

Relatório - Atividade Avaliativa 2

ALUNOS : Kauan Carlos Campos, Matheus felipe Pereira Lopes

Objetivo

O objetivo desta atividade é desenvolver um programa em Assembly capaz de realizar operações aritméticas entre dois operandos informados via linha de comando, por meio do operador correspondente.

Descrição do Programa

O programa desenvolvido integra funções escritas em linguagem C, sendo estas: `fprintf`, `fopen`, `atof` e `fclose`, sendo assim, é necessária a utilização do compilador **GCC** para a geração do executável. Para compilar e executar o código, os seguintes comandos devem ser utilizados:

```
nasm -f elf64 [LM_pratica_02]KauanCampos_MatheusLopes.asm && gcc -m64 -no-pie [LM_pratica_02]KauanCampos_MatheusLopes.o -o exercicio2
```

A execução do programa requer a passagem de três parâmetros via linha de comando, seguindo o formato:

```
./exercicio2 operando1 operador operando2
```

Dessa forma, o código opera como uma calculadora, executando a operação aritmética especificada entre os operandos fornecidos.

Armazenamento dos Resultados

O resultado obtido é registrado em um arquivo de texto denominado "**resultados.txt**", o qual mantém um histórico cumulativo das operações realizadas.

Estrutura do Código

O programa é composto por um total de **oito funções**:

- **Função principal (main):** Responsável por atribuir aos registradores os parâmetros recebidos via linha de comando, abrir o arquivo "**resultados.txt**" e

direcionar o fluxo de execução para a função correspondente à operação solicitada, por meio de uma estrutura similar a um **switch**.

- **Funções aritméticas:** Quatro funções distintas realizam as operações de soma, subtração, multiplicação e divisão.
- **Funções de escrita:** Duas funções específicas são responsáveis por registrar o resultado no arquivo de saída. Uma delas escreve os cálculos bem-sucedidos, enquanto a outra trata os casos de erro ou insucesso.

Tratamento de Erros

O código inclui mecanismos para tratar erros durante a execução:

1. Erro na abertura do arquivo:

- Caso o arquivo "**resultados.txt**" não possa ser aberto pela função **fopen**, o fluxo de execução é direcionado para a label **erro1**.
- Dentro dessa label, o fluxo de execução é levado até a função de escrita a fim de exibir no arquivo de saída a seguinte mensagem:

Erro ao abrir o arquivo!!!

2. Divisão por zero:

- Se o operador for divisão "d" e o segundo operando for zero, o fluxo de execução é redirecionado para a label **erro2**.
- Nessa label, a seguinte mensagem é registrada no arquivo de saída:

"%.2lf %c %.2lf = funcionalidade não disponível"

Dessa forma, o programa é capaz de lidar com erros comuns que poderiam comprometer sua execução.

Conclusão

Como conclusão, ficam os aprendizados obtidos sobre subrotinas em linguagens de montagem, uso de funções externas, uso correto da stack e o tratamento de floats e doubles.

