

Uppgift 2

Script

1. Skriv det sed-kommando som byter ut alla förekomster av "Linux programmerare" till Linuxprogrammerare i denna fil 1

sed -i -e 's/Linux programmerare/Linuxprogrammerare/g' text1.txt

Värt att notera är att ifall man bara vill se vad som händer genom sed kommandot och inte ändra filen så kan man ta bort -i och -e.

```
linus@debian:~/github_space/umu_linux/LINUX_2/uppgift2$ cat text1.txt
På kurserna Linux som utvecklingsmiljö 1 och 2 som riktar sig till Linux programmerare får man lära sig flera olika delområ-
det:
Del får man kunskap i administration och arbetssätt i Linux som varje Linux programmerare har nytta av i sitt dagliga jobb.
Dels får man testa olika verktyg som underlättar för en Linux programmerare såsom gdb, trace, diff mm.
Med dessa kurser bör man få med sig de förutsättningar som krävs för att gå vidare på mer specifika kurser som Linux i inby-
ggda system och Drivrutiner i Linux.

linus@debian:~/github_space/umu_linux/LINUX_2/uppgift2$ sed -i -e 's/Linux programmerare/Linuxprogrammerare/g' text1.txt
linus@debian:~/github_space/umu_linux/LINUX_2/uppgift2$ cat text1.txt
På kurserna Linux som utvecklingsmiljö 1 och 2 som riktar sig till Linuxprogrammerare får man lära sig flera olika delområ-
det:
Del får man kunskap i administration och arbetssätt i Linux som varje Linuxprogrammerare har nytta av i sitt dagliga jobb.
Dels får man testa olika verktyg som underlättar för en Linuxprogrammerare såsom gdb, trace, diff mm.
Med dessa kurser bör man få med sig de förutsättningar som krävs för att gå vidare på mer specifika kurser som Linux i inby-
ggda system och Drivrutiner i Linux.
```

2. Skriv det sed-kommando som ersätter alla personnummer med xxxxxx-xxxx i denna fil 2

sed -i -e 's/[0-9]\{6\}-[0-9]\{4\}/xxxxxx-xxxx/g' text2.txt

3. Skriv det sed-kommando som tar bort alla personnummer i ovan fil 2.

Här gör jag ett antagande att det ska vara originalfilen och inte ändrade filen.

sed -i -e 's/[0-9]\{6\}-[0-9]\{4\}//g' text2.txt

```
linus@debian:~/github_space/umu_linux/LINUX_2/uppgift2$ cat text2.txt
Björne Lindberg 555555-6677
John Berge 445566-7788
Sven Rönnbäck 223344-5566
Lisa Studentsson 665544-3322

linus@debian:~/github_space/umu_linux/LINUX_2/uppgift2$ sed -i -e 's/[0-9]\{6\}-[0-9]\{4\}//g' text2.txt
linus@debian:~/github_space/umu_linux/LINUX_2/uppgift2$ cat text2.txt
Björne Lindberg
John Berge
Sven Rönnbäck
Lisa Studentsson
```

Linus Frisk
010809-2933
4/3/2025

4. Skriv det sed-kommando som för säkerhets skull tar bort alla rader i ovan fil 2 som fortfarande innehåller personnummer (eller vilka valfria nummer som helst).

Tänker att jag tar originalfilen och tar bort en persons personnummer så man kan se en verkan på uppgiften. Alltså att 3 personers rader ska helt försvinna.

sed -i -e '/[0-9]\{6\}-[0-9]\{4\}/d' text2.txt

```
linus@debian:~/github_space/umu_linux/LINUX_2/uppgift2$ cat text2.txt
Björne Lindberg 555555-6677
John Berge 445566-7788
Sven rönnbäck
Lisa Studentsson 665544-3322

linus@debian:~/github_space/umu_linux/LINUX_2/uppgift2$ sed -i -e '/[0-9]\{6\}-[0-9]\{4\}/d' text2.txt
linus@debian:~/github_space/umu_linux/LINUX_2/uppgift2$ cat text2.txt
Sven rönnbäck

linus@debian:~/github_space/umu_linux/LINUX_2/uppgift2$
```

5. Skriv det awk-kommando som skriver ut personnumren i ovan fil 2.

awk '{for (i=1; i<=NF; i++) if (\$i ~ /^[0-9]\{6\}-[0-9]\{4\}/) print \$i}' text2.txt

NF: Number of fields.

Vi går igenom alla rader som finns där om pågående rad har siffrorna 0-9 i en följd av 6 siffror samt ett minustecken och siffrorna 0-9 i följd av 4 siffror. Just dessa siffror sparar vi och printar ut dessa.

```
linus@debian:~/github_space/umu_linux/LINUX_2/uppgift2$ awk '{for (i=1; i<=NF; i++) if ($i ~ /^[0-9]\{6\}-[0-9]\{4\}/) print $i}' text2.txt
555555-6677
445566-7788
223344-5566
665544-3322
```

Alternativt kan vi använda **awk '{print \$3}' text2.txt** eftersom vi vet att alla personnummer ligger på kolumn 3 men detta är inte giltigt för andra typer av textfiler där personnummrena kan sitta på en annan kolumn.

```
linus@debian:~/github_space/umu_linux/LINUX_2/uppgift2$ awk '{print $3}' text2.txt
555555-6677
445566-7788
223344-5566
665544-3322
```

7. Ladda ner och kör detta script. Försök avbryta med ctrl-C. Efter några försök kommer det att lyckas. Vad gör scriptet och varför vill man ibland kunna använda trap?

Scriptet är en oändlig loop som räknar upp med "1" varje sekund. Det lyssnar också efter en signal, vilket är när jag kör **ctrl+c** (SIGINT).

Själva increment funktionen skriver ut meddelandet om signalen togs emot, den ökar också värdet med 500 (på x) när man kör **ctrl+c**, samt att skulle x vara större än 2000 så avslutar vi programmet.

Trap kan vara bra att använda när man stänger ner ett script som skapat en temporär fil, där man senare tar bort den temporära filen genom trap precis innan man stänger av. Andra kända användningsområden av trap kan vara **error handling** och **graceful termination**.