

UMEÅ UNIVERSITET

November 10, 2014

Institutionen för Tillämpad fysik och elektronik

Laborationsrapport

Laboration 1

Script och webbprogrammering 7,5 hp

Scriptprogrammering

Namn Christer Jakobsson

E-mail dv12cjn@cs.umu.se

Handledare

Ola Ågren, Kalle Prorok

Contents

1	Del 1: Analys av befintligt script	2
2	Del 2: Enklare uppgift	2
2.0.1	Users	2
2.0.2	Groups	2
2.0.3	Användare inloggade just nu	3
3	Del 3: Något mer tillkrånglad uppgift	3
3.0.4	Användarhandledning	4
4	Källkod	4
4.0.5	Del 2: Enklare uppgift	4
4.0.6	Del 3: Något mer tillkrånglad uppgift	5

1 Del 1: Analys av befintligt script

Denna fil genererar en sträng som är av längd 8 och har slumpade tecken ifrån hela alfabetet, stora som små bokstäver. Scriptet initierar en variabel (*MATRIX*) som innehåller alla tecken i alfabetet i stora och små bokstäver. och loopar 8 gånger och plockar en random bokstav ifrån *MATRIX* och lägger till den i slutet på en variabel som heter *PASS*. Därefter så skrivs variabeln *PASS*:s data ut.

I och med att variabeln som skrivs ut på skärmen som resultat heter *PASS* så gissar jag att detta program är tänkt att skapa ett slumpgenererat lösenord och namnet torde stå för **new password** (np).

2 Del 2: Enklare uppgift

2.0.1 Users

För att ta reda på vilka användare som finns på datorn så använder jag mig av filen *passwd* som innehåller information om varje användare, filen ligger i */etc/passwd*. Så genom att köra *cat /etc/passwd* och sedan pipa vidare det till *cut -d -f1* som tar bort all annan data i filen förutom användarnamnen, så ger detta utdata med användarnamn i rader. Därefter så pipas detta vidare till *pr -s' ' -5 -l1 -t* som gör så att, ifrån det varit ett användarnamn per rad till att det är 5 användarnamn per rad. Därefter pipas detta vidare till *awk 'printf "%10s %10s %10s %10s %10s\n", \$1, \$2, \$3, \$4, \$5'* som skriver till standard out med 10 som mellanrum emellan varje användarnamn, detta skrivs sedan ut på standard out.

Kommandot i sin helhet:

```
cat /etc/passwd | cut -d: -f1 | pr -s' ' -5 -l1 -t | awk 'printf "%10s
%10s %10s %10s %10s\n", $1, $2, $3, $4, $5'
```

Ex:

root	daemon	bin	sys	sync
games	man	lp	mail	news
uucp	proxy	www-data	backup	list
irc	gnats	nobody	libuuid	syslog
messagebus	usbmux	dnsmasq	avahi-autoipd	kernoops
rtdkit	saned	whoopsie	speech-dispatcher	avahi
lightdm	colord	hplip	pulse	shinowa

2.0.2 Groups

Genom att använda kommandot *groups* så skrivs grupperna som finns på datorn ut på standard out.

Ex:

```
shinowa adm cdrom sudo dip plugdev lpadmin sambashare
```

2.0.3 Användare inloggade just nu

Genom kommandot *users* så visas de användare som är inloggade för tillfället ut på standard out.

Ex:

```
shinowa shinowa
```

3 Del 3: Något mer tillkrånglad uppgift

För att visa biblioteksstrukturen från en viss startaddress har jag gjort att användaren får skriva startadressen som programmets första argument. Sedan så använder jag kommandot *find \$1* där *\$1* är scriptets första argument för att skriva ut hela strukturen. Ex:

```
.  
./np  
./usergroup  
./libscript  
./Rapport  
./Rapport/Rapport.tex.backup  
./Rapport/Rapport.aux  
./Rapport/Rapport.pdf  
./Rapport/Rapport.tex  
./Rapport/.Rapport.tex.kate-swp  
./Rapport/Rapport.log
```

Sedan för att formatera utdatan så den liknar datan ifrån *tree* så pipar jag vidare till

```
\emph{sed -e "s;\$pwd;\char'\.\.;g;s;[~/]*\;/;|_.;g;s;_|; |;g"}
```

som kommer att strukturera varje rad så att man kan se hur djupt ner filen eller mappen är. För varje / så kommer namnet på filen att ha ett |__ extra framför sig och om filen är en mapp så kommer det att vara ett | under namnet som symboliserar att filerna under ligger i den mappen.

Ex:

```
.
|--np
|--usergroup
|--libscript
|--Rapport
| |--Rapport.tex.backup
| |--Rapport.aux
| |--Rapport.pdf
| |--Rapport.tex
| |--Rapport.log
```

Detta liknar tree till utseendet men använder sig bara av standardprogrammen *find* och *sed*.

3.0.4 Användarhandledning

Programmet heter *libscript* och för att köra programmet så öppnar man en terminal i den katalog som scriptet ligger i och skriver *./libscript pathToList*. *libscript* tar ett argument som är adressen för den katalog som det ska visa strukturen för.

Ex:

```
./libscript /home/shinowa/Dropbox/scriptprog/lab1
.
|--usergroup
|--libscript
|--Rapport
| |--Rapport.tex.backup
| |--Rapport.aux
| |--Rapport.toc
| |--Rapport.pdf
| |--Rapport.tex
| |--.Rapport.tex.kate-swp
| |--Rapport.log
```

4 Källkod

här följer källkoden för de två uppgifterna där script skulle skapas.

4.0.5 Del 2: Enklare uppgift

```
1 #!/bin/bash
2
```

```
3 printf "\n-----Users-----\n"
4 cat /etc/passwd | cut -d: -f1 | pr -s' ' -5 -l1 -t | awk
    '{printf "%10s %10s %10s %10s %10s\n", $1, $2, $3
    , $4, $5}'
5 printf "\n-----End-----\n"
6
7 printf "\n-----Groups-----\n"
8 "
8 groups
9
10 printf "\n-----Users logged in on computer
    -----\n"
11 users
```

4.0.6 Del 3: Något mer tillkrånglad uppgift

```
1 #!/bin/bash
2
3 if [ "$#" -ne 1 ];
4 then
5     printf "$0 takes a directory to list as argument,
        exiting...\n"
6     exit 1
7 fi
8
9 find $1 | sed -e "s;$1;\. ;g;s;[^\/*\/*];|_|g;s;_|;|;g"
```