OpenCASCADE Expression Interpreter by Flex & Bison

eryar@163.com

Abstract. OpenCASCADE provide data structure of any expression, relation or function used in mathematics. Flex and Bison are tools for building programs that handle structured input. They were originally tools for building compilers, but they have proven to be useful in many other areas. The Expression Interpreter in OpenCASCADE is made by Flex and Bison. So let's know something about Flex and Bison.

Key Words. OpenCASCADE, Expression Interpreter, Flex, Bison

1.Introduction

OpenCASCADE 的 TKMath 中提供了数学表达式求解的包 Expr 和 ExprIntrp，用来对数学表达式或函数进行解析计算。所谓的数学表达式如：sin(10)+20*6-6/3 等。以前在《数据结构》的书上看到用栈的方式来对简单表达式求值，感觉已经很不错了。但是如果表达式中包含三角函数、对数、指数等，程序就要复杂了。如何简单、优雅地去解决这个问题，当时也是很茫然。

工作中遇到前辈自己写了一个用于定义几何的语言，还可以计算表达式。虽然没有看到源码，对其已是崇拜不已。后来接触了脚本语言，像 Tcl/Tk，Python 之类，发现原来利用这些开源库，也是可以实现一个简单、稳定的自定义开发语言。因为这类脚本不仅可以计算表达式，还可以计算带参数的表达式，如：

```
set x 3
set y 4
set z [expr sqrt($x*$x+$y*$y)]
```

只要你在 Unix 环境中写过程序，你必定会邂逅神秘的 Lex&YACC，就如 GUN/Linux 用户所熟知的 Flex&Bison，这里的 Flex 就是由 Vern Paxon 实现的一个 Lex，Bison 则是 GNU 版本的 YACC。学习过《编译原理》的同学们对这两个神器应该不会陌生。使用这两个工具，不仅可以实现一个表达式解析计算器，还可以用来解析 SQL 语法，如 PostgreSQL 中就是使用他们实现 SQL 语法解析。甚至还可以实现一个编译器。

因为是 Unix 上的工具，在 Windows 上需要使用 winflexbison，可以从 sourceforge 上下载：https://sourceforge.net/projects/winflexbison/

本文介绍如何使用 Flex 来理解 OpenCASCADE 中 ExprInterp 的实现，开阔视野。OpenCASCADE 中 ExprInterp 的用法见：**Evaluate Math Expression**

http://www.cppblog.com/eryar/archive/2013/10/09/203625.html

2.Flex Example

FLEX 是一个自动化工具，可以按照定义好的规则自动生成一个 C 函数 yylex()，也成为扫描器（Scanner）。这个 C 函数把文本串作为输入，按照定义好的规则分析文本串中的字符，找到符合规则的一些字符序列后，就执行在规则中定义好的动作（Action）。例如在规则中可以这样定义：如果遇到一个换行字符\n，那么就把行计数器的值加一。

Flex 文件就是一个文本文件，内容包括定义好的一系列词法规则。文件的命名习惯上以小写字母 l(L)来作为文件后缀。如果为了清晰，也可以用.flx 或者.flex 作为文件的后缀名。Flex 文件完成后，就执行下列命令：

$ flex example.flex

这个命令执行后将生成一个 C 文件，默认文件名为 lex.yy.c。这个 C 文件主要内容就是函数 yylex()的定义。

如果要直接将这个文件编译成为一个可执行程序，还有一些要注意的地方。如果在 Flex 文件中没有提供 main()函数的定义，那么这个 C 文件中不会有 main()函数。此时单独编译这个 C 文件的时候，一定要加上-lfl 的连接库参数；若提供了 main()函数，就不必要提供这个连接库参数了。连接库 libfl 提供了一个缺省的 main 函数。缺省的 main()函数中只是简单地调用 yyflex()函数，而自己提供的 main()函数则可以根据需要加入许多其他的处理代码。

词法规范定义文件给出了单词构成规则。词法文件在习惯上用字母 l(即 L 的小写)来作为后缀。Flex 文件由三个部分组成。或者说三个段。三个段之间用两个%%分隔。

定义段(definitions)

%%

规则段(rules)

%%

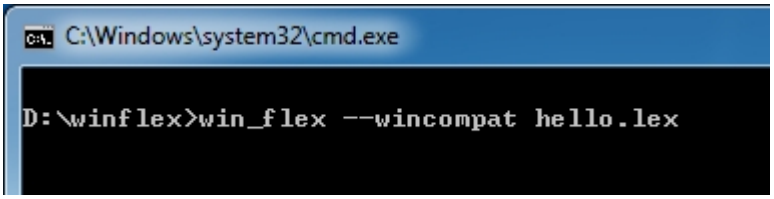
用户代码段(user code)

下面给出一个简单的 Flex 程序，代码如下所示：

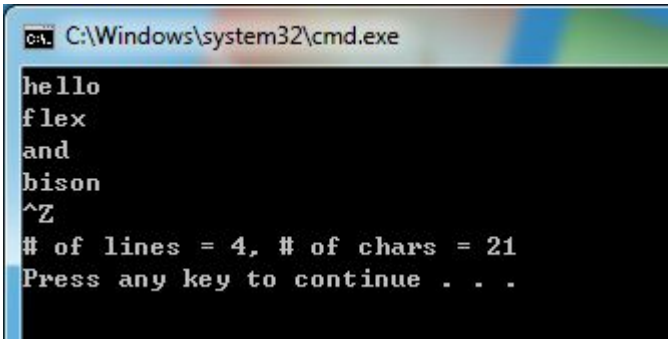
```
/* hello world for Flex. */
%option noyywrap
%{
int char_count = 0;
int line_count = 0;
%}

%%

\n { ++char_count; ++line_count; }
.  { ++char_count; }

%%

int main(int argc, char* argv[])
{
    yylex();
```

```
    printf("# of lines = %d, # of chars = %d\n", line_count, char_count);

    return 0;
}
```

将上述文件保存为 hello.lex，然后运行如下图所示命令：

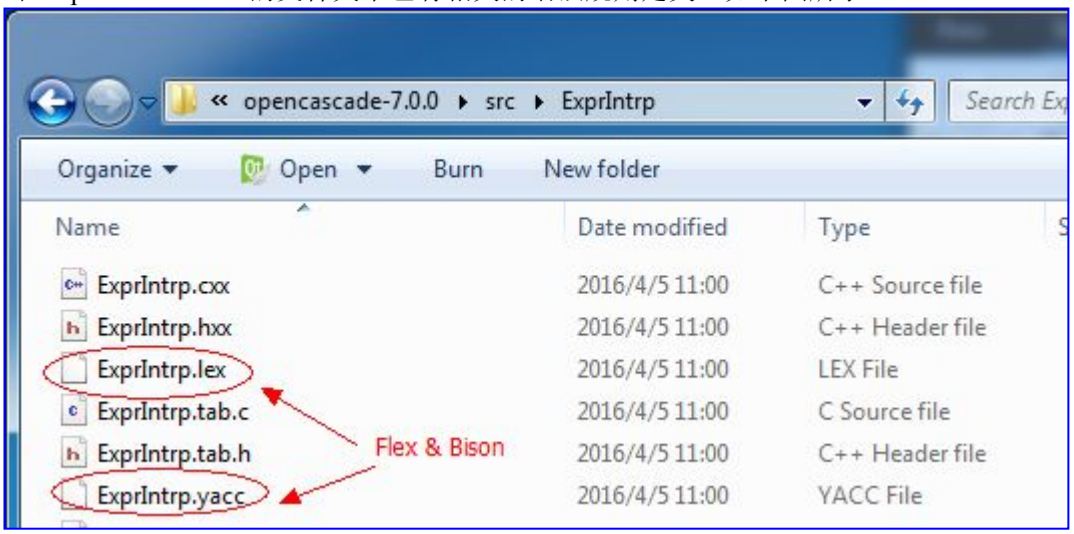


会生成一个 lex.yy.c 的源程序，将这个 C 源文件编译，链接即可生成一个可执行程序。运行程序如下图所示：

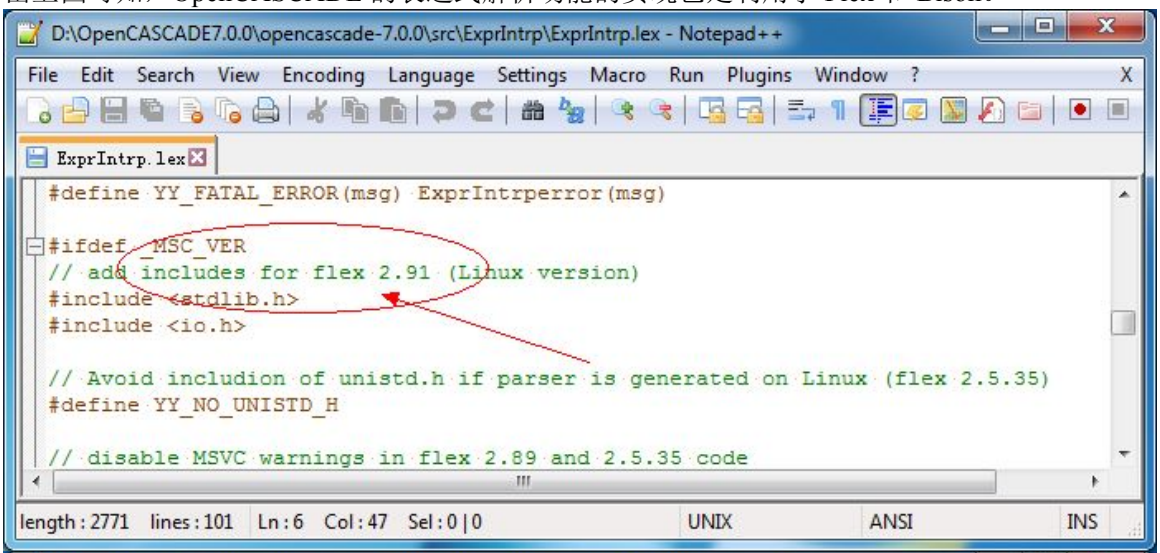


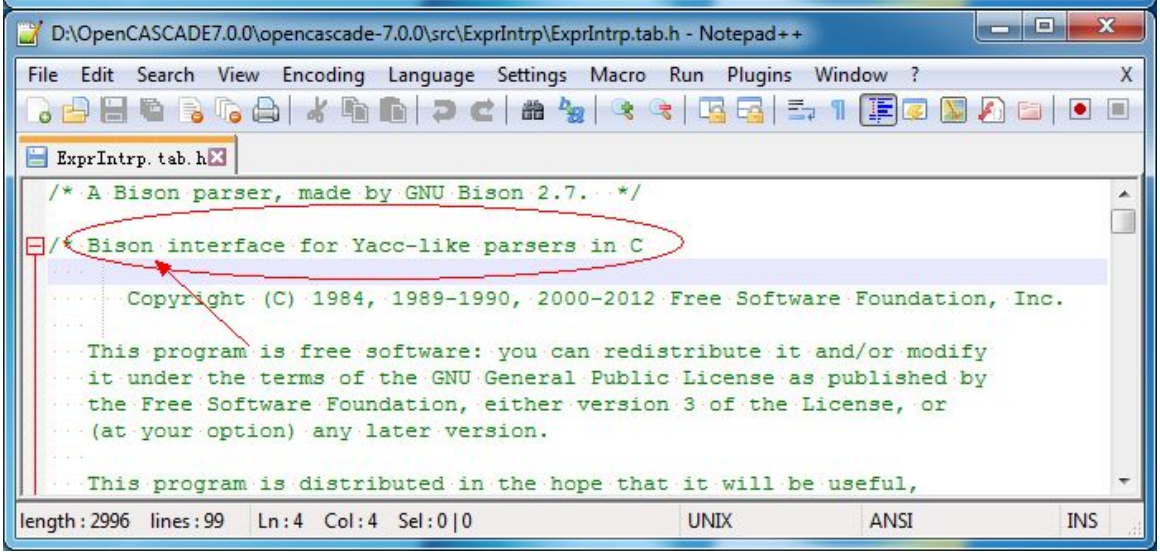
3.Flex and Bison in occ

在 OpenCASCADE 的文件夹中也有相关的语法规则定义，如下图所示：



由上图可知，OpenCASCADE 的表达式解析功能的实现也是利用了 Flex 和 Bison。





4.Conclusion

借助于 Flex 和 Bison 这两个强大的工具，你可以实现一个高级的计算器，即任意数学表达式计算器。

OpenCASCADE 的 ExprInterp 使用了 Flex 和 Bison 实现了数学表达式的解析计算。当理解了工具的用途，有兴趣的读者不妨结合《编译原理》等理论知识，对工具的原理进行一番探究。

5.References

1. 严蔚敏, 吴伟民. 数据结构(C 语言版). 清华大学出版社. 1997
2. 赵建华, 郑滔, 戴新宇 译. 编译原理. 机械工业出版社. 2011
3. John Levine, flex & bison. O'REILLY. 2009