

Aula de Laboratório 03

Arquitetura de Computadores (2019/02)

VISÃO GERAL

Realizar um trabalho prático de análise de desempenho (Benchmark) para duas diferentes arquiteturas, Intel 64bits e ARM11 32bits. Para avaliar o desempenho, dois algoritmos devem ser implementados e executados nessas arquiteturas. Por fim, um relatório técnico deve ser elaborada contendo os resultados e a análise comparativa entre as arquiteturas.

ARQUITETURAS

1. Intel 64 bits;
2. ARM11 32bits Emulado com QEMU

ALGORITMOS

- A. Cálculo do valor de Pi aproximado - Método Monte Carlo
 - a. Um gerador de números aleatórios - Método **Linear Congruential generator (LCG)**;
- B. Multiplicação de Matrizes.

TAREFAS

1. Capturar o tempo de execução de um programa.
 - a. Utilizando a bib <time.h> ou time do shell
2. Compilar código-fonte sem e com otimização.
 - a. Usar as opções **-O0** (sem otimização) e **-O1** (com otimização) no compilador GCC.
3. Implementar o gerador de números aleatórios em linguagem C;
4. Implementar o programa que calcula o valor aproximado de Pi;
5. Implementar o programa que multiplica matrizes.

MATERIAIS NECESSÁRIOS

1. Ambiente Linux de Programação;
2. GCC Complile;
3. Qemu com imagem do ARM11 e executado OS Rapsberry;

\$:	cd qemu
\$:	qemu-system-arm -M versatilepb -cpu arm1176 -m 256 -drive file=2019-07-10-raspbian-buster-lite.img,format=raw -net nic -net user,hostfwd=tcp::10022-:22 -dtb versatile-pb.dtb -kernel kernel-qemu-4.19.50-buster -append 'rw console=ttyAMA0 root=/dev/sda2 panic=1' -no-reboot -serial stdio

4. Editor de Texto de sua preferência.

Linear Congruential generator (LCG)

LCG é uma simples exemplo de gerador de números pseudo-aleatórios.

O algoritmo é baseado na seguinte equação:

$$r_{n+1} = a \times r_n + c \pmod{m}$$

Na qual,

- r_0 é a semente inicial .
- r_1, r_2, r_3, \dots , são números aleatórios.
- a, c, m são constantes.

Inteiros aleatórios r_i são gerados no intervalo $[0, m-1]$.

Para esta atividade utilizar as seguinte constantes:

- r_0 = semente
- $m = 2^{31} - 1$
- $a = 1664525$
- $c = 1013904223$

*fonte: Numerical Recipes in C: The Art of Scientific Computing Livro por William Henry Press

Calculando o valor de Pi via método de Monte Carlo

Método de Monte Carlo é um termo utilizado para se referir a qualquer método que resolve um problema gerando números aleatórios e observando se uma dada fração desses números satisfaz uma propriedade previamente estabelecida *. O método é útil para obter soluções numéricas para problemas complicados demais para serem resolvidos analiticamente.

fonte: Weisstein, Eric W. "Monte Carlo Method." From MathWorld--A Wolfram Web Resource.
<http://mathworld.wolfram.com/MonteCarloMethod.html>

O algoritmo em pseudocódigo é:

```
01. monteCarloPi(n)
02. |   acertos ← 0
03. |   para i ← 0 até n
04. |   |   x ← sorteie um número real entre 0 e 1
05. |   |   y ← sorteie um número real entre 0 e 1
06. |   |   se(x * x + y * y < 1)
07. |   |   |   acertos ← acertos + 1
08. |   |   fim_se
09. |   fim_para
10. |   retorne 4 * acertos / n
11. fim_monteCarloPi
```

Para esta atividade, deseja-se utilizar o gerador de números aleatórios implementado na seção anterior. Além disso, o valor de n deve ser igual a 5000000 .

Algoritmo Clássico de Multiplicação de Matrizes

```
1. multMatrix( $A_{m \times n}$ ,  $B_{n \times p}$ )
2.   inicializar a matriz  $C_{m \times p}$ 
3.   para i ← 1 até m
4.     para j ← 1 até p
5.       para k ← 1 até n
6.          $C_{ij} \leftarrow C_{ij} + A_{ik} \times B_{kj}$ 
7.       fim_para
8.     fim_para
9.   fim_para
10.  retorne  $C_{m \times p}$ 
11. fim_multMatrix
```

Para esta atividade, deseja-se calcular a multiplicação de duas matrizes de dimensões 10000 x 10000.

ATIVIDADE PARA ENTREGAR

Para a entrega da atividade, os seguintes artefatos são desejados:

- ❑ Código fonte para o algoritmo A (Valor de Pi)
- ❑ Código-fonte para o algoritmo B (Multiplicação de Matrizes)
- ❑ Relatório Técnico com a análise comparativa de desempenho das duas arquiteturas

Recomendações:

- ✓ Executar os programas para coleta dos tempos pelo menos 10 repetições cada. Em seguida, obter a média e o desvio padrão do tempo.
- ✓ Preencher a Tabela A com valores obtidos da experimentação e calcular os SPEEDUPS entre as arquiteturas.
- ✓ Resultados consolidados com a média e o desvio padrão dos tempos de execução para cada um dos algoritmos, sem e com otimização de compilador, e para cada arquitetura.
- ✓ Desenvolver um texto objetivo e claro com introdução, desenvolvimento e conclusões sobre a atividade (1 página pelo menos). Não esquecer de aprofundar a análise dos resultados apresentados na Tabela A.
- ✓ Enviar os arquivos compactados em formato zip e com seguinte formato de nome: [Lab03]_Nome_Aluno.zip

Tabela A - Comparativo de desempenho.

	ARM11	Intel 64 bits	SPEEDUP
Algoritmo PI -O0	00:000 ± 0.00	00:000 ± 0.00	0.000
Algoritmo MM - O0	00:000 ± 0.00	00:000 ± 0.00	0.000
Algoritmo PI -O1	00:000 ± 0.00	00:000 ± 0.00	0.000
Algoritmo MM -O1	00:000 ± 0.00	00:000 ± 0.00	0.000
Total	00:000	00:000	0.000