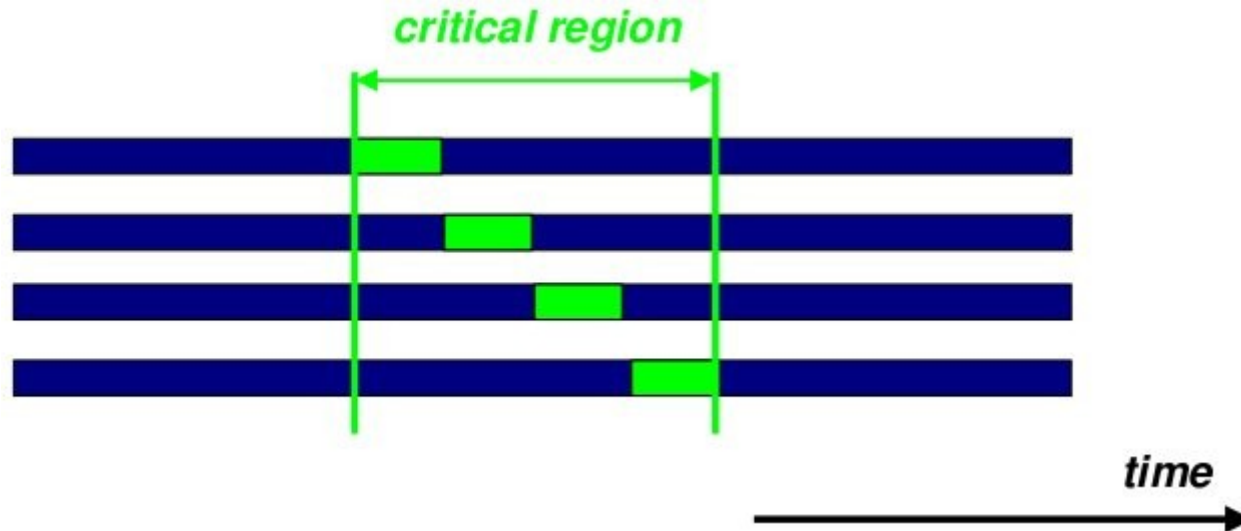


Seções Críticas em OpenMP

- Bloco de código que só pode ser executado por uma *thread* por vez
 - Pode ser utilizado para proteger a atualização de variáveis compartilhadas e evitar a condição de corrida
 - A diretiva *CRITICAL* permite que as seções críticas recebam nomes
 - Se uma *thread* está em uma seção crítica com um dado nome, nenhuma outra *thread* pode estar em uma seção crítica com o mesmo nome (embora elas possam estar em seções críticas com outros nomes)
 - Se o nome é omitido, um nome nulo é assumido



Exemplo básico para seção crítica em OpenMP

```
#include <omp.h>

main()
{
    int x;
    x = 0;

    #pragma omp parallel shared(x)
    {
        #pragma omp critical
        x = x + 1;
    } /* end of parallel section */
}
```

Seção Crítica – Exemplo 2

- Colocando e retirando dado de uma pilha

```
#pragma omp parallel shared(stack), private(inext, inew)
...
#pragma omp critical (stackprot)
{
    inext = getnext(stack);
}
work(inext, inew);
#pragma omp critical (stackprot)
if (inew > 0) putnew(inew, stack);
}
...
```

Diretiva *ATOMIC*

- Usada para proteger uma atualização única para uma variável compartilhada
- Aplica-se apenas a uma única sentença (forma "light" de se definir uma seção crítica)
- Sintaxe em C/C++:
#pragma omp atomic
statement
 - Onde *statement* deve ter uma das seguintes formas:
 - $x \text{ binop } = \text{expr}$, $x++$, $++x$, $x--$, or $--x$
 - and binop is one of $+$, $*$, $-$, $/$, $\&$, $^$, $<<$, or $>>$

Diretiva ATOMIC - Exemplo

- Computar o grau de cada vértice em um grafo:

```
#pragma omp parallel for
for (j=0; j<nedges; j++) {
    #pragma omp atomic
        degree[edge[j].vertex1]++;
    #pragma omp atomic
        degree[edge[j].vertex2]++; }
```