

How can a computer that is devoid of operating system code spring into action and begin executing? It cannot. Somehow a program must be available before a computer can start. On the oldest machines, restarting was a painful process because a human operator entered the initial program through switches on the front panel. Switches were replaced by keyboards, then by I/O devices such as tapes and disks, and eventually by Read-Only Memory (ROM), and Flash memory.

Some embedded devices do more than store an initial program in Flash; they store the entire operating system, which means that the device can start executing immediately after receiving power (e.g., after the batteries are changed or the device is powered on). However, most computers take multiple steps when restarting. When power is applied, the hardware executes an initial startup program from Flash. Although it may include mechanisms that allow engineers to debug the hardware, an initial program is usually quite small — its primary function consists of loading and running a larger program. In a typical personal computer, for example, the startup program powers on devices (e.g., a display, keyboard, and disk), searches the disk for a bootable operating system image, copies the operating system image from disk into memory, and then jumps to the entry point of the operating system.

Computer systems that do not have permanent storage or embedded systems may use a network: the initial startup program initializes the network interface, and then uses the network to download the operating system image from a remote server. Some Ethernet hardware includes firmware and a small onboard processor that can download an image over a network, store the image in memory, and then start the processor running the image.

In some cases, multiple steps are used. The initial startup program loads a larger, more powerful program, which then loads an operating system. Startup programs arranged in a sequence to load ever larger startup programs are often called *bootstraps*, and the entire process is known as *booting* the system.† Older names for the bootstrap process include *Initial Program Load (IPL)* and *cold start*.

22.3 An Example Of Booting Over A Network

The Galileo platform provides a specific example of a multi-stage bootstrap. Our lab at Purdue is designed so systems boot over a network. Unfortunately, a Galileo board does not have the hardware and software needed to boot over a network. Instead, when it receives power, a Galileo board runs an initial bootstrap program from the onboard Flash memory. The initial bootstrap, supplied by the vendor, can search local devices, find a specified image, download the image into memory, and start the processor executing the image. In particular, the initial bootstrap program can be configured to search the micro SD card for an image.

To enable network booting, we created a second bootstrap program that includes a network stack plus code to use the network to download an image. We placed a copy of the program on the SD card in each Galileo board. To boot a Galileo board, we power-cycle it. We built a special hardware system to handle the task: the system re-

†The terminology derives from the phrase “pulling one’s self up by one’s bootstraps,” a seemingly impossible task.