Uppgift 4, Script

Termin: HT20

Kurs: Linux som utvecklingsmiljö

Student: Fredrik Strömbäck, 19920823-1556

a:

```
#!/bin/bash

echo "$1"

exit
```

b:

```
#!/bin/bash

TODAYSDATE=$(date +%d/%m/%Y)

echo "Date of today: $TODAYSDATE"

POSIXTIME=$(date +"%s")

echo "POSIX time: $POSIXTIME"

exit
```

c:

```
#!/bin/bash

POSIXTIME=$(date +"%s")

echo "$POSIXTIME" >> datefile

exit
```

d:

```
#!/bin/bash

CURRENTPOSIXTIME=$(date +"%s")

OLDPOSIXTIME=$(cat datefile)

let diff=$CURRENTPOSIXTIME-$OLDPOSIXTIME

echo $diff

exit
```

e:

```
#!/bin/bash

function my_write () {
echo "$1 $2"
}

my_write "Hello world"
text="Hej hopp"
my_write "$text"

exit
```

f:

```
#!/bin/bash

COUNTFILE=countFile

if [ ! -e "$COUNTFILE" ]; then
touch "$COUNTFILE"
echo 0 > "$COUNTFILE"
fi

VALUE=$(cat $COUNTFILE)

function count() {
if [ "$1" == "in" ]; then
VALUE=$((VALUE += 1))
echo "$VALUE" > $COUNTFILE
elif [ "$1" == "ut" ] && [ $VALUE -gt 0 ]; then
VALUE=$((VALUE -= 1))

echo "$VALUE" > $COUNTFILE
fi
}

function print() {
if [ "$VALUE" -ge 1 ]; then
echo "Inloggad"
else
echo "Utloggad"
fi
}

count "$1"
print

exit
```

g:
Se script "timetrack.sh".

h:
Se script "checkuser.sh".

g:

```
#!/bin/bash

DIR="3D"
FILE="Makefile"
C_FILES=0

if [ -d "$DIR" ]; then
cd "$DIR" || exit 1
C_FILES=$(find ./*.c | wc -l)
else
echo "No directory named '$DIR' found."
exit 1
fi

if [ ! -f "$FILE" ]; then
echo "No $FILE found."
exit 1
fi

function compileFiles(){
local successCounter
successCounter=0

for f in ./*.c; do
if gcc -c "$f"; then
(( successCounter++ ))
fi
done

if [ "$successCounter" -eq "$C_FILES" ]; then
echo "All $C_FILES files are compilable."
else
echo "$(( "$C_FILES"-"$successCounter" )) file(s) failed."
fi
}

function runMakeAll() {
if make all ; then
echo "'make all' ran successfully!"
else
echo "'make all' encountered an error"
fi
}
```

```
function runMakeClean() {
local oCounter
oCounter=0
if make clean ; then
echo "'make clean' ran successfully!"
for f in ./*.o; do
if [ -f "$f" ]; then
(( oCounter++ ))
fi
done

if [ "$oCounter" -eq 0 ]; then
echo "All .o-files were removed"
else
echo "Still $oCounter .o-files left"
fi
else
echo "'make clean' encountered an error"
fi
}

compileFiles
runMakeAll
runMakeClean

exit
```

Rapport:

Det svåraste med uppgifterna är syntaxen för bash. Det är svårt då jag är mer van vid t.ex. Javascript. Självklart har mina programmeringskunsakper hjälpt mig att ta in och fundera ut lösningarna på problemen men syntaxen skiljer sig en hel del från språken jag har vana i. Detta var väl i stort sätt det enda stora problem jag kände att jag stötte på.

Något annat som var nytt för mig var ju detta med med att köra kommandon i terminalen. Alltså att man kunde köra t.ex. "cat", "find" eller "date" på ett mycket simplare sätt än jag stött på tidigare. Detta fick mig ju att tänka i banorna om hur mycket jag kan automatisera på, i mina studier och mina projekt. Men detta blev även en svårighet för att då behövde man hitta rätt kommandon och program för att få ut rätt data men även syntaxen för att få dem att exekveras där och på sättet man ville. Så även här blev syntaxen ett problem.

Den roligaste uppgiften var helt klart "timetrack" då jag ser att jag kan verkligen har användning för den och den var också den första uppgiften där jag kände att jag verkligen fick använda mig av funktioner på ett sätt jag är mer van vid.