Universidade de Brasília
Departamento de Ciência da Computação
Disciplina de Tópicos Avançados em Computadores - Programação Concorrente – Turma E
Prova 2 Período: 2014/1 Data: 25/06/2014
Aluno: Matrícula:

Questões

1 (1.5 pontos) - Explique o que são monitores, indicando como ocorre o controle de concorrência e a sincronização de processos em um monitor.

3 (2.0 pontos) - 3. Usando semáforos, complete o programa a seguir de maneira que o resultado impresso seja sempre "AAAAABBBBB" (isto é, cada processo só pode imprimir 'B' depois de todos os outros terem impresso 'A').

```
variáveis:
 . . .
 Processo (int i)  /* processo i = 0,1,2,3,4, ..., n; executa o seguinte código */ {
       ...
       imprime('A');
       ...
       imprime('B');
        . . .
 }
```

2 (2.0 pontos) - A seguir é apresentada uma solução para o problema do jantar dos filósofos.

```
semaphore garfo[5] = 1,1,1,1,1; /* cada semáforo é inicializado com 1*/

philosopher (int i)  /* filósofo i=0,1,2,3,4 executa o seguinte código */     {
   while(1) {
      think();                          \\pensa
      sem_wait(garfo[i]);          \\pega um garfo
      sem_wait(garfo[(i+1)%5]);    \\pega o outro garfo
      eat();                            \\come
      sem_post(garfo[(i+1)%5]);    \\ devolve um garfo
      sem_post(garfo[i]);          \\devolve o outro garfo
   }
}
```

Descreva como a solução apresentada pode levar a *deadlock*. Apresente ao menos uma alternativa para o problema de deadlock.

4 (2.0 pontos) - Considere a existência de dois monitores $m1$ e $m2$. Considere ainda que $m1$ e $m2$ possuem uma função/método chamado *metodo_m*1 e *metodo_m*2, respectivamente. Identifique um possível problema no algoritmo abaixo. Caso não exista nenhum problema, explique a sua resposta.

```
p_1() {
   m1.metodo_m1;
   m2.metodo_m2;
}

p_2() {
   m2.metodo_m2;
   m1.metodo_m1;
}
```

5 (2.5 pontos) - Implemente um solução para o problema de somar os elementos de uma matriz usando barreiras. Cada processo deve somar os elementos de uma linha (considere que serão criados um processo por linha da matriz e que seu identificador será recebido como parâmetro). Depois de cada processo computar os valores da sua linha correspondente, um processo qualquer deve totalizar a soma e imprimir o resultado na tela. Considere o seguinte protótipo para os processos.

```
somador(int id){
   ...
}
```