#include <pthread.h>

#include <semaphore.h>

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <time.h>

#define NUM\_PHILOSOPHERS 5

#define MEALS\_LIMIT 3

sem\_t chopsticks[NUM\_PHILOSOPHERS];

sem\_t waiter;

void\* philosopher(void\* num);

void think(int philosopher\_number);

void eat(int philosopher\_number, int meal\_number);

int main() {

pthread\_t philosophers[NUM\_PHILOSOPHERS];

int philosopher\_numbers[NUM\_PHILOSOPHERS];

for (int i = 0; i < NUM\_PHILOSOPHERS; i++) {

sem\_init(&chopsticks[i], 0, 1);

}

sem\_init(&waiter, 0, NUM\_PHILOSOPHERS - 1);

for (int i = 0; i < NUM\_PHILOSOPHERS; i++) {

philosopher\_numbers[i] = i;

pthread\_create(&philosophers[i], NULL, philosopher, &philosopher\_numbers[i]);

}

for (int i = 0; i < NUM\_PHILOSOPHERS; i++) {

pthread\_join(philosophers[i], NULL);

}

for (int i = 0; i < NUM\_PHILOSOPHERS; i++) {

sem\_destroy(&chopsticks[i]);

}

sem\_destroy(&waiter);

return 0;

}

void\* philosopher(void\* num) {

int philosopher\_number = \*(int\*)num;

int meals\_eaten = 0;

while (meals\_eaten < MEALS\_LIMIT) {

think(philosopher\_number);

sem\_wait(&waiter);

sem\_wait(&chopsticks[philosopher\_number]);

printf("Philosopher %d picked up left chopstick %d.\n", philosopher\_number, philosopher\_number);

sem\_wait(&chopsticks[(philosopher\_number + 1) % NUM\_PHILOSOPHERS]);

printf("Philosopher %d picked up right chopstick %d.\n", philosopher\_number, (philosopher\_number + 1) % NUM\_PHILOSOPHERS);

eat(philosopher\_number, meals\_eaten + 1);

meals\_eaten++;

sem\_post(&chopsticks[philosopher\_number]);

printf("Philosopher %d put down left chopstick %d.\n", philosopher\_number, philosopher\_number);

sem\_post(&chopsticks[(philosopher\_number + 1) % NUM\_PHILOSOPHERS]);

printf("Philosopher %d put down right chopstick %d.\n", philosopher\_number, (philosopher\_number + 1) % NUM\_PHILOSOPHERS);

sem\_post(&waiter);

}

return NULL;

}

void think(int philosopher\_number) {

printf("Philosopher %d is thinking.\n", philosopher\_number);

sleep(rand() % 3 + 1);

void eat(int philosopher\_number, int meal\_number) {

printf("Philosopher %d is eating. Meal number: %d\n", philosopher\_number, meal\_number);

sleep(rand() % 2 + 1);

}