

# Modul 1

Utforska filsystemet

Sebastian Bengtegård, Johan Holmberg

22 Januari 2025

## Innehållsförteckning

<b>1</b>	<b>Förord</b>	<b>2</b>
<b>2</b>	<b>Filsystemet</b>	<b>2</b>
<b>3</b>	<b>pwd: Var är jag någonstans?</b>	<b>2</b>
<b>4</b>	<b>cd: Navigera i filsystemet</b>	<b>3</b>
4.1	Absoluta sökvägar . . . . .	3
4.2	Relativa sökvägar . . . . .	4
4.3	Några genvägar . . . . .	5
<b>5</b>	<b>ls: Lista filer</b>	<b>6</b>
5.1	En närmare titt på <i>Long Format</i> . . . . .	6
<b>6</b>	<b>man: Ett enkelt manualkommando</b>	<b>7</b>
<b>7</b>	<b>Övning</b>	<b>8</b>
7.1	Klara, färdiga, gå! . . . . .	8
7.1.1	MacOS: Operation not permitted . . . . .	9

## 1 Förord

I denna modul kommer vi att introduceras till fyra nya kommandon:

- `pwd` (*print working directory*)
- `cd` (*change directory*)
- `ls` (*list files and directories*)
- `man` (*manual pages*).

Genom dessa nya kommandon kommer vi att kunna utforska vårt filsystem. Med “att utforska” menar vi att kunna se vilken mapp vi befinner oss i, vad den innehåller för filer och att kunna förflytta oss till en annan mapp. Vi kommer även att gå igenom hur vi kan ta reda på mer information om ett kommando genom dess manual.

Modulen avslutas sedan med en övning.

## 2 Filsystemet

Nästan alla datorer har någon form av ett system för sina filer (och i sin tur mappar), för att vi ska kunna organisera och strukturera allt vi har på vår dator (så som program, bilder, videos, m.m.). Detta är vad vi benämner som ett så kallat “filsystem”.

Detta system har vanligen en struktur likt ett träd. Där vi har en rot och från denna kommer trädstammen med alla sina grenar (våra mappar), som i sin tur består av löv (våra filer).

Beroende på vilket operativsystem du har finns det antingen en rot (även kallad “rotmappen”) eller flera. Till exempel har både MacOS och Linux en rotmapp (som brukar skrivas som ett snedstreck: /), och Windows har flera (där respektive rotmapp börjar med en bokstav, t.ex. C:/ och D:/).

Anledningen till varför detta är bra att känna till är för att när vi interagerar med skalet har vi inget grafiskt gränssnitt. Vi måste med andra ord bilda oss en uppfattning om hur vi till exempel kan nå en viss fil eller mapp. På så vis blir vårt filsystem en slags labyrint där vi tack vare trädstrukturen ändå har en idé om hur vi ska “hitta vägen” till dessa filer och mappar. Men för att göra detta behöver vi använda oss av våra första kommandon.

## 3 `pwd`: Var är jag någonstans?

Eftersom vi, i skalet, inte har ett grafiskt gränssnitt för vårt filsystem måste vi kunna representera detta på ett annorlunda vis. För att göra detta kan vi tänka oss att vårt

filsystem är som en labyrint och vi befinner oss någonstans i denna. Det enda vi kan se från vart vi står är vägen till föräldermappen (en trappa upp), filerna runt omkring oss och vägen till undermapparna (en trappa ner) från vår plats.

Den plats vi befinner oss i kallas för *working directory* och för att se vad denna plats heter använder vi kommandot `pwd` (*print working directory*).

```
sebastian@mindator:~$ pwd
/home/sebastian
```

När vi startar en terminal placerar den oss i vår hemmapp (*home directory*) som vår start-position. Beroende på operativsystem så kan namnet på denna plats skilja sig, till exempel på MacOS och Windows heter den `/Users/sebastian` respektive `/c/Users/sebastian`.

För att titta oss omkring använder vi kommandot `ls` (*list files and directories*).

```
sebastian@mindator:~$ ls
Desktop  Documents  dog.jpg    Downloads  example.txt
Mail     Music      Pictures   Videos
```

Vi kommer gå igenom kommandot `ls` i ett senare avsnitt. Det finns mycket vi kan göra med detta kommandot, men innan vi fortsätter med det måste vi lära oss lite om sökvägar och hur vi kan navigera filsystemet.

## 4 cd: Navigera i filsystemet

För att förflytta oss i labyrinten använder vi kommandot `cd` (*change directory*), som ändrar vår nuvarande position (*working directory*) till en annan. Detta sker genom att vi skriver `cd` följt av en *sökväg*, till exempel `cd Pictures` (notera mellanslaget). Sökvägen är den plats vi vill förflytta oss till i vårt filsystem och kan skrivas på två olika sätt: *absoluta sökvägar* och *relativa sökvägar*. Låt oss börja med absoluta sökvägar.

### 4.1 Absoluta sökvägar

En absolut sökväg börjar från starten av labyrinten (den sk. rotmappen, där vårt filsystem börjar) och följer sedan vägen fram till den mapp eller fil vi är ute efter. Till exempel är den absoluta sökvägen till min hemmapp `/home/sebastian`. Detta översätts till att rotmappen (det inledande snedstrecket) innehåller mappen `home` som i sin tur innehåller mappen `sebastian`. Notera att varje mapp som vi “går ner i” separeras med ett snedstreck.

För att förflytta oss till mappen “Pictures” i vår hemmapp skriver vi följande:

```
sebastian@mindator:~$ cd /home/sebastian/Pictures
sebastian@mindator:~/Pictures$ pwd
/home/sebastian/Pictures
sebastian@mindator:~/Pictures$ ls
Animals  Cats  Dogs  funny-dog.png  Originals  work-profile.jpg
```

Notera att Bash skiljer på stora och små bokstäver. Det är därför viktigt att vi skriver ut namnen precis som de står i listningen.

Här ser vi att vi ändrar vår *working directory* och att mappen “Pictures” även innehåller lite olika filer och mappar. Notera också att vår skalprompt har ändrats till `sebastian@mindator:~/Pictures$`. Vårt skal (Bash) är vanligtvis inställt på att visa vår nuvarande *working directory*.

*Det gör ingenting om din skalprompt inte ser ut på samma vis som vår.*

## 4.2 Relativa sökvägar

Där en absolut sökväg startar från rotmappen till en exakt plats, startar en relativ sökväg från vår nuvarande position (*working directory*). För att kunna göra detta finns det ett par notationer som representerar relativa positioner. Dessa är `.` (punkt) och `..` (punkt punkt).

Notationen `.` refererar till vår *working directory* och `..` refererar till föräldermappen av vår *working directory*. Låt oss utgå från mappen “Pictures” igen.

```
sebastian@mindator:~$ cd /home/sebastian/Pictures
sebastian@mindator:~/Pictures$ pwd
/home/sebastian/Pictures
```

Om vi nu vill återgå till vår hemmapp `/home/sebastian` kan vi antingen använda oss av en absolut sökväg:

```
sebastian@mindator:~/Pictures$ cd /home/sebastian
sebastian@mindator:~$ pwd
/home/sebastian
```

Eller, en relativ sökväg:

```
sebastian@mindator:~/Pictures$ cd ..
sebastian@mindator:~$ pwd
/home/sebastian
```

Vi kan åstadkomma samma resultat med båda. Vilken ska vi använda? Den som kräver minst skrivande!

Skulle vi vilja ändra vår *working directory* från vår hemmapp till “Pictures” kan vi med en absolut sökväg skriva:

```
sebastian@mindator:~$ cd /home/sebastian/Pictures
sebastian@mindator:~/Pictures$ pwd
/home/sebastian/Pictures
```

Eller, med en relativ sökväg:

```
sebastian@mindator:~$ cd ./Pictures
sebastian@mindator:~/Pictures$ pwd
/home/sebastian/Pictures
```

Detta är så vanligt förekommande att vi kan faktiskt utelämna `./` eftersom det är underförstått i vårt skal. Vi kan alltså istället skriva:

```
sebastian@mindator:~$ cd Pictures
sebastian@mindator:~/Pictures$ pwd
/home/sebastian/Pictures
```

Vilket hade gett oss samma resultat.

### 4.3 Några genvägar

Om vi skriver `cd` följt av *ingenting* kommer vår *working directory* ändras till vår hemmapp, vilket är ett snabbt sätt att komma hem på.

Skriver vi `cd -` (bindestreck) ändras vår *working directory* till den föregående *working directory* vi befann oss i. Vilket innebär att vi enkelt kan hoppa fram och tillbaka mellan två mappar.

Eftersom vi vanligtvis utgår från vår hemmapp när vi förflyttar oss så finns det även ett tecken som representerar denna, vilket är tildetecknet (`~`). Detta ger oss möjlighet att skriva kortare sökvägar utifrån vår hemmapp. Skulle vi till exempel skriva `cd ~/Pictures` är det detsamma som `cd /home/sebastian/Pictures`, så i vårt fall är tildetecknet synonymt med `/home/sebastian`.

Om vi även tar en titt på vår skalprompt igen kan vi se att tildetecknet (`~`) används som en representation av vår hemmapp: `sebastian@mindator:~$`.

## 5 ls: Lista filer

Kommandot `ls` (*list files and directories*) används för att lista innehållet i en mapp och är förmodligen det mest använda kommandot av dem alla. Vi kan även styra hur innehållet listas, här är några exempel:

Table 1: Några exempel på kommandot `ls`

Kommando	Resultat
<code>ls</code>	Lista filerna i vår <i>working directory</i> .
<code>ls /home</code>	Lista filerna i mappen <b>/home</b> .
<code>ls -l</code>	Lista filerna i vår <i>working directory</i> i <i>long format</i> (se nedan).
<code>ls -l /etc /home</code>	Lista filerna i mapparna <b>/etc</b> och <b>/home</b> i <i>long format</i> .
<code>ls -la</code>	Lista alla filer (inklusive dolda filer*) i vår <i>working directory</i> i <i>long format</i> .

Dessa exempel visar en viktig aspekt i hur kommandon fungerar. De flesta kommandon fungerar på följande vis:

```
kommando -tillval argument
```

Där *kommando* är namnet på kommandot, *-tillval* är ett eller flera justeringar på kommandots beteende, och *argument* är en eller flera “saker” som kommandot ska operera på.

I fallet med kommandot `ls` kan vi se att namnet är **ls**, det kan ha ett eller flera tillval så som **-l** och **-a** (vilket kan kombineras till **-la**), och det kan operera på en eller flera mappar.

\* de filer vars namn som börjar med en punkt är vanligtvis så kallade “dolda” filer.

### 5.1 En närmare titt på *Long Format*

Om vi till kommandot `ls` använder tillvalet **-l** (*long format*) kommer vår lista över filer innehålla en hel del information.

```
sebastian@mindator:~$ ls -l
-rw-r--r-- 1 sebastian sebastian 3526 Nov 10 22:16 example.txt
drwxr-xr-x 6 sebastian sebastian 4096 May 5 09:45 Desktop
drwxr-xr-x 3 sebastian sebastian 4096 Mar 27 18:06 Documents
drwxr-xr-x 4 sebastian sebastian 4096 May 4 10:36 Downloads
```

```
-rw-r--r-- 1 sebastian sebastian    252 Feb  3 14:28 essay.pdf
...
```

Informationen är uppdelad i ett antal kolumner från vänster till höger. Vi utgår från den översta filen:

1. **File permissions** `-rw-r--r--`. Det första tecknet representerar vilkentyp av fil det är. Ett `-` innebär att det är en vanlig fil och `d` att det är en mapp. De resterande tecknen är indelade i grupper om 3 (`rw`), som representerar vem som har åtkomst och vad dem kan göra med filen, detta kommer vi gå igenom i en senare modul.
2. **Number of hard links** 1. Inget vi kommer att gå igenom men skulle du vara nyfiken kan du [läsa mer här](#).
3. **Owner** `sebastian`. Namnet på den användare som äger filen.
4. **Group** `sebastian`. Namnet på den grupp (av användare) som har tillgång till filen utöver ägaren.
5. **Size** 3526. Filstorleken i *bytes*.
6. **Modification Time** Nov 10 22:16. Den senaste tiden då filen var modifierad.
7. **File Name** `example.txt`. Namnet på filen eller mappen.

## 6 man: Ett enkelt manualkommando

Kommandot `man` används för att läsa manualer om andra kommandon. Detta är väldigt praktiskt om vi till exempel vill upptäcka nya tillval eller har glömt bort hur vi använder ett kommando.

Vi skriver `man` följt av det kommando vi vill läsa manualen för, till exempel `man ls`. Eftersom vi inte arbetar med ett grafiskt gränssnitt behöver vi veta hur vi kan kontrollera programmet. För att stänga ner `man` trycker vi på q-tangenten. Här är ett urval av några vanliga kommandon:

Table 2: Ett urval av kommandon för `man`

Kommando	Funktion
Space eller <code>d</code>	Bläddra framåt en sida.
<code>b</code> eller <code>u</code>	Bläddra tillbaka en sida.
Pil-ned eller <code>j</code>	Skrolla fram en rad.
Pil-upp eller <code>k</code>	Skrolla tillbaka en rad.
<code>G</code>	Gå till slutet av manualen.
<code>g</code>	Gå till början av manualen.

Kommando	Funktion
<code>/sökterm</code>	Sök efter <i>sökterm</i> i manualen (observera snedstrecket).
<code>n</code>	Upprepa föregående sökning.
<code>q</code>	Stäng ner manualen.

Alla operativsystem har dessvärre inte tillgång till dessa manualer på förhand. Som tur är har de flesta kommandon tillvalet `--help` för att skriva ut information om hur kommandot i fråga används. Till exempel om vi skulle vilja ta reda på mer information om kommandot `ls` hade vi skrivit följande:

```
sebastian@mindator:~$ ls --help
```

Skulle tillvalet `--help` inte fungera så finns de flesta manualer även online, t.ex. kan du [klicka här för att komma till manualen för kommandot `ls`](#). En enkel sökning online ger dig säkerligen också goda resultat.

## 7 Övning

För att bli varm i kläderna behöver vi praktiskt öva på dessa nya kommandon. [Ladda därför ned denna zipfil](#), öppna den, och placera den sedan i en lämplig mapp, till exempel på “Skrivbordet”.

Under övningens gång kommer vi att anteckna ner vissa fynd vi gör, till exempel hur vi skrev ett kommando eller utskriften från ett kommando. I övningsfrågorna kommer vi be er anteckna ner dessa fynd under en viss etikett i form av **A1**, **A2**, **A3** och så vidare, bara för att vi ska kunna hålla ordning på alla våra fynd. Skriv ner dessa på en lämplig plats, såsom ett dokument på datorn eller med papper och penna.

### 7.1 Klara, färdiga, gå!

Börja med att öppna en ny terminal och kontrollera via kommandot `pwd` att vi faktiskt är i vår hemmapp. Navigera sedan till den plats där du placerat övningsmappen (*exercise-module-01*). Om du valde platsen “Skrivbordet” är det vanligen en mapp som kallas för *Desktop* i din hemmapp. Anteckna ner vilket/vilka kommandon du skrev för att navigera till övningsmappen som **A1**.

- I övningsmappen finns det en hel del filer. Men hur många av dessa är mappar? Använd kommandot `ls` för att lista ut detta. Anteckna ner antalet mappar som finns i övningsmappen som **A2**.



*MacOS: Om du får meddelandet “Operation not permitted” behöver du ge din terminal mer rättigheter, se nedan.*

- I övningsmappen finns det en dold fil som är tom (0 *bytes*). Anteckna ner namnet på den dolda filen i övningsmappen som **A3**.
- En av mapparna i övningsmappen innehåller filen `nemo.txt`. Använd kommandona `cd` och/eller `ls` för att lista ut vilken mapp som innehåller denna filen. Anteckna ner namnet på den mapp som innehåller `nemo.txt` som **A4**. Glöm inte bort skillnaden mellan absoluta och relativa sökvägar.
- Med hjälp av kommandot `man`, ta reda på vad tillvalet `-h` (liten bokstav) innebär för kommandot `ls`. Anteckna ner innebörden av tillvalet `-h` i form av en kort mening som **A5**. Om `man` inte finns att tillgå får du pröva tillvalet `--help` istället, och om dessa inte fungerar kan du falla tillbaka till online versionen.
- I övningsmappen finner du filen `compat.c`. Vad är dess filstorlek i kilobytes (K)? Anteckna ner filstorleken av `compat.c` i kilobytes som **A6**. Ta samtidigt reda på när denna filen senast modifierades (datum och klockslag) och anteckna ner det som **A7** (om det var länge sedan filen modifierades står det inget klockslag utan bara ett årtal).
- Kommandot `ls` kan även ge oss utskrift i färg, ta hjälp av kommandot `man` för att lista ut vilket tillval som används för få färgad utskrift. Sök på termen “color” för att slippa läsa igenom samtliga tillval. Anteckna ner det tillval som ger färgad utskrift till `ls` som **A8**.

### 7.1.1 MacOS: Operation not permitted

Skulle du stöta på följande felmeddelande:

```
sebastian@mindator:~$ ls
ls: .: Operation not permitted
```

Kan du behöva gå in i systeminställningar och klicka i följande:



Figure 1: Operation not permitted