

Trabalho Prático 1
Software Básico
Prof. Bruno Macchiavello
1 o Semestre de 2021

1 Introdução

O trabalho consiste em implementar um simulador do Assembly inventado em Ia-32 combinando com C/C++

2 Objetivo

Fixar o funcionamento de um processo de tradução. Aprender a programar em IA-32.

3 Especificação

3.1 Simulador

O simulador deve ser feito em uma pequena parte em C/C++ e o resto em Assembly ia-32. O simulador deve receber como entrada um arquivo objeto do assembly inventado. A princípio deve ser o arquivo objeto da saída do seu Trabalho 1. Porém, deve funcionar com qualquer arquivo objeto que esteja no formato indicado no trabalho 1 e utilizando a linguagem máquina do nosso assembly inventado.

O simulador deve receber o nome do arquivo objeto por linha de comando (ex.: /simulador myprogram.o). O programa principal deve ser feito na linguagem C/C++, o programa principal deve simplesmente ler o nome do arquivo a ser lido e chamar uma função em Assembly 32. O resto do programa deve ser feito em Assembly 32.

O simulador deve funcionar de forma similar ao simulador que foi proporcionado para fazer o Trabalho 1. Deve mostrar todas as saídas no monitor. Assim, como o simulador do trabalho 1. Porém, além disso, deve mostrar na tela o tamanho de bytes do arquivo de saída no final e deve criar um arquivo de saída. Que deve ser parcialmente um Disassembler. O arquivo de saída deve mostrar as instruções do Assembly inventado utilizadas no programa e na ordem correta. Somente instruções, sem operando e sem diretivas, como por exemplo:

INPUT
STORE
LOAD
SUB
MULT
JUMP
STOP

O simulador deve utilizar funções. Além dele ser uma função a ser chamada pelo programa principal. Ele deve utilizar pelo menos as seguintes funções:

- 1 - A leitura do arquivo objeto deve ser feita por uma função que deve receber por argumento (pelo menos) o ponteiro para um string com o nome do arquivo. Fechar o arquivo depois de ler
- 2 - A escrita do arquivo deve ser feita por uma função. Que deve receber como argumento (pelo menos) o ponteiro para um string com o nome do arquivo de objeto. A função deve modificar o arquivo o nome de forma de mudar a extensão para .diss e manter o nome original.
- 3 - o resto do programa pode ou não ser feito em funções. As funções podem receber mais argumentos se necessário. Deve devolver em EAX a quantidade de bytes do arquivo escrito (que deve ser mostrado na tela no final pelo programa principal).
- 4 - Toda passagem de argumentos deve ser feita pela pilha.

Notar que o simulador funciona similar a uma estrutura de "Switch". Vamos supor que o arquivo de objeto seja lido numa estrutura de array, então passando pelo array os códigos são lidas e tratados numa estrutura estilo switch, onde cada OPCODE seria um "case", por exemplo: caso seja o opcode 1, então deve-se ler a próxima posição do array que indica onde deve ser lido o operando no mesmo array, e ser somado o valor do operando com o valor do acumulador.

Não é necessário que um registrador específico seja utilizado como se fosse o acumulador da nossa máquina inventada. Pode ser utilizado UM ou Vários registradores para simular isso.