

## Relatório para o projeto 2- “Integer Multifind”

**Grupo:** Mochileiros das Galáxias

**Integrantes:**

Andrey Toshiro Okamura - 213119

Gustavo Adrien Polli -

Mateus Pim Santos -

**TADS** - Sistemas operacionais

**Repositório Github:** <https://github.com/mateuspim/IntegerMultifind>

**Playlist com vídeos:**

<https://www.youtube.com/playlist?list=PLGJJ1MuuVngsFRPf6fbDzOTE3W5hx6dYK>

### Problema:

Há vários ( $n \geq 1$ ) arquivos que contém números inteiros aleatórios e não ordenados. O programa deverá utilizar múltiplos threads para, em todos os  $n$  arquivos, encontrar um determinado valor, e informar em que linha está este número.

### Algoritmo em alto nível para solução do problema:

Aqui será apresentado o algoritmo utilizado em “main.c”.

1. Etapa de leitura de arquivos:
  - a. Separação e interpretação do array de entrada;
  - b. Recebe os inteiros dos arquivos e aloca em um vetor;
  - c. Fornece os valores início e fim de busca para as structs;
2. Etapa de construção/execução de threads:
  - a. Inicialização das threads;
  - b. Criação de threads de acordo com o número fornecido na entrada, para executar a função “IntegerMultifind”, que verifica a existência do valor procurado.
3. Etapa de exibição dos resultados; nela é chamado a função “printNumArray”, que:
  - a. Caso o valor verificador for -1, foi passado para outro arquivo;
  - b. Caso o valor seja -2, dado pelo “IntegerMultifind”, o número procurado foi encontrado;
  - c. Caso o próximo valor do array for -1 e o número não foi encontrado, informa a inexistência do número.

## Instruções de Compilação:

Está sendo disponibilizado duas formas de compilação, automático e manual.  
Automático:

No caso automático, o comando `make` permite a criação de um dado número de arquivos e a execução do programa, além da compilação.

Em um terminal ou interpretador de comando, digite:  
“sh automate.sh”

Manual:

O caso manual pode ser usado se desejar realizar todo o procedimento passo a passo, ou se não utilizou o *automate.sh*. Comando:

“make”

Caso queira limpar a saída, é possível utilizar o comando:  
“make clean”

Observações:

O sub-módulo *random* é um programa auxiliar para o projeto, e pode ser encontrado na pasta *random* no repositório Github, incluindo um guia de uso. Se *está usando o automate.sh*, não é necessário acessar os módulos manualmente.

## Instruções de Uso:

Em um terminal ou interpretador de comando, digite:  
“./multifind 16 123 arq1.in arq2.in arq3.in”

Onde:

16 é o número de threads a serem utilizados;

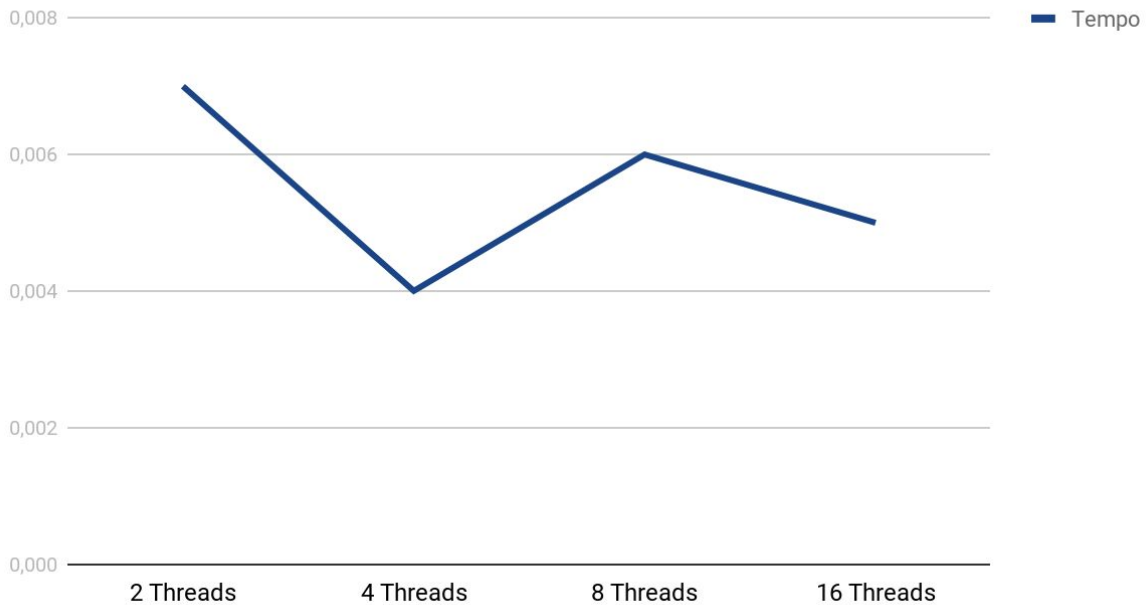
123 é o inteiro a ser procurado;

arq1.in, arq2.in, arq3.in são os arquivos a serem buscados.

### Gráfico com tempo de execução:

Para este projeto foi utilizado o processador Intel Core i5-3570 @ 3.40GHz. Os dados foram obtidos através do retorno fornecido pelo próprio programa, desconsiderando o tempo de leitura e o tempo de exibição dos resultados.

#### Tempo de execução



Em 2 Threads, o tempo de execução foi:

Em 4 Threads, o tempo de execução foi:

Em 8 Threads, o tempo de execução foi:

Em 16 Threads, o tempo de execução foi:

### Conclusões: