实验四 借助 Flex/Bison 进行语法分析

一．说明：

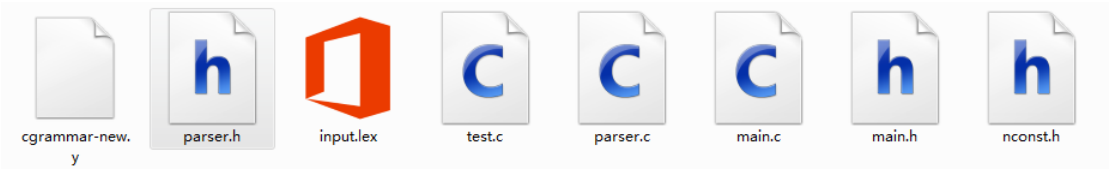
利用附录提供的 C 语言文法的相关参考资料，利用 Yacc/Bison 编写一个 C 语言分析器。

二．具体内容：

利用语法分析器生成工具 Bison 编写一个语法分析程序，与词法分析器结合，能够根据语言的上下文无关文法，识别输入的单词序列是否文法的句子。

三．实验要求：

实验资料如下：



3.1 阅读 Flex 源文件 input.lex、Bison 源文件 cgrammar-new.y。

3.2 实现 C 语言的语法分析功能，最后上机调试。

3.3 生成语法分析程序 2_2.exe，以给定的测试文件作为输入，输出运行结果到输出文件中。

四．实验过程：

（1）执行以下命令，生成 lex.yy.c、cgrammar-new.tab.h、cgrammar-new.tab.c。

```
C:\GnuWin32\bin>flex -l input.lex
C:\GnuWin32\bin>bison -d cgrammar-new.y
cgrammar-new.y: conflicts: 1 shift/reduce
```

（2）cgrammar-new.y 有移近规约冲突。

执行命令 bison -d cgrammar-new.y 后，Bison 提示移近规约冲突“cgrammar-new.y: conflicts: 1 shift/reduce”。以 Bison 的"-v"选项生成状态机描述文件 cgrammar-new.output，即执行 bison -d cgrammar-new.y。

```
C:\GnuWin32\bin>bison -v cgrammar-new.y
```

cgrammar-new.output 文件内容如下：

```
1  State 341 conflicts: 1 shift/reduce
```

```
6151  state 341
6152
6153    168 Stmt: IF '(' Exp ')' Stmt .
6154    169     | IF '(' Exp ')' Stmt . ELSE Stmt
6155
6156      ELSE  shift, and go to state 347
6157
6158      ELSE      [reduce using rule 168 (Stmt)]
6159      $default  reduce using rule 168 (Stmt)
```

修改以下两处：

2.1 在 yacc 的头部加入

%nonassoc LOWER_THAN_ELSE

%nonassoc ELSE

```
64  %start TransUnit
65
66  %nonassoc LOWER_THAN_ELSE
67  %nonassoc ELSE
68  %%
69
```

2.2 在 355 行加入

%prec LOWER_THAN_ELSE

```
355     | IF '(' Exp ')' Stmt %prec LOWER_THAN_ELSE { $$ = link(if_, $3, $5, 0); }
356     | IF '(' Exp ')' Stmt ELSE Stmt { $$ = link(ifelse_, $3, $5, $7, 0); }
357     | SWITCH '(' Exp ')' Stmt { $$ = link(switch_, $3, $5, 0);}
```

（3）编译

使用 cl.exe 或 gcc 编译器，编译 lex.yy.c cgrammar-new.tab.c main.c parser.c。使用 cl.exe 编译后，得到以下错误提示：

```
C:\GnuWin32\bin>cl lex.yy.c cgrammar-new.tab.c main.c parser.c
用于 80x86 的 Microsoft (R) 32 位 C/C++ 优化编译器 16.00.40219.01 版
版权所有(C) Microsoft Corporation。保留所有权利。

lex.yy.c
cgrammar-new.tab.c
main.c
parser.c
正在生成代码...
Microsoft (R) Incremental Linker Version 10.00.40219.01
Copyright (C) Microsoft Corporation.    All rights reserved.

/out:lex.yy.exe
lex.yy.obj
cgrammar-new.tab.obj
main.obj
```

```
parser.obj
lex.yy.obj : error LNK2019: 无法解析的外部符号 _yyinput，该符号在函数 _comment 中被引用
lex.yy.exe : fatal error LNK1120: 1 个无法解析的外部命令
```

修改 lex.yy.c，使其能顺利编译。

3.1 将 lex.yy.c 中的

#ifdef __cplusplus

static int yyinput()

#else

static int input()

#endif

```
1707  //#ifdef __cplusplus
1708  static int yyinput()
1709  #else
1710  static int input()
1711  #endif
1712      {
1713      int c;
1714
1715      *yy_c_buf_p = yy_hold_char;
```

改为

static int yyinput()

```
1707  static int yyinput()
1708      {
1709      int c;
1710
1711      *yy_c_buf_p = yy_hold_char;
1712
```

修改后重新编译，错误提示如下:

```
C:\GnuWin32\bin>cl lex.yy.c cgrammar-new.tab.c main.c parser.c
用于 80x86 的 Microsoft (R) 32 位 C/C++ 优化编译器 16.00.40219.01 版
版权所有(C) Microsoft Corporation。保留所有权利。

lex.yy.c
input.lex(223) : error C2129: 静态函数“int input()”已声明但未定义
        lex.yy.c(660) : 参见“input”的声明
cgrammar-new.tab.c
main.c
parser.c
正在生成代码...
```

2.2 将 lex.yy.c 中的

#ifdef __cplusplus

return yyinput();

#else

return input();

#endif

```
1746                    case EOB_ACT_END_OF_FILE:
1747                        {
1748                        if ( yywrap() )
1749                            return EOF;
1750
1751                        if ( ! yy_did_buffer_switch_on_eof )
1752                            YY_NEW_FILE;
1753    #ifdef __cplusplus
1754                        return yyinput();
1755    #else
1756                        return input();
1757    #endif
1758                        }
```

改为

return yyinput();

```
1746                    case EOB_ACT_END_OF_FILE:
1747                        {
1748                        if ( yywrap() )
1749                            return EOF;
1750
1751                        if ( ! yy_did_buffer_switch_on_eof )
1752                            YY_NEW_FILE;
1753
1754                        return yyinput();
1755
1756                        }
```

（3）生成可执行文件 2_2.exe，并分析源文件 test.c。

使用 cl 命令编译时，增加参数 o"2_2.exe"，制定输出的可执行文件名。

```
C:\GnuWin32\bin>cl lex.yy.c cgrammar-new.tab.c main.c parser.c -o"2_2.exe"
用于 80x86 的 Microsoft (R) 32 位 C/C++ 优化编译器 16.00.40219.01 版
版权所有(C) Microsoft Corporation。保留所有权利。

cl: 命令行 warning D9035 :“o”选项已否决，并将在将来的版本中移除
lex.yy.c
cgrammar-new.tab.c
main.c
parser.c
正在生成代码...
Microsoft (R) Incremental Linker Version 10.00.40219.01
Copyright (C) Microsoft Corporation.    All rights reserved.

/out:lex.yy.exe
/out:2_2.exe
lex.yy.obj
cgrammar-new.tab.obj
main.obj
parser.obj
```

执行 C:\GnuWin32\bin>2_2.exe < test.c，分析源文件 test.c。得到以下输出：

```
C:\GnuWin32\bin>2_2.exe < test.c



/
void main()
{
        int i = 0;
```

```
        int j = 0;

}

void t1()
{
        int i = 0;
}
/

typedef unsigned int uint;

uint xx;
uint yy;


Abstract Syntax Tree ...

  node   prev   next parent child   line    sti
    33      0      0      0     32      9      0      0      0   + goal_
    32      0     55     33     31      9      0      0      0     + extdef_
    31      0      0     32      7      9      0      0      0     | + funcdef_
     7      0     30     31      3      5      0      0      0     |     + funcdecl_
     3      0      6      7      1      4      0      0      0     |     | + decl_spec_
     1      0      0      3      0      4      0      0      0     |     | | + void_
     6      3      0      0      5      5      0      0      0     |     | + direct_decl_

     5      0      0      6      4      4      0      0      0     |     |     + funcdecl_
     4      0      0      5      2      4      0      0      0     |     |         + ident_
     2      0      0      4      0      4      1      0      0     |     |             + IDENT_(main)
    30      7      0      0     29      9      0      0      0     |     + funcbody_
    29      0      0     30     18      9      0      0      0     |         + compound_stmt_
    18      0      0     29     17      6      0      0      0     |             + declarations_
    17      0     28     18     10      6      0      0      0     |             + decl_init_
    10      0     16     17      8      6      0      0      0     |             | + decl_spec_
     8      0      0     10      0      6      0      0      0     |             | | + int_

    16     10      0      0     15      6      0      0      0     |             | + init_declarators_
    15      0      0     16     12      6      0      0      0     |             |     + declaratorinit_
    12      0     14     15     11      6      0      0      0     |             |         + direct_decl_
    11      0      0     12      9      6      0     56    157     |             |         | +ident_
     9      0      0     11      0      6      2      0      0     |             |         |+ IDENT_ (i)
    14     12      0      0     13      6      0      0      0     |             |         + assign_
    13      0      0     14      0      6      3      0      0     |             |             +CONST_ (0)
```

```
 28  17   0   0  21   7   0   0   0  |           + decl_init_
 21   0  27  28  19   7   0   0   0  |             + decl_spec_
 19   0   0  21   0   7   0   0   0  |             | + int_

 27  21   0   0  26   7   0   0   0  |             + init_declarators_
 26   0   0  27  23   7   0   0   0  |               + declaratorinit_
 23   0  25  26  22   7   0   0   0  |                 + direct_decl_
 22   0   0  23  20   7   0   0   0  |                 | +ident_
 20   0   0  22   0   7   4   0   0  |                 |+ IDENT_ (j)
 25  23   0   0  24   7   0   0   0  |                 + assign_
 24   0   0  25   0   7   3   0   0  |                   +CONST_ (0)
 55  32  66   0  54  14   0   0   0  + extdef_
 54   0   0  55  40  14   0   0   0  | + funcdef_
 40   0  53  54  36  12   0   0   0  |   + funcdecl_
 36   0  39  40  34  11   0   0   0  |   | + decl_spec_
 34   0   0  36   0  11   0   0   0  |   | | + void_
 39  36   0   0  38  12   0   0   0  |   | + direct_decl_

 38   0   0  39  37  11   0   0   0  |   |   + funcdecl_
 37   0   0  38  35  11   0   0   0  |   |     + ident_
 35   0   0  37   0  11   5   0   0  |   |       + IDENT_ (t1)
 53  40   0   0  52  14   0   0   0  |   + funcbody_
 52   0   0  53  51  14   0   0   0  |     + compound_stmt_
 51   0   0  52  50  13   0   0   0  |       + declarations_
 50   0   0  51  43  13   0   0   0  |         + decl_init_
 43   0  49  50  41  13   0   0   0  |           + decl_spec_
 41   0   0  43   0  13   0   0   0  |           | + int_

 49  43   0   0  48  13   0   0   0  |           + init_declarators_
 48   0   0  49  45  13   0   0   0  |             + declaratorinit_
 45   0  47  48  44  13   0   0   0  |               + direct_decl_
 44   0   0  45  42  13   0   0   0  |               | +ident_
 42   0   0  44   0  13   2   0   0  |               |+ IDENT_ (i)
 47  45   0   0  46  13   0   0   0  |               + assign_
 46   0   0  47   0  13   3   0   0  |                 +CONST_ (0)
 66  55  75   0  65  17   0   0   0  + extdef_
 65   0   0  66  56  17   0   0   0  | + decl_init_
 56   0  64  65  57  17   0   0   0  |   + typedef_
 57   0   0  56  60  17   0   0   0  |   | + unsigned_
 60   0   0  57  58  17   0   0   0  |   |   + decl_spec_

 58   0   0  60   0  17   0   0   0  |   |     + int_
 64  56   0   0  63  17   0   0   0  |   + init_declarators_
 63   0   0  64  62  17   0   0   0  |     + declarator_
```

```
    62     0     0    63    61    17     0     0     0     |           + direct_decl_
    61     0     0    62    59    17     0     0     0     |               + ident_
    59     0     0    61     0    17     6     0     0     |                   + IDENT_ (uint)
    75    66    84     0    74    19     0     0     0     + extdef_
    74     0     0    75    69    19     0     0     0     | + decl_init_
    69     0    73    74    67    19     0     0     0     |     + decl_spec_
    67     0     0    69     0    19     6     0     0     |     | + type_name_ (uint)
    73    69     0     0    72    19     0     0     0     |     + init_declarators_
    72     0     0    73    71    19     0     0     0     |         + declarator_
    71     0     0    72    70    19     0     0     0     |             + direct_decl_
    70     0     0    71    68    19     0     0     0     |                 + ident_
    68     0     0    70     0    19     7     0     0     |                     + IDENT_ (xx)
    84    75     0     0    83    20     0     0     0     + extdef_
    83     0     0    84    78    20     0     0     0       + decl_init_
    78     0    82    83    76    20     0     0     0         + decl_spec_
    76     0     0    78     0    20     6     0     0         | + type_name_ (uint)
    82    78     0     0    81    20     0     0     0         + init_declarators_
    81     0     0    82    80    20     0     0     0           + declarator_
    80     0     0    81    79    20     0     0     0             + direct_decl_
    79     0     0    80    77    20     0     0     0                 + ident_
    77     0     0    79     0    20     8     0     0                     + IDENT_ (yy)

AST is empty.

Created out.txt ...
```

输出文件 out.txt，包含符号表和抽象语法树，内容如下：

```
Symbol Table ...

  sti    leng    type    term
    1       4       0       1    <identifier>                    main
    2       1       0       1    <identifier>                    i
    3       1       0       2    <constant>                        0
    4       1       0       1    <identifier>                    j
    5       2       0       1    <identifier>                    t1
    6       4       0       4    {typedef}                         uint
    7       2       0       1    <identifier>                    xx
    8       2       0       1    <identifier>                    yy

Abstract Syntax Tree ...

  node    prev    next parent child    line     sti
    33       0       0       0    32       9       0       0       0     + goal_
    32       0      55      33    31       9       0       0       0         + extdef_
```

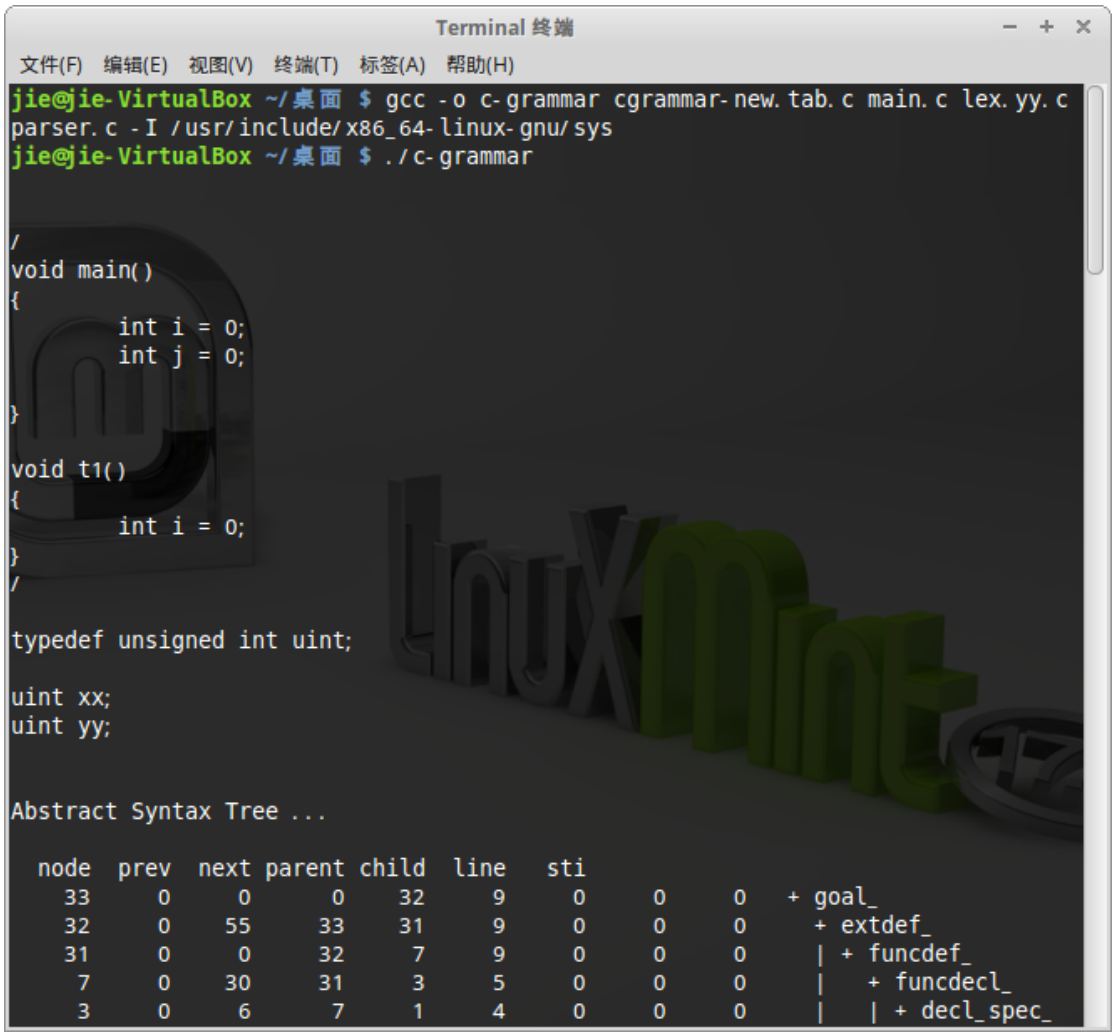
```
 31   0   0  32   7   9   0   0   0  | + funcdef_
  7   0  30  31   3   5   0   0   0  |     + funcdecl_
  3   0   6   7   1   4   0   0   0  |     | + decl_spec_
  1   0   0   3   0   4   0   0   0  |     | | + void_
  6   3   0   0   5   5   0   0   0  |     | + direct_decl_
  5   0   0   6   4   4   0   0   0  |     |     + funcdecl_
  4   0   0   5   2   4   0   0   0  |     |         + ident_
  2   0   0   4   0   4   1   0   0  |     |             + IDENT_ (main)
 30   7   0   0  29   9   0   0   0  |     + funcbody_
 29   0   0  30  18   9   0   0   0  |         + compound_stmt_
 18   0   0  29  17   6   0   0   0  |             + declarations_
 17   0  28  18  10   6   0   0   0  |                 + decl_init_
 10   0  16  17   8   6   0   0   0  |                 | + decl_spec_
  8   0   0  10   0   6   0   0   0  |                 | | + int_
 16  10   0   0  15   6   0   0   0  |                 | + init_declarators_
 15   0   0  16  12   6   0   0   0  |                 |     + declaratorinit_
 12   0  14  15  11   6   0   0   0  |                 |         + direct_decl_
 11   0   0  12   9   6   0  56 157  |                 |         | + ident_
  9   0   0  11   0   6   2   0   0  |                 |         |     + IDENT_ (i)
 14  12   0   0  13   6   0   0   0  |                 |         + assign_
 13   0   0  14   0   6   3   0   0  |                 |             + CONST_ (0)
 28  17   0   0  21   7   0   0   0  |                 + decl_init_
 21   0  27  28  19   7   0   0   0  |                     + decl_spec_
 19   0   0  21   0   7   0   0   0  |                     | + int_
 27  21   0   0  26   7   0   0   0  |                     + init_declarators_
 26   0   0  27  23   7   0   0   0  |                         + declaratorinit_
 23   0  25  26  22   7   0   0   0  |                             + direct_decl_
 22   0   0  23  20   7   0   0   0  |                             | + ident_
 20   0   0  22   0   7   4   0   0  |                             |     + IDENT_ (j)
 25  23   0   0  24   7   0   0   0  |                             + assign_
 24   0   0  25   0   7   3   0   0  |                                 + CONST_ (0)
 55  32  66   0  54  14   0   0   0  + extdef_
 54   0   0  55  40  14   0   0   0  | + funcdef_
 40   0  53  54  36  12   0   0   0  |     + funcdecl_
 36   0  39  40  34  11   0   0   0  |     | + decl_spec_
 34   0   0  36   0  11   0   0   0  |     | | + void_
 39  36   0   0  38  12   0   0   0  |     | + direct_decl_
 38   0   0  39  37  11   0   0   0  |     |     + funcdecl_
 37   0   0  38  35  11   0   0   0  |     |         + ident_
 35   0   0  37   0  11   5   0   0  |     |             + IDENT_ (t1)
 53  40   0   0  52  14   0   0   0  |     + funcbody_
 52   0   0  53  51  14   0   0   0  |         + compound_stmt_
 51   0   0  52  50  13   0   0   0  |             + declarations_
 50   0   0  51  43  13   0   0   0  |                 + decl_init_
```

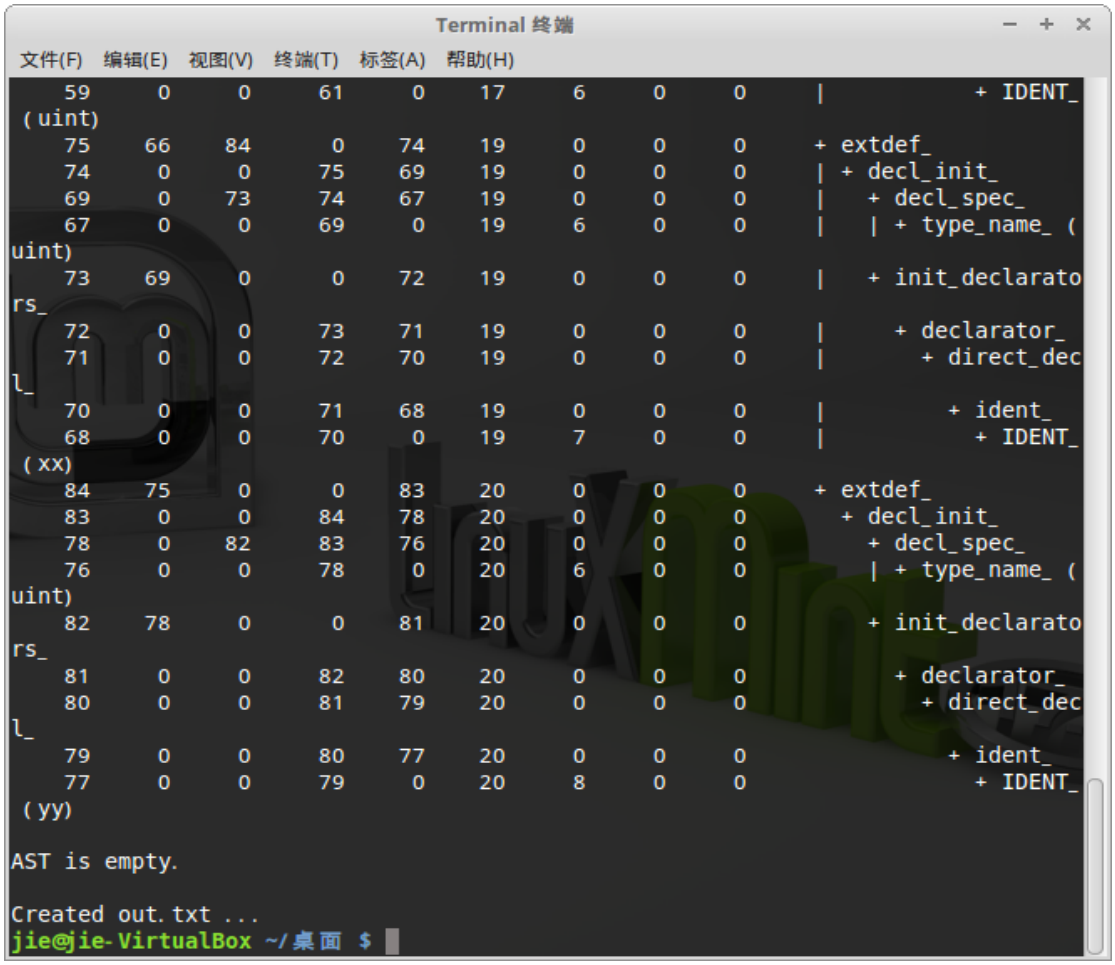
```
   43    0   49   50   41   13    0    0    0    |              + decl_spec_
   41    0    0   43    0   13    0    0    0    |              | + int_
   49   43    0    0   48   13    0    0    0    |              + init_declarators_
   48    0    0   49   45   13    0    0    0    |                + declaratorinit_
   45    0   47   48   44   13    0    0    0    |                  + direct_decl_
   44    0    0   45   42   13    0    0    0    |                  | + ident_
   42    0    0   44    0   13    2    0    0    |                  |     + IDENT_ (i)
   47   45    0    0   46   13    0    0    0    |                  + assign_
   46    0    0   47    0   13    3    0    0    |                    + CONST_ (0)
   66   55   75    0   65   17    0    0    0    + extdef_
   65    0    0   66   56   17    0    0    0    | + decl_init_
   56    0   64   65   57   17    0    0    0    |     + typedef_
   57    0    0   56   60   17    0    0    0    |     | + unsigned_
   60    0    0   57   58   17    0    0    0    |     |     + decl_spec_
   58    0    0   60    0   17    0    0    0    |     |       + int_
   64   56    0    0   63   17    0    0    0    |     + init_declarators_
   63    0    0   64   62   17    0    0    0    |       + declarator_
   62    0    0   63   61   17    0    0    0    |         + direct_decl_
   61    0    0   62   59   17    0    0    0    |           + ident_
   59    0    0   61    0   17    6    0    0    |             + IDENT_ (uint)
   75   66   84    0   74   19    0    0    0    + extdef_
   74    0    0   75   69   19    0    0    0    | + decl_init_
   69    0   73   74   67   19    0    0    0    |     + decl_spec_
   67    0    0   69    0   19    6    0    0    |     | + type_name_ (uint)
   73   69    0    0   72   19    0    0    0    |     + init_declarators_
   72    0    0   73   71   19    0    0    0    |       + declarator_
   71    0    0   72   70   19    0    0    0    |         + direct_decl_
   70    0    0   71   68   19    0    0    0    |           + ident_
   68    0    0   70    0   19    7    0    0    |             + IDENT_ (xx)
   84   75    0    0   83   20    0    0    0    + extdef_
   83    0    0   84   78   20    0    0    0      + decl_init_
   78    0   82   83   76   20    0    0    0        + decl_spec_
   76    0    0   78    0   20    6    0    0        | + type_name_ (uint)
   82   78    0    0   81   20    0    0    0        + init_declarators_
   81    0    0   82   80   20    0    0    0          + declarator_
   80    0    0   81   79   20    0    0    0            + direct_decl_
   79    0    0   80   77   20    0    0    0              + ident_
   77    0    0   79    0   20    8    0    0                + IDENT_ (yy)

End of Output.
```

【说明】Linux 上的 io.h 头文件没有包含在标准库/usr/include 中，使用命令 find /usr/include –name io.h 找到 io.h 的位置，使用参数-I 将其路径加入到头文件中。

结果及文件如下：

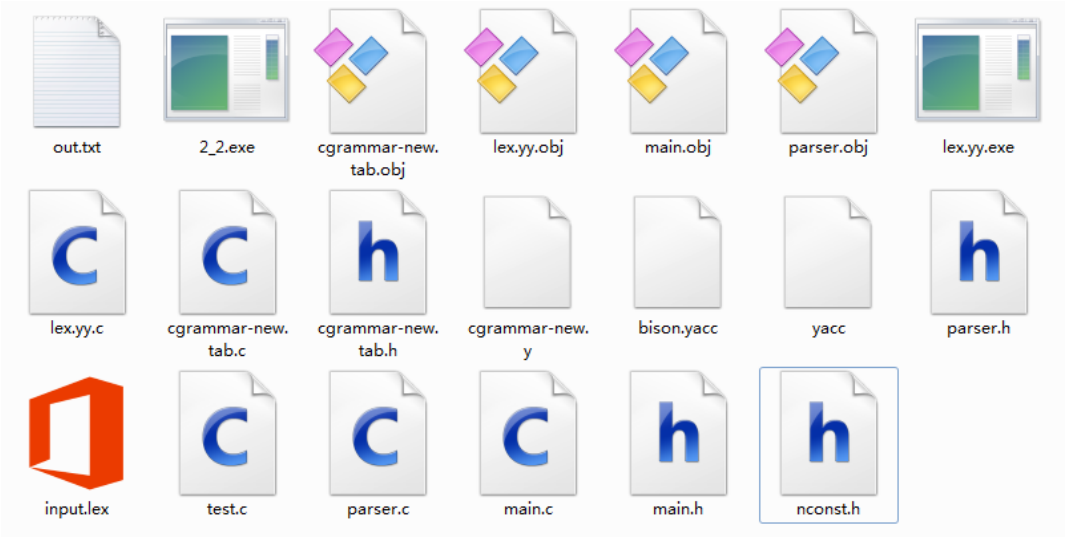

Terminal 终端
文件(F) 编辑(E) 视图(V) 终端(T) 标签(A) 帮助(H)
jie@jie-VirtualBox ~/桌面 $ gcc -o c-grammar cgrammar-new.tab.c main.c lex.yy.c parser.c -I /usr/include/x86_64-linux-gnu/sys
jie@jie-VirtualBox ~/桌面 $ ./c-grammar
/
void main()
{
int i = 0;
int j = 0;
}
void t1()
{
int i = 0;
}
/
typedef unsigned int uint;
uint xx;
uint yy;
Abstract Syntax Tree ...
node prev next parent child line sti
33 0 0 0 32 9 0 0 0 + goal_
32 0 55 33 31 9 0 0 0 + extdef_
31 0 0 32 7 9 0 0 0 | + funcdef_
7 0 30 31 3 5 0 0 0 | + funcdecl_
3 0 6 7 1 4 0 0 0 | | + decl_spec_


Terminal 终端
文件(F) 编辑(E) 视图(V) 终端(T) 标签(A) 帮助(H)
59 0 0 61 0 17 6 0 0 | + IDENT_
(uint)
75 66 84 0 74 19 0 0 0 + extdef_
74 0 0 75 69 19 0 0 0 | + decl_init_
69 0 73 74 67 19 0 0 0 | + decl_spec_
67 0 0 69 0 19 6 0 0 | | + type_name_ (
uint)
73 69 0 0 72 19 0 0 0 | + init_declarato
rs_
72 0 0 73 71 19 0 0 0 | + declarator_
71 0 0 72 70 19 0 0 0 | + direct_dec
l_
70 0 0 71 68 19 0 0 0 | + ident_
68 0 0 70 0 19 7 0 0 | + IDENT_
(xx)
84 75 0 0 83 20 0 0 0 + extdef_
83 0 0 84 78 20 0 0 0 + decl_init_
78 0 82 83 76 20 0 0 0 + decl_spec_
76 0 0 78 0 20 6 0 0 | + type_name_ (
uint)
82 78 0 0 81 20 0 0 0 + init_declarato
rs_
81 0 0 82 80 20 0 0 0 + declarator_
80 0 0 81 79 20 0 0 0 + direct_dec
l_
79 0 0 80 77 20 0 0 0 + ident_
77 0 0 79 0 20 8 0 0 + IDENT_
(yy)
AST is empty.
Created out.txt ...
jie@jie-VirtualBox ~/桌面 $


主文件夹
lex.yy.c
parser.h
cgrammar-new.tab.c
main.c
test.c
cgrammar-new.tab.h
main.h
c-grammar
cgrammar-new.y
nconst.h
out.txt
input.lex
parser.c

五．实验成果：

5.1 打包压缩以下文件为”2_2 代码.rar”，提交到服务器。



5.2 提交报告，详细描述实验过程。分析输出结果中符号表和抽象语法树的生成过程。