

Laboration nr 9 BashScripting

UDEV-REGLER och Skript för kopiering av viktiga filer via usb.
Ett backup-skript.

List of contents

| | |
|------------------------------------|---|
| Syfte:..... | 1 |
| Tillvägaprocess/Tillvägasätt:..... | 1 |
| Så här ser huvud skriptet ut:..... | 3 |
| Fördelar:..... | 4 |
| Nackdelar:..... | 4 |
| Reflektioner..... | 5 |
| Grupp Reflektion:..... | 5 |
| Oliver's Reflektion:..... | 5 |
| Källförteckning:..... | 6 |

Syfte:

Skriva ett fungerade skript så när man pluggar in en USB sticka ska ett backup-skript starta. Och det skriptet sparar dem viktiga dokumenten på USB-MINNET, ett viktiga dokument som i detta fall är .baschrc och .vimrc.

Vi skapar först UDEV-REGLER och en fil hand i hand och sedan skapade vi själva skriptet som vi kallar för casetest.sh till en början som sedan fick namnet Case8.sh när det var färdigt.

I laborationen arbetas mycket med filöverföring och även studera utanför det vanliga när det kommer till UDEV-REGLER och även bekanta sig med rsync.

Tillvägaprocess/Tillvägasätt:

Detta utförs torsdag 9 januari.

Väljer ut vilket case som laborationen kommer att handla om och gruppen bestämde sig för case 8 eftersom den har rimlig svårighetsgrad och tips ifrån läraren.

Vi har inge tidigare kundskap om Udev och börjar med att läsa från x-antal olika sidor (dem hänvisas till källförteckningen längst ner i uppsatsen)

Gruppen delar upp sidorna som dem hittat via google.se och information samlas om hur dem ska kunna gå tillväga för att nå ett lyckat resultat.

Efter att ha läst och teoretiskts försökta att lära sig börjar första bokstäverna rulla in och vi får det att fungera någorlunda men det är väldigt lång process för ett så kort skript. Det första skriptet vi gör är för UDEV-REGLERNA och det är 1 rad kod med timmars av teoretiskt arbete.

I skriptet så kopierar vi filerna som vi önskar att kopiera till USB-MINNET som är mountat innan. Allt blir i mer klartext efter vi har bifogat in skriptet.

Personerna får definiera en grupp variabler och kommer skriva en tydlig förklaring för var dem olika variablerna har för uppgift.

Efter att ha nåt ett lyckat resultat fokuseras det på förbättringar och rapportering av utförandet men vi är nöjda men inte mycket nöjda och kommer arbeta med förbättringar av skriptet tills det att vi har lämnat in laborationen.

Efter att ha fått casetest.sh fungerande och åter igen testat så skapar vi kopia av den filen och börjar även att försöka ge dig som läser detta en skaplig förklaring av det vi har skrivit i dem två skripten men även förbättra vårt nuvarande casetest.sh skript.

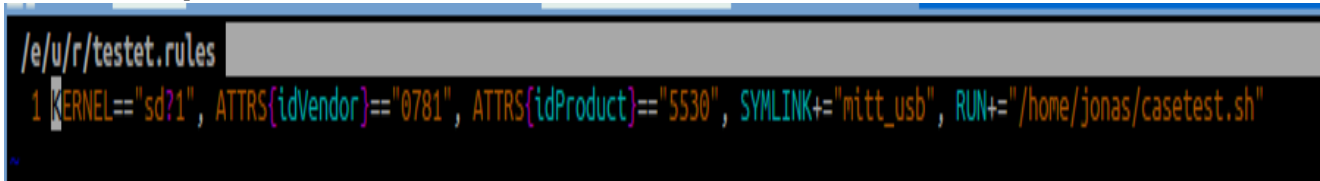
UDEV-REGLER skriptet heter testet.rules

Och backup-skriptet heter casetest.sh/Case8.sh

Vi kommer låta tankarna leka fritt efter skoldagens slut och fortsätta på fredag morgon.

Förklaring av skripten görs nedan:

UDEV-skriptet



```
/e/u/r/testet.rules
1 KERNEL=="sd?1", ATTRS{idVendor}=="0781", ATTRS{idProduct}=="5530", SYMLINK+="mitt_usb", RUN+="/home/jonas/casetest.sh"
```

Vad dem olika delarna betyder:

"KERNEL=="sd?1" = Namnet som USB-MINNET får ifrån kärnan.

"ATTRS{idVendor}=="0781"" = Där är id-numret ifrån tillverkaren av usb-minnet

"ATTRS{idProduct}=="5530"" =Usb-minnets egna produkt id

"SYMLINK+="mitt_usb" = skapar en symbolisk länk som heter /dev/mitt_usb som pekar på /dev/sd?1.

"RUN+="/home/jonas/casetest.sh" = Görs så att skriptet ska köras när usb:et kopplas in

Detta utförs fredag 10 januari.

Så här ser huvud skriptet ut:

```
#!/bin/bash
# Jonas Lundsholm - Oliver Gerling
# LX13
# Case 8
# This script will be activated when the right USB drive is connected.
# It will then copy the selected files to the USB drive automatically.

# Defining some variables
File1="home/jonas/.bashrc"
File2="/home/jonas/.vimrc"
Usb="/mnt/mitt_usb/"
Mkusb="mkdir $Usb"
Rmusb="rm -r $Usb"

if [ ! -e $Usb ]; then      # If the $Usb does not exist, it will be created first and then the USB drive will be mounted
there.
    $Mkusb
    mount /dev/mitt_usb $Usb
else
    mount /dev/mitt_usb $Usb # Mounts the connected USB drive to $Usb
fi

rsync -azvv $File1 $Usb/.bashrc # Copy .bashrc to $Usb
rsync -azvv $File2 $Usb/.vimrc  # Copy .vimrc to $Usb

# Här ska vi pipa till zenity för att kunna se överföringen av filerna grafiskt.
```

```
umount /dev/mitt_usb      # Unmounts the USB drive after file transfer is complete.

if [ -e $Usb ]; then      # Removes the $Usb after the USB drive has been unmounted.
    $Rmusb
fi
exit 0
```

Första stycket är information om skriptet.

Andra stycket defineras olika variabler. Där **file1** är .bashrc filen och **file2** är .vimrc filen.

Usb=/mnt/mitt_usb/ är där usb stickan kommer vara och det skapas i en mapp i mnt.

Dem återstående 2 usb variablerna skapar och tar bort usb.

Första if satsen är en sanity check där man kollar om usb är kopplat annars kopplar det och mountar usb-minnet.

Rsync raderna är det som kopierar filen, dem variablerna som används i den raden är definierade i andra stycket. Sedan efter det är gjort avmonteras usb-minnet.

Fördelar:

- Mycket arbete och informations letande på egen hand.
- Fördjupning med udev regler som inte har arbetas med sedan partitionering i linux grundkurs.
- 2 personer 1 skript gör att man kan rapportera mycket enklare och få en klar bild och uppfattning varför vi skriver skriptet som vi gör och även testa olika versioner av skriptet utan att bli "lost". 7
- Göra skriptet tillsammans med en nivå som är jämn mellan personerna.
- Slutföra och göra en fungerande funktion utan att få hjälp av lärare och klasskamrater.
- Går alltid att förbättra så arbetet sprids ut över en längre period.

Nackdelar:

- **Största nackdelen: 1 skript 2 personer** betyder också att det blir en del passivt arbete från partnern på bänken som får engagera sig i rapporteringen.
- Brist på tidigare erfarenheter och fanns inga kompletta sidor utan man fick ta bitar från olika sidor tills man fick ett fungerade pussel.
- Inget om if/case och var mycket arbete för så lite kod.
- Bra funktion och mycket arbete för lite kod – mer kod ser alltid bättre ut

- tyvär, ett arbete på 8 sidor är bättre än ett arbete på 5.
- Båda är väldigt ny på scripting och gör det på grundlig nivå men det negativa är hur tidskrävande det var.

Reflektioner

Grupp Reflektion:

Jonas började med caset innan Oliver och efter en kort tid så börjar gruppen att sammarbeta och börjar skriva skriptet. Gruppen fick uppfattningen av att man var klar men det visade sig alltid vara redigering på redigering så det blir "redigerings-ception" där man lätt kommer bort sig varför man gjorde ändringarna som man gjorde men stort tack till rapportering av arbetet och att vi jobbade i en grupp av 2.

Det var många steg och ett dussin av provförsök av skriptet innan det fungerade som planerat eftersom individerna i gruppen är på en väldigt grundlig nivå.

Vi började med att skriva grunderna och sedan bytte till rsync som gjorde det enklare – detta är inte ett universalt skript och måste redigeras om man vill använda det själv. Efter att ha nått ett lyckat resultat fokuseras det på förbättringar och rapportering av utförandet som har varit väldigt framgångsrikt under eftermiddagen.

Oliver's Reflektion:

Hoppade som sagt in lite senare i uppgiften men fick en stor roll av Jonas direkt framförallt när det kommer till valet att filer som ska göras en backup-skript på och varför jag tyckte dem. Även att definiera variabler som gör skriptet tydligare och enklare att tyda "tar-syntaxen".

Detta liknar laboration 6 öving 5 så det är bekant med backup-skript för mig men båda jag och Jonas är oerfarna på Udev-regler. Vill inte ljuga men Jonas gjorde större delen med UDEV-REGLERNA och jag lite mer kraft på rapportering och "casetest.sh"

Eftersom att det inte är ett universalt skript så måste ske lite redigering innan jag skulle kunna använda det på egen maskin. Men inga tvivel på att det skulle fungera om jag och Jonas vill lägga tid på det igen.

Det var ett bra case men alla olika caselabbar verkade vara väldigt utmanade och det var 1 case av 9 som passade mina förkunskaper och skicklighet. Blir för nuläget bara ett case för min del och kommer lägga all tid på att försöka klara av ett VG på tentamen.

Källförteckning:

www.google.se

<https://github.com/Gerling/habib> kollar gamla kap6o5 uppgiften för tar.

<https://help.ubuntu.com/community/rsync> rsync guide.

http://reactivated.net/writing_udev_rules.html#syntax för udev-skriptet

<http://hackaday.com/2009/09/18/how-to-write-udev-rules/> för udev-skriptet