

## Scientific Computing 1 1st Homework – 2nd Tutorial

**Handout:** 16<sup>th</sup> Oct. 2016

**Return:** 19<sup>th</sup> Oct. 2016

---

**Hint:** If you are not sure what a command does, or what the right parameters are, use `man`, `apropos` or `info` to get help.

---

### Exercise 1:

(6 Points)

`tar` is the standard Linux/Unix file archiver. Normally, it is combined with an additional compression tool like `gzip`, `bzip2`, or `xz`. Read the manpage of `tar` to figure out how:

- a.) a given archive can be extracted,
- b.) one creates an archive from a given set of files,
- c.) to list all files contained in an archive,
- d.) and how the compression is selected. How does the selected compression corresponds to the filename of the archive?

Demonstrate the usage of `tar` by realizing the following actions:

Create some files with arbitrary content (using `echo`).

- a.) Create a `.tar` archive of them.
- b.) Create a `.tar.gz` archive of them.
- c.) Create a `.tar.bz2` archive of them.
- d.) Extract the three archives to different directories.
- e.) Use the `diff` command to check if the extracted files are the same as the originals.

### Exercise 2:

(4 Points)

- a.) Use `grep` to find the string "copyright" in some of the license files in `/usr/share/common-licenses/`. Which option must be added to `grep` such that also the line numbers of the matching lines are displayed?

Search for a program which can count lines of a file using `apropos`. Use this program and `grep` to count how many lines contain the string "copyright" in the `/usr/share/common-licenses/` directory.

- b.) The # sign is used as comment identifier in many configuration files. Use `grep` to display all lines of a file which are not a comment. An example file which uses # for comments is `/etc/resolv.conf`<sup>1</sup>.
- c.) In MATLAB® the `tic` and `toc` commands can be used for time measurement. They produce a characteristic output like:

Elapsed time is XXXX seconds

Write a `grep` command which searches for such lines in a file and displays the match together with five lines before and two lines after it.

**Exercise 3:** **(4 Points)**

Write a regular expression which matches:

- a.) all strings ending with a dot.
- b.) all strings beginning with a # symbol.
- c.) a floating point number in the standard notation (e.g.: 19.456).
- d.) a floating point number in scientific notation (e.g.: 3.123e + 4).

**Exercise 4:** **(3 Points)**

The `date` command displays the current time and the date. Use `awk` to filter the output and

- a.) display only the time.
- b.) display the time zone.
- c.) get the day, the month, and the year and rearrange them to the German format.

**Exercise 5:** **(4 Points)**

The following table is an output from a special linear system solver:

it	VF/Z_F	R(ZZ')_2	Res-2 Change
0	1.000000000e+00	7.443409951e-01	1.000000000e+00
1	1.227312449e-01	7.436413735e-01	9.408051295e-04
2	3.323675855e-01	7.128056828e-01	4.325960273e-02
3	3.616833196e-01	6.558804004e-01	8.679216872e-02
4	4.003615306e-01	5.072151826e-01	2.931008829e-01
5	4.194196732e-01	2.580984101e-01	9.652007249e-01
6	2.408619304e-02	2.567466375e-01	5.265005836e-03
7	2.095617807e-01	2.596593240e-01	1.121733828e-02
8	2.338129836e-01	6.063714059e-02	3.282182858e+00
9	9.471210783e-02	6.198966248e-02	2.181850705e-02
10	3.249626967e-02	1.463117330e-02	3.236821012e+00
11	8.854898658e-02	1.351164483e-02	8.285656453e-02
12	2.775026128e-02	1.392461538e-02	2.965759103e-02
13	1.186741771e-01	5.089817861e-02	7.264221283e-01
14	4.586412631e-02	5.096929146e-02	1.395209723e-03
15	2.689489164e-03	5.097322108e-02	7.709189191e-05

<sup>1</sup>This file specifies the DNS server address for the name resolution in network applications, such as browsers.

In order to use this output in articles or to create plots from it, it needs to be transformed to other formats. Realize the following jobs using `sed` and `awk`:

- a.) Extract the values from the first and second column and rearrange them to a MATLAB compatible matrix of dimension  $maxit \times 2$ . That means it should fit in the following layout:

```
data = [  
  value from column 1, value from column 2;  
  ...  
];
```

- b.) Write an `awk`-script, which transforms the table into a  $\LaTeX$  table. A preprocessing with `sed` can be necessary. The table should look like:

```
\begin{tabular}{r|r|r|r}  
  Head 1 & Head 2 & Head 3 & Head 4 \\ \hline  
  value 1 & value 2 & value 3 & value 4 \\  
  ...  
\end{tabular}
```

- c.) **Bonus:** Modify all floating point numbers such that the exponent is written as  $\cdot 10^x$  instead of using the `e+x` notation. The  $\LaTeX$  code for  $\cdot 10^x$  is `\cdot 10^{x}`.

The input file can be downloaded from [http://www2.mpi-magdeburg.mpg.de/mpcsc/lehre/2018\\_WS\\_SC/tutorial/awk\\_input.txt](http://www2.mpi-magdeburg.mpg.de/mpcsc/lehre/2018_WS_SC/tutorial/awk_input.txt).

### Exercise 6:

(2 Points)

Write a `sed` command to replace all multiple spaces in a string by only one space.

### Exercise 7:

(2 Points)

A single-line comment in C is introduced by the sequence `//`. Alternatively the comment can be enclosed by `/*` and `*/`. Write a `sed` command which transforms the `//` comments to the enclosed ones.

**Overall Points: 25**