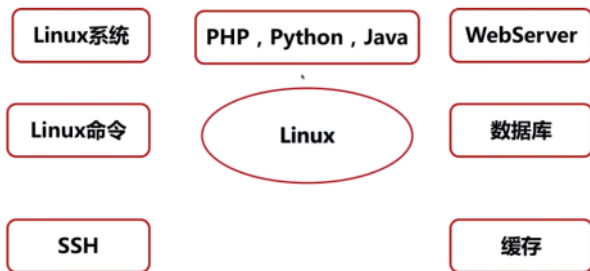


# Linux

2018年6月2日 11:02



## 第一章 研发必须知道的服务器Linux知识

### 1. Linux本身是一个开源软件



centOS是在企业中应用比较广发的Linux系统

### 2. Linux可以做什么

- ① 可以做企业服务器
- ② 嵌入式开发（安卓手机，路由器，机顶盒都是基于Linux系统进行开发的）

### 3. Linux的学习方法

- ① 需要一个Linux的学习环境（强迫给自己一个环境）
- ② 先自己尝试搜索解决问题（解决问题的能力是非常重要的）
- ③ 学会读懂Linux的错误提示（Linux的提示语句都非常详细）

### 4. 忘掉Windows平台的所有东西

在Linux中

- ① 没有exe安装程序
- ② 大小写是要区分的
- ③ 一切皆文件
- ④ 文件的后缀名不是那么重要，只是为了好识别而已

### 5. 虚拟机

虚拟机是什么：虚拟机可以隔离本身的操作环节

- 科研解释：虚拟机（Virtual Machine）指通过软件模拟的具有完整硬件系统功能的、运行在一个完全隔离环境中的完整计算机系统
- 白话解释：通过软件技术，可以在自己的电脑上安装另一个操作系统，而不影响原系统

## 虚拟机软件

更多教程加qq33799776



- Virtualbox ( V5.1.24 ) 下载地址 :

[https://www.virtualbox.org/wiki/Download\\_Old\\_Builds\\_5\\_1](https://www.virtualbox.org/wiki/Download_Old_Builds_5_1)

- Xshell下载(Win) : [http://dlsw.baidu.com/sw-search-](http://dlsw.baidu.com/sw-search-sp/soft/7b/15201/Xshell_5.0.0.37_setup.1459931786.exe)

[sp/soft/7b/15201/Xshell\\_5.0.0.37\\_setup.1459931786.exe](http://dlsw.baidu.com/sw-search-sp/soft/7b/15201/Xshell_5.0.0.37_setup.1459931786.exe)

- Virtualbox ( V5.1.24 ) 下载地址 :

<https://pan.baidu.com/s/1jHYGeuM> 密码: dt63

- Xshell下载(Win) :

<https://pan.baidu.com/s/1qYqKRnA> 密码: zrky

第一个用的

云服务器和虚拟机区别在于有公网IP,虚拟机是平时用来学习Linux的,一般云服务器在买的时候会提供公网IP

## 云服务器的平台

- 阿里云 : <https://www.aliyun.com>
- 腾讯云 : <https://cloud.tencent.com>
- 网易蜂巢 : <https://c.163.com/>

### 第四章 准备工作

#### 1.查看ip

- 查看ip
  - ifconfig
  - ip addr
  - vi /etc/sysconfig/network-scripts/ifcfg-xx
  - yum install net-tools

```
[root@localhost ~]#
[root@localhost ~]# ifconfig
-bash: ifconfig: command not found
[root@localhost ~]# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
    qlen 1000
    link/ether 08:00:27:2c:32:47 brd ff:ff:ff:ff:ff:ff
[root@localhost ~]#
```

```

root@localhost ~]# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
    qlen 1000
    link/ether 08:00:27:2c:32:47 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic enp0s3
        valid_lft 86398sec preferred_lft 86398sec
    inet6 fe80::7118:549c:5329:1cb5/64 scope link
        valid_lft forever preferred_lft forever

```

## 2. 替换默认源

- 替换默认源

- <http://mirrors.163.com/.help/centos.html>

- 替换默认源

- <http://mirrors.163.com/.help/centos.html>

- 安装Vim

```

last failed login: Sat Jun  2 01:27:32 EDT 2018 on tty1
there were 2 failed login attempts since the last successful login.
last login: Sat Jun  2 00:43:03 on tty1
root@localhost ~]# ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 219.223.247.162 netmask 255.255.248.0 broadcast 219.223.247.255
    inet6 fe80::7118:549c:5329:1cb5 prefixlen 64 scopeid 0x20<link>
    inet6 2001::250:3c02:301:6a39:4e6f:4486:8651 prefixlen 128 scopeid 0x0
    global
    ether 08:00:27:2c:32:47 txqueuelen 1000 (Ethernet)
    RX packets 68 bytes 7525 (7.3 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 90 bytes 9212 (8.9 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1 (Local Loopback)
    RX packets 72 bytes 6260 (6.1 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 72 bytes 6260 (6.1 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@localhost ~]#

```

## 第五章 远程服务器的管理工具软件SSH工具

### 1.

## SSH工具

- SSH是什么
- 服务器安装SSH服务
- 客户端安装SSH客户端
- SSH客户端连接服务器

## SSH工具

- SSH config用法详解
- 免密码登录 方案之 SSH Key
- SSH端口安全
- 个性化脚本一键登录服务器

SSH①:Secure Shell 安全外壳协议，②是建立在应用层基础上的安全协议③可靠，专为远程登录会话和其他网络服务提供安全性的协议④有效的防止远程管理过程中的信息泄露问题⑤SSH客户端适用于多种平台⑥SSH服务端几乎支持所有的UNIX平台

2.

### 服务器安装SSH服务

- 安装SSH  
`yum install openssh-server`
- 启动SSH  
`service sshd start`
- 设置开机运行  
`chkconfig sshd on`

3.

### 客户端安装SSH工具

更多教程加qq3379977

- SSH是典型的客户端和服务端的交互模式，客户端广泛的支持各个平台
- Windows 有很多工具可以支持SSH连接功能，例如  
Xshell,Putty, secureCRT
- Linux平台需要安装客户端软件  
`yum install openssh-clients`

centos一般在安装客户端的时候服务端会自动安装完成，无需再安装

4.客户端连接SSH服务（Linux）：

```

[c:\~]$ ssh root@192.168.224.128

Connecting to 192.168.224.128:22...
Connection established.
To escape to local shell, press 'Ctrl+Alt+J'.

Last login: Sun Jun  3 10:12:51 2018 from 192.168.224.1
[root@localhost ~]# ifconfig
-bash: ifconfig: 未找到命令
[root@localhost ~]# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 00:0c:29:90:45:a0 brd ff:ff:ff:ff:ff:ff
    inet 192.168.224.128/24 brd 192.168.224.255 scope global dynamic ens33
        valid_lft 1040sec preferred_lft 1040sec
    inet6 fe80::ae73:d6fa:4444:6b27/64 scope link
        valid_lft forever preferred_lft forever

```

```

[root@localhost ~]# ssh root@192.168.224.128
The authenticity of host '192.168.224.128 (192.168.224.128)' can't be established.
ECDSA key fingerprint is SHA256:9pVS0xSx0Ve2n7ki6tK7UzZqCQbFdagUU6H7F9IZvXE.
ECDSA key fingerprint is MD5:4e:97:98:7b:de:a5:1d:18:70:6c:d2:97:a3:eb:38:ff.
Are you sure you want to continue connecting (yes/no)? y
Please type 'yes' or 'no': yy^Hes
Please type 'yes' or 'no': yes
Warning: Permanently added '192.168.224.128' (ECDSA) to the list of known hosts.
root@192.168.224.128's password:
Last login: Sun Jun  3 10:25:40 2018 from 192.168.224.1
[root@localhost ~]#

```

## 5.SSH config (配置)

### SSH config 讲解

- config为了方便我们批量管理多个ssh
- config 存放在 ~/.ssh/config
- config 配置语法

### SSH config 语法关键字

更多教程册

Host	别名
HostName	主机名
Port	端口
User	用户名
IdentityFile	密钥文件的路径

SSH默认端口是22端口

注意：重启电脑和重启虚拟机的Ip可能会变

```

Connecting to 192.168.0.105:22...
Canceled.

Type 'help' to learn how to use Xshell prompt.
[c:\~]$ ping 192.168.0.105

正在 Ping 192.168.0.105 具有 32 字节的数据:
来自 10.0.2.2 的回复: 无法访问目标网。
来自 10.0.2.2 的回复: 无法访问目标网。
来自 10.0.2.2 的回复: 无法访问目标网。

192.168.0.105 的 Ping 统计信息:
    数据包: 已发送 = 3, 已接收 = 3, 丢失 = 0 (0% 丢失),
[c:\~]$

```

```

Connecting to 192.168.0.105:22...
Canceled.

Type 'help' to learn how to use Xshell prompt.
[c:\~]$
[c:\~]$ ping 192.168.0.105

正在 Ping 192.168.0.105 具有 32 字节的数据:
来自 10.0.2.2 的回复: 无法访问目标网。
来自 10.0.2.2 的回复: 无法访问目标网。
来自 10.0.2.2 的回复: 无法访问目标网。

192.168.0.105 的 Ping 统计信息:
    数据包: 已发送 = 3, 已接收 = 3, 丢失 = 0 (0% 丢失)。

[c:\~]$

```

## 6.SSH安全免密码登录:ssh key

### SSH安全免密码登录:ssh key

- ssh key 使用非对称加密方式生成公钥 和 私钥
- 私钥存放在本地 ~/.ssh 目录
- 公钥可以对外公开, 放在服务器的 ~/.ssh/ authorized\_keys

```

"config" 9L, 119C written
[root@localhost ~]# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa): imooc_linux
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in imooc_linux.
Your public key has been saved in imooc_linux.pub.
The key fingerprint is:
SHA256:bq/wIT7AabSOV6J0e+U1+0rGr1CAcDF2cBzozHCNrFo root@localhost.localdomain
The key's randomart image is:
+---[RSA 2048]-----+
|  ..B+. |
|  .+=o  |
|  B.    |
|  E + . |
|  o . S . |
|  .oo+.oo |
|  .oB+=o++ |
|  .oX.=o. |
|  ....*..+o |
+---[SHA256]-----+
[root@localhost ~]# ls -al
total 52
dr-xr-x---. 3 root root 253 Jun 3 11:05 .
dr-xr-xr-x. 17 root root 224 Jun 3 02:15 ..
-rw-----. 1 root root 1261 Jun 3 02:16 anaconda-ks.cfg
-rw-----. 1 root root 139 Jun 3 10:57 .bash_history
-rw-r--r--. 1 root root 18 Dec 29 2013 .bash_logout
-rw-r--r--. 1 root root 176 Dec 29 2013 .bash_profile
-rw-r--r--. 1 root root 176 Dec 29 2013 .bashrc
-rw-r--r--. 1 root root 119 Jun 3 10:56 config
-rw-r--r--. 1 root root 100 Dec 29 2013 .cshrc
-rw-----. 1 root root 1679 Jun 3 11:05 imooc_linux
-rw-r--r--. 1 root root 408 Jun 3 11:05 imooc_linux.pub
-rw-r--r--. 1 root root 9116 Apr 11 2016 mysql57-community-release-el7-8.noarch.rpm
drwx-----. 2 root root 39 Jun 3 10:40 .ssh
-rw-r--r--. 1 root root 129 Dec 29 2013 .tcshrc

```

## 第六章 Linux常用命令

### (1) 软件操作命令: 软件如何安装, 如何卸载

- 软件包管理器: yum
- 安装软件: yum install xxx
- 卸载软件: yum remove xxx
- 搜索软件: yum search xxx
- 清理缓存: yum clean packages
- 列出已安装: yum list

```

• 软件包信息: yum info xxx

```

### (2) 服务器硬件资源和磁盘操作: 如何看CPU, 内存是多大 (有四个指标)

## 服务器硬件资源信息

- 内存：free -m
- 硬盘：df -h
- 负载：w/top
- cpu个数 和 核数

```
[root@localhost ~]# free -m
              total        used          free      shared  buff/cache   available
Mem:           1823         173           1210          8         440         1457
Swap:          2047           0           2047
[root@localhost ~]# df -h
文件系统      容量  已用  可用  已用% 挂载点
/dev/mapper/cl-root    17G  1004M    17G    6% /
devtmpfs           901M     0   901M    0% /dev
tmpfs              912M     0   912M    0% /dev/shm
tmpfs              912M   8.8M   903M    1% /run
tmpfs              912M     0   912M    0% /sys/fs/cgroup
/dev/sdal          1014M  139M   876M   14% /boot
tmpfs              183M     0   183M    0% /run/user/0
```

```
[root@localhost ~]# w
12:51:27 up 18:32, 10 users, load average: 0.00, 0.02, 0.05
USER      TTY      FROM          LOGIN@   IDLE   JCPU   PCPU WHAT
root      tty1     192.168.224.1 11:09    1:37m  0.03s  0.03s ssh root@192.168.224.128
root      pts/0    192.168.224.1 10:25    1:54m  0.07s  0.03s ssh root@192.168.224.128
root      pts/2    192.168.224.128 10:53    1:54m  0.01s  0.01s -bash
root      pts/3    192.168.224.128 10:41    23:27  0.30s  0.27s ssh imooc
root      pts/4    192.168.224.128 10:43    23:27  0.06s  0.04s ssh root@192.168.224.128
root      pts/5    192.168.224.1 11:14    1:16m  0.03s  0.03s -bash
root      pts/6    192.168.224.128 11:31    23:27  0.03s  0.03s -bash
root      pts/7    192.168.224.1 12:29    7.00s  0.27s  0.22s ssh root@192.168.224.128
root      pts/8    192.168.224.128 12:42    7.00s  0.26s  0.20s w
```

```
[root@localhost ~]# cat /proc/cpuinfo
processor       : 0
vendor_id      : GenuineIntel
cpu family     : 6
model          : 94
model name     : Intel(R) Core(TM) i5-6500 CPU @ 3.20GHz
stepping       : 3
microcode      : 0x8a
cpu MHz        : 3192.771
cache size     : 6144 KB
physical id    : 0
siblings       : 2
core id        : 0
cpu cores      : 2
apicid         : 0
initial apicid : 0
fpu            : yes
fpu_exception  : yes
cpuid level    : 22
wp             : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush d
ts mmx fxsr sse sse2 ss ht syscall nx pdpe1gb rdtscp lm constant_tsc arch_perfmon pebs bts nopl xto
pology tsc_reliable nonstop_tsc aperfmperf eagerfpu pni pclmulqdq sse3 fma cx16 pcid sse4_1 sse4_2
x2apic movbe popcnt tsc_deadline_timer aes xsave avx f16c rdrand hypervisor lahf_lm abm 3dnowprefe
tch ida arat epb pln pts dtherm hwp hwp_noitfy hwp_act_window hwp_epp fsgsbase tsc_adjust bml hle
avx2 smep bmi2 invpcid rtm rdseed adx smap xsaveopt xsavec xgetbv1
bogomips       : 6384.12
clflush size   : 64
cache_alignme  : 64
address sizes   : 42 bits physical, 48 bits virtual
power managemen
```



```
processor      : 1
vendor_id     : GenuineIntel
cpu family    : 6
model         : 94
model name    : Intel(R) Core(TM) i5-6500 CPU @ 3.20GHz
stepping      : 3
microcode     : 0x8a
cpu MHz       : 3192.771
cache size    : 6144 KB
physical id   : 0
siblings      : 2
core id       : 1
cpu cores     : 2
apicid        : 1
initial apicid : 1
fpu           : yes
fpu_exception : yes
cpuid level   : 22
wp            : yes
flags         : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush d
ts mmx fxsr sse sse2 ss ht syscall nx pdpe1gb rdtscp lm constant_tsc arch_perfmon pebs bts nopl xto
pology tsc_reliable nonstop_tsc aperfmperf eagerfpu pni pclmulqdq ssse3 fma cx16 pcid sse4_1 sse4_2
x2apic movbe popcnt tsc_deadline_timer aes xsave avx f16c rdrand hypervisor lahf_lm abm 3dnowprefe
tch ida arat epb pln pts dtherm hwp hwp_notify hwp_act_window hwp_epp fsgsbase tsc_adjust bmi1 hle
avx2 smep bmi2 invpcid rtm rdseed adx smap xsaveopt xsavec xgetbv1
bogomips      : 6384.12
clflush size  : 64
cache_alignm  : 64
address sizes  : 42 bits physical, 48 bits virtual
power manage   :

```

格式化磁盘：

```
[root@localhost ~]# fdisk
```

用法:

```
fdisk [选项] <磁盘>    更改分区表
fdisk [选项] -l <磁盘>  列出分区表
fdisk -s <分区>         给出分区大小(块数)
```

选项:

```
-b <大小>          扇区大小(512、1024、2048或4096)
-c [= <模式>]      兼容模式: "dos"或"nondos"(默认)
-h                打印此帮助文本
-u [= <单位>]       显示单位: "cylinders"(柱面)或"sectors"(扇区, 默认)
-v                打印程序版本
-C <数字>          指定柱面数
-H <数字>          指定磁头数
-S <数字>          指定每个磁道的扇区数
```

第一个值表示最近一分钟的平均负载，第二个值表示最近5分钟的平均负载，第三个表示最近15分钟的平均负载

(3) 文件和文件夹操作命令：相当于windows中的文件和文件夹，怎样新建文件、文件夹，怎么删除文件和文件夹

## 文件操作命令

- Linux文件的目录结构

- 文件基本操作

- 文本编辑神器Vim

- 文件权限 421

- 文件搜索，查找，读取

- 文件压缩与解压

(一)

①根目录 "/"

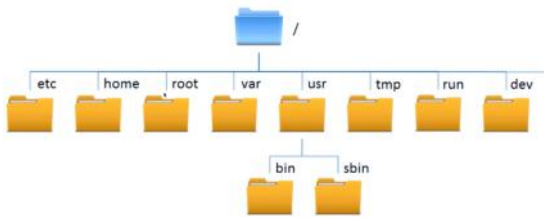
②家目录 "/home" (前面提到的家目录 "~"，表示当前页面的家目录)

③临时目录 "/tmp"

④配置目录 "/etc"

⑤用户程序目录 "/usr"





```

[root@localhost ~]# cd /
[root@localhost /]# ls -al
总用量 16
dr-xr-xr-x. 17 root root 224 6月 3 02:15 .
dr-xr-xr-x. 17 root root 224 6月 3 02:15 ..
lrwxrwxrwx. 1 root root 7 6月 3 02:13 bin -> usr/bin
dr-xr-xr-x. 4 root root 4096 6月 3 02:18 boot
drwxr-xr-x. 20 root root 3180 6月 3 02:18 dev
drwxr-xr-x. 75 root root 8192 6月 3 12:43 etc
drwxr-xr-x. 2 root root 6 11月 5 2016 home
lrwxrwxrwx. 1 root root 7 6月 3 02:13 lib -> usr/lib
lrwxrwxrwx. 1 root root 9 6月 3 02:13 lib64 -> usr/lib64
drwxr-xr-x. 2 root root 6 11月 5 2016 media
drwxr-xr-x. 2 root root 6 11月 5 2016 mnt
drwxr-xr-x. 2 root root 6 11月 5 2016 opt
dr-xr-xr-x. 125 root root 0 6月 3 02:18 proc
dr-xr-xr-x. 3 root root 253 6月 3 11:05 root
drwxr-xr-x. 23 root root 700 6月 3 12:45 run
lrwxrwxrwx. 1 root root 8 6月 3 02:13/sbin -> usr/sbin
drwxr-xr-x. 2 root root 6 11月 5 2016 srv
dr-xr-xr-x. 13 root root 0 6月 3 02:18 sys
drwxrwxrwt. 7 root root 132 6月 3 12:43 tmp
drwxr-xr-x. 13 root root 155 6月 3 02:13 usr
drwxr-xr-x. 19 root root 267 6月 3 02:18 var
  
```

```

[root@localhost /]# cd /home
[root@localhost home]# ls -al
总用量 0
drwxr-xr-x. 2 root root 6 11月 5 2016 .
dr-xr-xr-x. 17 root root 224 6月 3 02:15 ..
[root@localhost home]# cd /etc
[root@localhost etc]# ls -al
总用量 1084
drwxr-xr-x. 75 root root 8192 6月 3 12:43 .
dr-xr-xr-x. 17 root root 224 6月 3 02:15 ..
-rw-r--r--. 1 root root 16 6月 3 02:15 adjtime
-rw-r--r--. 1 root root 1518 6月 7 2013 aliases
-rw-r--r--. 1 root root 12288 6月 3 02:18 aliases.db
drwxr-xr-x. 2 root root 236 6月 3 02:14 alternatives
-rw-r--r--. 1 root root 541 3月 31 2016 anacrontab
-rw-r--r--. 1 root root 55 11月 5 2016 asound.conf
drwxr-xr-x. 3 root root 43 6月 3 02:14 audisp
drwxr-xr-x. 3 root root 83 6月 3 02:18 audit
drwxr-xr-x. 2 root root 22 6月 3 02:14 bash_completion.d
-rw-r--r--. 1 root root 2853 11月 6 2016 bashrc
drwxr-xr-x. 2 root root 6 11月 7 2016 binfmt.d
-rw-r--r--. 1 root root 38 11月 30 2016 centos-release
-rw-r--r--. 1 root root 51 11月 30 2016 centos-release-upstream
drwxr-xr-x. 2 root root 6 11月 6 2016 chkconfig.d
-rw-r--r--. 1 root root 1165 11月 15 2016 chrony.conf
-rw-r--r--. 1 root chrony 62 6月 3 02:18 chrony.keys
drwxr-xr-x. 2 root root 21 6月 3 02:13 cron.d
drwxr-xr-x. 2 root root 42 6月 3 02:14 cron.daily
-rw-r--r--. 1 root root 0 3月 31 2016 cron.deny
drwxr-xr-x. 2 root root 22 6月 10 2014 cron.hourly
drwxr-xr-x. 2 root root 6 6月 10 2014 cron.monthly
-rw-r--r--. 1 root root 451 6月 10 2014 crontab
drwxr-xr-x. 2 root root 6 6月 10 2014 cron.weekly
  
```

## (二) 文件操作的基本命令

### 文件操作基本命令

更多教程

命令	解释
ls	查看目录下的文件
touch	新建文件
mkdir	新建文件夹
cd	进入目录

## 文件操作基本命令

命令	解释
rm	删除文件和目录
cp	复制
mv	移动
pwd	显示路径

在centos中将“ls -al” 等同于 “ll”

接着

```
[root@localhost tmp]# cd /tmp/
[root@localhost tmp]# ll
总用量 4
-rwx-----. 1 root root 836 6月  3 02:16 ks-script-BtN5eD
-rw-----. 1 root root  0 6月  3 02:13 yum.log
[root@localhost tmp]# touch imooc
[root@localhost tmp]# ll
总用量 4
-rw-r--r--. 1 root root  0 6月  3 14:10 imooc
-rwx-----. 1 root root 836 6月  3 02:16 ks-script-BtN5eD
-rw-----. 1 root root  0 6月  3 02:13 yum.log
[root@localhost tmp]# touch imooc.log
[root@localhost tmp]# mkdir immoc
[root@localhost tmp]# ll
总用量 4
drwxr-xr-x. 2 root root  6 6月  3 14:11 immoc
-rw-r--r--. 1 root root  0 6月  3 14:10 imooc
-rw-r--r--. 1 root root  0 6月  3 14:11 imooc.log
-rwx-----. 1 root root 836 6月  3 02:16 ks-script-BtN5eD
-rw-----. 1 root root  0 6月  3 02:13 yum.log
```

循环创建

```
[root@localhost tmp]# mkdir -p imooc_dir/test1/test2
[root@localhost tmp]# cd imooc_dir
[root@localhost imooc_dir]# ll
总用量 0
drwxr-xr-x. 3 root root 19 6月  3 14:12 test1
[root@localhost imooc_dir]# cd test1/
[root@localhost test1]# ll
总用量 0
drwxr-xr-x. 2 root root  6 6月  3 14:12 test2
```

删除一个文件

```
总用量 4
drwxr-xr-x. 2 root root  6 6月  3 14:11 immoc
-rw-r--r--. 1 root root  0 6月  3 14:10 imooc
drwxr-xr-x. 3 root root 19 6月  3 14:12 imooc_dir
-rw-r--r--. 1 root root  0 6月  3 14:11 imooc.log
-rwx-----. 1 root root 836 6月  3 02:16 ks-script-BtN5eD
-rw-----. 1 root root  0 6月  3 02:13 yum.log
[root@localhost tmp]# rm imooc
rm: 是否删除普通空文件 "imooc"? y
[root@localhost tmp]# ll
总用量 4
drwxr-xr-x. 2 root root  6 6月  3 14:11 immoc
drwxr-xr-x. 3 root root 19 6月  3 14:12 imooc_dir
-rw-r--r--. 1 root root  0 6月  3 14:11 imooc.log
-rwx-----. 1 root root 836 6月  3 02:16 ks-script-BtN5eD
-rw-----. 1 root root  0 6月  3 02:13 yum.log
```

删除一个文件夹

```
[root@localhost tmp]# rm -r imooc_dir
rm: 是否进入目录"imooc_dir"? y
rm: 是否进入目录"imooc_dir/test1"? y
rm: 是否删除目录 "imooc_dir/test1/test2"? y
rm: 是否删除目录 "imooc_dir/test1"? y
rm: 是否删除目录 "imooc_dir"? y
[root@localhost tmp]# ll
总用量 4
drwxr-xr-x. 2 root root  6 6月  3 14:11 immoc
-rw-r--r--. 1 root root  0 6月  3 14:11 imooc.log
-rwx-----. 1 root root 836 6月  3 02:16 ks-script-BtN5eD
-rw-----. 1 root root  0 6月  3 02:13 yum.log
```

强制删除一个文件，不会出现是否要你删除的提示符

```
[root@localhost tmp]# rm -rf imooc.log
[root@localhost tmp]# ll
总用量 4
drwxr-xr-x. 2 root root  6 6月  3 14:11 immoc
-rwx-----. 1 root root 836 6月  3 02:16 ks-script-BtN5eD
-rw-----. 1 root root  0 6月  3 02:13 yum.log
```

新建一个文件夹并将其移动到“~” 目录下的一个不存在的文件夹“imooc.log2”中（就是复制操作，此时两个文件下面都有）

```

[root@localhost tmp]# touch imooc.log
[root@localhost tmp]# cp ./imooc.log ~/imooc.log2
[root@localhost tmp]# cd ~
[root@localhost ~]# ll
总用量 28
-rw-r--r-- 1 root root 1261 6月 3 02:16 anaconda-ks.cfg
-rw-r--r-- 1 root root 119 6月 3 10:56 config
-rw-r--r-- 1 root root 1679 6月 3 11:05 imooc_linux
-rw-r--r-- 1 root root 408 6月 3 11:05 imooc_linux.pub
-rw-r--r-- 1 root root 0 6月 3 14:19 imooc.log2
-rw-r--r-- 1 root root 9116 4月 11 2016 mysql57-community-release-el7-8.noarch.rpm

```

将tmp文件夹下的文件移动到~文件夹中（移动就相当于剪切）

```

[root@localhost tmp]# mv ./imooc.log ~/
[root@localhost tmp]# cd ~
[root@localhost ~]# ll
总用量 28
-rw-r--r-- 1 root root 1261 6月 3 02:16 anaconda-ks.cfg
-rw-r--r-- 1 root root 119 6月 3 10:56 config
-rw-r--r-- 1 root root 1679 6月 3 11:05 imooc_linux
-rw-r--r-- 1 root root 408 6月 3 11:05 imooc_linux.pub
-rw-r--r-- 1 root root 0 6月 3 14:19 imooc.log
-rw-r--r-- 1 root root 0 6月 3 14:19 imooc.log2
-rw-r--r-- 1 root root 9116 4月 11 2016 mysql57-community-release-el7-8.noarch.rpm
[root@localhost ~]#

```

显示路径

```

[root@localhost ~]# pwd
/root
[root@localhost ~]# cd /etc/yum.repos.d/
[root@localhost yum.repos.d]# pwd
/etc/yum.repos.d

```

（三）Linux文本编辑神器：Vim

在服务器上我们是没有类Windows的操作界面的，所有的GUI界面都无法使用；Vi是最出名的Linux文本编辑神器，可以对文件进行修稿操作；

Vim的安装命令：yum install vim

Vim操作也可以新建并操作一个文件夹

```

[root@localhost tmp]# vim imooc
2345
456789
89999999
131424235
22222222
33444444
~
~
~
~
~

```

按 “i” 可以进行编辑

按 “:wq” 进行保存

按住键盘的大写 “G” 可以将光标移动到行尾（当文件比较大的时候一屏可能看不完）

按住键盘的小写 “g” 两遍可以将光标移动到行头

“dd” :如果想删除一行，只需要将光标移动到一行，然后按dd即可

“u” :是恢复键

“yy” :是复制操作

## vi / vim 键盘图

Esc  
命令  
模式

动作	移动光标, 或者定义操作的范围
命令	直接执行的命令, <b>红色命令</b> 进入编辑模式
操作	后面跟随表示操作范围的指令

<b>extra</b>	特殊功能， 需要额外的输入
<b>q</b>	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**)

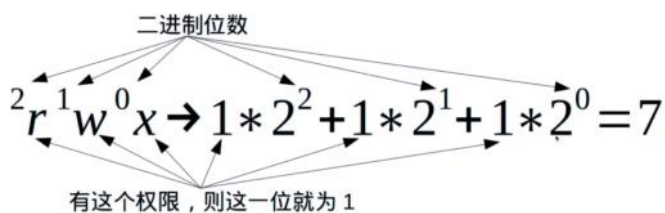
备注:

- (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(即贴板)  
(如: "ay8 拷贝剩余的行内容至寄存器 'a'")
- (2) 命令后添加数字  
多遇重复操作  
(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)

原图: [www.viemu.com](http://www.viemu.com) 翻译: *fdd (linuxsir)*

#### （四）文件权限 4-2-1

### 文件权限 4-2-1



## 文件权限 4-2-1

权限	数字
r	4
w	2
x	1

其中“777”表示的是最大的权限，意思就是“rwxrwxrwx”

一般只读设置为“4”，读加写“4+2”，可执行“4+2+1”

### （五）文件的搜索、查找、读取

## 文件搜索、查找、读取

命令	解释
tail	从文件尾部开始读
head	从文件头部读
cat	读取整个文件
more	分页读取

## 从尾部开始读取文件

grep:搜索关键字，下面的第一条操作是搜索“11”，第二条操作是搜索含有“11”的行数

统计行数：WC

分区 新分区 1 的第 13 页



```
[root@localhost tmp]# find . -name "*ooc"
./imoooc
[root@localhost tmp]#
```

```
[root@localhost tmp]# find . -type f
find: './-type': 没有那个文件或目录
find: 'f': 没有那个文件或目录
[root@localhost tmp]# find . -type d
.
./font-unix
./X11-unix
./ICE-unix
./Test-unix
./XIM-unix
./imoooc
```

"\*" 表示的是所有的字符；f表示文件；d表示文件夹

(六) 文件的压缩 (tar) 两种方式，特别是 ".gz"这个命令用的比较多，用的是二次压缩

①tar -cvf|tvf|xvf

②tar -czvf|tzvf|xzvf

```
EXAMPLES
tar -cf archive.tar foo bar
# Create archive.tar from files foo and bar.

tar -tvf archive.tar
# List all files in archive.tar verbosely.

tar -xf archive.tar
# Extract all files from archive.tar.
```

```
[root@localhost tmp]# man tar
[root@localhost tmp]# touch imoooc
[root@localhost tmp]# vim imoooc
[root@localhost tmp]# tar -cf imoooc.tar imoooc
[root@localhost tmp]# ls -al
总用量 32
drwxrwxrwt. 8 root root 193 6月 3 15:43 .
dr-xr-xr-x. 17 root root 224 6月 3 02:15 ..
drwxrwxrwt. 2 root root 6 6月 3 02:13 .font-unix
drwxrwxrwt. 2 root root 6 6月 3 02:13 .ICE-unix
drwxr-xr-x. 2 root root 6 6月 3 14:11 imoooc
-rw-r--r--. 1 root root 227 6月 3 15:43 imoooc
-rw-r--r--. 1 root root 12288 6月 3 14:34 .imoooc.swp
-rw-r--r--. 1 root root 10240 6月 3 15:43 imoooc.tar
-rwx-----. 1 root root 836 6月 3 02:16 ks-script-BtN5eD
drwxrwxrwt. 2 root root 6 6月 3 02:13 .Test-unix
drwxrwxrwt. 2 root root 6 6月 3 02:13 .X11-unix
drwxrwxrwt. 2 root root 6 6月 3 02:13 .XIM-unix
-rw-----. 1 root root 0 6月 3 02:13 yum.log
```

解压

```
-rw-r--r--. 1 root root 10240 6月 3 15:43 imoooc.tar
-rwx-----. 1 root root 836 6月 3 02:16 ks-script-BtN5eD
-rw-----. 1 root root 0 6月 3 02:13 yum.log
[root@localhost tmp]# tar -xf imoooc.tar
[root@localhost tmp]# ll
总用量 20
-rw-r--r--. 1 root root 227 6月 3 15:43 imoooc
-rw-r--r--. 1 root root 10240 6月 3 15:43 imoooc.tar
-rwx-----. 1 root root 836 6月 3 02:16 ks-script-BtN5eD
-rw-----. 1 root root 0 6月 3 02:13 yum.log
```

```

[root@localhost tmp]# tar -czvf imooc.tar.gz imooc
imooc
[root@localhost tmp]# ll
总用量 24
-rw-r--r--. 1 root root 227 6月 3 15:43 imooc
-rw-r--r--. 1 root root 10240 6月 3 15:43 imooc.tar
-rw-r--r--. 1 root root 148 6月 3 15:48 imooc.tar.gz
-rwx----- 1 root root 836 6月 3 02:16 ks-script-BtN5eD
-rw----- 1 root root 0 6月 3 02:13 yum.log

```

```

[root@localhost tmp]# tar -czvf imooc.tar.gz imooc
imooc
[root@localhost tmp]# ll
总用量 24
-rw-r--r--. 1 root root 227 6月 3 15:43 imooc
-rw-r--r--. 1 root root 10240 6月 3 15:43 imooc.tar
-rw-r--r--. 1 root root 148 6月 3 15:48 imooc.tar.gz
-rwx----- 1 root root 836 6月 3 02:16 ks-script-BtN5eD
-rw----- 1 root root 0 6月 3 02:13 yum.log
[root@localhost tmp]# tar -tzvf imooc.tar.gz
-rw-r--r-- root/root 227 2018-06-03 15:43 imooc
[root@localhost tmp]# rm -f imooc
[root@localhost tmp]# tar -xzvf imooc.tar.gz
imooc
[root@localhost tmp]# ll
总用量 24
-rw-r--r--. 1 root root 227 6月 3 15:43 imooc
-rw-r--r--. 1 root root 10240 6月 3 15:43 imooc.tar
-rw-r--r--. 1 root root 148 6月 3 15:48 imooc.tar.gz
-rwx----- 1 root root 836 6月 3 02:16 ks-script-BtN5eD
-rw----- 1 root root 0 6月 3 02:13 yum.log

```

(4) 系统用户操作命令：如何添加一个系统用户，但是开启的话会造成很多困扰（比较高级）

### 系统用户操作命令

命令	解释
useradd	添加用户
adduser	添加用户
userdel	删除用户
passwd	设置密码

```

[root@localhost home]# adduser imooc_test
[root@localhost home]# ls -al
总用量 0
drwxr-xr-x. 4 root root 37 6月 3 17:24 .
dr-xr-xr-x. 17 root root 224 6月 3 02:15 ..
drwx----- 2 imooc imooc 62 6月 3 15:57 imooc
drwx----- 2 imooc_test imooc_test 62 6月 3 17:24 imooc_test
[root@localhost home]# userdel imooc_test
[root@localhost home]# ls -al
总用量 0
drwxr-xr-x. 4 root root 37 6月 3 17:24 .
dr-xr-xr-x. 17 root root 224 6月 3 02:15 ..
drwx----- 2 imooc imooc 62 6月 3 15:57 imooc
drwx----- 2 1001 1001 62 6月 3 17:24 imooc_test

```

用下面操作彻底删除

```

[root@localhost home]# userdel -r imooc_test
[root@localhost home]# ll
总用量 0
drwx----- 2 imooc imooc 62 6月 3 15:57 imooc

```

(5) 防火墙相关设置：在centos中一般默认防火墙都是开启的

- ① 防火墙的作用：保护服务器的安全
- ② 设置防火墙的规则：（一）开放80（web server），22（ssh）端口
- ③ 关闭防火墙：只要是线上服务器，一定要使用防火墙。
- ④ 防火墙的基本命令：

### 防火墙设置

- 安装：yum install firewalld
- 启动：service firewalld start
- 检查状态：service firewalld status
- 关闭或禁用防火墙：service firewalld stop/disable

(6) 提权操作sudo和文件传输操作：当使用除了root之外的账户的话会发现很多操作都是需要提权的，此时需要使用sudo的操作；至于文件传输协议，比如想将本地windows里面的传输协议上传到centos中，此时需要用到文件传输操作。



## 提权 和 文件上传下载操作

- 提权 : sudo
  - visudo
- 文件下载
  - wget、curl
- 文件上传
  - scp

提权操作就是将root操作使其和普通用户能够进行等同

```
总需要 root 权限执行此命令。
[imooc@localhost ~]$ sudo yum install vim

我们信任您已经从系统管理员那里了解了日常注意事项。
总结起来无外乎这三点：

#1) 尊重别人的隐私。
#2) 输入前要先考虑(后果和风险)。
#3) 权力越大，责任越大。

[sudo] imooc 的密码：
imooc 不在 sudoers 文件中。此事将被报告。
[imooc@localhost ~]$ sudo yum install vim
[sudo] imooc 的密码：
已加载插件：fastestmirror
Loading mirror speeds from cached hostfile
软件包 2:vim-enhanced-7.4.160-4.el7.x86_64 已安装并且是最新版本
无须任何处理
```

```
drwx----- 2 imooc imooc 83 6月  3 18:05 imooc
[root@localhost home]# visudo
您在 /var/spool/mail/root 中有邮件
[root@localhost home]# ll
总用量 0
drwx----- 2 imooc imooc 83 6月  3 18:05 imooc
```

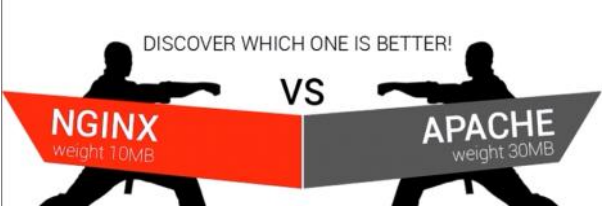
### 文件下载

```
imooc imooc.tar imooc.tar.gz ks-script-BYN58U yum.log
[imooc@localhost tmp]$ wget http://www.baidu.com
--2018-06-03 22:08:34-- http://www.baidu.com/
正在解析主机 www.baidu.com (www.baidu.com)... 183.232.231.173, 183.232.231.172
正在连接 www.baidu.com (www.baidu.com)|183.232.231.173|:80... 已连接。
已发出 HTTP 请求，正在等待回应... 200 OK
长度：2381 (2.3K) [text/html]
正在保存至：“index.html”

100%[=====>] 2,381      --.-K/s 用时 0s

2018-06-03 22:08:34 (128 MB/s) - 已保存 “index.html” (2381/2381))
```

## 七、WebServer



Apache基本操作	
解释	命令
安装	yum install httpd
启动	service httpd start
停止	service httpd stop

```

[imooc@localhost ~]$ sudo service httpd start
Redirecting to /bin/systemctl start httpd.service
[imooc@localhost ~]$ ps -ef | grep httpd
root      2366      1  0 21:34 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache    2367    2366  0 21:34 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache    2368    2366  0 21:34 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache    2369    2366  0 21:34 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache    2370    2366  0 21:34 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache    2371    2366  0 21:34 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
imooc     2373    2289  0 21:34 pts/1    00:00:00 grep --color=auto httpd
[imooc@localhost ~]$ sudo service httpd stop
Redirecting to /bin/systemctl stop httpd.service
[imooc@localhost ~]$ ps -ef | grep httpd
imooc     2398    2289  0 21:34 pts/1    00:00:00 grep --color=auto httpd

```

```

完毕！
[imooc@localhost ~]$ sudo netstat -anpl | grep 'http'
tcp6      0 :::80                :::*                  LISTEN      2420/httpd
unix      3  [ ]                  STREAM              CONNECTED    21086      2420/httpd
[imooc@localhost ~]$

```

当不能使用netstat时是因为没有安装net-tools的工具包，执行yum install net-tools即可

同样的执行上面操作已经知道80端口已经起来了，但是还是没有办法访问网站，是因为防火墙还没有关闭，执行下列操作关闭防火墙即可

```

完毕！
[imooc@localhost ~]$ sudo netstat -anpl | grep 'http'
tcp6      0 :::80                :::*                  LISTEN      2420/httpd
unix      3  [ ]                  STREAM              CONNECTED    21086      2420/httpd
[imooc@localhost ~]$
[imooc@localhost ~]$
[imooc@localhost ~]$ sudo service firewalld stop
Redirecting to /bin/systemctl stop firewalld.service

```



这就说明阿帕奇已经安装成功了

## 7 ( 2 ) 虚拟主机

```

[imooc@localhost ~]$ cd /etc/httpd
[imooc@localhost httpd]$ ls
conf  conf.d  conf.modules.d  logs  modules  run
[imooc@localhost httpd]$ ls -al
总用量 12
drwxr-xr-x. 5 root root 92 6月 10 21:33 .
drwxr-xr-x. 77 root root 8192 6月 10 21:33 ..
drwxr-xr-x. 2 root root 37 6月 10 21:33 conf
drwxr-xr-x. 2 root root 82 6月 10 21:33 conf.d
drwxr-xr-x. 2 root root 146 6月 10 21:33 conf.modules.d
lrwxrwxrwx. 1 root root 19 6月 10 21:33 logs -> ../../var/log/httpd
lrwxrwxrwx. 1 root root 29 6月 10 21:33 modules -> ../../usr/lib64/httpd/modules
lrwxrwxrwx. 1 root root 10 6月 10 21:33 run -> /run/httpd

```

```

:w 保存文件但不退出vi
:w file 将修改另外保存到file中，不退出vi
:w! 强制保存，不推出vi
:wq 保存文件并退出vi
:wq! 强制保存文件，并退出vi
:q 不保存文件，退出vi
:q! 不保存文件，强制退出vi
:el 放弃所有修改，从上次保存文件开始再编辑命令历史

```

还有一个问题：如何在vim的结果中查找一个单词，只需要在vim进入之后

### vim编辑器中如何查找某个词

2017年04月18日 21:13:33

在命令模式（进入后直接按shift+：）下输入

/xxx，这个是查找文件中“xxx”这个单词，是从文件上面到下面查找；

? xxx，这个是查找文件中“xxx”这个单词，是从文件下面到上面查找，

linux中使用 tail -f 可以不断的刷新日志信息，例如  
tail -f logs.log

但是，实时监控日志信息时，怎么让不断刷新的信息暂停一下，怎么退出到命令行呢？

tail命令本身不提供暂停功能。但可以结合其他命令来达到类似的效果。  
不清楚你需要暂停来做什么，是需要暂停输出来查看屏幕上的一些细节，还是需要转而去处理一些其他事物？

如果是想停下来看看屏幕上的东西，可以把tail -f的输出定向到less或者more这样的程序去。这样可以采用手工翻页的方式来查看tail -f的输出。

如果是想转而去处理其他事物，可以把tail程序放到后台去（Ctrl+z），等处理完其他事情，再用fg命令将tail命令转到前台来继续查看。

ctrl + c 退出

ctrl+c是最简单的方法

如果按照视频所讲配置完Directory还是不能访问那么需要这样子设置

```
[imooc@localhost conf]$ sudo setenforce 0
```

7（3）伪静态实际上是借助rewrite模块来实现的

```
rwxr-xr-x. 1 root root 69128 4月 21 02:11 mod_rewrite.so
```

7（4）Nginx的基本操作

**Nginx基本操作** 更多教程加qq31

解释	命令
安装	yum install nginx
启动	service nginx start
停止	service nginx stop
重载	service nginx reload

关于Nginx的安装在CetOs中的安装是比较奇怪的，首先要先安装资源库

- 添加CentOS 7 Nginx yum资源库
- sudo rpm -Uvh http://nginx.org/packages/centos/7/noarch/RPMS/nginx-release-centos-7-0.el7.noarch.rpm



解决root后面的主机名变化的问题

**MySQL基本操作** 更多教程加qq31

解释	命令
安装服务端	yum install mysql-community-server
启动	service mysqld start/restart
停止	service mysqld stop

- CentOS7默认安装mariadb数据库
  - `yum remove mariadb-libs.x86_64`
- 下载Mysql源
  - `https://dev.mysql.com/downloads/repo/yum/`
- 安装源
  - `yum localinstall mysql57-community-release-el7-8.noarch.rpm`

## MySQL安装

更多教程加qq33

- 安装MySQL
  - `yum install mysql-community-server`
- 默认密码
  - `cat /var/log/mysqld.log | grep " password"`

## MySQL拓展知识

- 远程连接
- 开启Genelog
- 新建用户和权限操作
- 忘记root密码怎么办

## MySQL客户端工具

- SQLyog
- Sequal Pro
- Navicat
- phpMyadmin
- HeidiSQL

查看端口以及关闭防火墙

```
[imooc@localhost ~]$ sudo netstat -anpl | grep 'http'
tcp6      0      0 :::80          :::*           LISTEN     2447/httpd
unix 3      [ ]          STREAM        CONNECTED     20886      2447/httpd
[imooc@localhost ~]$ sudo service firewalld stop
Redirecting to /bin/systemctl stop firewalld.service
```

如何终止ping



最佳答案

推荐于2018-04-12 18:46:56

按住键盘的Ctrl+c,终止它继续ping下去,就会停止了,会总结出运行的数据包有多少,通断的有多少了。

Mysql的密码

```
[imooc@localhost ~]$ cat /var/log/mysqld.log | grep password
2018-06-14T04:18:25.239725Z 5 [Note] [MY-010454] [Server] A temporary password is generated for root@localhost: p#U7hyEmn8F;
```

真正修改的密码是

- The `validate_password` plugin is installed and enabled.
- A superuser account 'root'@'localhost' is created. A password for the superuser is set and stored in the error log file. To reveal it, use the following command:

```
1 shell> sudo grep 'temporary password' /var/log/mysql.log
```

Change the root password as soon as possible by logging in with the generated, temporary password and set a custom password for the superuser account.

```
1 shell> mysql -uroot -p
```

```
1 mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'MyNewPass4!';
```

```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'ChangLu666!';
Query OK, 0 rows affected (0.01 sec)
```

通用日志的目的是记录所有的数据库语句：

```
[root@localhost ~]# tail -f /tmp/general.log
/usr/sbin/mysqld, Version: 8.0.11 (MySQL Community Server - GPL). started with:
Tcp port: 3306 Unix socket: /var/lib/mysql/mysql.sock
Time Id Command Argument
```

设置成这个样子就可以同步显示日志

```
mysql> set global general_log=on;
Query OK, 0 rows affected (0.01 sec)
```

设置成这个样子就不会显示

```
mysql> XshellXshellmysql>
mysql> set global general_log=off;
Query OK, 0 rows affected (0.01 sec)
```

总结：通用日志的设计步骤：

(1)

```
mysql> set global general_log_file="/tmp/general.log";
Query OK, 0 rows affected (0.00 sec)
```

(2)

```
mysql> set global general_log=on;
Query OK, 0 rows affected (0.01 sec)
```

(3) 新建用户

## 9.缓存服务

memcache和redis的区别：

memcache只支持key-val

**Redis扩展知识**



- Redis不仅仅支持简单的k/v类型的数据，同时还提供list, set, hash等数据结构的存储
- Redis支持数据的备份，即master-slave模式的数据备份
- Redis支持数据的持久化，可以将内存中的数据保持在磁盘中，重启的时候可以再次加载进行使用

**缓存服务**



## Memcached基本操作 [更多教程加q](#)

解释	命令
安装	yum install memcached
启动	memcached -d -l -m -p
停止	kill pid

## Redis基本操作 [更多教程加q](#)

解释	命令
安装	源码编译安装
启动	redis-server start/restart
停止	redis-server stop
客户端	redis-client

tomcat、nginx是为了

```
imooc@localhost ~]$ sudo netstat -anpl | grep mem
sudo] imooc 的密码:
tcp        0      0 0.0.0.0:11211          0.0.0.0:*              LISTEN
*2387/memcached
tcp6       0      0 :::11211               :::*                    LISTEN
*2387/memcached
udp        0      0 0.0.0.0:11211          0.0.0.0:*              *
*2387/memcached
udp6       0      0 :::11211               :::*                    *
*2387/memcached
unix 3      [ ]          STREAM  CONNECTED  18219      2387/memcached
unix 3      [ ]          STREAM  CONNECTED  18220      2387/memcached
unix 3      [ ]          STREAM  CONNECTED  18216      2387/memcached
unix 3      [ ]          STREAM  CONNECTED  18217      2387/memcached
unix 3      [ ]          STREAM  CONNECTED  18226      2387/memcached
unix 3      [ ]          STREAM  CONNECTED  18223      2387/memcached
unix 3      [ ]          STREAM  CONNECTED  18222      2387/memcached
```

```
先干：
[imooc@localhost ~]$ telnet 127.0.0.1 80
Trying 127.0.0.1...
telnet: connect to address 127.0.0.1: Connection refused
[imooc@localhost ~]$ ps -ef | grep nginx
imooc    2407    2335    0 22:22 pts/1    00:00:00 grep --color=auto nginx
[imooc@localhost ~]$ sudo service nginx start
Redirecting to /bin/systemctl start nginx.service
[imooc@localhost ~]$ ps -ef | grep nginx
root     2427     1    0 22:22 ?          00:00:00 nginx: master process /usr/sb
in/nginx -c /etc/nginx/nginx.conf
nginx    2428    2427    0 22:22 ?          00:00:00 nginx: worker process
imooc    2430    2335    0 22:22 pts/1    00:00:00 grep --color=auto nginx
[imooc@localhost ~]$ telnet 127.0.0.1 80
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
[
```

telnet是用来①检测一个端口是不是通的（非常好用）②

```
8.0.1Z4WZ["ydm(f&uPG(9!caching_sha2_passwordConnect
[imooc@localhost ~]$ telnet 127.0.0.1 11211
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
[
```

```

set imooc 0 60 5
hello
STORED
geet^H^H
ERROR
get imooc
VALUE imooc 0 5
hello
END
delette^H^H
ERROR
delete imooc
DELETED
get imooc
END

```

## Memcache 服务

`sudo yum install memcache`  
`memcached`

另开一个终端，进入服务器，  
`ssh imooc@你的 ip`

`telnet 127.0.0.1 11211`  
 -bash: telnet: command not found

`yum install telnet.*`  
`telnet 127.0.0.1 11211`  
`set imooc 0 60 5`  
`hello`  
`get imooc`

`telnet 127.0.0.1 11211`  
 -bash: telnet: command not found

`yum install telnet.*`  
`telnet 127.0.0.1 11211`  
`set imooc 0 60 5`  
`hello`  
`get imooc`  
`delete imooc`

附上一个 memcache 学习指导网址：

<http://blog.csdn.net/dailywater/article/details/50783636>

## Redis安装（源码编译）

### Redis 服务

#### 1:获取源码

`wget http://download.redis.io/releases/redis-4.0.2.tar.gz`

#### 2：解压

`tar xvfz redis-4.0.2.tar.gz`

#### 3：安装

`yum install gcc`  
`make malloc = libc`  
`sudo make install`

#### 4：启动

`cd src`  
`ls`

命令	简介
<code>redis-server</code>	Redis 服务器端启动程序
<code>redis-cli</code>	Redis 客户端操作工具。也可以用 telnet 根据其纯文本协议来操作
<code>redis-benchmark</code>	Redis 性能测试工具
<code>redis-check-aof</code>	数据修复工具
<code>redis-check-dump</code>	检查导出工具

启动 `./redis-server`

```

[imooc@localhost ~]$ ps -ef | grep redis
imooc 3576 2467 0 22:52 pts/0 00:00:00 ./redis-server *:6379
imooc 3606 3584 0 22:53 pts/1 00:00:00 grep --color=auto redis
[imooc@localhost ~]$ sudo netstat -anpl | grep redis
[sudo] imooc 的密码:
tcp        0      0 0.0.0.0:6379          0.0.0.0:*             LISTEN      3576/./redis-server
tcp6      0      0 :::6379              :::*                   LISTEN      3576/./redis-server

```



```
[imooc@localhost src]$ ./redis-server
3576:C 19 Jun 22:52:25.272 # o000o000o000o Redis is starting o000o000o000o
3576:C 19 Jun 22:52:25.272 # Redis version=4.0.10, bits=64, commit=00000000, modified=0, pid=3576,
just started
3576:C 19 Jun 22:52:25.272 # Warning: no config file specified, using the default config. In order
to specify a config file use ./redis-server /path/to/redis.conf
3576:M 19 Jun 22:52:25.274 # You requested maxclients of 10000 requiring at least 10032 max file de
scriptors.
3576:M 19 Jun 22:52:25.274 # Server can't set maximum open files to 10032 because of OS error: Oper
ation not permitted.
3576:M 19 Jun 22:52:25.274 # Current maximum open files is 4096. maxclients has been reduced to 409
6 to compensate for low ulimit. If you need higher maxclients increase 'ulimit -n'.

Redis 4.0.10 (00000000/0) 64 bit

Running in standalone mode
Port: 6379
PID: 3576

http://redis.io
```

```
[imooc@localhost src]$ ./redis-cli
127.0.0.1:6379> set imooc hello
OK
127.0.0.1:6379> get imooc
"hello"
127.0.0.1:6379> del imooc
(integer) 1
```

```
[imooc@localhost src]$ ./redis-cli
127.0.0.1:6379> set imooc hello
OK
127.0.0.1:6379> get imooc
"hello"
127.0.0.1:6379> del imooc
(integer) 1
127.0.0.1:6379> LPUSH imooc_list redis
(integer) 1
127.0.0.1:6379> LPUSH imooc_list mysql
(integer) 2
127.0.0.1:6379> LPUSH imooc_list memcache
(integer) 3
127.0.0.1:6379> LRANGE imooc_list 0 10
(empty list or set)
127.0.0.1:6379> LRANGE imooc_list 0 10
1) "memcache"
2) "mysql"
3) "redis"
```

启动 `./redis-server`

推荐学习 Redis 基本命令网址：<http://www.runoob.com/redis/redis-commands.html>

## 10.Git版本管理工具

### Git

- 安装：`yum install git`
- `git` 命令自动补全设置

### Git常用命令

更多教程

git config	git init
git clone	git remote
git fetch	git commit
git rebase	git push

## Git—最好用的分布式版本管理工具

1：安装

`sudo yum install git`

2：生成 SSH KEY

`ssh-keygen`

3：访问码云 (<https://gitee.com>)，配置 SSH Key

4：基本命令

<code>git config</code>	<code>git init</code>
<code>git clone</code>	<code>git remote</code>
<code>git fetch</code>	<code>git commit</code>
<code>git rebase</code>	<code>git push</code>

推荐学习 Git 基本命令网址：<http://www.runoob.com/git/git-tutorial.html>

11.

## 目录

- PHP基础环境配置
- PHP扩展安装
- Laravel5 环境配置
- Yii2 环境配置
- TP5环境配置
- 数据库网页工具：phpMyadmin
- 缓存redis管理工具：phpRedisadmin

## LAMP环境配置

缩写	解释
L	Linux 我们使用Centos
A	Apache WebServer
M	Mysql
P	FPM+PHP

## LNMP环境配置

缩写	解释
L	Linux 我们使用Centos
N	Nginx WebServer
M	Mysql
P	FPM+PHP

# PHP扩展安装

- Redis扩展
- Memcache扩展
- 数据库扩展

吉祥三宝搭建—Yii2、TP5、Laravel5

1: 默认版本太低 (5.4) 升级 php 到 5.6

1.1. 检查当前安装的 PHP 包

```
yum list installed | grep php
```

1.2 如果有安装的 PHP 包, 先删除他们

```
yum remove php.x86_64 php-cli.x86_64 php-common.x86_64 php-gd.x86_64 php-ldap.x86_64 php-mbstring.x86_64 php-mcrypt.x86_64 php-mysql.x86_64 php-pdo.x86_64
```

2: 配置源

```
sudo rpm -Uvh http://mirror.webtatic.com/yum/el7/epel-release.rpm
```

```
sudo rpm -Uvh http://mirror.webtatic.com/yum/el7/webtatic-release.rpm
```

如果想删除上面安装的包, 重新安装

```
rpm -qa | grep webtatic
```

```
rpm -e 上面搜索到的包即可
```

3: fpm 安装 和基本操作

```
sudo yum install php56w-fpm( 也可以 php55w-fpm php70w-fpm )
```

```
service php56w-fpm start/restart/stop
```

4: 安装 PHP 扩展

```
sudo yum install php56w.x86_64 php56w-cli.x86_64 php56w-common.x86_64 php56w-gd.x86_64 php56w-mbstring.x86_64 php56w-mcrypt.x86_64 php56w-mysql.x86_64 php56w-pdo.x86_64
```

yii2、tp5、laravel 请到对应下载目录中获取

三个框架



更多教程加qq337997

## Laravel5 运行环境配置

- 中文官网地址: <http://www.golaravel.com>
- 中文文档社区: [www.golaravel.com/laravel/docs/5.1/](http://www.golaravel.com/laravel/docs/5.1/)
- Laravel 5.1 下载: [down.golaravel.com/laravel/laravel-v5.1.11.zip](http://down.golaravel.com/laravel/laravel-v5.1.11.zip)

## Yii2 运行环境配置

- 中文官网地址: <http://www.yiichina.com/>
- 中文文档社区: [www.yiichina.com/doc/guide/2.0/intro-yii](http://www.yiichina.com/doc/guide/2.0/intro-yii)
- Yii2 下载: [github.com/yiisoft/yii2/releases/download/2.0.10/yii-basic-app-2.0.10.tgz](https://github.com/yiisoft/yii2/releases/download/2.0.10/yii-basic-app-2.0.10.tgz)

## ThinkPHP5 运行环境配置

官网地址：<http://www.thinkphp.cn/>

文档地址：[www.kancloud.cn/thinkphp/thinkphp5\\_quickstart](http://www.kancloud.cn/thinkphp/thinkphp5_quickstart)

应用项目：<https://github.com/top-think/think>

核心框架：<https://github.com/top-think/framework>

没有接着看下去了

12.java环境

### 目录

- Java安装
- Tomcat 安装
- Maven 安装
- Tomcat 和 Nginx配合
- 实例访问：java.imooc.test

### Java安装

```
sudo yum -y install java-1.8.0-openjdk*
```

安装完java之后查看一下版本：

```
[imooc@localhost ~]$ java -version
openjdk version "1.8.0_171"
OpenJDK Runtime Environment (build 1.8.0_171-b10)
OpenJDK 64-Bit Server VM (build 25.171-b10, mixed mode)
```

### Tomcat安装

• 官网：<http://tomcat.apache.org/>

• Tomcat9下载地址：

[http://apache.fayea.com/tomcat/tomcat-](http://apache.fayea.com/tomcat/tomcat-9/v9.0.0.M22/bin/apache-tomcat-9.0.0.M22.tar.gz)

[9/v9.0.0.M22/bin/apache-tomcat-9.0.0.M22.tar.gz](http://apache.fayea.com/tomcat/tomcat-9/v9.0.0.M22/bin/apache-tomcat-9.0.0.M22.tar.gz)

### Maven安装

• 官网：<http://maven.apache.org/>

• Maven下载地址：

[http://mirror.bit.edu.cn/apache/maven/maven-](http://mirror.bit.edu.cn/apache/maven/maven-3/3.5.0/binaries/apache-maven-3.5.0-bin.tar.gz)

[3/3.5.0/binaries/apache-maven-3.5.0-bin.tar.gz](http://mirror.bit.edu.cn/apache/maven/maven-3/3.5.0/binaries/apache-maven-3.5.0-bin.tar.gz)



\_\_\_\_\_