# **用户管理**

## 新增用户

|  |
| --- |
| 格式: useradd [-potions] username  参数:  -d 指定新建用户的主目录, 如果不指定, 则系统自动在 /home 目录创建  一个和用户名相同的文件夹作为新建用户的主目录（/home/username）    -m 自动创建主目录文件夹    -g 指定新建用户所在组名称, 如果不指定, 则系统自动创建一个和用户名  相同的组作为新建用户所属组  例子:  useradd -m user01 // 新建用户 user01, 自动创建主目录 /home/user01, 自动创建组 user01  useradd -md /home/u2 user02 // 新建用户 user02, 自动创建主目录 /home/u2, 自动创建组 user02  useradd -md /home/u3 -g g3 user03 // 新建用户 user03, 自动创建主目录 /home/u3, 指定组为 g3（指定的组必须已存在）  注意:  新创建的用户默认不能使用 sudo 命令, 需要另外添加该命令的权限, 实际上 sudo 是一个组的名称,  把 组sudo 追加到用户的附加组列表, 该用户便能在输入用户自己密码的情况下执行 sudo 命令。    例如让用户 user01 拥有 sudo 命令权限, 命令格式为: usermod -aG sudo user01 |

## 删除用户

|  |
| --- |
| 删除用户时, 如果用户所属组是创建用户时自动创建的和用户名称同名的组, 并且该组内没有其他用户, 则该组也会被删掉。  格式: userdel [-options] username  参数:  -f 强制删除用户, 即使用户当前已登录  -r 删除用户, 同时删除用户主目录  例子:  userdel user01 // 删除用户 user01, 保留用户主目录  userdel -r user02 // 删除用户 user02, 并同时删除用户主目录 |

## 修改用户

|  |
| --- |
| usermode 命令用于修改用户的基本信息。不允许修改正在线上的用户的名称, 不允许修改正在系统上执行程序的用户的ID。  格式: usermod [-options] username  参数:  -c comment, 设置帐户注释  -d home\_dir, 设置用户主目录  -e expiredate, 设置帐户过期日期  -g group, 强制修改用户的新主组  -G groups, 修改用户新的附加组列表  -a append group, 将用户追加至上边 -G 中提到的附加组中, 并不从其它组中删除此用户  -u uid, 修改账户的用户ID  -l login, 设置用户的登录名称  -s shell, 修改用户登录后使用的 shell  -L lock, 锁定用户帐户（等同于 passwd -l 命令锁定账户）  -U unlock, 解锁用户帐户（等同于 passwd -u 命令解账户）  例子:  usermod -g group01 user01 // 把用户 user01 的主组修改为 group01 组  usermod -u 9999 user01 // 把用户 user01 的 ID 修改为 9999  usermod -l u1 user01 // 把用户 user01 的名称修改为 u1    usermod -aG sudo user01 // 把 组sudo 添加到用户 user01 的附加组列表, 使该用户拥有执行 sudo 命令的权限 |

# 查询用户

|  |
| --- |
| 查询所有用户 cat /etc/passwd 查看某个用户的信息 id 用户名 |

# 组管理

### 增

|  |
| --- |
| 格式: groupadd [-options] groupname  参数:  -g 指定新建组的ID  例子:  groupadd group01 // 新建名称为 group01 的组  groupadd -g 1001 group02 // 新建名称为 group02 的组, 并给指定组的 ID 为 1001 |

删

|  |
| --- |
| 删除组, 如果组内还包含有用户, 需要将用户删除或移出该组后才能删除。否则提示"groupdel：不能移除用户“u1000”的主组"  格式: groupdel groupname |

改

|  |
| --- |
| 格式: groupmod [-options] groupname  参数:  -n 修改组的名称  例子:  groupmod -n NewGroupName OldGroupName |

查

|  |
| --- |
| 查看所有的组  cat /etc/group |

# 文件管理

增

|  |
| --- |
| touch 文件名  vim 文件名  mkdir [选项] 目录名  选项 -m 为目录指定访问权限，与chmod类似,例如 mkdir -m u=rwx,g=rwx,o=r-x bb  -p 若父目录不存在，则会强制性先创建父目录，在创建子目录 |

删

|  |
| --- |
| rm -rf 文件或者目录 |

改

|  |
| --- |
| 文件重命名或者移动 mv 原文件 新文件/目录 |

|  |
| --- |
| chmod修改文件权限有两种形式: 字母形式 和 数字形式  字母形式:  格式:  chmod [u/g/o/a][+/-/=][r/w/x] file/dir  参数:  u/g/o/a: 修改哪个组的权限, 分别表示 所有者(user), 所在组(group), 其他组(other), 所有(all)  +/-/=: 增加(+), 撤销(-) 或 重置(=) 权限  r/w/x: 需要操作的权限, 读(r)、写(w)、执行(x)  例子:  chmod u+x aa.sh // 给文件所有者添加执行权限  chmod a+x aa.sh // 给所有用户添加执行权限  chmod a=rwx aa.sh // 给所有用户添加读写和执行权限  chomd a= aa.sh // 撤销所有用户的所有权限, 相当于 a-rwx  chmod o-w aa.txt // 给其他组的用户撤销写权限  chmod u=rw,g=r,o= aa.sh // 可以用逗号分隔分别给不同权限组修改权限  数字形式:  权限有3位数字组成（注意: 每一位单独使用, 分别表示 所有者、所在组、其他组 的权限）,  读(r) = 4, 写(w) = 2, 执行(x) = 1, 没有(-) = 0    格式:  chmod [3位数字权限] file/dir    例子:  chmod 777 aa.sh // 给所有用户添加读写和执行权限  chmod 640 aa.sh // 所有者有读写权限, 所在组有读权限, 其他组没有任何权限 |

修改文件的所有者

|  |
| --- |
| 修改文件所有者(change owner)  格式: chown [-options] username file/dir  参数:  -R 递归处理, 如果修改的是目录, 则递归修改目录下的所有文件和子目录  例子:  chonw user01 aa.txt // 把文件 aa.txt 的所有者修改为 用户user01  chonw -R user01 bbDir // 递归修改 目录bbDir 的所有者为 用户user01 |

文件的复制

|  |
| --- |
| cp命令格式       cp （[选项]）源文件    目标文件  参数选项      -d    建立符号链接，并指向与源文件或目录链接的原址文件或目录                       -f     强行复制文件或目录，不论目标文件是否存在                        -i    覆盖既有文件之前会询问用户                        -l    只是建立链接，不是复制文件                        -p   保留源文件或目录的属性                        -r    递归复制                        -s    对源文件进行符号链接，不是复制文件                        -b    覆盖已存在的文件目标前将目标文件经行备份                        -u    只有在源文件更改才会复制 |