#include <stdio.h>

#include <stdlib.h>

#include <pthread.h>

#include <semaphore.h>

#include <unistd.h> // for sleep()

#define R 5

#define W 5

int readcount;

int writecount;

pthread\_mutex\_t x;

sem\_t wsem, y, rsem, z;

int clk[2] = {0, 0};

void \*reader1(void \*a);

void \*writer1(void \*a);

int main() {

int i, op;

pthread\_t thread\_read[R], thread\_write[W];

// Initialize semaphores and mutex

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

sem\_init(&wsem, 0, 1);

sem\_init(&y, 0, 1);

sem\_init(&rsem, 0, 1);

sem\_init(&z, 0, 1);

do {

printf("Menu:\n 1. Readers have priority\n 2. Exit\n");

scanf("%d", &op);

switch (op) {

case 1:

readcount = 0;

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

pthread\_create(&thread\_write[i], NULL, writer1, (void \*)i);

}

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

pthread\_create(&thread\_read[i], NULL, reader1, (void \*)i);

}

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

pthread\_join(thread\_write[i], NULL);

}

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

pthread\_join(thread\_read[i], NULL);

}

break;

case 2:

break;

default:

printf("Invalid option. Please try again.\n");

break;

}

} while (op != 2);

// Cleanup

pthread\_mutex\_destroy(&x);

sem\_destroy(&wsem);

sem\_destroy(&y);

sem\_destroy(&rsem);

sem\_destroy(&z);

return 0;

}

void \*reader1(void \*a) {

int r = (int)a;

int i = 0;

while (i < 5) {

// Simulate a delay

sleep(rand() % 10);

pthread\_mutex\_lock(&x);

readcount++;

if (readcount == 1)

sem\_wait(&wsem);

pthread\_mutex\_unlock(&x);

// Reading

printf("\t\tReader %d is reading: %d hrs %d mins\n", r, clk[0], clk[1]);

pthread\_mutex\_lock(&x);

readcount--;

if (readcount == 0)

sem\_post(&wsem);

pthread\_mutex\_unlock(&x);

sleep(rand() % 10);

i++;

}

return NULL;

}

void \*writer1(void \*a) {

int w = (int)a;

int i = 0;

while (i < 2) {

// Simulate a delay

sleep(rand() % 10);

sem\_wait(&wsem);

// Writing

clk[1] += 20;

if (clk[1] > 59) {

clk[0]++;

clk[1] -= 60;

}

printf("Writer %d is writing: %d hrs %d mins\n", w, clk[0], clk[1]);

sem\_post(&wsem);

sleep(rand() % 10);

i++;

}

return NULL;

}

Output:

Menu:

1. Readers have priority

2. Exit

1

Reader 0 is reading: 0 hrs 0 mins

Reader 1 is reading: 0 hrs 0 mins

Writer 0 is writing: 0 hrs 20 mins

Reader 2 is reading: 0 hrs 20 mins

Reader 3 is reading: 0 hrs 20 mins

Reader 4 is reading: 0 hrs 20 mins

Writer 1 is writing: 0 hrs 40 mins

Reader 0 is reading: 0 hrs 40 mins

Reader 1 is reading: 0 hrs 40 mins

Writer 0 is writing: 0 hrs 60 mins

Writer 1 is writing: 1 hrs 20 mins

Reader 2 is reading: 1 hrs 20 mins

Reader 3 is reading: 1 hrs 20 mins

Writer 0 is writing: 1 hrs 40 mins

Writer 1 is writing: 2 hrs 0 mins

Reader 4 is reading: 2 hrs 0 mins