Documentation for the server

- Main Page
- Compound List
- File List
- Header Files
- Compound Members
- File Members

Introduction

This is the documentation for the server program.
It is a simple multithreaded HTTP server. The second part of the report can be found here.

Building server

What you need

- Linux or Solaris
  (Tested with Linux 2.2.13 and libc6)
- pthread library
- gcc or similar compiler (tested with egcs 2.91-66)
- GNU make
- libfl (from the flex package)
- flex if you wish to remake the C-code for the lexical analyzers from the .lex files (usually not)

Building it

server needs a little bit more attention than client:

- Change to the server directory
- Type `./configure` to configure the source for your system. If configure reports any error, the source will very likely not build correctly. Please report bugs to the authors.
- Edit the file `src/w3server.c` and change the global variable `HTDOC_BASE` to the root directory containing your html files. Have a look at the other variables, even if they are unlikely to require any changes.
- Type `make` at the command prompt. This should build the server executable in the server directory.
  If something does not compile, it might be a good idea to call `make clean` first. Optional: Call `make clean` to delete all object files and auto-save files. `make stat` outputs a little code statistics. You need the `wc` program to use this.

Usage

The command line format to call `server` is:
`server port`
Where

- `port` is the port number the server is running on. The standard is 80, but you must supply the port number anyway.

Technical: Operation of server

main.c

Call to w3StartServer (in w3server.c), the argument being the portnumber. This call does everything the calling function wants, namely setting up a simple HTTP server for the given port :-)

w3StartServer

Calls w3ServerInit to initialize the server. This sets up the socket and fills in the given w3Server structure.

Sets up 5 threads of execution which do the following: 1. Call w3ServerAccept to accept a connection request from a client. 1.a) when a connection is accepted, a new thread is immediately started to replace the actual thread. 2. Call w3ServerServeClient to handle anything the client could want. 3. Call w3ServerDisconnect to close the connection.

w3ServerAccept

Calls the listen() library function to set the socket up to listen to one incoming connection request. Calls the accept() library function to accept a connection on the previously created socket (created by w3ServerInit). Returns a pointer to a w3ServerConnection structure.

w3ServerServeClient

Receives a request string from the (connected) client and responds to it using w3ServerRespond.

w3ServerRespond

Parses the given request string for the requests GET, POST, HEAD, of which currently only GET is handled properly (I hope). It responds to GET by sending the requested file to the client.

Error handling

It was tried to catch any error that is likely to occur, but since this is an early development stage, nothing is guaranteed ;) Errors like file-not-found in a request are handled by sending an error page to the client (see routines in w3html.h / w3html.c).

Known bugs

Currently, no bugs are known. If you encounter a bug, please report it to
Christian Gosch
or
Liuben Tonev

Generated at Mon May 29 19:37:04 2000 for HTTP Server advanced version by written by Dimitri van Heesch, © 1997-1999