ChkTEX v1.5

Jens T. Berger Thielemann

October 8, 1996

1 Introduction

This program has been written in frustration because some constructs in LATEX are sometimes non-intuitive, and easy to forget. It is *not* a replacement for the built-in checker in LATEX; however it catches some typographic errors LATEX oversees. In other words, it is Lint for LATEX. Filters are also provided for checking the LATEX parts of CWEB documents.

It is written in ANSI C and compiles silently under GCC using "`-Wall -ansi -pedantic`" and almost all warning flags. This means that you can compile & use the program on your favorite machine. Full source included.

The program also supports output formats suitable for further processing by editors or other programs, making errors easy to cycle through. Software is provided for beautifully interfacing against the AUC-TEX Emacs mode, CygnusED, GoldEd and various Amiga message browsers.

The program itself does not have any machine requirements; Amiga, MSDOS and OS/2 binaries are available. Of course, you'll need ARexx and ScMsg or VBrowse to benefit from the ARexx scripts.

2 Features

ChkTEX begins to get quite a few bells & whistles now. However, you should be aware of that in most cases, all this is transparent to the user. As you will see, ChkTEX offers the ability to adapt to many environments and configurations.

2.1 New features

Modifications and additions since v1.4:

2.1.1 Program

Modifications to the executable itself:

- As usual, a few more warnings:
  - No space or similar in front/after parenthesis.
  - Demands a consistent quote style.
  - Double spaces in input which will not be rendered as one.
  - Punctuation malplaced regarding to math mode.
  - Warns about TEX primitives.
  - Space in front of footnotes.
  - Bogus `\left` and `\right` commands.

- The abbreviation recognizer has (for the last time?) been redesigned. We now produce far less false warnings, catch more cases and do all this faster than before. Seems like a win.

  Done much of the same with the italic correction detection part, too...

- Some bugs have been silently fixed. Hot spots in the program have been optimized; in certain cases this in fact doubles the speed!

  Along with this goes more code elegance and utilization of macro processing and the C language. Take a look at "`Resource.[ch]`".

- It's possible to specify separate output-formats depending on whether you are sending the output to a file/pipe or to a terminal.

- Column positions are finally correct; we now expand tabs correctly.

- ChkTEX will now recursively search for `\input`'ed files, both in the document and on the commandline. See the "`chktexrc`" file for more info.

- The debug switch is now more intelligent; if you wish to hack a bit on ChkTEX for yourself, it is possible to produce selective debugging output. The feature can also be disabled altogether.

- MS-DOS and OS/2 version of the program is now more flexible and well-behaved, thanks to Gerd Böhm.

- You may now say "`-wall`" to make all messages warnings, and turn them on.

- Uses termcap on UNIX boxes; this should ensure that "`-v2`" (or more precisely: "`%i`" and "`%I`") works regardless to what terminal you are using.

2.1.2 Resource file

New concepts introduced in the setup file:

- You may now specify both case-sensitive and case-insensitive user patterns in the "`chktexrc`" file. In addition; it is now possible to reset/clear lists.

- It is possible to specify how many arguments (optional/required) "`WIPEARG`" should wipe; it behaves also somewhat more intelligent when the arguments stretch over multiple lines.

- Global files will be read *in addition* to local ones. The searching order has also been reversed in order to make this more intelligent.

2.1.3 Other

Various other stuff I've done to the product:

- The documentation has been polished and should now be easier to use in practical situations.

- "`check`" target in "`Makefile`", so you can check that the installation succeeded. In fact, the "`Makefile`" has been enhanced in several other ways too, amongst other it is now GNU conforming.

- "`deweb`" is now documented; you may say "`man deweb`" to get a few words of advice. The support script (`chkweb`) does now behave as the remaining package (accepting stdin input and flags).

- I've written an Emacs hack which magically adds ChkTEX to the list of AUC-TEX commands; thus making the use of the program even more trivial.

  For those of you who don't wish to mess with Emacs, I've included a trivial lacheck ↔ ChkTEX interface.

  This means that you now can use ChkTEX just as easily as lacheck when you're running AUC-TEX.

- Added an ARexx script which lets ChkTEX talk to VBrowse, the message browser of Volker Barthelmann's freely distributable ANSI C compiler. The browser itself is available on Aminet as "`dev/c/vbcc.lha`".

2.2 Feature list

- Supports over 40 warnings. Warnings include:

  - Commands terminated with space. Ignores "`\tt`", etc.
  - Space in front of references instead of "`~`".
  - Forgetting to group parenthesis characters when sub-/superscripting.
  - Italic correction ("`\/`") mistakes (double, missing, unnecessary).
  - Parenthesis and environment matching.
  - Ellipsis detection; also checks whether to use "`\dots`", "`\cdots`" or "`\ldots`".
  - Enforcement of normal space after abbreviation. Detects most abbreviations automagically.
  - Enforcement of end-of-sentence space when the last sentence ended with capital letter.
  - Math-mode on/off detection.
  - Quote checking, both wrong types ("`"`") and wrong direction.
  - Recommends splitting three quotes in a row.
  - Searching for user patterns.
  - Displays comments.
  - Space in front of "`\label`" and similar commands.
  - Use of "`x`" instead of "`$\times$`" between numbers.
  - Multiple spaces in input which will be rendered as one space (or multiple spaces, where that is undesirable).
  - Warns about text which may be ignored.
  - Mathematical operators typeset as variables.
  - No space in front of/after parenthesis.
  - Demands a consistent quote style.
  - Punctuation inside inner math mode/outside display math mode.
  - Use of TEX primitives where LATEX equivalents are available.
  - Space in front of footnotes.
  - Bogus characters following commands.

- Fully customizable. Intelligent resource format makes it possible to make ChkTEX respect your LATEX setup. Even command-line options may be specified globally in the "`chktexrc`" file.

- Supports "`\input`" command; both TEX and LATEX version. Actually includes the files. "`TEXINPUTS`"-equivalent search path.

- Intelligent warning/error handling. The user may promote/mute warnings to suit his preferences. You may also mute warnings in the header of a file; thus killing much unwanted garbage.
- Scripts included for checking CWEB files written in LaTeX. (Requires perl v5).
- Supports both LaTeX 2.09 and LaTeX 2_ε.
- Flexible output handling. Has some predefined formats and lets the user specify his own format. Uses a "`printf()`" similar syntax. "`lacheck`" compatible mode included for interfacing with the AUC-TeX Emacs mode.
- ARexx scripts for interfacing with ScMsg and VBrowse are included. Special scripts included for CygnusED/GoldED, for binding to hotkey.
- Amiga Workbench support. Parameters may be passed by shift-clicking the LaTeX files and setting the remaining options in the tooltypes.
- Wildcard matching (Amiga only). Matches file patterns internally, thus saving a lot of work. This is, however, platform-specific code — on UNIX boxes this is done by the shell.
- Written in ANSI C. "`configure`" script included for easy setup and installation on UNIX systems.

Still, it is important to realize that the output from ChkTeX is only intended as a *guide* to fixing faults. However, it is by no means always correct. This means that correct LaTeX code may produce errors in ChkTeX, and vice versa: Incorrect LaTeX code may pass silently through.

3 Legal stuff

ChkTeX, documentation, installations scripts, CWEB filters and other materials provided are copyright © 1995–96 Jens T. Berger Thielemann, unless explicitly stated otherwise.

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to:

The Free Software Foundation, Inc.
675 Mass Ave
Cambridge
MA 02139
USA

3.1 Registration

'ChkTEX' is made available under the concept of Gift-Ware, which is a variant of Share-Ware. Share-Ware software authors often release 'crippled' versions of their products, i.e. these programs do not support the same functionality as the registered versions you get when sending monetary contributions to the authors. It all comes down to 'pay for the software you are using' in Share-Ware terms. With Gift-Ware registration it is different, you are not required to contribute money, but a gift will do. With 'ChkTEX' you always get a fully functional program, there is no 'crippled' test release which you can try for a limited time and then have to pay for in order to receive the working registered version.

It may seem as if 'ChkTEX' was free, but this is not the case. Although there is no need to pay the author in order to get a fully functional version of the program you should consider making a contribution. You don't need to feel guilty if you cannot or do not want to give me something in return for the work I have put into 'ChkTEX'. Show me that it was worth spending so much time listening to users, updating, rewriting and enhancing this program. Your contributions will provide the motivation for me to keep developing the program.

Send your contribution to the following address:

Jens Berger
Spektrumvn. 4
N-0666 Oslo
Norway

4 Installation

A few words on installation on various platforms:

Amiga: Use the supplied Installer script.

MS-DOS & OS/2: Copy the binary ("`ChkTeX.exe`") to somewhere in your path, and place the "`chktexrc`" file in the same directory.

If you're running Emacs, you may wish to copy the file "`chktex.el`" to your elisp source directory, and add the following line to your "`.emacs`" file:

```
(require 'chktex)
```

UNIX: Type "`configure`", "`make`" and finally "`make install`". To make sure everything proceeded correctly, type "`make check`". If you don't have superuser privileges and thus access to the default system areas, you should type "`configure -help`" to help you set up correct paths.

If you haven't installed any software like this before, that is distributed in source form, here are some guidelines to help you install it locally at your account. Please note that a mail to the system administrator may be less work for you. :-)

We assume that you have put the archive ("`chktex.tar.gz`") in a subdir of yours, with path "`~/tmp`". We further assume that your shell is "`csh`" or "`tcsh`". Do the following:

1. First of all, unpack the archive contents.

```
> cd ~/tmp
> gunzip chktex.tar.gz
> tar xf chktex.tar
```

2. Now, we can configure the program. There are some configuration options you should know about:

   "`-enable-emacs-hack`": Install a small file which adds ChkTEX to the command menu of AUC-TEX. This is to be considered as a hack, and may break in future versions of AUC-TEX. It works fine under AUC-TEX v2.9, though.
   This option needs the path of your elisp directory as argument, e.g. "`-enable-emacs-hack=/home/myself/elisp`" or similar.
   You'll also have to add the following line to your "`.emacs`" file:

   ```
   (require 'chktex)
   ```

   You can now access ChkTEX from the "`Command`" menu in AUC-TEX. To cycle through the messages, type `C-x '`.

   "`-enable-lacheck-replace`" This enables a quick hack for making the AUC-TEX Emacs mode use ChkTEX instead of lacheck. This is done by installing a stub script which "overrides" the original lacheck executable.
   While more stable than the previous solution, this is also significantly less elegant — in computing terms, this is the "brute force" approach.

   "`-enable-debug-info`" ChkTEX has an ability to spit out various diagnostic messages using the "`-d`" flag. This behaviour is on by default. By adding the flag "`-disable-debug-info`" to the commandline, this will not be compiled in.
   This may be useful if you're running short of disk space (the time savings are neglible).

   If you are installing the program on your local account, use the following command:

   ```
   > configure --prefix ~/
   ```

   Add eventual extra flags as specified above. This command will generate a significant amount of output, this can usually be ignored.

3. Finally, we can just build the program and install it.

   ```
   > make
   > make install
   ```

4. Finished! The program is now installed and ready to use. You may now tell other people to put your bindir in their path in order to benefit from your work. All that remains is to make the shell aware of your installation.

   ```
   > rehash
   ```

   To make the remaining parts of your system aware of this, you'll have to log out and re-log in, I'm afraid. However, you should delay this until you've completed this installation procedure.

5. If you wish to make sure that everything is OK (you ought to), you may now ask ChkTEX to do a self-test:

   ```
   > make check
   ```

Other platforms: First of all, you have to copy the "`config.h.in`" file to a file named "`config.h`". Then, edit it to reflect your system. Do the same with "`OpSys.h`" (this file has been reduced significantly). If you wish, you may define "`DATADIR`" to the path you want the global resource file to be put.

Now, I would suggest that you take a peak at the "`OpSys.c`" file, and edit it appropiately, for more comfort. This should not be necessary, though, at least not the first time.

Finally, you may now compile and link all `.c` files. Define "`HAVE_CONFIG_H`" to 1 (on the command-line, for instance). If the "`config.h`" you wish to use has another name, define "`CONFIG_H_NAME`" to that (in that case, don't define "`HAVE_CONFIG_H`").

Put the directory path of the "`chktexrc`" file in a environment variable named "`CHKTEXRC`". The files "`deweb.in`" and "`chkweb`" should be moved to a directory in your path. These files may need further setup, as they haven't got the location of perl initialized.

If your compiler/the compiled program complains (or crashes!), you may try the hints listed below. Please note that it only makes sense to try these hints if your compiler fails to produce a working program.

1. Increase the preprocessor buffers and line buffers. The ChkTEX sources define macros sized 3–4k (expanding to about the same), and passes arguments sized about 1k.
2. Use the magic switch which lets us use large "`switch(...){...}`" statements; some of these statements have about 120 "`case`" entries.
3. The sources require that at least the first 12 of each identifier is significant.

Note: You *must* install the new "`chktexrc`" file; ChkTEX will fail to function unless!

After doing this, you may enhance ChkTEX' behaviour by reading/editing the "`chktexrc`" file.

5 Usage

5.1 ChkTEX

Amiga v2.04+ users may wish to note that you may pass the files that ChkTEX is to process by shift-clicking them. In addition, you may specify the arguments as tooltypes to the main program icon, using the Info menu option. For instance, if you wish to make the 'fancy' output the default when starting from Workbench, enter the following as a tooltype:

```
Verbosity=2
```

We use the WB2Argv package (also developed by me, available on Aminet as "`dev/c/WB2Argv.lha`"), and will fold case until the "`=`" character, in tradition with the case-insensitivity of Amiga OS. In all other respects (and platforms), the options listed below are case-sensitive.

5.1.1 Synopsis

A UNIX-compliant template format follows:

```
chktex  [-hiqrW] [-v[0-...]] [-l <rcfile>] [-[wemn] <[1-42]|all>]
        [-d[0-...]] [-p <pseudoname>] [-o <outputfile>]
        [-[btxgI][0|1]] file1 file2 ...
```

5.1.2 Options

These are the options ChkTEX currently accepts. Please note that single-lettered options requiring a numerical or no argument may be concatenated. E.g. saying "`-v0qb0w2`" is the same as saying "`-v0 -q -b0 -w2`", except for being less to type.

Enough general talk; here's a rather detailed description of all options:

Misc. options: General options which aren't related to some specific subpart of ChkTEX.

`-h [-help]` Gives you a command summary.

`-i [-license]` Shows distribution information.

`-l [-localrc]` Reads a resource-file formatted as the global resource-file "`chktexrc`", in addition to the global resource-file. This option needs the name of the resource-file as a parameter. See also `-g`.

`-r [-reset]` This will reset all settings to their defaults. This may be useful if you use the `CMDLINE` directive in your "`chktexrc`" file, and wish to do something unusual.

`-d [-debug]` Needs a numeric argument; a bitmask telling what to output. The values below may be added in order to output multiple debugging info.

Value	Dumps...
1	All warnings available and their current status.
2	Statistics for all lists in the resource file.
4	The contents of all lists in the resource file.
8	Misc. other status information.
16	Run-time info (note that this isn't widely used).

The info is produced after all switches and resource files have been processed.

It is possible to install versions of ChkTEX that ignore this flag; this means that it is not certain that this flag works.

`-W [-version]` Displays version information, and exits.

Muting warning messages: Controls whether and in what form error messages will appear. Usually they accept a specific warning number (e.g. "`-w2`"), but you may also say "`all`" (e.g. "`-wall`") which does the operation on *all* warnings.

`-w [-warnon]` Makes the message number passed as parameter a warning and turns it on.

`-e [-erroron]` Makes the message number passed as parameter an error and turns it on.

`-m [-msgon]` Makes the message number passed as parameter a message and turns it on. Messages are not counted.

`-n [-nowarn]` Turns the warning/error number passed as a parameter off.

Output control flags: Determines the appearance and destination of the error reports.

`-q [-quiet]` Shuts up about copyright information.

`-o [-output]` Normally, all errors are piped to `stdout`. Using this option with a parameter, errors will be sent to the named file instead. Only information relative to the LaTeX file will be sent to that file. Memory problems and similar will as as always be sent to `stderr`. If a file

with the name given already exists, it will be renamed to "`foobar.bak`" ("`foobar.$cl`" under MS-DOS), "`foobar`" being the name of the file. See also "`-b`".

`-v [-verbosity]` Specifies how much and how you wish the error reports to be displayed. This is specified in the "`chktexrc`" file; we'll list the default values below. If you wish, you may thus edit the "`chktexrc`" file to add or modify new formats.

The default is mode 1 (that is, the *second* entry in the "`chktexrc`" file), using `-v` without any parameter will give you mode 2.

0 Will show the information in a way that should be suitable for further parsing by `awk`, `sed` or similar. The format is as follows:

```
File:Line:Column:Warning number:Warning message
```

The colons may be replaced with another string; use the `-s` switch for this.

As the program does not output all errors in quite order, this output format is also suitable for piping through "`sort`".

1 Shows the information in a way which is more comprehensible for humans, but which still doesn't need anything but a glass tty.

2 Shows the information in a fancy way, using escape codes and stuff. It is the indeed most readable of all modes; however, it needs proper set up of the "`ChkTeX.h`" at compilation time. UNIX boxes, however, will find the information automatically.

3 Shows the information suitable for parsing by Emacs; this is the same format as `lacheck` uses. More formally, it is the following:

```
"File", line Line: Warning message
```

To utilize this, type `M-x compile RET`. Delete whatever is written in the minibuffer, and type `chktex -v3 texfile.tex`, and you should be able to browse through the error messages. Use `C-x ‘` to parse the messages.

4 More or less the same as `-v3`, but also includes information on where the error actually was found. Takes somewhat longer time to parse, but much more informative in use.

`-f [-format]` Specifies the format of the output. This is done using a format similar to "`printf()`", where we support the specifiers listed below.

Code	Description
`%b`	String to print **b**etween fields (from `-s` option).
`%c`	**C**olumn position of error.
`%d`	Length of error (**d**igit).
`%f`	Current **f**ilename.
`%i`	Turn on **i**nverse printing mode.
`%I`	Turn off **i**nverse printing mode.
`%k`	**k**ind of error (warning, error, message).
`%l`	**l**ine number of error.
`%m`	Warning **m**essage.
`%n`	Warning **n**umber.
`%u`	An **u**nderlining line (like the one which appears when using "`-v1`").
`%r`	Part of line in front of error ('S' − 1).
`%s`	Part of line which contains error (**s**tring).
`%t`	Part of line after error ('S' + 1).

Other characters will be passed literally; thus you can say "`%%`" to achieve a single percent sign in the output. Please note that we may introduce other specifiers in the future, so don't abuse this feature for other characters.

Also, note that we do *not* support field lengths (yet). This may come in the future, if I get the time...

The `-v` command is implemented by indexing into the "`chktexrc`" file; read that for seeing how each format is implemented. If you find yourself using a particular format often by using the -f switch, consider putting it in the "`chktexrc`" file instead.

`-V [-pipeverb]` Which entry we'll use in the "`chktexrc`" file whenever `stdout` isn't a terminal.

The default is to use the same mode as specified with the `-v` switch; using `-V` without any parameter will give you mode 1.

This switch was implemented because GNU less has problems with the escape codes ChkTEX uses for displaying inverse text. Under UNIX, there's another way around, though, which is slightly more elegant. Add the following line to your "`.envir`" file:

```
setenv LESS -r
```

`-p [-pseudoname]` With this switch, you can provide the filename which will be used when we report the errors. This may be useful in scripts, especially when doing pipes. It is in other words similar to C's `#line` directive.

We will only assume this name for the uppermost file; files that this one in turn `\input` are presented under their original names. This seems most logical to me.

`-s [-splitchar]` String to use instead of the colons when doing `-v0`; e.g. this string will be output between the fields.

Boolean switches: Common for all of these are that they take an optional parameter. If it is `0`, the feature will be disabled, if it is `1`, it will be enabled. All these features are on by default; and are toggled if you don't give any parameter.

`-b [-backup]` If you use the `-o` switch, and the named outputfile exists, it will be renamed to `filename.bak`.

`-I [-inputfiles]` Execute `\input` statements; e.g. include the file in the input. Our input parsing does of course nest; we use an input-stack to keep track of this.

`-H [-headererr]` Show errors found in front of the `\begin{document}` line. Some people keep *lots* of pure TEX code there, which errors can't be detected reliably (in other words, we will in most cases just produce a lot of garbage).

`-g [-globalrc]` Read in the global resource file. This switch may be useful together with the `-l` option.

`-t [-tictoc]` Display a twirling baton, to show that we're working. `-v0` does an `-t0`, too, as it assumes that the user then uses the program non-interactively. This is now a no-op.

`-x [-wipeverb]` Ignore the "`\verb`" command found within the LATEX file and its argument is completely by the checking routines. This is done by

simply overwriting them. If you somehow don't like that (for instance, you would like to count brackets inside those commands, too), use this switch.

If you don't specify any input LaTeX-files on the commandline, we'll read from `stdin`. To abort `stdin` input, press the following keycombinations:

Machine	Key-combination
Amiga	`Ctrl` + `\`
UNIX	`Ctrl` + `D`
MS-DOS	`Ctrl` + `Z`, followed by return.

By default, we're using the 1994 version of GNU's "`getopt()`" routine. For those of you who have only used the Amiga's "`ReadArgs()`", here are some quick instructions:

- Options may be given in any order; the names of the LaTeX-files do not have to be the last arguments. This behaviour may be turned off by creating an environment variable named "`POSIXLY_CORRECT`".
- The special argument "`-`" forces an end of option-scanning.
- Long-named options begin with "`-`" instead of "`-`". Their names may be abbreviated as long as the abbreviation is unique or is an exact match for some defined option. If they have an argument, it follows the option name in the argument, separated from the option name by a "`=`", or else the in next argument.

5.1.3 The "`chktexrc`" file

You should also take a look at the "`chktexrc`" file. As the file is self-documenting; you should be able to get the meaning of each keyword by simply reading the file. The method for *finding it*, however, has grown rather complex. An outline is given below.

If ChkTeX finds multiple files when searching, each and everyone will be read in the order specified below. The "`Keyword =ă{ item item ...}`" may thus be necessary to reset previously defined lists.

In this list, "`$foo`" is assumed to be the environment variable "`foo`":

1. First, we'll take a look at the directory which was specified as "`DATADIR`" during compilation (this does not happen on Amiga machines). On UNIX boxes, this usually evaluates to "`/usr/local/share/chktexrc`" or similar, under MS-DOS it is set to "`\emtex\data\chktexrc`".
2. Look in the following system directories:

Machine	Directory
2.04+ Amiga	`ENV:.chktexrc`
1.3 Amiga	`S:.chktexrc`
UNIX	"`$HOME/.chktexrc`" or "`$LOGDIR/.chktexrc`"
MSDOS	Program installation path

3. Look for it in the directory pointed to by an environment variable, as specified in the table below:

Machine	Directory
Amiga	"`$CHKTEXRC/.chktexrc`"
UNIX	"`$CHKTEXRC/.chktexrc`"
MSDOS	"`$CHKTEXRC\chktexrc`", "`$CHKTEX_HOME\chktexrc`" or "`$EMTEXDIR\data\chktexrc`"

4. Look for it in the current directory. On UNIX and Amiga boxes, we expect the filename to be "`.chktexrc`"; on other machines "`chktexrc`".

If you for some reason wish to undo what the previous files may have done, you may say "`CmdLine { -g0 -r }`" somewhere in the "`chktexrc`" file; this will reset all previous settings.

5.1.4 Hints

I've tried to collect some advice that may be useful — if you have a favourite hint, feel free to send it to me!

- If you use "`german.sty`"; it may be wise to put "`-n18`" in the "`CmdLine`" entry in the "`chktexrc`" file. This will probably reduce the amount of false warnings significantly.
- Put "`-v`" in the "`CmdLine`" entry of the "`chktexrc`" file; this makes the fancy printing the default.
- If you're working on a large project, it may pay off to make a local resource file which is included in addition to the global one. In this one, add the necessary info to reduce the amount of false warnings — these usually don't do anything but hide the real warnings.
- Create a total ignore environment, which ChkTEX will ignore completely. In here, you can place all that code which outsmarts ChkTEX completely. For instance, add the following lines at the top of your LATEX file:

  ```
  % ChkTeX will ignore material within this environment
  \newenvironment{ignore}{}{}
  ```

  In addition, you should add the item "`ignore`" to the "`VerbEnvir`" entry in the "`chktexrc`" file.

5.1.5 Bugs

No fatal ones, I think, but the program currently has some problems when a LATEX command/parameter stretch over a two lines — some extra spaces may be inserted into the input. I regard the program as fairly well tested; using the SAS/C "`cover`" utility I was able to make sure that approximately 95% of the code has actually been run successfully in the final version. This does indeed leave some lines; most of these are procedure terminating brackets or "can't happen" lines, though.

We've got some problems when isolating the arguments of a command. Although improved, it will certainly fail in certain cases; ChkTEX can for instance not handle arguments stretching over two lines. This also means that "`WIPEARG`" entries in the "`chktexrc`" file will only have the first half of their argument wiped if the argument stretches over two lines. We will, however, take care not to wipe parenthesis in such cases, in order to avoid false warnings.

Before submitting a bug report, please first see whether the problem can be solved by editing the "`chktexrc`" file appropiately.

5.2 ChkWEB

This shell script is provided for checking CWEB files. The template is as follows:

```
chkweb [options] file1 file2 ...
```

As you may see from the script, it is only a trivial interface towards `deweb` and ChkTEX. It does not support any individual options on the command line — all options found will be passed onto ChkTEX. If "`-`" or a filename is found, the remaining parameters will be ignored. The only real intelligence it features is that it will try to append `.w` to filenames it can't find.

If no filenames are given, we will read from `stdin`.

5.3 DeWEB

This program strips away C code and CWEB commands from CWEB sources. It is called with the following synopsis:

```
deweb file1 file2 ...
```

`deweb` filters away all C & CWEB commands from a CWEB source code. This leaves only the LATEX code. This stripped code, in turn, may then be passed to a suitable syntax checker for LATEX, like ChkTEX and `lacheck`, or spell-checkers like `ispell`.

When `deweb` strips away the C code from your CWEB source, it tries to preserve line breaks. This means that the error reports from *your favorite tool* will be correct regarding to line numbers. In most cases, the column position will also be correct. This significantly simplifies finding the errors in the LATEX source (in contrast to the output from `cweave`, which output is truly difficult to figure anything out from).

`deweb` accepts a list of filenames on the argument line, and will send its output to `stdout`. If no filenames are given, it will read from stdin, acting as a filter. No options are currently accepted.

Macho users may try to pipe the output from `deweb` directly into LATEX, theoretically, this should work. This would ease the debugging of the LATEX code significantly, as when LATEX complains about wrong syntax, you'll be able to find the erroneous line much more easily. Don't expect that the output looks very much like the final one, though.

`deweb` should now understand all correct `CWEB` opcodes. If it complains about not understanding a correct opcode, please inform the author.

5.3.1 Bugs

`deweb` will not even *compile* under Perl versions before perl v5. Unfortunately, this means that we can't even tell the user why we failed; Perl will just complain about not being able to compile the regexps.

6 Explanation of error messages

Below is a description of all error-messages ChkTEX outputs. Error messages set in *italic type* are turned off by default. Where margin paragraphs are listed in the text, they refer to the keyword in the "`chktexrc`" file which controls the discussed warning.

Warning 1: `Command terminated with space.` `Silent`

> You tried to terminate a command with a blank space. Usually, this is an error as these are ignored by LATEX. In most cases, you would like to have a real space there.

```
\LaTeXis a typesetter.
```
LaTeXis a typesetter.

```
\LaTeX\ is a typesetter.
```
LaTeX is a typesetter.

Warning 2: `Non-breaking space ('~') should have been used.` `Linker`

When reading a document, it is not very pretty when references are split across lines. If you use the ~ character, LaTeX will assign a very high penalty for splitting a line at that point. ChkTeX issues this warning if you have forgot to do this.

```
Please refer to figure_\ref{foo}.
```
Please refer to figure 11.

```
Please refer to figure~\ref{foo}.
```
Please refer to figure 11.

Warning 3: `You should enclose the previous parenthesis with '{}'.`

This is a warning which you may ignore, but for maximum aestethic pleasure, you should enclose your bracket characters with '{}'s.

```
$_[(ab)^{-1}]_\^{-2}$
```
$[(ab)^{-1}]^{-2}$

```
${[{(ab)}^{-1}]}^{-2}$
```
$[(ab)^{-1}]^{-2}$

Warning 4: `Italic correction ('\/') found in non-italic buffer.` `Italic` `ItalCmd` `NonItalic`

If you try to use the \/ command when ChkTeX believes that the buffer is not outputted as italic, you'll get this warning.

```
This is an\/ example
```
This is an example.

```
This is an example.
```
This is an example.

Warning 5: `Italic correction ('\/') found more than once.` `Italic` `ItalCmd` `NonItalic`

If the buffer is italic, and you try to use the \/ command more than once, you'll get this warning.

```
This {\it example\/\/} is not amusing.
```
This *example* is not amusing.

```
This {\it example\/} is not amusing.
```
This *example* is not amusing.

Warning 6: `No italic correction ('\/') found.`

`Italic`
`ItalCmd`
`NonItalic`

You get this error if ChkTEX believes that you are switching from italic to non-italic, and you've forgot to use the `\/` command to insert that extra little spacing. If you use the "`em`" option, you may ignore this warning.

```
This {\it example_} is not amusing, either.
```
This *example* is not amusing, either.

```
This {\it example\/} is not amusing, either.
```
This *example* is not amusing, either.

Warning 7: `Accent command 'command' needs use of 'command'.`

`IJAccent`

If you're using accenting commands, 'i' and 'j' should lose their dots before they get accented. This is accomplished by using the `\i`, `\j`, `\imath` and `\jmath` command.

```
This is an example of use of accents: \'{i}.
```
This is an example of use of accents: í.

```
This is an example of use of accents: \'{\i}.
```
This is an example of use of accents: í.

Warning 8: `Wrong length of dash may have been used.`

`HyphDash`
`NumDash`
`WordDash`

This warning suggests that a wrong number of dashes may have been used. It does this by classifying the dash according to the the character in front and after the dashes.

If they are of the same type, ChkTEX will determine which keyword to use in the "`chktexrc`" file. If not, it will shut up and accept that it doesn't know.

Character type	**Keyword in "`chktexrc`" file**
Space	`WordDash`
Number	`NumDash`
Alphabetic character	`HyphDash`

This is more or less correct, according to my references. Hopefully this check can be even more improved (suggestions?).

```
It wasn't anything - just a 2--3 star-shots.
```
It wasn't anything - just a 2—3 star–shots.

```
It wasn't anything --- just a 2--3 star-shots
```
It wasn't anything — just a 2–3 star-shots.

Warning 9: `'%s' expected, found '%s'.`
Warning 10: `Solo '%s' found.`

You get this warning whenever brackets or environments don't match. ChkTEX expect to find matching brackets/environments in the same order as their opposites were found, and no closing delimiters which haven't been preceded by an opening one.

While bracket matching is not an explicit error, it is usually a sign that something is wrong.

Warning 11: `You should use '%s' to achieve an ellipsis.` CenterDots LowDots

Simply typing three "`.`" in a row will not give a perfect spacing withing the between the dots. The `\ldots` is much more suitable for this.

In math mode, you should also distinguish between `\cdots` and `\ldots`; take a look at the example below.

`Foo...bar. $1,...,3$. $1+...+3$. $1,\cdots,3$.`
Foo...bar. $1, ..., 3$. $1 + ... + 3$. $1, \cdots, 3$.

`Foo\dots bar. $1,\ldots,3$. $1+\cdots+3$. $1,\ldots,3$.`
Foo… bar. $1, \ldots, 3$. $1 + \cdots + 3$. $1, \ldots, 3$.

Warning 12: `Interword spacing ('\ ') should perhaps be used.` Abbrev

One of the specified abbreviations were found. Unless you have previously said `\frenchspacing`, you'll have incorrect spacing, which one should avoid if possible.

`This is an example, i.e._an demonstration.`
This is an example, i.e. an demonstration.

`This is an example, i.e.\ an demonstration.`
This is an example, i.e. an demonstration.

Warning 13: `Intersentence spacing ('\@') should perhaps be used.`

LaTeX' detection of whether a period ends a sentence or not, is only based upon the character in front of the period. If it's uppercase, it assumes that it does not end a sentence. While this may be correct in many cases, it may be incorrect in others. ChkTeX thus outputs this warning in every such case.

`I've seen an UFO!_Right over there!`
I've seen an UFO! Right over there!

`I've seen an UFO\@! Right over there!`
I've seen an UFO! Right over there!

Warning 14: `Could not find argument for command.`

ChkTeX will in some cases need the argument of a function to detect an error. As ChkTeX currently processes the LaTeX file on a line-by-line basis, it won't find the argument if the command which needed it was on the previous line. On the other hand, this *may* also be an error; you ought to check it to be safe.

`$\hat$`
This will give a LaTeX error…

`$\hat{a}$`
$\hat{a}$

Warning 15: `No match found for '%s'.`

This warning is triggered if we find a single, *opening* bracket or environment. While bracket matching is not an explicit error, it is usually a sign that something is wrong.

Warning 16: `Mathmode still on at end of LaTeX file.` `MathEnvir`

This error is triggered if you at some point have turned on mathmode, and ChkTEX couldn't see that you remembered to turn it off.

Warning 17: `Number of 'character' doesn't match the number of 'character'.`

Should be self-explanatory. ChkTEX didn't find the same number of an opening bracket as it found of a closing bracket.

Warning 18: `You should use either `` or '' as an alternative to '"'.`

Self-explanatory. Look in the example, and you'll understand why.

```
This is an "example"
```
This is an "example"

```
This is an ``example''
```
This is an "example"

Warning 19: `You should use "'" (ASCII 39) instead of "'" (ASCII 180).`

On some keyboards you might get the wrong quote. This quote looks, IMHO, *ugly* compared to the standard quotes, it doesn't even come out as a quote! Just see in the example.

```
``There's quotes and there's quotes''
```
"There´s quotes and there´s quotes´´"

```
``There's quotes and there's quotes''
```
"There's quotes and there's quotes"

Warning 20: `User-specified pattern found.` `Userwarn`

A substring you've specified using `USERWARN` in the "`chktexrc`" file, has been found.

Warning 21: *`This command might not be intended.`*

I implemented this because a friend of mine kept on making these mistakes. Easily done if you haven't gotten quite into the syntax of LATEX.

```
\LaTeX\ is an extension of \TeX\. Right?
```
LATEX is an extension of TEX´Right?

```
\LaTeX\ is an extension of \TeX. Right?
```
LATEX is an extension of TEX. Right?

Warning 22: *`Comment displayed.`*

ChkTEX dumps all comments it finds, which in some cases is useful. I usually keep all my notes in the comments, and like to review them before I ship the final version. For commenting out parts of the document, the "`comment`" environment is better suited.

Warning 23: `Either ''\,' or '\,'' will look better.`

This error is generated whenever you try to typeset three quotes in a row; this will not look pretty, and one of them should be separated from the rest.

```
```Hello', I heard him said'', she remembered.
```
"'Hello', I heard him said", she remembered.

```
``\,`Hello', I heard him said'', she remembered.
```
" 'Hello', I heard him said", she remembered.

Warning 24: `Delete this space to maintain correct pagereferences.` PostLink

This message, issued when a space is found in front of a `\index`, `\label` or similar command (can be set in the "`chktexrc`" file). Sometimes, this space may cause that the word and the index happens on separate pages, if a pagebreak happens just there.

You might also use this warning to warn you about spaces in front of footnotes; however, the warning text may not be entirely correct then.

```
Indexing text_\index{text} is fun!
Indexing text\index{text} is fun!
```

Warning 25: `You might wish to put this between a pair of '{}'`

This warning is given whenever ChkTEX finds a "^" or a "_" followed by either two or more numberic digits or two or more alphabetic characters. In most situations, this means that you've forgotten some {}'s.

```
$5\cdot10^10$
```
$5 \cdot 10^{1}0$

```
$5\cdot10^{10}$
```
$5 \cdot 10^{10}$

Warning 26: `You ought to remove spaces in front of punctuation.`

This warning is issued if ChkTEX finds space in front of an end-of-sentence character.

```
Do you understand_?
```
Do you understand ?

```
Do you understand?
```
Do you understand?
```

Warning 27: `Could not execute LaTeX command.`

Some LaTeX commands will be interpreted by ChkTeX; however, some of them are sensible to errors in the LaTeX source. Most notably, the `\input` command relies on that the input file exists...

Warning 28: `Don't use \/ in front of small punctuation.`

`Italic`
`ItalCmd`
`NonItalic`

Italic correction should generally *not* be used in front of small punctuation characters like '.' and ','; as it looks better when the preceding italic character leans "over" the punctum or comma.

```
It is just a {\it test\/}, don't think anything else.
```
It is just a *test*, don't think anything else.

```
It is just a {\it test}, don't think anything else.
```
It is just a *test*, don't think anything else.

Warning 29: `$\times$ may look prettier here.`

In ASCII environments, it is usual to use the 'x' character as an infix operator to denote a dimension. The mathemathical symbol $\times$ provided by the `$\times$` command is better suited for this.

```
The program opens a screen sized 640x200 pixels.
```
The program opens a screen sized 640x200 pixels.

```
The program opens a screen sized $640\times200$ pixels.
```
The program opens a screen sized 640×200 pixels.

Warning 30: *`Multiple spaces detected in output.`*

This warning, intended for the novice, will remind you that even if you *type* multiple spaces in your input, only a single space will come out. Some ways to come around this is listed below.

```
White           is a beautiful colour.
```
White is a beautiful colour.

```
White~~~~~{ }{ }{ }\ \ \ is a beautiful colour.
```
White is a beautiful colour.

Warning 31: `This text may be ignored.`

`VerbEnvir`

Certain implementations of the `verbatim` environment and derivations of that, ignore all text on the same line as `\end{verbatim}`. This will warn you about this.

Warning 32: `Use ` to begin quotation, not '.`

Warning 33: `Use ' to end quotation, not `.`

Warning 34: `Don't mix quotes.`

Proper quotations should start with a ` and end with a '; anything else isn't very pretty. Both these warnings are relative to this; look in the example below.

```
There are ``examples'' and there are ``examples".
```
There are ``examples" and there are "examples".

```
There are ``examples'' and there are ``examples''.
```
There are "examples" and there are "examples".

Warning 35: `You should perhaps use 'cmd' instead.` `MathRoman`

Most mathematical operators should be set as standard roman font, instead of the math italic LaTeX uses for variables. For many operators, LaTeX provides a pre-defined command which will typeset the operator correctly. Look below for an illustration of the point.

```
$sin^2 x + cos^2 x = 1$
```
$sin^2 x + cos^2 x = 1$

```
$\sin^2 x + \cos^2 x = 1$
```
$\sin^2 x + \cos^2 x = 1$

Warning 36: `You should put a space in front of/after parenthesis.`

Warning 37: `You should avoid spaces in front of/after parenthesis.`

Outside math mode, you should put a space in front of any group of opening parenthesis, and no spaces after. If you have several after each other, you should of course not put a space in between each; look in the example below. Likewise, there should not be spaces in front of closing parenthesis, but there should be at least one after.

```
This( an example( Nuff said )), illustrates( ``my'' )point.
```
This(an example(Nuff said)), illustrates("my")point.

```
This (an example (Nuff said)), illustrates (``my'') point.
```
This (an example (Nuff said)), illustrates ("my") point.

Warning 38: `You should not use punctuation in front of/after quotes.` `QuoteStyle`

For best looking documents, you should decide on how you wish to put quotes relative to punctuation. ChkTeX recognizes two styles; you may specify which you use in the "`chktexrc`" file. A description on each style follows:

Traditional: This style is the most visually pleasing. It always puts the punctuation *in front of* the quotes, which gives a continuous bottom line.

However, it may in certain cases be ambigious. Consider the following example from a fictious "`vi(1)`" tutorial (quote taken from the Jargon file):

```
Then delete a line from the file by typing ``dd.''
```
Then delete a line from the file by typing "dd."

That would be very bad — because the reader would be prone to type the string d-d-dot, and it happens that in "`vi(1)`" dot repeats the last command accepted. The net result would be to delete *two* lines! This problem is avoided using logical style, described below.

Logical: This style uses quotes as balanced delimiters like parentheses. While this is not the most visual pleasing, it is can't be misunderstood. The above sentence would then become:

```
Then delete a line from the file by typing ``dd''.
```

Then delete a line from the file by typing "dd".

Warning 39: `Double space found.`

This warning is triggered whenever ChkTEX finds a space in front of a hard space, or vice versa. This will be rendered as two spaces (which you usually don't wish).

```
For output codes, see table_~_\ref{foo}.
```

For output codes, see table 1.1.

```
For output codes, see table~\ref{foo}.
```

For output codes, see table 1.1.

Warning 40: `You should put punctuation outside inner/inside display math mode.` `MathEnvir`

As recommended in the TEXbook, you should try to put punctuation outside inner math mode, as this is gets formatted better.

Similarily, you should let any final punctuation in display math mode end up within it. Look at the following example, which was taken from the TEXbook:

```
for $x = a,b$, or $c$.
```

for $x = a, b$, or c.

```
for $x = a$, $b$, or $c$.
```

for $x = a$, b, or c.

Warning 41: *`You ought to not use primitive TeX in LaTeX code.`* `Primitives`

This warning is triggered whenever you use a raw TEX command which has been replaced by a LATEX equivalent. If you consider yourself a purist (or want to be sure your code works under LATEX3), you should use the LATEX equivalent.

Warning 42: `You should remove spaces in front of '%s'` `NotPreSpaced`

Some commands should not be prepended by a space character, for cosmetical reasons. This notes you of this whenever this has happened.

```
This is a footnote_\footnotemark[1] mark.
```

This is a footnote [1] mark.

```
This is a footnote\footnotemark[1] mark.
```

This is a footnote[1] mark.

Warning 43: `'%s' is normally not followed by '%c'.` `NoCharNext`

LATEX' error message when calling `\left \{` instead of `left \{` is unfortunately rather poor. This warning detects this and similar errors.

7 Future plans

In a somewhat prioritized sequence, this is what I'd like to put into the program — if I have the time.

- Do a final fix for maths mode. Currently, ChkTEX doesn't recognize embedded math mode (i.e. constructions like `$$a+b\text{for $a \leq 0$}$$`).
- Support for regular expressions as user patterns. I'll do it at once I get the GNU "`rx`" package up and running (it doesn't produce correct include files).
- De-linearize the checker. Currently, it works on a line-by-line basis, in most respects, at least. I hope to be able to remove this barrier; as this will reduce the amount of false warnings somewhat.
- Probably some more warnings/errors; just have to think them out first. Suggestions are appreciated — I've "stolen" most that similar programs provides, and am running out of ideas, really.

  It would also be nice to investigate the field of "globally" oriented warnings; i.e. warnings regarding the document as a whole. Currently, ChkTEX operates mainly on a local/"greedy" basis.

  If you have suggestions/ideas on this topic, they're certainly welcome, including references to literature.
- Fix a few more bugs. :-)

8 Notes

8.1 Wish to help?

As most other living creatures, I have only a limited amount of time. If you like ChkTEX and would like to help improving it, here's a few things I would like to receive. The following ideas are given:

- Does anyone have a LATEX → `troff` conversion program? It would be really nice if I could extract the relevant sections from this manual, and present them as a man page. I will not, however, convert this manual to TEXinfo in order to be able to do this; IMHO TEXinfo documents have far too limited typographic possibilities.

  This doesn't mean that I'm not willing to restructure the document at all. This manual already uses some kind of preprocessing in order to achieve HTML output via LATEX2`html`, I'm willing to do the same in order to produce `troff` output.
- Help me port the program! This is a prioritized one. It's no fun writing ANSI C when people haven't got a C compiler.

  Of course, I'll provide whatever help necessary to modify the sources to fit to the new platform. Take contact if you're interested. I will include your compiled binary in the distribution, and give you credit where appropiate.

  Just one request: If you have to modify the sources in order to make ChkTEX compile & work on the new platform, *please* enclose your changes in something like "`#ifdef __PLATFORM__...code...#endif`"! It makes life so much easier later, when we try to merge the two source trees.

- Reports on problems configuring and compiling ChkTEX on supported (and unsupported) systems are welcomed.
- Filters for other file formats. I do believe that there are several formats using LATEX for its formatting purposes, combining that with something else. If you can write a program or script which filters everything away but the LATEX code, it will surely be appreciated (and included). Look at the `deweb` script to see what I mean.
- Arexx interfaces for other editors are also welcomed; these should be rather fast to write. They should to the following:
  1. Get the filename of the active file.
  2. If possible, save the file to disk if there has been any changes.
  3. Call the program "`ChkTeX.rexx`" with the filename as the only parameter.
- If somebody out there actually possesses (and uses) GoldED, it would be nice if they checked whether the ARexx script included actually work. If not, please send me a fixed copy; perhaps also one which supports point 2 above, too. If it does work, then please tell me so, so I can remove this item.

  I don't have GoldEd in my possession; the script was just modelled after Juergen Zeschky's, (`<juergen@sokrates.nbg.de>`) PGP ↔ GoldED interface.
- If you update the "`chktexrc`" file in anyway that is not strictly local, I would appreciate to receive your updated version.
- Suggestions for new warnings are always welcomed. Both formal (i.e. regexps or similar) and non-formal (plain English) descriptions are welcomed.

Of course, people doing any of this will be mentioned in this document, and thus receive eternal glory and appreciation.

8.2 Caps and stuff

This program uses the "`getopt()`" routine, as supplied from GNU. The source included in this distribution has been modified slightly. To make the use of C2LOCAL easier, portions which were `#ifdef`'ed out, have now been commented out.

Where trademarks have been used, the author is aware of that they belong to someone, and has tried to stick to the original caps.

9 About the author

A quick summary of who I am and what I do:

I'm 21 years old, and live in Oslo, the capital of Norway. I'm currently studying maths and computer science at the University of Oslo; planning to get a degree within mathematical modeling, with a dash of physics and emphazing the computer part of the study. More precisely, in autumn'96 my studies consist of mathematical analysis, statistics & probability calculations plus studying the relationship between society and computers.

At home I now possess 4 computers, of which 1 is regular use: A vanilla Amiga 1200, expanded only by a HD. The others are a `80286` PC and an Amiga 500, both semi-out-of-order. The last one is a Commodore VIC-20, which for some peculiar reason never seems to be used. Plans are to get a Linux-capable PC, though.

Most of the time in front of these computers (including SGI Indy's and SPARC stations at our university) is spent on C and shell programming, plus some text-processing.

C and shell programming are not my only knowledge areas regarding computers, however. I write the following languages more or less: Perl, Motorola `68000` assembly code, ARexx, Simula, C++, LaTeX, HTML, AmigaGuide, Amos Basic and Installer LISP. Once I also mastered Commodore Basic V2, the "language" included with my VIC-20.

However, I also try to not to end up as a computer nerd. Thus, in addition to the compulsory (?) interest for computers, I am a scout. Still running into the woods, climbing the trees, falling down and climbing up once more, in other words. To be more specific, I am a now a troop leader for 'Ulven' scout-group; Norwegian Scouts Association. I am also a active rover in 'Vålerenga' scout-group.

Certainly a lot more to tell (I play the piano and like cross-country skiing, for instance); but I'll stop here before you fall asleep. . . :-)

10 Thanks

The author wishes to thank the following people (in alphabetical order):

Russ Bubley
`russ@scs.leeds.ac.uk`
He has been the main external beta-tester for this program, sending me loads and loads of understandable and reproducible bug reports. If you somehow think that ChkTeX is well-behaved and free from bugs, send warm thoughts to Russ. He has also provided ideas for enhanced checks and so forth.

In addition, he sent me a huge list of 238 common English abbreviations, for inclusion in the "`chktexrc`" file! Together with the enhanced abbreviation recognizer, I do now believe most abbreviations should be catched. . . :-)

Finally, he has also given me valuable hints for improving the program's outputting routine, and given me lots of suggestions for filtering unnecessary/false warnings away.

Gerd Böhm
`Gerd.Boehm@physik.uni-regensburg.de`
Improved and bug-fixed the MS-DOS port of ChkTeX v1.4, sending me ready-to-yank code patches. The original port didn't respect all the peculiarities of the MS-DOS file-system, unfortunately.

Lars Frellesen
`frelle@math-tech.dk`
Sent a few bug reports regarding the filtering of messages. He has also helped me to expand the "`SILENT`" keyword in the "`chktexrc`" file.

Wolfgang Fritsch
`fritsch@hmi.de`
Author of the OS/2 port, which he did using the emx compiler. Please direct questions regarding strictly to that port to him (I would like to receive a carbon copy, though).

Stefan Gerberding
`stefan@inferenzsysteme.informatik.th-darmstadt.de`
First one to report the Enforcer hit in v1.2 when using ChkTeX as a pipe. Also came with suggestions to make ChkTeX more easily compile on early gcc compilers.

He has also kept on beta-testing later versions of ChkTEX, giving me bug-reports and enhancements requests.

Kasper B. Graversen
`kbg2001@internet.dk`
Lots of creative suggestions and improvements. Several of the warnings implemented were based on his ideas. In addition, he has given advice for improving the existing warnings.

Has also provided some OS-oriented code.

Frank Luithle
`f_luithle@outside.sb.sub.de`
Wrote a translation for v1.0. Unfortunately, he remained unreachable after that...:-/

Nat
`nat@nataa.frmug.fr.net`
Reported the same bug as Gerberding. In addition, he taught me a few tricks regarding the use of gcc + made me understand that the ANSI standard isn't unambigious; at least the `getenv()` call seem to be open for interpretations. Many possible incompatibilities have been removed due to these lessons.

Michael Sanders
`sanders@umich.edu`
Has found some of the bugs in this beast; both obscure and long-lived. Has also provided motivation to clarify this document.

Bjørn Ove Thue
`bjort@ifi.uio.no`
Author of the MSDOS port; please direct questions regarding strictly to that port to him (I would like to receive a carbon copy, though).

Martin Ward
`Martin.Ward@durham.ac.uk`
Sent a few bug-reports; also gave me information upon where to find regexp code. He also provided a Perl script for checking ordinary text, which ideas I was able to implement in ChkTEX. In addition, he sent me the source code for `lacheck`; which also inspired some of the warnings.

11 Contacting the author

If you wish to contact me for any reason or would like to participate in the development of ChkTEX, please write to:

Jens Berger
Spektrumvn. 4
N-0666 Oslo
Norway
E-mail: `<jensthi@ifi.uio.no>`

Any signs of intelligent life are welcomed; that should exclude piracy.

Have fun.