山东大学<u>计算机科学与技术</u>学院 操作系统 课程实验报告

学号: 202100130022 姓名: 郭家宁 班级: 21 数据

实验题目: 进程综合实验(Shell)

实验学时: 2 实验日期: 2023-4-24

实验目的: 掌握操作系统 shell 的工作机制与实现过程, 练习 Linux 系

统中进程创建与控制有关的编程和调试技术。

实验结果:

实验原理:

对输入的字符串按照空格进行分割,如果有"<"'>''|'进行特殊处理。使用 dup 系统调用,实现 I/0 的重定向,然后使用递归分析管道进行管道的处理

实现的功能:

(1) 执行用户输入的合法命令,允许命令携带参数,如 Is -la:

```
jianing@jianing-virtual-machine:~/桌面/os_ex/exe3_plus$ ./shell
SIGINT: Success
COMMAND-> ps -l
F S UID PID PPID C PRI NI ADDR SZ WCHAN TTY TIME CMD
0 S 1000 2790 2732 0 80 0 - 5024 do_wai pts/0 00:00:00 bash
0 S 1000 3249 2790 0 80 0 - 4425 do_wai pts/0 00:00:00 shell
0 R 1000 3250 3249 0 80 0 - 5335 - pts/0 00:00:00 ps
COMMAND->
```

(2) 支持命令的后台运行:

如果输入的指令最后有'&'字符,然后让父进程无需等待子进程结束

```
if (parseOrder(order, n, '&', 0) == n - 1)
{
    background = 1;
    ifvis[n - 1] = 1;
}
```

```
if (background == 0)

int status;
wait(&status); // 父进程等待子进程结束
}
```

(3) 支持文件的重定向功能。

```
jianing@jianing-virtual-machine:~/桌面/os_ex/exe3_plus$ ./shell
SIGINT: Success
COMMAND-> echo osYYDS >file.txt
redirect output to file.txt
COMMAND-> cat file.txt
osYYDS
COMMAND->
```

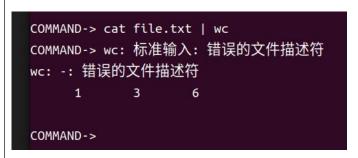
```
COMMAND-> echo pwd >file.txt
redirect output to file.txt
COMMAND-> ./ <file.txt >test_out.txt
redirect output to test_out.txt
redirect input to file.txt
COMMAND-> cat test_out.txt
COMMAND-> pwd
/home/jianing/桌面/os_ex/exe3_plus
COMMAND-> COMMAND->
```

(4) 保存用户最近输入的 30 个命令,可利用上下方向键进行选择使用 readline 库函数的 addhistory 函数实现,重载上下键。

```
// 调用readline接口实现命令获取、历史保存
char *order = readline("COMMAND-> ");
add_history(order);
```

```
jianing@jianing-virtual-machine:~/果田/os_ex/exe3_plus$ ./shell
SIGINT: Success
COMMAND-> ps -l
F S UID PID PPID C PRI NI ADDR SZ WCHAN TTY TIME CMD
0 S 1000 2790 2732 0 80 0 - 5024 do_wai pts/0 00:00:00 bash
0 S 1000 3249 2790 0 80 0 - 4425 do_wai pts/0 00:00:00 shell
0 R 1000 3250 3249 0 80 0 - 5335 - pts/0 00:00:00 ps
COMMAND-> cat file.txt
pwd
COMMAND-> cat file.txt
```

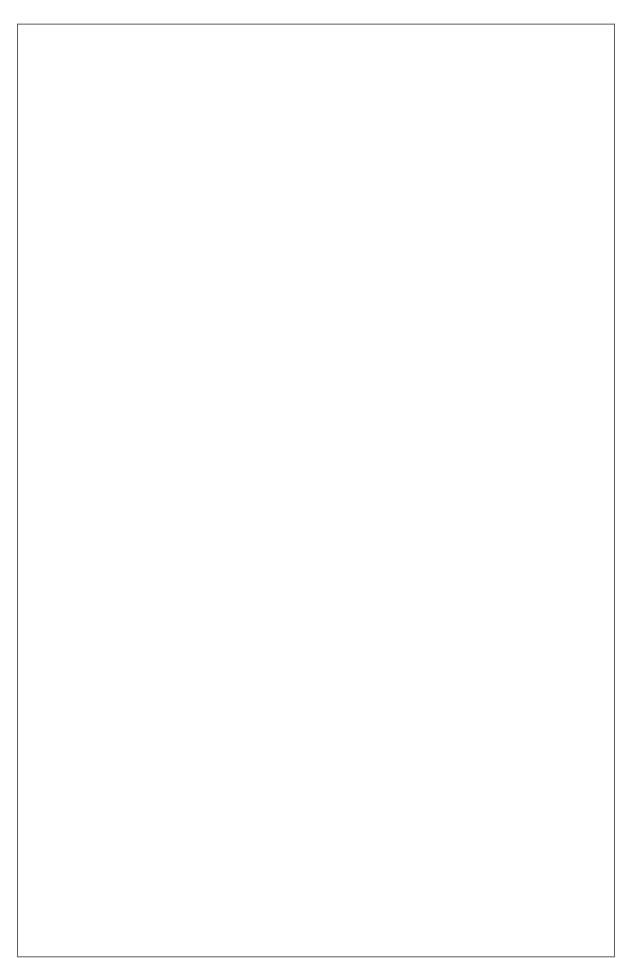
(5) 输入非法命令给出错误提示



对比一个真实的 Linux Shell, 你所实现的 Shell 需要做哪些改进?实现这些改进的思路如何?

对比一个真实的 Linux Shell, 我实现的这个 shell 无法使用 cd 命令移动文件夹, 而且在多级管道的参数传递中有一些问题。

改进思路,可以在移动文件夹的时候重新创建一个新的 shell 进程然后保留上一次的历史记录。 管道参数传递可以使用递归实现,递归调用子进程,然后再最后一个子进程中调用父进程来实现 参数的传递。



问题及收获:
更加掌握和了解了操作系统 shell 的工作机制与实现过程