

Projeto “Multicat integer sort”

Prof. André Leon S. Gradvohl, Dr.

gradvohl@ft.unicamp.br

Objetivo

Este projeto visa a criação de um programa que utilize múltiplas *threads* para ler números inteiros de vários ($n \geq 1$) arquivos com valores inteiros em diferentes quantidades, ordenar todos esses dados e concatená-los em um único arquivo resultante.

Toda essa operação – leitura dos arquivos, ordenação e concatenação dos dados – deve ocorrer utilizando 2, 4, 8 ou 16 *threads*, a critério do usuário.

O programa deverá ser escrito para o sistema operacional Linux e obrigatoriamente utilizar a biblioteca POSIX Threads.

Descrição do problema a ser resolvido

Considere que há vários ($n \geq 1$) arquivos, cada um deles com uma certa quantidade de números inteiros sem ordem estabelecida. O programa deverá ler os dados de todos esses n arquivos, organizá-los em ordem crescente e armazenar em um único arquivo resultante.

A ordenação pode ser feita com qualquer algoritmo estabelecido pela equipe. Entretanto, recomenda-se a utilização de algoritmos rápidos de ordenação como, por exemplo, o Quicksort ou o Heapsort.

A quantidade de arquivos é variável, podendo ser um único arquivo ou vários, a critério do usuário. Os arquivos originais devem ser preservados.

Entradas e saídas de valores para o programa

Entradas: o número de threads, os arquivos com os dados de entrada e o arquivo de saída que conterá os resultados. Recomenda-se que todas essas informações de entrada sejam fornecidas na própria linha de comando, conforme o exemplo a seguir.

Saídas: um arquivo com os valores inteiros ordenados.

Por exemplo, suponha que o programa multicat deva ser executado com 16 threads lendo os arquivos de números inteiros `arq1.in`, `arq2.in`, `arq3.in`, e produzindo o arquivo `arqSaida.out` com os dados ordenados. Assim, o comando para a execução do programa ficaria assim:

```
./multicat 16 arq1.in arq2.in arq3.in arqSaida.out
```

Detalhamento dos produtos do projeto

Os produtos que devem ser entregues como resultado do projeto são os seguintes:

- i. O código fonte do programa completo, documentado e pronto para ser compilado em sistemas Linux.
- ii. Um vídeo mostrando o código fonte do programa, a compilação do programa, trechos dos arquivos de entrada, a execução do programa para 2, 4, 8 e 16 threads e trechos dos arquivos de saída.
- iii. Um relatório contendo a descrição da solução do problema (algoritmo em alto nível); as instruções para compilá-lo; gráficos com os tempos de execução do programa para 2, 4, 8 e 16 threads; e as conclusões a respeito dos resultados obtidos. Meça o tempo de execução do programa considerando apenas os trechos com processamento, sem o tempo gasto para ler os arquivos.

Todo o código fonte documentado, o relatório e a documentação do projeto devem estar disponíveis no repositório Git. No Moodle da disciplina, deve ser publicado apenas o relatório em PDF e neste relatório deve constar o endereço do repositório Git e, se necessário, o endereço do vídeo na internet.

Para o item (i), a sugestão é a utilização de um `makefile` que facilite a compilação em sistemas Linux. Certifiquem-se de que não há erros na compilação do programa. **Qualquer erro na compilação significará que o programa não está correto e isso acarretará em nota zero no projeto.**

Para o item (ii), sugere-se o uso dos programas para capturar a tela do computador e criar vídeos, como o SimpleScreenRecorder (<http://www.maartenbaert.be/simplescreenrecorder>). O vídeo poderá estar armazenado no próprio repositório Git ou em outro local na internet, desde que possa ser acessado pelo professor.

Para o item (iii), o relatório deve estar no formato PDF ou disponível no próprio servidor Git. **Outros formatos não serão aceitos e acarretarão em nota ZERO, sem a necessidade de visualização do conteúdo.**

A falta de qualquer um dos itens acarretará em nota zero nesse componente de avaliação.