(Did a search engine send you to this .pdf, gentle reader?! It served you poorly. These steps could WRECK YOUR PHONE. Please continue only after soaking up the immortal prose in the related blog post.)

OVERVIEW

Android is based on the Linux kernel, could be regarded as another Linux distro. (Or not.) **System --> About Phone** (or something similar) should get you to a screen showing the Linux kernel version.

The Linux distro you install for yourself on a desktop is open, unlocked; you may transform yourself at will to the all-powerful root user, a la the 'Administrator' in Windows, may wield all powers of Mighty Linux, wisely or foolishly. The Android Linux on your stock smartphone t'aint open at all. You thumb paddle a new high score on *Candy Crush* as a plebe user with restricted permissions: are allowed to install apps, change wallpaper, invoke 'Do Not Disturb,' while barred from uninstalling bloatware, adware, from making larger changes. You own the phone, but do not fully control it. If you want full control, you must mod.

SMARTPHONE MODDING VOCABULARY

A glossary is at the end of this .pdf. PgDn gets you there. I suggest a look-see before continuing.

WHAT I DID

- I researched and bought phones with reps for easy modding.

  Most smartphone makers don't want you to mess with your móvil. They will yank your warranty if you try, engineer in barriers, roadblocks. A few makers are more charitable, perhaps because they hope to curry favor among app developers.

  Search engine terms like "best root friendly phones" led me to the Nexus 6 "Shamu" in 2015 and to HTC OnePlus 5T in 2018 to replace the Shamu stolen in Guatemala City. I paid cash up front for these handsets, did not acquire them as part of a long-term cell phone contract. Money. Hundreds of $ or €. A drawback. Sorry. A frugal modder might consider an older handset with an easy-to-mod reputation.

- I hunted online for device-specific modding guides.

  I'd check XDA Developers first. My choice of easily-modded phones made the step-by-steps almost identical, but I'd never mod without first searching for tips specific to my handset.

- I stuck with Android Platform Tools.

  Kingo Root, Towel Root, Odin and others offer one-click software rooting tools. I gave Kingo a whirl several years ago during fruitless efforts to hack into a Galaxy S4, observed no untoward behavior as it whizzed away ineffectually, but still regard these Swiss army knife mod programs with skepticism. The Platform Tools are 24 Karat Android, are available from android.com in Linux/Mac/Win editions.

- I turned on Developer Mode to enable ADB Debugging. The 5T also had a setting to enable OEM unlocking.

- I downloaded .zips for a custom ROM and GAPPs, and copied them onto my smartphone.

  Which ROM? I am a long-time Lineage OS user, although I took OmniRom for a spin to research this post, and liked the little I saw of it. Which GAPPs? ARM or ARM64, depending on my handset. I choose the Nano version, for no particular reason.

- I downloaded the .apk file for F-Droid, and copied it to the smartphone, too.

- I downloaded the .img file for the device appropriate version of TWRP recovery to my computer.

- I pressed the appropriate buttons to boot into fastboot mode, USB-plugged the phone into the computer, and ran some Platform

Tools commands from a terminal window:

fastboot devices told me that Platform Tools could see my Android handset. Whoopee!

fastboot oem unlock unlocked the bootloader, to allow me to install the above-mentioned TWRP recovery .img file. You will want to note that 'fastboot oem unlock' will WIPE EVERYTHING OFF THE HANDSET!

After fastboot oem unlock had done its thing, I booted again into fastboot mode, and ran:

```
fastboot flash recovery name_of_the_TWRP_img_file.img
```

to copy the TWRP .img file onto the handset.

With this copying completed, I unplugged the cable, booted into fastboot mode once more, and pressed the necessary buttons to get into 'recovery.'

Presto-change-o: a sparkling new TWRP recovery menu appeared, allowing all manner of powers heretofore unavailable.

- Within TWRP: I did *not* try to get root access.

  You could. Most modders do. I have, but decided later that I don't need to be all-powerful (and potentially vulnerable) root user to do what I want to do with a handset. I haven't experimented with Magisk or related root user tools.

- I selected **Backup** to make a copy of the stock OS, just in case. Touching **Name: (Auto Generate)** in TWRP will allow you to customize the backup name.

  Be careful about where you put your fingers on the backup menu screen; if you inadvertently uncheck an essential partition, you will create a backup that can't be restored.

- **Wipe** --> **Swipe to Factory Reset.**

- **Install** --> scroll down to the .zip file for the device-specific LineageOS ROM --> select it --> **Swipe to confirm Flash.** I do *not* 'Add more Zips' to install Gapps. Not for this configuration.

- TWRP now installs the ROM and boots into a bare bones Lineage OS. I find Lineage's built-in file manager, snoop the phone's innards until I locate the previously-copied F-Droid .apk, and long press to install it.

- I now have an F-Droid app store on my móvil, and spend the next many minutes downloading F-Droid apps and configuring OS settings. This will be the de-Googled set-up that will run my móvil ninety-nine hours out of a hundred, except for those rare occasions when I need to use something from Google Play.

- I boot into Recovery and make a Nandroid back-up copy of this fully-configured F-Droid set-up. I may copy the Nandroid to my computer, but always keep a copy on the phone itself, ready for restoration.

- With the F-Droid Nandroid tucked safely away, I select 'Wipe' in Recovery for a Factory Reset, and start from scratch to create a second, Google Play-enabled Android OS. I tap 'Install,' as before, select the LineageOS .zip, as before, but this time also tap 'Add more Zips' to install Gapps, too.

  The phone boots, and now confronts me with a Google sign-in prompt. I sign in to Google, as asked, find the Google Play icon brought aboard with Gapps, and again spend many minutes on configuration, this time downloading signed-in-to-Google-Play apps.

- When everything is just so, I boot back into Recovery to make a second Nandroid of my fully-configured Google Play set-up. I Wipe again (as I'll almost never use the Google Play OS), and tap Restore in TWRP to restore the F-Droid OS made earlier.

- My handset now sports two ready-to-roll Nandroids: for privacy-enhancing F-Droid, and for conventional Google Play.

My location history appears in my Google Account record when I restore the Google Play Nandroid, and disappears when I run the F-Droid version. To swap between the two, I need only load TWRP, wipe, then restore the desired Nandroid. Five minutes.

- Lineage issues weekly updates. If the version hasn't changed, I download the update with my PC, copy the .zip to the handset, load TWRP, touch 'install' -- *without* 'wipe' beforehand -- and wipe cache/Dalvik afterward. Voila! my móvil boots with an updated OS and all apps/settings intact. I remember to Nandroid it, if I want to restore it easily in the future.

  (And if the Lineage OS version *has* changed? From 14 to 15, say? I might try to get away with the ee-zee update just described, but generally prepare for a fresh install.)

ADDITIONAL THOUGHTS

- An ITWorld article describes a de-Googled set-up on a handset without mods. The user need only refuse all Google prompts while setting up a new móvil. Advantage: No technical chops needed! Disadvantage: No separate Nandroid with Google Play apps.

- A MakeUseOf article avers that the era of custom ROMs has come and gone, particularly for avid smartphone shutterbugs. I haven't tried it, but see no reason that a user couldn't strive for a best-of-both-worlds dual Nandroid set-up with the handset's stock Android OS. After installing TWRP, the user could still create separate Nandroids for F-Droid and Google Play versions, without installing a custom ROM or GaPPs. Why not?

- My #1 can't-live-without app is a static map that shows my GPS'd location on a street grid. Surprise, surprise: the GPS signal showed up on both the Nexus and OnePlus 5T with no sim card installed! The privacy minded can enjoy SIM-free GPS guidance in unfamiliar locales, slip the SIM into the phone only when ready to be connected again.

(Map alternatives: Maps.Me offers their .apk, which can be sideloaded like the .apk of F-Droid. The .apk of now-dated RMaps is available, too.)

- I'll treat myself to an only tangenitally related sidebar: now that you've installed Platform Tools, you may want to know that I couldn't tether either the Nexus or 5T's internet connection to a laptop without first following Platform Tools-centric instructions. (I did not have to install the ADB drivers mentioned in the article, at least not with my Linux rig.)

SHOULD I CARE SO MUCH ABOUT PRIVACY?

Consider what you now know about the internet, computers, smartphones, surveillance technology. Add what you've read about artificial intelligence, robotics, technology on the horizon.

Pause, please. Look away from the screen.

Ask yourself a question:

Is the technology of ten, twenty, thirty years hence likely to want to know more about us, or less?

More, of course. Much more. I get along comfortably today without a Siri-like virtual assistant, but that's because VAs are still relatively primitive. What happens when they improve? I'll likely crave a VA, too, and may not be able to get it without sharing a heretofore sacrosanct chunk of my private life with Big Data's petabytes.

What's in store in 2025, 2030? Are all the AI and Big Data start-ups likely to close up shop, abandon work on privacy-intruding tech that will automate what we now endure as drudgery?

Please don't get me wrong. I wouldn't have hacked my phone and written this post if I didn't regard privacy concerns as important. But I have to put the struggle in perspective. It's one thing to try to keep water out of the canoe when it's in storage, another when it's knifing through rapids. These challenges will continue.

I know a tech here who works professionally with GPS, who joked that he could dupe my móvil into thinking itself in Paris. He doesn't worry about the matters brooded about in this post. "Modern life," he seemed to shrug. He didn't confront me with the depths of my own ignorance, but I'll shine a spotlight on it anyway. I am a lay user. I lack any technical understanding of how the OS interacts with the hardware, how the chips work or how they are made, how the handset communicates with cell phone towers. I hitch a clueless consumer's piggyback ride on tech that is absolutely out of my league. Some could say I am lucky to merely exist in the same era as such wonders, to be able to use them at all.

You could decide to care less than I do, to judge comfort with privacy intrusion as an adaptation to twenty-first century life. I don't want to adapt that way, don't think I should have to. I presume that the U.S. Congress is too thoroughly compromised to protect consumer privacy, and continue to look with hope to the European Union.

SMARTPHONE MODDING VOCABULARY

Unlocked SIM: This means only that a handset is not locked to a single cell service provider. Said handset is likely stock, unmodded.

Developer Mode: A hidden Android menu that permits special system changes. **Settings --> System --> About Phone -->** (or something similar) should offer up a page that includes an entry for **Build number.** One enables Developer Mode by tapping Build number seven times. I do only what needs doing in Developer Mode, then disable it.

Root: A "rooted" handset enables Linux's root user powers.

Bootloader: You might never have seen it, but your handset employs the bootloader at start-up to, well, make Android boot. The modder unlocks the bootloader to install stuff. Some bootloaders are easily opened; others are off-limits to amateurs.

Fastboot: The correct button-presses on start-up will fire up a Android handset in bare bones fastboot mode. (Example: Press and hold "volume up" [or "volume down," on some handsets] then press power.) A modder may do stuff in fastboot mode via computer and USB cable. The Fastboot menu also can be used to start Android's ...

Recovery Mode / Recovery Environment: ... which could be seen as a separate mini operating system, intended for maintenance tasks. The stock Recovery Environment is feeble. Nearly all modders eagerly install TWRP or another custom Recovery Environment, which grants all manner of nifty new powers: to wipe the phone, to make "Nandroid" backups, to install new operating systems. I regard access to TWRP's powers as the biggest perk of modding.

Android Platform Tools, including **ADB** and **FASTBOOT:** Free-for-the-downloading Android tools widely used by modders, far more comfortably and quickly installed than the complete "Android SDK" used by developers.

Custom Rom: Linux users choose among variation-on-a-theme Linux *distros*: Ubuntu, Fedora, Debian, hundreds of others. A custom ROM is an Android distro. Fewer are available, and most of the attention now seems to go to only one: LineageOS. Alternatives include OmniROM, Paranoid Android, Bliss Rom.

Gapps: "G" for Google, "apps" for applications: a .zip file including some or many Google applications usually found on an Android phone. A modder may want nothing to do with Gapps, or may choose to install them on the heels of a custom ROM.

Nandroid: A backup, made easy through a Custom Recovery like TWRP.

APK File: The installation file for an Android program obtained AWOL of Google Play. I am interested only in the .apk for F-Droid. Apkpure, Apkmirror and other sites offer the .apks of household name apps, but I do not know if the "sideloading" of these apps is always legal, and am more skeptical still that downloaded .apks are always safe to use.