
EXP5 简易文件管理系统

汪楚文 2018202114&刘流 2018202116 06/18/2020

Copyright © 2020- by Wangchuwen, Liuliu, RUC. All rights reserved

实验概述

[实验环境]

Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-88-generic x86_64)

[功能要求]

- ①至少支持普通文件和目录文件两种类型
- ②创建新文件，删除已有文件
- ③根据文件名（绝对路径）检索到相应的文件（/d0/d1/filename）
- ④支持的操作：open、close、read、write
- ⑤至少支持三层目录（/d0/d1/filename）
- ⑥能将外部文件导入到文件系统中,以及将文件系统中的文件导出
- ⑦提供命令行界面，能够显示目录内容和文件内容

[实现要求]

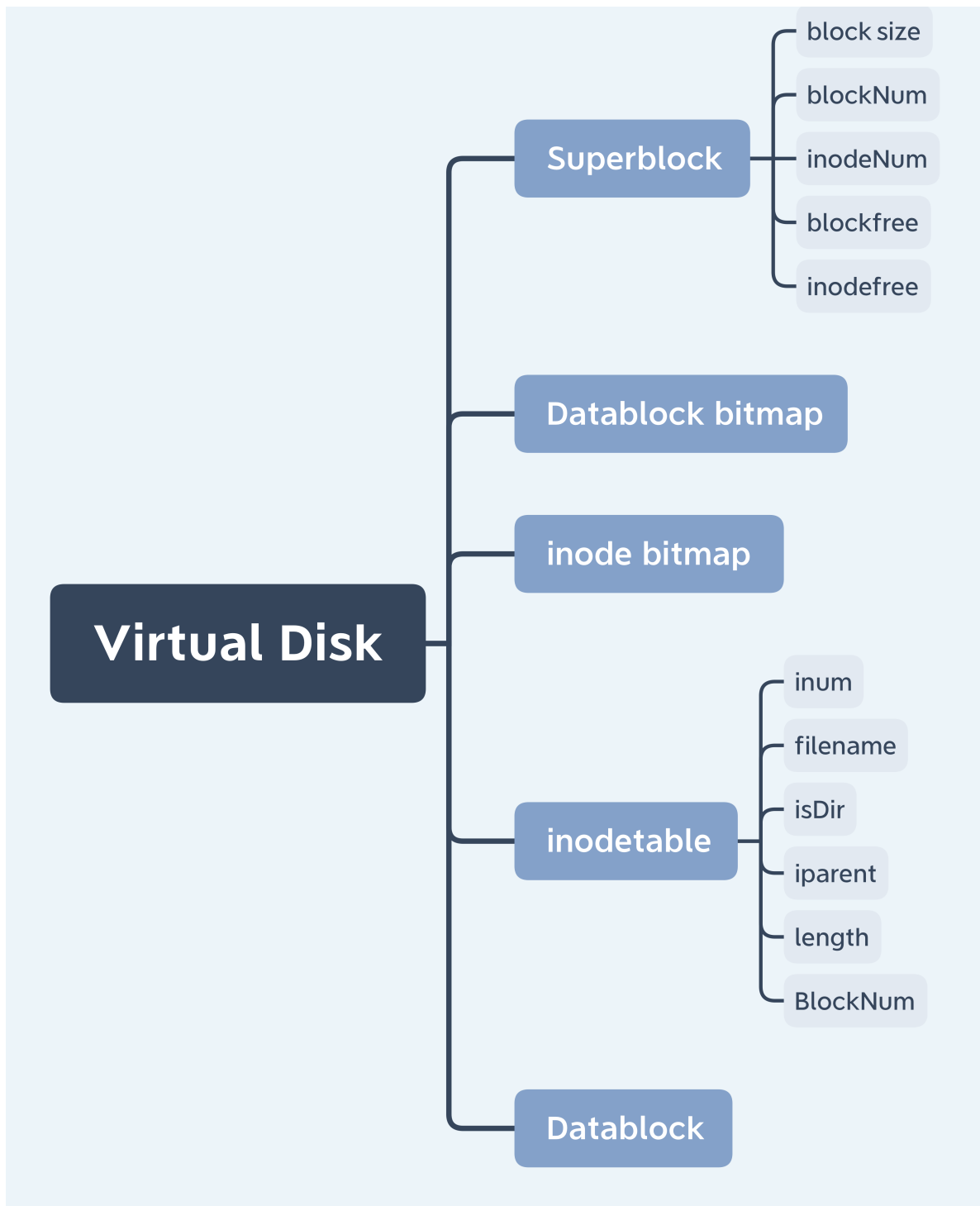
- ①使用一个空文件模拟一块磁盘，大小为 1MB，可预先创建好。
- ②以块为存取的基本单位，大小为 1KB
- ③每个文件最多占有 8 块，即文件的最大大小为 8KB
- ④文件名最长 8B
- ⑤简易文件系统支持最多 80 个文件（含目录文件）
- ⑥索引节点（inode）：
 - （1）每个索引节点大小 64B（1 个块可以存放 16 个 inode）
 - （2）系统需要维护 inode bitmap，记录每个索引节点分配与否
 - （3）每个索引节点至少包含：文件标识符、文件涉及的块（地址或块号）、文件创建时间等
 - （4）采用直接索引方式获取块位置
 - （5）根节点已知（即/目录对应的索引节点位置已知）
- ⑦需要实现两个基本函数：GETBLK(buf, blk_no); PUTBLK(buf, blk_no); 两个函数用于读取和写入指定块。

实验内容（思路）

[要求实现情况]

本实验的所有要求均已实现。

一.基本框架



二.函数功能简要说明

文件名	WenFS.c	主要函数功能说明
		<code>void createFileSystem()</code> 用来创建文件系统 <code>void openFileSystem()</code> 用来启动文件系统 <code>void createFile(char *name,int flag)</code> 用来创建文件或者创建目录 <code>void list()</code> 用来显示目录及其下文件信息 <code>int findInodeNum(char *name,int flag)</code> 用来寻找相应的inode <code>void cd(char *name)</code> 用来切换目录 <code>void cdParent()</code> 用来返回上一级目录 <code>void write(char *name)</code> 用来写文件 <code>void read(char *name)</code> 用来读文件 <code>void movein(char *name,char *name1)</code> 将文件系统外文件copy到文件系统里 <code>void moveout(char *name,char *name1)</code> 将文件系统里文件copy到文件系统外 <code>void delete(char *name)</code> 删除文件 <code>void updateResource()</code> 更新资源 <code>void pathSet()</code> 设置路径 <code>void systemInfo()</code> 输出文件系统信息 <code>void help()</code> 输出提示 <code>int analyse(char *str)</code> 分析输入命令 <code>void command(void)</code> 执行输入命令

三.结构说明

[1]SuperBlock

SuperBlock描述磁盘的整体信息，储存在磁盘文件偏移量为0的位置，每次对文件系统进行操作后需要更新SuperBlock信息。

[2]blockbitmap&inodebitmap

用char数组储存两个bitmap，0表示空闲，1表示占用

[3]inodetable

inodetable由两部分组成：Fcb和inode，inode用来存储文件名，文件类型，所占block等信息，Fcb用来构建目录信息链。

四.实现原理

[1]读文件

首先检查文件是否为目录，若是，则输出提示信息：This is a directory,not a file...

若为文件，则通过inodenum的返回值确定文件是否存在，若不存在，则输出提示信息：This is no %s file...，若存在，则通过fseek和fread找到其所在的datablock，然后逐行输出文件内容。

[2]写文件

首先检查文件是否为目录，若是，则输出提示信息：This is a directory,not a file...

若为文件，则通过inodenum的返回值确定文件是否存在，若不存在，则输出提示信息：This is no %s file...，若存在，则通过fseek和fread找到其所在的datablock，然后写入输入的内容（#为终止），最后再更新inode->length.

[3]创建文件

创建文件时函数第二个参数决定创建的是文件或者是目录。如果符合命名要求，则分配datablock，并更新superblock和blockbitmap里的信息。然后创建inode并初始化，更新inodebitmap，并将文件信息更新在上一级目录里。

[4]删除文件

删除文件时只需要找到对应的inodenum，并更新superblock和blockbitmap，datablock不需要做处理。如果删除的是目录文件，则递归删除目录下的子文件，然后再删除目录文件。

[5]复制文件到文件系统外

找到文件所在的datablock，并将其中的信息copy到buffer，最后打开文件系统外的文件，并把buffer里的内容写入文件中即可。

[6]复制文件到文件系统里

把文件信息写入buffer，然后在文件系统中对应目录下create同名文件，并将buffer里的内容写到该文件中。

[7]文件系统

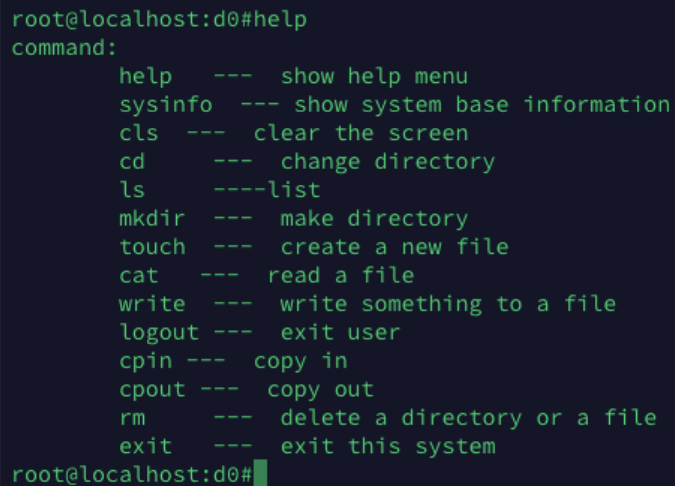
文件系统用file.dat模拟，大小为1M。

[8]命令控制

```
help --- show help menu ; sysinfo --- show system base information
cls --- clear the screen; cd --- change directory ; ls ----list
mkdir --- make directory ; touch --- create a new file; cat --- read a file
write --- write something to a file; logout --- exit user; cpin --- copy in
cpout --- copy out; rm --- delete a directory or a file; exit --- exit this system
```

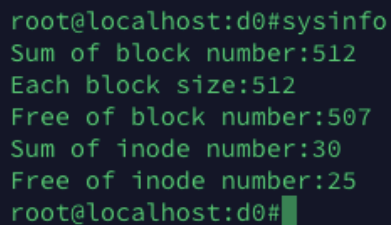
实验结果

一.引导语



```
root@localhost:d0#help
command:
    help    ---  show help menu
    sysinfo ---  show system base information
    cls     ---  clear the screen
    cd      ---  change directory
    ls      ----list
    mkdir   ---  make directory
    touch   ---  create a new file
    cat     ---  read a file
    write   ---  write something to a file
    logout  ---  exit user
    cpin    ---  copy in
    cpout   ---  copy out
    rm      ---  delete a directory or a file
    exit    ---  exit this system
root@localhost:d0#
```

二.sysinfo展示文件系统信息



```
root@localhost:d0#sysinfo
Sum of block number:512
Each block size:512
Free of block number:507
Sum of inode number:30
Free of inode number:25
root@localhost:d0#
```

三.mkdir创建文件夹并list

```
root@localhost:d0#mkdir d4
root@localhost:d0#ls
Filename: d1      Inode number: 4      Directory
Filename: d4      Inode number: 5      Directory
root@localhost:d0#
```

四.touch创建文件file1,后再用rm删除文件

```
root@localhost:d1#touch file1
root@localhost:d1#ls
Filename:          Inode number: 0      Regular file
Filename: file1     Inode number: 6      Regular file
root@localhost:d1#rm file1
root@localhost:d1#ls
Filename:          Inode number: 0      Regular file
root@localhost:d1#
```


七.将file2移动至文件系统内

[在文件系统外创建文件file2]

[illegible]

[将file2移入文件系统]

```
root@localhost:d0#ls
Filename: d1      Inode number: 4      Directory
Filename: d4      Inode number: 5      Directory
Filename: file2   Inode number: 8      Regular file
root@localhost:d0#cat file2
this is a file that is ready to be moved in my visual file system!
root@localhost:d0#
```

遇到的问题和解决办法

1.malloc.h找不到头文件

解决办法：经网上查询，发现在mac OS上此头文件要写作sys/malloc.h，在ubuntu上只需要写malloc.h即可。

2.write函数运行后会导致inode中的信息错误

解决办法：在write时不仅要修改datablock里的数据，还需要修改inode->length。

3.对齐错误

解决办法：结构体在对齐时并不是完全占满整个空间的，在对结构体的对齐规则进行了解后，修改了偏移量，程序即可正常运行。

4.磁盘文件无法正常read和write

解决办法：将磁盘文件类型改为二进制。

分工

汪楚文： (1) Superblock, inodetable, Fcb的设计

(2) 实现文件的读取，写入，创建和删除。

(3) 文件系统的创建和初始化。

(4) 撰写实验报告。

刘 流： (1) 实现把文件系统中文件复制到系统外

(2) 实现把系统外的文件复制到系统里。

(3) help打印帮助信息

(4) 实现ls函数

小结

通过这次学习，我的团队更加清楚了文件系统的结构，以及进行对应操作时的具体实现方法。

目录结构

```
OS_EXP5/  
|  
|-- WenFS.c  
|-- 实验报告.pdf
```

[超链接： 点击此处可在github中查看该项目](#)