



Problema do Barbeiro

```
#include "stdio.h"
#include "unistd.h"
#include "stdlib.h"
#include "pthread.h"
#include "semaphore.h"
#define CHAIRS 5                /* número de cadeiras para os clientes à espera */
/*
#define TRUE 1
sem_t customers;               /* número de cliente à espera de atendimento */
sem_t barbers;                 /* número de barbeiros à espera de clientes */
sem_t mutex;                   /* para exclusão mútua */
int waiting = 0;               /* clientes que estão esperando (não estão
cortando) */
/* protótipos */
void* barber(void *arg);
void* customer(void *arg);
void cut_hair();
void customer_arrived();
void get_haircut();
void giveup_haircut();
int main() {
sem_init(&customers, TRUE, 0);
sem_init(&barbers, TRUE, 0);
sem_init(&mutex, TRUE, 1);
pthread_t b, c;
/* criando único barbeiro */
pthread_create(&b, NULL, (void *) barber, NULL);
/* criação indefinida de clientes */
while(TRUE) {
pthread_create(&c, NULL, (void *) customer, NULL);
sleep(1);
}
return 0;
}
void* barber(void *arg) {
while(TRUE) {
sem_wait(&customers); /* vai dormir se o número de clientes for 0 */
sem_wait(&mutex); /* obtém acesso a 'waiting' */
waiting = waiting - 1; /* decresce de um o contador de clientes à espera */
sem_post(&barbers); /* um barbeiro está agora pronto para cortar cabelo */
sem_post(&mutex); /* libera 'waiting' */
cut_hair(); /* corta o cabelo (fora da região crítica) */
}
pthread_exit(NULL);
}
void* customer(void *arg) {
sem_wait(&mutex); /* entra na região crítica */
if(waiting < CHAIRS) { /* se não houver cadeiras vazias, saia */
customer_arrived();
waiting = waiting + 1; /* incrementa o contador de clientes à espera */
sem_post(&customers); /* acorda o barbeiro se necessário */
sem_post(&mutex); /* libera o acesso a 'waiting' */
sem_wait(&barbers); /* vai dormir se o número de barbeiros livres for 0 */
get_haircut(); /* sentado e sendo servido */
}
```



UNIVERSIDADE FEDERAL DE RORAIMA
PROFESSOR (A) : [Herbert Oliveira Rocha](#)
DISCIPLINA DE SISTEMAS OPERACIONAIS.
ALUNO : Ibukun Chife didier Adjitche

```
} else {  
sem_post(&mutex);           /* a barbearia está cheia; não espera */  
giveup_haircut();  
}  
pthread_exit(NULL);  
}  
void cut_hair() {  
printf("Barbeiro estah cortando o cabelo de alguem!\n");  
sleep(3);  
}  
void customer_arrived() {  
printf("Cliente chegou para cortar cabelo!\n");  
}  
void get_haircut() {  
printf("Cliente estah tendo o cabelo cortado!\n");  
}  
void giveup_haircut() {  
printf("Cliente desistiu! (O salao estah muito cheio!)\n");  
}
```