Trabalho 2
Software Básico

Introdução

O trabalho consiste em implementar em IA-32 um calculadora de números de 2 precisões diferentes.

O trabalho pode (e recomenda-se) ser feito em grupo de 2 alunos. Deve ser entregue somente o código e um arquivo README indicando o SO utilizado e como compilar o programa. O trabalho deve rodar em LINUX com o NASM e LD.

Não pode usar a biblioteca IO.MAC (IO.O)

Especificação

Faça um programa em IA-32 que representa uma calculadora que possa realizar as operações de SOMA, SUBTRAÇÃO, MULTIPLICAÇÃO, DIVISÃO, EXPONENCIAÇÃO e MOD.

O programa deve inicialmente perguntar para o usuário seu nome:

> *Bem-vindo. Digite seu nome:*

Depois dar uma mensagem de boas vindas:

> *Hola, <nome do usuario>, bem-vindo ao programa de CALC IA-32*

Depois deve perguntar q precisão:

> *Vai trabalhar com 16 ou 32 bits (digite 0 para 16, e 1 para 32):*

Depois deve mostrar um menu:

ESCOLHA UMA OPÇÃO:
- 1: SOMA
- 2: SUBTRACAO
- 3: MULTIPLICACAO
- 4: DIVISAO
- 5: EXPONENCIACAO
- 6: MOD
- 7: SAIR

A opção 6 deve terminar o programa.

Todas as mensagens de TEXTO devem ser mostradas usando uma ÚNICA função de saída de dados de string. Esta função deve receber pela pilha o ponteiro da variável global que contém o string e a quantidade de bytes a serem escritos. Não deve ter retorno. O programa principal deve UNICAMENTE chamar funções, e receber as saídas das funções que podem ser utilizadas como entrada de outras (se necessário). O programa principal NÃO deve processar dados nem fazer nenhuma operação de entrada e saída de dados diretamente. As funções de entrada e saída de dados podem ser chamadas pelo programa principal e outras funções.

As únicas variáveis globais podem ser os ponteiros para os strings de texto das mensagens, uma variável para receber o nome do usuário, uma variável para receber a resposta de precisão e uma variável para receber a opção do menu. TODAS as outras variáveis DEVEM ser LOCAIS NA PILHA.

Para o menu a ideia é chamar a função de saída de string uma vez por linha. A leitura do teclado deve ser feita por 2 funções: uma para ler strings, e outra para ler números. Porém, a de ler números deve ter duas versões: a de 16 bits e 32 bits. Deve trabalhar com números positivos e negativos.

Ao receber a opção do menu, que pode ser qualquer número entre 1 a 6, deve ir para uma função que vai executar a operação requerida. Tal função deve pedir 2 números inteiros (que pode ter o sinal negativo). Deve-se assumir que são de 16 ou 32 bits de acordo com a opção digitada pelo usuário. Devem ser armazenados em variáveis locais. Deve mostrar o resultado final. Esperar o usuário digitar ENTER e novamente mostrar menu de opções. Até o usuário digitar a opção de sair.

Nas operações de expoente e multiplicação deve-se verificar se o resultado gerou overflow. E caso positivo mostrar a mensagem: OCORREU OVERFLOW. E fechar o programa.

Todo parâmetro passado para funções deve ser na pilha. Toda saída deve ser feita pelo registrador EAX. Devem ser usadas as instruções de multiplicação e divisão do IA-32 para as opções 3, 4, 5 e 6 (conforme necessário). Não pode trocar multiplicação/divisão por outras operações.

A função principal e funções de entrada e saída de dados devem estar no mesmo arquivo CALCULADORA.ASM. Porém as operações deve estar cada uma num arquivo separado (ex.: SOMA.ASM, DIVISAO.ASM, etc.) o programa deve ser compilado e ligado para gerar um único executável. Para isso deve estar no arquivo README as instruções de compilar e ligar.

NOTA: lembre que não cumprir com parte da especificação não implica zerar o trabalho. Mande o que foi feito. É mais importante cumprir com o requerimento das duas precisões de dados do que o requerimento das variáveis locais, arquivos ASM separados e verificação de overflow. Soma, subtração, multiplicação e divisão valem mais que MOD e exponenciação.
Fazer perfeitamente as mensagens de entrada e menu, ler corretamente as respostas e executar perfeitamente 3 operações entre soma, subtração, divisão e multiplicação em ambas precisões garante uma nota 6.0.

Porém, é OBRIGATORIO que tudo seja implementado por função e que as funções recebam argumento pela pilha (não fazer isso zera o trabalho).