

## prog\_paralel\_for.c

```
1  #define _POSIX_C_SOURCE 200809L
2
3  #include <stdio.h>
4  #include <stdlib.h>
5  #include <omp.h>
6
7  // Definição global do número de passos para consistência
8  const long NUM_PASSOS = 100000000;
9  long pontos_no_circulo = 0; //variavel compartilhada pelas thread
10 unsigned int seed = 12345;
11
12 //versão Paralela
13 void pi_paralel_for() {
14
15     #pragma omp parallel for
16
17     for (long i = 0; i < NUM_PASSOS; i++){
18         unsigned int seed_T = seed ^ omp_get_thread_num(); //semente unica
19         por thread
20
21         double x = (double)rand_r(&seed_T) / RAND_MAX * 2.0 - 1.0;
22         double y = (double)rand_r(&seed_T) / RAND_MAX * 2.0 - 1.0;
23         if (x * x + y * y < 1.0) {
24             pontos_no_circulo++; //aqui esta a condição de corrida
25         }
26     }
27
28 int main() {
29     double start_time, end_time;
30
31     printf("Iniciando analise de desempenho para %ld passos.\n", NUM_PASSOS);
32     start_time = omp_get_wtime();
33     pi_paralel_for();
34     end_time = omp_get_wtime();
35     double tempo_paralelo = end_time - start_time;
36     double pi_estimado = 4.0 * pontos_no_circulo / NUM_PASSOS;
37
38     printf("Estimativa paralela de pi = %f\n", pi_estimado);
39     printf("Tempo Paralelo: %f segundos\n", tempo_paralelo);
40
41     return 0;
42 }
```