OS Lab5

学号：181860154

姓名：朱倩

邮箱：infinite0124@163.com

一、实验进度

完成了 **1. 完善格式化程序** 部分，即`func.c`中的`rmdir`和`rm`函数。

二、实验结果

1. **rm的测试：**

   在main.c中末尾添加

   ```
   ls /usr
   rm /usr/reader_writer
   ls /usr
   ```

   这三条指令的相关测试代码，make结果：

   ```
   ls /usr
   Name: ., Inode: 4, Type: 2, LinkCount: 2, BlockCount: 1, Size: 1024.
   Name: .., Inode: 1, Type: 2, LinkCount: 4, BlockCount: 1, Size: 1024.
   Name: print, Inode: 5, Type: 1, LinkCount: 1, BlockCount: 14, Size: 13604.
   Name: bounded_buffer, Inode: 6, Type: 1, LinkCount: 1, BlockCount: 14, Size: 13656.
   Name: philosopher, Inode: 7, Type: 1, LinkCount: 1, BlockCount: 14, Size: 13716.
   Name: reader_writer, Inode: 8, Type: 1, LinkCount: 1, BlockCount: 14, Size: 13652.
   LS success.
   1016 inodes and 3887 data blocks available.
   rm /usr/reader_writer
   RM success.
   1017 inodes and 3901 data blocks available.
   ls /usr
   Name: ., Inode: 4, Type: 2, LinkCount: 2, BlockCount: 1, Size: 1024.
   Name: .., Inode: 1, Type: 2, LinkCount: 4, BlockCount: 1, Size: 1024.
   Name: print, Inode: 5, Type: 1, LinkCount: 1, BlockCount: 14, Size: 13604.
   Name: bounded_buffer, Inode: 6, Type: 1, LinkCount: 1, BlockCount: 14, Size: 13656.
   Name: philosopher, Inode: 7, Type: 1, LinkCount: 1, BlockCount: 14, Size: 13716.
   LS success.
   1017 inodes and 3901 data blocks available.
   ```

   可以看到，在执行`rm /usr/reader_writer`指令后，可用的**inode数量**从1016变为1017，增加1；可用的**data block数量**从3887变为3901，增加14，为删除文件`reader_writer`的BlockCount.

   删除文件`reader_writer`后再次执行`ls /usr`指令，**没有看到文件**`reader_writer`。

   由此，删除文件时回收相应的iNode结构和数据块, 然后从父目录的表项中删除相应表项的功能基本得以验证。

2. **rmdir的测试：**

在main.c中末尾添加

```
rmdir /usr
ls /
```

这两条指令的相关测试代码，make结果：

```
rmdir /usr
rm /usr/.
RM success.
1018 inodes and 3902 data blocks available.
rm /usr/print
RM success.
1019 inodes and 3916 data blocks available.
rm /usr/bounded_buffer
RM success.
1020 inodes and 3930 data blocks available.
rm /usr/philosopher
RM success.
1021 inodes and 3944 data blocks available.
RMDIR success.
1021 inodes and 3944 data blocks available.
ls /
Name: ., Inode: 1, Type: 2, LinkCount: 4, BlockCount: 1, Size: 1024.
Name: .., Inode: 1, Type: 2, LinkCount: 4, BlockCount: 1, Size: 1024.
Name: boot, Inode: 2, Type: 2, LinkCount: 2, BlockCount: 1, Size: 1024.
LS success.
1021 inodes and 3944 data blocks available.
```

可以看到，执行rmdir指令后，递归地调用了rm函数进行目录下文件的删除（.文件为目录本身），可用的**inode数量**逐步变为1021，可用的**data block数量**逐步变为3944。

而执行`mkdir /usr`指令前inode数量和data block数量也分别为1021和3944，和上述结果相符。

```
1021 inodes and 3944 data blocks available.
mkdir /usr
MKDIR success.
1020 inodes and 3943 data blocks available.
```

`rmdir /usr`后执行`ls /`指令，可以看到`/`目录下不再有usr目录。

由此，删除目录时递归地对目录中的文件进行删除，并回收相应的iNode结构和数据块，然后从父目录的表项中删除相应表项的功能基本得以验证。

三、实验修改的代码位置

1. **删除文件rm：**

   lab4/utils/genFS/func.c:

   ```
   修改函数：int rm (const char *driver, const char *destFilePath);
   ```

   大致思路：

   （1）回收相应的inode结构和数据块： 将inodeBitmap， blockBitmap相应位置的bit置0，并更新superBlock的availInodeNum和availBlockNum。

   （2）从父目录的表项中删除相应文件表项

2. **删除目录（及目录下文件）rmdir：**

lab4/utils/genFS/func.c:

```
修改函数：int rmdir (const char *driver, const char *destDirPath);
```

大致思路：

（1）递归地对目录中的文件进行删除，同时释放目录本身所占的inode和data block

（2）从父目录的表项中删除该目录表项

3. **测试rm和rmdir功能：**

   lab4/utils/genFS/main.c:

   ```
   修改函数：int main(int argc, char *argv[]);
   ```

   添加rm和rmdir相关测试

Thanks for reading!