

Kursus LX-100 / SU-100

Linux / UNIX Grundkursus

Løsningsforslag

Indholdsfortegnelse

Modul 3 - Løsningsforslag - Filosofi og Syntaks.....	2
Modul 4 - Løsningsforslag – Dokumentation og hjælp.....	4
Modul 5 - Løsningsforslag - VI editoren.....	6
Modul 6 - Løsningsforslag - Filtræet.....	11
Modul 7 – Løsningsforslag – Data og filer.....	14
Modul 8 - Løsningsforslag - Håndtering af printere.....	18
Modul 9 – Løsningsforslag - Processer.....	19
Modul 10 – Løsningsforslag – Shellen og dens funktion.....	21
Modul 11 – Løsningsforslag - Tidsafhængig jobstart.....	23
Modul 12 – Løsningsforslag - Brugere.....	24
Modul 13 – Løsningsforslag - Filrettigheder.....	25
Modul 14 - Løsningsforslag – Kopiering og komprimering.....	27
Modul 14 - Løsningsforslag – lokalnet m.m.....	28

Modul 3 - Løsningsforslag - Filosofi og Syntaks

Øvelse 3.1. c

(Sammenlign følgende kommandoer.)

```
$ ls -l lark
    = normal listevisning med oprettelsetidspunkt for filen
$ ls -lc lark
    = Tidspunkt for sidste ændring af fil status information
$ ls -lu lark
    = viser det seneste filen har været tilgået
```

Øvelse 3. 1. d

(Hvad er forskellen på følgende kommandoer)

```
$ ls          (normal listevisning)
$ ls -m       (kommasepareret listevisning)
$ ls -x       (listen vises i rækker i stedet for kolonner)
$ ls -R       (listes rekursivt - man lister altså også indholdet af
underkataloger)
$ ls -F       (Tilføjer tegn alt efter filtype * / => @ |)
```

Øvelse 3.2. a

```
$ cat dyr          (viser indholdet af filen dyr)
$ cat gurues       (viser indholdet af filen gurues)
$ cat dyr gurues   (viser indholdet af filen dyr og gurues)
$ cat gurues dyr   (viser indholdet af filen gurues og dyr)
$ cat gurues dyr > fil4 (læser indholdet af gurues og dyr ind i fil4)
$ ls -l gurues dyr fil4 (lister de 3 filer)
$ cat fil4         (viser indholdet af filen fil4)

$ ls dyr gurues
$ ls gurues dyr    (Læg mærke rækkefølgen ikke ændres)

$ echo dyr gurues
$ echo gurues dyr  (echo udskriver den efterfølgende tekst,
                    det er her et tilfælde det er filnavne)
```

```
$ cat
```

(Udfører **cat** kommandoen med det input du taster efter hver gang du trykker på RETUR – afslut med CTRL + C)

Øvelse 3.2. b

(Iagttag forskellene mellem:)

```
$ cat dyr1
$ cat -v dyr1      (viser "ikke standard" tegn med ^ og M- notation)
$ cat -vt dyr1     (som ovenfor blot med notationer fo linieskift og
                   tabluring )
```

Øvelse 3.4. b

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

Hvad laver date-kommandoen?

```
$ whereis date
$ whatis date
```

Modul 4 - Løsningsforslag - Dokumentation og hjælp

Øvelse 4.1. a

Find nogle ikke gyldige flag til kommandoen ls **(brug man kommandoen)**

```
$ ls -e
$ ls -j
$ ls -y
```

Øvelse 4.1. b

Hvordan påvirker flaget -F kommandoen ls ? **(brug man kommandoen)**

```
Tilføjer tegn alt efter filtype * / => @ |
```

Hvad får man med: ls -lt **(brug man kommandoen)**

```
Udskriver lang liste og sorterer efter seneste ændringsdato
(nyeste først)
```

Øvelse 4.1. c

Hvad bruges kommandoen find til? **(brug man kommandoen)**

```
Grundlæggende finde kommandoer for at finde filer med navne
(med flaget -name)
```

```
eller efter ejer (med flaget -user)
```

```
eller efter type (med flaget -type) o.s.v.
```

```
Eks. På andre flag: -mtime -size
```

Øvelse 4.2. a

Find nogle korte forklaringer til følgende kommandoer:

```
cat find ls nl
```

```
$ man -f cat
```

```
eller
```

```
$ whatis cat
```

Øvelse 4.3. a

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

```
copy mail search
```

eks.

```
$ apropos copy
```

eller

```
$ man -k copy
```

Øvelse 4. 3. a

Afprøv disse tre hjælpekommandoer:

```
man
```

```
man -f
```

```
xman &
```

Afprøv det og læg bl.a. mærke vinduet der dukker op med xman kommandoen ...

Modul 5 - Løsningsforslag - VI editoren

Se evt <http://www.lagmonster.org/docs/vi.html> (Vi Cheat Sheet)

Øvelse 5.1. a

Åbn dokumentet lark med kommandoen:
vi lark

Flyt cursoren:

12 tegn ind på linjen: **12l**
ned til 2. vers: **2}** (**2 ALTGR+0**)
tilbage til starten af linjen: **0**
til sidste linje i dokumentet: **G**
op til næstsidste vers: **2{** (**2 ALTGR+7**)
til linjeslut: **\$** (**ALTGR4**)
3 ord baglæns: **3b**
til linje 4: **4G**
4 ord frem: **4w**

Øvelse 5.1. c

Åbn filen txt

Hvad sker, hvis du taster:

34w	(flytter cursor 34 ord fremad)
3j	(flytter cursor 3 linjer ned)
3 (efterfulgt af retur-tasten)	(flytter cursor 3 linjer ned)
5k	(flytter cursor 5 linjer op)
100k	(flytter cursor 100 linjer op/ el.til top)
6k	(flytter cursor 6 linjer op)

Øvelse 5.2. a

Åbn påny dokumentet lark med kommandoen:

\$ vi lark

Redigér, så der kun er én tom linje mellem versene. (**2dd**)

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). **(29G)**

Fjern de 3 punktummer. **(3x)**

Øvelse 5.3. a

Åbn filen saturday

Kopier linje 1 og indsæt den igen efter linje 3

yy 3 p

u

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

G dd A

melodi: folkemelodi

tekst: S. Grundtvig

ESC

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

Carl Nielsen (1924)

Harald Bergstedt (1921)

G i

Carl Nielsen (1924)

Harald Bergstedt (1921)

ESC

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:

```
3wr1
```

Øvelse 5.5. b

Ret tilsvarende udråbstegnet til punktum

```
$r.
```

Øvelse 5.5. c

Ret alle andre steder kolon til mellemrum.

```
:%s/:/ /g
```

Øvelse 5.6. a

Åbn filen lark

Søg efter tekststrengen "unge".

Fortsæt søgningen og tæl op hvor mange forekomster der er.

```
:/unge
```

```
RETURN
```

```
n
```

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.

```
:%s/svale/lærke/g
```

Øvelse 5.7. b

Åbn filen kina

Ret alle forekomster af bogstavet e til bogstavet i

```
:%s/e/i/g
```

Øvelse 5.7. c

Åbn filen kina

Ret alle forekomster af vokaler (aeiouyæøå) til bogstavet i

```
:%s/[aeiouyæøå]/i/g
```

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.

Arkivér dokumentet under navnet minfil

:w minfil

Øvelse 5.8. b

Tilføj forneden denne linje:

VI er standard i såvel Linux som UNIX

Afslut uden af arkivere.

:q!

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.

Arkivér og afslut med **zz**

Øvelse 5.9. a

Åbn filen cats

vi cats

Indsæt filen dogs før linje 1 og efter linje 6.

Arkivér som: hundogkat

:0r dogs

:w hundogkat (eller :w! hundogkat = overskriver gammel fil)

Modul 6 - Løsningsforslag - Filtræet

Opgave 6.1. a

```
pwd  
ls
```

Øvelse 6.1. b

```
cd /  
ls -a
```

```
cd /usr ; ls  
cd /bin ; ls  
cd /etc ; ls  
cd /dev ; ls  
cd /home ; ls
```

Øvelse 6.1. c

```
ls -l /etc/passwd
```

Svaret står i tallet foran månedsnavnet i datoen.

```
cd ; ls -l bibel
```

Svaret står i tallet foran månedsnavnet i datoen.

Øvelse 6.1. d

```
cd /tmp  
pwd  
cd  
cd ./tmp  
pwd
```

```
cd /tmp  
cd ~/tmp  
pwd  
cd /tmp  
pwd  
cd ~/tmp  
pwd
```

```
cd /tmp
cd
pwd
cd tmp
pwd
cd
```

Øvelse 6.1. e

```
cd

mkdir sub
cd dir
cd ../sub
cd ../dir

cd ~/sub
cd ~/dir
```

Øvelse 6.1. f

```
cd
./g

cd sub
../g

cd /tmp
~/g

cd
../bruger1/g
```

Øvelse 6.1. g

```
cd ~/sub
>a >b
ls -l
cd
```

```
rm -r sub
```

Modul 7 – Løsningsforslag – Data og filer

Øvelse 7.1. a

```
cat fil1  
cat fil1 fil2 fil3  
cat lark
```

Øvelse 7.1. b

```
less bibel
```

Øvelse 7.1. c

```
head -4 names  
tail -4 names  
tail -n +3 names
```

Øvelse 7.1. d

```
nl -ba lark  
cat -n names  
cat -n linefeed
```

Øvelse 7.2. a

```
mkdir links  
ln lark links/lark.ln  
ls -li links/lark.ln lark
```

Øvelse 7.2. b

```
ln -s ../lark links/lark.sy  
ls -li links/lark.sy lark
```

Øvelse 7.2.c

```
cp lark links  
ls -li links
```

```
head -4 lark.ln lark.sy
```

```
mv lark lark0  
head -4 lark.ln lark.sy
```

```
rm lark0
head -4 lark.ln lark.sy
```

```
cp saturday lark
head -4 lark.ln lark.sy
```

Øvelse 7.3.a

```
grep æbletræ bibel
```

Øvelse 7.3.b

```
grep -c æbletræ bibel
```

Øvelse 7.3.c

```
grep -n lærke lark*
eller
grep -n lærke lark[1-9] lark[A-Z]
```

Øvelse 7.3.d

```
grep -l lærke *
grep -l lærke * 2>/dev/null
```

Øvelse 7.3.e

```
grep -c lærke *
eller
grep -c lærke * | grep -v ':0'
eller
grep -c lærke * | grep -v '^:0'
```

Øvelse 7.3.f

```
grep \# fill
```

Øvelse 7.4.a

```
find /home -name txt
```

Øvelse 7.4.b

```
find ~ ! -user bruger1
```

Øvelse 7.4.c

```
find ~ -name '*.c'
```

Øvelse 7.4.d

```
find ~ -name 'fil*' -exec cat {} \;
```

Øvelse 7.5.a

```
sort -r dyr1
```

Øvelse 7.5.b

```
sort -k2n dyr1
```

Øvelse 7.5.c

```
sort -k3n dyr1
```

Øvelse 5.d

```
ls -l | sort -k5n
```

Øvelse 7.5.e

```
sort -k3n -k2n dyr1
```

Øvelse 7.6.a

```
sort dyr | uniq
```

Øvelse 7.6.b

```
sort dyr | uniq -u
```

Øvelse 7.6.c

```
sort dyr | uniq -c
```

Øvelse 7.7.b

```
cmp sat1 sat2
```

Øvelse 7.7.c

```
md5sum giraf* | sort
```


Øvelse 7.8.a

```
file /etc/passwd /usr/bin/passwd ~/main.c
```

Øvelse 7.8.b

```
file kap7.txt
```

```
cat kap7.txt
```

Øvelse 7.8.c

```
file kap7.doc
```

```
strings kap7.doc
```

Øvelse 7.9.a

```
wc bibel
```

Øvelse 7.9.b

```
strings kap7.doc | wc
```

```
wc kap7.txt
```

Øvelse 7.9.c

```
sort dyr | uniq | wc -l
```

Modul 8 - Løsningsforslag - Håndtering af printere

Øvelse 8.1. a

```
lpc status  
eller  
lpstat -t
```

Øvelse 8.1. b

```
lp -d PDF fil1  
eller  
lpr -P PDF < fil2  
eller  
cat fil2 | lpr -P PDF
```

Øvelse 8.1. c

```
ls -l PDF
```

Øvelse 8.1. d

```
evince PDF/fil1.pdf  
eller  
evince _stdin_.pdf
```

Modul 9 – Løsningsforslag - Processer

Øvelse 9.1. a

```
ps
```

Øvelse 9.1. b

```
ps -ef | more
```

```
ps -ef | less
```

Øvelse 9.1. c

```
ps -ef | wc -l
```

Der er – pga. headeren – een proces mindre end tallet der vises.

Øvelse 9.1. d

```
ssh dit_brugernavn@instructor
```

```
...
```

Øvelse 9.1. e

```
ps -ef | wc -l
```

Øvelse 9.1. f

```
ps -ef | grep " $$ "
```

Øvelse 9.1. g

```
ps -fu bruger1
```

Øvelse 9.1. h

```
ps -fu root
```

Øvelse 9.1. i

```
ps -ef | grep bash | grep -v grep
```

Øvelse 9.1. j

```
ps -ef | sort -k3n -k2n | less
```

Øvelse 9.1. k

```
ps -fp '124 456 1734'
```

eller:

```
ps -fp 124,456,1734
```

```
ps -fu 'bruger1 bruger3 bruger5'
```

eller:

```
ps -fu bruger1,bruger3,bruger5
```

Modul 10 – Løsningsforslag – Shellen og dens funktion

Øvelse 10.1. a

```
ls -d *.c
ls -d ????
ls -d *A*
ls -d fil*
```

Øvelse 10.1. b

```
ls *a a*   Lister filer der slutter med a og starter med a
           i alfabetisk rækkefølge.
echo *a a*   Udskriver liste med filer der slutter med a og
           herefter liste med filer der starter med a
```

Øvelse 10.1. c

```
echo '*a a*' og echo '*a a*'
I begge tilfælde udskrives en tekststreng indeholdende:
*a a*
```

Øvelse 10.2. b

```
sort -k2 SuperUsers > sort_navne
echo "Anders And $(date)" >> sort_navne
sort -ro sort_navne sort_navne
```

Øvelse 10.2. c

```
find /home -name txt 2>/dev/null > find.log
```

Øvelse 10.2. d

```
find .. -name '*.c' >find.log 2>&1
filen find.log indeholder både en liste over filer og
fejlmeldelser
stdout sendes til filen find.log og stderr sendes samme
sted hen som stdout (nemlig til filen find.log)
```

Øvelse 10.2. e

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

filen find.log indeholder en liste over filer (fejlene blev udskrevet til skærmen)

stderr(2) bliver sendt til skærmen (2>&1) da stdout(1)

endnu er default (nemlig skærm), herefter bliver stdout(1)

sendt til filen fejl.log

Øvelse 10.3. a

```
ps -ef | less
```

Øvelse 10.3. b

```
ps -u root | less
```

Øvelse 10.3. c

```
cat -n bibel | head -17437 | tail -8
```

eller

```
cat -n bibel | tail -n +17430 | head -8
```

Øvelse 10.3. d

```
./oftest < bibel.word > output
```

```
ls -l /usr/bin | sort -k5nr | less
```

Øvelse 10.3. e

```
find .. -name '*.c' | less
```

Øvelse 10.5. e

```
PS1='$PWD: '
```

eller

```
PS1='$ (pwd) : '
```

Øvelse 10.6. a

```
#!/bin/bash
```

```
echo Filnavne på 3 tegn:
```

```
ls -ld ???
```

```
echo
```

```
echo Filnavne på 4 tegn:
```

```
ls -ld ????
```

```
echo
```

```
echo Filnavne på 5 tegn:
```

```
ls -ld ?????
```

```
echo
```

```
echo Rapport fra $(date) af: $LOGNAME
```

Modul 11 – Løsningsforslag - Tidsafhængig jobstart

Øvelse 11.1. a

```
at now + 2 minutes  
    ls -l
```

CTRL+D

Øvelse 11.1. b

```
at now + 5 minutes  
    cat sat1 lark1 > sangbog
```

CTRL+D

Øvelse 11.1. c

```
at now + 5 minutes  
    head -4 sat1 lark1 > vers1
```

CTRL+D

Øvelse 11.1. d

```
at 15:45  
    echo "Kl. Er 15:45 og vi går snart hjem"
```

CTRL+D

Øvelse 11.1. e

```
at now + 5 minutes  
    cp lark lark.ny
```

CTRL+D

Øvelse 11.2. a

```
atq  
eller  
at -l
```

Øvelse 11.2. a

```
echo ls | at teatime  
atq
```


Modul 12 – Løsningsforslag - Brugere

Øvelse 12.1. a og b

```
vi mit_script.sh
```

```
#!/bin/bash
# Dette er et shell-script
# der laver en udskrivning af dato, klokkeslet, m.v.
date
echo
echo Kære $LOGNAME
echo
echo Indloggede personer:
echo
who | sort
```

OBS! For at scriptet skal kunne afvikles, skal det gøres eksekverbart:

```
chmod +x mit_script.sh
```

```
./ mit_script.sh
```

eller

```
sh mit_script.sh
```

Øvelse 12.2. a

Id

Øvelse 12.3. a – c

Ingen løsningsforslag – prøv selv

Øvelse 12.4. a

```
grep 'bruger*' /etc/passwd
```

Modul 13 – Løsningsforslag - Filrettigheder

Øvelse 13.1.a

```
> fil.ret
ls -l fil.ret
ln -s fil.ret symbol.ret
ls -l symbol.ret
```

```
mkdir dir.ret
ls -ld dir.ret
```

Øvelse 1.b

```
ln fil.ret link.ret
ls -l link.ret
```

```
chmod +x fil.ret
ls -l fil.ret link.ret
```

Øvelse 13.2.a

```
> fil.ret.1
chmod u+x fil.ret.1 ; ls -l fil.ret.1
chmod g-r fil.ret.1 ; ls -l fil.ret.1
chmod o+w fil.ret.1 ; ls -l fil.ret.1
chmod a-rwx fil.ret.1 ; ls -l fil.ret.1
chmod g+rx fil.ret.1 ; ls -l fil.ret.1
chmod a+rwx fil.ret.1 ; ls -l fil.ret.1
```

Øvelse 13.2.b

```
> fil.ret.2
chmod 744 fil.ret.2 ; ls -l fil.ret.2
chmod 704 fil.ret.2 ; ls -l fil.ret.2
chmod 706 fil.ret.2 ; ls -l fil.ret.2
chmod 0 fil.ret.2 ; ls -l fil.ret.2
```

```
chmod 777 fil.ret.2 ; ls -l fil.ret.2
```

Øvelse 13.3. a

```
umask 0  
> fil.ret.3  
ls -l fil.ret.3
```

```
umask 066  
rm fil.ret.3  
> fil.ret.3
```

```
umask 006  
rm fil.ret.3  
> fil.ret.3
```

```
umask 004  
rm fil.ret.3  
> fil.ret.3
```

```
umask 666  
rm fil.ret.3  
> fil.ret.3
```

Øvelse 13.3. b

Hvornår skal man vælge umask 033 fremfor umask 022 ?

Svar: Det giver ens filrettigheder, men forskellen på nye kataloger ligger i at både gruppen og alle andre ikke kan kigge i nye kataloger (x mangler)

Øvelse 13.3. c

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

Svar: at stille dem tilbage til det opreindelige her:

```
umask 022
```

Stilles også tilbage hvis du lukker terminalvinduet og åbner et nyt.

Modul 14 - Løsningsforslag - Kopiering og komprimering

Øvelse 14.1.a

```
cd
mkdir tardir
tar -cvf tardir/fil1-4.tar fil[1-4]
ls -l tardir/fil1-4.tar
tar -tvf tardir/fil1-4.tar
cd tardir/
tar -xvf fil1-4.tar
ls -l
cd ~/dir
ls -l
cat *
tar -xvf ../tardir/fil1-4.tar fil1 fil2 fil3
ls -l fil1 fil2 fil3
cat fil1 fil2 fil3
```

Øvelse 14.2.a

```
cd
ls -l txt
bzip2 txt
ls -l txt.bz2
bzip2 txt.bz2 | less
bzip2 txt.bz2

bunzip2 txt.bz2
gzip txt
ls -l txt.gz
zcat txt.gz | less
gzip txt.gz
gunzip txt.gz
```

Modul 15 - Løsningsforslag - lokalnet m.m.

Øvelse 15.1 a

```
ssh-keygen -f .ssh/peter-ssh-key
```

Øvelse 15.1. b

```
ls -l .ssh
```

```
cat .ssh/dit-navn-ssh-key
```

```
cat .ssh/dit-navn-ssh-key.pub
```

Øvelse 15.2. a

```
ssh bruger9@192.168.x.xxx
```

Øvelse 15.2. b

```
mkdir claus
```

Øvelse 15.2. c

```
cat fil1 >claus/min-fil1
```

```
cat fil2 >claus/min-fil2
```

```
cat bibel >claus/min-fil3
```

Øvelse 15.2. d

```
CTRL+N
```

Øvelse 15.2. e

```
scp -r bruger9@192.168.x.xxx:/home/bruger2/claus ~
```

Øvelse 15.2. f

```
ls -l ~/claus
```

Øvelse 15.3

(Fri øvelse)