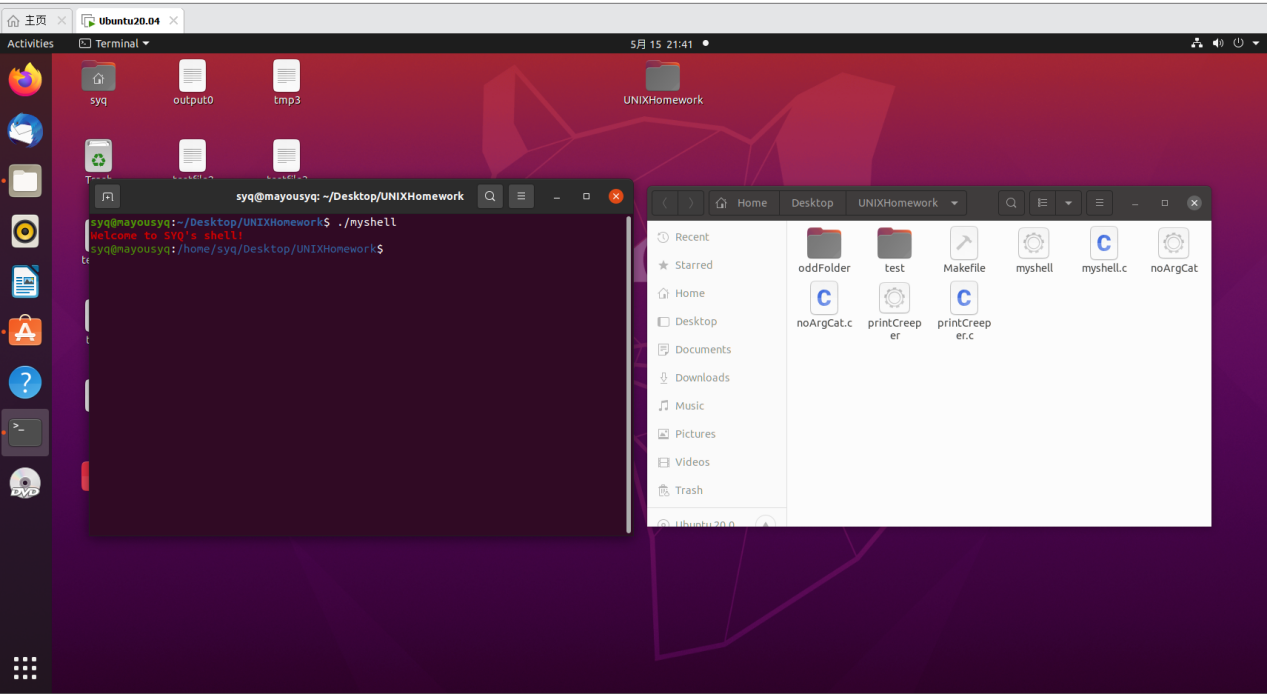
UNIX作业报告

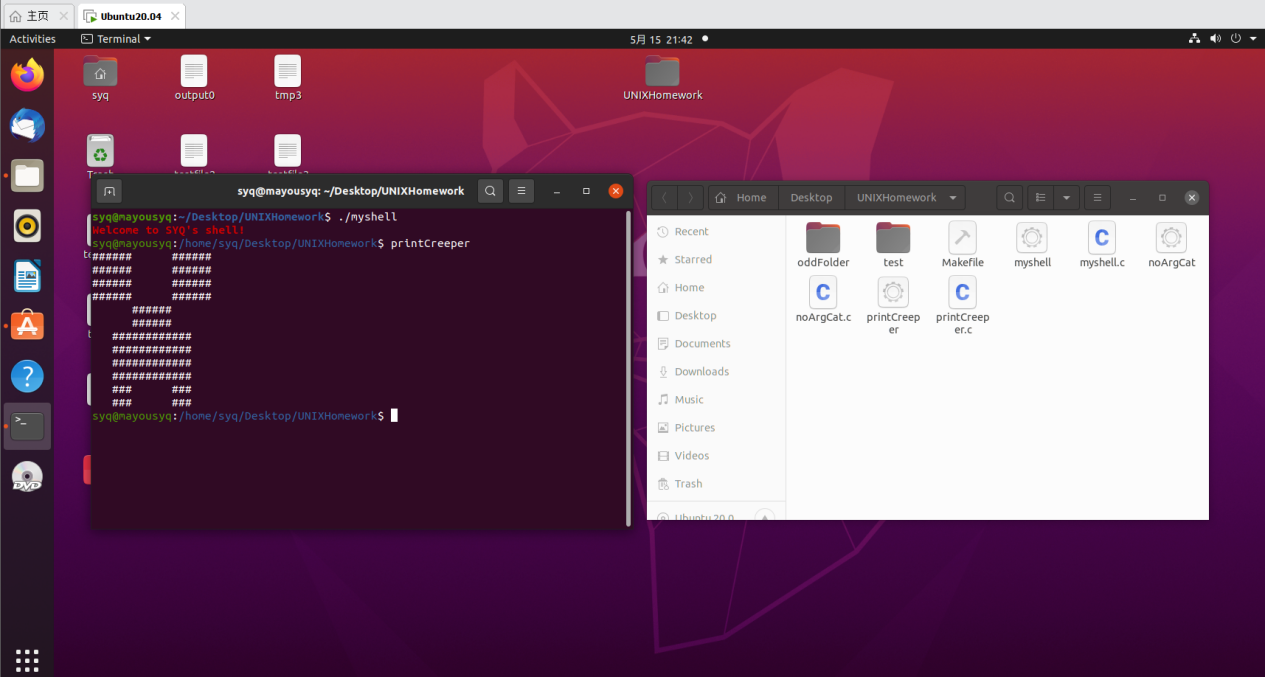
### 一、myshell功能概述

可以运行不带参数的外部命令，支持I/O重定向，支持管道。

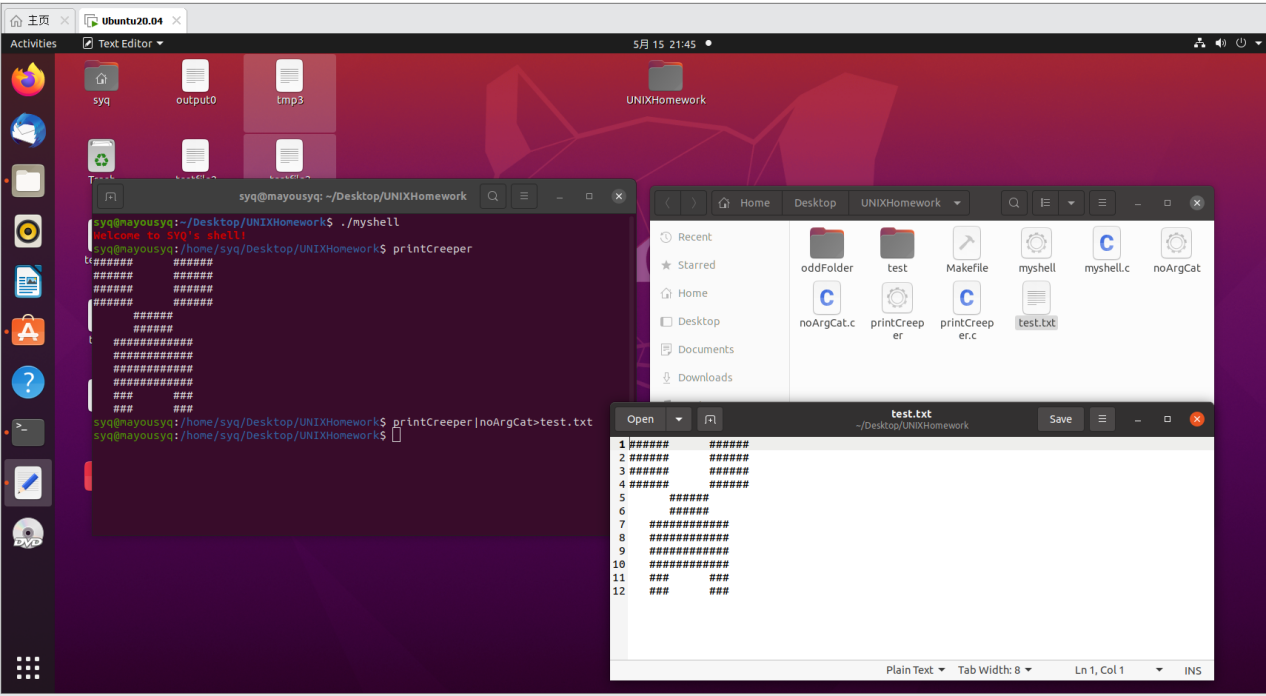
示例图如下



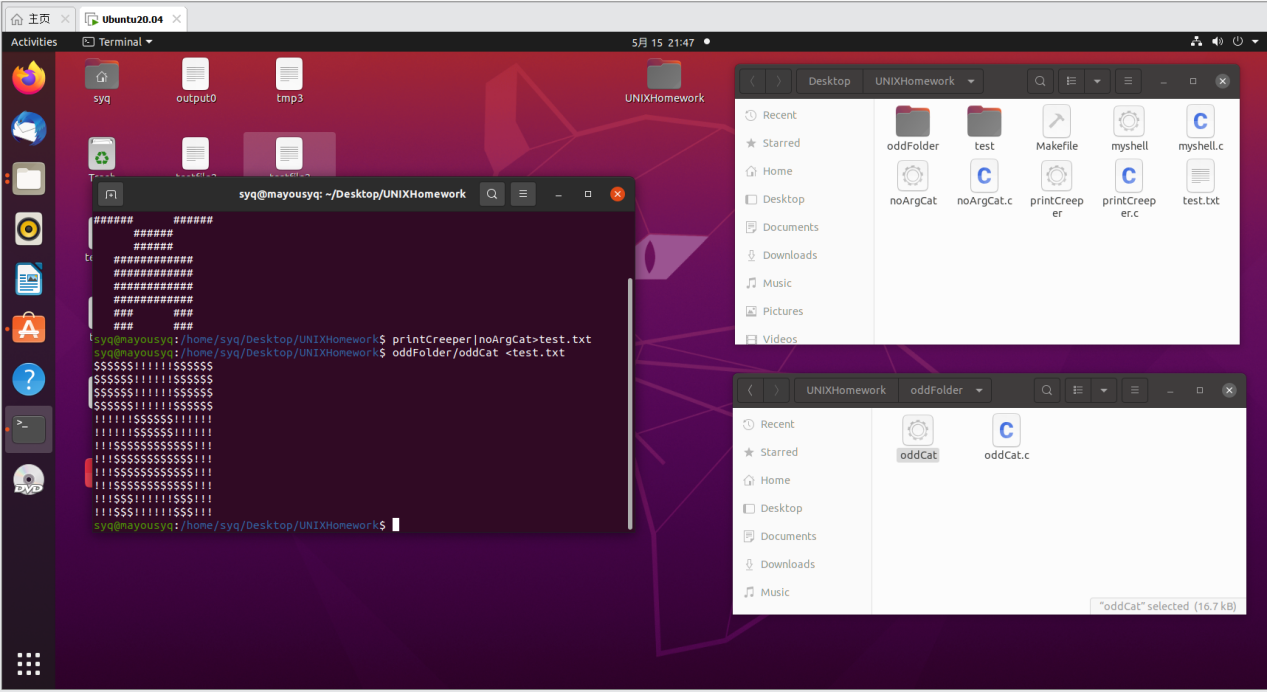
输入./myshell，myshell程序启动，欢迎语句出现，提示符出现



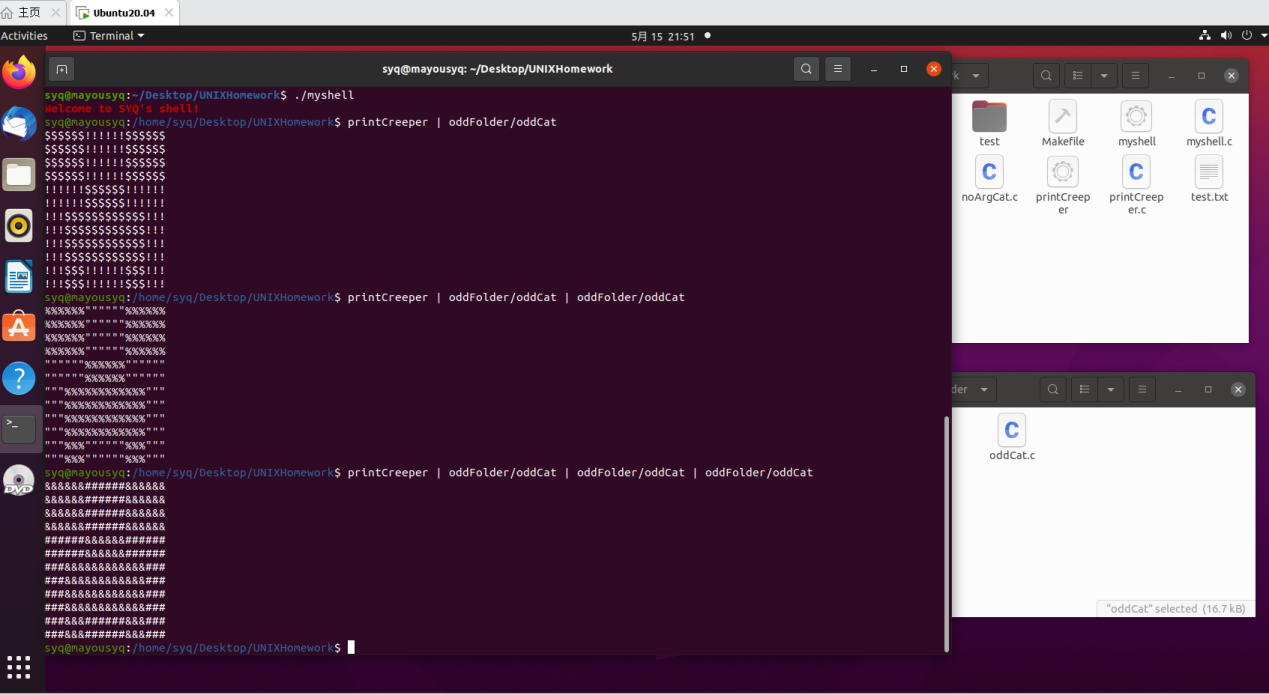
使用myshell运行当前目录下的printCreeper程序



将printCreeper的输出用管道连接到noArgCat的输入，再将标准输出重定向到test.txt



将标准输入重定向到test.txt，运行oddFolder下的oddCat文件



还可以使用多重管道

功能细节：

1.当前状态下，每行最多包含20条指令，每条指令最多包含50个字符，这些参数可以在myshell.c文件头部调节

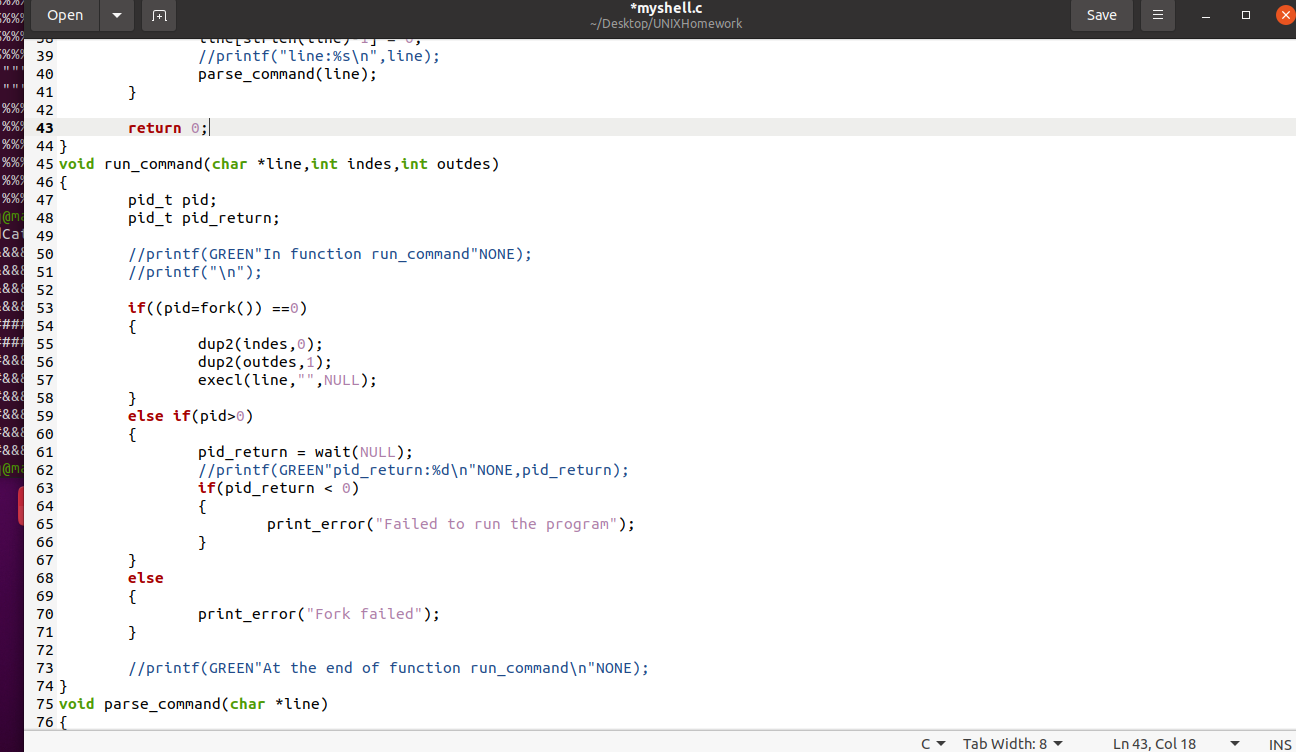
2.可以检查是否多次将文件重定向，但除此之外，没有过多的输入语法检查功能

3.有基本的异常抛出功能，例如：当遇到不存在的重定向输入文件名时，会打印Error并退出程序

### 二、实现原理

运行单条指令，使用了"fork()创建子进程+exec()运行外部程序+父进程调用wait()"的方式

同时，我还使用dup2()进行了输入输出重定向，以满足调用者的要求



大致流程：

if(pid=fork() == 0)

{

//子进程

输入输出重定向

运行外部程序

}

else if(pid>0)

{

//父进程

等待子进程

若运行失败，抛出异常

}

else

{

//创建子进程失败

抛出异常

}

解析重定向和管道时，我为一行中的每条外部指令维护了两个文件描述符，表示它们的输入和输出文件。其中，第一条指令的输入默认为标准输入，最后一条指令的输出默认为标准输出。

大致流程：

while循环检查输入字符串

{

检测到文件名([0-9a-zA-z\_./])

{

读入此指令

若指令数超出范围，抛出异常

若指令长度超出范围，抛出异常

存入指令列表

指令记录数++

}

检测到"<"

{

若输入已被重定向，抛出异常

排除空白符，读入文件名

若文件名长度超出范围，抛出异常

使用open()打开文件，代替标准输入

若文件打开失败，抛出异常

}

检测到">"

{

若输出已被重定向，抛出异常

若后面紧接一个">"，输出模式为添加，否则为覆盖

排除空白符，读入文件名

若文件名长度超出范围，抛出异常

使用open()打开文件，代替标准输出

若文件打开失败，抛出异常

}

检测到"|"

{

使用pipe()创建管道，将上一个文件的输出描述符和下一个文件的输入描述符分别 改为fd[1]和fd[0]

}

检测到空白符

{

直接跳过

}

检测到其它字符

{

抛出异常

}

}

以之前定义好的输入输出文件描述符，运行每一条指令

关闭除标准输入和输出之外所有的外部文件及管道