**操作系统 Project1**

*13331231*

*孙圣*

*教务四班（补修）*

1. **实验环境与使用**

操作系统：Windows8

使用说明：打开cmd，进入指定目录。

1. 批处理测试：执行execute.bat；
2. 可以直接在命令行参数中指定想要执行的文件，

例如minShell.exe test.exe；

1. 在命令行参数中载入相关的script文件，

例如minShell.exe script.txt。

1. **实验过程**

首先，判断命令行参数的个数，如果不是2个，则直接报错。

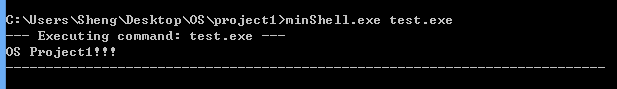
之后判断参数的后缀，如果是.exe，就直接调用execute();函数执行。如果

不是.exe，就尝试当作文本文件打开文件，文件不能打开则报错。否则，对于文本文件中的每一行执行execute();函数。

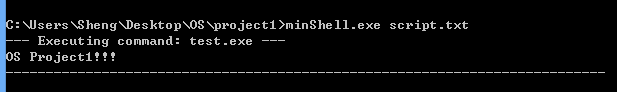
对于execute();函数：我一开始是考虑根据传入的参数分为两类。第一类是在路径中不含有’\’的情况，即可执行文件是和主程序在同一个文件夹下。如果满足这样的情况，则直接调用CreateProcess(); API。当调用失败时，考虑在系统中的环境变量中进行搜索。因此，先通过GetEnvironmentVariable(); API来得到环境变量，在将程序名加到每一个环境变量所指的目录后，再次调用CreateProcess(); API。

后来在测试的过程中发现，对于mspaint.exe这样的系统自带的程序来说，并没有执行到我写的在环境变量中搜索的代码。进一步的测试也证实：CreateProcess(); API本身会在系统的环境变量目录中搜索指定的可执行文件。因此，只需要将execute();方法改成简单的对CreateProcess(); API的调用即可。

1. **实验结果**
2. 命令行参数指定可执行文件 minShell.exe test.exe



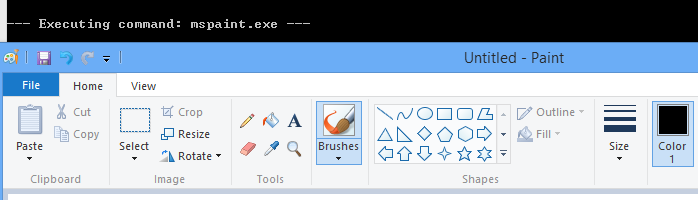
1. 命令行参数指定文本文件minShell.exe script.txt
2. 根目录下存在的可执行文件test.exe



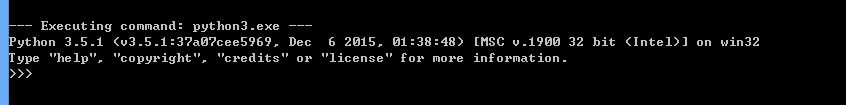
1. 根目录下不存在的文件 testNo.exe



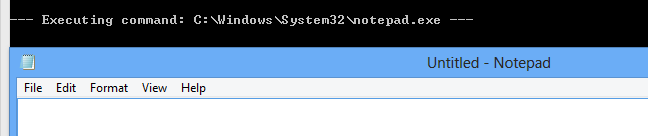
1. 系统自带的文件mspaint.exe



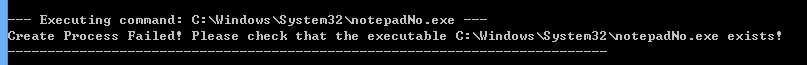
1. 用户安装的文件并配置了环境变量 python3.exe



1. 指定路径下存在的文件C:\Windows\System32\notepad.exe



1. 指定路径下不存在的文件C:\Windows\System32\notepadNo.exe



3. 命令行参数指定不存在的文本文件minShell.exe scriptNo.txt

