|  |  |  |  |
| --- | --- | --- | --- |
| Weber State University |  |  |  |

River Schenck

CS 3100 Professor Ball

Homework 7- Pthreads and Sloppy Counter Report

Table of Contents

No table of contents entries found.

Main Code:

-Most of the code was given in threadExample4.c supplied to us by Dr. Ball.

#include <stdio.h>

#include <stdlib.h>

#include <assert.h>

#include <pthread.h>

#include <time.h>

static volatile int counter = 0;

int threshold;

double time\_taken;

pthread\_mutex\_t lock;

void \*mythread(void \*arg) {

int i;

int localCounter = 0;

for (i = 0; i < 1000000; i++)

{

localCounter++;

if (localCounter == threshold) {

pthread\_mutex\_lock(&lock);

counter = counter + threshold;

pthread\_mutex\_unlock(&lock);

localCounter = 0;

}

}

return NULL;

}

int

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

{

if (argc < 2)

{

printf("usage: name number\n");

return 0;

}

threshold = atoi(argv[1]);

pthread\_t p1,p2,p3,p4,p5;

int pres = pthread\_mutex\_init(&lock, NULL);

int rc;

//DELETED ALL RC

printf("threshold #: %d\n", threshold);

clock\_t t; //THIS IS TO RECORD TIME

t = clock();

rc = pthread\_create(&p1, NULL, mythread, "A");

rc = pthread\_create(&p2, NULL, mythread, "B");

rc = pthread\_create(&p3, NULL, mythread, "C");

rc = pthread\_create(&p4, NULL, mythread, "D");

rc = pthread\_create(&p5, NULL, mythread, "E");

rc = pthread\_join(p1, NULL);

rc = pthread\_join(p2, NULL);

rc = pthread\_join(p3, NULL);

rc = pthread\_join(p4, NULL);

rc = pthread\_join(p5, NULL);

t = clock() - t;

time\_taken = ((double)t) / CLOCKS\_PER\_SEC;

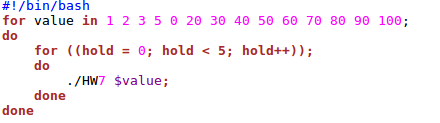
printf("processed time: %f seconds\n\n", time\_taken);

return 0;

}

Script:

-I was able to find how to make a script in Linux to run the program 65 times.



-To run this script I used bash.

