```
#ifndef THREADPOOL H
#define THREADPOOL H
#include <pthread.h>
#include <semaphore.h>
#include <list>
#include <queue>
using namespace std;
typedef struct _routine {
    void (*fct)(void *);
    void *arg;
} routine;
typedef struct _executor_arg {
    queue<routine> waiting; // Routines en attente d'exécution
    // Verrouille et patiente sur la liste des routines en attente
    sem t sem_waiting;
    pthread mutex t mutex waiting;
} executor args;
class ThreadPool
private:
    list<pthread t> _threads;
    executor args _queue;
    ThreadPool(const ThreadPool& other) {};
public:
    ThreadPool(const int n_threads);
    virtual ~ThreadPool();
    virtual ThreadPool& operator=(const ThreadPool& other);
    void inject(void (*fct)(void *), void *arg);
};
#endif // THREADPOOL H
```