Chair for Algorithms
and Data Structures
Prof. Dr. Hannah Bast
Marjan Celikik

Search Engines WS 09/10

http://ad.informatik.uni-freiburg.de/teaching

Exercise Sheet 2

complete until Tuesday, November 3rd

Exercise 1

Write your own small webserver in a programming language of your choice. Your program should take a parameter that specifies the port on which your webserver listens. In response to a GET query, your webserver should return the standard HTTP headers + a sentence of your choice repeated as many times as specified by your GET query, see below.

Test your webserver by running it on some port, say 8080, on your local machine and then typing a URL, say, „http://localhost:8080/sentence?repeat=87“, in a browser of your choice. For this URL, it should then show the sentence 87 times.

Exercise 2

Rewrite your query processing code from Exercise Sheet 1 as a server. Again, this server should take a parameter that specifices the port on which it is listening. Your server should speak HTTP, that is, it should accept GET requests and then return the standard HTTP result header + the answer to the query in some basic HTML formatting.

Test your server by running it, typing a URL with the corresponding port and some query (the exact syntax is up to you) into a browser and then check the resulting page.

Exercise 3

Create a simple HTML page with a search field and a search button such that when the search button is pressed, the query in the search field is send to your query processing server and the result page is shown on the screen.

Extend your code from Exercise 2 such that the result page again contains the search field (with the query you typed) and the search button.

Exercise 4

We will set up three servers on one of our local machines that deliver a requested amount of data, one via a TCP socket, the other via a UDP socket, the third via HTTP. Details will be given on the Wiki. Write a client that can communicate with both the TCP and the UDP servers and measure the respective data transfer rates and how much data gets lost. For comparison, also query the HTTP server (via a program like wget or curl).