OS实验2第二部分实验报告

实验过程

首先浏览整个 `shell.c` 的框架可以看到主要部分在于实现 **普通命令** 和 **管道命令**，在此基础上，又需要实现 **外部命令** 和 **内建命令**。因此我在助教给出框架的基础上实现了以下几个额外的部分：

- 每次等待命令输入前打印当前目录(类似Bash)
- 实现内建命令： `exit` 、 `about` 、 `cd`

其中 `exit` 由 `exit(0)` 直接实现， `cd` 由 `chdir` 实现， `about` 打印出该shell的相关信息。

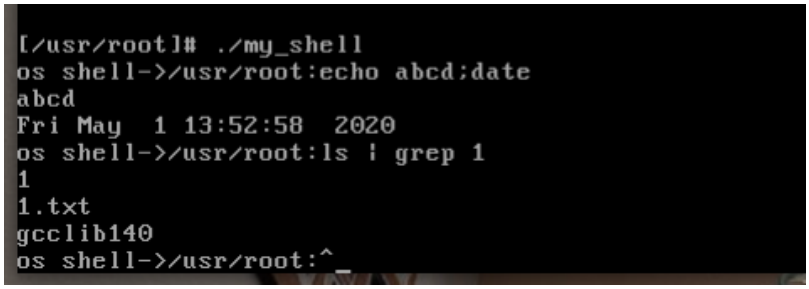
从以上的思路可以看到，只需要在助教框架里完善管道，以及 `fork()` 子进程来执行 `execl()` 。以下详细叙述如何实现管道：

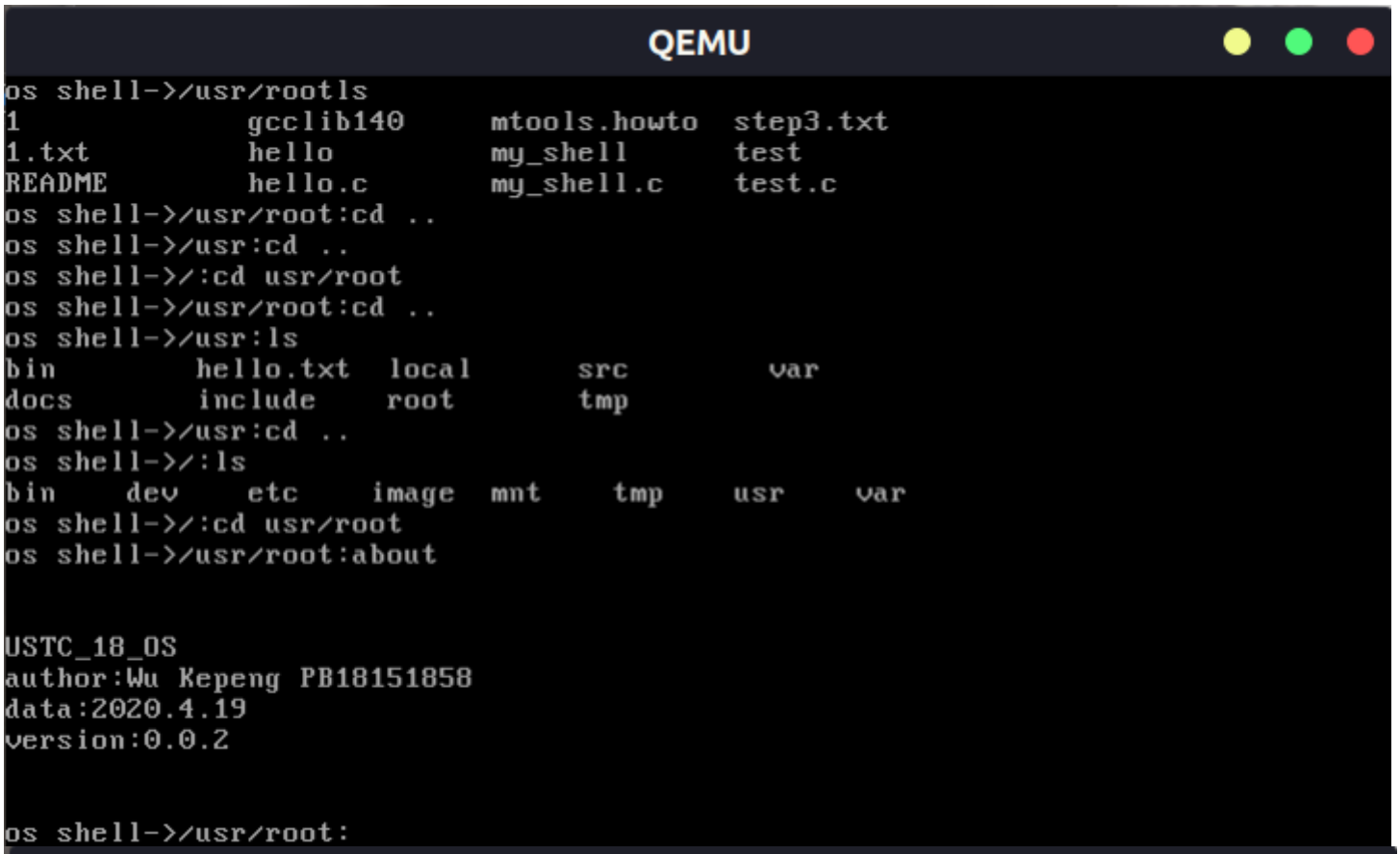
- 先使用 `pipe()` 生成管道，并通过 **pid** 保存文件描述符。
- 使用 `dup2()` 来进行重定向，将 `STDOUT` 或 `STDIN` 重定向到管道的 **写/读** 端。
- 父进程将 *cmd1* 的输出读入缓存(通过管道的读端口)，并写入与 *cmd2* 建立起的管道。
- 此即完成了管道符 `|` 的功能，把前一个命令的标准输出作为后一个命令的标准输入。

那么如何实现命令的执行呢：

- `fork()` 子进程。
- 父进程 `wait()` 子进程结束。
- 子进程调用 `execl(SHELL, "sh", "-c", cmdline, NULL)`

实验结果




QEMU
os shell->/usr/rootls
1 gcclib140 mtools.howto step3.txt
1.txt hello my_shell test
README hello.c my_shell.c test.c
os shell->/usr/root:cd ..
os shell->/usr:cd ..
os shell->/:cd usr/root
os shell->/usr/root:cd ..
os shell->/usr:ls
bin hello.txt local src var
docs include root tmp
os shell->/usr:cd ..
os shell->/:ls
bin dev etc image mnt tmp usr var
os shell->/:cd usr/root
os shell->/usr/root:about
USTC_18_OS
author:Wu Kepeng PB18151858
data:2020.4.19
version:0.0.2
os shell->/usr/root:


QEMU
os shell->/:cd usr/root
os shell->/usr/root:cd ..
os shell->/usr:ls
bin hello.txt local src var
docs include root tmp
os shell->/usr:cd ..
os shell->/:ls
bin dev etc image mnt tmp usr var
os shell->/:cd usr/root
os shell->/usr/root:about
USTC_18_OS
author:Wu Kepeng PB18151858
data:2020.4.19
version:0.0.2
os shell->/usr/root:exit
*** Have a nice day! ***
[/usr/root]#