PUC-Rio – Sistemas de Computação I
Prova 1 – 2/5/2000

1. Considere a seguinte situação. Um pombo correio leva mensagens entre os *sites* A e B, mas só quando o número de mensagens acumuladas chega a 20. Inicialmente, o pombo fica em A, esperando que existam 20 mensagens para carregar, e dormindo enquanto não houver. Quando as mensagens chegam a 20, o pombo deve levar exatamente (nenhuma a mais nem a menos) 20 mensagens de A para B, e em seguida voltar para A. Caso existam outras 20 mensagens, ele parte imediatamente; caso contrário, ele dorme de novo até que existam as 20 mensagens. As mensagens são escritas em um *post-it* pelos usuários; cada usuário, quando tem uma mensagem pronta, cola sua mensagem à mochila do pombo. O vigésimo usuário deve acordar o pombo caso ele esteja dormindo. Cada usuário tem seu bloquinho inesgotável de *post-it* e continuamente prepara uma mensagem e a leva ao pombo.

   Usando semáforos, modele o processo *pombo* e o processo *usuário*, lembrando que existem muitos usuários e apenas um pombo. Identifique regiÕes críticas e não críticas na vida do usuário e do pombo.

2. (a) O que é um escalonador *preemptivo*?
   (b) Dê exemplo(s) de motivo(s) para ocorrência da preempção.
   (c) Dê exemplos de contextos em que a preempção é essencial.

3. Considere a seguinte modelagem, por monitor, para o problema do produtor-consumidor com n produtores e m consumidores.

```
monitor buffer {
  int itens=0; cond temItens, temEspacos;
  ...
  int pega() {
    while (1) {
      if (!itens) wait(temItens);
      pega item no buffer
      itens--;
      signal(temEspacos);
      return (item);
    }
  }
  void coloca() {
    while (1) {
      if (itens==MAX) wait(temEspacos);
      coloca item no buffer
      itens++;
      signal(temItens);
    }
  }
}
```

   Suponha que esse monitor funciona com a disciplina "sinaliza e continua" dada em sala (mas observe que ele admite filas separadas por variáveis de condição!). Explique por que essa solução não funciona corretamente.

4. Considere o problema dos leitores e escritores, onde existem diversos processos que eventualmente fazem acessos de leitura a uma base de dados e diversos processos que eventualmente fazem acessos de escrita à mesma base. Vários acessos de leitura podem ocorrer simultaneamente, mas um acesso de escrita não pode ocorrer simultaneamente com acessos de nenhum tipo. Considere o código abaixo para os processos de leitura e de escrita. Suponha que todos os semáforos são inicializados com valor 1.

```
leitor:                          escritor:
...                              ...
1  while(1) {                    1  while(1) {
2    P(R);                       2    ...produz
3    P(M);                       3    P(R);
4    rc++;                       4    P(W);
5    if (rc==1) P(W);            5    ESCREVE;
6    V(M);                       6    V(W);
7    V(R);                       7    V(R);
8    LE;                         8  }
9    P(M);
10   rc--;
11   if (rc==0) V(W);
12   V(M);
13   ... consome
```

   (a) Essa solução pode levar a *starvation* de escritores? (pode acontecer de um escritor nunca conseguir o acesso à base devido à seguida chegada de novos leitores?) Explique sua resposta, usando os números das linhas de código para se referir aos passos do programa.

   (b) Explique o papel do semáforo `M`. Dê um exemplo de problema que poderia ocorrer caso as operações sobre ele fossem retiradas.