#include "include/concurrency\_layer.h"

int main(int argc, char \* argv[]){

int numberOfThreads=8;

pthread\_t tid[numberOfThreads];

stock\_market market\_madrid;

int exit = 0;

pthread\_mutex\_t exit\_mutex;

// Init market and concurrency mechanisms

init\_market(&market\_madrid, "stocks.txt");

init\_concurrency\_mechanisms();

pthread\_mutex\_init(&exit\_mutex,NULL);

// Init broker\_info structure for the broker thread

int numberOfBrokers=5;

broker\_info info\_b1;

strcpy(info\_b1.batch\_file, "batch\_operations.txt");

info\_b1.market = &market\_madrid;

broker\_info info\_b2;

strcpy(info\_b2.batch\_file, "batch\_operations2.txt");

info\_b2.market = &market\_madrid;

broker\_info info\_b3;

strcpy(info\_b3.batch\_file, "batch\_operations2.txt");

info\_b3.market = &market\_madrid;

broker\_info info\_b4;

strcpy(info\_b4.batch\_file, "batch\_operations.txt");

info\_b4.market = &market\_madrid;

broker\_info info\_b5;

strcpy(info\_b5.batch\_file, "batch\_operations2.txt");

info\_b5.market = &market\_madrid;

// Init exec\_info structure for the operation\_executer thread

exec\_info info\_ex1;

info\_ex1.market = &market\_madrid;

info\_ex1.exit = &exit;

info\_ex1.exit\_mutex = &exit\_mutex;

// Init reader\_info for the stats\_reader thread

reader\_info info\_re1;

info\_re1.market = &market\_madrid;

info\_re1.exit = &exit;

info\_re1.exit\_mutex = &exit\_mutex;

info\_re1.frequency = 100000;

reader\_info info\_re2;

info\_re2.market = &market\_madrid;

info\_re2.exit = &exit;

info\_re2.exit\_mutex = &exit\_mutex;

info\_re2.frequency = 100000;

// Create broker and exec threads

pthread\_create(&(tid[0]), NULL, &broker, (void\*) &info\_b1);

pthread\_create(&(tid[1]), NULL, &broker, (void\*) &info\_b2);

pthread\_create(&(tid[2]), NULL, &broker, (void\*) &info\_b3);

pthread\_create(&(tid[3]), NULL, &broker, (void\*) &info\_b4);

pthread\_create(&(tid[4]), NULL, &broker, (void\*) &info\_b5);

pthread\_create(&(tid[5]), NULL, &operation\_executer, (void\*) &info\_ex1);

pthread\_create(&(tid[6]), NULL, &stats\_reader, (void\*) &info\_re1);

pthread\_create(&(tid[7]), NULL, &stats\_reader, (void\*) &info\_re2);

// Join broker threads

void \* res;

for(int ii=0; ii<numberOfBrokers; ii++){

pthread\_join(tid[ii],&res);

}

// Put exit flag = 1 after brokers completion

pthread\_mutex\_lock(&exit\_mutex);

exit = 1;

pthread\_mutex\_unlock(&exit\_mutex);

// Join the rest of the threads

for(int ii=numberOfBrokers; ii<numberOfThreads; ii++){

pthread\_join(tid[ii],&res);

}

// Print final statistics of the market

print\_market\_status(&market\_madrid);

// Destroy market and concurrency mechanisms

delete\_market(&market\_madrid);

destroy\_concurrency\_mechanisms();

pthread\_mutex\_destroy(&exit\_mutex);

return 0;

}