



## 第一章：linux服务器系统的介绍与整套课程大纲

### 1.1 课程大纲总览

简介：介绍课程知识目录大纲

### 1.2 linux系统的简介介绍

简介：介绍linux系统

- Linux简介：

Linux 是一个免费使用和自由传播的类 Unix 操作系统，是一个基于 POSIX 和 UNIX 的多用户、多任务、支持多线程和多 CPU 性能稳定的网络操作系统。

linux的内核最初是由一个芬兰的大学生因为个人兴趣爱好而编写的，这个大学的名字叫林纳斯·托瓦兹 (Linus Torvalds)

目前市面上较知名的发行版有：Ubuntu、RedHat、CentOS、Debian、Fedora、SuSE、OpenSUSE、Arch Linux、SolusOS 等。其中这个CentOS是一个企业级的发行版，特别适合对稳定性，可靠性和功能要求较高用户

- 适合人群：

适合零基础，运维人员，程序开发人员，DBA数据库管理员等等所有从事IT行业的人。

- 学后达到的水平：

入门偏中等水平

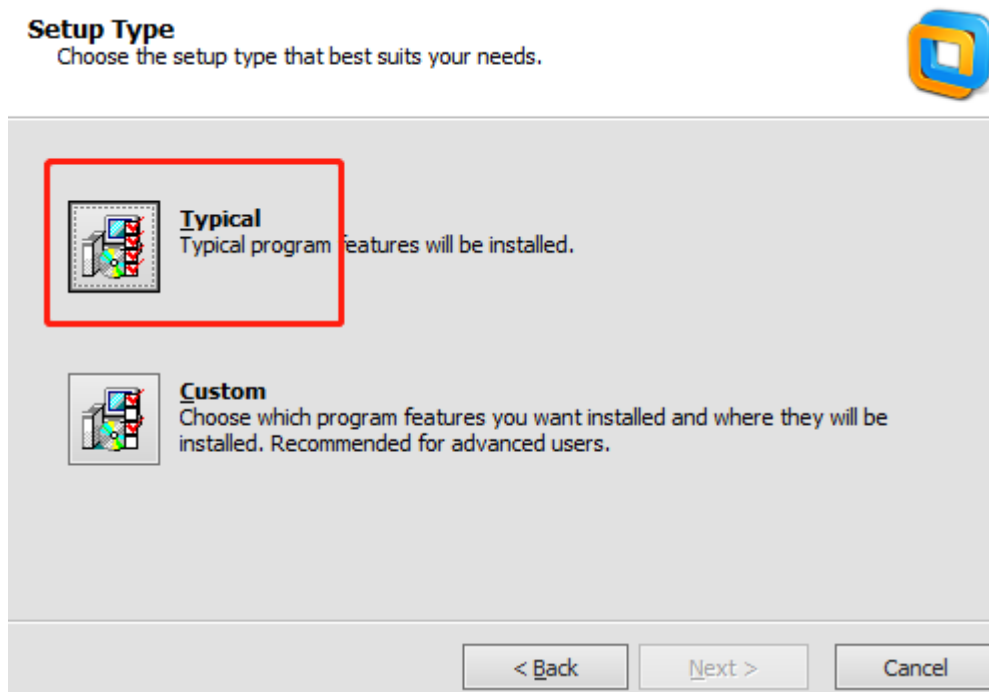
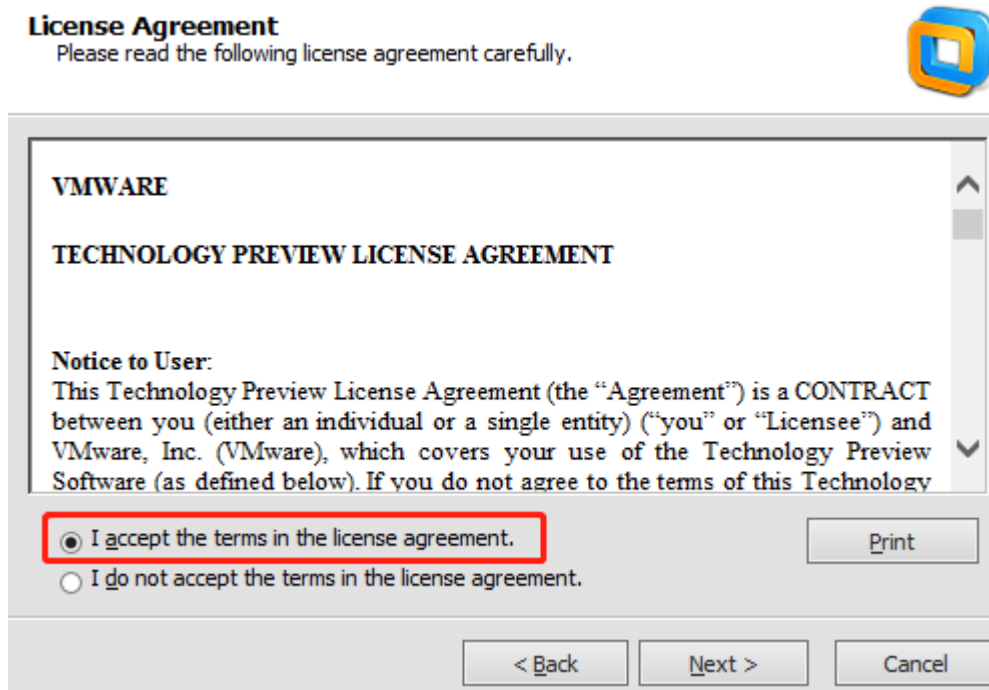


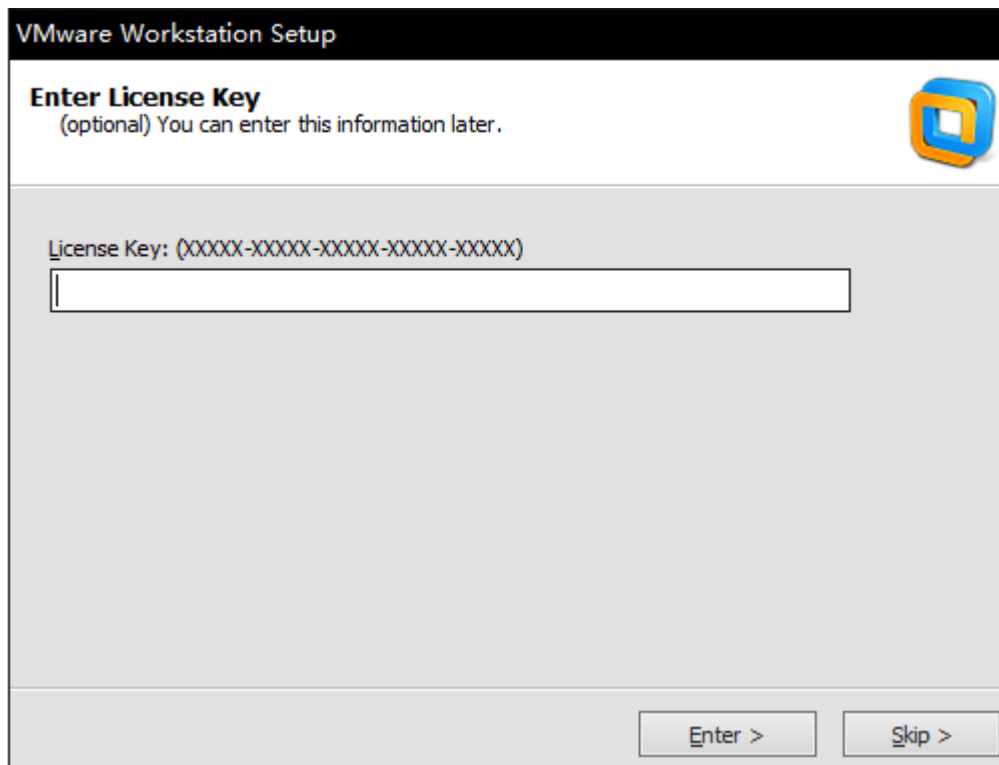
## 第二章：如何拥有一台属于自己的linux服务器

## 2.1 window 7环境下vmware虚拟机的安装

简介: vmware虚拟机的介绍与安装

- 安装步骤：双击安装，点下一步，需要选择的地方如下图





## 2.2 linux CentOS7系统的安装

简介：虚拟机CentOS7服务器的安装

## 2.3 vmware虚拟机的三种网络模式的设置介绍

简介：介绍网络设置

VMware 虚拟机的对应三种网络模式

- VMnet0 虚拟交换机：Bridged桥接模式

特点：

- 默认使用VMnet0，不提供DHCP服务（DHCP服务是指由服务器控制的一段IP地址范围，当客户机登录服务器时会自动获取服务器分配的IP地址与子网掩码）
- 虚拟机与外部主机需要在同一个网段上，与局域网的其它机器没有区别。
- 可以与局域网内其它主机通信，可以与外部网络通信
- 容易与局域网其他主机引起ip地址冲突

- VMnet1 虚拟交换机：Host-Only仅主机模式

特点：

- a. 默认使用VMnet1，提供DHCP服务
- b. 虚拟机可以和物理主机互相访问，但虚拟机无法访问外部网络

- VMnet8 虚拟交换机：NAT模式

特点：

- a. 默认使用VMnet8，提供DHCP服务
- b. 虚拟机可以和物理主机互相访问，可访问外部网络
- c. 局域网内其它机器访问不了

## 2.4 linux CentOS7的网络设置(上网)

简介：如何使得CentOS7上外网

- Bridged桥接模式
  - 重启主机的命令：reboot
  - 重启网卡的命令：systemctl restart network.service
  - 查看ip地址的命令：ip addr
  - ping命令可以检测网络是否畅通：ping ip地址
  - 结束ping命令：ctrl + c
  - 安装ctrl + l 可以清屏
  - 可以访问外网
  - 容易与局域网的其它机器ip地址冲突
- Host-Only仅主机模式
  - 一般情况下不能访问外网
  - 不会与局域网的其它机器ip地址冲突
- NAT模式
  - 可以访问外网
  - 不会与局域网的其它机器ip地址冲突
- 总结：

为了提供不同的测试环境，有的需要虚拟机与局域网内的其它机器环境隔离，有的不需要，所以就整出这么多花样

## 2.5 linux远程连接工具的使用

简介：常用的linux远程连接软件介绍

- 使用前保证网络畅通

- 查看ip地址是多少
- 安装上传下载功能lrzsz服务：yum -y install lrzsz（确认服务器可以联外网）
- 上传：执行rz命令
- 下载：sz 文件名

## 2.6 虚拟机的快速克隆与快照恢复

简介：如何快速克隆多台虚拟机

- 克隆后：systemctl restart network.service命令执行会报错，原因是MAC地址不正确
- 安装键盘的tab键可以对命令进行补全
- 网卡路径：/etc/sysconfig/network-scripts/ifcfg-eno16777728
- 使用vi工具进行编辑网卡信息：vi /etc/sysconfig/network-scripts/ifcfg-eno16777728（按住键盘的i进入编辑模式，按住键盘左上角esc键退出编辑模式，再输入：wq进行保存）
- 快照:相当于备份



小D课堂 愿景："让编程不在难学，让技术与生活更加有趣" 更多教程请访问 [xdclass.net](http://xdclass.net)

## 第三章：linux核心必备知识之常用操作

### 3.1 linux工作中常用之必备基础命令(一)

简介：讲解工作中常用的基础命令

#### 1、cd命令

功能说明：切换目录。

举 例：cd /usr/local/；cd ..；cd -

#### 2、ls命令

功能说明：列出目录内容。

举 例：ls -ltr；ls -lrt /home/

#### 3、pwd命令

功能说明：查询所在目录。

举 例：pwd

#### 4、cat命令

功能说明：查看小文件内容。

举 例：cat -n 123.txt

#### 5、more命令

功能说明：查看大文件内容

举 例：more System.map-3.10.0-123.el7.x86\_64

#### 6、head命令

功能说明：查看文件的前面N行。

举 例：head -20 system.map-3.10.0-123.el7.x86\_64

#### 7、tail命令

功能说明：查看文件的后面N行。

举 例：tail -f access.log ; tail -20 access.log

#### 8、touch命令

功能说明：创建一个空文件。

举 例：touch 123.txt

#### 9、mkdir命令

功能说明：创建目录。

举 例：mkdir -p /tmp/XD/XD/class

#### 10、rmdir命令

功能说明：删除目录。

举 例：rmdir /tmp/XD/XD/class

#### 11、cp命令

功能说明：拷贝文件。

举 例：cp 123.txt class/ ; cp -a 123.txt class/789.txt

#### 12、mv命令

功能说明：移动或更名现有的文件或目录。

举 例：mv 123.txt 345.php ; mv 789.txt /home/987.php

#### 13、rm命令

功能说明：删除文件或目录。

举 例：rm 987.php ; rm -rf 456.txt

#### 14、diff命令

功能说明：对比文件差异。

举 例：diff 123.txt 456.txt

#### 15、ssh命令

功能说明：远程安全登录方式。

举 例：ssh 192.168.226.131

#### 16、exit命令

功能说明：退出命令。

举 例：

#### 17、id命令

功能说明：查看用户。

举 例：id root

#### 18、uname命令

功能说明：查询主机信息。

举 例：uname -a

#### 19、ping命令

功能说明：查看网络是否通。

举 例：ping 192.168.226.131

#### 20、echo命令

功能说明：标准输出命令。

举 例：echo "this is echo 命令"

#### 21、man命令(ls --help)

功能说明：查看帮助文档

举 例：man ls

#### 22、help命令

功能说明：查看内部命令帮助

举 例：help if

## 3.2 linux工作中常用之必备基础命令(二)

简介：讲解工作中常用的基础命令

#### 1、clear命令

功能说明：清屏。

举 例：clear ; ctrl + l

#### 2、who命令

功能说明：当前在本地系统上的所有用户的信息

举 例：whoami ; who

#### 3、uptime命令

功能说明：查询系统信息

举 例：

load average: 0.00, 0.01, 0.05 1分钟的负载, 5分钟的负载, 15分钟的负载

#### 4、w命令

功能说明：查询系统信息

举 例：w

#### 5、free命令

功能说明：查看系统内存

举 例：free -h ; free -m

#### 6、wc命令

功能说明：统计行。

举 例：wc -l 123.txt

#### 7、grep命令

功能说明：查找文件里符合条件的字符串。

举 例：grep '119.4.253.206' 123.txt | wc -l

-n:输出行数 grep -n '80.82.70.187' 123.txt

-w:精确匹配 grep -w '113.66.107.198' 123.txt

-i:忽略大小写 grep -i 'IP:113.66.107.198' 123.txt

-v:反向选择 grep -v '113.66.107.198' 123.txt

#### 8、find命令

功能说明：查询文件。

举 例：find / -name -type f 123.txt

#### 9、uniq命令

功能说明：对排序好的内容进行统计

举 例：uniq -c 123.txt | sort -n

#### 10、sort命令

功能说明：对内容进行排序

举 例：uniq -c 123.txt | sort -n

#### 11、df命令

功能说明：文件系统的磁盘使用情况统计。

举 例：df -h

#### 12、netstat

功能说明：查看网络端口的使用情况

举 例：netstat -tunlp | grep nginx

-t : 显示tcp端口

-u : 显示UDP端口

-n : 指明拒绝显示别名

-l : 指明listen的

-p : 指明显示建立相关连接的程序名

安装netstat命令：yum -y install net-tools

#### 13、hostname命令

功能说明：查看主机名

举 例：hostname

#### 14、ps命令

功能说明：显示所有进程信息。 ps 与grep 常用组合用法，查找特定进程

举 例：ps -ef | grep nginx

ps -aux | grep nginx

#### 15、kill命令

功能说明：杀进程

举 例：kill -9 top

#### 16、top命令

功能说明：监控Linux系统状况，比如cpu、内存的使用

举 例：按住键盘q退出

#### 17、du命令

功能说明：统计大小

举 例：du -sh ; du -sm \*

#### 18、firewall-cmd命令

功能说明：查看防火墙的状态

举 例：firewall-cmd --state

centos 7 关闭防火墙：systemctl stop firewalld.service



#### 19、echo命令

功能说明：判断上一条命令是否正确

举 例：echo \$?

#### 20、cal命令

功能说明：查看日历

举 例：cal 2008

## 3.3 linux 下什么是输入输出错误重定向

简介：介绍输入输出错误重定向的使用

- 什么是输入重定向？

通俗的讲，输入重定向就是把要输入的信息写入到指定的文件中

- 什么是输出重定向？

通俗的讲，输出重定向就是把要输出的信息写入到一个文件中，而不是将要输出的文件信息输出到控制台（显示屏）

- 什么是错误重定向？

通俗的讲，错误重定向就是把错误的信息写入到一个文件中

- linux中一切皆文件

文件描述符：

posix名称	文件描述符	用途
/dev/stdin	0	标准输入
/dev/stdout	1	标准输出
/dev/stderr	2	标准错误输出

- 输入重定向：<

◦ eg：wc -l < 123.txt

- 输出重定向：> #代表覆盖写入；>> #代表追加写入

◦ eg：cat >> 123.txt；cat > 123.txt；ls -lrt >123.txt；echo '123455' > 123.txt

- 错误重定向：

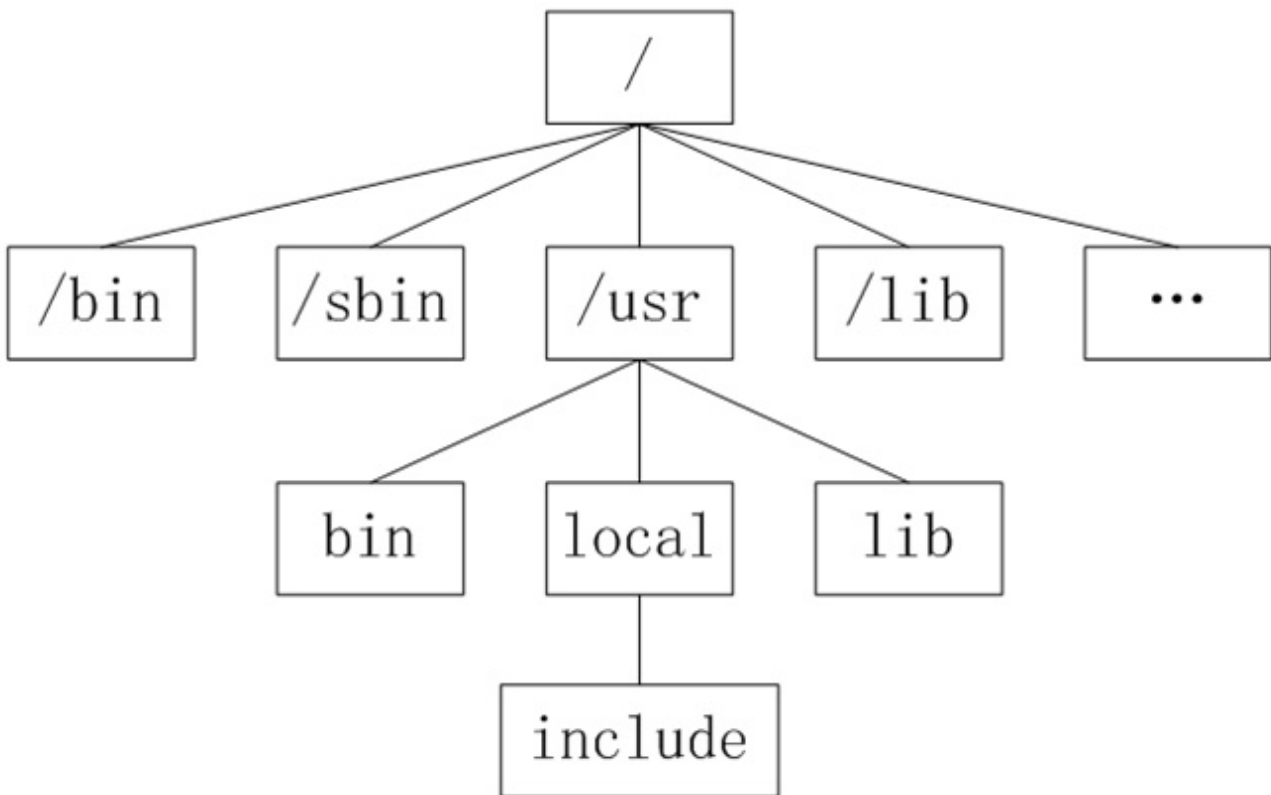
◦ eg：llll 2> 123.txt；llll 2> /dev/null #/dev/null 无底洞

- 几个符号：

- & #代表等同于是的意思 `ls -lrt /boot /test 1>/root/123.txt 2>&1`
- &> #代表不分正确还是错误的意思 `ls -lrt /boot /test &>123.txt`
- | #管道符
- ; #代表的是可以执行多条命令 `cat /etc/passwd | grep root ; ls -lrt`
- && #前面的命令执行成功的话，后面的才可以执行成功；前面的命令执行失败的话，后面的不可以执行
- || #前面的命令执行成功的话，后面的不可以执行；前面的命令执行失败的话，后面的可以执行

### 3.4 linux下的目录分类与作用介绍

简介：详细介绍CentOS7下常见的目录作用



- /：根目录，一般根目录下只存放目录，不要存放文件，也不要修改，或者删除目录下的内容
- /mnt：测试目录
- /root：root用户的家目录
- /home：普通用户的家目录
- /tmp：临时目录(比如文件上传时)
- /var：存放经常修改的数据，比如程序运行的日志文件
- /boot：存放的启动Linux 时使用的内核文件，包括连接文件以及镜像文件
- /etc：系统默认放置配置文件的地方
- /bin：所有用户都能执行的程序
- /sbin：只有root才能执行的程序
- /usr：用户自己的软件都可以放到这儿来
- /dev：存放硬件设备的地方(/dev/cdrom)
- /media：挂载光盘使用的
- 挂载光盘：`mount /dev/cdrom /media`
- 卸载光盘：`umount /dev/cdrom`

绝对路径：说白了就是完整的路径

相对路径：相对于当前位置路径 ./ 代表的是当前目录的意思 ../ 代表的是上一级目录的意思

## 3.5 虚拟机中Centos7与外部物理机的时间同步

简介：使内部机器与外部机器时间同步

- 卸载的光盘的时候：

```
[root@localhost media]# umount /dev/cdrom
umount: /media: target is busy.
(In some cases useful info about processes that use
the device is found by lsof(8) or fuser(1))
```

- 解决方法：
  - 首先确认联网状态
  - yum install -y psmisc
  - fuser -mv /media
  - fuser -kv /media
- date命令
  - date "+%Y-%m-%d %H:%M:%S"
  - date -s "2020-10-1 22:00:00"
  - date -d yesterday "+%Y-%m-%d %H:%M:%S"
  - date "+%w"
- 安装VMwareTools
  - 第一步打开虚拟机，安装VMwareTools使工具软件包下载到光盘
  - 挂载光盘到linux系统
  - cp VMWareTools-10.2.0-7259539.tar.gz /root/
  - umount /dev/cdrom
  - tar -xf VMWareTools-10.2.0-7259539.tar.gz
  - cd vmware-tools-distrib
  - yum -y install perl-Data-Dumper
  - ./vmware-install.pl
  - 一路按住键盘的 回车 键，选用默认
  - echo \$? 验证是否安装成功，返回0就是成功
  - 验证里面虚拟机的时间是否与外部物理机的时间同步

## 3.6 linux高手编辑神器vi的使用

简介：讲解编辑神器vi

- vi的基本概念: ( 三种模式 )
  - 命令模式
  - 插入模式
  - 底行模式

进入插入模式：按住键盘的 i 或者 o 或者 a

进入命令模式：按住键盘的左上角esc键

进入底行模式：前提是在命令模式，输入 : 进入

- 在命令行模式中的操作：

```
$ #移动到这一行的行尾
gg #移动到文档第一行行首
G #移动到文档最后一行行首
x #删除内容，删除一个字符
dd #删除光标所在的那一整行
u #复原原来的操作
v #选中范围按y即复制
p #粘贴
```

- 在底行模式中的操作：

```
n #n为数字。光标移动到第n 行
/ #寻找内容

%s/word1/word2/g #从第一行到最后一行寻找 word1 字符串，并将该字符串取代为 word2
n1,n2s/word1/word2/g #n1 与 n2 为数字。在第 n1 与 n2 行之间寻找 word1 这个字符串，并将该字符串取代为 word2

set nu #显示行号
set nonu #取消行号
q! #强制离开不保存
wq #离开并保存
wq! #强制离开并保存
!ls #暂时离开
```

## 3.7 linux的用户管理与组管理

简介：讲解linux用户管理与组管理

- Linux是一个多用户、多任务的操作系统
- linux用户的分类：
  - 超级用户root：拥有至高无上的权限 UID：0
  - 普通用户：权限有一定的限制，可以登录系统。一般可以执行/usr/local/bin或者/bin或者/usr/bin或者自己家目录的命令 UID：500 -60000（centos 6）UID：1000 - 60000（centos7）
  - 系统用户（伪用户）：一般不会登录系统，一般情况是用来维持某个服务程序 UID：1-499（centos 6）UID：1-1000（centos 7）
- 关于用户的相关配置文件

- 账号信息：/etc/passwd
- 密码信息：/etc/shadow

```
test :x :1000 :1000 : :/home/test :/bin/bash
用户 密码占位符 UID GID 用户描述 用户家目录 登录后使用的shell解释

/sbin/nologin #是不可登录的
/bin/bash #可以登录
```

- 添加用户命令：useradd
  - -u #指定用户UID
  - -d #指定用户主目录
  - -g #指定用户所属组
  - -r #指定用户是系统用户
  - -s #用户登录shell解释器
  - -M #不创建主目录
- eg：创建一个用户XD，指定UID为1010，指定家目录为/home/XD，指定所属组为root组，指定登录shell为/bin/bash
  - useradd -u 1010 -d /home/XD -g root -s /bin/bash XD
- 登录用户时出现以下信息如何解决：

```
如下：
bash-4.2$
bash-4.2$
解决：复制相关信息到家目录
cp -r /etc/skel/.bash* /home/XD/
```

- 删除用户命令：userdel
  - -r #连同家目录一块删除
- 添加用户组命令：groupadd
- 删除用户组命令：groupdel
- 修改用户的信息命令：usermod
  - -u #指定用户UID
  - -d #指定用户主目录
  - -g #指定用户所属组
- 设置用户密码命令passwd
  - passwd XD
  - echo "123456" | passwd --stdin XD

## 3.8 linux必备核心知识之文件属性与权限操作

简介：讲解文件属性与权限更改

- 文件的属性：ls -lrti

```
135088935 -rw-----. 1 root root 1778 Oct 1 2020 yum.log
```

第一列：i节点；i节点可以理解文件id，一个i节点号可以对应多个文件，一个文件只能对应一个i节点号

第二列：文件的类型与权限

- #代表的是文件；d#代表是目录；l #软链接文件；b #代表块设备；c #代表的是硬件设备（键盘）

r：表示读权限；w：表示写权限；x：表示执行权限

4：表示读权限；2：表示写权限；1：表示执行权限

rw-----:分为三列 rw- --- ---，第一列为所属者的权限，第二列为所属组的权限，第三列为其它的权限

第三列：有多少文件名链接到这个节点

第四列：文件的所有者

第五列：文件的所有组

第六列：容量大小，单位默认为B

第八列：创建或最近修改的时间

第九列：文件名

- 链接

软连接：ln -s

eg: ln -s /home/XD/yum.log /usr/local/

i节点号跟源文件不一样，源文件一旦删除，软链接将找不到源文件

硬链接：ln

eg: ln /home/XD/yum.log /usr/local/XD/

i节点与源文件一模一样，源文件删除，硬链接还可以继续使用。常用于防止重要文件被误删

- 修改文件的权限命令chmod：

- -R #递归的意思

- chmod -R 777 /home/XD/\*

eg：

```
chmod u+x,g+w,o+w boot.log
```

```
chmod u-x,g-w,o-w boot.log
```

```
chmod 777 boot.log
```

- 修改文件的所有者跟所属组命令chown：

- -R #递归的意思

eg：更改文件目录XD 的所属者为root用户 跟 所属组为XD组

```
chown -R root:XD XD
```

## 3.9 linux 的文件归档与解压缩详细介绍

简介：介绍如何对文件进行归档压缩

- 文件归档：

文件归档也称之为打包，指的是一个文件或者多个文件或者目录的一个集合，这个集合储存在一个文件中。归档文件是没有进行压缩的，所以占用的空间是所有文件或者目录的总和。工作中经常与压缩结合在一起使用

- 文件压缩：  
节约磁盘空间，加快文件的传输速率
- 解压缩命令：gzip；xz
  - gzip：不能压缩目录，只能压缩文件，压缩速度最快，但是压缩比例比较低。扩展名：.gz
    - 不保留源文件压缩：gzip 123.txt
    - 保留源文件压缩：gzip -c 345.txt > 345.txt.gz
    - 不保留源文件的解压：gunzip 123.txt.gz
    - 保留原文件的解压：gunzip -c 345.txt.gz > 234.txt
    - 不保留源文件解压：gzip -d 345.txt.gz
  - xz：可以压缩目录和文件压缩的速度比较慢，但是压缩比例最高。扩展名：.xz
    - 不保留源文件压缩：xz 123.txt
    - 保留源文件压缩：xz -c 345.txt > 345.txt.xz
    - 不保留源文件的解压：unxz 345.txt.xz
    - 保留原文件的解压：xz -d -k 123.txt.xz
    - 不保留源文件解压：xz -d 123.txt.xz
- 归档与压缩命令tar：
  - -c #创建新文件
  - -f #指定文件格式
  - -v #显示详细过程
  - eg：tar -cf vmware.tar vmware-tools-distrib；tar -cvf vmware-tools.tar vmware-tools-distrib
  - -z #以gzip方式归档压缩 eg：tar -zcvf vmware-tools.tar.gz vmware-tools-distrib
  - -j #以xz方式进行归档压缩 eg：tar -jcvf vmware-tools.tar.xz vmware-tools-distrib；tar -jcvf /home/XD/vmware-tools.tar.xz vmware-tools-distrib
  - -v #解档解压操作 tar -xf vmware-tools.tar.xz
  - -C #指定解压路径



小D课堂 愿景："让编程不在难学，让技术与生活更加有趣" 更多教程请访问 [xdclass.net](http://xdclass.net)

## 第四章：linux必须掌握的核心实用命令

### 4.1 linux搜索神器find命令高级用法

简介：详细讲解find命令的使用

- 基本用法：find 路径 选项
  - -type #根据文件类型 find /var/log -type f -name "\*.log"；find /var/log -type d
  - -name #根据文件名 find /var/log -type f -name "\*.log"

- -perm #根据文件权限 find /var/log -perm 600 -type f -name "\*.log"
- -user #根据文件所属主 find /var/log -user XD
- 高级用法：

```
find /var/log -type f -name "*.log" -exec wc -l {} \;
; #可以执行多条命令
\ #转义符,转义;使得这条命令结束
{} #把find命令匹配到的每一次结果传递给{}
-exec #执行

eg:
find /var/log -type f -name "*.log" -exec cp -a {} /home/test \;

-mtime #根据文件的变更时间来查找;-n表示更改时间距离现在n天以内;+n表示更改时间距离现在n天以前
eg:
find /var/log -mtime -2 -name "*.log" -exec ls -lrt {} \;
find /var/log -mtime +2 -name "*.log" -exec ls -lrt {} \;
```

## 4.2 CentOS7的防火墙以及selinux介绍

### 简介：firewall防火墙的使用

- 防火墙：主要用户信息安全防护，主要有软件防火墙和硬件防火墙。firewalld防火墙是软件防火墙
- 在centos7 之前默认采用的防火墙是iptables，而在centos 7则是采用firewall
- 查看firewalld服务状态
  - systemctl status firewalld
- 开启、重启、关闭firewalld服务
  - 开启：systemctl start firewalld.service
  - 关闭：systemctl stop firewalld.service
  - 重启：systemctl restart firewalld.service
- 查看firewall防火墙的状态
  - firewall-cmd --state
- 查看防火墙开放端口规则
  - firewall-cmd --list-port
- 开放80端口
  - firewall-cmd --permanent --add-port=80/tcp ( --permanent永久生效，没有此参数重启后就失效 )
- 加载生效开放的端口
  - firewall-cmd --reload
- 查询指定端口80是否开放
  - firewall-cmd --query-port=80/tcp
- 验证80端口是否开放：
  - 安装telnet命令：yum -y install xinetd telnet telnet-server ( 确认联网状态 )
  - 安装netstat与ifconfig命令：yum -y install net-tools ( 确认联网状态 )
- 关闭80端口



- firewall-cmd --remove-port=80/tcp
- SELinux 的三种工作模式；配置文件路径：/etc/selinux/config
  - enforcing：强制模式。违反selinux 规则的行为将会被阻止并记录到日志中去
  - permissive：宽容模式。违反selinux 规则的行为将会记录到日志中去
  - disabled：关闭模式。

## 4.3 linux服务器之间telnet与scp命令的用法

### 简介：讲解企业中scp命令的使用

- telnet命令：主要用于测试到某台机器的某个端口是否畅通
- telnet这个命令是依赖于 xinetd服务于telnet-server服务
- telnet命令的安装：yum -y install xinetd telnet telnet-server（确认联网状态）
- telnet命令用法：
  - telnet IP地址 端口
  - 应用场景：测试某个端口是否畅通
- scp命令：用于服务器之间的文件或者文件目录拷贝
  - 用法1：从本机拷贝文件到别的机器 scp 本机文件的存放路径 root@服务器IP:服务器目标路径

eg：从192.168.72.128这台机器的/root/VMwareTools-10.2.0-7259539.tar.gz文件拷贝到192.168.72.129的/root/下

```
scp /root/VMwareTools-10.2.0-7259539.tar.gz root@192.168.72.129:/root/
```

- 用法2：从别的机器拷贝文件到本地目录 scp root@服务器IP:服务器目标路径 本机文件的存放路径

eg：从192.168.72.129这台机器的/root/VMwareTools-10.2.0-7259539.tar.gz文件拷贝到192.168.72.128的/root/下

```
scp root@192.168.72.129:/root/VMwareTools-10.2.0-7259539.tar.gz /root/
```

- -r参数：递归的作用（可以拷贝目录）

eg：  
scp -r vmware-tools-distrib root@192.168.72.129:/root/

## 4.4 linux进程管理命令之ps -ef与ps aux详解

### 简介：介绍实际工作中ps命令的参数作用

```
[root@localhost ~]# ps -ef | more
UID          PID    PPID  C STIME TTY          TIME CMD
root          2        0  0 Jul30 ?        00:00:00 [kthreadd]
root          3        2  0 Jul30 ?        00:00:06 [ksoftirqd/0]
root          5        2  0 Jul30 ?        00:00:00 [kworker/0:0H]
root          7        2  0 Jul30 ?        00:00:04 [migration/0]
root          8        2  0 Jul30 ?        00:00:00 [rcu_bh]
root          9        2  0 Jul30 ?        00:00:00 [rcuob/0]
```

```
root          10      2  0 Jul30 ?          00:00:00 [rcuob/1]
```

UID：用户ID

PID：进程ID

PPID：父进程号

C：CPU的占用率

STIME：进程的启动时间

TTY：TTY终端

TIME：进程执行起到现在总的CPU占用时间

CMD：启动这个进程的命令

```
[root@localhost ~]# ps aux | more
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	2	0.0	0.0	0	0	?	S	Jul30	0:00	[kthreadd]
root	3	0.0	0.0	0	0	?	S	Jul30	0:06	[ksoftirqd/0]
root	5	0.0	0.0	0	0	?	S<	Jul30	0:00	[kworker/0:0H]
root	7	0.0	0.0	0	0	?	S	Jul30	0:04	[migration/0]
root	8	0.0	0.0	0	0	?	S	Jul30	0:00	[rcu_bh]
root	9	0.0	0.0	0	0	?	S	Jul30	0:00	[rcuob/0]
root	10	0.0	0.0	0	0	?	S	Jul30	0:00	[rcuob/1]
root	11	0.0	0.0	0	0	?	S	Jul30	0:00	[rcuob/2]

USER：哪个用户启动了这个命令

PID：进程的ID

%CPU：CPU的占用率

%MEM：内存的使用率

VSZ：如果一个程序完全驻留在内存中一共需要使用多少内存空间

RSS：进程当前占用了多少内存

TTY：tty终端

STAT：表示当前进程的状态（S#处于休眠的状态；D#不可中断的状态；Z#僵尸进程；X#死掉的进程）

START：启动这个命令的时间点

TIME：进程执行起到现在总的CPU占用时间

COMMAND：启动这个进程的命令

一般执行ps -ef 或者ps aux 命令是查看我们的进程是否启动成功，或者找出进程号，对进程的kill强制关闭

## 4.5 处理海量数据之cut命令

**简介：**介绍cut命令的基本使用

- cut应用场景：通常对数据进行列的提取
  - 语法：cut [选项]...[file]

选项：

-d #指定分割符

-f #指定截取区域

-c #以字符为单位进行分割

注意：不加-d选项，默认为制表符，不是空格

eg:

以':'为分隔符, 截取出/etc/passwd的第一列跟第三列

```
cut -d ':' -f 1,3 /etc/passwd
```

eg:

以':'为分隔符, 截取出/etc/passwd的第一列到第三列

```
cut -d ':' -f 1-3 /etc/passwd
```

eg:

以':'为分隔符, 截取出/etc/passwd的第二列到最后一列

```
cut -d ':' -f 2- /etc/passwd
```

eg:

截取/etc/passwd文件从第二个字符到第九个字符

```
cut -c 2-9 /etc/passwd
```

eg:

比如领导想叫你截取linux上面所有可登陆普通用户

```
cat /etc/passwd | grep '/bin/bash' | cut -d ':' -f 1 | grep -v root
```

## 4.6 处理海量数据之awk命令

**简介：**介绍awk命令的基本使用

- awk的简介：一个非常强大的数据处理命令，支持条件判断，数组，循环等功能，与grep，sed被称为linux三剑客
- awk的应用场景：通常对数据进行列的提取
  - 语法：
    - awk '条件1 {执行动作} 条件2 {执行动作} ...' 文件名
    - 或awk [选项] '条件1 {执行动作} 条件2 {执行动作} ...' 文件名
- 特殊要点与举例说明:
  - printf #格式化输出，不会自动换行。
  - print #打印出内容，默认会自动换行
  - %s #代表字符串
  - \t #制表符
  - \n #换行符

```
eg: printf '%s\t%s\t%s\t%s\t%s\t%s\n' 1 2 3 4 5 6
```

- awk的一些特殊要点与举例说明
  - NR #行号
  - \$1 #代表第一列
  - \$2 #代表第二列
  - \$NF#代表最后一列

```
df -h | awk 'NR==4 {print $1}'  
df -h | awk '(NR>=2 && NR <=5) {print $1}'  
df -h | awk '{print $NF}'
```

- -F #指定分割符

```
awk -F":" '{print $1}' /etc/passwd
```

- BEGIN #在读取所有行内容前就开始执行，一般用来初始化操作

```
eg :  
cat /etc/passwd | awk 'BEGIN {FS=":"} {print $1}'  
df -h |grep -v 'Filesystem' | awk '{printf $1} {printf "文件系统使用率："} {print $5}'  
df -h |grep -v 'Filesystem' | awk 'BEGIN {printf "文件系统使用情况：\n\n"} {printf $1}  
{printf "文 件系统使用率："} {print $5}'
```

- END #结束的时候 执行

```
df -h |grep -v 'Filesystem' | awk 'BEGIN {printf "文件系统使用情况：\n\n"} {printf $1}  
{printf "文件系统使用率："} {print $5} END {printf "一切正常\n"}'
```

## 4.7 处理海量数据之sed命令

### 简介：介绍sed命令的基本使用

- sed的应用场景：主要对数据进行处理（选取，新增，替换，删除，搜索）
- sed语法：
  - sed [选项] [动作] 文件名
- 常见的选项与参数：
  - -n #把匹配到的行输出打印到屏幕
  - p #以行为单位进行打印，通常与-n一起使用

```
df -h | sed -n '2p'
```

- d #删除

```
df -h | sed '2d'
```

- a #在行的下面插入新的内容

```
df -h | sed '2a 1234567890'
```

- i #在行的上面插入新的内容

```
df -h | sed '2i 1234567890'
```

- c #替换

```
df -h | sed '2c 1234567890'
```

- 指定字符串替换：s/要被取代的内容/新的字符串/g #指定内容进行替换

```
df -h | sed 's/centos-root/Centos7/g'
```

- -i #对源文件进行修改(高危操作，慎用，用之前需要备份源文件)

```
sed -i 's/Centos7/Centos8/g' df.txt
```

- 搜索：在文件中搜索内容

```
sed -n '/tmpfs/p' df.txt
```

- -e #表示可以执行多条动作

```
sed -e 's/Centos8/Centos7/g' -e 's/tmpfs/TMP/g' df.txt >123.txt
```



小D课堂 愿景："让编程不在难学，让技术与生活更加有趣" 更多教程请访问 [xdclass.net](http://xdclass.net)

## 第五章：linux服务器常用企业服务的安装

### 5.1 Linux下常用安装软件服务rpm方式的介绍

简介：介绍rpm软件包的管理

- rpm安装：安装别人编译好的软件包，rpm即Redhat Package Manager，是Redhat的软件包管理方式
- rpm安装优点：
  - 软件已经编译打包，所以传输和安装方便，让用户免除编译
  - 在安装之前，会先检查系统的磁盘、操作系统版本等，避免错误安装
- rpm安装缺点：
  - 软件包安装的环境必须与打包时的环境一致或相当
  - 必须安装了软件的依赖包
- RPM包的命名规则：

```
which-2.20-7.el7.x86_64.rpm
which  #代表的是软件名称
2.20   #代表的是软件版本号；
7      #代表的是发布版本号，指的是这个rpm软件包是第几次编译生成的
el7    #代表的是企业版的7操作系统
x86    #代表的是CPU架构
64     #代表的是系统的位数
```

- 安装rpm软件包：

```
-i #install 安装软件包
-v #输出更多的详情信息
-h #输出哈希标记( # )
--nodeps #不验证软件的依赖

rpm -ivh zsh-5.0.2-7.el7.x86_64.rpm
rpm -ivh mariadb-server-5.5.35-3.el7.x86_64.rpm --nodeps
```

- rpm包下载地址

```
http://rpmfind.net/
http://rpm.pbone.net/
http://www.rpmseek.com/index.html
```

- rpm 查询功能：rpm -q
  - -a #查询所有已安装的软件包 rpm -qa zsh
  - -f #查询文件所属软件包 rpm -qf /usr/bin/zsh
  - -p #查询软件包
  - -i #显示软件包信息
  - -l #显示软件包中的文件列表
  - -d #显示被标注为文档的文件列表
  - -c #显示被标注为配置文件的文件列表
- rpm 包升级:
  - -U #升级rpm软件服务
    - rpm -Uvh zsh-5.0.2-7.el7.x86\_64.rpm
- rpm 包卸载:
  - -e #卸载
    - rpm -e zsh

## 5.2 Linux下常用安装服务软件yum方式的介绍

简介：介绍yum软件包的管理并配置本地yum源

- yum安装：基于 C/S 架构，yum安装称之为傻瓜式安装
- yum安装优点：方便快捷，不用考虑包依赖，自动下载软件包。
- yum安装缺点：人为无法干预，无法设定想要的参数
- 配置本地yum源：

- 配置文件的路径：/etc/yum.repos.d/

```
[Centos7-yum]          #yum源名称，唯一的，用来区分不同的 yum 源
name=Centos7-source    #对yum源描述信息
baseurl=file:///mnt     #yum源的路径（ repodata目录所在的目录）
enabled=1              #表示启用 yum 源
gpgcheck=0             #为1表示使用公钥检验 rpm 的正确性
```

- yum安装方式的使用：
  - yum repolist #查看yum源列表
  - yum clean all #清空之前yum缓存
  - yum makecache #创建yum缓存，为后续安装更加快速
  - yum -y install #安装软件 yum -y install zsh
  - yum info zsh #查看zsh软件包信息（不管安装了没都会有信息）
  - yum info installed zsh #查看已经安装好的软件信息
  - yum -remove zsh #卸载软件
  - yum search gcc #搜索gcc软件
  - yum update #升级软件

## 5.3 Linux下常用安装服务软件源码编译安装方式的介绍

简介：介绍源码编译安装软件包的管理

- 源码安装优点：编译安装过程，可以设定参数，指定安装目录，按照需求进行安装，指定安装的版本，灵活性比较大。
- 源码安装的缺点：需要对依赖包一个一个的进行安装，不敢随便升级，一升级可能会由于依赖包的是不能使用导致一系列连锁反应
- 源码编译安装软件包4大步骤：

### 1. 解压源码包

```
tar -xf 源码包
```

### 2. 配置

进入解压后的目录，用./configure命令来配置相关信息（比如指定安装目录 --prefix=/usr/local/nginx）和生成Makefile文件

### 3. 编译

```
make -j4
```

### 4. 安装

```
make install
```

## 5.4 CentOS7服务器JDK8安装实战

简介:演练JDK8环境的安装

- 下载jdk官网：

```
https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html
```

- 全局环境变量的配置文件：vi /etc/profile

```
export JAVA_HOME=/usr/local/jdk1.8 #这个路径要改，其余不需要改
export JRE_HOME=$JAVA_HOME/jre
export CLASSPATH=$JAVA_HOME/lib:$JRE_HOME/lib:$CLASSPATH
export PATH=$JAVA_HOME/bin:$JRE_HOME/bin:$PATH
```

- 加载环境变量：source /etc/profile
- java -version

```
java version "1.8.0_211"
Java(TM) SE Runtime Environment (build 1.8.0_211-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.211-b12, mixed mode)
```

## 5.5 CentOS7 实战部署tomcat网站服务器

简介：实战演练tomcat网站服务器的搭建

- Tomcat：是一个开源免费的Web应用服务器，性能稳定，是目前比较流行的Web应用服务器
- tomcat官网下载：

```
https://tomcat.apache.org/download-80.cgi
下载：
yum install -y wget
wget http://mirrors.tuna.tsinghua.edu.cn/apache/tomcat/tomcat-8/v8.5.43/bin/apache-
tomcat-8.5.43.tar.gz
```

- 解压：tar -xf apache-tomcat-8.5.43.tar.gz
- mv apache-tomcat-8.5.43 /usr/local/tomcat8
- 检查java环境：java -version
- tomcat重要目录介绍

```
[root@localhost tomcat8]# ls -lrt /usr/local/tomcat8
total 132
drwxr-x---. 7 root root   76 Jul  5 04:53 webapps
-rw-r-----. 1 root root 16262 Jul  5 04:56 RUNNING.txt
-rw-r-----. 1 root root  7139 Jul  5 04:56 RELEASE-NOTES
-rw-r-----. 1 root root  3255 Jul  5 04:56 README.md
```



```
-rw-r-----. 1 root root 1726 Jul 5 04:56 NOTICE
-rw-r-----. 1 root root 57011 Jul 5 04:56 LICENSE
-rw-r-----. 1 root root 5407 Jul 5 04:56 CONTRIBUTING.md
-rw-r-----. 1 root root 19534 Jul 5 04:56 BUILDING.txt
drwxr-x---. 2 root root 4096 Aug 1 23:33 lib
drwxr-x---. 2 root root 4096 Aug 1 23:33 temp
drwxr-x---. 2 root root 4096 Aug 1 23:33 bin
drwx-----. 3 root root 4096 Aug 1 23:43 conf
drwxr-x---. 2 root root 4096 Aug 1 23:43 logs
drwxr-x---. 3 root root 4096 Aug 1 23:43 work
```

bin：存放可执行命令，比如开启和关闭；

conf：配置文件；

Context.xml：Tomcat公用的环境配置，tomcat 服务器会定时去扫描这个文件

web.xml：Web应用程序描述文件，都是关于Web应用程序的配置文件

server.xml：可以设置tomcat的端口号，添加虚拟机这些的，是对服务器的设置

tomcat-users.xml：用户配置文件

webapps：发布web应用；

lib：库文件；

- 关闭防火墙

```
systemctl stop firewalld.service
```

- 启动tomcat

```
sh startup.sh
```

- 查看是否监听tomcat

```
[root@localhost bin]# ps -ef | grep tomcat
root      68135      1  4 23:43 pts/0    00:00:25 /usr/local/jdk1.8/jre/bin/java -
Djava.util.logging.config.file=/usr/local/tomcat8/conf/logging.properties -
Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -
Djdk.tls.ephemeralDHKeySize=2048 -
Djava.protocol.handler.pkgs=org.apache.catalina.webresources -
Dorg.apache.catalina.security.SecurityListener.UMASK=0027 -Dignore.endorsed.dirs= -
classpath /usr/local/tomcat8/bin/bootstrap.jar:/usr/local/tomcat8/bin/tomcat-juli.jar
-Dcatalina.base=/usr/local/tomcat8 -Dcatalina.home=/usr/local/tomcat8 -
Djava.io.tmpdir=/usr/local/tomcat8/temp org.apache.catalina.startup.Bootstrap start

[root@localhost bin]# netstat -tunlp | grep 68135
tcp6      0      0 127.0.0.1:8005      :::*           LISTEN
68135/java
tcp6      0      0 :::8009            :::*           LISTEN
68135/java
tcp6      0      0 :::8080            :::*           LISTEN
68135/java
```

- 测试能否访问测试页面：

IP地址: 8080

## 5.6 CentOS7 实战源码部署apache网站服务器

### 简介：实战演练apache网站服务器的搭建

- Apache简介：

Apache软件基金会的一个开源免费的网页服务器，也是目前世界上使用最广泛的一种web server，apache最出名的是它跨平台，高效和稳定，可以运行在几乎所有广泛使用的计算机平台上。其特点是简单、速度快、性能稳定，并可做代理服务器来使用，并且可通过简单的 API 扩充，将 Perl/Python 等解释器编译到服务器中

- 源码编译安装 Apache：

- 下载组件apr并解压：

```
组件apr官方网站：http://apr.apache.org/download.cgi
wget http://mirror.bit.edu.cn/apache//apr/apr-1.7.0.tar.gz
tar -xf apr-1.7.0.tar.gz
```

- 下载组件apr-util并解压：

```
组件apr-util官方网站：http://apr.apache.org/download.cgi
wget http://mirrors.tuna.tsinghua.edu.cn/apache//apr/apr-util-1.6.1.tar.gz
tar -xf apr-util-1.6.1.tar.gz
```

- 下载apache并解压：

```
apache官方网站：http://httpd.apache.org/download.cgi
wget http://mirror.bit.edu.cn/apache/httpd/httpd-2.4.39.tar.gz
tar -xf httpd-2.4.39.tar.gz
```

- 安装gcc环境：

```
yum install -y gcc*
```

- 安装zlib-devel库：

```
yum install -y zlib-devel
```

- 安装xml的解析器：

```
yum install -y expat-devel
```

- 下载并安装pcre正则表达式库：

```
下载地址：https://ftp.pcre.org/pub/pcre/
tar -xf pcre-8.43.tar.gz
cd pcre-8.43
mkdir -p /usr/local/pcre
./configure --prefix=/usr/local/pcre
make && make install
```

- 把组件apr与组件apr-util拷贝到apache安装包里：

```
cp -rf apr-1.7.0 /root/test/httpd-2.4.39/src/lib/apr
cp -rf apr-util-1.6.1 /root/test/httpd-2.4.39/src/lib/apr-util
```

- 进入apache解压包进行./configure进行环境收集检验与配置相关模块：

```
cd /root/test/httpd-2.4.39
mkdir -p /usr/local/apache

-----

./configure --prefix=/usr/local/apache --sysconfdir=/usr/local/apache/etc --
with-apr=/home/test/apr-1.7.0 --with-apr-util=/home/test/apr-util-1.6.1 --with-
included-apr --with-pcre=/usr/local/pcre --enable-deflate --enable-expire --
enable-headers --enable-so --enable-modules=most --with-mpm=worker --enable-
rewrite

-----

选项说明：
--prefix                #指定安装目录
--sysconfdir            #指定配置文件的路径
```

--with-apr	#指定依赖文件的路径
--with-apr-util	#指定依赖文件的路径
--with-included-apr	#增加编译效率的
--with-pcre	#指定pcre正则表达式库安装路径
--enable-deflate	#开启压缩文件提高速度节约带宽
--enable-expire	#开启让浏览器缓存，减轻服务器压力，提高访问速度
--enable-headers	#使得支持http头
--enable-so	#使得支持动态模块
--enable-modules=most	#使得支持大多数模块
--with-mpm=worker	#使得Apache工作在worker模式下
--enable-rewrite	#使得支持地址重写

- 编译并安装apache：

```
make -j 4 && make install
```

- 设置服务器名称:

```
vi /usr/local/apache/etc/httpd.conf
```

输入：ServerName进行搜索

添加：

```
ServerName www.xdc1ss.com
```

- 测试并启停apache服务器：

```
ln -s /usr/local/apache/bin/* /usr/sbin/ #设置软连接
```

```
echo "export PATH=/usr/local/apache/bin:$PATH" >> /etc/profile #设置环境变量
```

```
source /etc/profile #加载环境变量
```

```
httpd -t #测试配置文件语法有没有错误
```

```
httpd -k start #启动apache服务
```

```
httpd -k stop #关闭apache服务
```

在防火墙关闭的条件下，打开浏览器，输入IP地址，成功打开apache测试页面

- 相关报错收集与解决办法：

```
(1) make[1]: *** [xml/apr_xml.lo] Error 1
```

```
make[1]: Entering directory `/home/test/apr-util-1.6.1'
```

```
/bin/sh /usr/local/apache/apr/build-1/libtool --silent --mode=compile gcc -g -O2 -pthread -DHAVE_CONFIG_H -DLINUX -D_REENTRANT -D_GNU_SOURCE -I/home/test/apr-util-1.6.1/include -I/home/test/apr-util-1.6.1/include/private -I/usr/local/apache/apr/include/apr-1 -o xml/apr_xml.lo -c xml/apr_xml.c && touch xml/apr_xml.lo
```

```
xml/apr_xml.c:35:19: fatal error: expat.h: No such file or directory
```

```
# include <expat.h>                                ^

compilation terminated.
make[1]: *** [xml/apr_xml.lo] Error 1
make[1]: Leaving directory `/home/test/apr-util-1.6.1'
make: *** [all-recursive] Error 1
```

错误原因：缺少xml 的解析器

解决方法：yum -y install expat-devel

---

## (2) configure 时 error: APR not found

错误详情：

configure:

checking for APR... no

configure: error: APR not found. Please read the documentation.

错误原因：没有指定 Apache 必需组件 APR 或没有加--with-apr 选项指定 APR 的安装位置。

解决方法：安装 APR 并且加--with-apr 选项指定正确的位置。

---

## (3) configure 时--with-apr 选项不正确

错误详情：

configure:

checking for APR... configure: error: -with-apr requires a directory or file to be provided

configure:

checking for APR... configure: error: the -with-apr parameter is incorrect. It must specify an install prefix, a build directory, or an apr-config file.

错误原因：--with-apr 选项不能为空或者指定的 APR 的位置不正确

解决方法：指定正确的 APR 选项。其实系统中已经使用 yum 安装了 APR，却不知道如何指定 yum 安装的 APR 的位置，故出此错误，也可以进行手动源代码编译安装 APR来解决这个错误。

---

## (4) configure 时 error: APR-util not found

错误详情：

configure:

checking for APR-util... no

configure: error: APR-util not found. Please read the documentation.

错误原因：没有安装 Apache 必需组件 APR-util 或没有加--with-apr-util 选项指定 APR-util 的位置

解决方法：--with-apr-util 选项指定正确的安装位置

---

(5) configure 时 error: pcre-config for libpcre not found

错误详情：

checking for pcre-config... false

configure: error: pcre-config for libpcre not found. PCRE is required and available from <http://pcre.org/>

错误原因：没有安装 Apache 必需组件 PCRE 或没有加-with-pcre 选项指定 PCRE 的安装位置。

解决方法：安装 PCRE 并且加-with-pcre 选项指定正确的安装位置

-----  
(6) configure 时 error: Did not find pcre-config script at /usr/local/pcre2

错误详情：

checking for pcre-config... false

configure: error: Did not find pcre-config script at /usr/local/pcre2

错误原因：httpd 2.4.39 不支持 pcre2？

解决方法：下载 pcre-8.43 安装即可。

-----  
(7) 启动 Apache 时提示设置 ServerName

错误详情：

AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using localhost.localdomain. Set the 'ServerName' directive globally to suppress this message

错误原因：没有设置服务器名称

解决方法：vi /usr/local/apache/etc/httpd.conf

## 5.7 CentOS7 实战源码部署nginx网站服务器

简介：实战演练nginx网站服务器的搭建

- nginx 简介：

Nginx是一款高性能的 HTTP 和反向代理服务器

- Nginx的优点：

1. 高并发量：根据官方给出的数据，能够支持高达 50,000 个并发连接数的响应
2. 内存消耗少：处理静态文件，同样起web 服务，比apache 占用更少的内存及资源，所有它是轻量级的
3. 简单稳定：配置简单，基本在一个conf文件中配置，性能比较稳定，可以7\*24小时长时间不间断运行
4. 模块化程度高：Nginx是高度模块化的设计，编写模块相对简单
5. 负载均衡服务器：Nginx可以做高并发的负载均衡，且Nginx是开源免费的，如果使用F5等硬件来做负载均衡，硬件成本比较高
6. 可移植性高：Nginx代码完全用C语言编写

- Nginx的缺点：

- 1.动态处理差：nginx处理静态文件好，耗费内存少，但是处理动态页面比较差
- 2.rewrite弱：虽然nginx支持rewrite功能，但是相比于Apache来说，Apache比nginx 的rewrite 强大。

- 编译安装：

安装gcc编译环境：

```
yum install -y gcc-c++
```

安装zlib-devel库：

```
yum install -y zlib-devel
```

安装OpenSSL密码库：

```
yum install -y openssl openssl-devel
```

安装pcre正则表达式库：

```
下载地址：https://ftp.pcre.org/pub/pcre/  
tar -xf pcre-8.43.tar.gz  
cd pcre-8.43  
mkdir -p /usr/local/pcre  
./configure --prefix=/usr/local/pcre  
make && make install
```

下载编译安装nginx：

```
nginx下载官网：http://nginx.org/en/download.html  
wget http://nginx.org/download/nginx-1.16.0.tar.gz  
mkdir -p /usr/local/nginx  
tar -xf nginx-1.16.0.tar.gz  
cd nginx-1.16.0  
  
./configure --prefix=/usr/local/nginx --with-http_ssl_module --with-  
http_stub_status_module --with-pcre  
  
make && make install
```

- 启停nginx服务：

启动：

```
/usr/local/nginx/sbin/nginx -c /usr/local/nginx/conf/nginx.conf
```

测试：

```
/usr/local/nginx/sbin/nginx -t
```

关闭：

```
/usr/local/nginx/sbin/nginx -s stop
```

打开浏览器测试能否访问到测试页面

## 5.8 CentOS7 实战源码安装mysql5.7.17数据库服务器

简介：实战演练mysql数据库服务器的搭建

- mysql简介：

mysql是一个开源的关系型数据库管理系统，现在是oracle公司旗下的一款产品，由C和C++语言编写，可移植性高。支持在多种操作系统上安装，最常见有AIX，linux，window。mysql因为开源免费，所以受到了目前互联网行业的欢迎。

以mysql作为数据库，linux系统作为操作系统，apache或者nginx作为web服务器，perl/php/python作为服务端的脚本解释器，就可以搭建起一个免费的网站。被业界称为LNMP或者LAMP

- 准备工作：

- o 强烈建议：

磁盘空间剩余量达到15GB

内存2G以上

步骤一致

确保依赖组件安装完毕

安装mysql5.7.17一致

centos 7

- o 关闭selinux，关闭防火墙：

```
systemctl stop firewalld.service
```

```
vi /etc/selinux/config
```

```
SELINUX=disabled
```

- o 安装cmake工具：



```
yum -y install cmake
```

- 下载boost路径 (mysql5.7.17的必需依赖组件) :

```
tar -xf boost_1_59_0.tar.gz
```

- yum安装其它依赖组件 :

```
yum -y install gcc gcc-c++ bzip2 bzip2-devel bzip2-libs python-devel ncurses  
ncurses-devel openssl openssl-devel
```

- 创建路径 :

```
mkdir -p /usr/local/mysql  
mkdir -p /data/mydata
```

- 创建mysql用户 :

```
useradd -M -s /sbin/nologin mysql
```

- 使用cmake工具对mysql5.7.17进行环境收集检验与配置相关模块 :

解压mysql源码包, 并进入解压后的路径

```
tar -xf mysql-5.7.17.tar.gz  
cd mysql-5.7.17
```

```
cmake . -DCMAKE_INSTALL_PREFIX=/usr/local/mysql \-DMYSQL_DATADIR=/data/mydata \-  
DSYSCONFDIR=/etc \-DWITH_INNOBASE_STORAGE_ENGINE=1 \-DWITH_ARCHIVE_STORAGE_ENGINE=1 \-  
DWITH_BLACKHOLE_STORAGE_ENGINE=1 \-DWITH_READLINE=1 \-DMYSQL_UNIX_ADDR=/tmp/mysql.sock \-  
DWITH_SSL=system \-DWITH_ZLIB=system \-DDEFAULT_CHARSET=utf8 \-  
DDEFAULT_COLLATION=utf8_general_ci \-DDOWNLOAD_BOOST=1 \-DWITH_BOOST=../boost_1_59_0  
\-DENABLE_DOWNLOADS=1
```

参数详细信息解释 :

-DCMAKE_INSTALL_PREFIX=/usr/local/mysql \	#指定mysql数据库安装目录
-DMYSQL_DATADIR=/data/mydata \	#指定数据库文件路径
-DSYSCONFDIR=/etc \	#指定配置文件目录
-DWITH_INNOBASE_STORAGE_ENGINE=1 \	#安装INNOBASE存储引擎
-DWITH_ARCHIVE_STORAGE_ENGINE=1 \	#安装ARCHIVE存储引擎
-DWITH_BLACKHOLE_STORAGE_ENGINE=1 \	#安装BLACKHOLE存储引擎
-DWITH_READLINE=1 \	#使用readline功能
-DMYSQL_UNIX_ADDR=/tmp/mysql.sock \	#连接文件位置
-DWITH_SSL=system \	#表示使用系统上的自带的SSL库
-DWITH_ZLIB=system \	#表示使用系统上的自带的ZLIB库
-DDEFAULT_CHARSET=utf8 \	#指定默认使用的字符集编码
-DDEFAULT_COLLATION=utf8_general_ci \	#指定默认使用的字符集校对规则
-DDOWNLOAD_BOOST=1 \	
-DWITH_BOOST=../boost_1_59_0 \	#指定Boost库的位置, mysql5.7必须添加该参数

-DENABLE\_DOWNLOADS=1

#支持下载可选文件

- 编译并安装：

```
make -j 4 && make install
```

- 初始化mysql：

```
/usr/local/mysql/bin/mysqld --initialize --user=mysql --basedir=/usr/local/mysql --datadir=/data/mydata --socket=/tmp/mysql.sock
```

- 对mysql的相关路径进行更改权限

```
chown -R mysql:mysql /usr/local/mysql /data/mydata
```

- 修改配置文件：

```
vi /etc/my.cnf

[mysqld]
datadir=/data/mydata
socket=/tmp/mysql.sock
symbolic-links=0

[mysqld_safe]
log-error=/usr/local/mysql/log/mysql.errlog
pid-file=/data/mydata/$hostname.pid
```

- 启停mysql：

```
[root@localhost support-files]# ./mysql.server start
Starting MySQL.2019-08-03T14:19:37.028727Z mysqld_safe error: log-error set to
'/usr/local/mysql/log/mysql.errlog', however file don't exists. Create writable for
user 'mysql'.
ERROR! The server quit without updating PID file
(/data/mydata/localhost.localdomain.pid).
```

解决：

```
touch /usr/local/mysql/log/mysql.errlog
```

```
chown -R mysql:mysql /usr/local/mysql/log/mysql.errlog
```

启动：

```
/usr/local/mysql/support-files/mysql.server start
```

关闭：

```
/usr/local/mysql/support-files/mysql.server stop
```

- 登录mysql：

```
/usr/local/mysql/bin/mysql -uroot -p
```

- 修改mysql密码：

```
set password for 'root'@'localhost'=password('密码');

mysql> flush privileges;
ERROR 1146 (42S02): Table 'mysql.servers' doesn't exist

use mysql;

drop table if exists mysql.servers;

CREATE TABLE `servers` (
  `Server_name` char(64) NOT NULL,
  `Host` char(64) NOT NULL, `Db` char(64) NOT NULL,
  `Username` char(64) NOT NULL,
  `Password` char(64) NOT NULL,
  `Port` int(4) DEFAULT NULL,
  `Socket` char(64) DEFAULT NULL,
  `wrapper` char(64) NOT NULL,
  `Owner` char(64) NOT NULL,
  PRIMARY KEY (`Server_name`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COMMENT='MySQL Foreign Servers table';
```

- 添加MySQL服务并设置mysql开机启动：

```
cp -a /usr/local/mysql/support-files/mysql.server /etc/rc.d/init.d/mysql
chkconfig --add mysql
```

```
chkconfig --list mysql
```

```
mysql          0:off    1:off    2:on     3:on     4:on     5:on     6:off
```

chkconfig命令主要用来更新（启动或停止）和查询系统服务的运行级信息

等级0表示：表示关机

等级1表示：单用户模式

等级2表示：无网络连接的多用户命令行模式

等级3表示：有网络连接的多用户命令行模式

等级4表示：不可用

等级5表示：带图形界面的多用户模式

等级6表示：重新启动

```
使用mysql服务的方式操作启停mysql服务：
service mysql start #启动mysql服务器
service mysql stop  #关闭mysql服务器
service mysql restart #重启mysql服务器
```

- 设置mysql环境变量

```
ln -s /usr/local/mysql/bin/* /usr/sbin/
```

- 成功使用 mysql -uroot -p 登录服务器

## 5.9 CentOS7 实战源码部署php服务与nginx 的整合

### 简介：实战演练php服务的搭建

- PHP是一种脚本语言，常用于做动态网站的。
- 源码编译安装：

```
安装依赖组件：
yum -y install gcc gcc-c++ bzip2 bzip2-devel bzip2-libs python-devel ncurses ncurses-
devel openssl openssl-devel

yum install -y libxml2-devel

解压php并进入解压后php包：tar -xf php-5.5.35.tar.gz  && cd php-5.5.35

./configure --prefix=/usr/local/php/ --enable-fpm --with-config-
file=/usr/local/php/etc

编译安装：
make -j 4 && make install
```

- 修改配置文件：

```
cp -a php.ini-production /usr/local/php/etc/php.ini
cp -a /usr/local/php/etc/php-fpm.conf.default /usr/local/php/etc/php-fpm.conf
```

- 启停php服务器：

```
/usr/local/php/sbin/php-fpm #启动

php的默认端口是：9000

cd /usr/local/php/sbin && pkill php-fpm #关闭
```

- 整合nginx测试php：

修改nginx配置文件并添加以下内容：vi /usr/local/nginx/conf/nginx.conf

```
location ~ /\.php$ {
    root    /usr/local/nginx/html;
    fastcgi_pass    127.0.0.1:9000;
    fastcgi_param   SCRIPT_FILENAME  $document_root$fastcgi_script_name;
    include          fastcgi_params;
}
```

修改后对nginx进行重启：/usr/local/nginx/sbin/nginx -s reload

FastCGI是一个在HTTP服务器和动态脚本语言间通信的接口

fastcgi\_pass 127.0.0.1:9000; #设置监听端口

fastcgi\_param SCRIPT\_FILENAME \$document\_root\$fastcgi\_script\_name; #设置脚本文件请求的路径

include fastcgi\_params; #引入fastcgi的配置文件

- 在nginx的网站发布路径下创建index.php文件

vi /usr/local/nginx/html/index.php

```
<?php
phpinfo();
?>
```

- 打开浏览器访问 IP/index.php，能出来就成功了



小D课堂 愿景："让编程不在难学，让技术与生活更加有趣" 更多教程请访问 [xdclass.net](http://xdclass.net)

## 第六章：linux下的shell脚本编程介绍

### 6.1 shell编程语言的介绍

简介：介绍shell脚本的重要性与应用场景

- shell简介：

Shell是一种脚本语言，又是一种命令语言。可以通俗一点来讲，Shell脚本就是一系列命令的集合，可以在Unix/Linux上面直接使用，并且直接调用大量系统内部的功能来解释执行程序把一些重复性工作交给Shell做，来实现自动化运维。

Shell 虽然没有C/C++、Java、Python等强大，但也支持了基本的编程元素。例如：if、for、while、case等循环，还有变量、数组、字符串、注释、加减乘除逻辑运算等

- 常见的脚本语言：

```
shell、perl、php、python
```

- shell的优点：

- 易用 #直接在linux系统上使用，不需要编译
- 高效 #程序开发的效率非常高，依赖于功能强大的命令可以迅速地完成任务
- 简单 #语法和结构比较简单，易于掌握

- shell应用场景：

- 监控linux系统的健康度
- 数据的处理 #日志的切割、分析、统计等
- 与数据库交互 #对数据库进行增，删，改，查等操作
- 监控进程，自动化启停服务
- 完成一些重复性的工作

## 6.2 shell编写第一个脚本

简介：编写第一个脚本

- 编写：vi first.sh

```
# !/bin/bash

# 作者：小C

# 编写时间：2019-08-20

# 功能：this is my first shell !

echo "this is my first shell !"
```

- 执行：

```
sh first.sh

chmod 755 first.sh
./first.sh
```

## 6.3 企业实战之shell脚本与crontab定时器的运用

简介：介绍实际工作中crontab定时器的重要性

- crond服务：  
以守护进程方式在无需人工干预的情况下来处理着一系列作业和指令的服务
- crond服务的启停命令：

```
启动
systemctl start crond.service

查看状态：
systemctl status crond.service

停止
systemctl stop crond.service

重新启动
systemctl restart crond.service
```

- crontab定时器的使用

```
语法：crontab 【选项】
crontab -l #列出crontab有哪些任务
crontab -e #编辑crontab任务
crontab -r #删除crontab里的所有任务
```

```
内容格式：
* * * * * 级别 命令
分 时 日 月 周
```

- crontab的例子

```
每分钟执行
* * * * * 或者 */1 * * * *

每小时执行
0 * * * *

每天执行
```

```
0 0 * * *
```

每周执行

```
0 0 * * 0
```

每月执行

```
0 0 1 * *
```

每年执行

```
0 0 1 1 *
```

每天早上6点执行

```
0 6 * * *
```

每两个小时执行

```
0 */2 * * *
```

每小时的10分, 40分执行

```
10,40 * * * *
```

每天的下午4点、5点、6点的5 min、15 min、25 min、35 min、45 min、55 min时执行命令

```
5,15,25,35,45,55 16,17,18 * * *
```

## 6.4 利用shell脚本企业实战nginx日志切割

简介：演练企业中的nginx服务器日志定时切割

- 需求：
  - nginx的日志文件路径
  - 每天0点对nginx 的access与error日志进行切割
  - 以前一天的日期为命名
- 脚本：

```
#!/bin/bash
#Auto cut nginx log script.
#create by 小C
#create date:2019-08-21

#nginx日志路径
logs_path=/usr/local/nginx/logs
Yesterday=$(date -d 'yesterday' +%Y-%m-%d)

#移动日志并以日期改名
mv ${logs_path}/access.log ${logs_path}/access_${Yesterday}.log
mv ${logs_path}/error.log ${logs_path}/error_${Yesterday}.log

#向nginx主进程发送信号，重新生成日志文件
kill -USR1 $(cat /usr/local/nginx/logs/nginx.pid)
```





## 第七章：不得不学的企业实战篇

### 7.1 企业实战静态ip地址配置

简介：演练企业静态ip地址配置过程

- 网卡的路径：

```
/etc/sysconfig/network-scripts/ifcfg-enxxxxxxxx
```

- alias命令实现别名：

临时设置别名：

```
alias vinet='vi /etc/sysconfig/network-scripts/ifcfg-eno16777728'
```

查看别名：

```
alias
```

取消临时别名：

```
alias vinet
```

永久设置别名：vi /root/.bashrc

```
alias vinet='vi /etc/sysconfig/network-scripts/ifcfg-eno16777728'
```

加载使立即生效：

```
source /root/.bashrc
```

- 修改ip地址为静态

```
BOOTPROTO="static"  
IPADDR=xxx.xxx.xxx.xxx  
GATEWAY=xxx.xxx.xxx.xxx  
NETMASK=255.255.255.0  
ONBOOT="yes"
```

修改以上信息，以下是我网卡信息

```
HWADDR="00:0c:29:dc:47:58"  
TYPE="Ethernet"  
BOOTPROTO="static"  
DEFROUTE="yes"  
PEERDNS="yes"  
PEERROUTES="yes"  
IPV4_FAILURE_FATAL="no"  
IPADDR=192.168.10.100  
GATEWAY=192.168.10.2  
NETMASK=255.255.255.0  
IPV6INIT="yes"  
IPV6_AUTOCONF="yes"
```

```
IPV6_DEFROUTE="yes"
IPV6_PEERDNS="yes"
IPV6_PEERROUTES="yes"
IPV6_FAILURE_FATAL="no"
NAME="eno16777728"
UUID="3199add9-379c-43a9-bab1-ae4e05c0f2cc"
ONBOOT="yes"
```

- 重启网卡

```
systemctl restart network.service
```

- ping不通域名：

```
[root@localhost ~]# ping www.baidu.com
ping: unknown host www.baidu.com
```

解决：vi /etc/resolv.conf 加上以下域名服务器解析地址

```
nameserver 114.114.114.114
nameserver 8.8.8.8
nameserver 1.1.1.1
```

## 7.2 企业CentOS7实战修改linux系统主机名

简介：演练如何修改linux系统主机名

- 查看主机名命令：

```
hostname
```

- 修改主机名的命令：

```
hostnamectl set-hostname XDapp1
```

- 修改后记得重启使得生效：

```
reboot
```

- 修改/etc/hosts文件：vi /etc/hosts

```
192.168.10.100 xdapp1
```

- 测试能否ping通

```
[root@xdapp2 ~]# ping xdapp1
PING xdapp1 (192.168.10.100) 56(84) bytes of data.
64 bytes from xdapp1 (192.168.10.100): icmp_seq=1 ttl=64 time=0.332 ms
64 bytes from xdapp1 (192.168.10.100): icmp_seq=2 ttl=64 time=0.411 ms
```

## 7.3 企业CentOS7实战ssh免密远程登录其它机器

简介：演练如何免密码在服务器之间随意切换登录

- 执行命令创建密钥：ssh-keygen -t rsa

```
[root@xdapp1 ~]# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
51:98:6e:75:8a:c4:50:1e:84:5f:ce:24:7b:53:e5:92 root@xdapp1
The key's randomart image is:
+--[ RSA 2048 ]-----+
|      .+o.  ..  |
|      ..+o+ oo  |
|      +OX +E   |
|      *. *    .  |
|      .S. .    |
|                |
|                |
|                |
|                |
+-----+
[root@xdapp1 ~]#
```

- 从xdapp1拷贝文件到xdapp2并重命名为authorized\_keys

```
scp /root/.ssh/id_rsa.pub root@xdapp2:/root/.ssh/authorized_keys
```

- 测试能不能免密登录xdapp2

```
ssh xdapp2
```

## 7.4 企业CentOS7实战搭建nfs文件共享服务器

简介：演练nfs文件共享服务器搭建

- NFS介绍：

NFS ( Network File System ) 即网络文件系统，它允许网络中服务器之间通过TCP/IP网络共享资源，NFS的一个最大优点就是可以节省本地存储空间

- NFS体系至少有两个主要部分：

一台NFS服务器和若干台客户机

- 环境准备：

A : 192.168.10.100 作服务端  
B : 192.168.10.3 作客户端

- 在A服务端机器 安装nfs-utils和rpcbind包：

```
yum install -y nfs-utils rpcbind
```

nfs-utils包提供了NFS服务器程序和相应的管理工具  
rpc则是能够获得nfs服务器端的端口号等信息

- 检查是否启动了rpcbind服务并监听着 111端口：

```
netstat -tunlp | grep 111
```

- 配置NFS服务的配置文件，参数配置：vi /etc/exports

```
/data/NFSdata 192.168.10.3/24(rw,sync)
```

/data/NFSdata                   #表示你要共享的目录  
192.168.10.3/24                #表示所允许访问的客户端IP  
(rw,sync)                      #rw：表示读写权限，sync：表示数据同步写入内存硬盘

- 启动nfs服务

```
systemctl start nfs
```

- 设置开机启动：

```
systemctl enable rpcbind.service  
systemctl enable nfs-server.service
```

- 在B客户端机器 安装nfs-utils和rpcbind包：

```
yum install -y nfs-utils rpcbind
```

- 设置开机启动：

```
systemctl enable rpcbind.service
```

- 挂载：

```
mount -t nfs 192.168.10.100:/data/NFSdata /NFS  
showmount -e 192.168.10.100
```

## 7.5 企业CentOS7 实战部署ftp文件传输服务器(一)

### 简介：演练ftp文件传输服务器搭建

- FTP文件服务器概述:

是一种文件传输服务器，默认使用TCP协议的20、21端口与客户端进行通信，20端口用于建立数据连接，并传输文件数据；21端口用于建立控制连接，并传输FTP控制命令；

- FTP数据连接分为主动模式和被动模式:

主动模式:服务器向客户端主动发起数据连接 ( PORT 模式 )  
被动模式:客户端向服务器主动发起数据连接 ( PASV 模式 )

- FTP服务器软件的种类

凡是可通过文件传输协议(File Transport Protocol)下载文件服务器上的文件, 以实现资源共享的, 我们称之为ftp服务器

FTP服务器软件包括Filezilla Sener、Serv-U、IIS等。而在Linux系统中, vsftpd是目前在Linux/UNIX领域应用比较广泛的一款FTP服务软件, 特点是速度快, 安全易用, 并发量高, 可以支持15000个用户并发连接

- FTP用户类型:

匿名用户: 就是不需要用户名, 密码。就能登录到服务器电脑里面

本地用户: 需要帐户名和密码才能登录。而且, 这个帐户名和密码, 都是在你 linux 系统里面, 已经有的用户。

虚拟用户: 同样需要用户名和密码才能登录。但是和上面的区别就是, 在linux 系统中是没有该用户帐号

- 搭建匿名访问 vsftpd 服务:

检查是否已经安装vsftpd软件包:

```
rpm -qa vsftpd
```

yum 安装 vsftpd:

```
yum -y install vsftpd
```

查找 vsftpd 的文件位置:

```
whereis vsftpd
```

启动 ftp 服务:

```
systemctl start vsftpd
```

查看监听的端口号:

```
netstat -tunlp | grep vsftpd
```

查看防火墙的状态:

```
firewall-cmd --state
```

创建测试文件:

```
mkdir -p /var/ftp/pub/ && echo "this is ftptest" >test.txt
```

使用 windows 客户端访问:

```
ftp://192.168.10.100/
```

安装 FTP 客户端工具:

```
yum -y install ftp
```

连接vsftpd:

```
ftp 192.168.10.100
```

下载文件:

```
get test.txt
```

```
上传文件：
put test.txt

退出：
quit 或者 exit
```

- 配置本地用户访问 vsftpd 服务：

```
关闭之前的匿名访问：
vi /etc/vsftpd/vsftpd.conf    #编辑配置文件
anonymous_enable=NO

systemctl restart vsftpd      #重新启动服务

mkdir -p /var/ftp/html        #创建目录

useradd -g root -M -d /var/ftp/html -s /sbin/nologin ftptest #添加系统用户

chown -R ftptest:root /var/ftp/html #配置权限

passwd ftptest #设置密码

cd /var/ftp/html && echo "this is ftptest" >ftptest.txt #添加测试文件

使用 windows 客户端访问：
ftp://192.168.10.100/

配置文件/etc/vsftp/vsftpd.conf 的一些常用权限解释：
anonymous_enable=YES 允许匿名用户访问
write_enable=YES 允许登录用户有写权限。属于全局设置
anon_root=/var/ftp 指定匿名用户登录的目录
anon_upload_enable=YES 允许匿名用户有上传文件的权限
anon_mkdir_write_enable=YES 允许匿名用户有新增目录的权限
chown_username=username 设置匿名用户上传文件的属主名
```

## 7.6 企业CentOS7 实战部署ftp文件传输服务器(二)

简介：演练ftp配置虚拟用户

- 配置虚拟用户的 vsftpd 服务
  - 虚拟用户访问 vsftpd 服务是得经过用户认证的，虚拟用户的用户认证是通过 pam 方式去认证的，pam 文件里面指定了认证的db 文件，db 文件又是通过明文用户名和密码文件生成而来
    - 第一步：修改配置文件指定用户认证是pam方式认证
    - 第二步：修改pam配置文件指定db文件的路径

### ■ 第三步：生成db文件

(1) 第一步指定 pam 文件(系统默认已经指定了):

```
vi /etc/vsftpd/vsftpd.conf
pam_service_name=vsftpd
```

(2) 第二步编辑pam 文件指定认证的db 文件：/etc/pam.d/vsftpd，把文件内容全部注释掉，在最上面 加上以下俩行配置：

```
auth required /lib64/security/pam_userdb.so db=/etc/vsftpd/vuser_passwd
account required /lib64/security/pam_userdb.so db=/etc/vsftpd/vuser_passwd
```

```
[root@xdapp1 html]# vi /etc/pam.d/vsftpd
auth required /lib64/security/pam_userdb.so db=/etc/vsftpd/vuser_passwd
account required /lib64/security/pam_userdb.so db=/etc/vsftpd/vuser_passwd
```

```
##PAM-1.0
#session    optional    pam_keyinit.so      force revoke
#auth       required    pam_listfile.so     item=user sense=deny
#           file=/etc/vsftpd/ftpusers onerr=succeed
#auth       required    pam_shells.so
#auth       include     password-auth
#account    include     password-auth
#session    required    pam_loginuid.so
#session    include     password-auth
```

(3) 第三步生成 db 文件，由于 db 文件是通过明文用户名和密码文件生成而来，所以先创建一个保存明文用户名和密码的文件 vuser\_passwd.txt

```
vi /etc/vsftpd/vuser_passwd.txt    #该文件奇行为用户名，偶行为密码
```

通过以下命令生成 db 文件：

```
cd /etc/vsftpd/ && db_load -T -t hash -f vuser_passwd.txt vuser_passwd.db
```

(4) 接下来就是根据需求和以上各文件信息来修改配置文件 /etc/vsftpd/vsftpd.conf 了，启用或更改以下配置的值：

```
vi /etc/vsftpd/vsftpd.conf
```

```
anonymous_enable=NO                #禁用匿名登录
ascii_upload_enable=YES
ascii_download_enable=YES
chroot_local_user=YES              #启用限定用户在其主目录下
手动添加：
guest_enable=YES                   #设定启用虚拟用户功能
guest_username=ftptest             #指定虚拟用户的宿主用户，CentOS 中已经有内置的ftp 用户了
user_config_dir=/etc/vsftpd/vuser_conf #虚拟用户配置文件存放的路径
allow_writeable_chroot=YES         #如果启用了限定用户在其主目录下需要添加这个配置
```



```
(6) 对虚拟用户权限与路径进行配置
mkdir /etc/vsftpd/vuser_conf          #创建一个目录存放文件
vi /etc/vsftpd/vuser_conf/xiaoming    #创建一个以虚拟用户名的文件(文件名一定要与登录名一样)

[root@xdapp1 vsftpd]# vi vuser_conf/xiaoming
local_root=/var/ftp/xiaoming
write_enable=YES
anon_umask=022
anon_world_readable_only=NO
anon_upload_enable=YES
anon_mkdir_write_enable=YES
anon_other_write_enable=yes
```

- 配置系统用户的 FTP与配置虚拟用户的 FTP 共存：

修改配置文件/etc/pam.d/vsftpd, 把原来的auth required改为auth sufficient

```
[root@xdapp1 html]# vi /etc/pam.d/vsftpd
auth sufficient /lib64/security/pam_userdb.so db=/etc/vsftpd/vuser_passwd
account sufficient /lib64/security/pam_userdb.so db=/etc/vsftpd/vuser_passwd

# %PAM-1.0
session optional pam_keyinit.so force revoke
auth required pam_listfile.so item=user sense=deny file=/etc/vsftpd/ftpusers
onerr=succeed
auth required pam_shells.so
auth include password-auth
account include password-auth
session required pam_loginuid.so
session include password-auth
```

## 7.7 云服务器介绍和阿里云服务器ECS服务器选购

简介：什么是云服务器及目前主要的几个厂商介绍

- 阿里云、腾讯云、亚马逊云

阿里云：<https://www.aliyun.com/>  
腾讯云：<https://cloud.tencent.com/>  
亚马逊云：<https://aws.amazon.com/>

## 7.8 域名购买和配置解析实战

简介：域名购买和配置解析实战

- 购买域名，备案

阿里云 备案地址：<https://beian.aliyun.com/>

- 购买服务器，阿里云，腾讯云，亚马逊云aws
- 配置域名解析到服务器



小吊课堂 愿景："让编程不在难学，让技术与生活更加有趣" 更多教程请访问 [xdclass.net](http://xdclass.net)

## 第八章：linux Centos 7课程总结与知识体系图

---

### 8.1 liunx课程总结与整套知识体系图

简介：课程总结与整套知识体系

