

## SCC0220 - Laboratorio de Introdução à Ciência da Computação II

Nome: Eduardo Tadahiro Nishizuka

7696393

### Exercício 1 - 07/08/2015

Análise dos códigos:

**mheap.c** - usuario entra com um valor "n" e o programa aloca na memoria 'n' espacos para palavras com 10 caracteres

Endereços do vetor de 10 strings

0x9f3070,0x9f3090,0x9f30b0,0x9f30d0,0x9f30f0,0x9f3110,0x9f3130,  
0x9f3150,0x9f3170,0x9f3190

Na saída do programa podemos observar um 'salto' (de 16bits) que ocorre entre o final da string com o ponteiro seguinte. Uma explicação para isso seria o SO alocar um espaço adicional(no caso 16bits) para cada ponteiro. Entre os caracteres de uma string, a alocação é feita sequencialmente (ex.: vetor[0][0] - 0x9f3070;  
vetor[0][1] - 0x9f3071;)

**mstack.c** - programa que demonstra o uso da memória stack

Neste código podemos perceber que a variável global e a *static* são alocadas numa parte da memória diferente da pilha(stack). Observamos que os endereços são bem diferentes das variáveis normais declaradas na main. Abaixo podemos ver onde foram alocadas as variáveis

#### Endereço de Variáveis

int soma (na função): 0x7ffeb1be674c

char a (na função): 0x7ffeb1be675f

static int num(na função): 0x601040

int count(global): 0x601048

char a (na main): 0x7ffeb1be678a

char b (na main): 0x7ffeb1be678b

char c (na main): 0x7ffeb1be678c

**overflow.c** - programa que analisa a stack quando ocorre um overflow(estouro de memória)

No exemplo podemos observar o quanto que faz a pilha estourar, comparando o uso de variáveis e parâmetros de função. Podemos ver que quando mais alocação, mais rápido estoura a pilha.

Tipo de estouro	# de recursões	Tempo de estouro
função rec. simples	523833	11.568s
usando 2 param. long double	174487	3.575s
usando 1 param. long double	261781	5.51

**recurs.c** - programa para demonstrar o uso de recursão e comparação com iteração

Este código mostra a diferença na performance entre funções recursivas e iterativas. No caso do fibonacci, apesar da função recursiva desse algoritmo ser mais simples de implementar, ele perde com enorme diferença para a versão iterativa, o que ressalta a importância da análise do algoritmo antes de implementá-lo.

N	Tempo recurs.	Tempo Iterativ	# iterações recursivas	# iterações vers. normal
10	0.000035s	0.0000001s	9	10
15	0.000058s	0.0000001s	109	15
20	0.000158s	0.0000001s	1219	20
25	0.001130s	0.0000001s	13529	25
30	0.007857s	0.0000001s	150049	30
35	0.103931s	0.0000001s	incontavel	35
40	1.012813s	0.0000001s	incontavel	40
45	10.823347s	0.0000001s	incontavel	45

**operacaobit.c** - programa para demonstrar operações bit a bit com o uso dos parametros da funcao main

Este codigo visa mostrar a diferença entre as operações de deslocamento de bits e a multiplicação. Porém, o compilador(gcc) otimiza o código, transformando a multiplicação em deslocamento. Por isso, não foi possível ver diferenças entre as operações.