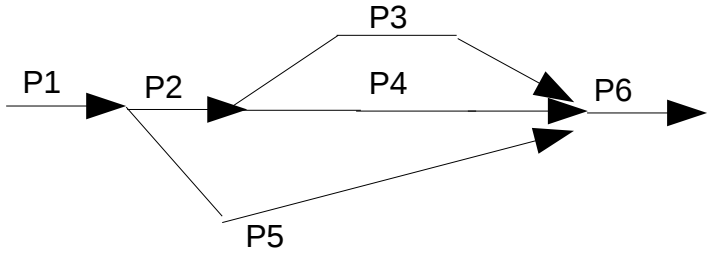
UNIVERSIDADE FEDERAL DE VIÇOSA
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
DEPARTAMENTO DE INFORMÁTICA
INF310 - Programação Concorrente e Distribuída
Segunda Prova – 29/10/2009 - Prof. Carlos Goulart

Nome:_____ Matr.: _____

1) a) Considere uma tarefa que é implementada por 6 processos, cuja ordem de execução e/ou concorrência estão mostradas na figura abaixo:



Esboce o código dos processos P1..P6, detalhando a definição das variáveis compartilhadas e os comandos de região para a sincronização baseada em RCCs. Considere que cada processo é um processo seqüencial e que executa apenas seu código apenas 1 vez.

b) Suponha que a tarefa fosse executada em 10 passos, ou seja, cada processo deve ser executado em um loop de 10 iterações, porém mantendo a precedência da figura acima: a segunda execução de P1 só deve ocorrer após terminar a primeira execução de P6, e assim, sucessivamente. Mostre que alterações seriam necessárias na slução do item “a” para atingir este objetivo.

---

2) a) Faça uma comparação das técnicas de sincronização de processos usando Semáforos e Regiões Críticas Condicionais (RCCs). Qual delas é a melhor? Quais as vantagens e desvantagens de cada uma?

b) Qual a vantagem principal da sincronização usando semáforos sobre a técnica de sincronização baseada em espera ocupada? Explique como esta vantagem é obtida.

c) Qual a diferença, no tocante ao acesso com exclusão mútua a uma variável compartilhada, quando se utiliza semáforos, RCC ou Monitores.

3) Considere o problema dos jantar dos selvagens descrito a seguir:

“Uma tribo de selvagens janta em conjunto, retirando pedaços de missionários assados de um grande pote que comporta até M pedaços. Quando um selvagem deseja comer ele se serve do pote, a menos que o pote esteja vazio. Se o pote estiver vazio, o selvagem acorda um cozinheiro e espera até que este tenha assado mais M pedaços de missionários e enchido o pote. Após encher o pote o coziheiro vai dormir”

a) Implemente uma solução baseada em semáforos para resolver este problema.

b) Analise sua solução e mostre que ela está correta.

---

4) Suponha que existam N passageiros e um carro da montanha russa. Os passageiros repetidamente esperar para andar no carro, que pode conter no máximo C passageiros, onde C < N. Porém, o carro só pode iniciar uma volta quando estiver cheio. Depois de terminar uma volta, cada passageiro sai do carro e passeia pelo parque de diversões antes de retornar à montanha-russa para uma outra volta. Por razões de segurança, o carro só dá V volta, quando pára para manutenção.

Suponha que o carro e os passageiros estão representados por processos, que devem cumprir os seguintes requisitos obrigatórios:

1. O carro sempre inicia uma volta com exatamente C passageiros;
2. Nenhum passageiro vai pular fora do carro enquanto o carro está dando uma volta;
3. Nenhum passageiro saltará para dentro do carro enquanto o carro está dando uma volta;
4. Nenhum passageiro vai permanecer no carro após o término de uma volta (tentar dar duas voltas pagando apenas 1 bilhete).

Os códigos dos processos Carro e Passageiros são esboçados a seguir:

```
Carro:
  c : integer init 0;
 loop
   …
   MR.espera_carro_encher();
   “dando volta”
   MR.espera_carro_esvaziar();
   c := c + 1
   exit when c = V
 endloop
```

```
Passageiro(i := 1 to N):

 loop
   …
   “passeia no parque”
   MR.entra_no_carro(i);
   “dando volta”
   MR.sai_do_carro(i);
   ...
 endloop
```

Implemente uma solução usando Monitor, detalhando:

a) as estruturas de dados para sincronizar os processos.

b) as operações usadas pelos processos Carro e Passageiros.