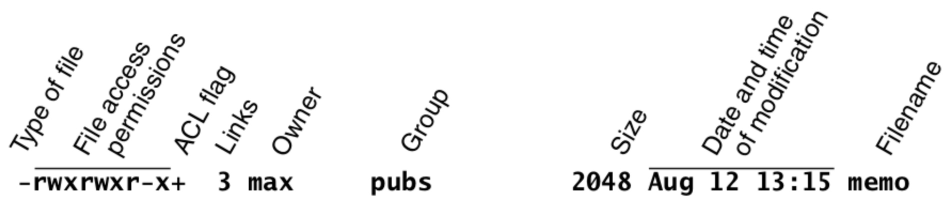
High Performance Computing Exercise Sheet 4

HS 25
Dr. Douglas Potter

http://www.ics.uzh.ch/
Teaching Assistants:
Cesare Cozza, cesare.cozza@uzh.ch
Mark Eberlein, mark.eberlein@uzh.ch

Issued: 10.10.2025
Due: 17.10.2025

Exercise 1 [File permissions]
With `ls -l file_name` and `ls -ld dir_name` you can check the access permissions of files and directories:



```
-rwxrwxr-x+  3 max     pubs          2048 Aug 12 13:15 memo
```

The first 3 letters are the permissions for the owner (`u`) of the file, the next 3 are for the group (`g`) and the last 3 for all the other users on the system (`o`). `rwx` stands for read, write and execute. For directories, `x` means that you can access it and examine what files are inside.

- What are the file permissions for your `$HOME` and `$SCRATCH` directory on Eiger? Can you access other student's (or members in your group) `$HOME` or `$SCRATCH`? Why (not)?
- If you create a new file in these directories, what are the default permissions?
- What are the permissions of `/users/meberlei`? If I tell you that there is a file `/users/meberlei/exam_solutions.txt`, could you open it and read its content? Would you be able to write it?

Access permissions are changed using `chmod`. You can use symbolic arguments, e.g `chmod a+rw file`, where you alter existing permissions by adding (`+`) or subtracting (`-`) read-/write/execute to the owner, group, others or all. Alternitively you can set the permissions with numeric arguments, e.g `chmod 755 file`, with a number XXX representing the permissions for owner, group, others. X = r(4) + w(2) + x(1).

- What command can you use to set the access permissions for a directory so only the owner has (full) access?
- Create a text file on `$SCRATCH` and set the permissions so that group members can write to it.
- Create a file and set its permissions to `000`. Can you do anything with it now? Is the file completely lost?

Exercise 2 [Bash scripting 1 - Regex - `grep`]

The command `grep` can be used to search a file for a specific expression. On the course repository you can find the file `binary.txt`, which contains a list of binary numbers. Give a regular expression that will match only the binary strings that satisfy the condition:

1) end with `00`
2) start and end with `1`
3) contain the pattern `110`
4) contain at least three times a `1`
5) contain at least three consecutive `1`s

- What are the regular expressions that would match the cases (1) – (5) above?

There is a folder named `measured` containing several types of files. Especially, text files `measurements_1.txt` to `measurements_5.txt` contain temperature and humidity data taken at different times.

Commit : Write a simple bash (or python) script which goes through each `measurement_*.txt`. Extract the data of `TEMPERATURE` and `TIME` (so humidity data are skipped) from all files and store it in a single file `out.txt`. In addition, the script should print the average measured temperature. Note there are also other files in the directory `measured`, which are irrelevant for this task. Make sure to skip them in your script.
Hint: Use a regular expression that matches the lines with temperature and time. Furthermore, `bash` is not able to do floating point arithmetic but you can use the program `bc` for calculations.

Exercise 3 [Bash scripting 2 - Is this a prime number?] Write a script that reads an argument from the command line and checks if the number given is prime or not. Make sure to add "safety nets", which assure your script will not crash when invalid input is given by the user.

```
./isprime 31
    31 is prime
./isprime 8
    8 is composite
./isprime seven
    ERROR: The given input is not a number: seven
./isprime -2
    ERROR: The given input is not a positive integer: -2
./isprime 4.5
    ERROR: The given input is not a positive integer: 4.5
```

Remember the “BASH way” is to run other programs to do most of the work.

- What does the `factor` command do? On MacOS it is called `gfactor` and you might need to install it, e.g., `brew install coreutils`. What is the output of this command? Try executing `factor 20` and `factor 31`. How does the output differ between the two?

What could you use to distinguish a prime from a composite number? In the class you learned about pipes. Is there another program you could pipe to that would help? Look at the man page for `wc` (`man wc`) and see if there is a flag that would help here.

- In the class you also learned about “output capture” and “test” with if/then/else. How would you capture the output of the above commands and construct an appropriate test to determine if the input was prime?
- Use the time function to measure the performance of your code. Test it on 4230283 and 4572862171001 (are these primes?). How much time did it take to compute the result?

Commit : Provide your script.

BONUS:

a) Modify the script from exercise 3 to take multiple numbers. It should check each number to see if they are prime or not. In the class you learned how to refer to parameters ($1 etc.). Could `shift` help here? For example,

```
./isprime_bonus1 7 15 32
```

```
    7 is prime
    15 is composite
    32 is composite
```

b) Write a script which will print all primes from a give range. In the class you learned about looping using `for` and `while`. There is a command called `seq` (`man seq`). Could you somehow combine a `for` loop and the `seq` command with output capture to do this. Example:

```
./isprime_bonus2 30 45
    31 is prime
    37 is prime
    41 is prime
```

Commit : Provide your scripts.