程设大作业：文件系统使用手册  
分工：all by李晨昊

设计思路：

所有文件相关操作由FileManager类管理。其构造函数接受一个参数表示文件系统所在位置，如果文件不存在则创建文件并初始化系统。其析构函数会向文件系统中完整写入内存中的相关内容（每次写操作后都会写入变化，所以这其实是不必要的）。

FileManager仅有一个public成员函数exec，它接受一个std::vector<std::string>作为参数。第0个字符串表示指令名称，此后的字符串表示具体参数。exec返回一个std::string来表征操作的状态（成功/失败）并在主函数中输出。

Util文件中定义了一些与FileManager本身状态无关的功能函数

Zip文件中定义了Zipper类和DeZipper类，两个类都重载了operator(std::string)，用于实现文件的压缩和压缩文件的展开。假定zip和dezip分别为两个类的对象，则始终保证dezip(zip(str))==str。压缩算法采用霍夫曼树。

使用方法

统一说明：

1. 各种错误都有对应的输出，这里不一一列出，下面也不会说明。
2. 部分简单操作不给出示例。
3. 在不导致误会的前提下，用“文件“来代指“文件和压缩包”。
4. 基本操作

>>exit

退出文件系统，保存一切修改。

>>cd arg1

将当前文件夹改至arg1对应的文件夹。

1. 读操作

>>pwd (-r)

显示当前文件夹下的文件夹和文件

接受一个可选的参数-r，此时将递归地显示当前文件夹下的所有文件夹和文件

>>ls arg1 (-r)

查看path对应的“东西”。也就是说并不要求arg1是文件。

第一行输出path的类型（文件/文件夹/压缩包）。

第二行输出path的大小。

大小的定义为：

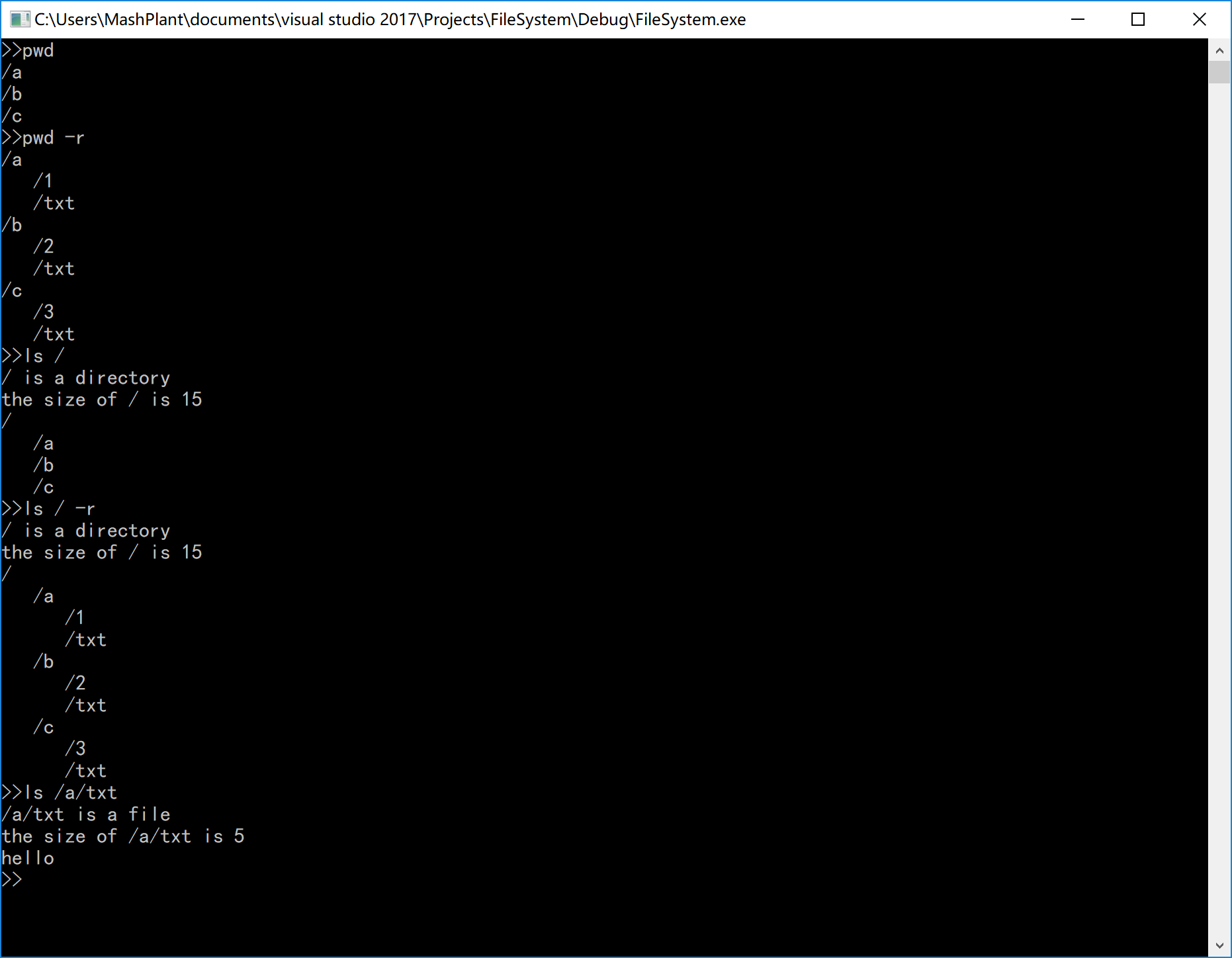
size(path) = path.data.size if isinstance(path,(File,Zip)) else sum([size(i) for i in path.child])

此后若干行，若arg1为文件夹，输出arg1下的文件和文件夹；若arg1为文件，输出文件内容；若arg1为压缩包，则没有此项输出。

接受一个可选的参数-r，此时对arg1是文件夹的情况下，此时将递归地显示arg1文件夹下的所有文件夹和文件。

若要求显示时间，则会显示该文件或文件夹最后一次修改的时间。

>>cat arg1

输出arg1对应的文件的内容。  


1. 写操作

>>mkdir arg1

创建arg1目录。

>>echo arg1 arg2

将arg1写入arg2文件。如果文件已存在则覆盖原文件，否则创建文件

arg1不需要由引号括起来，引号会被当成正常内容；如果arg1为””（仅有两个引号），会创建空文件。

如果arg1的长度超过了4095，则会在多个block中存放该字符串，这些block一定是连续分布的。所以可能会产生block总数足够但是连续的block不够从而无法创建长文件的情况。

>>append agr1 arg2

将arg1的内容添加到arg2文件之后

>>rm arg1

删除arg1文件

>>rmdir arg1

递归地删除arg1目录下的所有文件和文件夹

注意：若删除当前目录，则当前目录自动跳转到根目录。

>>cpy arg1 arg2

将文件arg2的内容复制到文件arg1

注意：arg1和arg2不能指向同一个文件，否则将做出提示。

>>cpydir arg1 arg2

创建文件夹arg1，递归地将arg2下的的文件和文件夹复制到path1下

注意：arg1不能是arg2的子目录，否则将做出提示。

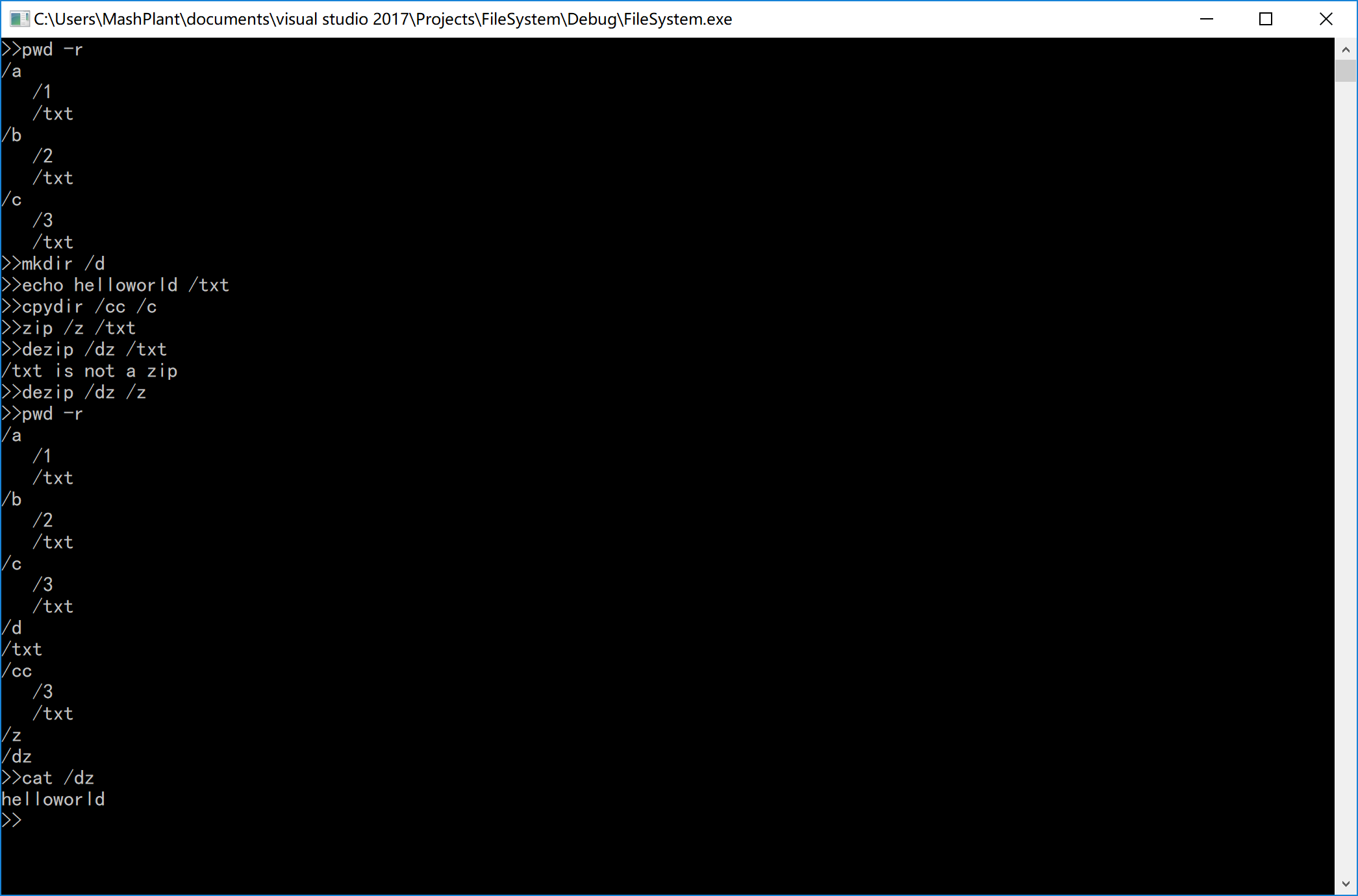
>>zip arg1 arg2

将arg2的内容压缩后写入arg1（类型为zip）。如果arg1已存在则覆盖原文件，否则创建文件。

通常而言对于较短的内容，压缩包会比原文件更大。对于较长的内容则可起到一定压缩效果。

>>dezip agr1 arg2

将arg2的内容解压缩后写入arg1（类型为zip）。如果arg1已存在则覆盖原文件，否则创建文件。

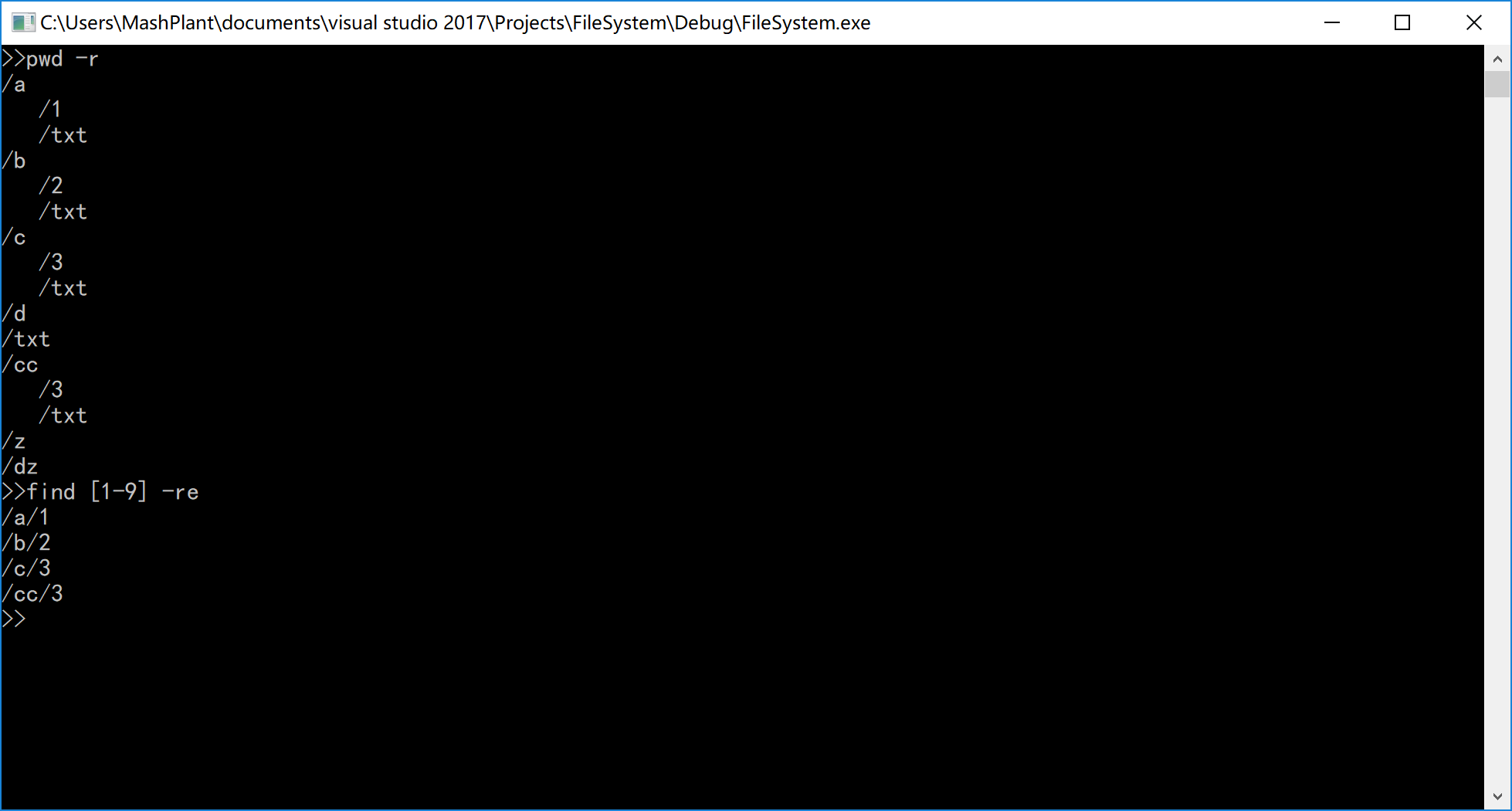


1. 查找操作

>>find arg1 (-re)

从根目录开始查找所有名字包含arg1的文件夹和文件并显示。

接受一个可选的参数-re，此时将查找所有名字中 含有能被正则表达式str匹配的子串 的文件夹和文件。



五 .其他

>>time

改变ls操作中是否显示时间。

time状态不会保存到文件系统中。