

The K Project

The K File System

LSE Team

EPITA

April 26, 2017

- Used to control how data is stored and retrieved
- Space management:
 - FAT12, FAT16, FAT32
 - NTFS
 - ext2, ext3, ext4
 - UFS
- User interface:
 - sysfs
 - procfs

- the superblock is your file system's entry point
- it's the first module in the multiboot struct:

```
(void *)((module_t*)info->mods_addr)[0].mod_start
```

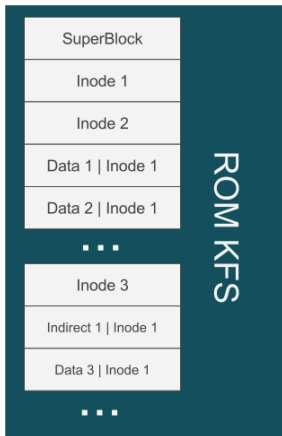


Figure: KFS Structure

- an inode describes a file representation.
- an indirect block references data blocks.
- a data block contains parts of the file data.
- structures are located in `include/k/kfs.h`

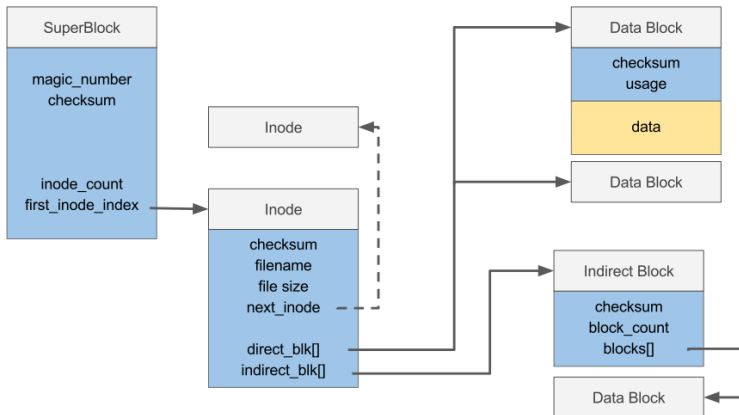


Figure: KFS

SERIOUSLY READ THE HEADER

- `include/k/kfs.h`

- get your superblock
- check your magic number
- check the checksum of your superblock
- allocate as a file descriptor table `inode_count` number of file structs
- memset this memory buffer

The K Project

LSE Team

File System

Conclusion

```
int open(const char *pathname, int flags);
ssize_t read(int fd, void *buf, size_t count);
off_t seek(int fd, off_t offset, int whence);
int close(int fd);
```

Open:

- search for the inode which filename is the pathname
- register an fd in your fd table

Close:

- free the fd

Seek:

- modify the file offset according to whence

Read:

- get the inode associated to the fd
- find the blocks of data to read and copy it
- modify the file offset

The K Project

LSE Team

File System

Conclusion

- #k (irc.rezosup.org)
- labos.lse with [K] tag
- k[at]lse.epita.fr
- xdbob[at]lse.epita.fr
- pierre.marsais[at]lse.epita.fr