

Modul 0

Terminalen och skalet

Sebastian Bengtegård, Johan Holmberg

22 Januari 2025

Innehållsförteckning

1	Förord	2
2	Skalet	2
2.1	Olika gränssnitt	2
3	Terminalen	3
4	Vad är nyttan med kunskap om terminalen och skalet?	3
5	Kom igång	4
5.1	Kom igång med Windows	4
6	En fungerande terminal	4

1 Förord

Detta material består av en serie moduler där respektive modul behandlar ett avgränsat kunskapsområde. Dessa moduler består av en introducerande text som beskriver området, vad vi kan göra inom detta och några exempel på hur. Därefter avslutas dessa med en praktisk övning. Därför rekommenderas det att först läsa igenom texten för att sedan påbörja övningen, där vi kan gå tillbaka till texten vid behov. Att pröva alla exempel längs vägen till övningen är valfritt och är ingenting vi förutsätter att du gör.

Detta material innehåller även en stor mängd olika termer som alla ursprungligen skrivs på engelska. Vi har däremot valt att i huvudsak skriva materialet med svenska termer och kommer istället att placera ursprungstermerna inom parenteser, eftersom dessa är viktiga att känna till, t.ex. "Skalet (*shell*)". Det finns några få undantag till denna regeln, t.ex. "en terminal (*terminal*)", där översättningen inte bidrar med något förtydligande.

I denna första modul kommer vi kort introducera terminalen, skalet och hur vi kommer igång med vår arbetsmiljö. Det finns ingen övning för denna modul, dessa börjar från och med modul 1.

2 Skalet

Skalet (*shell*) är det program där vi kan skriva kommandon som datorn sedan utför. Detta program kom i en tid då vi inte hade några grafiska användargränssnitt (*graphical user interface*) och var det enda sättet att interagera med en dator. Nuförtiden har vi både grafiska användargränssnitt och kommandoradsgränssnitt (*command line interface*) så som skalet.

I denna serie moduler om terminalen och skalet kommer vi att utgå från skalprogrammet **Bash** (**B**ourne **A**gain **S**hell, en förbättrad version av originalskalet **sh**, kort för *shell*, som skapades av Steve Bourne). Det finns idag flera alternativ till Bash, till exempel **zsh** och **fish**.

2.1 Olika gränssnitt

Grafiska användargränssnitt, som vanligen förkortas till GUI (*graphical user interface*), är det sätt som vi vanligen interagerar med en dator idag. Det vill säga genom program som webbläsaren Firefox, textredigeraren Word eller bildbehandlaren Photoshop. Ett kommandoradsgränssnitt, eller CLI (*command line interface*), är det sätt där vi endast kan skriva in text till datorn, där finns det inget koncept om olika fönster (program) som

vi kan flytta runt och så vidare. Det är genom detta gränssnitt vi i huvudsak kommer att arbeta.

3 Terminalen

Förr i tiden var det en terminal som tillät oss att fysiskt interagera med en dator, eftersom vi saknade grafiska gränssnitt. Med andra ord var terminalen hela datorn. Numera är terminalen ett program som efterliknar en gammaldags terminal genom ett grafiskt gränssnitt, så att vi kan skriva våra kommandon till ett skalprogram som Bash och samtidigt ha flera andra program igång.

Det finns en uppsjö olika terminalprogram och utbudet varierar beroende på vilket operativsystem som används. Alla har dock samma ändamål: att ge oss möjlighet att interagera med skalet.

Vilken terminal du kommer att använda dig av beror på ditt operativsystem, under avsnittet “Kom igång” får du ett lämpligt förslag.

4 Vad är nyttan med kunskap om terminalen och skalet?

Att förstå hur vi kan använda oss av en terminal, och då även skalet, ger oss i huvudsak en bättre förståelse för hur datorer fungerar. Det är inte ovanligt att ett program, bakom sitt grafiska gränssnitt, använder sig av kommandon från ett skal. Dessa kommandon kan vara allt från att ändra bredden på en bild till att konvertera en film från ett format till ett annat, och mycket mer. Denna kunskap skulle därmed vara ännu ett steg till att avmystifiera hur en dator fungerar.

Skulle vi istället diskutera de praktiska möjligheterna vi får av denna kunskap blir det svårt att bara nämna några få, eftersom datorer är en så stor del av vårt samhälle och används, mer eller mindre, överallt.

Däremot, som en utvecklare kan vi begränsa omfånget på nyttan till vad som faktiskt används ute i industrin. En stor del, kanske de flesta, datorsystem och drift av dessa kräver någon form av denna kunskap. Till exempel är det många som hört talas om så kallade “serverhallar”, det vill säga lokaler med oerhört många datorer. Dessa datorer har sällan en egen skärm och för att kunna interagera med dessa behöver vi en terminal och ett skal.

I industrin är det inte bara datorsystem som drar nytta av denna kunskap, det kan vara vardagliga saker som administrativt arbete. Vi kan med en terminal och ett skal enkelt

automatisera saker som annars är väldigt monotona, så som att sammanställa data från flera filer för att exempelvis skapa rapporter (t.ex. från olika Excel-filer till en PDF).

Att förstå nyttan kommer också från erfarenhet av att använda sig av terminalen och skalet, vilket du nu ska börja med. Så ge det lite tid så bildar du din egen uppfattning om varför kunskapen är nyttig (eller inte) för din del.

5 Kom igång

Innan vi kan gå vidare till att interagera med skalet behöver vi få igång vår arbetsmiljö: en fungerande terminal med skalprogrammet Bash. Lyckligtvis kommer både en terminal och Bash förinstallerat på operativsystemen Linux och MacOS. Dessvärre inte på Windows, i detta fall följer vi först instruktionerna under “Kom igång med Windows” innan vi går vidare till “En fungerande terminal”.

5.1 Kom igång med Windows

Det finns flera alternativ med att installera en terminal och skalet **Bash** på Windows. Ett av de enklare alternativen är programmet “Git Bash”, en terminal, som kommer med Bash installerat. Detta kan laddas ner via länken: <https://git-scm.com/download/win>.

När vi i modulerna skriver “terminalen” är det numera synonymt med programmet “Git Bash” på Windows.

[Här finner du ett exempel](#) med instruktioner för hur du installerar det. Observera att vi är endast intresserade av “Git Bash”, så när du följer instruktionerna ha detta i åtanke (vilket kan innebära att du får klicka i och ur vissa saker).

Nu kan vi återgå till avsnittet “**En fungerande terminal**”.

6 En fungerande terminal

De flesta operativsystem ger oss möjlighet att söka fram ett program att starta. Sök upp och starta programmet “terminal”, eller t.ex. “Git Bash” på Windows.

Det första vi ser i detta program är den så kallade skalprompten (*shell prompt*) där vi kan skriva in våra kommandon. Denna innehåller vanligen vårt användar- och datornamn följt av ett dollartecken, detta kan variera mellan operativsystem men ser vanligen ut på följande vis:

```
sebastian@mindator:~$
```

Pröva nu att skriva in några slumpade tecken och tryck sedan på Enter.

```
sebastian@mindator:~$ qwerty
```

Om allt fungerade ska vi ha fått ett felmeddelande om att kommandot inte finns.

```
sebastian@mindator:~$ qwerty
bash: qwerty: command not found
```

Operativsystemet MacOS kommer vanligen med skalet **zsh** förinstallerat, vilket kommer fungera lika bra som Bash för vårt material. Skulle du däremot vilja byta skriver du bara in **bash** och trycker Enter.

Perfekt! Nu har vi en fungerande arbetsmiljö. **Men** låt oss först prova ett fungerande kommando: **whoami**, som berättar för oss vem vi är inloggad som på datorn. Skriv nu in kommandot **whoami** och tryck Enter.

```
sebastian@mindator:~$ whoami
sebastian
```

Om allt fungerade ska vi ha fått ett meddelande om vem vi är inloggad som. Testa nu att trycka pil-upp och se hur kommandot **whoami** visas igen - vi har alltså en så kallad *kommandoradshistorik*. Trycker du nu pil-ned blir raden blank igen, vi går alltså fram och tillbaka i historiken genom upp- och nedpilarna.

Gå nu tillbaka i historiken till kommandot **whoami** och prova att trycka på vänster- och högerpilarna. Vi kan nämligen även navigera fram och tillbaka på kommandoraden för att kunna rätta till misstag så som felstavningar.

Nu är denna modul avklarad och du kan gå vidare till “[Modul 01: Utforska filsystemet](#)”.