

Threaded Web Server

Homework 3

POSIX Threads

- IEEE POSIX 1003.1c standard
- Compile with “-pthread”
 - `gcc -o webserver webserver.c -lpthread`
- Tutorial
 - <https://computing.llnl.gov/tutorials/pthreads/>

Creating/Exiting a Thread

```
#include <pthread.h>
```

```
int pthread_create(pthread_t *thread,  
                  const pthread_attr_t *attr,  
                  void *(*start_routine)(void *),  
                  void *arg)
```

```
#include <pthread.h>
```

```
int pthread_exit(void *value_ptr);
```

Threaded Web Server

```
pthread_t thqueue[QUEUELEN];
int connqueue[QUEUELEN];
...
for (;;) {
    i = nconn % QUEUELEN;
    len = sizeof(cliaddr);
    connfd = accept(listenfd, (struct sockaddr*)&cliaddr, &len);
    connqueue[i] = connfd;
    pthread_create(&thqueue[i], NULL, conn_handler, &connqueue[i]);
    nconn++;
}
pthread_exit(NULL);
```

Connection Handler

```
void *conn_handler(void *args) {
    int connfd = *((int *)args);
    read_request(connfd); /* read until "\r\n\r\n" */
    write(connfd, "200 OK HTTP/1.0\r\n\r\n"
        "That's it. You're welcome.", 19+27);
    close(connfd);
    pthread_exit(NULL);
}
```

What to Turn In

- Source code for two servers: forking (concurrent.c) and threaded (thread.c)
- Your environment info (CPU, Memory, OS)
- Graph for the forking and threaded servers
 - Throughput vs. # of connections
 - 1–2500 connections