

Introduction to Computer Science Lab Report 2

Part 1

1. Purpose of

-l (long format) displays Unix file types, permissions, number of hard links, owner, group, size, last-modified date and filename

-d shows information about a symbolic link or directory, rather than about the link's target or listing the contents of a directory

-a is to list 'all'; this lists absolutely every file and folder within the directory

2. Find a command line which lists all files and subfolders of the directory /etc starting with one of the letters a, b, or c.

```
ls -d [a-c]*
```

3. Find a command line which lists all files and subfolders in the directory /etc starting with "ant", "bl", or "ho".

```
ls -l -d ant* bl* ho*
```

4. Find a command line which lists all files and subfolders in the directory /etc which do not start with one of the letters a, b, or c.

```
ls -l [!a-c]*
```

5. What do "~", ".", ".." mean?

~ means users home directory . represents the current directory one is in

. represents the current directory

.. represents the parent directory

What might be the practical use of the "." directory?

It can be used on multiple devices to provide different results.

Introduction to Computer Science Lab Report 2

Part 2

1. Creating the Subdirectories:

```
cd home/practice
mkdir lab
cd lab
touch file-1
mkdir lab-1 lab-2 lab-3
cd lab-1
touch file-11 file-12 file-13 *.hidden
cd ..
cd lab-3
mkdir lab-31 lab-32
cd lab-31
touch file-311 file-312 file-313
```

2. Delete the subfolder “lab-3” as well as all contained files and subfolders

```
rm -R lab-3
```

3. Copy all files and subfolders of “lab-1” into a new subdirectory “lab3”

```
cp -a ~/lab/lab-1/ ~/lab/lab-3
```

4. Move all files of the subfolder “lab-1” into the subfolder “lab-2”

```
mv -v ~/lab/lab-1/. * ~/lab/lab-2
```

5. Create a hard link inside of “lab-1” with the name “file-11” to the existing file “file-11” which is located in “lab-2”

```
ln ~/lab/lab-2/file-11 ~/lab/lab-1/file-11
```

Can you see which file is the original and which one is the linked file by means of the “ls” command?

No, the created file is more or less a copy of the original file and can be accessed even if the original is deleted.

Remove the original file “file-11” inside the subdirectory “lab-2” and afterwards the entire subdirectory “lab-1” with the command “rmdir lab-1”. Does this work (yes/no)? If not, why?

It doesn't work because the directory is not empty.

8. Create a symbolic link (“soft link”) inside “lab-1” called “file-12” to the existing file “file-12” which is located in the subfolder “lab-2”

```
ln -s ~/lab/lab-2/file-12 ~/lab/lab-1/file-12
```

Introduction to Computer Science Lab Report 2

Can you see which file is the original and which one is the linked file by means of the “ls” command?

Yes, the created file is like a shortcut in Windows. It works only when the original is on the system and is unusable once the original is deleted. There is also a visible “arrow” in the top right corner of the file icon showing that it is a “shortcut” to another file.

Part 3

9. Extend the contents of the PATH environment variable so that you can call the script without typing the full pathname, simply by typing its filename

```
PATH=/home/practice:$PATH
```

10. How can you register this modification permanently, so that it still exists when you log in next time?

```
PATH="$PATH:/home/practice/" >> ~/.bashrc
```

11. Expand your script “script.bash” of the task before in that way, that it displays the following two lines:

The current time stamp is:

6:23 pm and 18 seconds

```
#!/bin/bash
echo "The current time stamp is:"
echo $(date "+ %l:%M %P and %S seconds")
echo "You are logged in as user: $USER"
echo "Your home directory is: $HOME"
echo "You currently use the shell: $SHELL"
```

Call the script in that way that the standard output of the script will be appended at the end of the file “/home/practice/output_script” in your home directory.

```
script.bash >> ~/output_script
```

Introduction to Computer Science Lab Report 2

12. Use the `at` command [`at(1)`] to start the script `script.bash` at a specific point of time, chosen by you. Check that it works correctly by listing the content of the file `output_script` with means of the `more` command.

```
echo "script.bash >> ~/output_script" | at 15:14
```

What happens when you forget to redirect the standard output of the script?

It outputs via Mail

What are the functions of the commands `atq` and `atrm`? Start several jobs at different points of time and test the two commands!

`atq` lists the user's pending jobs

`atrm` deletes jobs, by their job number.

13. Now try to start the script `script.bash` repeatedly every minute with the `crontab`

The stars are the schedule to run the task. In this case every minute, every

hour, every day, every week, every month.

```
crontab> * * * * * bash /home/practice/script.bash >> ~/output_script
```

`script.bash`:

```
#!/bin/bash
```

```
echo "The current time stamp is:"
```

```
echo "$(date "+ %l:%M %P and %S seconds")"
```

```
echo "You are logged in as user: $USER"
```

```
echo "Your home directory is: $HOME"
```

```
echo "You currently use the shell: $SHELL"
```

Example output:

The current time stamp is:

04:03 and 01 seconds

You are logged in as user:

Your home directory is: `/home/practice`

You currently use the shell: `/bin/sh`