|  |  |
| --- | --- |
| 弯道和树的图片  Liunx 实验笔记  日常使用linux大全 | 摘要  涵盖了日常使用linux可能遇到的各种问题  yongjian zhang  工具书 |

作者：张永健

目录

[Linux常用的拓展库 4](#_Toc21507794)

[安装方法 4](#_Toc21507795)

[Linux常见设置 5](#_Toc21507796)

[开机自启脚本 5](#_Toc21507797)

[Linux设置文件权限 5](#_Toc21507798)

[Linux访问windows主机文件 6](#_Toc21507799)

[设置时间 6](#_Toc21507800)

[设置网络 6](#_Toc21507801)

[添加环境变量 7](#_Toc21507802)

[编译需要安装的库 7](#_Toc21507803)

[临时和永久关闭Selinux 8](#_Toc21507804)

[Centos、redhat yum本地源的设置 8](#_Toc21507805)

[sudo设置方法： 9](#_Toc21507806)

[lvm硬盘拓展 10](#_Toc21507807)

[redhat防火墙iptables设置 12](#_Toc21507808)

[Linux常用命令 13](#_Toc21507809)

[tar解压缩命令 13](#_Toc21507810)

[zip压缩文件 15](#_Toc21507811)

[fdisk 17](#_Toc21507812)

[Mount挂载 18](#_Toc21507813)

[umount取消挂载 18](#_Toc21507814)

[top 18](#_Toc21507815)

[ps 18](#_Toc21507816)

[cp 18](#_Toc21507817)

[rpm安装 19](#_Toc21507818)

[Systemctl 19](#_Toc21507819)

[netstat 19](#_Toc21507820)

[find查找文件 19](#_Toc21507821)

[uptime 19](#_Toc21507822)

[init 20](#_Toc21507823)

[vi编辑 21](#_Toc21507824)

[df 21](#_Toc21507825)

[du 22](#_Toc21507826)

[wc 22](#_Toc21507827)

[more 22](#_Toc21507828)

[iconv 22](#_Toc21507829)

[passwd 23](#_Toc21507830)

[date 23](#_Toc21507831)

[awk 23](#_Toc21507832)

[chown 23](#_Toc21507833)

[chmod 24](#_Toc21507834)

[Yum 24](#_Toc21507835)

[Linux常见的运维方法 25](#_Toc21507836)

[Linux查看端口或网络 25](#_Toc21507837)

[查看域里面有哪些用户 25](#_Toc21507838)

[批量杀死进程 25](#_Toc21507839)

[怎样更改Linux中默认的openjdk为自己安装的JDK 25](#_Toc21507840)

[Linux 安装软件 26](#_Toc21507841)

[一、 安装nginx 26](#_Toc21507842)

[Linux资源查看 29](#_Toc21507843)

[Linux系统资源查看 29](#_Toc21507844)

[查看历史上机，下机记录 29](#_Toc21507845)

[常用系统日志资源查看 29](#_Toc21507846)

[查看网络信息 32](#_Toc21507847)

[批量删除大文件 34](#_Toc21507848)

[Linux 解释 35](#_Toc21507849)

[Linux开机过程 35](#_Toc21507850)

[Linux疑难杂症 35](#_Toc21507851)

[关于linux动态库 动态加载库 35](#_Toc21507852)

[需要注意的是 36](#_Toc21507853)

# **Linux常用的拓展库**

**安装方法**

yum -y install epel-release

# **Linux常见设置**

## 开机自启脚本

通常情况下该脚本位于

/etc/rc.d/rc.local（需要添加执行权限）

## 设置selinux开机自动关闭

假设selinux是正在运行的，我们可以使用setenforce 命令设置临时关闭，不用重启生效。

setenforce 0

我们还可以通过编辑文件，然后重启让它关闭生效。

vi /etc/sysconfig/selinux

把里边的一行改为  
SELINUX=disabled  
改了之后保存，然后重启就可以了。

## Linux设置文件权限

chown -R 用户名:用户所在的组 文件或目录

chmod -R ugo=rwx 文件或目录

ugo 表示为 用户 组 其他用户

chmod ug=wrx,o=wrx xiaojian.zip

## Linux访问windows主机文件

挂载共享文件

mount -t cifs -o username=frank,password=xiaojian5791041,vers=2.0,dir\_mode=0777,file\_mode=0777 //192.168.1.105/linux\_sys /mnt/windows

## 设置时间

ls时间格式调整；

export TIME\_STYLE='+%Y/%m/%d %H:%M:%S'

## 设置网络

1.用root账号登录

2.打开/etc/sysconfig/network-scrpts/目录

3.vi ifcfg-eth0 （centos7系统为ifcfg-eno16777736）

4.按键盘"i"键进入编辑模式，将ONBOOT改为yes

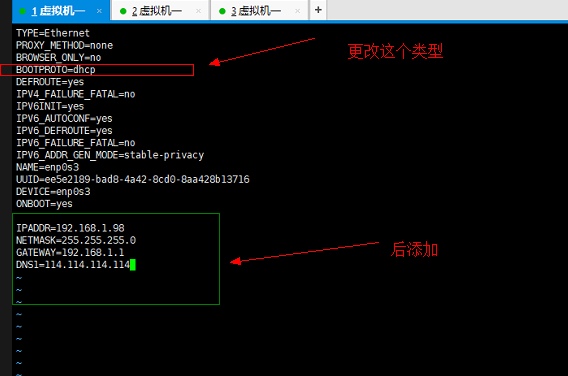
5.按下键盘ESC键之后输入:wq保存退出

重启网卡 网卡服务名是network

|  |  |
| --- | --- |
| 1 | service network restart |
|  |  |
| 2 | Systemctl stop network |

可以看到，小编这里使用的是dhcp自动获取IP地址（因为BOOTPROTO的参数为dhcp）

BOOTPROTO接受3个参数：none（禁止dhcp），static(静态IP)，dhcp（自动获取）

需要dhcp时只需更改 BOOTPROTO为dhcp删掉蓝色方框的内容重启服务即可

## 添加环境变量

添加 环境变量

修改profile文件：

#vi /etc/profile

在里面加入:

export PATH="$PATH:/opt/au1200\_rm/build\_tools/bin"

让环境变量立即生效需要执行如下命令：

#source /etc/profile

经过反复试验，得出 path路径在前面的生效，而不是覆盖，排在后面优先级最低

## 编译需要安装的库

有条件可以联网的话最好先安装上编译器

yum install -y gcc glibc

## 临时和永久关闭Selinux

临时关闭：

[root@localhost ~]# getenforce

Enforcing

[root@localhost ~]# setenforce 0

[root@localhost ~]# getenforce

Permissive

永久关闭：

[root@localhost ~]# vim /etc/sysconfig/selinux

SELINUX=enforcing 改为 SELINUX=disabled

重启电脑reboot

## Yum源介绍：

1. yum源三大命令：

清除缓存

yum clean all

生成缓存

yum makecache

检查yum源是否正常

yum repolist

1. 创建新的yum源文件
2. 创建本地yum源镜像文件模板

目录下/etc/yum.repos.d/ local.repo

[rhel-bendi]

name=bendi

baseurl=file:///mnt

gpgcheck=1

gpgkey=file:///mnt/RPM-GPG-KEY-redhat-release

1. 创建网络yum源

目录下/etc/yum.repos.d/ local.repo

[local\_docker]

name=loacl docker

baseurl=http://192.168.1.7/rpm\_package/

gpgcheck=0

enabled=1

1. 创建私有yum源（网络）

(1)、设置本地缓存rpm包

1.具体先安装createrepo工具

yum -y install createrepo -y

2.使用createrepo 生成缓存列表文件

其中rpm包在/root/RPM路径下

createrepo -o /root/RPM -v ./

yum.rpo里面设置 的路径写-o 的参数即可

当需要更新rpm包时，使用update参数

需要注意的是需要切换到repodata上级目录

createrepo --update ./Packages/

(2).配置nginx或本地添加访问即可

挂载镜像包：

mount -t iso9660 -o,loop /root/rhel-server-6.8-x86\_64-dvd.iso /mnt

创建本地源时需执行该命令挂载iso文件

vi /etc/rc.local

mount -t iso9660 -o,loop /root/rhel-server-6.8-x86\_64-dvd.iso /mnt

四、yum的常用命令

（1）检查已安装的包

yum list installed |grep libaio

五、yum的注意事项：

（1）只选择从一个yum源安装

yum install XXX --enablerepo=YYY

（2）如果是redhat 需要关闭redhat 订阅功能

vi /etc/yum/pluginconf.d/subscription-manager.conf

[main]

enabled=0

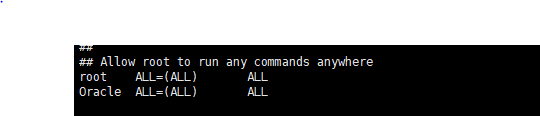
## sudo设置方法：

一、找到 /etc/sudoers文件

二、添加写权限

chmod -R +w /etc/sudoers

三、把目标用户加入文件中



四、收回写权限

chmod u-w /etc/sudoers

## lvm硬盘拓展

查看驱动器：lsblk

**第一步：创建pv**

创建 lvm pv pvcreate /dev/sdb /dev/sdc

查看物理卷信息 lvm

Pvs 、pvdisplay

**第二步：创建vg容器**

创建vg容器加入硬盘

vgcreate xiaojian /dev/sdb /dev/sdc

vgdisplay

查看vg详细列表

Vgs

简单查看详细列表

**第三步：创建逻辑使用空间**

创建逻辑使用空间

lvcreate -n mylv -L 10G xiaojian

详细信息

Lvdisplay

简单信息

Lvs

**第四步：格式化并挂载投入使用**

格式化信息

mkfs.ext4 /dev/xiaojian/mylv

注意如果不设置开机自动挂起 将不会自动挂载文件系统

挂载使用

mount /dev/xiaojian/mylv /10G

常见操作：

删除逻辑卷

lvremove /dev/xiaojian/ceshi

删除容器卷

vgremove

删除物理卷

pvremove

删除篇

首先先卸载驱动器

Umount /10G

先删除逻辑驱动器

lvremove /dev/xiaojian/mylv

查看是否还有

Lvs

删除容器

vgremove xiaojian

查看容器信息

vgs

去掉挂载pv的两个硬盘

fdisk -l

pvremove /dev/sdb /dev/sdc

查看信息

pvs

**扩容篇：**

当容器空间剩余时

首先先保证vg（容器里有剩余的空间vgdisplay）

Lvextend -L +1G /dev/xiaojian/ceshi 后面跟着的是lv（逻辑驱动器）

Resize2fs /dev/xiaojian/ceshi 更新文件系统

刷新容量

xfs\_growfs /dev/mapper/centos\_bogon-root

创建pv块

pvcreate /dev/sdb /dev/sdc

查看pv块信息

pvs

扩容vg容器纳入硬盘

vgextend centos\_bogon /dev/sdb /dev/sdc

查看扩容后的信息

vgs

查看lvs驱动器信息

Lvs

扩容lv（硬盘驱动器大小）

lvextend -L +40G /dev/centos\_bogon/root

刷新硬盘容量

resize2fs /dev/centos\_bogon/root

如果不行：

xfs\_growfs /dev/mapper/centos\_bogon-root

请参考：https://www.cnblogs.com/kevingrace/p/5825963.html

查看硬盘信息

df -h

查看文件系统

xfs\_info /dev/centos\_bogon/root

## redhat防火墙iptables设置

关于redhat6.8防火墙 iptables设置

首先是配置文件在

/etc/sysconfig/iptables

可以选择直接编辑该文件或者使用命令行（建议）

先打开iptables服务

这里举一个封杀特定端口的案例

这里是允许所有连接

iptables -P INPUT ACCEPT

iptables -P OUTPUT ACCEPT

iptables -P FORWARD ACCEPT

iptables -A INPUT -p tcp --dport 80 -j DROP

iptables -A INPUT -p udp --dport 80 -j DROP

iptables -A OUTPUT -p tcp --dport 80 -j DROP

iptables -A OUTPUT -p udp --dport 80 -j DROP

iptables -A FORWARD -p tcp --dport 80 -j DROP

iptables -A FORWARD -p udp --dport 80 -j DROP

封杀特定的端口。

iptables -L -n

保存配置文件

/etc/init.d/iptables save

或者service iptables save(有的时候不灵)

手动查看是否成功。

设置hostname

hostnamectl set-hostname 服务名称

## ssh无密钥配置

（1）免密登录原理，如图2-40所示

图2-40 免密登陆原理

（2）生成公钥和私钥：

[atguigu@hadoop102 .ssh]$ ssh-keygen -t rsa

然后敲（三个回车），就会生成两个文件id\_rsa（私钥）、id\_rsa.pub（公钥）

（3）将公钥拷贝到要免密登录的目标机器上

[atguigu@hadoop102 .ssh]$ ssh-copy-id hadoop102

[atguigu@hadoop102 .ssh]$ ssh-copy-id hadoop103

[atguigu@hadoop102 .ssh]$ ssh-copy-id hadoop104

## 手动清除缓存内存

echo 3 > /proc/sys/vm/drop\_caches

## 修改service文件后需要重新加载service

systemctl daemon-reload

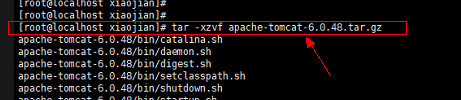
# Linux常用命令

## tar解压缩命令

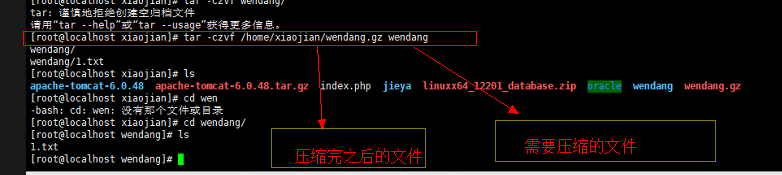
解压到指定目录



解压文件到本目录



压缩文件命令：



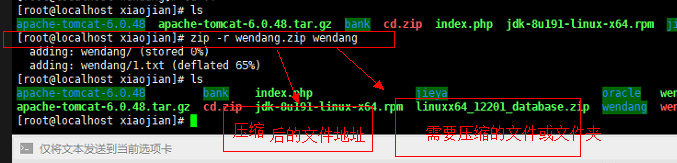
tar命令  
　　1. 作用 tar命令是Unix/Linux系统中备份文件的可靠方法，几乎可以工作于任何环境中，它的使用权限是所有用户。

tar -cvf jpg.tar \*.jpg jpg.tar 是压缩完之后的文件   
　　2. 格式 tar [主选项+辅选项] 文件或目录  eg:  tar zxvf libiconv-1.13.tar.gz ./libiconv    tar cxvf libiconv-1.13.tar.gz ./\*.php  
　　3. 主要参数  
　　使用该命令时，主选项是必须要有的，它告诉tar要做什么事情，辅选项是辅助使用的，可以选用。  
　　3.1 主选项：

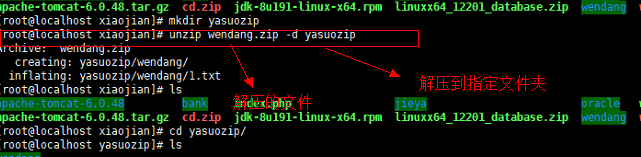
-----------------------------------------------------------------------------------------------  
　　-c 创建新的档案文件。如果用户想备份一个目录或是一些文件，就要选择这个选项。  
　　-r 把要存档的文件追加到档案文件的未尾。例如用户已经做好备份文件，又发现还有一个目录或是一些文件忘记备份了，这时可以使用该选项，将忘记的目录或文件追加到备份文件中。  
　　-t 列出档案文件的内容，查看已经备份了哪些文件。  
　　-u 更新文件。就是说，用新增的文件取代原备份文件，如果在备份文件中找不到要更新的文件，则把它追加到备份文件的最后。  
　　-x 从档案文件中释放文件。  
　　3.2 辅助选项：  
　　-b 该选项是为磁带机设定的，其后跟一数字，用来说明区块的大小，系统预设值为20（20×512 bytes）。  
　　-f 指定档案文件名或设备名，这个选项通常是必选的。  
　　-k 保存已经存在的文件。例如把某个文件还原，在还原的过程中遇到相同的文件，不会进行覆盖。  
　　-m 在还原文件时，把所有文件的修改时间设定为现在。  
　　-M 创建多卷的档案文件，以便在几个磁盘中存放。  
　　-v 详细报告tar处理的文件信息。如无此选项，tar不报告文件信息。  
　　-w 每一步都要求确认。  
　　-z 用gzip来压缩/解压缩文件，加上该选项后可以将档案文件进行压缩，但还原时也一定要使用该选项进行解压缩。

## zip压缩文件

Zip压缩



解压



语法

zip [-AcdDfFghjJKlLmoqrSTuvVwXyz$][-b <工作目录>][-ll][-n <字尾字符串>][-t <日期时间>][-<压缩效率>][压缩文件][文件...][-i <范本样式>][-x <范本样式>]

参数：

-A 调整可执行的自动解压缩文件。

-b<工作目录> 指定暂时存放文件的目录。

-c 替每个被压缩的文件加上注释。

-d 从压缩文件内删除指定的文件。

-D 压缩文件内不建立目录名称。

-f 此参数的效果和指定"-u"参数类似，但不仅更新既有文件，如果某些文件原本不存在于压缩文件内，使用本参数会一并将其加入压缩文件中。

-F 尝试修复已损坏的压缩文件。

-g 将文件压缩后附加在既有的压缩文件之后，而非另行建立新的压缩文件。

-h 在线帮助。

-i<范本样式> 只压缩符合条件的文件。

-j 只保存文件名称及其内容，而不存放任何目录名称。

-J 删除压缩文件前面不必要的数据。

-k 使用MS-DOS兼容格式的文件名称。

-l 压缩文件时，把LF字符置换成LF+CR字符。

-ll 压缩文件时，把LF+CR字符置换成LF字符。

-L 显示版权信息。

-m 将文件压缩并加入压缩文件后，删除原始文件，即把文件移到压缩文件中。

-n<字尾字符串> 不压缩具有特定字尾字符串的文件。

-o 以压缩文件内拥有最新更改时间的文件为准，将压缩文件的更改时间设成和该文件相同。

-q 不显示指令执行过程。

-r 递归处理，将指定目录下的所有文件和子目录一并处理。

-S 包含系统和隐藏文件。

-t<日期时间> 把压缩文件的日期设成指定的日期。

-T 检查备份文件内的每个文件是否正确无误。

-u 更换较新的文件到压缩文件内。

-v 显示指令执行过程或显示版本信息。

-V 保存VMS操作系统的文件属性。

-w 在文件名称里假如版本编号，本参数仅在VMS操作系统下有效。

-x<范本样式> 压缩时排除符合条件的文件。

-X 不保存额外的文件属性。

-y 直接保存符号连接，而非该连接所指向的文件，本参数仅在UNIX之类的系统下有效。

-z 替压缩文件加上注释。

-$ 保存第一个被压缩文件所在磁盘的卷册名称。

-<压缩效率> 压缩效率是一个介于1-9的数值。

unzip语 法：   
  
    unzip [-cflptuvz][-agCjLMnoqsVX][-P <密码>][.zip文件][文件][-d <目 录>][-x <文件>] 或 unzip [-Z]   
  
    补充说明：unzip为.zip压缩文件的解压缩程序。

unzip参 数：

    -c 将解压缩的结果显示到屏幕上，并对字符做适当的转换。

    -f 更新现有的文件。

    -l 显示压缩文件内所包含的文件。

    -p 与-c参数类似，会将解压缩的结果显示到屏幕上，但不会执行任何的转换。

    -t 检查压缩文件是否正确。

    -u 与-f参数类似，但是除了更新现有的文件外，也会将压缩文件中的其他文件解压缩到目录中。

    -v 执行是时显示详细的信息。

    -z 仅显示压缩文件的备注文字。

    -a 对文本文件进行必要的字符转换。

    -b 不要对文本文件进行字符转换。

    -C 压缩文件中的文件名称区分大小写。

    -j 不处理压缩文件中原有的目录路径。

    -L 将压缩文件中的全部文件名改为小写。

    -M 将输出结果送到more程序处理。

    -n 解压缩时不要覆盖原有的文件。

    -o 不必先询问用户，unzip执行后覆盖原有文件。

    -P<密码> 使用zip的密码选项。

    -q 执行时不显示任何信息。

    -s 将文件名中的空白字符转换为底线字符。

    -V 保留VMS的文件版本信息。

    -X 解压缩时同时回存文件原来的UID/GID。

    [.zip文件] 指定.zip压缩文件。

    [文件] 指定要处理.zip压缩文件中的哪些文件。

    -d<目录> 指定文件解压缩后所要存储的目录。

    -x<文件> 指定不要处理.zip压缩文件中的哪些文件。

    -Z unzip -Z等于执行zipinfo指令

## fdisk

fdisk -l 查看有哪些驱动盘

## Mount挂载

挂载命令：

mount -t iso9660 -o,loop /media/CentOS-7-x86\_64-DVD-1804.iso /mnt/Centos7

挂载ntfs u盘

mount -t ntfs /dev/mmcblk0p1 /mnt/xiaojian

对于fat32格式的u盘或驱动器

mount /dev/sdb4 guazai

## umount取消挂载

取消挂载

umount /dev/sdb4

## top

资源监视器

按大写C M 排序

top -u frank 监控frank用户的程序使用情况

## ps

常用语法：

ps -ef | grep mysql 查看有关于mysql的服务器进程命令

## cp

递归复制文件到目标目录

cp -r /home/frank/guazai/ /home/frank/wenjian/

## rpm安装

Rpm -ivh 包文件

Rpm -qa 查看所有安装包

Rpm -ql 产看安装路径

## Systemctl

查询安装的服务

Systemctl list-unit-files | grep nginx

## netstat

查看服务启动的端口等等

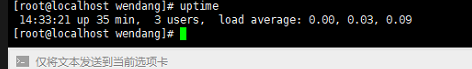
netstat -plnt |grep sshd

## find查找文件

find / -name xiaojian\*

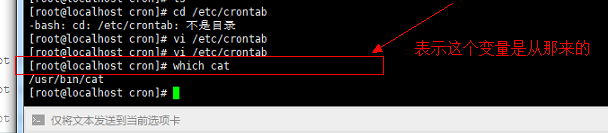
## uptime

查看系统负载（就是运行了多长时间，多少连接，）



which

查看命令的程序在哪。



## init

Init(谨慎使用)(突发情况)

Init6 杀死所有程序立即关机（等于强行关机）

# 0 - 停机（千万不能把initdefault 设置为0 ）

　　# 1 - 单用户模式

　　# 2 - 多用户，没有 NFS

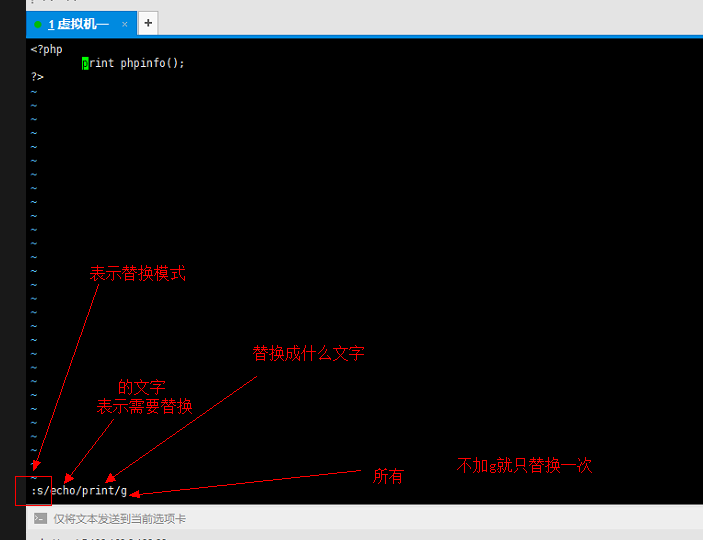
　　# 3 - 完全多用户模式(标准的运行级)

　　# 4 - 没有用到

　　# 5 - X11 （xwindow)

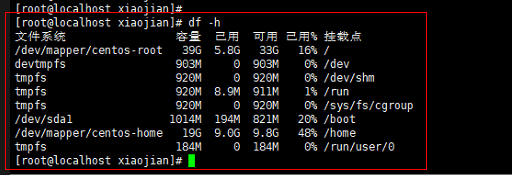
# 6 - 重新启动 （千万不要把initdefault 设置为6 ）

## vi编辑



## df

df - h 查看文件分区等



df [选项]... [FILE]...

文件-a, --all 包含所有的具有 0 Blocks 的文件系统

文件--block-size={SIZE} 使用 {SIZE} 大小的 Blocks

文件-h, --human-readable 使用人类可读的格式(预设值是不加这个选项的...)

文件-H, --si 很像 -h, 但是用 1000 为单位而不是用 1024

文件-i, --inodes 列出 inode 资讯，不列出已使用 block

文件-k, --kilobytes 就像是 --block-size=1024

文件-l, --local 限制列出的文件结构

文件-m, --megabytes 就像 --block-size=1048576

文件--no-sync 取得资讯前不 sync (预设值)

文件-P, --portability 使用 POSIX 输出格式

文件--sync 在取得资讯前 sync

文件-t, --type=TYPE 限制列出文件系统的 TYPE

文件-T, --print-type 显示文件系统的形式

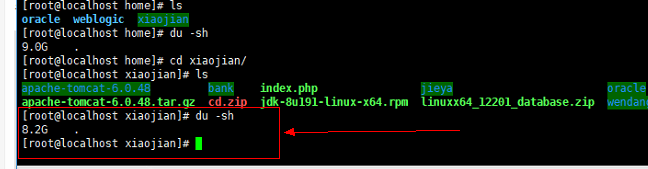
文件-x, --exclude-type=TYPE 限制列出文件系统不要显示 TYPE

文件-v (忽略)

文件--help 显示这个帮手并且离开

文件--version 输出版本资讯并且离开

## du



快速查看（仅查看一行不遍历）

du -ah --max-depth=1

du -h \* 查看各个文件夹的大小

du -sh \* 不一一列出文件夹里面的内容

## wc

wc -l Xorg.0.log 查看该文件有多少行

## more

more +1665 Xorg.0.log 从多少行开始读取

## iconv

iconv -f utf8 -t GBK -o xiaojians.txt xiaojian.txt

## passwd

passwd -l lynn

这就话的意思是锁定lynn用户，这样该用户就不能登录了。

passwd -u lynn

对锁定的用户lynn进行解锁，用户可登录了。

## date

date "+%Y-%m-%d %H:%M:%S"

## awk

awk '{print $2}' 这个命令

参考地址：https://blog.csdn.net/luman1991/article/details/52756630

$2：表示第二个字段

print $2 ： 打印第二个字段

awk '{print $2}' $fileName : 一行一行的读取指定的文件， 以空格作为分隔符，打印第二个字段

比如有这样一个文件

a1 b1 c1 d1

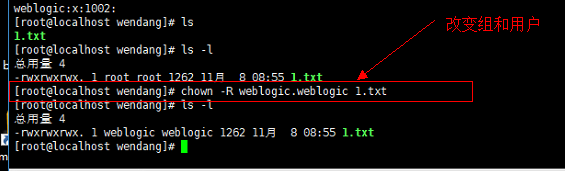
a2 b2 c2 d2

执行的结果是,输出

b1

b2

## chown



## chmod

Chmod -R

-rw------- (600) -- 只有属主有读写权限。

-rw-r--r-- (644) -- 只有属主有读写权限；而属组用户和其他用户只有读权限。

-rwx------ (700) -- 只有属主有读、写、执行权限。

-rwxr-xr-x (755) -- 属主有读、写、执行权限；而属组用户和其他用户只有读、执行权限。

-rwx--x--x (711) -- 属主有读、写、执行权限；而属组用户和其他用户只有执行权限。

-rw-rw-rw- (666) -- 所有用户都有文件读、写权限。这种做法不可取。

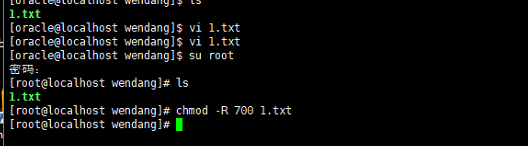
-rwxrwxrwx (777) -- 所有用户都有读、写、执行权限。更不可取的做法。

以下是对目录的两个普通设定:

drwx------ (700) - 只有属主可在目录中读、写。

drwxr-xr-x (755) - 所有用户可读该目录，但只有属主才能改变目录中的内容。

chmod -R u=rwx,g=rw,o=r xiaojian.txt



## Yum

检查已安装的包

yum list installed |grep libaio

## mkdir

创建文件夹

mkdir 路径

mkdir -p 路径 递归

# Linux常见的运维方法

## Linux查看端口或网络

nmap -v 192.168.2.110

ssh 192.168.2.110 -p 80 -v

nc -z 1172.28.5.31 10053

## 查看域里面有哪些用户

cat /etc/passwd |awk -F ':' '{ print $1":"$2":"$4":"$5":"$6 }' |grep 1001

## 批量杀死进程

ps -ef |grep httpd | awk '{ print $2 }'|xargs kill -9

ps -ef | grep test | grep -v grep | awk '{print $2}' | xargs kill -9

## [怎样更改Linux中默认的openjdk为自己安装的JDK](https://www.cnblogs.com/slgkaifa/p/7193730.html)

(1)  
/etc/profile  
export JAVA\_HOME=/usr/java/jdk1.7.0\_67-cloudera/  
export PATH=$PATH:$JAVA\_HOME/bin  
export JRE\_HOME="$JAVA\_HOME/jre"  
export CLASSPATH=.:$JAVA\_HOME/lib/tools.jar:$JAVA\_HOME/lib/rt.jar  
  
改动完毕后，使用source /etc/profile 命令进行更新；  
  
  
（2）以下这几步比較重要。由于在很多linux系统中都预装了openjdk，所以即使上面进行了这些设置，系统默认的还不是我们要设置的内容。假设要真正使用我们自己的jdk。还须要一下步骤：  
  
sudo update-alternatives --install /usr/bin/java java /usr/java/jdk1.7.0\_67-cloudera/bin/java 300  
  
sudo update-alternatives --install /usr/bin/javac javac /usr/java/jdk1.7.0\_67-cloudera/bin/javac 300  
  
(这里的300仅仅是标识一下。没什么意义，仅仅是下一步要用到)  
  
（3）大功告成，使用java -version 命令来查看我们的java环境变量，显演示样例如以下内容：  
  
java version "1.7.0\_05"  
Java(TM) SE Runtime Environment (build 1.7.0\_05-b06)

Java HotSpot(TM) Server VM (build 23.1-b03, mixed mode)

\* 对于更新一个 /usr/bin/xxx 命令指向一个新的路径能够用 sudo update-alternatives --config xxx

# Linux 常用工具使用

## Rsync远程同步

远程之前先建立ssh 信任

rsync -av /opt/software/ hadoop102:/opt/software

## Scp 远程传输工具

占位符

# Linux 中软件使用

## Filewalld 简介和使用

添加一条富规则

firewall-cmd --permanent --add-rich-rule="rule family="ipv4" source address=" 112.86.225.56" port protocol="tcp" port="443" accept"

删除一条富规则

firewall-cmd --permanent --remove-rich-rule="rule family="ipv4" source address=" 112.80.131.49" port port="443" protocol="tcp" accept"

#删除所有的富规则

firewall-cmd --list-rich-rules|xargs -I {} firewall-cmd --permanent --remove-rich-rule="{}"

查看所有的规则

firewall-cmd --list-all-zones

查看富规则

firewall-cmd --list-rich-rules

#重载

firewall-cmd --reload

## iptables 简介和使用

禁止端口

iptables -I INPUT -p tcp --dport 443 -j DROP

iptables -I INPUT -p udp --dport 443 -j DROP

##向后添加一条数据

iptables -I INPUT -s 153.3.90.185 -p tcp --dport 443 -j ACCEPT

iptables -I INPUT -s 153.3.90.185 -p udp --dport 443 -j ACCEPT

##查看条数

iptables -nL --line-number

# Linux 安装软件

## 安装nginx

官方安装地址：<https://unit.nginx.org/installation/>

参考地址：<https://www.cnblogs.com/herui1991/p/8996917.html>

依赖：

1.依赖环境介绍

gcc gcc-c++

gcc为GNU Compiler Collection的缩写，可以编译C和C++源代码等，它是GNU开发的C和C++以及其他很多种语言 的编译器（最早的时候只能编译C，后来很快进化成一个编译多种语言的集合，如Fortran、Pascal、Objective-C、Java、Ada、 Go等。）

gcc 在编译C++源代码的阶段，只能编译 C++ 源文件，而不能自动和 C++ 程序使用的库链接（编译过程分为编译、链接两个阶段，注意不要和可执行文件这个概念搞混，相对可执行文件来说有三个重要的概念：编译（compile）、链接（link）、加载（load）。源程序文件被编译成目标文件，多个目标文件连同库被链接成一个最终的可执行文件，可执行文件被加载到内存中运行）。因此，通常使用 g++ 命令来完成 C++ 程序的编译和连接，该程序会自动调用 gcc 实现编译。

gcc-c++也能编译C源代码，只不过把会把它当成C++源代码，后缀为.c的，gcc把它当作是C程序，而g++当作是c++程序；后缀为.cpp的，两者都会认为是c++程序，注意，虽然c++是c的超集，但是两者对语法的要求是有区别的。

make automake

make是一个用来控制可执行文件和其他一些从源文件来的非源代码文件版本的软件。Make可以从一个名为makefile的文件中获得如何构建你所写程序的依赖关系，Makefile中列出了每个目标文件以及如何由其他文件来生成它。

automake是一个从Makefile.am文件自动生成Makefile.in的工具。为了生成Makefile.in，automake还需用到perl，由于automake创建的发布完全遵循GNU标准，所以在创建中不需要perl。libtool是一款方便生成各种程序库的工具。

autoconf

autoconf是用来生成自动配置软件源代码脚本（configure）的工具

pcre pcre-devel

在Nginx编译需要 PCRE(Perl Compatible Regular Expression)，因为Nginx 的Rewrite模块和HTTP 核心模块会使用到PCRE正则表达式语法。

zlip zlib-devel

nginx启用压缩功能的时候，需要此模块的支持。

openssl openssl-devel

开启SSL的时候需要此模块的支持。

Libtool

libtool是一个通用库支持脚本，将使用动态库的复杂性隐藏在统一、可移植的接口中；使用libtool的标准方法，可以在不同平台上创建并调用动态库。

libtool主要的一个作用是在编译大型软件的过程中解决了库的依赖问题；将繁重的库依赖关系的维护工作承担下来，从而释放了程序员的人力资源。libtool提供统一的接口，隐藏了不同平台间库的名称的差异等细节，生成一个抽象的后缀名为la高层库libxx.la（其实是个文本文件），并将该库对其它库的依赖关系，都写在该la的文件中。

安装依赖包的命令：

yum -y install gcc gcc-c++ make automake autoconf pcre pcre-devel zlib zlib-devel openssl openssl-devel libtool

安装命令：

./configure --prefix=/usr/local/nginx --sbin-path=/usr/local/nginx/sbin/nginx --conf-path=/usr/local/nginx/conf/nginx.conf --error-log-path=/var/log/nginx/error.log --http-log-path=/var/log/nginx/access.log --pid-path=/var/run/nginx/nginx.pid --lock-path=/var/lock/nginx.lock --user=nginx --group=nginx --with-http\_ssl\_module --with-http\_stub\_status\_module --with-http\_gzip\_static\_module --http-client-body-temp-path=/var/tmp/nginx/client/ --http-proxy-temp-path=/var/tmp/nginx/proxy/ --http-fastcgi-temp-path=/var/tmp/nginx/fcgi/ --http-uwsgi-temp-path=/var/tmp/nginx/uwsgi --http-scgi-temp-path=/var/tmp/nginx/scgi --with-pcre

Make & make install

启动、停止Nginx服务

[root@herui ~]# /usr/local/nginx/sbin/nginx #启动Nginx

nginx: [emerg] getpwnam("nginx") failed #没有Nginx用户

[root@herui ~]# /usr/local/nginx/sbin/nginx -s stop #停止Nginx

[root@herui ~]# /usr/local/nginx/sbin/nginx -s reload #不停止服务，重新加载配置文件

验证Nginx是否开启成功

[root@herui ~]# ps aux|grep nginx

root 2218 0.0 0.1 47320 1188 ? Ss 10:21 0:00 nginx: master process /usr/local/nginx/sbin/nginx

nginx 2219 0.0 0.1 47756 1760 ? S 10:21 0:00 nginx: worker process

root 2222 0.0 0.0 103256 836 pts/0 S+ 10:21 0:00 grep nginx

#### nginx注意事项：

vim /usr/local/nginx/conf/nginx.conf

user nginx nginx; # 指定Nginx服务的用户和用户组

一般来说，这个用户是需要自己建的。测试一般写root即可

## 安装 openssl linux

1. 去官网下载最新版本，或wget下载也行 <http://www.openssl.org>
2. 解压并进入解压目录后执行：

./config --prefix=/usr/local/openssl shared zlib  
make depend  
make && make install

1. 备份当前openssl：

mv /usr/local/openssl /usr/local/openssl.bak

mv /usr/include/openssl /usr/include/openssl.bak

4、配置使用新版本：

ln -s /usr/local/openssl/bin/openssl /usr/bin/openssl

ln -s /usr/local/openssl/include/openssl /usr/include/openssl

5、更新动态链接库数据：（把liba添加到linux依赖动态链接中）

echo "/usr/local/openssl/lib" >> /etc/ld.so.conf

重新加载动态链接库  
ldconfig -v

6、重新查看版本号：

openssl version –a

如果在更新完后执行openssl version 命令报错如下：

https://img2018.cnblogs.com/blog/844003/201809/844003-20180925143700174-2047479030.png

这是由于openssl库的位置不正确造成的。  
可以做一个软连接：

ln -s /usr/local/openssl/lib/libssl.so.1.1 /usr/lib/

ln -s /usr/local/openssl/lib/libcrypto.so.1.1 /usr/lib/

https://img2018.cnblogs.com/blog/844003/201809/844003-20180925143905844-987417477.png

然后执行openssl version命令即可

参考地址：<https://www.cnblogs.com/caibao666/p/9698842.html>

# Linux资源查看

## Linux系统资源查看

### 查看历史上机，下机记录

查看关机

Last

### 常用系统日志资源查看

/var/log/secure 登录日志

/var/log/secure 登录日志

who /var/log/wtmp 历史登录日志

w 登录时间 登录

last -f /var/log/wtmp

查看系统内核版本

cat /proc/version

查看发行版本

cat /etc/redhat-release

cat /etc/issue

查看是多少位

getconf LONG\_BIT

查看基本信息

查看cpu是几核的

cat /proc/cpuinfo

查看内存信息

cat /proc/meminfo

查看路由网关

route -n

Uptime 用户的连接数,运行的时间

内存查看 free -bm

查看磁盘的使用情况

Df -h 直观的方式查看

Df -T 类型

查看文件或目录所占用的空间

Du -sh

查看最近登录的用户

Lastlo

查看这些用户是从哪个ip地址登录的

Who

查看登录信息

who /var/log/wtmp

查看端口信息

**netstat -plnt**

tail 命令可用于查看文件的内容，有一个常用的参数 **-f** 常用于查阅正在改变的日志文件。

**tail -f filename** 会把 filename 文件里的最尾部的内容显示在屏幕上，并且不断刷新，只要 filename 更新就可以看到最新的文件内容。

**日 志 文 件 说 明**

**/var/log/message 系统启动后的信息和错误日志，是Red Hat Linux中最常用的日志之一**

**/var/log/secure 与安全相关的日志信息**

**/var/log/maillog 与邮件相关的日志信息**

**/var/log/cron 与定时任务相关的日志信息**

**/var/log/spooler 与UUCP和news设备相关的日志信息**

**/var/log/boot.log 守护进程启动和停止相关的日志消息**

**系统：**

**# uname -a # 查看内核/操作系统/CPU信息**

**# cat /etc/issue**

**# cat /etc/redhat-release # 查看操作系统版本**

**# cat /proc/cpuinfo # 查看CPU信息**

**# hostname # 查看计算机名**

**# lspci -tv # 列出所有PCI设备**

**# lsusb -tv # 列出所有USB设备**

**# lsmod # 列出加载的内核模块**

**# env # 查看环境变量**

**资源：**

**# free -m # 查看内存使用量和交换区使用量**

**# df -h # 查看各分区使用情况**

**# du -sh <目录名> # 查看指定目录的大小**

**# grep MemTotal /proc/meminfo # 查看内存总量**

**# grep MemFree /proc/meminfo # 查看空闲内存量**

**# uptime # 查看系统运行时间、用户数、负载**

**# cat /proc/loadavg # 查看系统负载**

**磁盘和分区：**

**# mount | column -t # 查看挂接的分区状态**

**# fdisk -l # 查看所有分区**

**# swapon -s # 查看所有交换分区**

**# hdparm -i /dev/hda # 查看磁盘参数(仅适用于IDE设备)**

**# dmesg | grep IDE # 查看启动时IDE设备检测状况**

**网络：**

**# ifconfig # 查看所有网络接口的属性**

**# iptables -L # 查看防火墙设置**

**# route -n # 查看路由表**

**# netstat -lntp # 查看所有监听端口**

**# netstat -antp # 查看所有已经建立的连接**

**# netstat -s # 查看网络统计信息**

**进程：**

**# ps -ef # 查看所有进程**

**# top # 实时显示进程状态（另一篇文章里面有详细的介绍）**

**用户：**

**# w # 查看活动用户**

**# id <用户名> # 查看指定用户信息**

**# last # 查看用户登录日志**

**# cut -d: -f1 /etc/passwd # 查看系统所有用户**

**# cut -d: -f1 /etc/group # 查看系统所有组**

**# crontab -l # 查看当前用户的计划任务**

**服务：**

**# chkconfig –list # 列出所有系统服务**

**# chkconfig –list | grep on # 列出所有启动的系统服务**

**程序：**

**# rpm -qa # 查看所有安装的软件包**

## 查看网络信息

查看服务是否启动

ps -ef | grep sshd

查看服务是否启动端口

netstat -plnt | grep httpd

Netstat 命令用于显示各种网络相关信息，如网络连接，路由表，接口状态 (Interface Statistics)，masquerade 连接，多播成员 (Multicast Memberships) 等等。

firewall-cmd --zone=public --add-port=82/tcp --permanent

常见参数

-a (all)显示所有选项，默认不显示LISTEN相关

-t (tcp)仅显示tcp相关选项

-u (udp)仅显示udp相关选项

-n 拒绝显示别名，能显示数字的全部转化成数字。（可以直接显示端口名）

-l 仅列出有在 Listen (监听) 的服務状态

-p 显示建立相关链接的程序名

-r 显示路由信息，路由表

-e 显示扩展信息，例如uid等

-s 按各个协议进行统计

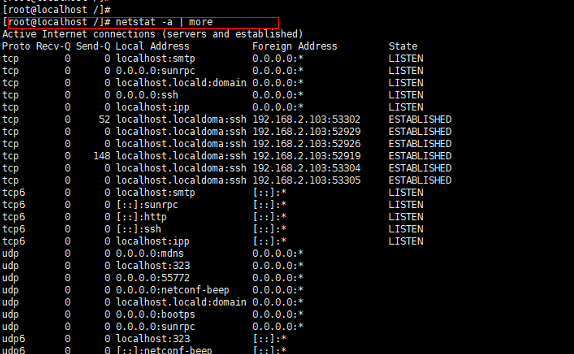
-c 每隔一个固定时间，执行该netstat命令。

提示：LISTEN和LISTENING的状态只有用-a或者-l才能看到

**列出所有端口**

Netstat -a | more

More用于隐藏更多信息，这里也可以直接Netstat -a



列出所有 tcp 端口 netstat -at

**列出所有 udp 端口 netstat -au**

**2. 列出所有处于监听状态的 Sockets**

**只显示监听端口 netstat -l**

**只列出所有监听 tcp 端口 netstat -lt**

**只列出所有监听 udp 端口 netstat -lu**

**只列出所有监听 UNIX 端口 netstat -lx**

**3. 显示每个协议的统计信息**

**显示所有端口的统计信息 netstat -s**

**显示 TCP 或 UDP 端口的统计信息 netstat -st 或 -su**

**4. 在 netstat 输出中显示 PID 和进程名称 netstat -p**

netstat -p 可以与其它开关一起使用，就可以添加 “PID/进程名称” 到 netstat 输出中，这样 debugging 的时候可以很方便的发现特定端口运行的程序。

**5. 在 netstat 输出中不显示主机，端口和用户名 (host, port or user)**

当你不想让主机，端口和用户名显示，使用 netstat -n。将会使用数字代替那些名称。

同样可以加速输出，因为不用进行比对查询。

# netstat -an

**6. 持续输出 netstat 信息**

netstat 将每隔一秒输出网络信息。

netstat -c

**7. 显示系统不支持的地址族 (Address Families)**

netstat --verbose

**9. 找出程序运行的端口**

并不是所有的进程都能找到，没有权限的会不显示，使用 root 权限查看所有的信息。

# netstat -ap | grep ssh  
 tcp 1 0 dev-db:ssh 101.174.100.22:39213 CLOSE\_WAIT -  
 tcp 1 0 dev-db:ssh 101.174.100.22:57643 CLOSE\_WAIT -

**找出运行在指定端口的进程**

# netstat -an | grep ':80'

**10. 显示网络接口列表**

# netstat -i

**11. IP和TCP分析**

**查看连接某服务端口最多的的IP地址**

[IMG_256](https://www.cnblogs.com/ggjucheng/archive/2012/01/08/javascript:void(0);)

wss8848@ubuntu:~$ netstat -nat | grep "192.168.1.15:22" |awk '{print $5}'|awk -F: '{print $1}'|sort|uniq -c|sort -nr|head -20

网址参考地址：https://www.cnblogs.com/ggjucheng/archive/2012/01/08/2316661.html

## 批量删除大文件

按照 文件大小 删除 列出前十条大文件

#显示前十条大文件

ls|grep log |xargs du -sm|sort -rn|head |awk '{print $2}'

# Linux 解释

## Linux开机过程

开机过程开机会先启动etc下目录的service文件 ，所有systemctl enable 实际上是复制一个快捷方式到etc目录下

/usr/lib/systemd/system

开启这个文件夹的

/usr/lib/systemd/system/multi-user.target.wants/

/etc/systemd/system

/etc/rc.d/init.d/

# Linux疑难杂症

## 关于linux动态库 动态加载库

vi /etc/ld.so.conf

/sbin/ldconfig -v

命令 /sbin/ldconf -v | grep "libaivse000" 查看是否系统有加载该库

查看动态库的地址

ldconfig -p

LD\_LIBRARY\_PATH 连接变量 = 需要加载.so的地址

export LD\_LIBRARY\_PATH=/opt/python/lib

然后重新加载 需要注意的时随意添加动态链接库可能会导致开机慢死机等。。。

## 需要注意的是

重要报错信息：

Yum是需要python2的如果将python2卸载或删除，yum将无法使用请注意。

## 账号密码都对，网络也没有限制，但怎么都登录不了Linux

正常情况下/var/log/tallylog权限：

[root@xuehen~]# ls -ald /var/log/tallylog

-rw- - - - - - -. 1 root root 23808 Nov 20 13:28 /var/log/tallylog

当权限为非此权限如下：

[root@xuehen~]# ls -ald /var/log/tallylog

-rwxrwxrwx. 1 root root 23808 Nov 20 13:30 /var/log/tallylog

报错信息：

pam\_tally2:/var/log/tallylog is either world wtitable or not a normal file

pam\_tally2:Authentication error

解决思路：

chmod 600 /var/log/tallylog