Rudy, a small web server

Oskar Olofsson

September 11, 2018

1 Introduction

This lab aims to understand the basic structure of a web server in Erlang.

2 Socket API

A socket API is consisting of libraries for handling Berkeley Sockets, which are abstractions that make handling connections over the Internet simpler for programmers. A socket is made up of an IP address and a port number.

3 Server process

A server process is a process waiting for incoming requests, it includes the procedures for handling the requests and send back a response accordingly. A server module Rudy was implemented as specified with procedures:

- init(Port) which init a listening socket for passing to the handler function.
- handler(Listen) uses the socket for listening for incoming requests
- request(Client) parses the request using the http module.
- reply(Request) replies to the request with the URI as argument.

4 HTTP

HTTP v1.1 is specified as a stateless application level protocol for distributed, collaborative, hypermedia information systems as specified by RFC 2616

The HTTP module is consisting of functions for parsing requests. The parser is set up to split the request into the parts request-line, body and headers. In the implementation we are recursively going through the string looking for characters marking the end.

5 Evaluation

Measuring the time with different sleep times for 100 requests resulted in:

Sleep time (s)	Benchmark time (s)
0.01	1.19
0.02	2.20
0.04	4.19
0.08	8.20
0.16	16.20

We can se that we have around 0.2 s actual delay in all cases (0.002 per request). This makes the delay significant for all cases. At 0.04 s delay time as in our program, every request takes 0.0419 s. We can make at most 1 / 0.0419 = 23.8, 23 requests per second.

6 Improvements

Possible improvements include using multi-threaded extensions. Creating a new process for every request is an alternative. An other alternative is to create a new thread for every request, which is cheaper than creating a new process. According to the course literature: Distributed systems, Concepts and design. It takes about 11 milliseconds to create a new UNIX process and about 1 millisecond to create a new thread, this was done using a CVAX processor running the Topaz kernel.

To know how long the body is you could look at the content-length parameter. Or if the client sends a message with the parameter: Transfer-Encoding: Chunked, it will have to be parsed until it finds a chunk with length 0.

7 Conclusion

This lab was a good introduction to server based Erlang programming.