Universidade Federal do Rio Grande do Norte
Departamento de Engenharia da Computação e Automação
DCA3703 - Programação Paralela
Tarefa 7 - Utilização de tasks
Aluno: Daniel Bruno Trindade da Silva

1 Introdução

O presente relatório tem como objetivo entender o funcionamento do paralelismo com múltiplas threads e tasks, para isso desenvolvemos um programa em linguagem C utilizando a biblioteca `OpenMP` para implementar uma estrutura de lista encadeada cujos nós contêm nomes de arquivos fictícios e, dentro de uma região paralela, percorrer essa lista criando uma task individual para o processamento de cada nó. Cada task é responsável por imprimir o nome do arquivo e o identificador da thread que a executou.

O experimento também buscou responder perguntas-chave: todas as tarefas foram executadas? Houve repetição ou omissão na execução dos nós? O comportamento do programa muda a cada execução? Além disso, foram discutidas estratégias para garantir que cada tarefa seja executada uma única vez e por apenas uma thread.

2 Metodologia

Nosso código foi estruturado para atender os requisitos da tarefa, assim criamos um `struct` para os nós da lista encadeada, cada nó recebeu dois atributos, `nome_arquivo` para dar o nome fictício ao arquivo e `prox` que é um ponteiro para o próximo nó.

```
typedef struct Node {
    char nome_arquivo[100];
    struct Node* prox;
} Node;
```

Também foram criadas funções como `criar_no()` que cria um novo nó, `adicionar_no` que adiciona um nó criado à lista encadeada

diretiva `#pragma omp single` para garantir que apenas uma thread fosse responsável pela criação das tarefas, enquanto todas as threads disponíveis na região paralela ficaram encarregadas de executá-las. Essa abordagem permitiu observar o comportamento dinâmico da distribuição das tarefas entre as threads, bem como analisar questões tais como concorrência, balanceamento de carga e ordem de execução.