

Laboratório 04 | Relatório

Computação Concorrente 2021.2

Carolina Naccarato | 117220395

O objetivo deste laboratório é praticar os conceitos estudados sobre o uso de memória compartilhada entre as threads e o problema de condição de corrida.

Dada uma aplicação que dispara três threads (T1, T2 e T3), temos que verificar as condições e os possíveis fluxos em que é possível que os valores 1, -1, 0, 2, -2, 3, -3, 4 e -4 sejam impressos na saída padrão quando a aplicação é executada. O código das handlers das threads se encontra abaixo.

```
int x=0; //variavel global
(0) T1:                T2:                T3:
(1) x = x-1;           x = x+1;           x = x+1;
(2) x = x+1;           x = x-1;           if (x == 1)
(3) x = x-1;                                   printf("%d", x);
(4) if (x == -1)
(5) printf("%d", x);
```

- 1

O valor 1 impresso na saída padrão é algo esperado, ou seja, é o caso em que a execução do nosso código foi bem sucedida. Para que esse valor seja impresso, basta que a função T3 seja executada, sem interrupções.

```
(0) T3:
(1) x = x+1;
(2) if (x == 1)
(3) printf("%d", x);
```

- -1

O valor -1 também é algo esperado pelo nosso programa. Para que esse valor seja impresso na saída padrão, basta que a função T1 seja executada, sem interrupções.

```
(0) T1:
(1) x = x-1;
(2) x = x+1;
(3) x = x-1;
(4) if (x == -1)
(5) printf("%d", x);
```

- 0

Se o valor 0 for impresso, assim como os demais abaixo, significa que mais de um fluxo foi executado de forma concorrente e observamos o problema de condição de corrida. Essa situação pode ocorrer caso a função T1 execute as suas instruções sem interrupções até a linha (4), de forma que a variável x estivesse com o valor igual a -1 e o fluxo de execução caia no if. Neste momento, a função T3 começa a sua execução até a linha (2),

fazendo com que a variável `x` passe a assumir o valor 0, que, por sua vez, é impresso na saída padrão.

```
(0) T1:
(1) x = x-1;
(2) x = x+1;
(3) x = x-1;
(4) if (x == -1)

(0) T3:
(1) x = x+1;

(0) T1:
(5) printf("%d", x);
```

- 2

Para que o valor 2 seja impresso na saída padrão, temos que a função `T3` começa a sua execução sem interrupções até a linha (2), de forma que `x = 1` e, assim, o fluxo de execução entra na condição do `if`. Neste momento, a primeira linha da função `T2` é executada, alterando o valor de `x` para 2, que, por sua vez, é impresso na saída padrão.

```
(0) T3:
(1) x = x+1;
(2) if(x == 1)

(0) T2:
(1) x = x+1;

(0) T3:
(3) printf("%d", x);
```

- -2

Para que a variável `x` assuma o valor de -2, teríamos que considerar situações em que os três fluxos se entrelaçam. A função `T1` começa a sua execução até a linha (1), fazendo `x = -1`. Neste momento, as funções `T2` e `T3` começam a sua execução, executando as linhas (1) respectivas a cada função, ao mesmo tempo que a função `T1` executa a linha (2). Como todas as funções leram o valor de `x` como -1, ao final disso a variável `x` termina sendo igual a 0. Assim, voltamos novamente para o fluxo de execução da função `T1`, de forma que as linhas (3) e (4) são executadas, caindo no `if` e avaliando como verdade pois `x = -1`. Nesse instante, a função `T2` finaliza a sua execução com a linha (2), decrementando a variável `x` de 1, fazendo com que seja impresso `x = -2`.

```
(0) T1:
(1) x = x-1;
```

--Executando ao mesmo tempo

```
(0) T1:
(2) x = x+1;
```

```

(0) T2:
(1) x = x+1;

(0) T3:
(1) x = x+1;

--x = 0

(0) T1:
(3) x = x-1;
(4) if (x == -1)

(0) T2:
(2) x = x-1;

(0) T1:
(5) printf("%d", x);

```

- 3

Para que o valor 3 seja impresso na saída padrão, novamente temos que analisar situações em que as três funções se entrelaçam. Começamos o nosso fluxo na função `T3`, executando até a linha (2). Após isso, $x = 1$ e, portanto, o controle avalia o `if` como verdade. Mas, antes de imprimir o valor de x , as funções `T1` e `T2` executam ao mesmo tempo as suas respectivas instruções na linha (1). Por conta dessa concorrência, perdemos o decremento da linha (1) de `T1`, terminando com $x = 2$. Após isso, a função `T1` executa a linha (2), incrementando o valor de x , finalizando com $x = 3$, que é impresso na saída padrão.

```

(0) T3:
(1) x = x+1;
(2) if(x == 1)

--Executando ao mesmo tempo

(0) T2:
(1) x = x+1;

(0) T1:
(1) x = x-1;

--x = 2

(0) T1:
(2) x = x+1;

(0) T3:
(3) printf("%d", x);

```

- -3

É possível que a variável x assumo o valor -3 se entrelaçarmos o fluxo das funções $T1$ e $T2$. Porém, o `if` da função $T1$ nunca seria avaliado como verdade, pois ele chegaria nele com $x = -3$. Assim, não é possível que o valor -3 seja impresso na saída padrão.

- 4 e -4

Não existe nenhum fluxo que permita que x assumo tais valores pois x começa sendo igual a 0.