Writeup: httpserver

Cagan Bakirci

CruzID: cbakirci

1. Testing

   The program has been tested using both unit testing and whole-system testing. The individual parts of the components have been thoroughly tested as they were being build. After the program was complete, an excessive whole-system testing has been applied. The details of the tests are described below.

   a) Unit Testing

      The unit testing has been applied after each phase of the algorithm. The initial testing was applied to the first phase of the algorithm to make sure the algorithm could receive the input without any mistakes or problems. This initial test also tested the cases where there are no outputs presented to server, hence the server should set the port to a default value (80) and the number of thread to default value(4).This test also tested and confirmed that the server breaks in case "-a " is not given as an argument. This unit testing also made sure that the algorithm was able to open a socket and connect without a problem.

      The second test conducted was aimed to detect the program's ability to parse the header. This test made sure that the program was capable of parsing the header and differing between a get request, a put request, and a patch request. The test concluded that the program was able to parse the header and detect the type of the request.

      The third unit test was conducted to test the function that checks the validity of the resource name. The test concluded that the function is able to produce the right output both when there is a slash before resource name and there is not.

      The fourth unit test was conducted to check the get request. The get request tests show that the get request worked on the local computer. The test also checked the cases where the program was getting input larger than 16Kib.

      The fifth unit test was conducted to check the put request. The put request tests show that the put request worked on the local computer. The test also checked the cases where the program was putting files larger than 16Kib.

      The sixth unit test was conducted to check the patch request. The patch request tests show that the patch request worked on the local computer with both get and put request. However, there is a bug with the patch request. In order for the patch request to work, the file should be created each time, otherwise the open return a -1 and breaks the entire request. The test also checked the cases where the program was getting input larger than 16Kib. Each cases seems to be working when the file is created for the first time.

      The seventh unit test was conducted to check the multithreading. The test aimed to find the system's ability to run multiple threads simultaneously without exceed 16 MiB memory for each thread.

   b) Whole-system Testing

      The whole-system testing was conducted to the final version of the server. It tested the system's ability of handling any case. The whole-system testing confirmed that the

program produced the right output for any given size of files, including the files with incorrect file names. The server has also been run against the valgrind to check for any possible memory leaks. It has been observed that the server is free from memory leaks and the memory allocation was successfully executed. The logging has not been implemented correctly, so the test results for logging have been ignored. The logging file is present, and it is written only when it should be; but the server is incapable of writing the correct value to the log file.

2. Questions
    - Explain the difference between fully resolving a name (to an httpname) when the name is created and the approach that you're taking for this assignment. Give an example of when it might be useful.

      The difference is using alias is clearly more comfortable for the users. An example of this might be web site addresses, where aliases are incredibly convenient.

    - What did you learn about system design from this class? In particular, describe how each of the basic techniques (abstraction, layering, hierarchy, and modularity) helped you by simplifying your design, making it more efficient, or making it easier to design.

      The modularity was by far the more useful one for me. I really understood the importance of modularity after taking this class. Also, this class really helped me to understand and internalize the abstraction concepts. I also understood layering and hierarchy really good, but not as good as I have understood modularity and abstraction.