

编译原理--词法分析

编译原理--词法分析

一、C语言子集

1.1 关键字 (32个)

数据类型 (12个)

控制语句 (12个)

存储类型 (4个)

其它关键字 (4个)

1.2 运算符 (29个)

算术运算符 (6个)

关系运算符 (6个)

逻辑运算符 (3个)

位运算符 (6个)

赋值运算符 (8个)

1.3 界符 (12个)

1.4 标识符

1.5 常量

二、注意点 (转换图主要关注这)

三、难点

四、分工

一、C语言子集

1.1 关键字 (32个)

数据类型 (12个)

- `int short long`
- `float double`
- `char`
- `signed unsigned`
- `struct union enum`
- `void`

控制语句 (12个)

- `for do while break continue`
- `if else goto`
- `switch case default`
- `return`

存储类型 (4个)

- `auto extern static register`(寄存器变量声明)

其它关键字 (4个)

- `const sizeof typedef volatile`

1.2 运算符 (29个)

算术运算符 (6个)

- `+ - * / %`
- `++ --`

关系运算符 (6个)

- `> < != == >= <=`

逻辑运算符 (3个)

- `&& || !`

位运算符 (6个)

- `<< >>`
- `& | ~ ^`

赋值运算符 (8个)

- `=`
- `+= -= *= /= %= <<= >>=`

1.3 界符 (12个)

- `{ } () []`
- `/* */ //`
- `" ' #`
- `, ;`

1.4 标识符

- 正则表达式: `letter(_|letter|digit)*`
- `letter=[a-zA-Z]+`
- `digit=[0-9]+`

1.5 常量

- 整数: `digit*`
- 实数: `digit+.digit+`
- 布尔: `true false`
- 字符串: `"*"`

二、注意点 (转换图主要关注这)

- `+ += ++; - -= --`
- `* *= */`
- `/ // /* /=`
- `< <= << <<=; > >= >> >>=`
- `= ==`
- `! !=`
- `& && &=; | || |=`

三、难点

- 中文字符识别与输出
- * 作为指针定义、乘号、指针解引用的区分

四、分工

算法设计：常为、吴扬俊

代码实现：武起龙、张嶢（结对编程）

测试设计：荆顺吉

ppt展示：荆顺吉