

Introdução ao Flex e Bison

Lucas Begnini Costa // André Moreira // George Lucas // Luis Sergio
// Walderi

Universidade do Estado do Amazonas

May 15, 2013



Introdução ao Flex e Bison

Lucas Begnini Costa // André Moreira // George Lucas // Luis Sergio
// Walderi

Universidade do Estado do Amazonas

May 15, 2013

Sumário

1 Introdução

2 FLEX

- Instalando Flex no Linux
- Criando um arquivo para o Flex

3 Bison

- Instalando Bison no Linux
- Criando um arquivo para o Bison

4 Calculadora

- Arquivo Flex da Calculadora
- Arquivo Bison da Calculadora
- Árvore Sintática da Calculadora

Introdução

Este trabalho tem como intuito apresentar a construção de uma calculadora que será utilizada através de linha de comando e interpretada através de:

- Flex - Analisador léxico

Mas para apresentar a calculadora será necessário uma breve apresentação a respeito de Flex e Bison

Introdução

Este trabalho tem como intuito apresentar a construção de uma calculadora que será utilizada através de linha de comando e interpretada através de:

- Flex - Analisador léxico
- Bison - Analisador semântico

Mas para apresentar a calculadora será necessário uma breve apresentação a respeito de Flex e Bison

FLEX

Como falado anteriormente o Flex é uma ferramenta para criação de analisadores Léxicos que procura padrões de escrita através de expressões regulares definidas dentro dos analisadores léxicos. Como resultado, depois da compilação do arquivo flex feito, é gerado um arquivo em C que valida todas as expressões regulares definidas no arquivo que passará pelo Flex

Instalando Flex no Linux

Antes de fazer qualquer coisa para criação do arquivo flex é necessário inicialmente a instalação do compilador flex no sistema operacional, no caso utilizado foi o Linux Ubuntu 13.04. Para instalação é necessário abrir o terminal e por linha de comando digitar:

```
$ sudo apt-get install flex
```

Criando um arquivo para o Flex

Para a criação de um arquivo para o flex não é necessário nenhuma extensão específica, podendo ser o que o desenvolvedor quiser, mas como um ato de boa pratica de programação, padronizou-se como `.lex` ou `.l` para melhor distinguir de outros arquivos.

Criando um arquivo para o Flex

Dentro do programa há duas partes, separado por "%%" dividindo o programa em duas partes: A primeira para delimitação das expressões regulares que serão consideradas, e a segunda as ações que serão tomadas caso encontre as expressões regulares definidas anteriormente. Como podemos ver no exemplo seguinte: **Exemplo de Flex**

Criando um arquivo para o Flex

Para compilação de um arquivo flex é necessário 2 etapas:

- Compilação em flex primeiro do seguinte modo, pela linha de comando do linux:

Na primeira compilação ele pega arquivo de flex e gera um arquivo .C que vai validar as expressões regulares O parametro "-lfl" passado na compilação do gcc serve pra identificar que foi usado o flex pra gerar o .C

Criando um arquivo para o Flex

Para compilação de um arquivo flex é necessário 2 etapas:

- Compilação em flex primeiro do seguinte modo, pela linha de comando do linux:
- `flex -o nome-do-programa.lex.c nome-do-programa.lex`

Na primeira compilação ele pega arquivo de flex e gera um arquivo .C que vai validar as expressões regulares O parametro "-lfl" passado na compilação do gcc serve pra identificar que foi usado o flex pra gerar o .C

Criando um arquivo para o Flex

Para compilação de um arquivo flex é necessário 2 etapas:

- Compilação em flex primeiro do seguinte modo, pela linha de comando do linux:
- `flex -o nome-do-programa.lex.c nome-do-programa.lex`
- Em seguida é necessária a compilação em GCC com o seguinte comando

Na primeira compilação ele pega arquivo de flex e gera um arquivo .C que vai validar as expressões regulares O parametro "-lfl" passado na compilação do gcc serve pra identificar que foi usado o flex pra gerar o .C

Criando um arquivo para o Flex

Para compilação de um arquivo flex é necessário 2 etapas:

- Compilação em flex primeiro do seguinte modo, pela linha de comando do linux:
- `flex -o nome-do-programa.lex.c nome-do-programa.lex`
- Em seguida é necessária a compilação em GCC com o seguinte comando
- `gcc -o nome-do-programa nome-do-programa.lex.c -lfl`

Na primeira compilação ele pega arquivo de flex e gera um arquivo .C que vai validar as expressões regulares O parametro "-lfl" passado na compilação do gcc serve pra identificar que foi usado o flex pra gerar o .C

Bison

O Bison é um analisador que lê um arquivo que codifica gramática uma linguagem e gera uma rotina de "parsing", agrupando os tokens em uma sequência de interesse e invoca ações para esses tokens

Instalando o Bison no Linux

FALAR SOBRE A INSTALAÇÃO DO BISON

Criando um arquivo para o Bison

Todos os arquivos de bison são terminados em `.y`

Criando um arquivo para o Bison

FALAR SOBRE COMO É ORGANIZADO O BISON E FAZER UM LINK COM ALGUM EXEMPLO "HELLO WORLD" PARA ILUSTRAR

Criando um arquivo para o Bison

FALAR SOBRE COMO É PARA COMPILAR UM ARQUIVO DO BISON
E OQUE RESULTA? SE RESULTA EM ALGUMA COISA? E SE DAR DE
COMPILAR.. ESSAS COISAS!

Calculadora

FALAR COMO FUNCIONARÁ A CALCULADORA

Arquivo Flex da Calculadora

FALAR COMO ESTÁ ORGANIZADO O ARQUIVO FLEX DA CALCULADORA E APONTAR UM LINK AO ESTILO DO SLIDE 07 PARA O NOSSO GIT, PARA MOSTRAR O ARQUIVO FLEX LÁ... PRA MOSTRAR QUE ESTAMOS FAZENDO SERVIÇO

Arquivo Bison da Calculadora

FALAR COMO ESTÁ ORGANIZADO O ARQUIVO BISON DA CALCULADORA E APONTAR UM LINK AO ESTILO DO SLIDE 07 PARA O NOSSO GIT, PARA MOSTRAR O ARQUIVO FLEX LÁ... PRA MOSTRAR QUE ESTAMOS FAZENDO SERVIÇO

Árvore Sintática da Calculadora

FALAR Oq é ARVORE SINTÁTICA E COMO ELA FUNCIONA

Árvore Sintática da Calculadora

EXEMPLIFICAR A ARVORE COM OS EXEMPLOS

1

1+3

1+3*4

1+3*4-2

Árvore Sintática da Calculadora

EXEMPLIFICAR A ARVORE COM OS EXEMPLOS

$(1+3)*4-2$

$(1+3)*(4-2)$

$(1+3)*4(-2)$