

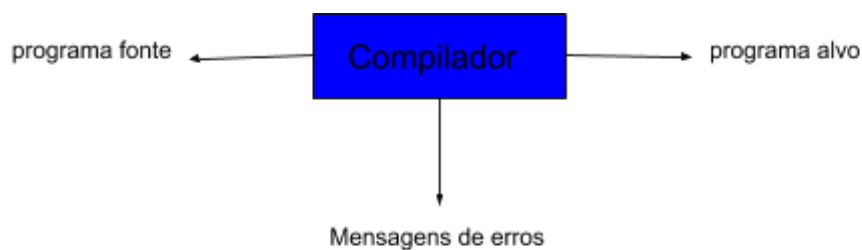
Trabalho Prático : Implementação de Compilador

Alunos : Eduarda Del Vale, Jonathan Ramsés

Disciplina : Compiladores - 7º Período

Professor : Rodrigo Evangelista

Compilador : Um compilador é um programa de computador que a partir de um código fonte escrito em uma linguagem compilada, cria um programa semanticamente equivalente, porém escrito em outra linguagem, código alvo.



1. Definição da Linguagem

1.1 Nome da Linguagem (Compilador)

Linguagem de Programação para Arduino

1.2 Característica da linguagem Léxica

- Utilizada para indexar uma cadeia de entrada, faz o uso de expressões regulares (linguagem regular);
- Cada Token é tratado como uma instância, retorna atributos TOKEN, LINHA, COLUNA e LEXEMA;
- Método `t_error(t)` retorna o lexema não reconhecido;
- Linguagem estruturada;
- A declaração de configurações deverá vir em um bloco iniciado pela palavra reservada: `def config()`
- Todas as constantes globais;

1.3 Linguagem utilizada para Analisador léxico

O compilador foi construído utilizando a linguagem Python com a biblioteca Ply, juntamente com Django para construir a interface gráfica.

1.4 Ferramenta utilizada - IDE

Visual Studio Code

2. Análise Léxica: Tokens, lexemas e Expressão Regular

| Tokens | Lexemas | Expressão Regular |
|------------|-------------------|----------------------------------------------------------------------------|
| VAR | x, nome, area | <code>r'[a-zA-Z_][a-zA-Z_0-9]*'</code> |
| DOUBLE_VAL | double | <code>r'[-+]?[0-9]+(\.([0-9]+)?([eE][-+]?[0-9]+)? [eE][-+]?[0-9]+)'</code> |
| PRINT | print | <code>r'print'</code> |
| STRING | string | <code>r'"([^\"]+)"'</code> |
| DEF | <code>^def</code> | <code>^def</code> |
| BEGIN | { | <code>r'\{'</code> |
| END | } | <code>r'\}'</code> |
| DELIMITER | ; | <code>;</code> |
| PLUS | + | <code>r'\+'</code> |
| MINUS | - | <code>r'\-'</code> |
| TIMES | * | <code>r'*'</code> |
| DIVIDE | / | <code>r'\/'</code> |
| LPAREN | (| <code>r'\('</code> |
| RPAREN |) | <code>r'\)'</code> |
| AT | = | <code>r'\=</code> |
| EQUALS | == | <code>r'\==</code> |
| COMMENT | #comentario | <code>r'\#.*'</code> |
| IGNORE | \r | <code>"\r"</code> |

2.1 Exemplos de códigos da linguagem Arduino :

```
def config(){  
  
    # Declarando pinos do arduino para ligar um led  
    int pinoLedAzul = 1;  
  
    #Configura o pino para saída de energia  
    configPin(pinoLedAzul, OUTPUT);  
  
}  
  
# Função de loop executada no arduino #~  
def init(){  
    # Função que muda o estado do pino para ON = ligado(5v)  
    turn(pinoLedAzul, ON)  
  
    # Aplica um intervalo de tempo até o próximo comando  
    delay(3000)  
  
    # Função que muda o estado do pino para OFF = desligado(0v)  
    turn(pinoLedAzul, OFF)  
  
}
```

Disponível na web : <https://dry-wave-43119.herokuapp.com/>