

# Introductory Linux Tutorial for Life Sciences

## Session 10: Exercises

- The following eight exercises are ideally solved in pairs.
- To solve these exercises, it is assumed that the knowledge gained from the previous tutorial sessions is used.

Each solution should be a shell script that makes use of variables where appropriate, and that is documented with moderate use of comments.

Scripts should be stored in the `~/myLinuxProject/scripts` directory and if an exercise requires you to create an output file, this should be written to the directory `~/myLinuxProject/results/courseDay_03` unless the exercise says something else (create these directories if they don't already exist).

The point of this is twofold:

1. Everyone has the same directory conventions, and
2. It shows that you are able to work with directories.

Use the `man` command to get a reference manual for commands that you are unsure of how to use. For example, use `man ls` to read the manual for the `ls` command.

# Exercise 1

Write a script called `fileBasename.sh` that iterates over all files in `~/data/fasta` and prints out the filenames (with no path) of all fasta files (only). Store and use the path to the directory in a variable in the script.

You may want to use a “for”-loop and the `basename` utility.

# Exercise 2

Write an interactive shell script named `register.sh`, which will request the following information from the user:

1. name,
2. surname,
3. username,
4. age,
5. affiliated university, and
6. password.

The user will provide this information via their keyboard. The password should not be visible while typing it in the terminal. Furthermore, you need to make sure that the password is at least 8 characters long. If the length of the password is shorter, then the user should be asked to type the password again, until it fulfills the length criteria.

The length of a string stored in the variable `var` is given by the expansion of `${#var}`.

After collecting all information, return the following message to the user (the password should not be shown on screen):

---

```
The registration was successful!  
We have received the following data from you:  
name          : name surname  
age           : age  
university    : university  
username      : username  
password      : *****
```

---

## Exercise 3

There are a few files in `~/data/mzML`, and they are all empty. They have names like `0_M5.mzML` (all are prefixed with the string `0_` and have a `.mzML` filename suffix).

For each file:

1. Copy the file to the directory `~/myLinuxProject/results/mzML` (this directory may have to be created as a first step). The copied file should not have the `0_` filename prefix (so `~/data/mzML/0_M5.mzML` should be renamed as `~/myLinuxProject/results/mzML/M5.mzML`).
2. Write the sentence “I’m file number N” into each file, where N should be 1 for the first file, 2 for the second, and so on.

While testing, it would be prudent to not use `mv` or `cp` until you know for sure that the file would be copied/moved to the correct place. You may simply use `echo` instead to output the source and target destination for visual inspection. When everything looks ok, you can use `cp` or `mv` “for real”, and then continue writing the rest of the script.

To modify the filename, you could possibly use the `cut` command together with `basename`, or a variable substitution that trims off the prefix.

Verify on the command line that you managed to copy all files and that they seem to contain the correct data after running the script. There should be 183 files.

## Exercise 4

Write a shell script called `lineByline.sh` which outputs every line of the file `~/data/others/Peters_shoppingsList.txt` that starts with a capital C.

There are at least three ways to do this task:

1. Use `grep` with a regular expression.
2. Read the file line by line and use a regular expression in an “if”-statement.
3. Read the file line by line and use a globbing pattern in an “if”-statement.

Try implementing each of these.

## Exercise 5

Write a script, `FastQC.sh`, which runs the command `fastqc` on all compressed fastq files whose names start with a capital `S` in the directory `~/data/fastq` (the `fastqc` utility knows how to read the compressed data, so decompressing the files is not necessary). Write the results to the directory `~/myLinuxProject/results/fastqc` (this directory may have to be created).

Use `fastqc --help` on the command line to get more information for how to run the `fastqc` tool.

Inspect the results after running the code.

## Exercise 6

Explain what the script below does:

---

```
#!/bin/bash -eu

infile="$HOME/data/fasta/brachy_CDS.fa"

for bp in A T C G; do
    var=$( grep -v '^>' "$infile" | grep -o "$bp" | wc -l )
    echo "$bp: $var"
done
```

---

## Exercise 7

Extend the above script (call it `countGC.sh`) so that it counts the total number of base pairs in the file as well as the overall GC contents (the percentage of G or C bases).

## Exercise 8

Write a script called `longest.txt` that will figure out which file out of all the files with a `.txt` filename suffix in `~/data/others` is the one with most lines of text in it.