

四川大学计算机学院、软件学院

实验报告

学号：2023141460321 姓名：孙谦昊 专业：计算机科学与技术 班级：行政七班 第 14 周

| | | | |
|--------------------------|--|------|------------|
| 课程名称 | 操作系统实验 | 实验课时 | 8-9 节 |
| 实验项目 | 文件系统 | 实验时间 | 2025.05.27 |
| 实验目的 | 1.补全 loadDirectoryFromDisk() 函数 2.补全 writeData2File()函数 3.补全 saveDirectory2Disk()函数 | | |
| 实验环境 | VM WorkStation Pro、ubuntu-24.04.2-desktop-amd64 | | |
| 实 验 内 容 (算法、程序、步骤和方法) | <div>1.补充 loadDirectoryFromDisk()函数； 首先根据 InodeNum 找到对应的 Inode，然后再根据 Inode 对象得到 OpenFile 对象的内容，最后将 openfile 的内容写入 Directory 中。由此，将 loadDirectoryFromDisk() 函数补充为： SDirectory loadDirectoryFromDisk(int vDirInodeNum) { SOpenFile DirOpenFile; DirOpenFile. InodeNum = vDirInodeNum; SInode dirInode = loadInodeFromDisk(vDirInodeNum); DirOpenFile. Inode = dirInode; DirOpenFile. CurSeekPos = 0; assert(DirOpenFile. Inode. FileType == 'd'); SDirectory TempDir; readDataFromFile(&TempDir, sizeof(SDirectory), DirOpenFile); return TempDir; } 2.补充 writeData2File()函数 首先应讨论操作系统中的 numBlocks 是否足够，如果文件大小不足以装下内容则重新分配对应的块数。在向 file 中写入内容时，需讨论 curseekpos</div> | | |

是否在某个块的中间，以及最后内容未将最后一个块占满。由此， writeData2File() 函数补充为：

```
int writeData2File(const void *vSrc, int vNumBytes, SOpenFile& vioFile)
{
    if (vNumBytes <= 0) return 0;

    if (vioFile. CurSeekPos + vNumBytes > g_MaxFileSize)
    {
        vNumBytes = g_MaxFileSize - vioFile. CurSeekPos;
    }

    if (vioFile. CurSeekPos + vNumBytes > vioFile. Inode.
e.
    FileSize)
    {
        int outRange = vNumBytes - (vioFile. Inode. Files
iz
        e - vioFile. CurSeekPos);

        SBitMap DataBlockBitMap;
        createEmptyBitMap(DataBlockBitMap, g_NumBlock
s);
        memcpy(DataBlockBitMap.pMapData, g_Disk, g_Block
BitMapSize);

        vioFile. Inode. FileSize += outRange;
        int addBlock = (int) ceil(outRange / (double)g_B
l
        ockSize);
        if (countClearBits(DataBlockBitMap) < addBlock)
        {
            return 0;
        }
        for (int i = vioFile. Inode. NumBlocks; i < vioFi
l
        e. Inode.NumBlocks + addBlock; i++)
```

```

        {
            vioFile. Inode.BlockNums[i] = findAndSetAvai
                lableBit(DataBlockBitMap);
        }
        vioFile. Inode. NumBlocks += addBlock;
    }

    int FirstBlock = vioFile. CurSeekPos / g_BlockSize;
    int LastBlock = (vioFile. CurSeekPos + vNumBytes -
1)    / g_BlockSize;
    int NumBlocks = 1 + LastBlock - FirstBlock;

    char *pDiskDataPos = g_Disk + g_BlockBitMapSize + g_
I
nodeBitMapSize + sizeof(SInode) * g_NumInodes;
    memcpy (pDiskDataPos + g_BlockSize*vioFile. Inode.
Bl
ockNums [FirstBlock], vSrc, g_BlockSize - vioFile. C
u
rSeekPos % g_BlockSize)
    vSrc += g_BlockSize - vioFile. CurSeekPos % g_BlockS
i
ze;
    vNumBytes -= g_BlockSize - vioFile. CurSeekPos % g_B
l
ockSize;
    for (int i = FirstBlock + 1; i < LastBlock; ++i)
    {
        memcpy(pDiskDataPos + g_BlockSize*vioFile. Inod
e.
        BlockNums[i], vSrc, g_BlockSize);
        vSrc += g_BlockSize;
        vNumBytes -= g_BlockSize;
    }
    memcmp (pDiskDataPos + g_BlockSize*vioFile.Inode.Bl
o
ockNums[LastBlock], vSrc, vNumBytes) ;
    return vNumBytes;
}

```

3.补充 saveDirectory2Disk()函数;

| | |
|--------------|--|
| | <p>将 saveDirectory2Disk() 函数补充为:</p> <pre> void saveDirectory2Disk(int vDirInodeNum, const SDirectory& vDirectory) { SOpenFile DirOpenFile; DirOpenFile.InodeNum = vDirInodeNum; SInode dirInode = loadInodeFromDisk(vDirInodeNum); DirOpenFile.Inode = dirInode; DirOpenFile.CurSeekPos = 0; assert(DirOpenFile.Inode.FileType == 'd'); writeData2File(&vDirectory, sizeof(SDirectory), DirOpenFile); } </pre> |
| 结 论 (结 果) | <p>将 char *info 中输入“sunqianhao2023141460321”并运行后, 可以看到输出结果:</p>  <pre> Creating file test.txt(type:f) size 832 at / result: sunqianhao@2023141460321 Inode: FileType: f FileSize:832 NumBlocks: 13 NumLinks: 1 NumBlocks: 3 4 5 6 7 8 9 10 11 12 13 14 15 </pre> |
| 小 结 | <p>实验通过模拟文件打开、读写等功能, 进一步熟悉 Linux 文件系统管理的原理, 并且综合运用位视图、Inode 等工具实现了相应功能</p> |
| 指导老师 评 议 | <p>成绩评定: _____ 指导教师签名: _____</p> |