|  |
| --- |
| Nullptr小组 |
| Minifs系统需求文档 |
|  |

|  |
| --- |
| Nullptr小组成员：刘硕，何祎君， 张嘉熙，张歆，彭青峰  2018-9-15 |

目录

**[一、 需求分析 2](#_Toc2139203747)**

**[二、 性能需求 3](#_Toc1899444972)**

**[三、功能设计 3](#_Toc936307571)**

[（一）操作空间： 3](#_Toc717474081)

[（二）操作目录及文件： 4](#_Toc1714785846)

## 需求分析

从这个系统的功能要求中，我们得出，这个系统需要划分出1G外存来充当我们自己的空间，所有的文件与文件相应的信息都是存储在这1G的空间里面的。也就是说，我们不能用Windows系统的文件管理系统去管理文件，而是得自行建立一个这样的MiniFS系统，由MIniFS系统在这1G中进行数据的读取与写入。所以我们需要自行设计数据结构，设计算法，写底层交互，去实现这个系统。这其实就是与FAT32，NTFS等一样的软件层面的文件系统，虽然我们所管理的磁盘空间只有1G的大小。

这个系统的最基本的功能就是与外部的操作系统的文件进行数据交互，也就是文件的导入导出；除此之外，还要实现一些在系统内部的文件管理功能，比如复制，删除，查看信息，查看文本等的功能。 但是在分析了整个系统的功能后，我们觉得这些功能不太够，这样的文件系统在使用时是无法满足用户的需求的，这样的单层目录结构会使所有的文件堆积在一起，根本无法做到方便管理。与真正的操作系统相比，我们可以明显发现，它少了文件夹这样的设计。文件夹可以方便用户对文件进行分类管理，也方便用户对文件进行批量操作。所以我们决定对整个系统的设计添加文件夹功能，将目录设计为多级的。这样，我们又多了许多新功能，比如创建、打开、移动、复制、删除文件夹，显示文件目录等。除此之外，我们还决定仿照Windows命令行的tree功能，做到以树形结构显示系统的文件夹结构。相信在拥有这些功能后，我们的系统就会真正地具备实用性。

## 性能需求

系统的性能水平主要表现在对文件和文件的可操作数量、大小和操作响应时间两方面。

对文件大小的操作限制即为文件系统的空间利用率，本系统应实现95%以上的空间利用率，即在1G磁盘空间的前提下，能够实现对0.95G或以上的文件的有效存储。

对文件和目录的操作数量则表现为目录的深度和最大子目录数以及目录下最大文件数量。为保证用户操作体验，目录深度应最少实现十层子目录的支持；每个目录下的最大目录数量和文件数量不少于十万，基本实现用户操作无受限感。

操作响应时间是用户使用的最直观体验。系统应实现在MiniFS系统内部进行操作时，0.5s内实现响应；Windows系统和MiniFS系统进行文件传输时，最大速度不低于50M/s，在20s内完成对0.9G文件的移动。

## 三、功能设计

在功能上，我们将需求大体分为两类，一类是对于MiniFS空间的操作，另一类则是对于空间中的文件或目录进行的操作：

### （一）操作空间：

create：建立一个新的MiniFS空间；

mount：安装所要操作的空间；

format：格式化一个MiniFS空间；

close：退出MiniFS空间；

help：显示帮助信息；

### （二）操作目录及文件：

attrib：显示文件或者目录属性；

cd：打开目标目录；

copy：复制文件或者子目录；

delete：删除空间中的文件或者子目录；

dir：显示空间的文件目录；

map：显示指定文件在空间所占用的全部块的块号；

md：创建子目录；

move：将指定路径下的文件移动到新的路径；

opt：优化整个空间，将每个文件尽可能连续存放；

import：导入硬盘上的文件到本系统中；

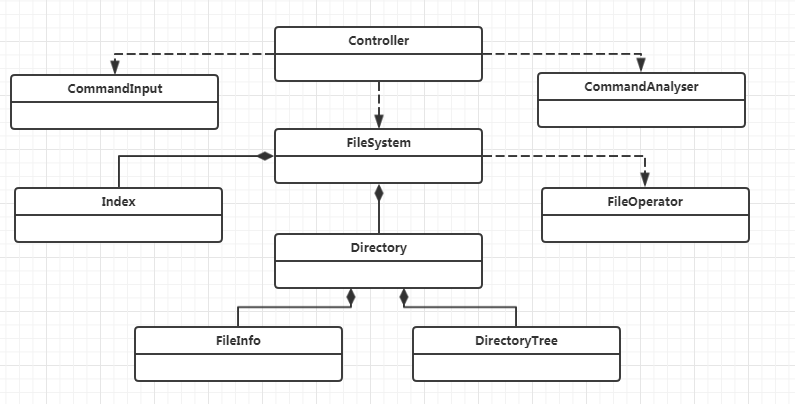
export：将本系统中的文件导出到外部硬盘中；

tree：以树型方式显示文件目录和文件。

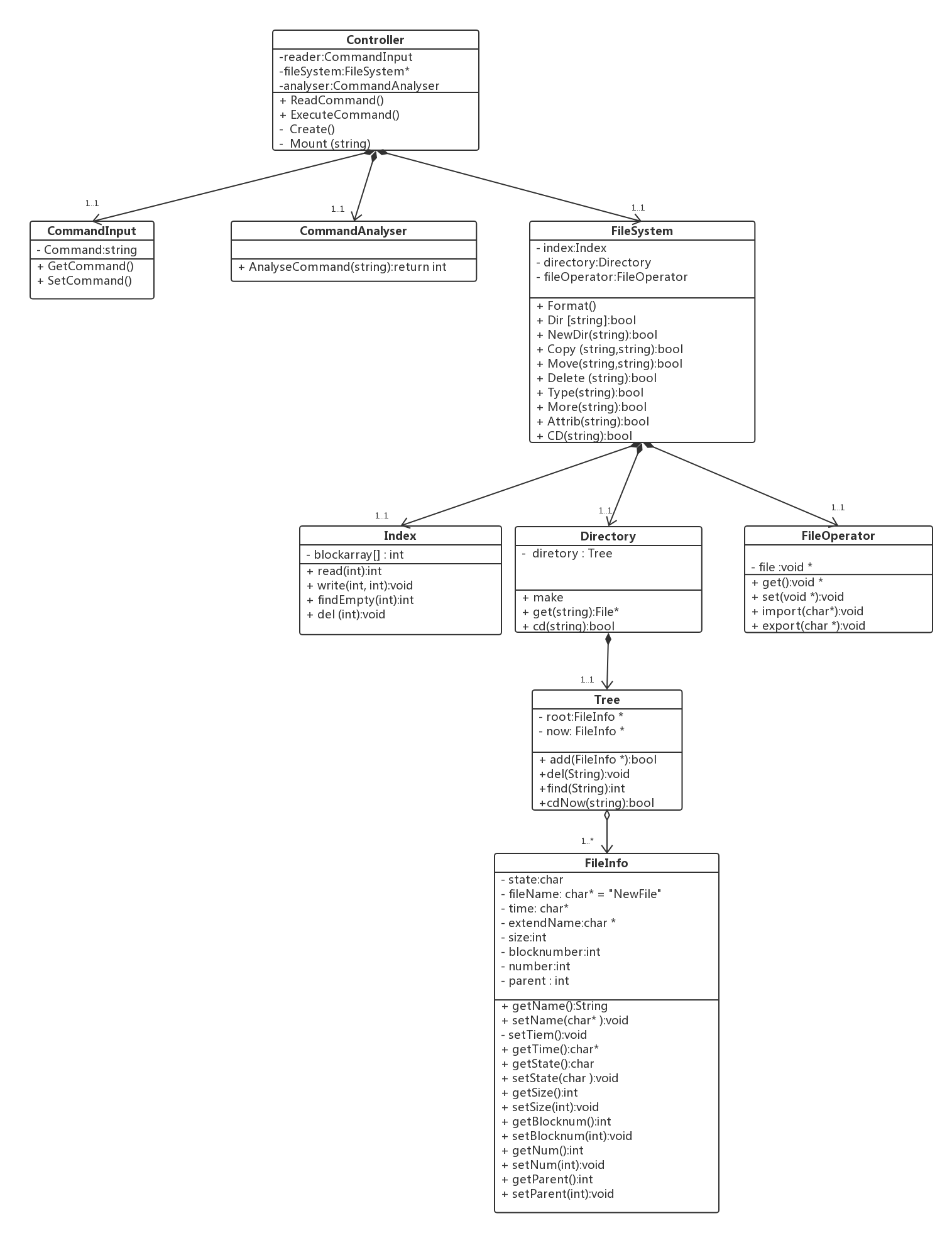
## UML建模

### 类图

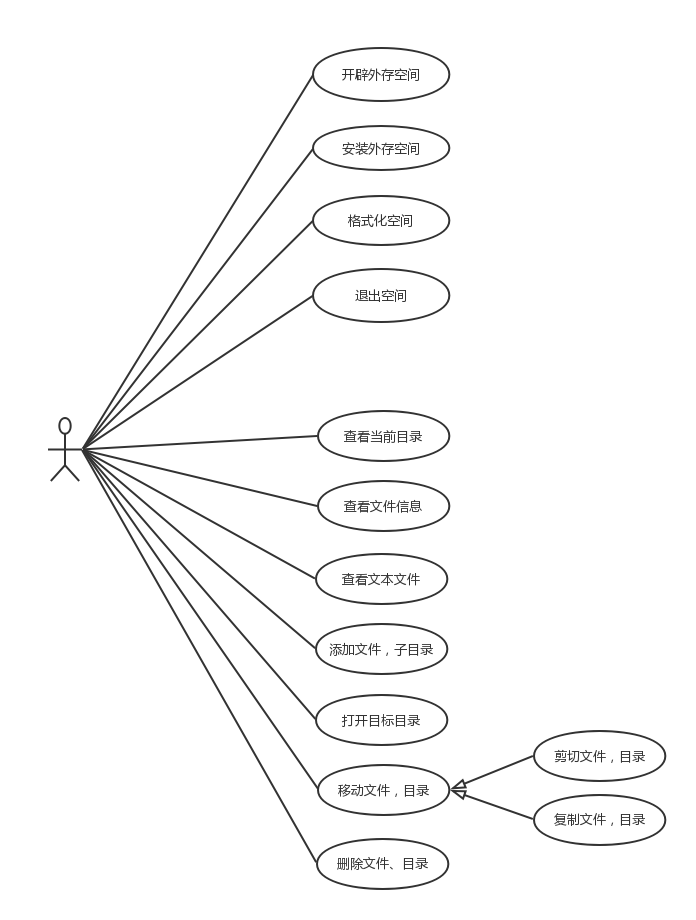
简化版类图框架



类图



### 用例图



### 时序图

