

# 《Linux 基础教程》

---

- 讲师: 魏明择
- 时间: 2019

## 目录

---

### 《Linux 基础教程》

#### 目录

#### UNIX/Linux组管理及命令

groups 命令

groupadd 命令

groupdel 命令

groupmod 命令

#### UNIX/Linux用户管理及命令

useradd 命令

userdel 命令

passwd 命令

usermod 命令

su 命令

#### 文件的权限管理及命令

chmod 命令

chown 命令

chgrp 命令

#### Linux 磁盘管理及命令

df命令

lsblk命令

fdisk 命令

#### UNIX/Linux进程管理及命令

ps 命令

kill命令:

top 命令:

输出重定向 >

#### 网络管理及命令

ifconfig 命令

ping 命令

hostname 命令

ssh 命令

who 命令

scp 命令

#### 系统的关机、重启以及登出

关机重启

#### 系统信息相关命令

uname 命令

date 命令

- cal 命令
- 输出重定向
  - 标准输出重定向
  - 标准错误输出重定向
  - 重定向所有输出
- echo 命令
- 管道操作
  - alias 命令：
- 简单shell编程
  - shell程序运行方法：
  - shell 程序的第一行注释：
  - 终端中常用快捷键

## UNIX/Linux组管理及命令

- 组相关的配置文件
  - /etc/group

### groups 命令

- 显示当前用户在哪儿些组中
- 格式

```
groups
```

### groupadd 命令

- 作用
  - 创建一个组
- 格式

```
groupadd 组名
```

### groupdel 命令

- 作用
  - 删除一个组
- 格式

```
groupdel 组名
```

### groupmod 命令

- 作用  
修改一个组的信息
- 格式

```
groupmod 组名
```

# UNIX/Linux用户管理及命令

- 用户相关的信息  
用户的名称  
用户的ID (UID)  
用户的主组  
用户的主目录  
用户的密码  
用户的登陆shell
- 用户相关的配置文件
  - 用户信息存储文件  
/etc/passwd
  - 用户密码信息存储文件  
/etc/shadow

## useradd 命令

- 作用：  
添加一个新用户
- 格式:

```
$ useradd [ 选项 ] 新用户名
```

- 常用选项

选项	说明
-g 组名或GID	指定主组
-G 组列表	将用户添加到其它组中，列表用逗号分开。
-p 密码	在创建时指定密码
-m	创建用户主目录
-d 用户主目录	指定其它的用户主目录
-s SHELL程序路径	指定新用户的登shell

- 示例：

```
$ useradd weimz -m -s /bin/bash -p 123456
$ useradd weimingze -g 1001 -m -s /bin/bash
```

## userdel 命令

- 作用：  
删除一个已存在的用户
- 格式：

```
$ userdel [ 选项 ] 用户名
```

- 常用选项  
-r 删除用户的主目录和相关文件
- 示例：

```
$ sudo userdel -r weimingze
```

## passwd 命令

- 作用：  
修改用户密码
- 格式：

```
$ passwd [ 选项 ] 用户名
```

- 常用选项  
-d 删除用户密码

## usermod 命令

- 作用：  
修改用户的相关信息
- 格式：

```
$ usermod [ 选项 ] 用户名
```

- 常用选项

选项	说明
-l 新登陆名	重新指定新的登陆名
-g 组名或GID	强制指定新主组
-G 组列表	新的附加组列，列表用逗号分开。
-p 密码	指定新密码
-d 用户主目录	指定其它的用户主目录
-s SHELL程序路径	指定新用户的登shell

## su 命令

- 作用：  
切换到指定的用户
- 格式:

```
$ su 用户名
```

- 说明  
如不指定参数，默认切换到root用户
- 练习(创建帐户):
  1. 在你的Ubuntu 虚拟机内创建你名字的用户 \$ sudo useradd weimz -m -s /bin/bash
  2. 用passwd给你名字的用户设置初始密码是123456
  3. 用su 命令切换到你名字的那个用户, 创建一个notes.txt文件，并用vi 写入你的姓名
  4. 退出以你名字命名的用户,回到tarena 用户
  5. 查看刚才你写过的文件
  6. 尝试修改你刚才写过的文件
  7. 查看/etc/shadow 下你名字帐户对应的密码值
  8. 修改你名字帐户的密码,再看看/etc/shadow里有什么变化
  9. 删除你名字的帐户（可以考虑不做此步）

## 文件的权限管理及命令

- 查看文件的权限  
ls -l
- Unix/Linux文件类型

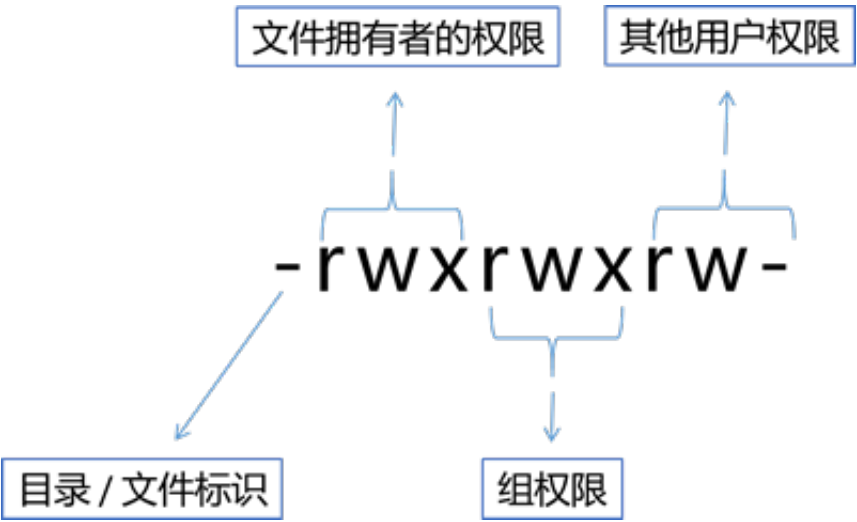
符号	类型
d	文件夹(d)
-	普通文件(-)
c/b	设备文件(c/b)
p	管道文件(p)
l	链接文件（Unix中是替身文件）(l)
s	套接字文件(s)

- Linux系统上对文件的权限有着严格的控制，用于如果相对某个文件执行某种操作，必须具有对应的权限方可执行 成功。

权限类型：权限类型一般包括读，写，执行，对应字母为 r、w、x

权限粒度：文件拥有者（创建文件的用户）、组（创建文件用户所在的组）、其它组三。

所以，一个文件的权限表示为9中状态，分别指拥有这、组、其它用户对该文件是否具有读、写、执行的权限



在rwx三个权限中，从右至左依次用值1,2,4来表示，所以拥有某个文件的读、写、执行权限也可以用三个权限值累加的方式来表示，如下表所示：

- 文件的权限类型
  - r 读权限
    - 文件，具有读取文件内容的权限。
    - 文件夹，具有读取文件夹内文件目录的权限
  - w 写权限
    - 文件，具有新增、修改、删除文件内容的权限。
    - 文件夹，具有新建、删除、修改文件名、移动目录内文件的权限。
  - x 执行权限
    - 文件，具有执行文件的权。
    - 文件夹，具有进入目录的权限。
  - 无权限
- 文件的权限的分组
  - 用户权限(user) 组权限(group) 其它(other)
- 最高权限

```
rwXrwxrwx
```

- 最低权限

```
-----
```

- 文件权限是可以改的

```
r--r--r--
-W--W----
----W----
```

- 属主
  - Unix/Linux系统中，每一个文件都必然属于某一用户。我们把这个用户叫做所有者，也叫属主
  - 文件的所有者（属主）才有权利修改文件的权限。
- 属组
  - Unix/Linux系统中，每一个文件都必然属于某一组。我们把这个组叫 属组
- 其他
  - 不属于属主也不属于属组的用户，叫做其它用户（other）

## chmod 命令

- 作用
  - 用来修改文件的权限
- 格式

```
chmod 权限 文件名/文件夹名
```

- 权限
  - u 用户（所有者）
  - g 同组用户
  - o 其他用户
  - a 所有用户
  - + 添加权限
  - - 去限权限
- 示例

```
chmod u+r yy.txt
chmod o-rwx yy.txt
chmod g+rw yy.txt
chmod g-w yy.txt
chmod a+x yy.txt
chmod a-r yy.txt
chmod 777 yy.txt
chmod 000 yy.txt
```

- 以数字方式去修改文件权限

r:4

w:2

x:1

```
chmod 764 file.txt
chmod 664 file.txt
chmod 777 file.txt
```

## chown 命令

- 作用：  
修改文件的属主（需要管理员权限）
- chown命令的格式

```
$ chown [owner][:group] [选项] FILES
```

- 常用选项  
-R 文件名 递归对所有文件进行相同的操作

## chgrp 命令

- 作用：  
修改文件的属组
- chgrp命令的格式

```
$ chgrp GROUP [选项] FILES
```

- 常用选项

```
-R 文件名 递归对所有文件进行相同的操作
```

- 示例

```
#将文件afile属组设置为weimz
$ chgrp weimz afile
等同于
$ chown :weimz afile
```



- 练习：
  1. 用tarena用户创建一个文件: myprog.sh
  2. 编辑 myprog.sh 添入以下内容:

```
#!/bin/bash

ls /
cd ~
mkdir "xxx来过"
```

3. 修改此文件的权限,让tarena用户有可执行此文件的权限
4. 用tarena用户执行此文件:  
\$ ./myprog.sh

- 练习
  - 操作文件权限:  
创建两个文件 a.txt b.txt写入一些内容  
让其它用户不能读取a.txt文件  
让用组户和其它用户不能读取b.txt  
让用户自身对所有文件有执行权限

## Linux 磁盘管理及命令

### df命令

- 作用  
用于显示目前在Linux系统上的文件系统的磁盘使用情况统
- 示例

```
$ df
文件系统              1K-块      已用      可用  已用%  挂载点
udev                  1971520          0   1971520    0% /dev
/dev/sda1             205373416  18106860 176764516   10% /
/dev/loop0             144128     144128          0  100% /snap/gnome-3-26-
1604/97
/dev/loop2             15104      15104          0  100% /snap/gnome-
characters/359
```

- 示例2

```
$ df -h
文件系统          容量  已用  可用 已用% 挂载点
udev              1.9G   0    1.9G   0% /dev
/dev/sda1         196G   18G   169G   10% /
/dev/loop0        141M   141M    0  100% /snap/gnome-3-26-1604/97
/dev/loop2        15M    15M    0  100% /snap/gnome-characters/359
```

## lsblk命令

- 作用  
用来查看磁盘(block设备)的信息.
- 示例:

```
$ lsblk
NAME      MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
loop0      7:0    0 140.7M  1 loop /snap/gnome-3-26-1604/97
loop1      7:1    0  14.8M  1 loop /snap/gnome-characters/367
sda        8:0    0   200G  0 disk
└─sda1     8:1    0   200G  0 part /
sr0       11:0    1  1024M  0 rom
```

## fdisk 命令

- 作用  
用来查看磁盘分区或为磁盘分区
- 常用选项  
-l 查看磁盘分区
- 示例:

```
sudo fdisk -l
[sudo] tarena 的密码:
Disk /dev/sda: 200 GiB, 214748364800 字节, 419430400 个扇区
单元: 扇区 / 1 * 512 = 512 字节
扇区大小(逻辑/物理): 512 字节 / 512 字节
I/O 大小(最小/最佳): 512 字节 / 512 字节
磁盘标签类型: dos
磁盘标识符: 0x1d3d809a

设备          启动  起点      末尾      扇区  大小  Id  类型
/dev/sda1     *      2048 419428351 419426304 200G 83 Linux
```

## UNIX/Linux进程管理及命令

## ps 命令

- 作用：  
查看进程
- 格式: ps [选项]
- 常用选项  
-aux 查看当前系统内的所有进程的详细信息
- 示例：

```
ps          # 查看当前环境的进程
ps aux      # 查看当前系统内的所有进程的详细信息
```

## kill命令:

- 作用：  
杀死进程
- 格式:

```
kill 进程的pid号
```

- 注: 在终端运行时 可以用 ctrl+c 终止进程

## top 命令:

- 作用:  
查看程序运行状态
- 退出  
Q 键

## 输出重定向 >

- 作用： 把输出到终端的内容输出到指定文件
- 格式：

```
命令      >      文件
```

- 示例：

```
ls > a.txt
cat passwd > a.txt
head -5 passwd > a.txt
echo "Python" > a.txt
```

- 注意

1. 如果文件存在，则覆盖原文件
  2. 如果文件不存在，则创建一个新文件
- 练习
    1. 在用户主目录下创建文件 echo.txt
    2. 用vi在echo.txt中添加如下内容  
人生难得起起落落,还是要坚强地生活!
    3. 用echo命令将echo.txt中的内容变为：我是echo过来覆盖你们的
    4. 在用户主目录下的文件以长格式的方式重定向输入到echo.txt中
    5. 查看echo.txt中的内容

## 网络管理及命令

---

- (以太网和WIFI无线)
- 什么是IP地址:  
IP地址是标识互联网上一块网络设备接口的逻辑地址
- IP地址分为:  
IPv4  
IPv6
- IPv4地址的组成: xxx.xxx.xxx.xxx  
192.168.1.100 (IPv4)  
(0~255).(0~255).(0~255).(0~255)

## ifconfig 命令

- 作用:  
查看本机IP地址，该地址可以理解为本机在网络上的门牌号，该编号是唯一的
- 查看和设置IP命令

```
$ ifconfig  
$ ifconfig eth0 显示一个以太网卡的配置
```

## ping 命令

- 测试网络的连接状态
- 格式

```
ping IP地址或域名
```

- 示例:

```
ping 172.60.50.78  
ping www.baidu.com
```

## hostname 命令

- 查看和设置 主机名

## ssh 命令

- 作用  
远程登录到一台主机
- 格式

```
ssh [用户名@]ip地址或域名
```

## who 命令

- 作用  
显示登录系统中的用户的信息  
查看当前主机有哪儿几个用户登陆
- 格式

```
who
```

## scp 命令

- 作用：  
远程复制文件或文件夹
- 格式：

```
scp 源文件地址 目标文件地址
```

- 远程文件地址格式: 用户名@IP地址:绝对路径
- 示例：

```
# 复制我的一个文件day03.tar.gz到172.60.50.78的 /home/weimz/下  
$ scp ./day02.tar.gz weimz@172.60.50.78:/home/weimz/
```

## 系统的关机、重启以及登出

---

### 关机重启

- 用法示例

```
poweroff # 现在关机
shutdown -h now # 关闭系统
shutdown -h hours:minutes # 按预定时间关闭系统
shutdown -c # 取消按预定时间关闭系统
shutdown -r now # 重启
reboot # 重启(2)
exit # 退出登陆
```

- 练习：
  1. 把你自己今天的笔记打包成为: day03.tar.gz
  2. 用scp 把笔记发送到你同桌的电脑上备份
  3. 用ssh 登陆，在你同桌的电脑上创建一个你自己的目录
  4. 把day04.tar.gz 复制到你自己的目录里
  5. 解包day04.tar.gz到你自己目录. 用vim 查看文件信息是否与你自己电脑上的相同

## 系统信息相关命令

### uname 命令

- 作用：  
查看操作系统信息(仅UNIX/Linux)
- 格式:

```
uname [-a]
```

- 常用选项:  
-a 表示显示全部信息  
如:

```
$ uname -a
```

### date 命令

- 作用：  
显示日期时间
- 格式:

```
date +%D # 月/日/年 格式显示日期
date +%T # 时:分:秒 格式显示时间
```

### cal 命令

- 作用：  
显示日历:

- 示例

```
$ cal          # 显示当月的日历
$ cal 2018     # 显示2018年的全部日历
$ cal 8 2020   # 显示2020年8月的日历
```

## 输出重定向

- 输出分两种：
  1. 标准输出
  2. 标准错误输出

## 标准输出重定向

```
^^^
>      将一个命令的标准输出重定向到指定文件
>>    将一个命令的标准输出追加到指定文件的末尾
^^^
```

- 示例：

```
$ find /usr -name "ls" > stdout.txt
```

- `>>` 和 `>` 比较:

<code>&gt;&gt;</code>	不删除原来文件内容
<code>&gt;</code>	清空原文件内容

- 示例：

```
grep "weimz" * 2> /dev/null
```

## 标准错误输出重定向

```
^^^
2>      将一个命令的标准错误输出重定向到一个文件
2>>    将一个命令的标准错误输出追加到一个文件的末尾
^^^
```

- 示例

```
mv aa.txt bb.txt
find /etc -name "passwd" 2> error.txt
find /etc -name "passwd" >f.txt 2> error.txt
```

## 重定向所有输出

```
~ ~ ~  
&>  
&>>  
~ ~ ~
```

- 作用：  
将所有标准输出和标准错误输出重定向到一个文件中

## echo 命令

- 作用  
将文本作为终端输出
- 格式：

```
echo "字符串"
```

- 示例

```
echo "hello"
```

- 练习：

已知有一个模板文件template.txt的内容是5行文本：

群通知

-----

xxxxxx

-----

群主：魏老师

别有两个文件：file1.txt file2.txt中写入一定的内容。

要求：写一个SHELL程序make\_notices.sh,用以上file1.txt 和file2.txt中的内容生成两个群通知。notice1.txt和 notice2.txt

提示：

用head, tail, 重定向等命令组合

- 练习：

已知有一个文件的内容是5行文本：

aaaa

bbbb

CDEF

dddd

eeee

用head, tail, echo 命令组合将源文件第三行改为：

cccc后存入另一个文件abcde.txt中

## 管道操作



## 运算符 |

- 作用：  
将命令的"输出"重定向为另一个命令的"输入"
- 格式:

```
命令1 [参数] | 命令2 [参数] | 命令3 [参数] ...
```

- 示例:

```
cat /etc/passwd | grep "tarena" -n
```

## alias 命令:

- 作用：  
给命令取别名
- 格式:

```
alias 新命令名='旧命令名 [选项...]'
```

- 示例:

```
alias delete='rm -i'
```

## 简单shell编程

- shell程序文件通常以 .sh 结尾
- 以#开头直至结尾的内容视为注释内容，不能与运行
- 第一行 #! xxx 用于制定解释此文件的 程序

## shell程序运行方法:

### 1. 方法1

```
$ bash shell程序文件名
```

### 2. 方法2

```
$ chmod +x myshell.sh  
$ ./myshell.sh # 需要执行权限
```

## shell 程序的第一行注释:

- 作用：  
指定解释执行此文件的程序是谁
- 示例:

```
#!/bin/bash
```

- 练习：
  - 用shell 编程来创建文件夹：
    1. 在~/aid1709/下创建pbase文件夹
    2. 在pbase文件夹内创建 day01, day02文件夹
    3. 在day01内创建 day01.txt文件,并写入"Python基础"
- 练习
  - 复习今天的命令
  - 玩Linux操作系统

## 终端中常用快捷键

快捷键	说明
Tab	自动补齐
Ctrl + c	终止当前命令执行
Ctrl + l	清屏
上下箭头	翻出之前执行过的命令 history
Ctrl + shift + '+'	终端字体放大
Ctrl + '-'	终端字体缩小
Ctrl + shift + t	添加一个终端
exit	退出终端