

Phase2_14307110274 文进

What I have done

实现了 namenode 与 datanode 包中对应类的功能，nameNode 类中除了继承来自 INameNode 类的所有方法外，还添加了其他方法。对于 mkdir create 而言，如果文件 uri 对应的目录不存在则抛出异常，如果目录下的文件存在也抛出异常，对于 open 而言，如果不存在对应文件就抛出异常。其他的功能包括一些磁盘读写方法，close 方法，块添加方法等。如图所示

```
@Override
public FileNode open(String fileUri) throws IOException, URISyntaxException, CL

@Override
public void close(FileNode fileNode) throws IOException, URISyntaxException {...}

@Override
public LocatedBlock addBlock(FileNode fileNode) throws IOException {...}
public Entity getEntityByName(DirNode currentDirNode, String name) {...}
public void writeDirNode(DirNode dirNode) throws IOException {...}
public DirNode readDirNode(int id) throws IOException {...}
public void writeFileNode(FileNode fileNode) throws IOException {...}
public FileNode readFileNode(int id) throws IOException, ClassNotFoundException
public int getMaxNodeID() throws IOException {...}
public int getMaxBlockID() throws IOException {...}
```

第二个重点做的工作就是两个流的实现，比较麻烦与复杂，比如 write

```
public void write(byte[] b) throws IOException {
    //todo your code here
    if (closed) {
        throw new IOException("OutputStream has closed");
    }
    int bLeftSpace = b.length;
    int bPos = 0;
    int bufferPos = BLOCKSIZE - bufferFreeSpace;
    while (bLeftSpace > 0) {
        if (bufferFreeSpace == 0) {
            flush();
            bufferPos = 0;
        }
        int writeSize = ((bufferFreeSpace > bLeftSpace)? bLeftSpace: bufferFreeSpace);
        System.arraycopy(b, bPos, buffer, bufferPos, writeSize);
        bPos += writeSize;
        bufferPos += writeSize;
        bLeftSpace -= writeSize;
        bufferFreeSpace -= writeSize;
    }
}
```

2.遇到的困难

InputStream 和 OutputStream 的实现上遇到了很多困难，因为要考虑的因素很多，比如 OutputStream 必须考虑什么时候应该 flush, flush 时该如何添加 Block, InputStream 必须考虑 read 的返回值，不同情况有不同的返回值，非常复杂。解决方案就是仔细弄清楚每个函数的功能，构建出清晰的思路，再来实现，这样的代码结构更清晰，会减少很多没有必要的 bug。