

**DEPARTAMENTO DE COMPUTAÇÃO (DECOM)
LABORATÓRIO DE ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES I**

Professor: Mateus Felipe Tymburibá Ferreira

Aluno: Igor Luciano de Paula

**PRÁTICA 1: AVALIAÇÃO DE DESEMPENHO ATRAVÉS DAS UNIDADES DE
MONITORAMENTO DE PERFORMANCE (PMU)**

1) Justifique a escolha dos 4 aplicativos selecionados.

R.: Editor de texto é uma das ferramentas mais usadas em qualquer sistema operacional. Além das funcionalidades básicas, este tipo de ferramenta integra hoje muitas funções direcionadas para a programação, registo de notas e outras funcionalidades mais avançadas. Foram escolhidos quatro dos principais editores de textos usado pelos programadores. Os aplicativos selecionados foram: Gedit, Sublime Text, Atom e Brackets. Os mesmos foram escolhidos para avaliar a performance do Hardware frente a essa classe de programas, bem como a maneira que estes aplicativos se comportam no mesmo Hardware.

2) Justifique a escolha dos eventos monitorados.

R.: Os eventos selecionados foram:

- cycles
- instructions
- cache-misses
- cache-references

A escolha dos eventos para monitoramento da performance do Hardware veio para ver na prática a aplicação do conteúdo apresentado na disciplina teórica de Arquitetura e Organização de Computadores. Sendo eles: ciclos de Clock por instruções (CPI) e porcentagem de erros de acesso a memória Cache (PEAMC).

$$CPI = \frac{cycles}{instructions}$$

$$PEAMC (\%) = \frac{(cache - misses)}{(cache - references)} * 100$$

3) Apresente uma discussão dos resultados obtidos, elencando possíveis relações de causa e efeito para os eventos e o desempenho observados.

Os resultados apresentados é uma média de 3 saídas do comando 'perf' para os eventos e programas selecionados.

Resultados obtidos:

Resultados				
	Gedit	Sublime Text	Atom	Brackets
Cycles	770.742.604	16.004.663	16.582.203.430	11.672.143.530
Instructions	688.351.862	13.077.258	11.438.545.069	10.512.108.398
Cache-misses	1.583.676	78.032	58.562.905	37.627.875
Cache-references	6.895.106	205.679	236.564.818	144.452.145

A execução que requereu o menor número de ciclos para sua execução foi o Sublime Text, e Atom o maior. Da mesma forma com relação ao número de instruções, Sublime Text o menor e Atom o maior.

Com relação a referência aos acessos da memória cache, o Sublime Text obteve menores dados, já a maior quantidade foi requerida pela aplicação Atom. Outrossim, os maiores erros de acesso a memória cache foram ainda da aplicação Atom, e os menores do Sublime Text.

Análise dos resultados obtidos:

Análise dos Resultados				
	Gedit	Sublime Text	Atom	Brackets
CPI	1,12	1,22	1,45	1,11
PEAMC (%)	22,97	37,94	24,76	26,05

Podemos concluir que a aplicação Atom foi a que teve um maior número médio de ciclos de Clock por instrução, já o aplicativo Brackets o menor.

Ademais, é possível notar que a porcentagem de erros de acesso a memória Cache foi maior para o programa Sublime Text, já o programa Gedit obteve a menor porcentagem de erros.

Não é possível concluir definitivamente qual programa obtém melhor performance do Hardware, e onde o Hardware teve melhor desempenho, em um caso geral com base simplesmente nesses 4 parâmetros. Contudo, é possível observar que mesmo a aplicação Gedit sendo a mais simples de todas, a mesma necessita de um número relativamente maior de ciclos e de instruções do que a aplicação Sublime Text. Porém a aplicação Gedit obteve o segundo menor CPI e o menor PEAMC. A aplicação Sublime Text foi a que necessitou um menor número de ciclos e instruções, obteve menor número de erros e o menor número de referência em relação a memória cache, porém, também atingiu a maior PEAMC.

Em contra partida, o programa Atom demonstrou ser o que mais necessitou da capacidade do Hardware em relação a todos os quesitos analisados, bem como o maior CPI de todas as aplicações. Já a aplicação Brackets apresentou números intermediários, em relação as outras aplicações, referentes aos quesitos avaliados. Todavia, esta aplicação foi a que apresentou o menor número de CPI dentre todas outras aplicações averiguadas.