

the codex

Life with Linux — A Zine


Typeset in L^AT_EX

Kenneth John Odle

2021
2 September

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. You don't need to, however, since this is licenced under a CC BY-NA-SA 4.0 Creative Commons license. More information is at <https://creativecommons.org/licenses/by-nc-sa/4.0/>. 

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.

The image on the front cover is courtesy JericoDelayah from the Wiki-Media Commons. The image is over here: https://commons.wikimedia.org/wiki/File:4_RETAT_04_Linus_Torvalds.jpg. You can also find a link to the Creative Commons CC BY-SA 3.0 license there, as well.

The image on the back cover is one that I highly agree with. We built it, it's ours, and we shouldn't be charged for using it. I want my tax dollars to serve my needs. I don't want my tax dollars used to make rich white old men richer.

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.*) Also, I use a lot of em-dashes and parentheses because that is also how my mind works. It's just one big long stream of consciousness up in here most days.

Contents

1	The Early Salad Days	5
1.1	Calculators	5
1.2	Speaking of Watches, Timex Used to make Home Computers	7
1.3	The Joy of a Trash-80	8
2	What's to Like About Linux	9
2.1	Control...and an Opportunity	9
2.2	Knowledge is Power	10
2.3	The Unix Philosophy	11
2.3.1	Where does the Unix Principle actually apply in real life?	12
2.3.2	Where else does the Unix Principle <i>not</i> apply that it probably should in real life?	15
2.3.3	Where does the Unix Principle not apply in real life and this is actually a good thing?	15
3	What Are All Those Files in the Linux Root?	16
3.1	bin	16
3.2	boot	17
3.3	dev	17
3.4	etc	17
3.5	home	17
3.6	lib	17
3.7	media	17
3.8	mnt	17
3.9	opt	18
3.10	proc	18

- 3.11 root 18
- 3.12 run 18
- 3.13 sbin 18
- 3.14 srv 18
- 3.15 sys 18
- 3.16 tmp 18
- 3.17 usr 19
- 3.18 var 19

- 4 Miscellany 20**
- 4.1 Is This Really a Hack? Or Is It Just a Tip? 20
- 4.2 What I Learned About L^AT_EX While Creating This Issue . . . 20
- 4.3 Coda: Why L^AT_EX? 21

Chapter 1

The Early Salad Days

Boring, early life stuff when my world smelled like sweat and disinfectant and room temperature 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 school starting with 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.

I mean, *sure* you can get your kid that Casio, which has all the same features and all the same buttons and is an order of magnitude cheaper, but you spent all that money on an expensive pre-school, and all that money on expensive tutors. Do you *really* (he asked snottily) want to risk little Jimmy's chances of getting into Harvard because you were temporarily too cheap to buy the right calculator? Just buy the TI already!

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 question is that Texas Instruments and the Major Textbook Pub-

lishers™ 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 calculators¹ [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 teachers² and the parts of the students, not to mention on the parts of parents and especially administrators, who would also be required to grow a spine—and learn how to use it. 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.³)

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.

Ironically, the earliest calculators I can remember seeing (not getting my hands on, because they didn't belong to me) were Texas Instruments calculators. I don't remember a lot about them, but an uncle had given a pair to two of my cousins. They took a *ton* of batteries, had red LEDs for outputs (meaning they glowed in the dark—you could use them in the dark if you memorized the keypad), and they were designed for students because they had a go-back-through-all-your-steps-function-to-see-where-you-done-screwed-up-boy function, which would be a useful fea-

¹A few years ago, I bought a scientific calculator at the **dollar store** and tested it against my very expensive TI-92. It was just as accurate as the more expensive calculator, and cheaper by two orders of magnitude. Did I mention that this is a racket? I really should do a YouTube video or blog post about this.

²To be fair, a lot of teachers would like to teach kids how to do actual math. But they also need to eat and when it comes down to the difference between doing what is right and doing what pays the bills, they will do the latter. It's not their fault, really; it's just that the system does not like anybody who sticks out. Keep your head down and the worksheets graded—that's what the system rewards.

³But that's capitalism for you!

ture on modern calculators to learn math, but again, we're not interested in kids actually learning how to think and do something as radical as math.

The other early calculator I remember was a Casio calculator and it was on a watch. A kid I knew for a short time had one, and even let me wear it for a while. (I wish I could remember his name, because this was a tremendous kindness on his part.) I swore that when I grew up, I would own one of these watches.



Well, I grew up and I didn't buy one of them, even though they are still available. I could just never justify spending the money on what is—let's face it: just a bit of full-frontal nerdity—when there were bills to pay. Nope, just could never bring myself to do it.

It's just me now, and my expenses are numerous but small, and a couple of years ago my local all-in-one-store had all their watches on sale for 40% off, including the name brand watches. I checked—it was in stock. At \$25 bucks it was a lot, but on sale it was only \$15. I could do this! So I picked it up and looked at it lovingly, thinking about all the good times we would have together as we went forth and explored the world one simple calculation at a time.

But there was a problem, a rather large problem, actually. The print on those buttons is tiny. And my eyes are bad. I couldn't actually read any of buttons. I use reading glasses when I'm reading or working on the computer, but I don't need them out in the wild. I could wear the watch with me everywhere, but unless I were at my desk, I wouldn't be able to actually use it.

Back on the shelf it went. I'm not going to spend money on something that is not actually useful to me.

At this point, my only hope is that maybe my eyes will get so bad that I'll need bifocals all day, every day. When that happens, will this watch be on sale for so little money ever again? I highly doubt it.

1.2 Speaking of Watches, Timex Used to make Home Computers

My earliest memory of a computer in somebody's home is of being in an aunt's apartment, and she had a Timex Sinclair hooked up to her televi-

sion.

I don't remember much about it, actually, other than it was small and sleek and very modern-looking. I do remember that I was not allowed to touch it.⁴

This is where memory gets wonky, because I remember seeing this when I was about ten years old. But according to Wikipedia, the Timex Sinclair⁵ was released in 1982, when I would have been 14 years old. So it's entirely possible that my memory is losing track of *when* things happened, or it's possible that this aunt had some other home computer that for whatever reason my brain thinks is a Timex Sinclair. Who knows? I certainly don't, and I'll probably never find out for sure.

1.3 The Joy of a Trash-80

One thing I'm quite sure about is that in seventh grade a select group of smart kids from my class were allowed to go to the local "skills center"⁶ one day a week (Wednesday afternoons, as I recall) to study computers. This was the first time I'd every laid my fingers on an actual computer keyboard.

Oh look, another diversion.

"TRS" actually stands for "The 80. Radio Shack," as in *The Radio Shack*

⁴This aunt bought things not because she found them useful, but because other people didn't have them and she wanted to always have a status symbol to point to. I don't remember her actually doing anything *useful* with this computer.

⁵It was called the Timex Sinclair because this was a collaboration between the Time Corporation and the Sinclair Corporation. I imagine Sinclair handled the R&D and manufacturing and Timex handled the marketing. If so, Timex didn't *technically* make a computer, but they wanted us to think that they did. Good enough for me.

⁶This was a centralized school where eleventh and twelfth graders who definitely weren't going on to college could take classes like agriculture and welding. We used to teach these classes in each school under the guise of "vocational education" but somehow lost our way.

Chapter 2

What's to Like About Linux

I could go on and on here, but I'll try to keep it short. I can always come back to this. (And I probably will.)

2.1 Control...and an Opportunity

What I like—not love (love is about aesthetics for me when it comes to computers)—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 software⁷ 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

⁷Which is never really true. What I really mean is that when you've learned everything *you* want to know about it.

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

That's the key when you're working with something that open-source: every problem is an opportunity for you to learn something. You might be able to find a workaround, or a fix, or even realize that you're doing something wrong, and that's why you're having a problem. Who knows, keep studying and trying things out and you might find an actual bug and be able to contribute a patch that fixes it.

That will never happen when you use Windows or Mac. Never.

2.2 Knowledge is Power

You know what I really, really like about Linux?

The command line.

I've already mentioned this earlier (and I'm sure that I'll probably write about this some more later), but my experience with computers goes back way before Macintosh made the mouse popular (and alas, necessary). You turned on the computer, and there was just this dark screen with a blinking cursor. If you wanted to make it 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.

I need to get that on a t-shirt.

Why? Because the command line is like real life. There is no undo button in real life. GUIs have made us lazy—lazy at thinking, lazy at figuring things out. Just do it, if you don't like it, just Ctrl-Z. Just throw that document away and leave it in the recycle bin. If you decide you want/need it later, you can just drag it on out of there.

With a GUI, that “undo” button is always an option.⁸ But in real life, you can't unmake a mistake. Sure, you can recover from a mistake, but

⁸Except for the rare occasion when it isn't. Those times are fun.

you are going to have to do some scrambling, my friend, and if you are at least halfway intelligent, you will definitely think twice about trying that again, or at least trying it *that way* again. You don't want to jump through all those hoops again, so you think about your end goal and try to develop a better workflow for next time.

The command line, in short, makes you think. It makes you plan, it makes you think about the end goal, it makes you remember past failures. The command line makes you think about *outcomes*.

A GUI only makes you think about the next step. Surely all the steps after that will be obvious, *n'est ce pas?* I've seen a lot of people ask questions online where they just want to be told which button to push. They are asking about how to cross the street when what they really want to do is get across town.

2.3 The Unix Philosophy

The Unix Philosophy was originated by Ken Thompson (one of the creators of Unix, upon which Linux is based) and basically says that each program should do one thing and do it well. (There is more to it than this; if you are interested, you can always google it.⁹)

This runs counter to physical life, where everything has to be a Swiss army watch. Watch any ad for a new kitchen gadget and this device does *everything* except walk the dog and take out the trash. If it *actually* did all those things and did them well, I would be happy to own one and more than happy to pay a couple of hundred dollars for it.

Unfortunately, it seems that it's impossible to build a device that will do a large number of things really, really well. I like to cook and so my parents inevitably give me a cooking-related gift every Christmas and every birthday. One year, I received a mandoline-type device—if you've spent any time watching the Home Shopping Network¹⁰ I'm sure you've seen

⁹Searching for something on the internet is *always* an option these days, and so many people seem to be unable to do just that. Honestly, this is the kind of stuff that gets my underpants in a twist.

Question: "Where can I find X?" Answer: The same place I would find it: At the other end of a google search.

Better question: "Which is the **best** source for X? Ah, *now* we have the basis for a discussion.

¹⁰Which is now called "HSN". Apparently, we are too busy to pronounce those two extra syllables. Modern life may be difficult, but I don't think the energy I save from not pronounc-

them. It's basically a plastic tray with different cutting inserts, a handle to hang on to the food item, and a box that snaps on to the bottom to hold whatever you are slicing.

I absolutely *love* this thing for slicing potatoes, and since I spend each autumn and winter making scalloped potatoes or au gratin potatoes, it sees a lot of use during those months when the days are short. It does a fantastic job slicing potatoes into a uniform thickness and does it far more quickly than I can do it with a knife.

It also includes inserts to make waffle slices (if you rotate the potato 90 degrees on each pass, you're supposed to be able to make waffle fries), inserts for dicing onions, and so forth. But here's the thing: as great as it is at slicing potatoes (and also carrots, which have the same general hardness as potatoes), it does a terrible job at slicing anything else. Basically, the thin and the thick slicing inserts work well for potatoes and carrots, and all the other inserts don't work at all for them, and any other vegetable just doesn't get cut or gets crushed because you have to hold onto it so hard.

I don't know how much my parents spent on this thing, but if it's anything north of \$20, that's a lot of money for something I can already do fairly easily (and actually enjoy doing) with a sharp knife. Don't get me wrong—I love the thing (even though it's a bit of a pain to clean), but if we had spent at least as much time and money engineering the thing as we did marketing it, we might have concluded that it would probably be better to just encourage people to buy decent knives and then teach them how to sharpen them and use them properly.

(Also, I'm not picking on the Home Shopping Network¹¹ because everybody does this. Monty Python, all those years ago, even had a skit about this, which you can find if you google "simpsons individual stringettes". Of course Monty Python was making fun of this tendency and 50 years later we just accept it as a part of life.)

2.3.1 Where does the Unix Principle actually apply in real life?

Kitchen knives — I know that technically, you can use about any knife to cut a tomato or chop an onion. No debate there. But certain types of knives are just better at some things than other types of knives are. Slicing

ing those two syllables is going to give me enough energy to overcome it.

¹¹Okay, "HSN".

bread is a perfect example. You can slice a loaf of bread with any knife, but a long, fairly rigid, serrated blade works best.

Most people have too many knives, myself included. (Again, this is due to marketing and FOMO—fear of missing out.) But really, you only have three or four basic tasks that you use a knife for:

- Slicing bread
- Cutting meats
- Peeling fruits and vegetables
- Cutting fruits and vegetables
- Cutting cheese¹²

Sure, there are always those oddball tasks that you have to do once a year or less, like using a butcher's cleaver to torque open a pumpkin, but lets talk about those tasks you do on a weekly basis. So let's talk about the knives I have that will accomplish those tasks:

- **Slicing bread** — The afore-mentioned serrated knife with a long, fairly rigid blade.
- **Cutting meats** — If I am cutting meat from the bone, I'll use a boning knife, which has a slightly curved blade that narrows at the tip. Kept sharp, it will also do a great job of cutting up meat for stir-fries, or trimming the fat off the edges of pork chops.
- **Peeling fruits and vegetables** — I don't use a knife for this, because a T-handle peeler is much more efficient, easier on the wrist, and safer.
- **Cutting fruits and vegetables** — Most fruits and vegetables tend to be firm enough that any sharp knife will do. I have a long chef's knife with an 8" blade which is great when I'm chopping a lot of something. (I do wish I had one with a 10" blade, because that would be even more efficient.) I also have a small santoku knife with a thin blade which I prefer when I need to slice something into thin slices, such as cucumbers. It also does a great job of cutting meat into tiny pieces, especially if you put the meat in the freezer for 20-30 minutes first.
- **Cutting cheese** — I have a knife with a 4" blade with holes in it. The idea is that holes prevent the cheese slices from sticking to the

¹²Yes, I know I said "three or four" and then immediately listed five. Bear with me.

cheese. I had an expensive (\$15!) that did all right; it also had a two-prong fork on the end to pick up the slice of cheese with. It did an okay job. I now have one with a shorter, thicker blade that does an excellent job. I paid \$4 for it at Menard's. It simply outperformed the more expensive knife.

I know that someone out there is itching to point out that tomatoes aren't potatoes and potatoes aren't carrots, *ad infinitum*, and thus the Unix Principle doesn't apply. Well, there are a lot of different pdfs out there, as well, and if I'm skilled with a command line application (such as pdf tk) it doesn't really matter which pdf I'm dealing with. The job of a knife is to cut. If you keep the knife sharp¹³ and *learn how to use it properly*, you'll be a lot more efficient in the kitchen. Notice the emphasis on learning how to use a tool properly. You can learn a lot just by reading the manual.

Breadmakers — I'm going on a limb here, because a lot of people will be happy to point out that there are sorts of breadmakers will all sorts of settings. Relax. Breadmakers are designed to do one thing: turn flour, water, yeast, and salt into dough, and then turn that dough into bread. All those settings are just options.

I've resisted buying a breadmaker for years, because I actually don't want a device in my kitchen that only does one thing, and I've always known how to make bread from scratch. But as I get older, I don't always have the time or patience to make homemade bread (it can be a messy process), and a breadmaker is ideal. It does one thing, and it does it really well. (Hint: bread machine yeast is your friend.)

Air fryers — Everything I said for breadmakers also applies to air fryers. They have one job: cook food fast and make it crispy. They work really well on certain items (tater tots!) and not so well on others. I have two analog air fryers, and all you really get to choose is time and temperature. I know that fancy digital ones have *programs*, but really, they are just different time and temperature combinations, which means less thinking for you. But I like to think (and I like to experiment) so I am perfectly happy with my little analog air fryers.

¹³Sharpening a knife is a skill in and of itself. If you watch someone that's good at it, you'll realize that it's also an art form. You can get the skill with practice; you can learn the art with long experience. This is something else that you need to learn.

2.3.2 Where else does the Unix Principle *not* apply that it probably should in real life?

Cars (i.e., portable entertainment centers)

Microwave ovens

2.3.3 Where does the Unix Principle not apply in real life and this is actually a good thing?

Instant Pots (and to a lesser degree, multicookers)

Chapter 3

What Are All Those Files in the Linux Root?

If you're using a Linux distro with a GUI (Ubuntu, Puppy OS, Mint, etc.) you land right in your Home folder whenever you click on "Files". But if you've ever gone all the way into the root of your computer (the Windows equivalent would be `C:\`) you'll see a lot of folders¹⁴ there with mysterious three-letter names. Let's take a look at the them and what they contain.¹⁵

3.1 bin

This directory contains essential command binaries that need to be available for all users. Many of these include binaries that bring up the system or repair it. Your basic binaries like `cat`, `ls`, and `mv` live here.

¹⁴Technically, these are *directories*, but let's not be pedantic. In a GUI, the icon usually looks like a folder.

¹⁵For more information about this, consult the Linux Foundation Referenced specifications, which are found at <https://refspecs.linuxfoundation.org/>. You're probably going to want the pdf version of this, which is at https://refspecs.linuxfoundation.org/FHS_3.0/fhs-3.0.pdf. It really is amazing how much you can learn just by reading the specs and manuals.

3.2 boot

Boot loader files. (It's complicated—kernels, and so forth. It is also four letters instead of three.)

3.3 dev

Device files, such as `/dev/disk0/`, `dev/sda1`, etc. Also the home of the wonderful `/dev/null`.

3.4 etc

System-wide static configuration files. It is not allowed to contain binaries.

3.5 home

Users' home directories. If you have multiple users, you will see a directory in here for each user, named after their login name.

3.6 lib

This directory contains libraries that are essential for the binaries in `/bin` and `/sbin`.

3.7 media

Mount points for removable media. Using a jump drive or SD card? This is where they are.

3.8 mnt

Temporarily mounted filesystems.

3.9 opt

Add-on application software packages.

3.10 proc

A virtual filesystem providing process and kernel information as files.

3.11 root

The home directory for the root user.

3.12 run

Run-time variable data. That is, information about the running system since the last boot, such as currently logged-in users and running daemons.

3.13 sbin

Essential system binaries, such as `fsck` and `init`.

3.14 srv

Site-specific data served by this system, such as data and scripts for web servers, and repositories for version control systems.

3.15 sys

This directory contains information about devices, drivers, and some kernel features.

3.16 tmp

Temporary files, natch. (These may also appear in `/var/tmp`.)

3.17 usr

This directory is a secondary hierarchy for read-only user data, and contains the majority of multi-user utilities and applications. This should be shareable and read-only.

3.18 var

This directory contains files whose content is expected to continually change during normal operation of the system, such as logs, spool files, and temporary email files.

Chapter 4

Miscellany

4.1 Is This Really a Hack? Or Is It Just a Tip?

4.2 What I Learned About L^AT_EX 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 links¹⁶? Use the **hyperref** package.
3. The **kpfonts** package has beautiful fonts.
4. Footnotes are easy! (Seriously, footnotes in L^AT_EX 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 (i.e., 5.5 inches by 8.5 inches).

¹⁶Yeah, I know these are irrelevant in a paper document.

7. You can make your top margin larger by using `\addtolength{\topmargin}{0.5in}` but there is not a similar parameter for the 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).
10. Footnotes reset back to the number one with each chapter. To prevent that, add `\counterwithout{footnote}{chapter}` to the preamble.

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 issue¹⁷

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.

4.3 Coda: Why L^AT_EX?

Ahem...

¹⁷Always assuming that there *will* be another issue.