

Using Latex on Berkeley Unix

Richard Furuta (modified from that prepared by Leslie Lamport)

September 28, 2000
For Latex Version 3.01

Contents

1	Getting Started	2
1.1	Running a Sample File	2
1.2	Preparing and Running Latex on Your Own Files	3
2	Carrying On	4
2.1	Latex on Unix	4
2.2	Document Styles	5
2.2.1	Special Styles	5
2.2.2	The <code>proc</code> Style Option	5
2.2.3	Letters	5
2.3	Where the Files Are	6
2.4	Running <code>labl1st.tex</code> and <code>idx.tex</code>	6
2.5	Differences from the Manual	6
2.6	Using BibTeX	6
2.7	Using S _L T _E X	7
3	DVI and Printer Support	7
3.1	Postscript	7
4	Previewing a DVI file	8
5	Including Graphics in TeX	9

Latex runs on a variety of computers at many different sites. This document tells you how to use Latex at Berkeley Unix sites running the standardly distributed Unix version of TeX and LaTeX. It is not about Latex itself, which is described in the manual—*LaTeX: A Document Preparation System*, available from Addison-Wesley (ISBN 0-201-15790-X). The Latex manual is available in the ACS Technical Library for faculty to review.

If you have a question that you can't answer by reading the manual and this document, ask systems@umbc.edu.

To generate a copy of this document on the laser printer in ECS 019, use the following commands.

```
Use netscape and select UCS Documentation http://www.umbc.edu/ucs
Then select helpful information, from there select Latex.
```

1 Getting Started

1.1 Running a Sample File

Before preparing your own documents, you may want to get acquainted with Latex by running it on a sample input file. First make your own copy of the file `sample.tex` by typing the following Ultrix command:

```
cp /usr/local/lib/tex/inputs/sample.tex .
```

(You must type the space followed by the period at the end. In Unix, the last period signifies to the operating system to copy the file into your current default directory. This and all other Unix commands are ended by typing *return*.) A copy of the file `sample.tex` is now in your current directory; you can edit it just like any other file. If you destroy or mess up your copy, typing the above command again gets you a fresh one.

Next, run Latex on the file `sample.tex` by typing:

```
latex sample
```

When Latex has finished, it will have produced the file `sample.dvi` in your directory.

Next convert the file to Postscript using the command

```
dvips -o sample.ps sample.dvi
```

Next print the file to using the command

```
lpr -Pacsp sample.ps
```

1.2 Preparing and Running Latex on Your Own Files

You must use a text editor to prepare an input file for L^AT_EX. Academic Computing recommends that the *emacs* text editor be used. Emacs offers a TeX mode, which is invoked on files ending in *.tex*. Tex mode offers a number of useful features, these include:

- Automatically match beginning and ending braces.
- Automatically match beginning and ending dollar signs in math mode.
- Automatically set up a begin..end block for you.
- Emacs has a command to validate your buffer looking for mismatched braces and parenthesis.
- From within emacs you may run your file through a spell checker.
- Tex can be invoked from within emacs.

The best reference available for using emacs is the *Gnu Emacs Manual*, by Richard Stallman. For more information on using emacs at umbc examine the document *emacs* from the command *doc*.

The easiest way to start learning about Latex is by examining the file `small.tex` with your text editor. You can obtain your own copy of this file, in your directory, by typing the command

```
cp /usr/local/lib/tex/inputs/small.tex .
```

After you have prepared your file, whose name should have the extension `tex`, you must run it through Latex and print the output. Follow the instructions in Section 1.1, except substitute the first name of your file for “`sample`”. Remember to save disk space by deleting the `dvi` file after printing the output.

If you want to stop Latex in the middle of its execution, perhaps because it is printing a seemingly unending string of uninformative error messages, type *CTRL-C* (press *C* while holding down the key labeled *CTRL*). This will make Latex stop as if it had encountered an ordinary error, and you can return to Unix command level by typing *X*, as described in the Latex manual.

To use the *spell* program for finding spelling errors in a Latex input file named `myfile.tex`, type the following command:

```
From within emacs use M-x ispell-buffer
```

Where M-x is produced by first typing the Meta key (usually the key labeled Escape or ESC) and then pressing the key *x*. Emacs will display each word that it finds spelled incorrectly. As spelling errors are found the spelling checker will display a list of words to replace the mis-spelled word with. Select the number corresponding to the correct word, you will then be prompted for whether you want to replace the mis-spelled word with the selected word, if so press the key *y*.

2 Carrying On

2.1 Latex on Unix

The only special problems in using Latex caused by the Unix operating system involve the way Unix handles files. The first problem arises because, when a program starts to write a file, Unix destroys the previous version of that file. Thus, if an error forces you to stop Latex prematurely (by typing *Control-C* or *Control-Z*), then the files that Latex was writing are incomplete, and the previous complete versions have been destroyed. You probably don't care about the output on the *dvi* file, but, if you are making a table of contents or using cross-referencing commands, then Latex also writes one or more *auxiliary files* that it reads the next time it processes the same input file. If the auxiliary files are incomplete because Latex was stopped before reaching the end of its input file, then the table of contents and cross-references will be incorrect the next time Latex is run on the same input file. You will have to run Latex a second time to get them right. If you want to avoid having to run Latex twice after making an error—for example, if your input is very long—then you should save copies of these auxiliary files before running \LaTeX . An input file named `myfile.tex` and all the auxiliary files produced by Latex from it are included in the Unix file specifier `myfile.*`. Use the Unix `cp` command to save copies of these files.

The second problem in using Latex on Unix involves the files that Latex reads. The file whose name you type with Unix's `latex` command is called the *root file*. In addition to reading the root file, Latex also reads the files specified by `\input` and `\include` commands. With the Unix directory system, Latex must know not only the names of these file but also on what directories they are. It will have no problem finding the correct files if you follow two simple rules:

1. Run Latex from the directory containing the root file.
2. Keep all files specified by `\input` and `\include` commands in the same directory as the root file.

If you follow these rules, you never have to type an Unix path specifier when using \LaTeX .

You should never break the first rule, otherwise Latex will have trouble finding auxiliary files. (To run Latex on someone else's file, copy the file to your directory.) If you break the second rule, specifying a file from another directory in an `\input` or `\include` command, you must use a complete path name. For example, to include the file `hisfile.tex` from Jones' directory `/foo/bar`, you can type

```
\include{/faculty/jones/foo/bar/hisfile}
```

A `~` character may not appear in the argument of an `\input` or `\include` command, so you *can't* use a file name such as `~jones/foo/bar/hisfile`.

For people who don't like to obey rules, here is exactly how Latex finds its files. The root file is found by Unix according to its usual rules. \LaTeX 's auxiliary files are read and written in the directory from which it is run. All file names specified in the Latex input, including the names of document-style (`sty`) files specified by the `\documentstyle` command, are interpreted relative to the directory from which Latex is run. If Latex does not find a file starting in this directory, it looks in the system directory `/usr/local/lib/tex/inputs`. You can change the directories in which Latex looks for its input files by setting the environment variable `TEXINPUTS`. Putting the command

```
setenv TEXINPUTS :./systems/jack/inputs:/usr/local/lib/tex/inputs:
```

in your `.login` file causes Latex to look for files first in the current directory, then in your personal Latex inputs directory, and then in the system directory.

2.2 Document Styles

2.2.1 Special Styles

The only document styles and style options currently available here that are not described in the manual are the `proc` style option for making camera-ready copy for conference proceedings and the `acsletter` style for making letters. They are described below.

2.2.2 The `proc` Style Option

The `proc` option is used with the `article` document style. It produces two-column output for ACM and IEEE conference proceedings. The command `\copyrightspace` makes the blank space at the bottom of the first column of the first page, where the proceedings editor will insert a copyright notice. This command works by producing a blank footnote, so it is placed in the text of the first column. It must go after any `\footnote` command that generates a footnote in that column.

Latex automatically numbers the output pages. It's a good idea to identify the paper on each page of output. Placing the command

```
\markright{Jones---Foo}
```

in the preamble (before the `\begin{document}` command) prints "Jones—Foo" at the bottom of each page.

2.2.3 Letters

There are a number of departmental letters styles available on the campus. I will be glad to place departmental letter styles in the system style directory if

departments desire. For faculty who would like to experiment creating their own letter style I have placed the style file used in Academic Computing *memo*, into the directory `/usr/local/lib/tex/inputs`. To copy this file into your directory use the command:

```
cp /usr/local/lib/tex/inputs/memo.sty .
```

2.3 Where the Files Are

All Latex files mentioned in the manual, including the `sty` and `doc` files, are on the directory `/usr/local/lib/tex/inputs`. Fonts are stored in `/usr/local/lib/tex/fonts`. The `*.tfm` files are used by TeX and L^AT_EX.

2.4 Running `lablst.tex` and `idx.tex`

A list of labels and citations in an input file is printed by running Latex on the input file `lablst.tex`, which is done by typing

```
latex lablst
```

Latex will then ask for the name of the input file, which should be typed without an extension, and for the name of the main document style (e.g., `article`), used by that file.

The index entries on an `idx` file are printed by running Latex on the file `idx.tex`, which is done by typing

```
latex idx
```

Latex will ask for the name of the `idx` file, which is typed without an extension.

2.5 Differences from the Manual

All Latex features described in the manual are provided by the Unix implementation.

2.6 Using BibTeX

BibTeX is a program for compiling a reference list for a document from a bibliographic database. It is run by typing

```
bibtex myfile
```

where `myfile.tex` is the name of your Latex input file. This reads the file `myfile.aux`, which was generated when you ran Latex on `myfile.tex`, and produces the file `myfile.bbl`. BibTeX should be run from the directory containing `myfile.tex` (which should be the same directory from which Latex was run on that file).

If the `bib` file is not in the same directory as the Latex input file—for example, if you’re using someone else’s `bib` file—then you must include a path as part of the file name specified by the `\bibliography` command. A `~` cannot appear in the argument of a `\bibliography` command, so you should use a complete path name. For example, the Latex command

```
\bibliography{/udir/jones/bibfiles/gnus}
```

specifies the file `gnus.bib` kept by Jones in his `/bibfiles` directory.

There is now no formal provision for sharing bibliographic database information, nor are there programs to assist in making your own `bib` files. Suggestions for forming one or more common `bib` files are welcome.

2.7 Using SLiTeX

Slitex is a version of Latex for making slides. To run Slitex with a root file `myroot.tex`, you type

```
slitex myroot
```

(You should be connected to the directory containing `myroot.tex`.) Refer to Section 2.1 if you want slide files or `\input` files to be in a different directory from your root file.

3 DVI and Printer Support

Under Unix, `lpr` is the command used to print a file. The `-P` option specifies the printer queue to use for printing the job (notice that there is no blank space between the `-P` option and the queue name).

3.1 Postscript

To produce files for the Postscript queue in the ECS Building use the commands

```
dvips -o file.ps file.dvi
lpr -Pacsps file.ps
```

Under Unix, `lpr` is the command used to print a file. The `-P` option specifies the printer queue to use for printing the job (notice that there is no blank space between the `-P` option and the queue name).

In addition under Unix the command `dvips` is available. The `dvips` command converts a TeX Dvi file into postscript. The form of the `dvips` command is

```
dvips file.dvi
```

The `dvips` command will produce as output a file containing postscript with a file type of `.ps` .

At the present time the Postscript queue(s) are the following:

1. `acsps` - ECS 019
2. `acspsdup` -ECS 019 Duplex printing
3. `acslp` - HPLJ4 printer with postscript
4. Any department postscript printer.

There are a number of options available with the `dvips` command. These include options to produce a postscript output file, which page to begin processing on, the number of pages to process, etc. The `-o file.ps` option instructs `dvips` to produce a output file containing the postscript commands instead of directly printing the file. The option `-s #` instructs `dvips` to begin processing on the page specified. The option `-n #` instructs `dvips` to process that number of pages. The following example instructs `dvips` to produce a 3 page postscript output file named `unixlocal-5-7.ps` that has a starting page of five:

```
dvips -o unixlocal-5-7.ps -s 5 -n 3 unixlocal.dvi
```

4 Previewing a DVI file

Academic Computing is supplying a TeX Previewer on machines supporting X-Windows. There are at least two previewers available under Unix for workstations, these are `xdvi` and `texx2`. Academic Computing is only supplying `xdvi` with workstations at the present time.

The `xdvi` previewer works by reading a `dvi` file and produces output that can be displayed in a window under X-Windows. To use the `dvi` previewer enter the command

```
xdvi file.dvi &
```

The `xdvi` program will bring up a rubber-band box that you can position on your screen. Once you have the box positioned at the proper location pres the left mouse button. The `xdvi` program maintains a series of buttons going down the right hand side of the window that function as the interface to the `xdvi` program. These buttons have the following meaning:

- QUIT – Exit `xdvi` program.
- SHRINK1– This button causes a small area of text to be displayed in the window. This text is then made quite large. This option is useful for examining formulas or other complicated structures; however because it reduces the amount of a page that is displayed most people do not use this option.

- SHRINK2– This button causes more text to be displayed but at small character size than SHRINK1.
- SHRINK3– This button will display about 75 percent of a page in an easy to read font. This button is the default when you enter xdvi.
- SHRINK4– This button will show the entire page of text on the display. Selecting this button will let you determine if your output is formatted correctly on the page; however, the characters are often difficult to read. I find that I switch between SHRINK3 and SHRINK4 for most work.
- NEXT – This button causes the next page to be displayed.
- PAGE+5 – This button cause xdvi to jump 5 pages ahead.
- PAGE+10– This button causes xdvi to jump 10 pages ahead.
- PREV – This button displays the previous page.
- PAGE-5 – This button causes xdvi to jump 5 pages back.
- PAGE-10– This button causes xdvi to jump 10 pages back.

In addition, placing the mouse cursor over a part of the previewed page and pressing one of the mouse buttons will cause the text to be enlarged. The amount of text area enlarged increases as you go from the left button to the middle button and up to the right button. I find selecting the right mouse button works best for me.

One nice feature of the xdvi program is that it will recognize when the dvi file it is previewing has been updated. This allows you to make changes to your tex file in emacs, produce a new dvi file, and instantly preview the changed dvi file.

5 Including Graphics in TeX

TeX has a very limited graphics capability. There are ways to overcome the limitations that TeX places on graphics. One method is to produce a postscript graphics file which will be included in your tex file. The second method is to use the latex picture mode.

Producing a postscript file is certainly the easiest method of including graphics. A postscript file can be produced by a number of utilities on Unix. These include the X-Windows based drawing programs xfig and idraw, graphing utilities such as Xgraph and Xgnuplot, and applications programs such as matlab and mathematica. Graphics displayed on an X-Window display can be saved and converted to a postscript file. Finally, postscript files can be generated on a Macintosh or PC and uploaded for inclusion in your TeX document.

The method used at UMBC for including postscript graphics relies on a set of TeX macros named *psfig*. These macros should be included at the beginning of your document via the command

```
input{psfig}
```

Once these macros are included you may import postscript graphics into your TeX document with the command:

```
psfig{figure=file.ps}
```

The psfig macros have many options that allow resizing the postscript graphic or producing special paging effects. To find out more about psfig read the document psfig available through doc.

Two utilities will produce a file containing latex picture mode commands. These utilities are Xgnuplot and xfig. The advantage of producing a file with picture mode commands is that the document can be printed on any laser printer. The problem with using LateX picture mode is that the graphics quality is not as good as postscript and a very complicated graph requires a large amount of internal tex memory. Under umbc5 we have produced a special version of teX and LaTeX named *bigtex* and *biglatex* with very large amounts of internal memory. These programs, due to their size are much slower than the normal TeX and LaTeX programs and should only be used when TeX or LateX aborts due to lack of memory. When TeX or LaTeX abort due to lack of memory an error message similar to the following is produced:

```
Memory Exceeded xxxxxxxx  
Contact a wizard to enlarge me
```