Installing teTeX on Linux Machines

Robert Kiesling v1.0, 31 March 1997 Contents of the teTeX HOWTO are Copyright (c) 1997 by Robert A. Kiesling. Permission is granted to copy this document, in whole or in part, provided that credit is given to the author and the Linux Documentation Project. Send all complaints, suggestions, errata, and any miscellaney to Robert_A._Kiesling@macline.com, so I can keep this document as complete and up to date as possible. Thanks! –Robert Kiesling

1 1. DO I REALLY WANT TO DO THIS?

FAQ No. 1. My computer just ate NINE high density diskettes' worth of data. WHAT HAPPENED?

Answer: Installing teTeX on Chanel3, my Compaq laptop, was like dropping a 20-foot concrete bridge section exactly into place from a height of 50 feet. teTeX is a BIG package. Even so, it is a moderately complete implementation of TeX 3.1415 and LaTeX 2e for Linux systems. TeX is very big subject anyway, so you can expect to spend the rest of your computing career keeping up-to-date on the latest in the world of TeX. That is to say, installing and using teTeX is not for the faint of heart, nor is it for daytrippers. Expect to spend some serious quality time with this package.

Thomas Esser, the author of teTeX, has gone to great lengths to make the package fast, complete, and easy to use. Because TeX is implemented for practically every serious computer system in the world, implementors must provide the installation facilities for all of them. This accounts in part for teTeX's size. It also accounts for the fact that the pieces necessary to make teTeX useful are spread all over your friendly neighborhood CTAN archive.

CTANistheCombinedTeXArchiveNetwork, aseriesofftpsiteswhicharchiveTeXprograms, macros, fonts, anddocumentation.You'llprobablybecomefamiliarwithatleastoneCTANsite.

Fortunately for us, some considerate implementors of the Linux Slackware Distribution have already assembled all of the necessary pieces for us. If you haven't installed the slackware distribution from scratch, however, you don't have the wonderful installpackage utility that comes with Slackware. Don't worry. It's not too difficult to install the individual pieces to teTeX by hand.

2 2. What is TeX? What is LaTeX? What is teTeX? What is the difference?

teTeX is Thomas Esser's implementation of TeX for Linux systems. This means getting the executable programs themselves to run under Linux and providing fonts in form usable by Linux users. The rest of the code, the TeX and LaTeX itself, is portable across various machines.

In addition to the executable programs themselves, the distribution includes all of the TeX, LaTeX, and Metafont sources, BibTeX, MakeIndex, and ALL of the documentation... about 3 megabyte's worth. In short, it's a reasonably complete implementation of TeX 3.1419 and LaTeX 2e, and the documentation covers everthing you will forseeably know to get started. So, install it. Not only will you eventually read all of it, the documentation also provides useful examples of "live" TeX and LaTeX code.

In comparison with other implementations of TeX, installation of teTeX is almost trivial even without the slackware setup program, not counting the effort necessary to insert and remove the nine diskettes of the slackware distribution. If you installed the slackware-distributed teTeX, you can pretty much skip the first part of this document. The next few sections are for the rest of you...

TeX is a typesetting system developed by Professor Donald Knuth. It is a lower-level typesetting language that powers all of the higher-level packages, like LaTeX. Essentially, LaTeX is a set of TeX macros that provide convenient, predefined document formats for end users. If you like the formats provided by LaTeX, you may never need to learn bare-bones TeX programming. The difference between the two languages is like the difference between assembly language and C. You can have the speed and flexibility of TeX, or the convenience of LaTeX. Which brings us to the next answer,

ANSWER NO. 2: YOU HAVE IT BACKWARDS!!! I WANT TO KNOW WHAT EXACTLY I NEED TO GET BEFORE I CAN HAVE TeX ON MY SYSTEM!

It's important to remember that TeX only handles the typesetting part of the document preparation. Generating output with TeX is like compiling source code into an executable program. You prepare an input file with a text editor – what most people think of as "word processing" – typeset the input file document with TeX to produce a device-independent output file, called a ".dvi" file. You will also need output output drivers for your printer and video display. These output drivers are collectively known as "dviware."

Any of the editors that come with Linux – jed, joe, jove, vi, vim, stevie, GNU emacs, microemacs – will work to prepare a TeX input file, as long as the editor reads and writes plain-vanilla ascii text. My preference is GNU emacs. There are several reasons for this:

1. Emacs's TeX and LaTeX modes obviate the need for a stand-alone TeX shell. 2. Emacs can automatically insert TeX-style, “curly quotes,” as you type, rather than the "ascii-vanilla" kind. 3. Emacs has integrated support for texinfo and makeinfo (more on this later). 4. Emacs is widely supported and does everything except butter the toast in the morning. 5. Emacs is free.

It's important (and probably overdue) to emphasize the difference between the TeX typesetting software and the .dvi imaging systems that are used for output, whether on-screen or on paper. TeX itself operates independent of both the fonts that you need to install for a particular output device, like a printer or plotter, and the output driver itself. In fact, much of TeX, and therefore LaTeX's complexity, arises from its implementation of various font systems, and the way these fonts are specified. A major improvement of LaTeX 2e over its predecessor was the way users specify fonts, the former New Font Selection Scheme.

teTeX comes distributed with about a dozen standard fonts preloaded, which is enough to get you started. Also provided are the font metrics descriptions, in .tfm (TeX font metric) files. To generate the other fonts you will need, it is simply a matter of installing the Metafont sources. teTeX's dvi utilities will invoke Metafont automatically and generate the Computer Modern fonts that you need, on-the-fly.

Later on, we'll discuss how to get TeX to use Postscript fonts on your system, and how to help unify all of the various font schemes on your system. Not only will this provide you with a more consistent look on-screen and in print, it can save you wads of disk space, too.

There's a lot of software to assemble. In the meantime, you can start in "learning" TeX and LaTeX. Remember that teTeX and the font packages have been designed as two separate entities: teTeX is specific to Linux, but the CM and DC font distributions have been designed for many different platforms. While you are working on assembling the files, you can take a few breaks to locate some of the documentation that you will need.

There are user manuals available both commercially and via the internet. The two most useful – and definitive – commmercially available texts for beginners are:

"LaTeX: A Document Preparation System," by Leslie Lamport, 272 pp. If you're using LaTeX instead of plain TeX (highly recommended), this is the definitive reference.

If you must use plain TeX, "The TeXBook," by Donald Knuth, 483 pp., is the definitive reference. It is also necessary if you plan to do any serious class, package, or macro writing for LaTeX.

At your nearest CTAN site you can retrieve these documents for free:

"The Not So Short Introduction to LaTeX2e," by Tobias Oetiker, Hubert Partl, Irene Hyna, and Elisabeth Schlegl, 69 pp. My URL to this wonderful document is:

ftp:wuarchive.wustl.edu:packages/TeX/info/lshort/*.

You can get a PostScript or .dvi version of the document ready for printing, or the native LaTeX document. There is also a version available in German: lkurz.*. Be sure to read the README file before assembling!

"A Gentle Introduction to TeX: A Manual for Self-Study," by Michael Doob, 91 pp. You can find this document at: ftp:wuarchive.wustl.edu:packages/TeX/info/gentle.tex Almost of necessity, this document covers less ground than its LaTeX counterpart, above. However, it will get you to the same place as the LaTeX manuals, if you still must use plain TeX for your documents.

There are, of course, other guides available to using TeX and LaTeX. They cover different aspects of these systems to varying degrees. The reference documents cited above, however, are the most comprehensive in scope that I have seen and the most suitable for someone who is approaching these systems for the first time.

You will note, however, that all of these references are hardware-independent. They won't tell you a thing about running teTeX specifically. That's where the documentation included with teTeX comes in, and what we'll spend most of the rest of this HOWTO on: getting output from this vast package.

3 3. PERFORMING THE ACTUAL INSTALLATION.

First, ftp to your nearest Linux archive site. Mine is

```
wuarchive.wustl.edu
```

then find the directory with the Slackware distribution diskettes. On wuarchive, this is

```
systems/linux/sunsite/distributions/slackware/slakware
```

Most Linux archives will store these diskettes in the directory distributions/slackware/slakware. teTeX, the full package, is contained on the slackware disk series t. So, grab all nine disks' worth of the t series, disks t1 - t9. It helps to keep them in order, too. Either store the files them in separate subdirectories labeled t1 - t9 on a hard drive partition, or on diskettes, and label the diskettes t1 through t9. We're going to install them, by hand, in order.

This is not difficult. The slackware installer creates the directories and unpacks the files. It also provides descriptions of each module in the distribution which allows you to decide whether you want to install it or not. In the case of teTeX, however, you are simply going to install everything, because that's what you should do anyway.

Installation will require about 30 Mb of disk space, so you should make sure that it's available before you start. You don't need to have the gcc compiler or the X Windows System installed (although X certainly is helpful because then you can preview documents on-screen). All you need is an editor that is capable of producing plain ascii text (see above). What could be simpler?

Let's assume that you have all nine diskettes' worth of the slackware teTeX distribution ready at hand, organized as described above. You'll have a lot of files that have the extension .tgz. This is shorthand for a gzipped tar archive. The names all fit the 8+3 filename limitations of MS-DOG. Aren't you glad you decided to scrap your DOG partitions and install Linux instead? You can use a MS-DOG hard disk partition or DOG-format diskettes to store the files. The archives also begin with the letters tb, td, or tm, and so on, which is the implementors' shorthand for TeX binary, TeX documentation, TeX macro, and so on. The difference to you is academic, because you'll be installing everything anyway.

Let's assume that you've assembled the slackware distribution on floppy diskettes, labelled t1 thru t9. Mount the t1 diskette like this

```
mount /dev/fd0 /mnt
```

if your Linux configuration is a standard Slackware configuration like mine. Actually, any mount point will do. You'll simply need to substitute the appropriate pathspec in the next few steps.

The next thing you want to do is create the teTeX top-level directory. teTeX's internal paths are specified relative to its binaries, but the Slackware distribution is archived relative to the root directory. So the top-level teTeX directory is

```
/usr/lib/teTeX
```

so, for each of the .tgz archive files in the distribution, copy the archive file to the /usr/lib/teTeX directory and repeat the following commands:

$Youshouldbeloggedinasrootandinthetop - leveldirectory, /, forthesesteps.I'veusedthetb - xfig.tgzarchivefordemonstrationpurposes.Ofcourse, you'llwanttosubstitutethenameofwhicheverarchiveyou'reunpacking.$

```
cp /mnt/tb-xfig.tgz /usr/lib/teTeX
gunzip /usr/lib/teTeX/tb-xfig.tgz
tar -xvf /usr/lib/teTeX/tb-xfig.tar  # v to see what's goint on!
rm /usr/lib/teTeX/tb-xfig.tar
```

If you're feeling REALLY adventurous, you can unpack the archives disk-by-disk. Repeat these steps for each t series diskette that you have mounted on /mnt:

```
cp /mnt/* /usr/lib/teTeX
gunzip /usr/lib/teTeX/*
tar -xvf /usr/lib/teTeX/*tar
rm /usr/lib/teTeX/*tar.
```

The important thing is to get all the files unpacked, and in the correct directories on your hard drive.

It's quite a rush, isn't it, watching all those filenames zipping by on the screen while the archives unpack onto your hard drive. Relax! Take a break, and freshen up your coffee (or grab another JOLT from the refrigerator, or otherwise replenish whatever you're drinking). There's only a few more steps. We'll take them in increasing order of difficulty.

The first thing you'll want to do is look at Thomas Esser's README file. It contains a lot of hints on how to configure teTeX for your output device (i.e., printer). The README file is located in the directory

```
/usr/lib/teTeX/texmf/doc/tetex
```

Read the file over with the command

```
less /usr/lib/teTeX/texmf/doc/tetex/README
```

or even better, print it out with the command

```
cat /usr/lib/teTeX/texmf/doc/tetex/README >/dev/lp0
```

assuming that your printer is connected to /dev/lp0. Substitute the appropriate device driver file as appropriate.

Repeat these steps with the teTeX-FAQ. Keep the FAQ handy because it contains some very useful hints for configuring teTeX's output drivers for your printer. We'll get to that in a moment.

It is disappointing that the Linux Slackware Distribution doesn't come with a standard lpr daemon. That's probably because of the wide variations in printing hardware, but that's only my semi-informed guess. This would be the subject of another HOWTO, but any lpr/lpq suite that conforms to the standard BSD configuration should work just fine.

Next, you want to define a directory to store your own TeX format files. teTeX searches the directories listed by the $TEXINPUTS environment variable for local TeX input files. On Chanel3, I added the line

```
export TEXINPUTS=".:~/texinputs:"
```

to the systemwide /etc/profile file. Of course, you must have logged in as root before you can do this. The $TEXINPUTS environment variable tells teTeX to look for users' individual TeX style files in the ~/texinputs directories under each user's home directory. It is CRITICAL that a colon appear before and after this directory. teTeX is going to append its own directory searches to your own. You want to have teTeX to search the local format files first, so it uses the local versions of any of the standard files you have edited.

Add the /usr/lib/teTeX/bin directory to systemwide path in whatever manner you usually do this, and restart the system to make sure the path and TEXINPUTS are registered properly; that is, globally.

Now, run texconfig per the instructions in the teTeX-FAQ and choose the printer that is attached to your system. Make sure that you have configured teTeX for both the correct printer and printer resolution.

4 4. USING TETEX.

Theoretically, at least, you should have everything installed correctly and ready to run. teTeX is a very large system, and you want to start by learning slowly, instead of being overwhelmed by the necessary complexities of the system.

So, instead of watching TeX typeset

"Hello, World!"

as Professor Knuth suggests in the TeXbook, we'll produce a couple of teTeX's own documents through the system in order to test it.

You should be logged in as root the first time you run teTeX. If you aren't, metafont may not be able to create the necessary directories for its fonts. Metafont will complain that it can't make the directories, and you won't see any output because you will have a bunch of zero-length font characters. This is no problem. Simply log out, re-login as root, and repeat the offending operation.

Of course, it doesn't mention this in teTeX's manual. But the nice thing about teTeX is that, if you blow it, no real harm is done. It's not like a compiler, where, say, you will trash the root partition if a pointer goes astray. What, you haven't read the teTeX manual yet? Of course you haven't! It's still in the distribution in source code form, waiting to be output!

So, without further delay, you will want to read the teTeX manual. It's located in the directory

```
/usr/lib/teTeX/texmf/doc/tetex.
```

Without further adieu, type

```
latex TETEXDOC.tex
```

LaTeX will print several warnings. One of them tells you to re-run the document through LaTeX to resolve its cross-references. This is a standard procedure for building a document's Table of Contents. So, repeat the command.

The other warnings can be safely ignored. They simply are warning you that some of the ftp paths mentioned in the documentation are too wide for their alloted spaces. (If you're really inquisitive, look at one of the TeX references for a discussion of \hbox and \vbox.)

LaTeX will have generated several files from TETEXDOC.tex. The one we're interested in is TETEXDOC.dvi. This is the device-independent output that you're going to send to the printer. There are several printer drivers included with teTeX, which allow you to change the dvi file into a fax, into plain text, or print it on a HP Laserjet. There are other dvi drivers available at CTAN sites. However, the best option, in my humble opinion, is convert the file to PostScript using dvips, which is included with the package. It isn't difficult. The command

```
dvips TETEXDOC.dvi -o TETEXDOC.ps
```

will do the job very nicely.

This is where the real work begins. (And you thought you were finished!) Dvips has invoked metafont, which is now happily generating all of the fonts that the teTeX User's Manual needs in order to print correctly. There are about half-a-dozen fonts in all, which can take anywhere from 5 minutes to several days (on a 10 MHZ 386 or something equally primitive) to generate.

If you're lucky (and rich) and have a PostScript printer, you can simply dump TETEXDOC.ps to the printer with a command like

```
lpr TETEXDOC.ps.
```

I haven't tested this, because I have a non-PostScript HP Deskjet 400 color inkjet printer connected to Chanel3's parallel port. This is where ghostscript comes into the picture. What ghostscript does is translate (actually, render) PostScript code into a form that any bitmapped output device can understand. Even though the Deskjet doesn't understand PostScript, ghostscript allows me to scale, rotate, blend, or otherwise alter text, or include graphics or colors, just like the expensive printers. A discussion of color printing is a little beyond the scope of this document, though, along with most other effects. We're going to stick to the basics for the moment.

The first thing you want to do (after dvips has completed writing TETEXDOC.ps) is invoke ghostscript to see its command line arguments, like this:

```
gs -help
```

You'll see a list of supported output devices and sundry other commands. Pick the output device that most nearly matches your printer. On Chanel3, because I generally produce black-and-white text, I use the cdjmono driver, which drives a color DeskJet in monochrome (black and white) mode.

The command line I would use is:

```
gs -dNOPAUSE -sDEVICE=cdjmono -sOutputFile=/tmp/gs.out TETEXDOC.ps -c quit
```

This will produce my HP-compatible output in the /tmp directory. It's a good idea to use a directory like /tmp, because gs can be particular about access permissions. Now you can print the file:

```
lpr /tmp/gs.out
```

Obviously, this can all go into a shell script. On my system, I have two simple scripts written, pv and pr, which simply outputs the PostScript file either to the display or the printer. Screen previewing is possible without X, but it's far from ideal. So, it's definitely worth the effort to install XFree86, or TinyX (which is what I did) in order to preview your output on the screen.

I'll probably write a HOWTO about dviware sometime in the future. Until then, the best thing to do is simply play with ghostscript to explore its various previewing options; failing that, you are strongly encouraged to use dvitty – this will give you ascii-readable text you can page through with less or Xless. It's only an approximation of what the printed output will look like, of course, but well worth the incovenience, given the amount of paper saved.

So, until you are able to view documents easily on-screen, please recycle your paper. If possible, purchase recycled photocopy paper to print on. You don't want your workplace to look like a branch office of Georgia-Pacific.

Remember: Save a tree... kill an editor.

Robert Kiesling

Robert_A._Kiesling@macline.com ... No writing gig too small ...