ubuntu manuals

Menu

noble (8) sfdisk.8.gz

Provided by: fdisk_2.39.3-9ubuntu6_amd64

NAME

sfdisk - display or manipulate a disk partition table

SYNOPSIS

sfdisk [options] device [**-N** partition-number]

sfdisk [options] command

DESCRIPTION

sfdisk is a script-oriented tool for partitioning any block device. It runs in mode if executed on a terminal (stdin refers to a terminal).

Since version 2.26 **sfdisk** supports MBR (DOS), GPT, SUN and SGI disk labels, bu provides any functionality for CHS (Cylinder-Head-Sector) addressing. CHS has important for Linux, and this addressing concept does not make any sense for n

sfdisk protects the first disk sector when create a new disk label. The option **always** disables this protection. Note that **fdisk**(8) and **cfdisk**(8) completely e area by default.

sfdisk (since version 2.26) **aligns the start and end of partitions** to block-de limits when relative sizes are specified, when the default values are used or multiplicative suffixes (e.g., MiB) are used for sizes. It is possible that pa will be optimized (reduced or enlarged) due to alignment if the start offset i exactly in sectors and partition size relative or by multiplicative suffixes.

The recommended way is not to specify start offsets at all and specify partiti MiB, GiB (or so). In this case **sfdisk** aligns all partitions to block-device I/ when I/O limits are too small then to megabyte boundary to keep disk layout po this default behaviour is unwanted (usually for very small partitions) then sp offsets and sizes in sectors. In this case **sfdisk** entirely follows specified n without any optimization.

sfdisk does not create the standard system partitions for SGI and SUN disk lab **fdisk**(8) does. It is necessary to explicitly create all partitions including w system partitions.

sfdisk uses **BLKRRPART** (reread partition table) ioctl to make sure that the dev used by system or other tools (see also **--no-reread**). It’s possible that this another **sfdisk** activity races with **systemd-udevd**(8). The recommended way how t possible collisions is to use **--lock** option. The exclusive lock will cause **sys** to skip the event handling on the device.

The **sfdisk** prompt is only a hint for users and a displayed partition number do that the same partition table entry will be created (if **-N** not specified), esp tables with gaps.

COMMANDS

The commands are mutually exclusive.

[**-N** partition-number] device
The default **sfdisk** command is to read the specification for the desired pa
of device from standard input, and then create a partition table according
specification. See below for the description of the input format. If stand
a terminal, then **sfdisk** starts an interactive session.

If the option **-N** is specified, then the changes are applied to the partiti
by partition-number. The unspecified fields of the partition are not modif

Note that it’s possible to address an unused partition with **-N**. For exampl
always contains 4 partitions, but the number of used partitions may be sma
this case **sfdisk** follows the default values from the partition table and d
built-in defaults for the unused partition given with **-N**. See also **--appen**

-A, --activate device [partition-number...]
Switch on the bootable flag for the specified partitions and switch off th
flag on all unspecified partitions. The special placeholder '-' may be use
the partition numbers to switch off the bootable flag on all partitions.

The activation command is supported for MBR and PMBR only. If a GPT label
then **sfdisk** prints warning and automatically enters PMBR.

If no partition-number is specified, then list the partitions with an enab

--backup-pt-sectors device
Back up the current partition table sectors in binary format and exit. See
UP THE PARTITION TABLE section.

--delete device [partition-number...]
Delete all or the specified partitions.

-d, --dump device
Dump the partitions of a device in a format that is usable as input to **sfd**
BACKING UP THE PARTITION TABLE section.

-g, --show-geometry [device...]
List the geometry of all or the specified devices. For backward compatibil
deprecated option **--show-pt-geometry** have the same meaning as this one.

-J, --json device
Dump the partitions of a device in JSON format. Note that **sfdisk** is not ab

JSON as input format.

-l, --list [device...]
List the partitions of all or the specified devices. This command can be u
with **--verify**.

-F, --list-free [device...]
List the free unpartitioned areas on all or the specified devices.

--part-attrs device partition-number [attributes]
Change the GPT partition attribute bits. If attributes is not specified, t
the current partition settings. The attributes argument is a comma- or spa
list of bits numbers or bit names. For example, the string "RequiredPartit
sets three bits. The currently supported attribute bits are:

Bit 0 (RequiredPartition)
If this bit is set, the partition is required for the platform to func
creator of the partition indicates that deletion or modification of th
can result in loss of platform features or failure for the platform to
operate. The system cannot function normally if this partition is remo
should be considered part of the hardware of the system.

Bit 1 (NoBlockIOProtocol)
EFI firmware should ignore the content of the partition and not try to
it.

Bit 2 (LegacyBIOSBootable)
The partition may be bootable by legacy BIOS firmware.

Bits 3-47
Undefined and must be zero. Reserved for expansion by future versions
specification.

Bits 48-63
Reserved for GUID specific use. The use of these bits will vary depend
partition type. For example Microsoft uses bit 60 to indicate read-onl
shadow copy of another partition, 62 for hidden partitions and 63 to d
automount.

--part-label device partition-number [label]
Change the GPT partition name (label). If label is not specified, then pri
current partition label.

--part-type device partition-number [type]
Change the partition type. If type is not specified, then print the curren
type.

The type argument is hexadecimal for MBR, GUID for GPT, type alias (e.g. "
type shortcut (e.g. 'L'). For backward compatibility the options **-c** and --
same meaning as this one.

--part-uuid device partition-number [uuid]
Change the GPT partition UUID. If uuid is not specified, then print the cu
partition UUID.

--disk-id device [id]
Change the disk identifier. If id is not specified, then print the current
The identifier is UUID for GPT or unsigned integer for MBR.

-r, --reorder device
Renumber the partitions, ordering them by their start offset.

-s, --show-size [device...]
List the sizes of all or the specified devices in units of 1024 byte size.
command is DEPRECATED in favour of **blockdev**(8).

-T, --list-types
Print all supported types for the current disk label or the label specifie
--label.

-V, --verify [device...]
Test whether the partition table and partitions seem correct.

--relocate oper device
Relocate partition table header. This command is currently supported for G
only. The argument oper can be:

gpt-bak-std
Move GPT backup header to the standard location at the end of the devi

gpt-bak-mini
Move GPT backup header behind the last partition. Note that UEFI stand
the backup header at the end of the device and partitioning tools can
automatically relocate the header to follow the standard.

OPTIONS

-a, --append
Don’t create a new partition table, but only append the specified partitio

Note that unused partition maybe be re-used in this case although it is no partition in the partition table. See also **-N** to specify entry in the part

-b, --backup
Back up the current partition table sectors before starting the partitioni default backup file name is ~/sfdisk-<device>-<offset>.bak; to use another option **-O, --backup-file**. See section **BACKING UP THE PARTITION TABLE** for m

--color[=when]
Colorize the output. The optional argument when can be **auto**, **never** or **alwa** when argument is omitted, it defaults to **auto**. The colors can be disabled; current built-in default see the **--help** output. See also the **COLORS** sectio

-f, --force
Disable all consistency checking.

--Linux
Deprecated and ignored option. Partitioning that is compatible with Linux modern operating systems) is the default.

--lock[=mode]
Use exclusive BSD lock for device or file it operates. The optional argume be **yes**, **no** (or 1 and 0) or **nonblock**. If the mode argument is omitted, it d **yes**. This option overwrites environment variable **$LOCK_BLOCK_DEVICE**. The d not to use any lock at all, but it’s recommended to avoid collisions with **systemd-udevd**(8) or other tools.

-n, --no-act
Do everything except writing to the device.

--no-reread
Do not check through the re-read-partition-table ioctl whether the device

--no-tell-kernel
Don’t tell the kernel about partition changes. This option is recommended with **--no-reread** to modify a partition on used disk. The modified partitio be used (e.g., mounted).

-O, --backup-file path
Override the default backup file name. Note that the device name and offse
appended to the file name.

--move-data[=path]
Move data after partition relocation, for example when moving the beginnin
partition to another place on the disk. The size of the partition has to r
same, the new and old location may overlap. This option requires option **-N**
be processed on one specific partition only.

The optional path specifies log file name. The log file contains informati
read/write operations on the partition data. The word "@default" as a path
sfdisk to use ~/sfdisk-<devname>.move for the log. The log is optional sin

Note that this operation is risky and not atomic. **Don't forget to backup y**

See also **--move-use-fsync**.

In the example below, the first command creates a 100MiB free area before
partition and moves the data it contains (e.g., a filesystem), the next co
creates a new partition from the free space (at offset 2048), and the last
reorders partitions to match disk order (the original sdc1 will become sdc

```
echo '+100M,' | sfdisk --move-data /dev/sdc -N 1

echo '2048,' | sfdisk /dev/sdc --append

sfdisk /dev/sdc --reorder
```

--move-use-fsync
Use the **fsync**(2) system call after each write when moving data to a new lo
--move-data.

-o, --output list
Specify which output columns to print. Use **--help** to get a list of all sup
columns.

The default list of columns may be extended if list is specified in the fo
(e.g., **-o +UUID**).

-q, --quiet
Suppress extra info messages.

-u, --unit S
Deprecated option. Only the sector unit is supported. This option is not s
when using the **--show-size** command.

-X, --label type
Specify the disk label type (e.g., **dos**, **gpt**, ...). If this option is not g
sfdisk defaults to the existing label, but if there is no label on the dev
then the type defaults to **dos**. The default or the current label may be ove
the "label: <name>" script header line. The option **--label** does not force
create empty disk label (see the **EMPTY DISK LABEL** section below).

-Y, --label-nested type
Force editing of a nested disk label. The primary disk label has to exist
This option allows editing for example a hybrid/protective MBR on devices

-w, --wipe when
Wipe filesystem, RAID and partition-table signatures from the device, in o
avoid possible collisions. The argument when can be **auto**, **never** or **always**.
option is not given, the default is **auto**, in which case signatures are wip
in interactive mode; except the old partition-table signatures which are a
before create a new partition-table if the argument when is not **never**. The
also does not wipe the first sector (boot sector), it is necessary to use
mode to wipe this area. In all cases detected signatures are reported by w
messages before a new partition table is created. See also the **wipefs**(8) c

-W, --wipe-partitions when
Wipe filesystem, RAID and partition-table signatures from a newly created
in order to avoid possible collisions. The argument when can be **auto**, **neve**
When this option is not given, the default is **auto**, in which case signatur
only when in interactive mode and after confirmation by user. In all cases
signatures are reported by warning messages after a new partition is creat
wipefs(8) command.

-v, --version
Display version information and exit.

-h, --help
Display help text and exit.

INPUT FORMATS

sfdisk supports two input formats and generic header lines.

Header lines

The optional header lines specify generic information that apply to the partit
The header-line format is:

<name>: <value>

The currently recognized headers are:

unit
Specify the partitioning unit. The only supported unit is **sectors**.

label
Specify the partition table type. For example **dos** or **gpt**.

label-id
Specify the partition table identifier. It should be a hexadecimal number
prefix) for MBR and a UUID for GPT.

first-lba
Specify the first usable sector for GPT partitions. This header is ignored
script and device sector size differ. In this case **sfdisk** uses label speci

last-lba
Specify the last usable sector for GPT partitions. This header is ignored
script and device sector size differ. In this case **sfdisk** uses label speci

table-length
Specify the maximal number of GPT partitions.

grain
Specify minimal size in bytes used to calculate partitions alignment. The
1MiB and it’s strongly recommended to use the default. Do not modify this
you’re not sure.

sector-size
Specify sector size. **sfdisk** always uses device sector size. Since version
recalculates sizes from dump if the script and device sector size differ.

Note that it is only possible to use header lines before the first partition i
in the input.

Unnamed-fields format

start size type bootable

where each line fills one partition descriptor.

Fields are separated by whitespace, comma (recommended) or semicolon possibly whitespace; initial and trailing whitespace is ignored. Numbers can be octal, hexadecimal; decimal is the default. When a field is absent, empty or specifie default value is used. But when the **-N** option (change a single partition) is g default for each field is its previous value.

The default value of start is the first non-assigned sector aligned according I/O limits. The default start offset for the first partition is 1 MiB. If the followed by the multiplicative suffixes (KiB, MiB, GiB, TiB, PiB, EiB, ZiB and the number is interpreted as offset in bytes. Since v2.38 when the **-N** option (single partition) is given, a '+' can be used to enlarge partition by move sta partition if there is a free space before the partition.

The default value of size indicates "as much as possible"; i.e., until the nex or end-of-device. A numerical argument is by default interpreted as a number o however if the size is followed by one of the multiplicative suffixes (KiB, Mi PiB, EiB, ZiB and YiB) then the number is interpreted as the size of the parti bytes and it is then aligned according to the device I/O limits. A '+' can be of a number to enlarge the partition as much as possible. Note '+' is equivale default behaviour for a new partition; existing partitions will be resized as

The partition type is given in hex for MBR (DOS) where 0x prefix is optional; string for GPT; a shortcut or an alias. It’s recommended to use two letters fo codes to avoid collision between deprecated shortcut 'E' and '0E' MBR hex code backward compatibility **sfdisk** tries to interpret type as a shortcut as a first in partitioning scripts although on other places (e.g. **--part-type** command) it shortcuts as the last possibility.

Since v2.36 libfdisk supports partition type aliases as extension to shortcuts is a simple human readable word (e.g. "linux").

Since v2.37 libfdisk supports partition type names on input, ignoring the case characters and all non-alphanumeric and non-digit characters in the name (e.g. x86" is the same as "linux usr-x86").

Supported shortcuts and aliases:

L - alias 'linux'

Linux; means 83 for MBR and 0FC63DAF-8483-4772-8E79-3D69D8477DE4 for GPT.

S - alias 'swap'
swap area; means 82 for MBR and 0657FD6D-A4AB-43C4-84E5-0933C84B4F4F for G

Ex - alias 'extended'
MBR extended partition; means 05 for MBR. The original shortcut 'E' is dep
to collision with 0x0E MBR partition type.

H - alias 'home'
home partition; means 933AC7E1-2EB4-4F13-B844-0E14E2AEF915 for GPT

U - alias 'uefi'
EFI System partition, means EF for MBR and C12A7328-F81F-11D2-BA4B-00A0C93
GPT

R - alias 'raid'
Linux RAID; means FD for MBR and A19D880F-05FC-4D3B-A006-743F0F84911E for

V - alias 'lvm'
LVM; means 8E for MBR and E6D6D379-F507-44C2-A23C-238F2A3DF928 for GPT

The default type value is linux.

The shortcut 'X' for Linux extended partition (85) is deprecated in favour of

bootable is specified as [*|-], with as default not-bootable. The value of thi
irrelevant for Linux - when Linux runs it has been booted already - but it mig
role for certain boot loaders and for other operating systems.

Named-fields format
This format is more readable, robust, extensible and allows specifying additio
information (e.g., a UUID). It is recommended to use this format to keep your
readable.

[device :] name[=value], ...

The device field is optional. **sfdisk** extracts the partition number from the de
It allows specifying the partitions in random order. This functionality is mos
--dump. Don’t use it if you are not sure.

The value can be between quotation marks (e.g., name="This is partition name")
start= and **size=** support '+' and '-' in the same way as **Unnamed-fields format.**

The currently supported fields are:

start=number
The first non-assigned sector aligned according to device I/O limits. The start offset for the first partition is 1 MiB. If the offset is followed b multiplicative suffixes (KiB, MiB, GiB, TiB, PiB, EiB, ZiB and YiB), then is interpreted as offset in bytes.

size=number
Specify the partition size in sectors. The number may be followed by the multiplicative suffixes (KiB, MiB, GiB, TiB, PiB, EiB, ZiB and YiB), then interpreted as size in bytes and the size is aligned according to device I

bootable
Mark the partition as bootable.

attrs=string
Partition attributes, usually GPT partition attribute bits. See **--part-att** details about the GPT-bits string format.

uuid=string
GPT partition UUID.

name=string
GPT partition name.

type=code
A hexadecimal number (without 0x) for an MBR partition, a GUID for a GPT p shortcut as for unnamed-fields format or a type name (e.g. type="Linux /us See above the section about the unnamed-fields format for more details. Fo compatibility the **Id=** field has the same meaning.

EMPTY DISK LABEL

sfdisk does not create partition table without partitions by default. The line partitions are expected in the script by default. The empty partition table ha explicitly requested by "label: <name>" script header line without any partiti For example:

```
echo 'label: gpt' | sfdisk /dev/sdb
```

creates empty GPT partition table. Note that the **--append** disables this featur

BACKING UP THE PARTITION TABLE

It is recommended to save the layout of your devices. **sfdisk** supports two ways

Dump in sfdisk compatible format

Use the **--dump** command to save a description of the device layout to a text fi format is suitable for later **sfdisk** input. For example:

```
sfdisk --dump /dev/sda > sda.dump
```

This can later be restored by:

```
sfdisk /dev/sda < sda.dump
```

Note that sfdisk completely restores partition types and partition UUIDs. This potentially become problematic if you duplicate the same layout to different d may result in duplicate UUIDs within your system.

Full binary backup

If you want to do a full binary backup of all sectors where the partition tabl then use the **--backup-pt-sectors** command. It writes the sectors to ~/sfdisk-<device>-<offset>.bak files. The default name of the backup file can with the **--backup-file** option. The backup files contain only raw data from the example:

```
sfdisk --backup-pt-sectors /dev/sda
```

The GPT header can later be restored by:

```
dd if=~/sfdisk-sda-0x00000200.bak of=/dev/sda seek=$((0x00000200)) bs=1
conv=notrunc
```

It’s also possible to use the **--backup** option to create the same backup immedi startup for other **sfdisk** commands. For example, backup partition table before partitions from partition table:

```
sfdisk --backup --delete /dev/sda
```

The same concept of backup files is used by **wipefs**(8).

Note that **sfdisk** since version 2.26 no longer provides the **-I** option to restor
dd(1) provides all necessary functionality.

COLORS

The output colorization is implemented by **terminal-colors.d**(5) functionality.
coloring can be disabled by an empty file

/etc/terminal-colors.d/sfdisk.disable

for the **sfdisk** command or for all tools by

/etc/terminal-colors.d/disable

The user-specific $XDG_CONFIG_HOME/terminal-colors.d or $HOME/.config/terminal
overrides the global setting.

Note that the output colorization may be enabled by default, and in this case
terminal-colors.d directories do not have to exist yet.

The logical color names supported by **sfdisk** are:

header
The header of the output tables.

warn
The warning messages.

welcome
The welcome message.

ENVIRONMENT

SFDISK_DEBUG=all
enables **sfdisk** debug output.

LIBFDISK_DEBUG=all
enables libfdisk debug output.

LIBBLKID_DEBUG=all

enables libblkid debug output.

LIBSMARTCOLS_DEBUG=all
enables libsmartcols debug output.

LOCK_BLOCK_DEVICE=<mode>
use exclusive BSD lock. The mode is "1" or "0". See **--lock** for more detail

NOTES

Since version 2.26 **sfdisk** no longer provides the **-R** or **--re-read** option to for kernel to reread the partition table. Use **blockdev --rereadpt** instead.

Since version 2.26 **sfdisk** does not provide the **--DOS**, **--IBM**, **--DOS-extended**, **--show-extended**, **--cylinders**, **--heads**, **--sectors**, **--inside-outer**, **--not-inside** options.

EXAMPLES

sfdisk --list --label-nested=mbr /dev/sda
Print protective MBR on device with GPT disk label.

echo -e ',10M,L\n,10M,L\n,+,\n' | sfdisk /dev/sdc
Create three Linux partitions, with the default start, the size of the fir
partitions is 10MiB, and the last partition fills all available space on t

echo -e 'size=10M, type=L\n size=10M, type=L\n size=+\n' | sfdisk /dev/sdc
The same as the previous example, but in named-fields format.

echo -e 'type=swap' | sfdisk -N 3 /dev/sdc
Set type of the 3rd partition to 'swap'.

sfdisk --part-type /dev/sdc 3 swap
The same as the previous example, but without script use.

sfdisk --delete /dev/sdc 2
Delete 2nd partition.

echo "+,+" | sfdisk -N 3 --move-data /dev/sdc
Enlarge 3rd partition in both directions, move start to use free space bef

```
partition and enlarge the size to use all free space after to the partitio
partition data too.
```

AUTHORS

```
Karel Zak <kzak@redhat.com>

The current sfdisk implementation is based on the original sfdisk from Andries
```

SEE ALSO

```
fdisk(8), cfdisk(8), parted(8), partprobe(8), partx(8)
```

REPORTING BUGS

```
For bug reports, use the issue tracker at https://github.com/util-linux/util-l
```

AVAILABILITY

```
The sfdisk command is part of the util-linux package which can be downloaded f
Kernel Archive <https://www.kernel.org/pub/linux/utils/util-linux/>.
```