

Ukesoppgaver 6 – C og assembly-programmering; bash scripting

1. **(Oblig)** skriver c-koden (høynivåkode) med «jed Hello.c» og etter det taster «gcc Hello.c» slik at den kompilerer til maskinkode:

Kan rename fila til hello: i stedet for a.out til noe annet med -o → gcc Hello.c -o hello

```
[s364520@data2500:~$ jed Hello.c
[s364520@data2500:~$ gcc Hello.c
[s364520@data2500:~$ cat Hello.c
#include <stdio.h>

void main()
{
    printf("Hello World!\n");
}
[s364520@data2500:~$ ./a.out
Hello World!
[s364520@data2500:~$ █
```

2. **(Oblig)** kompiler sum.c og sumMain.c med
gcc sumMain.c as.a -o sum

```
[s364520@data2500:~$ gcc sumMain.c as.s -o sum
as.s: Assembler messages:
as.s: Warning: end of file in comment; newline inserted
[s364520@data2500:~$ ./sum
Sum = 6
[s364520@data2500:~$ █
```

Endret \$3 til \$4 i assemblykoden:

```
[s364520@data2500:~$ ./sum
Sum = 10
[s364520@data2500:~$ █
```

Det som er likt mellom koden i as.s og maskinkoden i oppgave 5 (fra uke 5) er at det går gjennom en ekstra runde i løkken, og økningen er med i og ikke 2, slik som i maskinkoden.

3. Kopiert as.s → as2.s og sum.c → sum2.c også endret slik at følgende kode skjer:

```
S = 0;
for(i=1; i < 3; i++)
{
    S = S + 2;
}
```

```
sum:                # Standard

mov    $2, %rcx
mov    $1, %rdx
mov    $0, %rbx
mov    $0, %rax
```

Bildet over: første setning vil si så lenge i er mindre enn 3 skal for løkka kjøre. 2 setning velger hvor mye i øker med hver runde.

I løkka:

```
# LØKKE
start: # label
add    %rdx, %rbx
add    %rcx, %rax
cmp    %rcx, %rbx
jne    start
```

4. HJELP

5. HJELPP

6. HJELP

```
as2.s:13: Error: no such instruction: `ax, bx, cx, dx'
[s364520@data2500:~$ gcc sum2.c as2.s -o sum2
as2.s: Assembler messages:
as2.s:13: Error: no such instruction: `ax, bx, cx, dx'
s364520@data2500:~$
```

7. (Oblig)

```
[s264@os5264:~]$ grep $(whoami) /etc/passwd
s264:x:1000:998::/home/s264:/bin/bash
[s264@os5264:~]$ grep "$(id -un)" /etc/passwd
s264:x:1000:998::/home/s264:/bin/bash
[s264@os5264:~]$ grep "$(logname)" /etc/passwd
s264:x:1000:998::/home/s264:/bin/bash
[s264@os5264:~]$ c
```

8. (Oblig) shell-script som skriver ut verdien av (den globale) variabelen SHELL hvis den er satt og gir melding om at den er udefinert hvis den ikke er satt:

```
[s364520@data2500:~]$ cat SHELL.sh
#!/bin/bash

var=$1

if [ $var = "SHELL" ];
then
    echo $var er en global variabel
else
    echo $var er ikke en global variabel
fi

[s364520@data2500:~]$ ./SHELL.sh
./SHELL.sh: line 5: [: =: unary operator expected
er ikke en global variabel
[s364520@data2500:~]$ ./SHELL.sh SHELL
SHELL er en global variabel
```

9. (Oblig) Scriptet under:

```
[s364520@data2500:~]$ cat publiser.sh
#!/bin/bash

#setter rettigheter til alle filer i ~/www, kun eier skal lese og skrive, andre lese
#samme til alle filer i ~/www/bilder
#hvis de ikke eksisterer skal de opprettes, alle rettigheter for eier, kun lese og kjøre for andre
#om det finnes andre kataloger i ~/ww skal disse ikke endre rettigheter, ingenting under heller

#sjekker om ~/www eksisterer, hvis ikke oppretter og endre rettigheter
if [ ! -d ~/www ];
then
    mkdir ~/www
    chmod 700 ~/www
fi

#sjekker om ~/www/bilder eksisterer, hvis ikke oppretter og endrer rettigheter
if [ ! -d ~/www/bilder ];
then
    mkdir ~/www/bilder
    chmod 700 ~/www/bilder
fi

#endrer rettigheter til alle eksisterende filer av ~/www
chmod 700 ~/www/*

#endrer rettigheter til alle eksisterende filer av ~/www/bilder/
chmod 700 ~/www/bilder/
```

Jeg utdeler først rettigheter til publiser.sh og kjører scriptet. Også er det mulig å se at alle filer under får endra rettigheter slik som opplyst i oppgaven.

```
[s364520@data2500:~]$ cd ~/www  
[s364520@data2500:~/www$ ls -l  
total 4  
drwx----- 2 s364520 20364520 4096 Feb 10 17:24 bilder  
[s364520@data2500:~/www$ cd bilder  
[s364520@data2500:~/www/bilder$ ls -l  
total 0  
-rw-r--r-- 1 s364520 20364520 0 Feb 10 17:24 du  
-rw-r--r-- 1 s364520 20364520 0 Feb 10 17:24 hallo  
-rw-r--r-- 1 s364520 20364520 0 Feb 10 17:24 hei  
s364520@data2500:~/www/bilder$
```

10. (Oblig) husk at denne koden ikke teller filer, linker eller kataloger i noen mapper over kun alt under, og under der igjen.

```
[s364520@data2500:~]$ cat count.bash  
#!/bin/bash  
  
#initialisere variabler for å telle linker, filer og kataloger  
antall_linker=0  
antall_filer=0  
antall_kataloger=0  
  
#bruker find-kommando til å finne filer, linker og kataloger  
#øker telleren til variablene  
  
# 'find . -type f' finner alle filer i nåværende katalog ('.' og underkatalog, og returnerer navnene på filene. 'wc -l' tar denne  
# listen av filnavn og teller hvor mange linjer det er, og returnerer bare antall linjer.  
antall_linker=$(find . -type f | wc -l)  
antall_filer=$(find . -type d | wc -l)  
antall_kataloger=$(find . -type l | wc -l)  
  
#skriver ut resultater:  
echo "Lenker: $antall_linker"  
echo "Filer: $antall_filer"  
echo "Kataloger: $antall_kataloger"  
  
s364520@data2500:~$ chmod 700 count.bash  
s364520@data2500:~$ ./count.bash  
Lenker: 55  
Filer: 14  
Kataloger: 0
```

11. UKENS UTFORDRING! Koden under:

```

F10 key ==> File Edit Search Buffers Windows System Help
#!/bin/bash

# '~/www' eksisterer og er en sti til mappa
www_directory=~/.www

#alle rettigheter for eier, lese- og kjøre for alle andre
chmod 755 $www_directory

#finner alle filer (type f) i mappen $www_directory ( '~/www')
#leser filnavnene inn i while løkke med if

find $www_directory -type f | while read f; do
#sjekker om filen $f har x-rettigheter (kjørerrettigheter

    #sjekker om filen $f har rettigheter
    if [ -x "$f" ]; then
        #filen har x-rettigheter, settes 744 eier har alle rett)
        chmod 744 "$f"

    else
        #hvis filen ikke har x-rettigheter, settes 644 (eier har r & w)
        chmod 644 "$f"

    fi
done

```

Under først viser jeg at jeg har laget tre filer «publiser», «test» og «script» som jeg endrer rettigheter til 700:

```

[s364520@data2500:~/www$ chmod 700 publiser test script
[s364520@data2500:~/www$ ls -l
total 4
-rw-r--r-- 1 s364520 20364520 0 Feb 10 17:32 alle
drwx----- 2 s364520 20364520 4096 Feb 10 17:24 bilder
-rw-r--r-- 1 s364520 20364520 0 Feb 10 17:32 hei
-rwx----- 1 s364520 20364520 0 Feb 11 00:10 publiser
-rw-r--r-- 1 s364520 20364520 0 Feb 10 17:32 sammen
-rwx----- 1 s364520 20364520 0 Feb 11 00:10 script
-rwx----- 1 s364520 20364520 0 Feb 11 00:10 test

```

Så kjører jeg skriptet, og det er vist at skriptet fungerer:

```

[s364520@data2500:~$ ./publiser2.sh
[s364520@data2500:~$ cd ~/.www
[s364520@data2500:~/www$ ls -l
total 4
-rw-r--r-- 1 s364520 20364520 0 Feb 10 17:32 alle
drwx----- 2 s364520 20364520 4096 Feb 10 17:24 bilder
-rw-r--r-- 1 s364520 20364520 0 Feb 10 17:32 hei
-rwxr--r-- 1 s364520 20364520 0 Feb 11 00:10 publiser
-rw-r--r-- 1 s364520 20364520 0 Feb 10 17:32 sammen
-rwxr--r-- 1 s364520 20364520 0 Feb 11 00:10 script
-rwxr--r-- 1 s364520 20364520 0 Feb 11 00:10 test

```

12. UKENS UTFORDRING!

Jeg tror det eneste som skal endres i koden fra **UKENS UTFORDRING!** er at istedet for:

```
chmod -R 755 $www_directory
```

skal det skrives:

```
chmod -R 755 $www_directory/*
```

som vil gjøre akkurat det samme på tross av hvilke rettigheter filene hadde fra før.

13. (Oblig)

Koden til skriptet filSjekk.sh

```
#!/bin/bash

#tar inn en input og lagrer som variabel
fil=$1

#er den angitt med absolutt path?
if [[ $fil =~ ^/ ]];
then
    echo "Filen $fil er angitt med absolutt path"
#er den angitt med relativt path?
elif [[ $fil =~ ^\./ ]];
then
    #sjekker om filen er en relativ sti
    if [-f $fil ];
    then
        #sjekker om filen finnes relativt til der skriptet kj?res fra
        echo "Filen $fil er angitt med relativ path"
    else
        echo "Relativ path, filen eksisterer ikke"
    fi
else
    echo "Ikke noe gyldig path"
fi
```

Eksempel på output etter kjøring av skriptet med: ‘ \$./filSjekk.sh ’

```
[s364520@data2500:~]$ ./filSjekk.sh Hello
Ikke noe gyldig path
[s364520@data2500:~]$ ./filSjekk.sh /usr/bin
Filen /usr/bin er angitt med absolutt path
s364520@data2500:~$
```

14. (Oblig) Koden:

```
#!/bin/bash

#lokker gjennom avhengig av antall input
for fil in "$@"; do
    #absolutt sti starter med "/"
    if [[ $fil == /* ]];
    #fortell at det er en absolutt sti
    then
        echo "$fil: Absolutt path"
    #relativ sti starter med "."
    elif [[ $fil == ./* ]];
    then
        #finnes filen relativt til der skriptet kjøres fra
        if [ -f "$fil" ];
        then
            #sjekk om filen er en relativ sti og at den finnes
            echo "$fil: Relativ path, filen finnes"
        else
            #skriver ut at filen er en relativ sti, men eksisterer ikke
            echo "$fil: Relativ path, filen finnes ikke"
        fi
    else
        #skriver at argumentet ikke er angitt med sti
        echo "$fil: Ikke en gyldig path"
    fi
done
```

changer mods og kjører:

```
[s364520@data2500:~]$ ./filSjekk2.sh sum.c sumMain.c /usr/bin
sum.c: Ikke en gyldig path
sumMain.c: Ikke en gyldig path
/usr/bin: Absolutt path
s364520@data2500:~$
```

15.

16.

17.

18.

19.