

## Disciplina: Sistemas Operacionais I

### Exercício 4

Baseando se nos slides da Aula 12 e 13 e no GitHub do Professor.

1. Elabore um programa ilustrando o uso da estrita alternância.

- a. Explique o código fonte utilizado

Para fazer o processamento é necessário utilizar uma variável global em que os dois processos A e B utilizam. Pode causar lentidão e uso da memória desnecessárias.

```
alternancia.py
1  #Estrita Alternância
2  from threading import Thread
3  import time
4
5  global turn
6
7  def regiaoCritica():
8      time.sleep(1)
9
10 def processamentoA(times, delay):
11     global turn
12     for x in range(times):
13         print ("Secao de Entrada A - ",x+1)
14         while (turn != 0):
15             continue
16         print ("Regiao Critica A")
17         regiaoCritica()
18         print ("Secao de Saida A")
19         turn = 1
20         print ("Regiao nao critica A\n")
21         time.sleep(delay)
22
23 def processamentoB(times, delay):
24     global turn
25     for x in range(times):
26         print ("Secao de Entrada B - ",x+1)
27         while (turn != 1):
28             continue
29         print ("Regiao Critica B")
30         regiaoCritica()
31         print ("Secao de Saida B")
32         turn = 0
33         print ("Regiao nao critica B\n")
34         time.sleep(delay)
35
36
37 print ("Exemplo de Estrita Alternancia")
38 execTimes = 4
39 turn = 0
40
41 #no processamento você pode passar quantas vezes que a exec e
42 #qual o tempo de delay para simular o efeito comboio
43 tA = Thread(target=processamentoA, args=(execTimes,1,))
44 tA.start()
45 tB = Thread(target=processamentoB, args=(execTimes,5,))
46 tB.start()
```

b. Descreva os resultados obtidos

O Efeito Comboio é visualizado mais facilmente, pois o processo A fica esperando a finalização do processo B(que é mais lento), causando uma demora no processo geral.

```
ariana@arcursino:~/projetos/FATEC/so/ex4$ python3 alternancia.py
Exemplo de Estrita Alternancia
Secao de Entrada A - 1
Regiao Critica A
Secao de Entrada B - 1
Secao de Saida A
Regiao nao critica A

Regiao Critica B
Secao de Entrada A - 2
Secao de Saida B
Regiao nao critica B

Regiao Critica A
Secao de Saida A
Regiao nao critica A

Secao de Entrada A - 3
Secao de Entrada B - 2
Regiao Critica B
Secao de Saida B
Regiao nao critica B
Regiao Critica A

Secao de Saida A
Regiao nao critica A

Secao de Entrada A - 4
Secao de Entrada B - 3
Regiao Critica B
Secao de Saida B
Regiao nao critica B

Regiao Critica A
Secao de Saida A
Regiao nao critica A

Secao de Entrada B - 4
Regiao Critica B
Secao de Saida B
Regiao nao critica B
```

2. Elabore um programa ilustrando o uso da Solução de Peterson.

a. Explique o código fonte utilizado

Também é necessário utilizar uma variável global em que os dois processos A e B utilizam. Pode causar lentidão e uso da memória desnecessárias.

```
peterson.py
1  #Solução de Peterson
2  from threading import Thread
3  import time
4
5  global turn, i, j, flag
6
7  def regiaoCritica():
8      time.sleep(1)
9
10 def processamentoA(times, delay):
11     global turn, i, j, flag
12     for x in range(times):
13         print ("Secao de Entrada A - ",x+1)
14         flag[i] = True
15         turn = j
16         while (flag[j] and turn == j):
17             continue
18         print ("Regiao Critica A")
19         regiaoCritica()
20         print ("Secao de Saida A")
21         flag[i] = False
22         print ("Regiao nao critica A\n")
23         time.sleep(delay)
24
25 def processamentoB(times, delay):
26     global turn, i, j, flag
27     for x in range(times):
28         print ("Secao de Entrada B - ",x+1)
29         flag[j] = True
30         turn = i
31         while (flag[i] and turn == i):
32             continue
33         print ("Regiao Critica B")
34         regiaoCritica()
35         print ("Secao de Saida B")
36         flag[j] = False
37         print ("Regiao nao critica B\n")
38         time.sleep(delay)
39
40
41 print ("Exemplo de Solucao de Peterson")
42 execTimes = 5
43 turn = 0
44 i = 0
45 j = 1
46 flag = []
47 flag.append(False)
48 flag.append(False)
49
50 #no processamento você pode passar quantas vezes que a exec e
51 #qual o tempo de delay para simular o efeito comboio
52 tA = Thread(target=processamentoA, args=(execTimes,1))
53 tA.start()
54 tB = Thread(target=processamentoB, args=(execTimes,5))
55 tB.start()
```

b. Descreva os resultados obtidos

Com a Solução de Peterson, é possível observar que o processo A não fica esperando o processo B terminar e ele evolui.

```
ariana@arcursino:~/projetos/FATEC/so/ex4$ python3 peterson.py
Exemplo de Solucao de Peterson
Secao de Entrada A - 1
Regiao Critica A
Secao de Entrada B - 1
Secao de Saida A
Regiao nao critica A
Regiao Critica B

Secao de Saida B
Secao de Entrada A - 2
Regiao nao critica B

Regiao Critica A
Secao de Saida A
Regiao nao critica A

Secao de Entrada A - 3
Regiao Critica A
Secao de Saida A
Regiao nao critica A

Secao de Entrada A - 4
Regiao Critica A
Secao de Entrada B - 2
Secao de Saida A
Regiao nao critica A

Regiao Critica B
Secao de Saida B
Regiao nao critica B

Secao de Entrada A - 5
Regiao Critica A
Secao de Saida A
Regiao nao critica A

Secao de Entrada B - 3
Regiao Critica B
Secao de Saida B
Regiao nao critica B

Secao de Entrada B - 4
Regiao Critica B
Secao de Saida B
Regiao nao critica B

Secao de Entrada B - 5
Regiao Critica B
Secao de Saida B
Regiao nao critica B
```

3. Elabore um programa ilustrando a utilização de semáforo.

a. Explique o código fonte utilizado

Como é importado o Semaphore da biblioteca threading, o próprio SO já faz o gerenciamento das entradas e saídas das regiões críticas.

```
1  #Semáforo
2  from threading import Thread, Semaphore
3  import time
4
5  s = Semaphore()
6
7  def regiaoCritica():
8      time.sleep(1)
9
10 def processamentoA(times, delay):
11     for x in range(times):
12         print ("Secao de Entrada A - ", x+1)
13         s.acquire()
14         print ("Regiao Critica A")
15         regiaoCritica()
16         print ("Secao de Saida A")
17         s.release()
18         print ("Regiao nao critica A\n")
19         time.sleep(delay)
20
21 def processamentoB(times, delay):
22     for x in range(times):
23         print ("Secao de Entrada B - ", x+1)
24         s.acquire()
25         print ("Regiao Critica B")
26         regiaoCritica()
27         print ("Secao de Saida B")
28         s.release()
29         print ("Regiao nao critica B\n")
30         time.sleep(delay)
31
32
33 print ("Exemplo de Semafaro")
34 execTimes = 5
35
36 #no processamento você pode passar quantas vezes que a exec e
37 #qual o tempo de delay para simular o efeito comboio
38 tA = Thread(target=processamentoA, args=(execTimes, 1))
39 tA.start()
40 tB = Thread(target=processamentoB, args=(execTimes, 5))
41 tB.start()
```

b. Descreva os resultados obtidos

Assim como a Solução de Peterson, é possível observar que o processo A não fica esperando o processo B terminar e ele evolui.

```
ariana@arcursino:~/projetos/FATEC/so/ex4$ python3 semaforo.py
Exemplo de Semafaro
Secao de Entrada A - 1
Regiao Critica A
Secao de Entrada B - 1
Secao de Saida A
Regiao nao critica A
Regiao Critica B

Secao de Saida B
Regiao nao critica B

Secao de Entrada A - 2
Regiao Critica A
Secao de Saida A
Regiao nao critica A

Secao de Entrada A - 3
Regiao Critica A
Secao de Saida A
Regiao nao critica A

Secao de Entrada A - 4
Regiao Critica A
Secao de Entrada B - 2
Secao de Saida A
Regiao nao critica A

Regiao Critica B
Secao de Saida B
Regiao nao critica B

Secao de Entrada A - 5
Regiao Critica A
Secao de Saida A
Regiao nao critica A

Secao de Entrada B - 3
Regiao Critica B
Secao de Saida B
Regiao nao critica B

Secao de Entrada B - 4
Regiao Critica B
Secao de Saida B
Regiao nao critica B

Secao de Entrada B - 5
Regiao Critica B
Secao de Saida B
Regiao nao critica B
```