

Analísador léxico para a linguagem C- - Demonstrando o software

Victor E. Almeida Marco A. G. Pedroso

UNIOESTE

31 de Janeiro de 2022

Conteúdo

- 1 Introdução
- 2 lex/Flex
- 3 Analísador léxico
- 4 Conclusão

Softwares utilizados

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

Exemplo mínimo do flex

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

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 preprocessor:** “#” (.*)
- **palavras reservadas:**
“if” | “else” | “const” | “for” | “while” | “struct”
- **tipos de dados:** “int” | “float” | “double” | “char”
- **atribuição:**
(“+” | “-” | “*” | “/” | “%” | “<<” | “>>” | “&” | “|” | “^”) ? “=”
- **operador aritimetico:** “+” “+” ? | “-” “-”
“?” | “/” | “*” | “sizeof” | “[” {INTEGER_LITERAL} “]”
- **operador relacional:**
“&&” | “||” | “!” | (“=” | “!”) “=” | (“<” | “>”) “=” ?
- **fim de expressão:** ;
- **início de bloco:** {

Lista de Tokens reconhecidos II

- **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