Aluno: Nicholas de Araujo Figueira DRE: 121088218

Atividade 1.

Abra o arquivo hellobye.c e identifique qual é o requisito lógico/condicional da aplicação (qual é a ordem de impressão requerida para as expressões HELLO e BYEBYE). Acompanhe a explanação da professora.

A condição lógica é que para imprimir a mensagem de BYEBYE, precisa-se que sejam impressas duas vezes a expressão HELLO. Isso se deve ao fato que após a imprimir HELLO, a variável estado é incrementada e, ao igualar a 2, passa-se a ser possível entrar dentro da condicional da função A. Isso libera permissão para depois acessar a condicional da função B e seguir ao longo da execução do código. Pode ser que a thread B também seja executada antes da thread A, porém, nesse caso, ela não será acessível enquanto a condição anterior não for cumprida.

Atividade 2.

Leia o programa e verifique se os requisitos foram atendidos.

Os requisitos foram cumpridos. Ambas as mensagens “Oi Maria!” e “Oi José!” são impressas antes de “Sente-se por favor.” e “Até mais José!” e “Até mais Maria!” vem depois dessa mensagem. Em ambos os casos, a ordem das mensagens em si é irrelevante desde que venham antes ou depois de “Sente-se por favor.” Isso se deve ao uso de variáveis (chegadas e sentados) para controlar condicionais de tal forma que as mensagens sejam impressas apenas no devido momento.

Atividade 3.

Execute o programa e verifique seus resultados. As threads estão executando de forma sincronizada, concluindo uma interação para então iniciar outra?

Sim. As threads só continuam sendo executadas após todos os passos serem impressos, independentemente da ordem destes.

Descomente a linha 44 (que chama a “barreira”). Execute novamente o programa e avalie os resultados.

Nesse caso, realiza-se o processo “oposto” ao anterior. Agora, com a adição da barreira, as threads apenas avançam para os passos seguintes após todas as threads terem feito o mesmo passo.