

NOTE: The article was printed incorrectly in the August issue of Badger State Smoke Signals, and was reprinted correctly in the September issue.

THE BOOT PROCESS

This month I would like to express admiration for the software that gets our computers going. Many of the things that happen when we turn on the power are hidden from our view; they happen quickly and in the background and we really are not aware of them. Nevertheless, there is a fairly complicated series of happenings. A couple of years ago while reading, I jotted down some notes and made a flow diagram of the boot process. That is the subject, and most of the substance, of this article. (The flow diagram appears on the next page.)

There are at least 16 major steps a computer must take after the switch is flipped on, before we see the C:\ prompt on the screen. If we divided each step into sub-steps, it would add up to many more, easily filling all the pages of this month's publication. Even more striking is that most of the steps occur well ahead of loading COMMAND.COM, the traffic cop that takes charge of everything, including watching your keyboard to see if you press a key. Note that COMMAND.COM loaded very late in the process, down at the bottom of the flow chart, just after CONFIG.SYS is loaded and before AUTOEXEC.BAT is run.

That all this happens so quickly and efficiently is a tribute to the gals and guys that wrote the program code for the boot process and operating system. They wrote the programs in assembly language, a tedious process using a language that is just one step away from the binary language (zeros and ones) that the computer itself "thinks" with. The computer translates the human-written assembly language instructions to binary, so that it can understand the commands it is being asked to follow. The same translation process occurs with programs written in PASCAL or C or BASIC. These higher level languages are much easier for us humans to understand and work with, but the programmers who designed and wrote the boot and operating system software chose assembly language because it results in much smaller, efficient programs that run much faster than programs written in higher level languages. Had they chosen BASIC instead, it might take 5 or 6 minutes for the C:\ prompt to appear after turning on the power, instead of a couple of seconds. Think of that next time you flip the switch to the ON position! Happy computing .

