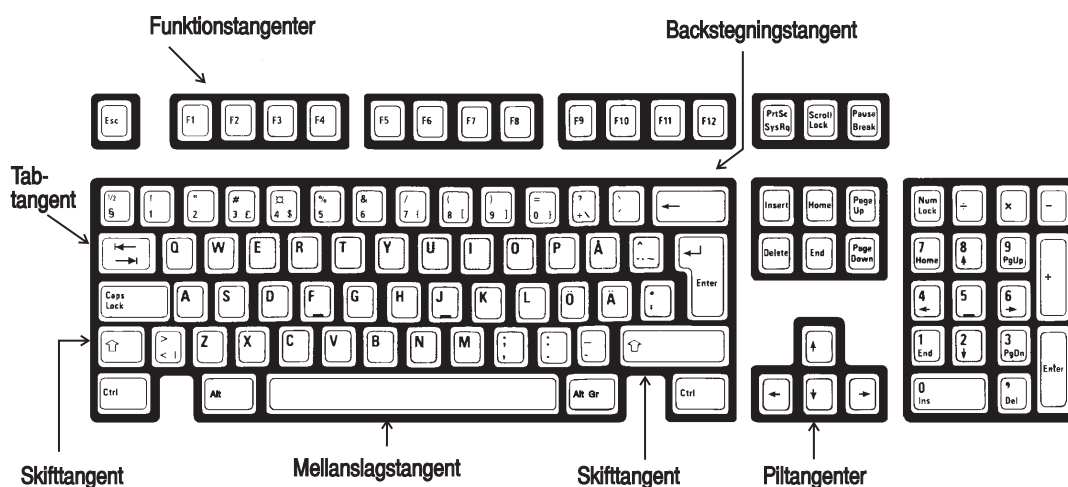


3. Kom igång med Unix

I detta kapitel beskrivs hur ”inloggning” och ”utloggning” går till och i anslutning till detta behandlas val av *lösenord*. Därefter introduceras några enkla kommandon, följt av mer principiell diskussion om hur kommandon ser ut och används (bl a behandlas och exemplifieras begreppet *argument*). Kapitlet avslutas med en introduktion till det grafiska *Fönstersystemet X*.

3.1 Tangentbordet

Tangentbordets utseende kan variera. Figuren nedan visar ett svenskt tangentbord till en modern IBM-kompatibel persondator (PC).



Figur 3.1: Tangentbord

På den vänstra, större delen av tangentbordet ligger bokstavs- och sifvertangenter. Dessa brukar med ett gemensamt namn kallas för *alfanumeriska* tangenter. Övriga tangenter på denna del av tangentbordet har följande funktioner (se figuren):

- Return** Returtangent, ibland märkt *Enter* och/eller med ↵. Används för att avsluta radinmatning och nedtrycks alltid för att utföra textbaserade kommandon.
- Skift** Skifttangent. När denna tangent är nedtryckt fås stora bokstäver resp de tecken som står uppe till vänster på tangenterna. När den släpps återgår skriften till det ”normala”. OBS: Tangentbordet har två skifttangenter, en på vardera sidan.
- Ctrl** Kontrolltangent. Används ihop med andra tangenter ungefär som skift-tangenten, dvs den ändrar de ”vanliga” tangenternas funktion. När *Ctrl*-tangenten ska nedtryckas tillsammans med en annan tangent skriver vi t ex *Ctrl-D* eller ^d, vilket betyder att *Ctrl* ska hållas nedtryckt när *D*-tangenten trycks. Moderna tangentbord brukar ha två *Ctrl*-tangenter.
- Alt** Alt-tangent. Används i kombination med andra tangenter ungefär som *Ctrl*-tangenten. Moderna PC-tangentbord brukar ha två *Alt*-tangenter.
- Caps Lock** Skiftlås. ”Låser” bokstavstangenterna så att stora bokstäver erhålls.

Backspace Backstegningstangent. Flyttar markören ett steg åt vänster och raderar det tecken som passerar. Tangenten används för att rätta felaktigt inskriven text.

Esc Escapetangent. Används bara i vissa program (t ex editorerna `vi` och `emacs`).

Tab Tabulatorstangent. Flyttar markören till nästa tabulatorstopp.

Av övriga tangenter används bl a piltangenterna i vissa program (editorer).

Olika teckenuppsättningar

Moderna system brukar kunna arbeta med 8-bitars ASCII, varvid svenska tecken inte bereder några problem (då utnyttjas vanligen samma teckenuppsättning som i Windows, *Latin 1*, eller i vissa fall *UTF-8* som är knutet till 16-bitars teckenkoden Unicode). Om man är hänvisad till 7-bitars ASCII är det värre. Då har man *antingen* tillgång till svenska *eller* amerikanska tecken. Om du har möjlighet att välja bör du i så fall offra de svenska till förmån för de amerikanska, ty flera amerikanska tecken har specialfunktioner i Unix (t ex hakparenteserna []), varför det är betydligt krångligare att ge vissa kommandon med svensk 7-bitars teckenuppsättning.

Om du måste arbeta med den svenska 7-bitars teckenuppsättningen, kan du ha nytta av tabellen nedan, som visar de svenska motsvarigheterna till de amerikanska tecknen.

Exempel: Om ett kommando ska innehålla tecknet | skriver man med svensk 7-bitars ASCII ö (litet ö) i stället.

Amerikansk ASCII	Svensk ASCII
@	É
\$	å
]	Å
[Ä
\	Ö
^	Ü
`	é
}	å
{	ä
	ö
~	ü

3.2 In- och utloggning

I Unix/Linux måste varje användare ha ett *användarkonto*. Detta ordnas av systemadministratören (kallas *root* i Unix), som ger den nye användaren ett *användarnamn* (eller *loginnamn*) och oftast också en hemlig kod, ett *lösenord* (eng password).

Varje gång man vill arbeta med systemet måste man identifiera sig med användarnamn och (vanligen även) lösenord, för att verifiera att man är den som man utger sig för att vara. Detta kallas för att *logga in* eller *logga på*. Loginnamnet och lösenordet måste anges korrekt. Om du glömmer ettdera kan du inte logga in och måste be systemadministratören om hjälp.

Inloggningsförfarandet försvårar för obehöriga att få tillgång till systemet. Det gör det också möjligt att ha privata filer, dvs man kan hindra att andra får tillgång till de egna filerna. En annan fördel med inloggning är att den gör det möjligt att anpassa datormiljön efter egna önskemål (även om datorn används av flera personer).

Om systemet är anslutet till ett nätverk, i synnerhet Internet, är det särskilt viktigt att inte slarva med användarnamn och lösenord. Det finns nämligen individer som finner stort nöje i att nästla sig in i och ibland även sabotera andra datorsystem.

Inloggning kan ske direkt vid Unix/Linux-datorn (arbetsstationen) eller från en terminal som på något sätt är ansluten till Unix/Linux-datorn. Terminalen kan vara en textterminal (som enbart arbetar i textläge) eller en grafisk terminal (där det grafiska gränssnittet vanligen är baserat på Fönstersystemet X, se nedan).

Innan inloggning kan ske från en terminal måste denna stå i kontakt med Unix-datorn. Används en PC som terminal måste dessutom ett terminalprogram startas på PC:n, t ex *telnet*, som ingår som standard i Windows, eller (om högre säkerhet eftersträvas) någon variant av *Secure Shell*, såsom *putty*.

Inloggningsförfarandet är i princip detsamma, oavsett om det sker i textläge eller i grafikläge: Man får skriva in sitt användarnamn och lösenord. Vi exemplifierar först med hur inloggning i textläge går till.

När kontakt upprättats med systemet möts du av texten (ibland krävs ett eller ett par tryck på **Return**-tangenter för att systemet ska reagera):

```
login:
```

Du skall nu skriva in ditt användarnamn. Gör detta och tryck **Return**:

```
login: loginnamn Return
```

Normalt ska loginnamnet skrivas med små bokstäver, ty i annat fall kan systemet tro att terminalen bara klarar stora bokstäver, vilket kan medföra problem.

Om du har ett lösenord fås följdfrågan:

```
Password:
```

Inmata lösenordet och tryck **Return** (OBS: Normalt syns inget på skärmen):

```
Password: lösenord Return
```

Saknas lösenord kan det hända att du inte får någon sådan fråga och en av dina första åtgärder bör vara att skaffa ett (hur det går till ska vi snart titta på).

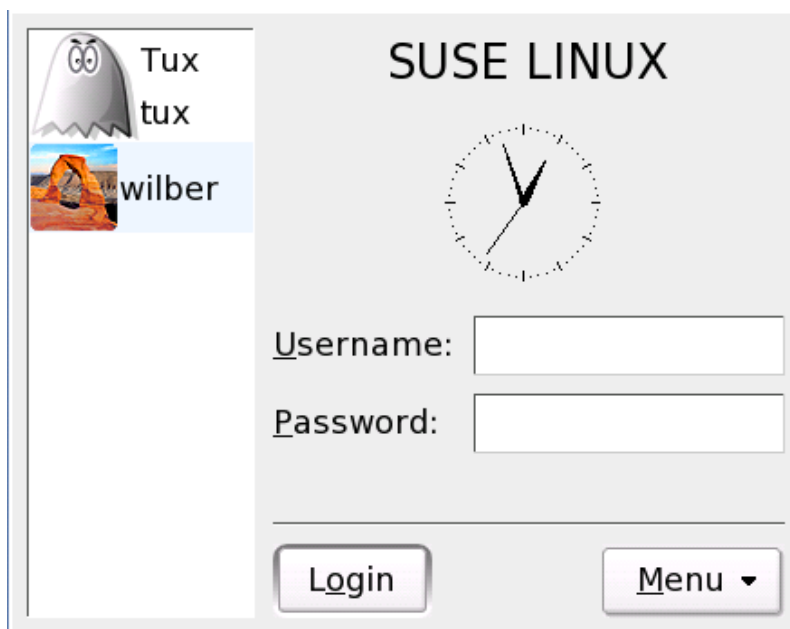
Om användarnamnet och det eventuella lösenordet accepterats är du därmed inloggad. Vad som sedan händer kan variera. Ofta visas tidpunkten för din senaste inloggning, olika meddelanden ges osv. När preludierna avklarats visas den s k systempromptern, som anger att systemet är redo att ta emot kommandon (systempromptern är ofta ett \$ eller % eller slutar på något av dessa tecken).

Skulle loginnamnet eller lösenordet vara felaktigt meddelar systemet detta, t ex:

```
Login incorrect
login:
```

Du får alltså en ny chans. Vissa system tillåter hur många inloggningsförsök som helst, medan andra godtar ett visst antal missar, och kopplar sedan ned förbindelsen. Misslyckade inloggningsförsök brukar registreras ("loggas"), vilket gör det möjligt att upptäcka om obehöriga försökt ta sig in i systemet.

Om inloggning sker i grafikläge (via Fönstersystemet X) brukar man mötas av en inloggningsruta, som exempelvis kan se ut så här:



Figur 3.2: Grafisk inloggning med SuSE Linux

Även här gäller det att mata in logginnamn och lösenord. Det som skiljer grafisk inloggning från textbaserad är främst kosmetika. Dessutom finns ofta andra alternativ att välja mellan (se bilden).

I fortsättningen förutsätter vi tillgång till en textbaserad kommandotolk. Om du loggat in i grafikläge så har du antingen redan fått ett sådant "textfönster" eller så får du öppna ett (oftast finns någon knapp eller något menyalternativ för att ordna detta). För mer information om grafiska användargränssnitt, se avsnittet *Introduktion till X* nedan.

För att logga ut från en terminal ger du kommandot:

exit *Return*

eller:

logout *Return*

Ibland kan man även logga ut genom att hålla *Ctrl*-tangents nedtryckt och samtidigt trycka *D*. Detta kan vi skriva:

^d (*Ctrl-D*)

Om man loggat in grafiskt kommer dock kommandot ovan att enbart avsluta det aktiva terminalfönstret. För att logga ut kan man i sådana fall tillgripa speciella menyalternativ.

När du loggat ut kan du lämna datorn/terminalen. Om du sitter vid en terminal kan du också stänga av denna (däremot bör man absolut inte stänga av själva Unix-datorn).

Anmärkning: I grafiska miljöer (t ex Windows eller vid inloggning direkt till en Unix-dator) kan man oftast öppna flera terminalfönster, vilket kan vara mycket fördelaktigt i och med att man kan arbeta med flera saker samtidigt, kopiera mellan fönster m m.

3.3 Lösenord

Användarnamnet fastställs av systemadministratören och kan inte ändras av vanliga användare. Lösenordet brukar man dock kunna byta ut.

Oavsett om du redan fått ett lösenord eller inte, så bör du snarast skaffa ett *eget*.

Lösenord kan bestå av minst 8 godtyckliga tecken (moderna system accepterar betydligt längre lösenord). Vanligen kontrolleras att det inte är för lätt att gissa (det får t ex inte vara samma som loginnamnet eller ord som "kilroy"). Ofta krävs också att det innehåller minst en icke-bokstav och att det inte är för kort (t ex minst 6 tecken). Ibland måste det dessutom bytas ut med jämna mellanrum.

För att byta lösenord, ge kommandot:

```
passwd Return
```

Ifall du redan har ett lösenord svarar systemet:

```
Old password:
```

Skriv det gamla lösenordet och tryck **Return** (inget syns på skärmen). Det gamla efterfrågas för minska risken att ändringen utförs av någon obehörig.

Det kan nu hända att systemet frågar om det ska förslå ett lösenord eller om du vill hitta på ett eget. Datorgenererade lösenord brukar bli väl krångliga och det är både roligare och lärorikare att hitta på egna, så välj i så fall att själv ange lösenord.

Du får därefter frågan:

```
New password:
```

Skriv det nya lösenordet och tryck **Return**. Om lösenordet godtas fås svaret:

```
Re-enter password:
```

eller något liknande. Det är nu meningen att du på nytt skall inmata lösenordet. Detta för att det verkligen ska bli korrekt (du kan ju ha skrivit fel första gången) och för att du skall kunna ångra dig. När du bekräftat lösenordet återkommer systempromptern.

3.4 Några enkla kommandon

I fortsättningen förutsätts att du är i din *hemkatalog* (home directory), den katalog där du hamnar vid inloggning (begreppet *katalog* behandlas grundligare i nästa kapitel). Vi "låtsas" också att du har användarnamnet `per`.

Testa om möjligt exemplen, men bli då inte förvånad över att utmatningen skiljer sig från den som visas här – vi kan ju inte veta ditt verkliga användarnamn osv.

Skriv (med *små* bokstäver):

```
date Return
```

Systemet svarar med aktuellt datum och klockslag, t ex (Linux):

```
tis jun 1 10:58:03 CEST 2004
```

För att ett kommando ska utföras måste **Return**-tangents nedtryckas. Detta låter vi i fortsättningen vara underförstått.

Om du skrivit fel (och det inte finns ett *annat* kommando som skrivs på just detta sätt) fås ett *felmeddelande* och du måste inmata kommandot på nytt:

```
datte
datte: command not found
```

Unix är ett fleranvändarsystem och därför kan andra vara inloggade samtidigt med dig (vid andra terminaler). För att undersöka detta kan du ge kommandot

```
who
```

Systemet svarar då t ex (Linux):

```
lotta      :0          Feb 18 13:08 (console)
per        pts/1      Feb 18 14:26 (kom.hem.se)
```

Detta betyder att två användare är inloggade: *lotta* vid systemets *konsol* (*console*), vilket är själva Linux-datorn, och *per* via en nätverksansluten terminal (*pts/1*). Båda loggade in den 18:e februari, *lotta* klockan 13:08, *per* klockan 14:26.

I stora system kan hundratals användare vara inloggade samtidigt, medan en persondator eller arbetsstation kanske bara har en användare åt gången.

Om du vill veta vilka filer som finns i hemkatalogen kan du använda kommandot **ls**, som "listar" kataloginnehåll. Ge kommandot

```
ls
```

Systemet svarar t ex:

```
adresser.txt      dokument      readme
```

I katalogen finns således tre filer.

Anmärkning: Utseenden av utmatningen från **ls**-kommandot kan variera en hel del (och kan konfigureras av den systemansvarige), exempelvis kan filerna visas i enbart *en* kolumn i st f flera.

Pröva:

```
echo Hej Unix
```

Systemet svarar:

```
Hej Unix
```

Kommandot **echo** upprepar, "ekar", den efterföljande texten (här: *Hej Unix*). Detta är kanske inte så meningsfullt, men genom att komplettera kommandot kan vi åstadkomma något betydligt intressantare. Ge kommandot

```
echo Hej Unix > memo1
```

Denna gång syns inget på skärmen. Istället har texten fram till ”större-än”-tecknet (>) lagrats på *filen* memo1. Detta är ett exempel på *omdirigering* (utmatningen från echo har ”omdirigerats” till filen memo1). Omdirigering behandlas närmare i kapitel 5.

Kontrollera att filen skapats:

```
ls
```

Systemet svarar:

```
adresser.txt      dokument          memo1             readme
```

Filen memo1 har alltså tillkommit. För att försäkra dig om att texten verkligen hamnat i memo1 ger du kommandot

```
cat memo1
```

Filinnehållet skrivs ut på bildskärmen:

```
Hej Unix
```

3.5 Rätta skrivfel

Skrivfel kan rättas innan *Return*-tangents nedtrycks. Enklast görs detta genom att ”backa över” den felaktiga texten med *Backspace* och sedan skriva in den korrekta.

Om man givit ett felaktigt kommando eller av någon annan anledning vill avbryta ett kommando innan det slutförts, kan man trycka på ”avbrottstangenten”: *Del*- eller *Break*-tangents eller tangentkombinationen *Ctrl-C* (^c) – detta är systemberoende.

Anmärkning: De flesta kommandotolkar medger betydligt mer sofistikerade redigeringsmöjligheter än de som nämns ovan. Med *bash* och *tcsh* kan man både förflytta sig på den aktuella kommandoraden med vänster- och högerpilarna (och göra ändringar) och ”bläddra” mellan gamla kommandon med pilupp- och pilner-tangenterna. Båda dessa kommandotolkar tillhandahåller även sk tabb-expansion, vilket betyder att kommandon och filnamn kompletteras automatiskt när man skrivit in en del av kommandot/filnamnet och sedan trycker *Tab*-tangents. Se kapitel 9, avsnittet ”Kommandoredigering”, för mer information.

3.6 Allmänt om Unix-kommandon

De enklaste kommandona består av endast kommandots namn, t ex

```
ls
```

Ofta måste dock kommandot kompletteras med ett eller flera *argument*, som anger *vad* som ska bearbetas (t ex en fil) eller ändrar på kommandots funktion i något avseende. Det allmänna kommandoformatet är:

```
kommandonamn argument1 argument2 ...
```

Kommandoraden inleds med ett *kommandonamn* (t ex *ls*). Nästan alla kommandonamn skrivs med *små* bokstäver (Unix skiljer mellan stora och små). Kommandonamn och de olika argumenten måste åtskiljas med minst ett blanktecken.

De vanligaste argumenten är ett eller flera *filnamn*, t ex

```
cat textfil
```

och *alternativ* (*flaggor* eller *växlar*), som ändrar kommandots beteende, t ex

ls -l

vilket ger en sk *lång* listning av innehållet i aktuell katalog. Alternativ inleds för det mesta med ett minustecken (**-l** i exemplet ovan).

En tredje (mindre vanlig) argumenttyp är *uttryck* av olika slag.

När två eller flera argumenttyper ingår i ett kommando är ordningen för det mesta:

kommandonamn alternativ uttryck filnamn

Lägg märke till att alternativ skrivs direkt efter kommandonamnet, dvs före eventuella uttryck och filnamn. Ordningen kan dock vara annorlunda, varför man bör rådfråga dokumentationen för det aktuella kommandot om man är osäker.

Som ett exempel på kommandon med argument kan vi ta **cal**-kommandot.

cal skriver ut kalendrar mellan år 1 och år 9999. Kalendrar kan erhållas för en månad eller ett helt år. Utan argument:

cal

visas aktuell månad (vissa system visar också månaden före och efter), t ex

```

      February 2003
Su Mo Tu We Th Fr Sa
          1
  2  3  4  5  6  7  8
  9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28

```

Moderna system brukar kunna konfigureras så att de använder olika språk (t ex svenska), vilket påverkar utmatningen från kommandon som **cal**. Om detta gäller för ditt användarkonto, då kommer kalendrar att visas i detta språk i stället för engelska.

Ett argument (ett tal från 1 till 9999) tolkas alltid som ett årtal. Kommandot

cal 2004

ger följaktligen en kalender för hela 2004.

Om vi ger ytterligare ett argument före årtalet, ett tal från 1 till 12, tolkas detta som ett månadsnummer och vi får en kalender för denna månad och år. Om systemet är inställt på att använda svenska kommer kommandot:

cal 2 1600

att visa:

```

      februari 1600
sö må ti on to fr lö
          1  2
  3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29

```