Tiago Ribeiro de Assunção - 13/0051187
Wilton Rodrigues - 13/0049212

Qual sistema operacional foi usado na construção do sistema;
* GNU/Linux - Debian 8.

Qual ambiente de desenvolvimento foi usado;
* Compilador GCC 4.9.2-10 com flag para std=c11 e VIM como editor de textos.

Quais são as telas (instruções de uso);
* Versão com threads
Para executar o programa, é necessário rodar o comando “make” e logo após executar o arquivo run com o comando “./run n x[0] x[1] x[2] ... x[n-1]”. Onde o n é a quantidade de inteiros que serão passados e em seguida cada um dos mesmos.

* Versão sem threads
Para executar o programa, é necessário rodar o comando “make” e logo após executar o arquivo run com o comando “./run_main-nothread n x[0] x[1] x[2] ... x[n-1]”. Onde o n é a quantidade de inteiros que serão passados e em seguida cada um dos mesmos.

Quais são as limitações conhecidas;
* As limitações se aplicam a ambas as soluções.
Caso o n seja menor ou igual a zero e maior que 100 a execução é finalizada para pela assertiva de programação defensiva.
Caso seja informado um valor de n e quantidade de parâmetros não corresponda ao valor de n, também há uma assertiva para tratar esse caso.
O programa não faz tratamento de string, pois a função atoi atribui 0 quando uma string é passada.
O programa não executa se não foi passada nenhum parâmetro pela linha de comando.
Caso sejam passados mais parâmetros do que o informado pelo n o programa irá desconsiderá-los, levando em conta apenas os valore do intervalo informado.

Casos de testes que demonstram que a aplicação contempla o comando do Trabalho.
* Entrada: ./run 3 1 2 3
* Ao executar, qualquer um dos programas deve-se visualizar a saída:
Number of input values = 3
Input values x = 1 2 3
After initialization w = 1 1 1
Thread T(0,1) compares x[0] = 1 and x[1] = 2, and writes 0 into w[0]
Thread T(0,2) compares x[0] = 1 and x[2] = 3, and writes 0 into w[0]
Thread T(1,2) compares x[1] = 2 and x[2] = 3, and writes 0 into w[1]
After Step 2
w = 0 0 1
Maximum = 3

Location = 2

* Entrada: ./run 3 1
* Ao executar, qualquer um dos programas deve-se visualizar a saída:
run: src/main.c:20: main: Assertion `argv[i+2] != NULL' failed.
Abortado (imagem do núcleo gravada)

* Entrada: ./run 2 1 2 3
* Ao executar, qualquer um dos programas deve-se visualizar a saída:
Number of input values = 2
Input values x = 1 2
After initialization w = 1 1
Thread T(0,1) compares x[0] = 1 and x[1] = 2, and writes 0 into w[0]
After Step 2
w = 0 1
Maximum = 2
Location = 1

QUESTÃO: Por que precisamos n(n-1)/2 em vez de n^2 threads?
* A partir da comparação n(n-1)/2 já é possível contemplar todos os casos de testes relevantes, porque qualquer outro caso de teste além destes já vai ter sido comparado implicando na repetição da comparação.

QUESTÃO: Verifique se há diferenças de desempenho observadas na busca pelo máximo ao se comparar o comportamento da implementação com threads com a implementação sequencial. Caso haja diferenças, explique-as.
* A solução com threads foi rodada com o seguinte comando: time ./run 50 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50

* Já a solução sem threads foi rodada com o comando: time ./run_main-nothread 50 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50

Após executar o comando 5 vezes para cada uma das soluções obteve-se a seguinte média de resultado:

COM THREAD	SEM THREAD
0,136	0,126
0,134	0,125
0,135	0,126
0,135	0,123
0,139	0,124
0,136	0,125

É possível observar que o programa com threads não obteve um desempenho melhor que o sem threads, como era esperado. Isso pode ser explicado pelo fato de que o programa solicitado é composto necessariamente de memory-bound e não de cpu-bound. Ou seja, a utilização de CPU nesse programa não é intensa. Por isso a queda de desempenho se dá pelo acesso à memória principal, pois as threads estão apenas se revezando para escrever na memória e não fazendo uso intensivo da CPU para processamento.

QUESTÃO: O que você sugeriria para otimizar essa implementação?
* Ao invés de utilizar uma implementação baseada no processamento por processadores, que são limitados, nós propomos uma implentação baseada na utilização de uma GPU para fazer o processamento em paralelo.