

Assembly do IA-32 em ambiente Linux

TPC9 e Guião laboratorial

Objetivo e notas

Os exercícios propostos no TPC9 introduzem conceitos **de avaliação do desempenho / hierarquia de memória**, no IA-32. Estes exercícios devem ser realizados no servidor remoto.

A resolução deverá ser entregue **impreterivelmente** no início da sessão PL, com a presença do estudante durante a sessão PL para que o TPC seja contabilizado na avaliação por participação.

Equação de desempenho do processador ($T_{exe} = \#I \times CPI \times T_{cc}$)

1. a) **(TPC)** Crie um ficheiro `media.c` com o seguinte código e execute a sua compilação para *assembly*, usando o comando `gcc -O2 -S media.c`.

<pre>#include <stdio.h> #include <stdlib.h> #define N 100 struct S { char s[4]; unsigned int a; }; void init(struct S *vec) { int i; for(i=0; i<N; i++) { vec[i].a = rand()%20; } }</pre>	<pre>int media(struct S *vec) { int i; unsigned int soma=0; for (i=0 ; i<N ; i++) vec[i].a++; for (i=0 ; i<N ; i++) soma += vec[i].a; return(soma/N); } int main() { struct S v[N]; init(v); printf("media=%d\n", media(v)); }</pre>
---	--

- b) **(TPC)** Identifique as instruções responsáveis pelos dois ciclos `for` na função `media` e apresente uma estimativa do número total de instruções executadas nessa função.
- c) **(TPC)** Repita a alínea anterior, agora compilando o código sem otimização (`gcc -O0 ...`).
- d) Qual será o ganho esperado ao compilar com otimização ($T_{exe_{O0}}/T_{exe_{O2}}$)?
- e) Quantas instruções são usadas para calcular `soma/N` nas duas versões? Porquê?

Análise da localidade no acesso aos dados

2. a) **(TPC)** Compare a forma como as variáveis `i` e `soma` são iniciadas nas versões `O0` e `O2`.
- b) Indique todas as instruções que acedem à memória na função `media`, compilada com `O2`.
- c) Nos acessos aos elementos do vector dentro da função `media` indique a localidade temporal e espacial existente.
- d) Altere o código C do programa para tirar melhor proveito da localidade temporal nos acessos ao vector. Estime o ganho em número de acessos à memória desta melhoria.
- e) Altere o código C do programa para tirar melhor proveito da localidade espacial nos acessos ao vector. Estime o ganho em termos de *cache misses* desta melhoria.

Nº

Nome:

Turma:

Resolução dos exercícios (deve ser redigido manualmente)**1. a) e 2. a)**

Indique aqui as instruções do primeiro ciclo `for` na função `media` na compilação com `-O2` e com `-O0` e as instruções que iniciam as variáveis `i` e `soma`.

	gcc -O2 ...	gcc -O0 ...
1º ciclo <code>for</code>		
Iniciação de <code>i</code> e de <code>soma</code>		

Indique aqui uma estimativa do número instruções da função `media`.

Justifique as diferenças na compilação com `-O2` e com `-O0`.