

Trabalho de Compiladores

Matheus Zap Souza, Vitor Bernstorff Clemes

1. Implementação

Nesta etapa do projeto, foi adicionado a partir final proposta ao projeto. Para implementar a parte que envolvia Java Byte Code, foi criado um novo arquivo chamado `bytecode.h`. Este arquivo possui as funções de escrita no arquivo “`output.j`” que depois será interpretado pela Java Virtual Machine. As funções são bem simples e possuem em sua maioria o nome do próprio bytecode. Logo no começo da execução, é gerado o cabeçalho. São permitidos 100 variáveis locais armazenadas e 100 carregamentos simultâneos na pilha. Foi utilizada uma estrutura de lista encadeada para implementação da pilha e do armazenamento das variáveis locais.

Para que fosse possível implementar, o arquivo “`list.c`” e seu cabeçalho “`list.h`” sofreram algumas modificações. A própria estrutura da lista teve que se adequar a novos conteúdos para adaptar-se à nova etapa. Além disso, novas funções de manipulação da estrutura de dados tiveram que ser implementadas.

2. Dificuldades do Trabalho

Até o momento, não está sendo possível repassar os valores fornecidos para as instruções do Jasmim. Mais especificamente, o código Jasmin é NÃO FUNCIONAL pois os valores escritos no arquivo a ser lido, apesar de possuírem a lógica correta, não são os valores corretos e esperados. Suspeita-se que seja um erro de implementação no arquivo “`compilador.y`” nas relações de `<ival>`, `<fval>`, `<sval>`, etc.

3. Configuração do Ambiente

Os analisadores léxicos e sintáticos foram escritos através de um editor de texto comum, denominado Visual Studio Code em sua versão para sistemas Linux (disponível em <https://code.visualstudio.com/download>). Qualquer extensão já instalada não afetou o desenvolvimento e a execução do trabalho.

A versão do Flex utilizada foi a 2.6.4, enquanto a versão do Bison utilizada foi a 3.5.1. Todo o desenvolvimento do código foi implementado em sistemas Linux, nas distribuições Ubuntu 20.04 LTS e Lubuntu.

Para facilitar os testes, foi criado dentro da pasta “`analisador_sintatico`” o arquivo `run.sh`. Este arquivo é um script que contém todos os comandos necessários para a compilação e execução do programa e pode ser acessado via terminal através do comando:

```
$ bash ./run.sh
```

Além disso, dentro da mesma pasta, foi criado um arquivo teste.txt, onde podem ser escritas as entradas que deverão ser executadas na execução do comando.

Então, o arquivo output.j será gerado, contendo as instruções de execução da Java Virtual Machine.

Para rodar o Jasmim, após utilizar o comando acima, executar:

```
//COMANDOS JASMIN
```