

# Project

ETH Zurich

**Hand-out:** 13 March 2012

**Team registration:** 21 March 2012

**Due:** 8 May 2012

## 1 Synopsis

As a new employee at the Algonquin web-server company, you have been tasked with developing a new experimental web-server. You, being on the cutting edge of software development, have decided to implement this system in SCOOP. This will allow you to make your employer aware of the newest developments in programming languages, and how they may help the company.

## 2 Requirements

You must implement a web-server that handles a very small subset of the HTTP/1.1 protocol, a full description of which is available in the W3C RFC-2616 [2]. The server must be concurrent, i.e., able to serve more than one request at the same time. Additionally, for record-keeping purposes, the server must be able to record the each client's IP address along with the requested files in a mutable data-structure shared among processors.

Since this is a proof of concept, you are only required to handle GET requests for relative URIs (i.e., no "\*" requests). As a minimum standard, the server must be able to handle requests from at least two of the major browsers: Internet Explorer, Firefox, Chrome, Safari, Opera. Please remember that not all requests may be small HTML files.

Additionally, the server must respect the *Connection* header of the request and keep the connection alive for future requests if asked to. This aspect differs from the wording in RFC 2616, which allows a server to immediately close a keep-alive connection if it wishes. If asked to close the connection, then the connection should be closed. More details on connections can be found in RFC 2616, Section 8. File preferences (via *Accept*) do not need to be strictly adhered to, any of the acceptable files may be served. You may assume that the identity content encoding is always acceptable. *Accept-Languages* may be ignored entirely. All other request headers can be ignored, as long as they do not contradict the above requirements.

## 3 Deliverables

### 3.1 Report (25 points)

The project report must include the following elements:

- An overview of the system design and its implementation. This should include an explanation of how SCOOP was used to cover the requirements. (10 Points)

- A performance evaluation. The evaluation must compare the performance of the concurrent web-server with a version of the web-server that processes requests sequentially. In particular, the evaluation must examine the effect on the speed-up of the concurrent web-server when the size of the served pages varies. The evaluation must also examine the effect of logging on the performance. To make the measurements, you can use `httperf` [1]. (8 Points)
- A discussion of SCOOP's usability for the task at hand. Analyze the aspects that SCOOP handled elegantly or effectively and analyze the aspects that were difficult to express with SCOOP. (3 Points)

The report will also be expected to have a clear structure and proper grammar and spelling. (4 points)

### 3.2 Design and implementation (25 points)

Design and implementation should adhere to the requirements (20 Points). The code should be well-documented (3 Points) and well-structured (2 Points).

## 4 Teams

You can work in teams of up to three persons. Please send an e-mail to one of the assistants ([benjamin.morandi@inf.ethz.ch](mailto:benjamin.morandi@inf.ethz.ch), [scott.west@inf.ethz.ch](mailto:scott.west@inf.ethz.ch)) to register your team. The due date for the registration is indicated at the top of this assignment.

## 5 Support

You have the possibility to ask questions during the exercise sessions. If you need immediate help, you can also contact the assistants directly. For resources on SCOOP and EiffelStudio, you can also consult the course website.

## 6 Submission

Please submit a zip file that contains the source code and the report in PDF format by e-mail by the due date. The source code should be *without* the EIFGENs directory (which contains compiler-generated files).

Your zip file should contain two folders: a *source* folder for the source code and a *documentation* folder for the report.

## References

- [1] HP Labs. `httperf`. <http://www.hpl.hp.com/research/linux/httperf/>, 2012.
- [2] World Wide Web Consortium. RFC 2616. <http://www.w3.org/Protocols/rfc2616/rfc2616.html>, 1999.