CST 334: Operating Systems

Dr. Glenn Bruns

# Lab: condition variables - solution

This is one solution to the condition variables lab:

/\*

\* reading and writing to a simple single cell buffer

\* Readers must wait until the buffer is non-empty.

\* Writers must wait until the buffer is empty.

\*

\*/

#include <pthread.h>

#include <stdio.h>

#include <stdlib.h>

#include <assert.h>

// state variables

int buf;

int count = 0;

// sync variables

pthread\_mutex\_t lock;

pthread\_cond\_t read\_go;

pthread\_cond\_t write\_go;

void \*reader(void \*arg) {

int i;

int val;

for (i = 0; i < 100; i++) {

pthread\_mutex\_lock(&lock);

while (count != 1) {

pthread\_cond\_wait(&read\_go, &lock);

}

count = 0;

val = buf;

printf("R: %d\n", val);

pthread\_cond\_signal(&write\_go);

pthread\_mutex\_unlock(&lock)

}

}

void \*writer(void \*arg) {

int i;

for (i = 0; i < 100; i++) {

pthread\_mutex\_lock(&lock);

while (count != 0) {

pthread\_cond\_wait(&write\_go, &lock);

}

count = 1;

buf = i;

printf("W: %d\n", i);

pthread\_cond\_signal(&read\_go);

pthread\_mutex\_unlock(&lock);

}

}

// run reader and writer

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

pthread\_t p1, p2;

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

pthread\_cond\_init(&read\_go, NULL);

pthread\_cond\_init(&write\_go, NULL);

int rc;

rc = pthread\_create(&p1, NULL, reader, NULL);

assert(rc == 0);

rc = pthread\_create(&p2, NULL, writer, NULL);

assert(rc == 0);

pthread\_join(p1, NULL);

pthread\_join(p2, NULL);

}