

A traditional barber pole with red and blue diagonal stripes and white vertical stripes on the top and bottom. It features a blue ribbon banner across the middle.

BARBEIRO SONOLENTO

Guilherme Miranda
Luiz Gustavo Cavalcante

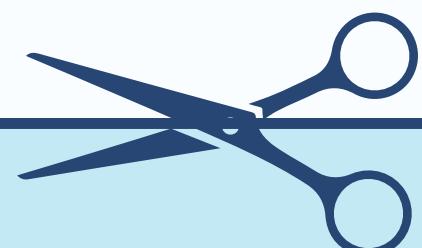
O PROBLEMA

O objetivo do problema é simular uma barbearia com:

- 1 barbeiro
- Cadeiras limitadas na sala de espera
- Clientes chegando em intervalos regulares

A lógica é:

- Se não há clientes, o barbeiro dorme.
- Se um cliente chega e há cadeiras vazias, ele espera.
- Se todas as cadeiras estão ocupadas, o cliente vai embora.



CÓDIGO

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <unistd.h>
4 #include <pthread.h>
5 #include <semaphore.h>
6
7 #define CHAIRS 5
8
9 sem_t customers;
10 sem_t barbers;
11 sem_t mutex;
12
13 int waiting = 0;
14
15 void* barber(void* arg);
16 void* customer(void* arg);
17 void cut_hair();
18 void customer_arrived();
19 void get_haircut();
20 void giveup_haircut();
```

- Importa as bibliotecas necessárias;
- Declara os semáforos e variáveis que serão utilizados;

CÓDIGO

```
22 int main() {
23     pthread_t barber_thread;
24
25     sem_init(&customers, 0, 0);
26     sem_init(&barbers, 0, 0);
27     sem_init(&mutex, 0, 1);
28
29     pthread_create(&barber_thread, NULL, barber, NULL);
30
31     while (1) {
32         pthread_t customer_thread;
33         pthread_create(&customer_thread, NULL, customer, NULL);
34         pthread_detach(customer_thread);
35         sleep(1);
36     }
37
38     pthread_join(barber_thread, NULL);
39     return 0;
40 }
```

- customers: começa em 0 (nenhum cliente esperando);
- barbers: começa em 0 (barbeiro inativo);
- mutex: começa em 1 (controle de acesso à variável waiting);
- Cria uma thread para o barbeiro;
- Um novo cliente é criado a cada segundo. Ele é imediatamente "destacado" (sem join) para que os recursos sejam liberados automaticamente quando a thread termina.

CÓDIGO

```
42 void* barber(void* arg) {  
43     while (1) {  
44         sem_wait(&customers);  
45         sem_wait(&mutex);  
46         waiting--;  
47         sem_post(&barbers);  
48         sem_post(&mutex);  
49         cut_hair();  
50     }  
51     return NULL;  
52 }
```

- O barbeiro:
 1. Espera clientes (sem_wait(&customers)).
 2. Reduz o número de clientes esperando (waiting--).
 3. Libera o barbeiro para o cliente (sem_post(&barbers)).
 4. Corta o cabelo.

CÓDIGO

```
54 void* customer(void* arg) {  
55     sem_wait(&mutex);  
56     if (waiting < CHAIRS) {  
57         customer_arrived();  
58         waiting++;  
59         sem_post(&customers);  
60         sem_post(&mutex);  
61         sem_wait(&barbers);  
62         get_haircut();  
63     } else {  
64         sem_post(&mutex);  
65         giveup_haircut();  
66     }  
67     return NULL;  
68 }
```

- O cliente:
 1. Entra na barbearia e tenta pegar o mutex.
 2. Se houver cadeira disponível, ele:
 - Aumenta waiting
 - Sinaliza sua chegada
 - Espera o barbeiro chamá-lo
 - Recebe o corte de cabelo
 3. Caso contrário, desiste e vai embora.

OBRIGADO
POR ASSISTIR!

Guilherme Miranda
Luiz Gustavo Cavalcante