NAME

lexgrog – parse header information in man pages

SYNOPSIS

lexgrog [**–m**|**-c**] [**–dfw?V**] [**–E** *encoding*] *file* ...

DESCRIPTION

lexgrog is an implementation of the traditional "groff guess" utility in **lex**. It reads the list of files on its command line as either man page source files or preformatted "cat" pages, and displays their name and description as used by **apropos** and **whatis**, the list of preprocessing filters required by the man page before it is passed to **nroff** or **troff**, or both.

If its input is badly formatted, **lexgrog** will print "parse failed"; this may be useful for external programs that need to check man pages for correctness. If one of **lexgrog**'s input files is "–", it will read from standard input; if any input file is compressed, a decompressed version will be read automatically.

OPTIONS

–d, **––debug**
: Print debugging information.

–m, **––man**
: Parse input as man page source files. This is the default if neither **––man** nor **––cat** is given.

–c, **––cat**
: Parse input as preformatted man pages ("cat pages"). **––man** and **––cat** may not be given simultaneously.

–w, **––whatis**
: Display the name and description from the man page's header, as used by **apropos** and **whatis**. This is the default if neither **––whatis** nor **––filters** is given.

–f, **––filters**
: Display the list of filters needed to preprocess the man page before formatting with **nroff** or **troff**.

–E *encoding*, **––encoding** *encoding*
: Override the guessed character set for the page to *encoding*.

–?, **––help**
: Print a help message and exit.

––usage
: Print a short usage message and exit.

–V, **––version**
: Display version information.

EXIT STATUS

0 Successful program execution.

1 Usage error.

2 **lexgrog** failed to parse one or more of its input files.

EXAMPLES

```
$ lexgrog man.1
man.1: "man – an interface to the on-line reference manuals"
$ lexgrog –fw man.1
man.1 (t): "man – an interface to the on-line reference manuals"
$ lexgrog –c whatis.cat1
whatis.cat1: "whatis – display manual page descriptions"
$ lexgrog broken.1
broken.1: parse failed
```

WHATIS PARSING

mandb (which uses the same code as **lexgrog**) parses the **NAME** section at the top of each manual page looking for names and descriptions of the features documented in each. While the parser is quite tolerant, as it has to cope with a number of different forms that have historically been used, it may sometimes fail to extract the required information.

When using the traditional *man* macro set, a correct **NAME** section looks something like this:

```
.SH NAME
foo \- program to do something
```

Some manual pagers require the '\–' to be exactly as shown; **mandb** is more tolerant, but for compatibility with other systems it is nevertheless a good idea to retain the backslash.

On the left-hand side, there may be several names, separated by commas. Names containing whitespace will be ignored to avoid pathological behaviour on certain ill-formed **NAME** sections. The text on the right-hand side is free-form, and may be spread over multiple lines. If several features with different descriptions are being documented in the same manual page, the following form is therefore used:

```
.SH NAME
foo, bar \- programs to do something
.br
baz \- program to do nothing
```

(A macro which starts a new paragraph, like `.PP`, may be used instead of the break macro `.br`.)

When using the BSD-derived *mdoc* macro set, a correct **NAME** section looks something like this:

```
.Sh NAME
.Nm foo
.Nd program to do something
```

There are several common reasons why whatis parsing fails. Sometimes authors of manual pages replace '.SH NAME' with '.SH MYPROGRAM', and then **mandb** cannot find the section from which to extract the information it needs. Sometimes authors include a NAME section, but place free-form text there rather than 'name \– description'. However, any syntax resembling the above should be accepted.

SEE ALSO

apropos(1), man(1), whatis(1), mandb(8)

NOTES

lexgrog attempts to parse files containing .so requests, but will only be able to do so correctly if the files are properly installed in a manual page hierarchy.

AUTHOR

The code used by **lexgrog** to scan man pages was written by:

Wilf. (G.Wilford@ee.surrey.ac.uk).
Fabrizio Polacco (fpolacco@debian.org).
Colin Watson (cjwatson@debian.org).

Colin Watson wrote the current incarnation of the command-line front-end, as well as this man page.