1. 文件传输：

```
scp、rsync
```

1.1 scp命令：

加密的方式在本地主机和远程主机之间复制文件。

1.1.1 本地目录复制到远程：

```
scp -r  lutixia/ 192.168.0.132:/data/

ps:lutixia后面不管带不带/，都可以同步lutixia目录

如果只想同步目录下文件：
scp -r  lutixia/*  192.168.0.132:/data/
```

1.1.2 本地文件复制到远程：

```
scp fstab 192.168.0.132:/data/
```

1.1.3 启动压缩：

```
scp -rC  lutixia/ 192.168.0.132:/data/
```

1.2 rsync命令：

远程数据同步工具，它传送两个文件的不同部分，而不是每次都整份传送，因此速度相当快。

需要注意：**本地与远程服务器都需要安装好rsync软件包。**

1.2.1 目录保持绝对一致：

--delete 参数会删除源目录中没有，而远程目录中存在的文件，以此保持文件一致。

```
# 同步本地目录。
rsync -av --delete  A/ /data/B
rsync -av --delete  A /data/B

ps1: B目录可以不存在，如果不存在则自动创建，但是上级data必须存在。

ps2:A目录带不带斜杠/，意义是不一样的，带斜杠表示同步A目录下的文件，
不带则同步A目录以及其中文件。

--delete: B与A保持绝对的一致，B中存在，而A中不存在的文件将被删除。
```

1.2.2 本地同步到远程：

```
rsync -av  A/ 192.168.0.132:/data/

将本地A目录所有文件，同步至远程服务器132上，
同理，要想实现绝对的一致，需要加上--delete参数。

使用ssh协议连接到远程，所以可以将本机公钥发给远程主机，
否则每次执行都会提示出入密码。
```

1.2.3 远程同步到本地：

```
rsync -av  192.168.0.132:/data/C .
rsync -av  192.168.0.132:/data/C/ .

.    表示将远程目录C，同步到本地C目录。

ps：要注意不带斜杠会在本地目录创建一个C目录，
    如果带上斜杠/,则只会同步C目录下的文件。
```

1.3 开启shell扩展：

```
shopt  -s extglob
rm -rf !(anaconda-ks.cfg)
```

2. 权限管理：

```
chmod、chown、setfacl、chattr、lsattr
```

2. 1 chmod命令：

变更文件或目录的权限

```
u   用户user，文件或目录的所有者。
g   用户组group，文件或目录所属群组
o   其它用户others
a   所有用户all，系统默认使用此项

+   添加某些权限
-   取消某些权限
=   设置文件的权限为给定的权限
r   表示可读权限
w   表示可写权限
x   表示可执行权限



s   设置权限suid和sgid，使用权限组合“u+s”设定文件的用户的ID
位，“g+s”设置组ID位
t   只有目录或文件的所有者才可以删除目录下的文件

-R 递归处理，将指令目录下的所有文件及子目录一并处理
```

2.1.1 设置可读可写：

```
# o+w | o=rw
+    表示直接加上写权限。
=    表示将其他用户权限设置为可读可写，如果以前还有可执行权限，现在也
去掉。
[root@localhost ~]# ll
-rw-r--r--  1 root root    0 Aug 22 05:09 file1
# 对普通文件授权：
[root@localhost ~]# chmod o+w file1
[root@localhost ~]# ll
-rw-r--rw-  1 root root    0 Aug 22 05:09 file1
或者：
[root@localhost ~]# chmod o=rw file1
# 对目录进行授权，R表示递归：
[root@localhost ~]# chmod -R o+w lutixia/
```

2.1.2 危险操作：

```
# 将目录权限设置为其他用户可写，这时普通用户是可以进入到目录中任意删除，修改文件名(即使不是自己的文件)，对文件的其他用户权限为不可写的文件也可以强制保存。

chmod o+w /data/

lutixia1用户修改lutixia2用户的文件：
[lutixia1@localhost data]$ ll
-rw-r--r-- 1 lutixia2 lutixia2        30 Aug 22 05:49 fs
[lutixia1@localhost data]$ vim fs
...
#提示不能修改
E45: 'readonly' option is set (add ! to override)
这时可以强制保存（:wq!）。
[lutixia1@localhost data]$ cat fs
this is test
this is lutixia2
this is lutixia1

【o-w】将目录权限设置为其他用户不可写（目录默认权限755），这时普通用户就无法删除，修改文件名，但是如果文件本身有其他可写权限，还是可以写数据的。
```

总结一下：父目录其他用户有可写权限，其下子文件不管有没有可写权限，均可强行写入，修改！
父目录其他用户没有可写权限，其下子文件只有可写才有写入权限，但不具备删除，修改文件名权限。

2.1.3 设置t权限：

-t 是对目录设置特殊权限，用户只能删除自己的文件。

```
[root@localhost ~]# chmod o+t /data/
#lutixia2想删除lutixia1的文件，失败
[lutixia2@localhost data]$ ll
-rw-r--r-- 1 lutixia1 lutixia1        63 Aug 22 05:58 fs
[lutixia2@localhost data]$ rm -rf fs
rm: cannot remove ‘fs’: Operation not permitted
```

2.1.4 设置s权限：

【-s】权限可以设置suid和sgid：

- “u+s”： 设置使文件在执行阶段具有文件所有者的权限；
- “g+s” 任何用户在此目录下创建的文件都具有和该目录所属的组相同的组。

【u+s】

```
#未设置前，普通用户执行netstat -ntlp，会提示没有root权限，普通用户看不了pid的属主：
[lutixia1@localhost data]$ netstat -nltp
(No info could be read for "-p": geteuid()=1003 but you should be root.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN      -

#设置suid后：
chmod u+s /usr/bin/netstat

[lutixia1@localhost data]$ netstat -nltp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN      1/systemd
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      1050/sshd
```

【g+s】

任何用户在此目录下创建的文件都具有和该目录所属的组相同的组

```
[root@localhost ~]# chmod g+s /data
[lutixia2@localhost data]$ ll
drwxrwsr-x 2 lutixia1 root         6 Aug 22 07:14 lutixia1
drwxrwsr-x 2 lutixia2 root         6 Aug 22 07:15 lutixia2
```

2.2 setfacl命令：

设置文件访问控制列表

2.2.1 修改acl规则：

通过-m参数，可以修改文件的acl规则。

```
# 设置用户权限
setfacl -m  u:lutixia:rw   abc.txt

# getfacl用于查看文件acl权限：
[root@localhost ~]# getfacl /data/abc.txt
# file: data/abc.txt
# owner: root
# group: root

#第一个user没有写用户名，代表是默认属主root的权限
user::rw-
#第二个user代表得是lutixia用户得权限
user:lutixia:rw-
group::r--
mask::rw-
other::r--

# 设置组权限
setfacl -m  g:web:rw     abc.txt
# 取消用户所有权限
setfacl -m  u:lutixia:-  abc.txt
# 取消其他用户得所有权限：
setfacl -m o::- abc.txt
```

```
#用户必须存在，否则会报错：
[root@localhost ~]# setfacl -m u:lutixia:rw /data/abc.txt
setfacl: Option -m: Invalid argument near character 3
[root@localhost ~]# id lutixia
id: lutixia: no such user
[root@localhost ~]# useradd lutixia
[root@localhost ~]# setfacl -m u:lutixia:rw /data/abc.txt
#现在lutixia可以写入数据到abc.txt文件了
[lutixia@localhost ~]$ echo "this is lutixia" >
/data/abc.txt
```

2.2.2 批量修改acl规则：

【-M】从**文件中读取相应得权限**进行设置，多用于批量管理用户权限。

```
#先创建一个文件：
touch acl.txt
#添加要设置得权限：
u:lutixia:rwx
u:lutixia1:rw
u:lutixia2:x
u:lutixia3:rx
#执行,其中abc.txt是要进行权限设置得文件：
setfacl -M acl.txt abc.txt
```

2.2.3 撤销acl权限

【-x】撤销某个用户得acl权限，恢复到普通ugo权限：

```
#不能单独撤销某一个权限，比如只撤销可写权限
setfacl -x u:lutixia abc.txt
```

【-b】撤销所有用户或者组得acl权限：

```
setfacl -b /data/abc.txt
```

2.2.4 复制acl规则：

【--set-file】复制一个文件acl权限到另外一个文件。

```
getfacl file1 | setfacl --set-file=- file2

-表示输出流

echo jfedu |cat -
```

2.2.5 临时降低权限：

【mask】会临时降低acl用户或者组的权限，只能降低用户权限，不能提升。

```
#设置acl用户的权限为rw:
[root@localhost data]# setfacl -m u:lutixia:rw abc.txt
#这时lutixia用户是可以写入的:
[lutixia@localhost data]$ echo "this is test" >> abc.txt

#设置mask:
[root@localhost data]# setfacl -m mask::r abc.txt
#这时acl用户只有读权限，不再可写:
[lutixia@localhost data]$ echo "this is test" >> abc.txt
-bash: abc.txt: 权限不够

注意：如果lutixia用户本身r权限（只读权限），即使mask设置为rw，也是不能写的。
而且设置了mask之后，如果再次使用setfacl进行权限的设置，那么mask的作用就失效了。
```

2.3 chattr命令：

改变文件属性

2.3.1 只允许追加：

【+a】让某个文件只能往里面追加内容，不能删除。

```
chattr +a /var/log/nginx/access.log

去掉属性就是-a
```

2.3.2 完全限制：

【+i】完全限制系统中某个关键文件，不能被修改，删除。

```
chattr +i /etc/fstab
```

3. 用户管理：

```
useradd、passwd、userdel、usermod、groupadd、groupdel
```

3.1 useradd命令：

创建的新的系统用户

默认是在/home下创建家目录：

```
[root@localhost ~]# useradd  lutixia1
[root@localhost ~]# su - lutixia1
[lutixia1@localhost ~]$ pwd
/home/lutixia1
```

3.1.1 指定家目录的根：

【-b】指定用户家目录的根目录，会自动创建用户家目录。

```
[root@localhost ~]# useradd -b /data/  lutixia2
[root@localhost ~]# su - lutixia2
[lutixia2@localhost ~]$ pwd
/data//lutixia2
```

3.1.2 指定家目录：

【-d】指定用户的家目录，即指定的目录即为家目录，也会自动创建。

```
[root@localhost ~]# useradd -d /data/lutixia4 lutixia4
[root@localhost ~]# su - lutixia4
[lutixia4@localhost ~]$ pwd
/data/lutixia4

#必须要指定到lutixia4
```

3.1.3 指定系统用户：

【-r】指定创建系统用户。

```
[root@localhost ~]# useradd -r lutixia6
[root@localhost ~]# id lutixia6
uid=994(lutixia6) gid=990(lutixia6) 组=990(lutixia6)
```

3.1.4 指定shell:

【-s】指定用户登入后所使用的shell。

```
[root@localhost ~]# useradd -s /sbin/nologin  lutixia7
[root@localhost ~]# grep "lutixia7" /etc/passwd
lutixia7:x:1014:1014::/home/lutixia7:/sbin/nologin

用户不能登录系统，多用于创建系统服务用户
```

3.2 passwd命令：

修改用户密码

```
#管理员修改用户密码：
passwd username
#用户修改自己的密码：
passwd
```

3.2.1 锁定密码：

【-l】锁定用户密码，用户登录不了。

```
[root@localhost ~]# passwd -l lutixia4
锁定用户 lutixia4 的密码 。
passwd: 操作成功
#用普通用户身份进行切换，因为root切换普通用户，不需要输入密码，所以会感觉没生效
[lutixia@localhost ~]$ su - lutixia4
密码：
su: 鉴定故障
#或者使用ssh登录普通用户，也可以验证
ssh  lutixia4@192.168.0.124
```

3.2.2 解锁密码：

【-u】解锁用户密码。

```
[root@localhost ~]# passwd -u lutixia4
解锁用户 lutixia4 的密码。
passwd: 操作成功
[lutixia@localhost ~]$ su - lutixia4
密码:
上一次登录：三 8月 21 16:42:12 CST 2019从 192.168.0.124pts/4 上
最后一次失败的登录：三 8月 21 16:43:07 CST 2019pts/3 上
最有一次成功登录后有 3 次失败的登录尝试。
```

3.2.3 注销密码：

【-e】使用户密码立即失效。

```
[root@localhost ~]# passwd -e lutixia4
正在终止用户 lutixia4 的密码。
passwd: 操作成功
[lutixia@localhost ~]$ su - lutixia4
密码:
^C

已经登录不上了。
```

3.2.4 非交互修改密码：

【--stdin】通过标准输入设置用户密码。

```
[root@localhost ~]# echo "123456" |passwd --stdin lutixia4
更改用户 lutixia4 的密码 。
passwd：所有的身份验证令牌已经成功更新。
```

4. 用户管理：

4.1 userdel命令：

删除用户以及相关数据，但是**如果不加参数，则只删除用户**，不删除其家目录

```
[root@localhost ~]# useradd lutixia
[root@localhost ~]# userdel lutixia
[root@localhost ~]# ls /home
jfedu  lutixia  www
[root@localhost ~]# id lutixia
id: lutixia: no such user
```

4.1.1 删除用户家目录：

【-r】删除用户的同时，删除与用户相关的所有文件。

```
[root@localhost ~]# userdel -r lutixia
[root@localhost ~]# ls /home
jfedu  www
```

4.1.2 强制删除在线用户：

【-f】强制删除用户，即使用户已经登录，默认情况下，用户在线，是不能删除的，因为用户可能正在执行相关操作。

```
#用户登录，不带参数删除：
[root@localhost ~]# !useradd
useradd lutixia
[root@localhost ~]# userdel lutixia
userdel: user lutixia is currently used by process 1958

#用户登录，强制删除：
[root@localhost ~]# userdel -rf lutixia
userdel: user lutixia is currently used by process 2166
#用户已经被删除了
[root@localhost ~]# id lutixia
id: lutixia: no such user
```

4.2 usermod命令：

修改用户的基本信息，不允许改变正在线上的使用者帐号名称。

4.2.1 修改用户名：

【-l】修改用户名：

```
[root@localhost ~]# usermod -l ltx lutixia
[root@localhost ~]# id lutixia
id: lutixia: no such user
[root@localhost ~]# id ltx
uid=1002(ltx) gid=1002(lutixia) groups=1002(lutixia)
```

4.2.2 添加附加组：

【-a】用户添加附加组。

```
[lutixia@localhost ~]$ sudo ps -ef |grep nginx
lutixia 不在 sudoers 文件中。此事将被报告。
[root@localhost ~]# usermod -a -G wheel lutixia
[lutixia@localhost ~]$ sudo ps -ef |grep nginx
lutixia  2186  2140  0 07:59 pts/1    00:00:00 grep --
color=auto nginx
```

4.2.3 修改用户家目录：

【-d】修改用户登入时的目录，只是**修改/etc/passwd中用户的家目录配置信息**，不会自动创建新的家目录，通常和-m一起使用。

```
如果已经使用-d执行了，可以立马去创建其家目录，并将/etc/skel/.bash*
文件拷贝进去。
```

4.2.4 移动用户家目录：

【-m】移动用户家目录到新的位置，不能单独使用，一般与-d一起使用。

```
[root@localhost ~]# useradd lutixia
[root@localhost ~]# usermod -md /data/new lutixia
[root@localhost ~]# su - lutixia
[lutixia@localhost ~]$
[lutixia@localhost ~]$
[lutixia@localhost ~]$ pwd
/data/new
```

4.3 groupadd命令：

创建一个新的工作组

```
#创建普通组
groupadd web
#创建系统组
groupadd -r web2

#创建指定id的组
[root@localhost ~]# groupadd -g 500 web3
[root@localhost ~]# grep "web3" /etc/group
web3:x:500:

查看组信息:
tail /etc/group
```

4.4 groupdel命令:

删除指定的工作组

```
[root@localhost ~]# groupdel web3
[root@localhost ~]# grep "web3" /etc/group
```