Project 2

Name: Fei Gao

Structure

The data structure is as follow:

```
typedef struct tpool_work {
        void*               (*routine)(void*);
        void                *arg;
        struct tpool_work   *next;
}tpool_work_t;

typedef struct tpool {
        int             shutdown;
        int             max_thr_num;
        pthread_t       *thr_id;
        tpool_work_t    *queue_head;
        pthread_mutex_t queue_lock;
        pthread_cond_t  queue_ready;
}tpool_t;
```

And I write functions named `tpool_create` `tpool_destory` `tpool_add_work` to achieve the operation of thread pool. And using `mutex_lock` `mutex_unlock` to handle.

When the server launches, the main thread creates[numThread] worker threads and one listener thread.

```
void thread_control()
{
    if(tpool_create(numThread) != 0){
        printf("threadPool create error!\n");
        exit(1);
    }

    pthread_t cThread;
    pthread_create(&cThread, NULL, listener, NULL);
    pthread_join(cThread,NULL);

}
```

When the request arrives, the listener thread accepts a new connection and places the connection into a linked list. Any available worker thread pulls a connection from the linked list , receives a request from the connection, produces the response and sends it to the client.

To maintain N worker threads in the thread pool,

```
for(i =0 ;i < numThread; i++){

    if(pthread_tryjoin_np(*(&tpool->thr_id[i]), NULL) == 0){

        if (pthread_create(*(&tpool->thr_id[i]), NULL, thread_routine, NULL) != 0){
        printf("%s:pthread_create failed, errno:%d, error:%s\n", __FUNCTION__,
        errno, strerror(errno));
        exit(1);
        }
    else
        printf("A new thread has been created successfully to maintain threadPool");
    }
}
```

using `pthread_tryjoin_np` to achieve non-blocking join. when some threads terminated, create a new one.

Result

Typing `./webserver_multi 2003 20` in one terminal And load htmlget_multi.c in another terminal. The output is as follow:

```
[pid 13071, tid 12] (from 127.0.0.1:43934) URL: GET / HTTP/1.0
[pid 13071, tid 10] Received a request from 127.0.0.1:43940
[pid 13071, tid 21] Received a request from 127.0.0.1:43944
[pid 13071, tid 6] Received a request from 127.0.0.1:43942
[pid 13071, tid 10] (from 127.0.0.1:43940) URL: GET / HTTP/1.0
[pid 13071, tid 6] (from 127.0.0.1:43942) URL: GET / HTTP/1.0
[pid 13071, tid 21] Received a request from 127.0.0.1:43946
Thread [pid 13071, tid 15] terminated!
[pid 13071, tid 100] Received a request from 127.0.0.1:43946
[pid 13071, tid 5] Reply: SUCCEED
[pid 13071, tid 12] Reply: SUCCEED
[pid 13071, tid 4] Reply: SUCCEED
[pid 13071, tid 10] Reply: SUCCEED
[pid 13071, tid 6] Reply: SUCCEED
[pid 13071, tid 100] (from 127.0.0.1:43946) URL: GET / HTTP/1.0
[pid 13071, tid 100] Reply: SUCCEED
A new thread has been created successfully to maintain threadPool
[pid 13071, tid 21] Received a request from 127.0.0.1:43948
[pid 13071, tid 21] Received a request from 127.0.0.1:43950
[pid 13071, tid 102] Received a request from 127.0.0.1:43948
[pid 13071, tid 21] Received a request from 127.0.0.1:43952
[pid 13071, tid 17] Received a request from 127.0.0.1:43950
[pid 13071, tid 21] Received a request from 127.0.0.1:43954
[pid 13071, tid 17] (from 127.0.0.1:43950) URL: GET / HTTP/1.0
```