

BASH - Övning - Kommandon

Enkelt sagt kan man säga att när vi skrivit in en rad med kod i terminalen och klickat på ENTER, då har vi skrivit ett *kommando*. Exempel på olika kommandon:

(lista alla filer och mappar i användarmappen)

```
ubuntu@elev0:~$ ls -a ~/
```

(kopiera alla filer och mappar från sökvägen `/etc` till sökvägen `~/backup-settings`)

```
ubuntu@elev0:~$ cp -r /etc ~/backup-settings
```

Vad består ett kommando utav?

BASH är ett *skal* (på engelska *shell*) också kallat *kommandotolk*, vilket betyder att man använder **BASH** till att köra program och att hantera filer, precis som man brukar göra grafiskt i ett operativsystem. Istället för att klicka oss fram i ett grafiskt skal som har en grafisk startmeny och en filhanterare (t.ex. *Utforskaren*), kör vi programmen med hjälp av textkommandon.

BASH kan också användas till programmering, men nu till en början fokuserar vi enbart på att använda **BASH** som ett skal. Och det betyder att vi på ett eller annat sätt alltid kör olika program.

Till en början kommer den första delen av kommandot ALLTID att vara namnet på ett program. Om vi tittar på exemplen ovanför ser vi att vi använt två olika program. **ls** och **cp**.

Kör ett program utan argument och flaggor

I exemplen ovan körs två olika program, men de ges också olika *argument* och *flaggor* (på engelska *options*). Många program kan man köra utan argument och tillval, genom att endast skriva namnet på programmet och klicka på ENTER.

!Kör programmet **ls** i din terminal.

```
ubuntu@elev0:~$ ls
```

Ett annat program, som vanligtvis alltid körs utan flaggor och argument är **pwd**.

!Kör också **pwd**.

```
ubuntu@elev0:~$ pwd
```

Program och flaggor (options)

Många program kan köras i olika lägen. **ls**, som listar filer och mappar kan köras i olika lägen. Normalt visar **ls** inte gömda filer och mappar, men vill man se de gömda filerna och mapparna måste man ange en flagga (på eng. *option*).

!Kör **ls** med flaggan **-a** ("show all")

```
ubuntu@elev0:~$ ls -a
```

!Kör **ls** med flaggan **-l** ("long listing format")

```
ubuntu@elev0:~$ ls -l
```

!Man kan också kombinera flaggor. Antingen såhär:

```
ubuntu@elev0:~$ ls -a -l
```

!Eller, såhär:

```
ubuntu@elev0:~$ ls -al
```

-a och **-l** är förkortningar. Det är en typ av flagga. En annan typ skrivs med dubbelt bindestreck och har i namnet en fullare beskrivning av vad flaggan gör. **-a**, kan t.ex även skrivas som **--all**. Båda gör exakt samma sak, men de är olika varianter på flaggor. Den ena förkortad, den andra fullt beskriven.

Alla flaggor finns inte i båda varianter. Ibland finns endast varianten med förkortningen, ibland enbart varianten med den fulla beskrivningen.

!Kör **ls** med den andra typen av flagga:

```
ubuntu@elev0:~$ ls --all
```

Det går att kombinera både förkortade och fullskrivna flaggor, men de fullskrivna måste skrivas för sig (vi använder nu tre olika flaggor, **--all** **-l** och **-t**).

!Kör **ls** med tre olika flaggor, både förkortningar och fullskrivna:

```
ubuntu@elev0:~$ ls --all -lt
```

!För att se alla flaggor som man kan använda med **ls** kan du använda flaggan **--help**

```
ubuntu@elev0:~$ ls --help
```

Program och argument

En del program kan köras med ett *argument*. Med *argument* menas målet för programmet - vad programmet skall processera. T.ex en *fil*, eller en *sökväg*. Det kan också vara ett namn av något slag.

ls kan köras med ett, eller flera argument, där argumentet måste vara en sökväg.

!Kör nedanstående exempel för att se resultatet:

```
ubuntu@elev0:~$ ls /home/
```

```
ubuntu@elev0:~$ ls ~
```

```
ubuntu@elev0:~$ ls /etc/apt/
```

```
ubuntu@elev0:~$ ls /dev/input/
```

!Sen kan vi kombinera flera och på så sätt lista filer och mappar från flera källor med ett enda kommando.

```
ubuntu@elev0:~$ ls /home/ /var/
```

Kombinera allt

Och till slut, så kan vi som i det allra första exemplet, kombinera programmet med både flaggor och argument (`$USER` är en systemvariabel som innehåller användarnamnet för den användare som är inloggad).

```
ubuntu@elev0:~$ ls -al /home/$USER
```