University of Antwerp
Faculty of Applied Engineering

Sten Hulsbergen - 20242689

Session 3

3.1

To make the script use `nano exercise-1`, when you are done editing the script, use `CTRL+O` to write to the file and follow up with `CTRL+X` to exit the file.

Next change permissions to be able to execute the file with `chmod u+x exercise-1` and execute the file with `./exercise-1 (number)`. The script and results are shown bellow.

```bash
#!/bin/bash

# Check if there is an argument given to the script.
# If no(!) argument($1), then exit script.
if [ ! "$1" ]; then
    echo "no argument given"
    exit 1
fi

# Check if the argument is a positive number.
# If argument($1) less than(-lt) 0, then exit script.
if [ $1 -lt 0 ]; then
    echo "number is not positive"
    exit 1
fi

# For index equal to argument($1), decrease index and echo each index until 0.
for (( i=$1; i>=0; i-- )) do
    echo "$i"
done
```

```
s0242689@3networkarchitecture-s0242689:~/session-3$ ./exercise-1
no argument given
s0242689@3networkarchitecture-s0242689:~/session-3$ ./exercise-1 -5
number is not positive
s0242689@3networkarchitecture-s0242689:~/session-3$ ./exercise-1 5
5
4
3
2
1
0
```

3.2

Create the "myFiles" folder and a number of files in it with `mkdir myFiles; touch myFiles/file{0..5}`, the results are shown with `tree` and are as followed:

```
s0242689@3networkarchitecture-s0242689:~/session-3$ mkdir myFiles; touch myFiles/file{0..5}
s0242689@3networkarchitecture-s0242689:~/session-3$ tree
.
├── exercise-1
├── exercise-2
└── myFiles
    ├── file0
    ├── file1
    ├── file2
    ├── file3
    ├── file4
    └── file5
```

By default, with `umask 0022`, the files can be read by everyone but only be written to by the user as shown bellow with `ls -l myFiles/`.

```
s0242689@3networkarchitecture-s0242689:~/session-3$ ls -l myFiles/
total 0
-rw-r--r-- 1 s0242689 s0242689 0 Oct 16 16:15 file0
-rw-r--r-- 1 s0242689 s0242689 0 Oct 16 16:15 file1
-rw-r--r-- 1 s0242689 s0242689 0 Oct 16 16:15 file2
-rw-r--r-- 1 s0242689 s0242689 0 Oct 16 16:15 file3
-rw-r--r-- 1 s0242689 s0242689 0 Oct 16 16:15 file4
-rw-r--r-- 1 s0242689 s0242689 0 Oct 16 16:15 file5
```

To make the script use `nano exercise-2`, when you are done editing the script, use `CTRL+O` to write to the file and follow up with `CTRL+X` to exit the file.

Next change permissions to be able to execute the file with `chmod u+x exercise-2` and execute the file with `./exercise-2 (directory)`. The script and results are shown bellow.

```bash
#!/bin/bash

# Check if there is an argument given to the script.
# If no(!) argument($1), then exit script.
if [ ! "$1" ]; then
    echo "no argument given"
    exit 1
fi

# Check if the argument is a directory.
# If no(!) directory(-d) argument($1), then exit script.
if [ ! -d "$1" ]; then
    echo "directory '$1' does not exist"
    exit 1
fi

# Change permissions of all the files within the given folder.
# The only permissions that are allowed are: the user can read and write.
sudo chmod -R 600 $1
echo "changed permissions of every file in '$1'"
```

```
s0242689@3networkarchitecture-s0242689:~/session-3$ ./exercise-2
no argument given
s0242689@3networkarchitecture-s0242689:~/session-3$ ./exercise-2 test
directory 'test' does not exist
s0242689@3networkarchitecture-s0242689:~/session-3$ ./exercise-2 myFiles
changed permissions of every file in 'myFiles'
s0242689@3networkarchitecture-s0242689:~/session-3$ sudo ls -l myFiles/
total 0
-rw------- 1 s0242689 s0242689 0 Oct 16 16:15 file0
-rw------- 1 s0242689 s0242689 0 Oct 16 16:15 file1
-rw------- 1 s0242689 s0242689 0 Oct 16 16:15 file2
-rw------- 1 s0242689 s0242689 0 Oct 16 16:15 file3
-rw------- 1 s0242689 s0242689 0 Oct 16 16:15 file4
-rw------- 1 s0242689 s0242689 0 Oct 16 16:15 file5
```

3.3

To make the script use `nano exercise-3`, when you are done editing the script, use `CTRL+O` to write to the file and follow up with `CTRL+X` to exit the file.

Next change permissions to be able to execute the file with `chmod u+x exercise-3` and execute the file with `./exercise-3 (bound) (bound)`. The script and results are shown bellow.

```bash
#!/bin/bash

# Check if there are arguments given to the script.
# If total of arguments($#) not equal(-ne) to 2, then exit script.
if [ $# -ne 2 ]; then
    echo "missing argument(s)"
    exit 1
fi

# Check wich bound is upper limit given to the script.
# If argument($2) greater than(-gt) argument($1), lower=$1 and upper=$2 else lower=$2
and upper=$1.
if [ $2 -gt $1 ]; then
    LOWER_BOUND=$1
    UPPER_BOUND=$2
else
    LOWER_BOUND=$2
    UPPER_BOUND=$1
fi

# For index equal to lower bound, increase index until upper bound.
for (( i=LOWER_BOUND; i<UPPER_BOUND; i++)) do
    getent passwd $i > /dev/null # Get entity by ID(getent passwd $i), output 0 when
user is found, other messages are ignored(> /dev/null).
    if [ $? -eq 0 ]; then # If output last command($?) equal(-eq) to 0, then echo
username and user ID.
        echo "$(id -nu $i): $i" # id command diplays UID, GID and groups, use
username(-un) of UID($i) to display username linked to UID.
    fi
done
```

```
s0242689@3networkarchitecture-s0242689:~/session-3$ ./exercise-3 100 2000
uuidd: 100
messagebus: 101
tcpdump: 102
sshd: 103
polkitd: 995
systemd-resolve: 996
systemd-timesync: 997
systemd-network: 998
netlab: 1000
s0242689: 1001
nbhat: 1002
rberkvens: 1003
rnietvelt: 1004
s0242689@3networkarchitecture-s0242689:~/session-3$ ./exercise-3 1000 100
uuidd: 100
messagebus: 101
tcpdump: 102
sshd: 103
polkitd: 995
systemd-resolve: 996
systemd-timesync: 997
systemd-network: 998
```