#include <pthread.h>

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#define PING 0

#define PONG 1

volatile unsigned NumRounds = 0;

volatile unsigned PrevVal = PONG;

pthread\_mutex\_t mutex1 = PTHREAD\_MUTEX\_INITIALIZER;

pthread\_cond\_t cond = PTHREAD\_COND\_INITIALIZER;

char \*Message[2] = {"PING", "PONG"};

/\*

\* pring the message, either "PING" or "PONG".

\* also checks to make sure that the correct message is being printed

\*/

void pingpongprint(int thisval)

{

/\* if the threads are taking turns then PrevVal should always be different from thisval \*/

if (PrevVal == thisval) {

fprintf(stderr, "ERROR: received '%s' but expected '%s'\n", Message[thisval], Message[!thisval]);

exit(-1);

}

printf("%s\n", Message[thisval]);

PrevVal = thisval;

}

void \*PingerPonger(void \*tidptr)

{

int tid = \*((int \*) tidptr);

for (unsigned dex = 0; dex <= NumRounds; dex++) {

pthread\_mutex\_lock(&mutex1);

while( PrevVal == tid)

{

pthread\_cond\_wait(&cond, &mutex1);

}

pingpongprint(tid);

PrevVal = tid;

pthread\_cond\_signal(&cond);

pthread\_mutex\_unlock(&mutex1);

}

return NULL;

}

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

{

if (argc != 2) {

fprintf(stderr, "USAGE: %s <numrounds>\n", argv[0]);

exit(-1);

}

NumRounds = atoi(argv[1]);

if (NumRounds < 1) {

fprintf(stderr, "ERROR: NumRounds must be >= 1\n");

exit(-1);

}

int ping\_tid = PING;

int pong\_tid = PONG;

pthread\_t pingthread;

pthread\_t pongthread;

int rc;

rc = pthread\_create(&pingthread, NULL, PingerPonger, (void \*) &ping\_tid);

if (rc) {

fprintf(stderr, "ERROR; could not create PING thread. return code from pthread\_create() is %d\n", rc);

exit(-1);

}

rc = pthread\_create(&pongthread, NULL, PingerPonger, (void \*) &pong\_tid);

if (rc) {

fprintf(stderr, "ERROR; could not create PONG thread. return code from pthread\_create() is %d\n", rc);

exit(-1);

}

rc = pthread\_join(pingthread, NULL);

if (rc != 0) {

fprintf(stderr, "ERROR joining with PING (rc==%d)\n", rc);

exit(-1);

}

rc = pthread\_join(pongthread, NULL);

if (rc != 0) {

fprintf(stderr, "ERROR joining with PONG (rc==%d)\n", rc);

exit(-1);

}

printf("SUCCESS! (parent exiting)\n");

return(0);

}