

Task 1

Consider the following scripts. Explain what means each line within these scripts. Explain what happens when you execute these scripts with and without input arguments.

```
#!/bin/bash
# Script 1
#
for FN in "$@"
do
chmod 0750 "$FN"
done
```

#For loop. FN = Variable.

#"\$@" is a shell variable which expands to a list of all arguments passed to the script

#chmod 0750 „\$FN“ – changes permission on a file

First number is for special privileges. In this case there are no privileges.(0)

#Second number describes the right of the owner. He has all the rights to do whatever he wants (7)

#Third number is about group rights. The group is allowed to read and execute it (5)

#Fourth number declares the rights of other users. In this case they have no rights(0)

The script takes the given arguments and tries to change the permission of the given file. If there is no file given → error

#7 – full rights

#6 – read and write

#5 – read and execute

#4 – just read

#3 – write and execute

#2 – just write

#1 – just execute

#0 – no rights

Special bit(first bit):

#4 – SUID (set user of the directory)

#3 – SGID (set group of the directory)

#2 – just Owner is allowed to delete and rename folders

#1 – nothing

```
#!/bin/bash
# Script 2
if (($# < 3))
then
printf "%b" "Error. Not enough arguments.\n" >&2
printf "%b" "usage: myscript file1 op file2\n" >&2
exit 1
elif (($# > 3))
then
printf "%b" "Error. Too many arguments.\n" >&2
printf "%b" "usage: myscript file1 op file2\n" >&2
exit 2
else
printf "%b" "Argument count correct. Proceeding...\n"
fi
```

#\$# - Number of arguments

#if command

#fi end of if command

##%b = backspace

##>&2 standard error output. >&1 would be standard output

#The script counts the number of given arguments and prints a specific message.

```
#!/bin/bash
# Script 3
INFILE=$1
OUTFILE=$2
if [-e "$INFILE"]
then
    if [-w "$OUTFILE"]
    then
        cat "$INFILE" >> "$OUTFILE"
    else
        echo "can not write to $OUTFILE"
    fi
else
    echo "can not read from $INFILE"
fi
```

```
#$1 first argument from command line
#$2 second argument from command line
#-e „File“ → TRUE if File exists
#-w exists and is writeable
```

```
#This scripts copies the content from file1 to file2 if it is possible. Otherwise it
#will give out an error message.
```