

Laboration 2 — mstatus

2017-08-30

Innehåll

1	Introduktion	
2	Syfte	
3	Formalia	
3.1	Handledning
4	Bakgrund	
5	Syfte	
6	Deluppgift 1	
6.1	Tips och saker att tänka på
6.1.1	Notera
7	Deluppgift 2	
7.1	Tips
8	Krav på hur inlämningen ska ske	

1 Introduktion

I denna laboration så kommer ni skapa ett skript-program i **Bourne Again Shell** (**bash**) samt skriva en tillhörande **man**-sida. Det är **inte** tillåtet att använda andra shell/skriptspråk som **csh**, **tcsh**, **ksh**, **Perl**, **php**, **Python**, etc. för att lösa uppgiften.

2 Syfte

Syftet med laborationen är att visa att ni kan skriva enklare skript-program, samt demonstrera att ni är bekant med formatet på en **man**-sida genom att skriva en sådan själv.

3 Formalia

Notera att laborationen *måste* lösas *enskilt*. Läs [riktlinjer för labgenomförande](#) för att ha en tydlig bild om vad det innebär att jobba enskilt och notera konsekvenserna som följer av att strida mot dessa regler.

Se avsnittet “[Krav på hur inlämningen ska ske](#)” för mer information om inlämningen.

3.1 Handledning

Skicka era handledningsfrågor till 5dv088-handl@cs.umu.se. Som referensmaterial för att formulera tydliga frågor läs då [denna text](#) och även [denna](#). När ni formulerar era frågor på det sättet kommer ni finna att ni löser problemen själva.

4 Bakgrund

På system som stöder kommandot **finger** så är innehållet i en användares **.plan**-fil det som visas när en annan användare exekverar **finger** på dem. Den ursprungliga tanken var att man, genom att anropa **finger** på en annan användare, kunde få reda på var denne var (och som namnet antyder) vad denne hade för planer.

Bolaget id Software hade under en tid, bland annat då de utvecklade Doom och Quake, sina **.plan**-filer offentliga vilket innebar att allmänheten kunde följa deras utveckling. **.plan**-filer var dåtidens bloggar.

Utdatan från **finger** kan se ut ungefär såhär,

```
ath@lina% finger -l ath
Name: Tomas Halvarsson
Plan:
I xlocked hoth at Thu, Sep 23 17:48:14 1999, leaving the following message:
Gone for today, maybe back tomorrow sometime...
If you want to contact me, send an e-mail to ath@cs.umu.se.

Hey! Stop touching me!
Project:
Work until I die? No ##$@!! way!

ath      Tomas Halvarsson      00:25      4 peppar      CS
```

I denna laboration kommer vi att skriva ett skript för att automatiskt uppdatera innehållet som följer rubriken “**Plan:**”

Observera: På institutionen så måste ni anropa **finger -l** för att få se rubriken **Plan** och dess innehåll.

Notera även att **.plan**-filen måste vara läsbara för alla. Kör kommandot **chmod 644 .plan** om ni vill försäkra er om att filen är läsbar.

5 Syfte

Syftet med uppgiften är att ni ska demonstrera en bekantskap med skript och visa att ni kan kombinera standardprogram i UNIX, såsom `date`, `head`, `tail`, `uname`, etc.

6 Deluppgift 1

Ni ska skriva ett shellskript som heter `mstatus`. Syntaxen för skriptet är

```
mstatus {login|logout|xlock|unlock} [meddelande]
```

`mstatus` kräver alltså något av orden `login`, `logout`, `xlock` eller `unlock` som första argument och sedan ett valfritt meddelande.

Det ni skriver till er `.plan`-fil ska vara,

- Datum på formatet “år-månad-dag”, till exempel 99-08-19.
- Tiden ska vara på formatet “timme:minut:sekund”, till exempel 16:00:00.
- Namnet på den maskin ni loggade in/ut på (eller låste/läste upp)
- Enlogg över de sista 5 ändringarna i er `.plan`-fil.

Det kan se ut på följande vis,

```
99-08-19 16:00:00: logged in on spik
99-08-19 16:00:47: xlocked spik with message "Back in a sec"
99-08-19 16:08:32: un-xlocked spik
99-08-19 17:04:13: logged out from spik with message "I'm tired!"
99-08-20 08:47:12: logged in on hammare
```

Ni ska även hantera scenariot att det inte finns mer än 5 saker i loggen.

6.1 Tips och saker att tänka på

- Titta på exemplen och övningarna från UNIX-introt för att komma igång med shellskriptsprogrammering.
- Om ni vill använda externa datafiler för mer statisk information som t.ex. er adress, måste den filen vara läsbar för oss (och fulla sökvägen till den måste anges i ert skript) då vi testkör. Vi kommer att kopiera ert skript till en katalog där vi har skrivrättigheter, så att vi kan skapa `.plan`-filer.
- Skriptet ska enbart skriva till filen `.plan` i aktuell katalog. Ni ska alltså inte ge någon fullständig sökväg till `.plan`-filen i ert skript.
- Filen `.plan` ska inte modifieras om skriptet startas med felaktiga parametrar.

- Använd inga temporära filer i din lösning, det vill säga skriv inte till någon annan fil än till just `.plan`.

6.1.1 Notera

- Om du gör *mer* än **det nödvändiga**, gör **inte** för mycket i det script du lämnar in. Vi vill inte behöva leta igenom en 1000-raders `.plan`-fil för att försöka se om du löst uppgiften eller ej. För er egen skull så kan ni göra en löjligt överarbetad lösning som du använder i “verkliga livet”. (`finger -l mr` är ett exempel på en statisk variant)

7 Deluppgift 2

I den här deluppgiften så ska du skriva en tillhörande `man`-sida för ditt program. Den bör åtminstone innefatta rubrikerna `NAME`, `SYNOPSIS`, `DESCRIPTION`, `AUTHOR` och `OPTIONS` om det är befogat.

7.1 Tips

Exempel på andra `man`-sidor finns i mappen `/usr/share/man`.

För att läsa dokumentation om hur en skriver `man`-sidor kan du skriva `man -S7 man` i din terminal.

För att hitta din egen `man`-sida bör du ändra i din `.tcshrc`-fil¹ i din hemkatalog. Öppna denna fil och lägg till följande:

```
setenv MANPATH `manpath`:"$HOME/sökväg/till/din/mstatus/mapp"
```

Så, om din `mstatus`-fil ligger i `edu/sysprog/lab2` så skriver du

```
setenv MANPATH `manpath`:"$HOME/edu/sysprog/lab2"
```

För att komma åt din `man`-sida måste du skapa en katalog som heter `man1` i mappen där du har ditt skript. I den nyskapade katalogen (`man1`) lägger du din `man`-fil. Den filen **måste** heta `mstatus.1`.² Kom även ihåg att starta om er terminal för att den nya inställningen ska tillämpas.

Undvik att använda svenska tecken i `man`-sidan eftersom vissa versioner av `man` inte alls klarar av detta utan coredumpar. Skriv på engelska.

Om ni vill använda ett punkt-tecken i början av ett stycke så råkar `man` tolka det som en rubrik. Ni kan istället för punkten skriva `\[char46]` ([Escaping dots in Groff](#))

¹Om den filen inte redan finns måste du skapa den

²Under Linux kan man också ge ‘`man`’-filen som argument direkt till ‘`man`’, till exempel ‘`man -l mstatus.1`’

8 Krav på hur inlämningen ska ske

Vi använder labres för inlämning och där kommer det även att finnas en automattestning så att ni kan testa er kod även innan inlämning. Observera att vi tittar på mer än bara testkörningarna i labres, dvs bara för att testerna är OK så kan uppgiften behöva kompletteras. Du behöver **inte** skriva en labbrapport.