Analisador léxico para a linguagem C- -

Demonstrando o software

Victor E. Almeida Marco A. G. Pedroso

UNIOESTE

31 de Janeiro de 2022

Conteúdo

Softwares utilizados

- **Sistema para compilar**: GNU make e gcc,
- **Linguagem de programação**: C11,
- **Gerador de analisador léxico**: GNU flex.

Exemplo mínimo do flex

```
1   %{
2   int num_lines = 0, num_chars = 0;
3   %}
4   %%
5   \n      ++num_lines; ++num_chars;
6   .       ++num_chars;
7   %%
8
9   int main() {
10      yylex();
11      printf("lines = %d, chars = %d\n",
12              num_lines, num_chars);
13      return 0;
14  }
```

Funcionamento do lex/flex

```
1    EXPRESSION_BEGIN "("
2    {EXPRESSION_BEGIN} {
3        doLog (
4            LOG_TYPE_INFO,
5            "inicio de expressao [(] encontrado"
6        );
7        is_open_expression++;
8    }
```

Lista de Tokens reconhecidos I

- **comando para o preprocessador**: “#” (.*)
- **palavras reservadas**:
  “if” | “else” | “const” | “for” | “while” | “struct”
- **tipos de dados**:
  “int” | “float” | “double” | “char”
- **atribuição**:
  (“+” | “-” | “*” | “/” | “%” | “≪” | “≫” | “&” | “|” | “^”)? “=”
- **operador aritmético**:
  “+” “+”? | “-” “-”? | “/” | “*” | “sizeof” |
  “[” {INTEGER_LITERAL} “]”
- **operador relacional**:
  “&&” | “||” | “!” | (“=” | “!”) “=” | (“<” | “>”) “=”?

Lista de Tokens reconhecidos II

- **fim de expressão**: ;
- **início de bloco**: {
- **fim de bloco**: }
- **abre parenteses**: (
- **fecha parenteses**:)
- **inteiro literal**: {DIGIT}+
- **float literal**: {INTEGER_LITERAL}"."{INTEGER_LITERAL}
- **string literal**: \"[^\\\n\"]+
- **char literal**: \'\\?.\'
- **identificador**: ({LETTER})({ALPHA_NUM}|_)*

Mão na massa!!

Agradecimentos

Perguntas?

Obrigado pela atenção