### 一、上堂回顾

#### 1.默写题目

#书写命令，完成下面操作  
#1.将文件text01.txt重命名为text02.txt mv text01.txt text02.txt  
#2.安装pygame第三方模块 pip install pygame  
#3.移除名为xxx的软件 apt remove xxx -purge/dpkg -r xxx  
#4.查看文件f1.txt的后5行内容 tail -5 f1.txt  
#5.创建具有嵌套关系的目录，a/b/c mkdir -p a/b/c  
#6.删除文件a1.txt中的第三行内容 sed -i '3d' a1.txt

#### 2.知识点回顾

--help/man

cd

ls

pwd

mv:移动，重命名

cp：拷贝

cat / head /tail/more:查看文件内容

touch/mkdir

rm -rf xxxx:强制删除

ln:软硬链接的区别

grep【结合管道符使用，搜索的范围不确定】/find【查找指定目录下的文件：-size -name -empty】

sort:搜索文件的过程中【-t:分隔符 -k:指明第几列】

tee:输入

cut:提取文本列【-d:分隔符 -f:指明第几列】

sed:增【a\ i\】删【d】改【c\】查【p】

tar:打包【-cvf -zcvf -jcvf 】

gzip:压缩【-d:解压缩 -r:递归处理】

### 二、常用命令

#### 1.vim编辑器

**vi命令**是UNIX操作系统最通用的全屏幕纯文本编辑器。Linux中的vi编辑器叫vim，它是vi的增强版（vi Improved），与vi编辑器完全兼容，而且实现了很多增强功能

三种工作模式之间的切换：命令模式-----》编辑模式【输入模式】----》命令模式----》末行模式【底线命令模式】

进入vi的命令   
 #vi filename :打开或新建文件，并将光标置于第一行首   
 vi +n filename ：打开文件，并将光标置于第n行首   
 vi + filename ：打开文件，并将光标置于最后一行首  
 vi filename1 filename2 :同时打开多个文件  
  
插入文本类命令   
 #i ：在光标前   
 I ：在当前行首   
 #a：光标后   
 A：在当前行尾   
 o：在当前行之下新开一行   
 O：在当前行之上新开一行   
 r：替换当前字符   
 R：替换当前字符及其后的字符，直至按ESC键   
  
移动光标 #注意：必须在命令模式下操作  
 j或下箭头 向下移动一行  
 k或上箭头 向上移动一行  
 h或左箭头 左移一个字符  
 l或右箭头 右移一个字符  
 #w 　　　　右移一个词  
 W 　　　　右移一个以空格分隔的词   
 #b 　　　　左移一个词  
 B 　　　　左移一个以空格分隔的词  
 0 　　　　移到行首  
 Ctrl-F　　向前翻页  
 Ctrl-B　　向后翻页  
 nG　　　　到第n行 ------》先按下数字，再按下G  
 G 　　　　到最后一行  
 gg 第一行 -----》先按下g，再按下g  
 n+ 光标下移n行   
 n- 光标上移n行  
   
 #shift + g/G :到最后一行  
   
esc:退出输入模式  
  
:set number：在命令模式下，用于在最左端显示行号；  
:set nonumber：在命令模式下，用于在最左端不显示行号；  
  
保存退出   
 :wq 执行存盘退出操作； -----》对内容做修改之后使用  
 :w 执行存盘操作；  
 :w！ 执行强制存盘操作；  
 :q 执行退出vi操作；  
 :q！ 执行强制退出vi操作； -----》如果没有任何修改的时使用  
 :e文件名 打开并编辑指定名称的文件；  
 :n! 如果同时打开多个文件，则保存上个文件继续编辑下一个文件；  
 :f 用于显示当前的文件名、光标所在行的行号以及显示比例；  
"""  
如果修改了文件内容，则:wq保存退出，如果不让退出，则:wq!退出  
如果未修改文件内容，则:q退出，如果不让退出，则:q!退出  
"""  
   
删除操作#注意：在命令模式下操作  
 x 删除光标处的单个字符   
 dd 删除光标所在行   
 dw 删除当前字符到单词尾（包括空格）的所有字符   
 de 删除当前字符到单词尾（不包括单词尾部的空格）的所有字符   
 d$ 删除当前字符到行尾的所有字符   
 d^ 删除当前字符到行首的所有字符   
 J 删除光标所在行行尾的换行符，相当于合并当前行和下一行的内容  
   
替换操作  
 :s/old/new 将当前行中查找到的第一个字符“old” 串替换为“new”  
 :#,#s/old/new 在行号“#,#”范围内替换所有的字符串“old”为“new”  
 :%s/old/new 在整个文件范围内替换所有的字符串“old”为“new”  
 :s/old/new/c 在替换命令末尾加入c命令，将对每个替换动作提示用户进行确认  
   
撤消操作 【注意：直接在输入模式下使用】  
 u取消最近一次的操作，并恢复操作结果  
 可以多次使用u命令恢复已进行的多步操作   
 U取消对当前行进行的所有操作   
 Ctrl + r对使用u命令撤销的操作进行恢复

#### 2.用户管理【重点掌握】

用户管理包括两部分内容：用户管理和用户组管理

包含的操作：

a.用户账号的添加，删除和修改

b.用户口令【密码】的管理

c.用户组的添加和删除

linux使用文件保存用户信息   
#/etc/passwd 用户账户信息。  
/etc/shadow 安全用户账户信息。  
/etc/group 组账户信息。  
/etc/gshadow 安全组账户信息。  
/etc/default/useradd 账户创建的默认值。  
/etc/skel/ 包含默认文件的目录。  
/etc/login.defs Shadow 密码套件配置。

1>whoami 查看当前系统当前用户的用户名

2>who 查看当前所有登录系统的用户信息

-q:只显示用户的登录账号的和登录用户的数量  
-u:显示列标题

3>chown 更改文件的所有者和所有组

4>sudo 用来以其他身份来执行命令，预设的身份为root【让当前用户暂时以管理员的身份root来执行命令】

5>exit 退出

演示命令：  
lichen@lichen-host:~$ whoami  
lichen  
lichen@lichen-host:~$ su root  
密码：   
root@lichen-host:/home/lichen# whoami  
root  
root@lichen-host:/home/lichen# who  
lichen tty7 2019-08-21 09:10 (:0)  
root@lichen-host:/home/lichen# su lichen  
lichen@lichen-host:~$ su root  
密码：   
root@lichen-host:/home/lichen# exit  
exit

6>su 切换用户

su root:默认的路径是普通用户的主目录【/home/yangyang】  
su - /su - root :默认的路径是root用户的主目录【/root】  
  
su可以在任意两个用户之间进行来回切换  
  
演示命令：  
lichen@lichen-host:~$ su root  
密码：   
root@lichen-host:/home/lichen# pwd  
/home/lichen  
root@lichen-host:/home/lichen# exit  
exit  
lichen@lichen-host:~$ su -  
密码：   
root@lichen-host:~# pwd  
/root  
root@lichen-host:~# exit  
注销  
lichen@lichen-host:~$ pwd  
/home/lichen  
lichen@lichen-host:~$ su - root  
密码：   
root@lichen-host:~# pwd  
/root  
root@lichen-host:~# su - lichen  
lichen@lichen-host:~$ pwd  
/home/lichen  
lichen@lichen-host:~$ ls /home/  
lichen  
lichen@lichen-host:~$

7>useradd 添加用户

注意：必须以root权限进行用户的添加和删除

-c 备注 加上备注。并会将此备注文字加在/etc/passwd中的第5项字段中  
-d 用户主文件夹。指定用户登录所进入的目录，并赋予用户对该目录的的完全控制权   
-e 有效期限。指定帐号的有效期限。格式为YYYY-MM-DD，将存储在/etc/shadow   
-f 缓冲天数。限定密码过期后多少天，将该用户帐号停用   
#-g 主要组。设置用户所属的主要组 www.cit.cn #chown   
-G 次要组。设置用户所属的次要组，可设置多组   
-M 强制不创建用户主文件夹   
#-m 强制建立用户主文件夹，并将/etc/skel/当中的文件复制到用户的根目录下 #mkdir cp   
-p 密码。输入该帐号的密码   
#-s shell。用户登录所使用的shell   
-u uid。指定帐号的标志符user id，简称uid  
  
演示命令：  
#方式一  
lichen@lichen-host:~$ sudo useradd zhangsan  
lichen@lichen-host:~$ sudo vim /etc/passwd  
lichen@lichen-host:~$ ls /home/  
lichen  
lichen@lichen-host:~$ sudo mkdir /home/zhangsan  
lichen@lichen-host:~$ ls /home/  
lichen zhangsan  
lichen@lichen-host:~$ ll /home/  
总用量 16  
drwxr-xr-x 4 root root 4096 8月 21 10:37 ./  
drwxr-xr-x 24 root root 4096 3月 25 2018 ../  
drwxr-xr-x 30 lichen rock 4096 8月 21 10:36 lichen/  
drwxr-xr-x 2 root root 4096 8月 21 10:37 zhangsan/  
lichen@lichen-host:~$ sudo chown zhangsan:zhangsan /home/zhangsan  
lichen@lichen-host:~$ ll /home/  
总用量 16  
drwxr-xr-x 4 root root 4096 8月 21 10:37 ./  
drwxr-xr-x 24 root root 4096 3月 25 2018 ../  
drwxr-xr-x 30 lichen rock 4096 8月 21 10:36 lichen/  
drwxr-xr-x 2 zhangsan zhangsan 4096 8月 21 10:37 zhangsan/  
lichen@lichen-host:~$ sudo passwd zhangsan  
输入新的 UNIX 密码：   
重新输入新的 UNIX 密码：   
passwd：已成功更新密码  
lichen@lichen-host:~$ su - zhangsan  
密码：   
$ pwd  
/home/zhangsan  
$ ls  
  
  
"""  
问题：  
yangyang@yangyang-host:~$su - zhangsan  
$  
  
原因：新创建的用户缺少配置文件  
  
解决办法  
$ ls -a /home/lichen/.bash\*  
/home/lichen/.bash\_history /home/lichen/.bash\_logout /home/lichen/.bashrc  
$ ls -a /etc/skel/  
. .. .bash\_logout .bashrc examples.desktop .profile  
$ pwd  
/home/zhangsan  
$ cp /etc/skel/.bash\* .  
$ ls -a  
. .. .bash\_logout .bashrc  
$ 注销 #ctrl + d注销当前用户  
lichen@lichen-host:~$ sudo vim /etc/passwd  
"""  
  
#方式二  
lichen@lichen-host:~$ sudo useradd -m -s /bin/bash lisi  
lichen@lichen-host:~$ ls /home/  
lichen lisi zhangsan  
lichen@lichen-host:~$ sudo vim /etc/passwd  
lichen@lichen-host:~$ sudo passwd lisi  
输入新的 UNIX 密码：   
重新输入新的 UNIX 密码：   
passwd：已成功更新密码  
lichen@lichen-host:~$ su - lisi  
密码：   
lisi@lichen-host:~$ pwd  
/home/lisi  
lisi@lichen-host:~$ ls -a  
. .. .bash\_logout .bashrc examples.desktop .profile  
  
#方式三  
lichen@lichen-host:~$ sudo useradd -m lisi -g python1904

8>userdel 删除用户

#注意：如果一个用户需要被删除，该用户一定是注销状态  
演示命令：  
#方式一  
ichen@lichen-host:~$ sudo userdel user1 #删除用户  
lichen@lichen-host:~$ ls /home/  
lichen lisi user1 zhangsan  
lichen@lichen-host:~$ cd /home/  
lichen@lichen-host:/home$ sudo rm -rf user1 #删除用户组  
lichen@lichen-host:/home$ ls  
lichen lisi zhangsan  
  
#方式二  
lichen@lichen-host:/home$ sudo userdel -r lisi #删除用户和用户组  
userdel: user lisi is currently used by process 10559 #用户已经被进程占用  
lichen@lichen-host:/home$ 注销 #ctlr + d注销需要被删除的用户  
lisi@lichen-host:~$ 注销  
lichen@lichen-host:~$ exit  
zhangsan@lichen-host:~$ 注销  
lichen@lichen-host:~$ 注销  
root@lichen-host:~# 注销  
lichen@lichen-host:~$ sudo userdel -r lisi  
userdel: lisi 邮件池 (/var/mail/lisi) 未找到  
lichen@lichen-host:~$ ls /home/  
lichen zhangsan

9>passwd 设置密码

演示命令：  
lichen@lichen-host:~$ sudo passwd lisi  
输入新的 UNIX 密码：   
重新输入新的 UNIX 密码：   
passwd：已成功更新密码

10>查看用户组

/etc/group

11>groupadd 添加组

用户组的作用：可以将多个用户管理在同一个组下，可以针对不同的用户设置不同的访问权限

#格式：sudo groupadd xxx  
  
演示命令：  
lichen@lichen-host:~$ cat /etc/group  
root:x:0:  
daemon:x:1:  
bin:x:2:  
sys:x:3:  
adm:x:4:syslog,rock  
tty:x:5:  
disk:x:6:  
lp:x:7:  
mail:x:8:  
news:x:9:  
uucp:x:10:  
man:x:12:  
proxy:x:13:  
kmem:x:15:  
dialout:x:20:  
fax:x:21:  
voice:x:22:  
cdrom:x:24:rock  
floppy:x:25:  
tape:x:26:  
sudo:x:27:rock  
audio:x:29:pulse  
dip:x:30:rock  
www-data:x:33:  
backup:x:34:  
operator:x:37:  
list:x:38:  
irc:x:39:  
src:x:40:  
gnats:x:41:  
shadow:x:42:  
utmp:x:43:  
video:x:44:  
sasl:x:45:  
plugdev:x:46:rock  
staff:x:50:  
games:x:60:  
users:x:100:  
nogroup:x:65534:  
systemd-journal:x:101:  
systemd-timesync:x:102:  
systemd-network:x:103:  
systemd-resolve:x:104:  
systemd-bus-proxy:x:105:  
input:x:106:  
crontab:x:107:  
syslog:x:108:  
netdev:x:109:  
messagebus:x:110:  
uuidd:x:111:  
ssl-cert:x:112:  
lpadmin:x:113:rock  
lightdm:x:114:  
nopasswdlogin:x:115:  
whoopsie:x:116:  
mlocate:x:117:  
ssh:x:118:  
avahi-autoipd:x:119:  
avahi:x:120:  
bluetooth:x:121:  
scanner:x:122:saned  
colord:x:123:  
pulse:x:124:  
pulse-access:x:125:  
rtkit:x:126:  
saned:x:127:  
rock:x:1000:  
sambashare:x:128:rock  
mysql:x:129:  
guest-wxpiwz:x:999:  
zhangsan:x:1001:  
lichen@lichen-host:~$ ll /home/  
总用量 16  
drwxr-xr-x 4 root root 4096 8月 21 11:07 ./  
drwxr-xr-x 24 root root 4096 3月 25 2018 ../  
drwxr-xr-x 30 lichen rock 4096 8月 21 10:58 lichen/  
drwxr-xr-x 2 zhangsan zhangsan 4096 8月 21 10:49 zhangsan/  
lichen@lichen-host:~$ sudo groupadd lichen  
[sudo] lichen 的密码：   
lichen@lichen-host:~$ cat /etc/group  
root:x:0:  
daemon:x:1:  
bin:x:2:  
sys:x:3:  
adm:x:4:syslog,rock  
tty:x:5:  
disk:x:6:  
lp:x:7:  
mail:x:8:  
news:x:9:  
uucp:x:10:  
man:x:12:  
proxy:x:13:  
kmem:x:15:  
dialout:x:20:  
fax:x:21:  
voice:x:22:  
cdrom:x:24:rock  
floppy:x:25:  
tape:x:26:  
sudo:x:27:rock  
audio:x:29:pulse  
dip:x:30:rock  
www-data:x:33:  
backup:x:34:  
operator:x:37:  
list:x:38:  
irc:x:39:  
src:x:40:  
gnats:x:41:  
shadow:x:42:  
utmp:x:43:  
video:x:44:  
sasl:x:45:  
plugdev:x:46:rock  
staff:x:50:  
games:x:60:  
users:x:100:  
nogroup:x:65534:  
systemd-journal:x:101:  
systemd-timesync:x:102:  
systemd-network:x:103:  
systemd-resolve:x:104:  
systemd-bus-proxy:x:105:  
input:x:106:  
crontab:x:107:  
syslog:x:108:  
netdev:x:109:  
messagebus:x:110:  
uuidd:x:111:  
ssl-cert:x:112:  
lpadmin:x:113:rock  
lightdm:x:114:  
nopasswdlogin:x:115:  
whoopsie:x:116:  
mlocate:x:117:  
ssh:x:118:  
avahi-autoipd:x:119:  
avahi:x:120:  
bluetooth:x:121:  
scanner:x:122:saned  
colord:x:123:  
pulse:x:124:  
pulse-access:x:125:  
rtkit:x:126:  
saned:x:127:  
rock:x:1000:  
sambashare:x:128:rock  
mysql:x:129:  
guest-wxpiwz:x:999:  
zhangsan:x:1001:  
lichen:x:1002:  
lichen@lichen-host:~$ sudo usermod -g lichen lichen  
lichen@lichen-host:~$ cat /etc/group  
root:x:0:  
daemon:x:1:  
bin:x:2:  
sys:x:3:  
adm:x:4:syslog,rock  
tty:x:5:  
disk:x:6:  
lp:x:7:  
mail:x:8:  
news:x:9:  
uucp:x:10:  
man:x:12:  
proxy:x:13:  
kmem:x:15:  
dialout:x:20:  
fax:x:21:  
voice:x:22:  
cdrom:x:24:rock  
floppy:x:25:  
tape:x:26:  
sudo:x:27:rock  
audio:x:29:pulse  
dip:x:30:rock  
www-data:x:33:  
backup:x:34:  
operator:x:37:  
list:x:38:  
irc:x:39:  
src:x:40:  
gnats:x:41:  
shadow:x:42:  
utmp:x:43:  
video:x:44:  
sasl:x:45:  
plugdev:x:46:rock  
staff:x:50:  
games:x:60:  
users:x:100:  
nogroup:x:65534:  
systemd-journal:x:101:  
systemd-timesync:x:102:  
systemd-network:x:103:  
systemd-resolve:x:104:  
systemd-bus-proxy:x:105:  
input:x:106:  
crontab:x:107:  
syslog:x:108:  
netdev:x:109:  
messagebus:x:110:  
uuidd:x:111:  
ssl-cert:x:112:  
lpadmin:x:113:rock  
lightdm:x:114:  
nopasswdlogin:x:115:  
whoopsie:x:116:  
mlocate:x:117:  
ssh:x:118:  
avahi-autoipd:x:119:  
avahi:x:120:  
bluetooth:x:121:  
scanner:x:122:saned  
colord:x:123:  
pulse:x:124:  
pulse-access:x:125:  
rtkit:x:126:  
saned:x:127:  
rock:x:1000:  
sambashare:x:128:rock  
mysql:x:129:  
guest-wxpiwz:x:999:  
zhangsan:x:1001:  
lichen:x:1002:  
lichen@lichen-host:~$ ll /home/  
总用量 16  
drwxr-xr-x 4 root root 4096 8月 21 11:07 ./  
drwxr-xr-x 24 root root 4096 3月 25 2018 ../  
drwxr-xr-x 30 lichen lichen 4096 8月 21 10:58 lichen/  
drwxr-xr-x 2 zhangsan zhangsan 4096 8月 21 10:49 zhangsan/  
lichen@lichen-host:~$ ls /home/  
lichen zhangsan  
lichen@lichen-host:~$ sudo groupadd python1904  
lichen@lichen-host:~$ sudo useradd -m lisi -g python1904  
lichen@lichen-host:~$ ll /home/  
总用量 20  
drwxr-xr-x 5 root root 4096 8月 21 11:31 ./  
drwxr-xr-x 24 root root 4096 3月 25 2018 ../  
drwxr-xr-x 30 lichen lichen 4096 8月 21 10:58 lichen/  
drwxr-xr-x 2 lisi python1904 4096 8月 21 11:31 lisi/  
drwxr-xr-x 2 zhangsan zhangsan 4096 8月 21 10:49 zhangsan/  
lichen@lichen-host:~$ cat /etc/group  
root:x:0:  
daemon:x:1:  
bin:x:2:  
sys:x:3:  
adm:x:4:syslog,rock  
tty:x:5:  
disk:x:6:  
lp:x:7:  
mail:x:8:  
news:x:9:  
uucp:x:10:  
man:x:12:  
proxy:x:13:  
kmem:x:15:  
dialout:x:20:  
fax:x:21:  
voice:x:22:  
cdrom:x:24:rock  
floppy:x:25:  
tape:x:26:  
sudo:x:27:rock  
audio:x:29:pulse  
dip:x:30:rock  
www-data:x:33:  
backup:x:34:  
operator:x:37:  
list:x:38:  
irc:x:39:  
src:x:40:  
gnats:x:41:  
shadow:x:42:  
utmp:x:43:  
video:x:44:  
sasl:x:45:  
plugdev:x:46:rock  
staff:x:50:  
games:x:60:  
users:x:100:  
nogroup:x:65534:  
systemd-journal:x:101:  
systemd-timesync:x:102:  
systemd-network:x:103:  
systemd-resolve:x:104:  
systemd-bus-proxy:x:105:  
input:x:106:  
crontab:x:107:  
syslog:x:108:  
netdev:x:109:  
messagebus:x:110:  
uuidd:x:111:  
ssl-cert:x:112:  
lpadmin:x:113:rock  
lightdm:x:114:  
nopasswdlogin:x:115:  
whoopsie:x:116:  
mlocate:x:117:  
ssh:x:118:  
avahi-autoipd:x:119:  
avahi:x:120:  
bluetooth:x:121:  
scanner:x:122:saned  
colord:x:123:  
pulse:x:124:  
pulse-access:x:125:  
rtkit:x:126:  
saned:x:127:  
rock:x:1000:  
sambashare:x:128:rock  
mysql:x:129:  
guest-wxpiwz:x:999:  
zhangsan:x:1001:  
lichen:x:1002:  
python1904:x:1003:  
lichen@lichen-host:~$ sudo groupadd lisi  
lichen@lichen-host:~$ sudo usermod -g lisi lisi  
lichen@lichen-host:~$ ll /home/  
总用量 20  
drwxr-xr-x 5 root root 4096 8月 21 11:31 ./  
drwxr-xr-x 24 root root 4096 3月 25 2018 ../  
drwxr-xr-x 30 lichen lichen 4096 8月 21 10:58 lichen/  
drwxr-xr-x 2 lisi lisi 4096 8月 21 11:31 lisi/  
drwxr-xr-x 2 zhangsan zhangsan 4096 8月 21 10:49 zhangsan/  
lichen@lichen-host:~$ sudo usermod -l jack lisi  
lichen@lichen-host:~$ ll /home/  
总用量 20  
drwxr-xr-x 5 root root 4096 8月 21 11:31 ./  
drwxr-xr-x 24 root root 4096 3月 25 2018 ../  
drwxr-xr-x 30 lichen lichen 4096 8月 21 10:58 lichen/  
drwxr-xr-x 2 jack lisi 4096 8月 21 11:31 lisi/  
drwxr-xr-x 2 zhangsan zhangsan 4096 8月 21 10:49 zhangsan/

12>usermod 修改用户的基本信息

-c<备注> 　修改用户帐号的备注文字。   
-d登入目录> 　修改用户登入时的目录。   
-e<有效期限> 　修改帐号的有效期限。   
-f<缓冲天数> 　修改在密码过期后多少天即关闭该帐号。   
#-g<群组> 　修改用户所属的群组。   
-G<群组> 　修改用户所属的附加群组。   
#-l<帐号名称> 　修改用户帐号名称。   
-L 　锁定用户密码，使密码无效。   
-s<shell> 　修改用户登入后所使用的shell。   
-u<uid> 　修改用户ID。   
-U 　解除密码锁定。  
  
#sudo usermod -g 组名 用户名 ：修改指定用户的组名  
#sudo usermod -l new old :修改用户的用户名

13>groupdel 删除组

"""  
1.如果用户组是某个用户的主组，则删除：  
 sudo userdel -r xxx  
 sudo groupdel xxx  
2.如果用户组不是任何一个用户的主组，则删除：  
 sudo groupdel xxxx  
"""  
  
演示命令：   
lichen@lichen-host:~$ cat /etc/group  
root:x:0:  
daemon:x:1:  
bin:x:2:  
sys:x:3:  
adm:x:4:syslog,rock  
tty:x:5:  
disk:x:6:  
lp:x:7:  
mail:x:8:  
news:x:9:  
uucp:x:10:  
man:x:12:  
proxy:x:13:  
kmem:x:15:  
dialout:x:20:  
fax:x:21:  
voice:x:22:  
cdrom:x:24:rock  
floppy:x:25:  
tape:x:26:  
sudo:x:27:rock  
audio:x:29:pulse  
dip:x:30:rock  
www-data:x:33:  
backup:x:34:  
operator:x:37:  
list:x:38:  
irc:x:39:  
src:x:40:  
gnats:x:41:  
shadow:x:42:  
utmp:x:43:  
video:x:44:  
sasl:x:45:  
plugdev:x:46:rock  
staff:x:50:  
games:x:60:  
users:x:100:  
nogroup:x:65534:  
systemd-journal:x:101:  
systemd-timesync:x:102:  
systemd-network:x:103:  
systemd-resolve:x:104:  
systemd-bus-proxy:x:105:  
input:x:106:  
crontab:x:107:  
syslog:x:108:  
netdev:x:109:  
messagebus:x:110:  
uuidd:x:111:  
ssl-cert:x:112:  
lpadmin:x:113:rock  
lightdm:x:114:  
nopasswdlogin:x:115:  
whoopsie:x:116:  
mlocate:x:117:  
ssh:x:118:  
avahi-autoipd:x:119:  
avahi:x:120:  
bluetooth:x:121:  
scanner:x:122:saned  
colord:x:123:  
pulse:x:124:  
pulse-access:x:125:  
rtkit:x:126:  
saned:x:127:  
rock:x:1000:  
sambashare:x:128:rock  
mysql:x:129:  
guest-wxpiwz:x:999:  
zhangsan:x:1001:  
lichen:x:1002:  
python1904:x:1003:  
lisi:x:1004:  
lichen@lichen-host:~$ sudo groupdel python1904  
lichen@lichen-host:~$ cat /etc/group  
root:x:0:  
daemon:x:1:  
bin:x:2:  
sys:x:3:  
adm:x:4:syslog,rock  
tty:x:5:  
disk:x:6:  
lp:x:7:  
mail:x:8:  
news:x:9:  
uucp:x:10:  
man:x:12:  
proxy:x:13:  
kmem:x:15:  
dialout:x:20:  
fax:x:21:  
voice:x:22:  
cdrom:x:24:rock  
floppy:x:25:  
tape:x:26:  
sudo:x:27:rock  
audio:x:29:pulse  
dip:x:30:rock  
www-data:x:33:  
backup:x:34:  
operator:x:37:  
list:x:38:  
irc:x:39:  
src:x:40:  
gnats:x:41:  
shadow:x:42:  
utmp:x:43:  
video:x:44:  
sasl:x:45:  
plugdev:x:46:rock  
staff:x:50:  
games:x:60:  
users:x:100:  
nogroup:x:65534:  
systemd-journal:x:101:  
systemd-timesync:x:102:  
systemd-network:x:103:  
systemd-resolve:x:104:  
systemd-bus-proxy:x:105:  
input:x:106:  
crontab:x:107:  
syslog:x:108:  
netdev:x:109:  
messagebus:x:110:  
uuidd:x:111:  
ssl-cert:x:112:  
lpadmin:x:113:rock  
lightdm:x:114:  
nopasswdlogin:x:115:  
whoopsie:x:116:  
mlocate:x:117:  
ssh:x:118:  
avahi-autoipd:x:119:  
avahi:x:120:  
bluetooth:x:121:  
scanner:x:122:saned  
colord:x:123:  
pulse:x:124:  
pulse-access:x:125:  
rtkit:x:126:  
saned:x:127:  
rock:x:1000:  
sambashare:x:128:rock  
mysql:x:129:  
guest-wxpiwz:x:999:  
zhangsan:x:1001:  
lichen:x:1002:  
lisi:x:1004:  
lichen@lichen-host:~$ sudo groupdel lisi  
groupdel：不能移除用户“jack”的主组  
lichen@lichen-host:~$ sudo userdel -r jack  
userdel: jack 邮件池 (/var/mail/jack) 未找到  
lichen@lichen-host:~$ sudo groupdel lisi  
lichen@lichen-host:~$ cat /etc/group  
root:x:0:  
daemon:x:1:  
bin:x:2:  
sys:x:3:  
adm:x:4:syslog,rock  
tty:x:5:  
disk:x:6:  
lp:x:7:  
mail:x:8:  
news:x:9:  
uucp:x:10:  
man:x:12:  
proxy:x:13:  
kmem:x:15:  
dialout:x:20:  
fax:x:21:  
voice:x:22:  
cdrom:x:24:rock  
floppy:x:25:  
tape:x:26:  
sudo:x:27:rock  
audio:x:29:pulse  
dip:x:30:rock  
www-data:x:33:  
backup:x:34:  
operator:x:37:  
list:x:38:  
irc:x:39:  
src:x:40:  
gnats:x:41:  
shadow:x:42:  
utmp:x:43:  
video:x:44:  
sasl:x:45:  
plugdev:x:46:rock  
staff:x:50:  
games:x:60:  
users:x:100:  
nogroup:x:65534:  
systemd-journal:x:101:  
systemd-timesync:x:102:  
systemd-network:x:103:  
systemd-resolve:x:104:  
systemd-bus-proxy:x:105:  
input:x:106:  
crontab:x:107:  
syslog:x:108:  
netdev:x:109:  
messagebus:x:110:  
uuidd:x:111:  
ssl-cert:x:112:  
lpadmin:x:113:rock  
lightdm:x:114:  
nopasswdlogin:x:115:  
whoopsie:x:116:  
mlocate:x:117:  
ssh:x:118:  
avahi-autoipd:x:119:  
avahi:x:120:  
bluetooth:x:121:  
scanner:x:122:saned  
colord:x:123:  
pulse:x:124:  
pulse-access:x:125:  
rtkit:x:126:  
saned:x:127:  
rock:x:1000:  
sambashare:x:128:rock  
mysql:x:129:  
guest-wxpiwz:x:999:  
zhangsan:x:1001:  
lichen:x:1002:  
lichen@lichen-host:~$

13>chmod 修改文件权限

#字母法：u[user] g[group] o[other] a[all]  
#数字法：r[4] w[2] x[1] -[0]  
#需要执行的操作：+[增加权限] -[减少权限] =[设置权限]  
  
#格式：  
 #chmod u/g/o/a +/-/= r/w/x filename  
 #chmod 751 filename

14>chown 修改文件所有者

演示命令：  
lichen@lichen-host:~$ sudo groupadd lichen  
[sudo] lichen 的密码：   
lichen@lichen-host:~$ sudo usermod -g lichen lichen

chown、usermod、chmod:三者之间的区别

chown修改文件的所有者，指明文件的用户名和用户组

usermod:修改用户的基本信息，例如：可以修改用户的主组和用户名

chmod:修改文件权限,分为字母法和数字法

#### 3.系统管理

1>ps 报告当前系统的进程状态

进程是处于执行期的程序以及它所包含的所有资源的总称，包括虚拟处理器，虚拟空间，寄存器，堆栈，全局数据段等。  
 针对用户进程，又可以分为交互进程、批处理进程和守护进程三类。   
 (1) 交互进程：由一个shell终端启动的进程，在执行过程中，需要与用户进行交互操作，可以运行于前台，也可以运行在后台。   
 (2) 批处理进程：该进程是一个进程集合，负责按顺序启动其他的进程。   
 (3) 守护进程：守护进程是一直运行的一种进程，经常在linux系统启动时启动，在系统关闭时终止。它们独立于控制终端并且周期性的执行某种任务或等待处理某些发生的事件。例如httpd进程，一直处于运行状态，等待用户的访问。还有经常用的crond进程，这个进程类似与windows的计划任务，可以周期性的执行用户设定的某些任务。  
 ps命令是最基本同时也是非常强大的进程查看命令，使用该命令可以确定有哪些进程正在运行和运行的状态、进程是否结束、进程有没有僵死、哪些进程占用了过多的资源等等，总之大部分信息都是可以通过执行该命令得到的

-a：显示所有进程（包括其他用户的进程）

-u：用户以及其他详细信息

-x：显示没有控制终端的进程

-ef：显示所有【包含后台的进程】

ps  
ps -a  
ps -u  
ps -x  
ps -ef  
ps -aux

2>kill 删除执行中的程序或工作

-a：当处理当前进程时，不限制命令名和进程号的对应关系；  
 -l <信息编号>：若不加<信息编号>选项，则-l参数会列出全部的信息名称；  
 -p：指定kill 命令只打印相关进程的进程号，而不发送任何信号；  
 -s <信息名称或编号>：指定要送出的信息；  
 -u：指定用户

格式：kill pid

注意：pid是由系统分配的，是一个进程唯一的标识

3>df 显示磁盘分区上的可使用的磁盘空间

注意：默认显示单位为KB

可以利用该命令来获取硬盘被占用了多少空间，目前还剩下多少空间等信息

4>du 显示文件的内存大小

注意：与[df](http://man.linuxde.net/df)命令不同的是du命令是对文件和目录磁盘使用的空间的查看,df查看每个文件的使用情况

5>crontab 提交和管理用户的需要周期性执行的任务

注意：当安装完成操作系统后，默认会安装此服务工具，并且会自动启动crond进程，crond进程每分钟会定期检查是否有要执行的任务，如果有要执行的任务，则自动执行该任务

6>free 显示当前系统未使用的和已使用的内存数目，还可以显示被内核使用的内存缓冲区

-b：以Byte为单位显示内存使用情况；

-k：以KB为单位显示内存使用情况；

-m：以MB为单位显示内存使用情况；

-o：不显示缓冲区调节列；

-s<间隔秒数>：持续观察内存使用状况；

-t：显示内存总和列；

-V：显示版本信息

7>mode 权限标记

#符号标记法修改权限：  
mode由三部分组成：操作对象，操作符，permission  
  
操作对象（可以是下面字母中的任何一个或者他们的组合）：  
 1、u 表示用户user  
 2、g 表示同组group用户  
 3、o 表示其他other用户  
 4、a 表示all用户  
   
操作符：  
 + 添加某个权限  
 - 取消某个权限  
 = 赋予给定权限并取消其他权限，即设定唯一权限  
   
permission（可以是下面字母中的任何一个或者他们的组合）：  
 r 可读  
 w 可写  
 x 可执行

8>alias 给命令起别名

格式：alias 别名=原名

$ sudo vim .bashrc  
[sudo] lichen 的密码：   
lichen@lichen-host:~$ source .bashrc

9>关机重启

#reboot 重启  
shutdown -h now 立即关机  
shutdown -r now 立即重启  
shutdown -h +1 一分钟后重启  
  
#init 0 :关机  
#init 6：重启  
  
telinit 0 :关机  
logout ：注销  
  
Linux有7个运行级别：  
 #0：关机，停机模式  
 #1：单用户模式  
 #2.多用户模式  
 3.完整的多用户文本模式  
 4.系统为使用，保留一般不用  
 #5.图形化模式  
 #6.重启模式  
   
补充：  
hostname:查看主机名  
last：查看最近的登录状态  
  
lichen@lichen-host:~$ kill -l  
1) SIGHUP 2) SIGINT 3) SIGQUIT 4) SIGILL 5) SIGTRAP  
6) SIGABRT 7) SIGBUS 8) SIGFPE 9) SIGKILL 10) SIGUSR1  
11) SIGSEGV 12) SIGUSR2 13) SIGPIPE 14) SIGALRM 15) SIGTERM  
16) SIGSTKFLT 17) SIGCHLD 18) SIGCONT 19) SIGSTOP 20) SIGTSTP  
21) SIGTTIN 22) SIGTTOU 23) SIGURG 24) SIGXCPU 25) SIGXFSZ  
26) SIGVTALRM 27) SIGPROF 28) SIGWINCH 29) SIGIO 30) SIGPWR  
31) SIGSYS 34) SIGRTMIN 35) SIGRTMIN+1 36) SIGRTMIN+2 37) SIGRTMIN+3  
38) SIGRTMIN+4 39) SIGRTMIN+5 40) SIGRTMIN+6 41) SIGRTMIN+7 42) SIGRTMIN+8  
43) SIGRTMIN+9 44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+13  
48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-12  
53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9 56) SIGRTMAX-8 57) SIGRTMAX-7  
58) SIGRTMAX-6 59) SIGRTMAX-5 60) SIGRTMAX-4 61) SIGRTMAX-3 62) SIGRTMAX-2  
63) SIGRTMAX-1 64) SIGRTMAX   
  
kill发送15给进程，告诉进程，你需要被关闭，请自行停止运行并退出  
kill发送9给进程，告诉进程，你被关闭了【kill -9表示强制终止某个进程】

10>date 显示当前日期

-s：设置时间，注意：只有root权限才能设置，其他只能查看  
日期格式化  
%Y year  
%m month (01..12)  
%d day of month (e.g., 01)  
%H hour (00..23)  
%I hour (01..12)  
%M minute (00..59)  
%S second (00..60)

11>cal 显示一个日历

#### 4.网络管理

ping

检测网络连通性【检测有没有网】

ping IP地址

ping -c n IP地址 连接 n 次结束

ifconfig[windows中ipconfig]

查看、设置网卡信息

演示命令：  
lichen@lichen-host:~/Desktop$ ifconfig  
ens33 Link encap:以太网 硬件地址 00:0c:29:c0:77:76   
 inet 地址:10.36.172.26 广播:10.36.172.255 掩码:255.255.255.0  
 inet6 地址: fe80::6927:79d:16e0:e9a0/64 Scope:Link  
 UP BROADCAST RUNNING MULTICAST MTU:1500 跃点数:1  
 接收数据包:44833 错误:0 丢弃:0 过载:0 帧数:0  
 发送数据包:8021 错误:0 丢弃:0 过载:0 载波:0  
 碰撞:0 发送队列长度:1000   
 接收字节:22580642 (22.5 MB) 发送字节:571354 (571.3 KB)  
  
lo Link encap:本地环回   
 inet 地址:127.0.0.1 掩码:255.0.0.0  
 inet6 地址: ::1/128 Scope:Host  
 UP LOOPBACK RUNNING MTU:65536 跃点数:1  
 接收数据包:304 错误:0 丢弃:0 过载:0 帧数:0  
 发送数据包:304 错误:0 丢弃:0 过载:0 载波:0  
 碰撞:0 发送队列长度:1000   
 接收字节:30802 (30.8 KB) 发送字节:30802 (30.8 KB)  
  
lichen@lichen-host:~/Desktop$ ping www.baidu.com  
PING www.a.shifen.com (182.61.200.7) 56(84) bytes of data.  
64 bytes from 182.61.200.7: icmp\_seq=1 ttl=52 time=23.6 ms  
64 bytes from 182.61.200.7: icmp\_seq=2 ttl=52 time=20.5 ms  
64 bytes from 182.61.200.7: icmp\_seq=3 ttl=52 time=44.8 ms  
64 bytes from 182.61.200.7: icmp\_seq=4 ttl=52 time=20.4 ms  
64 bytes from 182.61.200.7: icmp\_seq=5 ttl=52 time=20.7 ms  
64 bytes from 182.61.200.7: icmp\_seq=6 ttl=52 time=20.7 ms  
64 bytes from 182.61.200.7: icmp\_seq=7 ttl=52 time=20.7 ms  
64 bytes from 182.61.200.7: icmp\_seq=8 ttl=52 time=20.0 ms  
64 bytes from 182.61.200.7: icmp\_seq=9 ttl=52 time=20.7 ms  
^C  
--- www.a.shifen.com ping statistics ---  
9 packets transmitted, 9 received, 0% packet loss, time 8016ms  
rtt min/avg/max/mdev = 20.055/23.612/44.815/7.560 ms  
lichen@lichen-host:~/Desktop$

### 三、git的使用

#### 1.git简介

如果你用Word写过毕业论文，那你一定有这样的经历：

想删除一个段落，又怕将来想恢复找不回来怎么办？有办法，先把当前文件“另存为……”一个新的Word文件，再接着改，改到一定程度，再“另存为……”一个新文件，这样一直改下去，最后你的Word文档就会变得乱七八糟

过了一周，你想找回被删除的文字，但是已经记不清删除前保存在哪个文件里了，只好一个一个文件去找，真麻烦。

看着一堆乱七八糟的文件，想保留最新的一个，然后把其他的删掉，又怕哪天会用上，还不敢删，真郁闷。

更要命的是，有些部分需要你的财务同事帮助填写，于是你把文件Copy到U盘里给她（也可能通过Email发送一份给她），然后，你继续修改Word文件。一天后，同事再把Word文件传给你，此时，你必须想想，发给她之后到你收到她的文件期间，你作了哪些改动，得把你的改动和她的部分合并，真困难。

于是你想，如果有一个软件，不但能自动帮我记录每次文件的改动，还可以让同事协作编辑，这样就不用自己管理一堆类似的文件了，也不需要把文件传来传去。如果想查看某次改动，只需要在软件里瞄一眼就可以，岂不是很方便？

这个软件用起来就应该像这个样子，能记录每次文件的改动：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 文件名 | 用户 | 说明 | 日期 |
| 1 | service.doc | 张三 | 删除了软件服务条款5 | 7/12 10:38 |
| 2 | service.doc | 张三 | 增加了License人数限制 | 7/12 18:09 |
| 3 | service.doc | 李四 | 财务部门调整了合同金额 | 7/13 9:51 |
| 4 | service.doc | 张三 | 延长了免费升级周期 | 7/14 15:17 |

这样，你就结束了手动管理多个“版本”的史前时代，进入到版本控制的20世纪。

Git是目前世界上最先进的【分布式版本控制系统】（没有之一）。

Git有什么特点？简单来说就是：高端大气上档次！

##### 1.2git的由来

很多人都知道，Linus在1991年创建了开源的Linux，从此，Linux系统不断发展，已经成为最大的服务器系统软件了。

Linus虽然创建了Linux，但Linux的壮大是靠全世界热心的志愿者参与的，这么多人在世界各地为Linux编写代码，那Linux的代码是如何管理的呢？

事实是，在2002年以前，世界各地的志愿者把源代码文件通过diff的方式发给Linus，然后由Linus本人通过手工方式合并代码！

你也许会想，为什么Linus不把Linux代码放到版本控制系统里呢？不是有CVS、SVN这些免费的版本控制系统吗？因为Linus坚定地反对CVS和SVN，这些集中式的版本控制系统不但速度慢，而且必须联网才能使用。有一些商用的版本控制系统，虽然比CVS、SVN好用，但那是付费的，和Linux的开源精神不符。

不过，到了2002年，Linux系统已经发展了十年了，代码库之大让Linus很难继续通过手工方式管理了，社区的弟兄们也对这种方式表达了强烈不满，于是Linus选择了一个商业的版本控制系统BitKeeper，BitKeeper的东家BitMover公司出于人道主义精神，授权Linux社区免费使用这个版本控制系统。

安定团结的大好局面在2005年就被打破了，原因是Linux社区牛人聚集，不免沾染了一些梁山好汉的江湖习气。开发Samba的Andrew试图破解BitKeeper的协议（这么干的其实也不只他一个），被BitMover公司发现了（监控工作做得不错！），于是BitMover公司怒了，要收回Linux社区的免费使用权。

Linus可以向BitMover公司道个歉，保证以后严格管教弟兄们，嗯，这是不可能的。实际情况是这样的：

Linus花了两周时间自己用C写了一个分布式版本控制系统，这就是Git！一个月之内，Linux系统的源码已经由Git管理了！牛是怎么定义的呢？大家可以体会一下。

Git迅速成为最流行的分布式版本控制系统，尤其是2008年，GitHub网站上线了，它为开源项目免费提供Git存储，无数开源项目开始迁移至GitHub，包括jQuery，PHP，Ruby等等。

历史就是这么偶然，如果不是当年BitMover公司威胁Linux社区，可能现在我们就没有免费而超级好用的Git了

##### 1.3集中式和分布式【面试题】

git是一个分布式版本控制系统，svn是集中式版本控制系统

集中式版本控制系统：版本库是集中存放在中央服务器上，而干活的时候，每个人使用的是自己的电脑，所以先要中央服务器上获取最新的版本库，然后开始干活，再把自己干的活推送到中央服务器上【缺点：必须联网才能工作，在网速差或者没网的情况下，工作效率很低】

分布式版本控制系统：根本没有中央服务器，每个人的电脑上就是一个完整的版本库，这样，在你工作的时候，就不需要联网了，因为版本库就在你的电脑上

和集中式版本控制系统相比，分布式版本控制系统的好处：安全性更高，因为每人的电脑上都有一个完整的版本库

实际情况：使用分布式版本控制系统进行版本控制，很少在两个人的电脑之间直接进行推送修改，通常也有一个充当"中央服务器"的电脑，该电脑的作用仅仅是为了交换团队内不同人的修改

git的优势：不需要联网可以工作，分支管理和标签管理

git是一个工具，github是一个网站【充当远程服务器】

#### 2.安装git

sudo apt-get install git

#### 3.创建版本库

##### 3.1什么是版本库

版本库又称为仓库，英文名称repository,初期理解为一个目录，该目录中的所有的文件都可以被git管理，每个文件的修改，增加，删除等git都能够追踪，以便于任何时刻可以追踪历史版本，可以在将来某个时刻恢复到过去的某个版本

##### 3.2创建版本库

git init:将一个空的目录初始化为一个空的版本库，在该目录下进行的所有的工作都将被git追踪，

演示命令：  
ichen@lichen-host:~$ sudo apt-get install git  
正在读取软件包列表... 完成  
正在分析软件包的依赖关系树   
正在读取状态信息... 完成   
将会同时安装下列软件：  
git-man  
建议安装：  
git-daemon-run | git-daemon-sysvinit git-doc git-el  
git-email git-gui gitk gitweb git-cvs git-mediawiki  
git-svn  
下列软件包将被升级：  
git git-man  
升级了 2 个软件包，新安装了 0 个软件包，要卸载 0 个软件包，有 525 个软件包未被升级。  
需要下载 7,736 kB 的归档。  
解压缩后会消耗 6,342 kB 的额外空间。  
您希望继续执行吗？ [Y/n] y  
获取:1 http://ppa.launchpad.net/git-core/ppa/ubuntu xenial/main amd64 git amd64 1:2.23.0-0ppa1~ubuntu16.04.1 [6,059 kB]  
获取:2 http://ppa.launchpad.net/git-core/ppa/ubuntu xenial/main amd64 git-man all 1:2.23.0-0ppa1~ubuntu16.04.1 [1,678 kB]  
已下载 7,736 kB，耗时 34秒 (227 kB/s)   
debconf: 无法初始化前端界面：Dialog  
debconf: (对话框界面要求屏幕画面必须为至少 13 行高及 31 列宽.)  
debconf: 返回前端界面：Readline  
(正在读取数据库 ... 系统当前共安装有 222309 个文件和目录。)  
正准备解包 .../git\_1%3a2.23.0-0ppa1~ubuntu16.04.1\_amd64.deb ...  
正在将 git (1:2.23.0-0ppa1~ubuntu16.04.1) 解包到 (1:2.16.2-0ppa1~ubuntu16.04.1) 上 ...  
正准备解包 .../git-man\_1%3a2.23.0-0ppa1~ubuntu16.04.1\_all.deb ...  
正在将 git-man (1:2.23.0-0ppa1~ubuntu16.04.1) 解包到 (1:2.16.2-0ppa1~ubuntu16.04.1) 上 ...  
正在处理用于 man-db (2.7.5-1) 的触发器 ...  
正在设置 git-man (1:2.23.0-0ppa1~ubuntu16.04.1) ...  
正在设置 git (1:2.23.0-0ppa1~ubuntu16.04.1) ...  
lichen@lichen-host:~$ cd Desktop/  
lichen@lichen-host:~/Desktop$ ls  
a1.txt b1.txt  
lichen@lichen-host:~/Desktop$ rm -rf \*  
lichen@lichen-host:~/Desktop$ ls  
lichen@lichen-host:~/Desktop$ mkdir python1904  
lichen@lichen-host:~/Desktop$ ls  
python1904  
lichen@lichen-host:~/Desktop$ cd python1904/  
lichen@lichen-host:~/Desktop/python1904$ git init  
已初始化空的 Git 仓库于 /home/lichen/Desktop/python1904/.git/  
lichen@lichen-host:~/Desktop/python1904$ ls -a  
. .. .git  
lichen@lichen-host:~/Desktop/python1904$ cd .git/  
lichen@lichen-host:~/Desktop/python1904/.git$ ls  
branches config description HEAD hooks info objects refs  
lichen@lichen-host:~/Desktop/python1904/.git$ cd ..

##### 3.3将修改提交到版本库

注意1：git只能追踪修改【新建一个文件，删除一个文件，增加内容，删除内容，修改内容等】

注意2：在工作区【目录】中进行项目的开发工作，然后将修改提交到版本库

#步骤：在工作区进行工作------》使用git add filename 添加到暂存区-----》使用git commit -m "xxx"将暂存区的内容提交到本地版本库  
  
#注意：git add添加的是文件，git commit提交的是暂存区中的所有内容，所以，可以执行多次git add，最后执行一次git commit  
  
演示命令：  
lichen@lichen-host:~/Desktop/python1904$ ls  
lichen@lichen-host:~/Desktop/python1904$ touch a1.txt  
lichen@lichen-host:~/Desktop/python1904$ vim a1.txt   
lichen@lichen-host:~/Desktop/python1904$ git status  
位于分支 master  
  
尚无提交  
  
未跟踪的文件:  
（使用 "git add <文件>..." 以包含要提交的内容）  
 a1.txt  
  
提交为空，但是存在尚未跟踪的文件（使用 "git add" 建立跟踪）  
lichen@lichen-host:~/Desktop/python1904$ git add a1.txt   
lichen@lichen-host:~/Desktop/python1904$ git status  
位于分支 master  
  
尚无提交  
  
要提交的变更：  
（使用 "git rm --cached <文件>..." 以取消暂存）  
 新文件： a1.txt  
  
lichen@lichen-host:~/Desktop/python1904$ git commit -m "create file named a1.txt"  
  
\*\*\* 请告诉我你是谁。  
  
运行  
  
git config --global user.email "you@example.com"  
git config --global user.name "Your Name"  
  
来设置您账号的缺省身份标识。  
如果仅在本仓库设置身份标识，则省略 --global 参数。  
fatal: 无法自动探测邮件地址（得到 'lichen@lichen-host.(none)'）  
lichen@lichen-host:~/Desktop/python1904$ git config --global user.email "18501970795@163.com"  
lichen@lichen-host:~/Desktop/python1904$ git config --global user.name "yangyang-git"  
lichen@lichen-host:~/Desktop/python1904$ git commit -m "create file named a1.txt"  
[master（根提交） a70b59b] create file named a1.txt  
1 file changed, 1 insertion(+)  
create mode 100644 a1.txt  
lichen@lichen-host:~/Desktop/python1904$ git status  
位于分支 master  
无文件要提交，干净的工作区  
lichen@lichen-host:~/Desktop/python1904$ vim a1.txt   
lichen@lichen-host:~/Desktop/python1904$ git status  
位于分支 master  
尚未暂存以备提交的变更：  
（使用 "git add <文件>..." 更新要提交的内容）  
（使用 "git restore <文件>..." 丢弃工作区的改动）  
 修改： a1.txt  
  
修改尚未加入提交（使用 "git add" 和/或 "git commit -a"）  
lichen@lichen-host:~/Desktop/python1904$ git add a1.txt   
lichen@lichen-host:~/Desktop/python1904$ git commit -m "add 1234"  
[master d6699f5] add 1234  
1 file changed, 1 insertion(+)  
lichen@lichen-host:~/Desktop/python1904$ git status  
位于分支 master  
无文件要提交，干净的工作区  
lichen@lichen-host:~/Desktop/python1904$ vim a1.txt   
lichen@lichen-host:~/Desktop/python1904$ git add a1.txt   
lichen@lichen-host:~/Desktop/python1904$ touch b1.txt  
lichen@lichen-host:~/Desktop/python1904$ vim b1.txt   
lichen@lichen-host:~/Desktop/python1904$ git add b1.txt   
lichen@lichen-host:~/Desktop/python1904$ git status  
位于分支 master  
要提交的变更：  
（使用 "git restore --staged <文件>..." 以取消暂存）  
 修改： a1.txt  
 新文件： b1.txt  
  
lichen@lichen-host:~/Desktop/python1904$ git commit -m "modify a1 and create b1"  
[master 784af92] modify a1 and create b1  
2 files changed, 2 insertions(+)  
create mode 100644 b1.txt

#### 4.时光穿梭机

版本的任意切换【回退版本和回到未来的版本】

git status:查看仓库当前的状态【本地版本库和工作区之间的差异】，可以随时掌握工作区的状态

git diff:查看具体修改的内容

##### 4.1回退版本

git在修改文件的过程中，每提交一次，就相当于保存了一个“快照”，这个快照在git中被称为commit

版本回退：如果在某一时刻某个文件被修改乱了，可以通过commit恢复到原来的版本

HEAD是一个指针，指向最新一次的commit，所以回退版本其实就是改变HEAD指针的指向

需求：回到上一个版本

分析：要回退版本，首先需要让git直到回到哪个版本，在git中，HEAD表示当前版本，上一个版本是HEAD^,上上一个版本HEAD^^,向上数100个版本，HEAD~100

#git log:查看日志【查看提交的历史版本的信息：commitid,作者，日期，日志】,查看提交历史，以便于确定回退到哪个版本  
#git log --pretty=oneline：查看日志,以一行的形式显示日志信息  
#git reflog:要回到未来的版本，可以查看历史命令，以便于确定回到未来的哪个版本  
#git reset --hard HEAD^ :回退到上一个版本  
#git reset --hard commitid:回到指定的版本  
  
演示命令：  
lichen@lichen-host:~/Desktop/python1904$ git log  
commit 784af92d61d9327f6688e978ba280e5397e8b965 (HEAD -> master)  
Author: yangyang-git <18501970795@163.com>  
Date: Wed Aug 21 15:45:46 2019 +0800  
  
 modify a1 and create b1  
  
commit d6699f57ed76496db723955bf9007058ca523349  
Author: yangyang-git <18501970795@163.com>  
Date: Wed Aug 21 15:42:19 2019 +0800  
  
 add 1234  
  
commit a70b59bffdbb6a56f3c80f944591d36bb092a6b7  
Author: yangyang-git <18501970795@163.com>  
Date: Wed Aug 21 15:40:41 2019 +0800  
  
 create file named a1.txt  
lichen@lichen-host:~/Desktop/python1904$ vim b1.txt   
lichen@lichen-host:~/Desktop/python1904$ git add b1.txt  
lichen@lichen-host:~/Desktop/python1904$ git commit -m "modify b1"  
[master 28a7147] modify b1  
1 file changed, 2 insertions(+)  
lichen@lichen-host:~/Desktop/python1904$ git log  
commit 28a71476338622f3c3820201c794b3ba27320d5d (HEAD -> master)  
Author: yangyang-git <18501970795@163.com>  
Date: Wed Aug 21 15:59:28 2019 +0800  
  
 modify b1  
  
commit 784af92d61d9327f6688e978ba280e5397e8b965  
Author: yangyang-git <18501970795@163.com>  
Date: Wed Aug 21 15:45:46 2019 +0800  
  
 modify a1 and create b1  
  
commit d6699f57ed76496db723955bf9007058ca523349  
Author: yangyang-git <18501970795@163.com>  
Date: Wed Aug 21 15:42:19 2019 +0800  
  
 add 1234  
  
commit a70b59bffdbb6a56f3c80f944591d36bb092a6b7  
Author: yangyang-git <18501970795@163.com>  
Date: Wed Aug 21 15:40:41 2019 +0800  
  
:...skipping...  
commit 28a71476338622f3c3820201c794b3ba27320d5d (HEAD -> master)  
Author: yangyang-git <18501970795@163.com>  
Date: Wed Aug 21 15:59:28 2019 +0800  
  
 modify b1  
  
commit 784af92d61d9327f6688e978ba280e5397e8b965  
Author: yangyang-git <18501970795@163.com>  
Date: Wed Aug 21 15:45:46 2019 +0800  
  
 modify a1 and create b1  
  
commit d6699f57ed76496db723955bf9007058ca523349  
Author: yangyang-git <18501970795@163.com>  
Date: Wed Aug 21 15:42:19 2019 +0800  
  
 add 1234  
  
commit a70b59bffdbb6a56f3c80f944591d36bb092a6b7  
Author: yangyang-git <18501970795@163.com>  
Date: Wed Aug 21 15:40:41 2019 +0800  
  
 create file named a1.txt  
:...skipping...  
commit 28a71476338622f3c3820201c794b3ba27320d5d (HEAD -> master)  
Author: yangyang-git <18501970795@163.com>  
Date: Wed Aug 21 15:59:28 2019 +0800  
  
 modify b1  
  
commit 784af92d61d9327f6688e978ba280e5397e8b965  
Author: yangyang-git <18501970795@163.com>  
Date: Wed Aug 21 15:45:46 2019 +0800  
  
 modify a1 and create b1  
  
commit d6699f57ed76496db723955bf9007058ca523349  
Author: yangyang-git <18501970795@163.com>  
Date: Wed Aug 21 15:42:19 2019 +0800  
  
 add 1234  
  
commit a70b59bffdbb6a56f3c80f944591d36bb092a6b7  
Author: yangyang-git <18501970795@163.com>  
Date: Wed Aug 21 15:40:41 2019 +0800  
  
 create file named a1.txt  
~  
lichen@lichen-host:~/Desktop/python1904$ git log --pretty=oneline  
28a71476338622f3c3820201c794b3ba27320d5d (HEAD -> master) modify b1  
784af92d61d9327f6688e978ba280e5397e8b965 modify a1 and create b1  
d6699f57ed76496db723955bf9007058ca523349 add 1234  
a70b59bffdbb6a56f3c80f944591d36bb092a6b7 create file named a1.txt  
lichen@lichen-host:~/Desktop/python1904$ cat a1.txt   
aaaaaaa  
5454356  
hello  
lichen@lichen-host:~/Desktop/python1904$ cat b1.txt   
bbbbb  
fhajkhg  
fhajkgh  
lichen@lichen-host:~/Desktop/python1904$ vim a1.txt   
lichen@lichen-host:~/Desktop/python1904$ git add a1.txt   
lichen@lichen-host:~/Desktop/python1904$ git commit -m "add content at a1"  
[master a0468a2] add content at a1  
1 file changed, 1 insertion(+)  
lichen@lichen-host:~/Desktop/python1904$ git log --pretty=oneline  
a0468a21bdcb7ada619bcbe2dec67393ef415c72 (HEAD -> master) add content at a1  
28a71476338622f3c3820201c794b3ba27320d5d modify b1  
784af92d61d9327f6688e978ba280e5397e8b965 modify a1 and create b1  
d6699f57ed76496db723955bf9007058ca523349 add 1234  
a70b59bffdbb6a56f3c80f944591d36bb092a6b7 create file named a1.txt  
lichen@lichen-host:~/Desktop/python1904$ cat a1.txt   
aaaaaaa  
5454356  
hello  
this is a text  
lichen@lichen-host:~/Desktop/python1904$ git reset --hard HEAD^  
HEAD 现在位于 28a7147 modify b1  
lichen@lichen-host:~/Desktop/python1904$ cat a1.txt   
aaaaaaa  
5454356  
hello  
lichen@lichen-host:~/Desktop/python1904$ git reset --hard a0468a21  
HEAD 现在位于 a0468a2 add content at a1  
lichen@lichen-host:~/Desktop/python1904$ cat a1.txt   
aaaaaaa  
5454356  
hello  
this is a text  
lichen@lichen-host:~/Desktop/python1904$ git reset --hard a70b59bf  
HEAD 现在位于 a70b59b create file named a1.txt  
lichen@lichen-host:~/Desktop/python1904$ ls  
a1.txt  
lichen@lichen-host:~/Desktop/python1904$ git log --pretty=onelin  
a70b59bffdbb6a56f3c80f944591d36bb092a6b7 (HEAD -> master) create file named a1.txt  
lichen@lichen-host:~/Desktop/python1904$ git log --pretty=oneline  
a70b59bffdbb6a56f3c80f944591d36bb092a6b7 (HEAD -> master) create file named a1.txt  
lichen@lichen-host:~/Desktop/python1904$ git reflog  
a70b59b (HEAD -> master) HEAD@{0}: reset: moving to a70b59bf  
a0468a2 HEAD@{1}: reset: moving to a0468a21

##### 4.2撤销修改

流程：工作区----》暂存区----》版本库

a.在工作区中修改了文件，但是还未add

#git restore filename；丢弃工作区的改动  
lichen@lichen-host:~/Desktop/python1904$ vim a1.txt   
lichen@lichen-host:~/Desktop/python1904$ git status  
位于分支 master  
尚未暂存以备提交的变更：  
（使用 "git add <文件>..." 更新要提交的内容）  
（使用 "git restore <文件>..." 丢弃工作区的改动）  
 修改： a1.txt  
  
修改尚未加入提交（使用 "git add" 和/或 "git commit -a"）  
lichen@lichen-host:~/Desktop/python1904$ cat a1.txt   
aaaaaaa  
5454356  
hello  
this is a text  
11111  
lichen@lichen-host:~/Desktop/python1904$ git restore a1.txt  
lichen@lichen-host:~/Desktop/python1904$ cat a1.txt   
aaaaaaa  
5454356  
hello  
this is a text

b.将修改add到了暂存区

#1.取消暂存：git restore --staged filname #2.丢弃工作区的改动：git restore filename

c.将修改commit到了版本库

lichen@lichen-host:~/Desktop/python1904$ vim a1.txt   
lichen@lichen-host:~/Desktop/python1904$ git add a1.txt   
lichen@lichen-host:~/Desktop/python1904$ git commit -m "jfaf"  
[master 8c58c61] jfaf  
1 file changed, 1 insertion(+)  
lichen@lichen-host:~/Desktop/python1904$ cat a1.txt   
aaaaaaa  
5454356  
hello  
this is a text  
boss is stupid  
lichen@lichen-host:~/Desktop/python1904$ git reset --hard HEAD^  
HEAD 现在位于 a0468a2 add content at a1  
lichen@lichen-host:~/Desktop/python1904$ cat a1.txt   
aaaaaaa  
5454356  
hello  
this is a text

##### 4.3删除文件

步骤：1.删除工作区的文件 2.删除版本库中文件 3.提交修改

lichen@lichen-host:~/Desktop/python1904$ ls  
a1.txt b1.txt  
lichen@lichen-host:~/Desktop/python1904$ git status  
位于分支 master  
无文件要提交，干净的工作区  
lichen@lichen-host:~/Desktop/python1904$ rm -rf b1.txt   
lichen@lichen-host:~/Desktop/python1904$ ls  
a1.txt  
lichen@lichen-host:~/Desktop/python1904$ git status  
位于分支 master  
尚未暂存以备提交的变更：  
（使用 "git add/rm <文件>..." 更新要提交的内容）  
（使用 "git restore <文件>..." 丢弃工作区的改动）  
 删除： b1.txt  
  
修改尚未加入提交（使用 "git add" 和/或 "git commit -a"）  
lichen@lichen-host:~/Desktop/python1904$ git rm b1.txt   
rm 'b1.txt'  
lichen@lichen-host:~/Desktop/python1904$ git status  
位于分支 master  
要提交的变更：  
（使用 "git restore --staged <文件>..." 以取消暂存）  
 删除： b1.txt  
  
lichen@lichen-host:~/Desktop/python1904$ git commit -m "delete b1"  
[master 384052d] delete b1  
1 file changed, 3 deletions(-)  
delete mode 100644 b1.txt  
lichen@lichen-host:~/Desktop/python1904$ git status  
位于分支 master  
无文件要提交，干净的工作区

#### 5.远程仓库

##### 5.1准备工作

a.创建github账号

b.生成ssh key,添加到远程仓库中【作用：建立了本机和远程仓库之间的连接，在同一时刻一个ssh key只能被一个github账号绑定】

命令：ssh-keygen -t rsa -C "github的邮箱"

【补充：ssh key让github确定推送的修改是你推送的，github只需知道你的公钥，就可以确认自己才能推送，github允许同时添加多个公钥，多台电脑可以同时向同一个github账号上推送修改】

演示命令:  
lichen@lichen-host:~$ ssh-keygen -t rsa -C "18501970795@163.com"  
Generating public/private rsa key pair.  
Enter file in which to save the key (/home/lichen/.ssh/id\_rsa):   
Created directory '/home/lichen/.ssh'.  
Enter passphrase (empty for no passphrase):   
Enter same passphrase again:   
Your identification has been saved in /home/lichen/.ssh/id\_rsa.  
Your public key has been saved in /home/lichen/.ssh/id\_rsa.pub.  
The key fingerprint is:  
SHA256:Dd7tE3bqVttqtJFTkbwQXmuNNid+ZlsIQQ0X3SVDYQQ 18501970795@163.com  
The key's randomart image is:  
+---[RSA 2048]----+  
| .EB%==|  
| .=oB=|  
| . ..\*o=|  
| . + . +.\*.|  
| S o + +o=|  
| o +\*+o|  
| +o B |  
| ...+ .|  
| ..... |  
+----[SHA256]-----+  
lichen@lichen-host:~$ cd .ssh/  
lichen@lichen-host:~/.ssh$ ls  
id\_rsa id\_rsa.pub  
lichen@lichen-host:~/.ssh$ cat id\_rsa.pub   
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQD0sC7edEyLlzY6LvvApO8G0Dz6nNwm39+yDe86F+w6vNUPJQWzS2qVGL5YHX6tqsFKhv4Y7f2T+R5o2i5/dh6Nyjb7/VZJCRtKs4QirjxkXbT42wR/pSTIBhHDm0jJQO9pJSa+/vf7IGBVcK5b0pic6MMpMGM9llXfq4w6Z1HsqO+gtXT/p9seyT+GQDXsN+kGvoiwv6q+GO3o6CMIV1JNBJBcqurzffBTetN7dyqPMaBSK3Q3i1Hlw30O6Yl3B8+3ISrwrFyGz/lvjsxhHJ6LAdgVjczUHvqUq3HRp2UiNsnAQH8JCf6cpe+5wvU1TFQXHw9oxYnmzHRZcJt5txsr 18501970795@163.com  
lichen@lichen-host:~/.ssh$ ssh -T git@github.com  
The authenticity of host 'github.com (52.74.223.119)' can't be established.  
RSA key fingerprint is SHA256:nThbg6kXUpJWGl7E1IGOCspRomTxdCARLviKw6E5SY8.  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added 'github.com,52.74.223.119' (RSA) to the list of known hosts.  
Hi yangyang-git! You've successfully authenticated, but GitHub does not provide shell access.

#####5.2添加远程仓库【先有本地版本库，再有远程版本库】

#步骤：  
a.创建远程工作区  
b.建立本地工作区和远程工作区之间的连接：git remote add origin git@github.com:github账户名/远程工作区的名称.git  
c.建立本地版本库和远程版本库之间的连接：git push -u origin master  
#d。在工作的时候，git add filename---->git commit -m "xx"---->git push origin master  
  
#说明：origin其实可以是任意的标识符，但是默认为origin  
  
演示命令：  
lichen@lichen-host:~/.ssh$ cd ..  
lichen@lichen-host:~$ cd Desktop/  
lichen@lichen-host:~/Desktop$ cd python1904/  
lichen@lichen-host:~/Desktop/python1904$ git remote add origin git@github.com:yangyang-git/python1904.git  
lichen@lichen-host:~/Desktop/python1904$ ls  
a1.txt  
lichen@lichen-host:~/Desktop/python1904$ git push -u origin master  
枚举对象: 18, 完成.  
对象计数中: 100% (18/18), 完成.  
使用 8 个线程进行压缩  
压缩对象中: 100% (9/9), 完成.  
写入对象中: 100% (18/18), 1.39 KiB | 711.00 KiB/s, 完成.  
总共 18 （差异 0），复用 0 （差异 0）  
To github.com:yangyang-git/python1904.git  
 \* [new branch] master -> master  
分支 'master' 设置为跟踪来自 'origin' 的远程分支 'master'。  
lichen@lichen-host:~/Desktop/python1904$ touch b1.txt  
lichen@lichen-host:~/Desktop/python1904$ vim b1.txt   
lichen@lichen-host:~/Desktop/python1904$ git add b1.txt   
lichen@lichen-host:~/Desktop/python1904$ git commit -m "fsgd"  
[master b1283d3] fsgd  
1 file changed, 1 insertion(+)  
create mode 100644 b1.txt  
lichen@lichen-host:~/Desktop/python1904$ git status  
位于分支 master  
您的分支领先 'origin/master' 共 1 个提交。  
（使用 "git push" 来发布您的本地提交）  
  
无文件要提交，干净的工作区  
lichen@lichen-host:~/Desktop/python1904$ git push origin master  
枚举对象: 4, 完成.  
对象计数中: 100% (4/4), 完成.  
使用 8 个线程进行压缩  
压缩对象中: 100% (2/2), 完成.  
写入对象中: 100% (3/3), 271 字节 | 271.00 KiB/s, 完成.  
总共 3 （差异 0），复用 0 （差异 0）  
To github.com:yangyang-git/python1904.git  
384052d..b1283d3 master -> master  
lichen@lichen-host:~/Desktop/python1904$ git status  
位于分支 master  
您的分支与上游分支 'origin/master' 一致。  
  
无文件要提交，干净的工作区  
lichen@lichen-host:~/Desktop/python1904$ ls  
a1.txt b1.txt

##### 5.3克隆版本库【先有远程版本库，再有本地版本库】

#步骤：  
a。创建远程工作区【勾选initialize】  
b.克隆远程仓库：git clone git@github.com:yangyang-git/python1904clone.git  
  
演示命令：  
lichen@lichen-host:~/Desktop/python1904$ cd ..  
lichen@lichen-host:~/Desktop$ git clone git@github.com:yangyang-git/python1904clone.git  
正克隆到 'python1904clone'...  
remote: Enumerating objects: 3, done.  
remote: Counting objects: 100% (3/3), done.  
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0  
接收对象中: 100% (3/3), 完成.  
lichen@lichen-host:~/Desktop$ ls  
python1904 python1904clone  
lichen@lichen-host:~/Desktop$ cd python1904clone/  
lichen@lichen-host:~/Desktop/python1904clone$ ls  
README.md  
lichen@lichen-host:~/Desktop/python1904clone$ touch file1.txt  
lichen@lichen-host:~/Desktop/python1904clone$ vim file1.txt   
lichen@lichen-host:~/Desktop/python1904clone$ git add file1.txt   
lichen@lichen-host:~/Desktop/python1904clone$ git commit -m "file1"  
[master 8bbbeb3] file1  
1 file changed, 1 insertion(+)  
create mode 100644 file1.txt  
lichen@lichen-host:~/Desktop/python1904clone$ git status  
位于分支 master  
您的分支领先 'origin/master' 共 1 个提交。  
（使用 "git push" 来发布您的本地提交）  
  
无文件要提交，干净的工作区  
lichen@lichen-host:~/Desktop/python1904clone$ git push origin master  
Warning: Permanently added the RSA host key for IP address '13.250.177.223' to the list of known hosts.  
枚举对象: 4, 完成.  
对象计数中: 100% (4/4), 完成.  
使用 8 个线程进行压缩  
压缩对象中: 100% (2/2), 完成.  
写入对象中: 100% (3/3), 277 字节 | 277.00 KiB/s, 完成.  
总共 3 （差异 0），复用 0 （差异 0）  
To github.com:yangyang-git/python1904clone.git  
e53d3c1..8bbbeb3 master -> master