//File类list所有文件

public String[] list() {

SecurityManager security = System.getSecurityManager(); //文件系统权限获取

if (security != null) {

security.checkRead(path);

}

if (isInvalid()) {

return null;

}

return fs.list(this); //底层调用FileSystem的list

}

//FileSystem抽象类的list

//File类中定义

private static final FileSystem fs = DefaultFileSystem.getFileSystem();

//DefaultFileSystem类

class DefaultFileSystem {

/\*\*

\* Return the FileSystem object for Windows platform.

\*/

public static FileSystem getFileSystem() {

return new WinNTFileSystem();

}

}

//WinNtFileSystem类继承于FileSystem抽象类，这里我们主要观察它的list(File file)方法

@Override

public native String[] list(File f);

我们可以看到这是个native方法，说明list的操作是由操作系统的文件系统控制的，当目录中包含大量的文件时，这个方法的性能将会非常低。

由此，Files类设计了newDirectoryStream(Path dir)及其重载方法，将生成Iterable对象（可用foreach迭代）

//回调过滤

public String[] list(FilenameFilter filter) {

String names[] = list(); **//调用list所有**

if ((names == null) || (filter == null)) {

return names;

}

List<String> v = new ArrayList<>();

for (int i = 0 ; i < names.length ; i++) {

if (filter.accept(this, names[i])) { **//回调accept方法**

v.add(names[i]);

}

}

return v.toArray(new String[v.size()]);

}