

PCD - Quiz 6

O e-mail do participante (**igor.ribeiro@unifesp.br**) foi registrado durante o envio deste formulário.

1. Um monitor é um tipo de _____ *

- ☐ Semáforo
- ☐ Mecanismo de sincronização de baixo nível
- ☒ Mecanismo de sincronização de alto nível
- ☐ NDA

2. Um procedimento definido em um _____ pode acessar apenas variáveis definidas localmente dentro do _____, além de seus próprios argumentos e parâmetros. *

- ☐ processo, semáforo
- ☒ processo, monitor
- ☐ semáforo, semáforo
- ☐ monitor, monitor

Resposta correta

- ☒ monitor, monitor



3. Um código monitor assegura que _____ *

- ☐ Apenas um processo pode executar, por vez, o código definido no próprio procedimento
- ☐ Uma quantidade n de processos podem estar ativos dentro do código monitor
- ☒ Exista exclusão mútua, mas sem garantia de deadlocks
- ☐ N.D.A.

Resposta correta

- ☒ Apenas um processo pode executar, por vez, o código definido no próprio procedimento

4. Quais são as operações que normalmente são associadas a variáveis de condição? *

- ☒ wait & notify
- ☐ hold & wait
- ☐ continue & hold
- ☐ break & signal

5. O que acontece quando a operação de wait é invocada em uma variável de condição? *

- ☐ libera um processo recém bloqueado
- ☐ Permite que n processos entrem dentro de uma região protegida
- ☒ Suspende o processo até que outro processo invoque uma operação de signal/notify
- ☐ N.D.A.



6. Se nenhum processo está em suspensão, a operação signal/notify causa o seguinte efeito: *

- ☐ Põe o sistema em um estado de deadlock
- ☐ Suspende o processo default
- ☒ Nada ocorre
- ☐ É impossível prever o que ocorre



7. Qual a saída do código abaixo? *

<pre> Monitor funcoes string Last = "B2" condition A1,A2,B1,B2 operation A1() if (last!="B2") waitC(A1) last = "A1" print last signalC(B1) operation B1() if (last!="A1") waitC(B1) last = "B1" print last signalC(A2) operation A2() if (last!="B1") waitC(A2) last = "A2" print last signalC(B2) operation B2() if (last!="A2") waitC(B2) last = "B2" print last signalC(A1) </pre>	
Thread 1	Thread 2
funcoes.A1() funcoes.A2()	funcoes.B1() funcoes.B2()

- ☐ A1, A2, B1, B2
- ☒ A1, B1, A2, B2
- ☐ B1, B2, A1, A2
- ☐ N.D.A.



8. Qual a saída do código abaixo? *

```
Monitor funcoes
string Last = "B2"
condition A1,A2,B1,B2

operation A1()
    if (last!="B2")
        waitC(A1)
    last = "A1"
    print last
    signalC(A2)

operation B1()
    if (last!="A2")
        waitC(B1)
    last = "B1"
    print last
    signalC(B2)

operation A2()
    if (last!="A1")
        waitC(A2)
    last = "A2"
    print last
    signalC(B1)

operation B2()
    if (last!="B1")
        waitC(B2)
    last = "B2"
    print last
    signalC(A1)
```

Thread 1	Thread 2
funcoes.A1() funcoes.A2()	funcoes.B1() funcoes.B2()

- ☒ A1, A2, B1, B2
- ☐ A1, B1, A2, B2
- ☐ B1, B2, A1, A2
- ☐ N.D.A.



9. Uma seção crítica definida por "critical" em OpenMP tem maior relação com qual dos conceitos de programação concorrente relacionados abaixo? *

- ☐ Espera ocupada
- ☐ Semáforos
- ☒ Monitores
- ☐ Fork/Join

10. Um programador que está preocupado com as condições de corrida em seu código usa uma seção crítica para sincronizar o laço abaixo. Analise o código em indique a(s) afirmativa(s) correta(s). *

```
#pragma omp parallel for
for (i = 0; i < n; i++) {
    #pragma omp critical
    sum = sum + data[i];
}
```

- ☐ O programa ainda tem uma condição de corrida que retarda o programa.
- ☒ O programa não ganha velocidade porque o "pragma critical" permite que apenas uma thread execute o corpo do laço por vez.
- ☒ O programa poderia ter usado uma redução OpenMP (cláusula "reduction") para alcançar o mesmo resultado, porém de forma mais eficiente.
- ☐ Nenhuma das alternativas está correta.

Untitled Question

- ☒ Option 1

Este formulário foi criado em Universidade Federal de Sao Paulo.



Google Formulários



