Documantação - Montador

Trabalho Prático de Compiladores

Gustavo Dias de Oliveira[1] and Lucas Augusto Araújo Aguiar[2]

[1,2]Universidade Federal de Minas Gerais

July 30, 2021

1 Intrudução

O trabalho consiste na implementação de um Montador para uma Máquina Virtual customizada utilizando a linguagem C/C++. O Montador recebe como parâmetro um arquivo texto com um programa Assembly e gera como saída um arquivo no formato aceito pela Máquina Virtual, levando em consideração a tabela de instruções proposta referente a essa máquina.

2 Fluxo do código

2.1 Entrada

Para pegar a entrada, abriu-se e leu-se o arquivo texto passado como parâmetro, e armazenou-se cada linha em um vetor de *strings*. Após isso, eliminou-se todas as linhas vazias, removeu-se todos os comentários e espaços desnecessários presentes no início e final das linhas lidas.

2.2 Extraindo Operações

Depois de formatar a entrada lida, extraiu-se o *label*, simbolo da operação, primeiro operando e segundo operando de cada linha, sendo que, se tal atributo não existe, era adicionado uma *string* vazia como seu valor. Para armazenar essas informações foi criada uma classe **Operações**, que possui esses atributos.

2.3 Montador

Criou-se uma classe **Assembler**, onde nela armazenou-se a tabela de conversão, de operações para linguagem de máquina, como um atributo. Chamou-se então a função para conversão do código e passou como parâmetro as operações extraídas anteriormente. A função compara os tipos de operação; Se for do tipo *END*, encerra função. Se for do tipo *WORD*, adiciona-se na cadeia de saída o valor do inteiro de inicialização. Se for dos tipos descritos na tabela, primeiramente checa a quantidade e tipo de operando, se foi do tipo registrador, apenas realiza a conversão para a linguagem de máquina proposta e, se for do tipo memória, realiza a busca por todo vetor de operações até encontrar um *label* com o mesmo valor. Após isso, adiciona-se as posições de memória que o *PC* deve caminhar.

2.4 Saída

Para a saída do programa, ela deve ser exibida na sáida padrão. Assim, imprimiu-se a linha *"MV-EXE"*. Passou-se para a linha de baixo e imprimiu-se o tamanho total do código, seguido do endereço de carregamento, seguido do valor inicial da pilha e por fim o entry point do programa, que foi obtido durante a montagem do código. Ao fim, em outra linha, imprimiu-se as intruções para serem executadas pela Máquina Virtual.

3 Testes

Depois de finalizada a lógica acima, criou-se um novo comando no *Makefile* chamado de *test*. O intuíto da funcionalidade foi realizar testes de forma automatizada para garantir que cada parte do fluxo do código esteja executando de acordo. Testou-se casos especiais como entrada possuindo uma grande quantidade de espaços em branco no início, final e entre palavras de uma mesma linha, casos em que linhas eram constituídas apenas de comentários, casos em que havia linhas com operações e comentários, casos com várias linhas em branco e outros.

Ademais, implementou-se na linguagem assembler da Maquina Virtual alguns testes, como o teste da mediana, em que pode-se comparar a saída do programa com a saída esperada, a qual foi encontrada seguindo a tabela disponibilizada no arquivo que contém a descrição do trabalho.

4 Conclusão

Assim, foi-se implementado o montador para a máquina virtual previamente projetada, na qual utilizou-se os conceitos aprendidos no curso a respeito da montagem de programas. Pôde-se colocar na prática como funcionaria a montagem de um código, recebendo instruções em Assembly como entrada e imprimindo um código reconhecido pela máquina.