

the codex

Life with Linux — A Zine

Typeset in L^AT_EX

Kenneth John Odle

2021
August

Contents

1	The Early Salad Days	3
1.1	Calculators	3
2	What’s to Like About Linux	5
3	Coda	7
3.1	What I Learned About L ^A T _E X While Creating This Issue . . .	7

Impressum

All contents ©2021 Kenneth John Odle

Although this is now in your hands, and it’s also on the web, so if you really wanted to steal this, I’ve made it pretty darn easy. I can’t imagine why anyone would want to, though.

FYI, this is made in L^AT_EX using the report document class. It then gets exported to a letterhalf (5.5 in x 8.5 in) pdf, which then gets made into a booklet using Boomaga (<https://www.boomaga.org/>).

I’m pushing this to my own git server as I write this. You can find it here: <https://git.kjodle.net/kjodle/the-codex>. New issues will be pushed after they are complete.

You can just skip over all the diversions in you want. It’s just how my mind works. (And yes, there will be politics in this. *You have been warned.*)

Chapter 1

The Early Salad Days

Boring, early life stuff when my world smelled like sweat and warm bologna. Feel free to skip this. I wish I could.

1.1 Calculators

Before computers were in my life, there were calculators.

These days, every kid has to have an expensive graphing calculator for middle school math. Specifically, it has to be a Texas Instruments graphing calculator, because the examples in the textbook are all described in terms of a Texas Instruments calculator.

Oh my, a diversion already.

(A little off track here, but this begs two questions: 1) Why is it always a TI calculator that's required, and 2) Are we teaching kids to learn math or to learn how to operate a calculator? The answer to the first one is that Texas Instruments and the Major Textbook Publishers™ have colluded to produce expensive books that need to be replaced every two to three years [thereby costing the school district money] and that require expensive calculator [thereby costing you as a parent money]. It's a racket, but that's capitalism for you. The answer to the second question is that we are teaching kids how to use calculators. Teaching them how to do actual math would require thought on

both the parts of the teachers and the parts of the students. Again, education in the United States has become a racket, but that's capitalism for you.)

(Well, you can probably tell what my thoughts are on the dominant economic system on planet Earth. There *will* be more of that. If you're okay with that, I'm okay with that, too. If you're not okay with it and you want your money back, it's too late—I've already spent it.)

I have noticed that even little kids are required to bring little kid calculators to school with them in most of the local school districts. As I write this, the school supply buying season is coming to an end, but for the past six weeks every store was filled with school supply lists and yeah, you have to have a calculator to get into the second grade.

Chapter 2

What's to Like About Linux

I could go on and on here.

But really, what I like—not love (that's about aesthetics for me)—is that I'm in control.

Partly, that's the nature of open-source computing. If you want to know how something works, you can look at the source code. If you don't understand the source code, you can research how the source code works. You can ask questions. (Thank you, StackExchange!) You can do some more research and then learn how to ask *better* questions. There is always something to learn, and once you've learned everything there is to learn about a particular piece of software ¹ you can fork it and start contributing to the project yourself.

Wondering how something in Windows works? So is everybody else. There is nothing more frustrating than googling a problem in Windoze, getting hundreds or thousands of results, and every result is just somebody else asking the same question.

And yeah, you can write code and create applications for Windows, and you can solve a lot of problems that way, but you can never make Windows itself better. It is what it is, and if you don't like it, the feature that bugs you might be made better in the next release, or it might be made worse. It's a crap shoot, really.

For what it's worth, Mac OS X, even though it is based on Unix/Linux (I forget which—I dropped out of the Mac world at OS X version 4), is the

¹Which is never really true. What I really mean is that when you've learned everything *you* want to know about it.

same way. There *might* be an answer, there *might* be a solution, but you just *might* be on your own there, buddy.

But what I really, really like about Linux?

The command line.

I'll probably write about this some more later, but my experience with computers goes back way before Macintosh made the mouse popular (and necessary). You turned on the computer, and there was just this dark screen with a blinking cursor. If you wanted to make the thing do something, you had to *know* something. With a GUI, you can guess. You can guess a lot, actually, and just poke around all you want because most GUIs come with an undo feature.

There is no “undo” on the command line.

Chapter 3

Coda

3.1 What I Learned About \LaTeX While Creating This Issue

I'm still a relative newbie to LaTeX, so there's always something to learn. Here's a running list of what I've learned so far:

1. You might think you want the **book** document class, but you probably will find the **report** class just as handy.
2. You want links¹? Use the **hyperref** package.
3. The **kpfonts** package has beautiful fonts.
4. Footnotes are easy! (Seriously, footnotes in \LaTeX have got to be the easiest footnotes I've ever managed.)
5. Use the **fancyhdr** package to get more granular control over your headers and footers.
6. You can use the **geometry** package to make a document have a paper size of half letter.
7. You can make your top margin larger by using `\addtolength{\topmargin}{0.5in}` but there is not a similar parameter for the

¹Yeah, I know these are irrelevant in a paper document.

bottom margin. Instead, you need to make the text box shorter by using `\addtolength{\textheight}{-1in}`.

8. Want to show inline code without executing it? Use `verb` following by two pipes. Place your code between the pipes. (I had to use two of those in #7, because that code just went right off the edge of the page when I only used one.)
9. Need a little space between elements? Just insert `\ ,` (that is, a backslash followed by a comma).

Like I said, I'm still a newb and I may be completely wrong or off base on some of these things, in which case, I'll make a note of that in a future issue ²

If you are interested, there is a link in the Impressum to the git repo for this publication where you can check out the source code.

²Always assuming that there *will* be another issue.