



Exercícios sobre variáveis de condição

Exercício 01 – Notificação de eventos

Implemente um programa com múltiplas *threads* composto por threads apostadoras e uma *thread* auditora. Na inicialização, as *threads* apostadoras devem aguardar pela auditora informar o número da sorte (um inteiro entre 1 e 100, por exemplo). A partir da informação do número da sorte, cada thread apostadora fica em um laço de repetição repetindo os seguintes passos: soma um contador para o número de tentativas de descobrir o número da sorte, sorteia um número aleatório (chute) e verifica se acertou o número da sorte (chute = número da sorte). Caso a thread tenha acertado o número da sorte, ela notifica a *thread* auditora. Caso contrário repete o procedimento até que acerte o número.

A thread auditora fica bloqueada aguardando até que as threads apostadores desvendem o número secreto. Ao ser notificada por alguma thread que tenha acertado o número secreto. A thread auditora imprime na tela informações sobre a acertadora: ordem em que acertou o número (ex. 1ª, 2ª, etc..), qual foi a thread (informar o id) e quantas tentativas foram necessárias para acertar o valor.

Dica: você pode usar variáveis de condição para implementar o mecanismo de notificação da *thread* auditora.

Exercício 02 – Implementação de uma barreira

Você deve implementar uma estrutura de sincronização do tipo barreira usando variáveis de condição.

Para implementação da barreira, você deve implementar as seguintes estruturas e funções de manipulação da barreira:

- typedef: barreira (contém contador, mutex e variável de condição)
- inicializa_barreira(barreira b, int cont)
- espera(barreira b)

O tipo barreira contém os campos necessários para você implementar um mecanismo de espera do tipo barreira. O contador informa até quantas threads devem aguardar na barreira. A função `inicializa_barreira(barreira b)` garante que cada thread que execute esta operação fica bloqueada até que todas as threads necessárias para liberar a barreira tenham invocado esta função sobre a barreira b.