## 编程作业4--Shell管道和重定向功能的实现

### 一．实验内容

使用fork(), exec(), dup2(), pipe() ，open()系统调用完成与下列shell命令等价的功能。（提示：为简化编程，不需要用strtok断词，直接实现能达到下述shell命令相同功能的程序即可）

命令如下：

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

### 二、实验目的

1.熟悉Linux系统中的各类命令，以及能够通过编程使用系统调用实现相应的功能。

2.加强对Shell管道和重定向功能的理解。

### 实验步骤

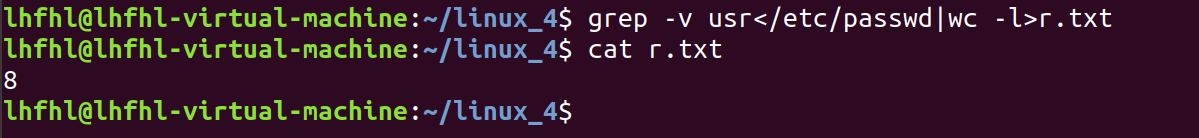
1.所给命令分析：

grep -v表示该命令的输入为/etc/passwd文件（输入重定向），-v usr选项表示从该文件筛选出不包含usr的行，并将它们输出到管道。

wc表示处理grep命令输出到管道的结果，通过wc -l命令统计行数，结果输出到r.txt文件（输出重定向）。

cat表示打印r.txt的内容到标准输出设备上。

2.查看所给命令的输出结果：

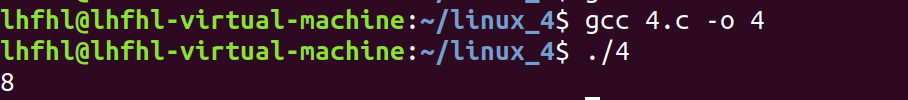


从上图可以看到所给命令的结果为8

3.编写代码



4.编译执行



结果与原命令执行一致。

### 四、实验总结

通过本次实验,提高了我的shell脚本的编写能力，也加强了我对于shell管道和重定向的理解，收获很多。

### 五、实验代码

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <fcntl.h>

#include <sys/wait.h>

int main(void)

{

int t, fd[2]; //t是wait函数的参数，fd[2]用于作为pipe函数的参数

pipe(fd); //建立管道

if (fork() == 0) //fork()函数创建子进程，开始执行grep

{

dup2(open("/etc/passwd", O\_RDONLY), 0);

dup2(fd[1], 1);

close(fd[0]);

execlp("grep", "grep", "-v", "usr", NULL);

}

if (fork() == 0) //fork()函数创建子进程，开始执行wc

{

int out = open("r.txt", O\_RDWR | O\_CREAT, S\_IRUSR | S\_IWUSR | S\_IRGRP | S\_IROTH);

dup2(fd[0], 0);

dup2(out, 1);

close(fd[1]);

execlp("wc", "wc", "-l", NULL);

}

close(fd[1]); //关闭管道

close(fd[0]); //关闭管道

wait(&t); //等待进程销毁

wait(&t);

if (fork() == 0) //fork()函数创建子进程，执行cat命令

{

execlp("cat", "cat", "r.txt", NULL);

}

wait(&t);

return 0;

}