

# 『编译技术』 SysY-Mips编译器设计——词法分析

## 章节目录

- [『编译技术』 SysY-Mips编译器设计——总体设计概述](#)
- [『编译技术』 SysY-Mips编译器设计——词法分析](#)
- [『编译技术』 SysY-Mips编译器设计——语法分析](#)
- [『编译技术』 SysY-Mips编译器设计——语义分析\(符号表管理与错误处理\)](#)
- [『编译技术』 SysY-Mips编译器设计——中间代码LLVM生成](#)
- [『编译技术』 SysY-Mips编译器设计——目标代码Mips生成](#)
- [『编译技术』 SysY-Mips编译器设计——中端代码优化](#)
- [『编译技术』 SysY-Mips编译器设计——后端代码优化](#)
- [『编译技术』 SysY-Mips编译器设计——实验总结](#)

## 零. 任务目标

设计并实现词法分析程序，从源程序中识别出单词，记录其单词类别和单词值。

## 一. SysY文法的单词（Token）

单词名称	类别码	单词名称	类别码	单词名称	类别码	单词名称	类别码
Ident	IDENFR	!	NOT	*	MULT	=	ASSIGN
IntConst	INTCON	&&	AND	/	DIV	;	SEMICN
FormatString	STRCON		OR	%	MOD	,	COMMA
main	MAINTK	for	FORTK	<	LSS	(	LPARENT
const	CONSTTK	getint	GETINTTK	<=	LEQ	)	RPARENT
int	INTTK	printf	PRINTF TK	>	GRE	[	LBRACK
break	BREAKTK	return	RETURNTK	>=	GEQ	]	RBRACK
continue	CONTINUETK	+	PLUS	==	EQL	{	LBRACE
if	IFTK	-	MINU	!=	NEQ	}	RBRACE
else	ELSETK	void	VOIDTK				

## 二. 词法解析器Lexer设计

词法解析器通过扫描输入的源程序字符串，将其分割成一个个单词，同时记录这些单词的类别信息，主要用途在于耦合进后续的语法解析器，使得语法解析器在解析文法时词法解析器可以源源不断为其提供语法解析的最小单元，**单词**。

因此我们的词法解析主要功能有二：

1. 不影响其他的基础上消除掉单行注释与多行注释
2. 依次将每个单词及其相应信息打包取出，直至输入文件结束

我们定义Lexer类，内部属性有：

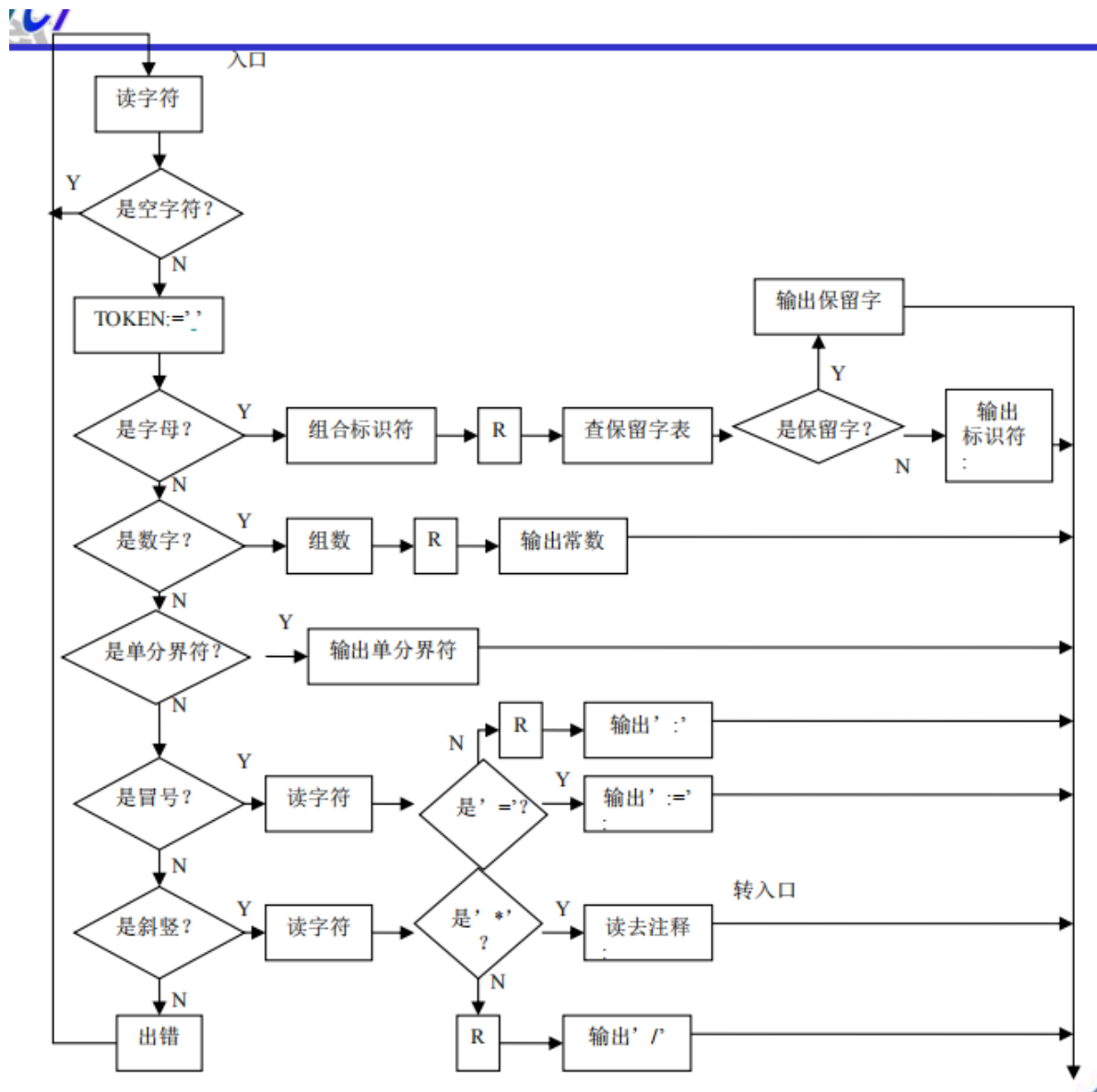
1. `input`：输入字符串

2. `pos`：当前解析到的位置
3. `curWordInfo`：记录当前读到的单词种类，单词值，行数等信息

方法有：

1. `next`：读取下一个单词
2. `peek`：获取当前单词信息

本质上我们读取单词的过程等价于一台有限状态自动机，词法解析(next函数)的基本逻辑如下图所示：



### 三. 词法分析的预读策略

我们发现单词中有一些单词存在前缀重叠的情况，如 '`<`' 和 '`<=`'，对于这些前缀重叠的单词，从词法层面我们既可以认为这是一个单词，也同样可以认为是两个单词，因此我们单单读入一个符号是无法判断的，需要预读下一个字符来判断是哪一种单词。

### 四. 词法分析Bug

- 注释消除有误，`strCon` 内存在 "`//...`" 时也会被当作注释消除，应在消除注释时特判此种情况。