FUSE example

[Posted January 27, 2004 by corbet]

```
/*
    FUSE: Filesystem in Userspace
    Copyright (C) 2001  Miklos Szeredi (mszeredi@inf.bme.hu)

    This program can be distributed under the terms of the GNU GPL.
    See the file COPYING.
*/

#ifdef linux
/* For pread()/pwrite() */
#define _XOPEN_SOURCE 500
#endif

#include <fuse.h>
#include <stdio.h>
#include <string.h>
#include <unistd.h>
#include <fcntl.h>
#include <dirent.h>
#include <errno.h>
#include <sys/statfs.h>

static int xmp_getattr(const char *path, struct stat *stbuf)
{
    int res;

    res = lstat(path, stbuf);
    if(res == -1)
        return -errno;

    return 0;
}

static int xmp_readlink(const char *path, char *buf, size_t size)
{
    int res;
    res = readlink(path, buf, size - 1);
    if(res == -1)
        return -errno;

    buf[res] = '\0';
    return 0;
}


static int xmp_getdir(const char *path, fuse_dirh_t h, fuse_dirfil_t filler)
{
    DIR *dp;
    struct dirent *de;
    int res = 0;

    dp = opendir(path);
    if(dp == NULL)
        return -errno;

    while((de = readdir(dp)) != NULL) {
        res = filler(h, de->d_name, de->d_type);
        if(res != 0)
            break;
    }

    closedir(dp);
    return res;
}

static int xmp_mknod(const char *path, mode_t mode, dev_t rdev)
{
    int res;

    res = mknod(path, mode, rdev);
    if(res == -1)
        return -errno;

    return 0;
}

static int xmp_mkdir(const char *path, mode_t mode)
{
    int res;

    res = mkdir(path, mode);
    if(res == -1)
        return -errno;

    return 0;
}
```

```
static int xmp_unlink(const char *path)
{
    int res;

    res = unlink(path);
    if(res == -1)
        return -errno;

    return 0;
}

static int xmp_rmdir(const char *path)
{
    int res;

    res = rmdir(path);
    if(res == -1)
        return -errno;

    return 0;
}

static int xmp_symlink(const char *from, const char *to)
{
    int res;

    res = symlink(from, to);
    if(res == -1)
        return -errno;

    return 0;
}

static int xmp_rename(const char *from, const char *to)
{
    int res;

    res = rename(from, to);
    if(res == -1)
        return -errno;

    return 0;
}

static int xmp_link(const char *from, const char *to)
{
    int res;

    res = link(from, to);
    if(res == -1)
        return -errno;

    return 0;
}

static int xmp_chmod(const char *path, mode_t mode)
{
    int res;

    res = chmod(path, mode);
    if(res == -1)
        return -errno;

    return 0;
}

static int xmp_chown(const char *path, uid_t uid, gid_t gid)
{
    int res;

    res = lchown(path, uid, gid);
    if(res == -1)
        return -errno;

    return 0;
}

static int xmp_truncate(const char *path, off_t size)
{
    int res;

    res = truncate(path, size);
    if(res == -1)
        return -errno;

    return 0;
}

static int xmp_utime(const char *path, struct utimbuf *buf)
{
    int res;

    res = utime(path, buf);
    if(res == -1)
        return -errno;

    return 0;
}
```

```
static int xmp_open(const char *path, int flags)
{
    int res;

    res = open(path, flags);
    if(res == -1)
        return -errno;

    close(res);
    return 0;
}

static int xmp_read(const char *path, char *buf, size_t size, off_t offset)
{
    int fd;
    int res;

    fd = open(path, O_RDONLY);
    if(fd == -1)
        return -errno;

    res = pread(fd, buf, size, offset);
    if(res == -1)
        res = -errno;

    close(fd);
    return res;
}

static int xmp_write(const char *path, const char *buf, size_t size,
                     off_t offset)
{
    int fd;
    int res;

    fd = open(path, O_WRONLY);
    if(fd == -1)
        return -errno;

    res = pwrite(fd, buf, size, offset);
    if(res == -1)
        res = -errno;

    close(fd);
    return res;
}

static int xmp_statfs(struct fuse_statfs *fst)
{
    struct statfs st;
    int rv = statfs("/",&st);
    if(!rv) {
        fst->block_size  = st.f_bsize;
        fst->blocks      = st.f_blocks;
        fst->blocks_free = st.f_bavail;
        fst->files       = st.f_files;
        fst->files_free  = st.f_ffree;
        fst->namelen     = st.f_namelen;
    }
    return rv;
}

static int xmp_release(const char *path, int flags)
{
    /* Just a stub.  This method is optional and can safely be left
       unimplemented */

    (void) path;
    (void) flags;
    return 0;
}

static int xmp_fsync(const char *path, int isdatasync)
{
    /* Just a stub.  This method is optional and can safely be left
       unimplemented */

    (void) path;
    (void) isdatasync;
    return 0;
}

static struct fuse_operations xmp_oper = {
    .getattr    = xmp_getattr,
    .readlink   = xmp_readlink,
    .getdir     = xmp_getdir,
    .mknod      = xmp_mknod,
    .mkdir      = xmp_mkdir,
    .symlink    = xmp_symlink,
    .unlink     = xmp_unlink,
    .rmdir      = xmp_rmdir,
    .rename     = xmp_rename,
    .link       = xmp_link,
    .chmod      = xmp_chmod,
    .chown      = xmp_chown,
    .truncate   = xmp_truncate,
    .utime      = xmp_utime,
    .open       = xmp_open,
    .read       = xmp_read,
    .write      = xmp_write,
    .statfs     = xmp_statfs,
    .release    = xmp_release,
    .fsync      = xmp_fsync

};

int main(int argc, char *argv[])
{
    fuse_main(argc, argv, &xmp_oper);
    return 0;
}
```

(Log in to post comments)

FUSE example
Posted Jul 29, 2015 13:45 UTC (Wed) by **sharut** (guest, #103787) [Link]

I'am new in FUSE and this example will help me. But when I've try to compile following example I've got following error messages:

```
warning: ‘struct fuse_statfs’ declared inside parameter list [enabled by default]
xmp.c:236:30: warning: its scope is only this definition or declaration, which is probably not what you want [enabled by default]
xmp.c: In function ‘xmp_statfs’:
xmp.c:241:6: error: dereferencing pointer to incomplete type
xmp.c:242:6: error: dereferencing pointer to incomplete type
xmp.c:243:6: error: dereferencing pointer to incomplete type
xmp.c:244:6: error: dereferencing pointer to incomplete type
xmp.c:245:6: error: dereferencing pointer to incomplete type
xmp.c:246:6: error: dereferencing pointer to incomplete type
xmp.c: At top level:
xmp.c:289:2: warning: initialization from incompatible pointer type [enabled by default]
xmp.c:289:2: warning: (near initialization for ‘xmp_oper.statfs’) [enabled by default]
```

Will you please clarify the situation?

Reply to this comment

FUSE example
Posted Jul 29, 2015 23:33 UTC (Wed) by **lsl** (subscriber, #86508) [Link]

The example is quite old, prior to the merging of FUSE into Linux. You can find an updated version with the libfuse sources.

http://fuse.sourceforge.net/doxygen/fusexmp_8c.html

You could also define FUSE_USE_VERSION to some older version, probably 1.X. See the comment in fuse.h for that.

Reply to this comment