

Linux下的用户和登录

- 前言：

Linux下Root是用户，也是管理员，当然我们的Linux可能要分配给多人进行作业，光有一个Root用户肯定是不行的。这一节，我们来创建多个Linux用户，并进行一些用户的实验以加深对Linux底层的理解。当然，别忘了，所有的子用户都归Root所管哦。

- 正文：

1.创建自己的用户：

①用useradd + 用户名：输入该命令，系统会自动创建一个用户，而且底层的配置都设置好了，无需操心。

②用id + 用户名：输入该命令，可以查看该用户的编号和所属的主组(gid)和副组(groups)。另外，whoami就会打印当前用户名称。

③用passwd + 用户名：给用户设置密码。第一次使用为设置密码，第二次后为更新密码。当我们在root用户下给其他用户设置密码时，即使密码简单，但是系统也会强制更新密码。但如果是进入了对应用户再passwd，要先确认原先密码，当密码不是数字、字母、符号等混合时，提示过于简单，并且更改失败。

④用su - 用户名：用于切换用户（是注销当前用户的），Ctrl+D可以快速切换回来。

⑤如Nginx和Ntp等账户，是系统账户，是无法登录的，因为它们的配置文件里是设置不可登录的，后面会再次说明。

2.深入理解Linux的用户：

①配置文件：Linux系统上运行着各种各样的软件，到了底层，就是程序在运行，而某个程序的运行，不是操作系统直接操控的，而是一些配置文件在控制。Linux的用户管理，例如登录、修改密码等，其实就是有对应的程序在运行。这样一来，我们可以通过更改配置文件，实现同样的操作。顺带一提，Linux的内核都是一样的，但配置文件不一样，也就表现出了不一样的功能。

②通过配置文件创建用户：如果我们在root用户中，我们返回到上一级，会看到如下的文件

```
[root@localhost ~]# ls
anaconda-ks.cfg  test.txt
[root@localhost ~]# cd ..
[root@localhost /]# ls
bin    dev    home   lib64  mnt    proc   run    srv    tmp    var
boot  etc    lib    media  opt    root   sbin   sys    usr
[root@localhost /]# cd
[root@localhost ~]# ls
anaconda-ks.cfg  test.txt
[root@localhost ~]# cd /
[root@localhost /]# ls
bin    dev    home   lib64  mnt    proc   run    srv    tmp    var
boot  etc    lib    media  opt    root   sbin   sys    usr
[root@localhost /]# _
```

root文件夹只是root用户独有的，home是存放其他用户的文件夹，而其他的文件夹呢，就是装系统的文件。我们要修改的配置文件在etc里。所以现在让我们查看一下/etc/passwd文件（顺带一提，cd /是返回到根目录的，如图，所以我们在root用户时，用vim /etc/passwd来打开passwd文件）。

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
systemd-network:x:192:192:systemd Network Management:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
polkitd:x:999:998:User for polkitd:/:/sbin/nologin
libstoragemgmt:x:998:997:daemon account for libstoragemgmt:/var/run/lsm:/sbin/nologin
abrt:x:173:173:/:etc/abrt:/sbin/nologin
rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/sbin/nologin
apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
postfix:x:89:89:/:var/spool/postfix:/sbin/nologin
chrony:x:997:995:/:var/lib/chrony:/sbin/nologin
ntp:x:38:38:/:etc/ntp:/sbin/nologin
tcpdump:x:72:72:/:/sbin/nologin
usertest:x:1000:1000:/:home/usertest:/bin/bash
linuxuser:x:1001:1001:/:home/linuxuser:/bin/bash
```

如图，在最后两行，我已经加了模仿第一行的格式创建了两个用户（usertest、linuxuser）。例如linuxuser用户，第一个数字“1001”指该用户的id，第二个“1001”指它的主组。后面的绿色路径不必多说，就是用户所在的目录了。至于后面的/bin/bash，这个可得好好说说了。

③关于/bin/bash这家伙：我们前面说了，Linux的内核不变的，而且我们其实不是直接控制它的，我们是通过一种手段对其进行控制的。我们称其为shell命令。我们敲打命令，传给shell，shell翻译给内核，内核再做出相应的回应。而其中bash是shell一种，也是最知名的一种，即我们一直在用的ls、cd那些命令。在上面的新建用户中，我们配置其在登录时使用bash命令，当然你也可以用别的，例如/bin/vi，用vi编辑器打开，然后登录该用户就会进入一个vi界面，没啥好操作的地方，随后退出vi编辑时，也自然地登出了该用户，大伙儿可自己试试。这样一来，想必你对Linux的开源可自行编辑的优点更深入体悟到了吧。

④继续改写配置文件使得用户真正意义上被创建：当我们进行了上述操作后，我们试试输入id root，再输入id linuxuser，然后比较两者输出的不同。没错，细心的你一定发现了，root的每个数字后面都有个括号，内有英文名字，linuxuser的gid和groups没有。这是为什么呢？很简单，因为我们前面改的配置文件，只是告诉了系统我们有这个用户，它是这么规定的——叫什么名，在哪个组。实际上，这个组并不存在，所以白叫了。我们需要通过vim /etc/group打开组的配置文件，在最后一行加上linuxuser:X:1001，这样我们就手动创建了一个组。先前我搞的那个usertest用户，是useradd创建的，所以啥都配置好了，不像这个linuxuser用户那么麻烦。另外，linuxuser的主组可以被更改为其他组，但绝不能没有空在那里。

⑤你以为这样就完了吗？并没有：有些机灵的小伙伴可能注意到了，前面我们说过用户都是用一个文件夹表示的，非root一般在home里，既然写配置文件建用户，组都不会自己补，更别提文件夹自动创建了。没错，配置文件只是登记用户而已，它连个家都没有。不信可以试试先su - linuxuser切到用户，然后whoami，pwd，你会发现我确实是linuxuser用户没错，但我所在的文件夹还是/root而不是/home/linuxuser。所以我们需要自建一个：切回root，然后mkdir /home/linuxuser创建专属文件夹，随后chown -R linuxuser:linuxuser /home/linuxuser/，把这个目录的权限修改一下。

⑥顺带一提，我们之前说了有些是系统用户，是禁止登录。在上图vim /etc/passwd中，后缀是/sbin/nologin就是标志不能登录的用户。

3.不切换用户快速地借助其他用户权限进行操作：

在实际工作中，这个还是蛮常见的，例如我在我的用户工作时，可能需要同伴的权限对某个他专属的文件进行操作，或者需要root的权限对系统进行更改时，都会需要在不切换用户的前提下暂时使用其他用户权限。

sudo就是用于切换的命令，也是个很常用的命令。sudo -u -linuxuser ls /bin/bash，这条命令意思是临时用linuxuser的身份来执行ls /bin/bash的命令。但是，当输入sudo -u linuxuser ls时，会显示拒绝访问（Permission denied），因为当前目录在root（确定你是用着root用户），linuxuser是没有权限访问root用户的。

然而在企业中，一般不是root临时用其他用户的，正好反过来，是其他用户临时借用root的权限，那我们就要考虑一个授权的问题了，总不可能一个子用户创建出来就默认能切换并使用root用户吧。为了给子用户授权，我们要接着改配置文件，visudo后，搜索Allow root，如下图所示：

```
# Adding HOME to env_keep may enable a user to run unrestricted
# commands via sudo.
#
# Defaults    env_keep += "HOME"
Defaults     secure_path = /sbin:/bin:/usr/sbin:/usr/bin

## Next comes the main part: which users can run what software on
## which machines (the sudoers file can be shared between multiple
## systems).
## Syntax:
##
##      user    MACHINE=COMMANDS
##
## The COMMANDS section may have other options added to it.
##

## Allow root to run any commands anywhere
root    ALL=(ALL)        ALL

linuxuser ALL=(ALL)      NOPASSWD: ALL

## Allows members of the 'sys' group to run networking, software,
## service management apps and more.
# %sys ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELEGATING, PROCESSES, LOCATE, DRIVERS

## Allows people in group wheel to run all commands
%wheel  ALL=(ALL)        ALL

## Same thing without a password
# %wheel    ALL=(ALL)      NOPASSWD: ALL

## Allows members of the users group to mount and unmount the
## cdrom as root
# %users    ALL=/sbin/mount /mnt/cdrom, /sbin/umount /mnt/cdrom

## Allows members of the users group to shutdown this system
```

可以看到，这里我已经授权了，你也来试试吧，别忘了NOPASSWD: ALL冒号后有个空格哦。这条配置指的是，linuxuser能随时随地sudo root并且不输入密码地使用。现在，你可以试试切换到linuxuser然后sudo -u root fdisk -l来查看只有root才能查看的硬盘信息了。

4.用户的密码管理：

可能有的小伙伴一直想问：我们的密码呢？为什么不在passwd里？其实，我们的密码在shadow里，不妨动手输入vim /etc/shadow，打开该配置文件：

