Winterterm 2012/2013

Otto-von-Guericke-University Magdeburg
Max Planck Institute for Dynamics of Complex Technical Systems
Computational Methods for Systems and Control Theory

Dr. Jens Saak, Dipl.-Math. Martin Köhler
Website: http://www.mpi-magdeburg.mpg.de/mpcsc/lehre/2012_WS_SC/

Scientific Computing 1

2nd Homework

Solution

Handout: 10/15/2012 **Return:** 10/22/2012

Exercise 1: (2 Points)

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 the line numbers of the matching line are displayed, too.

Solution:

```
 grep "copyright" /usr/share/common-licenses/GPL-2
```

The `-n` or `--line-number` options displays the lien numbers.

Exercise 2: (2 Points)

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 # as comment is `/etc/resolv.conf`[1].

Solution:

```
grep -v ^# /etc/resolv.conf
```

Exercise 3: (1 Point)

In MATLAB® the `tic` and `toc` commands can be used for time measuring. They produce an characteristic output:

```
Elapsed time is XXXX seconds
```

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

Solution:

```
grep -A 2 -B 5 "Elapsed time is"
```

Exercise 4: (6 Points)

Write a regular expression which matches:

a.) all strings ending with a dot.

[1] This file specifies the DNS server address for the name resolution in network applications, such as browsers.

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.: $3.123e + 4$).

Solution:

a.) `\.$`

b.) `^#`

c.) `[+-]\?[0-9]*\.[0-9]*` or `[+-]?[0-9]*\.[0-9]*`

d.) `[0-9]\?.[0-9][eE][+-][0-9]*`

Exercise 5: **(2 Points)**

A data set is given in a whitespace separated table:

```
value11 value12 value13 value14 value15
value21 value22 value23 value24 value25
```

Write an `awk` command line to fetch the first and the fourth column. Use `cat` and the pipe operator to supply the input data. A data file is available on: http://www.mpi-magdeburg.mpg.de/mpcsc/lehre/2012_WS_SC/data/easy_awk.txt

Solution:

```
cat file | awk '{print $1 $4;}'
```

Exercise 6: **(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.

Solution:

Date output:

```
date
Thu Oct 11 16:08:26 CEST 2012
```

a.) `date | awk '{ print $4;}'`

b.) `date | awk '{ print $5;}'`

c.) `date | awk '{ print $3 "  " $2 "" $6; }'`

Exercise 7: (2 Points)

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

Solution:

```
sed -e ’s/[ ]+/ /g’
```

Exercise 8: (3 Points)

A 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.

Solution:

```
 sed -e ’s,/\/\/\(.*$\)/\/\* \1 \*\//g’
```

Overall Points: 21

In the Tutorial

The following exercises are not part of the homework and will be solved in the tutorial. How ever if you solve them successfully during the homework, you can earn some extra points.

Exercise 9:

Write a regular expression which matches on a valid email address.

Solution:

```
[A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-Z]{2,4}

[A-Z0-9._%+-]+@[A-Z0-9.-]+\.(?:[A-Z]{2}
|com|org|net|edu|gov|mil|biz|info|mobi|name|aero|asia|jobs|museum)
```

Exercise 10:

Grep a C source code for all functions headers and write them to a file. Use `sed` to remove a possible terminal { characters and set a `;` after the last closing parenthesis. Use the pipe operator to glue both together to one command line call. You may assume that all function headers in the file span only one line.

Solution:

```
grep "[a-zA-Z0-9_]\+[ \t] \+ [a-zA-Z0-9_]\+(.*)" test.c \
| sed -e ’s/) [ \t]*{\?/);/g’
```

Exercise 11:

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
```

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;
  ...
  ];
```

Solution:

```
#!/usr/bin/awk -f
BEGIN{ FS="|";
       print "M = ["; }
$1~/[0-9]/ {print $1,  $2 ,";" ; }
END{print "];"; }
```

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}
```

Solution:

```
#!/usr/bin/awk -f
BEGIN{ FS="|";
       print "\\begin{tabular}{r|r|r|r}";  }
$1~/it/ { print $1, "&", $2, "&" , $3, "&", $4, "\\\\\\hline";   }
$1~/[0-9]/ {print $1, "&",$2 ,"&",$3,"&",$4,"\\\\" ; }
END{print "\\end{tabular}"; }
```

c.) 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}.

Solution:

```
cat awk_tex_matlab.dat | awk -f ./parse_to_tex.awk | \
sed -e ’s,e\([+-][0-9]*\),\\cdot 10^{\1},g’ -e ’s,_,\\_,g’
```