

Coding Task Description

Implement a program that monitors web sites and reports their availability. This tool is intended as a monitoring tool for web site administrators for detecting problems on their sites.

Main functions:

1. Reads a list of web pages (HTTP URLs) and corresponding page content requirements from a configuration file.
2. Periodically makes an HTTP request to each page.
3. Verifies that the page content received from the server matches the content requirements.
4. Measures the time it took for the web server to complete the whole request.
5. Writes a log file that shows the progress of the periodic checks.
6. *(OPTIONAL) Implement a single-page HTTP server interface in the same process that shows (HTML) each monitored web site and their current (last check) status.*

Details:

- The “content requirement” can for example be just a simple string that must be included in the response received from the server, e.g. one rule might be that the page at the URL <http://www.foobar.com/login> must contain the text “Please login:”.
- The checking period must be configurable via a command-line option or by a setting in the configuration file.
- The log file must contain the checked URLs, their status and the response times.
- The program must distinguish between connection level problems (e.g. the web site is down) and content problems (e.g. the content requirements were not fulfilled).

There is a lot of freedom to choose software technologies, tools and file formats to achieve the goal.

Note that the task is meant for evaluating both coding skills and software architecture design skills. Pay attention to the design of applications you create. Be ready to defend your architectural decisions in a discussion.

It is necessary to personally write the source code, but it does not have to be complete.

Deliverables:

The software is delivered with the full source code included. All used source code must be freely distributable.

If necessary, include readme.txt to describe the software and how it meets the requirements.

Design question (Optional)

Assuming we wanted to simultaneously monitor the connectivity (and latencies) from multiple geographically distributed locations and collect all the data to a single report that always reflects the current status across all locations. Describe how the design would be different. How would you transfer the data? Security considerations?