Hello, How Do You Reboot?

Paul S. Wang, Sofpower.com

July 31, 2024

Oh, no it froze again! I know how annoying it is when your computing device suddenly stops working, not responding to any input, from the mouse, touch screen/pad, or keyboard. You could call the help desk but that would be troublesome and very time consuming. Often, the help desk would suggest that you **reboot** your device.

So what is rebooting? When is it needed? Why does it work? How do you initiate a reboot when the device is stuck? What's its implications? We are going to talk about these and related issues so that you can quickly get unstuck and carry on with your important work.

This article is part of our *Computational Thinking* (CT) blog. You can find other interesting articles in *aroundKent* (`aroundkent.net`), an online magazine, and also in the author's new book

Becoming A Computational Thinker: Success in the Digital Age

See the website `computize.org/CTer` for more information.

What is Booting?

In computing, the term "*boot*" refers to creating an operating system (OS) when you turn on a computer from a powered-off state. An OS is a critical program required to make a computer work. Without an OS, any computing device is just a brick. If a computer were a physical person, then the OS would be its brain.

Computers come in many forms: desktop, laptop, tablet, smartphone, servers in the cloud, and other devices such as WiFi routers and smart TVs. The same computer hardware with a different OS would literally be a different

computer. Popular operating systems include Windows, macOS, Linux, iOS, and Android.

Under the control of an operating system, a computer can execute any **app** (application program) stored in its memory. Popular apps include web browsers, word processors, PDF readers, video/audio players, online chat/conference software, and so on.

But when a computer is first powered on, its RAM (memory) is basically blank space. There is not yet any program code in it. The first thing the computer hardware does after being powered on is to load a *boot loader* program to begin the building of the OS which is done in stages. This "coming into being" process has been described figuratively as "*pulling yourself up by the bootstraps*." Hence we have the term *bootstraping* or *booting* for short (Figure 1).

Figure 1: Bootstrapping

While it is impossible to pull yourself up by the bootstraps physically, it is actually just what happens when an operating system gives birth to itself.

Booting Actions

When a computer is powered on, it first performs the hardware *Power On Self Test* (POST) then runs the booting code to establish the full OS in a sequence of well-defined stages. Here is a simplified overview you can read through quickly. There is no need to dwell on details.

Figure 2: Boot Steps

1. **Power-On Self-Test (POST)**: When the power is turned on, the computer's power supply sends power to hardware components. This triggers the POST routine which performs basic checks on the hardware components, such as RAM memory, CPU, and storage devices, to ensure they are functioning correctly. If any critical hardware issues are detected, the POST process may halt and emit error codes (beep codes) or display error messages.

2. **Hardware Initialization**: Then comes initialization and configuration of system hardware, including CPU clock speeds, power management, and peripheral interfaces. This step also identifies and configures bootable devices (e.g., hard drives, SSDs, USB drives).

3. **Boot Loader Execution**: Next, the *boot loader*, a small program responsible for loading the operating system, is run. There may be multiple boot loaders in case alternative operating systems are made available on the same computer.

4. **Operating System Kernel Loading**: The boot loader locates the operating system *kernel* and loads it into memory. **The kernel is**

the most central and important part of the OS. It is responsible for managing hardware, system resources, providing an indispensable interface between hardware and software (Figure 3). Without the OS kernel, there would be no connection between any software to the hardware.

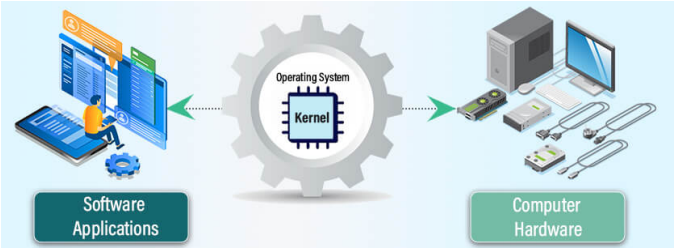

Figure 3: OS Kernel

5. **Kernel Initialization**: Once loaded, the kernel initializes hardware components, sets up memory management, and starts essential system services. The kernel mounts the root filesystem, which contains the core system files and directories needed for the operating system to function. For example, the root filestem is `C:\` on Windows, and `/` on Linux and macOS.

6. **Starting System Services**: The kernel starts one or more *master processes*. A master process is always present and controls and manages all other programs and services provided by the operating system.

7. **User Space Initialization**: The master process launches various background services required for system operation, such as networking, logging, and device management services. These background processes are also called *daemons* because they are hidden from users but always active. Then, the system enters a multi-user state.

8. **User Login and Session Start**: Users can log in via graphical or command-line interfaces. Upon login, user-specific settings and environments are loaded, and a user *session* is started, typically involving the launch of a desktop environment or command shell.

9. **Full System Readiness**: The system is now fully operational, with hardware and software components initialized and ready for use. Users

can start applications, perform tasks, and interact with the system as needed.

As soon as booting completes, your device becomes fully functional. As you use your device, there will be occasions when the booting process, or a part of the booting process, will need a redo, that is *rebooting*.

When Is Rebooting Required?

Rebooting can be triggered by the system or initiated by the user in various situations:

- Hardware Changes: Adding or removing hardware components like memory, CPUs, or other peripherals might necessitate a reboot. Changes in BIOS/UEFI settings can also require a reboot.
- Operating System Updates and Setting Changes: Kernel or critical system updates often require a reboot to take effect.
- Adding/Removing/updating Apps: Certain software installations or removals may require a reboot, especially those affecting core system libraries or services.
- Performance Issues: Severe system performance degradation, memory leaks, or resource exhaustion can sometimes be resolved by rebooting the system.
- System Slows, Crashes, or Freezes: If the system becomes unresponsive due to a kernel panic, severe application crash, or other critical errors, a reboot may be necessary to recover.
- Power Supply Issues: Power failures or issues with the power supply can cause an unexpected reboot.

Intentional Reboot

If a reboot takes place by itself, then a user just have to wait for it to complete. In case you wish to suspend (put to sleep), restart, or shutdown your system, you would normally use the *power icon* (Figure 4) which can be found easily on most systems.

Figure 4: Windows Power Icon

Furthermore, if your system supports voice input, you can also give verbal commands to restart/reboot and do other things. For example, on Windows you can use:

```
"Hey Cortana, lock my computer."
"Hey Cortana, shut down my computer."
"Hey Cortana, restart my computer."
```

You can also find such commands on other operating systems.

Figure 5: Restart in Progress

Reboot To Get Unstuck

When a system is stuck or unresponsive, instead of the power icon, you need a different way to trigger a reboot/restart.

There are two types of boot, cold boot and warm boot. The former is starting the computer from a completely powered-off state. The latter is

restarting the computer without turning off the power, usually done by the operating system's *restart* function. Restart is faster because it bypasses the POST and hardware initialization steps.

Here are several methods for users to initiate a restart/reboot, depending on the level of responsiveness of the system:

1. Use Keyboard Shortcuts

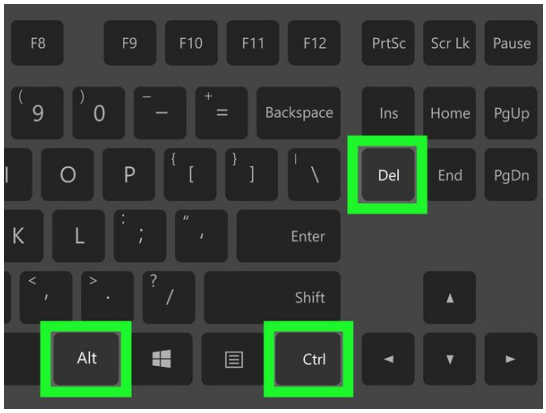

Figure 6: Simultaneous Key Combination: Ctrl + Alt + Del

On Windows systems, pressing Ctrl + Alt + Del (hold down Ctrl and Alt then press Del, Figure 6) brings up a screen with options to open the Task Manager, log out, restart, or power off the computer. On Linux systems, this or a similar key combination will also work.

On Apple macOS, pressing Ctrl + Cmd + Power button causes a restart, and pressing Ctrl + Alt + Cmd + Power button quits all apps then shut down the system.

2. Use Hardware Buttons

   Soft Power Off: Press and hold the power button for a few seconds (usually 5-10) to perform a soft shutdown. This method signals the operating system to shut down gracefully.

   Hard Power Off: If the system does not respond to a soft power off, pressing and holding the power button for a longer period (10-15 seconds) forces a hard shutdown. This immediately cuts the power to the system, which should be used as a last resort due to the risk of data loss.

3. Remove Power Supply (last resort)

   For desktops, unplug the power cord. For laptops and smartphones, press and hold the power button for a sufficient long time to force a shutdown. Alternatively, remove the battery, if possible. This method is a last resort because it can lead to data corruption or loss.

A reboot/restart usually takes a few minutes. It may take longer when doing system updates.

Problems Rebooting Can Solve

Now we know how to reboot. Let's list the types of problems a reboot can solve:

- *Performance Issues*: Slow system performance, lagging applications, and unresponsive user interfaces can often be improved with a reboot.
- *Memory Leaks*: Applications or processes that consume more memory over time due to poor memory management can be reset, freeing up memory.
- *Software Crashes and Freezes*: Applications that crash or freeze can be resolved as a reboot closes all running processes and starts them afresh.
- *Network Connectivity Issues*: Problems with network connections, such as no internet access, dropped connections, Bluetooth pairing problems, can sometimes be resolved by resetting network interfaces through a reboot.
- *Hardware Malfunctions*: Issues like unresponsive peripherals (mouse, keyboard, printers), malfunctioning USB ports, or graphics card problems can be fixed by reinitializing the hardware.
- *System Instability*: Random reboots, blue screens (Windows), kernel panics (Linux), or system crashes can often be resolved temporarily by a reboot.
- *Software Updates*: Completing the installation of updates for the operating system, drivers, and applications that require a system restart.

- *Clearing System Errors*: Miscellaneous system errors, such as those affecting the file system or specific applications, can be cleared out by a reboot.

Basically, a reboot can usually get you out of trouble and solve various problems. But if your system gets stuck during the reboot process time and again, that is serious. In that case, you may need to **reboot into safe mode** in order to fix or restore your system. Better call the help desk.

Why Booting Solves Problems

Rebooting a computer/device can solve a variety of problems by essentially resetting the system, clearing out temporary states, and starting fresh. Here' s why a reboot/restart can be effective and the types of problems it can address:

- *Gets you unstuck*: A reboot/restart returns your system to a working state and gives you a fresh start.
- *Frees memory and releases resources*: Rebooting clears the system's RAM, which can get cluttered with fragmented or corrupt data over time. This helps eliminate memory leaks caused by poorly written programs. It also frees up system resources that have been allocated to applications and processes that are no longer running, thus improving performance.
- *Clears Temporary Files and Caches*: A reboot can clear out temporary files and caches that might be causing conflicts or consuming unnecessary disk space. Corrupt cache files can lead to unexpected behavior in applications and a reboot clears these caches, forcing them to rebuild in a fresh state.
- *Resets Hardware Components*: Hardware components like network cards, USB ports, and graphics cards are reinitialized, which can resolve issues caused by hardware that is in an inconsistent state. Rebooting can resolve device conflicts where two hardware components are trying to use the same resources. Think of it as two individuals trying to access the same restroom.

- *Restarts Services and Processes*: Essential system services and background processes are restarted, which can fix issues caused by services that have crashed or are in a failed state. It also brings the operating system back to a known, clean state, resolving inconsistencies that can develop over long uptime periods.

- *Applies Updates and Patches*: Rebooting is often necessary to apply updates and patches, especially those affecting the kernel or core system libraries. Many software installations or updates require a reboot to complete the process and ensure the new software functions correctly.

Figure 7: Clean Slate

Basically, a reboot/restart wipes the slate clean (Figure 7) and gives you a new start.

Reboot and Apps

Often, multiple apps are running when a system reboots/restarts. The very first thing is that these apps are asked to quit by themselves or be closed forcefully.

A reboot/restart gives you a clean slate. But all previously running apps are also gone, unless they are restored. Different operating systems offer their own ways to support the restoration of apps after a reboot/restart. This feature, known as *session restore*, is popular with users.

To enable session restore: on Windows, set the `app restart` option in the settings menu and also via the task manager; on macOS use system preferences; on Linux use features in the GNOME or KDE desktop environment. No such feature is currently available on smartphones.

When an app is stuck or not working properly, you may not need a whole-system reboot. Try to quit and start that single app. To kill (force quit) an app you can use the task manager or commands (`kill`, `taskkill`) on major operating systems.

Reboot Implications

We see how a reboot often can fix things. But we also realize a reboot can break things, **because it may apply a faulty update or some program may stop working in the updated environment**.

According to a CNBC news article "*How a software update from cyber firm CrowdStrike caused one of the world's biggest IT blackouts*" (July 19, 2024):

Figure 8: IT Outage Caused Air Travel Chaos

> "*A fault with an update issued by cybersecurity company CrowdStrike led to a cascade effect among global IT systems Friday, with industries ranging from banking to airlines facing outages.*
>
> *Banks and health-care providers saw their services disrupted and TV broadcasters went offline as businesses worldwide grappled with the ongoing outage. Air travel has been hit hard, too, with planes grounded and services delayed.*
>
> *At the heart of the issue is Texas-based cybersecurity vendor CrowdStrike. On Friday, the cybersecurity firm experienced a major disruption following an issue with a software update.*"

Basically, CrowdStrike issued a software update that had a problem. Fortunately, the faulty software update only affected Windows operating systems where a reboot installed it. However, nearly 8.5 million Microsoft devices were disabled, according to a Microsoft blog post.

Summary

Computers are a new type of machines. They are *universal* because they can load and run any app. Hence, computers can achieve any tasks that can be programmed. That is almost anything you can think of.

The universality property comes from the unique ability of loading, storing, and running any program in the RAM memory. The operating system manages a computer and all other apps. It is the first program to be loaded and stored into a blank RAM memory. This *giving birth to itself* process is called **booting** and is performed every time a computer is turned on from a powered-off state.

When an OS installs updates or runs into difficulties, such as getting stuck or becoming unresponsive to user input, a reboot is usually required. And now we understand why.

A reboot can not only return the OS to its original state, but also apply fixes and updates to the system. In our real lives, we can only wish to wipe out errors and mistakes. What we can do is to improve and update our ways so that we can perform better in the future. If we keep track of our own desired self-improvements, we can use that data to do a *personal reboot* just like installing a computer update. Wouldn't that be nice?