UMSDOS HOW−TO

Table of Contents

Table of Contents

UMSDOS HOW−TO

Jacques Gelinas, `jacques@solucorp.qc.ca`

v1.1, 13 November 1995

Umsdos is a linux file system. It provide an alternative to the EXT2 file−system. Its main goal is to achieve easier coexistence with Ms−DOS data by sharing the same partition. This document explain first how to use Umsdos in different configuration, and later explain its operation and try to provide some information letting you decide if it is a good choice for you (see UMSDOS−WHY−TO at the end).

5.Basic principle

6.Installation/UN−installation and some tricks

7.Setting a *Linux* section in a *DOS* partition

8.UMSDOS−WHY−TO

1.UMSDOS: Where is it ?

1.1 History

The *Umsdos* project was started in 1992 and made available to the net in January 1994 as a patch. It was included in the standard kernel distribution in July, starting with kernel 1.1.36.

Umsdos was early adopted in the *Slackware* distribution even before it was officially included in the official kernel.

Umsdos was improved starting at kernel 1.1.60. Its performance has been dramatically enhanced, especially for writing. Since 1.1.70 (around this), it is stable again.

A major bug was solve in *Linux* 1.2.2. This bug was causing some grief to users since the beginning (some file were silently renamed, giving the sad impression that they were deleted). Beware that *Slackware* 2.2 is still shipping release 1.2.1 of the kernel, so has this bug.

1.2 Availability

It is available as a patch for kernel 1.0.x. It is built−in for kernel 1.2. It can be compiled in or load as a module. Beware that for now, if you intend to load *umsdos* as a module, you must also use the Ms−DOS fs as a module. This come from a limitation in the module system (some symbols are only export when the drivers is installed as a module).

1.3 Distribution supporting it

So far, I think only *Slackware* does support it. I am surely wrong, so please send me info to correct this.

1.4 Home site

The home site for *Umsdos* is sunsite.unc.edu. Look in the directory `/pub/Linux/system/Filesystems/umsdos`.

1.5 Technical documentation

There is quite a lot of documentation about the internal of *Umsdos*. It is available both in `HTML` and text format at the same location as the utilities.

As far as I know, the `HTML` version is not available online on any web site. You must down−load it and "UN−tar" it and read it locally.

1.6 Who wrote it

Jacques Gelinas jacques@solucorp.qc.ca

2.Umsdos as your root partition

2.1 The pseudo−root concept.

With *Umsdos*, *Linux* can be installed in a standard DOS partition. *Linux* is then installed as a second (or third) OS in the partition. To avoid name collision (there is maybe a bin or tmp directory in the drive C: already), *Umsdos* use a smart trick: The pseudo−root.

All *Linux* files are installed in a DOS subdirectory called `linux`, generally `C: LINUX`. The normal *Linux/Unix* directory structure goes there. So you get

- `C:\LINUX\BIN`
- `C:\LINUX\ETC`
- `C:\LINUX\LIB`
- `C:\LINUX\ROOT`
- `C:\LINUX\SBIN`
- `C:\LINUX\TMP`
- `C:\LINUX\USR`
- `C:\LINUX\VAR`

When the *Umsdos* boot, it probes for the directory `linux` and then `/linux/etc`. If it exist, it activates the pseudo−root mode.

Mostly, the pseudo−root mode switch the root of the partition to `C:\\LINUX` giving the conventional *Unix* directory layout

- `/bin`
- `/etc`
- `/lib`
-

- `/root`
- `/sbin`
- `/tmp`
- `/usr`
- `/var`

To this list, it adds a new one called `DOS`. This one is a virtual directory.

2.2 Things to know about the pseudo−root

- This mode can only be triggered at boot time. There is no way to activate this by a mount command.
- This mechanism is purely a different view of a normal *Umsdos* file−system. This means that a partition normally used as a root partition can be normally mounted. There won't be any pseudo−root effect. For example, if you boot linux with a maintenance floppy and mount your normal root partition in `/mnt`, you will find all your linux directory in `/mnt/linux/bin`, `/mnt/linux/etc` and so on.

3.Different topics about the operation of *Umsdos*

3.1 Mount option

You can use the same mount option as for the Ms−DOS file system. The option conv= is questionable on a *Umsdos* system. I suggest to avoid it. Mostly the option you may want to look at are

- uid=
- gid=
- umask=

Just remember that *Umsdos* manage non promoted directory the same way as the *Ms−DOS* file system. The option above will apply globally to all non promoted directory. *uid* setup the default owner, *gid* setup the default group and *umask* setup the default permissions.

3.2 How to set defaults for the root

umssetup was created to provide at run time default ownership for the root partition. For other *Umsdos* partition, mount option may be used or umssetup. Storing mount option in /etc/fstab is the prefered way for non root partition. Here is an example. Put this in /etc/rc.d/rc.S.

```
/sbin/umssetup −u jack −g group −m 0755 /
```

3.3 To swap or not to swap

Using a swap file is generally slower than a swap partition. It is however much more flexible. You can setup a swap file in a *Umsdos* partition the same way you do it for any other *Linux* file systems. For example, to setup a 8 megabytes swap file in the root directory:

```
dd if=/dev/zero bs=1024k count=8 of=/swap
mkswap /swap 8192
sync
swapon /swap
```

Once done, you can put the following line in /etc/fstab

```
/swap    swap    swap    default
```

And the swap file will be activated at each boot (There is generally a "swapon −a" in `/etc/rc.d/rc.S`).

4.How to boot a Umsdos system

4.1 Loadlin

The package lodlin15.tgz available from sunsite.unc.edu in `/pub/Linux/system/Bootutils`. This utility is particularly suited to boot a *Umsdos* system. Generally all you need to do is

```
Boot DOS
C:>loadlinx zimage root=D:
```

where zimage is a normal kernel image (compressed) simply copied somewhere in the DOS drive. `D:` is the DOS drive where you have installed *Linux*.

4.2 From a floppy

Booting a *Umsdos* system from a floppy is not different from booting a *Ext2* system. You need a kernel zImage file properly initialize to locate your root *Umsdos* partition. This is generally achieved using the command `rdev`. The following sequence will initialize a zImage and put it on a floppy.

```
rdev zImage /dev/hda1
rdev −R zImage 0
dd if=zImage bs=8192 of=/dev/fd0
```

If this looks confusing, just format a boot−able DOS floppy and put the following component on it.

- loadlin.exe
- loadlinx.exe
- zimage

and setup the autoexec.bat like this

```
loadlinx zimage rw root=C:
```

4.3 LILO

LILO, the official *Linux* boot loader can also be used to boot a *Umsdos* system. I have no experience with it though. Since 1.1.60, it should work. Please email if you know something.

4.4 How to defragment a *Umsdos* partition

It can be done using any popular DOS tool. There is nothing particular about file produced by *Umsdos*. And *Umsdos* do not expect anything particular (directory layout, directory entry sequence, etc...) from the file system under it.

As far as I know, there is no *Linux* tool to achieve this.

4.5 Advance tricks

Umsdos rely on the `--linux-.---` which rely on the *DOS* directory. Some users may want to experiment a bit. The utility `udosctl` part of the umsdos_progs package (containing `umssync` and `umssetup`) allows basic directory operation (listing, deletion) independently on the `--linux-.---` and the *DOS* directory.

5.Basic principle

5.1 Introduction

Umsdos map *Linux* files directly to *Ms−DOS* files. This is a one for one translation. File content is not manipulated at all. *Umsdos* only works on names. For special files (links and devices for example), it introduces special management.

For each directory, there is a file named `--linux-.---`.

5.2 *Umsdos* can replace the *Ms−DOS* file−system.

Umsdos can be thought as a general purpose superset of the *Ms−DOS* file system of linux. In fact this capability or flexibility yields much confusion about *Umsdos*. Here is why. Try to mount a newly formatted *DOS* floppy like this.

```
mount −t umsdos /dev/fd0 /mnt
```

And do this,

```
ls / >/mnt/LONGFILENAME
ls −l /mnt
```

You will get the following result

```
−rwxr−xr−x   1 root     root          302 Apr 14 23:25 longfile
```

So far, it seems that the *Umsdos* file system does not do much more (in fact nothing at all) than the normal *Ms−DOS* file system of *Linux*.

???

5.3 Directory promotion

Pretty unimpressive so far. Here is the trick. Unless promoted a *DOS* directory will be managed the same way with *Umsdos* than the *Ms−DOS* file−system will. *Umsdos* use a special file in each subdirectory to achieve the translation between the extended capabilities (long name, ownership, etc...) of *Umsdos* and the limitation of the *DOS* file−system. This file is invisible to *Umsdos* users, but visible when you boot *DOS*. To avoid cluttering the *DOS* partition with those file (`−−linux−.−−−`) uselessly, the file is now optional. If absent, *Umsdos* behave like *Ms−DOS*.

When a directory is promoted, any subsequent operation will be done with the full semantic normally available to *Unix* and *Linux* users. And all subdirectory created afterward will be silently promoted.

This feature allows you to logically organize your *DOS* partition into *DOS* stuff and *Linux* stuff. It is important to understand that those `−−linux−.−−−` file do take some place (generally 2k per directory).

DOS generally use large cluster (as big as 16k for a 500meg partition), so avoiding putting `--linux-.---` everywhere can save your day.

5.4 How to promote: `/sbin/umssync`

A directory can be promoted any time using `/sbin/umssync`. It can be used at any time. Promoting a directory do the following operation

- Create a `--linux-.---`.
- Establish a one to one relation between the `--linux-.---` and the current content of the directory.

`/sbin/umssync` maintain an existing `--linux-.---` file. It does not create it from scratch all the time. It simply add missing entries in it (Files created during a *DOS* session). It will also removed files which do not exist anymore in the *DOS* directory from the −−*linux*−.−−−. `umssync` gets its name from that. It put −−*linux*−.−−− in sync with the underlying *DOS* directory.

5.5 Using `/sbin/umssync` at boot time

It is a good idea to place a call to `/sbin/umssync` at the end of your /etc/rc.d/rc.S if it's not there. The following command is adequate for most system:

```
/sbin/umssync -r99 -c -i+ /
```

The `-c` option prevent `umssync` from promoting directories. It will only update existing `--linux-.---`.

This command is useful if you access *Linux* directory during a *DOS* session. *Linux* has no efficient way to tell that a directory has been modified by *DOS* so *Umsdos* can't do a *umssync* operation as needed.

5.6 How to UN−promote

Remove the `--linux-.---` file using *DOS*. You will be sorry.

5.7 What about files created during a *DOS* session ?

Unless you use `umssync` on a directory where files have been added or removed by *DOS*, you will notice some problems:

- It won't crash the system nor it won't cause major problems, only annoyance :−)
- Files created by *DOS*.
    - They will be invisible in *Linux*.
    - When trying to create a file with the same name, you will get an error message stating that the file already exist.
    - This creates more confusion that real problem. It does not harm the file system.
- Files deleted by *DOS* won't cause problem. *Umsdos* will notice the absence at the first access. A message will be output (and generally written into `/var/adm/syslog`).

6.Installation/UN−installation and some tricks

The installation of a *Umsdos* is not much different from the installation of an ordinary (*Ext2 based*) *Linux* system.

There are two main differences.

6.1 The pseudo−root `/mnt/linux`

The normal steps for an installation are

1. Setting a partition with fdisk and formatting it.
2. Mounting it as /mnt relative to our installation root disk.
3. Copy all packages into `/mnt`.

With *Umsdos*, the step 1 is not required (wasn't it the goal of *Umsdos* not to reformat ?).

It is possible to install a *Umsdos* system just by copying all packages into `/mnt`. This will certainly work. But it will create a bunch of subdirectories into your *DOS* root directory (C:) and you won't like it. This is the reason all *Umsdos* installation use the pseudo−root. And this is the major difference between a normal *Ext2* installation and a *Umsdos* one: All files are copied into `/mnt/linux`.

6.2 Preparing `/mnt/linux`

`/mnt/linux` is not an ordinary directory. It has to be promoted so it will correctly handle *Linux* long file name and special files (links, device). The step required to setup `/mnt/linux` are:

1. `mkdir /mnt/linux`
2. `umssync /mnt/linux`

That's it!

6.3 Making sure `/mnt/linux` is correctly setup

Even if the setup of `/mnt/linux` is pretty simple, there are many installation package out there who get it wrong. How can ?

The biggest installation problem come from an incompatible `umssync` program. *Umsdos* has been update in linux 1.1.88 (Can't remember exactly) and a flaw was uncovered in `umssync`. To avoid confusion in the *Linux* community, it was decided to raise the compatibility level required for all *Umsdos* tools. Old version of the tools were simply rejected.

It sounds like many distribution did not update their `umssync` utility on the installation disk.

There are still many distribution like this out there. The net result is that the directory `/mnt/linux` is not promoted at all and will truncate all long file name and will reject all special file.

It is possible to do a test very early during the installation to find out if something went wrong. Thanks to the pseudo console mechanism of *Linux*, you can do that without leaving the installation program. Do the following steps:

1. Press `Alt−F2` (`Alt` key at the same time as the `F2` key).
2. login as root.
3. `cd /mnt/linux` If this fail, you are trying this too early. A good time to do this is at the end of the packages selection.
4. `>TOTO`
5. `ls −l` You should see an empty file `TOTO` in uppercase. If you see it in lowercase, something went wrong. Try to do the `umssync` step again. `umssync` can be use over and over without problem. `umssync .` If there is no error message, try the `TOTO` test again. If `TOTO` appears fine, then all is OK. Something is strange in this installation, but you just save it. Continue
6. Press `Alt−F1` to get back to the installation screen.

If the test fail, the best fix is to get a newer installation root disk. You can generally fix this root disk by installing a newer version of `umssync`. This is not difficult but required a working *Linux* system. You simply have to mount the root disk floppy and replace the offending `umssync` with a new one.

6.4 Oops releasing pseudo root ...

Most *Umsdos* installation which fail, do this by printing this strange message. This is not a bug in *Umsdos* although the message looks strange. Here are the known causes.

- The most common one

  The *Slackware* installation try to setup a swap file very early during the installation. To do so, it asks you to select a partition (dos drive), then mount it and set the swap file.

  When installing a *Slackware* system, you must setup the target partition prior to install. This normally mounts the *DOS* partition on `/mnt`, creates the `/mnt/linux` directory and applies `umssync` on it.

  This is where most problems come from. Most user just forget the "setup target partition" step and go directly to the rest of the installation. Since `/mnt` is already mounted, this mistake goes unnotice. This means that `/mnt/linux` was not created properly (Not promoted). All special files and links and long names can't be created properly.
- Invalid umssync utility

  `/mnt/linux` was improperly setup−ed. Generally caused by an improper `umssync` utility on the installation root disk.
- Old bug in umsdos

  There was a bug in *Umsdos* prior to *Linux 1.2.2*. The pseudo−root mode would not activate properly if the file `/etc/init` was missing. `init` is now located in `/sbin`. You can fix it by getting a newer kernel. This is recommended because another bug was uncover and fixed in 1.2.2.

  If you can't upgrade, do this

  1. Boot from you installation disk.
  2. Login as root.
  3. `mount −t umsdos /dev/hdXX /mnt` where `/dev/hdXX` is your *DOS* partition.
  4. `cd /mnt/linux/etc`
  5. `ln −s ../sbin/init init`
  6. `cd /`
  7. `Ctrl−Alt−Del`
  8. Boot your *Umsdos* normally.

Unfortunatly, the first two (Installation problems) produce a completly unusable installation. Uninstall it (See next section) and install again.

6.5 How to UN−install a *Umsdos* system

One neat thing about *Umsdos* and its pseudo−root mechanism, is that you can UN−install it without pain. You just boot *DOS* and recursively delete the `linux` directory. That's all. *Umsdos* requires no special drivers in the config.sys, nor it creates anything special outside of the `linux` directory.

6.6 Moving a *Umsdos* system to another *DOS* drive

This can be done from *Linux* or from *DOS*. You just have to copy recursively the `linux` directory from one drive to the other. After that you will have to adjust you boot mechanism (generally loadlin command) and the `/etc/fstab` file.

Umsdos can live on any *DOS* drive. There is no need to install it on the `C:` drive, nor it is important to have it on the first hard drive. It does not matter at all.

In fact, one may decide to have several *Umsdos* installations on different drive just to do experiments.

6.7 About installing 50 *Umsdos* systems.

How about installing a bunch of *Linux* systems in no time ?

Umsdos systems are living in a *DOS* world. You can take advantage of this if you wish to install *Linux* easily.

You can install and configure a *Umsdos* system at your site. When you are satisfied with the configuration and the different packages you have selected, you can boot *DOS* and copy the complete `linux` directory to your *DOS* file server. Then you go to other *DOS* station and simply copy the files on the network drive to the local drive. That's it. Only adjust the boot script (*Loadlinx*) and go.

With minimal adjustment (Host name, IP number), anyone will be able to install a *Linux* system in a matter of minute.

Interest readers may note that installing *Linux* systems by copying running system also works for any other *Linux* systems, including *Ext2* based one.

One beauty of *Linux* is that there is no hidden files which have to be install by magic installation program.

7.Setting a *Linux* section in a *DOS* partition

Umsdos has some use even for *Ext2* (Native *Linux* file−system) users. One common scenario is this:

- *Linux* being your *OS* of choice, the *Linux* partition start to fill and fill and fill.
- Your *DOS* partition is collecting dust, being half empty.

* You are suddenly out of space in the *Ext2* partition.
* You are still not sure you want to get rid of *DOS*.

Umsdos may save the day here. You can setup a *Linux* directory in the *DOS* partition and use it without restriction for *Linux* usage. For example, say you want to setup a new directory named `"extra"` in your `C:` drive. And you want this directory to behave as a normal *Linux* directory. Do this (assuming that C: is /dev/hda1).

```
mkdir /c
/sbin/mount −t umsdos /dev/hda1 /c
mkdir /c/extra
umssync /c/extra
```

You must be root to do this.

By setting up `/etc/fstab` like this, you will always have access to the `/c/extra` directory.

8.UMSDOS−WHY−TO

Explaining how to operate or install a *Umsdos* system is not enough. Most people are seeking some advises about using *Umsdos* or not.

8.1 The goal of *Umsdos*

The goal of *Umsdos* was to ease the installation of *Linux*. An other goal was to ease its UN−installation. The idea here was to promote the spreading of *Linux*. Installing a new OS on a system is always troublesome. *OS/2* for one will happily pollute your `C:` root with a bunch of new directories. If you are clever like me, it will also erase your config.sys and autoexec.bat files :−(

The pseudo−root feature of *Umsdos* avoid this unwanted invasion. *Linux* can be UN−install without side effect.

8.2 Who needs it

If you have a small hard drive, *Umsdos* will allow you to share disk space between *DOS* and *Linux*. A disk below 300 megs is in my opinion a small disk. This opinion is based on the size of the different package available today. One popular word processor may eat as much as 70 megabytes if you select all features.

If you have a larger drive, you may consider having a dedicated *Linux* partition running the *Ext2* file−system. *Ext2* use a smaller cluster size that *DOS* (1k in fact) so installing many small files will eat less space than in a *Umsdos* partition.

8.3 Performance issue

The following point apply to *Umsdos* compared with *Ext2*.

- Directory management is faster on *Ext2*. This come from the overhead of the double directory structure of *Umsdos*.
- File access (reading and writing) is probably faster on *Umsdos* than *Ext2*. This come from the simplicity of the *FAT* file−system used by *DOS*. Beware that this simplicity come with a cost:
  - A maximum of around 65,000 files or clusters per partitions. This also means that a 500 megabytes partition will use cluster 16k large. In other word, a file containing a single byte will use 16k of disk storage.
  - Everything is controlled by the `FAT` located at the beginning of the hard drive. The *DOS* file−system is probably more fragile because of this.
  - No provision to avoid fragmentation of files. A *Umsdos* system will generally be used as a single user workstation. In this case, this does not matter much. As a multi−user engine, files will get spread−ed all around the drive, lowering file access performance.
- Symbolic links are stored in normal file. If you intend to have a lot of them, you will find that *Umsdos* use quite a lot of disk space compared to *Ext2*.