Filesystems HOWTO

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Filesystems HOWTO

Martin Hinner < mhi@penguin.cz>

Version 0.7.5, 22 August 2000

This small HOWTO is about filesystems and accessing filesystems. It is not Linux− or Unix−related document as you probably expect. You can find there also a lot of interesting information about non−Unix (file)systems, but Unix is my primary interest :−)

1.Introduction

2.Volumes

3.DOS FAT 12/16/32, VFAT

4.High Performance FileSystem (HPFS)

5.New Technology FileSystem (NTFS)

6.Extended filesystems (Ext, Ext2, Ext3)

7.Macintosh Hierarchical Filesystem − HFS

8.ISO 9660 − CD−ROM filesystem

9.Other filesystems

10.Raw partitions

11.Appendix

1.Introduction

The Filesystems HOWTO is about filesystems and accessing filesystems from various OS. Although this document has been put together to the best of my knowledge, it may and probably does contain mistakes. Please if you find some mistake or outdated information, let me know. I will try to keep this document up to date and as error free as possible. Any contributions are also welcome, so if you want to write anything about filesystems, please contact me via e−mail.

Before you read this HOWTO it's recommended to read Stein Gjoen's Disk−HOWTO (you can obtain it from http://sunsite.unc.edu/LDP/HOWTO/).

This HOWTO can be obtained from http://penguin.cz/~mhi/fs/ or http://metalab.unc.edu/filesystems/howto/.

If you are Japanese user, you might be interested that FUJIWARA Teruyoshi translated this HOWTO to Japanese. It is available at http://www.linux.or.jp/JF/JFdocs/Filesystems−HOWTO.html. SGML source file can be downloaded from ftp://ftp.linet.gr.jp/pub/JF/sgml/Filesystems−HOWTO.sgml.gz.

1.1 Copyright

1.2 Filesystems mailing−list

You may want to join Filesystems mailing list. It's intended to be a good source of information for both end−users and developers. So if you have anything to do with filesystems, join ;−) To subscribe send email to < majordomo@penguin.cz> and in the BODY (not the subject) of the email message put (without quotes): "**subscribe fs−l**".

Linux kernel filesystems mailing−list

To join Linux kernel filesystems mailing list linux−fsdev@vger.rutgers.edu, send e−mail to listserv@vger.rutgers.edu. Put `"subscribe linux-fsdev"` in message body.

FreeBSD filesystems mailing−list

To join techical FreeBSD filesystems mailing list freebsd−fs@FreeBSD.org, send e−mail to majordomo@FreeBSD.org. Put `"subscribe freebsd-fs"` in message body.

1.3 Filesystems collection at metalab.unc.edu

Filesystems collection is FTP/WWW site providing useful information about filesystems and filesystem−related programs and drivers. It lives at http://metalab.unc.edu/filesystems/, or FTP−only at ftp://metalab.unc.edu/pub/docs/filesystems/.

1.4 Credits

The original "Filesystems access HOWTO" was written by Georgatos Photis (see his homepage at http://students.ceid.upatras.gr/~gef/). This HOWTO contains a lot of information from his webpage. Thanks, Gef.

FUJIWARA Teruyoshi <fujiwara@linux.or.jp> translated this HOWTO to Japanese.

Other people who have contributed or helped me (directly or indirectly) with this HOWTO are, in alphabetical order:

- Mariusz Borkowski <borkowsm@ii.pw.edu.pl> – ISO9660/RR info

- Remy Card <card@masi.ibp.fr> – Ext2 filesystem introduction
- Peter A. Dinda <pdinda@cs.cmu.edu> – HFS filesystem description
- Alfonso De Gregorio <adg@speedcom.it> – TCFS filesystem info
- Radek Machacka <radekm@sco.com> – Thanks for SCO UnixWare and SCO OpenServer
- Andrey Shedel <andreys@cr.cyco.com> – Misc. updates
- Peter Todd <retep2@home.com> – SFS filesystem info
- Theodore Ts'o <tytso@mit.edu> – Ext2 filesystem introduction
- Stephen Tweedie <sct@dcs.ed.ac.uk> – Ext2 filesystem introduction

Many thanks to the above people. If I have forgotten anyone, please let me know.

1.5 Filesystems accessibility map

This is filesystem accessibility "map", alphabetically ordered by operating system. You may find this list a little bit chaotic. It's because Linux sgmltools don't know tables.

YOU SEE THAT THIS `MAP' IS NOT STILL COMPLETE. I WILL TRY TO FINISH IT IN THE NEAR FUTURE.

FreeBSD:BSD FFS | Ext2 | HPFS | NTFS

Linux:AFFS| BeFS| BFS| Ext2 FS| BSD FFS| HPFS| Qnx4 FS| Xia

NetBSD:BSD FFS | FAT12/16 | ISO9660

NetWare 2.x:NWFS–286

NetWare 3.x, 4.x:NWFS–386 | ISO9660

NetWare 5.x:NWFS–386 | NSS | ISO9660

OpenBSD:BSD FFS | FAT12/16

OS/2: Ext2 FS | FAT12/16/32 | HPFS | HPFS | ISO 9660 | JFS | VFAT

QNX 4:FAT12/16 | ISO 9660 | Qnx4 FS

SCO OpenServer:AFS| DTFS| EAFS| HTFS| ISO 9660 | S51K

SCO UnixWare:BFS| DTFS| ISO 9660 | System V| VxFS

1.6 Introduction to contiguous allocation filesystems

Some contiguous filesystems: BFS, ISO9660 and extensions.

1.7 Introduction to linked−list allocation filesystems

1.8 Introduction to FAT−based filesystems

(todo)

Some FAT filesystems: FAT12/16/32, VFAT and NetWare filestem.

1.9 Introduction to Inode filesystems

(todo)

1.10 Introduction to extent filesystems

(todo)

Some 'extent' filesystems: EFS and VxFS.

1.11 Introduction to filesystems using balanced trees

(todo)

Some filesystems which use B+ trees: HFS, NSS, Reiser FS and Spiralog filesystem.

1.12 Introduction to logging/journaling filesystems

File systems update their structural information (called metadata) by synchronous writes. Each metadata update may require many separate writes, and if the system crashes during the write sequence, metadata may be in inconsistent state.

At the next boot the filesystem check utility (called fsck) must walk through the metadata structures, examining and repairing them. This operation takes a very very long time on large filesystems. And the disk may not contain sufficient information to correct the structure. This results in misplaced or removed files.

A journaling file system uses a separate area called a log or journal. Before metadata changes are actually performed, they are logged to this separate area. The operation is then performed. If the system crashes during the operation, there is enough information in the log to "replay" the log record and complete the operation.

This approach does not require a full scan of the file system, yielding very quick filesystem check time on large file systems, generally a few seconds for a multiple−gigabyte file system. In addition, because all information for the pending operation is saved, no removals or lost−and−found moves are required. Disadvantage of journaling filesystems is that they are slower than other filesystems.

Some journaling filesystems: BeFS, HTFS, JFS, NSS, Spiralog filesystem, VxFS and XFS.

1.13 Other filesystem features

Quota

Snapshot

ACLs

2.Volumes

2.1 PC Partitions

- http://www.win.tue.nl/~aeb/partitions/ Partition types document by Andries Brouwer < aeb@cwi.nl>

GNU parted

- Homepage: http://www.gnu.org/software/parted
- Download: ftp://ftp.gnu.org/gnu/parted/
- Authors: Andrew Clausen < clausen@gnu.org>, Lennert Buytenhek < buytenh@dsv.nl> and Matt Wilson < msw@redhat.com>.
- Bug reports: < bug−parted@gnu.org>,
- Access: varies for each filesystem, see below.
- License: GPL

GNU Parted is a program for creating, destroying, resizing, checking and copying partitions, and the filesystems on them.

This is useful for creating space for new operating systems, reorganising disk usage, copying data between hard disks, and "disk imaging" − replicating installations over many computers.

Parted has support for these operations:

Filesystem	detect	create	resize	copy	check
ext2	*	*	*1	*2	*3
fat	*	*	*4	*4	*
linux−swap	*	*	*	*	*

NOTES:

(1) The start of the partition must stay fixed for ext2.

(2) The partition you copy to must be bigger (or exactly the same size) as the partition you copy from.

(3) Limited checking is done when the filesystem is opened. This is the only checking at the moment. All commands (including resize) will gracefully fail, leaving the filesystem in tact, if there are any errors in the file system (and the vast majority of errors in general).

(4) The size of the new partition, after resizing or copying, is restricted by the cluster size for fat (mainly affects FAT16). This is worse than you think, because you don't get to choose your cluster size (it's a bug in Windows, but you want compatibility, right?).

So, in practise, you can always shrink your partition (because Parted can shrink the cluster size), but you may not be able to grow the partition to the size you want. If you don't have any problems with using FAT32, you will always be able to grow the partition to the size you want.

Summary: you can always shrink your partition. If you can't use FAT32 for some reason, you may not be able to grow your partition.

Repairing corrupted partition table

Fixdisktable

- Homepage: http://bmrc.berkeley.edu/people/chaffee/fat32.html
- Download: ?
- Author: ?
- Access: ?
- License: ?

This is a utility that handles ext2, FAT, NTFS, ufs, BSD disklabels (but not yet old Linux swap partitions); it actually will rewrite the partition table, if you give it permission.

gpart

- Homepage: http://home.pages.de/~michab/gpart/
- Download: ?
- Author: ?
- Access: ?
- License: ?

GPART is a utility that handles ext2, FAT, Linux swap, HPFS, NTFS, FreeBSD and Solaris/x86 disklabels, minix, reiser fs; it prints a proposed contents for the primary partition table, and is well−documented.

rescuept

- Homepage: util−linux ?
- Download: ?

- Author: ?
- Access: ?
- License: ?

Recognizes ext2 superblocks, FAT partitions, swap partitions, and extended partition tables; it may also recognize BSD disklabels and Unixware 7 partitions. It prints out information that can be used with fdisk or sfdisk to reconstruct the partition table. It is in the non−installed part of the util−linux distribution.

findsuper

- Homepage: e2progs ?
- Download: ?
- Author: ?
- Access: ?
- License: ?

Small utility that finds blocks with the ext2 superblock signature, and prints out location and some info. It is in the non−installed part of the e2progs distribution.

2.2 Other partitions

Because I use **only** Intel x86 machines, any contributions (or non−x86 machine donation ;−)) are **very** welcome. If you can provide any useful information, don't hesitate to mail me.

ADFS partitions

Amiga partitions

ATARI partitions

Macintosh partitions

OSF partitions

Sun partitions

Ultrix partitions

2.3 Unix disklabels

(todo)

BSD disklabel

(todo)

UnixWare disklabel

UnixWare VTOC (Volume Table Of Contents) divides disk partition to 16 logical partitions. Linux kernel supports UnixWare VTOC, you must check "UnixWare slices support (EXPERIMENTAL)" and recompile your kernel. Another way of reading UnixWare disklabel is using GPL port of prtvtoc(1) command, which is in vxtools package.

SCO OpenServer disklabel

(todo)

Sun Solaris disklabel

(todo)

2.4 Windows NT volumes

- Homepage: http://www.penguin.cz/~mhi/fs/vol/
- Author: Martin Hinner < mhi@penguin.cz>
- Access: Read−only, supports OS/2 Volumes, Windows NT Stripe sets and volumes.
- Download: ftp://ftp.penguin.cz/pub/users/mhi/vol/
- License: GPL

This linux−kernel driver allows you to access and mount linear and stripe set volumes.

Repairing "fault tolerant" NTFS disks using FTEdit

- Homepage: ? MS ARTICLE ID: Q131658
- Download: ftp://ftp.rhrz.uni−bonn.de/pub/pc/winnt/intel/ftedit.zip
- Author: Microsoft Corp.
- License: ?

If you have a Windows NT Workstation or Server configured for fault tolerant (FT) partitions (such as stripes with parity and volume sets), and those partitions are inaccessible and appear in Disk Administrator as type Unknown, you can possibly make them accessible again by using the utility FTEDIT.

2.5 MD − Multiple Devices driver for Linux

- Homepage:?
- Author: Marc Zyngier < maz@wild−wind.fr.eu.org>
- Access: Read−write, supports linear mode, RAID−1, RAID−4 and RAID−5.
- Download: Linux kernel, tools are available at ftp://sweet−smoke.ufr−info−p7.ibp.fr/public/Linux/
- License: GPL

This driver lets you combine several hard disk partitions into one logical block device. This can be used to simply append one partition to another one or to combine several redundant hard disks to a RAID1/4/5 device so as to provide protection against hard disk failures. This is called "Software RAID" since the combining of the partitions is done by the kernel.

2.6 LVM − Logical Volume Manager (HP−UX LVM?)

Linux implementation is available here:

- Homepage: http://linux.msede.com/lvm/
- Author: Heinz Mauelshagen < mauelsha@ez−darmstadt.telekom.de>
- Access: ?
- Download: ftp://linux.msede.com/lvm/v0.6/
- License: GPL

2.7 VxVM − Veritas Volume Manager

For more information about Veritas Volume Manager see http://www.veritas.com/.

See also: VxFS (Veritas Journaling Filesystem).

2.8 IBM OS/2 LVM

Logical Volume Manager is available in OS/2 WarpServer 5. It allows you to create linear volumes on several disks/partitions. Some people say that it's compatible with IBM AIX Logical Volume Manager.

See also: HPFS, JFS

2.9 StackVM

StackVM is CrosStor's volume manager. Using StackVM the administrator can combine multiple physical disk slices into a single logical device know as a vdisk. Vdisk is short for virtual disk. The physical disks can be combined to form a concatenation, RAID 0 (stripe), RAID 1 (mirror), RAID 4 or RAID 5. In addition a single disk partition can be subdivided into multiple simple vdisks. For more information see CrosStor homepage at http://www.crosstor.com/.

2.10 Novell NetWare volumes

NetWare volumes are used for NWFS−386 filesystem.

3.DOS FAT 12/16/32, VFAT

3.1 VFAT: Long filenames

Windows 95/98 and Windows NT/2000 store long filenames on FAT in special directory entries with set attributes **ReadOnly**, **Hidden**, **System** and **Volume**, so if you access FAT volume from DOS you don't see these "files". These special entries have this mad structure:

```
byte              sequence number for slot
string(10)        first 5 characters in name
byte              attribute byte
byte              always 0
byte              checksum for 8.3 alias
string(12)        6 more characters in name
word              starting cluster number, 0 in long slots
string(4)         last 2 characters in name
```

Problem occur when you delete or modify file with long name from system without VFAT support, because only DOS 8+3 entry will be deleted or modified. Scandisk from Windows 95/98 can repair this problem.

3.2 UMSDOS: Linux LFN/attributes on FAT filesystem

Linux has it's own FAT extensions which gives you long filenames, permissions and owners, links and special devices on FAT partition, called UMSDOS. Each directory contains file named **"−−linux−.−−−"**. There are stored long names and other necessary fields. For more information see file **/usr/src/linux/Documentation/filesystems/umsdos.txt**. Author of Linux umsdos driver is Jacques Gelinas < jacques@solucorp.qc.ca> and it is currently maintained by Matija Nalis < mnalis@jagor.srce.hr>.

3.3 OS/2 Extended Attributes on FAT filesystems

OS/2 Warp version 3,4 and 5 stores long filenames and extended attributes on FAT volume in files "\ea data. sf" and "\wp root. sf" (both files are in root directory of filesystem). AFAIK there is no known implementation of OS/2 EAs for any other OS. If you can supply any information about EA structure, don't hesitate to mail them to me.

3.4 Star LFN

Star LFN is an emulator that allows programs, running under DOS 4.0 or above, to use the long filename functions present in Windows'95 DOS boxes. Currently, it can only read and write long filenames from and into a system+hidden file, which means you can't either read or write real Windows'95 long filenames. For more information see http://sta.c64.org/starlfn.html.

3.5 Accessing VFAT from OS/2 (VFAT−OS2)

- Homepage: http://www.dsteiner.com/products/software/os2/ifs.htm
- Author: Daniel Steiner < info@dsteiner.com>
- Access: Read−Write, no EAs supported.
- Mirror: ftp://hobbes.nmsu.edu/pub/os2/system/drivers/filesys/
- License: GPL

VFAT−OS2 is a package that will allow OS/2 to seamlessly access Windows 95 VFAT formatted partitions from OS/2 as if they were standard OS/2 drive letters. The ultimate aim of this package is to be able to use the VFAT file system as a replacement of FAT. It can now also access NTFS partitions in read−only mode.

3.6 Accessing VFAT from DOS (LFNDOS driver)

Some people say that Microsoft has released a driver called LFNDOS that provides the Microsoft Long Filename API under DOS. If you know where can this driver be downloaded, send me e−mail please.

3.7 Accessing VFAT from DOS (Free LFNDOS driver)

- Homepage: http://members.xoom.com/dosuser/
- Author: Chris Jones < dosuser@bigfoot.com>
- Access: Read−Write
- Mirror: http://www.simtel.net/pub/simtelnet/msdos/fileutil/lfnds106.zip
- License: Free, source code available

LFNDOS provides the Windows95 Long Filename (LFN) API to DOS programs. It uses the same format for storing the names on disk as Windows95 does, so you can view and use long filenames under both systems interchangeably. It runs as a memory−resident program, and while resident requires about 60k of conventional memory.

Under Windows95, a DOS program can use long filenames by calling a set of interrupt functions, which Windows provides. For example, COMMAND.COM will allow long filenames when run as a DOS Prompt from Windows, but not if you restart in MS−DOS mode. Other programs such as EDIT.COM and all DJGPP programs use long filenames if available.

3.8 Accessing VFAT from DOS (Odi's LFN tools)

- Homepage: http://odi.webjump.com/
- Author: Ortwin Glueck < glueck@freesurf.ch>
- Access: Read−Write, only DOS utilities
- Mirror: http://www.simtel.net/pub/simtelnet/msdos/fileutil/lfn141.zip
- License: ?

These tools provide easy file management under DOS with long filenames created by Windows 95/98 on FAT32, FAT16 and FAT12 file systems. Typing LDIR brings up the directory with its long filenames. Copying a file with LCOPY preserves long filenames. You can even create directories (LMD) with long names or rename files (LREN) with long names.

3.9 Accessing FAT32 from OS/2 (FAT32.IFS)

- Homepage: http://www.os2ss.com/information/kelder/index.html
- Author: Henks Kelder < hkelder@capgemini.nl >
- Access: Read−Write, long filenames, no EAs support.
- Download: http://www.os2ss.com/information/kelder/os2fat32.zip
- License: Free

FAT32.IFS for OS/2 will allow you to access FAT32 partitions from OS/2. You cannot create FAT32 partitions, you'll still need Win95 OSR2 to do that. Also, OS/2s CHKDSK cannot fix all possible errors that can occur, you'll have to use Windows 95 Scandisk to fix certain errors.

3.10 Accessing FAT32 from Windows NT 4.0

- Download: http://www.chat.ru/~ashedel/fat32/fastfat32.rar
- Author: Anonymous
- License: Free or GPL ?

FAT32 filesystem driver for NT 4.0 and NT 3.51.

3.11 Accessing FAT32 from Windows NT 4.0

- Homepage: http://www.sysinternals.com/fat32.htm
- Author: Mark Russinovich < mark@sysinternals.com> and Bryce Cogswell < cogswell@winternals.com>.
- Access: Read−only in free version, RW in commercial.
- Download: ?
- License: Free(read−only) or Commercial(read−write)

This is a FAT32 file system driver for Windows NT(R) 4.0. Once installed, any FAT32 drives present on your system will be fully accessible as native Windows NT volumes. Free version provides read−only capabilities. A read/write version is for sale.

3.12 Accessing Stac/Dblspaced/Drvspaced drives from Linux (DMSDOS)

- Homepage: http://fb9nt.uni−duisburg.de/mitarbeiter/gockel/software/dmsdos/
- Author: Frank Gockel < gockel@sent13.uni−duisburg.de> and Pavel Pisa < pisa@cmp.felk.cvut.cz>
- Access: Stacker, Dblspace and Drvspace in Read−Write mode, long filenames.
- Download: ftp://fb9nt.uni−duisburg.de/pub/linux/dmsdos/
- Freshmeat: Console/Filesystems
- License: GPL

DMSDOS reads and writes compressed DOS filesystems (CVF−FAT). The following configurations are supported:

- DoubleSpace / DriveSpace (MS−DOS 6.x)
- DoubleSpace / DriveSpace (Windows 95)
- DriveSpace 3 (Windows 95 with Plus! pack)
- Stacker 3
- Stacker 4

It works with FAT32, NLS, codepages (tested with fat32 patches version 0.2.8 under Linux 2.0.33 and with fat32 in standard 2.1.xx kernels and 2.0.34+35). Dmsdos can run together with vfat or umsdos for long filenames. It has been redesigned to be ready for SMP and should now compile completely under libc6.

3.13 Accessing Dblspaced/Drvspaced drives from Linux (thsfs)

- Download: ftp://ftp.ai−lab.fh−furtwangen.de/pub/os/linux/local/thsfs.tgz
- Author: Thomas Scheuermann < ths@ai−lab.fh−furtwangen.de>
- Access: Dblspace and Drvspace in Read−only mode.

- License: See copyright on files. Basically free

3.14 Fsresize − FAT16/32 resizer

- Homepage: http://www.alphalink.com.au/~clausen/fsresize/
- Author: Andrew Clausen < clausen@alphalink.com.au>
- Download: http://www.alphalink.com.au/~clausen/fsresize−0.8.tar.gz
- Freshmeat: Console/Filesystems
- Access: Read/Write, full FAT16/FAT32 support
- License: GPL

Resizes FAT16/FAT32 filesystems. It doesn't require any other programs (like a defrager). It has −−backup and −−restore options, so if there's a power failure, (or a bug), you can always go back. The backup files are usually < 1 meg.

The author probably won't be releasing any more versions of fsresize, because he is working on parted − a Partition Magic clone. It will be able to resize, copy, create and check filesystems/partitions.

3.15 FIPS − FAT16 resizer

- Homepage: ?
- Author: Arno Schaefer < schaefer@rbg.informatik.th−darmstadt.de>
- Download: ftp://sunsite.unc.edu/pub/Linux/system/Install/fips01alpha.tar.z
- License: GPL

4.High Performance FileSystem (HPFS)

Good HPFS links:

- ftp://ftp.leo.org/pub/comp/os/os2/leo/doc/hpfsinf.zip
- ftp://hobbes.nmsu.edu/pub/os2/info/tips/hpfs.zip
- http://www.globalxs.nl/home/c/cyborg/index.html − a good page about HPFS accessibility
- http://www−4.ibm.com/software/os/warp/warp−server/warp−server−adv/c2j.html − IBM OS/2 Warp Server : Features & Benefits : File & Print

4.1 Accessing HPFS from DOS (iHPFS)

- Homepage: http://www.student.nada.kth.se/~f96−bet/ihpfs/
- Author: Marcus Better Marcus.Better@abc.se
- Download: http://www.student.nada.kth.se/~f96−bet/ihpfs/ihpfs128.zip
- Access: Read−only
- License: GPL

iHPFS makes possible for OS/2 users to use their HPFS partitions when they boot plain DOS. The HPFS partition is assigned a drive letter, and can be accessed like any DOS drive.iHPFS is restricted to read−only access.

This program is no longer being developed, because author doesn't use OS/2. If you are willing to maintain the program, let him know.

4.2 Accessing HPFS from DOS (hpfsdos)

- Homepage: ?
- Author: Robert Muchsel < rmuchsel@iiic.ethz.ch> (this e−mail doesn't work)
- Access: Read−only
- License: Shareware ($23)

4.3 Accessing HPFS from DOS (hpfsa)

- Homepage: http://www.student.informatik.th−darmstadt.de/~akinzler/
- Author: Andreas Kinzler < akinzler@rbg.informatik.th−darmstadt.de> (this email doesn't work)
- Download: ftp://ftp.cdrom.com/.1/os2/mdos/hpfsa102.zip
- Access: Read/Write
- License: Shareware ($40)

4.4 Accessing HPFS from DOS (amos)

- Homepage: ?
- Author: Allan Mertner < mertner@login.dknet.dk> (this email doesn't work)
- Download: ftp://hobbes.nmsu.edu/pub/dos/amos320.zip
- License: Shareware ($50)

4.5 Accessing HPFS from Linux

- Homepage: http://artax.karlin.mff.cuni.cz/~mikulas/vyplody/hpfs/index−e.cgi
- Download: http://artax.karlin.mff.cuni.cz/~mikulas/vyplody/hpfs/hpfs−0.99b.tar.gz for 2.0 kernels; and http://artax.karlin.mff.cuni.cz/~mikulas/vyplody/hpfs/hpfs−1.98b.tar.gz for 2.2 kernels
- Author: Mikulas Patocka < mikulas@artax.karlin.mff.cuni.cz >
- Access: Read−Write, extended attributes, long names.
- License: GPL

This driver is part of Linux kernel (2.1.x+). It can read and write to HPFS partions. Access rights and owner can be stored in extended attributes. Few bugs in original read−only HPFS are corrected. It supports HPFS386 on Warp Server Advanced.

If you have kernel with HPFS support, say "Y"es to 'OS/2 HPFS filesystem support' in Filesystems submenu. Then recompile kernel using 'make dep bzImage', reboot and try to mount your HPFS partition (e.g. mount /dev/hda2 /mnt −t hpfs).

4.6 Accessing HPFS from FreeBSD

- Homepage: http://iclub.nsu.ru/~semen/
- Download: http://iclub.nsu.ru/~semen/hpfs/hpfs−0.3b.tar.gz
- Author: Semen A. Ustimenko < semenu@FreeBSD.org >
- Access: Read/Only
- License: BSD

Driver allows to mount HPFS volume into Unix namespace. ReadOnly access is only supported for now.

4.7 Accessing HPFS from Windows NT 3.5

- Homepage: http://www.htc.net/~nbehnken/
- Download: http://www.htc.net/~nbehnken/hpfs_nt.zip
- Author: Chris Behnken < nbehnken@htc.net >
- License: Freeware

This program will edit the Windows NT registry and enable HPFS support. Pinball.sys is the HPFS filesystem driver for Windows NT. It can be found on NT 3.5x's CD−ROM. Microsoft no longer supports HPFS. Installing this program will void your warranty and possibly the license agreement.

4.8 Accessing HPFS from Windows NT 4

- Download: ftp://hobbes.nmsu.edu/pub/windows/hpfsnt.zip
- Author: ?
- License: ?

HPFS driver for Windows NT 4.0

5.New Technology FileSystem (NTFS)

References:

- http://www.microsoft.com/msj/1198/ntfs/ntfstop.htm NTFS 5 information
- Rajeev Nagar, Windows NT File System Internals (O'Reilly).
- Helen Custer, Inside the Windows NT File System, ISBN: 1−55615−660−X.
- NTFS documentation by Regis Duchesne < regis@via.ecp.fr>, http://www.via.ecp.fr/~regis/ntfs.tar.bz2 or http://celine.via.ecp.fr/~regis/ntfs/new
- Microsoft TechNet, February 97, Windows NT Training: Support, NTFS
- http://www.stat.math.ethz.ch/~maechler/NTFS−docu

5.1 Accessing NTFS from DOS (NTFSDOS.EXE)

- Homepage: http://www.sysinternals.com/ntfs20.htm
- Authors: Mark Russinovich < mark@sysinternals.com> and Bryce Cogswell < cogswell@winternals.com>.
- Access: Read−only, Long filenames under DOS 7 and Win9x.

NTFSDOS.EXE is a network file system redirector for DOS/Windows that is able to recognize and mount NTFS drives for transparent access. It makes NTFS drives appear indistinguishable from standard FAT drives, providing the ability to navigate, view and execute programs on them from DOS or from Windows, including from the Windows 3.1 File Manager and Windows 95 Explorer.

5.2 Accessing NTFS from DOS (ntpwd)

- Homepage: http://www.esiea.fr/public_html/Christophe.GRENIER/
- Author: Grenier Christophe < grenier@nef.esiea.fr >
- Access: Read−only (rw experimental), long filenames supported, no driver letter (dos tools)

- License: GPL

NTPwd contains command line tools to access NTFS partition, it'a a Dos port of the driver used by Linux. It contains too a little utility to change NT password.

5.3 Accessing NTFS from OS/2

- Homepage: http://www.dsteiner.com/products/software/os2/ifs.htm
- Mirror: ftp://ftp−os2.nmsu.edu/pub/os2/system/drivers/filesys/ntfs_003.zip, ftp://ftp.leo.org/pub/comp/os/os2/leo/drivers/ifs
- Author: Daniel Steiner < info@dsteiner.com >
- Access: Read−only, Long filenames supported

ntfs_003.zip archive contains only command line tools to acccess a NTFS partition in OS/2. A true IFS for accessing NTFS is included in VFAT−OS2 v0.05.

5.4 Accessing NTFS from Linux

- Homepage: http://www.informatik.hu−berlin.de/~loewis/ntfs/
- Author: Martin von Löwis loewis@informatik.hu−berlin.de
- Freshmeat: Console/Filesystems
- Homepage: http://www.informatik.hu−berlin.de/~loewis/ntfs/ntfs−current.tgz
- Mirror: Included in official Linux kernel
- Access: RO, experimental RW, compression, no encryption
- License: GPL

Works both as a kernel driver, as well as a set of command line utilities.

5.5 Accessing NTFS from FreeBSD and NetBSD

- Homepage: http://iclub.nsu.ru/~semen/ntfs/
- Author: Semen A. Ustimenko < semenu@FreeBSD.org>
- Download: As part of FreeBSD (ftp://ftp.FreeBSD.org/pub/FreeBSD/), and NetBSD (ftp://ftp.NetBSD.org/pub/NetBSD/)
- Mirror: Lookup for FreeBSD's and NetBSD's mirrors
- Access: Read + limited writing, doesn't support codepages
- License: BSD

Driver allows to mount NTFS volumes under FreeBSD and NetBSD. We also support limited writing ability: you can write into not comressed files without holes, but you can't change the size of file yet. Write support was made to swap on NTFS volume.

5.6 Accessing NTFS from BeOS

- Homepage: http://www.cs.tamu.edu/people/tkg0143/be/
- Author: Travis Geiselbrecht < geist@tamu.edu >
- Download: http://www.cs.tamu.edu/people/tkg0143/be/downloads/ntfs−0.05−x86−r4.zip
- Access: ?
- License: Free

This is a ALPHA version of a NTFS driver for BeOS. It is not the most polished thing in the world, but every release that author puts out is more stable than the last. He just implemented compressed file reads, so be careful with those. He also finally worked with NTFS 5 volumes, and managed to root out a few bugs.

Author now works for Be Inc, so you will not see his NTFS and ext2 filesystem support updated on the web much more. The drivers will be pulled into future BeOS releases.

5.7 Accessing NTFS from BeOS (another)

- Homepage: http://www.sw.com.sg/solutions/ntfs−ro.shtml
- Author: Standard & Western Software, http://www.sw−soft.com
- Download: http://download.sw.com.sg/pub/Be/ntfs−rod−0302.tar.gz
- Access: Read−only.

5.8 Repairing NTFS using NTFSDOS Tools

- Homepage: http://www.sysinternals.com/
- Author: Winternals Software < info@winternals.com>
- Access: Read−Write: Copy and replace files.
- License: Commercial

An add−on to NTFSDOS that allows one to rename existing files, or to overwrite a file with new data. Very limited functionality.

5.9 Repairing NTFS using NTRecover

- Homepage: http://www.sysinternals.com/
- Author: Winternals Software < info@winternals.com>
- Access: Freeware version is read−only, commercial version is read/write.
- License: Freeware read−only version, commercial read/write version

Uses a boot floppy and a serial connection to a second NT system to provide full access to a NTFS drives on dead NT systems. Ideal for salvaging data or replacing drivers.

6.Extended filesystems (Ext, Ext2, Ext3)

Extended filesystem (ext fs), second extended filesystem (ext2fs) and third extended filesystem (ext3fs) were designed and implemented on Linux by Rémy Card, Laboratoire MASI−−Institut Blaise Pascal, < card@masi.ibp.fr>, Theodore Ts'o, Massachussets Institute of Technology, < tytso@mit.edu> and Stephen Tweedie, University of Edinburgh, < sct@redhat.com>

- http://web.mit.edu/tytso/www/linux/ext2.html − The ext2 homepage. This is the primary source of information about ext2.
- http://uranus.it.swin.edu.au/~jn/explore2fs/es2fs.htm − Document about ext2fs from John Newbigin.
- http://www.ing.umu.se/~bosse/ − Ext2fs_Rec (ext2 recognizer for WinNT).

6.1 Extended filesystem (ExtFS)

This is old filesystem used in early Linux systems.

6.2 Second Extended Filesystem (Ext2 FS)

The Second Extended File System is probably the most widely used filesystem in the Linux community. It provides standard Unix file semantics and advanced features. Moreover, thanks to the optimizations included in the kernel code, it is robust and offers excellent performance.

Since Ext2fs has been designed with evolution in mind, it contains hooks that can be used to add new features. Some people are working on extensions to the current filesystem: access control lists conforming to the Posix semantics, undelete, and on−the−fly file compression.

Ext2fs was first developed and integrated in the Linux kernel and is now actively being ported to other operating systems. An Ext2fs server running on top of the GNU Hurd has been implemented. People are also working on an Ext2fs port in the LITES server, running on top of the Mach microkernel and in the VSTa operating system. Last, but not least, Ext2fs is an important part of the Masix operating system, currently under development by one of the authors.

Motivations

The Second Extended File System has been designed and implemented to fix some problems present in the first Extended File System. Our goal was to provide a powerful filesystem, which implements Unix file semantics and offers advanced features.

Of course, we wanted to Ext2fs to have excellent performance. We also wanted to provide a very robust filesystem in order to reduce the risk of data loss in intensive use. Last, but not least, Ext2fs had to include provision for extensions to allow users to benefit from new features without reformatting their filesystem.

``Standard'' Ext2fs features

The Ext2fs supports standard Unix file types: regular files, directories, device special files and symbolic links.

Ext2fs is able to manage filesystems created on really big partitions. While the original kernel code restricted the maximal filesystem size to 2 GB, recent work in the VFS layer have raised this limit to 4 TB. Thus, it is now possible to use big disks without the need of creating many partitions.

Ext2fs provides long file names. It uses variable length directory entries. The maximal file name size is 255 characters. This limit could be extended to 1012 if needed.

Ext2fs reserves some blocks for the super user (`root`). Normally, 5% of the blocks are reserved. This allows the administrator to recover easily from situations where user processes fill up filesystems.

``Advanced'' Ext2fs features

In addition to the standard Unix features, Ext2fs supports some extensions which are not usually present in Unix filesystems.

File attributes allow the users to modify the kernel behavior when acting on a set of files. One can set attributes on a file or on a directory. In the later case, new files created in the directory inherit these attributes.

BSD or System V Release 4 semantics can be selected at mount time. A mount option allows the administrator to choose the file creation semantics. On a filesystem mounted with BSD semantics, files are created with the same group id as their parent directory. System V semantics are a bit more complex: if a directory has the setgid bit set, new files inherit the group id of the directory and subdirectories inherit the group id and the setgid bit; in the other case, files and subdirectories are created with the primary group id of the calling process.

BSD−like synchronous updates can be used in Ext2fs. A mount option allows the administrator to request that metadata (inodes, bitmap blocks, indirect blocks and directory blocks) be written synchronously on the disk when they are modified. This can be useful to maintain a strict metadata consistency but this leads to poor performances. Actually, this feature is not normally used, since in addition to the performance loss associated with using synchronous updates of the metadata, it can cause corruption in the user data which will not be flagged by the filesystem checker.

Ext2fs allows the administrator to choose the logical block size when creating the filesystem. Block sizes can typically be 1024, 2048 and 4096 bytes. Using big block sizes can speed up I/O since fewer I/O requests, and thus fewer disk head seeks, need to be done to access a file. On the other hand, big blocks waste more disk space: on the average, the last block allocated to a file is only half full, so as blocks get bigger, more space is wasted in the last block of each file. In addition, most of the advantages of larger block sizes are obtained by Ext2 filesystem's preallocation techniques.

Ext2fs implements fast symbolic links. A fast symbolic link does not use any data block on the filesystem. The target name is not stored in a data block but in the inode itself. This policy can save some disk space (no data block needs to be allocated) and speeds up link operations (there is no need to read a data block when accessing such a link). Of course, the space available in the inode is limited so not every link can be implemented as a fast symbolic link. The maximal size of the target name in a fast symbolic link is 60 characters. We plan to extend this scheme to small files in the near future.

Ext2fs keeps track of the filesystem state. A special field in the superblock is used by the kernel code to indicate the status of the file system. When a filesystem is mounted in read/write mode, its state is set to ``Not Clean''. When it is unmounted or remounted in read−only mode, its state is reset to ``Clean''. At boot time, the filesystem checker uses this information to decide if a filesystem must be checked. The kernel code also records errors in this field. When an inconsistency is detected by the kernel code, the filesystem is marked as ``Erroneous''. The filesystem checker tests this to force the check of the filesystem regardless of its apparently clean state.

Always skipping filesystem checks may sometimes be dangerous, so Ext2fs provides two ways to force checks at regular intervals. A mount counter is maintained in the superblock. Each time the filesystem is mounted in read/write mode, this counter is incremented. When it reaches a maximal value (also recorded in the superblock), the filesystem checker forces the check even if the filesystem is ``Clean''. A last check time and a maximal check interval are also maintained in the superblock. These two fields allow the administrator to request periodical checks. When the maximal check interval has been reached, the checker ignores the filesystem state and forces a filesystem check.

An attribute allows the users to request secure deletion on files. When such a file is deleted, random data is written in the disk blocks previously allocated to the file. This prevents malicious people from gaining access to the previous content of the file by using a disk editor.

Last, new types of files inspired from the 4.4 BSD filesystem have recently been added to Ext2fs. Immutable files can only be read: nobody can write or delete them. This can be used to protect sensitive configuration files. Append−only files can be opened in write mode but data is always appended at the end of the file. Like immutable files, they cannot be deleted or renamed. This is especially useful for log files which can only grow.

Physical Structure

The physical structure of Ext2 filesystems has been strongly influenced by the layout of the BSD filesystem. A filesystem is made up of block groups. Block groups are analogous to BSD FFS's cylinder groups. However, block groups are not tied to the physical layout of the blocks on the disk, since modern drives tend to be optimized for sequential access and hide their physical geometry to the operating system.

```
,---------+---------+---------+---------+---------,
| Boot    | Block   | Block   |   ...   | Block   |
| sector  | group 1 | group 2 |         | group n |
`---------+---------+---------+---------+---------'
```

Each block group contains a redundant copy of crucial filesystem control informations (superblock and the filesystem descriptors) and also contains a part of the filesystem (a block bitmap, an inode bitmap, a piece of the inode table, and data blocks). The structure of a block group is represented in this table:

```
,---------+---------+---------+---------+---------+---------,
| Super   | FS      | Block   | Inode   | Inode   | Data    |
| block   | desc.   | bitmap  | bitmap  | table   | blocks  |
`---------+---------+---------+---------+---------+---------'
```

Using block groups is a big win in terms of reliability: since the control structures are replicated in each block group, it is easy to recover from a filesystem where the superblock has been corrupted. This structure also helps to get good performances: by reducing the distance between the inode table and the data blocks, it is possible to reduce the disk head seeks during I/O on files.

In Ext2fs, directories are managed as linked lists of variable length entries. Each entry contains the inode number, the entry length, the file name and its length. By using variable length entries, it is possible to implement long file names without wasting disk space in directories.

Performance optimizations

In Linux, the Ext2fs kernel code contains many performance optimizations, which tend to improve I/O speed when reading and writing files.

Ext2fs takes advantage of the buffer cache management by performing readaheads: when a block has to be read, the kernel code requests the I/O on several contiguous blocks. This way, it tries to ensure that the next block to read will already be loaded into the buffer cache. Readaheads are normally performed during sequential reads on files and Ext2fs extends them to directory reads, either explicit reads (`readdir(2)` calls) or implicit ones (`namei` kernel directory lookup).

Ext2fs also contains many allocation optimizations. Block groups are used to cluster together related inodes and data: the kernel code always tries to allocate data blocks for a file in the same group as its inode. This is intended to reduce the disk head seeks made when the kernel reads an inode and its data blocks.

When writing data to a file, Ext2fs preallocates up to 8 adjacent blocks when allocating a new block. Preallocation hit rates are around 75% even on very full filesystems. This preallocation achieves good write performances under heavy load. It also allows contiguous blocks to be allocated to files, thus it speeds up the future sequential reads.

These two allocation optimizations produce a very good locality of:

- related files through block groups
- related blocks through the 8 bits clustering of block allocations.

6.3 Third Extended Filesystem (Ext3 FS)

Ext3 support the same features as Ext2, but includes also Journaling. You can download pre– version from ftp://ftp.uk.linux.org/pub/linux/sct/fs/jfs/.

6.4 E2compr – Ext2fs transparent compression

- Homepage: http://opensource.captech.com/e2compr/
- Download: ftp://opensource.captech.com/e2compr/
- Maintainer: Peter Moulder < reiter@netspace.net.au>
- Freshmeat: Console/Filesystems
- Access: As for ext2 (Read/Write, Long filenames)
- License: GPL except for compression algorithms (various licenses)

Implements `chattr +c' for the ext2 filesystem. Software consists of a patch to the linux kernel, and patched versions of various software (principally e2fsprogs i.e. e2fsck and friends). **Although some people have been relying on it for years, THIS SOFTWARE IS STILL IN DEVELOPMENT, AND IS NOT ,END−USER`−READY.**

6.5 Accessing Ext2 from DOS (Ext2 tools)

- Download: ftp://sunsite.unc.edu/pub/Linux/system/filesystems/ext2/
- Access: Read−only, no drive letters (special utilites)
- Author: Claus Tondering < ct@login.dknet.dk>
- Access: ?
- License: ?

A collection of DOS programs that allow you to read a Linux ext2 file system from DOS.

6.6 Accessing Ext2 from DOS, Windows 9x/NT and other Unixes (LTools)

- Homepage: http://www.it.fht−esslingen.de/~zimmerma/software/ltools.html
- Author: Werner Zimmermann < Werner.Zimmermann@fht−esslingen.de>
- Homepage: http://www.it.fht−esslingen.de/~zimmerma/software/ltools.htm
- Mirror: http://metalab.unc.edu/pub/linux/utils/dos/ (only major releases)
- Access: Read/Write/Modify, Long filenames
- License: GPL

The LTOOLS are under DOS/Windows 3.x/Windows 9x/Windows NT or non−Linux−UNIX, what the MTOOLS are under Linux. You can access (read, write, modify) your Linux files when running one of the other operating systems. The kernel of the LTOOLS is a set of command line programs. Additionally a JAVA program as a stand alone graphical user interface is available. Alternatively, you can use your standard web browser as a graphical user interface. The LTOOLS do not only provide access to Linux files on your own machine, but also remote access to files on other machines.

6.7 Accessing Ext2 from OS/2

- Homepage: http://perso.wanadoo.fr/matthieu.willm/ext2−os2/
- Author: Matthieu WILLM < willm@ibm.net> , < matthieu.willm@wanadoo.fr>
- Download: ftp://hobbes.nmsu.edu/pub/os2/system/drivers/filesys/ext2_240.zip
- Freshmeat: Console/Filesystems
- Access: Read/Write, swapping and booting to/from ext2, removable media support, but NO extended attributes.

EXT2−OS2 is a package that allows OS/2 to seamlessly access Linux ext2 formatted partitions from OS/2 as if they were standard OS/2 drive letters. The ultimate aim of this package is to be able to use the ext2 file system as a replacement of FAT or HPFS. For the moment the only lacking feature to achieve this goal is the support for OS/2 extended attributes.

6.8 Accessing Ext2 from Windows 95/98 (FSDEXT2)

- Homepage: http://www.yipton.demon.co.uk/
- Author: Peter van Sebille pvs@globalxs.nl , pese@nlnwgfsc.origin.nl
- Freshmeat: Console/Filesystems
- Access: Read−only, Long filenames supported

6.9 Accessing Ext2 from Windows 95 (Explore2fs)

- Homepage: http://uranus.it.swin.edu.au/~jn/linux/explore2fs.htm
- Access: Read/Write, Long filenames, symbolic links etc...
- Author: John Newbigin < jn@it.swin.edu.au>
- License: GPL

A user space application which can read and write the second extended file system ext2. Supports hard disks and removable media, including zip and floppy. Uses a windows explorer like interface to show files and details. Supports Drag& Drop, context menus etc. Written for Windows NT, but has some support for Windows 95. Large disks can cause problems.

6.10 Accessing Ext2 from Windows NT (ext2fsnt)

- Homepage: http://www.chat.ru/~ashedel/ext2fsnt/
- Download: http://www.chat.ru/~ashedel/ext2fsnt/ext2fsnt.rar
- Author: Andrey Shedel < andreys@cr.cyco.com >
- Freshmeat: Console/Filesystems
- License: Free
- Access: Read−write, LFN, Security, PageFile, Hardlinks.

6.11 Accessing Ext2 from BeOS

- Homepage: http://www.cs.tamu.edu/people/tkg0143/be/
- Author: Travis Geiselbrecht < geist@tamu.edu >
- Download: http://www.cs.tamu.edu/people/tkg0143/be/downloads/ext2fs−1.0.6−x86−r4.zip for R4

and http://www.cs.tamu.edu/people/tkg0143/be/downloads/ext2fs−1.0.3−x86−r3.zip for R3.
- Access: Read−only, long filenames supported.
- License: Free

This is a driver to allow BeOS to mount the Linux Ext2 filesystem. The version that is currently released author consider pretty stable. People have been using it for a long time, with no bug reports.

Authow now works for Be Inc, so you will not see his ext2 and NTFS filesystem support updated on the web much more. The drivers will be pulled into future BeOS releases.

6.12 Accessing Ext2 from MacOS (MountX)

- Homepage: http://calvaweb.calvacom.fr/bh40
- Author: ?
- Download: ?

MacOS driver which allows you to mount ext2 filesystems (Linux and MkLinux) on the Macintosh.

6.13 Accessing Ext2 from MiNT

- Homepage: http://?
- Author: < yescrew@capybara.sk−pttsc.lj.edus.si>
- Download: ?
- License: GPL

This is a full working Ext2 filesystem driver for FreeMiNT. It can read and write the actual ext2 version as implemented in Linux for example. The partition size is not limited and the logical sector size can be 1024, 2048 or 4096 bytes. The only restriction is that the physical sector size is smaller or equal to the logical sector size. The blocksize can be configured if you initialize the partition with mke2fs.

6.14 Ext2fs defrag

- Download: ftp://ftp.uk.linux.org/pub/linux/sct/defrag/
- Author: Stephen C. Tweedie < sct@redhat.com >
- License: GPL

Defragments your ext2 filesystem. Needs updated for glib libraries.

6.15 Ext2fs resize

- Homepage: http://www.dsv.nl/~buytenh/ext2resize/
- Download: http://www.dsv.nl/~buytenh/ext2resize/ext2resize−990617.tar.bz2
- Author: Lennert Buytenhek < buytenh@dsv.nl>.
- License: GPL

Resizes second extended filesystem.

6.16 Ext2end

- Homepage: http://linux.msede.com/ext2/ext2end.html
- Maintainer: Mike Field < mafield@the.net.nz>
- License: Copyright Mike Field. To be GPLed once stable.

For use with LVM Consists of 2 utilites. ext2endable reorganises an empty ext2 file systems to allow them to be extended, and ext2end that extends an unmounted ext2 file system. If ext2endable has not been run when the file system was created ext2end will only be able to extend it to the next multiple of 256MB

6.17 Repairing/analyzing/creating Ext2 using E2fsprogs

- Homepage: http://e2fsprogs.sourceforge.net/
- Download: ftp://download.sourceforge.net/pub/sourceforge/e2fsprogs/
- Authors: tytso@mit.edu and card@masi.ibp.fr
- Windows NT port: http://www.chat.ru/~ashedel/ext2fsnt/
- Freshmeat: Console/Filesystems
- License: GPL

The ext2fsprogs package contains essential ext2 filesystem utilities which consists of e2fsck, mke2fs, debugfs, dumpe2fs, tune2fs, and most of the other core ext2 filesystem utilities.

6.18 Ext2 filesystem editor − Ext2ed

- Homepage: ?
- Author: tgud@tochnapc2.technion.ac.il.
- Download: http://sunsite.unc.edu/pub/Linux/system/Filesystems/ext2/ext2ed−0.1.tar.gz
- License: GPL

EXT2ED is a disk editor for the extended2 filesystem. It will show you the ext2 filesystem structures in a nice and intuitive way, letting you easily "travel" between them and making the necessary modifications.

6.19 Linux filesystem editor − lde

- Homepage: ?
- Author: Scott D. Heavner < sdh@po.cwru.edu>.
- Download: http://sunsite.unc.edu/pub/Linux/system/Filesystems/lde−2.3.4.tar.gz
- License: GPL

This allows you to view some Linux fs's, hex block and inode editing are now supported and you can use it to dump an erased file to another partition with a little bit of work. Supports ext2, minix, and xiafs. Includes **LaTeX Introduction to the Minix fs**. You must patch sources to compile on 2.2.x and 2.3.x kernels beacuse of missing Xia header files in kernel.

6.20 Ext2 undelete utilities

- Homepage: http://amadeus.uprm.edu/~undelete
- Authors: Gunther Costas, Wilfredo Lugo, Jerry Ramirez < undelete@amadeus.uprm.edu>
- Freshmeat: Console/Filesystems
- License: GPL

This is a patch for kernel 2.0.30 that adds undelete capabilities using the "undeletable" attribute provided by the ext2fs. This patch include man pages, the undelete daemon and utilities. Check our web page for the latest and greatest version.

7.Macintosh Hierarchical Filesystem − HFS

All Macintosh storage devices except floppy disks are partitioned into one or more volumes. Volumes can contain four kinds of items: files, directories, directory threads and file threads. Each item is described by a catalog record which is analogous to a Unix inode. Catalog records are organized in the on−disk catalog B−Tree. Directory contents are derived from searching the catalog B−Tree. Only a file can occupy space outside of its catalog record.

A Macintosh "file" contains two components, or forks. The resource fork is an indexed file containing code segments, menu items, dialog boxes, etc. The data fork has the "stream of bytes" semantics of a Unix file contents. Each fork is comprised of one or more extents or contiguous runs of blocks. An extent descriptor encodes an extent's starting block and length into a 32bit quantity. The first extent record (three extent descriptors) of each fork is a part of the file's catalog record. Any further extent records are kept in the extents overflow B−Tree.

In addition to file and B−Tree extents a volume also contains two boot blocks, a volume information block, and a free space bitmap. There is a remarkable amount of redundancy in the on diskdata structures which improves crash recovery. While not strictly a part of the filesystem, it should be noted that several catalog record fields are reserved for the exclusive use of Finder, a program which handles user access to the filesystem and automatically maintains associations between applications and data files. Thus, HFS must also maintain this Finder info.

Every file and directory on an HFS volume has an identification number, similar to an inode number in the Unix filesystem. However, a file or directory is named by its parent's identification number and the file or directory's file name, which is a 32 character string that can contain nulls. This combination is the search key to the volume's catalog B−Tree. The catalog B−Tree differs from a traditional B−Tree structure in that all the nodes at each level of the B−Tree are linked together to form a doubly linked list and all of the records are in the leaf nodes. These variations permit accessing many items in the same directory by traversing the leaves using the linked list. Strictly speaking, the HFS B−Trees are a variant of B+−Trees although Apple's technical documentation calls them B*−Trees.

Each directory, including the root directory, contains its directory thread, which has the empty filename. The directory thread record contains the name of the directory and the id of the parent of the directory. Similarly, filethreads contain the name of a file and the id of the directory they are in. While every directory must contain a directory thread, file threads are very uncommon. In fact, both are examples of HFS redundancy − for undamaged trees, threads are not strictly necessary. Both file and directory records contain 32 bytes of information used by Finder. The first three extent descriptors for the catalog B−Tree are kept in the volume information block. If the catalog B−Tree file grows beyond three extents, the remaining extent descriptors are kept in the extents overfow.

HFS and HFS+ (also called Sequoia) filesystems are well documented. The best source of tech. information about HFS can be found in the **Inside Macintosh** series of books. Look at http://developer.apple.com/techpubs/mac/Files/Files−99.html. The HFS+ filesystem is described in **Technote 1150**, available online at http://developer.apple.com/technotes/tn/tn1150.html. A lot of information is available also in other technotes. This links are collected by Paul H. Hargrove:

- http://developer.apple.com/dev/technotes/fl/fl_22.html − HFS Ruminations.
- http://developer.apple.com/dev/technotes/fl/fl_32.html − Hey, Buddy, Can You Spare A Block?
- http://developer.apple.com/dev/technotes/fl/fl_505.html − Alias Manager Q&As
- http://developer.apple.com/dev/technotes/fl/fl_515.html − File Manager File Handling Q&As
- http://developer.apple.com/dev/technotes/fl/fl_530.html − File Manager Volume Handling Q&As
- http://developer.apple.com/dev/qa/ops/ops08.html − Bizarre Extension Loading Order: BackQuote Sorts Between "A" and "B"
- http://developer.apple.com/dev/technotes/tb/tb_535.html − Finder Q&As

7.1 Accessing HFS from Linux

- Homepage: http://www−sccm.stanford.edu/~hargrove/HFS/
- Author: Paul. Hargrove < hargrove@sccm.stanford.edu>
- Freshmeat: Console/Filesystems
- License: GPL

7.2 Accessing HFS from OS/2 (HFS/2)

- Homepage: http://www.student.nada.kth.se/~f96−bet/HFS/
- Author: Marcus Better < Marcus.Better@abc.se>

HFS/2 lets OS/2 users seamlessly read and write files on diskettes formatted with the Hierarchical File System, the file system used by Macintosh computers. With HFS/2, Macintosh diskettes can be used just as if they were regular diskettes.

This program is no longer being developed, because author doesn't use OS/2. If you are willing to maintain the program, let him know.

7.3 Accessing HFS from Windows 95/98/NT (HFV Explorer)

- Homepage: http://gamma.nic.fi/~lpesonen/HFVExplorer/
- Author: Lauri Pesonen < lpesonen@nic.fi>
- Access: R/W access to floppies, Zip disks and virtual volume files. Read access to HFS and hybrid CD's.
- License: GPL

An HFS volume browser for Windows NT and Windows 9x based on hfsutils. Launch pad support for all major Macintosh emulators running on Windows.

7.4 Accessing HFS from DOS (MAC−ETTE)

- Homepage: ?
- Author: Paul E. Thomson
- Download: http://home2.inet.tele.dk/shefan/macette3.zip
- Access: Read−Only
- License: Shareware ($34)

Mac−ette is a PC utility which can read, write, format and duplicate Macintosh HFS format 1.4 Meg diskettes on a PC equipped with a 3.5 inch high density diskette drive.

7.5 HFS utils

- Homepage: http://www.mars.org/home/rob/proj/hfs/
- Author: Robert Leslie < rob@mars.org>
- OS/2 port: http://www.f.kth.se/~f96−bet/hfsutils/

The hfsutils package contains a set of command−line utilities such as hformat, hmount, hdir, hcopy, etc. They allow read−write access of files and directories on HFS volumes.

7.6 MacFS: A Portable Macintosh File System Library

- Tech report: http://reports−archive.adm.cs.cmu.edu/anon/1998/abstracts/98−145.html
- Author: Peter A. Dinda < pdinda+macfs@cs.cmu.edu>, George C. Necula, and Morgan Price
- Download: ftp://ftp.cs.cmu.edu/user/pdinda/MacFS_0.1.tar.gz
- Access: Read/Write, full open/read/write/seek/close support
- License: Free for noncommercial and nonmilitary use, see ftp://ftp.cs.cmu.edu/user/pdinda/MacFS_0.1.LICENSE

This is a Macintosh file system library which is portable to a variety of operating systems and platforms. It presents a programming interface sufficient for creating a user level API as well as file system drivers for operating systems that support them. Authors implemented and tested such a user level API and utility programs based on it as well as an experimental Unix Virtual File System. They also describe the Macintosh Hierarchical File System and their implementation and note that the design is not well suited to reentrancy and that its complex data structures can lead to slow implementations in multiprogrammed environments. Performance measurements show that our implementation is faster than the native Macintosh implementation at creating, deleting, reading and writing files with small request sizes, but slower than the Berkeley Fast File System (FFS.) However, the native Macintosh implementation can perform large read and write operations faster that either our implementation or FFS.

8.ISO 9660 − CD−ROM filesystem

Useful ISO−9660 links:

- ftp://ftp.ecma.ch/ecma−st/e119−pdf.pdf − ISO−9660 (aka ECMA−119, aka High Sierra) specification

8.1 RockRidge extensions

Extensions allowing long filenames and Unix−style symbolic links.

Useful RockRidge links:

- ftp://ftp.ymi.com/pub/rockridge/susp112.ps − System Usage Sharing Protocol (SUSP, IEEE P1281)
- ftp://ftp.ymi.com/pub/rockridge/rrip112.ps − Rock Ridge Interchange Protocol (RRIP, IEEE P1282)

8.2 Joliet extensions

Joliet is a Microsoft extension to the ISO 9660 filesystem that allows Unicode characters to be used in filenames. This is a benefit when handling internationalization. Like the Rock Ridge extensions, Joliet also allows long filenames.

8.3 Hybrid CD−ROMs

Hybrid CDs contains three filesystems on one disc − ISO9660/RR, Joliet and HFS. Such CD−ROMs are accessible under DOS, Unix, Macintosh and Windows 9x/NT. All three filesystems use the same data, only metadata are the disc three times.

8.4 Novell NetWare indexes on ISO9660

(todo)

8.5 Accessing Joliet from Linux

- Homepage: http://bmrc.berkeley.edu/people/chaffee/joliet.html
- License: GPL

8.6 Accessing Joliet from BeOS

- Homepage: http://www.iae.nl/users/gertjan/be/
- Author: Gertjan van Ratingen < gertjan@iae.nl>
- License: ?

It is updated ISO9660 driver to be able to use a Joliet ISO9660 extensions.

8.7 Accessing Joliet from OS/2

- Download: ftp://hobbes.nmsu.edu/pub/os2/system/drivers/filesys/jcdfs.zip
- Author: IBM
- License: ?

Jcdfs.zip archive contains CDFS.IFS driver for OS/2 with Joliet level 3 support.

8.8 Accessing Audio CD as filesystem from Linux

- Homepage: http://www.ii.pw.edu.pl/~borkowsm/cdfs.htm
- Download: ?
- Author: Mariusz Borkowski < borkowsm@ii.pw.edu.pl>
- License: ?

This kernel module allows you to access an audio CD as a regular filesystem.

8.9 Accessing Audio CD as filesystem from BeOS

- Homepage: http://www.xs4all.nl/~marcone/be.html
- Download: http://www.xs4all.nl/~marcone/be/files/cdda5.zip (PPC/Intel archive)
- Author: Marco ?
- License: ?

This filesystem add−on will allow you (if your CD drive supports it) to treat a regular audio CD as if it were a bunch of WAV files. You can copy the files, encode them to mp3, play them slower, faster, even backwards.

8.10 Accessing all tracks from Linux (CDfs)

- Homepage: http://www.elis.rug.ac.be/~ronsse/cdfs/
- Download: http://www.elis.rug.ac.be/~ronsse/cdfs/cdfs.tgz
- Author: Michiel Ronsse < ronsse@elis.rug.ac.be>
- License: GPL

CDfs is a file system for Linux systems that `exports' all tracks and boot images on a CD as normal files. These files can then be mounted (e.g. for ISO and boot images), copied, played (audio tracks), etc. The primary goal for developing this file system was to `unlock' information in old ISO sessions. The file system also allows you to access data on faulty multi session disks, e.g. disks with multiple single sessions instead of a multi session.

8.11 Creating Hybrid CD−ROMs (mkhybrid)

- Homepage: http://www.ps.ucl.ac.uk/~jcpearso/mkhfs.html
- Download: ftp://ftp.ge.ucl.ac.uk/pub/mkhfs/
- Author: < j.pearson@ge.ucl.ac.uk>
- License: ?

Make an ISO9660/HFS/JOLIET shared hybrid CD volume

9.Other filesystems

9.1 ADFS − Acorn Disc File System

The Acorn Disc Filing System is the standard filesystem of the RiscOS operating system which runs on Acorn's ARM−based Risc PC systems and the Acorn Archimedes range of machines.

Linux kernel 2.1.x+ supports this filesystem. Author of Linux filesystem implementation is Russell King < rmk@arm.uk.linux.org>.

9.2 AFFS − Amiga fast filesystem

The Fast File System (FFS) is the common filesystem used on hard disks by Amiga(tm) systems since AmigaOS Version 1.3 (34.20).

Linux kernel 2.1.x+ supports this filesystem. Author of Linux filesystem implementation is Ray Burr < ryb@nightmare.com>.

9.3 BeFS − BeOS filesystem

BeFS is journaling filesystem used in BeOS. For more information about BeFS see Practical File System Design with the Be File System book or BeFS linux driver source code.

Linux BeFS implementation:

- Homepage: http://hp.vector.co.jp/authors/VA008030/bfs/
- Download: http://hp.vector.co.jp/authors/VA008030/bfs/bfs−19990528.tar.gz
- Author: Makoto Kato < m_kato@ga2.so−net.ne.jp>
- Access: Read−only
- License: GPL

This driver supports x86 and PowerPC Linux platform. Also, it only supports readable in hard disk and floppy disk.

9.4 BFS − UnixWare Boot Filesystem

UnixWare BFS filesystem type is a special−purpose filesystem. It was designed for loading and booting UnixWare kernel. BFS was designed as a contiguous filesystem. BFS supports only one (root) directory and you can create only regular files; no subdirs or special files such as devices or sockets can be created.

For more information about BFS see http://uw7doc.sco.com/FS_admin/_The_bfs_File_System_Type.html.

- http://uw7doc.sco.com/FS_admin/_The_bfs_Superblock.html − superblock
- http://uw7doc.sco.com/FS_admin/_bfs_Inodes.html − inodes
- http://uw7doc.sco.com/FS_admin/_bfs_Storage_Blocks.html − storage blocks

You can access BFS filesystem from Linux:

- Homepage: http://www.ocston.org/~tigran/patches/bfs/
- Download: In the Linux kernel, patches available at homepage.

* Author: Tigran A. Aivazian < tigran@ocston.org>
* License: GPL
* Access: Read/write (write part is limited, no compactification yet)

The support for BFS is included in the Linux kernel since version 2.3.25. If you are using an earlier kernel, check if BFS homepage contains a patch which adds support for this filesystem. The homepage also contains bugfixes/enhancement which are not yet merged into the official kernel.

There is also mine old implementation, which is now obsolete. My plan is to port this code to FreeBSD:

* Homepage: http://www.penguin.cz/~mhi/fs/bfs/
* Download: ftp://ftp.penguin.cz/pub/users/mhi/bfs/
* Author: Martin Hinner < mhi@penguin.cz>
* License: GPL
* Access: Read−only

This is read−only UnixWare Boot filesystem support for Linux. You can use it to mount read−only your UnixWare /stand partition or floppy disks. I don't plan a read−write version, but if you want it mail me. You might be also interested in VxFS Linux support.

9.5 CrosStor filesystem

This is new name for **High throughput filesystem (HTFS)**. For more information see CrosStor homepage at http://www.crosstor.com.

9.6 DTFS − Desktop filesystem

Goals in designing the Desktop File System were influenced by impression of what environment was like for small computer systems. DTFS compress the data stored in regular files to reduce disk space requirements (directories remain uncompressed). Compression is performed a page at a time and occur 'on−the−fly'. DTFS supports LZW and no−compression but you can add your own algorithms. Some space is saved by not pre−allocating inodes. Any disk block is fair game to be allocated as an inode. Each inode is stored as a B+tree. For more information see DTFS USENIX paper (you can download it from ftp://ftp.crosstor.com/pub/DTFS/papers/).

Read/Write **commercial** driver available from CrosStor for UnixWare and SUN Solaris:

* Download: ftp://ftp.crosstor.com/pub/DTFS/
* License: Commercial?
* Access: Read/Write

9.7 EFS − Enhanced filesystem (Linux)

The Enhanced Filing system project aims to create a new filing system for Linux and eventually other OSs which will allow the administrator to define mountable "file systems" on a set of block devices (either hard drives or partitions). The aim is to allow file systems to be added or removed from the partition set while the system is running and partitions may be added to a set (or removed if the remaining partitions have enough space to contain all the data) while the system is running.The two main aims are to allow a number of mountable file systems to share the same pool of storage space (IE have the user home dirs on the same drive as the news spool but have separate accounting for them), and to allow the easy addition of more hard drives to allow more space.

Some other features that authors want to implement are logging/journaling, support for as many OSs as possible (although all work will be initially done on Linux), and quotas in the FS so we don't need to waste ages running a silly quotacheck program at boot − the logging should avoid quotacheck the same way it avoids fsck! They want to be able to boot a system with 10gig of news spread over 4 hard drives with full quotas AFTER a power failure with less than 20 seconds for mounting file systems!

Homepage of Enhanced FS is at http://www.coker.com.au/~russell/enh/. Contact Russell Coker < russell@coker.com.au> for more information.

9.8 EFS − Extent filesystem (IRIX)

The Extent File System (efs) is Silicon Graphics' early block−device filesystem, widely used on pre−6.0 versions of IRIX. Since 6.0, xfs has been bundled with IRIX and users are being encouraged to migrate to xfs filesystems. IRIX support for efs will be read−only in versions of IRIX beyond 6.5, however efs is still very much in use on SGI software distribution CDs.

There are two kernel modules for linux to access EFS filesystem.

- Homepage: http://aeschi.ch.eu.org/efs/
- Download: http://aeschi.ch.eu.org/efs/efs−1.0b.tar.gz
- Author: Al Smith < Al.Smith@aeschi.ch.eu.org>
- License: GPL
- Access: Read−only

The efs kernel module is an implementation of the extent file system for linux 2.2 kernels. An efs implementation (efsmod−0.6.tar.gz) was originally written for 1.x kernels by Christian Vogelgsang. In this implementation the code has undergone a complete rewrite and is also endian−clean. To use the efs module, you will need to have at least a 2.2 kernel. To mount IRIX CDs, your CD−ROM will need to be able to cope with 512−byte blocks. This version of efs contains support for hard−disk partitions, and also contains a kernel patch to allow you to install the efs code into your linux kernel tree. Handling of large files has also been vastly improved.

Original efsmod is also available:

- Homepage: http://wwwcip.informatik.uni−erlangen.de/user/cnvogelg/proj.html
- Download: http://wwwcip.informatik.uni−erlangen.de/user/cnvogelg/bin/efsmod−0.6.tgz
- Author: Christian Vogelgsang
- License: GPL
- Access: Read−only

Efs−mod 0.6 is original EFS read/only module for Linux. Version 0.6 finished but Project frozen due to lack of time and information for implementing the write part.

EFS and FFS library, libfs

- Download: ftp://ivo.cps.unizar.es/pub/SPDsoft/libfs.tar.gz
- Author: J.A. Gutierrez < spd@ivo.cps.unizar.es>
- License: GPL
- Access: Read/Only IRIX EFS and Sun UFS

A C library to read EFS and FFS from WinNT x86, SunOS and IRIX. Easy to use (Posix like interface) and to links aginst existent code FTP server has also winefssh.exe and winufssh.exe, simple WinNT binaries to interactively read UFS and EFS file systems. Not a very polished/documented package, but somebody may find it useful.

Useful links:

- IRIX EFS filesystem brief description: http://squish.ucs.indiana.edu:80/ebt−bin/nph−dweb/dynaweb/SGI_Admin/IA_DiskFiles/@ebt−link;td=8?target=%25N%14_7484_START_RESTART_N%25

9.9 FFS − BSD Fast filesystem

This is native filesystem for most BSD unixes (FreeBSD, NetBSD, OpenBSD, Sun Solaris, ...).

See also: SFS, secure filesystem, UFS.

9.10 GPFS − General Parallel Filesystem

This is a UNIX(tm) operating system style file system designed for the RS/6000 SP(tm) server. It allows applications on multiple nodes to share file data. GPFS supports very large file systems and stripes data across multiple disks for higher performance. GPFS is based on a shared disk model which provides lower overhead access to disks not directly attached to the application nodes and uses a distributed locking protocol to provide full data coherence for access from any node. It offers many of the standard AIX(tm) file system interfaces allowing most applications to execute without modification or recompiling. These capabilities are available while allowing high speed access to the same data from all nodes of the SP system, and providing full data coherence for operations occurring on the various nodes. GPFS attempts to continue operation across various node and component failures assuming that sufficient resources exist to continue.

- http://www.austin.ibm.com/resource/technology/paper1.html

9.11 HFS − HP−UX Hi performance filesystem

This is the second hfs that appears in this howto. It is used in older HP−UX versions.

9.12 HTFS − High throughput filesystem

Useful links:

- SCO OpenServer 5 filesystems whitepaper: http://www.sco.com/products/Whitepapers/family/filesy4.htm

Read/Write **commercial** driver available from CrosStor:

- Download: ftp://ftp.crosstor.com/pub/HTFS/
- License: Commercial?
- Access: Read/Write

9.13 JFS − Journaled filesystem (HP−UX, AIX, OS/2 5, Linux)

- Homepage: http://oss.software.ibm.com/developerworks/opensource/jfs/
- Download: http://www10.software.ibm.com/developer/opensource/jfs/project/pub/
- Author: Steve Best < sbest@us.ibm.com> and Dave Kleikamp < shaggy@us.ibm.com>
- License: GPL
- Access: ?

JFS is IBM's journaled file system technology, currently used in IBM enterprise servers, and is designed for high−throughput server environments.

9.14 LFS − Linux log structured filesystem

Linux Log structured filesystem implementation called d(t)fs:

- Homepage: http://www.complang.tuwien.ac.at/czezatke/lfs.html
- Author: Christian Czezatke < e9025461@student.tuwien.ac.at>
- License: GPL
- Access: rw/long filenames, etc

d(t)fs is a log−structured filesystem project for Linux. Currently, the filesystem is mostly up and running, but no cleaner has been written so far.

There will also be a dtfs mailing list that will be announced on the homepage. For more information you can have a look at: http://www.xss.co.at/mailman/listinfo.cgi/dtfs

- http://collective.cpoint.net/lfs/ − The kfs Homepage Cornelius "Kees" Cook has started a Linux Log−−Structured Filesystem project before dtfs came to live.
- http://lucien.blight.com/~c−cook/prof/lfs/ − Another (death) LFS implementation ;−)
- http://www.eecs.harvard.edu/~margo/usenix.195/ − Margo Seltzer's < margo@das.harvard.edu> LFS page

9.15 MFS − Macintosh filesystem

MFS is original Macintosh filesystem. It has been replaced by HFS / HFS+. If you can provide further information, mail me please.

9.16 Minix filesystem

This is Minix native filesystem. It was also used in first versions of Linux.

9.17 NWFS − Novell NetWare filesystem

NWFS is native in Novell NetWare OS. It is modified FAT−based filesystem. Two variants of this filesystem exists. 16bit NWFS 286 is used in NetWare 2.x. NetWare 3.x, 4.x and 5 use 32bit NWFS 386.

NetWare filesystem / 286

(todo)

NetWare filesystem / 386

(todo)

Accessing NWFS−386 from Linux

- Homepage: http://www.timpanogas.com/html/fenris_for_linux.html
- Download: ftp://207.109.151.240/nwfs/
- Author: Timpanogas Research Group, Inc. (jmerkey@timpanogas.com)
- License: GPL
- Access: Read−Only

This driver allows you to mount NWFS−386 filesystem on Linux.

9.18 NSS − Novell Storage Services

This is a new 64bit journaling filesystem using a balanced tree algorithms. It is used in Novell NetWare 5.

- http://www.novell.com/whitepapers/nw5/nss.html − NSS Whitepaper

9.19 ODS − On Disk Structure filesystem

This is OpenVMS and VMS native filesystem.

9.20 QNX filesystem

This filesystem is used in QNX. Two major filesystem version exists, version 2 is used by QNX 2 and version 4 by QNX 4. QNX 4 doesn't support version 2 and vice versa.

QNX4 filesystem is now accessible from Linux 2.1.x+. Say "Y"es to 'QNX filesystem support';

- Download: In the kernel ;)
- Author: Frank Denis < j@4u.net> (maintainer), Richard Frowijn
- License: GPL
- Access: Read (except for multi−extents files), Write (experimental)

Driver for the QNX 4 filesystem.

9.21 Reiser filesystem

Reiserfs is a file system using a variant on classical balanced tree algorithms. The results when compared to the ext2fs conventional block allocation based file system running under the same operating system and employing the same buffering code suggest that these algorithms are more effective for large files and small files not near node size in time performance, become less effective in time performance and more significantly effective in space performance as one approaches files close to the node size, and become markedly more effective in both space and time as file size decreases substantially below node size (4k), reaching order of magnitude advantages for file sizes of 100bytes. The improvement in small file space and time performance suggests that we may now revisit a common OS design assumption that one should aggregate small objects using layers above the file system layer.

Useful links:

- Reiser fs homepage http://devlinux.org/namesys/

9.22 RFS (CD−ROM Filesystem)

Sony's incremental packet−writing filesystem.

9.23 RomFS − Rom filesystem

Author of Linux RomFS implemplementation is Janos Farkas < chexum@shadow.banki.hu> For more information see **/usr/src/linux/Documentation/filesystems/romfs.txt** file.

9.24 SFS − Secure filesystem

The sfs filesystem type is a variation of the FFS filesystem type. The boot block,superblock, storage blocks, and free blocks for the sfs filesystem type are, at the administrative level, identical to those for FFS. The inodes differ from FFS inodes, however. Each odd−numbered inode is reserved for security information. The information contains Access Control List information. I'm not sure if SFS has any other abilities though.

SFS links:

- http://uw7doc.sco.com/FS_admin/_The_sfs_File_System_Type.html − UnixWare 7 documentation: SFS Filesystem

9.25 Spiralog filesystem (OpenVMS)

Spiralog is a 64bit high−performance filesystem for the OpenVMS. The Spiralog combines log−structured technology with more traditional B−tree technology to provide a general abstraction. The B−tree mapping mechanism uses write−ahead logging to give stability and recoverability guarantees.

Spiralog−related links at Digital:

- http://www.digital.com/info/SP6048/ − Spiralog File System for OpenVMS Alpha
- http://www.digital.com/DTJM01/DTJM01AH.HTM − Overview of the Spiralog File System
- http://www.digital.com/DTJM02/DTJM02HM.HTM − Design of the Server for the Spiralog File System

9.26 System V and derived filesystems

Homepage of System V Linux project is at http://www.knm.org.pl/prezes/sysv.html. Maintainer of this project is <kgb@manjak.knm.pl.org>.

AFS − Acer Fast Filesystem

The Acer Fast Filesystem is used on SCO Open Server. It is similar to the System V Release 4 filesystem, but it is using bitmaps instead of chained free−list of blocks.

EAFS − Extended Acer Fast Filesystem

The AFS filesystem can be 'extended' to handle file names up to 255 characters, but directories entries still have 14−char names. This filesystem type is used on SCO Open Server.

Coherent filesystem

S5

This filesystem is used in UnixWare. It's probably SystemV compatible, but I haven't verified it yet. For more information see http://uw7doc.sco.com/FS_admin/_The_s5_File_System_Type.html.

S51K − SystemV 1K

Version 7 filesystem

This filesystem type is used on Version 7 Unix for PDP−11 machines.

Xenix filesystem

9.27 Text − (Philips' CD−ROM Filesystem)

Philips' standard for encoding disc and track data on audio CDs.

9.28 UDF − Universal Disk Format (DVD−ROM filesystem)

There is a Linux UDF filesystem driver:

- Homepage: http://trylinux.com/projects/udf/
- Download: http://trylinux.com/projects/udf/udf−0.8.0.1.tar.gz
- Author: Dave Boynton < dave@trylinux.com>
- Mailing−list: <linux_udf@hootie.lvld.hp.com>
- License: GPL
- Access: Read−only

9.29 UFS

Note: People often call BSD Fast Filesystem incorrectly UFS. FFS and UFS are *diferrent* filesystems. All modern Unixes use FFS filesystem, not UFS! UFS was used in early BSD versions. You can download source code at http://minnie.cs.adfa.edu.au/TUHS/.

Useful links:

* http://www.sun.ca/white−papers/ufs−cluster.html − Implementation of write−clustering for Sun's UFS

See also: BSD FFS

9.30 V7 Filesystem

The V7 Filesystem was used in Seventh Edition of UNIX Time Sharing system (about 1980). For more information see 7th Ed. source code, which is available from the Unix Archive: http://minnie.cs.adfa.edu.au/TUHS/.

9.31 VxFS − Veritas filesystem (HP−UX, SCO UnixWare, Solaris)

This is commercial filesystem developer by Veritas Inc. You can see it in HP−UX, SCO UnixWare, Solaris and probably other systems. It has very interesting features: Extent based allocation, Journaling, access control lists (ACLs), up to 2 terabyte large file support, online backup (snapshot filesystem), BSD style quotas and many more.

Three VxFS versions are available with VxFS:

Version 1: This is original VxFS, not commonly in use.

Version 2: Support for filesets and dynamic inode allocation.

Version 4: Latest version, supports large files and quotas.

Note that HP−UX, Solaris and UnixWare versions use slightly different structures, so you may not be able to read VxFS when you connect it to different system.

VxFS related links:

- http://www.veritas.com/ − Veritas Inc < vx−sales@veritas.com>.
- http://uw7doc.sco.com/ODM_FSadmin/CONTENTS.html − VxFS ODM FS Admin − UnixWare 7 (documentation, really good).
- http://uw7doc.sco.com/FS_manager/fsD.vxfsopt.html − VxFS FS Manager − UnixWare 7 (documentation).
- http://manuals.mchp.siemens.de:80/dynaweb/english/ru544e/drlugueb/o25636e1/@Generic__BookView/1641;cd=3 − VxFS − Reliant Unix.

See also: VxVM (Veritas volume manager) and journaling filesystems.

VxTools

Unix command−line utilities for accessing VxFS versions 2 and 4 are available under the GNU GPL:

- Homepage: http://www.penguin.cz/~mhi/fs/vxfs/
- Download: ftp://ftp.penguin.cz/pub/users/mhi/vxfs/
- Author: Martin Hinner < mhi@penguin.cz>
- Mailing−list: <fs−l@penguin.cz>
- License: GPL
- Access: Read−only, command−line utilites

Vxtools is a set of command−line utilites which allow you to access your VxFS filesystem from Linux (and possibly other Unixes). Current version can read VxFS versions 2 and 4.

I (mhi) plan also VxFS Linux kernel driver.

AFAIK, Rodney Ramdas < rodney@quicknet.nl> works on VxFS driver for FreeBSD. I don't know current status of his project, so if you want more info contact him directly.

9.32 XFS − Extended filesystem (IRIX)

XFS(tm) is the next−generation file system for Silicon Graphics[TM] systems, from desktop workstations to supercomputers. XFS provides full 64−bit file capabilities that scale easily to handle extremely large files and file systems that grow to 1 terabyte. The XFS file system integrates volume management, guaranteed rate I/O, and journaling technology for fast, reliable recovery. File systems can be backed up while still in use, significantly reducing administrative overhead.

XFS is designed for a very high performance; sustained throughput in excess of 300MB per second has been demonstrated on CHALLENGE systems. The XFS file system scales in performance to match the CHALLENGE MP architecture. Traditional files, directories, and file systems have reduced performance as they grow in size. With the XFS file system, there is no performance penalty. For example, XFS directories have been tested with up to 32 million files in a single directory.

XFS is a journalled file system. It logs changes to the inodes, directories and bitmaps to the disk before the original entries are updated. Should the system crash before the updates are done they can be recreated using the log and updated as intended.

XFS uses a space manager to allocate disk space for the file system and control the inodes. It uses a namespace manager to control allocation of directory files. These managers use B−tree indexing to store file location information, significantly decreasing the access time needed to retrieve file information.

Inodes are created as needed and are not restricted to a particular area on a disk partition. XFS tries to position the inodes close to the files and directories they reference. Very small files, such as symbolic links and some directories, are stored as part of the inode, to increase performance and save space. Large directories use B−tree indexing within the directory file to speed up directory searches, additions and deletions.

Useful XFS links:

- http://www.sgi.com/Technology/xfs−whitepaper.html XFS whitepaper

XFS Linux port covered by the GNU General Public License is available from SGI Inc.:

- Homepage: http://oss.sgi.com/projects/xfs/
- Download: ftp://oss.sgi.com/www/projects/xfs/download/
- Author: SGI Inc., http://www.sgi.com/
- License: GPL
- Access: Read−write

9.33 Xia FS

This filesystem was developed to replace old Minix filesystem in Linux. Author of this fs is Franx Xia < qx@math.columbia.edu>

10.Raw partitions

10.1 Backing up raw partitions using DBsnapshot

(todo: www.crosstor.com)

11.Appendix

11.1 Network filesystems

This HOWTO is not about Network filesystems, but I should mention them.

There is a brief list of some which I know:

AFS − Andrew Filesystem

- The AFS FAQ is at http://www.angelfire.com/hi/plutonic/afs−faq.html.
- Commercial clients and servers for almost all platforms (except win98) are available from IBM. See http://www.transarc.com/Product/EFS/AFS/index.html
- A free client for Unix is available from the Arla Team at http://www.stacken.kth.se/projekt/arla/.
- A free Server is also in preparation, but not in production yet.

CODA

- Homepage: http://www.coda.cs.cmu.edu/
- Download: ftp://ftp.coda.cs.cmu.edu/pub/coda/linux/
- Author: CMU Coda Group < braam@coda.cs.cmu.edu>.
- License: GPL
- Access: R/W

Coda is a distributed filesystem with novel features such as disconnected operation and server replication.

NFS − Network filesystem (Unix)

NCP − NetWare Core Protocol (Novell NetWare)

SMB − Session Message Block (Windows 3.x/9x/NT)

This protocol is used in Windows world.

Intermezzo

- Homepage: http://inter−mezzo.org/
- Download: http://inter−mezzo.org/downloads.html
- Author: Stelias and Redhat < info@stelias.com>.
- License: GPL
- Access: R/W

Intermezzo is a distributed file system for Linux. It was inspired from coda but uses the disk file system as a persistent cache. Intermezzo supports disconnected operation but does not yet implement an identification system.

11.2 Encrypted filesystems

CFS

- Homepage: ?
- Download: ?
- Author: Matt Blaze < mab@research.att.com>.
- License: ?
- Access: Read/Write, using DES/3DES.

CFS pushes encryption services into the Unix(tm) file system. It supports secure storage at the system level through a standard Unix file system interface to encrypted files. Users associate a cryptographic key with the directories they wish to protect. Files in these directories (as well as their pathname components) are transparently encrypted and decrypted with the specified key without further user intervention; cleartext is never stored on a disk or sent to a remote file server. CFS employs a novel combination of DES stream and codebook cipher modes to provide high security with good performance on a modern workstation. CFS can use any available file system for its underlying storage without modification, including remote file servers such as NFS. System management functions, such as file backup, work in a normal manner and without knowledge of the key.

TCFS

- Homepage: http://tcfs.dia.unisa.it/
- Download: ftp://tcfs.dia.unisa.it/pub/tcfs/
- Authors: Luigi Catuogno < luicat@tcfs.dia.unisa.it>, Aniello Del Sorbo < anidel@tcfs.dia.unisa.it>, Luigi Della Monica < dellui@tcfs.dia.unisa.it>, G.Cattaneo < cattaneo@dia.unisa.it>, G.Persiano (http://www.dia.unisa.it/~giuper/), Ermelindo (Erry) Mauriello < errmau@tcfs.dia.unisa.it>, Angelo Celentano < angcel@tcfs.dia.unisa.it>, Andrea Cozzolino < andcoz@tcfs.dia.unisa.it>.
- License: GPL
- Access: Read/Write transparently using CBC−DES/3DES/RC5/IDEA/Others..

The main difference between TCFS and CFS is the trasparency to user obtained by using TCFS. As a matter of fact, CFS works in user space while TCFS works in the kernel space thus resulting in improved performances and security. The dynamic encryption module feature of TCFS allows a user to specify the encryption engine of his/her choiche to be used by TCFS. Currently available only for Linux, TCFS will be relased soon also for NetBSD, and will support in a near future also other FS then NFS.

SFS

(TODO: http://www.cs.auckland.ac.nz/~pgut001/sfs/index.html)

VS3FS: Steganographic File System for Linux

- Homepage: http://www.linux−security.org/sfs/
- License: ?
- Access: ?

fspatch is a kernel patch which introduces module support for the steganographic file system (formerly known as vs3fs, an experimental type of filesytem that not only encrypts all information on the disk, but also tries to hide that information in such a way that it cannot be proven to even exist on the disk. This enables you to keep sensitive information on a disk, while not be prone to being forced to reveal that information. Even under extreme circumstances, fake documents could be stored on other parts of the disk, for which a pasword may be revealed. It should not be possible to find out whether any other information is stored on the disk.

11.3 Filesystem benchmarking utilities

IOzone

- Homepage: http://www.iozone.org/
- Download: http://www.iozone.org/src/stable/
- License: freely distributable

IOzone is a filesystem benchmark tool. The benchmark generates and measures a variety of file operations. Iozone has been ported to many machines and runs under many operating systems.

11.4 Writing your own filesystem driver

DOS

I haven't seen yet any good page about writing DOS filesystem drivers (Network redirectors) on the net. The best source is Ralf Brown's interrupt list and iHPFS source code.

OS/2

- ftp://ftp.leo.org/pub/comp/os/os2/leo/devtools/doc/ifsinf.zip
- ftp://hobbes.nmsu.edu/pub/os2/system/drivers/filesys/32drv170.zip – 32 bits OS/2 device driver and IFS support. Provides 32 bits kernel services (DevHelp) and utility functions to 32 bits OS/2 ring 0 code (device drivers and installable file system drivers).

Windows NT

Microsoft IFS kit page (http://www.microsoft.com/ddk/IFSkit/) will be useful as the best way to get into NT filesystems development (even for $1K it costs).

For more information about writing FS drivers for Windows NT see http://www.ing.umu.se/~bosse/ by < bosse@acc.umu.se>.

11.5 Related documents

- http://www.honeycomb.net/os/holistic/connect/filesys.htm – good page about filesystems
- http://home.att.net/~artnaseef/ – Linux overlay filesystem by < artnaseef@worldnet.att.net>.
- http://www.braysystems.com/linux/trustees.html – Linux trustees
- http://tcfs.dia.unisa.it – Transparent Cryptography Filesystem
- http://www.sas.com/standards/large.file – Large file summit – attacks the problem of 2gig+ of file in a 32bit computer
- http://www.coda.cs.cmu.edu/ – The CODA project (a distributed file system based on AFS)
- ftp://ftp.scis.org/pub/lfs/ – LFS related papers
- http://www.redhat.com:8080/HyperNews/get/khg.html – Linux Kernel Hacker's guide
- http://www.win.tue.nl/~aeb/linux/largedisk.html – Large disk HOWTO
- http://www.atnf.csiro.au/~rgooch/linux/kernel–patches.html – The Linux devfs
- http://gfs.lcse.umn.edu/ – The Global File System (GFS)
- ftp://hobbes.nmsu.edu/pub/os2/system/drivers/filesys/tvfs211.zip – The Toronto Virtual Filesystem/2.
- ftp://hobbes.nmsu.edu/pub/os2/system/drivers/filesys/ramfs64.zip Dynamic RAM drive IFS driver for OS/2
- http://doc.sco.com/ – UnixWare and SCO Unix documentation online
- http://uw7doc.sco.com/ – UnixWare 7 documentation online
- http://publib.boulder.ibm.com/cgi–bin/bookmgr/BOOKS/SG244428/CCONTENTS – Inside OS/2 LAN Server 4.0
- ftp://tsx–11.mit.edu/pub/linux/ALPHA/userfs/ – Linux UserFS, it allows you to write a Linux process which implements a filesystem.
- http://www.nyx.net/~sgjoen/disk.html – Stein Gjoen's Multi Disk System Tuning HOWTO.
- http://linuxtoday.com/stories/5556.html – Linux Today: Kragen's Amazing List of Filesystems.
- http://www.koehntopp.de/kris/artikel/dateisysteme/ – Kristian Kohntopp's Unix Filesystems (in German).