|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 多用户多任务管理之账户管理 | | | | |
| 计划学时 | |  | 授课时间 | 第 周 |
| **教学内容** | | | | |
| **1、账户管理**  **（1）账户管理概述**  **（1.1）账户实质**  Linux操作系统是多用户的操作系统，它允许多个用户同时登录到系统上，使用系统资原。当多个用户能同时使用系统时，为了使所有用户的工作都能顺利进行，保护每个用户的文件和进程，也为了系统自身的安全和稳定，必须建立起一种秩序，使每个用户的权限都能得到规范。为此，首先需要区分不同的用户，这样就产生了用户账户。  账户实质上就是一个用户在系统上的标识，系统依据账户ID来区分每个用户的文件、进程、任务，给每个用户提供特定的工作环境(如用户的工作目录、Shell 版本以及X Window环境的配置等)，使每个用户的工作都能独立不受干扰地进行。  **(1.2)用户和组**  广义上讲, Linux的账户包括用户账户和组账户两种。  Linux系统下的用户账户(简称用户)有两种，普通用户账户和超级用户账户(或管理员账户)。普通用户在系统上的任务是进行日常工作，管理员在系统上的任务是对普通用户和整个系统进行管理。管理员账户对系统具有绝对的控制权，能够对系统进行一切操作，如操作不当很容易对系统造成损坏。因此即使系统只有一个用户使用，也应该在管理员账户之外建立一个普通用户账户，在用户进行日常工作的时候以普通用户账户登录系统。  除了用户账户之外，在Linux 下还存在组账户(简称组)。组是用户的集合。在RHEL/CentOS中组有两种类型：私有组和标准组。当创建一个新用户时，若没有指定其所属的组，RHEL/CentOS 就建立一个和该用户同名的私有组，此私有组中只包含这个用户自己。  标准组可以包含多个用户，若使用标准组，在创建一个新的用户时就应该指定其所属的组。  从另一方面讲，同一个用户可以同属于多个组，如某单位有领导组和技术组等，Tom是该单位的技术主管，所以他既应该属于领导组又应该属于技术组。当一个用户属于多个组时，其登录后所属的组称为主组，其他的组称为附加组。  **(1.3) Linux 环境下的账户系统文件**  Linux下的账户系统文件主要有/etc/passwd、/etc/shadow、/etc/group和/etc/gshadow 4个文件。  **1)**/etc/passwd文件中每行定义一个用户账户，一行中又划分为多个字段定义用户账号的不同属性，各字段间用“:”分隔。例如：  **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**  **……（此处省略n行）**  下表中描述了这些字段的意义。   |  |  | | --- | --- | | /etc/passwd文件中各字段的含义 | | | 字 段 | 说 明 | | 用户名 | 这是用户登录系统时使用的用户名，在系统中是唯一的 | | 口令 | 此字段存放加密的口令。在此文件中的口令是X,表示用户的口令是被/etc/shadow文件保护的，所有加密的口令以及和口令有关的设置都保存在/etc/shadow中 | | 用户标识号 | 是一个整数，系统内部用它来标识用户。每个用户的UID都是唯一的。root用户的UID是0，从1~499是系统的标准账户。普通用户的UID从500开始 | | 组标识号 | 是一个整数，系统内部用它来标识用户所属的组。每个用户账户在建立好后都会有一个主组。主组相同的账户其GID相同。在默认情况下，每一个账户建立好后系统都会建立一个和账户名同名的组，作为该账户的主组，这个组只有用户本人这一个成员，即此组是私有组 | | GECOS | 例如存放用户全名、地理位置等信息 | | 宿主目录 | 用户登录系统后所进入的目录 | | 命令解释器 | 指示该用户使用的Shell，默认为bash |   **2)**/etc/passwd文件对任何用户均可读，为了增加系统的安全性，CentOS默认使用shadow passwords保护，将经过加密的用户口令保存在/etc/shadow文件里，它只对root用户可读且提供了一些口令时效字段。/etc/shadow文件的内容形式如下：  **root:$6$6OWB.8Dhhw179jID$syizd99UMbV2z.CWhR3Z52CzHUhtXRo51IizZgAtaRqTUuNYlCszISFyLX0BizXEbT6L3i5ybwnnXxeY9cJLQ0::0:99999:7:::**  **bin:\*:17110:0:99999:7:::**  **daemon:\*:17110:0:99999:7:::**  **adm:\*:17110:0:99999:7:::**  **lp:\*:17110:0:99999:7:::**  **sync:\*:17110:0:99999:7:::**  **shutdown:\*:17110:0:99999:7:::**  **halt:\*:17110:0:99999:7:::**  **……（此处省略n行）**  其中各字段的意义如下表所示。     |  |  | | --- | --- | | /etcsbadow 文件中各字段的含义 | | | 栏 位 | 说 明 | | 用户名1 | 用户的账户名 | | 口令 | 用户的口令，是SHA512加密过的 | | 最后一次修改的时间 | 从1970年1月1日起，到用户最后一次更改口令的天数 | | 最小时间间隔 | 从1970年1月1日起，到用户可以更改口令的天数 | | 最大时间间隔 | 从1970年1月1日起，到用户必须更改口令的天数 | | 警告时间 | 在用户口令过期之前多少天提醒用户更新 | | 不活动时间 | 在用户口令过期之后到集用账户的天数 | | 失效时间 | 从1970年1月1日起，到账户被禁用的天数 | | 标志 | 保留位 |   **3)** /et/group文件，将用户分组是Linux中对用户进行管理及控制访问权限的一种手段。每个用户都属于某一个组；一个组中可以有多个用户，一个用户也可以属于不同的组。当一个用户同时是多个组的成员时，在/etc/passwd文件中记录的是用户所属的主组，也就是登录时所属的主组，而其他组称为附加组。用户要访问附加组的文件时，必须首先使用newgrp 命令使自己成为所要访问的组的成员。组的所有属性都存放在/etc/group文件中。etc/group 文件对任何用户均可读。下面是一个/etc/group文件的例子。  **root:x:0:**  **bin:x:1:**  **daemon:x:2:**  **sys:x:3:**  **adm:x:4:**  **……（此处省略n行）**  与/etc/passwd文件类似，其中每一行记录了一个组的信息。每行包括4个字段，不同字段之间用冒号隔开。各字段的内容说明如下表所示。   |  |  | | --- | --- | | /etc/group文件中各字段说明 | | | 栏 位 | 说 明 | | 组名 | 该组的名称 | | 组口令 | 组口令，由于安全性原因，已不使用该字段保存口令，用x占位 | | GID | 组的识别号，和UID类似，每个组都有自己独有的识别号，不同组的GID不会相同 | | 组成员 | 属于这个组的成员，多个成员间用”,”分隔 |     **4)**/etc/gshadow文件用于定义用户组口令、组管理员等信息，该文件只有root用户可以读取。下面是一个/etc/gshadow文件的例子：  **root:::**  **bin:::**  **daemon:::**  **sys:::**  **adm:::**  **tty:::**  **disk:::**  **lp:::**  **mem:::**  **kmem:::**  **wheel:::localhost**  **cdrom:::**  **……（此处省略n行）**  与/etc/group文件类似，其中每一行记录了一个组的信息。每行包括4个字段，不同字段之间用冒号隔开。其中各字段的内容说明如下表所示。   |  |  | | --- | --- | | /etc/gshadow文件中各字段说明 | | | 栏 位 | 说 明 | | 组名 | 组名称，该字段与group文件中的组名称对应 | | 组口令 | 组口令，该字段用于保存已加密的口令 | | 组的管理员账号 | 组的管理负账号，管理员有权对该组添加、删除账号 | | 组成员 | 属于该组的用户成员列表，列表中多个用户间用“,”分隔 |   **注意：**  账户管理的实质就是管理上述的4个账户系统文件，可以使用图形界面工具进行，也可以使用命令行工具进行，甚至还可以使用Web工具进行。  **（2）使用命令行工具管理账户**  **1）管理账户的命令行工具**  下表中列出了管理账户的命令行工具的功能。   |  |  | | --- | --- | | **管理账户的命令行工具的功能** | | | 命 令 | 说 明 | | usernadd[<选项>]<用户名> | 添加新的用户 | | usermod[<选项>]<川户> | 修改已经存在的指定用户 | | userdel[-r]<用户名> | 删除已经存在的指定用户，-r参数用于删除用户自家目录 | | groupadd[<选项>]<组名> | 添加新的组 | | groupmod[<选项>]<组名> | 修改已经存在的指定组 | | groupdel<组名> | 删除已经存在的指定组 |   **2)命令行账户管理工具使用举例**  **操作步骤3.1命令行账户管理工具使用举例**  //创建一个新用户rzpt  # useradd rzpt  //创建一个新组staff  # groupadd staff  //创建一个新用户tom,同时加入staff附加组中  # useradd -G staff tom  //创建一个新用户webmaster，指定登录目录/www，不创建用户宿主目录(-M)  # mkdir /www；useradd -d /www -M webmaster  //将rzpt添加到附加组staff中  # usermod -G staff rzpt  //将rzpt用户修改为rizhao，jjh组修改为rizhao，宿主目录改为/home/rizhao  # usermod -l rizhao -d /home/rizhao -m rzpt  # groupmod -n rizhao jjh  //删除用户webmaster  # userdel webmaster  //删除用户rizhao，同时删除其自家目录  # userdel -r rizhao  //删除组staff  #groupdel staff  **(3)口令管理和口令时效**  **1)使用passwd命令进行口令管理**  **创建用户账户之后，还要给新用户设置口令。为此需使用命令passwd，其格式是:**  **passwd [<选项>][<登录用户名>]**  常用选项如下表所示。   |  |  |  |  | | --- | --- | --- | --- | | passwd 命令的常用选项 | | | | | 选项 | 说 明 | 选项 | 说 明 | | -S | 列出口令的状态信息 | -d | 删除口令 | | -l | 锁定用户账户 | -k | 保持口令不变，直到口令过期失效后方能更改 | | -u | 解除已锁定账户 | -stdin | 从标准输入读取口令(菲交互模式) |   **注意：**  (1)在输入口令时，屏幕上不会回显。口令的选取至少用8个字符，最好大小写字母和数字及特殊字符搭配使用，尽量不要用英文单词作为口令。  (2)只有管理员账户(root) 可以更改其他用户的口令，普通用户只能更改 自己的口令，且在更改口令之前，系统会要求用户输入旧的口令。  下面给出几个passwd命令的使用示例。  操作步骤3.2 passsd命令的使用示例  //1.创建新用户jason，显示口令状态，为其设置口令  # useradd jason  # passwd -S jason  jason LK 2020-03-13 0 99999 7 -1 (Password locked.)  # passwd jason  Changing password for user jason.  New password:  Retype new password:  passwd: all authentication tokens updated successfully.  //2.用户jason要更改自己的口令  #su- jason //切换到jason 用户  $ passwd  Changing password for user jason.  Changing password for jason.  (current) UNIX password:  New password:  Retype new password:  passwd: all authentication tokens updated successfully.  $exit //返回root的Shell  logout  #  //3.超级用户可以使用如下命令进行用户口令管理  # passwd -S jason //显示口令状态  jason PS 2020-03-13 0 99999 7 -1 (Password set, SHA512 crypt.)  # passwd -l jason //锁定用户jason  Locking password for user jason.  passwd: Success  # passwd -S jason //显示口令状态  jason LK 2020-03 -13 0 99999 7 -1 (Password locked.)  # passwd -u jason //解除对用户jason的锁定  Unlocking password for user jason.  passwd: Success.  # passwd -S jason //显示口令状态:  jason PS 2020-03 -13 0 99999 7-1 (Password set, SHA512 crypt)  # passwd -d jason //清空jason的口令  Removing password for user jason.  passwd: Success  # passwd -S jason //显示口令状态:  jason NP 2020-03-13 0 99999 7 -1 (Empty password.)  # passwd jason //重新设置用户jason的口令  Changing password for user jason.  New UNIX password:  Retype new UNIX password:  passwd: all authentication tokens updated successfully.  # passwd -S jason //显示口令状态  jason PS 2020-03-13 0 99999 7 -1 (Password set, SHA512 crypt)  #  **2)chage命令**  硬件计算能力如今已经非常强大，这大大地缩短了利用自动运行的程序来猜测口令的时间。口令时效是系统管理员用来防止机构内不良口令的一种技术。防止口令被攻击的方法就是经常改变口令。为安全起见，要求用户定期改变用户口令是明智之举。  口令时效意味着过了一段预先设定的时间后，用户会被提示创建一个新口令。它所根据的理论是，如果用户被强制定期改变口令，某个口令的破译对入侵者来说就只有有限的利用机会。这种用来强制用户在一段时间之后更改口令的机制称为口令时效。  **在Linux系统上，修改口令时效分为:**   * 对于未来要创建的所有用户的口令时效需修改/etc/login.defs文件的相关参数。 * 对已存在的用户修改口令时效是通过chage命令来管理。   (1)设置新添用户的口令时效  编辑/etc/login.defs，通过指定下表中描述的几个参数来设置口令实效的默认设定。   |  |  | | --- | --- | | **/etc/login.defs中与口令时效相关的参数** | | | 参 数 | 说 明 | | PASS\_MAX\_DAYS | 设定在多少天后要求用户修改口令。默认口令时效的天数为99999，即关闭了口令时效。明智的设定一般是60天(2个月)强制更改一次口令 | | PASS\_MIN\_DAYS | 设定在本次口令修改后，至少要经过多少天后才允许更改口令 | | PASS\_WARN\_AGE | 设定在口令失效前多少天开始通知用户更改口令(一般在用户刚刚登录系统时就会收到警告通知) |   (2)设置已存在用户的口令时效  chage命令的格式是:  chage [<选项>] <用户登录名>  下表中列出了chage命令的选项说明。   |  |  | | --- | --- | | chage命令的选项说明 | | | 选项 | 描 述 | | -m days | 指定用户必须改变口令所间隔的最少天数。如果值为0,口令就不会过期(PASS\_MIN\_DAYS) | | -M days | 指定口令有效的最多天数。当该选项指定的天数加上-d选项指定的天数小于当前的日期时，用户在使用该账号前就必须改变口令(PASS\_MAX\_DAYS) | | -d days | 指定自从1970年1月1日起，口令被改变的天数 | | -I days | 指定口令过期后，账号被锁前不活跃的天数。如果值为0,账号在口令过期后就不会被锁 | | -E date | 指定账号被锁的日期，日期格式为YYYY-MM-DD。若不用日期，也可以使用自1970年1月I日后约过的天数 | | -W days | 指定口令过期前要警告用户的天数(PASS\_WARN\_AGE) | | -l | 列出指定用户当前的口令时效信息，以确定账号何时过期 |   下面给出几个使用chage命令的例子。  操作步骤3.3 使用chage命令的例子  //1.使用户下次登录之后修改口令  # chage d 0 jason  //2.用户jason两天内不能更改口令，并且口令最长的存活期为30天  //并在口令过期前5天通知jason  # chage -m2 -M 30 \_W 5 jason  //3.查看用户jason当前的口令时效信息  # chage -l jason  Last password change : password must be changed  Password expires : password must be changed  Password inactive : password must be changed  Account expires : never  Minimum number of days between password change : 2  Maximum number of days between password change : 30  Number of days of warning before password expires : 5  **注意：**  **1.也可以使用chage <用户名>进入交互模式修改用户的口令时效。**  **2.使用chage命令实质上是修改影子口令文件/etc/shadow中的与口令时效相关的字段值。**  **3. chage命令仅适用于本地系统账户，对LDAP账户和数据库账户不起作用。**  **制订一项策略，定义多长时间后一个口令必须进行更改，然后强制执行该策略，是非常不错的一个做法。在解雇了某个雇员后，口令时效策略会保证该雇员不可能在被解雇3个月后发现他的口令依然可用。即使系统管理员忘了删除该雇员的账号，该账号也会因口令时效策略而被自动锁定。当然，这一点并不能成为不及时删除该雇员账号的理由，但是这个策略的确提供了一层额外的安全防护，尤其是在过去常常忽视及时清理账号的情况下。**  **(4)用户和组状态**  **1) 用户和组状态命令**  下表中列出了一些常用的用户和组状态命令。   |  |  | | --- | --- | | 常用的用户和组状态命令 | | | 命 令 | 功 能 | | whoami | 用于显示当前用户的名称 | | id | 用于显示用户身份 | | groups | 用于显示指定用户所属的组 | | newgrp | 用于将用户从当前组转换到指定的附加维，用户必须属于该组才可以使用 |   **2)用户和组状态命令举例**  下面给出用户和组状态命令的使用示例。  操作步骤3.4 用户和组状态命令的示例  //创建一个新组staff  [root@localhost ~]# groupadd -g 3001 staff  //将用户crq加入staff附加组，并为其设置口令  [root@localhost ~]# usermod G staff crq  //显示当前用户的名称  [root@localhost ~]# whoami  root  //显示当前用户所属的组  [root@localhost ~]# groups  root  //显示指定用户所属的组  [root@localhost ~]# groups crq  crq : crq staff  //显示用户当前的uid、gid和用户所属的组列表  [root@localhost ~]# id  uid=0(root) gid 0(root) groups 0(root)  //切换当前用户到crq (超级用户切换到普通用户无须口令)，同时切换用户工作环境(-)  [root@localhost ~]# su- crq  [crq@localhost ~]$  //显示用户当前的uid、gid和用户所属的组列表  (crq@localhost ~]$ id  uid=1005(crq) gid=3002(crq) groups=3002(crq),3001(staff)  //创建一个新文件，并查看其用户和组  [crq@localhost ~]$ touch abc  [crq@localhost ~J$ ll abc  -rw-rw-r--. 1 crq crq 0 Mar 13 19:50 abc  //切换用户的当前组到指定的附加组staff  [crq@localhost ~]$ newgrp staff  //显示用户当前的uid、gid和用户所属的组列表  [crq@localhost ~]$ id  uid=1005(crq) gid=3001(staff) groups=3001(staff),3002(crq)  //创建一个新文件，并查看其用户和组(比较abc和xyz的组)  [crq@localhost ~]$ touch xyz  [crq@localhost ~]$ ll  total 0  -rw-rw-r--. 1 crq crq 0 Mar 13 19:50 abc  -rw-r--r--. 1 crq staff 0 Mar 13 19:56 xyz  //返回上一次crq的登录  [crq@localhost ~]$ exit  exit  //返回上一次root的登录  [crq@localhost ~]$ exit  logout  [root@localhost ~]#  **2、权限管理**  **(1)操作权限概述**  **(1.1)操作权限简介**  Linux是多用户的操作系统，允许多个用户同时在系统上登录和工作。为了确保系统和用户的安全，Linux采取了很多的安全措施。用户在登入系统时需要输入用户名和口令，这样，就使系统可以通过用户的识别号(UID)来分别确定每个用户在登录系统后都做了些什么，也可以用来区别不同用户所建立的文件或目录。  普通用户在系统上受到权限的制约，一个普通用户若要切换至其他的普通用户甚至超级用户的工作目录，将会收到拒绝提示信息，例如:  $ whoami  $ osmond  $ cd /root  bash: /root：Permission denied  $ cd /home/jason  bash: /home/jason: Permission denied  $  **(1.2)3种基本的权限**  在Linux中，将使用系统资源的人员分为4类:超级用户、文件或目录的属主、属主的同 组人和世界上的其他人员。由于超级用户具有操作Linux系统的一切权限，所以不用指定超级 用户对文件和目录的访问权限。对于其他3类用户都要指定对文件和目录的访问权限，对每一类用户都有如下表所示的3种基本的权限需要说明。   |  |  |  |  | | --- | --- | --- | --- | | 文件或目录的3种基本的访问权限 | | | | | 代表字符 | 权 限 | 对文件的含义 | 对目录的含义 | | r | 读权限 | 可以读文件的内容 | 可以列出目录中的文件列表 | | w | 写权限 | 可以修改该文件 | 可以在该目录中创建、删除文件 | | x | 执行权限 | 可以执行该文件 | 可以使用cd命令进入该目录 |   **注意：**  1.目录上只有执行权限，表示可以进入或穿越该目录进入更深层次的子目录。  2.目录上只有执行权限，要访问该目录下的有读权限的文件，必须知道文件名。  3.目录上只有执行权限，不能列出目录列表也不能删除该目录。  4.目录上执行权限和读权限的组合，表示可以进入目录并列出目录列表。  5.目录上执行权限和写权限的组合，表示可以在目录中创建、删除和重命名文件。  **(1.3)查看文件和目录的权限**  在Linux中通过给3类用户分配3种基本权限，就产生了文件或目录的9个基本权限位。可以使用带l参数的ls命令查看文件或目录的权限，例如:  -rw-------. 1 root root 1549 Feb 1 11:42 anaconda-ks.cfg  -rw-r--r--. 1 root root 0 Mar 12 17:16 cd.play.list  -rw-r--r--. 1 root root 16 Mar 12 17:25 err\_file  -rw-r--r--. 1 root root 10 Mar 9 14:30 file2~  每一行显示一个文件或目录的信息，这些信息包括文件的类型、文件的权限、文件的属主(第3列)、文件的所属组(第4列)，还有文件的大小以及创建时间和文件名。  输出列表中每一行的第一列的第一个字母指示了该文件的类型。第一列的其余9个字母可分成3组，3个字母组。这3组分别代表文件属主的权限、文件所属组的权限、其他用户的权限。每组中的3个栏位分别表示了读取权限(r)、写入权限(w)、执行权限(x)或没有相应的权限(-)。  **注意：**  当某用户访问系统中的某文件时:  (1)若访问者的UID与文件的UID匹配，就应用用户(user) 权限。  (2)否则，若访问者的GID与文件的GID匹配，就应用群组(group)权限。  (3)如果都不匹配，就应用其他用户(other) 权限。  通常将由ls -l命令输出的第一列称为文件或目录的权限字符串。下表中列出了几个权限字符串的说明。   |  |  |  | | --- | --- | --- | | 权限字符串举例 | | | | 字 符 | 数 值 | 说 明 | | -rw------- | 600 | 只有属主才有读取和写入的权限 | | -rw-r--r-- | 644 | 只有属主才有读取和写入的权限。同组人和其他人只有读取的权限 | | -rwx------ | 700 | 只有属主才有读取、写入和执行的权限 | | -rwxr-xr-x | 755 | 属主有读取、写入和执行的权限。同组人和其他人只有读取和执行的权限 | | -rwx--x--x | 711 | 属主有读取、写入和执行权限，同组人和其他人只有执行权限 | | -rw-rw-rw- | 666 | 每个人都能够读取和写入文件 | | -rwxrwxrwx | 777 | 每个人都能够读取、写入和执行 | | drwx------ | 700 | 只有属主能在目录中读取、写入 | | drwxr-xr-x | 755 | 每个人都能够读取目录，但是其中的内容却只能被属主改变 |   **注意:**  把权限设为666 会允许每个人对文件或目录都有读取和写入的权限。把权限设为777允许每个人都有读取、写入和执行的权限。这些权限可能会允许对机密文件的篡改，因此，一般来说，使用这类设置是不明智的。  **(2)更改操作权限**  系统管理员和文件的属主可以根据需要来更改文件的权限。更改文件和目录的操作权限 使用chmod命令进行，有两种设置方法:文字设定法和数值设定法。  **(2.1)文字设定法**  chmod命令的文字设定法格式如下:  chmod [ugoa][+-=][rwxugo] <文件名或目录名>  其中第1个选项表示要赋予权限的用户，具体说明如下表所示。   |  |  |  |  | | --- | --- | --- | --- | | 用户选项说明 | | | | | 选项 | 说 明 | 选项 | 说 明 | | u | 表示属主(user) | o | 表示其他用户(other) | | g | 表示所属组用户(group) | a | 表示所有用户(all) |   第2个选项表示要进行的操作，具体说明如下表所示。   |  |  |  |  | | --- | --- | --- | --- | | 权限操作选项说明 | | | | | 选项 | 说明 | 选项 | 说明 | | + | 增加权限 | = | 分配权限。同时将原有权限删除 | | - | 删除权限 | —— | —— |   第3个选项是要分配的权限，具体说明如下表所示。   |  |  |  |  | | --- | --- | --- | --- | | 分配权限选项说明 | | | | | 选项 | 说明 | 选项 | 说明 | | r | 允许读取 | u | 和属主的权限相同 | | w | 允许写入 | g | 和所属组用户的权限相同 | | x | 允许执行 | o | 和其他用户的权限相同 |   操作步骤3.5 chmod 命令的文字设定法举例  //取消组用户和其他用户对文件users的读取权限  #cd;ls -l users  -rw-r--r--. 1 root root 0 Mar 13 21:16 users  # chmod go-r users  #ll users  -rw-------. 1 root root 0 Mar 13 21:16 users  //对文件addusers的属主添加执行权限  #  -rw-r--r--. 1 root root 0 Mar 13 21:44 addusers  # chmod u+x addusers  # ll addusers  -rwxr--r--. 1 root root 0 Mar 13 21:44 addusers  #  //对文件addusers1的属主添加执行权限  //同时取消组用户和其他用户对文件的读取权限  # ll addusers  -rw-r--r--. 1 root root 0 Mar 13 21:46 addusers1  # chmod u+x,go-r addusersl  # ll addusersl  -rwx------. 1 root root 0 Mar 13 21:46 addusers1  **(2.2)数值设定法**  chmod命令的数值设定法格式如下:  chmod n1n2n3 <文件名或目录名>  其中，n1代表属主的权限，n2代表组用户的权限，n3代表其他用户的权限，这3个选项都是八进制数字，其意义如下表所示。   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | 权限字符说明 | | | | | | | 权 限 | | | 数值表示 | | 说 明 | | 读 | 写 | 执行 | 二进制 | 八进制 | | - | - | - | 000 | 0 | 没有权限 | | - | - | x | 001 | 1 | 允许执行 | | - | w | - | 010 | 2 | 允许写入 | | - | w | x | 011 | 3 | 允许执行和写入 | | r | - | - | 100 | 4 | 允许读取 | | r | - | x | 101 | 5 | 允许执行和读取 | | r | w | - | 110 | 6 | 允许写入和读取 | | r | w | x | 111 | 7 | 允许执行、写入和读取 |   操作步骤3.6 chmod 命令的数值设定法举例  //对文件addusers的属主设置可读、写和执行权限  //所属组用户和其他用户只设置读和执行的权限，没有写的权限  #ll addusers  -rwxr--r--. 1 root root 0 Mar 13 21:44 addusers  # chmod 755 addusers  # ll addusers  -rwxr-xr-x. 1 root root 0 Mar 13 21:44 addusers  #  //取消组用户和其他用户对文件user1的一切权限  # ll userl  -rw-r--r--. 1 root root 0 Mar 13 21:59 user1  # chmod 600 userl  # ll userl  -rw-------. 1 root root 0 Mar 13 21:59 user1  **(3)更改属主和同组人**  管理员有时还需要更改文件的属主和所属的组。除了root 用户之外，只有文件的属主才有权更改其属主和所属组，即用户可以把属于自己的文件转让给他人。改变文件的属主和组可  以用chown命令，命令格式如下:  chown [-R] <用户[:组]> <文件或目录>  **操作步骤3.7** chown 命令使用举例  //将文件users1的属主改成osmond  # chown osmond usersl  //将文件users的属主和组都改成osmond  # chown osmond:osmond users  //将mydir目录及其子目录下的所有文件或目录的属主和组都改成osmond  # chown -R osmond:osmond mydir  **(4)设置文件和目录的生成掩码**  用户可以使用umask命令设置文件的默认生成掩码。默认的生成掩码告诉系统当创建一个文件或目录时不应该赋予其哪些权限。如果用户将umask 命令放在环境文件(.bash profile) 中，就可以控制所有新建的文件或目录的访问权限。  umask命令的格式如下:  umask [-S] [u1u2u3]  其中，u1表示的是不允许属主有的权限； u2表示的是不允许同组人有的权限； u3 表示的是不允许其他人有的权限。  **操作步骤3.8** umask 命令使用举例  //1.查看当前用户的文件默认生成掩码  $ umask  0022  $ umask -S  u=rwx,g=rx,o=rx  //下面显示了在默认的文件生成掩码为022的情况下(创建文件和目录的权限情况)  $ touch testfilel  $ ll testfilel  $ mkdir testdir2  $ ll -d testdir2  //2.设置当前用户的文件默认生成掩码(设置允许同组用户有写权限)  $ umask 002  $ touch tstfile3  $ ll testfile3 | | | | |
| 作业 |  | | | |