Software Básico
Trabalho Prático 1
Montador

Alunas:

Ingrid Elisabeth Spangler
Ingrid Rosselis Sant'Ana da Cunha

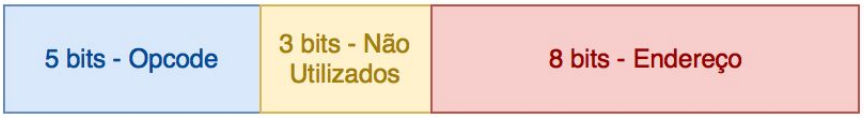
Introdução:

O trabalho prático 1 da disciplina de Software Básico é fazer a implementação de um montador para a máquina Swombat, simulada no programa CPUSim. Um montador é um programa que executa o processo de tradução por dois passos para um código cuja linguagem-fonte é Assembly e a linguagem-alvo é a própria linguagem de máquina.

A máquina Swombat utiliza palavras de 16 bits e faz operações sobre inteiros, possui memória de 256 posições com 8 bits cada célula e uma pilha de uso geral, para chamadas de procedimentos e para salvar o program counter nas instruções de call e de return. Possui também 8 registradores com 16 bits de comprimento de uso geral e 8 registradores de uso específico. As instruções possuem formato parecido com o do MIPS, tendo o tamanho fixo de 16 bits e suas divisões correspondentes ao tipo ao qual a instrução pertence (baseado no número de operandos):

- 0 operandos:
  - Especiais:

- 1 operando:
  - Tipo J:



- 2 operandos:
    - Tipo R (aritméticas):

    - Tipo I (imediatos):

    - Tipo J (branches):

Modelagem:

Implementamos um montador de dois passos que executa um processo de tradução completa de um código em linguagem assembly dividido em 4 campos: rótulo, operação, operandos e comentários.

O primeiro passo consiste em um pré-processamento do módulo, onde os rótulos são separados em uma tabela junto com o ILC (Instruction Location Counter) para evitar problemas de referenciamento posterior. O segundo passo é a tradução do módulo para linguagem de máquina propriamente dita, isto é, cada linha do arquivo de entrada é separada em seus 4 componentes e cada um deles é tratado separadamente até que seja traduzido para sua forma binária. Eles são tratados da seguinte forma:

- Rótulos: Quando estão como símbolos representando uma linha de código são tratados no primeiro passo do processo de montagem, já quando estão como operandos são tratados da mesma forma que outros símbolos.
- Operação: a operação é separada e é verificado em uma tabela quantos operandos esta operação aceita e qual o seu código, de acordo com a especificação. A seguir, a operação é traduzida e é escrita no primeiro campo da instrução.
- Operandos: Os operandos são salvos e posteriormente identificados e substituídos pelos seus valores numéricos correspondentes, o número do registrador se for um, seu próprio valor se for um literal, ou, se for uma label, o endereço correspondente.
- Comentários: Comentários são ignorados.

Decisões de planejamento:

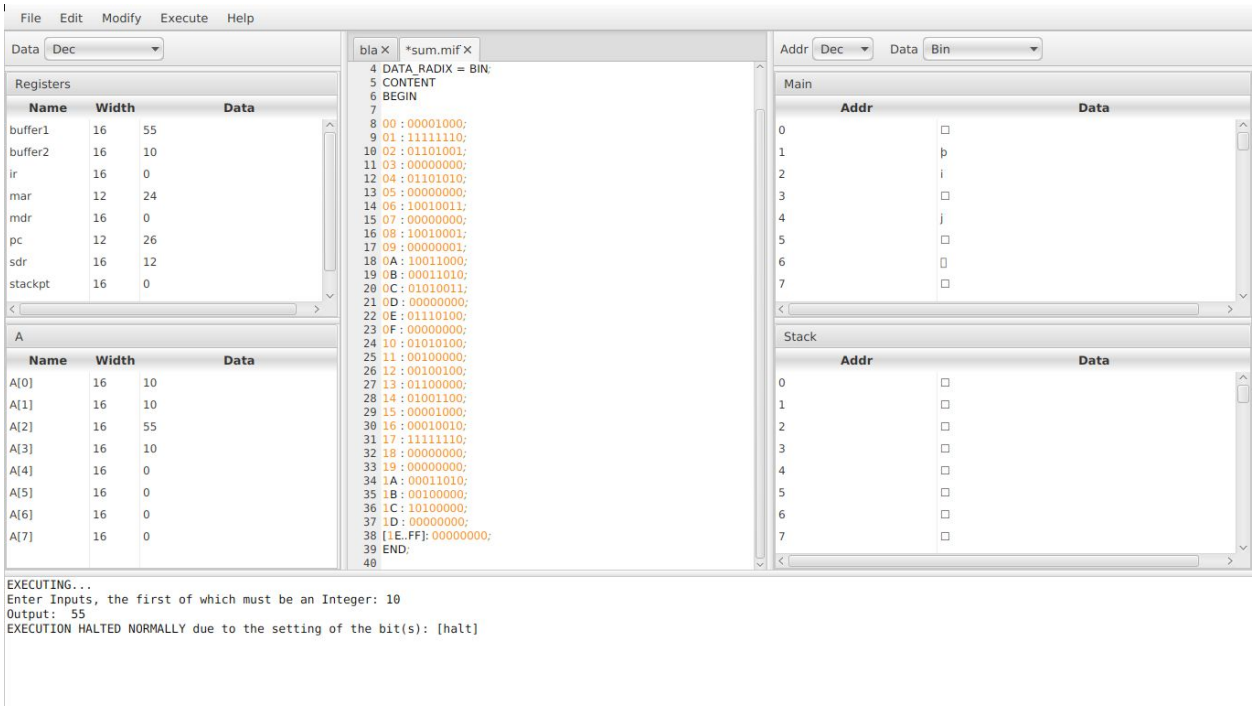
A linguagem utilizada no desenvolvimento do projeto foi C++, escolhida por conter ferramentas úteis e maior facilidade no tratamento de ponteiros, o que C puro não oferece. As ferramentas e estruturas de dados próprias de C++ presentes no código são:

- Classes: A máquina e suas funções estão em uma classe, em um namespace, assim ela pode ser utilizada em qualquer programa desta maneira, como uma biblioteca.
- Busca por expressão regular: Utilizamos esta busca para tratar qualquer tipo de indentação e separar os símbolos eficientemente, diferenciar labels e comentários de operações e operadores.
- Map: estrutura de dicionário que utiliza uma árvore red-black para armazenar os pares de chave e valor. O dicionário guarda o nome dos operadores e seus respectivos opcodes, juntamente com o número de operandos.
- Decidimos implementar o formato .mif, pela legibilidade e minimalismo das instruções em comparação com o formato .hex

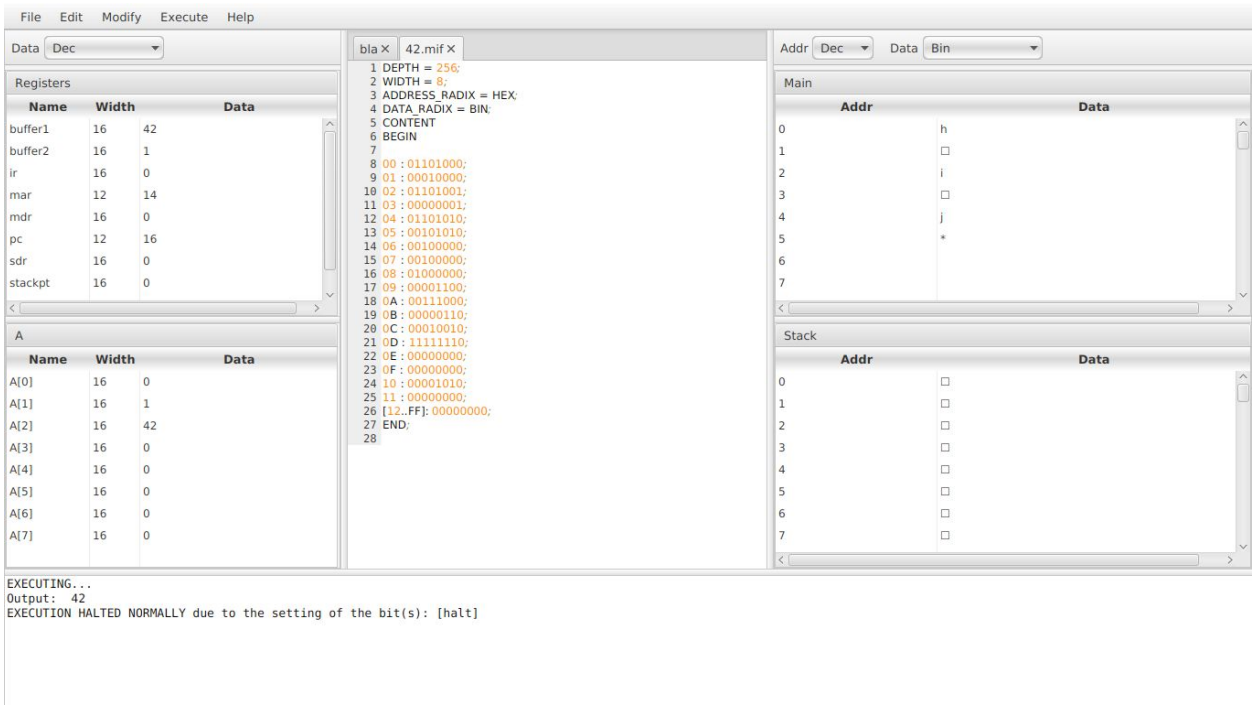
Análise de resultados:

O programa foi testado tanto com o programa disponibilizado no moodle como com os dois programas de testes escritos por nós: sum.a e 42.a.

Sum.a pede um valor de input para o usuário e retorna a soma de todos os números a partir do 1 até o número dado, usando uma function call:



42.a performa operações com dados .data da memória e imprime a resposta, que é sempre 42:



Conclusão:

Este trabalho foi um excelente aprendizado tanto pelo conteúdo quanto pela forma: obtemos maior conhecimento sobre o funcionamento de um montador, e nos proporcionou a oportunidade de desenvolver um projeto grande em cooperação, usando linguagem e conceitos ambos novos para nós.