**概述**

对于内联汇编的特性的引入，主要是源于我一开始的愚蠢想法，由于gcc的头文件被gcc的一些特有的但不在C11中的特性所污染，没法编译，因此试图完全使用linux的syscall来绕开gcc，最后做了个内联汇编出来。但是随后我意识到了头文件的定义和实现的分离，以及可以在之后使用gas和gcc的ld这个解决办法。

即便如此愚蠢，我仍然保留了和gcc类似的内联汇编的内容，因为C本就是面向底层的程序，没有内联汇编的C编译器是没有灵魂的。

如果将来有需要，也许努努力能够将gcc的runtime库所抛开吧。但这属于远期目标，目前是没有这样的打算的。

对于内联汇编的部分，主要是C的文法和汇编的文法两个部分，后者主要是引号里面的内容。

**GCC内联汇编语法（采用的部分）**

* 词法

主要涉及到关键字为\_\_asm\_\_和\_\_volatile\_\_，只需要在词法表中加入这两个词即可。

* 文法

在这里，GCC内联汇编主要包括两种，基本的汇编和扩展的汇编，但是我想，应该可以用一个语法来囊括，相关内容网上海了去了，我就不多说了

仿照C语言的文法，我做了以下设计：

|  |
| --- |
| KW\_asm  \_\_asm\_\_  KW\_volatile  \_\_volatile\_\_  Asm statement  KW\_asm KW\_volatile(opt) ( string Asm\_extend(opt) ) ;  Asm\_extend  : Asm\_output(opt) Asm\_extend\_postfix(opt)  Asm\_extend\_postfix  : Asm\_input(opt) Asm\_register\_postfix(opt)  Asm\_register\_postfix  : Asm\_register(opt)  Asm\_output  Asm\_restrict  Asm\_output , Asm\_restrict  Asm\_input  Asm\_restrict  Asm\_input , Asm\_restrict  Asm\_register  String  Asm\_register , string  Asm\_restrict  String ( identifier ) |

做几个说明，主体的string只有一个，也许在很多地方看起来可以有很多个，但是由于预编译的步骤里面，将没有用逗号分隔的string合并了，所以，他就是一个string

另外之所以写成这样，而不是三个冒号一排，是因为考虑到冒号后面内容缺省的情况，如果后面的东西缺省，那么可以取消冒号，但是后面的不缺而前面的缺省，则不能取消前面的冒号。但是由于有了缩减树的函数在，所以实际当中能够看到的语法树的节点情况，其实三个冒号后面的东西都是并列的。

相关的测试在par test 05.c当中测试。