

1.Linux下/etc/下有哪些配置文件，有什么作用？ resolve.conf是用来做什么的？

/etc 目录包含各种系统配置文件：/etc 是 UNIX®（LINUX）系统最基本的目录之一，因为它包含了全部特定主机的配置文件。绝不要为了回收空间而删除它！同样，如果您想要将您的文件树结构拓展到几个分区时，请记住 /etc 一定不能放到一个另外的分区中。系统初始化需要它，因此它必须在启动时存放于根分区。

1)./etc/passwd

用户数据库，其中的域给出了用户名、真实姓名、用户起始目录、加密口令和用户的其他信息。

2)./etc/shadow

在安装了影子(shadow)口令软件的系统上的影子口令文件。影子口令文件将/etc/passwd文件中的加密口令移动到/etc/shadow中，而后者只对超级用户(root)可读。这使破译口令更困难，以此增加系统的安全性。

3)./etc/inittab

init 的配置文件。

4)./etc/profile、/etc/csh.login、/etc/csh.cshrc

登录或启动时bourne或c shells执行的文件。这允许系统管理员为所有用户建立全局缺省环境。

5).linux根文件系统 /etc/resolv.conf 文件详解

它是DNS客户机配置文件，用于设置DNS服务器的IP地址及DNS域名，还包含了主机的域名搜索顺序。该文件是由域名解析器（resolver，一个根据主机名解析IP地址的库）使用的配置文件。它的格式很简单，每行以一个关键字开头，后接一个或多个由空格隔开的参数。resolv.conf的关键字主要有四个，分别是：

- nameserver //定义DNS服务器的IP地址
- domain //定义本地域名
- search //定义域名的搜索列表
- sortlist //对返回的域名进行排序

下面我们给出一个/etc/resolv.conf的示例：

domain 51osos.com

search www.51osos.com 51osos.com

nameserver 202.102.192.68

nameserver 202.102.192.69

- 最主要是nameserver关键字，如果没指定nameserver就找不到DNS服务器，其它关键字是可选的，nameserver表示解析域名时使用该地址指定的主机为域名服务器。其中域名服务器是按照文件中出现的顺序来查询的,且只有当第一个nameserver没有反应时才查询下面的nameserver。
- domain 声明主机的域名。很多程序用到它，如邮件系统；当为没有域名的主机进行DNS查询时，也要用到。如果没有域名，主机名将被使用，删除所有在第一个点(.)前面的内容。
- search 它的多个参数指明域名查询顺序。当要查询没有域名的主机，主机将在由search声明的域中分别查找。
- domain和search不能共存；如果同时存在，后面出现的将会被使用。
- sortlist 允许将得到域名结果进行特定的排序。它的参数为网络/掩码对，允许任意的排列顺序。

2.Linux的七个运行级别

- 运行级别0：系统停机状态，系统默认运行级别不能设为0，否则不能正常启动
- 运行级别1：单用户工作状态，root权限，用于系统维护，禁止远程登陆
- 运行级别2：多用户状态(没有NFS)
- 运行级别3：完全的多用户状态(有NFS)，登陆后进入控制台命令行模式
- 运行级别4：系统未使用，保留
- 运行级别5：X11控制台，登陆后进入图形GUI模式
- 运行级别6：系统正常关闭并重启，默认运行级别不能设为6，否则不能正常启动

运行级别的原理(以ubuntu说明):

- 在目录/etc/init.d下有许多服务器脚本程序，一般称为服务(service)
- 在/etc下有7个名为rcN.d的目录，对应系统的7个运行级别
- rcN.d目录下都是一些符号链接文件，这些链接文件都指向init.d目录下的service脚本文件，命名规则为K+nn+服务名或S+nn+服务名，其中nn为两位数字。
- 系统会根据指定的运行级别进入对应的rcN.d目录，并按照文件名顺序检索目录下的链接文件，对于以K开头的文件，系统将终止对应的服务，对于以S开头的文件，系统将启动对应的服务
- 查看运行级别用：runlevel
- 进入其它运行级别用：init N
- 另外init0为关机，init 6为重启系统

3.内核态与用户态的区别

4.用户态与内核态之间的转换

- 系统调用：这是用户态进程主动要求切换到内核态的一种方式，用户态进程通过系统调用申请使用操作系统提供的服务程序完成工作，比如前例中fork()实际上就是执行了一个创建新进程的系统调用。而系统调用的机制其核心还是使用了操作系统为用户特别开放的一个中断来实现，例如Linux的int 80h中断。
- 异常:当CPU在执行运行在用户态下的程序时，发生了某些事先不可知的异常，这时会触发由当前运行进程切换到处理此异常的内核相关程序中，也就转到了内核态，比如缺页异常。
- 外围设备的中断:当外围设备完成用户请求的操作后，会向CPU发出相应的中断信号，这时CPU会暂停执行下一条即将要执行的指令转而去执行与中断信号对应的处理程序，如果先前执行的指令是用户态下的程序，那么这个转换的过程自然也就发生了由用户态到内核态的切换。比如硬盘读写操作完成，系统会切换到硬盘读写的中断处理程序中执行后续操作等。

这3种方式是系统在运行时由用户态转到内核态的最主要方式，其中系统调用可以认为是用户进程主动发起的，异常和外围设备中断则是被动的。从触发方式上看，可以认为存在前述3种不同的类型，但是从最终实际完成由用户态到内核态的切换操作上来说，涉及的关键步骤是完全一致的，没有任何区别，都相当于执行了一个中断响应的过程，因为系统调用实际上最终是中断机制实现的，而异常和中断的处理机制基本上也是一致的，关于它们的具体区别这里不再赘述。关于中断处理机制的细节和步骤这里也不做过多分析，涉及到由用户态切换到内核态的步骤主要包括：

- 从当前进程的描述符中提取其内核栈的ss0及esp0信息。
- 使用ss0和esp0指向的内核栈将当前进程的cs,eip,eflags,ss,esp信息保存起来，这个过程也完成了由用户栈到内核栈的切换过程，同时保存了被暂停执行的程序的下一条指令。
- 将先前由中断向量检索得到的中断处理程序的cs,eip信息装入相应的寄存器，开始执行中断处理程序，这时就转到了内核态的程序执行了。

5.linux命令

利用Linux所提供的管道符“|”将两个命令隔开，管道符左边命令的输出就会作为管道符右边命令的输入。连续使用管道意味着第一个命令的输出会作为第二个命令的输入，第二个命令的输出又会作为第三个命令的输入，依此类推。

(Linux) 例如找到一个文件中含有aaa而不含bbb的行 `grep -rl "aaa" * | grep -v "bbb"`

cd命令

这是一个非常基本，也是大家经常需要使用的命令，它用于切换当前目录，它的参数是要切换到的目录的路径，可以是绝对路径，也可以是相对路径。如：

- `cd /root/Docements` # 切换到目录/root/Docements
- `cd ./path` # 切换到当前目录下的path目录中，“.”表示当前目录
- `cd ../path` # 切换到上层目录中的path目录中，“..”表示上一层目录

ls命令

这是一个非常有用的查看文件与目录的命令，list之意，它的参数非常多，下面就列出一些我常用的参数吧，如下：

- `-l`：列出长数据串，包含文件的属性与权限数据等
- `-a`：列出全部的文件，连同隐藏文件（开头为.的文件）一起列出来（常用）
- `-d`：仅列出目录本身，而不是列出目录的文件数据
- `-h`：将文件容量以较易读的方式（GB，kB等）列出来
- `-R`：连同子目录的内容一起列出（递归列出），等于该目录下的所有文件都会显示出来
- `ls -l #`以长数据串的形式列出当前目录下的数据文件和目录
- `ls -lR #`以长数据串的形式列出当前目录下的所有文件

grep命令

该命令常用于分析一行的信息，若当中有我们所需要的信息，就将该行显示出来，该命令通常与管道命令一起使用，用于对一些命令的输出进行筛选加工等等，它的简单语法为

`grep [-acinv] [--color=auto] '查找字符串' filename`

例如# 取出文件/etc/man.config中包含MANPATH的行，并把找到的关键字加上颜色：`grep --color=auto 'MANPATH' /etc/man.config`

把ls -l的输出中包含字母file（不区分大小写）的内容输出：`ls -l | grep -i file`
常用参数有：

- `-a`：将binary文件以text文件的方式查找数据
- `-c`：计算找到‘查找字符串’的次数
- `-i`：忽略大小写的区别，即把大小写视为相同
- `-v`：反向选择，即显示出没有‘查找字符串’内容的那一行

find命令

find是一个基于查找的功能非常强大的命令，相对而言，它的使用也相对较为复杂，参数也比较多，所以在这里将给把它们分类列出，它的基本语法如下：

find [PATH] [option] [action]

1).首先与时间有关的参数有：

- -mtime n : n为数字，意思为在n天之前的“一天内”被更改过的文件；
- -mtime +n : 列出在n天之前（不含n天本身）被更改过的文件名；
- -mtime -n : 列出在n天之内（含n天本身）被更改过的文件名；
- -newer file : 列出比file还要新的文件名

例如： find /root -mtime 0 : 在当前目录下查找今天之内有改动的文件

2).与用户或用户组名有关的参数：

- -user name : 列出文件所有者为name的文件
- -group name : 列出文件所属用户组为name的文件
- -uid n : 列出文件所有者为用户ID为n的文件
- -gid n : 列出文件所属用户组为用户组ID为n的文件

例如： find /home/ljianhui -user ljianhui : 在目录/home/ljianhui中找出所有者为ljianhui的文件

3).与文件权限及名称有关的参数：

- -name filename : 找出文件名为filename的文件
- -size [+ -]SIZE : 找出比SIZE还要大 (+) 或小 (-) 的文件
- -type TYPE : 查找文件的类型为TYPE的文件，TYPE的值主要有：一般文件 (f)、设备文件 (b、c)、目录 (d)、连接文件 (l)、socket (s)、FIFO管道文件 (p)；
- -perm mode : 查找文件权限刚好等于mode的文件，mode用数字表示，如0755；
- -perm -mode : 查找文件权限必须要全部包括mode权限的文件，mode用数字表示

- `-perm +mode` : 查找文件权限包含任一mode的权限的文件, mode用数字表示

例如:

`find / -name passwd` # 查找文件名为passwd的文件

`find . -perm 0755` # 查找当前目录中文件权限的0755的文件

`find . -size +12k` # 查找当前目录中大于12KB的文件, 注意c表示byte

cp命令

该命令用于复制文件, copy之意, 它还可以把多个文件一次性地复制到一个目录下, 它的常用参数如下:

- `-a` : 将文件的特性一起复制
- `-p` : 连同文件的属性一起复制, 而非使用默认方式, 与-a相似, 常用于备份
- `-i` : 若目标文件已经存在时, 在覆盖时会先询问操作的进行
- `-r` : 递归持续复制, 用于目录的复制行为
- `-u` : 目标文件与源文件有差异时才会复制

例如: `cp -a file1 file2` : 连同文件的所有特性把文件file1复制成文件file2

`cp file1 file2 file3 dir` : 把文件file1、file2、file3复制到目录dir中

mv命令

- 该命令用于移动文件、目录或更名, move之意, 它的常用参数如下:
- `-f` : force强制的意思, 如果目标文件已经存在, 不会询问而直接覆盖
- `-i` : 若目标文件已经存在, 就会询问是否覆盖
- `-u` : 若目标文件已经存在, 且比目标文件新, 才会更新

注: 该命令可以把一个文件或多个文件一次移动一个文件夹中, 但是最后一个目标文件一定要是“目录”。例如: `mv file1 file2 file3 dir` : 把文件file1、file2、file3移动到目录dir中

`mv file1 file2` : 把文件file1重命名为file2

rm命令

该命令用于删除文件或目录，remove之意，它的常用参数如下：

- -f：就是force的意思，忽略不存在的文件，不会出现警告消息
- -i：互动模式，在删除前会询问用户是否操作
- -r：递归删除，最常用于目录删除，它是一个非常危险的参数

例如：

rm -i file # 删除文件file，在删除之前会询问是否进行该操作

rm -fr dir # 强制删除目录dir中的所有文件

ps命令

该命令用于将某个时间点的进程运行情况选取下来并输出，process之意，它的常用参数如下：

- -A：所有的进程均显示出来
- -a：不与terminal有关的所有进程
- -u：有效用户的相关进程
- -x：一般与a参数一起使用，可列出较完整的信息
- -l：较长，较详细地将PID的信息列出

其实我们只要记住ps一般使用的命令参数搭配即可，它们并不多，如下：

- ps aux # 查看系统所有的进程数据
- ps ax # 查看不与terminal有关的所有进程
- ps -lA # 查看系统所有的进程数据
- ps axjf # 查看连同一部分进程树状态

kill命令

该命令用于向某个工作（%jobnumber）或者是某个PID（数字）传送一个信号，它通常与ps和jobs命令一起使用，它的基本语法如下：kill -signal PID
signal的常用参数如下：

注：最前面的数字为信号的代号，使用时可以用代号代替相应的信号。

- SIGHUP, 启动被终止的进程
- SIGINT, 相当于输入ctrl+c, 中断一个程序的进行
- SIGKILL, 强制中断一个进程的进程
- SIGTERM, 以正常的结束进程方式来终止进程
- SIGSTOP, 相当于输入ctrl+z, 暂停一个进程的进程

例如:

以正常的结束进程方式来终止第一个后台工作, 可用jobs命令查看后台中的第一个工作进程

```
kill -SIGTERM %1
```

重新改动进程ID为PID的进程, PID可用ps命令通过管道命令加上grep命令进行筛选获得

```
kill -SIGHUP PID
```

killall命令

该命令用于向一个命令启动的进程发送一个信号, 它的一般语法如下:

```
killall [-ile] [command name]
```

它的参数如下:

- -i : 交互式的意思, 若需要删除时, 会询问用户
- -e : 表示后面接的command name要一致, 但command name不能超过15个字符
- -l : 命令名称忽略大小写

例如: killall -SIGHUP syslogd # 重新启动syslogd

file命令

该命令用于判断接在file命令后的文件的基本数据, 因为在Linux下文件的类型并不是以后缀为分的, 所以这个命令对我们来说就很有用了, 它的用法非常简单, 基本语法如下:

```
file filename
```

例如: file ./test

tar命令

该命令用于对文件进行打包，默认情况并不会压缩，如果指定了相应的参数，它还会调用相应的压缩程序（如gzip和bzip等）进行压缩和解压。它的常用参数如下：

- -c：新建打包文件
- -t：查看打包文件的内容含有哪些文件名
- -x：解打包或解压缩的功能，可以搭配-C（大写）指定解压的目录，注意-c,-t,-x不能同时出现在同一条命令中
- -j：通过bzip2的支持进行压缩/解压缩
- -z：通过gzip的支持进行压缩/解压缩
- -v：在压缩/解压缩过程中，将正在处理的文件名显示出来
- -f filename：filename为要处理的文件
- -C dir：指定压缩/解压缩的目录dir

通常我们只需要记住下面三条命令即可：

- 压缩：tar -jcv -f filename.tar.bz2 要被处理的文件或目录名称
- 查询：tar -jtv -f filename.tar.bz2
- 解压：tar -jxv -f filename.tar.bz2 -C 欲解压缩的目录

注：文件名并不定要以后缀tar.bz2结尾，这里主要是为了说明使用的压缩程序为bzip2

cat命令

该命令用于查看文本文件的内容，后接要查看的文件名，通常可用管道与more和less一起使用，从而可以一页页地查看数据。例如：

cat text | less # 查看text文件中的内容

注：这条命令也可以使用less text来代替

chgrp命令

该命令用于改变文件所属用户组，它的使用非常简单，它的基本用法如下：

chgrp [-R] dirname/filename

-R：进行递归的持续对所有文件和子目录更改

例如: `chgrp users -R ./dir #` 递归地把dir目录下中的所有文件和子目录下所有文件的用户组修改为users

chown命令

该命令就是改变文件拥有者和所在用户组。每个文件都属于一个用户组和一个用户。在你的目录下, 使用"`ls -l`", 你就会看到像这样的东西。

```
root@tecmint:~# ls -l
```

```
drwxr-xr-x 3 server root 4096 May 10 11:14 Binary
```

```
drwxr-xr-x 2 server server 4096 May 13 09:42 Desktop
```

在这里, 目录Binary属于用户"server", 和用户组"root", 而目录"Desktop"属于用户 "server" 和用户组"server"

chown命令用来改变文件的所有权, 所以仅仅用来管理和提供文件的用户和用户组授权。例如:

```
root@tecmint:~# chown server:server Binary
```

```
drwxr-xr-x 3 server server 4096 May 10 11:14 Binary
```

```
drwxr-xr-x 2 server server 4096 May 13 09:42 Desktop
```

chmod命令

该命令用于改变文件的权限, 一般的用法如下:

`chmod [-R] xyz 文件或目录`

-R: 进行递归的持续更改, 即连同子目录下的所有文件都会更改

同时, chmod还可以使用u (user)、g (group)、o (other)、a (all) 和 + (加入)、- (删除)、= (设置) 跟rwx搭配来对文件的权限进行更改。例如:

- `chmod 0755 file #` 把file的文件权限改变为-rxwr-xr-x
- `chmod g+w file #` 向file的文件权限中加入用户组可写权限
- Read (r)=4
- Write(w)=2
- Execute(x)=1

3种用户和用户组权限。第一个是拥有者, 然后是用户所在的组, 最后是其它用户。

vim命令

该命令主要用于文本编辑，它接一个或多个文件名作为参数，如果文件存在就打开，如果文件不存在就以该文件名创建一个文件。vim是一个非常好用的文本编辑器，它里面有很多非常好用的命令。

df命令

df命令是Linux查看磁盘空间系统以磁盘分区为单位查看文件系统，可以加上参数查看磁盘剩余空间信息，命令格式：

- df -hl 查看磁盘剩余空间
- df -h 查看每个根路径的分区大小
- du -sh [目录名] 返回该目录的大小
- du -sm [文件夹] 返回该文件夹总M数