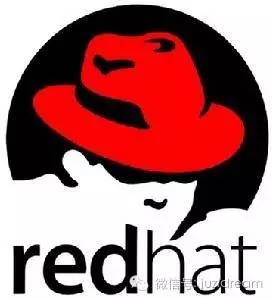
# IMG_256Linux Study Note



## Linux 内核

## Shell 与 shell script

## Linux 文件系统

## Linux 应用程序

## Linux 服务器搭建

## 1.Linux 应用程序

##### **1 用户账号**

* 先来查看系统里面有没有一个 jhadmin 用户

[root@testplan ~]*# grep --color jhadmin /etc/passwd*

jhadmin:x:501:501::/home/jhadmin:/bin/bash

* 查找前4个用户

[root@testplan ~]*# head -n 4 /etc/passwd*

root:x:0:0:root:/root:/bin/bash

bin:x:1:1:bin:/bin:/sbin/nologin

daemon:x:2:2:daemon:/sbin:/sbin/nologin

adm:x:3:4:adm:/var/adm:/sbin/nologin

* 详细说明
  + 账号名称
    - 就是账号，如 root 、bin
  + 密码
    - 实际密码放在 /etc/shadow 文件中
  + UID
    - 用户标识符
  + GID
    - 及组ID ，这个与 /etc/group 文件有个
  + 用户信息说明
    - 用来解释账号意义
  + 主文件夹
    - 及用户家目录，默认用户主文件夹在 /home/yourname 中
  + Shell
    - 与内核进行通信以完成用户操作

| **Id 范围** | **该 ID 特性** |
| --- | --- |
| 0（系统管理员） | 0 代表 root 用户，及系统管理员 |
| 1~499（系统账号） | 1~99 ：由 distribution 自行创建的系统账号；  100~499 ：若用户有系统账号，  可以使用的账号UID |
| 500~65535 | 给一般用户使用，目前Linux内核支持  4294967295这么多UID账号 |

* 查看前4个用户密码相关信息

[root@testplan ~]**# head -n 4 /etc/shadow**

root:$6$xkisBqYJJXxVdhK1$lVAPx7BFzO3sTVng2oOCvIWEqtjSubKiUgwVEY5Q8Q/12dsYx4TgPygkN6wBXQuDGu0h5p8yiPU2q..Hy4Gnp0:16985:0:99999:7:::

bin:\*:15980:0:99999:7:::

daemon:\*:15980:0:99999:7:::

adm:\*:15980:0:99999:7:::

* 相关说明
  + 账号名称
    - 与 /etc/passwd 文件账号对应
  + 密码
    - 经过编码加密 ，显示为特殊字符
  + 最近更动密码的日期
  + 密码不可被更动的天数
  + 密码需要重新更改的天数
  + 密码需要更改期限前的警告天数
  + 密码过期后的账号宽限时间
  + 账号失效日期
  + 保留
    - 最后一个字段是保留的 ^\_^

## Shell 与 shell script

##### echo 打印不同颜色shell

[root@testplan scripts]*# cat remindColor.sh*

**#!/bin/bash***#program:This is a about echo command different color display skill.*

*#date:2016-07-16*

*#author:racher*

*#email:1576768715@qq.com*

*#------------------font color--------------------------*

echo -e "\033[30m black font face. \033[0m"

echo -e "\033[31m red font face. \033[0m"

echo -e "\033[32m green font face. \033[0m"

echo -e "\033[33m yellow font face. \033[0m"

echo -e "\033[34m blue font face. \033[0m"

echo -e "\033[35m purple red font face. \033[0m"

echo -e "\033[36m skyblue font face. \033[0m"

echo -e "\033[37m white font face. \033[0m"

*#------------------font backgroung color---------------*

echo -e "\033[40;37m white on black.\033[0m"

echo -e "\033[41;37m white on red.\033[0m"

echo -e "\033[42;37m white on green.\033[0m"

echo -e "\033[43;37m white on yellow.\033[0m"

echo -e "\033[44;37m white on blue.\033[0m"

echo -e "\033[45;37m white on purple.\033[0m"

echo -e "\033[46;37m white on skyblue.\033[0m"

echo -e "\033[47;30m black on while.\033[0m"

## 3. linux 文件系统

### 3.1磁盘管理

##### **磁盘与存储设备**

##### **磁盘图片**



###### **磁盘的物理组成**

* 磁盘其实是有许多的盘片、机械手臂、磁头与主轴马达所组成。
* 圆形的盘片（主要记录数据的部分）；
* 机械手臂与机械手臂上的磁头（可读写盘片上的数据）；
* 主轴马达，可以转动盘片，让机械手臂的磁头在盘片上读写数据；
* 扇区（Sector）为最小的物理存储单位，每个扇区为512bytes；
* 将扇区组成一个圆，哪就是柱面（Cylinder），柱面是分区的最小单位；
* 第一个扇区最重要，里面有磁盘主引导记录（MastBootRecord，MBR）及分区表（Partition table）其中MBR占446bytes，而分区表占64bytes；
* 主分区与扩展分区最多可以有4个（硬盘限制）；
* 扩展分区最多只能有一个（操作系统限制）；
* 逻辑分区是由扩展分区持续分出来的分区；
* 能被格式化后作为数据访问的分区为主分区与逻辑分区，扩展分区无法格式化；
* 逻辑分区依操作系统而不同，在 Linux 系统中，IDE硬盘最多有59个逻辑分区（5号到63号），SATA硬盘则有11个逻辑分区（5号到15号）。

###### **磁盘接口**

* IDE
* SATA
* SCSI

##### **各种接口在 linux 中的文件名**

* /dev/sd[a-p][1-15] : 为 SCSI , SATA , USB , Flash 等接口的磁盘文件；
* /dev/sh[a-d][1-63] : 为 IDE 接口的磁盘文件名。

##### **Linux 安装目录大小建议**

| **挂载点** | **装 置** | **说明** |
| --- | --- | --- |
| / | /dev/hda1 | 15G |
| /boot | /dev/hda2 | 128MB |
| swap | /dev/hda3 | 大约内存大小或1.5~2倍 |
| /home | /dev/hda5 | 最大剩余空间 |
| /var | /dev/hda6 | 视服务器情况定 |

##### **分区**

1. 列出系统中的根目录所在的磁盘

[root@testplan ~]**# df /**

Filesystem 1K-blocks Used Available Use% Mounted on

/dev/sda2 7022156 5964440 701008 90% /

1. 查看帮助

[root@testplan ~]# fdisk /dev/sda

Command (m for **help**): **m**

Command **action**

**d** **delete** a **partition** //删除一个分区

**m** print this menu

**n** **add** a **new** **partition** //添加一个分区

**p** print the **partition** **table** //显示分区表

q quit **without** saving changes

w write **table** **to** disk **and** **exit** //将操作写入分区表

1. 查看目前系统的所有分区有哪些

[root*@testplan ~]# fdisk -l*

Disk /dev/sda: 8589 MB, 8589934592 **bytes**255 heads, 63 sectors/track, 1044 cylindersUnits = cylinders of 16065 \* 512 = 8225280 **bytes**Sector size (logical/physical): 512 **bytes** / 512 **bytes**I/O size (minimum/optimal): 512 **bytes** / 512 **bytes**Disk identifier: 0x000a6a2d

Device **Boot**  Start **End** **Blocks**  Id System

/dev/sda1 \* 1 26 204800 83 LinuxPartition 1 does not **end** on cylinder **boundary.**/dev/sda2 26 914 7134208 83 LinuxPartition 2 does not **end** on cylinder **boundary.**/dev/sda3 914 1045 1048576 82 Linux swap / Solaris

Device:设备文件名 Boot:是否为开机引导模块，/dev/sda1 开机引导模块 Start，End：表示分区从哪个柱面开始到那个柱面结束，可以决定分区大小 Blocks：以K为单位的容量 Id ,System：代表分区文件系统是啥

###### **同部分区表**

partprobe

###### **查看分区情况**

cat /proc/partitions

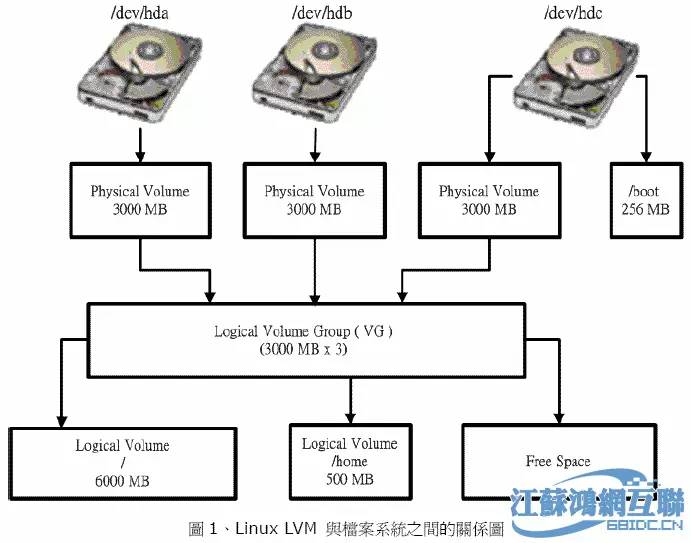
###### **磁盘格式**

mkfs -t ext3 | ext4 /dev/sdb1

### 3.2逻辑卷管理器

##### **1. 逻辑卷管理器 [logical volume manager]：**

###### **LVM 可以整合多个物理分区在一起，让这些分区看起来就像是一个磁盘一样，而且，还可以在将来其它的物理分区或将其从这个LVM管理的磁盘当中删除。LVM 重点在于可以单独调整文件系统的容量。**



##### **2. who ？ PV 、VG 、PE 、LV ：**

* Physical Volume，PV，物理卷

##### **指磁盘分区或从逻辑上与磁盘分区具有同样功能的设备（如RAID），是LVM的基本存储逻辑块，但和基本的物理存储介质（如分区、磁盘等）比较，却包含有与LVM相关的管理参数。**

* Volume Group，VG，卷用户组

##### **类似于非LVM系统中的物理磁盘，其由一个或多个物理卷PV组成。可以在卷组上创建一个或多个LV（逻辑卷）。**

* Physical Extend，PE,物理扩展卷

##### **每一个物理卷PV被划分为称为PE（Physical Extents）的基本单元，具有唯一编号的PE是可以被LVM寻址的最小单元。PE的大小是可配置的，默认为4MB。所以物理卷（PV）由大小等同的基本单元PE组成。这个 PE 有点像文件系统里面的 block 大小。**

* Logical Volume，LV,逻辑卷

##### **逻辑卷LV也被划分为可被寻址的基本单位，称为LE。在同一个卷组中，LE的大小和PE是相同的，并且一一对应。LV 最后可以被格式化使用的文件系统了。**

* Physical Storage Media，PSM,物理存储介质

##### **指系统的物理存储设备：磁盘，如：/dev/hda、/dev/sda等，是存储系统最底层的存储单元。**

##### **3. 创建和管理 LVM**

* 创建物理卷 PV
  + 与 PV 相关命令
    - pvcreate：初始化一个磁盘分区为 LVM
    - pvscan：显示目前系统具有PV的磁盘
    - pvdisplay：显示 PV 属性
    - pvremove：删除 PV
  + 检查有无 PV 在系统上，然后新建 /dev/sdb1~/dev/sdb4 位 PV 格式

Last login: Sun Aug 7 01:58:59 2016 from 192.168.88.1

Welcome To JuzlDream ^\_^

[root@testplan ~]**# pvscan**

No matching physical volumes found

[root@testplan ~]**# pvcreate /dev/sdb1**

Physical volume "/dev/sdb1" successfully created

[root@testplan ~]**# pvcreate /dev/sdb2**

Physical volume "/dev/sdb2" successfully created

[root@testplan ~]**# pvcreate /dev/sdb3**

Physical volume "/dev/sdb3" successfully created

[root@testplan ~]**# pvcreate /dev/sdb5**

Physical volume "/dev/sdb5" successfully created

[root@testplan ~]**# pvscan**

PV /dev/sdb1 lvm2 [15.66 MiB]

PV /dev/sdb2 lvm2 [7.84 MiB]

PV /dev/sdb3 lvm2 [7.84 MiB]

PV /dev/sdb5 lvm2 [7.81 MiB]

Total: 4 [39.16 MiB] / in use: 0 [0 ] / in no VG: 4 [39.16 MiB]

* 创建卷组 VG
  + 与vg相关的命令
    - vgextend：
    - vgeduce：
    - vgchange：
    - vgromve:
  + 建立 vg ，并制定 pe 为 16 MB.

[root@testplan ~]*# vgcreate -s 16M testplanVG /dev/sdb1*

Volume **group** **"testplanVG**" successfully created

* 创建逻辑卷 LV
  + 与lv相关的命令
    - lvextend：
    - lvreduce：
    - lvremove：
    - lvresize：
  + 将整个 vg 全部分配给 lv ，注意 pe 个数

[root@testplan ~]**# lvcreate -l 3 -n testplanLV testplanVG**

Logical volume "testplanLV" created

[root@testplan ~]**# ls -l /dev/testplanVG/testplanLV**

lrwxrwxrwx. 1 root root 7 Aug 7 03:14 /dev/testplanVG/testplanLV -> ../dm-0

* 文件系统
  + 格式化、挂载、查看

[root*@testplan ~]# mkfs -t ext3 /dev/testplanVG/testplanLV* mke2fs 1.41.12 (17-May-2010)Filesystem label=OS type: Linux**Block** size=1024 (log=0)Fragment size=1024 (log=0)**Stride=0 blocks, Stripe** width=0 **blocks**3072 inodes, 12288 **blocks**614 **blocks** (5.00%) reserved for the super userFirst **data** **block=1**Maximum filesystem **blocks=12582912**2 **block** groups8192 **blocks** per group, 8192 fragments per group1536 inodes per groupSuperblock **backups** stored on **blocks:**

8193

Writing inode tables: done Creating journal (1024 **blocks):** doneWriting superblocks **and** filesystem accounting information: done

This filesystem will **be** automatically checked every 20 mounts or180 days, whichever comes first. Use tune2fs -c or -i to override.

[root@testplan testLVM]**# mkdir /mnt/testLVM**

[root@testplan testLVM]# mount /dev/testplanVG/testplanLV /m

### 3.3磁盘配额（Quota）

所谓磁盘配额，就是设置某一个文件系统（分区）内的用户可使用空间大小。

#### **用途**

* ****WWW**** server 服务器，针对每个人的网页空间容量限制
* ****Mail**** server 每个人的邮件空间限制
* ****file**** server 每个人的最大可用的网络硬盘空间；本次试验就是为公司搭建一塔文件服务器。

#### **实现步骤如下：**

1. 创建3个用户

**#!/bin/bash**

*#program:Create 3 users and specify the user group.*

*#date:2017-02-27*

*#author:racher*

*#email:1576768715@qq.com*

groupadd myquotagrp

**for** username **in** myquota1 myquota2 myquota3

**do**

useradd -g myquotagrp $username

echo "123.com" | passwd --stdin $username

**done**

1. 文件系统支持
   1. 检查/home是否是独立的文件系统
   2. 用mount查询一哈/hoem的文件系统格式

[root@localhost home]**# df -h**

Filesystem Size Used Avail Use% Mounted on

/dev/sda5 4.8G 2.0G 2.6G 44% /

tmpfs 372M 0 372M 0% /dev/shm

/dev/sda1 190M 33M 147M 19% /boot

/dev/sda2 4.0G 33M 4.0G 1% /home

[root@localhost home]*# mount | grep home*

/dev/sda2 **on** /home type xfs (rw)

1. 手动加入quota支持

[root@localhost home]*# mount -o remount,usrquota,grpquota /home*

[root@localhost home]*# mount | grep home*

/dev/sda2 on /home type xfs (rw,usrquota,grpquota)

1. 开机自动挂载
   1. /etc/fstab

/dev/sda2 /home ext4 defaults,usrquota,grpquota 1 2

1. 生成配置文件

[root@rhel6 test]**# quotacheck -avug**

[root@rhel6 test]**# ls -a -l**

total 40

drwxr-xr-x 3 root root 4096 Feb 28 18:25 .

drwxr-xr-x. 30 root root 4096 Feb 26 14:07 ..

-rw------- 1 root root 6144 Feb 28 18:25 aquota.group

-rw------- 1 root root 6144 Feb 28 18:25 aquota.user

drwx------ 2 root root 16384 Feb 28 18:18 lost+found

1. 启动quota

[root@rhel6 test]*# quotaon -auvg*

/dev/sdb1 [/test]: **group** **quotas** turned on

/dev/sdb1 [/test]: **user** **quotas** turned on

quota 设置大小

命令编辑：

setquota -u jhadmin 100 150 0 0 /home/

内容设置：

[root@rhel6 test]*# edquota -u zz*

Disk quotas for **user** **zz** (uid 16777256):

Filesystem blocks soft hard inodes soft hard

/dev/sdb1 0 500000 600000 0 0 0

修改宽限时间

[root@rhel6 test]**# edquota -t**

Grace period before enforcing soft limits **for** users:Time units may be: days, hours, minutes, or seconds

Filesystem Block grace period Inode grace period

/dev/sdb1 3days 7days

~

~

1. 测试

[zz@rhel6 **test**]$ dd **if**=/dev/zero of=gkd **bs**=10M **count**=100

dd: writing `gkd': Disk quota exceeded

1+0 records **in**

0+0 records **out**

0 bytes (0 B) copied, 0.00526488 s, 0.0 kB/s

1. 查询使用情况

[root@rhel6 /]*# quota -uvs zz*

Disk quotas **for** **user** zz (uid 16777256):

Filesystem blocks quota **limit** grace files quota **limit** grace

/dev/sdb1 586M\* 489M 586M 6days 2 0 0

1. 复制zz用户给jhadmin

[root@rhel6 /]*# edquota -p zz jhadmin123*

[root@rhel6 /]*# quota -uvs jhadmin123*

Disk quotas **for** **user** jhadmin123 (uid 16777267):

Filesystem blocks quota **limit** grace files quota **limit** grace

/dev/sdb1 0 489M 586M 0 0 0

1. 查询所有用户情况

[root@rhel6 /]**# repquota -auvs**

\*\*\* Report **for** user quotas on device /dev/sdb1

Block grace time: 7days; Inode grace time: 7days

Block limits File limits

User used soft hard grace used soft hard grace

----------------------------------------------------------------------

root -- 20 0 0 2 0 0

nobody -- 0 489M 586M 0 0 0 **#user1 +- 586M 489M 586M 6days 2 0 0 #user2 -- 0 489M 586M 0 0 0 #user3 -- 0 489M 586M 0 0 0 #user4 -- 0 489M 586M 0 0 0**

## 5.Linux 服务器搭建

#### **5.1 Samba 之 AD 域控环境搭建**

1. 准备工作：

* Linux 服务器（redhat 6.8） 用于安装 samba AD IP：192.168.88.10
* Windows 客户端 用于加域测试
* Linux 端需要配置YUM源

2.下载samba软件包，本次采用源码安装方式

* 下载地址：<https://www.samba.org/samba/download/>

3.安装依赖库文件

* yum install -y perl gcc libacl-devel libblkid-devel gnutls-devel readline-devel python-devel gdb pkgconfig krb5-workstation zlib-devel setroubleshoot-server libaio-devel setroubleshoot-plugins policycoreutils-python libsemanage-python setools-libs-python setools-libs popt-devel libpcap-devel sqlite-devel libidn-devel libxml2-devel libsepol-devel libattr-devel keyutils-libs-devel cyrus-sasl-devel cups-devel bind-utils libxslt docbook-style-xsl openldap-devel

4.开始编译安装samba

./configure --prefix=/usr/local/samba --enable-selftest

make

make test

make install 4.环境变量

vim /root/.bash\_profile

PATH=$PATH:$HOME/bin:/usr/local/samba/bin:/usr/local/samba/sbin

export PATH

5.配置Samba AD

* samba-tool domain provision --server-role=dc --use-rfc2307 --dns-backend=SAMBA\_INTERNAL --realm=[xadev.com](http://xadev.com) --domain=xadev --adminpass=JHadmin999

1. 配置DNS

/etc/resolv.conf

domain [xadev.com](http://xadev.com)

nameserver 10.99.0.1

1. 配置kerberos

/etc/krb5.conf

rm /etc/krb5.conf ln -sf /usr/local/samba/private/krb5.conf /etc/krb5.conf

1. 启动Samba AD

samba

6.把客户端加入AD环境中

* 安装 samba 、krb5

yum **install** samba

yum **install** krb5-devel krb5-libs pam\_krb5 krb5-workstation

运行setup 加域







脚步配置加域

* + /etc/nsswitch.conf

passwd: files winbindshadow: files winbindgroup: files winbind

* + /etc/krb5.conf

[logging]

default = FILE:/var/log/krb5libs.log

kdc = FILE:/var/log/krb5kdc.log

admin\_server = FILE:/var/log/kadmind.log

[libdefaults]

default\_realm = XADEV.COM

dns\_lookup\_realm = **true**

dns\_lookup\_kdc = **true**

ticket\_lifetime = 24h

renew\_lifetime = 7d

forwardable = **true**

[realms]

EXAMPLE.COM = {

kdc = kerberos.example.com

admin\_server = kerberos.example.com

}

XADEV.COM = {

kdc = jhserver.xadev.com

default\_domain = xadev.com

admin\_server = jhserver.xadev.com\_\_\_\_

kdc = jhserver.xadev.com

}

[domain\_realm]

xadev.com = XADEV.COM

.xadev.com = XADEV.COM

* + /etc/samba/smb.conf

[**global**]

workgroup = XADEV

password server = jhserver.xadev.com

realm = XADEV.COM

security = ads

idmap config \* : **range** = 16777216-33554431

winbind separator = /

template homedir = /home/%**U**

template **shell** = /bin/bash

winbind **use** default domain = true

winbind offline logon = true

[homes]

comment = Home Directories

browseable = **no**

writable = yes

; valid users = %S

; valid users = MYDOMAIN\%S

path = /home/%**U**

valid users = XADEV.COM/%**U**

**read** only = **No**

browseable = **No**

root preexec = /root/mkhome.**sh** %**U** %**G**

create mode = 0777

directory mode = 0777

[printers]

comment = All Printers

path = /**var**/spool/samba

browseable = **no**

guest ok = **no**

writable = **no**

printable = yes

* + 加域

/usr/bin/net join -w XADEV -S jhserver.xadev.com -U Administrator

* + 自动创建用户目录 /etc/pam.d/sysconf-auth

session required pam\_mkhomedir.so silent skel=/etc/skel umask=0077

[WiKi Samba](https://wiki.samba.org/index.php/Setting_up_Samba_as_an_Active_Directory_Domain_Controller)

* 本次实验环境为：CentOS 6.5

#### **ftp 服务器简介**

##### **FTP 是File Transfer Protocol（文件传输协议）的英文简称，而中文简称为“文传协议”。用于Internet上的控制文件的双向传输。同时，它也是一个应用程序（Application）。基于不同的操作系统有不同的FTP应用程序，而所有这些应用程序都遵守同一种协议以传输文件。在FTP的使用当中，用户经常遇到两个概念："下载"（Download）和"上传"（Upload）。**

##### **安装步骤**

1. 检查vsftp 软件是否安装

[root@testplan ~]**# rpm -qa | grep vsftp**

1. 安装ftp软件

[root@testplan ~]**# yum install vsftp**

1. 启动服务

[root@localhost ~]**# service vsftpd start**

Starting vsftpd **for** vsftpd: [ OK ]

相关配置

* + 简单配置

anonymous\_enable=NO *//非匿名*

anon\_upload\_enable=YES *//上传*

anon\_mkdir\_write\_enable=YES *//创建*

[root@localhost ftp]**# passwd ftp**

Changing password **for** user ftp.

*//设置ftp用户密码*

ftp：*//192.168.88.88 //登录*

限制重要系统用户不能登录ftp

[root@localhost ~]*# cat /etc/vsftpd/ftpusers # Users that are not allowed to login via ftp*

root

bin

daemon

nobody

ftp

限制系统用户锁定在家目录

chroot\_list\_enable=YES*# (default follows)*

chroot\_list\_file=/etc/vsftpd/chroot\_list

cut -d : -f 1 /etc/passwd >> /etc/vsftpd/chroot\_list

利用ftp用户策略允许登录ftp的系统用户

userlist\_enable=**YES**serlist\_deny=**NO**userlist\_file=/etc/vsftpd/user\_list

访问测试

<ftp://192.168.88.88>

[参考文献](http://www.linuxidc.com/Linux/2015-06/118442.htm)

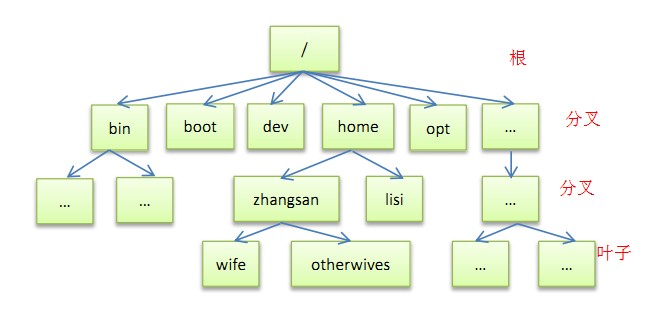
IMG_256

# Linux 服务器操作系统IMG_256

Red Hat 是 Linux 的一个版本

CentOS (社区企业操作系统（Community Enterprise Operating System）是 Red Hat 的一个版本

Linux 树形图：



**/bin** 二进制可执行命令

**/dev** 设备特殊文件  
**/etc** 系统管理和配置文件  
**/etc/rc.d** 启动的配置文件和脚本  
**/home** 用户主目录的基点，比如用户user的主目录就是/home/user，可以用~user表示  
/**lib** 标准程序设计库，又叫动态链接共享库，作用类似windows里的.dll文件  
/**sbin** 超级管理命令，这里存放的是系统管理员使用的管理程序  
/**tmp** 公共的临时文件存储点  
/**root** 系统管理员的主目录  
/**mnt** 系统提供这个目录是让用户临时挂载其他的文件系统  
/**lost+found**这个目录平时是空的，系统非正常关机而留下“无家可归”的文件（windows下叫什么.chk）就在这里  
/**proc** 虚拟的目录，是系统内存的映射。可直接访问这个目录来获取系统信息。  
/**var** 某些大文件的溢出区，比方说各种服务的日志文件  
/**usr** 最庞大的目录，要用到的应用程序和文件几乎都在这个目录，其中包含：

/usr/x11R6 存放x window的目录  
/usr/bin 众多的应用程序  
/usr/sbin 超级用户的一些管理程序  
/usr/doc linux文档  
/usr/include linux下开发和编译应用程序所需要的头文件  
/usr/lib 常用的动态链接库和软件包的配置文件  
/usr/man 帮助文档  
/usr/src 源代码，linux内核的源代码就放在/usr/src/linux里  
/usr/local/bin 本地增加的命令  
/usr/local/lib 本地增加的库根文件系统



exit 退出登录状态

shutdown now 关机

dir ls 均可查看当前目录下的东西

ll 除了可看见，文件名，还可看到很多其它属性。

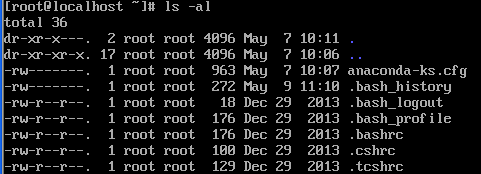
ll -a 连隐藏文件也可看见

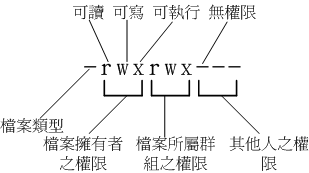
Linux 下，隐藏文件的文件名均是以 . .. 开头。

. 目录本身

.. 父目录

文件属性中





第一栏代表这个文件类型与权限(permission) d -

第二栏表示有多少档名连结到此节点(i-node)

第三栏表示这个文件(或目录)的『拥有者账号』

第四栏表示这个文件的所属群组

第五栏为这个文件的容量大小，默认单位为bytes

第六栏为这个文件的建档日期或者是最近的修改日期

第七栏为这个文件的档名

r read 读 4

w write 写 2

x execute 执行 1

chgrp user file 改变文件所属群组

chown user file 将此文件的拥有者，修改为user。

chmod -R 修改权限（符号法，数字法） r 表示修改到最下层

r 对文件而言，是查看文件中的内容，对目录而言，是查看目录中的文件列表。

x 对文件而言，是运行，对目录而言，是进入一个目录。

w 对文件而言，是修改文件内容。对目录而言，是在目录中创建或者删除文件的权限。

Linux 系统内文件身份：拥有者(user)、群组(group)、其他人(others)。三种权限（rwx）

只要对文件夹有写权限，修改里面的文件，就不是问题。

u owner

g group

o others

+ 添加权限

= 直接赋值

- 删除权限

r read

w write

x execute 执行 目录的x代表的是用户能否进入该目录成为工作目录的用途！

d 代表文件夹

- 常规文件

date 查看系统时间

Cal 日历

cal 2007 显示2007年的日历表

/ 根目录

~ 家目录 ，每个用户的家目录，都不一样。

root 的家目录位于，/root

其它用户的家目录位于，/home 里面那个与其用户名相同的文件夹。

cd 进入个人主页

cd 用来进入某个文件夹

Cd \ 根目录

Cd .. 返回上一级目录

Cd ../.. 返回上两级目录

Cd - 返回上次所在一目录

pwd 显示工作路径

ls -f 查找目录中文件

ls -l 显示文件和目录的详细资料   
ls -a 显示隐藏文件

tree 显示树形结构

ls \*[0-9]\* 显示包含数字的文件 目录

mkdir dir1 创建一个叫做 'dir1' 的目录'   
mkdir dir1 dir2 同时创建两个目录

rmdir dir1 删除dir1d

mv dir1 new\_dir 重命名/移动 一个目录

clear 清屏

useradd user1 添加一个用户

passwd user1 为user1设置密码，若没有参数，则代表修改自己的密码，只有root才可以去修改其它用户的密码。

# root

$ 非root

Useradd adduser 一样的

userdel user1 //将此用户删除

userdel -r user1 //参数 -r 表示，连同家目录，一起删除。

遇到假死 ctrl+c推出即可

cat //查看文件内容

~/.bash\_history //记录了此用户，使用过的历史命令。

tac //从下往上看

-n //查看前n行，n默认为10。

tail -n //查看最后n行，n默认为10。

/etc/\*-release //这些文件记录了，操作系统版本相关信息。

输入命令时，可以使用 \* 代替，任意多个任意字符，？代替一个任意字符。

mkdir //创建目录 , -p 可以一次创建多个目录

rmdir //删除一个空目录

mkdir --help //显示帮助信息，详细列出了，此命令的用法。

rm //用来删除文件

echo //可在屏幕输出字符串

字符串 > filename 写入某文件，覆盖此文件原有信息。

两个重定向，操作符。

> 会覆盖

>> 追加

echo “string” >a a文件里写string

echo ‘string’>> a a文件里写string

Cat a 查看a文件内容

touch filename.txt 新建一个txt文件;  
cat 你要写入的内容 >filename.txt 可以写txt文件, 但原有的内容会先被清除;  
cat 你要写入的内容 >> filename.txt 可以写txt文件,新的内容追加到文件末尾

rm -r //可用来删除非空文件夹，若要求你确认，则可加参数f，强制执行，则不需要输入y，太麻烦。

--help //查看帮助

-i //可再确认一下，更加慎重一些。

cp //可实现文件的复制 -r 加了这个参数，才可以复制文件夹。

mv //此命令用来移动文件或文件夹，此命令也常用来重命名。

eg: mv test.log test1.txt 文件改名

mv test.log . 文件移动到当前目录下

mv a.txt b.txt a 将a.txt b.txt 文件移动到a文件夹下

grep // 管线命令，可与任何命令结合起来使用，按行进行过滤。

eg: ll | grep abc >> a.txt

ls|grep \*

ctrl+c //终止命令的执行

find / -name filename //整个系统搜索，指定文件名的文件，也包括目录。

chkconfig 看到一些服务

service 对服务进行操作

vi linux下的记事本软件

i 从普通模式切换到插入状态

shift+; 从普通模式切换到指令模式

esc 任何模式切换到普通模式，均通过此方法。

指令模式下的指令

q 退出

w 保存

wq 保存再退出

若输入了命令，不管用，末尾加 ! 则一定会管用的。

dd 在普通模式下，删除光标所在的行。

插入状态下，insert键，可实现 insert/replace 两种模式的替换。

网卡配置文件

vi /etc/sysconfig/network-scripts/ifcfg-eno16777736 // ip addr 查看device

DEVICE=eno16777736

IPADDR=192.168.1.\*

GATEWAY=192.168.1.1

service 去重新启动 network 这个服务，即可连通局域网

配置 DNS

vi /etc/resolv.conf

nameserver 192.168.1.1

chkconfig 此命令可查看到本机的一些，基本服务。

service 服务名 status //查看此服务当前的状态

service 服务名 start //启动此服务

service 服务名 reload/restart //重启此服务

chkconfig network off/on //关闭 或 打开 此服务的开机自动启动

vim vi 的加强版，主要是写代码的时候，有色彩提示。

notepad notepad++ 一样的关系

yum info 软件名 //查看此软件的版本等相关信息

yum install 软件名 //在线安装某软件

yum remove 软件名 //卸载某软件

date //查看系统时间

tzselect //修改时区命令，稍后再作研究。

git //版本控制软件，团队做项目，必须的东西。（TortoiseGit-1.8.14.0-32bit）

新建仓库步骤：

mkdir test //选择一个合适的地方，创建一个空目录：

cd test

pwd

git init //第二步，通过git init命令把这个目录变成Git可以管理的仓库：

git clone ssh://user@192.168.1.106/tmp/test test //将仓库 clone 到本机

第三步，仓库路径下config ：

[receive]

denyCurrentBranch = ignore

添加了这两行之后，root，就可以 push 了。

chown -R user1 repos //执行此句话之后，user1 才有权限 push 。

git reset --hard //将服务器的仓库，更新到最新状态，此命令一般不用，服务器是否更新，并不影响仓库的使用。

目前流行版本控制软件有

svn ：集中式的

git ：分布式的

ssh user1@192.168.1.153 //此命令，用来从一台服务器，跳转至另外一台服务器。

Linux nginx 安装：

http://nginx.org/packages/ //下载 nginx yum 资源的地址

ssh 导入nginx-release-centos-7-0.el7.ngx.noarch.rpm 下载包；

rpm -i nginx-release-centos-7-0.el7.ngx.noarch.rpm //执行此命令后，方可 yum install nginx

yum install nginx //安装 nginx

nginx :/usr/share/nginx/html/

chkconfig 服务名 //此命令，可查看某服务，是否会开机自动启动，若为enabled表示开机会自动启动，否则不会。

chkconfig 服务名 on/off //打开/关闭，此服务的，开机自动启动。

reboot //重启centos

Linux-php 安装：

yum install php //安装PHP （/usr/share/nginx/html）

vim /etc/rc.local 添加一行：

/usr/bin/php-cgi -b9000& //设置 php 开机自动执行 ， & 表示此命令，转入后台执行。

chmod u+x rc.local //此文件必须具备执行权限，才会运行。（cd /etc/rc.d rc.local在rc.d文件夹下）

配置nginx 、etc/nginx/conf.d

location / {

root /usr/share/nginx/html/self; # 指明本服务器的根目录

index index.html index.htm index.php; # 默认主页设置]

}

location ~ \.php$ { #　php　交互说明

# root html;

fastcgi\_pass 127.0.0.1:9000; # 通用网关接口

# fastcgi\_index index.php;

fastcgi\_param SCRIPT\_FILENAME /usr/share/nginx/html/self$fastcgi\_script\_name;

# 接口交互文件路径，使用绝对路径，因为配置文件与 localhost 所在位置距离较远。

include fastcgi\_params; # 保证能在 url 中传递参数

}

Linux-Apache:

yum install httpd

Service httpd start

vi /etc/httpd/conf/httpd.conf

Linux-mysql：

yum install mariadb-server //安装 mariadb ，它是 mysql 的一个版本。

方法一：  
在mysql系统外，使用mysqladmin  
# **mysqladmin -u root -p password "test123"**  
Enter password: 【输入原来的密码】

mysqldump -uuser1 -p123456 db1 > ~/db1.sql // mysql 备份

Service mariadb status

Service mariadb reload/restart

Chkconfig mariadb-server on/off

mysql -p123456 -uuser1 [dbname] < ~/db1.sql // 在 centos 系统下，还原。dbname 这

delete from user where user<>'root'||host<>'localhost'; //只留此一行，只保留root用户。

flush privileges; //刷新权限

update user set password=password(123456); //修改密码

mysql.user 中的 host 字段，若设置为%，表示任何 ip 均可连接进来。

grant all privileges on stu.\* to user identified by '123456'; //创建一个密码位123456的user用户，并对所有数据库有查询、插入、修改、删除权限 All(select,insert,update,delete)

SELECT User, Password, Host FROM user; -- 查看现有用户,密码及允许连接的主机

grant all on \*.\* to root@"%" identified by "root"; 管理员权限

revoke all on books.\* from Bob; 取消bob在books数据库任何权限

Delete FROM user Where User='test' and Host='localhost';

删除账户及权限：>drop user 用户名@'%';

　　　　　　　　>drop user 用户名@ localhost;

service 服务名 status //查看此服务当前的状态

service 服务名 start //启动此服务

service 服务名 reload/restart //重启此服务

chkconfig network off/on //关闭 或 打开 此服务的开机自动启动

Smarty -[模板引擎](http://baike.baidu.com/view/4258079.htm" \t "http://baike.baidu.com/_blank) 它分离了[逻辑](http://baike.baidu.com/view/1838.htm" \t "http://baike.baidu.com/_blank)[代码](http://baike.baidu.com/view/41.htm" \t "http://baike.baidu.com/_blank)和外在的内容，提供了一种易于管理和使用的方法，用来将原本与[HTML](http://baike.baidu.com/view/692.htm" \t "http://baike.baidu.com/_blank)代码混杂在一起PHP代码逻辑分离。

克隆地址：git clone <http://github.com/smarty-php/smarty.git>

**require\_once** 'smarty/Smarty.class.php';

$sm=**new** Smarty();

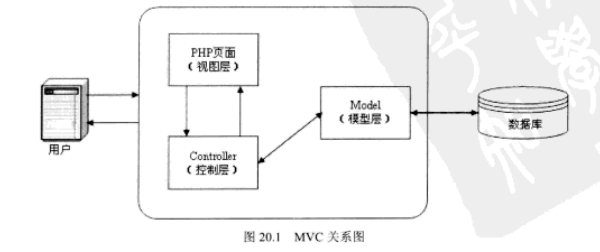
$sm->assign('name','张三');

$sm->assign('arr',**array**('name'=>'李四','age'=>28));

$sm->display('smt/index.tpl');

ThinkPHP

MVC设计模型的各自功能及相互关系



Model 一个实体对象或业务逻辑，具有重用性、可扩展性

View 应用程序与用户之间的交互界面

Controller 对应用请求进行控制，选择调用那些视图或模型

nginx 配置

location ~ \.php(/|$) {

set $path\_info '';

if ($fastcgi\_script\_name ~ ^(/.+\.php)(/.+)$) {

set $path\_info $2;}

fastcgi\_param PATH\_INFO $path\_info;

Controller 下index 地址：

<http://127.0.0.1/think/index.php/home/index/index/index.php>



http://localhost/index.php

http://localhost/index.php/Home

http://localhost/index.php/Home/Index

http://localhost/index.php/Home/Index/Index

数据库访问：

**public function** mariadb(){

$this->data=M()->query('select \* from phpclass');

$this->display();

}

HOME->conf-config.php 添加：

<?php

**return array**(

'db\_type' => 'mysql',

'db\_host' => 'localhost',

'db\_name' => 'stu1',

'db\_user' => 'root',

'db\_pwd' => '\*\*\*'

);

View ：

{$data}

<volist name=*'data'* id=*'vo'*>

<tr>

<td>{$vo.name}</td>

<td><if condition=*'$vo.sex ==0'*>男 <else />女 </if></td>

<td>{$vo.age}</td>

<td>{$vo.txt}</td>

</tr>

</volist>

events {}

http {

include mime.types;

server {

location / {

index index.htm index.html index.php;

if (!-e $request\_filename){

rewrite ^/think/(.\*) /think/index.php/home/$1;

}

}

location ~ \.php(/|$) {

fastcgi\_pass 127.0.0.1:8080;

fastcgi\_param SCRIPT\_FILENAME D:/sqlphp/nginx-1.9.0/html$fastcgi\_script\_name;

include fastcgi\_params;

set $path\_info '';

if ($fastcgi\_script\_name ~ ^(/.+\.php)(/.+)$) {

set $path\_info $2;

}

fastcgi\_param PATH\_INFO $path\_info;

}

}

<include file='pub:head' /> //加载公共页面

二次开发

论坛 discuz （康盛） , phpwind

博客 ucenter home（康盛）

微博 thinksns，iwebsns

CMS（内容管理系统） dedeCMS（织梦），ecms（帝国），supesite（康盛），phpcms（盛大，酷6），wordpress。

网店 ecshop （商派） ， iwebshop 。

网城 ecmall （商城），iwebmall 。

多用户整合 uc center （用户中心），康盛，同步注册，单点登录，退出。

1. 安装 ucenter

2. 安装 ucenter home

3. 安装 ecshop

4. cls\_mysql.php 最顶部添加一行代码

date\_default\_timezone\_set('PRC');

才可安装

5. cls\_template.php 422行

改为：

$arr=explode(' ', $tag);

$tag\_sel = array\_shift($arr);

1. lib\_main.php 1329行

改为：

$arr=explode('.', $tmp);

$ext = end($azzzzzzrr);

7. 在 data/config.php 末尾添加：

应用的 UCenter 配置信息: //此信息从 ucenter copy 过来。

8. 根据页面提示，修改includes\modules\integrates\ucenter.php文件中 login,add\_user,set\_cookie 函数的声明。

9. uhome common.php 第115行之后插入：

$\_SCONFIG['uc\_status'] = 1;

10. ecshop includes\modules\integrates\ucenter.php

//$this->ucdata = uc\_call("uc\_user\_synlogout"); //同步退出 ，秦，注释此行。

//qin begin

$this->ucdata = uc\_call("uc\_user\_synlogout",array());

//qin end