

PS7A

KRONOS TIME CLOCK: INTRODUCTION TO REGULAR EXPRESSION PARSING



In this assignment, we begin the analysis of the Kronos InTouch time clock log by using regular expressions to parse the file.

DETAILS

Here is a summary of the whole assignment:

1. Verify device boot up timing.
2. Verify device startup services.
3. Verify device software upgrades.

In this portion of the assignment, we will only be doing (1) – boot up timing.

Your job is to read in an entire InTouch log and report:

- each startup
- whether it completed
 - if so, how much time it required
- whether it failed

Your output should begin with the line number of the startup message “(log.c.166) server started”, the timestamp of the startup, and either: success followed by elapsed time, or failure.

- Success is determined by a line containing the string “oejs.AbstractConnector:Started SelectChannelConnector” that follows the startup message.
- Failure is determined by another startup message before the success message, or end of file.

NOTES

computing4summer2018

[Home](#)[portfolio](#)[psX](#)[ps7b](#)[ps7a](#)[ps6](#)[ps5](#)

- While it's possible to do the assignment without the use of regular expressions, the point of this assignment is to learn about regular expressions.

Therefore it is required that you use the C++ regular expression (regex) library.

- **The built-in regex library provided in C++98 gcc is incomplete and should not be used.** Instead, we will use the Boost regex library:

http://www.boost.org/doc/libs/1_58_0/libs/regex/doc/html/index.html

- To install the Boost regex library on your own machine, run this command:

```
sudo apt-get install libboost-regex-dev
```

- To use the Boost regex library in your source code, use the following:
 - header include directive `#include <boost/regex.hpp>`
- To link to the Boost regex library, use this compiler flag: `-lboost_regex`
- To install the boost date/time library, run the command:

```
sudo apt-get install libboost-date-time-dev
```

- Kronos log files and sample output may be **downloaded** here:

[device2_and_6_log.7z](#)

[device1_3-5_log.7z](#)

[device6_intouch.log_SOFTLOAD.zip](#)

- **For this week, you only have to do the Device Boot up Timing portion of the assignment.** Sample output is in the file `device5_intouch.log_BOOT.rpt` in

- Use the Boost time and date functions for figuring out elapsed time between server boot and boot completion. See:
 - Main docs: http://www.boost.org/doc/libs/1_57_0/doc/html/date_time.html
 - Time examples:
http://www.boost.org/doc/libs/1_57_0/doc/html/date_time/examples.html#date_time.examples.time_math
 - Using the Gregorian features:
http://www.boost.org/doc/libs/1_57_0/doc/html/date_time/gregorian.html#date_time.gregorian.date_class
- Your executable file must be named ps7a. It should accept a log file as the argument and create an output file with the same name but with a suffix .rpt. E.g. to run it, you might use at the shell:


```
./ps7a device1_intouch.log
```

and your code should would produce a file named device1_intouch.log.rpt.

SUBMIT

Submit the following:

- Your source code .cpp file and any header file(s).
- A Makefile for building the code.
- Output from running your code on each of the five InTouch log files. Your output files must be named device[1-5]_intouch.log.rpt.

(full & correct implementation=4 pts; nearly complete=3pts; part wa s; started=1 pt)

Use of Boost time methods: 2**Output files included: 2**

(full & correct = 2 pts; partial = 1 pt)

cpplint: 2

(your source files pass the style checks implemented in cpplint)

Makefile: 2

(makefile included)

Readme:4

(2 pts for describing regexs in readme; 2 pts for discussion)

Total: 16