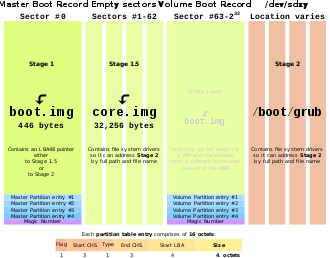
**GNU GRUB** (short for **GNU GRand Unified Bootloader**, commonly referred to as **GRUB**) is a [boot loader](https://en.wikipedia.org/wiki/Boot_loader) package from the [GNU Project](https://en.wikipedia.org/wiki/GNU_Project). GRUB is the [reference implementation](https://en.wikipedia.org/wiki/Reference_implementation) of the [Free Software Foundation](https://en.wikipedia.org/wiki/Free_Software_Foundation)'s [Multiboot Specification](https://en.wikipedia.org/wiki/Multiboot_Specification), which provides a user the choice to boot one of multiple [operating systems](https://en.wikipedia.org/wiki/Operating_system) installed on a computer or select a specific [kernel](https://en.wikipedia.org/wiki/Kernel_(computer_science)) configuration available on a particular operating system's partitions.

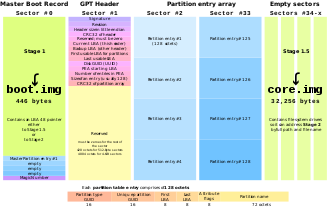
GNU GRUB was developed from a package called the *Grand Unified Bootloader* (a play on [Grand Unified Theory](https://en.wikipedia.org/wiki/Grand_Unified_Theory)[[5]](https://en.wikipedia.org/wiki/GNU_GRUB#cite_note-5)). It is predominantly used for [Unix-like](https://en.wikipedia.org/wiki/Unix-like) systems. The [GNU operating system](https://en.wikipedia.org/wiki/GNU) uses GNU GRUB as its boot loader, as do most [Linux distributions](https://en.wikipedia.org/wiki/Linux_distribution) and the [Solaris operating system](https://en.wikipedia.org/wiki/Solaris_(operating_system)) on x86 systems, starting with the Solaris 10 1/06 release.

GRUB is often used as a OS picker on startup for dual-boot computers, for instance, where one system is Linux and the other Windows.

**Operation**

[](https://en.wikipedia.org/wiki/File:GNU_GRUB_on_MBR_partitioned_hard_disk_drives.svg)

GRUB2 on [MBR](https://en.wikipedia.org/wiki/Master_boot_record)-partitioned hard drive; stage 1 (boot.img) can alternatively be written into one of the [partition boot sectors](https://en.wikipedia.org/wiki/Volume_boot_record).

[](https://en.wikipedia.org/wiki/File:GNU_GRUB_on_GPT_partitioned_hard_disk_drives.svg)

GRUB2 on a [GPT](https://en.wikipedia.org/wiki/GUID_Partition_Table)-partitioned hard drive, booting on [BIOS firmware](https://en.wikipedia.org/wiki/BIOS), or [UEFI compatibility mode (CSM)](https://en.wikipedia.org/wiki/Unified_Extensible_Firmware_Interface#CSM_booting)

**Booting**

See also: [UEFI disk device compatibility](https://en.wikipedia.org/wiki/UEFI_disk_device_compatibility)

When a computer is turned on, [BIOS](https://en.wikipedia.org/wiki/BIOS) finds the configured primary bootable device (usually the computer's hard disk) and loads and executes the initial [bootstrap](https://en.wikipedia.org/wiki/Bootstrapping_(computing)) program from the [master boot record](https://en.wikipedia.org/wiki/Master_boot_record) (MBR). The MBR is the first [sector](https://en.wikipedia.org/wiki/Disk_sector) of the hard disk, with zero as its [offset](https://en.wikipedia.org/wiki/Offset_(computer_science)) (sectors counting starts at zero). For a long time, the size of a sector has been 512 bytes, but since 2009 there are hard disks available with a sector size of 4096 bytes, called [Advanced Format](https://en.wikipedia.org/wiki/Advanced_Format) disks. As of October 2013, such hard disks are still accessed in 512-byte sectors, by utilizing the [512e emulation](https://en.wikipedia.org/wiki/Advanced_Format#512e).[[7]](https://en.wikipedia.org/wiki/GNU_GRUB#cite_note-7)

The legacy [MBR partition table](https://en.wikipedia.org/wiki/MBR_partition_table) supports a maximum of four partitions and occupies 64 bytes, combined. Together with the optional [disk signature](https://en.wikipedia.org/wiki/MBR_disk_signature) (four bytes) and [disk timestamp](https://en.wikipedia.org/wiki/MBR_disk_timestamp) (six bytes), this leaves between 434 and 446 bytes available for the [machine code](https://en.wikipedia.org/wiki/Machine_code) of a boot loader. Although such a small space can be sufficient for very simple boot loaders,[[8]](https://en.wikipedia.org/wiki/GNU_GRUB#cite_note-8) it is not big enough to contain a boot loader supporting complex and multiple [file systems](https://en.wikipedia.org/wiki/File_system), menu-driven selection of boot choices, etc. Boot loaders with bigger footprints are thus split into pieces, where the smallest piece fits into and resides within the MBR, while larger piece(s) are stored in other locations (for example, into empty sectors between the MBR and the first partition) and invoked by the boot loader's MBR code.

Operating system [kernel](https://en.wikipedia.org/wiki/Kernel_(computing)) images are in most cases files residing on appropriate file systems, but the concept of a file system is unknown to the BIOS. Thus, in BIOS-based systems, the duty of a boot loader is to access the content of those files, so it can be loaded into the [RAM](https://en.wikipedia.org/wiki/RAM) and executed.

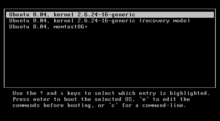
One possible approach for boot loaders to load kernel images is by directly accessing hard disk sectors without understanding the underlying file system. Usually, an additional level of [indirection](https://en.wikipedia.org/wiki/Indirection) is required, in form of *maps* or *map files* – auxiliary files that contain a list of physical sectors occupied by kernel images. Such maps need to be updated each time a kernel image changes its physical location on disk, due to installing new kernel images, file system defragmentation etc. Also, in case of the maps changing their physical location, their locations need to be updated within the boot loader's MBR code, so the sectors indirection mechanism continues to work. This is not only cumbersome, but it also leaves the system in need of manual repairs in case something goes wrong during system updates.[[9]](https://en.wikipedia.org/wiki/GNU_GRUB#cite_note-boot-admin-9)

Another approach is to make a boot loader aware of the underlying file systems, so kernel images are configured and accessed using their actual [file paths](https://en.wikipedia.org/wiki/Path_(computing)). That requires a boot loader to contain a driver for each of the supported file systems, so they can be understood and accessed by the boot loader itself. This approach eliminates the need for hardcoded locations of hard disk sectors and existence of map files, and does not require MBR updates after the kernel images are added or moved around. Configuration of a boot loader is stored in a regular file, which is also accessed in a file system-aware way to obtain boot configurations before the actual booting of any kernel images. As a result, the possibility for things to go wrong during various system updates is significantly reduced. As a downside, such boot loaders have increased internal complexity and even bigger footprints.[[9]](https://en.wikipedia.org/wiki/GNU_GRUB#cite_note-boot-admin-9)

GNU GRUB uses the second approach, by understanding the underlying file systems. The boot loader itself is split into multiple *stages*, allowing for itself to fit within the MBR boot scheme.

Two major versions of GRUB are in common use: GRUB version 1, called *GRUB legacy,* is only prevalent in older releases of Linux distributions, some of which are still in use and supported, for example [CentOS](https://en.wikipedia.org/wiki/CentOS) 5. *GRUB 2* was written from scratch and intended to replace its predecessor, and is now used by a majority of Linux distributions.

**Version 0 (GRUB Legacy)**

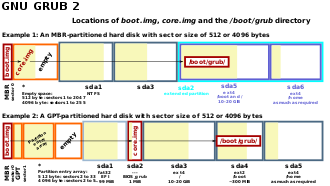
[](https://en.wikipedia.org/wiki/File:GRUB_screenshot.png)

GRUB v1 menu (running as part of [Ubuntu](https://en.wikipedia.org/wiki/Ubuntu_(operating_system)) 8.04 installation)

GRUB 0.x follows a two-stage approach. The master boot record (MBR) usually contains GRUB *stage 1,* or can contain a standard MBR implementation which [chainloads](https://en.wikipedia.org/wiki/Chain_loading) GRUB *stage 1* from the active [partition's boot sector](https://en.wikipedia.org/wiki/Partition_Boot_Sector). Given the small size of a boot sector (512 bytes), *stage 1* can do little more than load the next stage of GRUB by loading a few disk sectors from a fixed location near the start of the disk (within its first 1024 cylinders).

*Stage 1* can load *stage 2* directly, but it is normally set up to load the *stage 1.5.*, located in the first 30 [KiB](https://en.wikipedia.org/wiki/KiB) of hard disk immediately following the MBR and before the first partition. In case this space is not available (unusual partition table, special disk drivers, [GPT](https://en.wikipedia.org/wiki/GUID_Partition_Table) or [LVM](https://en.wikipedia.org/wiki/Logical_Volume_Manager_(Linux)) disk) the install of *stage 1.5* will fail. The *stage 1.5* image contains file system drivers, enabling it to directly load *stage 2* from any known location in the filesystem, for example from /boot/grub. *Stage 2* will then load the default configuration file and any other modules needed.

**Version 2 (GRUB 2)**

[](https://en.wikipedia.org/wiki/File:GNU_GRUB_components.svg)

GRUB 2 – MBR vs. GPT partitioning and boot sequence visualized (systems using BIOS firmware).

**Startup on systems using** [**BIOS firmware**](https://en.wikipedia.org/wiki/BIOS)

* See illustration in last image on the right.
* **1st stage:**boot.img is written to the first 440 bytes of the [Master Boot Record](https://en.wikipedia.org/wiki/Master_boot_record) (MBR in sector 0), or optionally in a [partition's boot sector](https://en.wikipedia.org/wiki/Partition_Boot_Sector) (PBR / VBR). It addresses diskboot.img by a [64-bit LBA](https://en.wikipedia.org/wiki/Logical_block_addressing) address, thus it can load from above the 2 GiB limit of the MBR. The actual sector number is written by grub-install.
* **2nd stage:** diskboot.img is the first sector of core.img (called stage 1.5 in GRUB Legacy) with the sole purpose to load the rest of core.img identified by LBA sector numbers also written by grub-install.
* On [MBR](https://en.wikipedia.org/wiki/Master_boot_record) partitioned disks: core.img is stored in the empty sectors (if available) between the MBR and the first partition. Recent operating systems suggest a 1 MiB gap here for alignment (2047\*512 byte or 255\*4KiB sectors). This gap used to be 62 sectors (31 KiB) as a reminder of the sector number limit of [C/H/S addressing](https://en.wikipedia.org/wiki/Cylinder-head-sector) used by Bios before 1998, therefore core.img is designed to be smaller than 32 KiB.
* On [GPT](https://en.wikipedia.org/wiki/GUID_Partition_Table) partitioned disks: partitions are not limited to 4, thus core.img is written to its own tiny (1 MiB), filesystem-less [BIOS boot partition](https://en.wikipedia.org/wiki/BIOS_boot_partition).
* **3rd stage:** core.img enters [32-bit protected mode](https://en.wikipedia.org/wiki/Protected_mode), uncompresses itself (the kernel of grub and filesystem modules to reach /boot/grub), then loads /boot/grub/<platform>/normal.mod from the partition configured by grub-install. If the partition index has changed, GRUB will be unable to find the normal.mod, and presents the user with the GRUB Rescue prompt, where the user “can” find and load normal.mod, or the linux kernel.
* The /boot/grub directory can be located on any partition (GRUB can read many filesystems, including [NTFS](https://en.wikipedia.org/wiki/NTFS)). Depending on how it was installed it's either in the *root* partition of the distribution, or a separate */boot* partition.
* **4th stage:** normal.mod (equivalent of stage 2 in GRUB Legacy) parses /boot/grub/grub.cfg, optionally loads modules (eg. for graphical UI) and shows the menu.

**Startup on systems using** [**UEFI firmware**](https://en.wikipedia.org/wiki/Unified_Extensible_Firmware_Interface)

* Common on motherboards since ca. 2012.
* /efi/<distro>/grubx64.efi is installed as a file in the [EFI System Partition](https://en.wikipedia.org/wiki/EFI_system_partition), and booted by the firmware directly, without a boot.img in sector 0.
* /boot/grub/ can be installed on the [EFI System Partition](https://en.wikipedia.org/wiki/EFI_system_partition) as well.

**After startup**

GRUB presents a menu where the user can choose from operating systems (OS) found by grub-install. GRUB can be configured to automatically load a specified OS after a user-defined timeout. If the timeout is set to zero seconds, pressing and holding ⇧ Shift while the computer is booting makes it possible to access the boot menu.[[10]](https://en.wikipedia.org/wiki/GNU_GRUB#cite_note-howtogeek.com-10)

In the operating system selection menu GRUB accepts a couple of commands:

* By pressing e, it is possible to edit kernel parameters of the selected menu item *before the operating system is started*. The reason for doing this in GRUB (i.e. not editing the parameters in an already booted system) can be an emergency case: the system has failed to boot. Using the kernel parameters line it is possible, among other things, to specify a module to be disabled (blacklisted) for the kernel. This could be required if the specific kernel module is broken and thus prevents boot-up. For example, to blacklist the kernel module nvidia-current, one could append modprobe.blacklist=nvidia-current at the end of the kernel parameters.
* By pressing c, the user enters the GRUB command line. The GRUB command line is not a regular Linux shell, like e.g. [bash](https://en.wikipedia.org/wiki/Bash_(Unix_shell)), and accepts only certain GRUB-specific commands, documented by various Linux distributions.[[11]](https://en.wikipedia.org/wiki/GNU_GRUB#cite_note-11)

Once boot options have been selected, GRUB loads the selected kernel into memory and passes control to the kernel. Alternatively, GRUB can pass control of the boot process to another boot loader, using [*chain loading*](https://en.wikipedia.org/wiki/Chain_loading). This is the method used to load operating systems that do not support the [Multiboot Specification](https://en.wikipedia.org/wiki/Multiboot_Specification) or are not supported directly by GRUB.

**History**

GRUB was initially developed by Erich Boleyn as part of work on booting the [operating system](https://en.wikipedia.org/wiki/Operating_system) [GNU](https://en.wikipedia.org/wiki/GNU)/[Hurd](https://en.wikipedia.org/wiki/GNU_Hurd), developed by the [Free Software Foundation](https://en.wikipedia.org/wiki/Free_Software_Foundation).[[12]](https://en.wikipedia.org/wiki/GNU_GRUB#cite_note-history-12) In 1999, Gordon Matzigkeit and Yoshinori K. Okuji made GRUB an official software package of the [GNU Project](https://en.wikipedia.org/wiki/GNU_Project) and opened the [development process](https://en.wikipedia.org/wiki/Software_development_process) to the public.[[12]](https://en.wikipedia.org/wiki/GNU_GRUB#cite_note-history-12) As of 2014, the majority of Linux distributions have adopted GNU GRUB 2, as well as other systems such as Sony's [PlayStation 4](https://en.wikipedia.org/wiki/PlayStation_4).[[13]](https://en.wikipedia.org/wiki/GNU_GRUB#cite_note-13)

**Development**

GRUB version 1 (also known as "GRUB Legacy") is no longer under development and is being phased out.[[14]](https://en.wikipedia.org/wiki/GNU_GRUB#cite_note-14) The GNU GRUB developers have switched their focus to GRUB 2,[[15]](https://en.wikipedia.org/wiki/GNU_GRUB#cite_note-15) a [complete rewrite](https://en.wikipedia.org/wiki/Rewrite_(programming)) with goals including making GNU GRUB cleaner, more robust, more portable and more powerful. GRUB 2 started under the name **PUPA**. PUPA was supported by the Information-technology Promotion Agency (IPA) in Japan. PUPA was integrated into GRUB 2 development around 2002, when GRUB version 0.9x was renamed GRUB Legacy.

Some of the goals of the GRUB 2 project include support for non-x86 [platforms](https://en.wikipedia.org/wiki/Platform_(computing)), [internationalization and localization](https://en.wikipedia.org/wiki/Internationalization_and_localization), non-ASCII characters, dynamic modules, [memory management](https://en.wikipedia.org/wiki/Memory_management), a scripting [mini-language](https://en.wikipedia.org/wiki/Mini-language), migrating platform specific (x86) code to platform specific modules, and an object-oriented framework. GNU GRUB version 2.00 was officially released on June 26, 2012.[[16]](https://en.wikipedia.org/wiki/GNU_GRUB#cite_note-16)[[17]](https://en.wikipedia.org/wiki/GNU_GRUB#cite_note-17)

Three of the most widely used [Linux distributions](https://en.wikipedia.org/wiki/Linux_distributions) use GRUB 2 as their mainstream boot loader.[[18]](https://en.wikipedia.org/wiki/GNU_GRUB#cite_note-18)[[19]](https://en.wikipedia.org/wiki/GNU_GRUB#cite_note-19)[[20]](https://en.wikipedia.org/wiki/GNU_GRUB#cite_note-20) [Ubuntu](https://en.wikipedia.org/wiki/Ubuntu_(operating_system)) adopted it as the default boot loader in its 9.10 version of October 2009.[[21]](https://en.wikipedia.org/wiki/GNU_GRUB#cite_note-21) [Fedora](https://en.wikipedia.org/wiki/Fedora_(operating_system)) followed suit with Fedora 16 released in November 2011.[[22]](https://en.wikipedia.org/wiki/GNU_GRUB#cite_note-22) [OpenSUSE](https://en.wikipedia.org/wiki/OpenSUSE) adopted GRUB 2 as the default boot loader with its 12.2 release of September 2012.[[23]](https://en.wikipedia.org/wiki/GNU_GRUB#cite_note-23) [Solaris](https://en.wikipedia.org/wiki/Solaris_(operating_system)) also adopted GRUB 2 on the x86 platform in the Solaris 11.1 release.[[24]](https://en.wikipedia.org/wiki/GNU_GRUB#cite_note-24)

In late 2015, the exploit of pressing backspace 28 times to bypass the login password was found and quickly fixed.[[25]](https://en.wikipedia.org/wiki/GNU_GRUB#cite_note-25)[[26]](https://en.wikipedia.org/wiki/GNU_GRUB#cite_note-26)

**Variants**

GNU GRUB is [free and open-source software](https://en.wikipedia.org/wiki/Free_and_open-source_software), so several variants have been created. Some notable ones, which have not been merged into GRUB mainline:

* [OpenSolaris](https://en.wikipedia.org/wiki/OpenSolaris) includes a modified GRUB Legacy that supports [BSD disklabels](https://en.wikipedia.org/wiki/BSD_disklabel), automatic 64-bit kernel selection, and booting from [ZFS](https://en.wikipedia.org/wiki/ZFS) (with [compression](https://en.wikipedia.org/wiki/Data_compression) and multiple boot environments).[[27]](https://en.wikipedia.org/wiki/GNU_GRUB#cite_note-27)[[28]](https://en.wikipedia.org/wiki/GNU_GRUB#cite_note-28)
* [Google Summer of Code](https://en.wikipedia.org/wiki/Google_Summer_of_Code) 2008 had a project to support GRUB legacy to boot from [ext4](https://en.wikipedia.org/wiki/Ext4) formatted partitions.[[29]](https://en.wikipedia.org/wiki/GNU_GRUB#cite_note-29)
* The [Syllable](https://en.wikipedia.org/wiki/Syllable_Desktop) project made a modified version of GRUB to load the system from its [AtheOS File System](https://en.wikipedia.org/wiki/AtheOS_File_System).[[30]](https://en.wikipedia.org/wiki/GNU_GRUB#cite_note-30)
* *TrustedGRUB* extends GRUB by implementing verification of the system integrity and boot process security, using the [Trusted Platform Module](https://en.wikipedia.org/wiki/Trusted_Platform_Module) (TPM).[[31]](https://en.wikipedia.org/wiki/GNU_GRUB#cite_note-31)
* The Intel BIOS Implementation Test Suite (BITS) provides a GRUB environment for testing BIOSes and in particular their initialization of Intel processors, hardware, and technologies. BITS supports scripting via Python, and includes Python APIs to access various low-level functionality of the hardware platform, including ACPI, CPU and chipset registers, PCI, and PCI Express.[[32]](https://en.wikipedia.org/wiki/GNU_GRUB#cite_note-32)
* GRUB4DOS was a now-defunct GRUB legacy fork that improves the installation experience on DOS and [Microsoft Windows](https://en.wikipedia.org/wiki/Microsoft_Windows) by putting everything besides the GRLDR config in one image file. It can be loaded by the [Windows Boot Manager](https://en.wikipedia.org/wiki/Windows_Boot_Manager).[[33]](https://en.wikipedia.org/wiki/GNU_GRUB#cite_note-33)[[34]](https://en.wikipedia.org/wiki/GNU_GRUB#cite_note-34)

**Utilities**

**GRUB configuration tools**

[](https://en.wikipedia.org/wiki/File:StartUp-Manager.png)

[StartUp-Manager](https://en.wikipedia.org/wiki/StartUp-Manager), a program used to configure GRUB

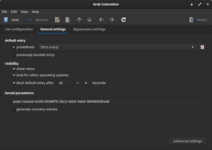
The setup tools in use by various distributions often include modules to set up GRUB. For example, [YaST2](https://en.wikipedia.org/wiki/YaST2) on [SUSE Linux](https://en.wikipedia.org/wiki/SUSE_Linux) and [openSUSE](https://en.wikipedia.org/wiki/OpenSUSE) distributions and [Anaconda](https://en.wikipedia.org/wiki/Anaconda_(installer)) on [Fedora](https://en.wikipedia.org/wiki/Fedora_(operating_system))/[RHEL](https://en.wikipedia.org/wiki/Red_Hat_Enterprise_Linux) distributions. [StartUp-Manager](https://en.wikipedia.org/wiki/StartUp-Manager) and GRUB Customizer are graphical configuration editors for Debian-based distributions. The development of StartUp-Manager stopped on 6 May 2011 after the lead developer cited personal reasons for not actively developing the program.[[35]](https://en.wikipedia.org/wiki/GNU_GRUB#cite_note-35) GRUB Customizer is also available for Arch-based distributions.

For GRUB 2 there are KDE Control Modules.[[36]](https://en.wikipedia.org/wiki/GNU_GRUB#cite_note-36)[[37]](https://en.wikipedia.org/wiki/GNU_GRUB#cite_note-37)

GRLDR ICE is a tiny tool for modifying the default configuration of grldr file for GRUB4DOS.[[38]](https://en.wikipedia.org/wiki/GNU_GRUB#cite_note-38)

**Boot repair utilities**

[Boot-Repair](https://sourceforge.net/p/boot-repair/home/Home/) is a simple graphical tool for recovering from frequent boot-related problems with GRUB and [Microsoft Windows](https://en.wikipedia.org/wiki/Microsoft_Windows) bootloader. This application is available under [GNU GPL license](https://en.wikipedia.org/wiki/GNU_General_Public_License). Boot-Repair can repair GRUB on multiple Linux distributions including, but not limited to, Debian, Ubuntu, [Mint](https://en.wikipedia.org/wiki/Linux_Mint), Fedora, openSUSE, and [Arch Linux](https://en.wikipedia.org/wiki/Arch_Linux).

[](https://en.wikipedia.org/wiki/File:GRUB_customizer.png)

GRUB Customizer

**Installer for Windows**

[Grub2Win](https://sourceforge.net/projects/grub2win) is a Windows open-source software package. It allows GNU GRUB to boot from a Windows directory. The setup program installs GNU GRUB version 2.04 to an NTFS partition. A Windows GUI application is then used to customize the GRUB boot menu, themes, UEFI boot order, scripts etc. All GNU GRUB scripts and commands are supported for both UEFI and legacy systems. Grub2Win can configure GRUB for multiboot of Windows, Ubuntu, openSuse, Fedora and many other Linux distributions. It is freely available under [GNU GPL License](https://en.wikipedia.org/wiki/GNU_General_Public_License) at [SourceForge](https://sourceforge.net/projects/grub2win).

**Alternative boot-managers**

See also: [Comparison of boot loaders](https://en.wikipedia.org/wiki/Comparison_of_boot_loaders)

The strength of GRUB is the wide range of supported platforms, file-systems, and operating systems, making it the default choice for distributions and embedded systems.

However, there are boot-managers targeted at the end user that gives more friendly user experience, graphical OS selector and simpler configuration:

* [rEFInd](https://en.wikipedia.org/wiki/REFInd) – Macintosh-style graphical boot-manager, only for UEFI-based computers (BIOS not supported).
* [CloverEFI](https://sourceforge.net/projects/cloverefiboot/) – Macintosh-style graphical boot-manager for BIOS and UEFI-based computers. Emulates UEFI with a heavily modified [DUET](https://github.com/tianocore/tianocore.github.io/wiki/DuetPkg) from the TianoCore project. Requires a FAT formatted partition even on BIOS systems. As a benefit, it has a basic filesystem driver in the partition boot sector, avoiding the brittleness of GRUB 2nd, 3rd stage and the infamous GRUB Rescue prompt. The user interface looks similar to rEFInd: both inherit from the abandoned boot-manager [rEFIt](http://refit.sourceforge.net/).
* [BOOTBOOT](https://gitlab.com/bztsrc/bootboot) – a simple and easy-to-use, 64 bit only boot protocol. Supports BIOS, UEFI, ARM Raspberry Pi (3 and up), El Torito CDROM, the Linux x86 boot protocol, UEFI PCI Option ROMs etc. It is also Multiboot compatible, so it can be added to a grub.cfg's menuentry. Graphical screen and serial port (for debug messages) are set up for the kernel regardless to the platform. For kernels ELF64 and PE32+ supported with higher-half linkage, and C/C++ and Rust examples provided for hobby OS developers.

Non-graphical alternatives:

* [systemd-boot](https://en.wikipedia.org/wiki/Systemd-boot) – Light, UEFI-only boot-manager with text-based OS selector menu.

**External links**

|  |  |
| --- | --- |
|  | Wikibooks has a book on the topic of: [***GRUB Installation After Windows Installation***](https://en.wikibooks.org/wiki/GRUB_Installation_After_Windows_Installation) |

**How-Tos and troubleshooting**

Distribution wikis have many solutions for common issues and custom setups that might help you:

* [Arch Linux /GRUB](https://wiki.archlinux.org/index.php/GRUB)
* [Ubuntu /Grub2](https://help.ubuntu.com/community/Grub2#Links) (also see Links at the bottom)
* [Fedora /GRUB\_2](https://fedoraproject.org/wiki/GRUB_2)
* [Gentoo /GRUB2](https://wiki.gentoo.org/wiki/GRUB2)
* [Grub2 theme tutorial](http://wiki.rosalab.ru/en/index.php/Grub2_theme_tutorial)

**Documentation**

* [GRUB manual](https://www.gnu.org/software/grub/manual/grub/) – most detailed documentation, including all commands
* [Official website](https://www.gnu.org/software/grub/) [Edit this at Wikidata](https://www.wikidata.org/wiki/Q212885#P856)
* [GRUB wiki archived in 2010](https://web.archive.org/web/20101005232749/http:/grub.enbug.org/)

**Introductory articles**

* [Boot with GRUB](http://www.linuxjournal.com/article/4622), an April 2001 article in [Linux Journal](https://en.wikipedia.org/wiki/Linux_Journal)

**Technicalities**

* [Booting Linux on x86 using Grub2](http://moi.vonos.net/linux/Booting_Linux_on_x86_with_Grub2/) – in-depth article
* [Unified Extensible Firmware Interface](https://en.wikipedia.org/wiki/Unified_Extensible_Firmware_Interface) (UEFI firmware, common since 2012)
* [GUID Partition Table](https://en.wikipedia.org/wiki/GUID_Partition_Table) (GPT) – handles hard drives bigger than 2 [TiB](https://en.wikipedia.org/wiki/TiB) and more than 4 partitions
* [Master boot record](https://en.wikipedia.org/wiki/Master_boot_record) used with [BIOS firmware](https://en.wikipedia.org/wiki/BIOS) (motherboards roughly before 2012)
* [BIOS Boot Specification Version 1.01 (January 11, 1996)](http://www.scs.stanford.edu/05au-cs240c/lab/specsbbs101.pdf) – hard to find

**See also**

* [Free and open-source software portal](https://en.wikipedia.org/wiki/Portal:Free_and_open-source_software)
* [SysLinux (IsoLinux)](https://en.wikipedia.org/wiki/SYSLINUX) – commonly used bootloader on CDs, DVDs
* [NTLDR (BOOTMGR)](https://en.wikipedia.org/wiki/NTLDR) – Windows bootloader
* [Comparison of boot loaders](https://en.wikipedia.org/wiki/Comparison_of_boot_loaders)

**References**

 <https://alpha.gnu.org/gnu/grub/>

  *Kiper, Daniel (July 4, 2019).* [*"GRUB 2.04 release"*](https://lists.gnu.org/archive/html/grub-devel/2019-07/msg00000.html)*. grub-devel (Mailing list). Retrieved July 5, 2019.*

  [*"Ohloh Analysis Summary – GNU GRUB"*](http://www.ohloh.net/p/grub)*.* [*Ohloh*](https://en.wikipedia.org/wiki/Ohloh)*. Retrieved May 12, 2010.*

  [*"GNU GRUB license"*](https://archive.is/20130911174027/http:/bzr.savannah.gnu.org/lh/grub/trunk/grub/annotate/head:/COPYING)*. Archived from* [*the original*](http://bzr.savannah.gnu.org/lh/grub/trunk/grub/annotate/head:/COPYING) *on September 11, 2013.*

  [EnterpriseLinux.com Definitions](http://searchenterpriselinux.techtarget.com/definition/GRUB-GRand-Unified-Bootloader) Definition of GRand Unified Bootloader

  [*"Dual boot with Windows"*](https://wiki.archlinux.org/index.php/Dual_boot_with_Windows)*. archlinux.org.*

  *Smith, Ryan (December 18, 2009).* [*"Western Digital's Advanced Format: The 4K Sector Transition Begins"*](http://www.anandtech.com/show/2888)*.* [*AnandTech*](https://en.wikipedia.org/wiki/AnandTech)*. Retrieved October 10, 2013.*

  [*"mbldr (Master Boot LoaDeR)"*](http://mbldr.sourceforge.net/)*. mbldr.sourceforge.net. 2009. Retrieved October 10, 2013.*

  [*"Booting and Boot Managers"*](http://www-uxsup.csx.cam.ac.uk/pub/doc/suse/suse9.0/adminguide-9.0/node8.html)*. SUSE. Retrieved October 10, 2013.*

  *Hoffman, Chris (September 22, 2014).* [*"How to Configure the GRUB2 Boot Loader's Settings"*](https://www.howtogeek.com/196655/how-to-configure-the-grub2-boot-loaders-settings/)*. HowToGeek.com.*

  [*"GNU GRUB documentation"*](https://www.gnu.org/software/grub/grub-documentation.html)*.*

  [GRUB Manual – 1.2 Grub History](https://www.gnu.org/software/grub/manual/html_node/History.html). Gnu.org (2012-06-23). Retrieved on 2012-12-01.

  [*"PS4 runs Orbis OS, a modified version of FreeBSD that's similar to Linux"*](http://www.extremetech.com/gaming/159476-ps4-runs-orbis-os-a-modified-version-of-freebsd-thats-similar-to-linux)*. extremetech.com. June 24, 2013. Retrieved October 23, 2013.*

  [GNU GRUB – GRUB Legacy](https://www.gnu.org/software/grub/grub-legacy.html). Gnu.org. Retrieved on 2012-12-01.

  [*"GNU GRUB – GRUB 2"*](https://web.archive.org/web/20080607115918/https:/www.gnu.org/software/grub/grub-2.en.html)*. Archived from the original on June 7, 2008. Retrieved April 18, 2014.*.

  *Serbinenko, Vladimir (June 28, 2012).* [*"GRUB 2.00 released"*](http://lists.gnu.org/archive/html/grub-devel/2012-06/msg00093.html)*. grub-devel (Mailing list). Retrieved December 1, 2012.*

  *Larabel, Michael.* [*"GRUB 2.00 Boot-Loader Officially Released"*](https://www.phoronix.com/scan.php?page=news_item&px=MTEyODc)*. Phoronix.com. Retrieved June 28, 2012.*

  *Haddon, Tom (January 26, 2012).* [*"An Introduction to Ubuntu"*](https://web.archive.org/web/20121028160701/http:/www.webjunction.org/documents/webjunction/An_Introduction_to_Ubuntu.html)*. WebJunction. Archived from* [*the original*](http://www.webjunction.org/documents/webjunction/An_Introduction_to_Ubuntu.html) *on October 28, 2012. Retrieved September 21, 2012.*

  *Janssen, Cory.* [*"What is Red Hat Enterprise Linux (RHEL)?"*](http://www.techopedia.com/definition/15777/red-hat-enterprise-linux-rhel)*. Technopedia. Retrieved September 21, 2012.*

  *Varghese, Sam (September 20, 2012).* [*"SUSE chief lists progress since privatisation"*](http://www.itwire.com/business-it-news/open-source/56686-suse-chief-lists-progress-since-privatisation)*. Retrieved September 21, 2012.*

  [*"9.10 Karmic GRUB version"*](http://distrowatch.com/table.php?distribution=ubuntu)*. Distrowatch.com. Retrieved July 8, 2012.*

  [GRUB 2](https://fedoraproject.org/wiki/GRUB_2). FedoraProject. Retrieved on 2012-12-01.

  [openSUSE:Upcoming features – openSUSE](http://en.opensuse.org/openSUSE:Upcoming_features) [Archived](https://web.archive.org/web/20120920065446/http:/en.opensuse.org/openSUSE%3AUpcoming_features) September 20, 2012, at the [Wayback Machine](https://en.wikipedia.org/wiki/Wayback_Machine). En.opensuse.org. Retrieved on 2012-12-01.

  [Solaris 11.1](http://www.oracle.com/technetwork/server-storage/solaris11/overview/solaris11-1-1845817.html). Oracle Retrieved 2015-03-19.

  *Khandelwal, Swati (December 16, 2015).* [*"You can Hack into a Linux Computer just by pressing 'Backspace' 28 times"*](http://thehackernews.com/2015/12/hack-linux-grub-password.html)*. thehackernews.com.*

  *Marco and, Hector; Ripoll, Ismael (December 2015).* [*"Back to 28: Grub2 Authentication 0-Day"*](http://hmarco.org/bugs/CVE-2015-8370-Grub2-authentication-bypass.html#exploit)*.*

  [x86: Modifying Boot Behavior by Editing the GRUB Menu at Boot Time](http://dlc.sun.com/osol/docs/content/SYSADV1/getov.html#fwbqs) [Archived](https://web.archive.org/web/20110719115732/http:/dlc.sun.com/osol/docs/content/SYSADV1/getov.html) July 19, 2011, at the [Wayback Machine](https://en.wikipedia.org/wiki/Wayback_Machine), Modifying Solaris Boot Behavior on x86 Based Systems (Task Map) – System Administration Guide: Basic Administration

  [x86: Supported GRUB Implementations](http://docs.sun.com/app/docs/doc/819-2379/getnk?a=view) [Archived](https://web.archive.org/web/20091021160244/http:/docs.sun.com/app/docs/doc/819-2379/getnk?a=view) October 21, 2009, at the [Wayback Machine](https://en.wikipedia.org/wiki/Wayback_Machine), (System Administration Guide: Basic Administration) – Sun Microsystems

  *Peng, Tao.* [*"Grub4ext4"*](https://code.google.com/archive/p/grub4ext4/)*. Retrieved June 13, 2017.*

  [2.3 Why does Syllable have its own version of GRUB?](http://web.syllable.org/documentation/FAQ.html#2_3) [Archived](https://web.archive.org/web/20110107003333/http:/web.syllable.org/documentation/FAQ.html) January 7, 2011, at the [Wayback Machine](https://en.wikipedia.org/wiki/Wayback_Machine), Syllable Documentation

  [*"TrustedGRUB project"*](http://sourceforge.net/projects/trustedgrub/)*. sourceforge.net. Retrieved November 18, 2014.*

  [BIOS Implementation Test Suite](https://biosbits.org:), Official BITS website

  [*"grub4dos"*](https://sites.google.com/site/grubdos/)*. Google Site. Retrieved April 7, 2019.*

  [*"GRUB for DOS Introduction"*](https://web.archive.org/web/20190407042958/http:/microsaint.narod.ru/_Info_Grub4Dos/Grub4dos_tutorial.html)*. grub4dos.sourceforge.net. 2007. Archived from* [*the original*](http://grub4dos.sourceforge.net/wiki/index.php/Grub4dos_tutorial) *on June 2, 2008. Retrieved April 7, 2019.*

  [*"StartUp-Manager is dead : StartUp-Manager"*](https://launchpad.net/startup-manager/+announcement/8300)*. launchpad.net. Retrieved April 9, 2017.*

  [GRUB2 Bootloader Editor](http://kde-apps.org/content/show.php/GRUB2+Bootloader+Editor?content=139643). Kde-apps.org (2012-06-18). Retrieved on 2012-12-01.

  [*"Grub2 KCM"*](http://kde-apps.org/content/show.php/Grub2+Kcm?content=137886)*. KDE-Apps.org. Retrieved January 27, 2011.*

 [*"Grub4dos tutorial"*](http://microsaint.narod.ru/_Info_Grub4Dos/Grub4dos_tutorial.html)*. Narod.ru.*