

1. Definição da gramática (Forma de Backus-Naur - https://en.wikipedia.org/wiki/Backus-Naur_form)

DIGITO	[0-9]
ID	[A-Za-z_][_A-Za-z0-9]*
HEX	[a-f0-9][a-f0-9][a-f0-9][a-f0-9][a-f0-9][a-f0-9]
TEXT0	[A-Za-z0-9][A-Za-z0-9]*
ARQUIVO	[A-Za-z0-9]*.[A-Za-z0-9]*

2. Especificação FLEX

{DIGITO}+	Um valor inteiro
{DIGITO}+."{DIGITO}*	Um valor real
void char short int long float double signed unsigned	Tipo Primitivo
"#include"	Incluindo algo
"#define"	Definição
"<{ARQUIVO}>"	Biblioteca
"\"{TEXT0} .\"{TEXT0}*"	String
"()" "	Parênteses
"{ }" "	Bloco de função
"[]" "	Índice de ponteiro
"==" "!=" "!==" "<=" ">=" "< ">"	Sinal lógico
"=" "*=" "/=" "%= "+=" "-=" "<=" ">=" "&=" "^=" " ="	Atribuição
"," ";"	Separador
"+" "-" "*" "/"	Um operador
"//\"{TEXT0} .\"{TEXT0}*"	Comentário de uma linha
"/*"	Início de comentário de múltiplas linhas
"*/"	Final de comentário de múltiplas linhas
return printf setlocale	Palavra reservada
for while	Laço de repetição
if else	Estrutura lógica condicional
{ID}	Um identificador
" "	Espaço em branco

\t	Tabulação
[\r] +	Carrier Return
[\n] +	Quebra de linha
.	Caractere não reconhecido

3. Comentários sobre a implementação (**tabela de símbolos**)
 - a. Formato dos tokens: tupla(id, indicador, tipo, posição)
4. Destaque para a compilação / ambiente / linguagem
 - a. MakeFile para auxiliar a compilação/execução do programa e também limpar os arquivos gerados.
 - b. Ambientes:
 - i. Windows 10 20H2, x64, Processador INTEL (i5 10400f)
 - ii. Ubuntu 20.10, x64, Processador AMD (ryzen 5 1600)
 - c. Quando usado LF no editor vscode o programa apresenta um comportamento irregular, somando todas as colunas iterativamente (3000 colunas ao final). Usando CRLF o programa identificou corretamente as linhas e colunas dos tokens no arquivo principal.lex