

Overview of Amiga Filesystems

Not all varieties of the Amiga filesystems are supported for reading and writing. The Amiga currently knows six different filesystems:

DOS\0	The old or original filesystem, not really suited for hard disks and normally not used on them, either. Supported read/write.
DOS\1	The original Fast File System. Supported read/write.
DOS\2	The old "international" filesystem. International means that a bug has been fixed so that accented ("international") letters in file names are case-insensitive, as they ought to be. Supported read/write.
DOS\3	The "international" Fast File System. Supported read/write.
DOS\4	The original filesystem with directory cache. The directory cache speeds up directory accesses on floppies considerably, but slows down file creation/deletion. Doesn't make much sense on hard disks. Supported read only.
DOS\5	The Fast File System with directory cache. Supported read only.

All of the above filesystems allow block sizes from 512 to 32K bytes. Supported block sizes are: 512, 1024, 2048 and 4096 bytes. Larger blocks speed up almost everything at the expense of wasted disk space. The speed gain above 4K seems not really worth the price, so you don't lose too much here, either.

The muFS (multi user File System) equivalents of the above file systems are supported, too.

Mount options for the AFFS

protect	If this option is set, the protection bits cannot be altered.
setuid[=uid]	This sets the owner of all files and directories in the file system to uid or the uid of the current user, respectively.
setgid[=gid]	Same as above, but for gid.
mode=mode	Sets the mode flags to the given (octal) value, regardless of the original permissions. Directories will get an x permission if the corresponding r bit is set. This is useful since most of the plain AmigaOS files will map to 600.
reserved=num	Sets the number of reserved blocks at the start of the partition to num. You should never need this option. Default is 2.
root=block	Sets the block number of the root block. This should never

affs.txt

be necessary.

bs=blksize	Sets the blocksize to blksize. Valid block sizes are 512, 1024, 2048 and 4096. Like the root option, this should never be necessary, as the affs can figure it out itself.
quiet	The file system will not return an error for disallowed mode changes.
verbose	The volume name, file system type and block size will be written to the syslog when the filesystem is mounted.
mufs	The filesystem is really a muFS, also it doesn't identify itself as one. This option is necessary if the filesystem wasn't formatted as muFS, but is used as one.
prefix=path	Path will be prefixed to every absolute path name of symbolic links on an AFFS partition. Default = "/". (See below.)
volume=name	When symbolic links with an absolute path are created on an AFFS partition, name will be prepended as the volume name. Default = "" (empty string). (See below.)

Handling of the Users/Groups and protection flags

Amiga -> Linux:

The Amiga protection flags RWEDRWEDHSPARWED are handled as follows:

- R maps to r for user, group and others. On directories, R implies x.
- If both W and D are allowed, w will be set.
- E maps to x.
- H and P are always retained and ignored under Linux.
- A is always reset when a file is written to.

User id and group id will be used unless set[gu]id are given as mount options. Since most of the Amiga file systems are single user systems they will be owned by root. The root directory (the mount point) of the Amiga filesystem will be owned by the user who actually mounts the filesystem (the root directory doesn't have uid/gid fields).

Linux -> Amiga:

The Linux rwxrwxrwx file mode is handled as follows:

- r permission will set R for user, group and others.
- w permission will set W and D for user, group and others.

affs.txt

- x permission of the user will set E for plain files.
- All other flags (suid, sgid, ...) are ignored and will not be retained.

Newly created files and directories will get the user and group ID of the current user and a mode according to the umask.

Symbolic links

Although the Amiga and Linux file systems resemble each other, there are some, not always subtle, differences. One of them becomes apparent with symbolic links. While Linux has a file system with exactly one root directory, the Amiga has a separate root directory for each file system (for example, partition, floppy disk, ...). With the Amiga, these entities are called "volumes". They have symbolic names which can be used to access them. Thus, symbolic links can point to a different volume. AFFS turns the volume name into a directory name and prepends the prefix path (see prefix option) to it.

Example:

You mount all your Amiga partitions under /amiga/<volume> (where <volume> is the name of the volume), and you give the option "prefix=/amiga/" when mounting all your AFFS partitions. (They might be "User", "WB" and "Graphics", the mount points /amiga/User, /amiga/WB and /amiga/Graphics). A symbolic link referring to "User:sc/include/dos/dos.h" will be followed to "/amiga/User/sc/include/dos/dos.h".

Examples

Command line:

```
mount Archive/Amiga/Workbench3.1.adf /mnt -t affs -o loop,verbose
mount /dev/sda3 /Amiga -t affs
```

/etc/fstab entry:

```
/dev/sdb5 /amiga/Workbench affs noauto,user,exec,verbose 0 0
```

IMPORTANT NOTE

If you boot Windows 95 (don't know about 3.x, 98 and NT) while you have an Amiga harddisk connected to your PC, it will overwrite the bytes 0x00dc..0x00df of block 0 with garbage, thus invalidating the Rigid Disk Block. Sheer luck has it that this is an unused area of the RDB, so only the checksum doesn't match anymore. Linux will ignore this garbage and recognize the RDB anyway, but before you connect that drive to your Amiga again, you must restore or repair your RDB. So please do make a backup copy of it before booting Windows!

If the damage is already done, the following should fix the RDB (where <disk> is the device name).

DO AT YOUR OWN RISK:

```
dd if=/dev/<disk> of=rdb.tmp count=1
cp rdb.tmp rdb.fixed
dd if=/dev/zero of=rdb.fixed bs=1 seek=220 count=4
dd if=rdb.fixed of=/dev/<disk>
```

Bugs, Restrictions, Caveats

=====

Quite a few things may not work as advertised. Not everything is tested, though several hundred MB have been read and written using this fs. For a most up-to-date list of bugs please consult fs/affs/Changes.

Filenames are truncated to 30 characters without warning (this can be changed by setting the compile-time option AFFS_NO_TRUNCATE in include/linux/amigaffs.h).

Case is ignored by the affs in filename matching, but Linux shells do care about the case. Example (with /wb being an affs mounted fs):
rm /wb/WrongCase
will remove /mnt/wrongcase, but
rm /wb/WR*
will not since the names are matched by the shell.

The block allocation is designed for hard disk partitions. If more than 1 process writes to a (small) diskette, the blocks are allocated in an ugly way (but the real AFFS doesn't do much better). This is also true when space gets tight.

You cannot execute programs on an OFS (Old File System), since the program files cannot be memory mapped due to the 488 byte blocks. For the same reason you cannot mount an image on such a filesystem via the loopback device.

The bitmap valid flag in the root block may not be accurate when the system crashes while an affs partition is mounted. There's currently no way to fix a garbled filesystem without an Amiga (disk validator) or manually (who would do this?). Maybe later.

If you mount affs partitions on system startup, you may want to tell fsck that the fs should not be checked (place a '0' in the sixth field of /etc/fstab).

It's not possible to read floppy disks with a normal PC or workstation due to an incompatibility with the Amiga floppy controller.

If you are interested in an Amiga Emulator for Linux, look at

<http://www.freiburg.linux.de/~uae/>