TeXprofile: Displaying TeX profiles

Martin Ruckert

September 9, 2024
Version 1.0

Abstract

texprofile - Displaying the profile data collected by *texprof*.

1 Synopsis

texprofile [Options] *inputfile*

2 Description

texprofile reads the binary *inputfile* as produced by *texprof*. The extension `.tprof` of the input file can be omitted. It analyses the data and presents the results in a variety of tables, either optimized for human readability or as CSV (comma separated values) files for further processing.

The output is written to the standard output stream where it can be redirected to a file or viewed immediately.

3 Options

There are four types of options: general options, table options, selection options, and formatting options.

The general options:

-? -h −help Display a short help text and exit the program.

−version Display the version information and exit.

The table options determine the tables that will be displayed. These options all use upper case letters. If no table option is given, only some global information is shown.

-T Show the table of the top 10 input lines.

-G Show the table of the macro call graph.

-C Show the table of times per TeX command.

-L Show the table of times per input line.

-R Show the table of raw time stamps.

-S Show the table of macro stack changes.

-F Show the table of all input files.

-M Show the table of all macros called.

-A Show important tables (equal to -TGFC) tables.

-N Do not show the table of global information.

The selection options allow to reduce the amount of data that is shown in the table by omitting table entries that contribute little to the overall runtime.

-pn Do not show information for table entries with cumulative time below n percent. The default is 1.0 percent.

-tn Limit the number of input lines shown in the table of the "Top Ten" to n. This option is only useful with the **-T** option.

The formatting option determine the formatting of the information in the tables as well as the selection of information that is shown in the tables.

-i Add the macros file and line numbers after the macro name. This option is useful to distinguish two macros that share the same name.

-m Optimize output for machine readability. When optimized for human readability, times are rounded and given in a short form adding a unit identifier: s for seconds, ms for milliseconds, us for micro seconds, and ns for nanoseconds. This format is inconvenient if you want to import the data for example into a spread-sheet for further processing. With the **-m** option times are simply given in nano seconds without units. Similar the combined time for the total time used for a macro and the time used as a sub-macro from the call graph is given in a single column with a slash separating both numbers is nice for human readers. With the **-m** option both numbers will simply be given in separate columns.

-n Show the time stamp numbers. This option is useful together with the **-R** option, if you import the table into a database or spread-sheet. If you sort the data by various properties into different categories, the time stamp number can be used to keeping the entries within a category sorted by time.

-s Show the changes of the macro stack. This option is useful with the **-R** option if you want to see the changes in the macro stack (see **-S**) together with the timing information in a single table.

4   Data Formats

4.1   File Numbers

While *texprof* is running, every TeX input file is given a unique file number. The file name alone is often not unique because two files in different directories might have the same file name. Displaying the full file name with the complete path is however often not very convenient. So if you observe the same file name together with different file numbers, you can use the option **-F** to get a table of all file numbers and their full file names.

But even within the same directory, TeX can read different files with the same name during one run: TeX can open a file for writing, write content to the file, close it, open it for reading, read it and close it again. Then TeX might repeat this process a second time, or multiple times, reusing the same file name over and over again. *texprof* will assign a new file number to this file each time it is opened for reading. So you can tell from the sequence of file numbers which one was the first, the second, or the third.

Last not least, there are some special file numbers:

0 - unknown If the file is unknown, which should rarely happen, the file number 0 is used.

1- system *texprof* will map time intervals that are used to execute certain system routines to the "system" pseudo file using the line numbers to identify the specific routine like producing the output DVI file (shipout), breaking a paragraph into lines (linebrk), or breaking the document into pages (buildpg). These times do not depend on the use of macros but simply on the size of the document.

2 - terminal TeX commands entered on the command line or interactively are mapped to file number 2.

4.2   Time

If the option **-m** is given times are given as nanoseconds. Otherwise, times are rounded to at most 4 digits precission and displayed with a unit: seconds (s), milliseconds (ms), microseconds (um), or nanoseconds (ns).

4.3   Macro Names

Macro names are shown with the leading backslash. Since macro names are often not unique, the command line option **-i** can be used to show after the macro name in square brackets the file number and the line number where the macro is defined. Unless two macros with the same name are defined in the same file and line this is sufficient to uniquely identify a macro. A macro defined with "let" results in a complet copy of the original macro. Therefore it will not reference the file and line of the "let" command but the same file and line as the original macro.

5   Examples

Let's assume that you issue the command *texprof* **-prof** *hello.tex*. This will run *texprof* on the input file *hello.tex* with option **-prof**. The program *texprof* will load the plain TeX format and then process `hello.tex` to produce `hello.log` and `hello.dvi`. It will execute exactly the same steps that TeX would execute if you had issued the command *tex hello.tex*.

In addition to `hello.log` and `hello.dvi`, *texprof* will also produce the file `hello.tprof` containing time measurements made while *texprof* was running. The option **-prof** will switch on the gathering of timing data as soon as *texprof* enters its main control procedure; The file `hello.tprof` will contain a time measurement, called a time stamp, for every command that TeX executed while processing the input. A binary format is used to store all that data in a simple and compact form. Still the file `hello.tprof` might become very big.

The program *texprofile* is used to extract and analyse the data contained in `hello.tprof` and produce useful output. Here are some examples:

texprofile hello Without further options *texprofile* will write some general information to the standard output, like the total time measured, the number of samples, the average time per sample, etc. The general information is always given unless explicitely disabled with the **-N** option or with the **-m** option.

texprofile* -T *hello With the option **-T**, *texprofile* will map each time stamp to a specific line of input, add up the time intervalls for each input line separately, and output a table showing the ten lines that have the highest cummulative time. The table has the following seven columns:

1. **file** The first column contains the input file number as explained above. The input file name if shown in column 7.

2. **line** The line number.

3. **percent** The time spent in this line and file as a percentage of the total time measured as given as part of the general information.

4. **absolute** The absolute time spent in this line and file.

5. **count** The number of times the execution path entered this line. Note, that a macro call usually redirects the execution path to another line from where the execution will return after the macro call has completed. Reentering the line after such an excursion to an other line will cause this counter to be incremented. In summary, this counter might reflect the number of partial executions of a line not the number of full executions of the entire line.

6. **average** The average time spent in the line is simply computed by dividing the value in column 4 by the value in column 5.

7. **file** The input file name. The corresponding file number is shown in column 1.

texprofile -G hello With the option **-G**, *texprofile* will map each time stamp either to file input or to a macro body. The table shown is divided into several sections, the first section is devoted to file input, each of the following section is devoted to a specific macro.

Each section starts with a header. The header of the first section is "File" the header of a later section is the macro name.

The first line after the header gives the total time spent in the section in column 1 and 3 in different formats:

1. **time** Column 1 gives it as an absolut time.

3. **percentage** Column 3 gives it as a percentage of the total time measured.

For the first section with the header "File", the absolute time will be equal to the total time measured because *texprof* did spent all the time processing the file *hello*. And consequently the value in column 3 will be 100%.

To accomplish a task, a macro usually calls other macros, that we call child macros in the following. The following lines will give a breakdown of the time shown in the first line. The next line after the header gives the time spent in the section excluding the time spent in child macros. It adds in column 4 the number of times the macro was called.

1. **time** Column 1 gives the absolut time spent in the section excluding the time spent in child macros

3. **percentage** Column 3 gives the time from column 1 as a percentage of the total time spent in this section as given in the previous line.

4. **count/total** Column 4 gives the number of times the section was called.

6   Bugs

If the last command in a macro body is a macro call, we call this a tail call. If such a tail call reads ahead to scan the following input for possible arguments, the look-ahead mechanism of TeX might push further macros or new input files on TeX's input stack. These entries will remain on top of TeXprof's macro nesting stack, even if TeX backs up all these tokens on its input stack. This can cause an attribution of runtime to those entries as sub entries of the tail call. If this explanation sounds complicated to you, it is because the situation is indeed complicated.

7   See Also

texprof(1)

8   Version

Version: 1.0 of September 9, 2024

Copyright © 2024, Martin Ruckert,
    Hochschule München, Lothstrasse 64, D-80335 München

Distribution *texprofile* is distributed with TeX Live. More recent versions might be found on
    GitHub `https://github.com/ruckertm/HINT`.

License This program can be redistributed and/or modified under the terms of the MIT/X11
    license.

9   Author

Martin Ruckert
Munich University of Applied Sciences
Email: `martin.ruckert@hm.edu`