**GRUB 2 Manual by a1ve**

**Manual page:** [**https://a1ive.github.io/grub2\_zh.html**](https://a1ive.github.io/grub2_zh.html)

**Author: a1ive** [**a1ive@github**](https://a1ive.github.io/)

Translated by: alacrán from reboot.pro by means of Google translate <http://reboot.pro/>

[**Sample menu**](https://ufc7i3dbowvp7y3sgxok3ewjua--a1ive-github-io.translate.goog/grub2_menu_zh.html)

[**Command list**](https://ufc7i3dbowvp7y3sgxok3ewjua--a1ive-github-io.translate.goog/grub2_cmd_zh.html)

[**Environment variable**](https://ufc7i3dbowvp7y3sgxok3ewjua--a1ive-github-io.translate.goog/grub2_var_zh.html)

[**Equipment and files**](https://ufc7i3dbowvp7y3sgxok3ewjua--a1ive-github-io.translate.goog/grub2_disk_zh.html)

[**theme**](https://ufc7i3dbowvp7y3sgxok3ewjua--a1ive-github-io.translate.goog/grub2_gfx_zh.html)

[**Lua**](https://ufc7i3dbowvp7y3sgxok3ewjua--a1ive-github-io.translate.goog/grub2_lua_zh.html)

[**FatFs**](https://ufc7i3dbowvp7y3sgxok3ewjua--a1ive-github-io.translate.goog/grub2_fatfs_zh.html)

**Sample menu**

The encoding of the menu file must be **UTF-8 without BOM** .

export pager=0;

cat --set=modlist ${prefix}/insmod.lst;

for module in ${modlist}; do insmod ${module};

done;

if [ "${grub\_platform}" = "efi" ]; then getenv -t uint8 SecureBoot grub\_secureboot;

if [ "${grub\_secureboot}" = "1" ]; then sbpolicy -i;

fi;

fi;

loadfont ${prefix}/fonts/unicode.pf2;

export enable\_progress\_indicator=0;

export locale\_dir=${prefix}/locale;

export lang=zh\_CN;

export gfxmode=1024x768;

export gfxpayload=keep;

export color\_normal=white/black;

export color\_highlight=black/white;

terminal\_output gfxterm;

#### FUNCTION ####

function to\_g4d\_path {

unset g4d\_path;

if regexp --set=1:num '^\(hd[0-9]+,[a-zA-Z]\*([0-9]+)\).\*' "${1}"; then

# (hdx,msdosy) (hdx,gpty) (hdx,y)

expr --set=num "${num} - 1";

regexp --set=1:path\_1 --set=2:path\_2 '^(\(hd[0-9]+,)[a-zA-Z]\*[0-9]+(\).\*)' "${1}";

set g4d\_path="${path\_1}${num}${path\_2}";

elif regexp '^\([chf]d[0-9]\*\).\*' "${1}"; then

# (hd) (cd) (fd) (hdx) (cdx) (fdx)

set g4d\_path="${1}";

fi;

}

menuentry "Load grubfm.cfg" {

configfile "${prefix}/grubfm.cfg";

}

menuentry "Boot Arch Linux" {

set root\_uuid=c55da16f-e2af-4603-9e0b-03f5f565ec4a;

search --set=root --file /boot/vmlinuz-linux;

linux /boot/vmlinuz-linux root=/dev/disk/by-uuid/$root\_uuid ro;

initrd /boot/initramfs-linux.img;

}

menuentry "Boot Manjaro LiveCD" {

export iso\_path="/livecd/manjaro-kde.iso";

search --set=root --file "${iso\_path}";

loopback loop "${iso\_path}";

probe --set=rootuuid -u (${root});

export rootuuid;

set root=loop;

configfile /boot/grub/loopback.cfg;

}

menuentry "Boot deepin" {

export iso\_path="/livecd/deepin-15.11-amd64.iso";

search --set=root --file "${iso\_path}";

loopback loop "${iso\_path}";

linux (loop)/live/vmlinuz findiso=${iso\_path} locales=zh\_CN.UTF-8 username=user boot=live config;

initrd (loop)/live/initrd.lz;

}

menuentry "Boot Microsoft Windows 8/10" {

search -s -f /EFI/Microsoft/Boot/bootmgfw.efi;

chainloader -t /EFI/Microsoft/Boot/bootmgfw.efi;

}

menuentry "Boot Microsoft Windows 8/10" {

set winload="${prefix}/${grub\_cpu}-${grub\_platform}/bootmgfw.efi";

set lang=en\_US;

terminal\_output console;

ntboot --win --efi="${winload}" (hd1,gpt4);

}

menuentry "Boot WinPE ISO" {

set iso\_file="(hd0,2)/winpe.iso";

if [ "$grub\_platform" = "efi" ]; then map -f "${iso\_file}";

elif [ "$grub\_platform" = "pc" ]; then to\_g4d\_path "${iso\_file}";

if [ -n "${g4d\_path}" ]; then set g4d\_cmd="map ${g4d\_path} (0xff);map --hook;chainloader (0xff);boot";

linux ${prefix}/grub.exe --config-file=${g4d\_cmd};

else set enable\_progress\_indicator=1;

linux16 ${prefix}/memdisk iso raw;

initrd16 "${iso\_file}";

fi;

boot;

fi;

}

menuentry "Boot WinPE WIM (wimboot)" {

set wim\_file="(hd0,2)/winpe.wim";

set winload="${prefix}/${grub\_cpu}-${grub\_platform}/bootmgfw.efi";

set lang=en\_US;

terminal\_output console;

wimboot --rawwim --testmode=no \

@:bootmgfw.efi:"${winload}" @:boot.wim:"${wim\_file}";

}

menuentry "Boot WinPE WIM (ntboot)" {

set wim\_file="(hd0,2)/winpe.wim";

set winload="${prefix}/${grub\_cpu}-${grub\_platform}/bootmgfw.efi";

set lang=en\_US;

terminal\_output console;

ntboot --testmode=no --efi="${winload}" "${wim\_file}";

}

menuentry "Boot Windows Nt6+ VHD/VHDX" {

set vhd\_file="(hd0,3)/win10.vhd";

set winload="${prefix}/${grub\_cpu}-${grub\_platform}/bootmgfw.efi";

set lang=en\_US;

terminal\_output console;

ntboot --vhd --efi="${winload}" "${vhd\_file}";

}

if [ "$grub\_platform" = "efi" ]; then menuentry "UEFI Firmware Setup" {

reset --fwui;

}

menuentry "UEFI Shell" {

set lang=en\_US;

terminal\_output console;

shell;

}

fi;

menuentry "GRUB Shell" {

commandline;

}

menuentry "Reboot (R)" --hotkey "r" {

reboot;

}

menuentry "Halt (H)" --hotkey "h" {

halt;

}

## Command list

### [EXPRESSION]

​ Same as "test"

### acpi [OPTIONS] FILE1 [FILE2 …]

​ Load ACPI table

​ Modern BIOS systems usually implement ACPI and define various tables to describe the interface between ACPI-compliant operating systems and firmware. In some cases, the tables provided by default are only suitable for certain operating systems, and it may be necessary to replace some of them.

​ Generally, this command will replace the root system description pointer (RSDP) in the extended BIOS data area to point to the new table. If you use the –no-ebda option, only GRUB will get the new table, but GRUB's EFI simulation can use the new table.

* --exclude=TABLE1,TABLE2,…, -x do not load the list
* --load-only=TABLE1,TABLE2,…, -n load list
* --v1, -1 Import version 1 tables into the operating system
* --v2, -2 Import version 2 and 3 tables into the operating system
* --oemid=STRING, -o set the OEMID of RSDP, XSDT and RSDT
* --oemtable=STRING, -t set OEMTABLE ID for RSDP, XSDT and RSDT
* --oemtablerev=n, -r set the OEMTABLE version of RSDP, XSDT and RSDT
* --oemtablecreator=STRING, -c Set the OEMTABLE creator of RSDP, XSDT and RSDT.
* --oemtablecreatorrev=n, -d Set the OEMTABLE creator version of RSDP, XSDT and RSDT.
* --no-ebda, -e Do not update EBDA. It can prevent part of the BIOS from crashing, and it is invalid for the OS that cannot receive RSDP from GRUB.
* **--slic, -s Load as SLIC, automatically modify OEMID and OEMTABLE ID.**
* **--msdm display/load MSDM table**
* **--bgrt use the BMP file as the startup Logo**

### ****alias NAME COMMAND [SUMMARY]****

​ Set alias

### appleloader CMDLINE

​ Apple legacy boot loader.

### authenticate [userlist]

​ Check whether the user is in the user list or is listed in the value of the variable "superusers"

​ If "Super User" is empty, this command returns true.

### background\_color COLOR

​ Set the background color of the mobile terminal

​ The background color can only be changed when using "gfxterm". This command sets the color of the blank area without text. The background color of the text is controlled by the environment variables color\_normal, color\_highlight, menu\_color\_normal, and menu\_color\_highlight.

### background\_image [OPTIONS] [FILE]

​ Set a background image for the activated terminal

​ By default the image will be stretched to fill the entire screen (--mode=stretch)

​ Without parameters, delete the currently loaded background image.

* --mode=stretch/normal, -m Set the background image mode to stretch or normal

### backtrace

​ Print backtracking information

### badram ADDR1,MASK1[,ADDR2,MASK2[,…]]

​ Shield the wrong memory

​ This command informs the memory manager that the specified area of ​​RAM should be filtered out. As long as the loaded kernel obtains its memory map from GRUB, it is still valid after the kernel is loaded. The kernels that support this feature usually include Linux, GNU Mach, FreeBSD kernel and Multiboot kernel.

​ The syntax is the same as that provided by Memtest86+: a list of address/mask pairs. Given a page-aligned address and a base address/mask pair, if all bits of the page-aligned address enabled by the mask match the base address, it means that the page will be filtered.

### blocklist ****[OPTIONS]**** FILE

​ Print the block list of the file

* --set=VAR, -s save to variable
* --disk, -d Calculate the starting sector based on the disk instead of the partition

### ****blscfg**** FILE

​ Import BootLoaderSpec (BLS) configuration

### ****bls\_import**** FILE

​ Same as "blscfg"

### boot

​ Start the loaded operating system

### ****btrfs-info**** DEVICE

​ Display the btrfs partition information of the device

### ****btrfs-mount-subvol**** DEVICE DIRECTORY SUBVOL

​ Set the DIRECTORY directory of the btrfs device as the mount point of the subvolume SUBVOL

### ****btrfs-list-subvols**** [OPTIONS] DEVICE

​ Show all subvolumes on the device DEVICE

* --output=VARIABLE, -o save the output as a variable
* --path-only, -p only show the path of the subvolume
* --id-only, -i only show the id of the subvolume

### ****btrfs-get-default-subvol**** [OPTIONS] DEVICE

​ Display the default subvolume on the device DEVICE

* The parameters are the same as btrfs-list-subvols

### cat [OPTIONS] FILE

​ Display text file content

* --dos allow DOS format newline (CR-LF)
* **--set=VARIABLE, -s save content to variable**

### chainloader [–force|–bpb] FILE ****[ADDR]****

​ Start another bootloader, the default loading address is 0x7c00

### chainloader [OPTIONS] FILE CMDLINE

​ Start EFI executable file

**Warning: Use this command may cause security issues**

* **--alt, -a Use GRUB 2's built-in EFI loader**
* **--text, -t switch to text mode before starting EFI**
* **--boot, -b execute boot immediately**

### ****checktime**** minute hour day month day\_of\_week

​ Check whether the current time meets the requirements, if yes, return 0, otherwise return 1. The syntax is similar to cron under unix.

| **symbol** | **significance** | **Example** |
| --- | --- | --- |
| \\* | Any value (note the asterisk escape) | \\* |
| , | List multiple values | 10, 20, 30 |
| - | Range value | 5-45 |
| / | Step value | 10/5 |

### clear

​ Clear screen

​ Note: You need to use this command before unset debug

### ****clear\_menu****

​ Clear the current menu

**Warning: Be sure to use this command before the disable ESCexport grub\_disable\_esc=1**

### cmosclean byte:bit

​ Clear the value of the CMOS bit in byte:bit

​ Only available on platforms that support CMOS.

### cmosdump

​ Display CMOS raw data

### cmostest byte:bit

​ Test the value of CMOS at byte:bit

​ If the bit is 1, it returns true (0), otherwise it is non-zero.

### cmp FILE1 FILE2

​ Compare two files

​ If the sizes of the two files are different, the sizes will be displayed separately. If the size is the same but the data is different, the first different position and data will be displayed. If they are exactly the same, there is no output. The return value is 0, it means the files are the same, otherwise the files are different.

* **--quiet, -q do not display information**

### ****commandline****

​ Enter GRUB command line

### configfile FILE

​ Load GRUB2 configuration file

### ****crc32**** FILE [VARIABLE]

​ Calculate the CRC32 check code of the file

### cpuid [OPTIONS]

​ Check CPU characteristics

​ If no parameter is added, the default parameter is -l. If the CPU supports this feature, it returns 0.

* --long-mode, -l Check whether the CPU supports 64-bit long mode
* --pae, -p detect whether the CPU supports physical address extension (PAE)

### cryptomount DEVICE | -u UUID | -a | -b

​ Mount the encrypted device (support LUKS/geli), in some cases you need to enter the password interactively

* --uuid, -u mount device by UUID
* --all, -a mount all devices
* --boot, -b mount all devices marked with "boot"

### cutmem FROM[K|M|G] TO[K|M|G]

​ Delete all memory areas within the specified range

### date [OPTIONS] [[year-]month-day] [hour:minute[:second]]

​ Display/set current time

* **--set=VARIABLE, -s save time to variable**

### ****decrement**** VARIABLE

​ Decrease the value of the variable by one

### devicetree FILE

​ Load device tree blob (.dtb)

### ****dd**** OPTIONS

​ Write file/string/hexadecimal number to file

**Warning: This command will cause the loss of data**

* --if=FILE, -i specify the input file
* --str=STRING, -s specify the input string
* --hex=HEX, -h specifies the input hexadecimal number
* --of=FILE, -o specify output file
* --bs=BYTES, -b specifies the block size
* --count=n, -c specifies the number of blocks
* --skip=n skip the first n blocks of input
* --seek=n skip the first n blocks of output

### distrust PUBKEY\_ID

​ Remove PUBKEY\_ID from the trust list

### ****dp**** FILE/DEVICE

​ UEFI Device Path of the output device or file

### drivemap [OPTIONS] FROM\_DEVICE TO\_DEVICE

​ Swap BIOS disk order

* --list, -l list the current disk mapping
* --reset, -r reset all maps to default values
* --swap, -s perform disk mapping

### dump ADDR [SIZE]

​ Display memory content

### echo [OPTIONS] STRING…

​ Display string

​ If the -n option is not added, it will automatically wrap. The backslash escape supports the following sequences:

​ \\-backslash \a-alarm (BEL) \c-prohibit trailing line feed \f-page feed

​ \n – line feed\r – carriage return\t – horizontal tab \v – vertical tab

​ \e0xBF - set character color

* -n do not wrap
* -e enables backslash escape parsing

### ****efi-export-env**** VARIABLE

​ Save GRUB variables to the EFI environment variable GRUB\_ENV

**Warning: Use this command to modify UEFI environment variables**

### ****efi-load-env****

​ Read variables from the EFI environment variable GRUB\_ENV

### ****efiload**** [OPTIONS] FILE

​ Load UEFI driver

* --nc, -n only load the driver, do not connect

### ****efiusb DEVICE****

​ Print USB information

### eval STRING…

​ Connect the parameters together with a single space as a separator, and execute the result as a sequence of GRUB commands.

### exit

​ Exit GRUB

### export VARIABLE ****[=VALUE]**** …

​ Set the variable as a global environment variable

### ****expr**** [OPTIONS] EXPRESSION

​ Calculate mathematical expressions, support +-\* \% operator

**Warning: Division by zero results in unexpected situations such as crash or restart**

* --set=VARIABLE, -s save the result to a variable

### fakebios

​ Create a Legacy-BIOS-like structure to be compatible with existing systems

### false

​ returns false (false)

### file OPTIONS FILE

​ Detect file type

* --is-i386-xen-pae-domu
* --is-x86\_64-xen-domu
* --is-x86-xen-dom0
* --is-x86-multiboot
* --is-x86-multiboot2
* --is-arm-linux
* --is-arm64-linux
* --is-ia64-linux
* --is-mips-linux
* --is-mipsel-linux
* --is-sparc64-linux
* --is-powerpc-linux
* --is-x86-linux
* --is-x86-linux32
* --is-x86-kfreebsd
* --is-i386-kfreebsd
* --is-x86\_64-kfreebsd
* --is-x86-knetbsd
* --is-i386-knetbsd
* --is-x86\_64-knetbsd
* --is-i386-efi
* --is-x86\_64-efi
* --is-ia64-efi
* --is-arm64-efi
* --is-arm-efi
* --is-riscv32-efi
* --is-riscv64-efi
* --is-hibernated-hiberfil
* --is-x86\_64-xnu
* --is-i386-xnu
* --is-xnu-hibr
* --is-x86-bios-bootsector

### fixmmap

​ Fixed the "BlInitializeLibrary failed 0xc000009a" error that appeared when starting Windows on some computers

### fix\_video

​ Fix image display problem

### ****fucksb**** [OPTIONS]

​ Hide the firmware safe boot state in the bootloader stage. If there is no parameter, it will return whether to enable this function, and it will return 1.

* --install, -i enable camouflage function
* --on, -y disguise the safe boot state as on
* --off, -n disguise the safe boot state as off

### fwsetup

​ Restart to enter UEFI firmware settings

### ****getargs**** OPTIONS STRING VARIABLE

​ Get parameters from the command line received from the GRUB 2 EFI file

​ If the command is executed successfully (parameter/value exists), 0 is returned.

* --key, -k Get whether to set this parameter
* --value, -v get the value of the parameter

### ****getenv**** [OPTIONS] EFI\_ENV VARIABLE

​ Get UEFI environment variables

* --guid=GUID, -g set the GUID of the variable to be queried, the default is a global variable
* --type=string/wstring/uint8/hex, -t specifies the variable type as string/wide string/8-bit unsigned integer/hexadecimal data, the default is hexadecimal data

### ****getkey**** [-n] [VARIABLE]

​ Wait for key press and output keyboard scan code

### gettext STRING

​ Translate the string into the current language

​ The current language code is stored in the "lang" variable in the GRUB environment. The translation file in MO format is read from "locale\_dir", usually /boot/grub/locale.

### ****gptprio.next**** OPTIONS [DEVICE]

​ Select the next partition of the GPT disk to boot

* --set-device=VARIABLE, -d save the partition name to a variable
* --set-uuid=VARIABLE, -u save partition UUID to variable

### ****gptrepair**** DEVICE

​ Verify and repair the GPT partition table of the device

**Warning: Use this command may cause data loss**

### gptsync DEVICE [PARTITION[+/-[TYPE]]]…

​ Modify the MBR compatible partition table of the GPT partition table hard disk

​ TYPE is the MBR partition type code. "+" means to activate the partition, "-" means to deactivate the partition.

### halt [–no-apm]

​ Turn off the computer

​ If the "--no-apm" parameter is added, the APM BIOS call will not be executed. Otherwise, APM will be used to shut down the computer.

### hashsum -h HASH [OPTIONS] [-c FILE [-p PREFIX]] [FILE1 [FILE2 …]]

​ Calculate or verify the hash value, if the hash verification succeeds, it returns 0.

* --hash=HASH, -h specifies the hash value type, supports'adler32','crc64','crc32','crc32rfc1510','crc24rfc2440','md4','md5','ripemd160','sha1' ,'sha224','sha256','sha512','sha384','tiger192','tiger','tiger2','whirlpool'
* --check=FILE, -c specifies the hash list file (generated by md5sum under UNIX)
* --prefix=PREFIX, -p specify file directory
* --keep-going, -k Do not stop checking after the first error, stop checking without this option
* --uncompress, -u unzip the file before verifying

### hdparm [OPTIONS] DISK

​ Get/set ATA disk parameters

* --apm=n, -B set advanced power management (APM), 1=low,...,254=high, 255=off
* --power, -C show power mode
* --security-freeze, -F Freeze ATA security settings until reset
* --health, -H show SMART health status
* --aam=n, -M set automatic noise management (AAM), 0=off, 128=quiet,...,254=fast
* --standby-timeout=n, -S set standby timeout, 0=off, 1=5s, 2=10s,..., 240=20m, 241=30m,...
* --standby, -y set to standby mode
* --sleep, -Y set to sleep mode
* --identify, -i show device identification and settings
* --dumpid, -I display the original content of the ATA IDENTIFY sector
* --smart=n disable/enable SMART (0/1)
* --quiet, -q do not display information

### help [PATTERN …]

​ Display help information for built-in commands. If no parameters are added, all available commands are displayed.

### hexdump [OPTIONS] FILE/DEVICE ****[VARIABLE]****

​ Display the hexadecimal data of the file or device. (mem) is a memory device.

* --skip=n, -s skip the first n bytes
* --length=n, -n set the number of bytes read
* **--quiet, -q do not display output**

### ****hiddenentry**** “TITLE” [OPTIONS] [arg …] {COMMAND;…}

​ Add hidden menu, only valid for gfxmenu

​ The parameters are the same as "menuentry"

### inb [OPTIONS] PORT

​ Read 8-bit value from port

* -v=VARIABLE write the read value to the variable

### ****increment**** VARIABLE

​ Increase the value of the variable by one

### ****ini\_get**** [OPTIONS] FILE [SECTION:]KEY

​ Get data from ini file

* --set=VARIABLE, -s save data to variable

### initrd FILE…

​ Load Linux initial memory disk, use after linux

### initrd16 FILE…

​ Load Linux initial memory disk, use after linux16

### ****initrdefi**** FILE

​ Load the Linux initial memory disk, use it after linuxefi

### inl [OPTIONS] PORT

​ Read the 32-bit value from the port, the parameter is the same as "inb"

### insmod MODULE

​ Load GRUB2 module

### inw [OPTIONS] PORT

​ Read 16-bit value from the port, the parameter is the same as "inb"

### ****isotools OPTIONS FILE [VARIABLE]****

​ ISO El Torito related tools

* --offset, -o UEFI El Torito image offset (unit is sector)
* --length, -l UEFI El Torito image size (unit is sector)
* --ventoy, -v check whether the ISO image contains Ventoy Compatible information

### keymap FILE

​ Load keyboard layout

### keystatus [OPTIONS]

​ If the Shift/Ctrl/Alt key is pressed, it returns 0

​ Only some platforms support detecting modifier key status. If no parameters are added, this command is used to detect whether modifier key status is supported.

* --shift, -s detect Shift key
* --ctrl, -c detect Ctrl key
* --alt, -a detect the Alt key

### kfreebsd [OPTIONS] FILE [CMDLINE]

​ Load FreeBSD kernel

### kfreebsd\_loadenv FILE

Load FreeBSD environment variables

### kfreebsd\_module FILE [CMDLINE]

​ Load FreeBSD modules

### kfreebsd\_module\_elf FILE [CMDLINE]

​ Load FreeBSD module (ELF)

### knetbsd [OPTIONS] FILE [CMDLINE]

​ Load NetBSD kernel

### knetbsd\_module FILE [CMDLINE]

​ Load NetBSD module

### knetbsd\_module\_elf FILE [CMDLINE]

​ Load NetBSD module (ELF)

### kopenbsd [OPTIONS] FILE [CMDLINE]

​ Load OpenBSD kernel

### kopenbsd\_ramdisk FILE

​ Load OpenBSD memory disk

### ****linuxefi**** FILE [CMDLINE]

​ Load the Linux kernel

### list\_env [OPTIONS]

​ List all variables in the environment block file

* --file=FILE, -f specifies the file name, the default file name is ${prefix}/grubenv
* --skip-sig, -s skip the signature check of environment files

### list\_trusted

​ List the list of trusted keys

### load\_env [OPTIONS] [VARIABLE …]

​ Load variables from the environment block file, the parameters are the same as "list\_env"

### loadbios BIOS\_DUMP [INT10\_DUMP]

​ Load BIOS dump

### ****loadfile [OPTIONS] FILE****

​ Load file into memory

* --skip=n, -k skip n bytes of the file header
* --length=n, -l specifies the number of bytes read
* --addr=ADDR, -a specify the memory address to load
* --nodecompress, -n do not decompress files automatically
* --set=VARIABLE, -s save the memory file name to the variable

### loopback [OPTIONS] DEVICE FILE

​ Mount the file as a virtual disk

* --delete, -d delete the specified virtual disk
* **--mem, -m copy the file to the memory and mount it, allowing write operations**

### ls [OPTIONS] [FILE …]

​ List devices or files

​ If there are no parameters, all devices are listed. If the parameter is a device name enclosed in brackets, the name of the device file system is printed. If the parameter is a directory specified as an absolute file name, the contents of the directory are listed.

* --long, -l show more detailed information
* --human-readable, -h Display file size in readable format (KB, MB …)
* --all, -a list all files

### lsacpi [OPTIONS]

​ Display ACPI information

* --v1, -1 only show version 1 ACPI table
* --v2, -2 Only show version 2 and version 3 ACPI tables

### lspci [OPTIONS]

​ List PCI devices

* --iospace, -i show I/O space

### lsefi

​ Show EFI handle

### ****lsefienv****

​ List all EFI environment variables

### lsefimmap

​ Show EFI memory map

### lsefisystab

​ Display EFI system table

### ****lua**** [OPTIONS] [FILE]

​ Execute Lua script

* --execute, -e execute a single-line Lua statement
* --load=NAME, -l load library
* --interactive, -i enter interactive mode
* --version, -v show version information

### map [OPTIONS] FILE [DEVICE]

​ Create UEFI virtual disk

* --mem, -m load into memory
* --rt specified memory disk to use RUNTIME\_SERVICES\_DATAthe type of memory
* --blocklist, -l convert to blocklist type disk, which will speed up the read speed of virtual disk and enable write function
* --type=CD/HD/FD, -t specifies the disk type as CD/HD/FD
* --ro, -o prohibit writing to virtual disk
* --eltorito=DISK, -e also specify the drive letter for mounting the El Torito image
* --nb, -n do not start this virtual disk
* --unmap=DISK, -x shield a disk
* --first, -f make the virtual disk ranked first in the disk list to solve Windows startup problems

### md5sum arg…

​ means "hashsum –hash md5 arg …"

### menuentry “TITLE” [OPTIONS] [arg …] {COMMAND;…}

​ Define GRUB menu item, the menu name is TITLE

​ When the menu item is selected and executed, if --id is specified, the value of the environment variable chosen will be set to the value of --id. The commands in the braces will be executed. If the last command is executed successfully and the kernel has been loaded, the boot command will be executed automatically.

​ All parameters (arg …) including TITLE will be passed as positional parameters, and TITLE will be assigned to $1.

* --class=STRING is the category of menu items, and icons are displayed in different categories
* --users=UESR[,USER] List the users allowed to execute this menu
* --hotkey=KEY set hotkey
* --source=STRING Use STRING as menu entry body.
* --id=STRING associate a unique identifier with the menu item. id cannot start with a number, only supports ASCII alphanumeric, underscore and hyphen
* --unrestricted allows all users to execute this menu

​ Special hotkey:

* backspace backspace
* tab
* delete delete key
* insert insert key
* esc exit key
* f1~f12 function keys
* You can also use key codes in the form of hexadecimal numbers, such as 0x46

### ****nes**** FILE [PIXEL\_SIZE WAIT\_KEY\_TIME]

​ NES emulator

### normal [FILE]

​ Enter normal mode and display the GRUB menu

​ In normal mode, commands, file system modules and encryption modules will be automatically loaded, and a complete GRUB script parser can be used. Other modules can be loaded explicitly using insmod.

​ If a file is given, the command will be read from the file. Otherwise, they will be read from $prefix/grub.cfg (if they exist).

​ You can call normal again from normal mode to create a nested environment. For this, configfile is usually used.

### normal\_exit

​ Exit normal mode

​ If this normal instance is not nested in another instance, it returns to rescue mode.

### ****ntboot**** [OPTIONS] FILE

​ Start NT6+ VHD/VHDX/WIM

* --gui, -g enable graphical startup information
* --pause, -p pause before starting
* --vhd, -v specify the file type as VHD/VHDX
* --wim, -w specify the file type as WIM
* --win, -n Windows on boot disk
* --efi=FILE, -e specify the path of bootmgfw.efi, the default is /efi/microsoft/boot/bootmgfw.efi
* --sdi=FILE, -s specify the path of boot.sdi, the default is /boot/boot.sdi
* --dll=FILE, -d specify bootvhd.dll path

### ****nthibr FILE****

​ Check whether hiberfil.sys of NTFS is in hibernation

### ****ntversion (hdx,y) VARIABLE****

​ Get the version information of Windows NT installed on the disk partition (hdx, y)

### outb PORT VALUE [MASK]

​ Write 8-bit value to the port

### outl PORT VALUE [MASK]

​ Write a 32-bit value to the port

### outw PORT VALUE [MASK]

​ Write a 16-bit value to the port

### ****partnew**** OPTIONS DISK PARTNUM

​ Create a primary partition for the disk of the msdos partition table

**Warning: Use this command may cause data loss**

* --active, -a activate the partition
* --file=FILE, -f use file as partition content
* --type=HEX, -t specifies the partition type, 0x00 is to automatically detect the partition type (support FAT, exFAT, NTFS, EXT), 0x10 is to automatically detect and set as a hidden partition.
* --start=n, -s specify the start address (unit is sector)
* --length=n, -l specifies the length (unit is sector)

### parttool PARTITION COMMANDS

​ Modify the partition table (currently only supports mbr partition table)

**Warning: Use this command may cause data loss**

​ Each command is either a boolean option, in which case it must be followed with'+' or'-' (with no intervening space) to enable or disable that option, or else it takes a value in the form'command=value '.

Available options:

* boot (boolean) When enabled, this makes the selected partition be the active (bootable) partition on its disk, clearing the active flag on all other partitions. This command is limited to primary partitions.
* type (value) Change the type of an existing partition. The value must be a number in the range 0-0xFF (prefix with '0x' to enter it in hexadecimal).
* hidden (boolean) When enabled, this hides the selected partition by setting the hidden bit in its partition type code; when disabled, unhides the selected partition by clearing this bit. This is useful only when booting DOS or Windows and multiple primary FAT partitions exist in one disk. See also DOS/Windows.

### password USER clear-password

​ Define a user named user with password clear-password.

### password\_pbkdf2 USER hashed-password

​ Define a user named user with password hash hashed-password. Use grub-mkpasswd-pbkdf2 to generate password hashes.

### pcidump OPTIONS

​ Display the original dump of PCI configuration space

* -d [vendor]:[device] Select device by vendor and device ID
* -s [bus]:[slot][.func] select the device according to its position on the bus

### play FILE | TEMPO [PITCH1 DURATION1] [PITCH2 DURATION2]…

​ Use PC Speaker to play tunes

​ If the parameter is a file, the tune recorded by the file is played.

### ****pop\_env**** VARIABLE…

​ Pass the variables in the submenu to the previous menu

### probe OPTIONS DEVICE

​ Testing equipment information

* --set=VARIABLE, -s set the return value as a variable
* --driver, -d detect drive
* --partmap, -p detect partition table type
* --fs, -f detect file system type
* --fs-uuid, -u detect file system UUID
* --label, -l detect file system volume label
* --partuuid, -g detect partition UUID (GPT partition table)
* **--bootable, -b detect whether there is a bootable logo**
* **--quiet, -q do not display errors**

### ****rand**** [OPTIONS] VARIABLE

​ Generate pseudo-random numbers

* --from=n, -f set the lower bound of the random number
* --to=n, -t set random number upper bound

### rdmsr [OPTIONS] ADDR

​ Read a model-specific register at address ADDR.

​ Please note that on SMP systems, reading from a MSR that has a scope per hardware thread, implies that the value that is returned only applies to the particular cpu/core/thread that runs the command.

​ Also, if you specify a reserved or unimpl

​ Also, if you specify a reserved or unimplemented MSR address, it will cause a general protection exception (which is not currently being handled) and the system will reboot.

* -v VARIABLE save the value to a variable

### read [VARIABLE] ****[hide | asterisk]****

​ Read a line of user input

### read\_byte [OPTIONS] ADDR

​ Read 8-bit value from ADDR

* -v VARIABLE save the value to a variable

### read\_dword [OPTIONS] ADDR

​ Read 32-bit value from ADDR, the parameter is the same as "read\_byte"

### ****read\_file**** FILE VARIABLE…

​ Read the file and set the content of the file as a variable line by line

### read\_word [OPTIONS] ADDR

​ Read 16-bit value from ADDR, the parameter is the same as "read\_byte"

### reboot

​ Restart the computer

### regexp [OPTIONS]'REGEXP' “STRING”

​ Test whether the regular expression REGEXP matches the string STRING

​ The supported regular expressions are POSIX.2 extended regular expressions.

​ If the -set option is given, the first sub-expression that matches is stored in the variable var. The sub-expression is numbered from 1 in the opening bracket. The number defaults to 1.

* --set=[NUMBER:][VARIABLE], -s save the nth matching string to a variable

### ****reset [OPTIONS]****

​ Restart the computer (the efiusage is the same as "reboot" under non- platform, no parameters are supported)

* --shutdown, -s execute shutdown
* --warm, -w perform a warm start
* --cold, -c perform a cold start
* --fwui, -f return to the firmware setting user interface at the next startup

### save\_env [OPTIONS] VARIABLE…

​ Save variables from GRUB environment to environment block file, the parameters are the same as "load\_env"

### ****sbpolicy**** [OPTIONS]

​ Install a security policy that bypasses secure boot. Enabling this function can load unsigned EFI programs when secure boot is enabled.

**Warning: Use this command may cause security issues**

* --install, -i install security policy
* --uninstall, -u uninstall security policy
* --status, -s show security policy status

### search OPTIONS STRING

​ Search Disk

* --file, -f search by file
* --label, -l search by file system volume label
* --fs-uuid, -u search by file system UUID
* --part-label, -L search by partition label
* **--part-uuid, g Search by partition UUID (GPT)**
* **--disk-uuid, U Search by disk UUID (GPT)**
* --set=VARIABLE, -s save the first device found to a variable
* --no-floppy, -n do not detect floppy disks
* **--quiet, -q If there is no match, no error is displayed**
* --hint=HINT, -h Specifies to start searching from a device first, if it ends with a comma, it will also search for subpartitions
* --hint-ieee1275=HINT If running under the IEEE1275 environment, specify to start searching from a certain device first
* --hint-bios=HINT If running under the BIOS environment, specify to start searching from a certain device first
* --hint-baremetal=HINT If running in a baremetal environment, specify to start searching from a device
* --hint-efi=HINT If running under EFI environment, specify to start searching from a certain device first
* --hint-arc=HINT If running under ARC environment, specify to start searching from a certain device first

### serial [OPTIONS]

​ Configure serial port

### ****setenv**** [OPTIONS] EFI\_ENV VALUE

​ Write UEFI environment variables

**Warning: Use this command to modify UEFI environment variables**

**Warning: Use this command may cause security issues**

* --guid=GUID, -g set the GUID of the variable to be written, the default is a global variable
* --type=string/wstring/uint8/hex, -t specifies the variable type as string/wide string/8-bit unsigned integer/hexadecimal data, the default is hexadecimal data

### ****setkey**** the OPTIONS NEW\_KEY USA\_KEY

​ Map a key (USA\_KEY) of the American keyboard to another key (NEW\_KEY)

​ Supports up to 255 groups of mapping relationships at the same time.

* --reset, -r reset mapping
* --enable, -e enable key mapping
* --disable, -d disable key mapping
* --status, -s show current keyboard mapping

### setpci [OPTIONS] REGISTER[=VALUE[:MASK]]

​ Operating PCI devices

* -d [vendor]:[device] Select device by vendor and device ID
* -s [bus]:[slot][.func] select the device according to its position on the bus
* -v VARIABLE save data to variable

### setup\_var offset [setval]

​ Read/Write specific (byte) offset of setup variable.

**Warning: Use this command to modify UEFI environment variables**

**Warning: Use this command may cause security issues**

### sha1sum arg…

​ means "hashsum -hash sha1 arg …"

### sha256sum arg…

​ means "hashsum –hash sha256 arg …"

### sha512sum arg…

​ means "hashsum –hash sha512 arg …"

### ****shell**** [OPTIONS] CMDLINE

​ Start UEFI Shell

* --nostartup do not execute the default startup script
* --noconsoleout Do not display terminal output
* --noconsolein does not accept user input
* --delay=n Set the waiting time before executing the startup script
* --nomap don't display device mapping list
* --noversion do not display version information
* --startup execute the default startup script
* --nointerrupt prohibit interruption of execution
* --exit automatically exit after the command is executed
* --device=DEVICE specify the default Device Path

### sleep [OPTIONS] n

​ Wait for n seconds

​ If the countdown is over, it returns 0. If the countdown is interrupted by ESC, it returns to 1.

* --verbose, -v show countdown seconds
* --interruptible, -i allows the use of ESC to interrupt the countdown

### smbios [OPTIONS]

​ Retrieve SMBIOS information

* --type=TYPE, -t Match structures with the given type.
* --handle=HANDLE, -h Match structures with the given handle.
* --match=MATCH, -m Select a structure when several match.
* --get-byte=OFFSET, -b Get the byte's value at the given offset.
* --get-word=OFFSET, -w Get two bytes' value at the given offset.
* --get-dword=OFFSET, -d Get four bytes' value at the given offset.
* --get-qword=OFFSET, -q Get eight bytes' value at the given offset.
* --get-string=OFFSET, -s Get the string specified at the given offset.
* --get-uuid=OFFSET, -u Get the UUID's value at the given offset.
* --set=VARIABLE save data to variable

### ****stat**** [OPTIONS] FILE

​ Display file and file system information

* --set=VARIABLE, -s set the return value as a variable
* --size, -z show file size
* --human, -m display file size in readable form
* --offset, -o show the offset of the file on the disk
* --fs, -f display file system information
* --ram, -r Display memory size in MiB
* --contig, -c check if the file is continuous
* --quiet, -q do not display output

### ****strconv**** [OPTIONS] STRING

​ String UTF-8/GBK encoding conversion

* --gbk, -g UTF-8 -> GBK
* --utf8, -u GBK -> UTF-8 (default)
* --set=VARIABLE, -s set the return value as a variable

### submenu “TITLE” [OPTIONS] {MENU…}

​ Define the sub-menu, when this item is selected for execution, a new menu containing the menu items in braces will be displayed

​ The parameters are the same as "menuentry"

### ****submenu\_exit****

​ Exit from the current submenu

### terminfo [OPTIONS] [TERM]

​ Set terminal type

### test EXPRESSION

​ Calculation expression, if the result is true, it returns zero, otherwise it returns a non-zero state

​ Support the following expressions

* string1 ==string2

the strings are equal

* string1 !=string2

the strings are not equal

* string1 <string2

string1 is lexicographically less than string2

* string1 <=string2

string1 is lexicographically less or equal than string2

* string1 >string2

string1 is lexicographically greater than string2

* string1 >=string2

string1 is lexicographically greater or equal than string2

* integer1 -eqinteger2

integer1 is equal to integer2

* integer1 -geinteger2

integer1 is greater than or equal to integer2

* integer1 -gtinteger2

integer1 is greater than integer2

* integer1 -leinteger2

integer1 is less than or equal to integer2

* integer1 -ltinteger2

integer1 is less than integer2

* integer1 -neinteger2

integer1 is not equal to integer2

* prefixinteger1 -pgtprefixinteger2

integer1 is greater than integer2 after stripping off common non-numeric prefix.

* prefixinteger1 -pltprefixinteger2

integer1 is less than integer2 after stripping off common non-numeric prefix.

* file1 -ntfile2

file1 is newer than file2 (modification time). Optionally numeric bias may be directly appended to -ntin which case it is added to the first file modification time.

* file1 -otfile2

file1 is older than file2 (modification time). Optionally numeric bias may be directly appended to -otin which case it is added to the first file modification time.

* -d file

file exists and is a directory

* -e file

file exists

* -f file

file exists and is not a directory

* -s file

file exists and has a size greater than zero

* -n string

the length of string is nonzero

* string

string is equivalent to -n string

* -z string

the length of string is zero

* ( expression )

expression is true

* ! expression

expression is false

* expression1 -aexpression2

both expression1 and expression2 are true

* expression1 expression2

Both expression1 and expression2 are true. This syntax is not POSIX-compliant and is not recommended.

* expression1 -oexpression2

either expression1 or expression2 is true

### testspeed [OPTIONS] FILE

​ Test file reading speed

* --size=n, -s specify the size of each read

### ****tetris****

​ Tetris game. Please execute first terminal\_output console.

### tr [OPTIONS] [SET1] [SET2] [STRING]

​ Replace the character SET1 in the string STRING with SET2

​ If you enter two parameters, the input string must be specified by the --set option.

* --set=VARIABLE, -s save the return value to a variable
* --upcase, -U convert to uppercase
* --downcase, -D convert to lowercase

### true

​ Return TRUE (0) directly

​ is used in if or while statements.

### trust [OPTIONS] PUBKEY\_FILE

​ Add PUBKEY to the list of trusted keys

* --skip-sig, -s skip the signature check of the public key file

### ****unalias NAME****

​ Unset alias

### ****uuid4**** VARIABLE

​ Generate UUID string

### ****vboot**** harddisk=FILE floppy=FILE cdrom=FILE boot=harddisk/floppy/cdrom

​ To start vboot, specify the hard disk image/floppy disk image/CD image and the default boot device.

​ You must execute vbootinsmod to load vbootcore.mod first.

​ The hard disk needs to contain the vboot file.

​ The default path of vboot is /vboot/vboot, modify the variable vbootloader to specify the path.

### ****vbootinsmod**** FILE

​ Load the vboot core file vbootcore.mod

### verify\_detached [OPTIONS] FILE SIGNATURE\_FILE [PUBKEY\_FILE]

​ Verify detached signature.

​ The parameters are the same as "trust"

### version

​ Show GRUB version information

### ****vfat**** OPTIONS

​ UEFI virtual FAT disk operation

* --create, -c create virtual FAT disk
* --add=NAME FILE, -a Specifies to add the file FILE to the virtual disk, the file name in the virtual disk is NAME
* --mem, -m specify the file to be loaded into the memory, allowing writing to the file
* --install, -i make the virtual disk visible in UEFI
* --boot, -b start virtual disk
* --ls, -l list all files in the virtual disk
* --patch=FILE STRING, -p specifies the name of the file to be modified, the file must be in memory
* --offset=n, -o specify the modified offset
* --search=STRING, -s search string
* --count=n, -c specifies the number of searches

The search and replace strings support the following formats

| **format** | **Types of** | **Sample input** | **Sample output** |
| --- | --- | --- | --- |
| sSTRING starts with's' | Ordinary string | sHello | Hello |
| wSTRING starts with'w' | Wide character string | wWin32 | W\0i\0n\03\02\0 |
| STRING does not start with's' or'w' | Hexadecimal | 74657374 | test |

### ****vhd**** [OPTIONS] DEVICE FILE

​ Mount the vhd file as a virtual disk

* --delete, -d delete virtual disk
* --partitions -p simulate a hard disk with partitions

### videoinfo [WxH[xD]]

​ List available display modes

### vbeinfo [WxH[xD]]

​ Same as'videoinfo'

### ****videomode [OPTIONS] VARIABLE****

​ Get the current/available display mode and save it to a variable

* --list, -l list available display modes
* --current, -c Get the current display mode

### ****wimboot**** [OPTIONS] @:NAME:FILE

​ Start WIM file

* --gui, -g enable graphical startup information
* --rawbcd, -b disable BCD automatic modification (.exe changed to .efi)
* --rawwim, -w disable WIM automatic modification
* --index=n, -i specify the WIM volume number to start
* --pause, -p pause before starting
* --inject=WIN32\_PATH, -j specifies the injection folder, the default is \Windows\Syatem32

### ****wimtools OPTIONS FILE [WIN32\_PATH]****

​ WIM analysis tool

* --index=n, i specify WIM volume number
* --exist, -e check if the file exists
* --is64, -a Check whether the WIM internal system is 64-bit (x86\_64)

### ****write\_bytes**** ADDR VALUE…

​ Write a series of 8-bit values ​​to the address ADDR

**Warning: Use this command may cause security issues**

## Environment variable

GRUB supports Unix-like environment variables. Some of these variables have special meaning to GRUB.

### Special environment variables

#### color\_normal

The foreground and background colors of the terminal when "normal" is specified, separated by a slash, and the following colors are supported:

* black (black)
* blue (blue)
* green (green)
* cyan (青)
* red (red)
* magenta (magenta)
* brown (brown)
* light-gray (light gray)
* dark-gray (dark gray)
* light-blue (light blue)
* light-green (light green)
* light-cyan
* light-red (light red)
* light-magenta (light magenta)
* yellow (yellow)
* white (white)

The default value is "light-gray/black".

#### color\_highlight

The foreground and background colors of the terminal when "highlight" is specified, are the same as "color\_normal".

#### debug

This variable is used to display debugging information.

#### default

Set the menu item selected by default. The menu item can be a number (starting from 0) or a menu id.

#### enable\_progress\_indicator

Set whether to display the progress and speed when reading files. Value 0disables progress display, otherwise enable progress display.

Note: This feature conflicts with the graphics mode theme. If you use the theme, please disable the progress display and enable it only when needed.

export enable\_progress\_indicator=0

#### fallback

If the default item fails to start, select this menu item.

#### gfxmode

Set the resolution of the "gfxterm" terminal. You can specify multiple resolutions, separated by commas or semicolons, each mode must be in the form of "auto", "width x height" or "width x height x color depth". For example set gfxmode=1024x768,640x480,auto.

#### gfxpayload

Set the display mode when the Linux kernel starts. If it is set to "text", it is forced to enter the character mode, and set to "keep" to keep the display mode consistent with "gfxmode.

If there is a problem during startup, please try to set it to "text".

#### grub\_build\_date

GRUB build date.

#### grub\_cpu

The CPU type selected when GRUB was built, such as "i386", "x86\_64", etc.

#### grub\_detect\_floppies

Set whether to detect floppy drive under Legacy BIOS. The default value is 0, ignoring the floppy drive.

#### grub\_disable\_console

Set whether to disable press Center GRUB console.

#### grub\_disable\_edit

Set whether to prohibit Eediting menu items by pressing the key.

#### grub\_disable\_esc

Set whether to disable ESCkey return to the previous menu. The default value is 0, allowing to press ESC to return.

#### grub\_draw\_border

Set whether to draw the menu border. The value is 1, the menu border is drawn.

#### grub\_enable\_menu\_hotkey

Set whether to display the shortcut key of the menu on the menu item.

#### grub\_enable\_menu\_jump

Whether to enable the press A~Zjumps to the first letter corresponding to the next menu item.

#### grub\_frame\_speed

The frame rate of the animation theme, the unit is milliseconds/frame, and the recommended value is 110.

If this variable is set, the dynamic theme function will be enabled.

#### grub\_fs\_case\_sensitive

Set whether the file name is case sensitive. The value is 1, the file name is case sensitive.

#### grub\_normal\_menu\_title

Specify the text content of the menu title without a theme.

#### grub\_pkg\_version

GRUB major version number.

#### grub\_platform

The platform type selected when GRUB was built, such as "pc", "efi", "emu", etc.

#### grub\_prompt

GRUB command prompt, the default is "grub>".

#### grub\_sound\_speed

The speed of the buzzer sound, in milliseconds/note, the recommended value is 110.

If this variable is set, the buzzer playback function will be enabled.

#### grub\_sound\_select

Customize the up and down keys to select the menu sound effects, "freq1 freq2 freq3 …", where freq1, freq2, freq3… are frequencies and the unit is hertz.

The default value is "880 0 880 0 880 698 1046".

#### grub\_sound\_start

Custom loop music, like "freq1 freq2 freq3 …", where freq1, freq2, freq3… are frequencies and the unit is hertz.

The default value is "220 277 330 440 185 220 277 370 294 370 440 587 330 415 494 659".

#### grub\_uefi\_version

UEFI firmware version.

#### icondir

Specify the folder where the icon is located when the theme is displayed.

#### lang

Set the localized language code, such as "zh\_CN".

#### locale\_dir

Set the folder where the translation file is located, usually "/boot/grub/locale".

#### menu\_color\_highlight

Specify the foreground and background colors of the highlighted menu item.

#### menu\_color\_normal

Specify the foreground and background colors of non-highlighted menu items.

#### pager

If it is set to 1, it will be displayed in pages and paused for keyboard input.

#### prefix

Specify the default path for GRUB module/theme/menu loading. This variable must be set correctly, otherwise GRUB will not work properly.

#### root

Specify the root device. This variable must be set correctly, otherwise GRUB will not work properly.

#### theme

Set the folder where the theme is located.

#### timeout

Set the waiting time.

#### timeout\_style

Set the "timeout" timing method. If it is "countdown" or "hidden", the timing will be performed before the menu is displayed; if it is set to "menu", the timing will be performed after the menu is displayed.

### Variable settings

set List all variables

set variable=value Set the value of the variable

export variable Set as global variable

export variable=value Set the value of a global variable

unset variable Unset variable

## Equipment and files

### equipment

The syntax of the device is like

(device[,partmap-name1part-num1[,partmap-name2part-num2[,...]]])

Under BIOS and UEFI, the names of hard disks, floppy disks, and optical disks are "hd", "fd", "cd" plus a number (starting from 0), such as "hd0", "fd1", "cd0". If booting from CD under BIOS, the device name may be "cd".

partmap-nameAnd part-numrespectively represent the name of the partition table, partition numbers (starting from 1), for example, (hd0,msdos1)represents a first partition of a hard 0 MBR (msdos) partition table, (hd1,gpt3)a third partition the hard disk partition table of the GPT 1.

### file

GRUB supports file representation methods such as absolute paths and block lists.

#### Absolute path

Similar to Unix systems, GRUB uses a slash "/" as a folder separator.

##### Example

(hd1,msdos2)/boot/grub/grub.cfg

#### relative path

If the file path omits the device name, it will default to the file on the root device (root).

##### Example

set root=hd0,1

It is /efi/boot/bootx64.efiequivalent to (hd0,1)/efi/boot/bootx64.efi.

#### Block list

The form [offset]+length[,[offset]+length]can be used to represent files that do not exist in the file system, such as the master boot record of a disk.

##### Example

(hd0)0+100,200+1,300+300

This means GRUB reads blocks 0 to 99, block 200 and blocks 300 to 599. If the offset is omitted, the offset is assumed to be 0. You can blocklistsee a list of file block commands.

#### Memory file

It has the form mem:addr:size:num, you can read a certain section of memory as a file.

##### Example

mem:0x1234:size:567

#### Treat the device as a file

GRUB supports directly reading a disk or partition as a file

##### Example

(hd1) (hd0,msdos1) (memdisk) (http)

## theme

Note: The theme conflicts with the file reading progress function. If you have problems with the image display, disable the progress display.

export enable\_progress\_indicator=0

### Font

GRUB supports the PFF2 font format. You can use lsfontsa command to list the available fonts, use loadfontscommand to load fonts.

### image

GRUB supports pictures in bmp, jpg, jpeg, png, tga format.

### colour

The following colors are supported:

* #RRGGBB (Hexadecimal number, such as "#8899FF")
* #RGB (same as above)
* R,G,B (decimal number, such as "128,128,255")
* Lowercase "SVG 1.0 color name"

### coordinate

Coordinate positions in the following formats are supported:

* x (in pixels)
* p% (in units of percentage of the container)
* p%+x, p%-x (mix of the above two)

### Global attributes

Global attributes are specified in the following format:

* name1: value1
* name2: "value which may contain spaces"
* name3: #88F
* name4: p%+x

Global attribute list:

* title-text

Specify the title text.

* title-font

Specify the title font.

* title-color

Specify the title text color.

* desktop-image

Specify the desktop background image.

* desktop-image-scale-method

Specify the scaling mode of the desktop background image. Optional values ​​are "stretch", "crop", "padding", "fitwidth", "fitheight".

* desktop-image-h-align

Specify the horizontal alignment mode of the desktop background image. Optional values ​​are "left", "center", "right".

* desktop-image-v-align

Specify the vertical alignment mode of the desktop background image. Optional values ​​are "top", "center", "bottom".

* desktop-color

Specify the desktop background color.

* terminal-box

Specify the style picture of the command line terminal window, such as "terminal\_\*.png".

* terminal-border

Specify the border width of the command line terminal window.

* terminal-left

Specify the left coordinate of the command line terminal window.

* terminal-top

Specify the top coordinate of the command line terminal window.

* terminal-width

Specifies the width of the command line terminal window.

* terminal-height

Specify the height of the command line terminal window.

### Theme component

Create a component in the theme by adding a "+" sign in front of the component type:

+ label {

text = "GNU GRUB 2"

font = "Dos VGA"

color = "#8FF"

align = center

preferred\_size = (120, 80)

}

#### label

The component displays a line of text.

##### Attributes

* id

If it is "\_\_timeout\_\_" and there is no "text" attribute, the start countdown will be displayed.

* text

The text to be displayed. If the text begins with "@@" variable value is displayed, for example, is set to "@@ grub\_cpu" to display the variable grub\_cpuvalue.

* var

The name of the variable to be displayed.

* font

The font of the text.

* color

The color of the text.

* align

The horizontal alignment of the text in the component, the options are "left", "center" and "right".

* visible

Set to "false" to hide this component.

#### image

The component displays a picture.

##### Attributes

* file

The full path of the image.

#### progress\_bar

Display a horizontal countdown progress bar.

##### Attributes

* id

Set to "\_\_timeout\_\_" to display the countdown.

* fg\_color

Set the foreground color.

* bg\_color

Set the background color.

* border\_color

Set the border color.

* text\_color

Set the text color.

* bar\_style

Set the progress bar style picture, such as "progress\_frame\_\*.png".

* highlight\_style

Set the style picture of the highlight area of ​​the progress bar, such as "progress\_hl\_\*.png".

* highlight\_overlay

If set to "true", the highlighted edge image will cover the edge image of the progress bar frame. The default is "false".

* font

Set the font.

* text

Set the text displayed on the progress bar. If the value is "@TIMEOUT\_NOTIFICATION\_SHORT@", "@TIMEOUT\_NOTIFICATION\_MIDDLE@" or "@TIMEOUT\_NOTIFICATION\_LONG@", GRUB will automatically update the prompt information.

#### circular\_progress

A circular countdown indicator is displayed. The appearance is determined by the center image and scale image. The center image is at the center of the component. Around the circle, draw several scale images.

##### Attributes

* id

Set to "\_\_timeout\_\_" to display the countdown.

* center\_bitmap

Set the file name of the center image.

* tick\_bitmap

Set the file name of the scale image.

* num\_ticks

Set the number of graduations on the circle.

* ticks\_disappear

Set whether to disappear when the progress arrives, the option is "true" or "false" (default).

* start\_angle

Set the position of the first scale mark, the unit is 1/256 of the circle. Use "xxx deg" or "xxx \xc2\xb0" to set the position by angle.

#### boot\_menu

The GRUB boot menu is displayed.

##### Attributes

* item\_font

The font of the menu item title.

* selected\_item\_font

The font of the title of the selected menu item. The default is "inherit", inherit the item\_fontfont settings.

* item\_color

The color of the menu item title.

* selected\_item\_color

The color of the title of the selected menu item. The default is "inherit", inherited item\_colorcolor settings.

* icon\_width

The width of the menu item icon.

* icon\_height

The height of the menu item icon.

* item\_height

The height of the menu item.

* item\_padding

The space to be reserved on each side of the menu item content.

* item\_icon\_space

The space between the menu item icon and the title.

* item\_spacing

The space reserved between menu items.

* menu\_pixmap\_style

Start the style picture of the menu box, such as "menu\_\*.png".

* item\_pixmap\_style

The style picture of the menu item box.

* selected\_item\_pixmap\_style

Select the style picture of the menu item box.

* scrollbar

Whether to display the scroll bar, the options are "true" or "false".

* scrollbar\_frame

Scroll bar style picture, such as "scrollbar\_\*.png".

* scrollbar\_thumb

The style picture of the scroll bar slider, such as "scrollbar\_thumb\_\*.png".

* scrollbar\_thumb\_overlay

The options are "true" or "false". If it is "true", the side of the slider will cover the side of the scroll bar frame.

* scrollbar\_slice

The position of the scroll bar in the menu frame, the options are "west", "center", "east" (default). "West" will be drawn on the left (right-aligned), and "east" will be drawn on the right (left-aligned). "Center" will be drawn in the center. Note: If it is "center", if you draw a scroll bar, the width of the startup menu item will decrease the width of the scroll bar, and the scroll bar will be drawn to the right of the center picture. If the scroll bar is not drawn, the width of the menu item is the width of the center image.

* scrollbar\_left\_pad

The space to the left of the scroll bar. If scrollbar\_slicefor the "west", then this property has no effect.

* scrollbar\_right\_pad

The space to the right of the scroll bar. If scrollbar\_slicefor the "east", then this property has no effect.

* scrollbar\_top\_pad

The space above the scroll bar.

* scrollbar\_bottom\_pad

The space at the bottom of the scroll bar.

* visible

Set to "false" to hide the boot menu.

#### canvas

canvas Is a container, you can put any component inside it, and does not change the position of its sub-components.

#### hbox

hboxThe container arranges its subcomponents from left to right, and sets its preferred width for each subcomponent. The height of each subcomponent will be set to the highest preferred height among the subcomponents.

#### vbox

vboxThe container arranges its subcomponents from top to bottom, and sets its preferred height for each subcomponent. The width of each subcomponent will be set to the widest preferred width in the subcomponent.

#### animation

Show animation. Need to properly set the variable grub\_frame\_speedto enable animation.

export grub\_frame\_speed=110

Among them, 110 is the frame rate, and the unit is milliseconds per frame.

##### Example

+ animation {

dir\_name = "IMAGE\_DIR"

image\_format = png/jpg/jpeg/tga

frame\_number = n

left = p%

width = p%

top = p%

height = p%

size\_ratio = n

start\_x = n

start\_y = n

move\_speed = n

move\_direction = up/down/left/right

play\_once = pause/disappear

hit\_wall = pause/stop/disappear

bind\_menu = fixed\_position/follow\_single/foll\_variety/full\_screen

bind\_direction = left/right

}

##### Attributes

* dirname = "IMAGE\_DIR"

The folder where the animation sequence pictures are located, refers to the folder under the directory where theme.txt is located

* image\_format = png/jpg/jpeg/tga

Specify the extension of the animated picture, support png, jpg, jpeg, tga

* frame\_number = n

The total number of pictures in the animation sequence. The name of the picture file must be a number plus an extension, the number starts from 1, such as 1.jpg, 2.jpg, 3.jpg…

* size\_ratio = n

The ratio of the animation size to the container.

* start\_x/start\_y = n

The offset value of the x/y coordinates of the animation in the container. If the animation moves with the position of the selected item in the main menu, start\_y is invalid.

* move\_direction = up/down/left/right

Set the initial moving direction of the animation, which is only effective in random moving animations.

* move\_speed = n

Set the animation movement speed. The value is 0, the animation is played in place. The value cannot be negative.

* play\_once = pause/disappear

After setting the animation to play once, the last frame will be paused or disappear. If you don't set this option, it will play in a loop.

* hit\_wall = pause/stop/disappear

Set the behavior when the animation hits the container wall, pause-pause the animation, stop-stop moving, continue to play the animation, disappear-disappear.

* bind\_menu = fixed\_position/follow\_single/foll\_variety/full\_screen

Set the animation as a logo that changes with the menu options. If the parameter is not follow\_single, the animation sequence should be placed in each subdirectory of the dir\_name parameter directory, the subdirectory name is the same as the parameter of menuentry -class.

* + fixed\_position-play animation with the menu item in a fixed position
  + follow\_single-follow the menu item, display the same set of animations in different positions
  + follow\_variety-follow the menu item, play different animations in different positions
  + full\_screen-full screen display animation that changes with menu items
  + bind\_direction = left/right After enabling the animation that follows the menu item, set the position of the animation to the left or right of the left border of the main menu

#### General attributes

* left

Specifies the position of the left border of the component.

* top

Specify the top border position of the component.

* width

Specify the width of the component.

* height

Specify the height of the component.

## Lua

### Load Lua module

​ insmod lua

### Enter Lua interactive programming mode

​ Enter in the console lua

### Execute Lua script

​ lua /path/to/script.lua

### grub library

* grub.run ( stringcommand)

Execute the GRUB command, if the execution is successful, it returns zero, otherwise it returns a non-zero value.

* grub.script ( stringscript)

Execute a line of GRUB script, and return zero if the script is executed successfully.

* grub.getenv ( stringvariable)

Get the value of GRUB environment variable, if the environment variable exists, return the value of the variable, otherwise, return nil.

* grub.setenv ( stringvariable, stringvalue)

Set the value of the GRUB environment variable. The first parameter is the variable name, and the second parameter is the variable content.

* grub.exportenv ( stringvariable[, stringvalue])

Set the value of GRUB global environment variables. The first parameter is the variable name, and the second parameter is optional and is the variable content.

* grub.enum\_device ( function( stringdevice[, stringfs, stringuuid, stringlabel, stringsize]))

Enumerate GRUB's disk devices.

The parameter is a function to be executed by enumeration, which can obtain the device name, file system, UUID and volume label.

* grub.enum\_file ( function( stringfilename[, intisdir]))

Enumerate files and folders in a directory (including .and ..)

* userdatafile = grub.file\_open ( stringfilename[, stringflag])

Open the file and return a value of userdatatype file handle.

The first parameter is the file name. The second parameter is the open mode, if it is "w", it will open in writable mode.

* grub.file\_close ( userdatafile)

Close the file.

* grub.file\_seek ( userdatafile, integeroffset)

Set the position offset of the file handle for reading and writing, and the return value is the offset.

* grub.file\_read ( userdatafile, integerlength)

Read the file, the length is "length", and return the file content.

* grub.file\_write ( userdatafile, stringdata)

Write the string "data" to the file.

* grub.file\_getline ( userdatafile)

Read a line of text from the file, and the return value is that text.

* grub.file\_getsize ( userdatafile)

Returns the file size.

* grub.file\_getpos ( userdatafile)

Returns the current offset of the file.

* grub.file\_eof ( userdatafile)

Determine whether the file has been read to the end.

* grub.file\_exist ( stringfilename)

Determine whether the file exists.

* stringbuf, stringhex = grub.hexdump ( userdatafile, integerskip, integerlength)

Get the hexadecimal data at the specified location of the file.

* grub.add\_menu ( stringsource, stringtitle[, …])

Add a menu item, the content of the menu is "source" and the title is "title".

* grub.add\_icon\_menu ( stringicon, stringsource, stringtitle[, …])

Add menu items with icons (–class).

* grub.add\_hidden\_menu ( stringhotkey, stringsource, stringtitle[, …])

Add a hidden menu item, the hotkey is "hotkey".

* grub.clear\_menu ( nil)

Clear the menu.

* grub.cls ( nil)

Clear the screen.

* grub.setcolorstate ( integerstate)
* grub.refresh ( nil)

Perform grub\_refreshfunctions

* stringtext = grub.gettext ( stringsrc)

Translation string.

* integerrand = grub.random ( integerm)

Return a random number less than "m".

* grub.enum\_block ( function( stringblock)[, intpart\_start])

Enumerate the block list of the file in the form of sector+size. If there is part\_startargument, a sector number is calculated based on the starting sector of the partition.

### ini library

* userdatainifile = ini.load ( stringfilename)

Load the ini configuration file.

* ini.free ( userdatainifile)

Release the ini configuration file.

* stringini.get ( userdatainifile, [ stringsection,] stringkey)

Read configuration items from ini.

### input library

* integerascii\_code, integerscan\_code = input.getkey ( nil)

Wait for the user to press the key, return ASCII code and scan code.

* integerascii\_code, integerscan\_code = input.getkey\_noblock ( nil)

Return the ASCII code and scan code (used in the loop).

* stringline = input.read ( nil)

Wait for the user to enter a line of string.

### video library

* video.swap\_buffers ( nil)
* video.fill\_rect ( tablecolor{ integerr, integerg, integerb, integera}, integerx, integery, integerw, integerh)

Draw a rectangle at the specified position.

* video.draw\_string ( stringtext, stringfont, tablecolor{ integerr, integerg, integerb, integera}, integerx, integery)

Display the string at the specified position.

* stringvideo\_mode = video.info ( nil)

Get a list of image modes.

* video.draw\_pixel ( tablecolor{ integerr, integerg, integerb, integera}, integerx, integery)

Draw a pixel at the specified position.

* integerx, integery = video.get\_info ( nil)

Get the width and height of the current display mode.

* userdatabitmap = video.bitmap\_load ( stringfilename)

Load image files (support bmp, jpg, jpeg, png, tga).

* video.bitmap\_close ( userdatabitmap)

Close the image file.

* integerx, integery = video.bitmap\_info ( userdatabitmap)

Get the width and height of the image.

* video.bitmap\_blit ( userdatabitmap, integerx, integery, integeroffset\_x, integeroffset\_y, integerw, integerh)

Display the image at the specified location.

* userdatascaled\_bitmap = video.bitmap\_rescale ( userdatabitmap, integerw, integerh)

Zoom the specified image.

### gbk function library

* stringgbk\_str = gbk.fromutf8 ( stringutf8\_str)

Convert UTF-8 string to GBK encoded string.

* stringutf8\_str = gbk.toutf8 ( stringgbk\_str)

Convert GBK encoded string to UTF-8 string.

* stringutf8\_simp = gbk.tosimp ( stringutf8\_trad)

Convert UTF-8 traditional Chinese character string to UTF-8 simplified Chinese character string.

### disk function library

* userdatadev = disk.open ( stringdiskname)

Open the disk, the return value userdatatype of disk handle.

* disk.close ( userdatadev)

Turn off the disk.

* stringbuf = disk.read ( userdatadev, integersector, integeroffset, integerlength)

Read the data of the specified sector/offset of the disk and return a character string.

* disk.write ( userdatadev, integersector, integeroffset, integerlength, stringbuf)

Write to disk.

* stringpartmap = disk.partmap ( userdatadev)

Get the name of the partition table.

* stringdriver = disk.driver ( userdatadev)

Get the disk drive name.

* stringfs = disk.fs ( userdatadev)

Get the file system name.

* stringuuid = disk.fsuuid ( userdatadev)

Obtain the file system UUID.

* stringlabel = disk.label ( userdatadev)

Obtain the disk volume label.

* stringsize = disk.size ( userdatadev[, flag])

Get the disk size.

* booleanboot = disk.bootable ( userdatadev)

Determine whether the partition has a bootable logo.

### fatfs library

* fat.mount
* fat.umount
* fat.disk\_status
* fat.get\_label
* fat.set\_label
* fat.mkdir
* fat.rename
* fat.unlink
* fat.open
* fat.close
* fat.read
* fat.write
* fat.lseek
* fat.tell
* fat.eof
* fat.size
* fat.truncate

### memrw library

* integervalue = memrw.read\_byte ( integeraddr)

Read one byte of data from the memory address.

* integervalue = memrw.read\_word ( integeraddr)

Read double-byte data from the memory address.

* integervalue = memrw.read\_dword ( integeraddr)

Read double word data from memory address.

* memrw.write\_byte ( integeraddr, integervalue)

Write a byte of data to the memory address.

* memrw.write\_word ( integeraddr, integervalue)

Write double-byte data to the memory address.

* memrw.write\_dword ( integeraddr, integervalue)

Write double word data to memory address.

## FatFs

​ FAT/exFAT file read and write function

### Load FatFs module

​ insmod fatfs

### Path format

​ FatFs supports up to 10 partitions at the same time.

​ The user can mount up to 9 partitions, the drive letters are 1:, 2:, 3:, …, 9:. Drive letter 0: reserved for the system.

​ The path is expressed as X:/path/to/file, such as 2:/EFI/BOOT/bootx64.efi, and is not case sensitive.

### Mount partition

​ mount PARTITION NUM[1-9]

​ Example:

mount (hd0,1) 2The (hd0,1) mounted to the partition 2:

### Unmount partition

​ umount NUM[1-9]

​ Example:

umount 2Uninstall 2:

### View the partition mounting situation

​ mount status

### Create folder

​ mkdir PATH

​ Example:

​ mkdir 1:/foo/bar/new\_dir

### Copy files

​ cp SRC\_FILE DST\_FILE

​ The source file can be either GRUB file path or FatFs file path

​ Example:

​ cp 1:/foo/bar/sys.vhd 2:/boot.vhd

​ cp (hd0,2)/foo/bar/sys.vhd 2:/boot.vhd

### Rename file/folder

​ rename SRC\_FILE DST\_FILE

​ The target file must be the same disk as the source file, which is equivalent to moving the file under the same disk.

​ Example:

​ rename 1:/foo/bar.tar.gz 1:/abc.gz

### Delete files/folders

​ rm FILE

​ Cannot delete non-empty folders.

​ Example:

​ rm 1:/foo/bar.txt

### Move files

​ mv SRC\_FILE DST\_FILE

If the destination file with the source files in the same disk, it's the same renameoperation.

​ Example:

​ mv 1:/foo/bar.tar.gz 2:/abc.gz

### Create file/modify file access time

​ touch FILE [YEAR MONTH DAY HOUR MINUTE SECOND]

​ If no time is specified, the file timestamp will be modified to the current time. If the file does not exist, an empty file is created.

​ Example:

​ touch 1:/foo/bar.txt

​ touch 1:/foo/bar.txt 2000 1 1

### Modify file

​ write\_file FILE STRING [OFFSET]

​ Write a character string to the file at offset OFFSET bytes. If the file size is insufficient, the file size is automatically expanded.

​ Example:

​ write\_file 1:/foo/bar.txt "The quick brown fox jumps over the lazy dog"

​ write\_file 1:/foo/bar.txt "I am not a journalist, but I have seen too much. " 0x786