

UMEÅ UNIVERSITET

November 13, 2014

Institutionen för Tillämpad fysik och elektronik

Laborationsrapport

Laboration 2

# Script och webbprogrammering 7,5 hp

Scriptprogrammering

**Namn** Christer Jakobsson

**E-mail** dv12cjn@cs.umu.se

**Handledare**

Ola Ågren, Kalle Prorok

## Contents

<b>1</b>	<b>Del 1: Analys av befintligt script</b>	<b>2</b>
<b>2</b>	<b>Del 2: awk-uppgift.</b>	<b>2</b>
2.1	Källkod . . . . .	2
<b>3</b>	<b>Del 3: perl-uppgift</b>	<b>2</b>
3.1	Användarhandledning . . . . .	2
3.2	Källkod . . . . .	3

## 1 Del 1: Analys av befintligt script

Detta script gör om utdatan till att beroende på hur djupt ner i katalogstrukturen som något ligger så får dess utskrift ifrån ls ett mellanslag framför sig. Om ls listar katalogerna i */home/användare* så kommer sed att göra så all utskrift ifrån ls rörande denna katalog kommer ha tre mellanslag framför sig.

När jag kör ls -lR och pipar detta till scriptet så kommer ls att rekursivt lista alla undermappar, och scriptet kommer att lägga till mellanslag så man enklare kan se katalogstrukturen.

Eftersom att programmet ändrar utskriften på katalogstrukturen så att man kan se hur trädstrukturen ser ut så skulle scriptet kunna heta *tabulateLsOutput*.

## 2 Del 2: awk-uppgift.

Sed scriptet `sed -e '/:$/d' /etc/group | cut -d: -f1` skriver ut de rader där det finns ett värde efter sista avgränsaren. Formatet på filen */etc/group* är: `Group name:Password:Group ID:Group List`. Så de rader där *Group List* innehåller ett värde är de som scriptet pipar vidare till `cut -d: -f1` som tar bort alla värden i raden och lämnar kvar *Group name* som skrivs ut.

Mitt awk program fungerar så att jag säger att avgränsaren ska vara `:` så awk kommer att dela upp raden i fyra stycken delar, och om del fyra inte är tom så ska raden skrivas ut.

### 2.1 Källkod

```
1 awk -F ':' '{if ($4 != "") print $1}' /etc/group
```

## 3 Del 3: perl-uppgift

Detta perl script ska visa biblioteksstrukturen för en viss startaddress. Det får inte använda sig av några standardprogram.

### 3.1 Användarhandledning

För att köra scriptet så öppnar man en terminal i den katalog som man har scriptet och skriver `./[scriptnamn].pl [pathToSearch]`. där *scriptnamn* är det namn man har gett scriptet , och *pathToSearch* är den sökväg man vill visa biblioteksstrukturen på. Programmet kan behövas göra exekverbart och detta görs genom att skriva `chmod 777 [scriptnamn].pl`

Exempelkörning:

```
/home/shinowa/Dropbox/scriptprog//lab3
```

```
/home/shinowa/Dropbox/scriptprog//lab3/awkuppgift~  
/home/shinowa/Dropbox/scriptprog//lab2  
/home/shinowa/Dropbox/scriptprog//lab2/libscript.pl~  
/home/shinowa/Dropbox/scriptprog//lab2/libscript~  
/home/shinowa/Dropbox/scriptprog//lab2/awkuppgift  
/home/shinowa/Dropbox/scriptprog//lab2/namn~  
/home/shinowa/Dropbox/scriptprog//lab2/Rapport  
/home/shinowa/Dropbox/scriptprog//lab2/Rapport/Rapport.tex.backup  
/home/shinowa/Dropbox/scriptprog//lab2/Rapport/Rapport.aux  
/home/shinowa/Dropbox/scriptprog//lab2/Rapport/Rapport.toc  
/home/shinowa/Dropbox/scriptprog//lab2/Rapport/Rapport.pdf  
/home/shinowa/Dropbox/scriptprog//lab2/Rapport/Rapport.tex  
/home/shinowa/Dropbox/scriptprog//lab2/Rapport/Rapport.log  
/home/shinowa/Dropbox/scriptprog//lab2/libscript.pl  
/home/shinowa/Dropbox/scriptprog//lab1  
/home/shinowa/Dropbox/scriptprog//lab1/Rapport.pdf  
/home/shinowa/Dropbox/scriptprog//lab1/usergroup  
/home/shinowa/Dropbox/scriptprog//lab1/libscript  
/home/shinowa/Dropbox/scriptprog//lab1/Rapport  
/home/shinowa/Dropbox/scriptprog//lab1/Rapport/Rapport.tex.backup  
/home/shinowa/Dropbox/scriptprog//lab1/Rapport/Rapport.aux  
/home/shinowa/Dropbox/scriptprog//lab1/Rapport/Rapport.toc  
/home/shinowa/Dropbox/scriptprog//lab1/Rapport/Rapport.pdf  
/home/shinowa/Dropbox/scriptprog//lab1/Rapport/Rapport.tex  
/home/shinowa/Dropbox/scriptprog//lab1/Rapport/Rapport.log
```

### 3.2 Källkod

```
1  #!/usr/bin/perl  
2  
3  use strict;  
4  use warnings;  
5  
6  my $dir = $ARGV[0];  
7  
8  traverse($dir);  
9  
10     sub traverse {  
11         my ($thing) = @_;  
12  
13         return if not -d $thing;  
14  
15         opendir(my $DIR, $thing) or die $!;  
16         while (my $file = readdir($DIR)) {  
17  
18             # A file test to check that it is a directory  
19             # Use -f to test for a file  
20  
21             next if $file eq '.' or $file eq '..';  
22             print "$thing/$file\n";
```

```
23         traverse("$thing/$file");
24     }
25
26     closedir($DIR);
27 }
28 exit 0;
```