

计算机系统工程 LAB1

—— UNIX FILE SYSTEM LAYERING AND NAMING

发布时间：2015-3-26

DEADLINE：2015-4-09（卓越班为 4-16）

目的：

了解 UNIX 文件系统的基本架构，深入理解文件系统的 NAMING LAYER 和相关 API 的具体实现

要求：

1. 阅读理解教材第二章第五节的内容（Case study: unix® file system layering and naming）
2. 阅读理解项目 SimpleUnixFileSystemLayerSimulator 的框架代码。
该框架包含了 UNIX 文件系统的各层接口，并通过读写 diskblocks.data 文件模拟了 BLOCK 层的相关操作。diskblocks.data 是一个二进制流文件，我们通过对该文件不同位置的读写，模拟了 BLOCK 层对磁盘的读写。
在 SimpleUnixFileSystemLayerSimulator.cpp 中，我们定义了一系列的宏，这些宏决定了文件系统对 blocks 的区域划分（详见教材 P94 Figure2.20），其中包含了 block 的数量以及大小,bootblock, superblock, bitmap for free blocks, inode table 以及 file blocks 等各个区域的起始位置和大小，inode 的数据结构和数量等信息。
3. 根据教材第二章第五节，以 SimpleUnixFileSystemLayerSimulator 框架为基础，在 block 读写操作函数（WRITE_BLOCK_BY_BLOCK_NUMBER 和 BLOCK_NUMBER_TO_BLOCK）的基础上，实现 unix 文件系统各个 Layer 中的函数。
4. 添加额外功能函数，并修改 main 函数中的代码，使得最终编译出的 SimpleUnixFileSystemLayerSimulator.exe 能够支持如下命令行参数，简单起见，之后的 fs.exe 均指代 SimpleUnixFileSystemLayerSimulator.exe：

ls 指令，列出某个目录下的文件和子目录

>fs.exe ls <某个目录的绝对路径>

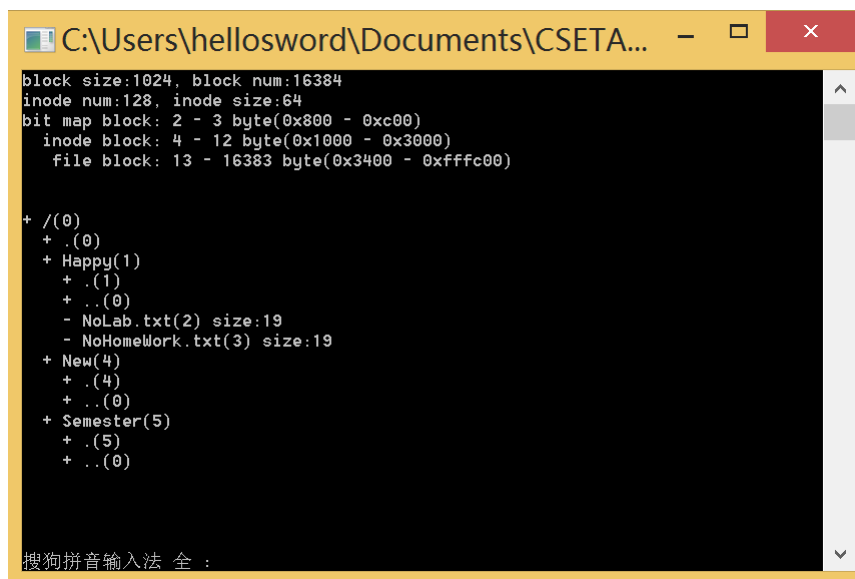
export 指令，从文件系统中提取某个文件及其内容，并以原文件名保存在 fs.exe 所在的同一目录下

>fs.exe export <某个文件的绝对路径>

tree 指令，生成某个目录的目录树，须包含该目录的全部子目录和文件，以及子目录下的目录和文件，以及子目录的子目录下的目录和文件，以及子目录的子目录的子目录下的目录和文件。。。依次类推，学过数据结构的同学会知道这个叫做递归。。。orz

>fs.exe tree <某个目录的绝对路径>

Sample: 下图第七行开始为“fs.exe tree /”指令的输出，也就是从根目录开始打印目录树。
+前缀表示目录，-前缀表示文件，中间是文件/目录名称，名称后的括号为对应的inode_number，如果是文件则最后还须包括该文件的大小。每进入一层子目录，均增加两个空格的缩进。



```
block size:1024, block num:16384
inode num:128, inode size:64
bit map block: 2 - 3 byte(0x800 - 0xc00)
inode block: 4 - 12 byte(0x1000 - 0x3000)
file block: 13 - 16383 byte(0x3400 - 0xffffc00)

+ /(0)
+ .(0)
+ Happy(1)
+ .(1)
+ ..(0)
- NoLab.txt(2) size:19
- NoHomeWork.txt(3) size:19
+ New(4)
+ .(4)
+ ..(0)
+ Semester(5)
+ .(5)
+ ..(0)
```

find 指令，在某个目录下查询所有名称为某关键字的目录和文件

>fs.exe find <某个目录的绝对路径> <要查询的关键字>

5. 如果命令行参数中的路径不存在，或出现其它错误，需给出合理提示。

6. simpledisk.data 中包含了一个简单的目录结构，同学们可以以此来调试自己的程序。
diskblocks.data 模拟文件中包含了比较多的文件和目录，且 inode number 和 block 并不连续，同学们可以以此来验证你们的代码是否正确。如果你觉得各 layer 函数的参数和返回值设置不够合理，可以自行修改，但尽量保证函数名不变，并写好注释

7. 关键步骤务必添加注释

说明：

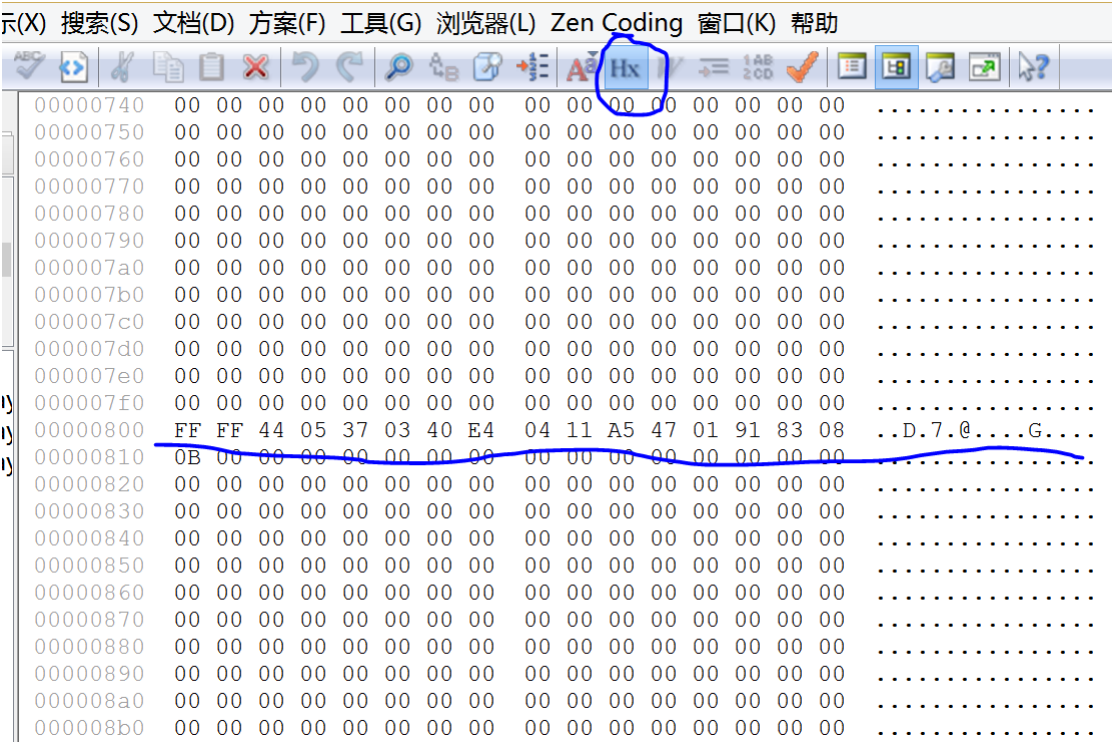
SimpleUnixFileSystemLayerSimulator 是以 C/C++语言编写一个 win32 控制台项目，请在 windows 下使用 Microsoft Visual Studio 2012（该软件可以从 mvls.fudan.edu.cn 中下载到）或 2010 版本进行编译运行，编译出的可执行文件 exe 通常位于 CSE_LAB1\SimpleUnixFileSystemLayerSimulator\Debug\目录下。对 C++不熟悉的同学可以完

全使用 C 语言的语法完成项目。

提示：

1. 命令中的<某个目录的绝对路径>中可能包括.和..两个特殊的目录，其中点（.）表示当前目录，点点(..)表示上一级目录。比如"/foo/bar"和"/./foo/bar/sth/.."两个路径是等价的。你们需要了解这些特殊目录在 UNIX 文件系统中的解析过程，并在自己的代码中进行相应的处理。卓越班的同学在完成第二阶段任务的时候务必注意。

2. 同学们可以使用较新版本的 EditPlus 以 16 进制的方式查看二进制流文件(diskblocks.data)中每个 byte 的具体数据。



3. 可以使用 windows 批处理来进行程序测试，范例详见：
CSE_LAB1\SimpleUnixFileSystemLayerSimulator\Debug\Sample.bat

卓越班同学的第二阶段任务：

卓越班的同学在完成了上述内容之后，使用你们的指令在 diskblocks.data 模拟文件中找到一个名叫 lab1_phase2_document.txt 文件导出之，并按该文档的说明完成你们第二阶段的任务。

额外工作

学有余力的同学可以尝试完成一些额外工作，比如 block layer 层的 cache，symbolic link/hard

link 等，并在文档中予以清晰的说明。会有最高 20% 的额外加分。

提交物

1. 项目说明文档，内容不做强制要求，可以包括：已完成的内容，额外工作（如果有的话），程序运行方法，项目过程中遇到的问题及解决方法，感想等。
2. 请在文档中写出对以下几个问题的回答：LINUX 文件系统所支持的最大文件/目录数量，文件名的最大长度，文件大小的最大值，一个目录所能容纳的子目录和文件的最大数量等这些“最大值”分别收到那些因素的影响？如果由你来设计一个类 UNIX 文件系统，在决定 Inode 结构，inodetable 大小以及 filebloks 数量的时候，你会考虑哪些因素。
3. SimpleUnixFileSystemLayerSimulator 项目的全部代码和文件，（建议上传前删除所有*.sdf 文件，否则你的 ftp 空间可能不够用）
4. 非卓越班同学如果觉得本次 lab 难度过高，可选择阅读经典论文 KJL84(a fast file system for UNIX)，并写出一篇 recitation，要求类似 recitation1。Recitation 报告末尾仍需附加对“提交物-2”中几个问题的回答。（较 lab 而言，做 recitation 的起评分会降低 30%）
5. 将所有上述内容完成后，统一打包（学号_姓名_lab1.zip）上传至 ftp

非卓越班同学上传至:

<ftp://10.132.141.33/classes/13/142> 计算机系统工程/WORK UPLOAD/LAB1/phase1

卓越班同学上传至:

ftp://10.132.141.33/classes/13/142 计算机系统工程/WORK_UPLOAD/LAB1/phase2

有任何问题请发邮件至: 14212010014@fudan.edu.cn