

## DVA249/DVA267 Linux, HT2023

# - Laboration 2 -

### Förberedelser

Läs instruktionerna på Canvas och läs veckans kurslitteratur innan du börjar med laborationen. Vi rekommenderar även att du tittar på videomaterialet som tillhör laborationen. Svaren till flera av uppgifterna finns i kursmaterialet.

## 1 Filhantering i det grafiska gränssnittet

I denna laboration kommer vi titta på grunderna i Linux filsystem. Målet med uppgiften är att förstå den hierarkiska strukturen hos filsystemet, lära dig hur du navigerar, skapar/tar bort/döper om filer och kataloger och vart viktiga filer finns. Vi börjar med att titta på det via det grafiska gränssnittet.

### 1.1 Filer och kataloger

Starta en filhanterare ('*file manager*') via GUI och hitta din hemkatalog ('*home directory*'). Sökvägen till din hemkatalog är oftast `/home/<yourusername>`. Om ditt användarnamn (namnet du använda för att logga in) är `abc123` skulle din hemkatalog vara `/home/abc123`.

1. Undersök vilka kataloger som finns i din hemkatalog.
2. Använd sökfunktionen i filhanteraren och sök efter följande kataloger eller filer: `local` och `messages`. Hur kan du se ifall det är en katalog eller vanlig fil?

**OBS!** Du kan behöva starta din sökning i en annan katalog än din hemkatalog, till exempel i root-katalogen (`/`).

3. Gå tillbaka till din hemkatalog. Skapa en ny fil med namn `file1`. Skapa även en ny katalog i din hemkatalog och döp den till `testDirectory`. Kopiera filen du skapade (`file1`) till `testDirectory`. Markera sedan filen `file1` i `testDirectory`, kopiera den och klistra in den i samma katalog. Vad händer?
4. Ta bort alla filer i katalogen `testDirectory` och ta sedan bort katalogen `testDirectory`.
5. Som standard visar inte filhanteraren dolda eller backup-filer. Hitta ett sätt att (i filhanteraren) permanent visa dessa filer. Hur gör du det?

## 2 Filhantering i kommandotolken

Nu ska vi utforska hur filhantering fungerar i kommandotolken via kommandon.

### 2.1 Bläddra i filsystemet

Använd kortkommandot (med tangentbordet) för att öppna en ny kommandotolk. Kör kommandot `ls`. Du ser filerna i din hemkatalog och det bör vara samma filer som du såg i den grafiska filhanteraren.

1. Prova nu att köra `ls` med flaggan `-a`, sedan med flaggan `-l` och till sist med flaggan `-F`. Förklara vad de olika flaggorna har för funktioner.

**Tips:** Du kan hitta information om kommandon med kommandona `info` och `man`.

2. Experimentera med kommandona `cd` och `pwd` i kommandotolken och beskriv vad de används till.
3. Bläddra igenom följande underkataloger till root-katalogen (`/`). Hitta och beskriv funktionen av dessa kataloger. Vilka typer av filer är sparade i de olika katalogerna?

<code>/home</code>	<code>/usr</code>	<code>/var</code>	<code>/tmp</code>	<code>/etc</code>
--------------------	-------------------	-------------------	-------------------	-------------------

## 3 Filanvändning

1. Skapa en katalog `lab3test` i din hemkatalog. Vilket kommando använde du? Vilka bokstäver och symboler kan och kan inte användas i ett filnamn?
2. Byt nu din arbetskatalog till `lab3test` och skapa en fil vid namn `lab4test.txt`. Hur kan du skapa en tom fil (utan att använda dig av en texteditor)?
3. I katalogen `lab3test`, skapa två underkataloger `subdir1` och `subdir2`.
  - (a) Vad är den absoluta sökvägen till `subdir1`?
  - (b) Vad är den relativa sökvägen till `subdir1`?
4. Alla kataloger innehåller två specialkataloger: `.` och `..`, beskriv vad de betyder och hur de kan användas.
5. När du ska hantera flera filer samtidigt kan dessa skrivas var för sig i kommandoraden. Ett mer effektivt sätt är att använda sig av globbing.
  - (a) Beskriv hur och när asterisk (`*`) och frågetecken (`?`) används.
  - (b) Ge ett exempel med kommandot `ls` och `*`
  - (c) Ge ett exempel med kommandot `ls` och `?`

### 3.1 Filhantering

I denna uppgift ska vi lära oss att kopiera, flytta, döpa om och ta bort filer.

**OBS!** Att flytta, döpa om, ta bort och kopiera filer kan allvarligt skada ditt humör. Många timmars arbete kan gå förlorat om du råkar ta bort eller skriva över filer. Var försiktig när du gör denna del av laborationen. Flaggan `-i` är en bra livlina. Skapa gärna ett alias för `cp`, `mv` och `rm` med flaggan `-i`.

1. Använd kommandot `cp` för att kopiera `lab3test.txt` till en fil med namnet `lab3test_backup.txt`
  - (a) Vilket kommando använde du?
  - (b) Läs manualsidan för `cp` och beskriv följande flaggor: `-v`, `-r`, `-p`, `-b`, `-u` och `-i`.
2. Vilka av följande kombinationer av kommandon och argument är korrekta? Beskriv varför de inkorrekta kombinationerna inte fungerar.
  - (a) `cp file1.txt file2.txt`
  - (b) `cp file1.txt file1.txt`
  - (c) `cp file1.txt file2.txt file3.txt`
  - (d) `cp file1.txt file2.txt subdir1/`
  - (e) `cp subdir1/ file2.txt`
  - (f) `cp subdir1/ file1.txt file2.txt`
  - (g) `cp file1.txt subdir1/file4.txt`
  - (h) `cp file1.txt subdir1/`
3. Hur kan du kopiera alla filer med filändelsen `.txt` till en katalog? **Tips:** Använd globbing
4. Vilket kommando kan användas för att
  - (a) *flytta* filen `file2.txt` från den aktuella katalogen till underkatalogen `subdir2`?
  - (b) *ta bort* filen `file4.txt` från underkatalogen `subdir1`?
5. Anta att du i din hemkatalog har katalogerna `dir1` och `dir2`. I `dir1` finns två filer och en underkatalog enligt Figur 1. Vad händer om du kör kommandot `mv -i dir1/* dir2/` från hemkatalogen?

```
awk03@ubuntu:~/dir1$ ll
total 16
drwxrwxr-x  3 awk03 awk03 4096 nov 23 14:52 ./
drwxr-xr-x 20 awk03 awk03 4096 nov 23 14:49 ../
-rw-rw-r--  1 awk03 awk03  108 nov 23 14:52 aa.log
-rw-rw-r--  1 awk03 awk03    0 nov 23 14:21 bb.txt
drwxrwxr-x  2 awk03 awk03 4096 nov 23 14:49 subdir/
awk03@ubuntu:~$ mv -i dir1/* dir2/
```

Figur 1: Listar innehållet i `dir1`

### 3.2 Filsökning

Tidigare i laborationen har du använt sökfunktionen i det grafiska gränssnittet. I denna uppgift ska du söka efter filer i terminalen istället.

1. Beskriv kortfattat funktionen av följande kommandon. Hur skiljer de sig?
  - (a) `find`
  - (b) `locate`
  - (c) `whereis`
  - (d) `which`
2. Hur kan du utföra en skiftlägesokänslig ('*case insensitive*') sökning med `find` och `locate`?

### 3.3 Arkivering och komprimering

Filer kan paketeras för säkerhetskopiering eller för smidigare filöverföring. Dessa arkiv kan även komprimeras för att spara diskutrymme.

I denna uppgift tittar vi på olika arkiveringstyper och komprimeringsverktyg.

1. Titta på funktionen av följande arkiveringsverktyg: `tar`, `gzip` och `gunzip`. Beskriv varje verktyg med en mening.
2. Skapa ett arkiv med `tar`-kommandot. Du kan exempelvis skapa en ny katalog, skapa filer i katalogen och sedan skapa ett arkiv som innehåller alla filer.

**Tips:** Använd arkiveringsverktyget i en katalog där du inte har något viktigt eftersom filer kan bli ersatta ('*overwritten*') eller borttagna vid felaktig användning.

**Tips:** Enligt praxis döps tar-arkiv med filändelse `.tar`. Ett tar-arkiv kan exempelvis heta `projekt_version_2.tar`.

3. Skapa en ny katalog och packa upp ('*extract*') ditt arkiv i den katalogen med hjälp av `tar`. Vilket kommando använder du?
4. Gör nu samma övning som i de föregående två stegen men denna gång ska du komprimera tar-arkivet med `gzip`.

**Tips:** Man brukar döpa ett komprimerat tar-arkiv med filändelse baserat på vilken komprimeringsteknik som används. Ett tar-arkiv komprimerat med `gzip` ges filändelsen `.tar.gz`, till exempel `projekt_version_2.tar.gz`.

## 4 Introduktion till skript

Skalskript ('*shell scripts*') är enkla program skrivna i ett tolkat programmeringsspråk som är inbyggda i ett Linuxskal. Eftersom du kommer skriva dina skript i **BASH** så bör du döpa dina filer med filändelsen **.bash** och använda dig av **BASH** (**bash**) när du kör dina skript.<sup>1</sup> Kör *inte* dina skript med **sh** (detta är en gammal version av **BASH** och inte samma sak). I denna laboration kommer vi enbart skapa väldigt enkla skript som består av flera kommandon som körs efter varandra.

1. Öppna en texteditor och skriv in kommandon (ett per rad) enligt Kodruta 1.

```
#!/bin/bash
echo "Första kommandot!"
echo "Kommando nummer två..."
echo "Tredje kommandot är det sista."
```

Kodruta 1: kommandon.bash

2. Fundera på vad du tror att du kommer få för output.
3. Spara filen som **kommandon.bash**
4. Prova att köra skriptet genom att köra kommandot **bash kommandon.bash**
5. Vad får du för output? Är det samma som du trodde?

**Tips:** Notera att skriptet börjar med en speciell rad, den så kallade '*shebang*'-raden: **#!/bin/bash**. I denna laboration behöver du inte förstå vad den betyder, men varje skript du skapar kommer att ha den raden längst upp.

Man använder sig ofta av skript för att automatisera uppgifter. Om man har ett antal kommandon som man vill köra ofta, kan man skapa ett skript och istället för att skriva in alla kommandon varje gång räcker det med att köra skriptet. I laboration 1 använde vi oss av alias för att skapa kortkommandon. Aliasen försvinner varje gång man stänger och öppnar en ny terminal. För att slippa skriva om dem kan vi skapa ett skript som skapar aliasen åt oss.

1. Skriv ned tre kommandon för att skapa tre alias **alone**, **altwo** och **althree**. Vad aliasen gör för något får du bestämma själv.
2. Skapa en skript-fil **aliases.bash** som innehåller en shebang följt av de tre kommandon som du skrivit ned, se Kodruta 2.

```
#!/bin/bash
alias alone=...
alias altwo=...
alias althree=...
```

Kodruta 2: aliases.bash

3. Öppna en ny terminal, se till att aliasen är odefinierade genom att köra dem. Kör sedan **source aliases.bash** för att definiera dem, och kolla att de fungerar som du tänkt.

**Tips:** För att alias ska skapas i den BASH-instans som du befinner dig i måste **source** användas istället för **bash**.

Vi kommer i senare laborationer titta på var man kan placera sina kommandon för att de ska köras automatiskt varje gång man startar en ny terminal.

<sup>1</sup>Ifall du har installerat ett annat skal, se till att använda dig av **BASH** när du gör inlämningsuppgifterna.

## DVA249/DVA267 Linux, HT2023

### - Quiz 2 -

Denna uppgift genomförs **individuellt och lämnas in genom att göra quiz "Quiz 2" i Canvas** när du är klar med uppgiften. Uppgiften ska vara inlämnad innan deadline, annars kan vi inte garantera att vi hinner titta igenom era inlämningar innan examinationstillfället.

Vilka av följande kombinationer av kommandon och argument är korrekta?

- (a) `cp file1.txt file2.txt`
- (b) `cp file1.txt file1.txt`
- (c) `cp file1.txt file2.txt file3.txt`
- (d) `cp file1.txt file2.txt subdir1/`
- (e) `cp subdir1/ file2.txt`
- (f) `cp subdir1/ file1.txt file2.txt`
- (g) `cp file1.txt subdir1/file4.txt`
- (h) `cp file1.txt subdir1/`

Quizet anses vara klar när du har 1 poäng på uppgiften i Canvas.

## DVA249/DVA267 Linux, HT2023

### - Inlämningsuppgift 2 -

Denna uppgift genomförs **i grupp och lämnas in genom att ladda upp filen i Canvas** när du är klar med uppgiften. Uppgiften ska vara inlämnad innan deadline, annars kan vi inte garantera att vi hinner titta igenom era inlämningar innan examinationstillfället.

Skapa ett skript med namnet `logs.bash` som gör följande:

1. Byter arbetskatalog till hemkatalogen `/home/<username>/`. Använd `$HOME` (eller `~`) för att hänvisa till din hemkatalog.
2. Skapa en katalog med namnet `Logs` i din hemkatalog.
3. Kopiera alla filer som slutar på `.log` från katalogen `/var/log/` till `Logs`-katalogen.
4. Skapa ett komprimerat tar-arkiv som innehåller filerna i `Logs`-katalogen. Arkivet ska sparas i `Logs`-katalogen.
5. Ta bort alla filer som slutar på `.log` i `Logs`-katalogen.
6. Lista innehållet i `Logs`-katalogen.

Skriptet ska använda den **relativa sökvägen** till `/home/<username>/Logs/` och den **absoluta sökvägen** till `/var/log/`. Använd skriptet i Kodruta 3 som grund för ditt skript. Skriv klart skriptet enligt punkterna ovanför.

```
#!/bin/bash
# Kommandorad nr 1
# Kommandorad nr 2
# Kommandorad nr 3
echo '-----'
# Kommandorad nr 4
# Kommandorad nr 5
echo '-----'
# Kommandorad nr 6
```

Kodruta 3: `logs.bash`

Kör ditt skript genom att ange skriptet som argument till kommandot `bash`. Figur 2 visar ett exempel på det förväntade resultatet.

**OBS!** Antalet `log`-filer och vilka ni har rättigheter att kopiera kan skilja sig mellan olika Linux-maskiner. Därför är Figur 2 endast ett exempel.

```
cp: cannot open '/var/log/boot.log' for reading: Permission denied
-----
<Eventuell output fr n kommandorad 4 eller 5>
-----
logs.tar.gz
```

Figur 2: Exempel output

Uppgiften anses vara klar när du har 1 poäng på uppgiften i Canvas. Deadline för uppgiften är 17/11 kl 23:59.