Modul 2

Arbeta med filsystemet

Sebastian Bengtegård, Johan Holmberg

22 Januari 2025

Innehållsförteckning

1 Förord

I denna modul kommer vi att introduceras till sex nya kommandon/program:

- `less` (*view text files*)
- `file` (*determine file type*)
- `cp` (*copy files and directories*)
- `mv` (*move or rename files and directories*)
- `rm` (*remove files or directories*)
- `mkdir` (*make directories*)

Genom dessa nya kommandon kommer vi att kunna arbeta med filsystemet, det vill säga att kopiera filer, flytta filer, radera filer och så vidare. Detta är däremot något som är väldigt enkelt att göra via ett grafiskt gränssnitt, till exempel via *Utforskaren* på Windows eller *Finder* på MacOS. Så varför ska vi göra detta via ett kommandoradsgränssnitt?

Svaret är att det är enklare att utföra mer komplexa uppgifter. Säg att vi skulle vilja kopiera bilder från en mapp till en annan, men bara de bilder som inte redan finns och de som är nyare än de befintliga. Det hade förmodligen tagit lite tid med ett grafiskt gränssnitt. Via vårt skal är det inte svårare än:

```
sebastian@mindator:~$ cp -u *.jpg destination
```

Hur detta kommando fungerar och varför får vi läsa mer om under avsnittet “cp”.

Modulen avslutas med en övning, men eftersom att denna modul handlar om att arbeta med filsystemet kan du redan nu ladda ner övningsmappen, om du vill ha något annat än dina egna filer att experimentera med.

2 `less`: Att läsa textfiler

`less` är ett program som låter oss läsa innehållet i *textfiler*, vilket är väldigt behändigt eftersom en stor del av de filer vi arbetar med, via vårt skal, är just textfiler.

Men, med det sagt måste vi först förstå *vad* en textfil egentligen är. Till exempel är ett Word-dokument eller en PDF **inte** en textfil, eftersom de kan innehålla allt från kursiverad text till bilder. En textfil består nämligen bara av text, det vill säga olika typer av tecken, så som “a, b, c” eller specialtecken som “!, #, $”.

För att läsa innehållet i en fil skriver vi `less` följt av sökvägen till en textfil, till exempel `less example.txt`. Eftersom vi inte arbetar med ett grafiskt gränssnitt behöver vi veta hur vi kan kontrollera programmet, vilket vi faktiskt redan gjort i “Modul 01” via kommandot

`man`. Kommandot `man` är nämligen bara en genväg till att läsa manualer via `less`, men vi slipper skriva (eller komma ihåg) sökvägarna till dessa manualer.

Så, återigen stänger vi `less` genom att trycka på "q". Och här är samma urval av några vanliga kommandon för att kontrollera `less`:

Table 1: Ett urval av kommandon för `less`

Kommando	Funktion
Space eller `d`	Bläddra framåt en sida.
`b` eller `u`	Bläddra tillbaka en sida.
Pil-ned eller `j`	Skrolla fram en rad.
Pil-upp eller `k`	Skrolla tillbaka en rad.
`G`	Gå till slutet av textfilen.
`g`	Gå till början av textfilen.
/sökterm	Sök efter *sökterm* i textfilen (observera snedstrecket).
`n`	Upprepa föregående sökning.
`q`	Stäng ner textfilen.

Om vi skulle råka öppna en fil som inte är en textfil kommer programmet `less` att säga ifrån. Skulle vi däremot på förhand vilja ta reda på vilken typ en fil är använder vi kommandot `file`.

3 `file`: Vad är det här för fil?

För att ta reda på vilken typ en fil är använder vi kommandot `file` följt av en sökväg. Detta kommando kan även i vissa fall ge oss mer information om filen än vilken typ det är, till exempel dimensionerna (höjd och bredd) av en bild.

```
sebastian@mindator:~$ ls
Desktop  Documents  dog.jpg  Downloads  example.txt
Mail     movie.mp4  Music    Pictures   Videos
sebastian@mindator:~$ file dog.jpg
dog.jpg: JPEG image data, ...
sebastian@mindator:~$ file example.txt
example.txt: ASCII text
sebastian@mindator:~$ file movie.mp4
movie.mp4: ISO Media, MP4 Base Media v1 ...
```

Det finns en uppsjö olika typer av filer och med hjälp av detta kommando kan vi, för det mesta, ta reda på vilken typ en fil är. Detta brukar även vara en bra utgångspunkt inför att göra en informationssökning på webben. Skulle vi till exempel vara osäkra på vad "ASCII text" är för någon typ av fil skulle vi kunna läsa mer om det på Wikipedia, eller någon annanstans.

4 Jokertecken

I introduktionen av denna modul fick vi se kommandot `cp -u *.jpg destination` och detta beskrevs som att vi skulle "kopiera **alla bilder** från en mapp till en annan [...]".

Eftersom det är så vanligt att vi arbetar med filnamn (dvs. sökvägar) erbjuder vårt skal några specialtecken för att underlätta detta. Dessa kallas för jokertecken (*wildcards*) och gör det möjligt att välja filer utifrån ett *mönster*, till exempel "alla bilder" eller "alla filer som börjar med bokstaven A".

I kommandot ovan har vi specialtecknet `*` (ett jokertecken) som översätts till "matchar alla tecken". Givet detta specialtecken kan vi forma många olika typer av mönster. Här är ett litet urval för att demonstrera några varianter:

Table 2: Några exempel på jokertecknet `*`

Mönster	Matchar
`*`	Alla filnamn.
`b*`	Alla filnamn som börjar med tecknet "b".
`*.jpg`	Alla filnamn som slutar på tecknen ".jpg".
`a*.txt`	Alla filnamn som börjar med tecknet "a" och slutar på tecknen ".txt".
`screenshot-*.png`	Alla filnamn som börjar med tecknen "screenshot-" och slutar på ".png".

Vi kan använda jokertecken i alla kommandon som arbetar med filnamn. Det finns fler jokertecken att upptäcka men det lämnar vi som en övning åt dig (kom ihåg att söka på termen *wildcards*).

5 cp: Kopiera filer och mappar

Kommandot `cp` används för att kopiera filer och mappar. I dess enklaste form kan vi kopiera en fil:

```
sebastian@mindator:~$ cp file1 file2
```

Notationen `file`, `file1` och `file2` är synonymt med sökvägar till filer, detsamma gäller även för `directory`, `dir1` och `dir2` fast för mappar.

Låt oss säga att vi vill kopiera filen `notes.txt` från mappen "Documents" till "Desktop", då hade vi skrivit följande:

```
sebastian@mindator:~$ cp Documents/notes.txt Desktop/notes.txt
```

Det kan även användas för att kopiera en eller flera filer (och/eller mappar) till en annan mapp:

```
sebastian@mindator:~$ cp file... directory
```

Notationen `...` innebär att vi kan skriva ett eller flera filnamn efter varandra, separerade med ett mellanslag. Denna notation används i manualerna och är något vi kommer att återanvända i våra beskrivningar här.

Låt oss säga att vi vill kopiera filerna `notes.txt` och `names.txt` från mappen "Documents" till "Desktop", då hade vi skrivit följande:

```
sebastian@mindator:~$ cp Documents/notes.txt Documents/names.txt Desktop
```

Här är några fler exempel på kommandot `cp` och dess tillval:

Table 3: Några exempel på kommandot `cp`

Kommando	Resultat
`cp file1 file2`	Kopierar innehållet från *file1* till *file2*. Om *file2* inte existerar, så skapas den, **annars kommer *file2* skrivas över med innehållet från *file1*.**
`cp -i file1 file2`	Som kommandot ovan, men med tillvalet `-i` (*interactive*) som innebär: om *file2* existerar kommer vi att behöva godkänna* kommandot först.
`cp file1 dir1`	Kopiera innehållet från *file1* (till en fil med namnet *file1*) till mappen *dir1*.

Kommando	Resultat
`cp -u file1 dir1`	Som kommandot ovan, men med tillvalet `-u` (*update*) som innebär: kopiera *file1* om den är nyare än den fil med samma namn i *dir1* eller om den inte existerar.
`cp -R dir1 dir2`	Kopiera innehållet från mappen *dir1*. Om mappen *dir2* inte existerar, så skapas den, annars skapas en mapp med namnet *dir1* inuti *dir2*.

** för att godkänna ett kommando skriver vi in* `y` *(kort för* yes*) och trycker sedan Enter, för att avböja skriver vi istället in* `n` *(kort för* no*) och trycker Enter.*

6 `mv`: Flytta filer och mappar

Kommandot `mv` används för att flytta eller döpa om filer och mappar. Antingen flyttar vi en eller flera filer från en mapp till en annan, eller så döper vi om en fil eller mapp från ett namn till ett annat. Ungefär som kommandot `cp` i föregående avsnitt.

För att döpa om något skriver vi så här:

```
sebastian@mindator:~$ mv file1 file2
```

Låt oss säga att vi skulle vilja döpa om filen `stuff.txt` till `notes.txt` i mappen "Documents", då hade vi skrivit följande:

```
sebastian@mindator:~$ mv Documents/stuff.txt Documents/notes.txt
```

Vi behöver alltså alltid skriva kompletta sökvägar, eftersom en dator inte kan förutsätta någonting.

Och om vi vill flytta en eller flera filer (och/eller mappar) till en annan mapp:

```
sebastian@mindator:~$ mv file... directory
```

Här är några fler exempel på kommandot `mv` och dess tillval:

Table 4: Några exempel på kommandot `mv`

Kommando	Resultat
`mv file1 file2`	Om *file2* inte existerar, så döps *file1* om till *file2*. **Om *file2* existerar kommer den att ersättas med innehållet av *file1*.**
`mv -i file1 file2`	Som kommandot ovan, men med tillvalet `-i` (*interactive*) som innebär: om *file2* existerar kommer vi att behöva godkänna kommandot först.
`mv file1 file2 dir1`	Flytta filerna *file1* och *file2* till mappen *dir1*. Om *dir1* inte existerar kommer kommandot att avbrytas.
`mv dir1 dir2`	Om *dir2* inte existerar, så döps *dir1* om till *dir2*. Om *dir2* existerar, så kommer *dir1* att flyttas in i *dir2*.

7 `rm`: Ta bort filer och mappar

Kommandot `rm` används för att radera filer och mappar.

Observera att vi raderar saker och ting **permanent**. Det vill säga **filer och mappar hamnar inte i en "papperskorg"**. **Detta kan skada din dator** om du inte är försiktiga! Är du osäker kan du alltid använda tillvalet `-i`. En bra tumregel är att skriva relativa sökvägar, då minskar du risken för att råka skriva en sökväg till en systemfil eller liknande.

```
sebastian@mindator:~$ rm file...
```

För att radera en mapp behöver vi däremot använda tillvalet `-r` (*recursive*), se detta som en säkerhetsåtgärd så att vi inte råkar radera en mapp.

```
sebastian@mindator:~$ rm -r directory...
```

Här är några exempel på kommandot `rm` och dess tillval:

Table 5: Några exempel på kommandot `rm`

Kommando	Resultat
`rm file1 file2`	Radera *file1* och *file2*.

Kommando	Resultat
`rm -i file1 file2`	Som kommandot ovan, men med tillvalet `-i` (*interactive*) som innebär att vi kommer behöva godkänna varje fil som ska raderas.
`rm -r dir1 dir2`	Radera mapparna *dir1* och *dir2* samt deras innehåll.

8 `mkdir`: Skapa en mapp

Kommandot `mkdir` används för att skapa nya mappar. Du skriver `mkdir` följt av en eller flera mappnamn som ska skapas. Dessa skapas i den nuvarande *working directory*.

```
sebastian@mindator:~$ mkdir directory...
```

9 Att använda kommandon med jokertecken

Eftersom de kommandon vi nu gått igenom arbetar med filnamn, kan vi använda *jokertecken* för att skriva mindre, och samtidigt komma åt fler filer och mappar. Här är några exempel:

Table 6: Några exempel på jokertecknet `*`

Kommando	Resultat
`cp *.png images`	Kopiera alla filer i vår *working directory* som slutar på ".png" till mappen *images*.
`mv ../*.txt Documents`	Flytta alla filer i föräldermappen från vår *working directory* som slutar på ".txt", till mappen *Documents* i vår *working directory*.
`rm Documents/*.pdf`	Radera alla filer som slutar på ".pdf" i mappen *Documents* i vår *working directory*.

10 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.

10.1 Klara, färdiga, gå!

Börja med att öppna en ny terminal och navigera till den plats där du placerat övningsmappen (*exercise-module-02*).

- I övningsmappen finner du filerna/mapparna `rcu`, `.karen`, `README` och `exit.c`. Vilken/vilka av dessa är av typen "ASCII text" och får vi reda på någon mer information om dessa? Anteckna ner detta som **A1**. Kom ihåg att filer som böjar med en punkt är dolda och kräver därför att vi använder ett visst tillval till `ls`.
- Vad står på sista raden i filen `pull-requests.rst`? Använd programmet `less` för att ta reda på detta. Antecka ner detta som **A2**. I samma fil finns termen "Kurragömma", sök upp den och antecka ner den korta text som kommer direkt efter som **A3** (tänk på att du behöver placera dig överst i filen innan du söker, eftersom en sökning sker från *där du är till slutet*).
- Kopiera filerna `exit.c` och `README` till mappen `events`. Anteckna ner kommandot du skrev som **A4**.
- Döp om filen `Makefile` i mappen `power` till `Fika`, utan att du befinner dig i mappen `power` (dvs. att du får inte använda kommandot `cd` för att först gå ner till mappen `power`). Antecka ner kommandot du skrev som **A5**.
- Radera alla filer som slutar på "delete_us" i mappen `futex`, använd samtidigt tillvalet `-i` så att du inte råkar radera fel filer. Antecka ner det/de kommandon du skrev som **A6**.