Linux笔记

1. **Linux概述**

为什么服务器尤其大型服务器都使用Linux系统

服务器尤其是大型服务器一般都使用Linux系统，有以下几点原因：

1、成本低，Linux操作系统是免费的

2、安全性好，Linux采取了许多的安全措施，包括对读写权限控制、带保护的子系统、审计跟踪、核心授权等，为网络多用户环境中的用户提供了必要的安全保障

3、网络功能强大，Linux在通信和网络功能方面优于其他操作系统

4、开源性，这意味着管理员可以对系统随意进行定制

Linux用户和组的概念

1、用户

用户是能够获取系统资源的权限的集合

2、Linux用户组分类

（1）管理员root：具有使用系统所有权限的用户，UID为0

（2）普通用户：即一般用户，其使用系统的权限受限，UID在500-60000之间

（3）系统用户：保障系统运行的用户，一般不提供密码登录系统，UID在1-499之间

3、与用户相关的文件

（1）/etc/passwd

其格式为account:password:UID:GID:GECOS:diretory:shell

a）account表示用户名或账号

b）password表示用户密码占位符

c）UID表示用户ID号

d）GID表示用户所在组的ID号

e）diretory表示用户所在的家目录

f）shell表示用户所在的编程环境

我自己家里面装了一个ubuntu12，该文件的内容为：

（2）/etc/shadow

其格式为account:password:最近更改密码的日期:密码不可更改的天数:密码需要重新更改的天数:密码更改前的告警期限:密码过期的宽限时间:账号失效日期:保留

4、用户组分类

（1）普通用户组：可以加入多个用户

（2）系统组：一般加入一些系统用户

（3）私有组：当创建用户时，如果没有为其指明所属组，则为其定义一个私有的用户组，其名称与用户名同名。如果把其他用户加入了该私有组中，那么私有组就变成了普通组

同组用户有一样的权限

5、与用户组相关的文件

/etc/group，其格式为group\_name:password:GID:user\_list

（1）group\_name表示组名

（2）password表示组密码

（3）GID表示组ID号

（4）user\_list表示已group\_name为附加组的用户列表

su和sudo的区别

1、su

su就是切换用户的工具。比如以用户A登录，但要添加用户任务，执行useradd，A用户并没有这个权限，而这个权限恰恰root拥有。解决办法无非两个，一个是退出A，以root登录；另一个是通过su切换到root，等任务完成后再退出root。

注意，"su"和"su -"同样是切换到root用户，二者之间还是有区别的。前者只是切换了root身份，但是Shell环境仍然是普通用户的Shell；后者连用户和Shell环境一起切换成了root身份。只有切换了Shell环境才不会出现PATH环境变量错误，可以通过pwd命令验证。

su确实给管理带来了方便，通过切换到root下，就能完成所有系统管理，只要把root密码交给任何一个普通用户，他都能切换到root来完成所有的系统管理工作。但通过su切换到root也有不安全的因素，多用户操作时，只要有任何一个人操作失误，都可能造成系统崩溃或数据损失。

2、sudo

通过sudo，我们能把某些超级权限有针对性地下方，并且不需要普通用户知道root密码，所以sudo相对于权限无限制的su来说，还是比较安全的，所以sudo也称为受限制的su。另外sudo是需要授权许可的，所以它也被称为授权许可的su。

sudo执行命令的流程是当前用户切换到root（或其它执行切换到的用户），然后以root（或其他执行切换到的用户）身份执行命令，执行完成后，直接退回到当前用户。而这些的前提是要通过sudo配置文件/etc/sudoers来进行授权。

文件权限

我截取自己ubuntu下某路径下的文件列表：

详细解释一下：

1、第一列表示文件/文件夹的种类和权限

（1）第一个字符表示文件种类，"d"表示目录，"-"表示普通文件

（2）第二至第四个字符表示文件/文件夹所有者的权限，r表示读权限、w表示写权限、x表示执行权限

（3）第五至第七个字符表示文件/文件夹所属群组的权限，r表示读权限、w表示写权限、x表示执行权限

（4）第八至第十个字符表示文件/文件夹其他用户的权限，r表示读权限、w表示写权限、x表示执行权限

2、第二列表示文件/文件夹的连接数

3、第三列表示文件/文件夹的拥有者

4、第四列表示文件/文件夹所属的组

5、第五列表示文件/文件夹大小

6、第六列表示文件/文件夹最后修改时间

7、第七列表示文件/文件夹名称

1. **常用命令**

**2.1 vi、ls、cd、pwd、mkdir、rm、mv、cp、cat、tail**

**vi**

**Linux环境下的文本编辑器**

vi基本上可以分为三种操作状态，分别是命令模式、插入模式、底线命令模式，各功能的功能区分如下：

1、命令模式

控制屏幕光标的移动，字符或光标的删除，移动复制某区段及进入插入模式

2、插入模式

只有在插入模式下，才可以做文字数据输入，按ESC等可以回到命令模式

3、底线命令模式

将存储文件或离开编辑器，也可以设置编辑环境，如寻找字符串、列出行号等

注意，进入vi后是处于命令模式下的，要切换到插入模式才能输入文字。在命令模式下按"i"、"a"或是"o"切换到插入模式。在插入模式下，有一些命令：

i：插入，从目前光标所在之处插入所输入的文字

a：增加，从目前光标所在的下一个字开始输入文字

o：插入新的一行，从行首开始输入文字

在插入模式下，只能一直打字，如果发现字打错了，要先按ESC切换回命令模式，再把光标往回移动，最后删除打错的字符。删除有一些命令：

x：每按一次删除光标所在位置的后面一个字符

#x：例如6x表示删除光标所在位置后面的6个字符

dd：删除光标所在行

这是命令模式，在命令模式下按":"、"/"、"?"都可以进入底线命令模式，底线命令模式下有一些命令：

:set nu：在文章的每一行前面列出行号

/关键字：向后搜索关键字，假如不是想要的，按n继续向后搜索

?关键字：向前搜索关键字，假如不是想要的，按n继续向前搜索

:w：保存文件

**ls**

**用来打印当前目录的清单，可以指定其他目录**

-a：列出目录下所有的文件，包括以"."开头的隐藏文件

-l：除了文件名外，还列出文件权限、所有者、大小等详细信息

-r：反向次序排列

-t：以文件修改时间排序

-g：类似-l，但是不列出所有者

-h：以容易理解的文件格式列出（如1K、234M、2G）

例如：

ls -l t\*：列出所有文件名以"t"开头的文件信息

**cd**

**切换文件目录至dirName**

例如：

cd /：进入系统根目录

cd aa/bb/cc：跳转到指定目录，注意这里"aa"前没有"/"，指从当前目录跳转到指定目录

cd ~：进入系统主目录

cd ..：跳转到上一级

**pwd**

**查看当前工作目录的完整路径**

**mkdir**

**用来创建指定的名称的目录，要求创建目录的用户在当前目录中具有写权限，并且指定的目录名不能是当前目录中已有的目录**

-m：创建目录的时候设定权限

-p：若创建目录的路径中有些路径不存在，可一次性地建立多个目录

例如：

mkdir test1：创建一个空目录test1

mkdir -p test1/test2：递归创建多个目录

mkdir -m 777 test：在创建目录test3的时候给此目录赋予777的权限

**rm**

**删除一个目录中的一个或多个文件或目录，它可以将某个目录及其下所有文件及子目录删除。对于链接文件，只是删除了链接，原有文件均保持不变**

-f：强制删除且不给出提示

-r：将参数中列出的目录路和子目录均递归删除

-v：显示详细进行的步骤

例如：

rm -rf \*.log：删除任何.log文件并且不询问用户

**mv**

**可以用来移动文件或者将文件改名，常用来备份文件或目录**

例如：

mv log1.txt log.txt test3：将log1.txt、log2.txt移动到test3文件夹中

mv dir1 dir2：如果dir2不存在，则将dir1改名为dir2；如果dir2存在，则将dir1移入dir2中

mv \* ../：移动当前文件夹下所有文件至上一级目录

mv test3/\*.txt test5：将test目录下所有的.txt文件全部移动到test5文件夹下

**cp**

**复制文件或目录**

例如：

cp log.log test6：将log.log复制进入test6目录下

**cat**

**cat主要有三个功能：**

**1、一次性显示整个文件**

**2、从键盘创建一个文件**

**3、将几个文件合并为一个文件**

顺便提一句，tac这个命令和cat一样，无非是将文件内容从最后一行到第一行显示而已。

-E：在每行结束显示"$"

-n：对输出的所有行进行从1开始的编号

-s：对连续两行的空白行代替为一行

-T：将跳格字符（tab）显示为"^I"

例如：

cat -n log1.log log2.log：将log1.log加上行号后输入到log2.log文件中，log2.log在目录中必须存在

cat -n log1.log > log.log：将log1.log加上行号后输入到log.log文件中，log.log在目录中不存在

**tail**

**用于显示指定文件末尾内容，不指定文件时，作为信息输入进行处理。常用查看日志文件**

-f：不断读取，循环刷新，这样可以看到最新的文件内容

-q：不显示处理信息

-v：显示详细的处理信息

-c<数目>：显示的字节数

-n<行数>：显示的行数

例如：

tail -n 5 log1.log：显示log1.log最后5行的内容

tail -f log1.log：循环查看log1.log的内容

**2.2 more、which、find、chmod、tar、diff、grep、ps、netstat、uname**

**more**

**类似cat，不过more不是将整个文件内容从上到下显示在屏幕上的，而是以一页一页的显示方便使用者逐页阅读。more最基本的指令就是space即往下翻一页，b即往回翻一页显示，而且还有搜索字符串的功能**

+n：从第n行开始显示

-n：定义屏幕大小为n行

+/pattern：在每个档案显示前搜索该字符串patern，然后从该字符串前两行之后显示

-p：通过清除窗口而不是滚屏来对文件进行换页，与-c选项相似

-s：把连续的多个空行显示为一行

-u：把文件内容中的下划线去掉

常用操作命令：

Enter：向下n行，需要定义，默认为1行

空格键：向下滚动一屏

b：向上滚动一屏

=：输出当前行号

V：调用vi编辑器

q：退出more

例如：

more +3 log1.log：从第3行起显示log1.log里面的内容

more +/day3 log1.log：从文件中查找第一个出现'day3'的字符串的行，并从该行前两行开始显示输出

more -5 log1.log：设定每屏显示行数

ls -l | more -5：more常与别的命令连用，一般是因为某些命令输出的内容太多，所以使用more来分页显示

**which**

**在PATH变量指定的路径中，搜索某个系统命令的位置，并返回第一个搜索结果。即，which命令可以看到某个系统命令是否存在以及执行的到底是哪一个位置的命令**

**find**

**用于在文件树中查找文件并作相应的处理**

-name：按照文件名查找文件

-perm：按照文件权限查找文件

-user：按照文件属主来查找文件

-group：按照文件所属的组来查找文件

-mtime<-n><+n>：按照文件更改时间来查找文件，-n表示n天之内，+n表示n天以前

-newer file1 file2：查找更改时间在file1和file2之间的文件

例如：

find -mtime -2：查找48小时修改过的文件

find -name "\*.log"：查找当前目录以"\*.log"结尾的文件

find /opt/soft/test -perm 777：查找指定目录下权限为777的文件

find -type f -name "\*.log"：查找当前目录下以".log"结尾的普通文件

find -type d | sort：查找当前所有目标目录并排序

**chmod**

**用于改变Linux系统文件或目录的访问权限**

-f：错误信息不输出

-c：当发生改变时，报告处理信息

-R：处理执行目录及其子目录下所有文件

-v：运行时显示详细处理信息

<权限范围>+<权限设置>：使权限范围内的目录或文件具有指定权限

<权限范围>-<权限设置>：删除权限范围内的目录或文件的指定权限

<权限范围>=<权限设置>：设置权限范围内的目录或文件的权限为指定的值

|  |  |
| --- | --- |
| **权 限 范 围** | **权 限 代 号** |
| u：目录或文件的当前用户 | r：读，代号为4 |
| g：目录或文件的当前群组 | w：写，代号为2 |
| o：其他用户或群组 | x：执行，代号为1 |
| a：所有用户和群组 | -：删除，代号为0 |
|  | s：特权权限 |

例如：

chmod a+x log1.log：设定文件log1.log所有用户或群组都有执行权限

chmod ug+x,o-x log1.log：同时设定不同的用户权限

chmod u=x log1.log：撤销用户原有对log1.log的权限并设定为x

chmod -R u+x test4：递归地给test4下所有文件与子目录分配执行权限

chmod 751 file：给file、属主分配读、写、执行权限，给所在组分配读、执行权限，其他用户分配执行权限

**tar**

**用来压缩和解压文件，tar本身不具备压缩功能，它是调用压缩功能实现的**

-c：建立新的压缩文件

-d：记录文件差别

-r：添加文件到已压缩的文件中

-x：从压缩的文件中提取文件

-t：显示压缩的文件内容

-z：支持gzip解压文件

-v：显示操作过程

-f：指定压缩文件

例如：

tar -cvf log.tar log1.log：仅把log1.log打包成log.tar，不压缩

tar -zcvf log.tar.gz log1.log：把log1.log打包成log.tar.gz，并指出gzip压缩

tar -ztvf log.tar.gz：查阅log.tar.gz中的内容，因log.tar.gz是以gzip压缩的，所以要加上z

tar -zcvf log30.tar.gz log1.log log2.log：把多个文件打包压缩成一个文件

tar -zxvf log30.tar.gz log2015.log：把log30.tar.gz中的log2015.log解压出来

**diff**

**比较单个文件或目录内容，如果指定比较的是文件，则只有当输入为文本文件时才有效。以逐行的方式，比较文本文件的异同处。如果指定的是目录，diff命令会比较两个目录下名字相同的文本文件，列出不同的二进制文件、公共子目录和只在一个目录下出现的文件。**

-b：不检查空格字符的不同

-B：不检查空白行

-i：不检查大小写的不同

-q：只显示有无差别，不现实详细信息

-y：以并排方式显式文件的异同处

-w：在使用参数-y时，指定栏宽

例如：

diff log1.log log2.log：比较两个文件，输出结果比如"3c3"和"8c8"表示两个文件在第三行和第8行有所不同，"11,12d10"表示第一个文件比第二个文件多了第11行、第12行。diff命令的normal格式模式显示三种提示，即a-add、c-change、d-delete

diff test3 test4：比较test3和test4两个文件夹的不同

diff log1.log log2.log -y -w 50：并排输出，并制定栏宽为50

**grep**

**用于过滤/搜索指定字符串。可以使用正则表达式，能配合多种命令使用**

-A：显示符合的行及之后的内容

-B：显示符合的行及之前的内容

-C：显示符合的行及前后的内容

-c：计算符合的行数目

-i：忽略大小写

规则表达式

^：'^grep'表示匹配所有以grep开头的行

$：'grep$'表示匹配所有以grep结尾的行

.：'gr.p'表示匹配'gr'后面接任意字符然后是'p'

\*：'\*grep'表示匹配一个或多个空格后紧跟grep的行

[]：[Gg]rep表示匹配Grep与grep

[^]：[^A-FH-Z]rep表示匹配不以A-F、H-Z开头并紧跟rep的行

例如（grep命令大多数时候多要和别的命令一起使用才有意义）：

ps -ef | grep -c SVN：查找指定的进程的个数

cat test.txt | grep -f test2.txt：从text2.txt中读取关键字后在test.txt中搜索

cat test.txt | grep nf text2.txt：从text2.txt中读取关键字后在text.txt中搜索并显示行号

grep 'linux' text.txt test2.txt：从多个文件中搜索关键字'linux'

cat test.txt | grep -E 'ed|at'：显示包含ed或at的行

**ps**

**用来列出系统中运行的那些进程**

Linux中的进程状态有：

|  |  |
| --- | --- |
| **状 态** | **含 义** |
| R | 运行：正在运行或在运行队列中等待 |
| S | 中断：休眠中，受阻 |
| D | 不可中断：收到信号不唤醒和不可运行，进程必须等待直至有中断发生 |
| Z | 僵死：进程已终止 |
| T | 停止：进程收到SIGSTOP、SIGSTP、SIGSIN、SIGSOU信号后停止运行 |

-A：显示所有进程

-e：等同于-A

-c：显示进程的真实名称

-f：显示程序间的关系

例如：

ps -u root：显示root的进程信息

ps -ef：显示所有进程信息连同其命令行

ps -ef | grep ssh：ps常与grep连赢来显示特定进程

ps -l：将目前属于自己登录的pid与相关信息列出来

ps aux：列出目前所有的在内存中的进程

**netstat**

**用于显示各种网络相关信息，如网络连接、路由表、接口状态、多播成员等**

netstat输出：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Proto | Recv-Q | Send-Q | Local Address | Foreign Address | State |
| 指协议，有tcp、udp、unix等 | 接收队列 | 发送队列 | 本机地址 | 远程地址 | 状态 |

-a：显示所有选项

-t：仅显示tcp选项

-u：仅显示udp选项

-n：不显示主机别名，能显示数字的全部显示数字

-r：显示路由信息、路由表

-s：按各种协议进行统计

-l：仅列出LISTEN的服务状态

例如：

netstat -at：列出所有tcp端口

netstat -lt：只列出所有监听tcp的端口

netstat -s：显示所有端口的统计信息

netstat -p：在输出中显示pid和进程名称

**uname**

**获取操作系统相关信息**

-a：列出详细信息，依次为内核名、主机名、内核版本号、内核版本、硬件名、处理器类型、硬件平台类型、操作系统名称

-m：显示主机CPU名

-n：显示主机在网络上的节点或网络名

-r：显示Linux操作系统内核版本号

-s：显示Linux内核名称

-v：显示操作系统是第几个version版本

-p：显示处理器类型

-i：显示硬件平台类型

-o：显示操作系统名

**2.3 useradd、userdel、passwd、groupadd、chgrp、chown、df、du、sort、wget**

**useradd**

**添加新的用户账号，只有root账户可以操作**

-d 目录：指定用户主目录（默认在home下），若此目录不存在可同时使用-m创建主目录

-g 用户组：指定用户所属的用户组

-G 用户组：指定用户所属的附加组

-s shell文件：指定用户登陆使用的shell

例如：

useradd -d /usr/sam -m sam：为登录名sam创建一个主目录/usr/sam

useradd -s /bin/sh -g group -G adm,root sam：创建一个登录用户sam，该用户登录的shell为/bin/sh，属于group用户组，同时属于用户组adm与root，但group是其主组

**userdel**

**删除用户账号**

-r：最常用的一个选项，用户的主目录被一起删除

例如：

userdel -r sam：删除sam在文件系统中（/etc/password、/etc/shadow、/etc/group），同时删除用户的主目录

**passwd**

**管理用户的口令，用户账号刚创建时无口令，被系统锁定，无法使用，必须为其指定口令后才能使用，即使是空口令。超级用户可为自己与其他用户指定口令，普通用户只能指定自己的口令**

-l：锁定口令

-u：口令解锁

-d：使账号无口令

-f：强迫用户下次登录时修改口令

例如：

passwd：修改当前用户口令，如果是超级用户的话则不需要知道原口令，普通用户修改口令会先询问原口令

passwd sam：超级用户指定用户sam的口令

passwd -d sam：超级用户删除用户sam的口令，使sam下次登录不需要口令

passwd -l sam：超级用户锁定sam，使其无法登陆

**groupadd**

**增加一个新的用户组**

-g：指定用户组标识号GID

-o：一般与-g一起使用，表示新用户组的GID可以与已有用户组GID相同

例如：

groupadd group1：增加一个用户组group1，GID是在当前已有GID最大值上加1

groupadd -g 101 group1：增加一个用户组group1，并指定其GID为101

**chgrp**

**采用群组名或群组识别码GID的方式改变文件或目录的所属群组，超级用户使用。被改变的组名必须在/etc/group文件内存在**

-c：当发生改变时输出调试信息

-f：不显示错误信息

-R：处理指定目录及其下所有子目录、子文件

-v：运行时显示详细处理信息

例如：

chgrp -v bin log1.log：将log1.log的群组改为bin，并显示详细处理信息

chgrp --reference=log2.log log1.log：改变log1.log群组属性，使之参考log2.log

chgrp -R bin test：改变test下所有目录及其子目录的群组属性为bin

chgrp -R 100 test：根据群组识别码GID改变test群组属性

**chown**

**改变文件的拥有者与群组，拥有者可用用户名或用户ID、群组可用组名或组ID。chown命令的操作权限一般为系统管理员，系统管理员常将文件拷贝到另一用户目录名下，使之拥有使用该文件的权限**

-c：显示更改的部分信息

-f：忽略错误信息

-R：处理指定目录及其下所有子目录、子文件

-v：显示详细处理信息

例如：

chown mail:mail log1.log：改变log1.log的拥有者与所属群组为mail

chown :mail log1.log：改变log1.log的文件群组为mail

chown -R -v root:mail test6：改变test6文件夹及其下所有文件/文件夹的所有者为root，群组为mail，修改时显示处理信息

**df**

**显示指定磁盘文件可用空间，如果不指定文件名，则所有当前被挂在的文件系统的可用空间将被显示。默认展示单位为kb**

-a：展示全部的文件系统列表

-h：以方便阅读的形式展示

-l：只显示本地文件系统

例如：

df：显示磁盘使用情况

df -T：列出文件系统类型

df -h：以方便阅读的方式显示目前磁盘空间和使用情况

df -t ext3：显示磁盘ext3下的文件使用情况

**du**

**显示每个文件和目录的磁盘使用情况，默认展示单位为kb**

-b：以byte为展示单位

-k：以kb为展示单位

-m：以mb为展示单位

-s：只显示总计

-h：以K、M、G为单位，提高信息的可读性

例如：

du：显示目录或文件所占空间，只显示当前目录下的子目录的目录大小和当前目录总大小（当前目录总大小在最下面）

du abc：显示abc目录所占用的空间

du log1.log log2.log log3.log：显示多文件所占空间

du -s：只显示总和大小

du | sort -nr | more：按空间大小排序并使用more输出

du --max-depth=1：输出当前目录下的子目录所占空间大小

**sort**

**依据不同数据类型进行排序，默认为升序**

-b：忽略每行开始的空格

-c：检查文件是否已按顺序排列

-f：排列时忽略大小写字母

-n：按数值大小排序

-o<输出文件>：将排序后的结果存入指定文件中

-r：以相反顺序排序

例如：

sort seq.txt：按ASCII码值进行比较并排序，最后将它们按升序输出

sort -u seq.txt：按ASCII码值进行比较、排序、去重复，最后将它们按升序输出

sort -r seq.txt：前面是升序，加个-r就变成降序了

sort -r number.txt -o number.txt：把number.txt中的内容反向排序后覆盖number.txt，若是别的文件名则直接输出至指定文件中

sort -n number.txt：以数值排序number.txt中的数值，这样就不会出现10<2的情况了

sort -n -k 2 -t ':' fruit.txt：将fruit.txt中的每行内容以':'分隔，并以第2列为基准按数值排序

**wget**

**用于从网络上下载资源，若不指定目录，则下载资源默认为当前目录。wget支持：**

**1、断点下载**

**2、同时支持FTP与HTTP**

**3、支持代理服务器**

**4、设置简单**

-b：启动后转入后台执行

-o：把记录写到文件中

-a：把记录追加到文件中

-tries=NUMBER：设定最大尝试连接次数，0表示无限制，默认为20次

-timeout=SECOND：设定响应超时的秒数

例如：

wget URL：使用wget从URL下载单个文件，会有进度条

wget -O abc.zip URL：下载一个文件并以abc.zip命名

wget --limit-rate=300k URL：限制下载速率为300K下载某个文件

wget -c URL：断点续传某文件，防止下载大文件时的网络中断

wget -b URL：后台下载某文件，可用tail -f wget-log查看下载进度

wget --user-agent="XXX" URL：使用伪代理名称下载

wget --tries=40 URL：下载一个文件，失败可重试40次

wget --reject=gif URL：下载一个文件，但不下载gif文件，用于过滤指定格式

wget -o download.log URL：将下载日志写入download.log

wget --ftp-user=USERNAME --ftp-password=PASSWORD URL：指定用户名、密码，使用ftp下载某文件