

Labb 1

Syfte

Få grundläggande kunskap om bash och hur det fungerar och konfigureras.

Tillvägagångssätt

Start filer:

Här tittade jag runt i alla startfiler. Jag noterade även att jag använder en ~/.bash_aliases vilket jag tycker ger en mer strukturerad konfiguration. Jag såg här att .bash_aliases filen exekveras i .bashrc.

Alias:

Jag valde att lägga in följande aliases:

```
# enklare sätt att skriva ls på, latmask :)
alias l='ls'

# generera ett 8 tecken långt lösenord
alias generatepw='tr -cd '[:alnum:]' < /dev/urandom | fold -w8 | head -n1'
```

Redirection:

```
ls -l 2> /dev/null
```

Ovanstående rad redirectar stderr till /dev/null. Dvs alla felmedelanden kommer inte visas.

Man kan även köra ">>", då kommer den bara att appenda stderr, stdout eller stdin på den angivna filen eller programmet. Ex:

```
ls -l 2>> /var/log/error.log
```

Variabler:

Här skapar jag några variabler:

```
lsOutput='ls'
workingDir=$PWD
```

Skillnaden mellan en miljövariabel är att den sträcker sig över hela systemet. Inte bara i det aktuella scriptet. Jag brukar se dom som konstanter även om dom inte är det, men jag hanterar dom som det. Detta förenklar det för mig då jag har viss bakgrund med att skriva i andra språk som hanterar konstanter.

GUSTAF LARSSON
LX13
EC UTBILDNING

För att skapa miljövariabler använder jag mig av kommandot **export**. Ett exempel på detta är:

```
Export hemmapp=/home/larsson/
```

Wildcards:

I kommandot **rm** använder jag wildcards flitigt. Det är ett enkelt sätt att ta bort många filer samtidigt. Ett exempel på detta är:

```
Rm ~/.tmp
```

Ett annat exempel på vart man använder wildcards är i **find**.

Frågor och uppgifter:

1:

```
ls -la [a]*[!e]
```

Alla filer som börjar med a och inte slutar med e

```
ls -la *[f-i]
```

Visar alla filer där näst sista bokstaven är **f** eller **i**.

```
ls -la ????.co
```

Visar alla filer som består av 4 följande med en punkt samt sist c eller o som sista tecken.

2: Jag skulle använda mig av **""**. Tex:

```
Ls -la a*.c
```

3: Du ”appendar” outputen(stdout) från `ls -la` till filen `/test/test`. Den kommer inte skriva över befintlig data i filen utan bara lägga till det på nya rader under den befintliga texten.

4:

```
ls / -laR 2> ls-laR-errors.txt | gzip > ls-laR.gz 2> ls-laR-errors.txt
```

Not: Jag kör `2>` till `errors.txt` 2 gånger för att ta eventuella felmeddelanden från både `ls` och `gzip`.

5: Env variabeln `$PATH` innehåller sökvägar till mappar där alla binära och körbara filer ligger. För att lägga till en till sökväg gör jag följande:

```
PATH=$PATH:/home/larsson/bin/
```

Notera att det är mycket viktigt att `$PATH` skrivs med när man lägger till en extra sökväg. Skulle jag inte göra detta skulle alla befintliga sökvägar försvinna. För att göra så att det alltid läggs till lägger jag till raden i min `.bash_profile`.

GUSTAF LARSSON
LX13
EC UTBILDNING

6: Den startar programmet **top** vid utloggning.

7: Dom används för att ”spara” mappar i minnet så att man lätt kan komma tillbaka dit. **Dirs** visar alla som är sparade i listan. **Popd** tar bort mappar i listan och byter till den mappen. **Pushd** sparar ner specifierad eller nuvarande mapp i listan.

Exempel:

```
larsson@laptop:~/temp$ ls
dir1 dir2 dir3
larsson@laptop:~/temp$ dirs
~/temp ~/temp/dir3 ~/temp/dir2 ~/temp/dir1
larsson@laptop:~/temp$ popd
~/temp/dir3 ~/temp/dir2 ~/temp/dir1
larsson@laptop:~/temp/dir3$ popd
~/temp/dir2 ~/temp/dir1
larsson@laptop:~/temp/dir2$ popd
~/temp/dir1
larsson@laptop:~/temp/dir1$ dirs
~/temp/dir1
larsson@laptop:~/temp/dir1$
```

8: Ett null kommando (jag antar att du syftar på /dev/null) är en fil/device som inte returnerar något. Detta är värdefullt när man scriptat, tex när man vill rensa bort data i en variabel eller vill skicka in en output dit för att det bara ska ”försvinna”.

Reflektion

Jag tycker labben va enkel men ändå kul. Jag har aldrig testat popd/pushd/dirs innan och det va ganska intressant. Men jag älskar bash, det är alltid kul att upptäcka nya saker där och att få vara kreativ. Jag ser nog mer fram emot kommande labbar där jag kan leka runt lite och sjunka in i lite kod.