

Compilador LAMP

Analizador léxico e Tabela de simbolos

O compilador consiste em um unico arquivo que compila todos os valores necessarios para sua compilação. O nosso compiladore é baseado funções estaticas que são chamadas pela main. Possui apenas um cabeçalho `hashtable.h` que é a interface que da forma a nossa tabela de simbolos.

Arquitetura do compilador

O compilador contem funções que são chamadas estaticamente.

A primeira função é a função `error` que é responsavél por printar a mensagem de erro e parar a execução da compilação.

A função `readin` tem como função abrir o arquivo e popular o nosse raw com aquela string que está no arquivo. Essa função pode lançar diversas exceções e finalizar a compilação.

A função `ident` tem como obrigação buscar os valores dos tokens literais, ou seja, aquelas strings que não são simbolos logicos, como `&&`, etc. Essa função tem como finalidade mostrar tanto os tokens reservados, como as variáveis - denominadas identificadores.

A função `number` tem como responsabilidade a construção dos valores numéricos inteiros ou flutuantes.

A função `comment` tem como responsabilidade a leitura de um comentário.

A função `literal` tem como responsabilidade a leitura de uma string constante.

A função `parser` é o parser do programa que usa o lexico como função - aqui ele não tem um função bem definida, ele apenas printa os tokens lidos.

A função `words` consiste nas palavras reservadas, que usam a tabela de simbolos para bloquear essas palavras.

Execução

O programa faz uso de Makefile, logo para uma pre compilação usa-se

`make all` para a compilação e para a execução dos testes usa-se `make test`. Como nessa primeira entrega o make test vai printar toda a tabela de simbolos com todos os tokens pre definidos e os tokens escaneados.

Primeira Execução

A primeira execução dos casos de teste mostra que os testes 1, 3 e 7 fizeram uma analise lexica correta, agora devemos corrigir os outros casos de teste para que passem pela analise léxica.

```
lamp compiler test suite
=====
test1.lamp...
Token: program - Type: 1
Token: teste1 - Type: 17
Token: a - Type: 17
Token: is - Type: 18
Token: int - Type: 14
Token: int - Type: 59
Token: b - Type: 17
Token: is - Type: 18
Token: float - Type: 13
Token: float - Type: 59
Token: begin - Type: 19
Token: write - Type: 11
Token: write - Type: 40
Token: a - Type: 17
Token: a - Type: 43
Token: b - Type: 17
Token: b - Type: 41
Token: b - Type: 59
Token: end - Type: 69

---- Hash Table ---
@0: float => 13 | char => 15 | if => 2 | while => 8 | write => 11 |
@1: program => 1 | begin => 19 | is => 18 | int => 14 | until => 7 | do => 9 | read => 10 | teste1 => 17 | a => 17 |
@2: end => 69 | then => 3 | else => 5 | repeat => 6 | b => 17 |
```

test2.lamp...

Token: program - Type: 1

Token: teste1 - Type: 17

Token: a - Type: 17

Token: a - Type: 44

Token: b - Type: 17

Token: is - Type: 18

Token: int - Type: 14

Token: int - Type: 59

Token: result - Type: 17

Token: is - Type: 18

Token: int - Type: 14

Token: int - Type: 59

Token: a - Type: 17

Token: a - Type: 44

Token: x - Type: 17

Token: is - Type: 18

Token: float - Type: 13

Token: float - Type: 59

Token: begin - Type: 19

Token: a - Type: 17

Token: a - Type: 61

Token: 12 - Type: 73

Token: a - Type: 17

Token: a - Type: 59

Token: x - Type: 17

Token: x - Type: 61

lamp exception near line: 7

Exception: FLOAT CONST NOT FORMATTED

test3.lamp...

Token: program - Type: 1

Token: teste2 - Type: 17

Token: a - Type: 17

Token: a - Type: 44

Token: b - Type: 17

Token: b - Type: 44

Token: c - Type: 17

DEFAULTS 142

lamp exception near line: 2

Exception: MALFORMED RAW STRING

test4.lamp...

Token: program - Type: 1

Token: a - Type: 17

Token: a - Type: 44

Token: aux - Type: 17

Token: is - Type: 18

Token: int - Type: 14

Token: int - Type: 59

Token: b - Type: 17

Token: is - Type: 18

Token: float - Type: 13

Token: begin - Type: 19

Token: b - Type: 17

Token: b - Type: 61

Token: 0 - Type: 73

Token: 0 - Type: 59

Token: in - Type: 17

Token: in - Type: 40

Token: a - Type: 17

Token: a - Type: 41

Token: a - Type: 59

Token: in - Type: 17

```
Token: in - Type: 17
Token: in - Type: 40
Token: b - Type: 17
Token: b - Type: 41
Token: b - Type: 59
Token: if - Type: 2
Token: if - Type: 40
Token: a - Type: 17
Token: a - Type: 62
Token: a - Type: 41
Token: then - Type: 3
Token: aux - Type: 17
Token: aux - Type: 61
Token: b - Type: 17
Token: b - Type: 59
Token: b - Type: 17
Token: b - Type: 61
Token: a - Type: 17
Token: a - Type: 59
Token: a - Type: 17
Token: a - Type: 61
Token: aux - Type: 17
Token: end - Type: 69
Token: end - Type: 59
Token: write - Type: 11
Token: write - Type: 40
Token: a - Type: 17
Token: a - Type: 59
Token: write - Type: 11
Token: write - Type: 40
Token: b - Type: 17
Token: b - Type: 41
```

---- Hash Table ----

```
@0: float => 13 | char => 15 | if => 2 | while => 8 | write => 11 |
@1: program => 1 | begin => 19 | is => 18 | int => 14 | until => 7 | do => 9 | read => 10 | a => 17 | aux => 17 |
@2: end => 69 | then => 3 | else => 5 | repeat => 6 | b => 17 | in => 17 |
```

test5.lamp...

```
Token: programa - Type: 17
Token: teste4 - Type: 17
lamp exception near line: 24
Exception: unterminated comment
```

test6.lamp...

```
Token: program - Type: 1
Token: teste5 - Type: 17
Token: a - Type: 17
Token: a - Type: 44
Token: b - Type: 17
Token: b - Type: 44
Token: c - Type: 17
Token: c - Type: 44
Token: maior - Type: 17
Token: is - Type: 18
Token: int - Type: 14
Token: int - Type: 59
Token: outro - Type: 17
Token: is - Type: 18
Token: char - Type: 15
Token: char - Type: 59
Token: begin - Type: 19
Token: repeat - Type: 6
Token: write - Type: 11
```

Token: write - Type: 40
Token: A - Type: 16
Token: A - Type: 41
Token: A - Type: 59
Token: read - Type: 10
Token: read - Type: 40
Token: a - Type: 17
Token: a - Type: 41
Token: a - Type: 59
Token: write - Type: 11
Token: write - Type: 40
Token: B - Type: 16
Token: B - Type: 41
Token: B - Type: 59
Token: read - Type: 10
Token: read - Type: 40
Token: b - Type: 17
Token: b - Type: 41
Token: b - Type: 59
Token: write - Type: 11
Token: write - Type: 40
Token: C - Type: 16
Token: C - Type: 41
Token: C - Type: 59
Token: read - Type: 10
Token: read - Type: 40
Token: c - Type: 17
Token: c - Type: 41
Token: c - Type: 59
Token: if - Type: 2
Token: if - Type: 40
Token: if - Type: 40
Token: a - Type: 17
Token: a - Type: 62
Token: a - Type: 41
Token: a - Type: 97
Token: a - Type: 40
Token: a - Type: 17
Token: a - Type: 62
Token: a - Type: 41
Token: a - Type: 41
Token: end - Type: 69
Token: maior - Type: 17
Token: maior - Type: 61
Token: a - Type: 17
Token: else - Type: 5
Token: if - Type: 2
Token: if - Type: 40
Token: b - Type: 17
Token: b - Type: 62
Token: b - Type: 41
Token: then - Type: 3
Token: maior - Type: 17
Token: maior - Type: 61
Token: b - Type: 17
Token: b - Type: 59
Token: else - Type: 5
Token: maior - Type: 17
Token: maior - Type: 61
Token: c - Type: 17
Token: end - Type: 69
Token: end - Type: 69
Token: end - Type: 59
Token: write - Type: 11
Token: write - Type: 40

```

Token: Maior valor: - Type: 16
DEFAULTS 99
lamp exception near line: 24
Exception: MALFORMED RAW STRING

test7.lamp...
Token: program - Type: 1
Token: teste1 - Type: 17
Token: a - Type: 17
Token: is - Type: 18
Token: int - Type: 14
Token: int - Type: 59
Token: b - Type: 17
Token: is - Type: 18
Token: float - Type: 13
Token: float - Type: 59
Token: begin - Type: 19
Token: write - Type: 11
Token: write - Type: 40
Token: a - Type: 17
Token: a - Type: 43
Token: b - Type: 17
Token: b - Type: 41
Token: b - Type: 59
Token: end - Type: 69

---- Hash Table ---
@0: float => 13 | char => 15 | if => 2 | while => 8 | write => 11 |
@1: program => 1 | begin => 19 | is => 18 | int => 14 | until => 7 | do => 9 | read => 10 | teste1 => 17 | a => 17 |
@2: end => 69 | then => 3 | else => 5 | repeat => 6 | b => 17 |
-----

```

Test 2

```

./main tests/test2.lamp
Token: program - Type: 1
Token: teste1 - Type: 17
Token: a - Type: 17
Token: a - Type: 44
Token: b - Type: 17
Token: is - Type: 18
Token: int - Type: 14
Token: int - Type: 59
Token: result - Type: 17
Token: is - Type: 18
Token: int - Type: 14
Token: int - Type: 59
Token: a - Type: 17
Token: a - Type: 44
Token: x - Type: 17
Token: is - Type: 18
Token: float - Type: 13
Token: float - Type: 59
Token: begin - Type: 19
Token: a - Type: 17
Token: a - Type: 61
Token: 12 - Type: 73
Token: a - Type: 17
Token: a - Type: 59
Token: x - Type: 17
Token: x - Type: 61
lamp exception near line: 7
Exception: FLOAT CONST NOT FORMATTED

```

O floatpoint está com um problema de formatação, ele consta como `x = 12.;` e deve constar como `x = 12.0`.

Teste corrigido:

```
program teste1
a, b is int;
result is int;
a,x is float;
begin
a = 12a;
x = 12.0;
read (a);
read (b);
read (c)
result = (a*b + 1) / (c+2);
write {Resultado: };
write (result);
end.
```

Compilação corrigida

```
./main tests/test2.lamp
Token: program - Type: 1
Token: teste1 - Type: 17
Token: a - Type: 17
Token: a - Type: 44
Token: b - Type: 17
Token: is - Type: 18
Token: int - Type: 14
Token: int - Type: 59
Token: result - Type: 17
Token: is - Type: 18
Token: int - Type: 14
Token: int - Type: 59
Token: a - Type: 17
Token: a - Type: 44
Token: x - Type: 17
Token: is - Type: 18
Token: float - Type: 13
Token: float - Type: 59
Token: begin - Type: 19
Token: a - Type: 17
Token: a - Type: 61
Token: 12 - Type: 73
Token: a - Type: 17
Token: a - Type: 59
Token: x - Type: 17
Token: x - Type: 61
Token: 12.0 - Type: 70
Token: 12.0 - Type: 59
Token: read - Type: 10
Token: read - Type: 40
Token: a - Type: 17
Token: a - Type: 41
Token: a - Type: 59
Token: read - Type: 10
Token: read - Type: 40
Token: b - Type: 17
Token: b - Type: 41
Token: b - Type: 59
Token: read - Type: 10
Token: read - Type: 40
Token: c - Type: 17
Token: c - Type: 41
Token: result - Type: 17
Token: result - Type: 61
```

```

Token: result - Type: 40
Token: a - Type: 17
Token: a - Type: 42
Token: b - Type: 17
Token: b - Type: 43
Token: 1 - Type: 73
Token: 1 - Type: 41
Token: 1 - Type: 40
Token: c - Type: 17
Token: c - Type: 43
Token: 2 - Type: 73
Token: 2 - Type: 41
Token: 2 - Type: 59
Token: write - Type: 11
Token: Resultado: - Type: 16
Token: Resultado: - Type: 59
Token: write - Type: 11
Token: write - Type: 40
Token: result - Type: 17
Token: result - Type: 41
Token: result - Type: 59
Token: end - Type: 69
Token: end - Type: 46

---- Hash Table ---
@0: float => 13 | char => 15 | if => 2 | while => 8 | write => 11 | x => 17 | c => 17 |
@1: program => 1 | begin => 19 | is => 18 | int => 14 | until => 7 | do => 9 | read => 10 | teste1 => 17 | a => 17 |
@2: end => 69 | then => 3 | else => 5 | repeat => 6 | b => 17 | result => 17 |
-----

```

Teste 5

O teste 5 tem um erro de formatação no comentário. fechando o comentario ele vai compilar corretamente.

Correção do código

```

programa teste4
/* Teste4 do meu compilador */
pontuacao, pontuacaoMaxima, disponibilidade is inteiro;
pontuacaoMinima is char;
begin
pontuacaoMinima = 50;
pontuacaoMaxima = 100;
write({Pontuacao do candidato: });
read(pontuacao);
write({Disponibilidade do candidato: });
read(disponibilidade);

while (pontuacao>0 && (pontuacao<=pontuacaoMaxima) do
if ((pontuação > pontuacaoMinima) && (disponibilidade==1)) then
write({Candidato aprovado.})
else
write({Candidato reprovado.})
end
write({Pontuacao do candidato: });
read(pontuacao);
write({Disponibilidade do candidato: });
read(disponibilidade);
end
end

```

Recompilação

```
./main tests/test5.lamp
```

Token: programa - Type: 17
Token: teste4 - Type: 17
Token: pontuacao - Type: 17
Token: pontuacao - Type: 44
Token: pontuacaoMaxima - Type: 17
Token: pontuacaoMaxima - Type: 44
Token: disponibilidade - Type: 17
Token: is - Type: 18
Token: inteiro - Type: 17
Token: inteiro - Type: 59
Token: pontuacaoMinima - Type: 17
Token: is - Type: 18
Token: char - Type: 15
Token: char - Type: 59
Token: begin - Type: 19
Token: pontuacaoMinima - Type: 17
Token: pontuacaoMinima - Type: 61
Token: 50 - Type: 73
Token: 50 - Type: 59
Token: pontuacaoMaxima - Type: 17
Token: pontuacaoMaxima - Type: 61
Token: 100 - Type: 73
Token: 100 - Type: 59
Token: write - Type: 11
Token: write - Type: 40
Token: Pontuacao do candidato: - Type: 16
Token: Pontuacao do candidato: - Type: 41
Token: Pontuacao do candidato: - Type: 59
Token: read - Type: 10
Token: read - Type: 40
Token: pontuacao - Type: 17
Token: pontuacao - Type: 41
Token: pontuacao - Type: 59
Token: write - Type: 11
Token: write - Type: 40
Token: Disponibilidade do candidato: - Type: 16
Token: Disponibilidade do candidato: - Type: 41
Token: Disponibilidade do candidato: - Type: 59
Token: read - Type: 10
Token: read - Type: 40
Token: disponibilidade - Type: 17
Token: disponibilidade - Type: 41
Token: disponibilidade - Type: 59
Token: while - Type: 8
Token: while - Type: 40
Token: pontuacao - Type: 17
Token: pontuacao - Type: 62
Token: pontuacao - Type: 97
Token: pontuacao - Type: 40
Token: pontuacao - Type: 17
Token: pontuacao - Type: 108
Token: pontuacaoMaxima - Type: 17
Token: pontuacaoMaxima - Type: 41
Token: do - Type: 9
Token: if - Type: 2
Token: if - Type: 40
Token: if - Type: 40
Token: pontua - Type: 17
Token: pontua - Type: 17
Token: pontua - Type: 17
Token: pontua - Type: 17
Token: pontua - Type: 17
Token: o - Type: 17
Token: o - Type: 62
Token: pontuacaoMinima - Type: 17
Token: pontuacaoMinima - Type: 41


```
Token: pontuacaoMinima - Type: 97
Token: pontuacaoMinima - Type: 40
Token: disponibilidade - Type: 17
Token: disponibilidade - Type: 22
Token: 1 - Type: 73
Token: 1 - Type: 41
Token: 1 - Type: 41
Token: then - Type: 3
Token: write - Type: 11
Token: write - Type: 40
Token: Candidato aprovado. - Type: 16
Token: Candidato aprovado. - Type: 41
Token: else - Type: 5
Token: write - Type: 11
Token: write - Type: 40
Token: Candidato reprovado. - Type: 16
Token: Candidato reprovado. - Type: 41
Token: end - Type: 69
Token: write - Type: 11
Token: write - Type: 40
Token: Pontuacao do candidato: - Type: 16
Token: Pontuacao do candidato: - Type: 41
Token: Pontuacao do candidato: - Type: 59
Token: read - Type: 10
Token: read - Type: 40
Token: pontuacao - Type: 17
Token: pontuacao - Type: 41
Token: pontuacao - Type: 59
Token: write - Type: 11
Token: write - Type: 40
Token: Disponibilidade do candidato: - Type: 16
Token: Disponibilidade do candidato: - Type: 41
Token: Disponibilidade do candidato: - Type: 59
Token: read - Type: 10
Token: read - Type: 40
Token: disponibilidade - Type: 17
Token: disponibilidade - Type: 41
Token: disponibilidade - Type: 59
Token: end - Type: 69
Token: end - Type: 69

---- Hash Table ---
@0: float => 13 | char => 15 | if => 2 | while => 8 | write => 11 | disponibilidade => 17 | inteiro => 17 | pontuacaoMaxima => 17 |
@1: program => 1 | begin => 19 | is => 18 | int => 14 | until => 7 | do => 9 | read => 10 | teste4 => 17 | pontuacao => 17 | pontuac
@2: end => 69 | then => 3 | else => 5 | repeat => 6 | programa => 17 |
-----
```

Teste 6

O caractere 'n' está mal formatado, a correção para 'n' corrige a compilação.

Correção do código

```

/* Teste do meu compilador */
program teste5
a, b, c, maior is int;
outro is char;
begin
repeat
write({A});
read(a);
write({B});
read(b);
write({C});
read(c);
if ( (a>b) && (a>c) ) end
maior = a

else
if (b>c) then
maior = b;

else
maior = c
end
end;
write({Maior valor:});
write (maior);
write ({Outro? (S/N)});
read(outro);
until (outro == 'N' || outro == 'n')
end

```

Recompilação

```

test6.lamp...
Token: program - Type: 1
Token: teste5 - Type: 17
Token: a - Type: 17
Token: a - Type: 44
Token: b - Type: 17
Token: b - Type: 44
Token: c - Type: 17
Token: c - Type: 44
Token: maior - Type: 17
Token: is - Type: 18
Token: int - Type: 14
Token: int - Type: 59
Token: outro - Type: 17
Token: is - Type: 18
Token: char - Type: 15
Token: char - Type: 59
Token: begin - Type: 19
Token: repeat - Type: 6
Token: write - Type: 11
Token: write - Type: 40
Token: A - Type: 16
Token: A - Type: 41
Token: A - Type: 59
Token: read - Type: 10
Token: read - Type: 40
Token: a - Type: 17
Token: a - Type: 41
Token: a - Type: 59
Token: write - Type: 11
Token: write - Type: 40
Token: B - Type: 16
Token: B - Type: 41

```

Token: B - Type: 59
Token: read - Type: 10
Token: read - Type: 40
Token: b - Type: 17
Token: b - Type: 41
Token: b - Type: 59
Token: write - Type: 11
Token: write - Type: 40
Token: C - Type: 16
Token: C - Type: 41
Token: C - Type: 59
Token: read - Type: 10
Token: read - Type: 40
Token: c - Type: 17
Token: c - Type: 41
Token: c - Type: 59
Token: if - Type: 2
Token: if - Type: 40
Token: if - Type: 40
Token: a - Type: 17
Token: a - Type: 62
Token: a - Type: 41
Token: a - Type: 97
Token: a - Type: 40
Token: a - Type: 17
Token: a - Type: 62
Token: a - Type: 41
Token: a - Type: 41
Token: end - Type: 69
Token: maior - Type: 17
Token: maior - Type: 61
Token: a - Type: 17
Token: else - Type: 5
Token: if - Type: 2
Token: if - Type: 40
Token: b - Type: 17
Token: b - Type: 62
Token: b - Type: 41
Token: then - Type: 3
Token: maior - Type: 17
Token: maior - Type: 61
Token: b - Type: 17
Token: b - Type: 59
Token: else - Type: 5
Token: maior - Type: 17
Token: maior - Type: 61
Token: c - Type: 17
Token: end - Type: 69
Token: end - Type: 69
Token: end - Type: 59
Token: write - Type: 11
Token: write - Type: 40
Token: Maior valor: - Type: 16
Token: Maior valor: - Type: 16
Token: Maior valor: - Type: 41
Token: Maior valor: - Type: 59
Token: write - Type: 11
Token: write - Type: 40
Token: maior - Type: 17
Token: maior - Type: 41
Token: maior - Type: 59
Token: write - Type: 11
Token: write - Type: 40
Token: Outro? (S/N) - Type: 16
Token: Outro? (S/N) - Type: 41

Token: Outro? (S/N) - Type: 59

Token: read - Type: 10

Token: read - Type: 40

Token: outro - Type: 17

Token: outro - Type: 41

Token: outro - Type: 59

Token: until - Type: 7

Token: until - Type: 40

Token: outro - Type: 17

Token: outro - Type: 22

Token: outro - Type: 67

Token: outro - Type: 111

Token: outro - Type: 17

Token: outro - Type: 22

Token: outro - Type: 67

Token: outro - Type: 41

Token: end - Type: 69

---- Hash Table ---

@0: float => 13 | char => 15 | if => 2 | while => 8 | write => 11 | c => 17 |

@1: program => 1 | begin => 19 | is => 18 | int => 14 | until => 7 | do => 9 | read => 10 | a => 17 |

@2: end => 69 | then => 3 | else => 5 | repeat => 6 | teste5 => 17 | b => 17 | maior => 17 | outro => 17 |

Autores João Victor Mazagão e Lara Loures
