**Trabalho 03: Matrizes Esparsas**

SSC0300 – Linguagens de Programação e Aplicações

**Aluno:** Felipe Augusto Senger Lopes de Souza - No USP: 5513317

**Introdução:** Este projeto contém o exercício do 3º Trabalho da disciplina de Linguagens de Programação e Aplicações. Ele é um programa em C que implementa uma matriz esparsa.

**Definição:** A estrutura da matriz esparsa é utilizada para armazenar matrizes de grandes ordens no computador, que obedecem a uma condição: possuem uma grande quantidade de zeros (0). Quando uma matriz tem ordem de cerca de 106 x 106, por exemplo, é necessário muita memória para armazená-la. Entretanto, se ela possui muitos zeros (cerca de 90% da matriz), é possível implementar uma estrutura que otimize o uso da memória, armazenando apenas algarismos diferentes do 0. Essa implementação usa uma série de listas encadeadas para armazenar as linhas da matriz, e se uma determinada posição não consta na lista, assume-se que ela é zero. Mais detalhes sobre a utilização de listas foram incluídos a seguir.

**Descrição do Projeto:** Todo o programa foi desenvolvido no Windows 7 64 bits, no programa Dev C++. O compilador utilizado foi o default do Dev C++, v5.10, nas configurações TDM-GCC 4.8.1 64-bit Release. Todas as bibliotecas usadas foram as consideradas padrão do Dev C++. A compilação não requer nenhum parâmetro especial e os programas foram compilados usando o botão de compilação do Dev C++.

Segue a descrição de como o programa funciona e é executado:

**Algoritmo:** O programa recebe a entrada de dois números, que representam o número de linhas e o número de colunas que a matriz irá possuir. Ao receber esses números, ele utiliza o primeiro (número de linhas) para criar uma lista encadeada “base”, fazendo uso de *structs*. Ela será a base para o início de cada uma das linhas da matriz (uma ilustração se encontra mais adiante). Após isso, o usuário pode inserir números em sua matriz.

Cada item da lista base está aterrado, apontando para *NULL****,*** que posteriormente irão apontar para o início de cada lista que representa as linhas da matriz, que são listas utilizando um segundo tipo de *struct* definido. Quando o usuário insere um elemento na linha, o programa chama a função *malloc* para alocar espaço para o novo item da matriz. Então ele é inserido em sua respectiva linha, e possui um marcador “coluna” para saber de qual coluna ele pertence.

OBS: Deve-se observar que a numeração das linhas e colunas da matriz inicia-se em 0. Portanto uma matriz de 4 linhas e 5 colunas tem suas linhas numeradas de 0 a 3 e colunas numeradas de 0 a 4.