

Computação Concorrente, 2021.2

Exercício teórico sobre uso de memória compartilhada entre as threads

Aluno: Kevin Sena

DRE: 120085762

E-mail: kevinsa@dcc.ufrr.br

Exercício

Uma aplicação dispara três threads ($T1$, $T2$ e $T3$) para execução (códigos mostrados abaixo). Verifique se os valores 1, -1, 0, 2, -2, 3, -3, 4 e -4 podem ser impressos na saída padrão quando essa aplicação é executada. Em caso afirmativo, mostre uma sequência de execução das threads que gere o valor correspondente.

`int x = 0; // variável global`

	T1:	T2:	T3:
(0)			
(1)	<code>x = x - 1;</code>	<code>x = x + 1;</code>	<code>x = x + 1;</code>
(2)	<code>x = x + 1;</code>	<code>x = x - 1;</code>	<code>if(x == 1)</code>
(3)	<code>x = x - 1;</code>		<code>printf("%d",x);</code>
(4)	<code>if (x == -1)</code>		
(5)	<code>printf("%d",x);</code>		
(6)			

❖ Saída: 1

- A thread **T3** executa primeiro que as demais threads e termina sua tarefa de forma atômica, ou seja, finaliza tudo sem que haja interrupções de outras threads.
- A thread **T1** executa sua tarefa até a linha (4) e passa no if (ou seja, $x = -1$), porém, antes de imprimir na tela, a thread **T2** entra em execução e realiza a linha (1), agora $x = 0$. Antes que a thread **T2** realize a linha (2), a thread **T3** entra em execução e faz toda a sua tarefa, incrementando +1 ao x , logo $x = 1$, depois entra no if e imprime na tela o valor 1.

❖ Saída: -1

- A thread **T1** executa primeiro que as demais threads e termina sua tarefa de forma atômica.
- A thread **T2** executa completamente e depois a thread **T1**.

❖ Saída: 0

- A thread **T1** executa até a linha (4) e passa pelo if ($x = -1$), mas antes de imprimir o valor na tela, a thread **T2** entra em execução e realiza a linha (1), ou seja $x = 0$, em seguida o fluxo volta para **T1** e imprime o 0 na tela.
- A thread **T3** executa até a linha (2) e passa pelo if ($x = 1$), mas antes de imprimir o valor na tela, a thread **T1** entra em execução e realiza a

linha (1), ou seja $x = 0$, em seguida o fluxo volta para **T3** e imprime o 0 na tela.

❖ **Saída: 2**

- A thread **T3** executa até a linha (2) e passa pelo if ($x = 1$), mas antes de imprimir o valor na tela, a thread **T2** entra em execução e realiza a linha (1), ou seja $x = 2$, em seguida o fluxo volta para **T3** e imprime o 2 na tela.

❖ **Saída: -2**

- A thread **T1** executa até a linha (1), logo $x = -1$, em seguida a thread **T2** entra em execução e executa a linha (1) ao mesmo tempo em que a **T1** realiza a linha (2) e, dessa forma, apenas 1 incremento é computado, logo $x = 0$. Após isso, a thread **T2** perde a CPU e a **T1** executa a linha (3), computando $x = -1$, entra e passa pelo if e, antes de imprimir na tela, a **T2** volta e termina sua execução decrementando o x novamente com a linha (2), logo $x = -2$. Por fim, o fluxo volta para **T1** e imprime o valor de -2 na tela.

❖ **Saída: 3**

- A thread **T3** executa até a linha (2) e passa pelo if ($x = 1$), mas antes de imprimir na tela, as threads **T1** e **T2** entram em execução e realizam a linha (1) ao mesmo tempo, mas a linha (1) de **T2** prevalece sobre **T1** e incrementa o valor de x , logo $x = 2$. Em seguida, a thread **T1** realiza a linha (2) e incrementa novamente o valor de x , logo $x = 3$. Por fim, o fluxo volta para **T3** e imprime o valor de 3 na tela.

❖ **Saída: -3**

- A thread **T1** inicia sua tarefa, lê $x = 0$ e guarda essa informação em um registrador, salvando o valor de x no contexto da thread **T1** e, em seguida, ocorre a interrupção por tempo. Então, a **T3** entra em execução, incrementa 1 ao x , logo $x = 1$ e entra no if, mas antes de imprimir na tela, o fluxo volta para a thread **T1**, ela restaura todo o seu contexto e recupera de seu registrador o valor de $x = 0$. Depois disso, a thread **T2** é disparada e então as linhas (1) e (2) são executadas ao mesmo tempo por **T1** e **T2**. Na linha (1) **T1** se sobrepõe ($x = -1$) e na linha (2) **T2** se sobrepõe ($x = -2$). Em seguida **T1** realiza a linha (3), deixando $x = -3$ e o fluxo volta para a thread **T3**, que imprime o valor -3 na tela.

❖ **Saída: 4**

- Não é possível que 4 seja impresso na tela, pois só há 3 instruções de incremento entre as threads.

❖ **Saída: -4**

- Da mesma forma que o 4, o -4 também não pode ser impresso na tela, pois só há 3 instruções de decremento entre as threads.