Advanced Programming Techniques in Health Care

Health Care IT
WS 2018

Using Unix Tools
1. Assignment
(Deadline: 2018-11-13 10:00)

In the directory `I:\EngIT\HCIT\Tools\unixutils` you will find some UNIX tools runnable on Windows.
Use them (or the tools on a real unix system) to solve the following tasks.

1. Using Regular Expressions and `grep`
   Using the file `faust.txt`, write commands (`grep` with regular expressions) to output the lines with the following characteristics:

   (a) all lines, that contain a number (digit)

   (b) all lines, that start with a number (digit)

   (c) all lines, that start with a underscore

   (d) all lines, that start and end with a underscore

   (e) all lines, that contain the letters `Strom`

2. Using Filters
   Using the file `faust.txt`, write commands (`grep` with regular expressions and/or filters like `sort`, `uniq`, `wc`, `awk` and `cut`) to produce an output with the following characteristics:

   (a) count the number of lines in the file `faust.txt`

   (b) all **different** lines, that start and end with a underscore

   (c) the **number** of all lines that start with the word `Und`.

   (d) give the **different** entries of the `Username` field in the files used in assignment 1
   Try to solve this example with `cut` and `gawk`

3. Using `awk`
   Write a small `awk` script that analyses the log file and calculates how often a user started a task. Your output shout look like the following (with the help of `sort`):

   ```
   andreas Task 0 count: 874
   andreas Task 1 count: 908
   andreas Task 2 count: 872
   .........
   susanne Task 14 count: 849
   susanne Task 15 count: 871
   ```

   Every Line in the log-file has the following (space delimited) structure:
   `Loglevel Thread-Identifier Username Date Time Task optionalText`

`Loglevel` is a numerical value, describing the log-level in which this output has been generated.
The log-level 0 is only used to show that a new thread has been created.

`Thread-Identifier` shows which threads has generated this output.

`Username` is a string containing the username (the username must not contain any spaces).

`Date Time` shows at which date and time the output was generated (`Date Time` is in the C# format: `dd.MM.yyyy HH:mm:ss.fff`).

`Task` is a number containing the executed task. To enable an analysis of the duration of a task the start and end of the task is encoded in a bit:

- bits 0..7: task number
- bit 8: start of the task
- bit 9: end of the task

Start and end of a task will always be logged in log-level 1, and it is guaranteed that the end of a task occurs in the log file after the start of the task.

Example:

```
1 Thread_5 kerstin 15.09.2010 12:49:49.153 266
0 Thread_7 no_user 15.09.2010 12:49:49.153 1
1 Thread_10 andreas 15.09.2010 12:49:49.153 258
5 Thread_5 kerstin 15.09.2010 12:49:49.296 10
4 Thread_5 kerstin 15.09.2010 12:49:49.426 10
1 Thread_5 kerstin 15.09.2010 12:49:50.271 522
```

The above example can be interpreted as follows:

Line 1: User kerstin has started the action 10 (bit 8 set) at 15.9.2010 12:49:49.153, and `Thread_5` is working on it.

Line 2: `Thread_7` has been started.

Line 3: User andreas has started the action 2 (bit 8 set) at 15.9.2010 12:49:49.153, and `Thread_10` is working on it.

Line 4: `Thread_5` is still working on action 10.

Line 5: `Thread_5` is still working on action 10.

Line 6: `Thread_5` has finished action 10 at 15.09.2010 12:49:50.271 (bit 9 is set). The duration of the task was 1.118 seconds.

A thread always works on a task from start to end and can not be interrupted by another task.

A few example log files are given on:
`I:\EngIT\HCIT\1. Semester\AdvProTe\exercises\log(0|1|2|3|4).txt`

4. Using `sed`
In the file `faust1.txt` many lines start with two or more spaces; use `sed` to delete these spaces on all lines and write the output to a file named: `faust1ws.txt`.