

Resultados do Laboratorio 4

Leonardo Santiago (120036072)

1 Análise

Podemos analisar a execução do código olhando para cada um dos valores.

Primeiramente notamos que se rodarmos as threads sequencialmente, T1 printará -1, T2 deixará $x = -1$ e T3 transforma $x = 0$ e não printa nada.

Sabemos então que pelo menos o valor -1 pode ser printado.

Notamos também que se a thread T3 for a primeira a executar, ela fará $x = 1$ e printará corretamente o valor 1 (assumindo que não aconteça nenhuma interrupção entre o if e o printf). Continuando, T2 continuará deixando x intacto, e T1 não printará nada.

Para que os outros valores sejam printados, precisamos que alguma das threads (T1 ou T3) sejam interrompidas entre o teste feito no if e execução da função print, para que o valor de x seja alterado nesse meio tempo.

Analizando primeiro quando T1 é interrompida nesse tempo, temos que $x = -1$, e portanto (como as Threads ainda precisam ser executadas sequencialmente dentro da própria thread), poderíamos ter que T2, incrementa x e depois T1 resume, printando 0; que T2 deixa o valor inalterado e printa -1; que T2 incrementa, T3 incrementa e depois printamos o valor $x = 1$; ou por fim que T2 incrementa, T3 incrementa, T2 decrementa e novamente temos o valor 0. Em todos os casos, o único valor novo que pode ser printado é o número 0.

Analizando agora quando T3 é interrompida, temos exatamente $x = 1$, e portanto se T2 incrementar, podemos printar o número 2. Se T2 incrementar e decrementar, voltamos para o valor 1, e qualquer ação feita por T1 volta para um dos valores que já poderia ser alcançado antes (2, 1 ou 0), já que ele primeiro deve decrementar (chegando em 1 ou 0), e depois incrementar novamente (chegando em 2 ou 1), e por fim decrementar de novo (1 ou 0). Se em qualquer um desses momentos o valor for printado, não teremos nenhum valor novo.

Portanto, analisando cada possível caso, vemos que os únicos valores que podem ser printados são -1, 0, 1 e 2, analisando somente caso todas as instruções rodem sem acontecer overwrites.

Podemos analisar então os outros valores assumindo que possíveis instruções sejam executadas ao mesmo tempo (e uma acabe sobreescrevendo a outra). Vemos que se a primeira linha de T1 rodar ao mesmo tempo que a primeira linha de T2, e a primeira linha de T1 sobreescrever $x = -1$, podemos rodar o resto de T1, passar da comparação, depois rodar a segunda linha de T2 e acabar printando o valor $x = -2$.

De maneira similar, podemos printar o valor 3 caso a primeiro T3 execute sem interrupções até a comparação, depois a segunda linha de T2 sobreescreva a primeira linha de T1 ($x = 2$), depois a segunda linha de T1 execute normalmente ($x = 3$) e depois escreva x na tela.

Para os outros valores, podemos fazer uma pequena análise para ver que não são possíveis. Se considerarmos que, independente do fluxo que ocorreu, chegamos na comparação de T1, e queremos que ela seja verdadeira ($x = -1$) para imprimir o valor na tela, podemos ver que temos no máximo, 2 incrementos e 1 decremento. Ainda que de alguma forma conseguíssemos combinar esses valores, podemos chegar a no máximo valores no intervalo $[-1 - 1, -1 + 2] = [-2, 1]$.

Olhando da mesma forma para a comparação em T3, temos que $x = 1$ e nas outras duas threads, temos 2 incrementos e 3 decrementos. Portanto, ainda que de alguma forma pudéssemos combinar eles da maneira que quiséssemos (o que não podemos exatamente), só poderíamos chegar em valores de $[1 - 3, 1 + 2] = [-2, 3]$. Portanto, vemos que nos dois casos, os valores -3 , 4 e -4 não serão alcançáveis, independentemente da ordem que sejam executadas, ainda que assumamos que qualquer ordem de execução ou de sobreescrita seja possível.

Logo, os valores possíveis serão $-2, -1, 0, 1, 2, 3$.