

The World is Your Filesystem

An explanation of what FUSE is and why it matters

Josh Glover <jmglov@wmalumni.com>

Tokyo Linux Users Group - <http://tlug.jp/>
2007/06/14



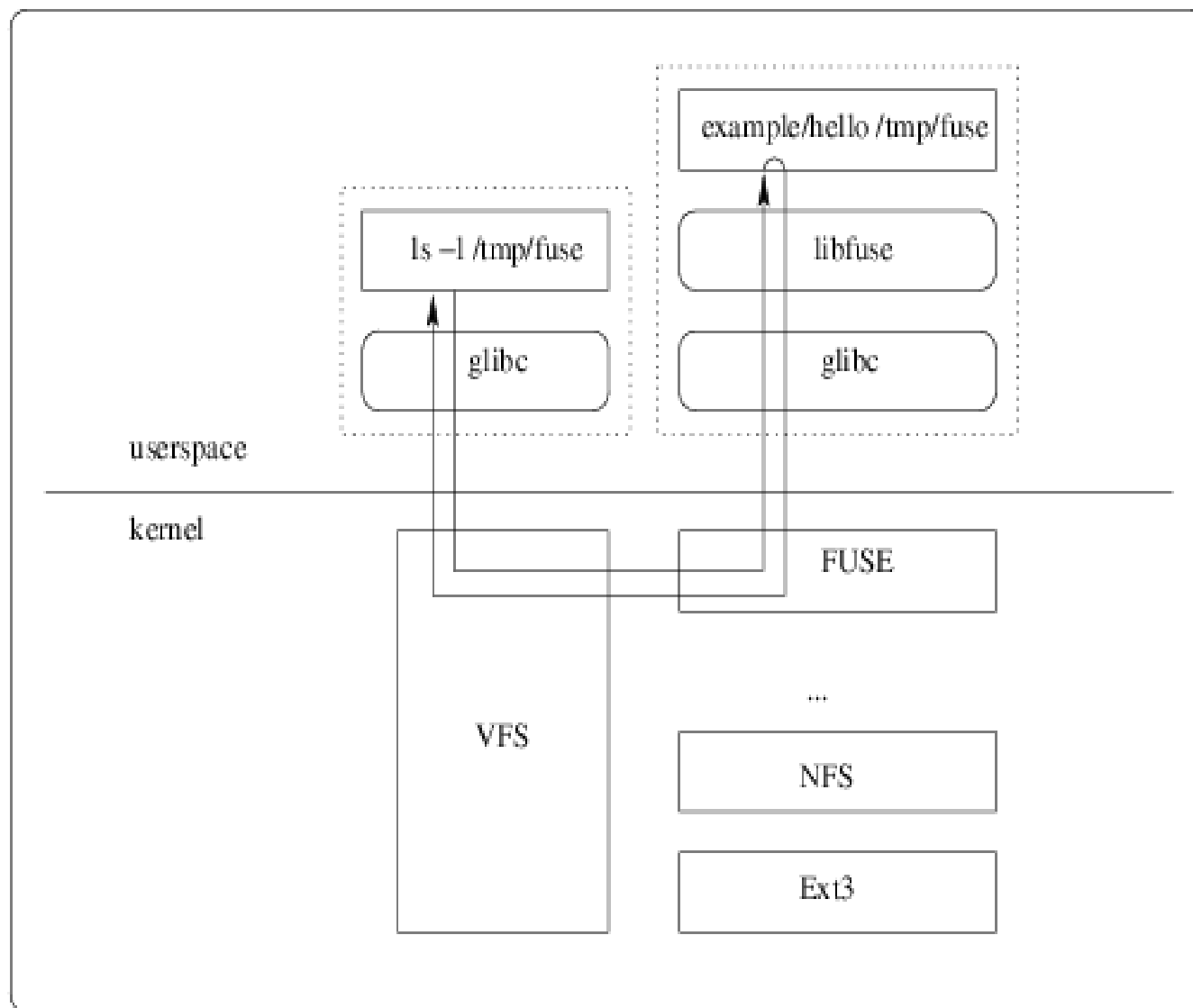
*This file is licensed under the Creative Commons Attribution-ShareAlike License.
To view a copy of this licence, visit:*

<http://creativecommons.org/licenses/by-sa/3.0/>

FUSE

- Filesystem in **Userspace**
- <http://fuse.sourceforge.net/>
- FUSE provides a kernelspace interface for userspace filesystems
- Allows you to assign filesystem semantics to virtually any task; your imagination is the limit!

Architecture



Installing FUSE

- Check for FUSE:
 - `zcat /proc/config.gz \`
`| grep CONFIG_FUSE_FS`
- Building FUSE as a module
 - Gentoo:
 - `sudo emerge -av sys-fs/fuse`
 - Generic:
 - `cd /usr/src/linux`
 - `sed -i.BAK.pre-fuse \`
`-e 's/^# CONFIG_FUSE_FS`
`.\+$ /CONFIG_FUSE_FS=m/' \`
`.config`
 - `make modules && sudo make modules_install`

Using FUSE

- Load the module:
 - `sudo modprobe fuse`
- Use one of the many userspace filesystems listed in the FUSE Wiki:
 - <http://fuse.sourceforge.net/wiki/index.php/FileSystems>
 - <http://fuse.sourceforge.net/wiki/index.php/NetworkFileSystems>
 - <http://fuse.sourceforge.net/wiki/index.php/ArchiveFileSystems>

SSHFS - 1/2

- Mount remote filesystems with ssh
- <http://fuse.sourceforge.net/sshfs.html>
- Implementation: C
- Install:
 - `sudo emerge -av sshfs-fuse`
 - `sudo apt-get install sshfs`
 - `sudo yum install sshfs`
- Use:
 - `mkdir -p ~/mnt/<hostname>`
 - `sshfs <hostname>:/ ~/mnt/<hostname>`
 - `ls ~/mnt/<hostname>/${HOME}`
 - `fusermount -u ~/mnt/<hostname>`

SSHFS - 2/2

- Really use:
 - `echo 'sshfs#<hostname>:/home/<username>/mnt/<hostname> fuse user,noauto 0 0' \`
`| sudo tee -a /etc/fstab`
 - `mount ~/mnt/<hostname>`
 - `ls ~/mnt/<hostname>`
 - `umount ~/mnt/<hostname>`

avfs

- Mount archive files (tar, gzip, zip, bzip2, ar, rar, etc.)
- <http://avf.sourceforge.net/>
- Implementation: C
- Install:
 - `sudo emerge -av avfs`
 - `sudo apt-get install avfs`
 - `sudo yum install avfs`
- Use:
 - `mkdir -p ~/.avfs`
 - `ln -s ~/.avfs ~/mnt/avfs`
 - `mountavfs`
 - `ls ~/mnt/avfs/path/to/archive#`
 - `ls \`
 - `~/mnt/avfs/path/to/archive#path/inside/archive`
 - `fusermount -u ~/mnt/avfs`

wdfs

- Mount WebDAV shares (Subversion!)
- <http://noedler.de/projekte/wdfs/>
- Implementation: C
- Install:
 - `sudo emerge -av wdfs`
 - `sudo yum install wdfs`
- Use:
 - `mkdir -p ~/mnt/svn/<project>`
 - `wdfs <URI> ~/mnt/svn/<project>`
 - `ls ~/mnt/svn/<project>/`
 - `fusermount -u ~/mnt/svn/<project>`

Gmail FS

- Mount your Gmail folders as a filesystem
- <http://richard.jones.name/google-hacks/gmail-filesystem/gma>
- Implementation: Python
- Install:
 - http://bugs.gentoo.org/show_bug.cgi?id=63785
 - `sudo apt-get install gmailfs`
- Use:
 - <http://richard.jones.name/google-hacks/gmail-filesystem/g>
 - I cannot get it to work :(

Flickrfs - 1/2

- Mount your Flickr photostream
- <http://manishrjain.googlepages.com/flickrfs>
- Implementation: Python
- Install:
 - `sudo emerge -av flickrfs`
 - `sudo apt-get install flickrfs`
- Use:
 - `mkdir -p ~/mnt/flickr`
 - `flickrfs ~/mnt/flickr`
 - `mkdir ~/mnt/flickr/stream`
 - `display ~/mnt/flickr/stream/<fn>`
 - `fusermount -u ~/mnt/flickr`

Flickrfs - 2/2

- Uploading (does not work for me)
 - `cp <fn> \`
`~/mnt/flickr/stream:tag1:tag2:tag3/`
 - `chmod 755 ~/mnt/flickr/stream/<fn>`
 - `ln ~/mnt/flickr/stream/<fn> \`
`~/mnt/flickr/sets/<set>/`

SMB for FUSE

- Seamlessly browse your network neighbourhood
- <http://www.ricardis.tudelft.nl/~vincent/fusesmb/>
- Implementation: C
- Install:
 - `sudo emerge -av fusesmb`
 - `sudo apt-get install fusesmb`
 - `sudo yum install fusesmb`
- Use:
 - `mkdir -p ~/mnt/smb`
 - `smbtree -N`
 - `fusesmb ~/mnt/smb`
 - `ls ~/mnt/smb`
 - `fusermount -u ~/mnt/smb`

WikipediaFS

- View and edit Wikipedia articles as if they were real files!
- <http://wikipediafs.sourceforge.net/>
- Implementation: Python
- Install:
 - http://bugs.gentoo.org/show_bug.cgi?id=183705
 - `sudo apt-get install wikipediafs`
 - `sudo yum install wikipediafs`
- Use:
 - `mkdir ~/mnt/wiki`
 - `mount.wikipediafs ~/mnt/wiki`
 - `mkdir ~/mnt/wiki/wikipedia-en`
 - `ls ~/mnt/wiki/wikipedia-en/`
 - `fusermount -u ~/mnt/wiki`
- Does not work for me :(

CurlFtpFS

- Mount FTP sites
- <http://curlftpfs.sourceforge.net/>
- Implementation: C
- Install:
 - `sudo emerge -av curlftpfs`
 - `sudo apt-get install curlftpfs`
 - `sudo yum install curlftpfs`
- Use:
 - `mkdir ~/mnt/gentoo`
 - `curlftpfs ftp://ftp.ecc.u-tokyo.ac.jp/GENTOO \`
`~/mnt/gentoo`
 - `cp ~/mnt/gentoo/snapshots/portage-`
`latest.tar.bz2 \`
`/tmp/`
 - `fusermount -u ~/mnt/gentoo`

Other FUSE Filesystems 1/4

- OWFS
 - <http://www.owfs.org/>
 - One Wire File System (OWFS) uses FUSE to expose all the Dallas 1-wire sensors, iButtons and memory chips as a filesystem. Devices are dynamically included in the directory, and properties like temperature are obtained by reading a file.
- PhoneBook
 - <http://www.freenet.org.nz/phonebook>
 - PhoneBook is expressly designed for use in situations where someone can be under pressure (legal, military and/or criminal) to disclose decryption keys, and has a 'chaffing' scheme whereby the user can disclose only passphrases for non-sensitive material, and credibly deny the existence of anything else.

Other FUSE Filesystems 2/4

- KIO Fuse Gateway
 - <http://kde.ground.cz/tiki-index.php?page=KIO+Fuse+Gateway>
 - This gateway makes it possible to mount ioslaves or a general ioslave-gateway via fuse and make them this way available to all linux apps.
- Bluetooth File System
 - <http://www.mulliner.org/bluetooth/btfs.php>
 - Btfs is a simple application to map some basic bluetooth functions into the filesystem. With btfs a simple `ls DEVICES` shows you all bluetooth devices within range and `cp somefile OPUSH/devicename` sends the given file to the device.
- gphoto2-fuse-fs
 - <http://www.hep.phy.cam.ac.uk/~lester/gphoto2-fuse-fs/>
 - This program allows mounting a gphoto2 based digital camera so that you can access the files via "standard" programs like "ls, cat, tar, gthumb, netscape, firefox, etc" rather than just through "gtkam and gphoto2"

Other FUSE Filesystems 3/4

- BitTorrent File System
 - <http://btslave.sourceforge.net/>
 - BTSlave allows users to mount a BitTorrent? .torrent file as a file system.
- GnomeVFS2 FUSE
 - <http://www.evolware.org/chri/>
 - Gateway between FUSE and Gnome VFS2. Allows you to mount everything that Nautilus can and looks like a directory.
- Fuse::DBI
 - http://www.rot13.org/~dpavlin/fuse_db.html
 - Simple way to export one type of data (e.g. html templates or content) from any database supported by perl's DBI modules back to filesystem for quick editing.

Other FUSE Filesystems 4/4

- FunionFS: An UnionFS over FUSE
- <http://funionfs.apiou.org/>
- FunionFS is the aggregation of two filesystems: a read-only and a read-write one. The read-only filesystem could be a CDROM or a flash disk for an embedded system. The read-write filesystem could be a Ramdisk or a partition on an USB key ... All datas are read from the read-only filesystem if they are not present on the read write one. Data are written to the read-write filesystem. it's the same principle as the unionfs driver used in the well known Knoppix CDROM.
- MythTVfs
- <http://outflux.net/software/pkgsg/mythtvfs-fuse/>
- Designed to communicate with a MythTV backend server. It creates an overlay filesystem that encodes TV Program metadata (title, episode, description) into a filename so that systems that do not natively talk to MythTV can still get information about a given show.

Archive Filesystems

- fusepak - Support for PACK and WAD files
- <http://fusepak.sourceforge.net/>
- Fusepak allows mounting PACK and WAD files (used by many games based on idSoftware engine, ie. Quake, Doom, Half-Life).
- FuseNRG
- <http://fusenrg.sourceforge.net/>
- FuseNRG allows you to mount Ahead Nero(R) NRG files on your Unix system with FUSE. On the mounted directory, there will be an ISO file, equivalent to the original NRG file. Such ISO file can be mounted with fuseiso or burned to a CD with cdrecord or even Ahead Nero(R) itself.

Network Filesystems 1/4

- GfarmFS-FUSE (Japanese project!)
- <http://datafarm.apgrid.org/software/gfarmfs-fu>
- Gfarm is a reference implementation of the Grid Datafarm architecture designed for global petascale data-intensive computing. It provides Gfarm Grid file system that is a shared file system in cluster or Grid that can scale up to petascale storage, and realize scalable I/O bandwidth and scalable parallel processing.

Network Filesystems 2/4

- CORNFS

- <http://ian.blenke.com/projects/cornfs/cornfs.html>
- CORNFS is an attempt at creating a distributed filesystem that mirrors N copies of files across a group of M number of servers. Everything in CORNFS is stored as a file. At any time, it is possible to reconstruct the entire filesystem via a simple overlay rsync from the remote filesystems.

- HTTP-FUSE KNOPPIX (Japanese project!)

- <http://unit.aist.go.jp/itri/knoppix/http-fuse/index-en.html>
- HTTP-FUSE-KNOPPIX-4.0 is only 5MB CD image and enables us to use same contents of 3.8GB DVD KNOPPIX 4.0. We don't need to download 3.8GB iso image at one time and burn DVD.

Network Filesystems 3/4

- FSFS - the Fast Secure File System
 - <http://fsfs.sf.net/>
 - The Fast Secure File System exports files and directories securely over the network, and lets users store and retrieve encrypted data. It moves most cryptography to clients, achieving better scalability.
- Grifi: GridFTP File System
 - <http://grifi.sourceforge.net/>
 - grifi is a virtual file system (developed with FUSE) that allows a user to mount a remote directory using the GridFTP protocol. It is based on the UberFTP client and on FTPFS from the LUGS project.

Network Filesystems 4/4

- BlogFS

- <http://rohanpm.net/blogfs>
- Mount your WordPress (and maybe other MetaWeblog? compatible) blog(s). Supports reading and writing posts.

- 9pfuse

- <http://swtch.com/usr/local/plan9/src/cmd/9pfuse/>
- 9pfuse is a FUSE <-> 9P translator. It is part of Plan 9 from User Space

- DuggFS

- <http://www.omnigia.com/scheme/duggfs/>
- DuggFS uses the Digg API to organize Digg.com content as a file system.

How it Works - hello.c

- <http://fuse.cvs.sourceforge.net/fuse/fuse/example/hello.c?revision=1.21&view=markup>

```
#include <fuse.h>
```

```
static int hello_getattr(const char *path, struct stat *stbuf)
static int hello_readdir(const char *path, void *buf,
    fuse_fill_dir_t filler, off_t offset, struct fuse_file_info *fi)
static int hello_open(const char *path, struct fuse_file_info *fi)
static int hello_read(const char *path, char *buf, size_t size,
    off_t offset, struct fuse_file_info *fi)
```

```
static struct fuse_operations hello_oper = {
    .getattr= hello_getattr,
    .readdir= hello_readdir,
    .open    = hello_open,
    .read    = hello_read,
};
```

```
int main(int argc, char *argv[])
{
    return fuse_main(argc, argv, &hello_oper, NULL);
}
```

How it Works - Userland 1/2

- <http://fuse.cvs.sourceforge.net/fuse/fuse/doc/how-fuse-works?revision=1.2&view=markup>
- `hello.c:main()` calls `fuse_main()`
- <http://fuse.cvs.sourceforge.net/fuse/fuse/lib/helper.c?revision=1.73&view=markup>
- `fuse_main()` calls `fuse_mount()`
- <http://fuse.cvs.sourceforge.net/fuse/fuse/lib/mount.c?revision=1.41&view=markup>
- `fuse_mount()` creates Unix socket pair, then forks and execs `fusermount`, passing it one end of the socket
- <http://fuse.cvs.sourceforge.net/fuse/fuse/util/fusermount.c?revision=1.90&view=markup>
- `fusermount` opens `/dev/fuse` and sends the filehandle over the socket back to `fuse_mount()`
- `fuse_mount()` returns the filehandle to `fuse_main()`

How it Works - Userland 2/2

- <http://fuse.cvs.sourceforge.net/fuse/fuse/doc/how-fuse-works?revision=1.2&view=markup>
- `fuse_main()` calls `fuse_new()` which allocates the fuse datastructure that stores and maintains a cached image of the filesystem data.
- http://fuse.cvs.sourceforge.net/fuse/fuse/lib/fuse.c?revision=1.203&view=markup#l_3017
- `fuse_main()` calls `fuse_loop()` which starts to read the filesystem system calls from the `/dev/fuse`
- http://fuse.cvs.sourceforge.net/fuse/fuse/lib/fuse.c?revision=1.203&view=markup#l_2805
- `fuse_loop()` calls the usermode functions stored in the `fuse_operations` datastructure
- http://fuse.cvs.sourceforge.net/fuse/fuse/lib/fuse_session.c?revision=1.10&view=markup#l_86
- The results of those calls are then written back to the `/dev/fuse` file where they can be forwarded back to the system calls.