

**Linux实验报告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 学 期 | 2024-2025学年第一学期 | | | | |
| 课 程 | Linux操作系统 | | | | |
| 专　 业 | 计算机科学与技术 | | 班 　 级 | | 22 计科2 |
| 学生姓名 | 陈伟 | | 学　 号 | | 22219111239 |
| 组 号 | 无 | 同组学生姓名 | | 无 | |
| 实验（项目）名称 | | Linux操作系统（第三次）实验 | | | |
| 任课教师 | 周旭立 | | 成 绩 | |  |

# 第3章 shell （上）

## 一、实验目的

（1）了解shell字符，了解文件系统目录结构，掌握文件命名方法、文件属性查询与确定；

（2）掌握shell命令格式和命令行编辑与历史特性的使用；

（3）理解shell命令的类型及搜索优先顺序；

（4）掌握标准I/O流、I/O重定向与管道机制；

（5）掌握变量、环境变量的定义及使用；

（6）掌握三对引号和各种替换机制；

## 二、实验环境（以后略）

装有Ubuntu 的工作站或桌面的系统，若不涉及图形界面的话，也可以是服务器系统。如果，不考虑图形界面的的话，也可以是其它版本的Linux系统。

出于学习和研究的目的，在虚拟机控制下的Linux系统可能更好。在虚拟机软件的控制下，可在同一Windows或Linux系统下安装（本书使用或其它）多个版本的Linux、Unix或Windows系统，这对于虚拟化、云计算、大数据和集群等部署与研究具有特殊意义。

## 三、实验方法与注意事项（以后略）

为了保证各种实验的成功和顺利进行，建议实验者以root用户登录系统（可以使用sudo -s 命令后输入密码进行root）。当然，以一个具有管理员属性的普通用户身份工作，也是可以的。在此种情况下，管理命令可能需要使用sudo命令来执行。但不论怎么样，实验室内的实验系统与环境是共用的，请不要在系统内做对系统或对其他用户不安全的事情。

## 四、实验过程

#### 1.shell的字符

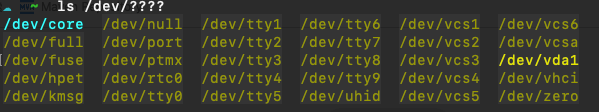
1. 通配符

#什么是通配符

通配符（Wildcard）是一种特殊符号或字符组合，用于替代或匹配未知、任意或部分已知的数据

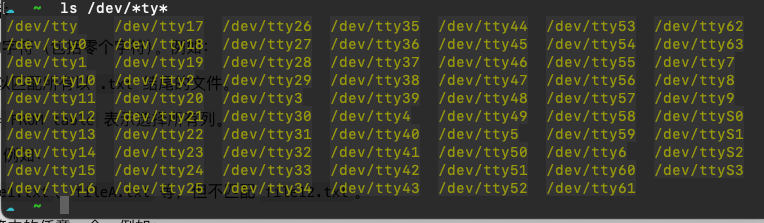
#用通配符显示 /dev 目录中所有由4个字符组成的文件 （用什么命令？并且把结果截图至此）

**Ls /dev/????**



#用通配符显示 /dev 目录中所有包含“ty”字母的文件 （用什么命令？并且把结果截图至此）

**Ls /dev/\*ty\***



#用通配符显示 /dev 目录中所有以“s”字母开头文件 （用什么命令？并且把结果截图至此）

**ls /dev/s\***



#用通配符显示 /dev 目录中所有以“n,p,r”字母开头、总长度是4个字母的文件 （用什么命令？并且把结果截图至此）

**ls /dev/[npr]???**



#用通配符显示 /dev 目录中所有不以“l,s,t”字母开头、以“d”字母结尾的文件 （用什么命令？并且把结果截图至此）

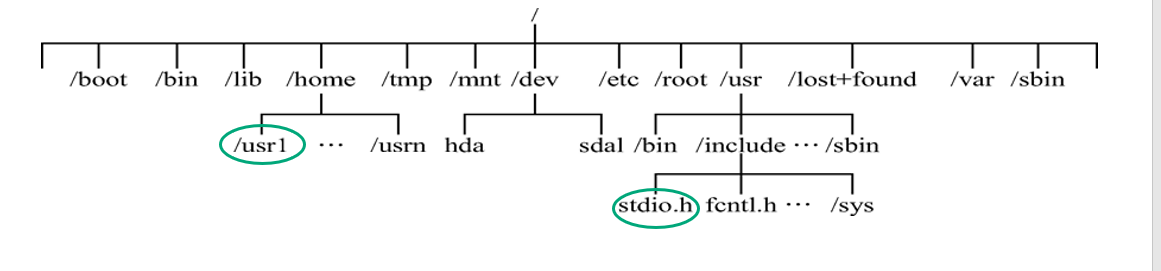
**ls /dev/[^lst]\*d**



2）shell中用什么符号做注释？

#

#### 2. 系统目录



如果目录usr1的用户要访问目录include中的文件stdio.h,相对路径和绝对路径怎么实现？

#相对路径：/usr/include/stdio.h

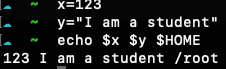
#绝对路径：../usr/include/stdio.h

#### 3．变量、三种引号及变量替换、命令替换和参数替换

1）变量（写出命令，并且截图结果）

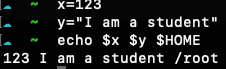
# 定义一个数值变量x，其值为123

X=123



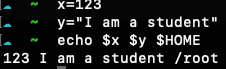
# 定义一个字符串变量y，其值为‘I am a student'

y="I am a student"



# 显示用户变量x，y和环境变量HOME的值：123 I am a student /root

echo $x $y $HOME



# 由变量x和y定义变量z，并且显示出来。其值为 123 I am a student

z="$x $y"

echo $z



# echo I am $whoami and working in $pwd 这个命令行能否成功运行？请解释原因。

不能，因为whoami和pwd是未定义的变量，可以把它们转换为命令来执行

**echo I am $(whoami) and working in $(pwd)**

2）引号机制

（1）单引号（写出命令，并且截图结果）

# 单引号在shell中起什么作用？

在 Shell 中，单引号 (') 用于**完全引用**字符串。它会将字符串中的所有字符都视为字面含义，不进行变量替换或命令替换，也不会解释任何特殊字符（如 $、\ 等）。

# 定义变量string，其值为$HOME

string='$HOME'



# 显示变量的值：$HOME （显示的结果为 “$HOME”,结果不需要有双引号）



（2）反单引号（写出命令，并且截图结果）

# 反单引号的有什么作用？

反单引号（`）在 Shell 中用于**命令替换**。它会执行其中的命令，并将命令的输出替换为相应的结果。反单引号的作用与 $() 形式的命令替换相同，但 $() 是现代 Shell 中更推荐的使用方式，因为它在嵌套命令时更直观。

# 通过命令定义变量x。并且把pwd的功能替换给变量x

x=`pwd`

# 通过命令定义变量y。并且把whoami的功能替换给变量y

y=`whoami`

# 显示变量x和y的内容



（3）双引号（写出命令，并且截图结果）

# 双引号和单引号有什么区别？双引号起什么作用？

 **单引号 (')**：完全引用，字符串中的所有字符都会按原样处理，**不会**对变量进行替换或解析特殊字符。即使字符串中有 $、\、命令等，也不会被解释。

 **双引号 (")**：部分引用，字符串中的大多数字符都会按原样处理，但**允许变量替换**和**命令替换**。例如，字符串中的 $变量名 会被替换为变量的值，反引号或 $() 中的命令会被执行并替换为输出。

# 设置变量myname= ‘xiaoming’

myname='xiaoming'

# 利用变量myname，并显示 I am a student, my name is “xiaoming” (注意要结果显示双引号)



#### 4．I/O重定向与管道

1）I/O重定向

（1）标准输入重定向

# 统计/etc/passwd的行、词和字符信息。 （写出命令，并且截图结果）

**wc /etc/passwd**



# 使用标准输入重定向的写法统计/etc/passwd的行、词和字符信息。（写出命令，并且截图结果）

wc < /etc/passwd



（2）标准输出重定向

# 覆盖方式。将当前目录信息重定向到/tmp/dir.out （写出命令，并且截图结果）



# 追加方式，将/usr信息重定向追加到/tmp/dir.out （写出命令，并且截图结果）



（3）标准错误重定向

$ # 先用命令“ls -l /home/w-w-w”产生错误，然后将此错误重定向到文件/tmp/err.out（写出命令、并且截图）

ls -l /home/w-w-w 2> /tmp/err.out

（4）标准输出和标准错误同时重定

# 先用命令“ls -l -d / /home/w-w-w”输出并且产生错误，然后标准输出和标准错误同

时重定到/tmp/err.out （写出命令、并且截图）



# 使用“cat /tmp/err.out”查看文件/tmp/err.out内容（写出命令，并且截图结果）



2）管道

# 使用管道统计系统设备目录/dev内有多少文件或子目录 （写出命令，并且截图结果）



#### 5. shell 的历史

1. shell和 terminal 有什么区别？

Shell 是一种命令解释器，用于接受用户输入的命令并与操作系统交互。它解释和执行用户输入的命令，并将结果返回给用户。常见的 Shell 有 bash、zsh、sh、csh 等

Terminal 是用户与计算机之间的界面，原本是指早期计算机系统中的物理终端，现在多指**终端仿真器**，它是一个窗口，允许用户通过键盘与计算机交互。

 **Terminal** 是一个提供用户输入界面的应用程序，用来运行 **Shell** 或其他命令行工具。

 **Shell** 是命令解释器，负责解释你在 Terminal 中输入的命令并执行相应的操作。

1. Shell大致有什么种类？

bash、zsh、sh、csh