**概述**

符号表的管理可以说是贯穿整个编译过程的。

对于符号表的构造，由于作用域的存在，因此使用了一个树状的符号表，每个作用域则用一个哈希表来进行管理。

**哈希表**

（继续说个丢脸的事情，在哈希表上出的问题太多了，以至于我迄今为止仍然无法确定，这个哈希表用起来是不是不会出错）

**树状符号表**

之所以这样使用一个树状的符号表，是因为，这个符号表不仅仅表示了符号，还包括了作用域的问题，换而言之，所有跟作用域相关的部分，都推荐建立在符号表中，而不是AST树。

例如，对于typedef name的使用，是和符号表的作用域息息相关的，如果再下一层的符号表中定义了同名但不是typedef name的符号，那么在下一层当中会被合法的当做下一层的定义，而不是一个错误。