

```
1: %{
2:  // $Id: lexer.l,v 1.9 2014-10-09 18:39:47-07 - - $
3:
4:  #include "tokens.h"
5:
6:  %{
7:
8:  %option 8bit
9:  %option debug
10: %option nodefault
11: %option nounput
12: %option noyywrap
13: %option verbose
14: %option warn
15:
16: LETTER      [a-zA-Z_]
17: DIGIT       [0-9]
18: IDENT       {LETTER} ({LETTER} | {DIGIT}) *
19: MANTISSA     {DIGIT}+ \. ? {DIGIT} * | \. {DIGIT}+
20: EXPONENT     [Ee] [+-] ? {DIGIT}+
21: NUMBER       ({MANTISSA}) ({EXPONENT}) ?
22:
23: %%
24:
25: { IDENT }      { return IDENT; }
26: { NUMBER }     { return NUMBER; }
27: "+"           { return '+'; }
28: "-"           { return '-'; }
29: "*"           { return '*'; }
30: "/"           { return '/'; }
31: "="           { return '='; }
32: ";"           { return ';' ; }
33: \n            { return '\n'; }
34: [ \t]+        { /* Skip whitespace. */ }
35: "#".*         { /* Ignore comment. */ }
36: .             { return ERROR; }
37:
38: %%
39:
```

```
1: // $Id: tokens.h,v 1.7 2014-10-09 18:39:25-07 - - $
2:
3: #ifndef __TOKENS_H__
4: #define __TOKENS_H__
5:
6: #define YYEOF      0
7: #define IDENT      260
8: #define NUMBER     261
9: #define ERROR      262
10:
11: extern int yy_flex_debug;
12: extern char* yytext;
13: extern int yylex (void);
14:
15: #endif
16:
```

```
1: // $Id: main.cc,v 1.3 2014-10-09 18:39:25-07 - - $
2:
3: #include <errno.h>
4: #include <getopt.h>
5: #include <libgen.h>
6: #include <stdio.h>
7: #include <stdlib.h>
8: #include <string.h>
9:
10: #include "tokens.h"
11:
12: int main (void) {
13:     yy_flex_debug = 1;
14:     int linenr = 1;
15:
16:     for (;;) {
17:         int token = yylex();
18:         if (yy_flex_debug) fflush (NULL);
19:         switch (token) {
20:             case YYEOF:
21:                 printf ("END OF FILE\n");
22:                 return 0;
23:             case IDENT:
24:                 printf ("IDENT \"%s\"\n", yytext);
25:                 break;
26:             case NUMBER:
27:                 printf ("NUMBER \"%s\"\n", yytext);
28:                 break;
29:             case '+':
30:             case '-':
31:             case '*':
32:             case '/':
33:             case '=':
34:             case ';':
35:                 printf ("OPERATOR \"%s\"\n", yytext);
36:                 break;
37:             case '\n':
38:                 printf ("NEWLINE\n");
39:                 ++linenr;
40:                 break;
41:             default:
42:                 printf ("ERROR \"%s\"\n", yytext);
43:         }
44:     }
45:
46:     return EXIT_SUCCESS;
47: }
48:
```

```
1: # $Id$
2:
3: GPP      = g++ -g -O0 -std=gnu++11 -Wall -Wextra
4:
5: SOURCES = lexer.l tokens.h main.cc Makefile
6: EXECBIN = scantokens
7: OBJECTS = lexer.o main.o
8:
9: all : ${EXECBIN}
10:
11: ${EXECBIN} : ${OBJECTS}
12:             ${GPP} ${OBJECTS} -o ${EXECBIN}
13:
14: %.o : %.cc tokens.h
15:             ${GPP} -c $<
16:
17: lexer.o : lexer.l
18:             flex -olexer.cc lexer.l
19:             ${GPP} -c lexer.cc -o lexer.o
20:
21: clean :
22:         - rm lexer.cc ${OBJECTS} test.out
23:
24: spotless : clean
25:         - rm ${EXECBIN} Listing.ps Listing.pdf
26:
27: ci : ${SOURCES}
28:         - checksource ${SOURCES}
29:         cid + ${SOURCES}
30:
31: test : ${EXECBIN}
32:         (echo '# This is a comment.' \
33:         ;echo '12.3 12.0 .3 1123.4e4 ;' \
34:         ;echo 'hello = world +.1234;' \
35:         ;echo 'this * that + foo / bar;' \
36:         ;echo '+ - * /' \
37:         ;echo 'Errors . : ( )' \
38:         ) | ${EXECBIN} >test.out 2>&1
39:
40: lis : test
41:         mkpspdf Listing.ps ${SOURCES} test.out
42:
43: again :
44:         gmake spotless ci all lis
45:
```

```
1: --(end of buffer or a NUL)
2: --accepting rule at line 35 ("# This is a comment.")
3: --accepting rule at line 33 ("
4: ")
5: --accepting rule at line 26 ("12.3")
6: NEWLINE
7: --accepting rule at line 34 (" ")
8: --accepting rule at line 26 ("12.0")
9: NUMBER "12.3"
10: --accepting rule at line 34 (" ")
11: --accepting rule at line 26 (".3")
12: NUMBER "12.0"
13: --accepting rule at line 34 (" ")
14: --accepting rule at line 26 ("1123.4e4")
15: NUMBER ".3"
16: --accepting rule at line 34 (" ")
17: --accepting rule at line 32 (";")
18: NUMBER "1123.4e4"
19: --accepting rule at line 33 ("
20: ")
21: OPERATOR ";"
22: --accepting rule at line 25 ("hello")
23: NEWLINE
24: --accepting rule at line 34 (" ")
25: --accepting rule at line 31 ("=")
26: IDENT "hello"
27: --accepting rule at line 34 (" ")
28: --accepting rule at line 25 ("world")
29: OPERATOR "="
30: --accepting rule at line 34 (" ")
31: --accepting rule at line 27 ("+")
32: IDENT "world"
33: --accepting rule at line 26 (".1234")
34: OPERATOR "+"
35: --accepting rule at line 32 (";")
36: NUMBER ".1234"
37: --accepting rule at line 33 ("
38: ")
39: OPERATOR ";"
40: --accepting rule at line 25 ("this")
41: NEWLINE
42: --accepting rule at line 34 (" ")
43: --accepting rule at line 29 ("*")
44: IDENT "this"
45: --accepting rule at line 34 (" ")
46: --accepting rule at line 25 ("that")
47: OPERATOR "*"
48: --accepting rule at line 34 (" ")
49: --accepting rule at line 27 ("+")
50: IDENT "that"
51: --accepting rule at line 34 (" ")
52: --accepting rule at line 25 ("foo")
53: OPERATOR "+"
54: --accepting rule at line 34 (" ")
55: --accepting rule at line 30 ("/")
56: IDENT "foo"
57: --accepting rule at line 34 (" ")
58: --accepting rule at line 25 ("bar")
```

```
59: OPERATOR "/"
60: --accepting rule at line 32 (";")
61: IDENT "bar"
62: --accepting rule at line 33 ("
63: ")
64: OPERATOR ";"
65: --accepting rule at line 27 ("+"
66: NEWLINE
67: --accepting rule at line 34 (" ")
68: --accepting rule at line 28 ("-"
69: OPERATOR "+"
70: --accepting rule at line 34 (" ")
71: --accepting rule at line 29 ("*"
72: OPERATOR "-"
73: --accepting rule at line 34 (" ")
74: --accepting rule at line 30 ("/"
75: OPERATOR "*"
76: --accepting rule at line 33 ("
77: ")
78: OPERATOR "/"
79: --accepting rule at line 25 ("Errors")
80: NEWLINE
81: --accepting rule at line 34 (" ")
82: --accepting rule at line 36 (".")
83: IDENT "Errors"
84: --accepting rule at line 34 (" ")
85: --accepting rule at line 36 (":")
86: ERROR "."
87: --accepting rule at line 34 (" ")
88: --accepting rule at line 36 ("("
89: ERROR ":"
90: --accepting rule at line 34 (" ")
91: --accepting rule at line 36 (")")
92: ERROR "("
93: --accepting rule at line 33 ("
94: ")
95: ERROR ")"
96: --(end of buffer or a NUL)
97: --EOF (start condition 0)
98: NEWLINE
99: END OF FILE
```