto → IDA-konto
- Särskilda ordningsregler för IDA-konto (se webbkursmaterialet)

Hur mycket måste man kunna?



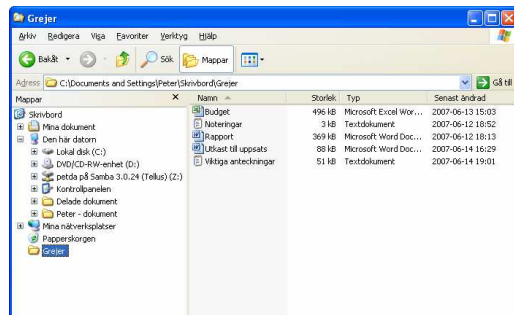
Några allmänna åsikter

- En professionell datoranvändare behöver inte vara administratör.
- Det är viktigt att behärska många olika verktyg och arbetsätt.
- Det tar ofta lika lång tid att tweeka systemet så att det blir som man är van vid, som att helt enkelt lära sig att det funkar olika på olika ställen.
- Endast dåliga hantverkare skyller på verktygen.

Två sätt att använda datorn

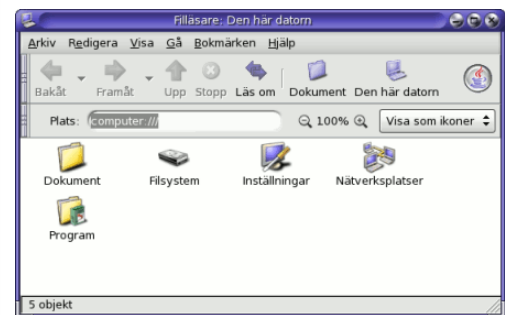
- Fönstersystem
 - Varje program körs i ett eller flera *fönster* som har en grafisk representation.
 - Man kommunicerar med både mus och tangentbord.
- Kommandoskal
 - Programmen visar enbart text.
 - Man kommunicerar genom att skriva korta kommandon.

Filhantering i fönstersystem



Utforskaren i Microsoft Windows

Filhantering i fönstersystem



Filhanteraren i Java Desktop System

Filhantering i kommandoskal



```
Terminal
Arktiv Redigera Visa Terminal Filkar Hjälp
zaza8 <1> pwd
/home/petjo450
zaza8 <2> ls
ada          lisp          old           pub
bin          mail          pascal        www-pub
exempelfil.txt Mail          pc
zaza8 <3> cd pascal
zaza8 <4> ls
matrix.p     perfect.txt  primesum.p
perfect.p    perm.p      untar.pas
zaza8 <5>
```

Exempel

```
zaza8 <1> pwd
/home/petjo450/
zaza8 <2> ls
ada          lisp          old           pub
bin          mail          pascal        www-pub
exempelfil.txt Mail          pc
zaza8 <3> cd pascal
zaza8 <4> ls
matrix.p     perfect.txt  primesum.p
perfect.p    perm.p      untar.pas
zaza8 <5>
```

Fördelar och nackdelar

- Man blir mycket mer effektiv när man lärt sig kommandospråket
- Man kan utföra mer komplexa uppgifter.
- Inläringströskeln är högre jämfört med grafiskt gränssnitt.
- Det kan vara svårt att få överblick eftersom man får hålla saker i huvudet.

Bakgrund till UNIX



Bakgrund till UNIX

- Nya idéer: *Multiuser, interactive, time-sharing operating system*
- CTSS (1961) från MIT
- MULTICS (1962) med MIT, GE, Bell Labs
- UNIX (1969) från Bell Labs

Tidiga riktlinjer för UNIX

1. Varje program ska utföra en liten väldefinierad uppgift och göra det bra
2. Förutsatt att ett programs utdata kan bli indata till ett annat program
3. Utveckla program som går att testa tidigt

Man ville skapa en gemenskap!

