

<https://www.bilibili.com/video/BV1Sv411r7vd?p=40>

## 基础篇

- Linux的应用领域
- 网络连接的三种方式
- Linux的目录结构
- Linux远程登陆
- Linux-vim和vi编辑器
- Linux的关机重启，用户登录和注销
- Linux的用户管理
  - 基本介绍
  - 密码修改
  - 用户删除
  - 查询用户信息指令
  - 用户组
  - 用户组相关文件
  - 指定运行级别
  - 找回root密码（经典面试题）
- 帮助指令（p27）
- 文件目录类指令
  - cp指令 拷贝指令
  - rm指令 移除文件或目录
  - mv指令 移动文件与目录或者重命名
  - cat指令 查看文件内容（p31）
- 搜索查找类
  - find指令
  - locate指令
- 压缩和解压类p36
- 组管理和权限管理
  - 文件/目录所有者
  - 组的创建
  - 其他组
  - 权限的基本介绍
  - 修改权限-chmod
  - 修改文件/目录所有者-chown
  - 修改文件/目录所在组
- 定时任务的调度（p52）
  - crontab 进行 定时任务的设置
  - at 定时任务
- Linux磁盘分区，挂载（p58）

# 基础篇

---

## Linux的应用领域

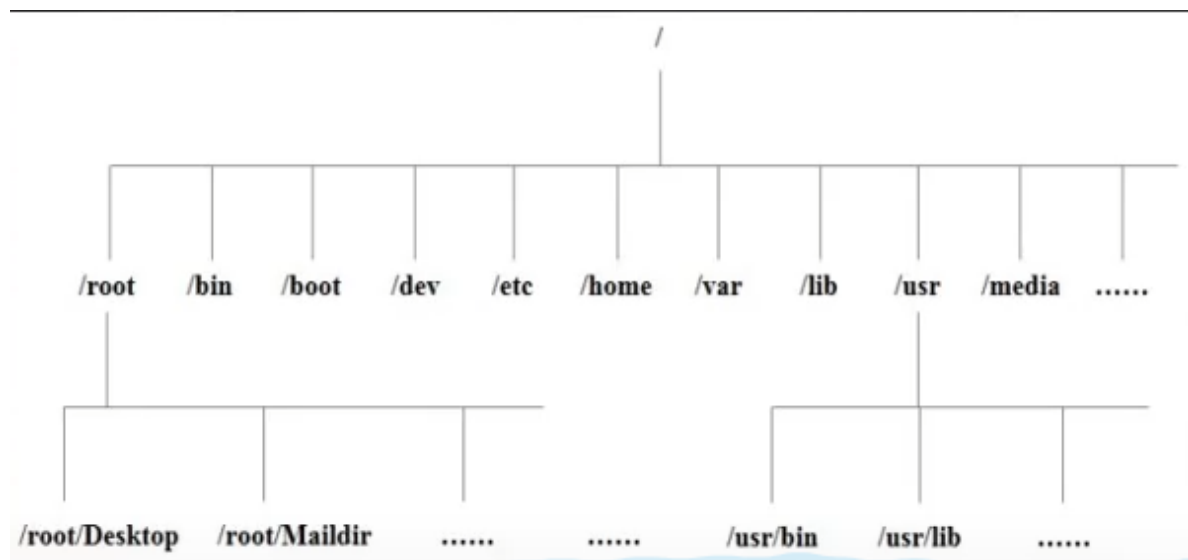
- 服务器领域（最强的领域）
- 嵌入式领域（免费，运行稳定，可以根据要求进行软件剪裁，内核最小可以达到几百KB）
- 个人桌面开发（弱）

## 网络连接的三种方式

1. 桥接模式:同一个网段最多255个IP地址，桥接模式下的虚拟机可以和外部系统通讯但是会占用一个IP地址，容易导致IP冲突。
2. NAT模式（网络地址转换模式）:虚拟机创建一个虚拟IP和主机通讯，由主机的IP和外界系统通讯，不会造成IP冲突。
3. 仅主机模式：独立系统不和外部发生联系

## Linux的目录结构

Linux的文件系统是采用层级式的树状目录结构，在此种最上层是根目录“/”，然后在此目录下在创建其他目录。（在Linux世界力，一切皆文件）把主体结构背下来！



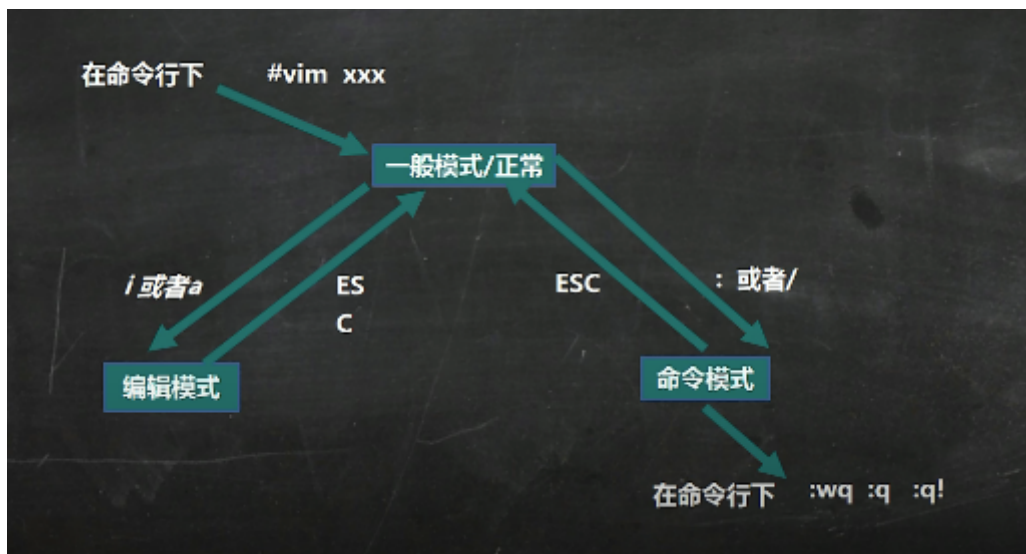
- **【常用】** /bin (/usr/bin,/usr/local/bin) 是Binary的缩写，这个目录存放最经常使用的命令
- /sbin (/usr/sbin,/usr/local/sbin) s是super user的意思，这里存放系统管理员的系统管理程序
- **【常用】** /home 存放普通用户的主目录
- **【常用】** /root 系统管理员目录，也称作超级权限者的用户主目录
- /lib 系统开机所需要的最基本的动态链接库，其作用类似于Windows的DLL文件。几乎所有应用程序都需要这些共享库。
- /lost+found 这个目录一般情况下是空的，当系统非法关机后，这里存放了一些文件。
- **【常用】** /etc 所有系统管理所需要的配置文件和子目录 (my.conf)
- **【常用】** /usr 这是一个非常重要的目录，用户的很多应用程序和文件都放在这个目录下，类似于windows的program files目录。
- **【常用】** /boot 存放的是启动Linux时使用的一些核心文件，包括一些链接文件和镜像文件
- /proc **【不能动，动了容易系统崩溃】** 这个目录是一个虚拟目录，他是系统内存的映射，访问这个目录来获取系统信息
- /srv **【不能动，动了容易系统崩溃】** service的缩写，该目录存放一些服务启动之后需要提取的数据
- /sys **【不能动，动了容易系统崩溃】** 这是Linux2.6内核的很大的变化。该目录下安装了2.6内核中出现的一个文件系统sysfs
- /tmp 这个目录是用来存放一些临时文件的
- **【重要】** /dev 类似于windows的设备管理器，把所有的硬件用文件形式存储
- **【常用】** /media Linux系统会自动识别一些设备，例如U盘，光驱，当识别后，Linux会把识别的设备挂载到这个目录下
- **【常用】** /mnt 系统提供该目录是为了让用户临时挂载别的文件系统，我们可以将外部存储挂载在/mnt/上，然后进入目录就可以查看里面内容了。（主机和虚拟机文件共享）

- /opt 这是给主机额外安装软件存放的目录。如安装ORACLE数据库就可以放到该目录下。默认为空。
- **【常用】** /usr/local 安装软件所安装的目录。一般是通过编译源码方式安装的程序
- **【常用】** /var 这个目录中存放着不断扩充的东西，习惯将经常修改的目录放在这个目录下，包括各种日志文件
- /selinux [security-enhanced linux] SELinux是一种安全子系统，它能控制程序只能访问特定文件，有三种工作模式，自行设置。

## Linux远程登陆

Xshell6, xftp6

## Linux-vim和vi编辑器



进入/创建文件 vim hello.cpp

1. 拷贝当前行一般模式下 yy, 拷贝向下的5行 5yy, 粘贴 p
2. 删除当前行 dd, 删除向下5行 5dd
3. 在文件中查找某个关键字，进入命令模式，输入 /关键字 然后回车，输入n 查找下一个
4. 设置行数显示 : setnu 取消 : setnonu
5. 快速到首行 gg 末行 G
6. 撤销动作 一般模式下 u
7. 光标瞬间移动 在一般模式下 20 然后 shift+g , 直接到20行
8. ....

## vi / vim 键盘图

**Esc**  
命令模式

转换大小写 ~	外部过滤器 !	@ 运行宏 @	# prev ident 3	\$ 行末 4	% 括号匹配 5	^ "款" 行首 6	& 重复 :s 7	* next ident 8	( 句首 9	) 下一句首 0	"soft" bol down - 前一"硬"行首	+ 后一行首 = 自动格式化
跳转到标注 .	1	2	3	4	5	6	7	8	9	0	-	=
切换到 ex 模式 Q	下一单词 W	词尾 E	替换模式 R	back till T	拷贝行 Y	撤销行内命令 U	到行首插入 I	分段(前) O	分段(后) o	粘贴(前) P	{ 段首	} 段尾
录制宏 q	下一单词 w	词尾 e	替换字符 r	t till t	拷贝 y	撤销命令 u	插入模式 i			p 粘贴(后)	. 杂项	. 杂项
A 在行末附加 a	S 删除行并插入 s	D 删除整行 d	F 行内字符反向查找 f	G 文尾/行号 g	H 屏幕顶行 h	J 合并两行 j	K 帮助 k	L 屏幕底行 l	:	ex 命令	" 寄存器标识	行首/列
Z 退出 Z	X 退格 X	C 修改至行末 c	V 可视行模式 v	B 前一行 b	N 查找上一处 n	M 屏幕中间行 m	< 反缩进	> 缩进	?	向前搜索	! 跳转到标注的行首	未使用!
										重复 :u/T/f/F	! 跳转到标注的行首	未使用!

**动作** 移动光标, 或者定义操作的范围  
**命令** 直接执行的命令, 红色命令进入编辑模式  
**操作** 后面跟圆表示操作范围的指令

**extra** 特殊功能, 需要额外的输入  
Q 后跟字符参数

**w,e,b 命令**  
小写(b): quux(foo, bar, baz);  
大写(B): QUUX(FOO, BAR, BAZ);

**主要 ex 命令:**  
:w (保存), :q (退出), :q! (不保存退出)  
:e f (打开文件 f),  
:%s/x/y/g ('y' 全局替换 'x'),  
:h (帮助 in vim), :new (新建文件 in vim)

**其它重要命令:**  
CTRL-R: 重复 (vim),  
CTRL-F/-B: 上翻/下翻,  
CTRL-E/-Y: 上滚/下滚,  
CTRL-V: 块可视模式 (vim only)

**可视模式:**  
漫游后对选中的区域执行操作 (vim only)

**备注:**  
(1) 在 拷贝/粘贴/删除 命令前使用 "x (x=a..z,\*) 使用命令的寄存器('剪贴板') (如: "ay\$ 拷贝剩余的行内容至寄存器 'a')  
(2) 命令前添加数字 多遍重复操作 (e.g.: 2p, d2w, 5i, d4j)  
(3) 重复本字符在光标所在行执行操作 (dd = 删除本行, >> = 行首缩进)  
(4) ZZ 保存退出, ZQ 不保存退出  
(5) zt: 移动光标所在行至屏幕顶端, zb: 底端, zz: 中间  
(6) gg: 文首 (vim only), gf: 打开光标处的文件名 (vim only)

## Linux的关机重启, 用户登录和注销

- shutdown -h now 立即关机 或者 halt
- shutdown -h 1 1分钟后关机
- shutdown -r now 立即重启 或者 reboot

注意事项, 不管事重启还是关闭系统首先运行sync, 把内存中的数据写到磁盘

虽然上述命令中 在关机前进行了sync, 但是小心驶得万年船

- 进入管理员权限 su - root (-两边都有空格)
- logout 退出管理员账户

## Linux的用户管理

## 基本介绍

Linux系统是一个多用户多任务的操作系统, 任何一个要使用系统系统资源的用户, 都必须向系统管理员申请一个账号, 然后以这个账号的身份进入系统。

基本语法: useradd 用户名

应用案例

添加一个用户 ros, 默认该用户在家目录 /home/ros

当创建用户成功后, 会自动的创建和用户同名的家目录

也可以通过 useradd -d指定目录 新的用户名, 给新创建的用户指定家目录 useradd -d /home/test ros (/home/test为指定目录)

## 密码修改

passwd 用户名 （不加用户名的话，默认为当前登陆的用户）

补充，显示当前用户所在的目录 pwd

---

## 用户删除

1. 保留家目录的情况：
  - 获得管理员权限
  - userdel 用户名
2. 不保留全部删除（谨慎使用）：
  - 获得管理员权限
  - user -r 用户名

## 查询用户信息指令

基本语法：id 用户名

指令：who am i （查询最初登陆到系统的用户）

## 用户组

介绍：类似于角色，系统可以对有共性（相同权限）的多个用户进行统一管理

新增组指令：groupadd 组名 （不加组名，默认创建一个和用户名相同的新组）

删除组：groupdel 组名

修改组（将一个用户放入另一个组）：usermod -g 用户组 用户名

## 用户组相关文件

- /etc/passwd文件：用户（user）的配置文件，记录用户的各种信息  
每行的含义：用户名：口令：用户标识号：（注释性描述）：主目录：登陆Shell(shell是命令和Linux内核中间的解释器，国内一般是bash)
- /etc/shadow文件：口令配置文件  
每行含义：登录名：加密口令：最后一次修改时间：最大时间间隔：警告时间：不活动时间；失效时间；标志
- /etc/group 文件：组（group）的配置文件，记录Linux包含的组的信息  
每行含义：组名：口令：组标识号：组内用户列表

## 指定运行级别

multi-user.target 多用户有网络 （类似于无桌面的，仅有命令行。节省资源）

graphical.target 图形化（默认一般情况）

查看运行级别：systemctl get-default

设置运行级别：systemctl set-default TARGET.target （设置完之后，重启依然是修改后的运行级别）

## 找回root密码（经典面试题）

1. 启动系统，在开机界面按 e，进入编辑界面
2. 使用键盘控制光标，找到Linux16开头所在的行数，在最后输入 init=bin/sh
3. 输入完成之后，直接快捷键Ctrl+x 进入单用户模式
4. 接着在光标闪烁的位置输入：mount -o remount.rw / 完成后回车
5. 在新的一行最后输入：passwd 输入完成回车。输入密码，回车然后在此确认
6. 接着在光标闪烁处，输入：touch /.autorelabel 回车
7. 输入exce /sbin init 回车，等待系统自动需改密码，过程有点长，需要耐心等待

## 帮助指令（p27）

- man

基本语法：man 【命令或配置文件】（获得帮助信息）

案例：查看ls命令的帮助信息 man ls

在linux下，隐藏文件是.开头的，选项可以组合使用 比如ls -al，ls -la /root

- help

基本语法：help +命令（获得shell内置命令的帮助信息）

案例：查看cd命令的帮追信息 help cd

（看不懂英文可以直接百度查）

## 文件目录类指令

- 基本语法：pwd （显示当前工作目录的绝对路径）
- 基本语法：ls [选项 目录或者文件]

常用选项

-a 显示当前目录的所有文件和目录，包括隐藏的（隐藏目录都是 . 开头的）

-l 以列表的方式显示信息

---

### • cd指令

基本语法：cd [参数]（切换到指定目录）

理解绝对路径和相对路径：

绝对路径：从根目录到目标文件位置

相对路径：从当前位置到目标文件位置

cd ~ 或者cd：回到自己的家目录，比如你是root，cd~后 到 /root

cd.. 回到当前目录的上一级目录

- 应用实例：

1. 使用绝对路径切换到root目录：cd /root
2. 使用相对路径切换到root目录,比如在/home/ros：cd ../../root
3. 回到上级目录：cd..
4. 回到家目录：cd~

- 
- **mkdir指令** （用于创建目录）

基本语法：mkdir [选项] 要创建的目录

常用选项：

-p 创建多级目录

应用实例

创建一个目录：mkdir /home/dog

创建多级目录：mkdir -p /home/animal/dog

- **rmdir指令** （用于删除空目录）

基本语法：rmdir [选项] 要删除的空目录

应用实例：删除一个目录 /home/dog

rmdir /home/dog

注意细节：rmdir 删除的是空目录，如果目录下有内容时无法删除

如果要删除非空目录，需要用 rm -rf,比如rm -rf /home/animal

---

- **touch命令** （创建空文件）

基本语法：touch 文件名称

---

## cp指令 拷贝指令

基本语法：cp [选项] source 【需要拷贝的文件】 dest 【拷贝到的位置】

常用选项

-r 递归复制整个文件夹

强制覆盖不提示 \cp

---

## rm指令 移除文件或目录

基本语法：rm [选项] source

常用选项

-r ： 递归删除整个文件夹

-f: 强制删除不提示

---

## mv指令 移动文件或目录或者重命名

基本语法

mv oldnamefile newnamefile (重命名) 两个文件在同目录下

mv /temp/movefile/targetFolder (移动文件) 类似于cut剪切

## cat指令 查看文件内容 (p31)

cat [选项] source

常用选项

-n 显示行号

使用细节：cat只能浏览文件，而不能修改，为了浏览方便，一般会带上 |more（管道命令）

知识点补充：管道命令---把前面命令得到的结果，给下一个命令处理好

more指令

more指令是基于VI编辑器的文本过滤器，它以全屏幕的方式按页显示文本文件的内容。more指令中内置了若干快捷键，详见操作说明。

操作	功能说明
空白键 (space)	代表向下翻一页；
Enter	代表向下翻『一行』；
q	代表立刻离开 more ，不再显示该文件内容。
Ctrl+F	向下滚动一屏
Ctrl+B	返回上一屏
=	输出当前行的行号
:f	输出文件名和当前行的行号

基本语法：more 查看的文件

### • ln 指令

软连接也称为符号链接，类似于windows里的快捷方式，主要存放了链接其他文件的路径

基本语法：ln -s [原文件目录] [软链接名] （给原文件创建一个软链接）

应用实例：

1.在/home目录下创建一个软链接myroot，链接到/root目录

```
ln -s /root/ /home/myroot
```

2.删除软链接myroot

```
rm /home/myroot
```



细节说明：当我们使用pwd指令查看目录是，仍然看到的时软链接所在的目录

---

## • history 指令

查看已执行过的历史指令

基本语法：history

应用实例

1.显示所有历史指令

history

2.显示最近使用过的10个指令

history 10

3.执行历史编号为5的指令

i5

---

## • date指令 显示当前日期

基本语法

date 显示当前时间

date +%Y 显示年份

date +%m 显示月份

date +%d 显示哪一天

date "+%Y-%m-%d %H:%M:%S " 显示年月日时分秒

## • 时间设置

基本语法

date -s 字符串时间

案例：设置系统时间，比如设置成2022-03-31 14: 00: 00

date -s "2022-03-31 14:00:00"

---

## • cal指令

查看日历指令

基本语法

cal[选项] (不加选项，显示本月日历)

案例

1.显示当前日历

cal

2.显示2020年日历

## 搜索查找类

### find指令

find指令将从指定目录向下递归遍历其各个子目录，将满足条件的文件或者目录显示在终端

基本语法

find [搜索范围] [选项]

选项说明

选项	功能
<b>-name&lt;查询方式&gt;</b>	<b>按照指定的文件名查找模式查找文件</b>
<b>-user&lt;用户名&gt;</b>	<b>查找属于指定用户名所有文件</b>
<b>-size&lt;文件大小&gt;</b>	<b>按照指定的文件大小查找文件。</b>

应用实例

1.按文件名：根据名称查找 /home目录下的hello.txt文件

find /home -name hello.txt （若是所有txt文件 最后改成 \*.txt）

2.按拥有者：查找 /opt目录下，用户名称为nobody的文件

find /opt -user nobody

3.查找整个linux系统下大于200M的文件（+n 大于 -n小于 n等于,k,MG）

find / -size +200M

补充 ls -lh 查看文件 （h代表适合人类的观看方式）

---

### locate指令

快速定位文件路径。利用事先建立的系统中虽有文件名称及路径的locate数据库实现快速定位。locate指令无需遍历整个文件系统，查询速度较快，为了保证查询的结果准确度，管理员必须定期更新locate时刻

基本语法

locate 搜索文件

特别说明:由于locate 指令基于数据库进行查询，所以第一次运行前，必须使用update指令创建locate数据库

应用实例

使用locate指令快速定位hello.txt文件所在目录

补充 which指令：查找指令在某个目录 比如which ls

---

## 压缩和解压类p36

### • gzip/gunzip 指令

基本语法

gzip 文件 (压缩文件, 只能将文件压缩为\*.gz文件)

gunzip 文件.gz (解压缩文件命令)

应用实例

---

### • zip/unzip指令

zip用于压缩文件, unzip用于解压, 在项目打包发布中很有用

基本语法

zip [选项] xxx.zip (压缩文件和目录命令)

unzip [选项] xxx.zip (解压文件)

常用选项

zip: -r 递归压缩, 即压缩目录

unzip: -d 指定解压后存放目录

案例

1.将 /home下的所有文件/文件夹进行压缩成myhome.zip

zip -r myhome.zip/home /home (/home 本身也被被压缩)

2.将myhome.zip 解压到 /opt/tmp目录下

unzip -d /opt/tmp home/myhome.zip

---

### • tar指令

基本语法

tar [选项] xxx.tar.gz 打包内容 (打包目录,压缩后的为文件格式 tar.gz)

常用选项

选项	功能
<b>-c</b>	产生.tar打包文件
<b>-v</b>	显示详细信息
<b>-f</b>	指定压缩后的文件名
<b>-z</b>	打包同时压缩
<b>-x</b>	解包.tar文件

应用案例

1.压缩多个文件, 将 /home/pig.txt和/home/cat.txt压缩成pc.tar.gz

```
tar -zcvf pc.tar.gz /home/pig.txt /home/cat.txt
```

2.将/home的文件夹压缩成myhome.tar.gz

```
tar -zcvf myhome.tar.gz /home/
```

3.将pc.tar.gz解压到当前目录

```
tar -zxvf pc.tar.gz
```

4.将myhome.tar.gz解压到 /opt/tmp2目录下

```
tar -zxvf /home/myhome.tar.gz -C /opt/tmp2/
```

---

## 组管理和权限管理

在Linux中的每个用户必须属于一个组，不能独立于组外。在linux中的每个文件有所有者，所在组，其他组的概念。（1.所有者，2.所在组，3.其他组，4.改变用户所在组） P40

### 文件/目录所有者

一般为文件的创建者，谁创建了该文件，就自然的成为该文件的所有者

查看文件所有者

指令：ls -ahl

修改文件所有者

指令：chown 用户名 文件名

---

### 组的创建

基本指令 groupadd 组名

应用实例

1.创建一个组monster

```
groupadd monster
```

2.创建一个用户fox,并放入monster组中

```
useradd -g monster fox
```

（当某个用户创建了一个文件后哦，这个文件的所在组就是该用户的所在组）

查看文件/目录所在组

ls -ahl

修改文件所在的组

基本指令

```
chgrp 组名 文件名
```

---

## 其他组

除文件的所有者和所在组的用户外，系统的其他用户都是文件的其他组

改变用户所在组：

在添加用户时，可以指定将该用户添加到哪个组中，同样的用root的管理权限可以改变某用户所在的组

`usermod -g 组名 用户名`

`usermod -d 目录名 用户名` 改变用户登陆的初始目录

---

## 权限的基本介绍

### 1.第0位

确定文件类型 (d, -,l,c,b)

l是链接，相当于windows的快捷方式

d是目录，相当于windows的文件夹

c是字符设备文件，鼠标，键盘等

b是块设备，比如硬盘

2.第1-3位确定所有者（该文件的所有者）拥有该文件的权限。--User

3.第4-6位确定所属组（同用户的组）拥有该文件的权限，--Group

4.第7-9位确定其他用户对该文件的权限 --Other

rwX作用到文件

1. r 代表可读：可以读取查看

2. w代表可写：可以修改，但并不代表可以删除（删除一个文件的条件是对该文件所在目录有写的权限）

3. x代表可执行：可以被执行

rwX作用到目录

1.r 代表可读：可以读取，查看目录内容

2.w代表可写：可以修改，对目录内创建+删除+重命名

3.x代表可执行：可以进入该目录

实际案例-

ls -l 现实的内容如下：

`-rwxrw-r--1 root root 1213 Feb 2 09:39 abc`

(10个字符确定不同用户能对文件干什么)

第一个字符代表文件类型：-l d c b

第一组rwx：文件拥有的权限是读，写，执行

第二组rw-：与文件所有者同一组的用户的权限是读，写但不能执行

第三组r--：不与拥有者同组的其他用户的权限是读，不能写与执行

可用数字表示为：r=4,w=2,x=1因此 $rw\bar{x}=4+2+1=7$

其他说明

1	文件：硬连接数或 目录：子目录数
root	用户
root	组
1213	文件大小（字节），如果是文件夹，显示4096字节
Feb 2 09:39	最后修改时间
abc	文件名

---

## 修改权限-chmod

通过chmod，可以修改文件或者目录的权限。

第一种方式：+,-,=

u:所有者 g:所有组 o:其他人 a:所有人（u, g, o的总和）

1.chmod u=rwx,g=rx,o=x 文件名/目录名

2.chmod o+w 文件/目录名

3.chmod a-x 文件/目录名

案例

1.给abc文件的所有者读写执行的权限，给所在组读执行权限，给其他组读执行权限

chmod u=rwx,g=rw,o=rw abc

2.给abc文件的所有者除去执行权限，增加组写的权限

chmod u-x,g+w abc

3.给abc文件的所有用户添加读的权限

chmod a+r abc

第二种方式：通过数字改变权限

可用数字表示为：r=4,w=2,x=1因此 $rw\bar{x}=4+2+1=7$

chmod u=rwx,g=rx,o=x 文件名/目录名

相当于：chmod 751 abc

案例：将/home/abc.txt 文件的权限修改成 rwxr-xr-x

chmod 755 /home/abc.txt

---

## 修改文件/目录所有者-chown

基本介绍

chown newowner 文件/目录 改变所有者

chown newowner: newgroup 文件/目录 改变所有者和所在组

-R 如果是目录 则使其下所有子文件或目录递归生效

案例演示

1.将 /home/abc.txt文件的所有者修改成tom

chown tom /home/abc.txt

2.将 /home/kkk目录下所有的文件和目录的所有者改成tom

chown -R /home/abc.txt

---

## 修改文件/目录所在组

chgrp newgroup 文件/目录 改变所有组

演示案例

1.将 /home/abc.tx文件所在组改成yuasnhen

groupadd yuasnhen

chgrp yuasnhen /home/abc.tx

2.将 /home/kkk目录下所有的文件和目录的所有组改成yua

chgrp -R yuasnhen /home/abc.tx

---

p(48-51 关于组管理和权限管理的练习略)

---

## 定时任务的调度 (p52)

### crontab 进行 定时任务的设置

概述：任务调度：是指系统在某个时间执行的特定的命令或程序

任务调度分类：1.系统工作：有些重要的工作必须周而复始地进行（如病毒扫描等）

2.个别用户工作：个别用户可能希望执行某些程序，比如对mysql数据库备份

基本语法： crontab [tab]

-e 编辑crontab定时任务

-l 查询crontab任务

-r 删除当前用户所有的crontab任务

快速入门：

1.设置任务调度文件： /etc/crontab

2.设置个人调度：执行 `crontab -e`

3.接着输入任务调度文件

如：`* / * * * * ls -l /etc/ > /tmp/to.txt` (每小时的每分钟执行ls -l /etc/ > /tmp/to.txt)

4.上方5个占位符的说明

项目	含义	范围
第一个 “*”	一小时当中的第几分钟	0-59
第二个 “*”	一天当中的第几小时	0-23
第三个 “*”	一个月当中的第几天	1-31
第四个 “*”	一年当中的第几月	1-12
第五个 “*”	一周当中的星期几	0-7 ( 0和7都代表星期日 )

特殊符号	含义
*	代表任何时间。比如第一个 “*” 就代表一小时中每分钟都执行一次的意思。
,	代表不连续的时间。比如 “0 8,12,16 * * * 命令”，就代表在每天的8点0分，12点0分，16点0分都执行一次命令
-	代表连续的时间范围。比如 “0 5 * * 1-6命令”，代表在周一到周六的凌晨5点0分执行命令
*/n	代表每隔多久执行一次。比如 “*/10 * * * * 命令”，代表每隔10分钟就执行一遍命令

时间	含义
45 22 * * * 命令	在22点45分执行命令
0 17 * * 1 命令	每周1 的17点0分执行命令
0 5 1,15 * * 命令	每月1号和15号的凌晨5点0分执行命令
40 4 * * 1-5 命令	每周一到周五的凌晨4点40分执行命令
*/10 4 * * * 命令	每天的凌晨4点，每隔10分钟执行一次命令
0 0 1,15 * 1 命令	每月1号和15号，每周1的0点0分都会执行命令。注意：星期几和几号最好不要同时出现，因为他们定义的都是天。非常容易让管理员混乱。

应用实例：

案例1：每隔1分钟，就当前的日期信息，追加到 /tmp/mydate 文件中

```
* / 1 * * * * date >> /tmp/mydate
```

案例2：每隔1分钟，将当前日期和日历都追加到 /home/mycal文件中

1.vim /home/my.sh 写入内容： `date >> /home/mycal`和`cal >> /home/mycal`

2.给my.sh 增加执行权限， `chmod u+x /home/my.sh`

3. `crontab -e */1 * * * * date >> /home/my.sh`

案例3：每天凌晨2：00将mysql数据库testdb,备份到文件中。提示：指令为：`mysqldump -u root -p密码 数据库 >> /home/db.bak`

1.crontab -e

2. `0 2 * * * mysqldump -u root -proot testbd > /home/db.bak`

crond相关指令



crontab -r :

crontab -l:

service crond restart [重启任务调度]

## at 定时任务

### 基本介绍

- 1.at命令是一次性定时计划任务，at的守护进程atd会以后台模式运行，检查作业队列来运行
- 2.默认情况下，atd守护进程每60S检查作业队列，有作业时，会检查作业运行时间，如果时间与当前时间匹配，则运行。
- 3.at命令是一次性定时计划任务，执行完一个任务后不再执行此任务了
- 4.在使用at命令的时候，一定要保证atd进程的启动，可以使用相关指令来查看

ps -ef | grep atd            (ps -ef : 查看所有正在运行的程序    |grep atd : 过滤出关于atd的程序)

### 基本语法

at [选项] [时间]

Ctrl +D 结束at命令的输入

选项	含义
-m	当指定的任务被完成后，将给用户发送邮件，即使没有标准输出
-l	atq的别名
-d	atrm的别名
-v	显示任务将被执行的时间
-c	打印任务的内容到标准输出
-V	显示版本信息
-q <队列>	使用指定的队列
-f <文件>	从指定文件读入任务而不是从标准输入读入
-t <时间参数>	以时间参数的形式提交要运行的任务

### ● at时间定义

at指定时间的方法：

1. 接受在当天的hh:mm（小时:分钟）式的时间指定。假如该时间已过去，那么就放在第二天执行。例如：04:00
2. 使用midnight（深夜），noon（中午），teatime（饮茶时间，一般是下午4点）等比较模糊的词语来指定时间。
3. 采用12小时计时制，即在时间后面加上AM（上午）或PM（下午）来说明是上午还是下午。例如：12pm
4. 指定命令执行的具体日期，指定格式为month day（月 日）或mm/dd/yy（月/日/年）或dd.mm.yy（日.月.年），指定的日期必须跟在指定时间的后面。例如：04:00 2021-03-1
5. 使用相对计时法。指定格式为：now + count time-units，now就是当前时间，time-units是时间单位，这里能够是minutes（分钟）、hours（小时）、days（天）、weeks（星期）。count是时间的数量，几天，几小时。例如：now + 5 minutes
6. 直接使用today（今天）、tomorrow（明天）来指定完成命令的时间。

### 应用实例

1.2天后的下午5点执行 /bin/lis /home

at 5pm + 2 days

/bin/ls /home (输入完之后 按两次 Ctrl +D 结束at命令的输入)

```
[root@hspEdu01 ~]# at 5pm + 2 days
at> /bin/ls /home<EOT>
job 5 at Sat Nov 7 17:00:00 2020
```

2.atq命令来查看系统中没有执行的工作任务

atq

3.明天17点中, 输出时间到指定文件内 比如 /root/date100.log

at 5pm tomorrow

date > /root/date100.log(输入完之后 按两次 Ctrl +D 结束at命令的输入)

4.2分钟后, 输出时间到指定文件内, 比如 /root/date200.log

at now + 2 mintues

date > /root/date200.log(输入完之后 按两次 Ctrl +D 结束at命令的输入)

5.删除已经设置的任务, atrm编号

atq 查看等待执行的任务及其编号

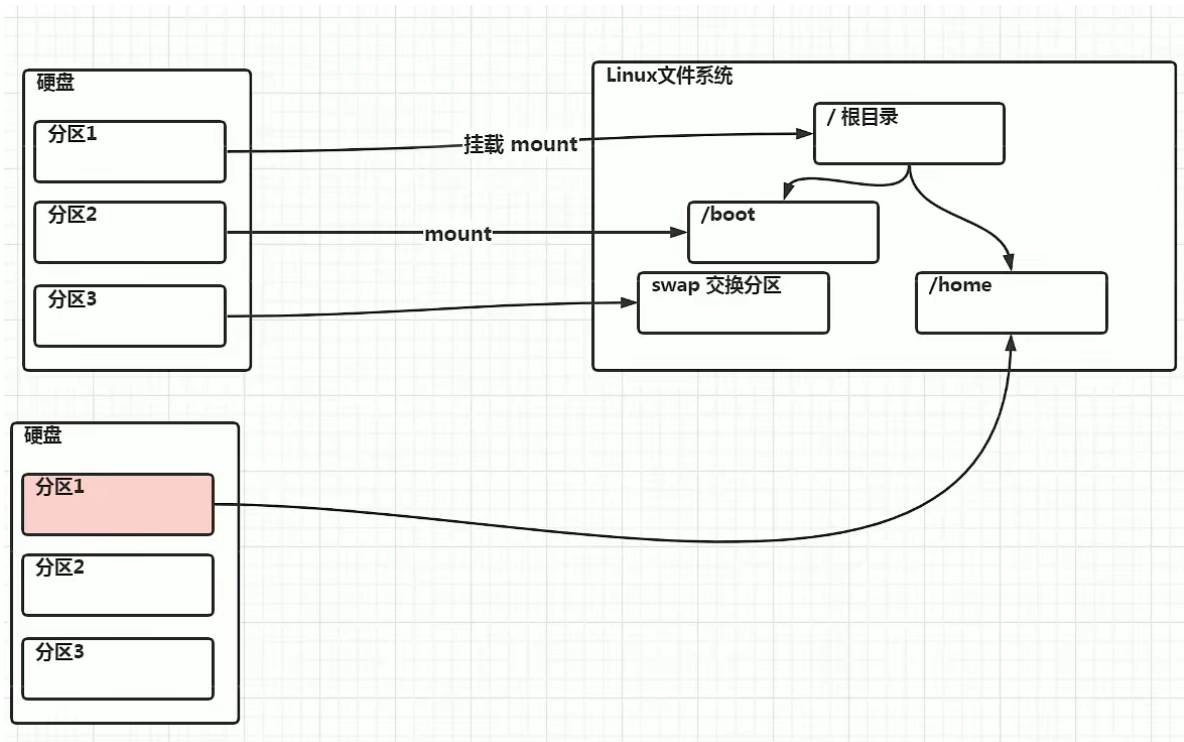
atrm [任务编号]

## Linux磁盘分区, 挂载 (p58)

基本原理

1.Linux来说, 无论有几个分区, 分给哪一个目录使用, 它归根结底就只有一个根目录, 一个独立且唯一的文件结构, Linux中每个分区都是用来组成整个文件系统的一部分

2.Linux采用了一种叫“载入”的处理方式, 它的整个文件系统包含了一整套的文件和目录, 且将一个分区和一个目录联系起来。这时要载入的一个分区将使它的存储空间在一个目录下获得。



3.查看设备挂载情况: lsblk or lsblk -f

```
[root@hspEdu01 /]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda          8:0    0   20G  0 disk
├─sda1       8:1    0    1G  0 part /boot
├─sda2       8:2    0    2G  0 part [SWAP]
└─sda3       8:3    0   17G  0 part /
sr0         11:0    1   4.3G  0 rom
```

```
sr0         11:0    1   4.3G  0 rom
[root@hspEdu01 /]# lsblk -f
NAME        FSTYPE LABEL                UUID                MOUNTPOINT
sda          分区情况
├─sda1 ext4                df4d64bd-981e-41eb-8f73-16ac20f0371e /boot
├─sda2 swap                494c7f94-5656-45a3-b990-008c2b0e39b7 [SWAP]
└─sda3 ext4                12ae1cc1-09e0-42ce-b659-063df3e7c941 /
sr0      iso9660 CentOS 7 x86_64 2018-11-25-23-54-16-00
```

*分区情况* (handwritten red text)

*分区唯一性* (handwritten red text with arrow pointing to UUID)

**MOUNTPOINT** (circled in red)