Exploring Android
By Dick Maybach, Member, Brookdale Computer Users' Group, NJ
November 2017 issue, BUG Bytes
www.bcug.com
n2nd (at) att.net

The most popular cell-phone operating system, Android, is more like a tool kit than an OS. Cell phone vendors readily change not only its included applications, but also its structure, to differentiate their products. As a result, there is more variation among devices than Windows users are accustomed to. (Windows on a Dell is almost identical to Windows on a Lenovo, but Android on a Samsung may differ substantially from Android on a Nexus.) This doesn't concern most cell-phone users, who have little interest about what goes on inside their devices, but computer enthusiasts are more curious. In this article, I'll use my own phone, a Motorola G^4 using Android 7, as an example, but yours is most likely different. You should not study my results, but use the tools I describe to explore your own device.

Smart phones are far more powerful and complex than early PCs, and one area you will want to explore is its hardware; the app My Device by 3k Developers is useful for this. Figure 1 shows the areas you can learn about. (This is the screen on my phone; yours probably differs.)

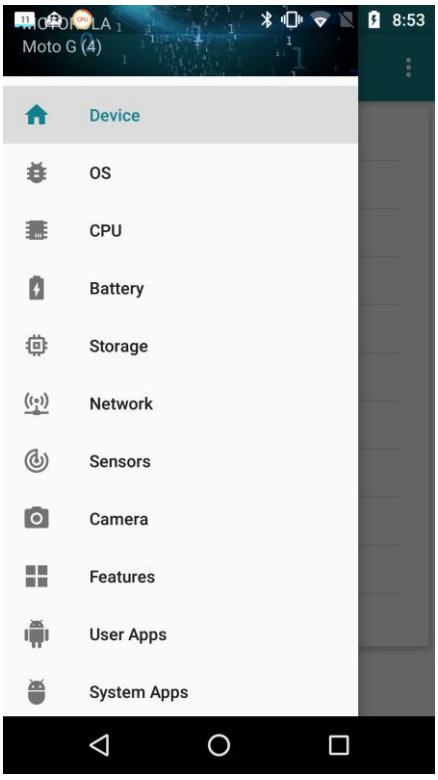

Figure 1. My Device Home Screen.

Figure 2 lists the CPU characteristics; it's eight cores and 2 Gbytes of RAM far exceed the resources available to desktop PCs of just a few years ago. This hints at the potential hidden in the small device you carry in your pocket.

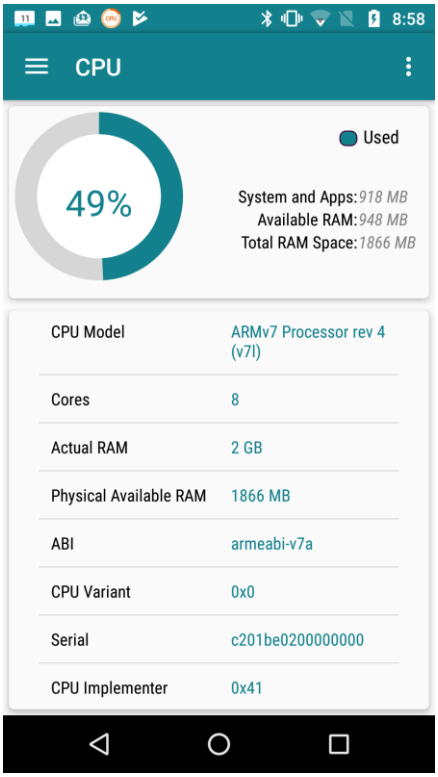

Figure 2. CPU Characteristics.

The camera is another interesting device, as phone cameras probably take far more pictures than dedicated ones. Figure 3 shows what Motorola included. Here too, the front camera's 13-Megapixel resolution is higher than most digital cameras of just a few years ago.

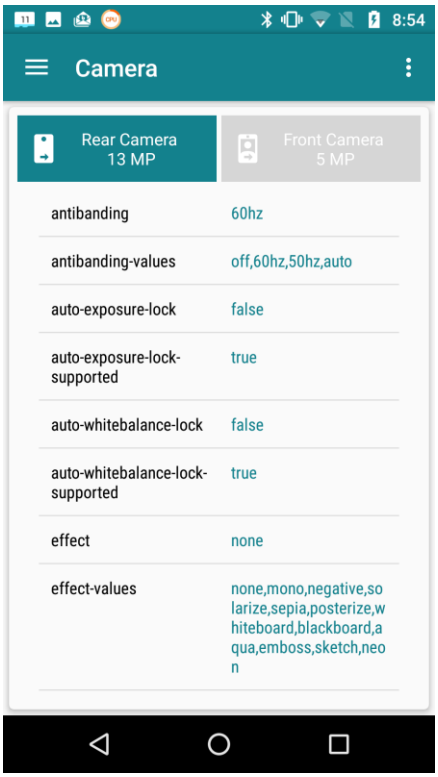

Figure 3. Camera Characteristics.

As you can see, the My-Device app provides much information about the capabilities of your phone's hardware and, although I haven't shown it, its software.

It's important to properly configure your phone, especially the resources that each app can access. Motorola included the Settings app for this, and although your vendor may have given it a different name, you have something similar. While My Device gives you detailed information on your phone, Settings lets you configure it. Figure shows a portion of its home screen. Taking the time to explore the entries will not only tell you about your phone's features, but will let you configure it to your liking, rather than to what its manufacturer might prefer.

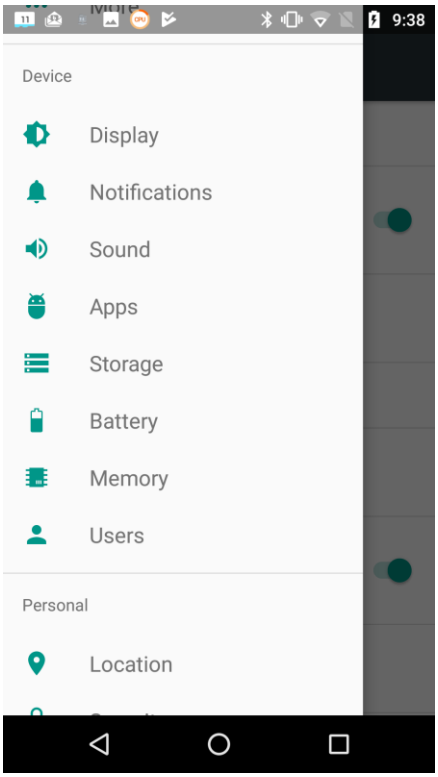

Figure 4. Settings Hone Screen.

Exploring the Apps item is especially useful. For example, Figure 5 shows a portion of the email app BlueMail. As you can see, it can access my calendar, contacts, phone, and storage, but I've silenced its notifications. (I prefer that my personal devices speak only when spoken to; I don't want them breaking into my life at their convenience. As a result, I've silenced almost all notifications.) Some apps by default can access far more than what they need, and it's important when you install a new one to review what information they ask for and to disable what is not appropriate. As in the case of My Device, there is far more to explore in Settings than I have space to discuss here.

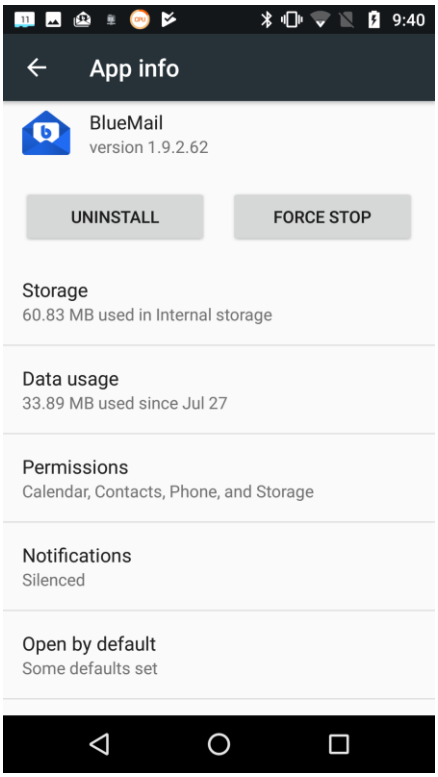

Figure 5. BlueMail App Information.

Unlike in the case of a personal computer, much of the Android file system is not available to users; instead you are limited to exploring your home directory, where your data are stored. (To access the system area, you must do what's called "rooting," but unless you are an experienced Android developer, little but mischief will result.) Of course, you'll use a file manager for this exploration. If your vendor didn't include one, try the ES File Explorer File Manager from ES Global. The directory screen-shots below are from this app. Figure 6 shows my home directory.

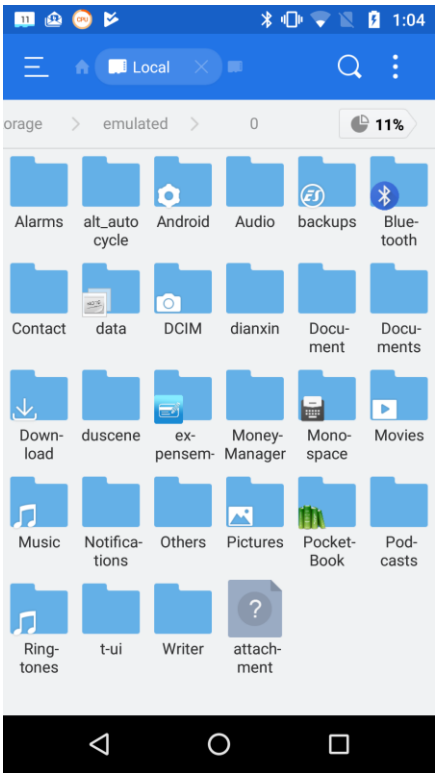

Figure 6. Android Home Directory

On a typical PC, most of the directories in the home directory are created by its user, but this isn't the case for Android, where typically the directories are created either by apps or Android itself. A worthwhile directory to explore is Android, which has two subdirectories, data and media. Figure 7 shows a portion of data's contents.

Figure 7. Contents of Android/data.

As you can see, it's full of subdirectories, most associated with a particular app, which is where the app will keep its data. If you can't find something in your home directory, this is a good place to look.

However, there is a better way to find where a particular app keeps its files. Use it to create a test file, then use the File Analyzer tool in ES File Manager (accessible when you touch the vertical line of three dots in its menu bar). Scroll down on the result to the Recently Created Files area and select Details to see a list of new files (Figure 8).

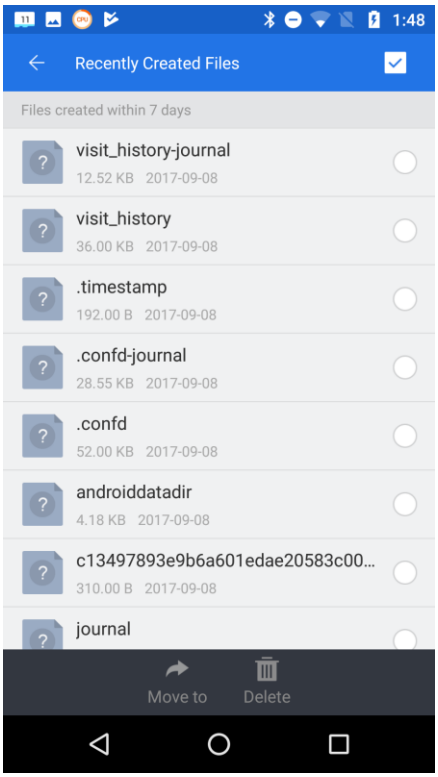

Figure 8. Recently Created Files.

When I selected androiddatadir, I obtained Figure 9.

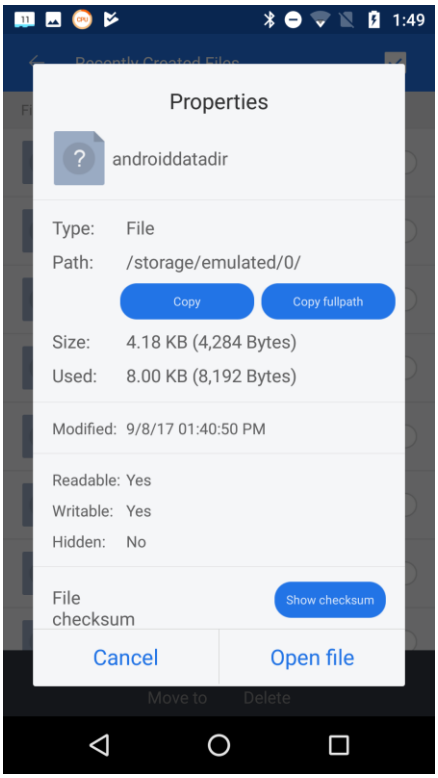

Figure 9. Details on a New File.

The storage/emulated/0/ may be confusing, but it's just the location of the home directory. In this case, since the file we were searching for was in the root directory, finding it by brute force would have been easy. The directory structure on your device will be different of course. (You may have noticed that this file doesn't appear in the listing of Figure 6; I had deleted it before I made that screen-shot.)

For our last expeditions we'll either (depending on your attitude) return to computing's roots or its dark age; yes, we'll be using the command line. You'll first need to install a terminal app, and I use T-UI from Everyone. (Starting T-UI and using the command "tutorial", will take you to a website with a lot more help than you probably want. The command "help" will show commands that have help pages; for example, "help time" will show you what the time command does.) Figure 10 shows a screen-shot with the command pwd (print working directory), which is the complete path to the home directory. It also shows the result of "echo $Path", which displays the directories where Android looks for commands, in this case /sbin, /vendor/bin, /system/bin, and /system/xbin. None of these directories listed by $PATH are accessible with a file manager, unless the device has been rooted.

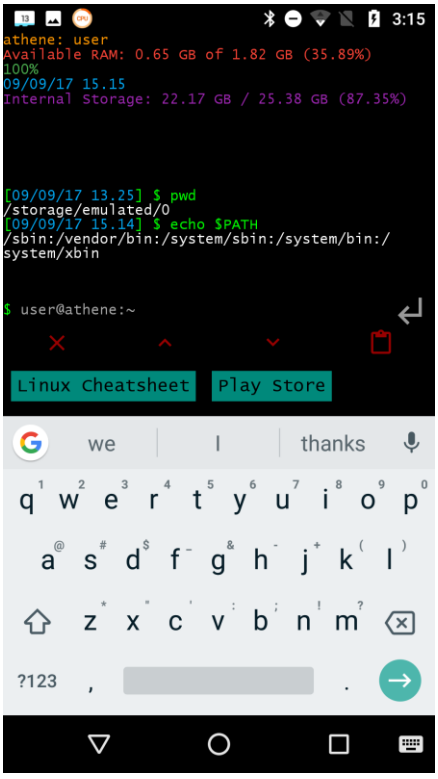

Figure 10. T-UI Screen.

The command "cat proc/partitions" shows all the partitions that exist. (I'm going to skip the explanation of exactly what this command means; the object here is to learn about Android, not become proficient in using the command line.) I won't show the output, as yours will be different. Each line of result (after the header) is a different partition; the first number (major) is the device, the second (minor) the partition on that device, the

third the number of blocks in it, and the last its name. Mine had three different major numbers, showing three separate storage devices, one with a single partition, one with 17, and one with 35. You are probably surprised at these numbers; just because Android devices are small doesn't mean they are simple. However, many of these partitions are not used in normal operation, but only during start-up and recovery.

We obtain a better sense if the storage environment with the "df -h" command, which shows the free space on each partition. (The -h option makes the numbers more readable.) I've formatted the result into the table of Figure 11, where we see that of the 53 defined partitions, only 13 are in use.

File System	Size	Used	Avail	Use%	Mounted on
tmpfs	923M	692K	923M	1%	/dev
tmpfs	923M	0	923M	0%	/mnt
/dev/block/bootdevice/by-name/system	2.3G	1.7G	617M	75%	/system
/dev/block/bootdevice/by-name/userdata	25G	2.8G	22G	12%	/data
/dev/block/bootdevice/by-name/cache	248M	1.6M	241M	1%	/cache
/dev/block/bootdevice/by-name/dsp	12M	3.6M	7.7M	32%	/dsp
/dev/block/bootdevice/by-name/modem	94M	70M	23M	76%	/firmware
/dev/block/bootdevice/by-name/fsg	1.9M	1.9M	0	100%	/fsg
/dev/block/bootdevice/by-name/oem	635M	96M	527M	16%	/oem
tmpfs	923M	0	923M	0%	/storage
/data/media	25G	2.8G	22G	12%	/storage/emulated
/mnt/media_rw/F549-17EF	30G	818M	29G	3%	/storage/F549-17EF
tmpfs	923M	0	923M	0%	/storage/self

Figure 11. Output of the mount command.

Note the data/media file system, which is mounted at /storage/emulated, which we saw in Figure 10 is our home directory. From other explorations, I know that F549-17EF designates my SD card.

Using such readily-available tools as these, you can explore your Android device to learn about it and to make you more efficient while using it.