



Prof. Esther Colombini
esther@ic.unicamp.br

Atividade prática 1: Funções, Ponteiros e Strings

Entrega: 30/08/2023

1 Objetivo

O objetivo deste trabalho é permitir que o aluno aplique seus conhecimentos em C++ para manipulação de strings, funções, alocação de memória, ponteiros e interface com o usuário.

2 Passos da Atividade

Passo 1: Configuração do Ambiente:

Antes de iniciar a atividade, certifique-se de que o ambiente de desenvolvimento C++ está configurado. Você pode usar qualquer IDE de sua escolha (Visual Studio Code, Code::Blocks, Dev-C++, etc.) ou até mesmo um editor de texto simples com o compilador g++ instalado. Lembre-se que a instalação do compilador g++ é dependente de plataforma.

Passo 2: Escrevendo o Código:

Desenvolva um programa que utilize uma matriz de caracteres para simular uma tela de arte ASCII com uma dimensão inicial de 20x40, para isso use alocação dinâmica de memória. Implemente uma função destinada a preencher a tela com espaços em branco. Use essa função para criar a matriz inicial do programa. Crie uma função que permita ao usuário escolher um caractere ASCII e indicar uma posição na tela onde deseja desenhar. Adicione uma função para que o usuário possa ajustar o tamanho da tela, modificando as dimensões da matriz. Certifique-se de gerenciar de maneira adequada a alocação e liberação dinâmica de memória ao redimensionar a matriz.

Para exibir a arte ASCII gerada, crie outra função responsável por apresentar a matriz na tela. Por fim, estabeleça um loop de interação com o usuário, fornecendo-lhe as seguintes opções numeradas: 1) Desenhar, 2) Redimensionar, 3) Visualizar, 4) Sair do Programa.

Certifique-se de comunicar as opções de forma clara e fornecer números correspondentes a cada alternativa. Caso o usuário introduza uma opção inexistente, assegure-se de exibir uma mensagem de erro apropriada.

Passo 3: Criando o Arquivo Makefile:

Um arquivo Makefile desempenha um papel fundamental na compilação de projetos maiores e mais complexos em linguagens de programação como C++ ou C. Ele automatiza o processo de compilação e torna a construção do projeto mais eficiente e organizada, permitindo o gerenciamento de dependências, a compilação incremental, a compatibilidade de plataforma e a documentação do processo de compilação.

Para o nosso projeto, crie um arquivo Makefile para compilar o programa matriz.cpp em um executável chamado **matriz**. Crie uma opção de apagar todos os .obj e todos os .exe criados.

Passo 4: Compilação e Execução:

Abra um terminal no diretório onde os arquivos matriz.cpp e Makefile estão localizados. Use os comandos make para compilar o programa e, em seguida, executar o executável gerado.