



Linguajar

Linguagem de programação em português (BR)

Pedro de la Peña - 7º Semestre - Engenharia da Computação
Insper - Lógica da Computação - Professor Raul Ikeda



O que é uma linguagem de programação?

“Uma linguagem de programação é um método padronizado para comunicar instruções para um computador. É um conjunto de regras sintáticas e semânticas usadas para definir um programa de computador. Permite que um programador especifique precisamente sobre quais dados um computador vai atuar, como estes dados serão armazenados ou transmitidos e quais ações devem ser tomadas sob várias circunstâncias.”

Fonte: Wikipédia



Linguagem de programação em português?

- Inclusão de matérias de programação no ensino fundamental e médio de todas as escolas
 - Desenvolvimento de raciocínio lógico e solução de problemas
- Barreira linguística em linguagens já existentes (Python, Java, C, Pascal)
 - Documentação, tutoriais e vários outros fatores **geralmente** estão em inglês
 - Por mais que seja possível encontrar referências excelentes em português, algumas coisas como “tokens” da linguagem sempre serão em inglês

Linguajar - EBNF e Tokens






O Projeto

O desenvolvimento da linguagem foi dividida em 3 etapas:

1. Desenvolvimento da EBNF
2. Uso das ferramentas Bison e Flex
3. Implementação da linguagem em compilador próprio



Linguajar - ENBF (1)

```
program: func_def | decl | ;
func_def: type_spec declaration compound_statement ;
decl: type_spec declaration tk_break;
type_spec: TIPO ;
declaration: tk_id | ( declaration ) ;
compound_statement: INICIO declaration statement_list FIM | INICIO statement_list FIM
statement_list: statement ;
statement: exp_statement | compound_statement | if_statement | while_statement
exp_statement: exp tk_break;
if_statement: SE ( exp ) tk_then FAZER statement FIM | SE ( exp ) tk_then FAZER statement SENAO statement FIM SE
while_statement: ENQUANTO ( exp ) statement FINALIZADO;
exp: assign_exp ;
assign_exp: log_exp | un_exp IGUAL assign_exp ;
log_exp: log_or | log_and ;
log_or: eq_exp | log_or assign_exp OU eq_exp ;
```



Linguajar - ENBF (2)

```
log_and: eq_exp | log_and ETMB eq_exp ;
eq_exp: rel_exp | eq_exp = rel_exp ;
rel_exp: addsub_exp | rel_exp MAIOR addsub_exp | rel_exp MAIORIG addsub_exp | rel_exp MENOR addsub_exp | rel_exp MENORIG
addsub_exp
addsub_exp: multdiv_exp | addsub_exp MAIS multdiv_exp | addsub_exp MENOS multdiv_exp ;
multdiv_exp: un_exp | multdiv_exp MULTIP un_exp | multdiv_exp DIVIDE un_exp ;
un_exp: primary_exp | un_op primary_exp ;
un_op: MAIS | MENOS | NEG ;
primary_exp: tk_id | string | const | ( exp ) ;
const: TIPO ;
```

Linguajar - Bison e Flex





Linguajar - Tokens Principais

A intenção é facilitar a legibilidade para pessoas que não estudaram inglês ou que não se sentem à vontade em relação a sua proficiência com esta linguagem

- “SE” - if
- “FAZER” - then
- “SENAO” - else
- “ETMB” - and
- “OU” - or
- “ENQUANTO” - while
- “FINALIZADO” - while end
- “FIM” - end



Linguajar - Bison e Flex

O Bison foi desenvolvido a partir da EBNF da linguagem, enquanto o Flex contém os tokens da linguagem.

Para compilar o arquivo *parser.y* é preciso instalar o Bison com *apt install bison* (Ubuntu), e em seguida executar o comando *bison parser.y*

Já o arquivo *tokens.l* demanda o Flex, que pode ser instalado com *apt install flex* (Ubuntu) e depois deve-se executar o comando *flex tokens.l*

Linguajar - Compilador





Linguajar - Compilador

Para compilar a linguagem, poderia ser utilizada uma LLVM ou um compilador próprio. Sendo assim, o compilador de VBA simplificado desenvolvido nesta matéria foi adaptado e utilizado.

O compilador é totalmente feito em python3 e recebe um arquivo de entrada do tipo “.vbs”

Para compilar, basta executar o comando

```
python3 lang.py (arquivo input).vbs
```

Um arquivo de testes foi disponibilizado para mostrar o funcionamento do compilador

Arquivo de entrada de exemplo (test.vbs)

```
Function Soma(x como integer, y como integer) como integer
    Soma = x + y
FIM Function

Sub principal()
    definir a como integer
    definir c como integer
    definir b como integer

    c = 100
    b = 99999
    a = 0

    ENQUANTO (c > 1)
        a = a + 1
        c = c-5
        SE (a > c) FAZER
            print b
            print a
            print c
            print nao c
            c = 0
        SENAO
            print Soma(a,c)
        FIM SE
    FINALIZADO
FIM sub
```



Disponibilização dos arquivos

Todos os arquivos citados nesta apresentação podem ser encontrados em

github.com/pedrodelapena/Linguajar

Muito
obrigado :)

