COMPARAÇÃO DO TEMPO DE EXECUÇÃO DE ALGORITMOS EXECUTADOS EM COMPUTADORES COM MEMÓRIA CACHE

Izaura Pereira Pradela[1], Lucineida Nara de Andrade Oliveira[1], Marcela Dantas Queiroz[1], Walteno Martins Parreira Júnior[2]

[1]Graduandas do Curso de Engenharia de Computação da UEMG - campus de Ituiutaba (UEMG-FEIT-ISEPI)
email: izaurapradela@bol.com.br, lucineida_nara@hotmail.com, cellysindy@hotmail.com

[2]Professor dos cursos de Engenharia da Computação, Engenharia Elétrica e Sistema de Informação da UEMG - campus de Ituiutaba (UEMG-FEIT-ISEPI) – email: walteno@ituiutaba.uemg.br.

10º SEMINÁRIO DE INICIAÇÃO CIENTÍFICA E EXTENSÃO DA UEMG

Palavras-chave: algoritmo, memória cachê, tempo de execução.

1. Introdução

Avaliar a variação de tempo entre as execuções de dois algoritmos de ordenação, ocasionados por recursos internos aos processadores (memória cache).

2. Fundamentações Teóricas

2.1. Memória cachê

A memória cache surgiu quando percebeu-se que as memórias já não eram capazes de acompanhar os processadores em termos de velocidade, fazendo com que ele ficasse esperando a memória RAM liberar os dados para poder concluir suas tarefas, perdendo muito tempo em desempenho. Para solucionar este problema passou a ser utilizada a memória cache, pois o acesso é rápido a esse dispositivo e ela serve para armazenar dados que possuem uma grande probabilidade de serem utilizados novamente pelo processador.

2.2. Métodos de Ordenação

Os métodos de ordenação constituem um bom exemplo de como resolver problemas utilizando computadores. As técnicas de ordenação permitem apresentar um conjunto amplo de algoritmos distintos para resolver uma mesma tarefa.

Os algoritmos de ordenação implementados na linguagem C para comparação foram: a) por seleção e b) por inserção. São métodos de fácil compreensão e de implementação.

Ordenação por seleção:

	1	2	3	4	5	6
	O	R	D	E	N	A
i =1:	A	R	D	E	N	O
i =2:	A	D	R	E	N	O
i =3:	A	D	E	R	N	O
i =4:	A	D	E	N	R	O
i =5:	A	D	E	N	O	R

A complexidade deste método é O(n2) para qualquer que seja os valores iniciais; e o número de comparações é C(n) = ½ (n2 – n).

Ordenação por inserção:

	1	2	3	4	5	6
Vetor Inicial	O	R	D	E	N	A
i = 2	O	R	D	E	N	A
i = 3	O	R	D	E	N	A
i = 4	D	O	R	E	N	A
i = 5	D	E	O	R	N	A
i = 6	D	E	N	O	R	A
Res.:	A	D	E	N	O	R

A complexidade deste método é O(n2) para os casos médio e pior e o número de comparações para o caso médio é C(n) = ¼ (n2 + 11 n) - 3.

3. Desenvolvimento

Os algoritmos foram implementados na Linguagem C, compilados no TurboC e executados várias vezes no mesmo computador.

O computador tem a seguinte configuração: Intel (R) Pentium (R) 4, CPU 2.80GHz, 256MB de RAM, CACHE L2 512KB.

4. Resultados obtidos

4.1 Análise do algoritmo de ordenação por seleção:

Tamanho do vetor	Nro Comparações	Média (segundos)	Tempo 1	Tempo 2	Tempo 3
10	49,5	43,3s	43s	44s	43s
20	152	183,7s	187s	187s	177s
30	304,5	238,3s	312s	220s	183s
40	507	767s	761s	770s	770s

Tabela 2- número de comparações e tempo de execução

4.2 Análise do algoritmo de ordenação por inserção:

Tamanho do vetor	Nro Comparações	Média (segundos)	Tempo 1	Tempo 2	Tempo 3
10	45	32,7s	30s	34s	34s
20	190	116s	112s	118s	118s
30	435	210,3s	239s	243s	149s
40	780	335s	434s	227s	344s

Tabela 1-número de comparações e tempo de execução

Observa-se que a diferença de tempo de execução é grande para os vetores de 30 e 40, isso se deve ao fato do computador estar usando a memória cache ao invés de acessar a memória RAM na execução do algoritmo. Como esse algoritmo faz muitas comparações, o bloco da memória RAM que o processador mais utilizava foi transferida para a memória cache a partir da segunda vez em que o algoritmo foi executado, diminuindo o tempo de acesso para leitura e assim o tempo de processamento (diminuição de até 51% do tempo) do algoritmo como um todo.

5. Conclusão

Durante a análise desses algoritmos, concluímos que para desenvolver um software qualquer não é necessário apenas ter conhecimento das linguagens de programação e uma ferramenta de implementação em mãos. É necessário entender como funcionam os componentes que compõem a arquitetura dos computadores, pois o software pode não atender os valores de limite de tempo que se deseja que ele execute.

6.Bibliografia

MORIMOTO, Carlos E. Memória Cachê. 1999-2007 **Guia do Hardware**. Disponível em: <http://www.guiadohardware.net/termos/memoria-cache> acesso em 08 set. 2008.

PARREIRA JÚNIOR, Walteno Martins. **Métodos de ordenação** (Apostila). Ituiutaba: FEIT-UEMG, 2006.

[1] STALLINGS, W. **Arquitetura e Organização de Computadores**, 5ª Edição, São Paulo: Prentice Hall, , 2002. (Cap. 4)

ZIVIANI, Nivio. **Projeto de Algoritmos: Com Implementação em Pascal e C**. São Paulo: Pioneira – Thonson Learning, 2002.