

Variáveis de condição usando PThreads

O objetivo deste Laboratório é introduzir o uso de variáveis de condição na biblioteca PThreads para implementar sincronização por condição. Para cada atividade, siga o roteiro proposto antes de passar para a próxima atividade.

Atividade 1

Objetivo: Exemplificar o uso de variáveis de condição provido pela biblioteca Pthreads.

Roteiro:

1. Abra o arquivo `hellobye.c` e compreenda como a aplicação funciona.
2. Execute o programa várias vezes. O log de execução impresso na tela foi sempre o esperado? A condição lógica da aplicação foi atendida em todas as execuções?
3. Abra o arquivo `byehello.c` e compreenda como a aplicação funciona.
4. Execute o programa várias vezes. O log de execução impresso na tela foi sempre o esperado? A condição lógica da aplicação foi atendida em todas as execuções?

Relatório da atividade: Em um arquivo `.txt` descreva as respostas das questões colocadas e inclua exemplos dos resultados obtidos nas execuções realizadas.

Atividade 2

Objetivo: Explorar características do funcionamento das operações sobre variáveis de condição em PThreads.

Roteiro:

1. Abra o arquivo `printX.c` e compreenda como a aplicação funciona.
2. Execute o programa várias vezes. O log de execução impresso na tela foi sempre correto? Atendeu à condição lógica da aplicação?
3. Podemos substituir a linha 54 pela linha 55? Justifique.
4. Comente a linha 54 e descomente a linha 55 e reexecute a aplicação várias vezes. O log de execução impresso na tela foi sempre correto? Atendeu à condição lógica da aplicação?

Relatório da atividade: Em um arquivo `.txt` descreva as respostas das questões colocadas e inclua exemplos dos resultados obtidos nas execuções realizadas.

Atividade 3

Objetivo: Projetar e implementar um programa *multithreading* em C onde a ordem de execução das *threads* é controlada no programa (usar *locks* e variáveis de condição).

Roteiro:

1. Implemente um programa com 4 *threads*:
 - A *thread* 1 imprime a frase “olá, tudo bem?”;
 - A *thread* 2 imprime a frase “hello!”;
 - A *thread* 3 imprime a frase “até mais tarde.”;
 - A *thread* 4 imprime a frase “tchau!”.
2. As *threads* 1 e 2 devem executar antes das *threads* 3 e 4 sempre (a ordem de execução entre as *threads* 1 e 2 não importa, assim como a ordem de execução entre as *threads* 3 e 4).

Relatório da atividade: Em um arquivo `.txt` descreva as respostas das questões colocadas e inclua exemplos dos resultados obtidos nas execuções realizadas.

Atividade 4

Objetivo: Mostrar uma implementação do problema dos produtores/consumidores usando *locks* e variáveis de condição.

Roteiro:

1. Abra o arquivo `pc.c` e compreenda como a aplicação funciona.
2. O programa está implementado corretamente?
3. Execute o programa várias vezes. O log de execução impresso na tela foi sempre o esperado? As condições lógicas da aplicação foram atendidas em todas as execuções?

Relatório da atividade: Em um arquivo `.txt` descreva as respostas das questões colocadas e inclua exemplos dos resultados obtidos nas execuções realizadas.

Atividade 5

Objetivo: Altere o programa `pc.c` acrescentando uma nova função para inserção de itens no *buffer* `InsererTudo`. Essa função deverá inserir N elementos de uma vez no *buffer* (precisa que o *buffer* esteja vazio). Acrescente ao programa uma *thread* produtora especial que chama a função `InsererTudo`. Teste seu programa.

Relatório da atividade: Em um arquivo `.txt` descreva as respostas das questões colocadas e inclua exemplos dos resultados obtidos nas execuções realizadas.