

**Kursus LX-100 / SU-100**

# **Linux / UNIX Grundkursus**

## **Øvelser**

## Indholdsfortegnelse øvelser

Modul 3 UNIX/Linux: Filosofi og Syntaks .....	3
Modul 4 - Dokumentation og hjælp .....	8
Modul 5 - vi-editoren .....	10
Modul 6 - Filtræet .....	16
Modul 7 - Håndtering af data og filer. ....	18
Modul 8 - Håndtering af printere .....	26
Modul 9 - Processer .....	29
Modul 10 - Shellen og dens funktion .....	31
Modul 11 - Tidsafhængig jobstart .....	37
Modul 12 - Brugere .....	38
Modul 13 - Filrettigheder .....	40
Modul 14 - Kopiering og komprimering .....	42

## Modul 3 UNIX/Linux: Filosofi og Syntaks

### Øvelse 3.1. a

Sammenlign kommandoerne inden for hver "gruppe".

(\$-tegnet angiver prompten. Afslut med RETUR efter hver kommando)

```
$ ls gurues
```

```
$ ls -l gurues
```

```
$ ls dir
```

```
$ ls -l dir
```

```
$ ls -l larkA larkB
```

```
$ ls -l -Ir larkA larkB
```

Er der forskel på følgende to kommandoer:

```
$ ls -li 1. b
```

```
$ ls -li gurues
```

### Øvelse 3.1. c

Sammenlign følgende kommandoer.

```
$ ls -l lark
```

```
$ ls -le lark
```

```
$ ls -lu lark
```

## Øvelse 3.1. d

Hvad er forskellen på følgende kommandoer:

```
$ ls
```

```
$ ls -ro
```

```
$ ls -x
```

```
$ ls -R
```

```
$ ls - F
```

## Øvelse 3.2. a

**Afprøv følgende kommandoer og iagttag forskellene.**

```
$ cat dyr
$ cat gurues
$ cat dyr gurues
$ cat gurues dyr
$ cat gurues dyr > fil4
$ Is -l gurues dyr fil4
$ cat fil4
```

```
$ Is dyr gurues
$ Is gurues dyr
```

```
$ echo dyr gurues
$ echo gurues dyr
```

```
$ cat
```

**" indtast dernæst et tilfældigt ord og tryk på RETUR. Gentag dette nogle gange og tryk tilsidst *ne* (Ctrl+e).**

## Øvelse 3.2. b

**lagttag forskellene mellem:**

```
$ cat dyrl)
$ cat -v dyrl
```

### Øvelse 3.3. a

Hvilken ugedag blev du født? Brug cal-kommandoen til at vise en kalender over din fødselsmåned. Du skal give cal 2 parametre: månedsnummer og årstal- fx:

```
$ cal 2 1947  
(februar 1947)
```

Afprøv også at indtaste disse eksempler:

```
$ cal
```

```
$ cal 2
```

```
$ cal 1947
```

```
$ cal 9 1752  
(en lidt speciel måned!)
```

```
$ cal 5000
```

```
$ cal 0
```

### **Øvelse 3.4. a**

**Gentag 5 af dine tidligere kommandoer.**

**Vælg selv hvilke og i hvilken rækkefølge du vil gøre det i.**

### **Øvelse 3.4. b**

**Hvor findes date-kommandoen (dvs. filen/programmet)?**

**Hvad laver date-kommandoen?**

## Modul 4 - Dokumentation og hjælp

### Øvelse 4.1. a

Find nogle ikke gyldige flag til kommandoen `ls` -

### Øvelse 4.1. b

Hvordan påvirker flaget `-F` kommandoen `ls` ?

Hvad får man med: `ls -lt`

### Øvelse 4.1. c

Hvad bruges kommandoen `find` til?

### Øvelse 4.2. a

Find nogle korte forklaringer til følgende kommandoer:

`cat`

`find`

`ls`

`nl`

*Lenier*

### Øvelse 4.3. a

Find de kommandoer, der har følgende nøgleord i deres korte beskrivelser:

`copy`

`mail`

`search`



## Øvelse 4.4. a

**Afprøv disse tre hjælpekommandoer:**

`man`

`man -f`

`xman &`

**... til at finde information om:**

**`vi`**

**`ex`**

**`ed`**

**`emacs`**

## Modul 5 - vi-editoren

### Øvelse 5.1. a

Åbn dokumentet lark med kommandoen:

```
$ vi lark
```

#### **Flyt cursoren:**

- 12 tegn ind på linjen
- ned til 2. vers
- tilbage til starten af linjen
- til sidste linje i dokumentet
- op til næstsidste vers
- tillinjeslut
- 3 ord baglæns
- til linje 4
- 4 ord frem

Afslut vi uden at arkivere - tast følgende:

"ESC"- tasten og derefter

```
:q!
```

## Øvelse 5.1. b

Åbn **en anden fil med vi.**

Flyt "på egen hånd" cursoren rundt i dokumentet.

**Afslut vi uden at arkivere - tast følgende:**

"Esc"- tasten og derefter

:q!

## Øvelse 5.1. c

Åbn filen txt

Hvad sker, hvis du taster:

34w

3j

3 (efterfulgt af retur-tasten)

5k

100k (hvad sker?)

6k

**Afslut vi uden at arkivere - tast følgende:**

"Esc"-tasten og derefter

:q!

### Øvelse 5.2. a

Åbn **påny dokumentet lark** med kommandoen:

```
$ vi lark
```

Rediqr, så der kun er en tom linje mellem versene.

**Afslut vi uden at arkivere - tast følgende:**

"Esc"-tasten

:q!

### Øvelse 5.2. b

Åbn **påny dokumentet lark**

Gå til linje 29 (med de 3 punktummer i starten af linjen).

Fjern de 3 punktummer.

### Øvelse 5.3. a

Åbn **filen saturday**

Kopier linje 1 og indsæt den igen efter linje 3

Fortryd indsætningen

### Øvelse 5.3. b

I hvert vers i filen saturday er første verslinje blevet delt i to dele.

Nu skal du forene de to dele.

### Øvelse 5.3. c

I sangen "Det var en lørdag aften" starter hvert vers med gentagelse af de to første linjer.

Sørg for at de pågældende linjer også bliver skrevet to gange i filen `saturday`.

### Øvelse 5.4. a

Åbn **filen `saturday`**

på nederste linje mangler lidt oplysninger. Ret teksten, så den ser således ud:

`melodi: folkemelodi`

`tekst: S. Grundtvig`

**I filen `lark`** skal du tilføje oplysningslinjer om henholdsvis melodi og tekst:

`Carl Nielsen (1924)`

`Harald Bergstedt (1921)`

### Øvelse 5.5. a

**I første linje af filen `saturday`** er der anvendt et `ettal` i stedet for et lille `l`.

Ret det med `r`-kommandoen.

### Øvelse 5.5. b

Ret tilsvarende udråbstegnet til punktum

### Øvelse 5.5. c

Ret alle andre steder `kolon` til mellemrum.

### Øvelse 5.6. a

Åbn **filen lark**

Søg efter tekststrengen "unge".

Fortsæt søgningen og tæl op hvor mange forekomster der er.

### Øvelse 5.7. a

I **filen lark** er der skrevet "svale" i stedet for "lærke".

Nu skal du fra bundlinjen med substitute-kommandoen rette alle steder.

### Øvelse 5.7. b

Åbn **filen kina**

Ret alle forekomster af bogstavet e til bogstavet i

### Øvelse 5.7. c

Åbn **filen kina**

Ret alle forekomster af vokaler (aeiouyæøå) til bogstavet i

### Øvelse 5.8. a

Åbn **et nyt tomt dokument.**

Indtast følgende 4 linjer:

```
Det er ikke så svært  
at bruge VI-editoren.  
Man kan komme langt  
med det vi har lært.
```

Arkiver dokumentet under navnet minfil

### Øvelse 5.8. b

Tilføj forneden denne linje:

```
VI er standard i såvel Linux som UNIX
```

Arkiver seneste ændringer.

### Øvelse 5.8. c

Tilføj forneden denne linje:

```
Emacs er bedre.
```

Afslut uden af arkivere.

Cat filen: minfil

### Øvelse 5.8. d

Åbn filen påny i vi og tilføj forneden denne linje:

```
Kender vi VI, kan vi  
altid redigere tekstfiler.
```

Arkiver og afslut med zz

### Øvelse 5.9. a

Åbn **filen cats**

Indsæt filen dogs før linje 1 og efter linje 6.

Arkiver som: **hundogkat**

## Modul 6 - Filtræet

### Øvelse 6.1. a

Undersøg hvor du står i filtræet efter login, og se hvilke filer der ligger her.

### Øvelse 6.1. b

Gå til rodkataloget og list alle de filer og kataloger der ligger her (ikke underkatalogernes indhold).

Gå herefter til et eller flere af følgende kataloger og list indholdet:

/usr

/bin

/etc

/dev

/home

### Øvelse 6.1. c

Find ud af hvor mange bytes filen /etc/passwd indeholder .

... og hvor mange bytes indeholder filen bibel i dit hjemmekatalog?

### Øvelse 6.1. d

Gå til kataloget /tmp

- Check at det er her du står.

Gå herefter til kataloget tmp i dit hjemmekatalog.

- Check at det er her du står.

Skift mellem katalogerne /tmp og tmp i dit hjemmekatalog.



### Øvelse 6.1. e

Gå til hjemmekataloget

Opret et katalog med navnet sub

Gå til kataloget dir i dit hjemmekatalog.

Skift mellem katalogerne sub og dir Brug både relativ og absolut sti-angivelse.

### Øvelse 6.1. f

Eksekver filen g i dit hjemmekatalog.

Gå til dit sub-katalog og eksekver derefter filen g en gang til.

Gå til /tmp-kataloget og eksekver filen g endnu en gang.

### Øvelse 6.1. g

Opret i kataloget sub et par tomme filer (giv dem fx. navnene a og b).

Slet på en gang filerne a og b samt kataloget sub

## Modul 7 - Håndtering af data og filer

### Øvelse 7.1. a

Brug kommandoen `cat` til at:

- Se indholdet af `fill`
- Se indholdet af `fill`, `fil2` og `fil3` på en gang
- Se indholdet af filen `lark`

### Øvelse 7.1. b

Brug `more` eller `less` til at se indholdet af filen `bibel`

### Øvelse 7.1. c

Vis de første 4 linjer af `names`

Vis de sidste 4 linjer af `names`

Vis de sidste linjer af `names` fra og med linje nummer 3

### Øvelse 7.1. d

Vis filen `lark` med `linjenumre`.

Vis filen `names` med `linjenumre`.

Vis filen `linefeed` med `linjenumre`.

### Øvelse 7.2. a

Opret et underkatalog med navnet links

Opret i links-kataloget et ekstra link med navnet lark.In til filen lark i hjemmekataloget.

Sammenlign i-noderne og størrelsen af de to filer.

### Øvelse 7.2. b

Opret i links-kataloget et symbolsk link med navnet lark.sy til filen lark

Sammenlign i-noderne og størrelsen af de to filer i links

### Øvelse 7.2. c

Kopier filen lark i hjemmekataloget tillinks-kataloget.

Sammenlign i-noderne og størrelsen af de tre filer i links.

Vis de første 4 linjer i lark.In

Vis de første 4 linjer i lark.sy

Omdøb filen lark i hjemmekataloget til navnet larkO.

Vis de første 4 linjer i lark.In

Vis de første 4 linjer i lark.sy

Find en forklaring på hvad der er sket.

Slet filen larkO i hjemmekataloget

Vis de første 4 linjer i lark.In

Vis de første 4 linjer i lark.sy

Find en forklaring på hvad der er sket.

Kopier filen saturday i hjemmekataloget og kald kopien lark

Vis de første 4 linjer i lark.In

Vis de første 4 linjer i lark.sy

**Find en forklaring på hvad der er sket.**

### Øvelse 7.3. a

Udskriv fra filen:

bibel

linjer, som indeholder tekststrengen:

æbletræ

### Øvelse 7.3. b

Udskriv fra filen:

bibel

antallet af de linjer, som ikke indeholder tekststrengen:

æbletræ

### Øvelse 7.3. c

Udskriv fra filerne:

lark1 lark2 .. osv.

de linjer (med linjenummer), som indeholder tekststrengen:

lærke

### Øvelse 7.3. d

Udskriv filnavne på filer som indeholder tekststrengen:

lærke

### Øvelse 7.3. e

Udskriv for hver fil i hjemmekataloget antallet af linjer som indeholder tekststrengen:

lærke

### Øvelse 7.3. f

Udskriv de linjer i fill som indeholder #-tegn.

#### Øvelse 7.4. a

Find alle filer med navnet txt i /home og underliggende kataloger.

#### Øvelse 7.4. b

Find alle filer der ikke er ejet af din brugerkonto i dit hjemmekatalog.

#### Øvelse 7.4. c

Find alle de filer i eller under dit hjemmekatalog og som har ".e" i slutningen af filnavnet.

#### Øvelse 7.4. d

Vis indholdet af alle filer, hvis navn starter med fil

### Øvelse 7.5. a

Sorter linjerne i filen dyrl i faldende alfabetisk orden

### Øvelse 7.5. b

Sorter linjerne i filen dyrl i numerisk orden efter 2. felt

### Øvelse 7.5. c

Sorter linjerne i filen dyrl i numerisk orden efter 3. felt

### Øvelse 7.5. d

List filerne i hjemmekataloget og sorter linjerne efter filstørrelse.

### Øvelse 7.5. e

Sorter linjerne i filen dyrl i numerisk orden efter 3. felt.

De linjer, hvor det 3. felt har samme værdi, skal indbyrdes sorteres numerisk efter 2. felt.

### Øvelse 7.6. a

Udskriv linjerne i filen dyr, således at alle dyr forekommer, men kun en gang hver.

### Øvelse 7.6. b

Udskriv KUN de linjer i filen dyr hvor dyret kun optræder en gang i filen (truly unikke dyr)

### Øvelse 7.6. c

Udskriv linjerne i filen dyr med antallet af forekomster.

## Øvelse 7.6. d

lagttag indholdet af filen oftest

Denne fil er et shellscript.

Kan du gennemskue, hvad det laver?

Hvor kommer inddata fra?

Eksekver filen med kommandoen:

`.loftest < dyr`

Eksekver filen med kommandoen:

`.loftest`

og indtast derefter:

kat

kat

hund

egern

hund

kat

***afs/ut med <ctrl> d***





### Øvelse 7.7. a

Sammenlign filerne lark1 og lark2 ved hjælp af kommandoerne

cmp

diff

sdiff

md5sum

### Øvelse 7.7. b

Undersøg om filerne satlog sat2 er ens.

### Øvelse 7.7. c

Hvilke af filerne, som matcher giraf\*, er nøjagtig ens?

### Øvelse 7.8. a

Undersøg filtypen på

/etc/passwd

/usr/bin/passwd

-/main.c

### Øvelse 7.8. b

Undersøg indholdet af tekstfilen kap7.txt

### Øvelse 7.8. c

Undersøg indholdet af Word-dokumentet kap7.doc

### Øvelse 7.9. a

Hvor mange linjer, ord og tegn er der i filen bibel

### Øvelse 7.9. b

Hvor mange linjer, ord og tegn er der i filerne :

**kapl.doc**

**kapl.txt**

### Øvelse 7.9. c

Find ud af hvor mange forskellige dyr, der er repræsenteret i filen dyr

## Modul 8 - Håndtering af printere

### Øvelse 8.1. a

Vis status på alle tilgængelige printere.

### Øvelse 8.1. b

Find ud af om der er jobs i en af printkøerne.

### Øvelse 8.1. c

Tilføj dit eget navn til filen fill i dit eget tmp-katalog.

Sæt filen i jobkøen til udskrivning.

### Øvelse 8.1. d

Få systemadministratoren (instruktøren) til at stoppe for udskrivning fra jobkøen.

Sæt nogle flere filer i køen til udskrivning.

Kig på jobkøen i dens nye udseende.

### Øvelse 8.1. e

Fjern et eller flere jobs fra jobkøen

### Øvelse 8.1. f

Forsøg at fjerne nogen af de andre brugeres jobs fra køen.

### Øvelse 8.1. g

Sæt flere jobs i køen til udskrivning, men nu skal output fra en anden kommando (fx. who) læses af printkommandoen (via et pipe-tegn).

Kig på jobkøen i dens nye udseende.

### Øvelse 8.1. h

Hvorfor kan man ikke fjerne alle sine jobs med:

`lprm brugera2*`

eller:

`cancel brugera2*`

### Øvelse 8.1. i

Forklar hvad der sker med følgende kommandoer:

`lp -d beck lark*`

`lp -d beck brugera2*`

`lpr -p beck lark*`

`lpr -p beck brugera2*`

### Øvelse 8.1. j

Vis status på alle tilgængelige printere.

### Øvelse 8.1. k

Find ud af om der er jobs i en af printkøerne.

### Øvelse 8.1. l

Tilføj dit eget navn til filen fill i dit eget tmp-katalog.

Sæt filen i jobkøen til udskrivning.

### Øvelse 8.1. m

Få systemadministratoren (instruktøren) til at stoppe for udskrivning fra jobkøen.

Sæt nogle flere filer i køen til udskrivning.

Kig på jobkøen i dens nye udseende.

### Øvelse 8.1. n

Fjern et eller flere jobs fra jobkøen

### Øvelse 8.1. o

Forsøg at fjerne nogen af de andre brugeres jobs fra køen.

### Øvelse 8.1. p

Sæt flere jobs i køen til udskrivning, men nu skal output fra en anden kommando (fx. who) læses af printkommandoen (via et pipe-tegn).

Kig på jobkøen i dens nye udseende.

### Øvelse 8.1. q

Hvorfor kan man ikke fjerne alle sine jobs med:

`lprm bruger2*`

eller:

`cancel bruger2*`

### Øvelse 8.1. r

Forklar hvad der sker med følgende kommandoer:

`lp -d beck lark*`

`lp -d beck bruger2*`

`lpr -p beck lark*`

`lpr -p beck bruger2*`

## Modul 9 - Processer

### Øvelse 9.1. a

Undersøg hvilke processer der kører på din egen terminal.

### Øvelse 9.1. b

Laven liste over alle processer der kører på din maskine.

### Øvelse 9.1. c

Få kommandoen `wc` til at tælle hvor mange processer der kører på din maskine.

### Øvelse 9.1. d

Log ind på instruktør-maskinen før du går i gang med de følgende øvelser.

(For at logge ind, skal du bruge kommandoen `ssh`, som - når du allerførste gang anvender `ssh` mod instruktørmaskinen - kræver en bekræftelse på, at maskinerne udveksler nøgler for at kryptere forbindelsen.)

**Husk: Log ud fra instruktørmaskinen efter afsluttet øvelse.**

```
$ ssh dit_brugernavn@instructor
```

*The authenticity of host 'instructor (192.168.1.100)' can't be established.*

*RSA key fingerprint is c8:6S:3?:S6:a3:1e:61:c6:56:52:25:d1:03:c5:d4:5e.*

*Are you sure you want to continue connecting (yes/no)? yes*

*Warning: Permanently added 'instructar' (RSA) to the list of known hosts.*

brugerl@instructor's password: password

```
$
```

### Øvelse 9.1. e

Undersøg hvor mange processer der kører på instruktørmaskinen

### Øvelse 9.1. f

Laven liste over de processer, der vedrører din shell.

### Øvelse 9.1. g

Laven liste over netop de processer, der er knyttet til din sidemand's brugerkonto.

### Øvelse 9.1. h

Laven liste over netop de processer, der er knyttet til root.

### Øvelse 9.1. i

Hvad skal der til for at liste de processer, der er bash-shells?

### Øvelse 9.1. j

Hvordan får man listet processerne i sorteret orden efter dels PPID dels PID?

### Øvelse 9.1. k

Hvordan kan man med en kommando få vist procesoplysninger om flere udvalgte PID-numre eller flere forskellige bruger-ID'er?

(Brug man-siden for ps).

**Husk: Log ud fra instruktørmaskinen efter afsluttet øvelse!**

## Modul 10 - Shellen og dens funktion

### Øvelse 10.1. a - Wildcards

Find de wildcards til at liste de filer der opfylder følgende betingelser:

- filnavn ender på .c
- filnavn på 4 tegn
- filnavn der indeholder A
- filnavn starter med fil

### Øvelse 10.1. b

Hvad sker her:

```
ls *a a*
```

og

```
echo *a a*
```

### Øvelse 10.1. c

Hvad sker her:

```
echo "*a a"
```

og

```
echo ,*a a= '
```



## Øvelse 10.2. a - Redirigering

Hvad er forskellen på:

```
Is > lsfil | sort -r
```

og

```
Is | sort -r > lsfil
```

## Øvelse 10.2. b

Sorter linierne i filen SuperUsers efter efternavn og få de sorterede linier redirigeret til en fil, du kalder sort\_navne

Tilføj til sort\_navne ved hjælp af kommandoen echo en linie med dit eget fornavn og efternavn samt dato

Sorter nu linierne sort\_navne i omvendt orden og skriv resultatet tilbage til filen sort\_navne

## Øvelse 10.2. c

Brug find til at lede i /home efter filer med navnet txt. Al output skal skrives til en fil med navnet find. log

## Øvelse 10.2. d

Hvad indeholder filen find.log efter følgende kommando:

```
find .. -name '*.c' >find.log 2>&1
```

## Øvelse 10.2. e

Hvad indeholder filen find.log efter følgende kommando:

```
find .. -name '*.c' 2>&1 >find.log
```

### Øvelse 10.3. a - Pipe

Vis samtlige processer på din maskine med en skærmside ad gangen.

### Øvelse 10.3. b

Vis processer knyttet til root med en skærmside ad gangen.

### Øvelse 10.3. c

Linierne 17430-17437 fra filen bibel skal udskrives på skærmen med linienumre forrest.

Gør det ved hjælp af flere kommandoer i en pipeline

*Tips:*

`cat -n` sætter linjenumre på

`head` viser begyndelsen af en datastrøm

`tail` viser slutningen af en datastrøm

### Øvelse 10.3. d

Kør scriptet oftest og lad inddata komme fra filen bibel.word og skriv uddata til en fil.

Lav en lang listning af filerne i `/usr/bin` og sorter linierne i omvendt orden efter filstørrelsen. Vis output med en side ad gangen.

### Øvelse 10.3. e

Hvad sker her:

```
find .. - name , * . c | l e s s
```

### Øvelse 10.3. f

Åbn to vinduer og placer dem, så du kan se dem begge.

Se navnet på det ene terminalvindue ved at køre kommandoen `tty`.

Gå til det andet vindue, kør kommandoen `date` og rediriger output til det første vindue.

### Øvelse 10.4. a - Variabler

Udskriv indholdet af følgende variabler:

HOME

PATH

TERM

PS1

LOGNAME

PWD

### Øvelse 10.4. b

Kan du ændre din prompt fra

\$ █

til:

Yes, Sir?> █

... eller til maskinens navn

(variablen HOST)

fx.:

```
embla.kl.suloeal.dk: █
```

### Øvelse 10.5. a - Plinger

Afprøv og forklar forskellen mellem

```
sted:::::Karlebo Kro
```

```
s ted::::: "Karlebo Kro"
```

```
sted:::::'Karlebo Kro'
```

### Øvelse 10.5. b

Afprøv og forklar forskellen mellem

```
ls -l sat*
```

```
ls -l IIsat*"
```

```
ls -l 'sat*'
```

### Øvelse 10.5. c

Afprøv og forklar forskellen mellem

```
eeho █ dag er det: $(date)
```

```
eeho "I dag er de t: $(da te) "
```

```
echo 'I dag er det: $(date) ,
```

### Øvelse 10.5. d

Hvad sker her:

```
eeho Der er \ og "\,, og '\,
```

### Øvelse 10.5. e

Kan du ændre din prompt, så den viser aktuelle katalog (variablen PWD) og bliver ændret hver gang man bruger ed-kommandoen.

```
/home/joe: ed tmp
```

```
/home/joe/tmp: ed /tmp
```

```
/tmp: ed /ete
```

/etc:

## Øvelse 10.6. a - Shellsript

**Skriv et script, der laver en listning af de filer, der har 3, 4 eller 5 tegn i filnavnet.**

**Udskriv forklarende overskrifter.**

**Slut af med udskrivning af dato og navn.**

### Eksempel på output:

#### Filnavne på 3 tegn:

```
-rwxr-xr-x      1 kresten users          50213 Oct 5 2008 txt
```

#### Filnavne på 4 tegn:

```
-r--r--r--      1 kresten users          469 Jun 15 14:39 lark
```

#### Filnavne på 5 tegn:

```
-rw-r--r--      1. kresten users        3877682 Jun 7 15:16 bibel
```

## Modul 11 - Tidsafhængig jobstart

### Øvelse 11.1. a

Udfør om 2 minutter en lang listning af dit nuværende arbejdskatalog.  
(Output mail/es til dit kursus-brugernavn)

Når mailen er modtaget, kan du læse den med kommandoen: mail

### Øvelse 11.1. b

Opret om 5 minutter en ny fil ved navn sangbog, der skal indeholde filerne satl og larkl

### Øvelse 11.1. c

Opret om 5 minutter en ny fil ved navn versl  
Filen må kun indeholde de første 4 linjer af filerne san og lark1

### Øvelse 11.1. d

Afsend med mail kl 15:45 meddelelsen Klokken er 15:45 og vi går snart hjem

### Øvelse 11.1. e

Udfør denne kommando om 10 minutter:  
ep lark lark.ny

### Øvelse 11.2. a

Se hvilke jobs du har liggende til udførelse.

### Øvelse 11.2. b

Undersøg hvornår det er teatime.  
Prøv kommandoen:  
eeho Is lat teatime

## Modul 12 - Brugere

### Øvelse 12.1. a

Lavet eksekverbart script, der laver en udskrivning af dato, klokkeslet, m.v.

### Øvelse 12.1. b

Udvid scriptet med udskrivning af en liste over alle de brugere der er logget ind - sorteret efter brugernavn .

Scriptet skal afslutte med udskrivning af antallet af brugere der er logget ind (der skal ikke tages hensyn til at nogle brugere kan være logget ind flere gange).

### Øvelse 12.2. a

Undersøg alle oplysninger in dit user-ID.

### Øvelse 12.3. a

Ændre dit bruger-id, så du udgiver dig for en anden af brugerne på holdet.

Udfør whoami-kommandoen.

Log tilbage med `^d` (Ctrl+d).

### Øvelse 12.3. b

Log ind på instruktørmaskinen og se på output fra who kommandoen.

### Øvelse 12.3. c

Skift user-id med:

`su - bruger9`

(password: password)

Check dit nuværende user ID med id kommandoen  
logname kommandoen  
samt  
who kommandoen  
whoami kommandoen  
og bemærk forskellene.

### Øvelse 12.4. a

Udskriv de linier fra passwd-filen der starter med bruger.



## Modul 13 - Filrettigheder

### Øvelse 13.1. a

Opret en ny fil med navnet: **fil.ret**

Se hvilke rettigheder filen har.

Opret et symbolsk link til **fil.ret** med navnet: **symbol.ret**

Se hvilke rettigheder det symbolske link har.

Opret et nyt katalog med navnet: **dir.ret**

Se hvilke rettigheder kataloget har.

### Øvelse 13.1. b

Opret et almindeligt link til **fil.ret** med navnet: **link.ret**

Se hvilke rettigheder det almindelige link har.

Du skal nu tilføje eksekveringsret til filen **fil.ret**:

```
chmod +x fil.ret
```

Se hvilke rettigheder er på filerne **fil.ret** og **link.ret**

### Øvelse 13.2. a

Opret en ny fil med navnet: **fil.ret.l**

Du skal ændre rettighederne på filen med *symbolske* operatorer.

Hver gang du ændrer noget, skal du kontrollere at det er sket:

- Tilføj eksekveringsret til brugerkategorien **user** (dig selv).
- Fratag group læseret.
- Tilføj skriveret for kategorien **other**.
- Fjern alle rettigheder for alle brugerkategorier.
- Tildel gruppen læse- og eksekveringsret.
- Tildel alle brugerkategorier alle rettigheder.

## Øvelse 13.2. b

Opret en ny fil med navnet: **fil.ret.2**

Du skal ændre rettighederne på filen med *oktale* operatorer.

Hver gang du ændrer noget, skal du kontrollere at det er sket:

- Tilføj eksekveringsret til brugerkategorien owner (dig selv).
- Fratag group læseret.
- Tilføj skriveret for kategorien other.
- Fjern alle rettigheder for alle brugerkategorier.
- Tildel gruppen læse- og eksekveringsret.
- Tildel alle brugerkategorier alle rettigheder.

## Øvelse 13.3. a

Sæt umask til 000

Opret en ny fil med navnet:

**fil.ret.3**

Se dens rettigheder. Dette er de "nøgne" ublokerede standard-rettigheder.

Du skal nu finde de værdier umask skal have for at nyoprettede almindelige filer får følgende rettigheder:

rw-

rw- rw-

rw- rw- -w-

## Øvelse 13.3. b

Hvornår skal man vælge umask 033 fremfor umask 022 ?

## Øvelse 13.3. c

Hvad skal du huske m.h.t. din umask-værdi før du går videre med Øvelserne?

når du skal  
oprette filer

umask

## Modul 14 - Kopiering og komprimering

### Øvelse 14.1. a

Gå til dit hjemmekatalog.

Opret et nyt katalog: **tardir**

Du skal vha. tar-kommandoen lave en **backup-fil af** *fi/1*, *fi/2*, *fi/3* og *fi/4* efter følgende regler:

Filen skal hedde **fil1-4.tar**

Den skal placeres i tardir

Filerne skal angives uden stiangivelse

List med **ls -l** -kommandoen filen *fil1-4.tar*

List med tar-kommandoen "backup-indholdet" af *fil1-4.tar*

Skift katalog til tardir

Indlæs (extract) alle filerne i *fil1-4.tar*, så de bliver placeret i tardir

Kontroller, at det lykkedes.

Gå til underkataloget *dir* (i hjemmekataloget) og list filerne der er her og se deres indhold.

Indlæs (extract) her *fil1* og *fil2* samt *fil3* fra backuppen *fil1-4.tar*.

Kontroller, at det lykkedes, dels ved at liste filerne, dels ved at kigge på indholdet.

### Øvelse 14.2. a

Lav denne øvelse to gange. Første gang med **bzip2**-kommandoen og anden gang med **gzip**-kommandoen.

Gå til dit hjemmekatalog.

Find størrelsen på filen *txt*

Komprimer txt

Hvor meget fylder filen (txt.bz2 henholdsvis txt.gz) nu?

Noter størrelsen på den komprimerede fil.

Prøv at se indholdet af filen txt.bz2 henholdsvis txt.gz

Komprimer txt.bz2 henholdsvis txt.gz

Hvad skete der her?

Dekomprimer den komprimerede fil.

Kontroller at det er sket.

Sammenlign de noterede størrelser af de to komprimeringer.

### Øvelse 14.2. b

Find størrelsen på backupfilerne fra de foregående Øvelser.

Komprimer filerne og undersøg graden af komprimering.

