ChkTEX v1.4

Jens T. Berger Thielemann

April 30, 1996

1 Introduction

This program was written in frustration of that 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.

While written on an Amiga, it is written in ANSI C, so you can use this at your UN*X/MS-DOS/whatever site also. Full source included.

The program also supports output formats suitable for further processing by editors or other programs, making errors easy to catch. An ARexx script for interfacing with SCMSG (and via that, other editors) is included. A special script for CygnusED and an experimental script for GoldED is included. The program can also produce output suitable for Emacs.

The program itself does not have any machine requirements, Amiga and MSDOS binaries are included. Of course, you'll need ARexx and SCMSG to benefit from the ARexx scripts.

2 Legal stuff

3 Feature list

- Supports over 30 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. Detect most abbreviations automagically.
  - Enforcement of end-of-sentence space when the last sentence ended with capital letter.
  - Mathmode 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.
  - Text which may be ignored.
  - Mathematical operators typeset as variables.
- 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 LATEX2$_\varepsilon$.
- Flexible output handling. Has some predefined format, but lets the user specify his own format. Uses a "`printf()`" similar syntax. "`lacheck`" compatible mode included for interfacing with the AUC TeX Emacs mode.
- An ARexx script for interfacing with SCMSG, the SAS/C message browser, is included. Special script 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.

3.1 New features

Modifications and additions since v1.3:

- The whole subpart containing the error-checking routines has been restructured. This added the following benefits:
  - We are now *much* more context-sensitive; warnings are to a much larger degree muted/promoted according to whether we are in math-mode or not.
  - Detects math and verbatim environment (you may configure which are which), and acts accordingly.
- Yet more flexible resource format. UNIX systems may now have a *global* resource file; thus saving the individual users a lot of work installing the program.
- You may now customize which dashlengths you allow in different contexts.
- Removed that silly twirling baton which no-one used anyway.
- A few more warnings:
  - Checks that quotes are used the right way.
  - Checks for use of "`sin`" instead of "`\sin`" in mathmode and similar.
- "`configure`" script.
- Compiles happily on MSDOS computers; binary included.
- Output formats may now be specified globally in the "`chktexrc`" file now, for easy access.
- Most other suggestions incorporated.

4 Installation

A few words on installation on various platforms:

Amiga: Use the supplied Installer script.

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

UNIX: Type "`configure`", "`make`" and finally "`make install`". 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.

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 "`ChkTeX.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 also take a peak at the "`OpSys.c`" file first, 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.in`" should be moved to a directory in your path. These files may need further setup, as they haven't got the location of perl and csh initialized.

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.

Template: A UNIX-compliant template format follows:

```
chktex  [-hiqr] [-v[0-3]] [-l <rcfile>] [-[wemn] <[1-35]>]
        [-d <number>] [-p <pseudoname>] [-o <outputfile>]
        [-[btxgI][0|1]] file1 file2 ...
```

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]` Given a numeric argument above zero, it will output some status information during run-time.

Muting warning messages: Controls whether and in what form error messages will appear.

`-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, etc., 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` being the name of the file. See also `-b`.

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

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 (string).
`%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 [--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.

By default, we'll choose format 1 (that is, the *second* entry in the "`chktexrc`" file).

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.

This format will also turn off any twirling baton's, and other things which make the program unsuitable for pipes and batch scripts. As the program does yet 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`' file.

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.pl -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.

The default is mode 1, using `-v` without any parameter will give you mode 2.

`-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}` statement. 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 Ctrl + \ on the Amiga; on UNIX Ctrl + D. 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.

You should also take a look at the "`chktexrc`" file. The method for finding has been changed, it is now done as follows:

1. Look for it in the current directory.
2. Look for it in the directory pointed to by the environment variable `CHKTEXRC`.
3. Look in the following directories:

Machine	**Directory**
2.04+ Amiga	`ENV:`
1.3 Amiga	`S:`
UNIX	`$HOME`

If ChkTEX fails to find the "`chktexrc`" file it any of these places, it will look for it in the directory you specified as `DATADIR` when installing it, but then omitting the leading dot. Usually, this evaluates to `/usr/local/share/chktexrc` or `/usr/local/lib/chktexrc`.

It should be rather self-explanatory, and should help you customize the program to your own needs.

5.2 ChkWEB

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

`chkweb file1 file2 ...`

As you may see from the script, it is only a trivial interface towards "`deweb`" and ChkTEX. It does currently not support any options on the command line; the only intelligence it features is that it will try to append `.w` to filenames it can't find.

Please note that the "`deweb`" Perl5 script does in fact preserve linebreaks, and should also mostly keep column positions correct. `|...|` uses are currently not dealt with, though.

The "`deweb`" script is supposed to deal with all correct CWEB escape sequences; if I have accidentally omitted some, please let me know.

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.

Warning/error 1: `Command terminated with space.`

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.

Example:

`\LaTeX_is a typesetter.`
LaTEXis a typesetter.
`\LaTeX\ is a typesetter.`
LaTEX is a typesetter.

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

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.

Example:

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

Warning/error 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.

Example:

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

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

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

Example:

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

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

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

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

Example:

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

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

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

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.

Example:

`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/error 7: `Accent command 'command' needs use of 'command'.`

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.

Example:

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

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

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

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?).

Example:

```
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/error 9: `'%s' expected, found '%s'.`

You get this warning when you try to mix brackets or environments — ChkTEX expect to find matching brackets/environments in the same order as their opposites were found. While bracket matching is not an explicit error, it is usually a sign that something is wrong.

Warning/error 10: `Solo '%s' found.`

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

Warning/error 11: `You should use '%s' to achieve an ellipsis.`

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

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

Example:

```
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/error 12: `Interword spacing ('\ ') should perhaps be used.`

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

Example:

`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/error 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.

Example:

`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/error 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.

Example:

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

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

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

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

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

Example:

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

```
This is an ``example''
```
This is an “example”

Warning/error 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.

Example:

```
``There´s quotes and there´s quotes´´
```
“Theres quotes and theres quotes

```
``There's quotes and there's quotes''
```
“There's quotes and there's quotes”

Warning/error 20: `User-specified pattern found.`

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

Warning/error 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.

Example:

```
\LaTeX\ is an extension of \TeX\. Right?
```
LaTeX is an extension of TEXṘight?

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

Warning/error 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/error 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.

Example:

```
```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/error 24: `Delete this space to maintain correct pagereferences.`

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.

Example:

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

Warning/error 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.

Example:

```
$5\cdot10^10$
```

$5 \cdot 10^1 0$

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

$5 \cdot 10^{10}$

Warning/error 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.

Example:

```
Do you understand _?
```

Do you understand ?

```
Do you understand?
```

Do you understand?

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

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.

Example:

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

It is just a *prototype*, don't think anything else.

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

It is just a *prototype*, don't think anything else.

Warning/error 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.

Example:

```
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 $\times$ 200 pixels.

Warning/error 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.

Example:

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

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

Warning/error 31: `This text may be ignored.`

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

Warning/error 32: `Use ‘ to begin quotation, not ’.`
Warning/error 33: `Use ’ to end quotation, not ‘.`
Warning/error 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.

Example:

```
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/error 35: `You should perhaps use ‘cmd’ instead.`

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.

Example:

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

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

7 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. Currently, this should cause nothing but a few extra warnings.

There is also a small bug which will occur if you try to type the following:

```
Please take a look at ChkTeX.DVI. Please.
```

ChkTeX will then report that you both should use interword spacing and intersentence spacing. This happens whenever you use an uppercase abbreviation. Will be fixed — but it requires some restructuring of the program.

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

8 Future plans

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

- Support for regular expressions as user patterns. Have the code, just have to interface to it.
- Differentiate whether strings in the resource file should be checked case-sensitively or case-insensitively; we do now only do a dumb enforcement of what seems most practical for each keyword.
- Clean up the italic checking; it doesn't nest braces (in fact, that's why it magically works with LaTeX 2_ε...).
- Probably some more warnings/errors; just have to think them out first. Suggestions are appreciated — I've "stolen" most that lacheck provides, and am running out of ideas, really.
- Fix a few more bugs. :-)
- Put some garbage collecting into the thing. We're buffering almost one half of the lines in the file for the life of the program. 99% of these can be expunged *much* earlier. However, if you're dealing with so large files that you get out-of-mem errors, please consider splitting and using `\input` instead.

  Besides; if ChkTeX runs out of memory when processing your files, LaTeX will indeed do the same!
- Rewrite the thing in Perl? Get regexp's for free, at least. I have received positive reactions on this; however I feel that Perl hasn't the structural support C gives the programmer (I'm especially thinking of `struct`'s now, which simply doesn't feel the same when realized as an associative array).

9 Notes

9.1 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. Think of it as giftware — if you like it, you help improving it. The following ideas are given:

- 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.

- Reports on problems configuring and compiling ChkTEX on supported systems are welcomed.

- Information on the "`_doprnt`" function; prototypes and use. I've understood that not all systems support the "`vfprintf`" function (which ChkTEX uses a lot); however, some of them have a similar function (according to the Autoconf manual).

  I would like to receive any information on this; as I would like to create an interface towards that function.

- 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 have access to a dictionary listing abbreviations (not only English), I would appreciate an updated version of the "`chktexrc`" file with these filled in. In fact, 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.

9.2 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.

10 About the author

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

I'm 20 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 modelling, with a dash of physics and emphasing the computer part of the study. More precisely, in spring'96 my studies consist of discrete mathematics, signal- and image-processing plus co-operating in writing a program simulating the effects of acid rainfall in a test-area in the south of Norway (the Birkenes-field, close to Kristiansand).

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 textprocessing.

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 one included with my VIC-20 :-)).

However, I also try to not to end up as a computer nerd. Thus, in addition to the obligatory (?) 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' scoutgroup; Norwegian Scouts Association. I am also a active rover in 'Vlerenga' scoutgroup.

If you *really* like this program, and wish to ensure that development is continued, please consider sending one of the scout-badges of your country. It will surely be appreciated. Remember, all this development is done on my spare time, and I would be very grateful for your initiative.

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. . . :-)

11 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 suggested enhanced methods for filtering unnecessary warnings away.

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. 5–6 of the warnings implemented were based on his ideas.

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`
Found a long-time bug in the "`command not intended`" section. This one survived a few revisions, unfortunately (I haven't changed that part for quite a while).

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 few Perl scripts for inspiration and sent me the source code for `lacheck`; which inspired some of the warnings.

12 Contacting the author

If you wish to contact me for any reason (*scout-badges*, more tests, bug reports, porting, suggestions, hellos, smiley's, etc.) 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 is welcomed; that should exclude piracy.

Have fun.