Exercise Sheet #2

Bulcsú Sándor <sandor@th.physik.uni-frankfurt.de>
Hendrik Wernecke <wernecke@th.physik.uni-frankfurt.de>
Laura Martin <lmartin@th.physik.uni-frankfurt.de>
Christopher Czaban <czaban@th.physik.uni-frankfurt.de>

How to submit your solutions
Here some advice how to make communication more efficient.

(a) In order to get the points for solving a problem you need to submit your fully **commented and compilable solution** in time.

(b) Therefore send all relevant files **named by your name and the problem** it solves, e. g. `SmithJoe_problem2a_additionalInfo.xyz`.

(c) **Never** use a space-character in a file name!

(d) The **subject** of your submission email must start with `CPP` and contain the number of the problem sheet, e. g. `CPP: sheet 2`. Else your mail might be disregarded.

(e) The **deadline for submission** is on Friday 12 pm and will always be announced on top of each problem sheet.

Problem 1 (*Shell script*) 10 Pts

Write a shell script called `mySpamCreator.sh` that sends an email every minute to an email adress given to the script as an argument. You can choose the text you are sending with the email, but also attach a time stamp. In order not to crash any mail account make sure that the programm terminates after having sent maximally 5 mails.

Hint: A quick overview of linux commands is given here:
`https://gist.github.com/LeCoupa/122b12050f5fb267e75f`.

For sending emails from command line you can use the programm `mutt`. The typical syntax looks like:

```
echo "my message!" | mutt -s "my subject" -- email@adress.to
```

Advanced: Modify the program such that it sends an email at a given point in time. Try to send the 'spam' to several adresses, which you read from a file.

Note: Sending spam to a tutor of this course will decrease your number of points by one point per received spam mail.

Problem 2 (*Gnuplot*) 10 Pts

(a) Create a gnuplot script following the instructions below: (6 Pts)

(i) Set up the terminal and output commands to create a postscript file with color as an output. **Hint:** For output terminals see: `http://www.gnuplotting.org/output-terminals/`.

(ii) Set the x-axis range to (0, 10) and the xlabel to `x`, and the y-label to `f(x)`.

(iii) Plot the two functions $e^{1/x}$ and e^{x} in the same plot, using different line widths, line types and line colors. Give appropriate titles to each function. **Hint:** For plotting functions see: `http://www.gnuplotting.org/plotting-functions/`.

(iv) Create the output file by calling `gnuplot <scriptname.gnu>`.

(b) Get the data file from `http://itp.uni-frankfurt.de/~wernecke/data.dat`, which contains as first column the x-, as second column the noisy measurement data. (4 Pts)

(i) Create again a plot as postscript file similar to above to plot the data. **Hint:** For plotting data see: `http://www.gnuplotting.org/plotting-data/`.

(ii) Define a polynomial of second order and fit it to the data. Plot the data and the fit into a single file, using different line widths, line types and line colors and proper titles. **Hint:** For plotting data see: `http://gnuplot.sourceforge.net/demo/fit.html`.