



UTFPR - CÂMPUS PONTA GROSSA - COCIC

Disciplina de Compiladores – 2014.1

Exemplos de programas em FLEX

Professor: Gleifer Vaz Alves

Neste arquivo estão listados alguns exemplos de programas, no caso pequenos analisadores léxicos criados na linguagem FLEX.

Para compilar os arquivos FLEX (considerando: ambiente *linux* e compilador *gcc*) os seguintes passos são executados:

- 1º opção:

```
– $ flex teste.l
– $ gcc lex.yy.c -lfl
– $ ./a.out
```

- 2º opção:

```
– $ flex -o teste.c teste.l
– $ gcc teste.c -lfl -o teste
– $ ./teste
```

- Nessa 2ª opção, utiliza-se a opção **-o** para especificar o nome do arquivo de saída. Caso contrário, o **flex** gera automaticamente o nome *lex.yy.c*

- *Obs.:* a opção **-lfl** é utilizada pois o **flex** utiliza uma pequena biblioteca para algumas funções, no caso a biblioteca **-lfl**.

1. O programa FLEX ilustrado no Código 1 serve para reconhecimento simples de dígitos e palavras.

Note a separação das partes de um programa FLEX entre as definições de expressões regulares e regras de tradução.

Além disso, utiliza-se a função `yylex()`, que é responsável por fazer o escaneamento dos tokens da entrada e ainda o apontador `yytext`, o qual funciona como o *lexemeBegin*.

```
1 %{
3  /* Exemplo 0 :: Flex
   reconhecimento simples de digitos e palavras */
5
6  %{
7
8  DIGITO    [0-9]
9  PALAVRA   [a-zA-Z]+
10
11 %%
12
13 {DIGITO} { printf("Digito: \n%s\n", yytext); }
14 {PALAVRA} { printf("Palavra: \n%s\n", yytext); }
15 . { printf("Nao reconhecido: \n%s\n", yytext); }
16
17 %%
```

```

19 int main ( int nArgs, char* szArgs[] ) {
21     yylex();
    return 0;
}

```

Listing 1: Exemplo FLEX 0

2. O programa FLEX ilustrado no Código 2 serve para reconhecimento simples de dígitos e palavras. Além de espaços.

```

%{
2
    /* Exemplo 1 :: Flex
4     reconhecimento simples de digitos e palavras
       com reconhecimento de espaços também */
6
    int numPalavras = 0;
8
%}
10
DIGITO    [0-9]
12 PALAVRA [a-zA-Z]+
ESPACO    " "
14 %%

16 {DIGITO} { printf("Digito:\n%s\n", yytext); }
{PALAVRA} { printf("Palavra:\n%s\n", yytext); numPalavras++; }
18 {ESPACO} { printf("Espaco\n"); }
.         { printf("Nao-reconhecido:\n%s\n", yytext); }
20 %%

22 int main ( int nArgs, char* szArgs[] ) {
    yylex();
24     printf("Numero_palavras: %d\n", numPalavras);
    return 0;
26 }

```

Listing 2: Exemplo FLEX 1

3. O programa FLEX ilustrado no Código 3 serve para reconhecimento e contagem de novas linhas, caracteres e tabulações.

```

%{
2
    /* Exemplo 2 :: Flex
4     reconhecimento e contagem de novas linhas, caracteres e
       tabulacoes */
6
    int linhas = 0, caracteres = 0, tabs = 0;
8
%}
10
NOVALINHA [\n]+
12 TAB     [\t]+
14 %%

16 {NOVALINHA} { ++linhas; ++caracteres; }
18 {TAB}      { ++tabs; }
.           { ++caracteres; }
20
%%
22
int main ( int nArgs, char* szArgs[] ) {
2

```

```

24     yylex();
    printf("Numero_linhas: %d\n", linhas);
26     printf("Numero_tabulacoes: %d\n", tabs);
    printf("Numero_caracteres: %d\n", caracteres);
28     return 0;
}

```

Listing 3: Exemplo FLEX 2

4. O programa FLEX ilustrado no Código 4 serve para escanear tags, por exemplo de um código em HTML.

```

1  %{
3  /* Exemplo 3 :: Flex
   realiza o escaneamento de tags (e.g., HTML) */
5
   int linhas = 0;
7
   %{
9
   TAG    \<(.)*\>
11  WHITESPACE  [\ \t]
   LINHA    [\n]
13
15 %%
17 {TAG}      ;
   {LINHA}    { linhas++; }
19 {WHITESPACE} { printf("%s", yytext); }
   .         { printf("%s", yytext); }
21
23 %%
25 int main ( int nArgs, char* szArgs[] ) {
   yylex();
   printf("\nTotal_de_linhas_processadas: %d\n", linhas);
27     return 0;
}

```

Listing 4: Exemplo FLEX 3

5. O programa FLEX ilustrado no Código 5 serve, não apenas para escanear tags (HTML), mas também reconhecê-las.

```

1  %{
2
   /* Exemplo 4 :: Flex
   realiza o escaneamento e o reconhecimento
   de tags (e.g., HTML) */
6
   int linhas = 0, qtdTags = 0;
8
   %{
10
   TAG    \<(.)*\>
12  WHITESPACE  [\ \t]
   LINHA    [\n]
14
16 %%
18 {TAG}    { qtdTags++; printf("Tag: %s\n", yytext); }
   {LINHA} { linhas++; }

```

```

20 {WHITESPACE}    ;
21     .           ;
22
23
24 %%
25
26 int main ( int nArgs, char* szArgs[] ) {
27     yylex();
28     printf("\nTotal_de_linhas_processadas: %d\n", linhas);
29     printf("\nQuantidade_de_tags_reconhecidas: %d\n", qtdTags);
30     return 0;
31 }

```

Listing 5: Exemplo FLEX 4