

## 一、Flex 的常用命令流程

假设你的词法分析文件名为 `lexer.l`：

```
flex lexer.l  
gcc lex.yy.c -o lexer -lfl  
./lexer < input.txt
```

下面我们详细拆解说明每一步。

## 二、常用命令及其作用

### 1 flex source.l

- 功能：

将 Flex 源文件 (`.l`) 编译生成一个 C 语言文件 `lex.yy.c`。

- 结果：

生成的 `lex.yy.c` 文件中包含自动生成的 `yylex()` 函数等实现代码。

💡 如果不指定输出文件名，默认生成 `lex.yy.c`。

若希望指定输出文件名：

```
flex -o mylexer.c lexer.l
```

### 2 gcc lex.yy.c -o lexer -lfl

- 功能：

使用 GCC 将 Flex 生成的 C 源文件编译为可执行文件。

- 参数解释：

- `lex.yy.c`：输入的 C 源文件。
- `-o lexer`：输出可执行文件名为 `lexer`。
- `-lfl`：链接 Flex 库 (`libfl`)，其中包含 `yywrap()`、`yyrestart()` 等函数定义。

⚠ 若省略 `-lfl`，可能会报错：

```
undefined reference to `yywrap`
```

因为编译器找不到 Flex 库函数。

### 3 ./lexer < input.txt

- 功能：

将 `input.txt` 文件的内容重定向为 `lexer` 的标准输入，从而让 `yylex()` 从文件中读取并分析内容。

- 相当于：

手动键入文件内容再按 `Ctrl+D` 结束输入。

---

```
4 echo "x = 10 + 2" | ./lexer
```

- 功能:

使用管道符 (|) 将 echo 输出直接送入 lexer 的标准输入。  
不需要创建额外的文件。

---



### 三、常见的 Flex 辅助命令选项

命令	作用
<code>flex -v lexer.l</code>	显示 Flex 的状态机统计信息 (状态数、规则数等)
<code>flex -t lexer.l</code>	将生成的代码输出到终端 (而不是生成文件)
<code>flex -o mylexer.c lexer.l</code>	指定输出文件名
<code>flex --help</code>	查看所有命令行选项
<code>flex --version</code>	查看版本号

---



### 四、调试与查看生成文件

```
cat lex.yy.c      # 查看生成的C源代码  
less lex.yy.c    # 分页查看  
grep yylex lex.yy.c # 查找yylex函数位置
```

---



### 五、扩展用法

如果你希望直接对某个 C 源文件 (如 example.c) 进行词法分析:

```
./lexer < example.c
```

就能让你的 Flex 程序自动读取并分析 example.c 的内容。