



Trabalho 2 Fundamentos de Sistemas Operacionais - Questão 3

Alunos:

Eduardo Q. Gomes 14/0137068

Miguel Pimentel 14/0156143

Material Usado

- Sistema Operacional Linux, distribuição Ubuntu 16.04 com a interface Gnome;
- Compilador gcc (5.4.0);
- Editor de texto Sublime.
- Valgrind versão 3.11.0

Conteúdo Utilizado

- Conteúdo do Moodle disponível em <http://aprender.ead.unb.br/course/view.php?id=3806>
- Livro Autor: Autor: SILBERSCHATZ, A.; GAGNE, G.; GALVIN, P.B. Obra: Operating System Concepts

Questão 3

Escreva programas em C que implementem a multiplicação de matrizes de coeficientes inteiros de acordo com os requisitos abaixo:

Implemente o programa q03a que realize a multiplicação de matrizes de forma sequencial, ou seja, execute todas as operações em uma única thread (a thread principal do processo).

O programa contém uma estrutura que pode ser compilada utilizando um Makefile, além disso conta com um README.md para o usuário poder rodar o software. O programa trabalha com um conjunto de arquivos, são eles:

- main.c - que contém a função principal para a execução do programa

Para utilizar o software, o usuário deverá digitar no terminal

- make
- make run

Dependências: Depende das bibliotecas: <stdio.h> <pthread.h> <time.h> <stdlib.h>

Tela da execução do software:

```
eduardo@Nautilus2:~/Documentos/GitE/SO_Trabalhos/T2/q03a$ make
gcc main.c -o prog -W -Wall -pedantic -ansi -lm
eduardo@Nautilus2:~/Documentos/GitE/SO_Trabalhos/T2/q03a$ make run
./prog
É possível multiplicar essa matriz
Matriz do Resultado:

[0][0]: -2 [0][1]: -17 [0][2]: 15
[1][0]: -20 [1][1]: 6 [1][2]: -26
[2][0]: 22 [2][1]: 0 [2][2]: 22

Tempo de Execucao: 0.000140
```

Implemente o programa q03b que realize a multiplicação de matrizes de forma concorrente, lançando uma thread para o cálculo simultâneo de cada célula da matriz de saída.

- Considere o produto de duas matrizes inteiras do seguinte suporte: 3x2 e 2x3. A matriz de saída deverá ter dimensões 3x3; logo, seu programa deverá calcular as 9 células da matriz de saída disparando 9 threads simultaneamente, cada uma responsável pelo cálculo da sua respectiva célula de saída.

O programa contém uma estrutura que pode ser compilada utilizando um Makefile, além disso conta com um README.md para o usuário poder rodar o software. O programa trabalha com um conjunto de arquivos, são eles:

- main.c - que contém a função principal para a execução do programa

Para utilizar o software, o usuário deverá digitar no terminal

- make
- make run

Tela da execução do software:

```
eduardo@Nautilus2:~/Documentos/GitE/SO_Trabalhos/T2/q03b$ make
gcc main.c -o prog -W -Wall -pedantic -ansi -lm -lpthread
eduardo@Nautilus2:~/Documentos/GitE/SO_Trabalhos/T2/q03b$ make run
./prog
Calculus done on Thread!
Calculus done on Thread!
Calculus done on Thread!
Calculus done on Thread!
Calculus done on Thread!
Calculus done on Thread!
Calculus done on Thread!
Calculus done on Thread!
Calculus done on Thread!

[-2] [-17] [15]
[-20] [6] [-26]
[22] [0] [22]

Tempo de Execucao: 0.000497
```

Implemente o programa q03c que realize a multiplicação de matrizes de forma concorrente, porém, lançando uma thread para o cálculo de cada célula da matriz de saída.

- O número máximo de threads disparadas simultaneamente deverá ser igual ao inteiro que resulta do seguinte comando: `cat /proc/cpuinfo | grep c processor`

O retorno dessa função foi 2.

- Se o número retornado pelo comando acima for igual a 4, por exemplo, e a matriz de saída possuir 9 células, será necessário o disparo de 3 lotes de threads: o primeiro e o segundo lotes com 4 threads e, no lote final, sobrá uma thread para ser lançada e calcular o valor do produto da célula remanescente.

Pelo fato de ser 2 a resposta acima, serão usados 5 pacotes para executar as 9 células da matriz 3x3 usada.

O programa contém uma estrutura que pode ser compilada utilizando um Makefile, além disso conta com um README.md para o usuário poder rodar o software. O programa trabalha com um conjunto de arquivos, são eles:

- main.c - que contém a função principal para a execução do programa

Para utilizar o software, o usuário deverá digitar no terminal

- make
- make run

Tela da execução do software:

```
eduardo@Nautilus2:~/Documentos/GitE/SO_Trabalhos/T2/q03c$ make
gcc main.c -o prog -W -Wall -pedantic -ansi -lm -lpthread
eduardo@Nautilus2:~/Documentos/GitE/SO_Trabalhos/T2/q03c$ make run
./prog
Calculus done on Thread!
Calculus done on Thread!
Calculus done on Thread!
Calculus done on Thread!
Calculus done on Thread!
Calculus done on Thread!
Calculus done on Thread!
Calculus done on Thread!
Calculus done on Thread!
Calculus done on Thread!

[-2] [-17] [15]
[-20] [6] [-26]
[22] [0] [22]

Tempo de Execucao: 0.000396
```

Questão de análise:

- Compare o desempenho em termos de tempo de execução das três aplicações e justifique eventuais diferenças de desempenho obtidas. Em sua resposta, indique qual foi a implementação mais rápida em termos de tempo de execução.
- Para essa solução, será necessário usar esquemas de bechmarking para comparar os tempos de execução das aplicações.

Para a análise dos tempos de execução serão utilizados:

- Biblioteca <time.h> que será implementada no código
- time ./prog
- times ./prog da biblioteca <sys/times.h>

Os resultados obtidos foram:

- Resultados com <time.h> no código

q03a

Tempo de Execucao: 0.000118

Tempo de Execucao: 0.000138

Tempo de Execucao: 0.000145



q03b

Tempo de Execucao: 0.000811

Tempo de Execucao: 0.000514

Tempo de Execucao: 0.001431

q03c

Tempo de Execucao: 0.000474

Tempo de Execucao: 0.000380

Tempo de Execucao: 0.000523

- Resultados com time ./prog

q03a

real 0m0.002s

user 0m0.000s

sys 0m0.000s

real 0m0.002s

user 0m0.000s

sys 0m0.000s

real 0m0.001s

user 0m0.000s

sys 0m0.000s

q03b

real 0m0.003s

user 0m0.004s

sys 0m0.000s

real 0m0.002s

user 0m0.000s

sys 0m0.000s

real 0m0.003s

user 0m0.000s

sys 0m0.000s

q03c

real 0m0.004s

user 0m0.004s

```
sys 0m0.000s
```

```
real 0m0.002s  
user 0m0.000s  
sys 0m0.000s
```

```
real 0m0.002s  
user 0m0.000s  
sys 0m0.000s
```

- Resultados com times ./prog

q03a

```
0m0.840s 0m0.240s  
0m40.792s 0m2.308s
```

q03b

```
0m0.144s 0m0.024s  
0m0.000s 0m0.004s
```

q03c

```
0m0.100s 0m0.012s  
0m2.620s 0m0.092s
```

Resultados:

Com os resultados obtidos pode-se observar algumas análises pertinentes a cada caso.

O tempo de execução da q03a foi mais rápido que o tempo de execução das outras questões, a partir dessa afirmação, pode se afirmar a partir dos dados que na verdade há um queda de desempenho ao executar threads.

O programa por ser simples obteve um melhor resultado em tempo de execução na q03a em relação as outras letras, porém, em relação a programas que tem um custo de processamento maior, o uso das threads poderiam compensar e ganhar em velocidade de programas que trabalham com uma só thread.

Já na questão q03c e q03b, é perceptível o aumento de processamento do sistema ao executar com time ./prog, isso deve ao fato do próprio sistema ter que gerenciar as requisições feitas.