FUSE
(Filesystem in Userspace)

Krerk Piromsopa, Ph.D.

Why Userspace?

- Writing (good) code is not easy.

Let’s speed it up….

- Writing (good) code in Kernel is HARD!
    - No libc (no printf, Qt, stdio…)
- Too many reboots, Kernel Panic

Building a filesystem is difficult?

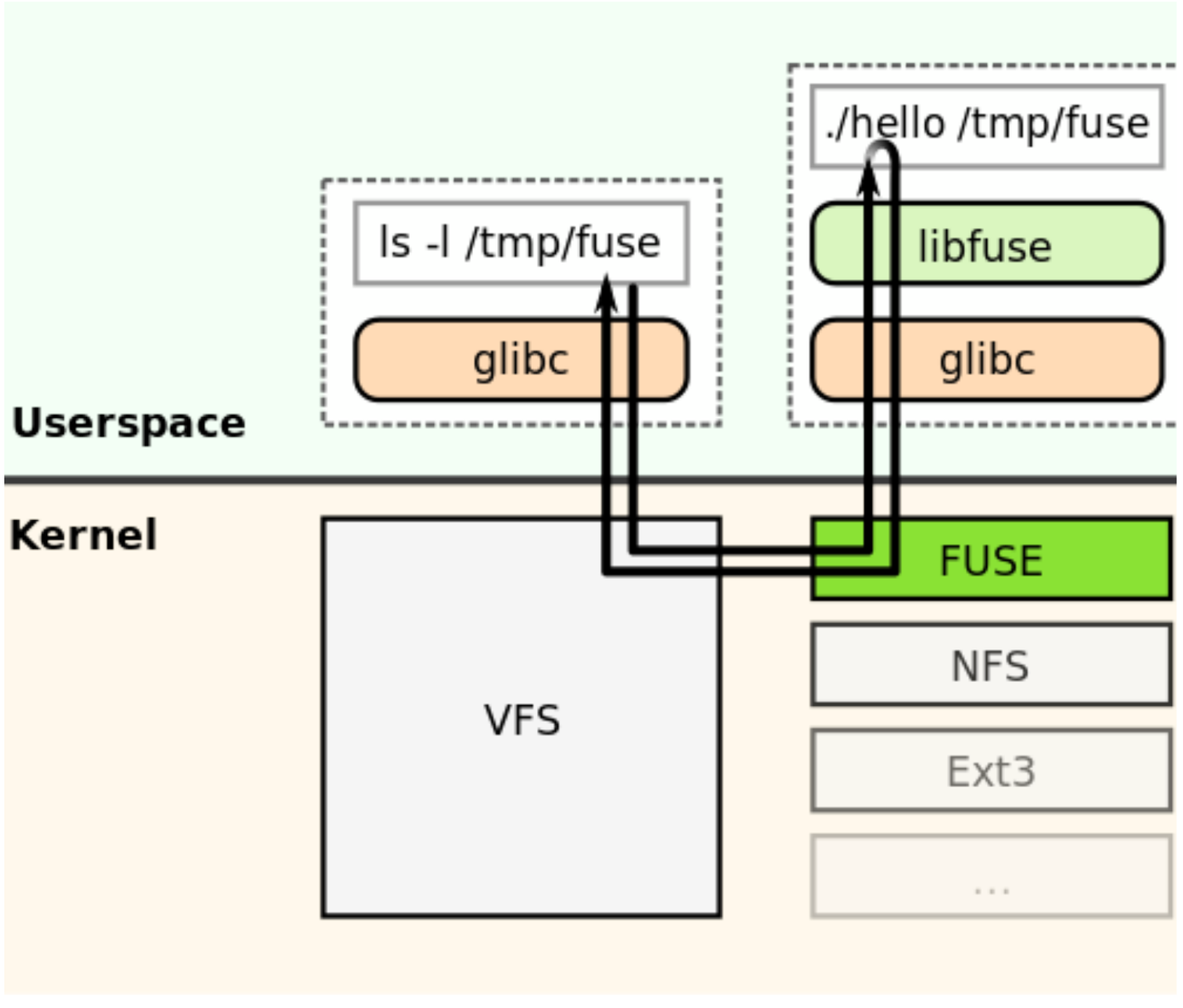
- Performance reduce disk seek
- Life cycle of disk (SSD block limited life cycle)
- Maximum filesize limitations
- Metadata, permission, security (encryption)
- Deduplication, compression, snapshots
- etc….

What is FUSE?

- Writing a filesystem in your favorite environments (language, library, debugging tool)
- e
  running file system code in user space while the FUSE module provides only a "bridge" to the actual kernel interfaces. [wikipedia]

How does it work

- File/Filesystem API are passed to a process.
- A kernel module (driver) is a bridge
- Available on Linux, BSD (MacOSX), Android, Minix, …
- Doken is a Windows implementation (incomplete?)



binding for C, Python, Objective-C, Ruby, Java, C#,…..

Operations (1)

- int (*getattr) (const char *, struct stat *);
    - Get file attributes.
- int (*readlink) (const char *, char *, size_t);
    - Read the target of a symbolic link
- int (*mknod) (const char *, mode_t, dev_t);
    - Create a file node.
- int (*mkdir) (const char *, mode_t);
    - Create a directory. Note that the mode argument may not have the type specification bits set, i.e. S_ISDIR(mode) can be false. To obtain the correct directory type bits use mode | S_IFDIR

https://lastlog.de/misc/fuse-doc/doc/html/structfuse__operations.html

Operations (2)

- int (*unlink) (const char *);
  - Remove a file
- int (*rmdir) (const char *);
  - Remove a directory
- int (*symlink) (const char *, const char *);
  - Create a symbolic link
- int (*rename) (const char *, const char *);
  - Rename a file
- int (*link) (const char *, const char *);
  - Create a hard link to a file
- int (*chmod) (const char *, mode_t);
  - Change the permission bits of a file
- int (*chown) (const char *, uid_t, gid_t);
  - Change the owner and group of a file
- int (*truncate) (const char *, off_t);
  - Change the size of a file
- int (*open) (const char *, struct fuse_file_info *);
  - File open operation.
-

https://lastlog.de/misc/fuse-doc/doc/html/structfuse__operations.html

Operations (3)

- int (*read) (const char *, char *, size_t, off_t, struct fuse_file_info *);
    - Read data from an open file.
- int (*write) (const char *, const char *, size_t, off_t, struct fuse_file_info *);
    - Write data to an open file
- int (*statfs) (const char *, struct statvfs *);
    - Get file system statistics
- int (*flush) (const char *, struct fuse_file_info *);
    - Possibly flush cached data

Operations (4)

- int (*opendir) (const char *, struct fuse_file_info *);
    - Open directory. Unless the 'default_permissions' mount option is given, this method should check if opendir is permitted for this directory. Optionally opendir may also return an arbitrary filehandle in the fuse_file_info structure, which will be passed to readdir, closedir and fsyncdir.
- int (*readdir) (const char *, void *, fuse_fill_dir_t, off_t, struct fuse_file_info *);
    - Read directory
- int (*releasedir) (const char *, struct fuse_file_info *);
    - Release directory
- int (*fsyncdir) (const char *, int, struct fuse_file_info *);
    - Synchronize directory contents
- void *(*init) (struct fuse_conn_info *conn);
    - Initialize file system.
- int (*lock) (const char *, struct fuse_file_info *, int cmd, struct flock *);
    - Perform POSIX file locking operation

Some Implementations

- GmailFS
- EncFS
- NTFS-3G
- WikipediaFS
- Lustre
- SSHFS
- FTPFS
- ImapFS
- YoutubeFS
- Gdrive, Grive?
- gitFS
- etc… (build your own)

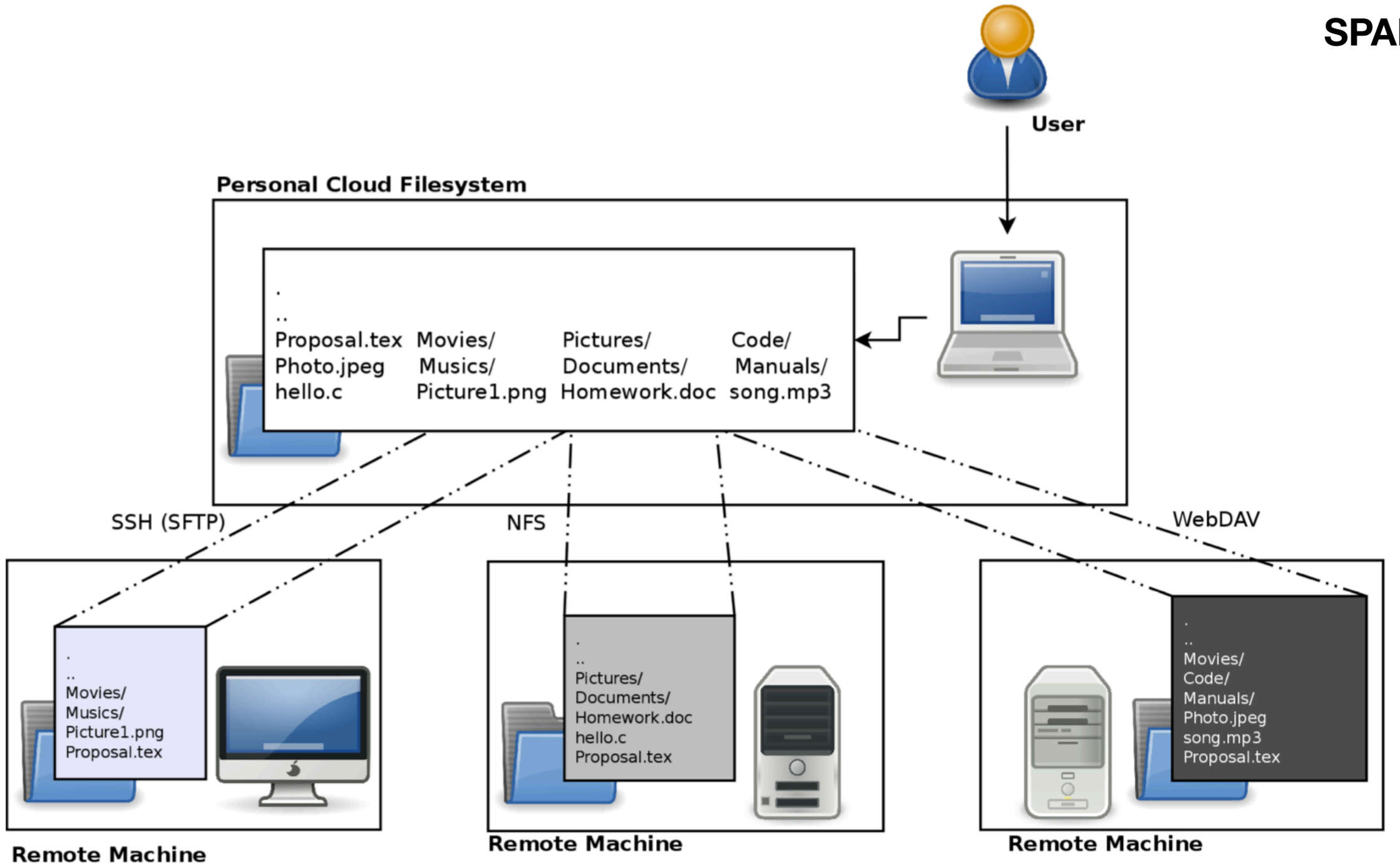
Some from my collections

- S. Dhumbumroong and K. Piromsopa, “Personal Cloud Filesystem: A distributed unification filesystem for personal computer and portable device,” in Computer Science and Software Engineering (JCSSE), 2011 Eighth International Joint Conference on, 2011, pp. 58–62.
  (for both Linux and Windows) **SPAFS**

- W. Ratinimittum and K. Piromsopa, “An implementation of RESTful-based Scalable File System,” in JCSSE 2012 - 9th International Joint Conference on Computer Science and Software Engineering, 2012, pp. 136–141.

- More from students’ projects in the past

SPAFS


User
Personal Cloud Filesystem
.
..
Proposal.tex Movies/ Pictures/ Code/
Photo.jpeg Musics/ Documents/ Manuals/
hello.c Picture1.png Homework.doc song.mp3
SSH (SFTP)
NFS
WebDAV
.
..
Movies/
Musics/
Picture1.png
Proposal.tex
Remote Machine
.
..
Pictures/
Documents/
Homework.doc
hello.c
Proposal.tex
Remote Machine
.
..
Movies/
Code/
Manuals/
Photo.jpeg
song.mp3
Proposal.tex
Remote Machine

Python-FUSE API

- create(path, mode)
- truncate(path, size)
- mknod(path, mode, dev)
- open(path, mode)
- write(path, data, offset)
- read(path, length, offset)
- release(path)
- fsync(path)
- chmod(path, mode)
- chown(path, oid, gid)
- mkdir(path, mode)
- unlink(path)
- readdir(path)
- rmdir(path)
- rename(opath, npath) l
- ink(srcpath, dstpath)

File Operations and API

Reading
(e.g. cat a.txt)

getattr

open

read

release

Writing
(e.g. echo “a” > a.txt)

getattr

create

write

flush

release

Appending
(e.g. echo “a” >> a.txt)

getattr

open

write

flush

release

Truncating
(e.g. echo “a” > a.txt)

getattr

truncate

open

write

flush

release

Removing
(e.g. rm a.txt)

getattr

unlink

Hint… Try strace

File Operations and API

Creating Directory
(e.g. mkdir demo)

getattr

mkdir

Reading Directory
(e.g. ls demo/)

getattr

opendir

readdir

releasedir

Removing Directory
(e.g. rmdir demo)

getattr

rmdir

Changing Permission
(e.g. chmod 777 demo)

getattr

chmod

Changing Ownership
(e.g. chmod 777 demo)

getattr

chown

Linking
(e.g. ln -s demo d)

getattr

symlink

Renaming
(e.g. mv demo d)

getattr

rename

Let’s see a demo

MyFS.py

```
#!/usr/bin/env python2

# 2110352 Operating System
# FUSE/Filesystem exercise
# By Krerk Piromsopa, Ph.D.
<Krerk.P@chula.ac.th>
#    Department of Computer Engineering
#    Chulalongkorn University.

import os, stat, errno
import fuse
from fuse import Fuse

if not hasattr(fuse, '__version__'):
    raise RuntimeError, \
        "your fuse-py doesn't know of
fuse.__version__, probably it's too old."

fuse.fuse_python_api = (0, 2)

containers={ \
                '/subject':"2018S1 -
Operating Systems\nCP ENG CU\n" ,
                '/instructors':"0:CP ENG CU
OS 2018S1 - Instructors\n" \
"1:      Veera Muangsin, Ph.D.\n" \
"2:      Krerk Piromsopa, Ph. D.\n" \
"3:      Kunwadee Sripanidkulchai, Ph.D.\n"
"4:      Thongchai Rojkangsadan\n"
        }

class MyStat(fuse.Stat):
    def __init__(self):
        self.st_mode = 0
        self.st_ino = 0
        self.st_dev = 0
        self.st_nlink = 0
        self.st_uid = 0
        self.st_gid = 0
        self.st_size = 0
        self.st_atime = 0
        self.st_mtime = 0
        self.st_ctime = 0
```

MyFS.py (ctd)

```
class MyFS(Fuse):

    def getattr(self, path):
        st = MyStat()
        if path == '/':
            st.st_mode = stat.S_IFDIR | 0777
            st.st_nlink = 2
        elif path in containers:
            st.st_mode = stat.S_IFREG | 0444
            st.st_nlink = 1
            content=containers[path]
            st.st_size = len(content)
        else:
            return -errno.ENOENT
        return st

    def readdir(self, path, offset):
        filenames=containers.keys()
        for r in  '.', '..':
            yield fuse.Direntry(r)

        for r in filenames:
            yield fuse.Direntry(r[1:])
```

```
    def open(self, path, flags):
        if path not in containers:
            return -errno.ENOENT
        accmode = os.O_RDONLY | os.O_WRONLY
| os.O_RDWR
        if (flags & accmode) != os.O_RDONLY:
            return -errno.EACCES

    def read(self, path, size, offset):
        if path not in containers:
            return -errno.ENOENT

        content = containers[path]
        slen = len(content)
        if offset < slen:
            if offset + size > slen:
                size = slen - offset
            buf =
content[offset:offset+size]
        else:
            buf = ''
        return buf
```

Let’s rethink and build your own filesystem