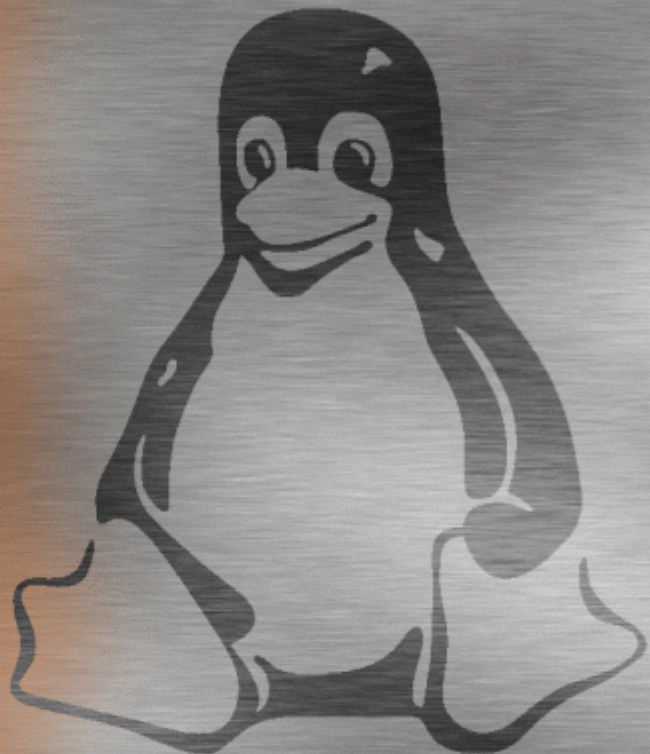


```

detected 501.143 MHz processor.
Console: colour dummy device 80x25
Calibrating delay loop... 999.42 BogoMIPS
Memory: 126420k/131008k available (1557k kernel c
Entry cache hash table entries: 16384 (order: 5,
Inode cache hash table entries: 8192 (order: 4, 65
Mount cache hash table entries: 512 (order: 0, 409
Buffer-cache hash table entries: 4096 (order: 2,
Page-cache hash table entries: 32768 (order: 5,
CPU: L1 I-Cache: 32K (32 bytes/line), D-cache 32K
CPU: AMD-K6(tm) 3D processor stepping 0c
checking 'hlt' instruction... OK.
POSIX conformance testing by UNIFIX
PCI: PCI BIOS revision 2.10 entry at 0xf240, la
PCI: Using configuration type 1
PCI: Probing PCI hardware
PCI: Using IRQ router VIA [1106/0586] at 00:07.0
Activating ISA DMA hang workarounds.
lsaprg: Scanning for PnP cards...
lsaprg: Card 'ADM ISDN-Controller FRITZ!Card'
lsaprg: 1 Plug & Play card detected total
Linux NET4.0 for Linux 2.4
Based upon Swansea University Computer Society NET3.03
Initialising MT netlink socket
iprm: BIOS version 1.2 Flags 0x07 (Driver version 1.16)
Starting kswapd
Installing knfsd (copyright (C) 1994 okir@sonad.meb
parport0: PC-style at 0x378 (0x778) [PC8FP(,...)]
parport0: irq 7 detected
vramfb: framebuffer at 0xd6000000, mapped to 0xc06800
vramfb: mode is 1024x768x32, linelen=10496, pages=
vramfb: protected mode interface info: 0xc000:03c5
vramfb: scrolling: redraw
vramfb: directcolor: size=8:8:8:8, shift=24:16:8:0
Console: switching to colour frame buffer device 7
fb0: VESA VGA frame buffer device
pty: 256 Unix98 pty's configured
Serial driver version 5.05c (2001-07-08) with MA
ttyS00 at 0x03f8 (irq = 4) is a 16550A
ttyS01 at 0x02f8 (irq = 3) is a 16550A
Floppy drive(s): fd0 is 1.44M
FDC 0 is a post-1991 82077
loop: loaded (max 8 devices)
nl39too Fast Ethernet driver 0.9.26
PCI: Found IRQ 10 for device 00:09.0
eth0: Realtek RTL8139 Fast Ethernet at 0xc00e0000, 00:3
Uniform Multi-Platform E-IDE driver Revision: 7.00beta
ide: Assuming 33MHz system bus speed for PIO modes; c
idebus=xxx
VF_IDE: IDE controller at PCI slot 00:07.1
VF_IDE: chipset revision 6
VF_IDE: not 100% native mode: will probe irqs later
VF_IDE: VIA vt82c586b (rev 47) IDE UDMA33 controller
hda0: BM-CMA at 0xc000-0xc007, BIOS settings: hda
hda1: BM-CMA at 0xc008-0xc00f, BIOS settings: hd
hda: SAMSUNG SV4003N, ATA DISK drive
hdb: PCRM1208, ATAPI CD/DVD-ROM drive
hbk: qswse c03722e0, I/O limit 4095Mb (mask 0xff
hdd: LTN301, ATAPI CD/DVD-ROM drive

```



Pripájanie diskových oddielov

Tomáš Hanzel

Close The Windows, Open The Source

Úvod

Ak používate dual-boot s windows-om, určite budete chcieť zdieľať súbory medzi operačnými systémami.. Našťastie, Linux poskytuje plnú podporu čítania/zápisu pre [FAT32](#) a staršie systémy súborou FAT. Pre NTFS podporu čítania/zápisu je taktiež možná v jadre možnosť *experimental write support*.

Niekoľko flash jednotiek (vrátane digitálnej kamery, mp3 prehrávača, atď.) sú sformátované FAT systémom súborov, a tento sprievodca vám pomôže jako ich sprístupniť v Linuxe, pre prácu s nimi.

Plná podpora zápisu na NTFS je možná použitím [captive](#) third-party modulu (prístupného na [portage](#)), ktorý používa Window's NTFS DLLs (knižnice)

Ak chcete formátovať vašu partíciu jedným zo systému súborov FAT, mkfs.vfat a mkfs.msdos sú prístupné v [dosfstools](#) balíku.

Pripojenie vašej jednotky z Windowsom je jednoduché, záleží od skonfigurovaní vášho jadra so správnymi možnosťami.

Doporučujem prečítať si [Linux-NTFS project](#).

[[edit](#)]

Kernel Configuration (konfigurácia jadra)

Ak používate genkernel, DOS / VFAT / NTFS podpora je už vo vašom jadre a prístupná jako modul. Môžete ich zaviesť jako root - **modprobe** a msdos, vfat, ntfs.

Ak si komilujete svoje vlastné jadro a chcete mať podporu FAT/DOS/NT systému súborov napíšte make menuconfig nájdite v menu File Systems menu. Vstúpte do *DOS/FAT/NT Filesystems* podmenu

```
Linux Kernel Configuration: file systems
File Systems --->
  DOS/FAT/NT Filesystems --->
    <M> MSDOS fs support
    <M> VFAT (Windows-95) fs support
    <M> NTFS file system support
```

V príklade hore sú všetky možnosti jako moduly. Vo viacerých prípadoch nebudete potrebovať *MSDOS fs podporu*, odkedy je podpora zápisu FAT16 / FAT32 fs. Väčšina používateľov potrebuje pridať *VFAT (Windows-95) fs podporu k prístupu na ich disky s FAT fs*, a *NTFS fs podporu na prístup k ich disku s Windowsom*.

Před ukončením konfigurácie jadra, treba pridať: Native language support.

"Ak chcete zobrazovať názvy súborov s pôvodnými jazykovými znakmi z Microsoft FAT fs family alebo z JOLIET CD-ROMs na obrazovke správne, musíte zahrnúť vhodnú input/output znakovú sadu."

Ku NLS podpore, výjdite z DOS/FAT/NT pod-menu, vstúpte do pod-menu vo File Systems for *Native Language Support*.

Linux Kernel Configuration: NLS

```
Native Language Support
--- Base native language support
(iso8859-1) Default NLS Option
<M>   Codepage 437 (United States, Canada)
<M>   NLS ISO 8859-1 (Latin 1; Western European
Languages)
```

Ak ste už povolili jeden zo systému súborov v predchádzajúcom menu, prvá možnosť v tomto menu bude zaškrtnutá ako zahrnutá (--- Base native language support). Druhá možnosť má defaultnú NLS možnosť. Nemeňte ju pokiaľ viete čo robíte!

Ak sa chystáte používať MS-DOS podporu (čítať/zápis na MS-DOS partíciu, nie VFAT), potom budete musieť označiť jednu z *Codepage* možností ako modul. Každé codepage možnosť má krajinu alebo región vypísanú vedľa, takže výber by nemal byť zložitý. Ak používate len VFAT a/alebo NTFS podporu, túto možnosť môžete preskočiť

Používatelia VFAT a/alebo NTFS read/write podpory, budú musieť označiť jednu z NLS ISOs. Znovu, zvolte si kódovanie pre vás.. Pre Slovensky-hovoriacich, možnosť NLS ISO 8859-2 je čo budete potrebovať. (Ospravedlňujem sa tým ktorých kódovanie neuvádzam...neviem...:-))

Uložte svoje nastavenia jadra, skompilujte (a reštart). Alebo nahrajte moduly keď skončíte.

[\[edit\]](#)

Pripájanie partícií

Ak ste tak neurobili, urobte tak **man mount!**.

Takže podporu máme zavedenú, ideme na to. Ak chcete pripojiť nejakú existujúcu partíciu na váš disk, prvá vec ktorú treba urobiť je nájsť ktoré zariadenie je aké. Dobré je spustiť fdisk **fdisk -l** (l jako list) jako root .Tento príkaz jednoducho vypíše vaše diskové partície a ukončí sa.

Výstup z príkazu fdisk-l

Code: fdisk -l						
	Device	Boot	Start	End	Blocks	Id
System	/dev/hda1	*	x	xxxx	xxxx+	7
HPFS/NTFS	/dev/hda2		xxxx	xxxx	xxxx	b
W95 FAT32						

V tomto príklade, the harddrive running Windows has its C:\ drive formatted as NTFS and its D:\ drive formatted as Windows 95 FAT32 (VFAT). And, from the Device column, we can see that they are listed as **/dev/hda1** and **/dev/hda2**.

Teraz máme všetky informácie a môžeme pokračovať

[\[edit\]](#)

Základy

Pre príklad my použijeme NTFS partíciu takže jako root vytvorte v zložke **/mnt** zložku **ntfs**.

```
# mkdir /mnt/ntfs
```

Ďalej pripojíme partíciu. Keď pripájate, použite -t argument pre špecifikovanie systému súborov (ntfs, v tomto prípade), potom názov zariadenia (/dev/hda1 napríklad) a zložku kde sa má pripojiť (/mnt/ntfs):

```
# mount -t ntfs /dev/hda1 /mnt/ntfs
```

Pre pripojenie FAT partície nahradte X a vytvorte si zložku na pripojenie teda v /mnt zložku vfat kde X znamená číslo zariadenia.

```
# mount -t vfat /dev/hdxX /mnt/vfat
```

Pre MS-DOS

```
# mount -t msdos /dev/hdxX /mnt/msdos
```

[\[edit\]](#)

/etc/fstab

Teraz keď už viete ako pripojiť partície Windows alebo MS-DOS systémov, môžete pridať riadok do /etc/fstab súboru, a neskôr si tak zjednodušiť pripájanie

.

The filesystem tabular file (fstab) zápisy by mali byť v jednom riadku, s medzerami alebo tabmi medzi stĺpcami. Stĺpce sú v poradí filesystem (alebo zariadenie), prípojný bod, typ súborového systému, možnosti, a premenné..

Otvorte si súbor z vašim obľúbeným editorom (ako root), a pridajte nový riadok na konci súboru.

Prvé v píšete zariadenie (/dev/hdaX napr.). Z príkladu hore ,ak chcete pripojiť windows disk C:\ formátovaný ako NTFS s default hodnotou (viac možností vysvetlíme neskôr), tak je toto čo by ste mali napísať.

```
/dev/hda1 /mnt/ntfs ntfs defaults 0 0
```

"Defaults" hodnota pripojí disk už pri boot-e, a nastaví na read/write prístup. Čítajte ďalej ak chcete pridať viac možností keď pripájate partície vo fstab.

[\[edit\]](#)

Pripájanie - možnosti

ro: len na čítanie

Ak chcete túto možnosť, nezapisovať na disk, zmeňte "defaults" na "ro"

```
/dev/hda1 /mnt/ntfs ntfs ro 0 0
```

noauto: neprípajaj při boot-e

Ak nechcete aby sa disk pripájal při bootovani, ale chete pripájať manuálne, pridajte "noauto" jako možnosť.

```
/dev/hda1 /mnt/ntfs ntfs ro,noauto 0 0
```

[\[edit\]](#)

user, users: dovol' užívateľovi mount a unmount

"Zvyčajne, pripájať môže len superužívateľ-root. Ničmenej, keď fstab obsahuje **user** možnosť tak so systémom môže pracovať každý užívateľ-pripájať odpájať..

```
/dev/cdrom /mnt/cdrom iso9660 ro,user,noauto,unhide
```

ktorýkoľvek užívateľ môže pripojiť iso9660 file system obsahujúci CDROM použitím príkazu

```
mount /dev/cdrom
```

alebo

```
mount /mnt/cdrom
```

uid,gid: pripoj jako (užívateľ,skupina)

Na všetkých troch fs (MS-DOS, VFAT, NTFS), môžete pripájať podľa **uid** a **gid** možností, ktorá vám dovoľí nastaviť užívateľové a skupinové IDs ktorých sú vlastníci mountovaných systémov

Pre zistenie hodnoty **uid** a **gid**, príkaz **id <user>**.

```
$ id larry
uid=1000(larry) gid=100(users)
```

Takže ak by ste chceli všetko vlastnené jako larry:užívateľ na **/mnt/ntfs**, toto je čo by ste mali použiť vo **/etc/fstab**

```
/dev/hda1 /mnt/ntfs ntfs uid=1000,gid=100 0 0
```

Bez nejakej možnosti, pripojenie bude default ako user 0 a group 0, alebo root.

umask: octalové súborové práva

Môžete zmeniť práva pridaním **umask**. Je to octálové číslo, vypadá nejak takto::

- **character '0':** Ukazuje že je to oktálové číslo, nie decimálne.
- **first digit:** vlastnícke práva užívateľa
- **second digit:** vlastnícke práva skupiny

- **third digit:** celkové práva ostatných (ostatný užívateľ systému)

Módy sú nasledujúce (prvý stĺpec je oktálové číslo):

M	R	W	X
0		*	*
1		*	*
2		*	-
3		*	-
4		-	*
5		-	*
6		-	-
7		-	-

File: /etc/fstab

Napríklad, ak chcete aby **každý** mohol **čítať**, **zapisovať** a **spúšťať** každý súbor na **/mnt/c**, špecifikujte **mask 0000**:

```
/dev/hda1 /mnt/vfat vfat umask=0000 0 0
```

Ak chcete aby len užívatelia zo skupiny 610 mohli **čítať**, **zapisovať**, a **spúšťať**:

```
/dev/hda1 /mnt/vfat vfat gid=610,umask=0707 0 0
```

Ak chcete aby len užívatelia zo skupiny 610 mohli **čítať**, a **spúšťať** (**nie zapisovať**):

```
/dev/hda1 /mnt/vfat vfat gid=610,umask=0727 0 0
```

utf8: kódovanie

Taktiež môžete pridať **iso8859-2** možnosť, Ak názvy súborov správne:

```
/dev/hda1 /mnt/vfat vfat ro,utf8 0 0
```

Tento dokument je preložený z Anglického jazyka do Slovenského jazyka s chybami, avšak príkazy a možnosti sú plne funkčné a bez chýb.

Originál Dokument nájdete nižšie. Hlavným cieľom bolo prispieť k slovenským manuálom.

Dokument obsahuje tie najzákladnejšie možnosti pri pripájaní zariadení (USB kľúč, diskové partície, digitálnu kameru...). Všetky príkazy sa zapisujú pod superužívateľom -root. Dbajte na správnosť zápisov do súboru fstab, i pri najmenšej chybe či zamenení písmenka sa môže stať že sa vám disk či disková partícia nepripojí.

V prípade že preklad nieje zrozumiteľný a obsahuje chyby vo vysvetlení príkazov, máte plnú moc tento dokument opraviť, nahradiť chyby. Dokument bol prevzatý z www.wikipedia.org pre gentoo ale jeho príkazy budú pracovať správne i na iných Linux systémov.

Pre vašu spokojnosť www.linuxos.sk a používateľ Linux Open SuSE 10, Gentoo, Slackware 10.2

Tomáš Hanzel thanzel@atlas.sk www.cvcpart.edu.sk
Close The Windows, Open The Source!

If you are dual-booting your system with Windows, at some point you may want to share files between the two operating systems. Fortunately, the linux kernel provides full read/write support for [FAT32](#) and the older FAT filesystems. Full [NTFS](#) read-only support is natively available in the kernel as well, along with *experimental* write support.

Many hardware flash drives (included in digital cameras, mp3 players, for example) format their partitions with the VFAT filesystem, and this guide will apply to those who want to access their data on those devices as well.

Full write support for NTFS is available using the [captive](#) third-party modules (available in [portage](#)), which uses Window's NTFS DLLs from your installation to access the data.

If you are interested in formatting your partitions to one of the FAT filesystems, mkfs.vfat and mkfs.msdos are available in the [dosfstools](#) package.

Mounting your Windows drives is simply a matter of reconfiguring your kernel with the right options.

Recommended reading before proceeding includes the documentation for the [Linux-NTFS project](#), which provides the [ntfsprogs user tools](#), and the MAN page for mount.

[[edit](#)]

Kernel Configuration

If you are using genkernel, chances are that DOS / VFAT / NTFS support are already included into your kernel, and are readily available as modules. You can load them respectively as root with **modprobe** and msdos, vfat, ntfs.

If you are compiling your own kernel and wish to add modular or included support into yours, go to make menuconfig and navigate to the File Systems menu. From there will be an entry for *DOS/FAT/NT Filesystems* leading to a sub-menu.

```
Linux Kernel Configuration: file systems
File Systems --->
  DOS/FAT/NT Filesystems --->
    <M> MSDOS fs support
    <M> VFAT (Windows-95) fs support
    <M> NTFS file system support
```

In the above example, all three options are selected as modules. In most cases, you'll never need *MSDOS fs support*, since that is for writing to very old FAT16 / FAT32 file systems. Most general users would need to add *VFAT (Windows-95) fs support* to access their Windows system and/or flash devices, and *NTFS file system support* to access their Windows partitions.

Before exiting your kernel configuration, there is one more thing to add: Native language support. The kernel's Help sub-option describes the necessity well:

"If you want to display filenames with native language characters from the Microsoft FAT file system family or from JOLIET CD-ROMs correctly on the screen, you need to include the appropriate input/output character sets."

To get to NLS support, exit out of the DOS/FAT/NT sub-menu, and enter the sub-menu in File Systems for *Native Language Support*.

Linux Kernel Configuration: NLS

```
Native Language Support
--- Base native language support
    (iso8859-1) Default NLS Option
    <M>   Codepage 437 (United States, Canada)
    <M>   NLS ISO 8859-1 (Latin 1; Western European
Languages)
```

If you've already enabled one of the filesystem options in the other menu, the first option in this menu will be marked as included (--- Base native language support). The second line has a default NLS option. Don't change it unless you know you need to.

If you are going to use MS-DOS support (read/write to MS-DOS partitions, not VFAT), then you will have to either compile in or select one of the *Codepage* options as a module. Each codepage option has a country or region listed next to it, so choosing yours shouldn't be difficult. If you are only using VFAT and/or NTFS support, you can skip that option.

Users of VFAT and/or NTFS read/write support, will need to choose one of the NLS ISOs that are available. Again, choose the selection that most closely resembles your region or language. For English-speaking cultures, the first option, NLS ISO 8859-1 is what you will need. (My apologies to others, this was written by a gringo.)

If you need to add support for files saved with Unicode characters, then NLS UTF8 is available as an option as well.

Save your settings, recompile your kernel (and reboot), or load the modules when finished.

[\[edit\]](#)

Mounting Partitions

If you haven't yet, read **man mount** for some really good information on what your options and limitations are in mounting each filesystem type.

Once support is added into your kernel, you're ready to go. If you want to mount some existing partitions on your harddrives, the first thing to do is find out what their device names are listed as. A good way to find out is to run **fdisk -l** (that's l as in list) as root. That command will simply list your disk partitions and exit.

Below is a sample output:

Code: fdisk -l					
Device	Boot	Start	End	Blocks	Id
System					
/dev/hda1	*	x	xxxx	xxxx+	7
HPFS/NTFS					
/dev/hda2		xxxx	xxxx	xxxx	b
W95 FAT32					

In this example, the harddrive running Windows has its C:\ drive formatted as NTFS and its D:\ drive formatted as Windows 95 FAT32 (VFAT). And, from the Device column, we can see that they are listed as `/dev/hda1` and `/dev/hda2`.

Now you should have everything you need to mount the filesystems.

[\[edit\]](#)

Basics

Assuming you are root, create a directory to access your drive in `/mnt`. For this example, we'll use `ntfs` as the directory name for the NTFS drive.

```
# mkdir /mnt/ntfs
```

Next, mount the partition using the mount command. When using mount, use the `-t` argument to specify the filesystem type (`ntfs`, in this case), then the device name (`/dev/hda1` from the example) and the directory to access the drive (`/mnt/ntfs`):

```
# mount -t ntfs /dev/hda1 /mnt/ntfs
```

If you were going to mount a FAT32 partition, the `-t` type would be `vfat`. Just replace the `x` and `X` in `/dev/hdxX` with the partition device and number.

```
# mount -t vfat /dev/hdxX /mnt/vfat
```

For MS-DOS partitions, the option would be `msdos` instead.

```
# mount -t msdos /dev/hdxX /mnt/msdos
```

[\[edit\]](#)

/etc/fstab

Now that you know how to mount a Windows or MS-DOS filesystem, you can add an entry to your `/etc/fstab` file, so that next time you can simplify the process.

The filesystem tabular file (`fstab`) entries should be one per line, with spaces or tabs between columns. The columns are in order of filesystem (or device), mountpoint, filesystem type, options, and dump / pass variables.

Open up the file with your favorite editor (as root), and add a new line at the end of the file.

First, put your filesystem's device. From our example above, if you wanted to mount your Windows C:\ drive formatted as NTFS with the default values (more options explained below), this is what you would add to your file:

```
/dev/hda1 /mnt/ntfs ntfs defaults 0 0
```

The "defaults" option will both mount the partition at every time at boot, and set it to read/write access. Read below for more options you can pass when mounting the partitions in `fstab`.

[\[edit\]](#)

Mount Options

The same options that are available with the mount program can be passed in fstab as well. Again, read [man mount](#) for a full explanation of your options, as this wiki entry will only cover the basics.

[\[edit\]](#)

ro: readonly filesystem

If you wanted it so you can only read data, and not write to the hard drive, change "defaults" to "ro" instead.

```
/dev/hda1 /mnt/ntfs ntfs ro 0 0
```

[\[edit\]](#)

noauto: don't mount at boot

With that option [/mnt/ntfs](#) will always be mounted everytime you boot. If you didn't want it to automatically mount each time, but instead mount it manually, add "noauto" to the options. Options in the fstab file are comma separated, with no spaces between them.

```
/dev/hda1 /mnt/ntfs ntfs ro,noauto 0 0
```

[\[edit\]](#)

user, users: let users mount and unmount

Taking a page from [man mount](#) (which explains it much better than we ever could),

"Normally, only the superuser can mount file systems. However, when fstab contains the **user** option on a line, then anybody can mount the corresponding system.

Thus, given a line

```
/dev/cdrom /mnt/cdrom iso9660 ro,user,noauto,unhide
```

any user can mount the iso9660 file system found on his CDROM using the command

```
mount /dev/cdrom
```

or

```
mount /mnt/cdrom
```

For more details, see [fstab\(5\)](#). Only the user that mounted a filesystem can unmount it again. If any user should be able to unmount, then use **users** instead of **user** in the fstab line."

In summary, add the **user** option to allow any user mount the partition, but that user will be the only one who can unmount it. Or add the **users** option to let any user mount or unmount that partition.

[\[edit\]](#)

uid,gid: mount as (user,group)

On all three filesystems (MS-DOS, VFAT, NTFS), mount lets you pass the **uid** and **gid** options, which let you set the user and group IDs of who the files are chown'd to when the filesystem is mounted.

To find out the values for **uid** and **gid**, run **id <user>**.

```
$ id larry
uid=1000(larry) gid=100(users)
```

So if you wanted everything owned as larry:users on **/mnt/ntfs**, this is what you would use in **/etc/fstab**

```
/dev/hda1 /mnt/ntfs ntfs uid=1000,gid=100 0 0
```

Without any options, mount will default to user 0 and group 0, or root.

[\[edit\]](#)

umask: octal file permissions

You can change permissions using the parameter **umask**. But be aware that it must be the bitmask of permissions that are not present for the mountpoint. It is an octal number, formed like this:

- **character '0'**: Indicates that this is an octal number, not decimal.
- **first digit**: owner user permissions
- **second digit**: owner group permissions
- **third digit**: world permissions (every other user on the system)

The modes are as follows (the first column is the mode octal number):

M		R	W	X
0		*	*	*
1		*	*	-
2		*	-	*
3		*	-	-
4		-	*	*
5		-	*	-
6		-	-	*
7		-	-	-

File: /etc/fstab

For example, if you want that **everybody** be able to **read, write, and execute** every file in your **/mnt/c**, you should specify the **mask 0000**:

```
/dev/hda1 /mnt/vfat vfat umask=0000 0 0
```

If you want that only users from group 610 be able to **read, write, and execute**:

```
/dev/hda1 /mnt/vfat vfat gid=610,umask=0707 0 0
```

If you want that only users from group 610 be able to **read, and execute (not write)**:

```
/dev/hda1 /mnt/vfat vfat gid=610,umask=0727 0 0
```

[\[edit\]](#)

utf8: Unicode support

You may also add **utf8** option, if you can't see some files with international characters in filenames.

```
/dev/hda1 /mnt/vfat vfat ro,utf8 0 0
```