

Linux 开发环境及应用 上机作业四：Shell 管道和重定向功能的实现 实验报告

毛子恒

2019211397

北京邮电大学 计算机学院

日期：2022 年 5 月 26 日

1 实验内容

使用 `fork()`, `exec()`, `dup2()`, `pipe()`, `open()`, `wait()` 等系统调用编写 C 语言程序完成与下列 Shell 命令等价的功能。

```
grep -v usr < /etc/passwd | wc -l > r.txt; cat r.txt
```

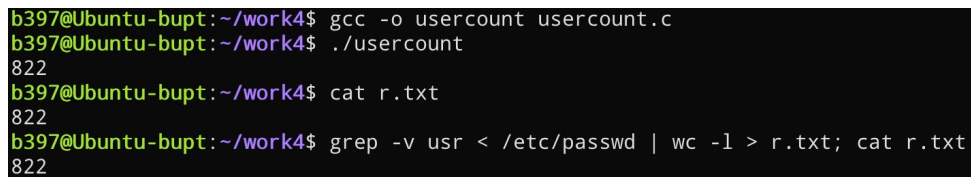
2 实验步骤

编写代码 使用 `pipe()` 初始化管道, `fork()` 创建子进程, `open()` 打开文件, `dup2()` 重定向文件描述符, `execlp()` 执行程序。

```
1  #include <fcntl.h>
2  #include <stdio.h>
3  #include <unistd.h>
4  #include <sys/types.h>
5  #include <sys/stat.h>
6  #include <sys/wait.h>
7
8  int main()
9  {
10     int fd[2];
11     pipe(fd);
12     if (!fork())
13     {
14         close(fd[0]);
15         int passwd_fd = open("/etc/passwd", O_RDONLY);
16         dup2(passwd_fd, 0);
17         dup2(fd[1], 1);
18         close(fd[1]);
19         execlp("grep", "grep", "-v", "usr", NULL);
20     }
21     if (!fork())
22     {
```

```
23     close(fd[1]);
24     int r_fd = open("r.txt", O_WRONLY | O_CREAT, 0777);
25     dup2(fd[0], 0);
26     close(fd[0]);
27     dup2(r_fd, 1);
28     execlp("wc", "wc", "-l", NULL);
29 }
30 wait(NULL);
31 close(fd[0]);
32 close(fd[1]);
33 execlp("cat", "cat", "r.txt", NULL);
34 }
```

运行结果 运行结果如图 1。



```
b397@Ubuntu-bupt:~/work4$ gcc -o usercount usercount.c
b397@Ubuntu-bupt:~/work4$ ./usercount
822
b397@Ubuntu-bupt:~/work4$ cat r.txt
822
b397@Ubuntu-bupt:~/work4$ grep -v usr < /etc/passwd | wc -l > r.txt; cat r.txt
822
```

图 1: 运行结果

3 实验总结

本次实验中我主要应用了 `fork()`, `exec()`, `dup2()`, `pipe()`, `open()`, `wait()` 系统调用模拟了 `shell` 的一小部分功能, 使我对进程管理和文件描述符更加熟悉。