module introduction

- Module introduction
- Components of an Operating System
- Supplemental Reading for Chrome OS
- File Systems
- Supplemental Reading for ReFS File System
- Process Management
- Memory Management and Virtual Memory
- I/O Management
- Interacting with the OS: User Space
- Logs
- The Boot Process
- Boot Methods
- Boot Methods Best Practices
- Mobile Operating Systems
- Cindy: Drive and career path

Module introduction

Welcome back. You've learned about the basics of computing with binary and the hardware layer of the computer architecture. Now, it's time to move on to the next layer, the operating system. By the end of this lesson, you'll know what an operating system is, and what makes up an operating system. You also get some hands-on experience with the three biggest operating systems used today. Before we get deeper into operating systems, I'd like to introduce myself to you. My name is Cindy Gouache and I'm a Site Reliability Engineer at Google. The team I work on is responsible for the management and support of Google's entire internal mobile fleet. Android OS, IOS, and Chrome OS. Before focusing on mobile, I was a systems administrator on the Linux team. Before that, I was an Operations Engineer. But like a lot of the Googlers you've met and will meet, I started my career as an IT support specialist. I've been working in IT for seven years now. The first time I can remember interacting with computers was in middle school when my teacher brought them into our classroom so we could create fun video and multimedia projects. It was my brother who brought technology into our house. My parents were immigrants from Vietnam and we didn't have a lot of money growing up. We had to be creative if we wanted to play with a computer at home. I can remember spending hours with my brother as he assembled a computer and I will just ask a million questions. Eventually, I wanted to try and build my own computer, so I gathered up some old parts and save money to buy new components. I finally put all the parts together from what I remembered my brother doing, but it just didn't work. It turns out that I use some incompatible parts. But through a lot of trial and error, troubleshooting, and long search sessions on the Internet, I finally got it to work. The feeling I got when I heard my computer boot up for the first time was amazing. Before I knew it, I also worked on computers. I really enjoyed the intense concentration and problem-solving required in IT. But I didn't think a career in tech was even possible back then. Once I got to college, I had to find a job to help pay for tuition and that job was an IT support specialist on campus. That's when I realized that take is actually something I could pursue as a career. Operating systems aren't essential part of IT support. Everyone uses their computer to accomplish something. Whether that's browsing the web, writing a novel, making graphics, playing video games, etc. Whatever the case may be, they need to interact with their operating system to do it. IT support, it's essential to understand how operating systems work, so you can help someone accomplish the task they set out to do. Whether that's figuring out why an application won't start, why the graphics look funny on their video games or anything else. Things can get really messy and challenging, and that's part of the fun. Let's begin.

Components of an Operating System

A lot of us hear the term operating system and think of the interfaces of our desktops and phones, like the menus, buttons and backgrounds. Technically these are part of the operating system but it's a little more complex than that. An operating system is the whole package that manages our computers resources and lets us interact with it. There are two main parts to an operating system, the kernel and the user space. The kernel is the main core of an operating system. It talks directly to our hardware and manages our system's resources. As users we don't interact with the kernel directly. Instead, we interact with the second part of an operating system, the user space. The user space is basically made up of everything outside the kernel. These are things that we interact with directly like system programs, user interfaces, etcetera. When we say operating system, we're talking about both the kernel and the user space. There are hundreds of operating systems out there, but we'll focus on the major ones used in IT, Windows, Mac and Linux. Windows OS is developed by Microsoft and used widely in the business and consumer space. Most pcs you buy come with Windows as the default operating system. PC means personal computer, which technically means a computer that one person uses. But in today's world, PC is more commonly referred to as a Windows computer. So we'll just referred to a PC as a Windows computer from here. Mac OS by Apple is mainly used in the consumer space. If you buy an Apple computer, it'll come with Mac OS preloaded. The last operating system will dive into as the Linux operating system. Linux is an open source operating system which means its software is free to share, modify and distribute. Linux is used heavily in business infrastructure and in the consumer space. Linux itself is actually a kernel developed by Linus Torvalds. Because of the way it evolved, we call the kernel the Linux operating system. Today, Linux has become a huge community effort with developers all over the world contributing to its success. Because Linux is open source lots of different organizations package their own version of it. Operating systems like Windows or Macintosh, on the other hand, are solely developed by the respective companies. We call these different Linux OSs distributions. Some common Linux distributions are ubuntu, debian and red hat. Another operating system that has started to gain popularity is chrome OS but we won't go into detail on that one. Mobile phone usage around the world is more prevalent than desktop computers. One cool thing to call out is that chrome OS and android OS both run the Linux kernel underneath the hood. So there's a chance you've already worked with Linux and didn't even know it. There are lots of operating systems out there and they all share common characteristics. If you're able to understand the basic building blocks of one OS, you can apply that to any operating system and understand how it works. In IT support It's super common to work with many different operating systems, desktop OSs, to smartphone OSs and more. Before we get there let's do a rundown of the basics. The kernel does file storage and file management. You can compare it to a physical office file where we store data in paper form.

A computer file is just data that we store and a file can be anything, a word document, a picture of song, literally anything. A file system is how we manage these files, just like in an office we use a system to store our files. We don't just put all our files in one cabinet that would be seriously messy. Instead we organize those files in folders or directories to make them easier to find. There are lots of different types of file systems which will cover more in depth in future videos. Another important function of the kernel is process management. We have many programs that we want to run on our system. To run them we manage the order they run in, how many resources they take up, how long they run etcetera. Our kernel helps us do this with its process management capabilities. For example, you've probably used your computer to do several tasks at once. Maybe you writing a text document while listening to music or playing a video. The process scheduler is part of the kernel that makes this multitasking possible. It switches the execution of each different process on the CPU faster than you can blink and it gives you the illusion that things are happening simultaneously. Next up is memory management. Our kernel optimizes memory usage and make sure our applications have enough memory to run. The last important function that a kernel performs is input, output or IO management. This is how our kernel talks to external devices, like disks, keyboards, networks, connections, audio devices and more. IO management is anything that can give us input or that we can use for output of data. If you've ever saved a file to disk, click the mouse button or used a microphone when video chatting with a friend, you've got the kernel's ability to manage IO to think. And that's the basic rundown of the main functions of the kernel file management, process management, memory management and IO management. Finally, we'll talk about the other component of an operating system, the user space. The user space is everything outside the kernel. These are the things that we interact with directly, like programs such as text editors, music players, system settings, user interfaces, etcetera. By the end of this module, you'll hopefully have a solid understanding of all these functions of an operating system. Let's start by taking a deeper dive into the kernel's file management

Supplemental Reading for Chrome OS

Another operating system that has started to gain popularity is Chrome OS, which you can read about in more detail here.

bro also check out Vanilla os

one of the most user friendly linux distros, it started as an ubuntu flavor but is becoming its own thing using the debian base, kindof link how ubuntu does.
also it uses containers of linux to run native code for any os. say u want to use aur programs, use that, u want rhel apps? gotchu running that nativly. o regular flatpacs? YES PLEASE
PS: Chromebook keyboards suck so bad that its legit the only reason i refuse to use them. BUT u can now install it on any os :D

Heads up: A big part of being successful in an IT role is the ability to be a self-led learner -- someone who finds key resources and reads up on the latest tech trends and solutions. The supplemental readings we've provided have been designed to show you just some of the support materials available to you online; they're not meant to be considered a comprehensive list.

File Systems

Imagine if you had to store a single file in a cabinet. That sounds so bad. What if instead of one file you had to store 100,000? Can you see a problem here? Well, on our computers we can easily store hundreds of thousands of files, if not more, problem-solved. Not quite. We have to be able to keep track of all these files. The kernel handles file storage and file systems on our machines. In this lesson, we're going to dig a little deeper on how it does that. There are three main components to handling files handlers, the file data, metadata, and file system. Let's start with the file system. Well, we have a brand new hard disks that we want to store data on. We need to erase and configure the disk. This way operating system can read and write data to it. This is important since it's how our operating system keeps tracks of files. We must know what kind of file system is used. There are lots of file systems and are used for different purposes. Some file systems support the storage of large amounts of data others only support small amounts. They can operate in different speeds and have varying resiliency towards file corruption and so on. We won't get into which file system is best. That's for you to decide. But the major OS manufacturers have their own unique file systems that they recommend, for Windows, the major file system that's used is NTFS. It was introduced in the previous version of Windows OS, Windows NT, and includes many features like encryption, faster access speeds, security, and more. Microsoft is developing another file system called ReFS, but it isn't quite ready for consumer use just yet. For Linux, different distributions will use different file system types. A standard for file systems for Linux is EX T4, which is compatible with older EXT file systems. In general, different file system types don't play nicely with each other. You might not be able to easily move files across different file systems depending on the file system type. A good guideline to use is just to use the file system that your operating system recommends. Another important part of file management is the storage of actual file data. We write data to our hard drive in the form of data blocks. When we say something to our hard disks, it doesn't always sit in one piece. It can be broken down into many pieces and written to different parts of the disk. Block storage improves faster handling of data, because the data isn't stored on one long piece, and it can be accessed quicker. It's also better for utilizing storage space. Lastly, we need to keep the metadata that contains the information about our file. There's a lot of information about our file that we want to know, who created it, when it was last modified, who has access to it, and so on. The file metadata tells us everything we need to know about our file. It also tells us what type of file it is. A file extension is the appended part of a filename that tells us what type of file it is in certain operating systems. Take cool_image.jpg. JPG is a file extension associated with image files. You'll see different types of file extensions like this. When you're working with your operating system, your working knowledge of file systems and the differences between them is a great skill to have in your IT support specialist toolbox. It can be super useful when you need to do things like recovered data from damaged discs or explore ways to boot from two different kinds of operating systems, like Windows and Linux on the same computer.

Supplemental Reading for ReFS File System

Microsoft is currently developing another filesystem called ReFS, it isn’t quite ready for consumer use, but if you’re interested in learning more you can read more about here.

Heads up: A big part of being successful in an IT role is the ability to be a self-led learner -- someone who finds key resources and reads up on the latest tech trends and solutions. The supplemental readings we’ve provided have been designed to show you just some of the support materials available to you online; they’re not meant to be considered a comprehensive list. Feel free to add to the conversation by posting other useful resources for learners to the discussion forum.

Process Management

One of the most important tasks that our kernel performs is process management. A process is a program that's executing, like our Internet browser or text editor. A program is an application that we can run, like Chrome. Take note of the difference. We can have many processes of the same program running at the same time. Think of how many Chrome windows you can open. These are all different processes for the same program. When we want to run our programs, we have to dedicate computer resources to them, like RAM and CPU. We only have a finite amount of resources, and we want to be able to run multiple programs. Our kernel has to manage our resources efficiently so that all the programs we want to use can be run. Our kernel doesn't just dedicate all of our computers resources to one process. Our system is actually constantly running multiple processes that are necessary for it to function. Our kernel has to worry about all of these processes at once. What a program wants to run, a process needs to be created for it. This process needs to have hardware resources like RAM and CPU. The kernel has to schedule time for the CPU to execute the instructions in the process, but there's only one CPU and many processes. How is the CPU able to execute multiple processes at once? It actually doesn't. It executes processes one-by-one through something known as a time slice. A time slice is a very short interval of time that gets allocated to a process for CPU execution. It's so short that you don't even notice it. It's super short. The CPU executes one process in milliseconds, then executes another process, then another. To the human eye, everything looks like it runs simultaneously. That's how fast the CPU works. If your computer is running slowly, and your CPU resources are being maxed out, there can be many factors at play. It's possible that one process is taking up more time slices than it should. This means that the next process can't be executed. Another possibility is that there are too many processes that want CPU time and the CPU can't keep up with them. Whatever the case may be, even though the kernel does its best to manage processes for us, we might need to step in manually from time-to-time. The kernel creates processes, efficiently schedules them, and manages how processes are terminated. This is important since we need a way to collect all the previously used resources that active processes were taking up and reallocate them to another process.

Memory Management and Virtual Memory

Remember that when a process runs, it needs CPU time, but it also needs memory. When processes are run, they have to take up space in memory, so that the computer can read and load them quickly. However, compared to our hard disk drives, memory comes in smaller quantities. So to give us more memory than we physically have, we use something called virtual memory. Virtual memory is the combination of hard drive space and RAM that acts like memory that our processes can use. When we execute a process, we take the data of the program in chunks we call pages. We store these pages in virtual memory. If we want to read and execute these pages, they have to be sent to physical memory or RAM. Why don't we just store the entire program in RAM so we can execute it quickly? Well you could, if it was small enough, but for large applications it would be wasteful. Have you ever worked in a word processor and then gone to a menu don't normally use and noticed the application slow down a little? It's because your computer had to load the page for that menu from virtual memory into RAM. We don't use all the features of our application at once. So why load it up at once? It's similar to cooking a recipe from a cookbook. You don't need to read the whole book just to make one recipe. You only need to read the pages of the recipe you're currently using. When we store our virtual memory on our hard drive, we call the allocated space swap space. When we get into practical applications of disk partitioning, we'll allocate space for swap. The kernel takes care of all of this for us of course. It handles the process of taking pages of data and swapping them between RAM and virtual memory. But, the colonel isn't the only hard worker around. You've done great getting through the lessons so far. Nice work.

I/O Management

So far we've learned how hard our kernel works by handling files, managing file storage, juggling all the different processes running on our computer, and allocating memory. Another important task that our kernel handles is managing input and output. We refer to devices that perform input and output as I/O devices. These include our monitors, keyboards, mice, hard disk drives, speakers, Luther's headsets, webcams and network adapters. These I/O devices are all managed by our kernel, the kernel needs to be able to load up drivers that are used so that we can recognize and speak to these different types of hardware. When the kernel is able to start the drivers to communicate with hardware, it also manages the transfer of data in and out of the devices. I/O doesn't just mean the transfer of data between us and our devices. The devices also need to be able to talk to each other. Our kernel handles all the inter communication between devices. It also figures out what the most efficient method of transfer is and it tries its best to make sure our data doesn't have errors during process. When you're troubleshooting or solving a problem with a slow machine it's usually some sort of hardware resource deficiency. If you don't have enough RAM you can't load up as many processes. If you don't have enough CPU you can't execute programs fast enough. If you have too much input coming into the device or too much output going somewhere you'll also block other data from being sent or received. It's slow is one of the most common problems you'll solve in an IT support role. Knowing the potential sources of that slowness is a big help when you're trying to narrow down the cause of the latency. Troubleshooting is such an important part of any IT support role. Beyond desktop support, identifying the source of a resource bottleneck and a server or large IT system like a Web application can unlock performance gains and new heights of responsiveness for your users.

Interacting with the OS: User Space

We've covered the kernel's major responsibilities. Now, let's discuss the final major aspect of an operating system, how humans interact with it. This is what we call the userspace. When we interact with an operating system, we want to do certain functions like creating files and folders, open applications, and deleting items, you get the idea. There are two ways that we can interact with our OS. With a shell or graphical user interface. There are also some shells that use graphical user interfaces, but we'll work with a Command Line Interface or CLI shell. For the most part, this just means that we'll use text commands. A Graphical User Interface or GUI is a visual way to interact with the computer. We use our mouse to click and drag, to open folders, etc. We can see everything we do with it. You probably use a GUI every day without realizing you're using one. To watch this video, you probably used GUI clicking icons and navigating menus to open your web browser and navigate to the website. People usually recognize a device or product based on its GUI. You might be able to spot the difference between a computer running Microsoft Windows or Mac OS based on the design of the windows, menus, and icons, you've probably seen GUI's and other places too, like mobile phones and tablets, ATM machines, and airport kiosks. A shell is basically a program that interprets text commands and sends them to the OS to execute. Before we had fancy visual interfaces, commands like create a file had to be typed out. While we have GUI is today, the shell is still commonly used to run commands, especially by power users. Power users are above average computer users. Linux, especially, it's essential that you actually know commands, not just a GUI. This is because most of them Linux machines you interact with in IT support will be accessed remotely. Most of the time, you won't be given a GUI. There are lots of different types of shelves. Some have different features, some handled performance differently. It's the same concept behind different operating systems. For our purposes, we'll just be using the most common shell, Bash or bourne. Again, shell in Linux, you might be thinking, but it's easier for me to navigate a GUI than it is to use commands to do the same thing. Why would I want to learn both? I can't stress this enough. It's vital for you to know how to use a shell in an IT support role. Some tests can only be completed through commands. In more advanced IT roles, you might have to manage thousands of machines. You don't want to have to click a button or drag a window on every machine when you can just run a command once.

Logs

Imagine this scenario. You're playing your favorite video game and you finally get to the big boss. You spent countless hours finding this boss, neglecting all other responsibilities, like your job, school, even hygiene. That's pretty gross, but I get it. So you're right about to kill the big boss when suddenly your game console shuts off completely. You probably freak out for a second. But then you remember it's okay, you saved the game before the boss came along. So now you can turn it back on and you'll be at the same spot. But then your console shuts off again. This happens over and over. You like most people are devastated. You into a fit of rage, but then just before you toss your console out, you make one last dish effort and yell, tell me what's wrong with you. Suddenly you hear a faint voice telling you what you want to hear. Wouldn't that be amazing? Sure, that scenario was a bit exaggerated. But my point is that our computers actually can talk to us and tell us what's wrong. Maybe they won't whisper answers to us, but they speak to us in the form of logs. Logs or files that record system events on our computer, just like a system's diary. A computer will record events like when it was turned on, when a driver was loaded. And even when something isn't working in the form of error messages. In all operating systems, logs are kept so we can refer back to them when we need to find out something that happened. But logs can be hard to navigate because our computer will essentially record everything. Here's what a log looks like. As you can see, it can be tough to make your way through a log. But with a little bit of elbow grease, we can figure out what happened on our computer and piece together a solution. Unfortunately, our computers, cars and machines don't have a little voice that tells us what's wrong when there's a problem. But by the end of this program, you'll be able to navigate and read logs so you won't even need it.
magine this scenario. You're playing your favorite video game and you finally get to the big boss. You spent countless hours finding this boss, neglecting all other responsibilities, like your job, school, even hygiene. That's pretty gross, but I get it. So you're right about to kill the big boss when suddenly your game console shu: Added to Selection. Press [CTRL + S] to save as a note (Required)
en

The Boot Process

In this lesson, we're going to learn how our operating system starts up. As an IT support specialist, you'll probably work on lots of computers that won't start. It's important to know the steps and operating system takes, so you can help diagnose the issue. Booting a computer or starting a computer comes from the phrase to pull oneself up by one's bootstraps. Basically, it means to start from nothing and follow a series of steps to arrive at a fully operational system. When we start up with computer, will use the term boot. For most operating systems, the boot process follows a general pattern, much like how we have different cars startup in the same way. Put in the key, turn on the ignition, etc. Here's a rundown of the boot process. First, the computer is powered on. The BIOS/UEFI is a low-level software that initializes our computer's hardware to make sure everything is good to go. Next, the bios UEFI runs a process called the power-on self-test, or post. The post performs a series of diagnostic tests to make sure that the computer is in proper working order. Next, depending on the bios or UEFI configuration of boot device will be selected. Devices that are attached to our system, like hard drives, USB drives, CD drives, etc, are configured in a certain boot order. The devices will be checked in this order and the computer will search for what's known as a bootloader. The bootloader is a small program that loads the operating system. Once our computer finds a bootloader on a device in the listed order, it will start to execute this program. This will then start to load a larger and more complex program and eventually loads our operating system. Once the bootloader loads up our operating system, our kernel gets loaded. The kernel controls access to our computer's resources. It also loads up drivers and more so that our hardware can talk to our software. Next, essential system processes and user space items are launched. These include processes like user login, spinning up a desktop environment, and more which basically allows us to interact with our system. And that's it. After these simple steps, you'll be able to get to work.

English

Boot Methods

While the most common way to boot a computer is to simply push the power button and allow the normal process to run, there are many other boot options. This reading covers the various methods you can use to boot a computer.

Internal method

You can create partitions on the computer's drive so that only one part of the drive runs the boot process. A common reason to partition your drive is to have two separate operating systems on your computer, such as both Windows and Linux. When you have two operating systems on your drive, you must choose which one will run the boot process. Having two possible systems to boot into is called dual booting.

While having two operating systems can be helpful for various reasons, it is especially helpful when one system is failing or unable to boot. If this happens, you can still boot the computer using the other system and troubleshoot from there.

External tools

External tools can be used to boot the computer. You can load the needed resources on an external tool to boot a system before any problems happen.

External bootable devices include:

- **USB drive**: You use a USB drive loaded with resources needed to boot the computer. This drive is inserted into a USB port and chosen at startup.

- **Optical Media:** You use a disk loaded with booting resources. This disk can be a DVD, CD, or Blu-ray disk and is loaded through the computer's optical drive.

- **Solid State Boot Drive:** You use a solid state drive to boot. Solid state drives do not use spinning discs or moving parts. This solid state drive can be installed in your computer or can be a smaller device such as a flash drive.

- **External hot-swappable drive**: You boot from an external hard drive that can be moved between computers without turning it off.

- **Network boot:** You boot the operating system directly from a local area network (LAN) without using a storage device. Your computer must be connected to a LAN for this option.

- **Internet-based boot:** You boot the computer from an internet source, as long as it is a secure source. Your computer must be connected to the internet for this option.

Window OS or Linux OS

In order to boot either Windows OS or Linux OS with an external tool, you'll need to enter BIOS at startup by pressing F2/F12/Del keys. From there you can change the boot order so that the first option is the external tool you want to use.

macOS

If booting macOS, press and hold the Option key at startup. This will open up the Startup Manager, which will scan your computer and identify bootable devices. Then you can choose the bootable device you want to use.

Key Takeaways

There are multiple ways to boot a computer.

- A computer can be partitioned into different operating systems and you can select which OS to use when booting.

- You can boot from an external tool. External tools include USB drives, optical media, solid state boot drives, external hot-swappable drives, network booting, and internet-based booting.

- Choosing a boot method on startup varies depending on which operating system you use.

Boot Methods Best Practices

The most common way to boot a computer is to simply push the power button and allow the normal startup process to run. But what happens if the normal startup process becomes corrupted and the computer will not boot? Or maybe you would like to run a computer on a different operating system than the one specified by your normal boot process. For situations like these, you have several options for booting your operating system. This reading covers the various methods you can use to boot a computer.

The boot process

When your computer is powered on, the BIOS/UEFI (BIOS) runs a series of diagnostic tests to make sure that the computer is in proper working order. The BIOS is a low-level software that initializes a computer's hardware to make sure everything is good to go. A boot device is selected based on a boot order that is configured in the BIOS. Devices that are attached to your system, like hard drives, USB drives, and CD drives are checked in this configured boot order and the computer searches these devices for a small program called a “bootloader.” Once your computer finds a bootloader on a device, it executes this program. The bootloader program then initiates a process that loads the specific operating system setup that you want to use.

You can choose a computer’s boot method by telling the BIOS on which device to search for the bootloader. If you want to run an OS setup that’s stored on a USB drive, you can configure the boot order in your computer’s BIOS to search for a bootloader on a USB drive first.

Configuring boot options

Boot order is the order in which a computer chooses which boot files to use to startup. The boot order determines your boot method. To set the boot order for a computer, you need to enter the BIOS and configure the boot options.

To enter your computer’s BIOS on a Windows or Linux computer, power on the system and look for an on-screen message that says which function key you should press to enter setup. The function keys used for entering the BIOS vary between computer manufacturers and the version of BIOS. Some of the more common function key messages are "Press DEL to enter SETUP," "F2=SETUP," or "Press F12 to enter SETUP." If booting macOS, press and hold the Option key at startup. This will open up the Startup Manager, which will scan your computer and identify bootable devices. Then you can choose the bootable device you want to use.

If you press the specified function key during the Windows or Linux power up process (before the OS begins to load), you will open your BIOS program. A BIOS screen will look similar to this:

Image of a BIOS screen featuring the boot options menu.

The BIOS screen will vary depending on your computer manufacturer and BIOS version, but all BIOS programs will feature a Boot Options menu. The Boot Options menu is where you can set your preferred boot method.

The boot options menu lists all the devices attached to your system where it may find a bootloader program. These include devices like internal hard drives, USB drives, CD drives, as well as other storage options, like network storage or cloud storage. In the BIOS boot options menu you can set the specific order you want to search these devices for the bootloader that will load your OS setup. The BIOS will run the first bootloader that it finds.

Boot method options

You may find the following boot methods listed in your BIOS boot options:

External options

- **USB drive:** You use a USB drive loaded with resources needed to boot the computer. This drive is inserted into a USB port and chosen at startup.

- **Optical Media:** You use an optical media disk loaded with booting resources. This disk can be a DVD, CD, or Blu-ray disk and is loaded through the computer's optical drive.

The USB drive and optical media methods are useful for recovering a computer with a corrupted OS. They can also be used to start up a computer with a different OS. For example, you might boot a Windows computer in a Linux environment by using a USB with Linux OS. You will need to

prepare these media with a bootable OS in order to use them as a boot method (see resources linked below).

- **Solid State Boot Drive:** You can use a solid state drive to boot your computer. Solid state drives do not use spinning discs or moving parts. This solid state drive can be installed in the computer or can be a smaller device such as a flash drive.

- **External hot-swappable drive:** You may boot from an external hard drive that can be moved between computers without turning it off.

- **Network boot:** You can boot an operating system directly from a local area network (LAN) without using a storage device. Your computer must be connected to a LAN for this option. The network boot is used when the computer does not have an OS installed, among other things. To boot from a network, you will need to set up the Preboot Execution Environment (PXE) capability on the BIOS and have the network environment prepared for this type of request (see resources linked below).

- **Internet-based boot:** You boot the computer from an internet source, as long as it is a secure source. If you are in charge of a network and your server is down for any reason, you can use this boot method to remotely power on the server and restart network operations. Internet-based boot can be achieved in one of two ways:

1. Remote access. Remote Access Controller (IPMI or similar) has to be enabled on the BIOS and the computer needs to have a Remote access control device, such as IDRAC (see resources linked below).

2. Wake on LAN (WoL). This process requires the WoL option enabled on the BIOS (see resources linked below). The WoL instruction should come from a device in the network or use a WoL gateway, and the network card should have WoL capability.

Internal options

Disk partitions: You can create partitions on your computer's drive so that only one part of the drive runs the boot process. A common reason to partition your drive is to have two separate operating systems on your computer. For example, you could have Windows on one partition of your drive and Linux on the other. When you have two operating systems on your drive, you must choose which one will run the boot process. Having two possible systems to boot into is called dual booting.

While having two operating systems can be helpful for various reasons, it is especially helpful when one system is failing or unable to boot. If this happens, you can still boot the computer using the other system and troubleshoot from there.

Key Takeaways

There are multiple ways to boot a computer.

- A computer can be partitioned into different operating systems and you can select which OS to use when booting.

- You can boot from an external tool. External tools include USB drives, optical media, solid state boot drives, external hot-swappable drives, network booting, and internet-based booting.

- Choosing a boot method on startup varies depending on which operating system you use.

Resource Links:

- How to make a bootable CD/DVD/USB to install windows
- How to build your own bootable Linux Live CD
- Create a bootable installer for macOS
- What is preboot execution environment (PXE)?
- How to set up PXE boot for UEFI hardware
- Installing and configuring the RAC software
- How to enable and use Wake on LAN (WoL) on Windows 10

Mobile Operating Systems

Some mobile devices are general-purpose computing devices like tablets or smartphones. Other mobile devices like fitness monitors, e-readers, and smartwatches, are designed to do a smaller set of tasks. General-purpose mobile devices generally use a mobile operating system that's derived from other operating systems. For example Android is derived from Linux and iOS shares a lot of core components with MacOS. So how are mobile operating systems different from the OSs that they're based on? Mobile devices run on batteries that have to be recharged or replaced on a regular basis and you want the device to last as long as possible between charges. So mobile operating systems are optimized to use as little power as possible, for example, by removing OS features and applications that the mobile device doesn't need. We also use motion, touch, and voice to interact with mobile devices in very different ways from desktop or server computers. This requires adding device drivers and support to the mobile operating system. More specialized mobile devices like fitness trackers, e-readers, and GPS devices, often use custom OSs that are optimized for what the device is designed to do. These devices are even more slimmed down to run on very minimal hardware with very minimal battery power. They might also be built using specialized chips and peripherals, which more general-purpose operating systems don't know how to run on.

Cindy: Drive and career path

I'd say when I first started I thought there were two jobs you could do, you could be a sys admin or you could be IT support. But that's completely false. There's like a huge amount of opportunity in IT. You could be specialized in networks. You could be specialized in databases or like reliability engineering. I'd say it doesn't matter if you're a guy a girl an alien ideas for everyone. It's just problem solving with technology and anyone can do it. I know people who are in IT that degrees in liberal arts or like cooking and all these other things. People come from IT from all sorts of backgrounds. I've always drive to be different. And for me I think that was just learning skills like growing up girls didn't use computers. I was like Igirl relly use a computer? Girls didn't know how to drive manual cars. I'm going to go learn how to drive a manual car. They don't know how to ride motorcycles. I can do that too. And I enjoy learning I enjoy learning a lot of things. I enjoy picking up new skills. Technology is a real equalizer for people who don't have the current educational like background. You can load up a website. All these learning websites. There's so much information on the Internet and I think that technology really equalizes that for people who want to get into certain careers. They want to learn something. I think what's available now is amazing and I think I wish I'd had that ten years ago when I started.
I'd say when I first started I thought there were two jobs you could do, you could be a sys admin or you could be IT support. But that's completely false. There's like a huge amount of opportunity in IT. You could be specialized in networks. You could be specialized in databases or like reliability engineering. I'd say it doesn't matter if you're a guy a girl an alien ideas for everyone. It's just problem solving with technology and anyone can do it. I know people who are in IT that degrees in liberal arts or like cooking and all these other things. People come from IT from all sorts of backgrounds. I've always drive to be different. And for me I think that was just learning skills like growing up girls didn't use computers. I was like Igirl relly use a computer? Girls didn't know how to drive manual cars. I'm going to go learn how to drive a manual car. They don't know how to ride motorcycles. I can do that too. And I enjoy learning I enjoy learning a lot of things. I enjoy picking up new skills. Technology is a real equalizer for people who don't have the current educational like background. You can load up a website. All these learning websites. There's so much information on the Internet and I think that technology really equalizes that for people who want to get into certain careers. They want to learn something. I think what's available now is amazing and I think I wish I'd had that ten years ago when I started.