

UDESC – Universidade do Estado de Santa Catarina  
DCC – Departamento de Ciências da Computação  
Disciplina de Compiladores  
Prof.: Fabiano Baldo, Dr. Eng.

## Primeiro Trabalho de Implementação Analisador Léxico (scanner)

**Descrição:** Considere a linguagem detalhada abaixo:

### Características da linguagem

1. Suporta os tipos: integer, real, char, string e boolean
2. Comandos:
  - a. Atribuição com o operador “:=”, “+=”, “-=”
  - b. Entrada com o comando **read( )**
  - c. Saída com os comandos **write( )** e **writeln( )**
  - d. Condicional com o comando **if - then - else**
  - e. Repetições com os comandos: **while - do**, **repeat - until**, **for - to - do**
3. Constantes caracteres delimitados por (‘ ’) e constantes strings por (“ ”)
4. Operadores relacionais: “=”, “>”, “<=”, “>”, “<”, “<>”
5. Operadores lógicos **and**, **or**, **not**
6. Operadores aritméticos: “+”, “-”, “\*”, “/”
7. Símbolos especiais: “,”, “:”, “;”, “.”
8. Bloco de comandos delimitados por **begin** e **end**
9. Comentário de bloco com os delimitadores “{” e “}”
10. Lista de Palavras Reservadas: **var**, **const**, **while**, **do**, **for**, **read**, **write**, **writeln**, **if**, **then**, **else**, **true**, **false**, **integer**, **real**, **char**, **string**, **boolean**, **and**, **or**, **not**, **to**, **repeat**, **until**, **program**

Abaixo, é apresentado um exemplo de um programa escrito nessa linguagem:

{Exemplo completo de um programa}

Program Piloto;

{ declarações de variáveis e constantes globais }

var: integer cont; real Nota1, Nota2, Media\_das\_medias, med;

const: integer total = 10;

{Início do Programa}

```

begin
  media_das_medias := 0;
  for cont=0 to total do
  begin
    writeln("***** ENTRADA DE DADOS *****");
    writeln("Digite os valores da primeira nota:");
    read(Nota1);
    writeln("Digite os valores da segunda nota:");
    read(Nota2);
    med := (a+b)/2.0;
    media_das_medias := media_das_medias + med;
    write("Media = ",med);
  end;
  write("Media Geral = ",Media_das_medias/total);
end.

```

**Objetivo:** Implemente um módulo (sub-rotina) para a análise léxica para um protótipo de compilador para a linguagem apresentada acima.

Características:

Do módulo scanner:

- A sub-rotina retorna um token (classe e valor) cada vez que for chamada;
- Considera que o programa fonte para análise já está aberto (deve-se abrir os arquivos no programa principal);
- Não retorna nada quando atingir o fim de arquivo;
- Implementa um AFD para o reconhecimento de tokens.

Do programa principal:

- Abre um arquivo fonte para a análise;
- Chama (sucessivas vezes) a rotina scanner e exibe o valor do token na tela;
- Fecha o arquivo fonte ao final da compilação;
- Pára o processo de compilação caso um erro seja encontrado;
- Exibe erros de compilação (se ocorrerem) ou mensagem de sucesso.

Critérios de Avaliação:

- Desenvolvimento em dupla;
- Implementação usando linguagem C ou C++ (pode ser utilizado o LEX para a geração do programa);
- Entrega de fontes e executável (em um arquivo zipado) via e-mail: [fabianobaldo@gmail.com](mailto:fabianobaldo@gmail.com)
- Data de Entrega: **16/10/2008**;
- Será descontado ½ ponto por dia de atraso da entrega do trabalho;
- Será descontado ½ ponto para cada análise incorreta;

- Será descontado 50% da nota para programas que travarem os estiverem incompletos;
- Será descontado 100% da nota para trabalhos copiados, de ambos os grupos.

Observação:

Para facilitar a implementação do analisador léxico, pode ser utilizado como base o analisador disponível no site na disciplina, criado por K. C. Loudon e descrito em seu livro, intitulado: "Compiladores – Princípios e Práticas". Parte deste livro está no Xerox para ser utilizado como guia para a implementação.