lex (software)

From Wikipedia, the free encyclopedia

In computer science, **lex** is a program that generates lexical analyzers ("scanners" or "lexers").[1][2] Lex is commonly used with the yacc parser generator. Lex, originally written by Eric Schmidt and Mike Lesk, is the standard lexical analyzer generator on many Unix systems, and a tool exhibiting its behavior is specified as part of the POSIX standard.

Lex reads an input stream specifying the lexical analyzer and outputs source code implementing the lexer in the C programming language.

Though traditionally proprietary software, versions of Lex based on the original AT&T code are available as open source, as part of systems such as OpenSolaris and Plan 9 from Bell Labs. Another popular open source version of Lex is Flex, the "fast lexical analyzer".

Contents

Structure of a lex file

The structure of a lex file is intentionally similar to that of a yacc file; files are divided up into three sections, separated by lines that contain only two percent signs, as follows:

```
Definition section
%%
Rules section
%%
C code section
```

- The **definition** section is the place to define macros and to import header files written in C. It is also possible to write any C code here, which will be copied verbatim into the generated source file.
- The **rules** section is the most important section; it associates patterns with C statements. Patterns are simply regular expressions. When the lexer sees some text in the input matching a given pattern, it executes the associated C code. This is the basis of how lex operates.
- The **C code** section contains C statements and functions that are copied verbatim to the generated source file. These statements presumably contain code called by the rules in the rules section. In large programs it is more convenient to place this code in a separate file and link it in at compile time.

Example of a lex file

The following is an example lex file for the flex version of lex. It recognizes strings of numbers (integers) in the input, and simply prints them out.

```
/*** Definition section ***/

%{
/* C code to be copied verbatim */
#include <stdio.h>
%}

/* This tells flex to read only one input file */
%option noyywrap

%%
    /*** Rules section ***/

    /* [0-9]+ matches a string of one or more digits */
[0-9]+  {
            /* yytext is a string containing the matched text. */
            printf("Saw an integer: %s\n", yytext);
        }

.       {   /* Ignore all other characters. */   }

%%
/*** C Code section ***/

int main(void)
{
    /* Call the lexer, then quit. */
    yylex();
    return 0;
}
```

If this input is given to flex, it will be converted into a C file, lex.yy.c. This can be compiled into an executable which matches and outputs strings of integers. For example, given the input:

```
abc123z.!&*2ghj6
```

the program will print:

```
Saw an integer: 123
Saw an integer: 2
Saw an integer: 6
```

Using Lex with other programming tools

Using Lex with parser generators

Lex and parser generators, such as Yacc or Bison, are commonly used together. Parser generators use a formal grammar to parse an input stream, something which Lex cannot do using simple regular expressions (Lex is limited to simple finite state automata). However, parser generators cannot read from a simple input stream – they require a series of tokens. Lex is often used to provide the parser generator with these tokens.

Lex and make

make is a utility that can be used to maintain programs involving lex. Make assumes that a file that has an extension of `.l` is a lex source file. The make internal macro `LFLAGS` can be used to specify lex options to be invoked automatically by make.[3]

See also

- Flex lexical analyser
- Yacc
- Ragel
- Quex
- List of C# lexer generators

References

1. ^ Levine, John R; Mason, Tony; Brown, Doug (1992). *LEX & YACC* (2 ed.). O'Reilly. pp. 1–2. ISBN 1-56592-000-7. http://books.google.co.in/books?id=YrzpxNYegEkC&printsec=frontcover#PPA1,M1.
2. ^ Levine, John (August 2009). *flex & bison*. O'Reilly Media. pp. 304. ISBN 978-0-596-15597-1. http://oreilly.com/catalog/9780596155988.
3. ^ "make", *The Open Group Base Specifications Issue 6, IEEE Std 1003.1, 2004 Edition* (The IEEE and The Open Group), 2004, http://www.opengroup.org/onlinepubs/009695399/utilities/make.html

External links

- Using Flex and Bison at Macworld.com

Retrieved from "http://en.wikipedia.org/wiki/Lex_(software)"
Categories: Compiling tools | Parser generators | Unix programming tools | Unix SUS2008 utilities