For submission to The PracTEX Journal
Draft of July 9, 2007

The "hacking for learning" paradigm in LaTeX – Part 2 *

S. Parthasarathy†
drpartha@gmail.com

Abstract This article is a sequel to an article published earlier in this journal [1]. The earlier article argued a case for making hacking an accepted way of learning LaTeX . We now examine certain basic principles, which will make hacking an effective way of mastering LaTeX . Much of this advise comes from the author's own experiences.

PRACTEX JOURNAL STYLE. DRAFT PAPER.

1 Preamble

Hacking has become a culture, and a way of life, for people who have to depend on self-tutoring. Hacking is a very effective means of learning several subjects e.g. scripting languages like Python, and Ruby, HTML, Linux, shell scripting, and LaTeX . In particular, LaTeX is eminently suitable for the "hacking for learning" paradigm. The earlier paper [1] focused on giving the basis and justifications for using "hacking-for learning". We now examine certain principles which will make hacking-for-learning, much more effective.

2 Effective hacking

Learners must appreciate that there is no magic-wand for learning. Learners, particularly those who wish to master LaTeX (and the art of typesetting), must

*This article is dedicated to Richard M Stallman, President of the Free Software Foundation. He is an inspiration for all good-intentioned hackers (like the author of this article)

†Algologic Research and Solutions, 78 Sancharpuri Colony, Bowenpally, Secunderabad 500 011, India WWW-URL – http://algolog.tripod.com/nupartha.htm

exhibit enormous patience and perseverance (even when they use hacking-for-learning). Here are a few simple dos and don'ts which will help the learner, if he/she chooses to adopt "hacking for learning".

Hacking is NOT black-magic. Hacking is no magic pill. "Hacking for learning" does not make hacking a voodoo. It has also to obey certain logic, and principles of pedagogy. A systematic approach to hacking will help the learner to avoid various hazards. Learners must avoid the "shot in the dark" approach when trying hacking.

1. Try to chase one goal at a time. Explore one command at a time. Build up progressively, trying out one command at a time.
2. When you want to try out an alternate command, in place of an existing command, do NOT delete the existing command. Just nullify it, by commenting it out. Do NOT delete unwanted text until you are sure of what you did. You may sometimes like to retrace your steps.
3. Continue exploring till the question you had in mind, is settled decisively. Do not give up half-way, and leave unfinished commands in your source text. This will help you avoid a lot of headaches later on.
4. If you have found some clever or unusual solution, remember to document it, using comments. Add a few words about the rationale why you did it this way. People tend to forget their own inventions.
5. If you change your mind, or if your command does not achieve what you want, and if you want to revert to an earlier command, simply comment out the defective command and uncomment the one you want to try again.
6. Nested commands are most slippery. Be careful.
7. Commands with more than one argument can be very confusing. Take care.
8. Take care of your parantheses. Wrongly matched parantheses can cause havoc, and send you on a wild-goose chase.
9. Study your style sheet carefully. Understand what is being done there. Many of your questions will get answered if you do this.
10. A little help and hand-holding is always useful, no matter how good you are at LaTeX . Join a Latex User Group . Ask questions. There is no such

thing as a silly question. Conversely, help others, and try to let people know, if you have found some innovative solution.

11. Create your own "bag of tricks". Document your experimental files and store them in a convenient place. Dig into this bag, whenever you are looking for some help (auto-hacking). Remember, this bag of tricks will be useful to you, even several years after you have created the collection. So preserve this carefully (in more than one format, and in more than one medium).
12. Choose your inspiration carefully – the text which you wish to hack, can decide whether you will succeed or not. Like in every profession, there are bad LaTeX programmers, and there are good LaTeX programmers. Make sure you are following a good example.
13. LaTeX is often used for typesetting maths. Make sure that your maths is right, and that your maths expressions are well-formed. Make sure you have respected all the rules of precedence, and all the usual conventions of good mathematics.
14. Be fair to all. Help others to hack, just like you used hacking for learning. Make the source text (.tex file) available, along with the rendered file (usually a pdf file). Encourage people to use hacking, by sharing your own creations with them. Who knows ? Someday, your own files may be useful to you too (auto-hacking).
15. Reduce your dependence on word-processors. Try to shun your word-processor habits and hangovers. Do not try to find an equivalent function or procedure. It is like comparing apples with oranges.
16. Avoid hacking large files. You will end up confusing yourself. Use divide-and-rule policy. You can make small files, debug them, and save them for future use. You can then merge these files into one, using the `\include` facility of LaTeX .

The above steps will help you in using hacking as an effective way of learning LaTeX . However, you should also consider reducing progressively, your dependence on hacking. It is generally a good idea to to keep the LaTeX bible [2] and the LaTeX gospel [3] handy. Try to read and understand the ideas and pinciples of LaTeX .

3 IDEs and front-ends

Integrated development environments (IDE) and front-end tools like Kile or LyX, reduce the effort of learning LaTeX . There are also a variety of LaTeX - aware text-editors like emacs or Crimson. But, these tools also have a learning curve. Use them whenever you can. They can be used to support your hacking-for-learning efforts.

For larger projects e.g. books, multi-volume compendia, conference proceedings, multi-author publications etc., it would be wise to use a version-control system like RCS, CVS or subversion. This adds another layer of complexity, to the learning process.

4 Postscript

An overdose of anything can be lethal. Hacking, when practised in moderation, *along with other forms of learning*, can be an excellent way of jumpstarting the learning process for LaTeX .

This article was prepared under Suse Linux 10, using KDE-3.4.2 b, and the Kile 1.8 front-end. Hacking was not necessary, in view of the simplicity of the text. The author welcomes suggestions and reports on real-life experiences.

5 About the author

Parthasarathy teaches discrete mathematics, and preaches LaTeX and Linux, to undergaduate students of Computer Science, at Hyderabad, India. he runs a specialised enterprise which sues Linux and LaTeX exclusively. His website (marked in the footnote, in the title page) will give more specific details about him.

References

[1] S. Parthasarathy, The hacking-for-learning paradigm in LaTeX , PracTeX Journal, Vol. 2007, No. 1, Jan. 2007.

[2] L. Lamport, LaTeX : A document preparation system, Pub.: Addison Wesly, 1986 (The LaTeX bible)

[3] M Goossens, F Mittelbach, A Samarin, The LaTeX Companion, Pub.: Addison Wesley, 1994. (The LaTeX gospel)

* * * * *