

A Web Server: Threads

Objectives

- Learn basic thread concepts and thread synchronization.
- Understand thread-safety and error handling.
- Learn how real-world multi-threaded servers are built.

Specification

This project will modify the webserver previously developed by implementing threads to handle multiple simultaneous client connections instead of processes. This is typical producer-consumer synchronization problem with a single producer (web server) and multiple consumers (web clients).

The most basic thread implementation of a web server is a single-threaded web server. The disadvantage of this type of server is that it does not scale to multiple simultaneous client requests. The next step is a multi-threaded web server. In this case, a worker thread is created for each client connection. This will scale much better than the single-threaded web server but still suffers by the limit on the maximum number of threads for the thread package utilized. If more clients try to connect than threads are available, the server will not be able to satisfy all the requests. In this case, the request should be placed into a waiting queue so it can be serviced when a thread is available.

The next step is a thread-pool approach. In this type of server the main thread will create a pool of worker threads. As the main thread processes connection requests, it creates request objects and places them in a data structure. Each worker thread will retrieve request objects from the data structure according to some scheduling algorithm and respond to the request. The only limit with this type of threaded server is memory.

You will modify the previous web server project by implementing it as a thread-pool web server.

The main thread will:

- Set up the socket
- Create thread pool
- Accept new connections from clients
- Create request structure/object and add it to a request queue
- Assign request objects to worker thread

The worker threads will:

- Read request from request queue
- Process the HTTP request
- Send the HTTP response
- Do any clean up
- Exit the thread

Usage Clause

webserver <port_number>

Notes

- Request queue must be protected with mutex so items are protected.

Comparison

Add a new section to the readme file. This section will be a performance comparison of the process web server and the thread web server. Include the tests you ran to compare the performance as well as the results of each test. Also include a brief summary of your findings.