03ex_Bash

January 18, 2025

1 Lab 03: BASH

The following exercises are meant to be solved by gathering the bash commands incrimentally in **two scripts**, one for ex 1.* the other for ex 2.* **AT THE EXAM THERE IS ALWAYS at least 1 EXERCISE ABOUT BASH. DONT OVERLOOK THIS**

1.1 Ex 1 (done)

1.a Make a new directory called `students` in your home. Download a csv file with the list of students of this lab from here (use the `wget` command) and copy that to `students`. First check whether the file is already there “‘

For ZSH

`wget`is used to retrieve files from the internet. Examples of usage:

```
wget https://example.com/file.zip #download file in the pwd
wget -O newfilename.zip https://example.com/file.zip #download and rename (big O stands for "O
wget -r https://example.com #download all the site and liked pages recursively
wget -c https://example.com/largefile.zip #continue a download if it was interrupted
```

I have MacOS so I cannot use `wget`: my default command is `curl`which works similarly:

“‘bash curl -O https://www.dropbox.com/scl/fi/bxv17nrbrl83vw6qrkiu9/LCP_22-23_students.csv #saves with the original name from the URL curl -o custom_name.csv https://www.dropbox.com/scl/fi/bxv17nrbrl83vw6qrkiu9/LCP_22-23_students.csv #specifies custom name

My solution

```
mkdir students
find /students -name LCP_22-23_students.csv #search for the file in dir /students: no of cours
curl -L -O "https://www.dropbox.com/scl/fi/bxv17nrbrl83vw6qrkiu9/LCP_22-23_students.csv?rlkey=4
# without the -L option, the content of the downloaded file was "<!--status=302-->" which indi
# -L specifies to curl to follow the redirect automatically, and so enables to download the ac
find . -name LCP_22-23_students.csv# "." stands for here in pwd
cp LCP_22-23_students.csv ./students/LCP_22-23_students_copy.csv
cd students
ls
cd -
```

1.b Make two new files, one containing the students belonging to PoD, the other to Physics.

My solution

```
cd students
head -n 5 LCP_22-23_students.csv
cut -d',' -f4 LCP_22-23_students.csv | grep -c "PoD"
# focus on the 4th column, consider ',' as a delimiter, then retrieve and count the entries eq
# this is the basic command
grep "PoD" LCP_22-23_students.csv > "PoD_students.csv" #std output redirection
grep "Physics" LCP_22-23_students.csv > "Physics_students.csv"
# this to also keep the header line (line n 1): ">>" appends, ">" overwrites
head -n 1 LCP_22-23_students.csv > "PoD_students.csv" && grep "PoD" LCP_22-23_students.csv >> "
head -n 1 LCP_22-23_students.csv > "Physics_students.csv" && grep "Physics" LCP_22-23_students
```

1.c For each letter of the alphabet, count the number of students whose surname starts with that letter.

1.d Find out which is the letter with most counts.

1.e Assume an obvious numbering of the students in the file (first line is 1, second line is 2, etc.), group students "modulo 18", i.e. 1,19,37,.. 2,20,38,.. etc. and put each group in a separate file

My solution see file /03_Lab_Bash/students/ex1_script.sh

chmod +x ex1c_script.sh #makes it executable ./ex1c_script.sh #runs it

1.1.1 Ex 2 (done)

2.a Make a copy of the file `data.csv` removing the metadata and the commas between numbers; call it `data.txt`

2.b How many even numbers are there?

2.c Distinguish the entries on the basis of `sqrt(X^2 + Y^2 + Z^2)` is greater or smaller than `100*sqrt(3)/2`. Count the entries of each of the two groups

2.d Make `n` copies of data.txt (with `n` an input parameter of the script), where the i-th copy has all the numbers divided by i (with `1<=i<=n`).