1 Reboot or turn on the computer. As soon as the computer restarts, you'll see a black screen that says ″lenovo″ in large white letters.[1] The screen only appears for a few seconds, so you'll need to perform the next step quickly. If you're using Windows 8/8.1, you'll need to restart the computer from Windows to access the BIOS. From the Windows desktop, press ⊞ Win+i, click Power, and then click Restart. 2 Repeatedly tap F2 until the BIOS appears. Aim to press the key twice per second. The key you need to use for your model appears at the bottom of the ″lenovo″ screen next to ″to Setup.″[2] You may need to press FN + F2. Other models and Linux computers use F1 instead of F2.[3] This may take a few attempts until you can enter BIOS. Advertisement 1 2 Press the One Key Recovery (Novo) button. This button usually looks like a curved arrow pointing backward and is usually on the top or side of your device.[4] Only Ideapad Lenovo's have the Novo button.[5] Your computer will boot directly into BIOS when you press this button. Advertisement 1 Click the Start menu. It's usually at the bottom-left corner of the desktop.[6] 2 Click Settings . It's near the bottom of the menu.[7] 3 Click Update & security. It's the option with a curved arrow. 4 Click Recovery. It's in the left column.[8] 5 Click Restart now. It's under ″Advanced startup″ in the right panel. The PC will reboot to a blue menu. 6 Click Troubleshoot on the menu. It's the option with a screwdriver and a wrench.[9] 7 Click Advanced options. It's the last option. 8 Click UEFI Firmware Settings. It's in the right column.[10] 9 Click Restart. The PC will restart and boot into the BIOS. Advertisement 1 Log out of Windows. Windows 10: Click Start . Click your user name. It's at the top-left corner of the menu. Click Sign out. Windows 8.1/8: Press ⊞ Win+X. Click Shut down or sign out. It's at the bottom of the menu. Click Sign out. 2 Hold ⇧ Shift as you click the menu. Don't lift your finger when the menu appears. 3 Continue holding ⇧ Shift and click Restart. Keep holding the key down as the PC reboots to a blue menu screen.[11] 4 Click Troubleshoot on the menu. It's the option with a screwdriver and a wrench.[12] 5 Click Advanced options. It's the last option.[13] 6 Click UEFI Firmware Settings. It's in the right column.[14] 7 Click Restart. The PC will restart and boot into the BIOS.[15] Advertisement 1 Move the mouse to the top-right corner of the home screen and then move it downward. A menu will appear.[16] 2 Click Settings. 3 Click Change PC Settings. It's at the bottom of the menu.[17] 4 Click Update and recovery. It's at the bottom of the left column. 5 Click Recovery. It's in the left column. 6 Click Restart now. It's under the ″Advanced startup″ header in the right panel. The PC will reboot to a blue menu. 7 Click Troubleshoot on the menu. It's the option with a screwdriver and a wrench. 8 Click Advanced options. It's the last option. 9 Click UEFI Firmware Settings. It's in the right column. 10 Click Restart. The PC will restart and boot into the BIOS. Advertisement Ask a Question Advertisement Thanks Advertisement Written by: wikiHow Technology Writer This article was co-authored by wikiHow staff writer, Darlene Antonelli, MA. Darlene has been writing and editing tech content at wikiHow since 2019. She previously worked for AppleCare, served as a writing tutor, volunteered in IT at an animal rescue, and taught as an adjunct professor for EN101 and EN102. Darlene has completed Coursera courses on technology, writing, and language. She holds both a BA (2011) and an MA (2012) from Rowan University in Writing, with a focus on workplace communication. With her extensive experience, academic background, and ongoing learning, Darlene has become the go-to grammar expert for her friends and family, as well as a skilled wordsmith for anyone in need. This article has been viewed 628,666 times. Co-authors: 7 Updated: September 26, 2024 Views: 628,666 Categories: Laptop Maintenance and Repair Print Send fan mail to authors Thanks to all authors for creating a page that has been read 628,666 times. In some cases we need to enter BIOS to make changes in order to fix some PC issues. Here we will guide you how to enter BIOS setup on Lenovo laptops preinstalled with Windows 10/8.1/8, including Thinkpad P series, Ideapad series, Yoga series, Z series, B series, and more. On older Lenovo laptops, accessing BIOS is easily that you just need to Press certain key like F1 during startup when the computer logo is displayed. However, this becomes difficult because it is impossible to enter BIOS Setup by pressing F1 on Windows 8 and newer Windows OS. Thus, how to access BIOS setup on Lenovo laptop pre-installed with Windows 10/8.1/8? Part 1: How to Enter BIOS Setup on Lenovo Laptop from within Windows 8/8.1/10 You can easily access to BIOS setup on your laptop in following steps instead of using the traditional F1 button. Boot into Windows. Open the Charm Bar by pressing Windows Key-C. Choose Settings at the bottom and when the next set of options shows up choose Change PC settings. On the left, at the bottom choose Update and Recovery, then on the screen that comes up, choose Recovery at the bottom. On the next screen, go to Advanced startup -> Restart Now. To enter Setup Utility, you will want to choose Troubleshoot -> Advanced options -> UEFI Firmware Settings -> Restart. To access Boot menu, click on Use A Device ->Boot Menu. Part 2: How To Get the "Press F1 to Enter the BIOS" Prompt on the Bottom Left on the Screen This is a case that you can't boot into your PC such as you forgot Windows login password. In this case you can't enter BIOS in method above, thus you can try to get the "Press F2 to Enter the BIOS" prompt on the bottom left on the screen in following ways: Press and hold the Shift key while selecting the Shutdown option in Windows 8. This will make the PC perform a full shutdown instead of a hybrid shutdown. Then F1 or F12 can be pressed successfully during startup. Select Restart instead of Shutdown. Then F1 or F12 can be pressed successfully during startup. Choose "Yes, I'd Like to Start Rescue Now."Follow the on-screen prompts to back up your data; otherwise, click "Skip Rescue". Disable the fast startup option in Control Panel -> Hardware and Sound -> Power Options -> Choose what the power buttons do. After this, you can the "Press F1 to Enter the BIOS" prompt will appear on the bottom left on the screen, you can either press F1 to enter Setup Utility or press F12 to enter the Boot Menu. In this artile we share how to enter UEFI Bios in Lenovo laptops. And if you have any problems on Lenovo password reset, get Windows Password Key to have your problem solved instantly. Related Articles How to Enter BIOS on Toshiba Laptop Running Windows 10/8.1/8 How to Crack Laptop Password on Windows 8 How to Reset Password on Windows 7 with USB comments powered by 1 Reboot or turn on the computer. As soon as the computer restarts, you'll see a black screen that says ″lenovo″ in large white letters.[1] The screen only appears for a few seconds, so you'll need to perform the next step quickly. If you're using Windows 8/8.1, you'll need to restart the computer from Windows to access the BIOS. From the Windows desktop, press ⊞ Win+i, click Power, and then click Restart. 2 Repeatedly tap F2 until the BIOS appears. Aim to press the key twice per second. The key you need to use for your model appears at the bottom of the ″lenovo″ screen next to ″to Setup.″[2] You may need to press FN + F2. Other models and Linux computers use F1 instead of F2.[3] This may take a few attempts until you can enter BIOS. Advertisement 1 2 Press the One Key Recovery (Novo) button. This button usually looks like a curved arrow pointing backward and is usually on the top or side of your device.[4] Only Ideapad Lenovo's have the Novo button.[5] Your computer will boot directly into BIOS when you press this button. Advertisement 1 Click the Start menu. It's usually at the bottom-left corner of the desktop.[6] 2 Click Settings . It's near the bottom of the menu.[7] 3 Click Update & security. It's the option with a curved arrow. 4 Click Recovery. It's in the left column.[8] 5 Click Restart now. It's under ″Advanced startup″ in the right panel. The PC will reboot to a blue menu. 6 Click Troubleshoot on the menu. It's the option with a screwdriver and a wrench.[9] 7 Click Advanced options. It's the last option. 8 Click UEFI Firmware Settings. It's in the right column.[10] 9 Click Restart. The PC will restart and boot into the BIOS. Advertisement 1 Log out of Windows. Windows 10: Click Start . Click your user name. It's at the top-left corner of the menu. Click Sign out. Windows 8.1/8: Press ⊞ Win+X. Click Shut down or sign out. It's at the bottom of the menu. Click Sign out. 2 Hold ⇧ Shift as you click the menu. Don't lift your finger when the menu appears. 3 Continue holding ⇧ Shift and click Restart. Keep holding the key down as the PC reboots to a blue menu screen.[11] 4 Click Troubleshoot on the menu. It's the option with a screwdriver and a wrench.[12] 5 Click Advanced options. It's the last option.[13] 6 Click UEFI Firmware Settings. It's in the right column.[14] 7 Click Restart. The PC will restart and boot into the BIOS.[15] Advertisement 1 Move the mouse to the top-right corner of the home screen and then move it downward. A menu will appear.[16] 2 Click Settings. 3 Click Change PC Settings. It's at the bottom of the menu.[17] 4 Click Update and recovery. It's at the bottom of the left column. 5 Click Recovery. It's in the left column. 6 Click Restart now. It's under the ″Advanced startup″ header in the right panel. The PC will reboot to a blue menu. 7 Click Troubleshoot on the menu. It's the option with a screwdriver and a wrench. 8 Click Advanced options. It's the last option. 9 Click UEFI Firmware Settings. It's in the right column. 10 Click Restart. The PC will restart and boot into the BIOS. Advertisement Ask a Question Advertisement Thanks Advertisement Written by: wikiHow Technology Writer This article was co-authored by wikiHow staff writer, Darlene Antonelli, MA. Darlene has been writing and editing tech content at wikiHow since 2019. She previously worked for AppleCare, served as a writing tutor, volunteered in IT at an animal rescue, and taught as an adjunct professor for EN101 and EN102. Darlene has completed Coursera courses on technology, writing, and language. She holds both a BA (2011) and an MA (2012) from Rowan University in Writing, with a focus on workplace communication. With her extensive experience, academic background, and ongoing learning, Darlene has become the go-to grammar expert for her friends and family, as well as a skilled wordsmith for anyone in need. This article has been viewed 628,666 times. Co-authors: 7 Updated: September 26, 2024 Views: 628,666 Categories: Laptop Maintenance and Repair Print Send fan mail to authors Thanks to all authors for creating a page that has been read 628,666 times. Entering the BIOS on a Lenovo laptop is essential for adjusting settings and resolving problems. Entering the BIOS can save you time and hassle. It lets you tweak boot options and manage your hardware. Different Lenovo models use different keys such as F1, F2, or Delete to get in. If you get stuck, do not worry; we will go over some tips that will assist you with this. Your laptop's true power is about to be unleashed… Key Takeaways Understanding BIOS is critical for managing hardware settings and troubleshooting issues on your Lenovo laptop. Not all Lenovo laptops use the same method to access BIOS; verify this compatibility for your specific model. Get used to various ways to enter BIOS, like through function keys when booting up. Follow the step-by-step instructions carefully to ensure you enter BIOS correctly and safely. If you are still having trouble getting into BIOS, try some troubleshooting methods that fix common issues. Consider additional tips to improve your experience when using BIOS settings for your Lenovo laptop. Read More: How to Take a Screenshot on a Lenovo Laptop: 5 Easy Steps How to Reset the BIOS Password on a Lenovo Laptop Easily? How to enable the camera on your Lenovo laptop? Why is my Lenovo laptop so slow on Windows 10/11? Understanding BIOS What is BIOS? BIOS refers to Basic Input/Output System. It serves as the bridge between your computer's hardware and operating system. Whenever you switch on your Lenovo laptop, BIOS runs first. It tests all the connected hardware, such as your hard drive and memory, to ensure everything works properly. If something goes wrong, BIOS will warn you about it. Since BIOS settings affect the performance of your laptop, they are a big deal. If you change the boot order in the BIOS, you decide on which device your laptop will check first when it starts up. That means it gives you control over how your laptop boots up! A misconfigured setting can slow your system or even prevent it from booting at all. Importance of BIOS in Laptops BIOS is important for changing boot order and hardware settings. You can configure it to prioritize your SSD over an external drive, which speeds up your startup. If your laptop is having trouble booting, taking a look at BIOS could help. It may uncover a hardware issue or incorrect settings that need to be corrected. Security also receives a boost with BIOS. You can then set a password that locks access to BIOS settings. This extra layer of protection helps keep your laptop free from people making changes that you don't want. To enter the BIOS on a Lenovo laptop, tap F2 as soon as you start it up. You should aim to press the key twice per second until the BIOS screen appears. If you have another model or if you use Linux, try hitting F1 instead. Knowledge of BIOS makes you the master and not the servant of your laptop's performance and security. It's worth taking the time to explore these settings so you can enjoy a smoother experience overall. Lenovo Laptop Compatibility Supported Lenovo Models Different models of Lenovo laptops may have different ways to access the BIOS setup utility. Well-known models like the ThinkPad series, IdeaPad series, and Legion series each have unique mechanisms for entering the BIOS. For instance, many users find that tapping the F2 key works well for most models, while some other models or Linux computers may require the F1 key instead to access the UEFI boot options. If you're uncertain about your specific model, it's advisable to consult the article or documentation that came with the laptop. This will provide you with essential information, including the specific key you need to press to enter the BIOS setup. Following these steps can help you skip the guesswork and navigate directly to the settings window you're looking for. By understanding the startup process and utilizing the correct function key, you can effectively manage your laptop's firmware interface. This knowledge is particularly useful for troubleshooting issues or configuring specific settings. Whether you're a tech enthusiast or a casual user, having access to the right resources can significantly enhance your experience with Lenovo products. In conclusion, knowing how to access the BIOS setup is crucial for optimizing your laptop's performance. Remember to check the manufacturer's documentation for your model and familiarize yourself with the necessary key combinations. This way, you'll be better equipped to handle any configuration needs and ensure your laptop operates smoothly. Compatible Operating Systems Depending on what operating system your laptop runs, you'll experience a slightly different method for accessing the BIOS setup utility. Almost all Lenovo laptops come with Windows, which allows you simple access to the BIOS. If you're using Windows, spamming the F2 key until the BIOS screen loads generally does the job. Aim to tap that function key twice a second to ensure a smooth transition. If you're using a different OS, like Linux, be mindful of this; some UEFI boot features may look different or act differently. Certain settings in the BIOS setup may be limited or enhanced depending on the OS you're using. Always keep in mind how your configuration impacts your access to BIOS settings. Understanding the startup process is crucial, as some options may vary in visibility based on the operating system. The mechanisms for accessing the boot menu can differ significantly, so it's essential to follow the correct steps for your specific setup. This knowledge will help you navigate through your BIOS effectively. For those interested in exploring advanced features, such as virtualization, it is important to familiarize yourself with the UEFI specifications. This standard firmware interface provides a more modern approach to system management than traditional BIOS. Knowing how to access and utilize these settings can enhance your overall computing experience. Ultimately, the way your laptop's firmware interface operates can greatly influence your system's performance. Whether you're repairing issues, applying software updates, or simply adjusting preferences, understanding the BIOS setup and UEFI can lead to more efficient use of your device. Methods to Access BIOS 1. Function Key Access To access the BIOS setup utility, you need to reboot your Lenovo laptop. As it begins the startup process, repeatedly tap the F2 key; for certain models, the F12 key may be required instead. Timing is crucial, so aim to hit the function key twice per second. If you delay too long, your laptop will proceed into the operating system rather than the BIOS setup. Following these steps ensures you can navigate the boot menu effectively. If you encounter issues, consult support resources or articles online that delve into the UEFI boot process and the necessary settings window adjustments. This knowledge can enhance your overall web experience and help you manage your device's firmware interface more efficiently. 2. Using Novo Button Find the Novo button on your Lenovo laptop. Usually, it's right next to the power button. Press this button or use Fn + Novo to open the OneKey Recovery menu. From there, click on "UEFI Firmware Settings." This step boots you straight into the BIOS. 3. Access from Windows Desktop You can open the BIOS setup utility from the Windows desktop by following these steps. Head to your settings, then select Update & Security. Next, locate "Recovery" and choose "Restart now," which brings up advanced startup options, including access to the UEFI boot menu. Once in the UEFI, you can configure your preferences for the startup process. This allows you to manage settings related to the system clock and other essential configurations for your desktop PC. 4. Shift and Restart Method Another way is to use the Shift key. By holding Shift while you click Restart in Windows, you access an advanced startup menu that allows you to enter the BIOS setup utility. From there, select "Troubleshoot," then navigate to "Advanced options" to access the BIOS settings. This process is essential for configuring your system's startup sequence, enabling you to make necessary adjustments in the BIOS setup. Following these steps ensures you can effectively manage your UEFI boot settings and optimize your desktop PC's performance. Step-by-Step Instructions 1. Access with Function Key To access the BIOS setup utility on your Lenovo laptop, boot it up and search for the F2 or F12 function key. As soon as the Lenovo logo appears, start tapping that key continuously, aiming for about twice per second. If you miss it, don't worry—just reboot and try again! Timing is crucial here, so stay focused. Once successful, the BIOS setup screen will display, allowing you to configure your system settings. 2. Enter with Novo Button Another way to access the BIOS setup utility is by using the Novo button, which is typically located next to the power button or on the side of your laptop. To utilize this function key, first switch off your laptop. Then, tap the Novo button, and a boot menu will appear on your screen. From this menu, ensure that "BIOS Setup" is selected, as this is the primary function for entering the BIOS setup. Following these steps will allow you to navigate the UEFI boot settings effectively. Understanding the BIOS setup process is crucial for configuring your system preferences and managing essential firmware interface settings. This knowledge can significantly enhance your overall web experience and control over your device's startup sequence. 3. Use Windows 10/11 Desktop If you're already logged into Windows, accessing the BIOS setup utility is simple. Click the Start menu and select "Settings." Then, navigate to "Update & Security" and select "Recovery." Under "Advanced startup," hit "Restart now." The computer restarts, and you select "Troubleshoot," then "Advanced options," and finally "UEFI Firmware Settings." After hitting "Restart" again, you will enter the BIOS setup, where you can adjust various settings related to your system's startup process. This includes configuring options in the UEFI boot menu, which is essential for optimizing your desktop PC's performance. Familiarizing yourself with these settings can enhance your overall web experience and system functionality. The BIOS setup is a crucial component of your computer's firmware interface, allowing you to control specific preferences for your hardware. By accessing the UEFI settings window, you can manage elements like the system clock and boot sequence, ensuring your PC operates smoothly. Remember, making changes in the BIOS can affect how your computer functions, so it's important to proceed with caution. For more detailed guidance, consider referring to a comprehensive article or support resources that explain the necessary steps and mechanisms involved in BIOS configuration. 4. Shift and Restart Technique To initiate the advanced startup process, try the Shift and Restart technique. Hold down the Shift key while you click Restart in the Start menu, which will reboot your desktop PC into advanced options. From this menu, select "Troubleshoot," then "Advanced options," and finally "UEFI Firmware Settings" to access the BIOS setup utility. Once in the BIOS setup, you can adjust certain settings related to the UEFI boot process. This allows you to configure preferences that may improve your system's functioning. If needed, you can also tap the F1 key for help, ensuring you navigate the settings window effectively. Following these steps will guide you through the troubleshooting process, allowing for better control over your PC's startup sequence. Understanding these mechanisms can enhance your experience with technology, making it easier to manage software updates and other essential tasks. Troubleshooting BIOS Access Common Issues Encountered Many users seem to have trouble accessing their Lenovo laptops' BIOS setup utility. A common issue is that you don't see the BIOS screen at all; instead, the laptop boots directly into the operating system. This is usually due to fast startup settings that streamline the startup process, often skipping the BIOS entirely, which complicates access to the UEFI boot options. If pressing the function key does nothing, it may be time to check your keyboard. Sometimes, a malfunctioning keyboard can prevent you from entering the BIOS setup. Ensure that your keys are functioning correctly before attempting to access the BIOS again. To resolve this, consider disabling the fast startup feature in your system's settings window. This adjustment can help you access the BIOS setup more reliably during the boot menu phase. Following these steps will enhance your ability to manage your laptop's firmware interface effectively. If you continue to face issues, it may be worthwhile to consult online communities or support resources for further assistance. Engaging with these technology forums can provide valuable insights and solutions from other users who have encountered similar problems. Ultimately, ensuring your laptop's BIOS is accessible is crucial for configuring preferences and performing necessary updates. By following the right steps, you can enhance your overall web experience and maintain your device's functionality. Solutions for Access Problems To enter the BIOS setup utility, start by tapping the F2 key repeatedly as soon as you turn on your laptop. You should hit that function key twice per second until you access the BIOS. If F2 doesn't work, consider trying the F1 key or other function keys, as different Lenovo models or Linux computers may have varying requirements for entering the BIOS setup. Disabling fast startup in Windows can also assist in resolving access issues. This setting often conflicts with the ability to enter the BIOS and can disrupt the startup process. To adjust this, navigate to Control Panel > Power Options > Choose what the power buttons do. From there, uncheck "Turn on fast startup" and save your changes to enhance your system's performance. If you're still experiencing difficulties, consult the specific article related to your laptop model for tailored access instructions. Each model may have distinct key combinations or mechanisms for entering the BIOS setup, which can vary significantly across different products. Additionally, ensuring that you have the latest firmware updates for your device can improve overall functionality. Firmware updates can rectify bugs and enhance compatibility with the UEFI boot process, allowing for a smoother startup sequence. Lastly, remember that accessing the BIOS is crucial for configuring certain settings, such as boot order and hardware preferences. By following these steps, you can effectively manage your laptop's BIOS setup and ensure optimal performance for your technology needs. Additional Tips for Users Preparing Your Laptop Ensure your laptop is sufficiently charged to avoid any interruptions during the BIOS setup utility. Charge it up fully or leave it plugged in, as you do not want it to shut down while navigating through the UEFI boot options. Additionally, close all open apps to prevent issues when the laptop reboots and enters the BIOS setup. Before accessing the BIOS, take a moment to review the preferences and settings window. Familiarizing yourself with the configurations can enhance your confidence in managing the startup process effectively. Understanding these elements will prepare you for any adjustments you may need to make. The following steps will guide you through the BIOS setup. Knowing what to expect will streamline your experience and help you navigate the boot menu with ease, ensuring a successful interaction with your laptop's firmware interface. Taking the time to familiarize yourself with these mechanisms is essential for efficient operation. Ensuring Successful Entry Following the correct steps is crucial for entering the BIOS setup utility. Timing matters, so as your laptop is starting up, tap the function key promptly. This may take several attempts, so keep your cool during the startup process. If it doesn't work at first, don't give up hope; repeat the actions until you nail it. The BIOS setup is essential for configuring various settings, including the UEFI boot options. If you are having trouble accessing it, remember that the F1 key is often the primary function for many systems. Keep trying, as persistence is key to successfully navigating this firmware interface and ensuring your device operates smoothly. Don't hesitate to consult online resources like a WikiHow article for more detailed guidance on this process. These support resources can provide valuable information on troubleshooting and accessing the boot menu, ensuring that you can effectively manage your laptop's settings and preferences. Keeping Your BIOS Updated Keeping your BIOS updated helps improve performance and security. Check for updates periodically on Lenovo's site. An updated BIOS can fix bugs while also increasing system stability. Backing Up Important Data Before making any changes in BIOS, back up your important files. Changes may seem harmless, but sometimes they just don't work out the way you'd hoped, and that's where a backup comes in; you won't lose any good information. Documenting Changes Whatever you change in BIOS, write it down. Keeping track of what you modify pays off if you need to troubleshoot later. It gives you a reference point for curation updates down the line. Conclusion Viewing the BIOS on your Lenovo laptop brings up a plethora of options. You can modify settings, configure hardware, and increase performance. There is a nice clear set of key commands to reach what you want to do. You can also dive into the setup menu if you prefer; either way, it is easier than you may think! If you're running into trouble, don't sweat it. Follow the troubleshooting tips to get you back on the right track. Keep in mind that every laptop has its quirks, and with patience and practice, you'll figure it out. Now that you know how to access BIOS, why not give it a try? Tinker around and see what works best for your needs. Jump in, have a look around, and take charge of your device. Frequently Asked Questions How do I know if my Lenovo laptop has BIOS? To check if your Lenovo laptop has a UEFI boot option, restart the device and pay attention to the prompt during the startup process. This prompt usually indicates which function key, such as F1 or F2, to tap to access the BIOS setup utility for configuration. What is the purpose of BIOS on a laptop? The BIOS setup utility, an essential component of the UEFI boot process, initializes hardware during booting while providing vital runtime services for operating systems and hardware management. Can I access BIOS without a keyboard? No, you need a keyboard to access the BIOS setup utility. There are specific function keys you must tap at startup to enter the BIOS setup, with the F1 key being essential to this startup process. What should I do if I cannot access BIOS on my Lenovo laptop? If you can't enter the BIOS setup utility or UEFI boot, ensure you're pressing the correct function key during the startup process. If problems persist, consider rebooting your laptop or checking for hardware issues. Is it safe to change settings in BIOS? Change settings in the BIOS setup utility, and the system will run differently. Only change settings if you know what they do, as improper adjustments can affect the overall startup process. Always document original settings before making changes to avoid potential issues. How do I reset BIOS to factory settings? To reset BIOS, tap the F1 key during startup to enter the BIOS setup utility, navigate to the "Exit" tab, and select Load Setup Defaults to restore factory settings. Can I update my Lenovo laptop's BIOS? Yes, you can update your laptop's BIOS setup utility. Check Lenovo's official website for the latest firmware updates, and read the article closely to avoid issues during the update process. Firmware for hardware initialization and OS runtime services This article is about the BIOS as found in IBM PC/AT and compatibles. For the modern replacement that is often still called BIOS, see UEFI. For the general concept, see Firmware. For other uses, see Bios (disambiguation). A pair of AMD BIOS chips for a Dell 310 computer from the 1980s. The bottom one shows the distinct window of an EPROM chip.Year started1981[a]OrganizationOriginally IBM as proprietary software, later industry wide as a de facto standard. In 1996, the BIOS Boot Specification was written by Compaq, Phoenix Technologies and Intel.SuccessorUEFI In computing, BIOS (/ˈbaɪɒs, -oʊs/, BY-oss, -ohss; Basic Input/Output System, also known as the System BIOS, ROM BIOS, BIOS ROM or PC BIOS) is a type of firmware used to provide runtime services for operating systems and programs and to perform hardware initialization during the booting process (power-on startup).[1] The firmware comes pre-installed on the computer's motherboard. The name originates from the Basic Input/Output System used in the CP/M operating system in 1975.[2][3] The BIOS firmware was originally proprietary to the IBM PC; it was reverse engineered by some companies (such as Phoenix Technologies) looking to create compatible systems. The interface of that original system serves as a de facto standard. The BIOS in older PCs initializes and tests the system hardware components (power-on self-test or POST for short), and loads a boot loader from a mass storage device which then initializes a kernel. In the era of DOS, the BIOS provided BIOS interrupt calls for the keyboard, display, storage, and other input/output (I/O) devices that standardized an interface to application programs and the operating system. More recent operating systems do not use the BIOS interrupt calls after startup.[4] Most BIOS implementations are specifically designed to work with a particular computer or motherboard model, by interfacing with various devices especially system chipset. Originally, BIOS firmware was stored in a ROM chip on the PC motherboard. In later computer systems, the BIOS contents are stored on flash memory so it can be rewritten without removing the chip from the motherboard. This allows easy, end-user updates to the BIOS firmware so new features can be added or bugs can be fixed, but it also creates a possibility for the computer to become infected with BIOS rootkits. Furthermore, a BIOS upgrade that fails could brick the motherboard. Unified Extensible Firmware Interface (UEFI) is a successor to the PC BIOS, aiming to address its technical limitations.[5] UEFI firmware may include legacy BIOS compatibility to maintain compatibility with operating systems and option cards that do not support UEFI native operation.[6][7][8] Since 2020, all PCs for Intel platforms no longer support legacy BIOS.[9] The last version of Microsoft Windows to officially support running on PCs which use legacy BIOS firmware is Windows 10 as Windows 11 requires a UEFI-compliant system (except for IoT Enterprise editions of Windows 11 since version 24H2[10]). /* C P / M B A S I C I / O S Y S T E M (B I O S) COPYRIGHT (C) GARY A. KILDALL JUNE, 1975 */ […] /* B A S I C D I S K O P E R A T I N G S Y S T E M (B D O S) COPYRIGHT (C) GARY A. KILDALL JUNE, 1975 */—An excerpt from the BDOS.PLM file header in the PL/M source code of CP/M 1.1 or 1.2 for Lawrence Livermore Laboratories (LLL)[2] The term BIOS (Basic Input/Output System) was created by Gary Kildall[11][12] and first appeared in the CP/M operating system in 1975,[2][3][12][13][14][15] describing the machine-specific part of CP/M loaded during boot time that interfaces directly with the hardware.[3] (A CP/M machine usually has only a simple boot loader in its ROM.) Versions of MS-DOS, PC DOS or DR-DOS contain a file called variously "IO.SYS", "IBMBIO.COM", "IBMBIO.SYS", or "DRBIOS.SYS"; this file is known as the "DOS BIOS" (also known as the "DOS I/O System") and contains the lower-level hardware-specific part of the operating system. Together with the underlying hardware-specific but operating system-independent "System BIOS", which resides in ROM, it represents the analogue to the "CP/M BIOS". The BIOS originally proprietary to the IBM PC has been reverse engineered by some companies (such as Phoenix Technologies) looking to create compatible systems. With the introduction of PS/2 machines, IBM divided the System BIOS into real- and protected-mode portions. The real-mode portion was meant to provide backward compatibility with existing operating systems such as DOS, and therefore was named "CBIOS" (for "Compatibility BIOS"), whereas the "ABIOS" (for "Advanced BIOS") provided new interfaces specifically suited for multitasking operating systems such as OS/2.[16] The BIOS of the original IBM PC and XT had no interactive user interface. Error codes or messages were displayed on the screen, or coded series of sounds were generated to signal errors when the power-on self-test (POST) had not proceeded to the point of successfully initializing a video display adapter. Options on the IBM PC and XT were set by switches and jumpers on the main board and on expansion cards. Starting around the mid-1990s, it became typical for the BIOS ROM to include a "BIOS configuration utility" (BCU[17]) or "BIOS setup utility", accessed at system power-up by a particular key sequence. This program allowed the user to set system configuration options, of the type formerly set using DIP switches, through an interactive menu system controlled through the keyboard. In the interim period, IBM-compatible PCs—including the IBM AT—held configuration settings in battery-backed RAM and used a bootable configuration program on floppy disk, not in the ROM, to set the configuration options contained in this memory. The floppy disk was supplied with the computer, and if it was lost the system settings could not be changed. The same applied in general to computers with an EISA bus, for which the configuration program was called an EISA Configuration Utility (ECU). A modern Wintel-compatible computer provides a setup routine essentially unchanged in nature from the ROM-resident BIOS setup utilities of the late 1990s; the user can configure hardware options using the keyboard and video display. The modern Wintel machine may store the BIOS configuration settings in flash ROM, perhaps the same flash ROM that holds the BIOS itself. See also: Option ROM Peripheral cards such as hard disk drive host bus adapters and video cards have their own firmware, and BIOS extension option ROM code may be a part of the expansion card firmware; that code provides additional capabilities in the BIOS. Code in option ROMs runs before the BIOS boots the operating system from mass storage. These ROMs typically test and initialize hardware, add new BIOS services, or replace existing BIOS services with their own services. For example, a SCSI controller usually has a BIOS extension ROM that adds support for hard drives connected through that controller. An extension ROM could in principle contain operating system, or it could implement an entirely different boot process such as network booting. Operation of an IBM-compatible computer system can be completely changed by removing or inserting an adapter card (or a ROM chip) that contains a BIOS extension ROM. The motherboard BIOS typically contains code for initializing and bootstrapping integrated display and integrated storage. The initialization process can involve the execution of code related to the device being initialized, for locating the device, verifying the type of device, then establishing base registers, setting pointers, establishing interrupt vector tables,[18] selecting paging modes which are ways for organizing available registers in devices, setting default values for accessing software routines related to interrupts,[19] and setting the device's configuration using default values.[20] In addition, plug-in adapter cards such as SCSI, RAID, network interface cards, and video cards often include their own BIOS (e.g. Video BIOS), complementing or replacing the system BIOS code for the given component. Even devices built into the motherboard can behave in this way; their option ROMs can be a part of the motherboard BIOS. An add-in card requires an option ROM if the card is not supported by the motherboard BIOS and the card needs to be initialized or made accessible through BIOS services before the operating system can be loaded (usually this means it is required in the boot process). An additional advantage of ROM on some early PC systems (notably including the IBM PCjr) was that ROM was faster than main system RAM. (On modern systems, the case is very much the reverse of this, and BIOS ROM code is usually copied ("shadowed") into RAM so it will run faster.) BIOS chips in a Dell 310 that were updated by replacing the chips Option ROMs normally reside on adapter cards. However, the original PC, and perhaps also the PC XT, have a spare ROM socket on the motherboard (the "system board" in IBM's terms) into which an option ROM can be inserted, and the four ROMs that contain the BASIC interpreter can also be removed and replaced with custom ROMs which can be option ROMs. The IBM PCjr is unique among PCs in having two ROM cartridge slots on the front. Cartridges in these slots map into the same region of the upper memory area used for option ROMs, and the cartridges can contain option ROM modules that the BIOS would recognize. The cartridges can also contain other types of ROM modules, such as BASIC programs, that are handled differently. One PCjr cartridge can contain several ROM modules of different types, possibly stored together in one ROM chip. The 8086 and 8088 start at physical address FFFF0h.[21] The 80286 starts at physical address FFFFF0h.[22] The 80386 and later x86 processors start at physical address FFFFFFF0h.[23][24][25] When the system is initialized, the first instruction of the BIOS appears at that address. If the system has just been powered up or the reset button was pressed ("cold boot"), the full power-on self-test (POST) is run. If Ctrl+Alt+Delete was pressed ("warm boot"), a special flag value stored in nonvolatile BIOS memory ("CMOS") tested by the BIOS allows bypass of the lengthy POST and memory detection. The POST identifies, tests and initializes system devices such as the CPU, chipset, RAM, motherboard, video card, keyboard, mouse, hard disk drive, optical disc drive and other hardware, including integrated peripherals. Early IBM PCs had a routine in the POST that would download a program into RAM through the keyboard port and run it.[26][27] This feature was intended for factory test or diagnostic purposes. After the motherboard BIOS completes its POST, most BIOS versions search for option ROM modules, also called BIOS extension ROMs, and execute them. The motherboard BIOS scans for extension ROMs in a portion of the "upper memory area" (the part of the x86 real-mode address space at and above address 0xA0000) and runs each ROM found, in order. To discover memory-mapped option ROMs, a BIOS implementation scans the real-mode address space from 0x0C0000 to 0x0F0000 on 2 KB (2,048 bytes) boundaries, looking for a two-byte ROM signature: 0x55 followed by 0xAA. In a valid expansion ROM, this signature is followed by a single byte indicating the number of 512-byte blocks the expansion ROM occupies in real memory, and the next byte is the option ROM's entry point (also known as its "entry offset"). If the ROM has a valid checksum, the BIOS transfers control to the entry address, which in a normal BIOS extension ROM should be the beginning of the extension's initialization routine. At this point, the extension ROM code takes over, typically testing and initializing the hardware it controls and registering interrupt vectors for use by post-boot applications. It may use BIOS services (including those provided by previously initialized option ROMs) to provide a user configuration interface, to display diagnostic information, or to do anything else that it requires. An option ROM should normally return to the BIOS after completing its initialization process. Once (and if) an option ROM returns, the BIOS continues searching for more option ROMs, calling each as it is found, until the entire option ROM area in the memory space has been scanned. It is possible that an option ROM will not return to BIOS, pre-empting the BIOS's boot sequence altogether. Boot process After the POST completes and, in a BIOS that supports option ROMs, after the option ROM scan is completed and all detected ROM modules with valid checksums have been called, the BIOS calls interrupt 19h to start boot processing. Post-boot, programs loaded can also call interrupt 19h to reboot the system, but they must be careful to disable interrupts and other asynchronous hardware processes that may interfere with the BIOS rebooting process, or else the system may hang or crash while it is rebooting. When interrupt 19h is called, the BIOS attempts to locate boot loader software on a "boot device", such as a hard disk, a floppy disk, CD, or DVD. It loads and executes the first boot software it finds, giving it control of the PC.[28] The BIOS uses the boot devices set in Nonvolatile BIOS memory (CMOS), or, in the earliest PCs, DIP switches. The BIOS checks each device in order to see if it is bootable by attempting to load the first sector (boot sector). If the sector cannot be read, the BIOS proceeds to the next device. If the sector is read successfully, some BIOSes will also check for the boot sector signature 0x55 0xAA in the last two bytes of the sector (which is 512 bytes long), before accepting a boot sector and considering the device bootable.[b] When a bootable device is found, the BIOS transfers control to the loaded sector. The BIOS does not interpret the contents of the boot sector other than to possibly check for the boot sector signature in the last two bytes. Interpretation of data structures like partition tables and BIOS Parameter Blocks is done by the boot program in the boot sector itself or by other programs loaded through the boot process. A non-disk device such as a network adapter attempts booting by a procedure that is defined by its option ROM or the equivalent integrated into the motherboard BIOS ROM. As such, option ROMs may also influence or supplant the boot process defined by the motherboard BIOS ROM. With the El Torito optical media boot standard, the optical drive actually emulates a 3.5" high-density floppy disk to the BIOS for boot purposes. Reading the "first sector" of a CD-ROM or DVD-ROM is not a simply defined operation like it is on a floppy disk or a hard disk. Furthermore, the complexity of the medium makes it difficult to write a useful boot program in one sector. The bootable virtual floppy disk can contain software that provides access to the optical medium in its native format. If an expansion ROM wishes to change the way the system boots (such as from a network device or a SCSI adapter) in a cooperative way, it can use the BIOS Boot Specification (BBS) API to register its ability to do so. Once the expansion ROMs have registered using the BBS APIs, the user can select among the available boot options from within the BIOS's user interface. This is why most BBS compliant PC BIOS implementations will not allow the user to enter the BIOS's user interface until the expansion ROMs have finished executing and registering themselves with the BBS API.[citation needed] Also, if an expansion ROM wishes to change the way the system boots unilaterally, it can simply hook interrupt 19h or other interrupts normally called from interrupt 19h, such as interrupt 13h, the BIOS disk service, to intercept the BIOS boot process. Then it can replace the BIOS boot process with one of its own, or it can merely modify the boot sequence by inserting its own boot actions into it, by preventing the BIOS from detecting certain devices as bootable, or both. Before the BIOS Boot Specification was promulgated, this was the only way for expansion ROMs to implement boot capability for devices not supported for booting by the native BIOS of the motherboard.[citation needed] The user can select the boot priority implemented by the BIOS. For example, most computers have a hard disk that is bootable, but sometimes there is a removable-media drive that has higher boot priority, so the user can cause a removable disk to be booted. In most modern BIOSes, the boot priority order can be configured by the user. In older BIOSes, limited boot priority options are selectable; in the earliest BIOSes, a fixed priority scheme was implemented, with floppy disk drives first, fixed disks (i.e., hard disks) second, and typically no other boot devices supported, subject to modification of these rules by installed option ROMs. The BIOS in an early PC also usually would only boot from the first floppy disk drive or the first hard disk drive, even if there were two drives installed. On the original IBM PC and XT, if no bootable disk was found, the BIOS would try to start ROM BASIC with the interrupt call to interrupt 18h. Since few programs used BASIC in ROM, clone PC makers left it out; then a computer that failed to boot from a disk would display "No ROM BASIC" and halt (in response to interrupt 18h). Later computers would display a message like "No bootable disk found"; some would prompt for a disk to be inserted and a key to be pressed to retry the boot process. A modern BIOS may display nothing or may automatically enter the BIOS configuration utility when the boot process fails. The environment for the boot program is very simple: the CPU is in real mode and the general-purpose and segment registers are undefined, except SS, SP, CS, and DL. CS:IP always points to physical address 0x07C00. What values CS and IP actually have is not well defined. Some BIOSes use a CS:IP of 0x0000:0x7C00 while others may use 0x07C0:0x0000.[29] Because boot programs are always loaded at this fixed address, there is no need for a boot program to be relocatable. DL may contain the drive number, as used with interrupt 13h, of the boot device. SS:SP points to a valid stack that is presumably large enough to support hardware interrupts, but otherwise SS and SP are undefined. (A stack must be already set up in order for interrupts to be serviced, and interrupts must be enabled in order for the system timer-tick interrupt, which BIOS always uses at least to maintain the time-of-day count and which it initializes during POST, to be active and for the keyboard to work. The keyboard works even if the BIOS keyboard service is not called; keystrokes are received and placed in the 15-character type-ahead buffer maintained by BIOS.) The boot program must set up its own stack, because the size of the stack set up by BIOS is unknown and its location is likewise variable; although the boot program can investigate the default stack by examining SS:SP, it is easier and shorter to just unconditionally set up a new stack.[30] At boot time, all BIOS services are available, and the memory below address 0x00400 contains the interrupt vector table. BIOS POST has initialized the system timers, interrupt controller(s), DMA controller(s), and other motherboard/chipset hardware as necessary to bring all BIOS services to ready status. DRAM refresh for all system DRAM in conventional memory and extended memory, but not necessarily expanded memory, has been set up and is running. The interrupt vectors corresponding to the BIOS interrupts have been set to point at the appropriate entry points in the BIOS, hardware interrupt vectors for devices initialized by the BIOS have been set to point to the BIOS-provided ISRs, and some other interrupts, including ones that BIOS generates for programs to hook, have been set to a default dummy ISR that immediately returns. The BIOS maintains a reserved block of system RAM at addresses 0x00400–0x004FF with various parameters initialized during the POST. All memory at and above address 0x00500 can be used by the boot program; it may even overwrite itself.[31][32] The BIOS ROM is customized to the particular manufacturer's hardware, allowing low-level services (such as reading a keystroke or writing a sector of data to diskette) to be provided in a standardized way to programs, including operating systems. For example, an IBM PC might have either a monochrome or a color display adapter (using different display memory addresses and hardware), but a single, standard, BIOS system call may be invoked to display a character at a specified position on the screen in text mode or graphics mode. The BIOS provides a small library of basic input/output functions to operate peripherals (such as the keyboard, rudimentary text and graphics display functions and so forth). When using MS-DOS, BIOS services could be accessed by an application program (or by MS-DOS) by executing an interrupt 13h interrupt instruction to access disk functions, or by executing one of a number of other documented BIOS interrupt calls to access video display, keyboard, cassette, and other device functions. Operating systems and executive software that are designed to supersede this basic firmware functionality provide replacement software interfaces to application software. Applications can also provide these services to themselves. This began even in the 1980s under MS-DOS, when programmers observed that using the BIOS video services for graphics display were very slow. To increase the speed of screen output, many programs bypassed the BIOS and programmed the video display hardware directly. Other graphics programmers, particularly but not exclusively in the demoscene, observed that there were technical capabilities of the PC display adapters that were not supported by the IBM BIOS and could not be taken advantage of without circumventing it. Since the AT-compatible BIOS ran in Intel real mode, operating systems that ran in protected mode on 286 and later processors required hardware device drivers compatible with protected mode operation to replace BIOS services. In modern PCs running modern operating systems (such as Windows and Linux) the BIOS interrupt calls are used only during booting and initial loading of operating systems. Before the operating system's first graphical screen is displayed, input and output are typically handled through BIOS. A boot menu such as the textual menu of Windows, which allows users to choose an operating system to boot, to boot into the safe mode, or to use the last known good configuration, is displayed through BIOS and receives keyboard input through BIOS.[4] Many modern PCs can still boot and run legacy operating systems such as MS-DOS or DR-DOS that rely heavily on BIOS for their console and disk I/O, providing that the system has a BIOS, or a CSM-capable UEFI firmware. Intel processors have reprogrammable microcode since the P6 microarchitecture.[33][34][35] AMD processors have reprogrammable microcode since the K7 microarchitecture. The BIOS contain patches to the processor microcode that fix errors in the initial processor microcode; microcode is loaded into processor's SRAM so reprogramming is not persistent, thus loading of microcode updates is performed each time the system is powered up. Without reprogrammable microcode, an expensive processor swap would be required;[36] for example, the Pentium FDIV bug became an expensive fiasco for Intel as it required a product recall because the original Pentium processor's defective microcode could not be reprogrammed. Operating systems can update main processor microcode also.[37][38] Some BIOSes contain a software licensing description table (SLIC), a digital signature placed inside the BIOS by the original equipment manufacturer (OEM), for example Dell. The SLIC is inserted into the ACPI data table and contains no active code.[39][40] Computer manufacturers that distribute OEM versions of Microsoft Windows and Microsoft application software can use the SLIC to authenticate licensing to the OEM Windows Installation disk and system recovery disc containing Windows software. Systems with a SLIC can be preactivated with an OEM product key, and they verify an XML formatted OEM certificate against the SLIC in the BIOS as a means of self-activating (see System Locked Preinstallation, SLP). If a user performs a fresh install of Windows, they will need to have possession of both the OEM key (either SLP or COA) and the digital certificate for their SLIC in order to bypass activation.[39] This can be achieved if the user performs a restore using a pre-customised image provided by the OEM. Power users can copy the necessary certificate files from the OEM image, decode the SLP product key, then perform SLP activation manually. Some BIOS implementations allow overclocking, an action in which the CPU is adjusted to a higher clock rate than its manufacturer rating for guaranteed capability. Overclocking may, however, seriously compromise system reliability in insufficiently cooled computers and generally shorten component lifespan. Overclocking, when incorrectly performed, may also cause components to overheat so quickly that they mechanically destroy themselves.[41] Some older operating systems, for example MS-DOS, rely on the BIOS to carry out most input/output tasks within the PC.[42] Calling real mode BIOS services directly is inefficient for protected mode (and long mode) operating systems. BIOS interrupt calls are not used by modern multitasking operating systems after they initially load. In the 1990s, BIOS provided some protected mode interfaces for Microsoft Windows and Unix-like operating systems, such as Advanced Power Management (APM), Plug and Play BIOS, Desktop Management Interface (DMI), VESA BIOS Extensions (VBE), e820 and MultiProcessor Specification (MPS). Starting from the year 2000, most BIOSes provide ACPI, SMBIOS, VBE and e820 interfaces for modern operating systems.[43][44][45][46][47] After operating systems load, the System Management Mode code is still running in SMRAM. Since 2010, BIOS technology is in a transitional process toward UEFI.[5] Historically, the BIOS in the IBM PC and XT had no built-in user interface. The BIOS versions in earlier PCs (XT-class) were not software configurable; instead, users set the options via DIP switches on the motherboard. Later computers, including most IBM-compatibles with 80286 CPUs, had a battery-backed nonvolatile BIOS memory (CMOS RAM chip) that held BIOS settings.[48] These settings, such as video-adapter type, memory size, and hard-disk parameters, could only be configured by running a configuration program from a disk, not built into the ROM. A special "reference diskette" was inserted in an IBM AT to configure settings such as memory size.[49] Early BIOS versions did not have passwords or boot-device selection options. The BIOS was hard-coded to boot from the first floppy drive, or, if that failed, the first hard disk. Access control in early AT-class machines was by a physical keylock switch (which was not hard to defeat if the computer case could be opened). Anyone who could switch on the computer could boot it.[citation needed] Later, 386-class computers started integrating the BIOS setup utility in the ROM itself, alongside the BIOS code; these computers usually boot into the BIOS setup utility if a certain key or key combination is pressed, otherwise the BIOS POST and boot process are executed. Award BIOS setup utility on a standard PC A modern BIOS setup utility has a text user interface (TUI) or graphical user interface (GUI) accessed by pressing a certain key on the keyboard when the PC starts. Usually, the key is advertised for short time during the early startup, for example "Press DEL to enter Setup". The actual key depends on specific hardware. The settings key is most often Delete (Acer, ASRock, Asus PC, ECS, Gigabyte, MSI, Zotac) and F2 (Asus motherboard, Dell, Lenovo laptop, Origin PC, Samsung, Toshiba), but it can also be F1 (Lenovo desktop) and F10 (HP).[50] Features present in the BIOS setup utility typically include: Configuring, enabling and disabling the hardware components Setting the system time Setting the boot order Setting various passwords, such as a password for securing access to the BIOS user interface and preventing malicious users from booting the system from unauthorized portable storage devices, or a password for booting the system Main article: Hardware monitoring A modern BIOS setup screen often features a PC Health Status or a Hardware Monitoring tab, which directly interfaces with a Hardware Monitor chip of the mainboard.[51] This makes it possible to monitor CPU and chassis temperature, the voltage provided by the power supply unit, as well as monitor and control the speed of the fans connected to the motherboard. Once the system is booted, hardware monitoring and computer fan control is normally done directly by the Hardware Monitor chip itself, which can be a separate chip, interfaced through I²C or SMBus, or come as a part of a Super I/O solution, interfaced through Industry Standard Architecture (ISA) or Low Pin Count (LPC).[52] Some operating systems, like NetBSD with envsys and OpenBSD with sysctl hw.sensors, feature integrated interfacing with hardware monitors. However, in some circumstances, the BIOS also provides the underlying information about hardware monitoring through ACPI, in which case, the operating system may be using ACPI to perform hardware monitoring.[53][54] BIOS replacement kit for a Dell 310 from the late 1980s. Included are two chips, a plastic holder for the chips, and a IC extractor. In modern PCs the BIOS is stored in rewritable EEPROM[55] or NOR flash memory,[56] allowing the contents to be replaced and modified. This rewriting of the contents is sometimes termed flashing. It can be done by a special program, usually provided by the system's manufacturer, or at POST, with a BIOS image in a hard drive or USB flash drive. A file containing such contents is sometimes termed "a BIOS image". A BIOS might be reflashed in order to upgrade to a newer version to fix bugs or provide improved performance or to support newer hardware. Some computers also support updating the BIOS via an update floppy disk or a special partition on the hard drive.[57] American Megatrends BIOS 686. This BIOS chip is housed in a PLCC package in a socket. The original IBM PC BIOS (and cassette BASIC) was stored on mask-programmed read-only memory (ROM) chips in sockets on the motherboard. ROMs could be replaced,[58] but not altered, by users. To allow for updates, many compatible computers used re-programmable BIOS memory devices such as EPROM, EEPROM and later flash memory (usually NOR flash) devices. According to Robert Braver, the president of the BIOS manufacturer Micro Firmware, Flash BIOS chips became common around 1995 because the electrically erasable PROM (EEPROM) chips are cheaper and easier to program than standard ultraviolet erasable PROM (EPROM) chips. Flash chips are programmed (and re-programmed) in-circuit, while EPROM chips need to be removed from the motherboard for re-programming.[59] BIOS versions are upgraded to take advantage of newer versions of hardware and to correct bugs in previous revisions of BIOSes.[60] Beginning with the IBM AT, PCs supported a hardware clock settable through BIOS. It had a century bit which allowed for manually changing the century when the year 2000

happened. Most BIOS revisions created in 1995 and nearly all BIOS revisions in 1997 supported the year 2000 by setting the century bit automatically when the clock rolled past midnight, 31 December 1999.[61] The first flash chips were attached to the ISA bus. Starting in 1998, the BIOS flash moved to the LPC bus, following a new standard implementation known as "firmware hub" (FWH). In 2005, the BIOS flash memory moved to the SPI bus.[62] The size of the BIOS, and the capacity of the ROM, EEPROM, or other media it may be stored on, has increased over time as new features have been added to the code; BIOS versions now exist with sizes up to 32 megabytes. For contrast, the original IBM PC BIOS was contained in an 8 KB mask ROM. Some modern motherboards are including even bigger NAND flash memory ICs on board which are capable of storing whole compact operating systems, such as some Linux distributions. For example, some ASUS notebooks included Splashtop OS embedded into their NAND flash memory ICs.[63] However, the idea of including an operating system along with BIOS in the ROM of a PC is not new; in the 1980s, Microsoft offered a ROM option for MS-DOS, and it was included in the ROMs of some PC clones such as the Tandy 1000 HX. Another type of firmware chip was found on the IBM PC AT and early compatibles. In the AT, the keyboard interface was controlled by a microcontroller with its own programmable memory. On the IBM AT, that was a 40-pin socketed device, while some manufacturers used an EPROM version of this chip which resembled an EPROM. This controller was also assigned the A20 gate function to manage memory above the one-megabyte range; occasionally an upgrade of this "keyboard BIOS" was necessary to take advantage of software that could use upper memory.[citation needed] The BIOS may contain components such as the Memory Reference Code (MRC), which is responsible for the memory initialization (e.g. SPD and memory timings initialization).[64]:8[65] Modern BIOS[66] includes Intel Management Engine or AMD Platform Security Processor firmware. Comparison of different BIOS implementations Company AwardBIOS AMIBIOS Insyde SeaBIOS License Proprietary Proprietary Proprietary LGPL v3 Maintained / developed Terminated Terminated Terminated Yes 32-bit PCI BIOS calls Yes Yes Yes AHCI Yes Yes Yes Yes APM Yes Yes Yes (1.2) Yes (1.2) BBS Yes Yes Yes Yes Boot menu Yes Yes Yes Yes Compression Yes (LHA[67]) Yes (LHA) Yes (RLE) Yes (LZMA) CMOS Yes Yes Yes Yes EDD Yes Yes Yes Yes ESCD Yes Yes ? No Flash from ROM ? Yes ? No Language Assembly Assembly Assembly C LBA Yes (48) Yes (48) Yes (48) MultiProcessor Specification Yes Yes Yes Yes Option ROM Yes Yes Yes Yes Password Yes Yes Yes No PMM ? Yes ? Yes Setup screen Yes Yes Yes No SMBIOS Yes Yes Yes Yes Splash screen Yes (EPA)[68] Yes (PCX) Yes Yes (BMP, JPG) TPM Unknown Yes Unknown Some USB booting Yes Yes Yes Yes USB hub ? ? ? Yes USB keyboard Yes Yes Yes Yes USB mouse Yes Yes Yes Yes IBM published the entire listings of the BIOS for its original PC, PC XT, PC AT, and other contemporary PC models, in an appendix of the IBM PC Technical Reference Manual for each machine type. The effect of the publication of the BIOS listings is that anyone can see exactly what a definitive BIOS does and how it does it. Compaq Portable 386 BIOS In May 1984, Phoenix Software Associates released its first ROM-BIOS. This BIOS enabled OEMs to build essentially fully compatible clones without having to reverse-engineer the IBM PC BIOS themselves, as Compaq had done for the Portable; it also helped fuel the growth in the PC-compatibles industry and sales of non-IBM versions of DOS.[69] The first American Megatrends (AMI) BIOS was released in 1986. New standards grafted onto the BIOS are usually without complete public documentation or any BIOS listings. As a result, it is not as easy to learn the intimate details about the many non-IBM additions to BIOS as about the core BIOS services. Many PC motherboard suppliers licensed the BIOS "core" and toolkit from a commercial third party, known as an "independent BIOS vendor" or IBV. The motherboard manufacturer then customized this BIOS to suit its own hardware. For this reason, updated BIOSes are normally obtained directly from the motherboard manufacturer. Major IBVs included American Megatrends (AMI), Insyde Software, Phoenix Technologies, and Byosoft. Microid Research and Award Software were acquired by Phoenix Technologies in 1998; Phoenix later phased out the Award brand name (although Award Software is still credited in newer AwardBIOS versions and in UEFI firmwares).[when?] General Software, which was also acquired by Phoenix in 2007, sold BIOS for embedded systems based on Intel processors. SeaBIOS is an open-source BIOS implementation. The open-source community increased their effort to develop a replacement for proprietary BIOSes and their future incarnations with an open-sourced counterparts. Open Firmware was an early attempt to make an open specification for boot firmware. It was initially endorsed by IEEE in its IEEE 1275-1994 standard but was withdrawn in 2005.[70][71] Later examples include the OpenBIOS, coreboot and libreboot projects. AMD provided product specifications for some chipsets using coreboot, and Google is sponsoring the project. Motherboard manufacturer Tyan offers coreboot next to the standard BIOS with their Opteron line of motherboards. This section needs additional citations for verification. Please help improve this article by adding citations to reliable sources in this section. Unsourced material may be challenged and removed. (March 2019) (Learn how and when to remove this message) Gigabyte DualBIOS PLCC32 A detached BIOS chip EEPROM and flash memory chips are advantageous because they can be easily updated by the user; it is customary for hardware manufacturers to issue BIOS updates to upgrade their products, improve compatibility and remove bugs. However, this advantage had the risk that an improperly executed or aborted BIOS update could render the computer or device unusable. To avoid these situations, more recent BIOSes use a "boot block"; a portion of the BIOS which runs first and must be updated separately. This code verifies if the rest of the BIOS is intact (using hash checksums or other methods) before transferring control to it. If the boot block detects any corruption in the main BIOS, it will typically warn the user that a recovery process must be initiated by booting from removable media (floppy, CD or USB flash drive) so the user can try flashing the BIOS again. Some motherboards have a backup BIOS (sometimes referred to as DualBIOS boards) to recover from BIOS corruptions. There are at least five known viruses that attack the BIOS. Two of which were for demonstration purposes. The first one found in the wild was Mebromi, targeting Chinese users. The first BIOS virus was BIOS Meningitis, which instead of erasing BIOS chips it infected them. BIOS Meningitis was relatively harmless, compared to a virus like CIH. The second BIOS virus was CIH, also known as the "Chernobyl Virus", which was able to erase flash ROM BIOS content on compatible chipsets. CIH appeared in mid-1998 and became active in April 1999. Often, infected computers could no longer boot, and people had to remove the flash ROM IC from the motherboard and reprogram it. CIH targeted the then-widespread Intel i430TX motherboard chipset and took advantage of the fact that the Windows 9x operating systems, also widespread at the time, allowed direct hardware access to all programs. Modern systems are not vulnerable to CIH because of a variety of chipsets being used which are incompatible with the Intel i430TX chipset, and also other flash ROM IC types. There is also extra protection from accidental BIOS rewrites in the form of boot blocks which are protected from accidental overwrite or dual and quad BIOS equipped systems which may, in the event of a crash, use a backup BIOS. Also, all modern operating systems such as FreeBSD, Linux, macOS, Windows NT-based Windows OS like Windows 2000, Windows XP and newer, do not allow user-mode programs to have direct hardware access using a hardware abstraction layer.[72] As a result, as of 2008, CIH has become essentially harmless, at worst causing annoyance by infecting executable files and triggering antivirus software. Other BIOS viruses remain possible, however;[73] since most Windows home users without Windows Vista/7's UAC run all applications with administrative privileges, a modern CIH-like virus could in principle still gain access to hardware without first using an exploit.[citation needed] The operating system OpenBSD prevents all users from having this access and the grsecurity patch for the Linux kernel also prevents this direct hardware access by default, the difference being an attacker requiring a much more difficult kernel level exploit or reboot of the machine.[citation needed] The third BIOS virus was a technique presented by John Heasman, principal security consultant for UK-based Next-Generation Security Software. In 2006, at the Black Hat Security Conference, he showed how to elevate privileges and read physical memory, using malicious procedures that replaced normal ACPI functions stored in flash memory.[74] The fourth BIOS virus was a technique called "Persistent BIOS infection." It appeared in 2009 at the CanSecWest Security Conference in Vancouver, and at the SyScan Security Conference in Singapore. Researchers Anibal Sacco[75] and Alfredo Ortega, from Core Security Technologies, demonstrated how to insert malicious code into the decompression routines in the BIOS, allowing for nearly full control of the PC at start-up, even before the operating system is booted. The proof-of-concept does not exploit a flaw in the BIOS implementation, but only involves the normal BIOS flashing procedures. Thus, it requires physical access to the machine, or for the user to be root. Despite these requirements, Ortega underlined the profound implications of his and Sacco's discovery: "We can patch a driver to drop a fully working rootkit. We even have a little code that can remove or disable antivirus."[76] Mebromi is a trojan which targets computers with AwardBIOS, Microsoft Windows, and antivirus software from two Chinese companies: Rising Antivirus and Jiangmin KV Antivirus.[77][78][79] Mebromi installs a rootkit which infects the Master boot record. In a December 2013 interview with 60 Minutes, Deborah Plunkett, Information Assurance Director for the US National Security Agency claimed the NSA had uncovered and thwarted a possible BIOS attack by a foreign nation state, targeting the US financial system.[80] The program cited anonymous sources alleging it was a Chinese plot.[80] However follow-up articles in The Guardian,[81] The Atlantic,[82] Wired[83] and The Register[84] refuted the NSA's claims. Newer Intel platforms have Intel Boot Guard (IBG) technology enabled, this technology will check the BIOS digital signature at startup, and the IBG public key is fused into the PCH. End users can't disable this function. For comparable software on other computer systems, see booting. Unified Extensible Firmware Interface (UEFI) supplements the BIOS in many new machines. Initially written for the Intel Itanium architecture, UEFI is now available for x86 and Arm platforms; the specification development is driven by the Unified EFI Forum, an industry special interest group. EFI booting has been supported in only Microsoft Windows versions supporting GPT,[85] the Linux kernel 2.6.1 and later, and macOS on Intel-based Macs.[86] As of 2014[update], new PC hardware predominantly ships with UEFI firmware. The architecture of the rootkit safeguard can also prevent the system from running the user's own software changes, which makes UEFI controversial as a legacy BIOS replacement in the open hardware community. Also, Windows 11 requires UEFI to boot,[87] with the exception of IoT Enterprise editions of Windows 11.[10] UEFI is required for devices shipping with Windows 8[88][89] and above. After the popularity of UEFI in 2010s, the older BIOS that supported BIOS interrupt calls was renamed to "legacy BIOS".[citation needed] Other alternatives to the functionality of the "Legacy BIOS" in the x86 world include coreboot and libreboot. Some servers and workstations use a platform-independent Open Firmware (IEEE-1275) based on the Forth programming language; it is included with Sun's SPARC computers, IBM's RS/6000 line, and other PowerPC systems such as the CHRP motherboards, along with the x86-based OLPC XO-1. As of at least 2015, Apple has removed legacy BIOS support from the UEFI monitor in Intel-based Macs. As such, the BIOS utility no longer supports the legacy option, and prints "Legacy mode not supported on this system". In 2017, Intel announced that it would remove legacy BIOS support by 2020. Since 2019, new Intel platform OEM PCs no longer support the legacy option.[90] Double boot Extended System Configuration Data (ESCD) Input/Output Control System ACPI (Advanced Configuration and Power Interface) Ralf Brown's Interrupt List (RBIL) – interrupts, calls, interfaces, data structures, memory and port addresses, and processor opcodes for the x86 architecture System Management BIOS (SMBIOS) UEFI (Unified Extensible Firmware Interface) Das U-Boot, often used on embedded systems ^ Although the term BIOS predates 1981, the standard for IBM PC–compatible computers started with the release of the original IBM Personal Computer. ^ The signature at offset +0x1FE in boot sectors is 0x55 0xAA, that is 0x55 at offset +0x1FE and 0xAA at offset +0x1FF. Since little-endian representation must be assumed in the context of IBM PC–compatible compatible, this can be written as 16-bit word 0xAA55 in programs for x86 processors (note the swapped order), whereas it would have to be written as 0x55AA in programs for other CPU architectures using a big-endian representation. Since this has been mixed up numerous times in books and even in original Microsoft reference documents, this article uses the offset-based byte-wise on-disk representation to avoid any possible misinterpretation. ^ "Ref — System BIOS". PCGuide. Archived from the original on 2014-12-21. Retrieved 2014-12-06. ^ a b c Kildall, Gary Arlen (June 1975), CP/M 1.1 or 1.2 BIOS and BDOS for Lawrence Livermore Laboratories ^ a b c Kildall, Gary Arlen (January 1980). "The History of CP/M - The Evolution of an Industry: One Person's Viewpoint" (Vol. 5, No. 1, Number 41 ed.). Dr. Dobb's Journal of Computer Calisthenics & Orthodontia. pp. 6–7. Archived from the original on 2016-11-24. Retrieved 2013-06-03. ^ a b "Booting · Linux Inside". 0xax.gitbooks.io. Retrieved 2020-11-10. ^ a b Bradley, Tony. "R.I.P. BIOS: A UEFI Primer". PCWorld. Archived from the original on 2014-01-27. Retrieved 2014-01-27. ^ "Unified Extensible Firmware Interface". Intel. ^ "UEFI". OSDev.org. ^ "Intel® Platform Innovation Framework for EFI Compatibility Support Module Specification (revision 0.97)" (PDF). Intel. 2007-09-04. Retrieved 2013-10-06. ^ "Removal of Legacy Boot Support for Intel Platforms Technical Advisory". Retrieved 2024-07-25. ^ a b "Minimum System Requirements for Windows IoT Enterprise". Microsoft Learn. 2024-05-22. Retrieved 2024-06-07. ^ Swaine, Michael (1997-04-01). "Gary Kildall and Collegial Entrepreneurship". Dr. Dobb's Journal. Archived from the original on 2007-01-24. Retrieved 2006-11-20. ^ a b "IEEE Milestone in Electrical Engineering and Computing - CP/M - Microcomputer Operating System, 1974" (PDF). Computer History Museum. 2014-04-25. Archived (PDF) from the original on 2019-04-03. Retrieved 2019-04-03. ^ Shustek, Len (2016-08-02). "In His Own Words: Gary Kildall". Remarkable People. Computer History Museum. Archived from the original on 2016-12-17. ^ Killian, A. Joseph "Joe" (2001). "Gary Kildall's CP/M: Some early CP/M history - 1976-1977". Thomas "Todd" Fischer, IMSAI. Archived from the original on 2012-12-29. Retrieved 2013-06-03. ^ Fraley, Bob; Spicer, Dag (2007-01-26). "Oral History of Joseph Killian, Interviewed by: Bob Fraley, Edited by: Dag Spicer, Recorded: January 26, 2007, Mountain View, California, CHM Reference number: X3879.2007" (PDF). Computer History Museum. Archived from the original (PDF) on 2014-07-14. Retrieved 2013-06-03. ^ Glass, Brett (1989). "The IBM PC BIOS". Byte: 303–310. Retrieved 2021-12-31. ^ "HP BIOS Configuration Utility". Hewlett-Packard. 2013. Archived from the original on 2015-01-12. Retrieved 2015-01-12. ^ Phoenix Technologies, Ltd. (June 1991). System BIOS for IBM PCs, Compatibles, and EISA Computers — The Complete Guide to ROM-Based System Software. Phoenix Technical Reference Series (2nd ed.). Amsterdam: Addison Wesley Publishing Company, Inc. ISBN 0-201-57760-7. ^ Phoenix Technologies, Ltd. (1989) [1987]. System BIOS for IBM PC/XT/AT Computers and Compatibles — The Complete Guide to ROM-Based System Software. Phoenix Technical Reference Series (1st ed.). Addison Wesley Publishing Company, Inc. ISBN 0-201-51806-6. ^ Sanchez, Julio; Canton, Maria P. (2003-02-26). The PC Graphics Handbook. CRC Press. ISBN 978-0-203-01053-2. ^ "iAPX 86,88 User's Manual" (PDF). Intel. 1981. System Reset, p. 2-29, table 2-4. Retrieved 2018-04-15. ^ "AMD 80286 Datasheet" (PDF). AMD. 1985. p. 13. the 286 begins execution in real mode with the instruction at physical location FFFFF0H. ^ "80386 Programmer's Reference Manual" (PDF). Intel. 1990. Section 10.1 Processor State After Reset, pages 10-1 - 10.3. ^ "80386 Programmer's Reference Manual" (PDF). Intel. 1990. Section 10.2.3 First Instruction, p. 10-4. Retrieved 2013-11-03. Execution begins with the instruction addressed by the initial contents of the CS and IP registers. To allow the initialization software to be placed in a ROM at the top of the address space, the high 12 bits of addresses issued for the code segment are set, until the first instruction which loads the CS register, such as a far jump or call. As a result, instruction fetching begins from address 0FFFFFFF0H. ^ "Intel® 64 and IA-32 Architectures Software Developer's Manual" (PDF). Intel. May 2012. Section 9.1.4 First Instruction Executed, p. 2611. Archived from the original (PDF) on 2012-08-08. Retrieved 2012-08-23. The first instruction that is fetched and executed following a hardware reset is located at physical address FFFFFFF0h. This address is 16 bytes below the processor's uppermost physical address. The EPROM containing the software-initialization code must be located at this address. ^ page 5-27 IBM Personal Computer Hardware Reference Library Technical Reference, 1984, publication number 6361459 ^ "IBM 5162 PC XT286 TechRef 68X2537 Technical Reference manual" (PDF). August 1986. p. 35 (System BIOS A-5). Archived (PDF) from the original on 2014-12-11. Retrieved 2014-12-11. ^ How StuffWorks: What BIOS Does Archived 2008-02-07 at the Wayback Machine. ^ Akeljic, Bekir (2017-01-01). "BIOS BASIC INPUT/ OUTPUT SYSTEM BIOS FUNCTIONS AND MODIFICATIONS". BIOS: 12. Archived from the original on 2022-08-08. Retrieved 2022-08-08 – via INTERNATIONAL UNIVERSITY TRAVNIKFACULITY OF INFORMATION TECHNOLOGY TRAVNIKSOFTWARE PROGRAMMING. ^ "BIOS - CodeDocs". codedocs.org. Retrieved 2022-08-08. ^ "Memory Layout and Memory Map". flint.cs.yale.edu. Retrieved 2022-08-08. ^ "BIOS Data ACPI Table (BDAT)" (PDF). Interface Specification. 4 (5): 67. 2020. Archived (PDF) from the original on 2021-07-03. Retrieved 2022-08-08. ^ Stiller, Andreas; Paul, Matthias R. (1996-05-12). "Prozessorgeflüster". c't – magazin für computertechnik. Trends & News / aktuell - Prozessoren (in German). Vol. 1996, no. 6. Verlag Heinz Heise GmbH & Co KG. p. 20. ISSN 0724-8679. Archived from the original on 2017-08-28. Retrieved 2017-08-28. ^ Mueller, Scott (2001-06-08). Processor Update Feature | Microprocessor Types and Specifications. InformIT. Archived from the original on 2014-04-16. Retrieved 2014-04-15. ^ "Linux* Processor Microcode Data File". Download Center. Downloadcenter.intel.com. 2009-09-23. Archived from the original on 2014-04-16. Retrieved 2014-04-15. ^ Scott Mueller, Upgrading and repairing PCs 15th edition, Que Publishing, 2003 ISBN 0-7897-2974-1, pages 109-110 ^ "KB4100347: Intel microcode updates". support.microsoft.com. Retrieved 2020-09-20. ^ "Microcode - Debian Wiki". wiki.debian.org. Retrieved 2020-09-19. ^ a b "How SLP and SLIC Works". guytechie.com. 2010-02-25. Archived from the original on 2015-02-03. Retrieved 2015-02-03. ^ "Create and add an OEM ACPI SLIC table module to a congatec BIOS" (PDF). congatec.com. 2011-06-16. Archived (PDF) from the original on 2014-08-02. Retrieved 2015-02-03. ^ Whitson Gordon (2014-01-13). "A Beginner's Introduction to Overclocking Your Intel Processor". Lifehacker. Gawker Media. Archived from the original on 2014-12-07. Retrieved 2014-12-06. ^ Smart Computing Article - What Is The BIOS? Archived 2012-03-10 at the Wayback Machine - Computing Basics July 1994 • Vol.5 Issue 7 ^ "What is ACPI (Advanced Configuration and Power Interface)? - Definition from WhatIs.com". SearchWindowsServer. Retrieved 2020-09-18. ^ "Changing hardware abstraction layer in Windows 2000 / XP – Smallvoid.com". 2001-01-15. Retrieved 2020-09-18. ^ "What is ACPI?". www.spo-comm.de. Retrieved 2020-09-18. ^ lorihollasch. "Support for headless systems - Windows drivers". docs.microsoft.com. Retrieved 2020-12-05. ^ "Memory Map (x86) - OSDev Wiki". wiki.osdev.org. Retrieved 2020-12-11. ^ Torres, Gabriel (2004-11-24). "Introduction and Lithium Battery". Replacing the Motherboard Battery. hardwaresecrets.com. Archived from the original on 2013-12-24. Retrieved 2013-06-20. ^ Beales, R. P. (2006-08-11). PC Systems, Installation and Maintenance. Routledge. ISBN 978-1-136-37441-8. ^ "How to Enter the BIOS on Any PC: Access Keys by Manufacturer". Tom's Hardware. 2022-02-04. ^ Constantine A. Murenin (2010-05-21). "11.1. Interfacing from the BIOS". OpenBSD Hardware Sensors – Environmental Monitoring and Fan Control (MMath thesis). University of Waterloo: UWSpace. hdl:10012/5234. Document ID: ab71498b6b1a60ff817b29d56997a418. ^ Constantine A. Murenin (2007-04-17). "2. Hardware review". Generalised Interfacing with Microprocessor System Hardware Monitors. Proceedings of 2007 IEEE International Conference on Networking, Sensing and Control, 15–17 April 2007. London, United Kingdom: IEEE. pp. 901–906. doi:10.1109/ICNSC.2007.372901. ISBN 978-1-4244-1076-7. IEEE ICNSC 2007, pp. 901—906. ^ "aibs – ASUSTeK AI Booster ACPI ATK0110 voltage, temperature and fan sensor". OpenBSD, DragonFly BSD, NetBSD and FreeBSD. 2010. ^ "acpi_thermal(4)". www.freebsd.org. Retrieved 2021-02-24. ^ Clarke, Glen E.; Tetz, Edward (2007-01-30). CompTIA A+ Certification All-In-One Desk Reference for Dummies. John Wiley & Sons. ISBN 978-0-471-74811-3. ^ Micheloni, Rino; Crippa, Luca; Marelli, Alessia (2010-07-27). Inside NAND Flash Memories. Springer. ISBN 978-90-481-9431-5. ^ Mueller, Scott (2004). Upgrading and Repairing PCS. Que. ISBN 978-0-7897-2974-3. ^ Clarke, Glen E.; Tetz, Edward (2007-01-30). CompTIA A+ Certification All-In-One Desk Reference for Dummies. John Wiley & Sons. ISBN 978-0-471-74811-3. ^ "Decoding RAM & ROM Archived 2012-04-06 at the Wayback Machine." Smart Computing. June 1997. Volume 8, Issue 6. ^ "Upgrading Your Flash BIOS For Plug And Play Archived 2012-04-06 at the Wayback Machine." Smart Computing. March 1996. Volume 7, Issue 3. ^ "Time To Check BIOS Archived 2011-07-16 at the Wayback Machine." Smart Computing. April 1999. Volume 7, Issue 4. ^ "BIOS Flash Location". Archived from the original on 2021-08-18. Retrieved 2025-03-30. ^ "SplashTop's Instant-On Linux Desktop | Geek.com". Archived from the original on 2008-09-07. ^ Posted by Alex Watson, possibly repost from original content on custompc.com [unclear]. "The life and times of the modern motherboard". 2007-11-27. Archived from the original on 2012-07-24. Retrieved 2013-02-02. ^ David Hilber Jr. (August 2009). "Considerations for Designing an Embedded Intel Architecture System with System Memory Down" (PDF). Intel. Archived (PDF) from the original on 2012-10-18. Retrieved 2013-02-02. ^ "Types of BIOS". rompacks.com. Retrieved 2021-09-20. ^ Stiller, Andreas (2001). "Prozessor-Patches". c't (in German) (5). Heise: 240. Archived from the original on 2015-11-22. Retrieved 2015-11-21. ^ "Award BIOS logo". 2015-06-15. Archived from the original on 2015-12-21. Retrieved 2015-12-06. ^ Phoenix Eagerly Waiting to Clone Next-Generation IBM BIOS. Archived 2014-01-22 at the Wayback Machine, InfoWorld, 9 March 1987. ^ IEEE Standard for Boot (Initialization Configuration) Firmware: Core Requirements and Practices. 1994-10-28. pp. 1–262. doi:10.1109/IEEESTD.1994.89427. ISBN 978-0-7381-1194-0. IEEE STD 1275-1994. ^ "IEEE Standards Association". IEEE Standards Association. Retrieved 2023-04-26. ^ "Definition of hardware abstraction layer". PCMAG. Retrieved 2022-07-11. ^ New BIOS Virus Withstands HDD Wipes, 27 March 2009. Marcus Yam. Tom's Hardware US ^ "Black Hat 2006 Multimedia - Presentation, Audio and Video Archives". www.blackhat.com. Retrieved 2019-04-21. ^ Sacco, Anibal; Alfredo Ortéga (2009-03-23). "Persistent BIOS Infection". Exploiting Stuff. Archived from the original on 2009-08-04. Retrieved 2010-02-06. ^ Fisher, Dennis. "Researchers unveil persistent BIOS attack methods". Threat Post. Archived from the original on 2010-01-30. Retrieved 2010-02-06. ^ Giuliani, Marco (2011-09-13). "Mebromi: the first BIOS rootkit in the wild". blog. Archived from the original on 2011-09-23. Retrieved 2011-09-19. ^ "360发布"BMW病毒"技术分析报告". blog. Archived from the original on 2011-09-25. Retrieved 2011-09-19. ^ Yuan, Liang. "Trojan.Mebromi". Threat Response. Archived from the original on 2011-09-23. Retrieved 2011-09-19. ^ a b "How did 60 Minutes get cameras into a spy agency?". CBS News. 2013-12-15. Archived from the original on 2014-04-22. Retrieved 2014-04-15. ^ Spencer Ackerman in Washington (2013-12-16). "NSA goes on 60 Minutes: the definitive facts behind CBS's flawed report". theguardian.com. Archived from the original on 2014-01-25. Retrieved 2014-01-27. ^ Friedersdorf, Conor (2013-12-16). "A Question for 60 Minutes: Why Would China Want to Destroy the Global Economy?". The Atlantic. Retrieved 2019-03-26. ^ Poulsen, Kevin (2013-12-16). "60 Minutes Puff Piece Claims NSA Saved U.S. From Cyberterrorism". Wired. ISSN 1059-1028. Retrieved 2019-03-26 – via www.wired.com. ^ Sharwood, Simon (2013-12-16). "NSA alleges 'BIOS plot to destroy PCs'". The Register. Retrieved 2019-03-26. ^ "Windows and GPT FAQ". microsoft.com. Microsoft. Archived from the original on 2011-02-19. Retrieved 2014-12-06. ^ "Extensible Firmware Interface (EFI) and Unified EFI (UEFI)". Intel. Archived from the original on 2010-01-05. Retrieved 2014-12-06. ^ "Windows 11 Specs and System Requirements | Microsoft". Microsoft. Retrieved 2021-10-14. ^ "Next-gen boot spec could forever lock Linux off Windows 8 PCS". ^ "Windows 8 secure boot could complicate Linux installs". 2011-09-21. ^ Tung, Liam (2017-11-20). "Intel: We're ending all legacy BIOS support by 2020". ZDNET. IBM Personal Computer Technical Reference (Revised ed.). IBM Corporation. March 1983. IBM Personal Computer AT Technical Reference. IBM Personal Computer Hardware Reference Library. Vol. 0, 1, 2 (Revised ed.). IBM Corporation. March 1986 [1984-03]. 1502494, 6139362, 6183310, 6183312, 6183355, 6280070, 6280099. Phoenix Technologies, Ltd. (1989) [1987]. System BIOS for IBM PC/XT/AT Computers and Compatibles — The Complete Guide to ROM-Based System Software. Phoenix Technical Reference Series (1st ed.). Addison Wesley Publishing Company, Inc. ISBN 0-201-51806-6. Phoenix Technologies, Ltd. (1989) [1987]. CBIOS for IBM PS/2 Computers and Compatibles — The Complete Guide to ROM-Based System Software for DOS. Phoenix Technical Reference Series (1st ed.). Addison Wesley Publishing Company, Inc. ISBN 0-201-51804-X. Phoenix Technologies, Ltd. (1989) [1987]. ABIOS for IBM PS/2 Computers and Compatibles — The Complete Guide to ROM-Based System Software for OS/2. Phoenix Technical Reference Series (1st ed.). Addison Wesley Publishing Company, Inc. ISBN 0-201-51805-8. Phoenix Technologies, Ltd. (June 1991). System BIOS for IBM PCs, Compatibles, and EISA Computers — The Complete Guide to ROM-Based System Software. Phoenix Technical Reference Series (2nd ed.). Amsterdam: Addison Wesley Publishing Company, Inc. ISBN 0-201-57760-7. BIOS Disassembly Ninjutsu Uncovered, 1st edition, a freely available book in PDF format [1] More Power To Firmware, free bonus chapter to the Mac OS X Internals: A Systems Approach book Wikimedia Commons has media related to the BIOS. Look up BIOS in Wiktionary, the free dictionary. "BIOS Boot Specification 1.01" (PDF). Phoenix.com. 1996-01-11. Archived from the original (PDF) on 2011-07-15. "How BIOS Works". How Stuff Works. 2000-09-06. "Implementing a Plug and Play BIOS Using Intel's Boot Block Flash Memory" (PDF). Intel. February 1995. Archived from the original (PDF) on 2007-11-28. Retrieved 2007-11-28. "List of BIOS options". techarp.com. Archived from the original on 2014-01-27. "Persistent BIOS Infection". Phrack. No. 66. 2009-06-01. Archived from the original on 2011-04-30. Retrieved 2011-04-30. "Preventing BIOS Failures Using Intel Boot Block Flash Memory" (PDF). Intel. December 1998. Archived from the original (PDF) on 2007-03-29. Retrieved 2007-03-29. Retrieved from " 2Operating system from IBM "CP/DOS" redirects here. For the similarly named Digital Research operating system, see CP/M. Operating system OS/2OS/2 Warp 4 desktop. This version was released on 25 September 1996.[1]DeveloperIBMMicrosoft (1.0–1.3)Written inC, C++ and assembly languageWorking stateHistorical, now developed as ArcaOSSource modelClosed sourceInitial releaseDecember 1987; 37 years ago (1987-12)Latest release4.52 / December 2001; 23 years ago (2001-12)Marketing targetProfessionals, serversAvailable inChinese, English, French, German, Italian, Japanese, Korean, Spanish, Slovenian, Portuguese, RussianPlatforms32-bit x86, PowerPCKernel typeHybrid kernelInfluenced byMS-DOS, IBM PC DOSDefaultuser interfaceWorkplace ShellLicenseProprietarySucceeded byeComStationArcaOSOfficial websiteOS/2 Warp (Archived) OS/2 is a proprietary computer operating system for x86 and PowerPC based personal computers. It was created and initially developed jointly by IBM and Microsoft, under the leadership of IBM software designer Ed Iacobucci,[2] intended as a replacement for DOS. The first version was released in 1987. A feud between the two companies beginning in 1990 led to Microsoft's leaving development solely to IBM, which continued development on its own. OS/2 Warp 4 in 1996 was the last major upgrade, after which IBM slowly halted the product as it failed to compete against Microsoft's Windows; updated versions of OS/2 were released by IBM until 2001. The name stands for "Operating System/2", because it was introduced as part of the same generation change release as IBM's "Personal System/2 (PS/2)" line of second-generation PCs. OS/2 was intended as a protected-mode successor of PC DOS targeting the Intel 80286 processor. Notably, basic system calls were modelled after MS-DOS calls; their names even started with "Dos" and it was possible to create "Family Mode" applications – text mode applications that could work on both systems.[3] Because of this heritage, OS/2 shares similarities with Unix, Xenix, and Windows NT. OS/2 sales were largely concentrated in networked computing used by corporate professionals. OS/2 2.0 was released in 1992 as the first 32-bit version as well as the first to be entirely developed by IBM, after Microsoft severed ties over a dispute over how to position OS/2 relative to Microsoft's new Windows 3.1 operating environment.[4][5] With OS/2 Warp 3 in 1994, IBM attempted to also target home consumers through a multi-million dollar advertising campaign.[6] However it continued to struggle in the marketplace, partly due to strategic business measures imposed by Microsoft in the industry that have been considered anti-competitive.[7][8] Following the failure of IBM's Workplace OS project, OS/2 Warp 4 became the final major release in 1996; IBM discontinued its support for OS/2 on December 31, 2006.[9] Since then, OS/2 has been developed, supported and sold by two different third-party vendors under license from IBM – first by Serenity Systems as eComStation from 2001 to 2011,[10] and later by Arca Noae LLC as ArcaOS since 2017.[11][12][13] This section needs additional citations for verification. Please help improve this article by adding citations to reliable sources in this section. Unsourced material may be challenged and removed. (April 2012) (Learn how and when to remove this message) Logo of OS/2 1.xOS/2 1.0 featured a text-mode interface similar to MS-DOS. The development of OS/2 began when IBM and Microsoft signed the "Joint Development Agreement" in August 1985.[14][15] It was code-named "CP/DOS" and it took two years for the first product to be delivered. OS/2's release was long delayed. It was widely believed that all IBM programmers used assembly language, and a rumor said that the delay was because they had to learn C.[16] OS/2 1.0 was announced in April 1987 and released in December. The original release only ran in text mode, and a GUI was introduced with OS/2 1.1 about a year later. OS/2 features an API for controlling the video display (VIO) and handling keyboard and mouse events so that programmers writing for protected mode need not call the BIOS or access hardware directly. Other development tools included a subset of the video and keyboard APIs as linkable libraries so that family mode programs are able to run under MS-DOS,[citation needed] and, in the OS/2 Extended Edition v1.0, a database engine called Database Manager or DBM (this was related to DB2, and should not be confused with the DBM family of database engines for Unix and Unix-like operating systems).[17] A task-switcher named Program Selector was available through the Ctrl-Esc hotkey combination, allowing the user to select among multitasked text-mode sessions (or screen groups; each can run multiple programs).[18] Communications and database-oriented extensions were delivered in 1988, as part of OS/2 1.0 Extended Edition: SNA, X.25/APPC/LU 6.2, LAN Manager, Query Manager, SQL. OS/2 1.1 was the first version to feature the Presentation Manager GUI. The promised user interface, Presentation Manager, was introduced with OS/2 1.1 in October 1988.[19] It had a similar user interface to Windows 2.1, which was released in May of that year. (The interface was replaced in versions 1.2 and 1.3 by a look closer in appearance to Windows 3.0.) The Extended Edition of 1.1, sold only through IBM sales channels, introduced distributed database support to IBM database systems and SNA communications support to IBM mainframe networks. In 1989, Version 1.2 introduced Installable Filesystems and, notably, the HPFS filesystem. HPFS provided a number of improvements over the older FAT file system, including long filenames and a form of alternate data streams called Extended Attributes.[20] In addition, extended attributes were also added to the FAT file system.[21] Installation Disk A of Microsoft OS/2 1.3 (3½-inch floppy disk) The Extended Edition of 1.2 introduced TCP/IP and Ethernet support. OS/2- and Windows-related books of the late 1980s from both Microsoft's Gordon Letwin and his IBM counterpart Ed Iacobucci acknowledged the existence of both systems and promoted OS/2 as the system of the future.[22][16] Logo of Microsoft's OS/2 until the breakup The collaboration between IBM and Microsoft unravelled in 1990, between the releases of Windows 3.0 and OS/2 1.3. During this time, Windows 3.0 became a tremendous success, selling millions of copies in its first year.[23] Much of its success was because Windows 3.0 (along with MS-DOS) was bundled with most new computers.[24] OS/2, on the other hand, was available only as an additional stand-alone software package. In addition, OS/2 lacked device drivers for many common devices such as printers, particularly non-IBM hardware.[25] Windows, on the other hand, supported a much larger variety of hardware. The increasing popularity of Windows prompted Microsoft to shift its development focus from cooperating on OS/2 with IBM to building its own business based on Windows.[26] Several technical and practical reasons contributed to this breakup. The two companies had significant differences in culture and vision. Microsoft favored the open hardware system approach that contributed to its success on the PC. IBM sought to use OS/2 to drive sales of its own hardware, and urged Microsoft to drop features, such as fonts, that IBM's hardware did not support. Microsoft programmers also became frustrated with IBM's bureaucracy and its use of lines of code to measure programmer productivity.[27] IBM developers complained about the terseness and lack of comments in Microsoft's code, while Microsoft developers complained that IBM's code was bloated.[28] The two products have significant differences in API. OS/2 was announced when Windows 2.0 was near completion, and the Windows API already defined. However, IBM requested that this API be significantly changed for OS/2.[29] Therefore, issues surrounding application compatibility appeared immediately. OS/2 designers hoped for source code conversion tools, allowing complete migration of Windows application source code to OS/2 at some point. However, OS/2 1.x did not gain enough momentum to allow vendors to avoid developing for both OS/2 and Windows in parallel. OS/2 1.3 was the final 16-bit only version of OS/2, and the last to be sold by Microsoft. OS/2 1.x targets the Intel 80286 processor and DOS fundamentally does not. IBM insisted on supporting the 80286 processor, with its 16-bit segmented memory mode, because of commitments made to customers who had purchased many 80286-based PS/2s as a result of IBM's promises surrounding OS/2.[30] Until release 2.0 in April 1992, OS/2 ran in 16-bit protected mode and therefore could not benefit from the Intel 80386's much simpler 32-bit flat memory model and virtual 8086 mode features. This was especially painful in providing support for DOS applications. While, in 1988, Windows/386 2.1 could run several cooperatively multitasked DOS applications, including expanded memory (EMS) emulation, OS/2 1.3, released in 1991, was still limited to one 640 kB "DOS box". Given these issues, Microsoft started to work in parallel on a version of Windows which was more future-oriented and more portable. The hiring of Dave Cutler, former VAX/VMS architect, in 1988 created an immediate competition with the OS/2 team, as Cutler did not think much of the OS/2 technology and wanted to build on his work on the MICA project at Digital rather than creating a "DOS plus". His NT OS/2 was a completely new architecture.[31] IBM grew concerned about the delays in development of OS/2 2.0. Initially, the companies agreed that IBM would take over maintenance of OS/2 1.0 and development of OS/2 2.0, while Microsoft would continue development of OS/2 3.0. In the end, Microsoft decided to recast NT OS/2 3.0 as Windows NT, leaving all future OS/2 development to IBM. From a business perspective, it was logical to concentrate on a consumer line of operating systems based on DOS and Windows, and to prepare a new high-end system in such a way as to keep good compatibility with existing Windows applications. While it waited for this new high-end system to develop, Microsoft would still receive licensing money from Xenix and OS/2 sales. Windows NT's OS/2 heritage can be seen in its initial support for the HPFS filesystem, text mode OS/2 1.x applications, and OS/2 LAN Manager network support. Some early NT materials even included OS/2 copyright notices embedded in the software.[citation needed] One example of NT OS/2 1.x support is in the WIN2K resource kit. Windows NT could also support OS/2 1.x Presentation Manager and AVIO applications with the addition of the Windows NT Add-On Subsystem for Presentation Manager.[32] OS/2 2.0 was the first 32-bit release of OS/2, and the first to feature the Workplace Shell. OS/2 2.0 was released in April 1992. At the time, the suggested retail price was US$195, while Windows retailed for $150.[33] OS/2 2.0 provided a 32-bit API for native programs, though the OS itself still contained some 16-bit code and drivers. It also included a new OOUI (object-oriented user interface) called the Workplace Shell. This was a fully object-oriented interface that was a significant departure from the previous GUI. Rather than merely providing an environment for program windows (such as the Program Manager), the Workplace Shell provided an environment in which the user could manage programs, files and devices by manipulating objects on the screen. With the Workplace Shell, everything in the system is an "object" to be manipulated. OS/2 2.0 was touted by IBM as "a better DOS than DOS and a better Windows than Windows".[34] It managed this by including the fully-licensed MS-DOS 5.0, which had been patched and improved upon. For the first time, OS/2 was able to run more than one DOS application at a time. This was so effective that it allowed OS/2 to run a modified copy of Windows 3.0, itself a DOS extender, including Windows 3.0 applications. Because of the limitations of the Intel 80286 processor, OS/2 1.x could run only one DOS program at a time, and did this in a way that allowed the DOS program to have total control over the computer. A problem in DOS mode could crash the entire computer. In contrast, OS/2 2.0 could leverage the virtual 8086 mode of the Intel 80386 processor to create a much safer virtual machine in which to run DOS programs. This included an extensive set of configuration options to optimize the performance and capabilities given to each DOS program. Any real-mode operating system (such as 8086 Xenix) could also be made to run using OS/2's virtual machine capabilities, subject to certain direct hardware access limitations. The OS/2 2.0 upgrade box Like most 32-bit environments, OS/2 could not run protected-mode DOS programs using the older VCPI interface, unlike the Standard mode of Windows 3.1; it only supported programs written according to DPMI. (Microsoft discouraged the use of VCPI under Windows 3.1, however, due to performance degradation.)[35] Unlike Windows NT, OS/2 always allowed DOS programs the possibility of masking real hardware interrupts, so any DOS program could deadlock the machine in this way. OS/2 could, however, use a hardware watchdog on selected machines (notably IBM machines) to break out of such a deadlock. Later, release 3.0 leveraged the enhancements of newer Intel 80486 and Intel Pentium processors—the Virtual Interrupt Flag (VIF), which was part of the Virtual Mode Extensions (VME)—to solve this problem. To accommodate those who wanted to have multiple operating systems on their machine, Boot Manager was introduced that allowed for the creation of separate partitions on the boot drive which could be used to install different versions of DOS, Windows and OS/2 and give the user a choice of which partition to boot from.[36] Further information: VME (CONFIG.SYS directive) OS/2 2.1 was released in 1993. This version of OS/2 achieved compatibility with Windows 3.0 (and later Windows 3.1) by adapting Windows user-mode code components to run inside a virtual DOS machine (VDM). Originally, a nearly complete version of Windows code was included with OS/2 itself: Windows 3.0 in OS/2 2.0, and Windows 3.1 in OS/2 2.1. Later, IBM developed versions of OS/2 that would use whatever Windows version the user had installed previously, patching it on the fly, and sparing the cost of an additional Windows license.[37] It could either run full-screen, using its own set of video drivers, or "seamlessly," where Windows programs would appear directly on the OS/2 desktop. The process containing Windows was given fairly extensive access to hardware, especially video, and the result was that switching between a full-screen WinOS/2 session and the Workplace Shell could occasionally cause issues.[38] Because OS/2 only runs the user-mode system components of Windows, it is incompatible with Windows device drivers (VxDs) and applications that require them. Multiple Windows applications run by default in a single Windows session – multitasking cooperatively and without memory protection – just as they would under native Windows 3.x. However, to achieve true isolation between Windows 3.x programs, OS/2 can also run multiple copies of Windows in parallel, with each copy residing in a separate VDM. The user can then optionally place each program either in its own Windows session – with preemptive multitasking and full memory protection between sessions, though not within them – or allow some applications to run together cooperatively in a shared Windows session while isolating other applications in one or more separate Windows sessions. At the cost of additional hardware resources, this approach can protect each program in any given Windows session (and each instance of Windows itself) from every other program running in any separate Windows session (though not from other programs running in the same Windows session).[39] Whether Windows applications are running in full-screen or windowed mode, and in one Windows session or several, it is possible to use DDE between OS/2 and Windows applications, and OLE between Windows applications only.[40] IBM's OS/2 for Windows product (codename Ferengi), also known as "OS/2, Special Edition", was interpreted as a deliberate strategy "of cashing in on the pervasive success of the Microsoft platform" but risked confusing consumers with the notion that the product was a mere accessory or utility running on Windows such as Norton Desktop for Windows when, in fact, it was "a complete, modern, multi-tasking, pre-emptive operating system", itself hosting Windows instead of running on it. Available on CD-ROM or 18 floppy disks, the product documentation reportedly suggested Windows as a prerequisite for installing the product, also being confined to its original FAT partition, whereas the product apparently supported the later installation of Windows running from an HPFS partition, particularly beneficial for users of larger hard drives. Windows compatibility, relying on patching specific memory locations, was reportedly broken by the release of Windows 3.11, prompting accusations of arbitrary changes to Windows in order to perpetrate "a deliberate act of Microsoft sabotage" against IBM's product.[39] Wordmark of OS/2 Warp 3.0OS/2 Warp Connect 3.0, showing the Windows 3.1 Program Manager, QBASIC in a DOS window, and the LaunchPad (bottom center) Released in 1994, OS/2 version 3.0 was labelled as OS/2 Warp to highlight the new performance benefits, and generally to freshen the product image. "Warp" had originally been the internal IBM name for the release: IBM claimed that it had used Star Trek terms as internal names for prior OS/2 releases, and that this one seemed appropriate for external use as well. At the launch of OS/2 Warp in 1994, Patrick Stewart was to be the Master of Ceremonies; however Kate Mulgrew[41] of the then-upcoming series Star Trek: Voyager substituted for him at the last minute.[42][43] OS/2 Warp offers a host of benefits over OS/2 2.1, notably broader hardware support, greater multimedia capabilities, Internet-compatible networking, and it includes a basic office application suite known as IBM Works. It was released in two versions: the less expensive "Red Spine" and the more expensive "Blue Spine" (named for the color of their boxes). "Red Spine" was designed to support Microsoft Windows applications by utilizing any existing installation of Windows on the computer's hard drive. "Blue Spine" includes Windows support in its own installation, and so can support Windows applications without a Windows installation. As most computers were sold with Microsoft Windows pre-installed and the price was less, "Red Spine" was the more popular product.[44] OS/2 Warp Connect—which has full LAN client support built-in—followed in mid-1995. Warp Connect was nicknamed "Grape".[19] In OS/2 2.0, most performance-sensitive subsystems, including the graphics (Gre) and multimedia (MMPM/2) systems, were updated to 32-bit code in a fixpack, and included as part of OS/2 2.1. Warp 3 brought about a fully 32-bit windowing system, while Warp 4 introduced the object-oriented 32-bit GRADD display driver model. Main article: Workplace OS In 1991, IBM started development on an intended replacement for OS/2 called Workplace OS. This was an entirely new product, brand new code, that borrowed only a few sections of code from both the existing OS/2 and AIX products. It used an entirely new microkernel code base, intended (eventually) to host several of IBM's operating systems (including OS/2) as microkernel "personalities". It also included major new architectural features including a system registry, JFS, support for UNIX graphics libraries, and a new driver model.[45] Workplace OS was developed solely for POWER platforms, and IBM intended to market a full line of PowerPCs in an effort to take over the market from Intel. A mission was formed to create prototypes of these machines and they were disclosed to several corporate customers, all of whom raised issues with the idea of dropping Intel. Advanced plans for the new code base would eventually include replacement of the OS/400 operating system by Workplace OS, as well as a microkernel product that would have been used in industries such as telecommunications and set-top television receivers. A partially functional pre-alpha version of Workplace OS was demonstrated at Comdex, where a bemused Bill Gates stopped by the booth. The second and last time it would be shown in public was at an OS/2 user group in Phoenix, Arizona; the pre-alpha code refused to boot. It was released in 1995. But with $990 million being spent per year on development of this as well as Workplace OS, and no possible profit or widespread adoption, the end of the entire Workplace OS and OS/2 product line was near. Firefox 3.5.4 for OS/2 Warp 4 Wordmark of OS/2 Warp 4OS/2 Warp 4 desktop after installation In 1996, Warp 4 added Java and speech recognition software.[46] IBM also released server editions of Warp 3 and Warp 4 which bundled IBM's LAN Server product directly into the operating system installation. A personal version of Lotus Notes was also included, with a number of template databases for contact management, brainstorming, and so forth. The UK-distributed free demo CD-ROM of OS/2 Warp essentially contained the entire OS and was easily, even accidentally, cracked[clarification needed], meaning that even people who liked it did not have to buy it. This was seen as a backdoor tactic to increase the number of OS/2 users, in the belief that this would increase sales and demand for third-party applications, and thus strengthen OS/2's desktop numbers.[citation needed] This suggestion was bolstered by the fact that this demo version had replaced another which was not so easily cracked, but which had been released with trial versions of various applications.[citation needed] In 2000, the July edition of Australian Personal Computer magazine bundled software CD-ROMs, included a full version of Warp 4 that required no activation and was essentially a free release. Special versions of OS/2 2.11 and Warp 4 also included symmetric multiprocessing (SMP) support. OS/2 sales were largely concentrated in networked computing used by corporate professionals; however, by the early 1990s, it was overtaken by Microsoft Windows NT. While OS/2 was arguably technically superior to Microsoft Windows 95, OS/2 failed to develop much penetration in the consumer and stand-alone desktop PC segments; there were reports that it could not be installed properly on IBM's own Aptiva series of home PCs.[47] Microsoft made an offer in 1994 where IBM would receive the same terms as Compaq (the largest PC manufacturer at the time) for a license of Windows 95, if IBM ended development of OS/2 completely. IBM refused and instead went with an "IBM First" strategy of promoting OS/2 Warp and disparaging Windows, as IBM aimed to drive sales of its own software as well as hardware. By 1995, Windows 95 negotiations between IBM and Microsoft, which were already difficult, stalled when IBM purchased Lotus SmartSuite, which would have directly competed with Microsoft Office. As a result of the dispute, IBM signed the license agreement 15 minutes before Microsoft's Windows 95 launch event, which was later than their competitors and this badly hurt sales of IBM PCs. IBM officials later conceded that OS/2 would not have been a viable operating system to keep them in the PC business.[48][49] This section does not cite any sources. Please help improve this section by adding citations to reliable sources. Unsourced material may be challenged and removed. (March 2013) (Learn how and when to remove this message) A project was launched internally by IBM to evaluate the looming competitive situation with Microsoft Windows 95. Primary concerns included the major code quality issues in the existing OS/2 product (resulting in over 20 service packs, each requiring more diskettes than the original installation), and the ineffective and heavily matrixed development organization in Boca Raton (where the consultants reported that "basically, everybody reports to everybody") and Austin. That study, tightly classified as "Registered Confidential" and printed only in numbered copies, identified untenable weaknesses and failures across the board in the Personal Systems Division as well as across IBM as a whole. This resulted in a decision being made at a level above the Division to cut over 95% of the overall budget for the entire product line, end all new development (including Workplace OS), eliminate the Boca Raton development lab, end all sales and marketing efforts of the product, and lay off over 1,300 development individuals (as well as sales and support personnel). $990 million had been spent in the last full year. Warp 4 became the last distributed version of OS/2. Although a small and dedicated community remains faithful to OS/2,[50] OS/2 failed to catch on in the mass market and is little used outside certain niches where IBM traditionally had a stronghold. For example, many bank installations, especially automated teller machines, run OS/2 with a customized user interface; French SNCF national railways used OS/2 1.x in thousands of ticket selling machines.[citation needed] Telecom companies such as Nortel used OS/2 in some voicemail systems. Also, OS/2 was used for the host PC used to control the Satellite Operations Support System equipment installed at NPR member stations from 1994 to 2007, and used to receive the network's programming via satellite.[citation needed] Although IBM began indicating shortly after the release of Warp 4 that OS/2 would eventually be withdrawn, the company did not end support until December 31, 2006,[51] with sales of OS/2 stopping on December 23, 2005. The latest IBM OS/2 Warp version is 4.52, which was released for both desktop and server systems in December 2001. IBM is still delivering defect support for a fee.[51][52] IBM urges customers to migrate their often highly complex applications to e-business technologies such as Java in a platform-neutral manner. Once application migration is completed, IBM recommends migration to a different operating system, suggesting Linux as an alternative.[53][54][55] Main articles: eComStation and ArcaOS ArcaOS is the most recent OS/2-based operating system developed outside of IBM. After IBM discontinued development of OS/2, various third parties approached IBM to take over future development of the operating system. The OS/2 software vendor Stardock made such a proposal to IBM in 1999, but it was not followed through by the company.[56] Serenity Systems succeeded in negotiating an agreement with IBM, and began reselling OS/2 as eComStation in 2001.[57] eComStation is now sold by XEU.com, the most recent version (2.1) was released in 2011.[58] In 2015, Arca Noae, LLC announced that they had secured an agreement with IBM to resell OS/2.[11] They released the first version of their OS/2-based operating system in 2017 as ArcaOS.[13] As of 2023, there have been multiple releases of ArcaOS, and it remains under active development.[59] Many people hoped that IBM would release OS/2 or a significant part of it as open source. Petitions were held in 2005 and 2007, but IBM refused them, citing legal and technical reasons.[60] It is unlikely that the entire OS will be open at any point in the future because it contains third-party code to which IBM does not have copyright, and much of this code is from Microsoft. IBM also once engaged in a technology transfer with Commodore, licensing Amiga technology for OS/2 2.0 and above, in exchange for the REXX scripting language.[61][unreliable source?] This means that OS/2 may have some code that was not written by IBM, which can therefore prevent the OS from being re-announced as open-sourced in the future.[62][failed verification][63] On the other hand, IBM donated Object REXX for Windows and OS/2 to the Open Object REXX project maintained by the REXX Language Association on SourceForge.[64] There was a petition, arranged by OS2World, to open parts of the OS. Open source operating systems such as Linux have already profited from OS/2 indirectly through IBM's release of the improved JFS file system, which was ported from the OS/2 code base. As IBM didn't release the source of the OS/2 JFS driver, developers ported the Linux driver back to eComStation and added the functionality to boot from a JFS partition. This new JFS driver has been integrated into eComStation v2.0, and later into ArcaOS 5.0. Release dates refer to the US English editions unless otherwise noted.[65][66] Date Version December 1987 OS/2 1.0 November 1988 OS/2 1.1 October 1989 OS/2 1.2 December 1990 OS/2 1.3 October 1991 OS/2 2.0 LA (Limited Availability) April 1992 OS/2 2.0 October 1992 OS/2 2.00.1 May 1993 OS/2 2.1 November 1993 OS/2 for Windows February 1994 OS/2 2.11 July 1994 OS/2 2.11 SMP October 1994 OS/2 Warp 3 May 1995 OS/2 Warp Connect December 1995 OS/2 Warp, PowerPC Edition February 1996 OS/2 Warp Server 4 September 1996 OS/2 Warp 4 September 1996 OS/2 Warp Server Advanced SMP November 1997 WorkSpace On-Demand 1.0 October 1998 WorkSpace On-Demand 2.0 April 1999 OS/2 Warp Server for e-Business (version 4.50) November 2000 OS/2 Convenience Pack 1 (version 4.51) November 2001 OS/2 Convenience Pack 2 (version 4.52) The graphic system has a layer named Presentation Manager that manages windows, fonts, and icons. This is similar in functionality to a non-networked version of X11 or the Windows GDI. On top of this lies the Workplace Shell (WPS) introduced in OS/2 2.0. WPS is an object-oriented shell allowing the user to perform traditional computing tasks such as accessing files, printers, launching legacy programs, and advanced object oriented tasks using built-in and third-party application objects that extended the shell in an integrated fashion not available on any other mainstream operating system. WPS follows IBM's Common User Access user interface standards. WPS represents objects such as disks, folders, files, program objects, and printers using the System Object Model (SOM), which allows code to be shared among applications, possibly written in different programming languages. A distributed version called DSOM allowed objects on different computers to communicate. DSOM is based on CORBA. The object oriented aspect of SOM is similar to, and a direct competitor to, Microsoft's Component Object Model, though it is implemented in a radically different manner; for instance, one of the most notable differences between SOM and COM is SOM's support for inheritance (one of the most fundamental concepts of OO programming)—COM does not have such support. SOM and DSOM are no longer being developed. The multimedia capabilities of OS/2 are accessible through Media Control Interface commands. The last update (bundled with the IBM version of Netscape Navigator plugins) added support for MPEG files. Support for newer formats such as PNG, progressive JPEG, DivX, Ogg, and MP3 comes from third parties. Sometimes it is integrated with the multimedia system, but in other offers it comes as standalone applications. OS/2 Window (cmd.exe) on Microsoft OS/2 Version 1.3 The following list of commands is supported by cmd.exe on OS/2.[67][68] ansi append assign attrib backup boot break cache call cd chcp chdir chkdsk cls cmd codepage command comp copy createdd date ddinstal debug del detach dir diskcomp diskcopy doskey dpath eautil echo endlocal erase exit extproc fdisk fdiskpm find for format fsaccess goto graftabl help if join keyb keys label makeini md mem mkdir mode more move patch path pause picview pmrexx print prompt pstat rd recover rem ren rename replace restore rmdir set setboot setcom40 setlocal share shift sort spool start subst syslevel syslog time trace tracebuf tracefmt tree type undelete unpack ver verify view vmdisk vol xcopy This section needs expansion. You can help by adding to it. (April 2019) The TCP/IP stack is based on the open source BSD stack as visible with SCCS what compatible tools. IBM included tools such as ftp and telnet and even servers for both commands. IBM sold several networking extensions including NFS support and an X11 server. Architecture of OS/2 Warp under x86 Hardware vendors were reluctant to support device drivers for alternative operating systems including OS/2, leaving users with few choices from a select few vendors. To relieve this issue for video cards, IBM licensed a reduced version of the Scitech display drivers, allowing users to choose from a wide selection of cards supported through Scitech's modular driver design.[69] Document detailing OS/2's architecture. OS/2 has historically been more difficult to run in a virtual machine than most other legacy x86 operating systems because of its extensive reliance on the full set of features of the x86 CPU; in particular, OS/2's use of ring 2 prevented it from running in early versions of VMware.[70] Newer versions of VMware provide official support for OS/2, specifically for eComStation.[71] VirtualPC from Microsoft (originally Connectix) has been able to run OS/2 without hardware virtualization support for many years. It also provided "additions" code which greatly improves host–guest OS interactions in OS/2. The additions are not provided with the current version of VirtualPC, but the version last included with a release may still be used with current releases. At one point, OS/2 was a supported host for VirtualPC in addition to a guest. Note that OS/2 runs only as a guest on those versions of VirtualPC that use virtualization (x86 based hosts) and not those doing full emulation (VirtualPC for Mac). VirtualBox from Oracle Corporation (originally InnoTek, later Sun) supports OS/2 1.x, Warp 3 through 4.5, and eComStation as well as "Other OS/2" as guests. However, attempting to run OS/2 and eComStation can still be difficult, if not impossible, because of the strict requirements of VT-x/AMD-V hardware-enabled virtualization and only ACP2/MCP2 is reported to work in a reliable manner.[72] ArcaOS supports being run as a virtual machine guest inside VirtualBox, VMware ESXi and VMWare Workstation.[73] It ships with VirtualBox Guest Additions, and driver improvements to improve performance as a guest operating system.[74] The difficulties in efficiently running OS/2 have, at least once, created an opportunity for a new virtualization company. A large bank in Moscow needed a way to use OS/2 on newer hardware that OS/2 did not support. As virtualization software is an easy way around this, the company desired to run OS/2 under a hypervisor. Once it was determined that VMware was not a possibility, it hired a group of Russian software developers to write a host-based hypervisor that would officially support OS/2. Thus, the Parallels, Inc. company and their Parallels Workstation product was born.[75] OS/2 has few native computer viruses;[76] while it is not invulnerable by design, its reduced market share appears to have discouraged virus writers. There are, however, OS/2-based antivirus programs, dealing with DOS viruses and Windows viruses that could pass through an OS/2 server.[77] Some problems were classic subjects of comparison with other operating systems: Synchronous input queue (SIQ): if a GUI application was not servicing its window messages, the entire GUI system could get stuck and a reboot was required. This problem was considerably reduced with later Warp 3 fixpacks and refined by Warp 4, by taking control over the application after it had not responded for several seconds.[78][79]:565 No unified object handles (OS/2 v2.11 and earlier): The availability of threads probably led system designers to overlook mechanisms which allow a single thread to wait for different types of asynchronous events at the same time, for example the keyboard and the mouse in a "console" program. Even though select was added later, it only worked on network sockets. In case of a console program, dedicating a separate thread for waiting on each source of events made it difficult to properly release all the input devices before starting other programs in the same "session". As a result, console programs usually polled the keyboard and the mouse alternately, which resulted in wasted CPU and a characteristic "jerky" reactivity to user input. In OS/2 3.0 IBM introduced a new call for this specific problem.[80] This section needs additional citations for verification. Please help improve this article by adding citations to reliable sources in this section. Unsourced material may be challenged and removed. (June 2011) (Learn how and when to remove this message) OS/2 has been widely used by Iran Export Bank (Bank Saderat Iran) in their teller machines, ATMs and local servers (over 35,000 working stations). As of 2011, the bank moved to virtualize and renew their infrastructure by moving OS/2 to Virtual Machines running over Windows. OS/2 was widely used by Brazilian banks. Banco do Brasil had a peak 10,000 machines running OS/2 Warp in the 1990s. OS/2 was used in automated teller machines until 2006. The workstations and automated teller machines and attendant computers have been migrated to Linux.[81] An ATM in Australia revealing during a reboot that it is based on OS/2 Warp OS/2 has been used in the banking industry. Suncorp bank in Australia still ran its ATM network on OS/2 as late as 2002. ATMs at Perisher Blue used OS/2 as late as 2009, and even the turn of the decade.[82] OS/2 was widely adopted by accounting professionals and auditing companies. In mid-1990s native 32-bit accounting software were well developed and serving corporate markets. OS/2 ran the faulty baggage handling system at Denver International Airport. The OS was eventually scrapped, but the software written for the system led to massive delays in the opening of the new airport. The OS itself was not at fault, but the software written to run on the OS was. The baggage handling system was eventually removed. OS/2 was used by radio personality Howard Stern. He once had a 10-minute on-air rant about OS/2 versus Windows 95 and recommended OS/2. He also used OS/2 on his IBM 760CD laptop. OS/2 was used as part of the Satellite Operations Support System (SOSS) for NPR's Public Radio Satellite System. SOSS was a computer-controlled system using OS/2 that NPR member stations used to receive programming feeds via satellite. SOSS was introduced in 1994 using OS/2 3.0, and was retired in 2007, when NPR switched over to its successor, the ContentDepot. OS/2 was used to control the SkyTrain automated light rail system in Vancouver, Canada until the late 2000s when it was replaced by Windows XP. OS/2 was used in the London Underground Jubilee Line Extension Signals Control System (JLESCS) in London, England. This control system delivered by Alcatel was in use from 1999 to 2011 i.e. between abandonment before opening of the line's unimplemented original automatic train control system and the present SelTrac system. JLESCS did not provide automatic train operation only manual train supervision. Six OS/2 local site computers were distributed along the railway between Stratford and Westminster, the shunting tower at Stratford Market Depot, and several formed the central equipment located at Neasden Depot. It was once intended to cover the rest of the line between Green Park and Stanmore but this was never introduced. OS/2 has been used by The Co-operative Bank in the UK for its domestic call centre staff, using a bespoke program created to access customer accounts which cannot easily be migrated to Windows. OS/2 has been used by the Stop & Shop supermarket chain (and has been installed in new stores as recently as March 2010). OS/2 has been used on ticket machines for Tramlink in outer-London. OS/2 has been used in New York City's subway system for MetroCards.[83] Rather than interfacing with the user, it connects simple computers and the mainframes. When NYC MTA finishes its transition to contactless payment, OS/2 will be removed.[84] OS/2 was used in checkout systems at Safeway supermarkets.[83] OS/2 was used by Trenitalia, both for the desktops at Ticket Counters and for the Automatic Ticket Counters up to 2011. Incidentally, the Automatic Ticket Counters with OS/2 were more reliable than the current ones running a flavor of Windows.[citation needed] OS/2 was used as the main operating system for Abbey National General Insurance motor and home direct call centre products using the PMSC Series III insurance platform on DB2.2 from 1996 to 2001. BYTE in 1989 listed OS/2 as among the "Excellence" winners of the BYTE Awards, stating that it "is today where the Macintosh was in 1984: It's a development platform in search of developers". The magazine predicted that "When it's complete and bug-free, when it can really use the 80386, and when more desktops sport OS/2-capable PCs, OS/2 will—deservedly—supersede DOS. But even as it stands, OS/2 is a milestone product".[85] In March 1995 OS/2 won seven awards[86] InfoWorld Product of the Year.[87] Five Awards at CeBIT. PC Professional Magazine - Innovation of the Year award. CHIP Magazine named OS/2 Warp the Operating System of the Year. DOS International named OS/2 Warp the Operating System of the Year. 1+1 Magazine awarded it with the Software Marketing Quality award. Industrie Forum awarded it with its Design Excellence. SPA Best Business Software Award. IBM has used OS/2 in a wide variety of hardware products, effectively as a form of embedded operating system. Product Product type Usage of OS/2 IBM 2074 Console support controller Used to connect 3270 sessions to host via ESCON channels. Introduced in September 2000 as a replacement for local, non-SNA 3174 Control Units. All models were withdrawn in 2006 and replaced by the Open System Adapter Integrated Console Controller (OSA ICC).[88] IBM 3494 Tape library Used as the operating system for the Library Manager (LM) that controlled the tape accessor (robot)[89] IBM 3745 Communications controller Used as the operating system for the Service Processor (SP) and if installed, the Network Node Processor (NNP).[90] IBM 3890 Document processor The 3890/XP1 was announced November 12, 1988. It initially used OS/2 1.1 Extended Edition[91] on a PS/2 Model 80 to emulate the stacker control software that previously ran on a System/360. IBM later switched to OS/2 Warp.[92] IBM 473x ATM Used in a range of automatic teller machines manufactured by IBM. Was also used in later 478x ATMs manufactured with Diebold. IBM 9672 IBM mainframe Used as the operating system for the Hardware Management Console (HMC) and Support Element (SE).[93] Was also used in later mainframe models such as the IBM 2064. History of the graphical user interface Multiple Virtual DOS Machine (MVDM) – OS/2 virtual DOS machine and seamless Windows integration OpenDoc – Software standard System Object Model – Programming frameworkPages displaying short descriptions of redirect targets Team OS/2 Windows Libraries for OS/2 LAN Manager ^ Matthew Nawrocki (26 February 2013). "Preview: eComStation 2.2 Beta, the legacy of OS/2 lives on". TechRepublic.com. Archived from the original on 2 September 2022. Retrieved 22 September 2013. ^ Nuska, Andrew (June 21, 2013). "Ed Iacobucci, co-founder of Citrix, dies of cancer". ZDNet. Archived from the original on 25 February 2023. Retrieved 20 May 2017. ^ Necasek, Michal (2001-09-08). "OS/2 1.3: Ten Years Ago". The History of OS/2. Archived from the original on 2007-10-12. ^ McCracken, Harry (2012-04-02). "25 Years of IBM's OS/2: The Strange Days and Surprising Afterlife of a Legendary Operating System". Time. ISSN 0040-781X. Archived from the original on 2022-11-28. Retrieved 2024-07-26. ^ Markoff, John (1992-06-28). "I.B.M. and Microsoft Settle Operating-System Feud". The New York Times. ISSN 0362-4331. Retrieved 2024-07-26. ^ Lewis, Peter H. (8 August 1995). "PERSONAL COMPUTERS; OS/2 No Longer at Home at Home". The New York Times. ^ Lea, Graham (15 June 1999). "IBM chief: Microsoft killed OS/2". BBC. ^ "Microsoft Hampered OS/2, I.B.M. Official Tells Court (Published 1998)". The New York Times. ^ "Changes in support for IBM OS/2 Warp 4 and OS/2 Warp Server for e-business". IBM. 12 July 2005. Archived from the original on 27 April 2006. Retrieved 9 June 2006. ^ "eComStation (eCS) history". Archived from the original on 2021-02-04. Retrieved 2020-08-29. ^ a b Sanders, James (2015-11-02). "OS/2: Blue Lion to be the next distro of the 28-year-old OS". Archived from the original on 25 February 2023. ^ Sanders, James (2016-08-31). "OS/2 resurrected: Blue Lion becomes ArcaOS, details emerge for upcoming release". TechRepublic. Archived from the original on 25 October 2022. ^ a b Sharwood, Simon (19 May 2017). "What is dead may never die: a new version of OS/2 just arrived". The Register. Archived from the original on 19 January 2023. ^ "Joint Development Agreement Between International Business Machines Corporation And Microsoft Corporation" (PDF). Tech Insider. Archived (PDF) from the original on August 15, 2021. Retrieved August 15, 2021. ^ Michal Necasek (April 2, 1987). "Microsoft Operating System/2 With Windows Presentation Manager Provides Foundation for Next Generation of Personal Computer Industry". The History of OS/2 (Press release). Archived from the original on April 10, 2010. Retrieved March 25, 2013. ^ a b Vose, G. Michael (July 1988). "Reader's Guide to OS/2". Book Reviews. Byte. pp. 51–54. Retrieved 2025-04-12. ^ "DBA Certification Course (Summer 2008) Chapter 1: DB2 Products and Tools" (PDF). Users.informatik.uni-halle.de. Retrieved 2 April 2019. ^ Michal Necasek (2001-07-16). "OS/2 1.0". The History of OS/2. Archived from the original on 2010-08-11. ^ a b David Both (May 2, 2012) [originally published December 19, 1996]. "A Short History of OS/2". DataBook for OS/2 Warp. Archived from the original on February 18, 2013. Retrieved April 9, 2013. ^ H. Gilbert (1995). "HPFS". Das Boot. PC Lube and Tune. Archived from the original on 2006-07-14. Retrieved 2006-06-09. ^ Bob Eager (28 October 2000). "Implementation of extended attributes on the FAT file system". Tavi Systems. ^ Iacobucci, Ed; foreword by Bill Gates (1988). "Foreword". OS/2 Programmer's Guide. McGraw-Hill Osborne Media. ISBN 0-07-881300-X. I believe OS/2 is destined to be the most important operating system, and possibly program, of all time. As the successor to DOS, which has over 10,000,000 systems in use, it creates incredible opportunities for everyone involved with PCs. ^ Bellis, Mary. "The Unusual History of Microsoft Windows". ThoughtCo. Dotdash. ^ Thomas Hormby (25 May 2005). "Windows History (1985–1994)". osviews.com. Archived from the original on March 12, 2006. Retrieved April 9, 2013. ^ Michal Necasek (2001-10-29). "OS/2 1.1 and 1.2: The Early Years". The History of OS/2. Archived from the original on 2006-06-13. ^ Paul Thurrott (24 January 2003). "Windows Server 2003: The Road To Gold". winsupersite.com. Archived from the original on 4 June 2010. ^ Gates, Bill; Myhrvold, Nathan; Rinearson, Peter (1996-10-08). The Road Ahead. ISBN 0-670-77289-5. ^ Michael Lee Vasu; Debra W. Stewart; G. David Garson (1998-03-03). Organizational behavior and public management. Taylor & Francis. p. 268. ISBN 978-0-8247-0135-2. ^ Gordon Letwin. ""What's happening to OS/2," a Usenet post by Gordon Letwin from August 1995, the point of view of a Microsoft employee". Google Groups. Archived from the original on 2007-09-23. Retrieved 2007-10-18. ^ Gates, Bill. "Bill Gates Interview". Computer History Collection (transcript of a Video History interview). Interviewed by David Allison. National Museum of American History, Smithsonian Institution. Retrieved April 10, 2013. ^ Microsoft (2000). Inside Out, Microsoft—In Our Own Words. Warner Business Books. ISBN 0-446-52739-4. ^ Microsoft (11 September 2008). "Compatibility of OS/2-based Applications and APIs". Microsoft TechNet. Microsoft. Retrieved April 9, 2013. ^ Corr, O. Casey (1992-04-06). "GIANT-KILLER? Microsoft mounts challenge to IBM". The Baltimore Sun. Archived from the original (Not available in the EU; check the archive URL instead) on 2020-03-05. ^ Robert X. Cringely (27 April 2006). "Killer Apps: For Apple's Windows Strategy to Work, It Must Replace Microsoft Office and Buy Adobe Systems". pbs.org. ^ Microsoft (6 November 1999). "Windows 3.1 Standard Mode and the VCPI". support.microsoft.com. Archived from the original on March 9, 2013. ^ "Using OS/2 Boot Manager to install multiple operating systems on a single hard file - IBM PC/IBM IntelliStation". www.ibm.com. 2014-08-05. Retrieved 2025-05-05. ^ Christian Alice Scarborough (15 September 1998). "Team OS/2 Frequently Asked Questions". ^ "OS/2 Games Setting Archive". Game Zero magazine. 6 March 1995. Archived from the original on 14 June 2006. Retrieved 9 June 2006. ^ a b Bidmead, Chris (July 1994). "OS/2 for Windows". Personal Computer World. pp. 251–252. ^ Timothy F. Sipples (20 February 1995). "OS/2 Warp Frequently Asked Questions List". ^ "Biography for Kate Mulgrew". Internet Movie Database. In 1996, was contracted by IBM to help promote the latest release of OS/2 Warp, version 4 (previously codenamed Merlin), due to associations with Star Trek. ^ "In Search of Stupidity, Excerpts from Chapter 6". Insearchofstupidity.com. Archived from the original on January 27, 2013. Retrieved April 9, 2013. ^ Chapman, Merrill R. (2006). In Search of Stupidity: Over 20 Years of High-tech Marketing Disasters (Paperback) (2nd ed.). Berkeley, California: Apress. p. 108). ISBN 9781590597217. OCLC 71275572. They rented a hall in New York City and invited hundreds to see Patrick Stewart, the then current captain of the Starship Enterprise to help roll out the product in a gala event. (Stewart was a no-show. ^ Reimer, Jeremy (November 2013). "Half an operating system: The triumph and tragedy of OS/2". Ars Technica. Archived from the original on 26 March 2023. Retrieved 4 April 2023. ^ Fleisch, Brett D; Allan, Mark (September 23, 1997). Workplace Microkernel and OS: A Case Study (Technical report). John Wiley & Sons, Ltd. Archived from the original (PostScript) on August 24, 2007. Retrieved March 25, 2013. ^ Edwards, Ben J. (26 September 2021). "OS/2's Last Stand: IBM OS/2 Warp 4 Turns 25". www.howtogeek.com. HowToGeek. Retrieved 26 September 2021. ^ Lewis, Peter H. (8 August 1995). "PERSONAL COMPUTERS; OS/2 No Longer at Home at Home". The New York Times. ^ Wasserman, Elizabeth (June 8, 1999). "MS/DoJ: Microsoft urged IBM to yank Smart Suite". PC World. Archived from the original on January 16, 2014. Retrieved April 10, 2013. ^ "C. The Similar Experiences of Other Firms in Dealing with Microsoft". Albion.com. Retrieved 2013-03-20. ^ "OS2 World Community Forum – Index". Os2world.com. Retrieved 2013-03-20. ^ a b IBM. "End of Standard Support". Archived from the original on 2006-09-09. Retrieved 2006-08-08. ^ "OS/2 Warp: Warranties and Licenses". IBM. Retrieved April 9, 2013. ^ "Migration Station". IBM. Archived from the original on May 13, 2010. Retrieved April 9, 2013. ^ "IBM Redbooks | OS/2 Server Transition". Redbooks.ibm.com. 2003-10-06. Archived from the original on 2014-02-01. Retrieved 2013-03-20. ^ "3346/GEN/K (1–32) (Page 1)" (PDF). Archived from the original (PDF) on 2012-03-18. Retrieved 2013-03-20. ^ Dan Casey: OS/2 Voice Press Release Archived 2011-10-08 at the Wayback Machine - 21 Sept 1999. ^ "OS/2, ArcaOS and eComStation Versions and Languages". Retrieved 2020-08-24. ^ "News :: eComStation 2.1 GA available". eComStation.com. Archived from the original on 2013-05-12. Retrieved 2013-03-20. ^ "Roadmap: ArcaOS". arcanoae.com. 2023-09-02. Retrieved 2023-10-21. ^ "Slashdot: IBM Won't Open-Source OS/2". 22 January 2008. ^ "OS/2 Warp History". Archived from the original on January 27, 2013. Retrieved April 9, 2013. ^ "IBM OS/2 Warp History". 2004-06-24. Archived from the original on 2008-09-19. Retrieved 2008-09-09. ^ The Art of Unix Programming p. 66 ISBN 0-13-142901-9, ISBN 978-0-13-142901-7 ^ "Open Object REXX FAQ". Retrieved 2011-07-05. ^ "OS/2 Timeline". OS/2 Museum. Retrieved 2016-12-19. ^ "PC Magazine (archives scanned by Google)" (PDF). PC Magazine. Retrieved 2016-12-19. ^ "JaTomes Help - OS/2 Batch File Commands". www.jatomes.com. Archived from the original on 2019-04-14. Retrieved 2019-04-14. ^ "JaTomes Help - OS/2 Commands". www.jatomes.com. Archived from the original on 2019-04-14. Retrieved 2019-04-14. ^ "OS/2 to Linux Client Transition" (PDF). IBM redbook. 2004. p. 9. Archived from the original (PDF) on 2011-11-24. Retrieved 2011-07-05. ^ The VMware Team. "A Notice to OS/2 Users". Archived from the original on 2006-04-20. ^ "Guest Operating System Installation Guide - eComStation 2.x". VMware. 2014. Retrieved 2022-01-13. ^ VirtualBox. "VirtualBox Guest_OSes". ^ "Virtual Machine Configuration". Arca Noae. Retrieved 2020-09-05. ^ "ArcaOS 5.0 Changes". arcanoae.com. Retrieved 2020-09-04. ^ Timothy Prickett Morgan. "Breaking News—Parallels Joins the PC and Server Virtualization Fray". itjungle.com. Archived from the original on 2007-04-03. Retrieved 2007-10-19. ^ John F. Morar, David M. Chess. "the effect of computer viruses on OS/2 and Warp". research.ibm.com. Archived from the original on 2011-06-04. ^ "OS/2 Power Wiki : ClamAV". January 2011. Archived from the original on 2010-10-03. Retrieved 2012-08-17. ^ see IBM Developer Connection for OS/2, Internal Fixpack 12J ^ Bidmead, Chris (July 1994). "Just jamming". Personal Computer World. pp. 565–568. ^ KbdGetConsole() and DosWaitMuxWaitSem(), see Control Program Programming Guide and Reference, IBM OS/2 Toolkit ^ "Brazilian banks look to Linux for ATMs". Archived from the original on 2012-07-28. Retrieved 2008-05-27. ^ "NetIQ Manages Over 1000 Windows Servers for One of Australia's Largest Banks; ANZ Bank Completes Roll Out of NetIQ Management Software to 1,300 Servers" (fee). Business Wire. January 24, 2002. Retrieved 2010-01-24.[permanent dead link][dead link] ^ a b McCracken, Harry (April 2, 2012). "25 Years of IBM's OS/2: The Strange Days and Surprising Afterlife of a Legendary Operating System". Time. Technologizer. Retrieved April 9, 2013. ^ Subway History: How OS/2 Powered The NYC Subway For Decades Archived 2019-07-30 at the Wayback Machine, Andrew Egan, Jun 13, 2019, Tedium ^ "The BYTE Awards". BYTE. January 1989. p. 327. ^ "IBM's Developer Support News 1995 Issue 4". Public.dhe.ibm.com. Archived from the original on 2022-04-08. Retrieved 2021-10-11. ^ "InfoWorld". Books.google.com. 25 March 1996. p. 72. ^ "IBM 2074 Model 2 Console Support Controller — Consolidating and Streamlining Console Support Function". 26 February 2002. Retrieved 19 June 2014. ^ IBM TotalStorage 3494 Tape Library: A Practical Guide to Tape Drives and Tape Automation. IBM. 2009. p. 420. ISBN 978-0-7384-3222-9. ^ "IBM 3746 Software Plan" (PDF). IBM. ^ "IBM Announcement Letter". IBM. Retrieved 30 December 2011. ^ "IBM Announcement Letter". Retrieved 30 December 2011. ^ "Support Element Operations Guide". Archived from the original on 2014-03-05. Retrieved 30 December 2011. Harvey M. Deitel and Michael S. Kogan (1992). The Design of OS/2. Addison-Wesley. ISBN 0-201-54889-5. Letwin, Gordon (1988). Inside OS/2. Microsoft Press. ISBN 1-55615-117-9. Pascal, Zachary (1994). Showstopper! The Breakneck Race to Create Windows NT and the Next Generation at Microsoft. Warner Books. ISBN 0-02-935671-7. Peter Moylan (2004-07-23). "Some fundamental OS/2 concepts". Archived from the original on 2022-12-17. Michal Necasek (2005-12-03). "OS/2 Warp, PowerPC Edition". The History of OS/2. Archived from the original on August 12, 2010. Retrieved April 10, 2013.—Necasek discusses an aborted port to PowerPC machines. Reimer, Jeremy. "Half an operating system: The triumph and tragedy of OS/2". Archived from the original on 10 December 2013. Retrieved 12 December 2013. os2world.com – Community of OS/2 users ecomstation.ru – Community of eComStation and OS/2 users netlabs.org – OpenSource Software for OS/2 and eCS OS/2 FAQ hobbes.nmsu.edu – The OS/2 software repository Archived 2020-04-06 at the Wayback Machine EDM/2 – The source for OS/2 developers eCSoft/2 – The OS/2 and eComstation software guide Archived 2010-06-08 at the Wayback Machine osFree an open source project to build an OS/2 clone operating system Voyager Project, a defunct project to reimplement OS/2 on modern technology OS/2 to Linux API porting project Open Source OS/2 API implementation for Windows Microsoft documentation of OS/2 API compatibility with Windows NT The History of OS/2 Technical details of OS/2 OS/2 Warp 4 Installation and Update Manual; with boot disks and many links Retrieved from " 3 This is the talk page for discussing improvements to the OS/2 article. This is not a forum for general discussion of the article's subject. Put new text under old text. Click here to start a new topic. New to Wikipedia? Welcome! Learn to edit; get help. Assume good faith Be polite and avoid personal attacks Be welcoming to newcomers Seek dispute resolution if needed Article policies Neutral point of view No original research Verifiability Find sources: Google (books · news · scholar · free images · WP refs) · FENS · JSTOR · TWLArchives: 1Auto-archiving period: 12 months This article is rated C-class on Wikipedia's content assessment scale.It is of interest to multiple WikiProjects.Computing: Software Mid-importanceThis article is within the scope of WikiProject Computing, a collaborative effort to improve the coverage of computers, computing, and information technology on Wikipedia. If you would like to participate, please visit the project page, where you can join the discussion and see a list of open tasks.ComputingWikipedia:WikiProject ComputingTemplate:WikiProject ComputingComputingMidThis article has been rated as Mid-importance on the project's importance scale.This article is supported by WikiProject Software (assessed as Low-importance). Microsoft Mid-importanceThis article is within the scope of WikiProject Microsoft, a collaborative effort to improve the coverage of articles relating to Microsoft on Wikipedia. If you would like to participate, please visit the project page, where you can join the discussion and see a list of open tasks.MicrosoftWikipedia:WikiProject MicrosoftTemplate:WikiProject MicrosoftMicrosoftMidThis article has been rated as Mid-importance on the project's importance scale. Technology Technology portalThis article is within the scope of WikiProject Technology, a collaborative effort to improve the coverage of technology on Wikipedia. If you would like to participate, please visit the project page, where you can join the discussion and see a list of open tasks.TechnologyWikipedia:WikiProject TechnologyTemplate:WikiProject TechnologyTechnology The following references may be useful when improving this article in the future: Reimer, Jeremy (25 November 2013). "Half an operating system: The triumph and tragedy of OS/2". Ars Technica. Condé Nast. ESR has a good resume of history of OS/2 in The Art of Unix Programming: see The Art of Unix Programming Davidme — Preceding undated comment added 07:53, 29 June 2007 (UTC)[reply] Someone(s) should flesh out the following summary of a neglected part of the OS/2 history. I can contribute, but certain authoritative references might take considerable effort to obtain. IBM was a member of the Open Software Foundation, and participated with Digital Equipment Corp. on the OSF Research Institute's OSF/1 Mach kernel (OSFMK 7.3 kernel) microkernel design. That microkernel was derived from both the CMU CS Mach microkernel and the CMU CS Alpha real-time microkernel. Alpha's contributions were primarily application-specific scheduling algorithms and distributed threads. IBM Boca and Austin (VP Larry Loucks) decided to base the microkernel of Workplace OS for Power PC on that OSF microkernel. DEC had acquired most of the key Alpha designers, and was developing a real-time OS product based on MK7.3A. IBM contracted with the DEC Real-Time Business Unit to consult on the design and implementation of the WPOS/PPC microkernel. That collaboration included some key Alpha and MK7.3A designers at the OSF/RI. The IBM WPOS/PPC project was subsequently cancelled (as is documented herein), the OSF morphed into being the Object Management Group, dissolving the RI (its key MK7.3A designer went to Apple), and DEC's Real-Time Business Unit was terminated when Compaq purchased DEC. E Douglas Jensen (talk) 21:13, 28 January 2023 (UTC)[reply] during the mid 90s, os/2 was still under development and was a new thing. a coming soon thing. the article also says that protected mode was present in the 286 processors. it was not. i tested protected mode games on a 286 processor and they did not work, because of a lack of protected mode. this is not a joke 84.212.100.141 (talk) 14:58, 1 April 2023 (UTC)[reply] I am unsure as to what specifically your objection "during the mid 90s, os/2 was still under development and was a new thing. a coming soon thing." refers, but feel free to be more specific as to what sections or sentences you feel are incorrect or inappropriate. As to the 286 protected mode, all RS I have been able to find documents the ability of the 286 to enter protected mode, but it certainly seemed like it was very difficult to use in practice, and one of it's main criticisms was especially the fact that you could not escape protected mode without resetting the 286. This thread on stackexchange explains in detail how to put a 286 in protected mode, if you are interested. askeuhd (talk) 10:36, 4 April 2023 (UTC)[reply] 286 did have a protected mode and it could be 'exited' (return to RM). I think the OP was trying to run 386 PM software on a 286 which isn't going to work for obvious reasons. 57.135.233.22 (talk) 15:37, 8 May 2024 (UTC)[reply] The commands list (OS/2#Commands) seems to break up the general flow and seems to provide excessive information considering the rest of the article. Seems out of place, placing {{summarize section}} tag until further input has been received on what should be done with this section. Vghfr (talk) 04:53, 8 January 2024 (UTC)[reply] This section heading seems fundamentally wrong – Windows 3.0 compatibility already shipped in 1992's OS/2 2.0, as the text of this section itself confirms. OS/2 2.1 brought about improved Windows compatibility (i.e. support for Windows 3.1 as opposed to Windows 3.0), but Windows support was already there in 1992's OS/2 2.0 SomethingForDeletion (talk) 07:47, 6 April 2025 (UTC)[reply] Retrieved from " 4 The following pages link to Talk:OS/2 External tools (link count transclusion count sorted list) · See help page for transcluding these entries Showing 26 items. View (previous 50 | next 50) (20 | 50 | 100 | 250 | 500)Talk:OS (links | edit) Talk:The Hobbit (film series)/Archive 2 (links | edit) Talk:Derry/Archive 9 (links | edit) Talk:OS/2/Archive 1 (transclusion) (links | edit) User:Blaxthos/IBM (links | edit) User:Blaxthos/IBM/Articles/Software (links | edit) User:Blaxthos/IBM/Articles (links | edit) User:Saikat 2012/sandbox (links | edit) User:Becky Sayles/sandbox/awb (links | edit) User:Syed yasser ahmed/sandbox (links | edit) User talk:Bruce Bathurst (links | edit) User talk:Abd/Archive 16 (links | edit) User talk:Davidgothberg/Archive 6 (links | edit) User talk:DMarti (links | edit) User talk:Tom29739/Archive 2 (links | edit) User talk:Tom29739/Archive 5 (links | edit) User talk:Doniago/Archive 70 (links | edit) User talk:182.66.30.147 (links | edit) User talk:Callmemirela/Archive 25 (links | edit) Wikipedia:Subpages (links | edit) Wikipedia:WikiProject Spam/LinkSearch/groups.google.com (links | edit) Wikipedia:Bot requests/Archive 22 (links | edit) Wikipedia:Village pump (technical)/Archive 89 (links | edit) Wikipedia:Redirects for discussion/Log/2014 January 5 (links | edit) Wikipedia:Redirects for discussion/Log/2014 January 15 (links | edit) Wikipedia:Village pump (technical)/Archive 174 (links | edit) View (previous 50 | next 50) (20 | 50 | 100 | 250 | 500) Retrieved from " WhatLinksHere/Talk:OS/2" How can financial brands set themselves apart through visual storytelling? Our experts explain how.Learn MoreThe Motorsport Images Collections captures events from 1895 to today’s most recent coverage.Discover The CollectionCurated, compelling, and worth your time. Explore our latest gallery of Editors’ Picks.Browse Editors' FavoritesHow can financial brands set themselves apart through visual storytelling? Our experts explain how.Learn MoreThe Motorsport Images Collections captures events from 1895 to today’s most recent coverage.Discover The CollectionCurated, compelling, and worth your time. Explore our latest gallery of Editors’ Picks.Browse Editors' Favorites