
Trabalho 1: analisador léxico para LALG

- Especificação
 - Desenvolver o analisador léxico para a linguagem LALG, com tratamento de erro



Trabalho 1: analisador léxico para LALG

■ Exemplo

<u>Programa-fonte em LALG</u>	<u>Saída</u>
program lalg; {entrada} var a: integer; begin readd(a, @, 1); end.	program - program lalg - id ;-; var - var a - id :-; integer - integer ;-; begin - begin readd - id (- (a - id , - , @ - erro , - , 1 - num) -) :-; end - end . - .

Trabalho 1: analisador léxico para LALG

- Especificação

1. Modelar a tarefa do analisador léxico

- Tipos de tokens, expressões regulares, erros

2. Buscar e **estudar** o Lex/Flex

3. Gerar o analisador léxico usando o Lex/Flex

- Incorporar função principal que analise todo o arquivo de entrada, chamando o analisador léxico várias vezes, o qual, a cada chamada, deve retornar um único par <cadeia,token>

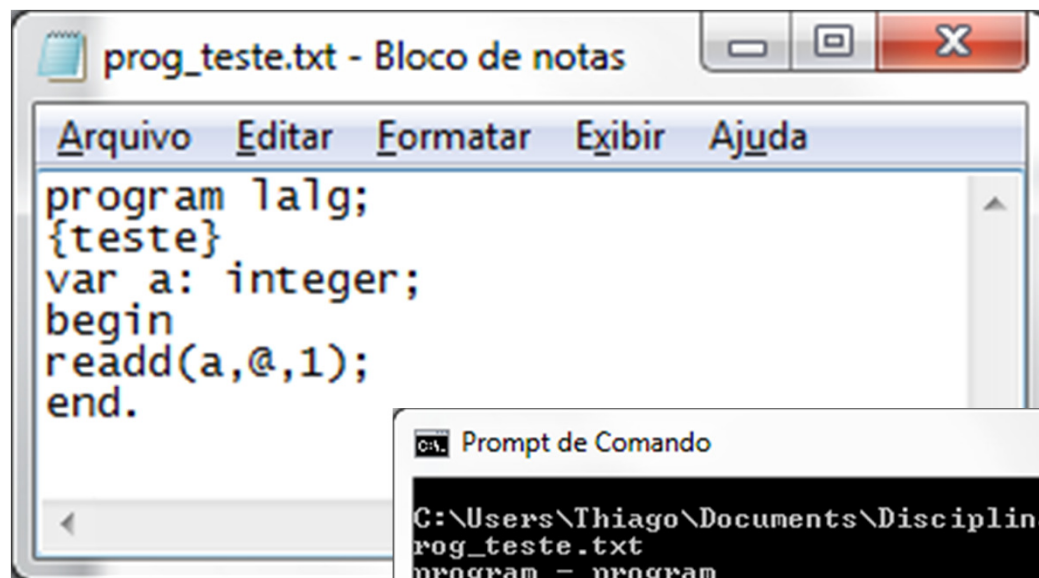
- Essa função será substituída posteriormente pelo analisador sintático
-

Trabalho 1: analisador léxico para LALG

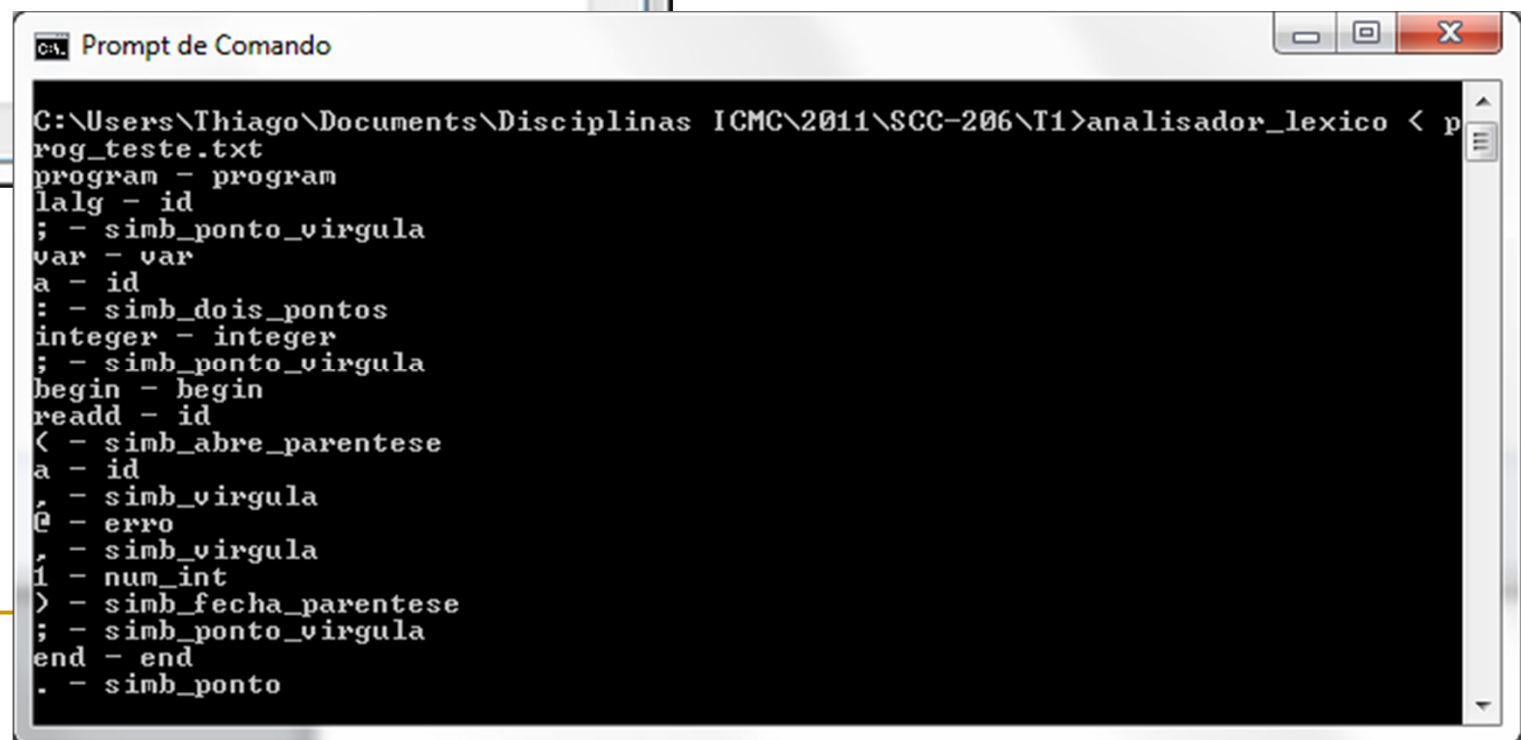
- Decisões de projeto e de implementação
 - Como serão os comentários
 - Possibilidade de várias linhas?
 - Quais os tokens para as cadeias
 - <palavra_reservada, palavra_reservada> ou <palavra_reservada, simb_palavra_reservada>?
 - Não usem códigos numéricos para os tokens!
 - Como implementar a tabela de palavras reservadas
 - Estrutura de dados, busca
 - Lembrem-se: busca deve ser eficiente
 - Como lidar com os erros
-

Trabalho 1: analisador léxico para LALG

- Exemplo



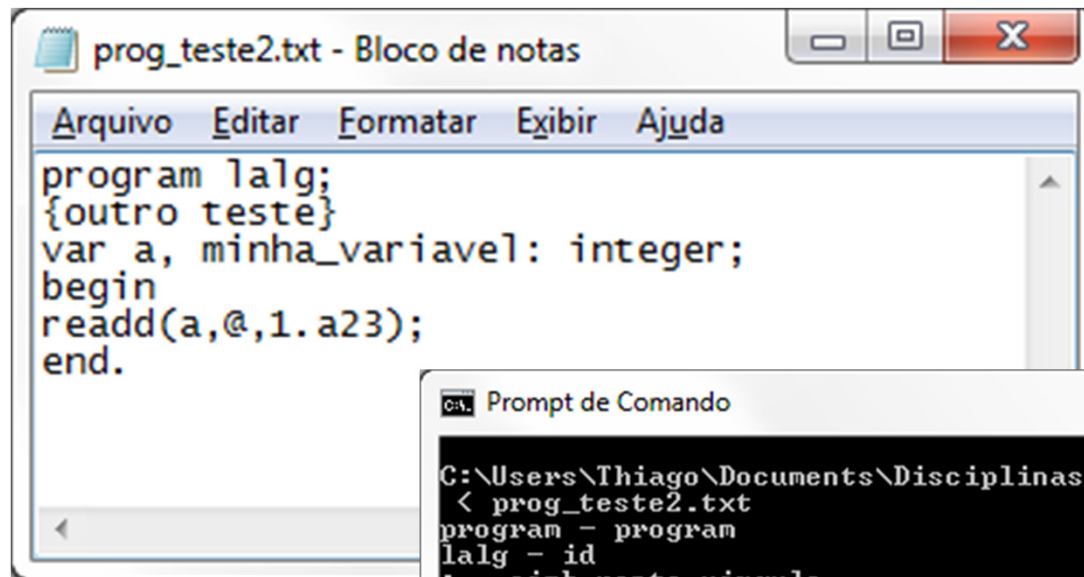
```
program lalg;  
{teste}  
var a: integer;  
begin  
  readd(a,@,1);  
end.
```



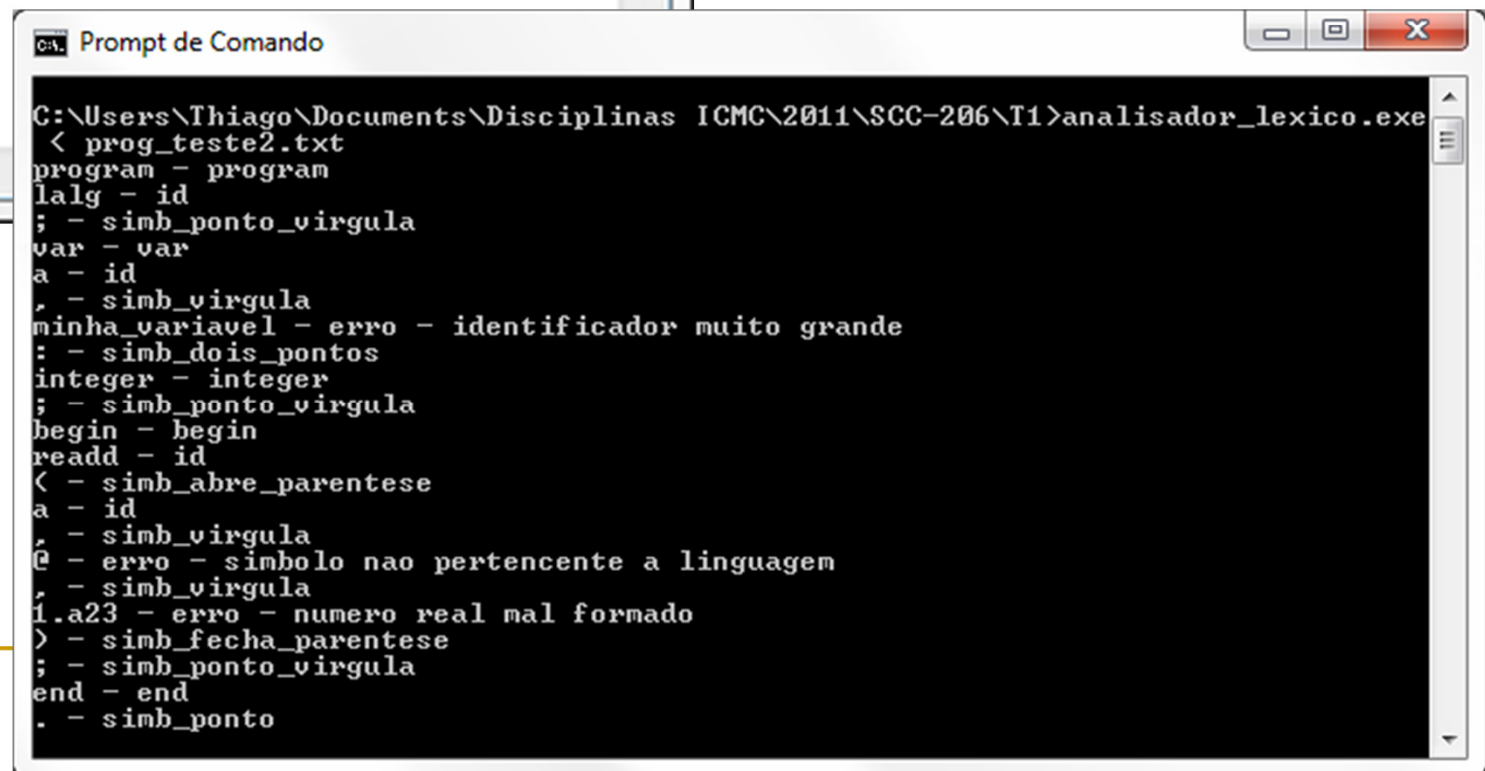
```
C:\Users\Thiago\Documents\Disciplinas ICMC\2011\SCC-206\T1>analisador_lexico < prog_teste.txt  
program - program  
lalg - id  
; - simb_ponto_virgula  
var - var  
a - id  
: - simb_dois_pontos  
integer - integer  
; - simb_ponto_virgula  
begin - begin  
readd - id  
a - id  
, - simb_virgula  
@ - erro  
, - simb_virgula  
i - num_int  
) - simb_fecha_parentese  
; - simb_ponto_virgula  
end - end  
. - simb_ponto
```

Trabalho 1: analisador léxico para LALG

- Outro exemplo



```
Arquivo  E_ditar  F_ormatar  E_xibir  A_juda
program lalg;
{outro teste}
var a, minha_variavel: integer;
begin
readd(a,@,1.a23);
end.
```



```
C:\Users\Thiago\Documents\Disciplinas ICMC\2011\SCC-206\T1>analisador_lexico.exe
< prog_teste2.txt
program - program
lalg - id
; - simb_ponto_virgula
var - var
a - id
, - simb_virgula
minha_variavel - erro - identificador muito grande
: - simb_dois_pontos
integer - integer
; - simb_ponto_virgula
begin - begin
readd - id
< - simb_abre_parentese
a - id
, - simb_virgula
@ - erro - simbolo nao pertencente a linguagem
, - simb_virgula
1.a23 - erro - numero real mal formado
) - simb_fecha_parentese
; - simb_ponto_virgula
end - end
. - simb_ponto
```

Trabalho 1: analisador léxico para LALG

- Grupos de 3 alunos, no máximo
 - Entrega
 - E-mail para professor e estagiária PAE com:
 - Especificação do analisador léxico na linguagem do lex/flex
 - Código-fonte produzido pelo lex
 - Executável (para evitar problemas com e-mail, alterem o nome: de **arq.exe** para **arq.ex**, por exemplo)
 - Relatório *sucinto e objetivo* (máximo de 5 páginas) contendo
 - Decisões de projeto, justificativas
 - Descrição da especificação do analisador léxico na linguagem do lex/flex
 - Passo a passo para compilar o analisador léxico e executá-lo
 - Exemplo de execução
 - Data de entrega: **04/04** até meia noite
 - A cada dia de atraso, 1 ponto a menos
 - Se cópia de trabalhos detectada: zero para todos os grupos envolvidos!
-