**操作系统课程实践（网安）**

# 实验 5：简单文件系统的实现

## 实验介绍

本实验通过具体编写文件存储空间的管理、文件物理结构、目录结构和文件操作的实现，加深对文件系统内部数据结构、功能以及实现过程的理解。

## 任务描述

在内存中开辟一个虚拟磁盘空间作为文件存储分区，在其上实现一个简单的基于多级目录的单用户单任务系统中的文件系统。在退出该文件系统的使用时，应将虚拟磁盘上的内容以一个文件的方式保存到磁盘上，以便下次可以再将它恢复到内存的虚拟磁盘中。

文件物理结构可采用显式链接或其他结构。

空闲磁盘空间的管理可选择FAT表、位示图或其他办法。

文件目录结构采用多级目录结构。为简单起见，可以不使用索引结点，每个目录项应包含文件名、物理地址、长度等信息，还可以通过目录项实现对文件的读写保护。

## 实验目的

通过具体的文件存储空间的管理、文件物理结构、目录结构和文件操作的实现，加深对文件系统内部数据结构、功能以及实现过程的理解。

通过用户态文件系统框架FUSE，实现一个可以挂在到Linux或Windows下的文件系统。

请根据自身情况，进一步阅读分析相关系统调用的内核源码实现。

## 实验内容

1. 文件系统预备知识

### 参考教材7.6.2节，预备知识。

1. 文件系统的设计与实现（下面两种方式采用一种即可）

### 参考教材7.6.3节实例系统的设计与实现，采用命令方式演示文件系统的工作流程，参考教材，程序能够提供以下操作命令：

* my\_format：对文件存储器进行格式化，即按照文件系统的结构对虚拟磁盘空间进行布局，并在其上创建根目录以及用于管理文件存储空间等的数据结构。
* my\_mkdir：用于创建子目录。
* my\_rmdir：用于删除子目录。
* my\_ls：用于显示目录中的内容。
* my\_cd：用于更改当前目录。
* my\_create：用于创建文件。
* my\_open：用于打开文件。
* my\_close：用于关闭文件。
* my\_write：用于写文件。
* my\_read：用于读文件。
* my\_rm：用于删除文件。
* my\_exitsys：用于退出文件系统。

### 参考教材7.6.3节实例系统的设计与实现，采用用户态文件系统框架FUSE实现文件系统，并能够挂载，挂载后文件系统需能够支持：

* 文件的创建 touch test.txt
* 文件的写入 echo hello > test.txt
* 文件的读出 cat test.txt
* 文件的编辑 vim test.txt
* 文件的删除 rm test.txt
* 文件夹的创建 mkdir testdir
* 进入文件夹 cd testdir
* 文件夹内完成上述文件创建、写入、读出、编辑、删除操作
* 文件夹的删除 rmdir testdir
* 文件列表命令 ls -l

FUSE的官方网站<https://github.com/libfuse/libfuse>，目前FUSE版本是3.0，和前面版本有较大区别。FUSE的接口函数介绍可以参考<https://blog.csdn.net/u012587561/article/details/50908656>（该版本为FUSE2.6，注意差异），此外大家可以自行百度FUSE的相关资料。OpenEuler系统上编译并安装Fuse3.0的步骤如下：

### 先安装libiconv：

**wget https://ftp.gnu.org/pub/gnu/libiconv/libiconv-1.16.tar.gz**

**tar zxvf libiconv-1.16.tar.gz**

**cd libiconv-1.16**

**./configure**

**make**

**make install**

**echo /usr/local/lib > /etc/ld.so.conf.d/libiconv.conf**

**ldconfig**

### 安装libfuse

**yum install meson**

**wget** [**https://github.com/libfuse/libfuse/releases/download/fuse-3.10.3/fuse-3.10.3.tar.xz**](https://github.com/libfuse/libfuse/releases/download/fuse-3.10.3/fuse-3.10.3.tar.xz)

**tar xvf fuse-3.10.3.tar.xz**

**cd fuse-3.10.3**

**mkdir build; cd build**

**meson ..**

**meson configure**

**ninja**

**ninja install**

**ldconfig**

### Example目录下的hello.c是一个简单的文件系统例子（写死一个只读文件），大家可以学习下。编译后的hello在build目录下的example目录下：

**cd exmaple**

**mkdir /mnt/hello**

**./hello /mnt/hello**

### simplefs是一个简单的采用FUSE3.0实现的内存文件系统，只实现了一层目录结构，演示了fuse接口函数的使用，大家可以参考。该程序也可以在Windows下采用winfsp编译。

