



UFOP

Universidade Federal de Ouro Preto – UFOP
Instituto de Ciências Exatas e Biológicas – ICEB
Departamento de Computação – DECOM

Programação de Sistemas em Tempo Real
Prof. Charles Garrocho

Exercício de Sincronização de Processos

Cada aluno irá resolver o problema de seção crítica, do código abaixo (Figura 1), utilizando ALGORITMO 1, 2, 3 e semáforos. Cada aluno deverá montar um relatório explicando as implementações e quais requisitos é satisfeito em busca de solucionar o problema de seção crítica.

Entenda que nem todos os algoritmos satisfazem todos os requisitos de seção crítica, quero que você explique por que alguns algoritmos satisfaz e outros não. Utilizar 2 threads nos três primeiros algoritmos e 10 threads em semáforos. Na implementação de semáforos deve satisfazer todos os requisitos de seção crítica.

```
1  from threading import Thread
2  from time import sleep
3
4  contador = 0
5
6  class Processo(Thread):
7
8      def __init__(self, p):
9          Thread.__init__(self)
10         self.p = p
11
12     def run(self):
13         global contador
14         while True:
15             print "processo: {0}, entrou na região crítica, contador: {1}".format(self.p, contador)
16             contador = contador + 1
17             print "processo: {0}, saiu da região crítica, contador: {1}".format(self.p, contador)
18
19
20 p1 = Processo(0)
21 p2 = Processo(1)
22
23 p1.start()
24 p2.start()
```

Figura 1: Código com seção crítica (linha 15, entrada, linha 17, saída).

Obs. 1: Exercícios com códigos e relatórios iguais levaram nota zero.

Obs. 2: Imprimir o relatório e entregar no dia da apresentação (21/11/2014).