Sleeping Barber

written by Nik Tsonev

Intro

A barbershop has a barber, a barber chair, and a waiting room with a limited number of chairs. When no customers are present, the barber sleeps. When a customer arrives, they wake the barber or, if the barber is busy, wait for an empty chair. If all chairs are full, the customer leaves.

Solution

so we have 3 semaphores

1. if the barber is ready to cut
2. if the customer is ready
3. if the waiting room can be changed

so the barber is always waiting and first checks if the customer is ready, then checks if the waiting room is so that he can decrement the amount of people waiting

then he cuts hair and notifies that he is ready for the next customer and that waiting room will not be modified.

the customers thread waits if he can access the waiting room in, then if there is space, he increments the room by 1 and signals that he is ready and that the waiting room will not be modified, then he waits for the barber semaphore to be ready

```
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#include <semaphore.h>
#include <unistd.h>
```

```
#define NUM_CHAIRS 3

sem_t barber_ready;
sem_t cust_ready;
sem_t access_waiting_room;
int waiting_cust = 0'

void* barber(void *arg) {
        (void) arg;

        while (1) {
                sem_wait(&cust_ready);
                sem_wait(&access_waiting_room);
                waiting_cust--;
                printf("Cutting hair\n");
                sem_post(&barber_ready);
                sem_post(&access_waiting_room);
        }
        return NULL;
}

void* customer(void *arg) {
        int id = *(int*) arg;
        sem_wait(&access_waiting_room);
        if (waiting_cust < NUM_CHAIRS) {
                wait_cust++;
                printf("cust %d is waiting\n", id);
                sem_post(&access_waiting_room);
                sem_post(&cust_ready);
                sem_wait(&barber_ready);
        } else {
                printf("no spacer - customer leaveing\n");
        }
        return NULL
}
```