

Universidade de São Paulo
Escola de Artes Ciências e Humanidades
Sistemas Operacionais (ACH2044)

EP2 - Seção Crítica

Jadno Barbosa

Professora Gisele Craveiro

Sumário

1	Implementação	2
1.1	Inicialização	2
1.2	Seção Crítica e Não-Crítica	2
1.3	Processos	3
1.4	Main	4
2	Execução	4

1 Implementação

1.1 Inicialização

Primeiramente foram importadas as bibliotecas necessárias e declaradas as variáveis globais **count** e **vez**.

```
1 #include <error.h>
2 #include <pthread.h>
3 #include <stdbool.h>
4 #include <stdio.h>
5 #include <unistd.h>
6
7 int count;
8 int vez;
9
```

1.2 Seção Crítica e Não-Crítica

Na seção crítica a variável **count** é incrementada, o id do processo na seção crítica é impresso na tela junto com a variável **count** e o processo dorme por 1 segundo. Na seção não-crítica o id do processo na seção não-crítica é impresso na tela.

```
10 void secao_critica(int id) {
11     count++;
12     printf("Processo %d na secao critica\n", id);
13     printf("\tcount: %d\n", count);
14     sleep(1);
15 }
16
17 void secao_nao_critica(int id) {
18     printf("Processo %d na secao nao critica\n", id);
19 }
20
```

1.3 Processos

```
21 void *p1() {
22     int meu_id = 0;
23     int outro = 1;
24     while (true) {
25         while (vez != meu_id) {
26             };
27         secao_critica(meu_id);
28         vez = outro;
29         secao_nao_critica(meu_id);
30     }
31 }
32
33 void *p2() {
34     int meu_id = 1;
35     int outro = 0;
36     while (true) {
37         while (vez != meu_id) {
38             };
39         secao_critica(meu_id);
40         vez = outro;
41         secao_nao_critica(meu_id);
42     }
43 }
44
```

1.4 Main

Na função main as variáveis globais são inicializadas e as duas threads são criadas e executadas.

```
45 int main() {  
46  
47     count = 0;  
48     vez = 0;  
49  
50     pthread_t filhos[2];  
51     pthread_create(&filhos[0], NULL, p1, NULL);  
52     pthread_create(&filhos[1], NULL, p2, NULL);  
53     pthread_join(filhos[0], NULL);  
54     pthread_join(filhos[1], NULL);  
55  
56     return 0;  
57 }
```

2 Execução

Na saída do programa é possível observar que algoritmo respeita o princípio da exclusão mútua pois só entra um processo por vez na seção crítica. O progresso não é respeitado pois um processo não consegue entrar na seção crítica logo após ter saído dela, mesmo que o outro processo não vá utilizar seção crítica. A espera limitada é respeitada pois um processo nunca ficará esperando indefinidamente para entrar na seção crítica, uma vez que, ao sair de uma seção crítica, um processo imediatamente libera a vez para o outro processo.

```
^ ~/projetos/S0/ep2 .....  
[>> gcc secaocritica.c -lpthread -o secaocritica  
  
^ ~/projetos/S0/ep2 .....  
[>> ./secaocritica  
Processo 0 na secao critica  
      count: 1  
Processo 0 na secao nao critica  
Processo 1 na secao critica  
      count: 2  
Processo 1 na secao nao critica  
Processo 0 na secao critica  
      count: 3  
Processo 0 na secao nao critica  
Processo 1 na secao critica  
      count: 4  
Processo 1 na secao nao critica  
Processo 0 na secao critica  
      count: 5  
Processo 0 na secao nao critica  
Processo 1 na secao critica  
      count: 6
```