

Lektionstillfälle 3

Terminalen och skalet (shell)
med Mikael Larsson

Återblick

Förra gången:

- Förutsättningar för installation
- Installera Linux
- Inställningar och alternativ
- Skillnader och likheter med Windows och macOS
- Hemkatalogen och att navigera i filsystemet

Begrepp:

Firmware, BIOS, EFI, ESP, boot loader, partition, LVM

Dagens lektion

Mål: Att kunna navigera i terminalen, använda enkla kommandon och få hjälp.

- Introduktion till Bash
- Kommandoraden
- Miljövariabler
- Att utforska och få hjälp
- Redigera filer

TLCL-boken, kapitel 1 - 3 (till s19), 5, 8

Termer och begrepp

- Terminal, konsol, emulator
- Skal / shell
- Kommandon och kommandohistorik

Terminalen

- Terminal-emulator – vad emuleras?
- Vi kommer använda konsolen som terminal
- Växla mellan olika virtuella terminaler med Alt+F1, Alt+F2, osv.



DEC VT-100 (1978)

Skalet - BASH

Bourne Again SHell

Skalet är ett program som tar tangentbordsinmatningar i form av kommandon och skickar vidare dem till operativsystemet.

Skalet är det första program som startar när en användare loggar in.

Finns många olika, vi kommer bara använda **bash**, som är standard i Linux.

Bash har ersatt det ursprungliga skalet **sh** som är skrivet av Steve Bourne. Därav ordvitsen i bash, **Bourne Again** istället för **Born Again** (född på nytt).



▶ [Colt Steel on the shell](#)

Kommandoraden

- Prompt, väntar på inmatning, slutar ofta med \$ eller #
 - \$ = vanlig användare
 - # = superanvändare (**root** eller **sudo su**)
- Historik– använd pil upp/ned
- [tab]-komplettering
 - Kommandon
 - Filargument
 - Skriptad komplettering

Kommandon kan ta optioner och argument

```
$ command [argument]  
$ command [option]... [argument]
```

Ofta kan man använda mer än en option. Ett argument kan vara ett filnamn.

Du kommer väl ihåg "ls", "pwd" och "cd" från förra lektionen?

Några kommandon

- **date** – visa eller sätt datum och tid
exempel: **date -Is; date +%s**
- **df** – visa filsystemsinformation
exempel: **df -h; df -T /boot/efi; df -TBM /dev/shm**
- **free** – visa minnesanvändande i systemet
exempel: **free -h; free -s 5**
- **exit** – avsluta skalet \

Kommandon som visar minne och storlek tar ofta option **-h** för att visa i formatet "human", alltså för människor att läsa. Optioner kan också ta värden som i fallet **free -s 5** ovan som får free att repetera sig själv var femte sekund.

Vad händer om man trycker [ctrl+d] vid kommando-prompten?

Varför?

The 50 most popular commands

whoami	rmdir	tail	unig	ps	sleep	ln
man	rm	date	diff	top	gzip	who
clear	touch	cat	find	kill	gunzip	su
pwd	open	less	grep	killall	tar	sudo
ls	mv	echo	du	jobs	nano	passwd
cd	cp	ws	df	fg	alias	chown
mkdir	head	sort	history	bg	xargs	chmod

Colt Steele går igenom dessa kolumnvis av en pedagogisk anledning.

Titta på Colt Steele's video, vid tillfälle, på youtube om du inte har gjort det redan (5 timmar):

 <https://www.youtube.com/watch?v=ZtqBQ68cfJc>

Vi kommer att stifta bekantskap med många av dessa kommandon.

Kanske några fler dessutom, **what****is**, **free**, **exit**. 😊

Kontrolltecken i terminalen

^ = tangenten [ctrl]

terminal	ahead of	word	back	cursor	forward	word	end
clear				^L			
up (prev)				^P			
move	^A	Alt-B	^B		^F	Alt-F	^E
delete	^U	^W	^H,BS	^D	^D	Alt+D	^K
down (next)				^N			
undo				^7			
hist.search				^R			

- ^R = reverse-i-search
- ^C = avbryt körande process (program)
- ^D = avsluta ström
- ^S/^Q = **Stop** / **Quit stop** terminal output, xoff/xon

ASCII-tabell: <http://www.physics.udel.edu/~watson/scen103/ascii.html>

Filsystemet

Vandra i filsystemet

Linux har ett hierarkiskt filsystem med en rot. Man kan tänka sig ett fällt träd med roten till vänster och kronan med sina grenar till höger, där löven är filer.

- Filnamn är skiftlägeskänsliga.
- Nuvarande katalog heter punkt, ".", föräldrakatalogen heter "..".
- Absoluta sökvägar börjar alltid från roten, relativa sökvägar börjar från nuvarande katalog. Jämför `cd /home` och `cd ..`.
- Filer som börjar med punkt är dolda, och kan listas med `ls -a`.
- tilde, "~", pekar på hemkatalogen. Kan bara användas först i en sökväg.
- `pwd` visar var vi är nu.

Det finns ett kommando som heter `tree` som visar hierarkin.
Den måste installeras med `sudo apt install tree` i ubuntu.

Text – vad är text?

Textfiler är filer avsedda för att direkt läsas eller redigeras av människor.

Textfiler kodas med olika teckenuppsättningar, som översätter filens bytes till tecken.

ASCII, UTF-8, iso-8859-1 (Latin-1)

- **type** – vad för slags kommando?
- **file** – vad för slags fil?
- **less, more** – visa filens innehåll
- **head / tail** – visa filstart / filslut

```
$ type cd
cd is a shell builtin
```

```
$ type quote
quote is a function
quote ()
{
    local quoted=${1//\'/\'\'\\\'\'\'\'};
```

```
/var/log# file *
apt:                directory
auth.log:           ASCII text, with very long lines
boot.log:           UTF-8 Unicode text, with CRLF, LF
                    line terminators, with escape sequences
cups:               directory
demo:               ISO-8859 text
dmesg:              UTF-8 Unicode text, with very
                    long lines
dpkg.log:           empty
faillog:            data
fontconfig.log:     CSV text
syslog:             ASCII text
syslog.1:           UTF-8 Unicode text
syslog.2.gz:        gzip compressed data, last
                    modified: Thu Feb  9 23:00:15 2023...
wtmp:               dBase III DBT, version number 0,
                    next free block index 2
```

Laboration 1

Ta hjälp av boken [TLCL](#), sidan 2 – 19.

Se instruktion för Laboration 1 i studentportalen.

Intro till docker

Olika slags kommandon

Exekverbara filer

- /usr/bin/ls – binär
- hello.sh – shell script

Inbyggda kommandon

- `exit`, `type`
- `alias`
- `ls`

Shell-funktioner

Funktioner i bash är en samling kommandon under ett namn. När namnet anropas utförs alla kommandon i följd.

Viktigt! Funktioner exekveras i det environment som de startades i till skillnad från ett shell-skript.

Använd kommandot "type" för att avgöra vilket slags kommando du har att göra med.

```
$ type type  
type is a shell builtin
```

Exekverbara filer är märkta med en x-flagga. Se utskrift från "ls -l".

```
-rwxr-xr-x 1 root root 76672 feb  5 2022 ping
```

Miljövariabler 1

Environment variables

- Namngivna variabler med strängvärden.
- Mycket likt motsvarande i Windows.
- Variablerna bor i processen, hanteras av skalet.

Sätts genom tilldelning:

```
$ hello=world
```

Visas med **echo**, **printenv** eller **declare -p**:

```
$ echo $hello  
$ printenv hello  
$ declare -p hello
```

Finns också interna shell-variabler – styr skalets funktion.

Kommandot **set** och **declare** listar både interna- och miljö-variabler, även icke-exporterade.

Mer detaljer i [TLCL kapitel 11](#)

Miljövariabler 2

Måste exporteras för att synas i barnprocesser:

```
$ hello=world  
$ export hello  
    ... eller ...  
$ export hello=planet
```

Kan tillfälligt exporteras och ändras inte i skalet:

```
$ hello=olleh printenv hello
```

Lista **alla** exporterade variabler

```
$ set  
$ env|sort  
$ declare -p
```

Lista **icke** exporterade variabler

```
$ declare -p | grep "\-\"
```

Lista exporterade variabler

```
$ printenv | sort
```


Att få hjälp

Hjälp-system

- `apropos` – Visar alla man-sidors rubrik för angivna nyckelord, behöver inte vara ett kommando
- `what is` — Visar man-sidans rubrik
- `man` – Visar **man**ualen för ett kommando”
- `help` – Visar hjälp för shell-inbyggda kommandon

Hjälp inbyggt i kommandon, "usage"

- `what is -h`
- `apropos -?` *ovanligt*
- `head --help`

Varför inte bara Googla?

Att redigera textfiler — utan GUI

Viktigt att kunna redigera text utan GUI

- Klassiska editorer, **vi** och **emacs**
- **vi**, **vim** finns *överallt* – förinstallerad, men är lite udda om man kommer från GUI. Välj **vim** över **vi** om den finns eller går att installera, den är enklare.
- Oftast finns **nano** installerad – funkar lite mer *som vanligt*

Du måste kunna en av dessa bra.

De andra måste du kunna öppna och spara i, för att inte bli strandad i en nödsituation.

Om du vill testa **emacs** – använd **jove**!

Det är en mycket liten – i stort Emacskompatibel editor med en egen lärobok, **learnjove**

Att redigera textfiler — VIM

vi Visual Editor kom som en räddare när endast ed fanns att tillgå.

vim = Vi IMproved är en förbättrad version. Använd den, installera den om den saknas.

Vim använder en "penna nere" analogi som indikeras av att det står -- INSERT -- längst ner på skärmen. Man "lyfter pennan" med [esc] eller [^C] (Ctrl+C). Inga kommandon kan ges i läge "penna nere" men vanliga markörförflyttningar fungerar med pilar och [home] [end] [page up] och [page down].

De viktigaste kommandona:

kommando	förklaring
i	Lägg ner pennan, starta läge -- INSERT --
[esc]	lämna läge -- INSERT --
[^C]	lämna läge -- INSERT --
u	undo senaste ändring
:w	spara (Write) utan att lämna vim
:wq	spara (Write) och lämna vim (Quit)
:q	lämna vim (Quit) utan att spara
:q!	lämna vim (Quit) tvingande (!) kan behövas om du redigerat i en fil du inte äger.

När pennan **inte** är nere (-- INSERT -- visas inte) kan du söka på samma sätt som i less och många navigeringskommandon fungerar lika.

Laboration 2

Se instruktion i studentportalen.

Laboration 2.5

Välj en lämplig text-editor, skapa en fil i din hemkatalog. Skriv en liten lathund för kontroll-kommandon och kommandon vi pratat om idag.

Om du blir klar – välj en annan editor och fortsätt redigera.

Om du vill testa `jove` behöver du installera den med `$ sudo apt install jove`.

Det finns även en emacs-kompatibel editor som heter jed med menyer.

Har du ännu mer tid över, spela vim-adventures:

<https://vim-adventures.com/>

Summering

Idag har vi arbetat med:

- Introduktion till Bash
- Kommandoraden
- Miljövariabler
- Att utforska och få hjälp
- Redigera filer

TLCL-boken, kapitel 1 - 3 (till s19), 5, 8

Läs kapitlen!

Frågor?

Nästa gång

Mål: Att förstå Linux filer, filstruktur och veta var saker finns.

- Linuxdistributioner
- Filsystem och filstruktur
- Filrättigheter
- Symboliska länkar
- Att söka efter filer

Stort tack!