NAME

lgrind – grind nice program listings using LaTeX

NOTE

This man page is not yet much outdated, but might be soon except somebody asks me to work on it. Consider the LaTeX docs the real docs.

SYNOPSIS

lgrind [**-e**] [**-i**] [**-**] [**-n**] [**-c**] [**-t** <*width*>] [**-h** <*header*>] [**-d** <*description file*>] [**-l**<*language*>] [**-s**] <*name*> ...

DESCRIPTION

LGrind formats program sources in a nice style using LaTeX(1). Comments are placed in roman, keywords in bold face, variables in italics, and strings in typewriter font. Source file line numbers appear in the right margin (every 10 lines).

LGrind processes its input file(s) and writes the result to standard output. This output can be saved for later editting, inclusion in a larger document, etc.

The options are:

–e process a LaTeX file for embedded code.

–i process source code file for inclusion in a LaTeX document.

– take input from standard input.

–n don't boldface keywords.

–c don't treat @, etc. specially in comments.

–t change tab width (default 8).

–h specifies text to go into the header.

–d specifies the language definitions file (default is **texmf-dist/tex/latex/lgrind/lgrindef**).

–d! same as above, but write patched executable.

–l specifies the language to use.

–s shows a list of currently known languages.

If LGrind is called without parameters, a help screen will be shown. If neither **–e** nor **–i** are specified, a complete LaTeX file is produced. When no language is specified, LGrind tries to find out the language used itself; C is used when this fails.

USAGE

For example, to include a C file named **foo.c** into your LaTeX document, first give the command:

```
lgrind -i -lc foo.c > foo.tex
```

This will generate `foo.tex`, which will have the pretty-printed version of `foo.c` with a lot of LaTeX commands.

Then include `lgrind.sty` as you include any other style, namely with the `\usepackage{lgrind}` line at the beginning of your LaTeX document. Having done this, within the document you can include `foo.tex` using one of the following commands:

`\lgrindfile{foo.tex}`
: which will simply include the file at that point of text, and will draw horizontal lines before and after the listing.

`\lagrind[htbp]{foo.tex}{caption}{label}`
: which will put the listing also within a figure environment, using the float options, caption and label you gave.

To produce a standalone LaTeX file from, say, a Yacc file:

```
lgrind -ly bary.y > bary.tex
```

This uses Piet van Oostrum's `fancyhdr.sty` to make the headers and footers.

For a more detailed explanation of these commands, refer to `texmf-dist/doc/latex/lgrind.pdf`.

EMBEDDED PROGRAMS WITHIN A LaTeX FILE

(From Jerry Leichter's notes.)

Within the text of your LaTeX file, you mark groups of lines as either text- or display-style program code:

Text style: 1 l. The expression %(a + 3 %) produces 10. prints something like: "The expression *a* + 3 produces 10." (with "a + 3" set as a program.)

The same effect can be achieved with inline @'s. 1 l. The expression @a + 3@ produces 10.

Display style: 1 l. The statement %[a += 3; %] is an example of an incrementing operator. prints something like: 1 l. The statement a += 3; is an example of an incrementing operator.

Important rules:

`%` and the following character must be the first two characters on the line to be recognized.

Put *nothing* on the line after the `%` and the key character. If you do that, LGrind will provide a default environment that will produce an `\hbox` for `%( )%`, and a `\vbox` for `%[ - %]`. If you put stuff on the line, LGrind assumes you want to control the format completely. Doing this requires understanding *exactly* what the code LGrind produces is doing. (Sometimes I'm not sure I do!)

`%)` and `%]` are, if I remember right, simply ignored outside of a code group, but any extra `%(` or `%[` produces a warning, so a missing `%)` or `%]` is usually caught.

You can insert your own code by using a line starting with `%=` in the program text. Whatever you enter after that is left in the output, exactly as you typed it. It will be executed in a strange environment, so doing anything fancy is very tricky. A macro, `\Line`, is provided to help you do simple things. For example, tab (/); l. %[%=\Line{_____\vdots}
a = 1; %] produces: tab (/); l.
.
.
.
a = 1;

(Within the program text, _ is active and expands to a fixed-width space. A whole bunch of macros are also defined. If you understand how LGrind sets lines up, you can replace the 8 _'s with a call to `\Tab` — but I'll let you hang yourself on that one.)

The output of LGrind always contains exactly one output line for each input line. Hence, you can look up line numbers in TeX error messages in your original file, rather than in the lgrind'ed (lground?) file. (Of course, if the problem is in the LGrind output....)

Many things are controllable by re-defining various macros. You can change what fonts LGrind will use for various kinds of things, how much it indents the output, whether it adds line numbers, and if so at what interval it prints them and whether it sticks them on the left or right, and so on. This stuff is all described in `lgrind.dvi`, though probably not very well. The default settings produce output that looks reasonable to me, though I can't say I'm ecstatic about it. Doing a *really* good job would require defining some special fonts.

FILES

`bin/lgrind`
Executable

`texmf-dist/doc/lgrind/lgrind.pdf`
: Documentation

`texmf-dist/tex/latex/lgrind/lgrind.sty`
: LaTeX style file

`texmf-dist/tex/latex/lgrind/lgrindef`
: Language descriptions

AUTHORS

Van Jacobson, Lawrence Berkeley Laboratory (based on "vgrind" by Dave Presotto & William Joy of UC Berkeley), wrote it for TeX.

Jerry Leichter of Yale University modified it for LaTeX.

George V. Reilly of Brown University changed the name to lgrind, fixed up the man page, and added the program-text-within-comments and @-within-LaTeX features.

Michael Piefel of Humboldt-University Berlin adapted it to LaTeX2e and wrote decent documentation.

SEE ALSO

latex(1), **tex**(1), **vgrind**(1), **lgrindef**(5)