Programming Exercise CSS/R&D/Connectivity

Goal of the exercise

To assess the knowledge level, skill set and abilities of the applicant in the field of C# development targeted towards the Connectivity team.

Secondary goal is to give the applicant an idea of the issues and challenges faced in the R&D department.

Exercise description

At the R&D department we often have challenges where we have to deal with SDP (Session Description Protocol) messages. In this exercise we ask of you to extract some fields from a high volume of these messages. A SDP message specifies the audio path of a telephony session and defines: IP address, UDP port and codecs used.

This exercise is split into multiple assignments. Each following assignment adds functionality to the single program you will be developing for all assignments. It is optional to complete all assignments in the given time and you may simplify or skip assignments. The exercises are given to find your strengths in C# development and understand your way of approaching problems.

The exercise material consist of:

- a development environment with working internet connection
- sample input file with SDP messages

If anything is unclear or unspecified, do not stop but log your assumptions and continue. A working or functionally complete solution is not the most important goal of this assigment.

The time allotted for this exercise is 1,5 hours and half an hour to discuss the results.

Assignment 1

- Create a C# project to read in the supplied file: sdp_input.txt
- Add the logic to split input into separate messages. Messages are delimited by a double newline or when the file ends.
- Output each separate message

Assignment 2

Parse the following information from each individual message and store this in-memory:

- Extract IP address from *'c='* line
- Extract port from *'m='* line
- Extract codec(s) (G722/PCMU/PCMA/etc) from the *'a='* lines. Store this as a sequence of elements.

Some lines in the input are not required and must therefore be ignored. Messages with invalid values may be handled however is convenient.

Output now only the information in the objects created in this assignment. The information must be output together, such that the combinations of IP, port and codec are grouped in the same way as in the input document.

Assignment 3

Add timing measurement to the program and output the total processing time required (in milliseconds) in such a way that the time required to output the information is not measured. Record (on paper, in notepad) the initial result of the functionallity complete program.

Optimize the program however you see fit to reduce the processing time required. Detail the optimizations you made, why you made them and the new time. The goal is to reduce the runtime of the program as much as possible.

Consider:

- Improving the way the file is read
- Improving parsing
- How the data is stored
- Multi threaded processing