

2.6 利用 Lex 自动生成扫描程序

- Lex(Flex)：主要用于 Unix 系统中的扫描器自动生成程序
- 要点：为每个记号列出正则表达式及相应的处理
- 注意：表示正则表达式时有一些特殊的约定 (2.6.1)

2.6 利用 Lex 自动生成扫描程序

- Lex(Flex): 主要用于 Unix 系统中的扫描器自动生成程序
- 要点: 为每个记号列出正则表达式及相应的处理
- 注意: 表示正则表达式时有一些特殊的约定 (2.6.1)

Lex 输入文件格式	输入例
{definitions}	digit [0-9]
	number {digit}+
%%	%%
{rules}	{ number } { return number }
%%	%%
{auxiliary routines}	void main(void)
	{ yylex(); } //生成扫描器 yylex

输出: 扫描程序 yylex

构建给文本添加行号的扫描程序

```
1  %{ /* a Lex program that adds line numbers
2      to lines of text, printing the new text
3      to the standard output */
4  #include <stdio.h>
5  int lineno = 1;
6  // 引入必要的头文件和全局变量等,
7  // 将直接插入到生成的C代码中
8  %}
9  line .*\\n
10 %%
11 line { printf("%5d\\n", lineno++, yytext); }
12 %%
13 main()
14 { yylex(); return 0; }
```

2.6 利用 Lex 自动生成扫描程序

- Lex(Flex): 主要用于 Unix 系统中的扫描器自动生成程序
- 要点: 为每个记号列出正则表达式及相应的处理
- 注意: 表示正则表达式时有一些特殊的约定 (2.6.1)

Lex 输入文件格式	输入例
{definitions}	digit [0-9]
	number {digit}+
%%	%%
{rules}	{ number } { return number }
%%	%%
{auxiliary routines}	void main(void)
	{ yylex(); } //生成扫描器 yylex

输出: 扫描程序 yylex

构建给文本添加行号的扫描程序

```
1  %{ /* a Lex program that adds line numbers
2      to lines of text, printing the new text
3      to the standard output */
4  #include <stdio.h>
5  int lineno = 1;
6  // 引入必要的头文件和全局变量等,
7  // 将直接插入到生成的C代码中
8  %}
9  line .*\\n
10 %%
11 line { printf("%5d\\n", lineno++, yytext); }
12 %%
13 main()
14 { yylex(); return 0; }
```

将文本中的数转换成 16 进制表示

```
1  %{ /* a Lex program that change all numbers
2     from decimal to hexadecimal notation ,
3     printing a summary statistic to stderr */
4  #include <stdlib.h>
5  #include <stdio.h>
6  int count = 0;
7  %{
8  digit [0-9]
9  number {digit}+
10 %%
11 number { int n = atoi(yytext); // yytext: 匹配的字符串
12         printf("%x", n);
13         if (n > 9) count++; }
14 %%
15 main()
16 {    yylex();
17     fprintf(stderr, "number_of_replacements=%d",
18             count);
19     return 0;
20 }
```

回显以 a 开头或结尾的文本行

```
1  %{ /* Selects only lines that end or
2     begin with the letter 'a'.
3     Deletes everything else. */
4  #include <stdio.h>
5  %{
6  ends_with_a .*a\n
7  begins_with_a a.*\n
8  %%
9  ends_with_a ECHO;
10 begins_with_a ECHO;
11 .* \n ;
12 %%
13 main()
14 {    yylex();
15     return 0;
16 }
```

将注释外的大写字母转换成小写字母

```
1  %{ #include <stdio.h> %}  
2  %%  
3  [A-Z] { putchar(tolower(yytext[0]));  
4  "/*" {  char c;  
5          int done = 0;  
6          ECHO;  
7          do  
8              { while ((c = input()) != '*') putchar(c);  
9                  putchar(c);  
10                 while ((c = input()) == '*') putchar(c);  
11                 if (c == '/') done = 1;  
12             } while (!done);  
13         }  
14  %%  
15  main()  
16  {    yylex();  }
```


Tiny 词法分析 Lex 代码

- 研究 Tiny 词法分析程序的 Lex 代码

编 译 方 法

第二章 词法分析

第三部分

冯速

fengsu@bnu.edu.cn

北京师范大学

信息科学与技术学院