

# Linux-julbord

Andreas Ehliar & Kent Engström

2010-11-30

- 1 Processhantering
- 2 Nätverk
- 3 Bildhantering med netpbm
- 4 Överlevnadstricks för bash
- 5 dbus

# screen

Screen är en textbaserad fönsterhanterare. Bra till exempel för att...

- ... få flera fönster över en enda SSH-inloggning
- ... hålla igång program trots att nätverkskoppel går ner
- ... starta pratsamma program vid uppstart

## screen

### Grunder

- Starta och avsluta screen
- Kommando-knappen (normalt C-a, inte bra ihop med Emacs)
- `screen -e ^zz`
- escape `^zz` i `~/ .screenrc`
- Hjälp (C-z ?)
- Skapa flera fönster (C-z c)
- Byta fönster (C-z 0, C-z 1, ..., C-z SPC, C-z DEL)

# screen

## Detach och reattach, multidisplay

- Detach (C-z d)
- `screen -r`
- `screen -d -r`
- `screen -d -R`
- `screen -D -R`
- `screen -D -RR`
- ...
- `screen -x`

## screen

### Klippa och klistra, scrollback

- Inled med C-z ESC
- Spring runt med pilar eller i, j, k, l
- Siffror före: 100k går 100 rader upp
- Markera med SPC (början och slutet)
- Klipp in med C-z ] (obekvämt för svenskar?)
- bind p paste . i ~/.screenrc
- Mest för historien bakåt, sällan klipp och klistra?

## screen

Starta flera program i screen

- Smidigt till exempel vid systemstart
- `screen -d -m -c configfile -S mydaemons`
- `screen -r -S mydaemons`

### configfile

```
escape ^zz
```

```
screen top
```

```
screen emacs -nw
```

# pstree

## Överblick över processer

- `pstree`
- `pstree -p`
- `pstree -u`
- `pstree -h`



# lsof

## List Open Files

Visa vilka filer och liknande processer har öppna.

- `lsof`
- `lsof -i`
- `lsof -p PID`
- `lsof /mountpoint`
- `lsof +D /arbitrary/directory`

## Scenario

- Du får följande meddelande i sysloggen ifrån en demon en konsult lämnade efter sig:
  - Nov 26 11:37:42 yendor bigbinaryblob: Warning:  
The file is corrupted
  - Nov 26 11:37:42 yendor bigbinaryblob: Exiting
- Vad gör du? (Konsulten har tagit med sig källkoden till Bahamas)

# strace

- strace kan användas för att få spårutskriften av alla systemanrop ett program gör
- Starta spårning: `strace filnamn`
- Anslut till redan existerande process: `strace -p PID`
- Kör som annan användare (om du behöver felsöka ett program med suid-biten satt): `strace -u username`

## strace - Skriva ut strängar

- strace skriver normalt sett bara ut 32 tecken av en sträng
- För att se längre strängar: `strace -s 1000`

## strace - Filtrera systemanrop

- För mycket utdata?
- Begränsa till vissa systemanrop:  
`strace -e write,open,read,close`
- Begränsa till enbart systemanrop som tar ett filnamn:  
`strace -e file`
- Det finns andra alias också som `network`, `process`, `ipc`.  
(Se mansidan)
- Skriv utdata till fil istället för `stderr`:  
`strace -o TRACEFIL`

## Andra program för spårutskriften

- Scenario: bigbinaryblob använder 100% CPU men gör inga systemanrop
- Har det kraschat? Eller gör det något vettigt?
- Möjlig lösning: `ltrace`

# ltrace

- ltrace används på samma sätt som strace förutom att det även kan spåra biblioteksanrop
- Används på ungefär samma sätt som strace. (-s, -f, -p, -u fungerar likadant)

# truss

- strace fungerar nog inte som du vill på Solaris
- strace (1m) - print STREAMS trace messages
- Använd truss istället. (Finns även på FreeBSD)



## Prioritetering av processer

- De flesta torde känna till `nice`
- Starta en kompilering med lägsta prioritet:  
`nice -n 19 make -j 4`
- Se till att din server får högsta prioritet:  
`nice -n -20 ./myserver`
- (Endast root får sätta en högre prioritet i normalfallet)

# Mer processprioritering

- Ändra prioritet på en körande process: `renice`
- Exempel: Du har av misstag surfat in på en sida med 100 flash-applikationer:
  - `renice 19 $(pgrep firefox)`

# Avancerad processhantering

- I vissa fall kan du vilja sätta realtidsprioritet på en process
- Exempel: `chrt -r 1 ./myserver`

## Avancerad processhantering (forts)

- I en del fall är det hårddisken som sätter begränsningen:
- Exempel: Se till att din find enbart använder disken när inga andra processer gör det:
  - `ionice -c 3 find / -name '*.v'`
  - (3 är för idle-prioritet)

## ssh: Port forwarding

- Scenario: Du sitter hemma bakom en brandvägg men vill ändå komma åt din dator hemma ifrån jobbet
- Lösning: Logga in på din jobbdator och forwarda port 4000 till ssh-porten på den dator du loggat in ifrån
- `ssh -R 4000:localhost:22 loginserver.isy.liu.se`

## ssh: Port forwarding (forts)

- Port forwarding kan också användas för att skapa en krypterad förbindelse ifrån din lokala dator till en annan dator.
- Exempel: Logga in på `stalhein.lysator.liu.se` och forwarda den lokala porten 4894 till `kom.lysator.liu.se` (port 4894).
  - `ssh -L 4894:kom.lysator.liu.se:4894 stalhein.lysator.liu.se`
  - (På så sätt kan bara användare med åtkomst till lysators nätverk avlyssna din KOM-anlutning)

## ssh: Master mode

- Behöver du ofta köra program på en viss annan dator?
- Använd isåfall master-mode i ssh
- Exempel: Logga in på en server och starta ssh i Master-mode som styrs av av socketen `$HOME/.ssh-master`
- `ssh -S ~/.ssh-master -M stalhein.lysator.liu.se`

## ssh: Master mode (forts)

- För att koppla upp sig igen mot denna dator:
- `ssh -S ~/.ssh-master stalhein.lysator.liu.se`
- Den befintliga anslutningen återanvänds vilket innebär att det går väldigt snabbt att göra en ny anslutning
- I sökvägen för den socket som anges med `-S` kan du även använda exempelvis `%h` och `%r`. Dessa ersätts med datornamnet respektive användarnamnet.



## Escapetecken i ssh

- Precis som i telnet finns det ett escapetecken i ssh. Som standard är detta satt till ~
- Användning:
  - Lista antalet forwardade sessioner: ~#
  - Lägga ssh i bakgrunden: ~^Z
  - Avsluta ssh: ~.
  - Du behöver ibland trycka enter innan ssh accepterar din escape-character.
- (Det finns några fler möjligheter, se mansidan för mer info.)

## ssh: Smidig filkopiering

- Det händer ganska ofta att jag vill kopiera en fil ifrån den dator jag är inloggad på.
- Här kan du givetvis använda scp, sftp eller sshfs, men det är ofta lite drygt att ordna detta i efterhand
- Lösning: Använd master-mode och ett par funktioner, `sshpush` respektive `sshpop`

# ssh: Smidig filkopiering: sshpush

```
function sshpush {  
    if echo "$1" | egrep -q '^/'  
    then  
        echo "$1" > $HOME/.ssh-pushed  
    else  
        echo "$PWD/$1" > $HOME/.ssh-pushed  
    fi  
    echo "Scheduling $(cat $HOME/.ssh-pushed) for copying."  
    echo 'To copy the file, press ^^Z and then type sshpop'  
}
```

# ssh: Smidig filkopiering: sshpop

```
function sshpop {
    SN=$HOME/.ssh-master-$(basename $(tty))
    (FILENAME="/usr/bin/ssh -S $SN dummy cat .ssh-pushed)"
    if test -f "$(basename "$FILENAME")"
    then
        echo "$FILENAME already exists, aborting!"
    else
        echo "Scheduling remote file $FILENAME for copying"
        scp -o "ControlPath $SN" "dummy:$FILENAME" .
    fi) &
    %/usr/bin/ssh
}
```

# ssh: alias för att starta ssh i samband med sshpush/sshpop

```
function sshp {  
    SN=$HOME/.ssh-master-$(basename $(tty))  
    /usr/bin/ssh -M -S $SN "$@"  
}
```

# ssh: Övrigt

- Använd `$HOME/.ssh/config`
- Exempel:
- Host jobb  
Hostname `loginserver.isy.liu.se`  
User `ehliar`  
Port `80`

# pdsh

Kör kommando (via SSH) på många datorer samtidigt:

- `pdsh -w mail,www,dns uname -r`
- `pdsh -w n[1-10,101-120] uname -r`
- `pdsh -a service openssh restart`
- Option `-f`: hur många samtidiga SSH-anrop
- Option `-u`: timeout

# dshbak

Sammanställning av data från pdsh och liknande program:

- `pdsh -w n[1-100] rpm -q glibc | dshbak`
- `pdsh -w n[1-100] rpm -q glibc | dshbak -c`
- Option `-c`: slå samman datorer med identisk utmatning



## pdcp och rpdcp

Kopiera data till (pdcp) och från (rpdcp) många datorer:

- `pdcp -w n[1-100] /etc/hosts /etc/hosts`
- `pdsh -w n[1-100] 'rpm -qa|sort >allpackages.txt'`
- `mkdir resultdir`
- `rpdcp -w n[1-100] allpackages.txt resultdir`

# socat

- “Kopplar ihop något med något.”
- “Netcat på steroider med inbyggd motorsåg”
- “Multipurpose relay (SOcket CAT)”
- `socat [options] something something`

# socat

Vi börjar med några fåniga enkla exempel:

- `socat EXEC:ls STDOUT`
- `socat -u EXEC:ls CREATE:listing.txt`
- `socat OPEN:listing.txt STDOUT`
- `socat OPEN:listing.txt -`

# socat

Ersättning för netcat:

- `socat TCP-LISTEN:12345 -`
- `socat -u TCP-LISTEN:12345 CREATE:saved-stuff.txt`
- `socat OPEN:datatosend.txt TCP:192.168.1.1:12345`
- `somecommand | socat - TCP:192.168.1.1:12345`

# socat

Felsöka nätverksprotokoll?

- `socat - TCP:smtp.liu.se:25`
- `socat READLINE,history=$HOME/.smtphistory  
TCP:smtp.liu.se:25,crnl`

# socat

Ett TCP-relä, en fortune-server och en top-server:

- `socat TCP-LISTEN:8000,reuseaddr,fork  
TCP:www.liu.se:80`
- `socat TCP-LISTEN:8000,reuseaddr,fork  
EXEC:/usr/bin/fortune`
- `socat -U TCP-LISTEN:8000,reuseaddr,fork  
EXEC:/usr/bin/top,pty`

# socat

Vem behöver `tail -f`?

- `socat -u /var/log/messages,seek-end=0,ignoreeof -`

Läs mer i man-sidan, men börja gärna med exempeldelen!

# openssl s\_client

Felsökning av SSL/TLS-förbindelser:

- `openssl s_client -connect www.liu.se:443`
- Visar certifikatkedjan (eller avsaknad av densamma)
- Visar servercertifikatet (eller alla certifikat med `-showcerts`) i PEM-format
- Låter dig prata protokollet som ligger ovanpå SSL/TLS



# openssl x509

Visning av X.509-certifikat:

- `openssl x509 -noout -text < mycert.pem`
- Visar innehåller i certifikatet.

# rsync

Visst använder ni redan `rsync` för effektiv kopiering av kataloger med många och stora filer i?

- `rsync -av dir1/ dir2/`
- `rsync -av localdir/ server:remotedir/`
- `rsync -av --delete localdir/ server:remotedir/`

# unison

Tvåvägsynkronisering av kataloger:

- `unison dir1 dir2`
- `unison common ssh://server/common`

# collectl

“vmstat på steroider”

- `collectl`
- `collectl -i2 -oT -scnm`
- Kan köras som server för senare uppspelning med `-p`

# Bildhantering med netpbm

- Scenario: Du behöver skapa thumbnails av ett stort antal JPEG-filer
- Lösning:

```
• for i in ./*.jpg
do
    jpegtopnm "$i" | pnmscale -ysize 160 120 |\
    pnmtojpeg --quality 60 > "scaled_$i"
done
```

## Mer om netpbm

- Netpbm innehåller ett stort antal program för att konvertera till PNM: `jpegtopnm`, `pngtopnm`, `tifftopnm`, ...
- På samma sätt kan man konvertera från PNM till ett stort antal format: `pnmtojpeg`, `pnmtopng`, `pnmtotiff`, ...
- Slutligen finns det ett antal program för att manipulera PNM-filer, exempelvis `pnm-scale`, `pnmcut` och `pnmcomp`

## Klippa och klistra i bilder

- Scenario: Du har ett stort antal bilder som du vill göra om till en slideshow. Längst uppe till vänster ska det finnas en thumbnail av nästa bild.
- Lösning: Använd `pnm-scale` för att skala ner nästa bild och `pnmcomp` för att lägga ihop två bilder.
- (Du kan även använda `pnmcut` för att beskära bilder.)

## Exempelscript där pnmcomp används

```
previmg="";num=0
for i in *.jpg ""
do
    if test -n "$previmg"; then
        jpegtopnm "$previmg" |\
            pnm scale -xysize 1024 768 > image.pnm
    fi
    if test -n "$i"; then
        jpegtopnm "$i" |\
            pnm scale -xysize 160 120 > thumb.pnm
        pnmcomp -xoff 0 -yoff 0 thumb.pnm image.pnm |\
            pnmtjpeg > $num.jpg
    else
        pnmtjpeg image.pnm > $num.jpg
    fi
    previmg="$i";num=$((num+1))
done
```



# Överlevnadstricks för bash

- Hantera historik:
  - pil-upp/pil-ner är välkänt
  - Inkrementell bakåtsökning i historiken: `^R`
  - Klipp in sista argumentet ifrån föregående kommando: `Alt-.` (eller `Escape .`)

## Överlevnadstricks för bash (forts)

- Att ta bort gamla backupfiler är ganska farligt då `~` och `*` är alldeles för nära return.
- Lösning: `alias rmbak="echo rm *~;rm *~"`

## bash: Flytta sig mellan bibliotek

- Gå till det förra biblioteket: `cd -`
- Lägg ett bibliotek på stacken: `pushd /lib/modules/  
2.6.35-22-generic/kernel/drivers/net/usb/`
- Poppa ett bibliotek från stacken: `popd`

## Viktiga miljövariabler

- Scenario 1: Ett program vägrar starta, kanske på grund av problem med inställningsfilerna
- Lösning 1: Ändra HOME till ett temporärt bibliotek och testa att starta programmet igen:
- `HOME=/home/ehliar/tmphome firefox`

## Viktiga miljövariabler

- Scenario 2: Ett visst universitet envisas med att ha `/bin/tcsh` som standardskal
- Lösning 2: Ändra SHELL i antingen `.login` eller `.xinitrc`<sup>1</sup>
- `setenv SHELL /bin/bash`  
`exec /bin/bash --login`

---

<sup>1</sup>Lämnas som en övning åt lyssnaren.

## Tips för att skriva shellsript

- Se till att ett fel avslutar ett script: `set -e`
- (Fungerar inte riktigt för allting, men bra mycket bättre än ingen felkontroll alls)

## Vanligt fel: Hanterar ej mellanslag korrekt

- Dåligt exempel:

```
for i in *.jpg
do
    mv $i $HOME/tmp
done
```

- Bättre exempel:

```
for i in *.jpg
do
    mv "$i" $HOME/tmp
done
```

- (Hanterar dock inte filnamn som börjar med -)

## Hantera filnamn som börjar med -

- Förslag på lösningar:
  - Lägg till ett ./ i början av filnamnet: `for i in ./*.jpg`
  - Använd `--` argumentet till de kommandon som stödjer detta:  
`mv -- "$i" $HOME/tmp`



## Hantera argument med mellanslag i sig

- Felaktigt: `for i in $*`
- Korrekt: `for i in "$@"`
- (Under förutsättning att du sedan använder "\$i")

# Vad är dbus?

- dbus används för interprocesskommunikation på de flesta linuxdistributioner idag
- Användningsområden:
  - Slå av/starta om datorn
  - Styra mediaspelare
  - Styra backlight
  - Hantera diskar
  - Hantera nätverk
  - etc. . .

## qdbus: Lista klienter

- Lista alla klienter på sessionsbussen: `qdbus`
- Lista alla klienter på systembussen: `qdbus --system`

## qdbus: Hitta vilka metoder som finns hos en klient

- `qdbus org.gnome.ScreenSaver`
- `qdbus org.gnome.ScreenSaver /`

## qdbus: Anropa metoder hos en klient

- `qdbus org.gnome.PowerManager  
/org/gnome/PowerManager/Backlight  
org.gnome.PowerManager.Backlight.SetBrightness 50`

## qdbus: Några exempel:

- Styra skärmläckaren via `org.gnome.ScreenSaver`
- Styra mediaspelaren Rhythmbox via `org.gnome.RhythmBox`
- Logga ut via `org.gnome.SessionManager`

# Slutsatser

- Det finns många småknep i Linux som gör livet betydligt enklare eller roligare
- Vi har visat några av våra favoritknep, vilka är dina?