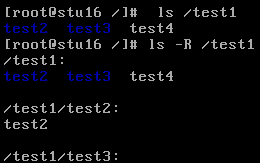
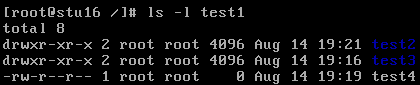
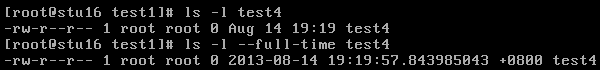
<https://blog.csdn.net/wangzhenyu177/article/details/52080031>

# 1 .命令ls

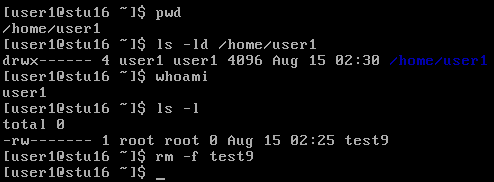
**命令ls**，用于列出（list）各文件。   
默认仅显示非隐藏文件文件名。   


各选项：

* 选项“**-a**”：显示包括隐藏文件在内的所有文件名。   
  + 文件名第一个字符为“.”的即为隐藏文件。
* 选项“**-d**”：若参数为目录[1](https://blog.csdn.net/wangzhenyu177/article/details/52080031#fn:2)，该选项表示仅列出目录本身，而非目录内的文件。
* 选项“**-R**”：若参数为目录，该选项表示连同子目录及其下文件都列出。   
  
* 选项“**-l**”：列出详细的文件和属性。   
     
  各字段以空格分隔，各字段意义：   
  + 共10个字符。第一个字符表示文件类型[2](https://blog.csdn.net/wangzhenyu177/article/details/52080031#fn:1)；后9个字符，每3个一组分别表示文件属主（u）、属组（g）、其他用户（o）对于此文件是否有读（r）、写（w）、执行（x）权限。
  + 该文件被硬链接的次数。
  + 文件属主。
  + 文件属组。
  + 文件大小，默认单位为B。[3](https://blog.csdn.net/wangzhenyu177/article/details/52080031" \l "fn:14" \o "See footnote" \t "_blank)
  + 文件内容最近一次被修改的时间，即mtime。
  + 文件名。
* 选项“**-h**”：与选项“-l”连用，使显示的文件大小使用人类（human）常用单位，如KB、MB等，而非B。
* 选项“**–time={atime,ctime}**”：与选项“-l”连用，输出文件最近访问时间（atime）或属性改变时间（ctime），而非内容修改时间（mtime）。
* 选项“**–full-time**”：与选项“-l”连用。选项“-l”输出的时间默认显示月、日、时间；若距离现在过久，则仅显示年月。该选项表示显示完整时间。   
  

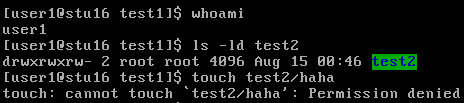
# 2 .文件权限

## 2.1 读、写、执行权限的意义

* 对于普通文件   
  + **r**：表示读取文件的内容的权限。   
    - 例如读取一个文本文件的文字；
    - 对某文件执行**复制操作**时，需用户对该文件有“r”权限[4](https://blog.csdn.net/wangzhenyu177/article/details/52080031#fn:3)。
  + **w**：表示编辑修改文件内容的权限（**但不含删除该文件**）。
  + **x**：表示文件是否可以被执行。   
    - windows可通过文件扩展名判断文件是否可以被执行，而linux的文件名并无特别意义，文件是否可执行完全看是否有此权限。
* 对于目录文件
  + **r**：表示读取目录结构列表的权限，即可使用命令“ls”查看目录中的文件名列表。   
    - 目录的主要内容就是记录**文件名列表**。
  + **w**：表示更改目录结构列表的权限，即：
    - 可在该目录创建新的目录或文件；
    - 可在该目录删除已存在的目录或文件，不论被删除的文件权限为何；   
      
    - 可在该目录重命名已存在的目录和文件；
    - 可在该目录剪切已存在的目录和文件等。

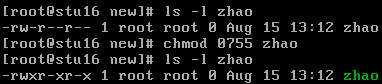
综上，“w”权限对目录来讲可理解为，变动该目录下的文件名列表的权限。

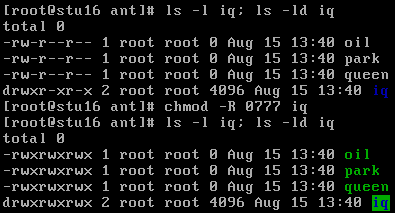
* + **x**：表示能否切换到该目录作为当前工作目录，即：
    - 可使用命令“cd”至该目录；
    - **可读取、执行该目录下的文件**（如果文件本身开放读、执行权限的话），若目录无“x”权限，则无法读取、执行该目录下的文件；
    - 可使用“ls -l”查看该目录下各文件属性等。

注：   
1、目录开放“x”权限时，一般也要开放“r”权限。否则虽然可以“cd”至该目录，但不能“ls”（所以更不能“ls -l”）。   
2、用户在某目录下创建、删除、剪切文件（这些操作属于该目录的“w”权限），或对该目录下的文件进行复制等操作时，若**仅有“w”**权限**而无“x”**权限，则**仍会提示无权限操作**，需同时对该目录有“x”权限才可[5](https://blog.csdn.net/wangzhenyu177/article/details/52080031#fn:4)。   
   
从上图也可看出，若开放目录的“w”权限给“其他用户（others）”，系统会为目录名着背景色。

## 2.2 读、写、执行权限的修改

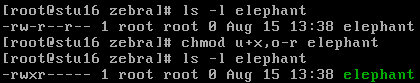
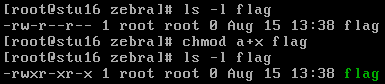
**命令chmod**，用于修改文件权限。**仅root和文件属主可修改文件权限**，两种方式：

* 使用8进制数修改   
  文件的读、写、执行权限共9位（每位取值0或1），每3位一组分别代表属主（u）、属组用户（g）、其他用户（o）的读、写、执行权限，共3组。每组取值范围是0到7（- - - 到rwx），可视作3个8进制数。所以修改的权限可用3个8进制数表示：   
  

   
如上，可使用3位8进制数来修改属主、属组用户、其他用户（上图显示的是4位8进制数，第1位是用来修改特殊权限的，如无需修改特殊权限，可省略此位）。使用**“-R”选项**可一次修改目录及其下所有文件的权限。

* 使用符号修改

|  | **修改谁的权限** | **添加、减去、指定** | **权限** |  |
| --- | --- | --- | --- | --- |
| chmod | u、g、o、a | +、-、= | r、w、x | 文件 |

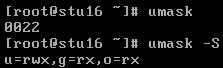
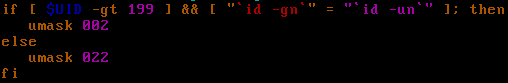
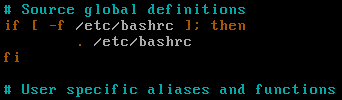
* 如上所示，可修改属主（u）、属组用户（g）、其他用户（o）或全部（a，包含ugo）用户的“rwx”权限，修改方式包括添加、删除、指定。
* 例如使某文件的属主权限添加“x”权限，其他用户去掉“r”权限：   
     
  没有指定改变属组用户的权限，故属组用户权限不会变化。
* 又如，不改变其他权限，使所有人都有文件的“x”权限：   
  

## 2.3 文件默认权限

默认权限计算

* 普通文件常常是为了记录数据，且为安全起见，不随便开放“x”权限，所以默认最大为开放读写（rw-rw-rw-），即“666”；
* 目录则需开放“x”以便切换至该目录，默认最大是“777”；
* 创建一个新目录或普通文件，其默认权限为上述的最大默认权限减去遮罩码。   
  + 比如遮罩码设置为0022（第一位用于特殊权限，下述），则新建文件即为“rw-rw-rw-”减去“- - - -w - - w -”等于“rw- r- -r- -”；新建目录用同样方法计算可得默认权限为“rwxr-xr-x”。

**命令umask**，用于显示、设置遮罩码（仅在当前shell生效，）

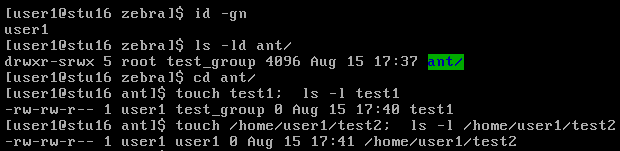
* 显示遮罩码，选项“-S”表示以符号（Symbolic）方式显示   
  
* 设置遮罩码   
  这里写图片描述   
  + 使用命令设置的遮罩码仅在当前shell生效。永久生效的是配置文件，/etc/bashrc中使用了该命令设置了各用户的默认遮罩码。根据这个设置可知，root的默认遮罩码为0022；普通用户的遮罩码为0002。
  + 如需使设置的遮罩码永久生效，一般也不在/etc/bashrc中直接修改（且普通用户无该文件“w”权限）。可在bashshell的环境参数配置文件“~/.bashrc”中设置，可以看到该文件的内容也就是读取了文件/etc/bashrc。   
    

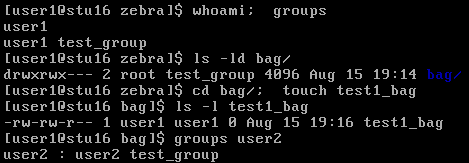
## 2.4 文件特殊权限及修改

* SUID权限   
  Set UID，简称SUID。该权限**仅对二进制文件有效**[6](https://blog.csdn.net/wangzhenyu177/article/details/52080031#fn:9)，且仅在该文件**执行过程中生效**。当文件具有SUID权限时，文件属主的“x”权限位置变为“s”。

效果：若文件具有SUID权限，且执行者对于该文件有“x”权限，当文件被其执行时，执行者将具有**该文件属主的权限**。

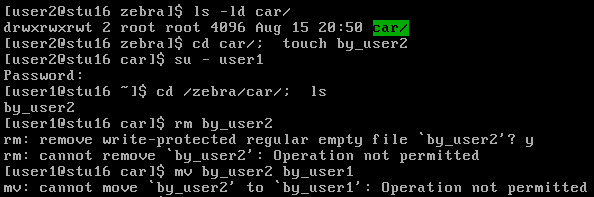
以命令“passwd”为例，它具有SUID权限，可以看到具有SUID的文件会被系统着背景色。   
这里写图片描述   
命令“passwd”可以修改用户的密码，即改变了文件“/etc/shadow”中的内容，该文件权限如下：   
这里写图片描述   
显然只有root可以读取和强制写入[7](https://blog.csdn.net/wangzhenyu177/article/details/52080031#fn:8)该文件。   
那么普通用户在使用命令“passwd”修改密码时，为何可修改文件“/etc/shadow”中的内容？   
因为文件“/usr/bin/passwd”有SUID权限，普通用户又对其有执行权限。所以普通用户在执行“passwd”的过程中（SUID仅在执行过程中生效），会具有文件“/usr/bin/passwd”属主（即root）的权限，所以可修改文件“/etc/shadow”的内容。

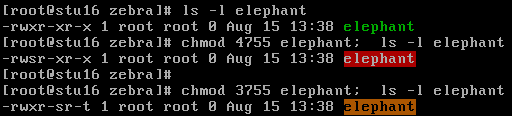
* SGID权限   
  Set GID，简称SGID。文件具有此权限时，属组用户的“x”权限位置变为“s”。SGID仅对于二进制文件和目录有效，二者具有不同的意义：
  + **对于二进制文件**   
    类似SUID权限。若执行者对于文件具备“x”权限，当文件被其执行时，执行者在执行过程中将具有**该文件属组用户的权限**。   
    例如命令“locate”执行时，可以查询文件“/var/lib/mlocate/mloca   
    te.db”，二者权限如下：   
    这里写图片描述   
    文件“/usr/bin/locate”也会被着背景色。这类似于上述的命令“passwd”与文件“/etc/shadow”的关系，不再赘述。
  + **对于目录**   
    当一目录具有SGID权限时，若用户对该目录有“r、x”权限（可“cd”至该目录），则用户“cd”至该目录后，用户的有效用户组不会变化，但在这个目录中创建的文件的属组，则不是用户的当前有效用户组，而是该目录的属组。如下所示：   
       
    可以看到用户user1在家目录下创建的文件的属组是user1，而在目录ant下创建的文件test1的属组是目录ant的属组test\_group。

**该功能常用于共同编辑文件**。比如用户user1、user2均属于用户组test\_group，目录bag的属组应为test\_group，且属组用户权限开放为“rwx”。用户user1、user2在同一目录bag下共同开发项目[8](https://blog.csdn.net/wangzhenyu177/article/details/52080031#fn:10)。   
   
如上，若目录“bag”没有SGID权限，则user1、user2虽然都可以在其下创建文件，但属组用户仍然是各自的当前有效用户组，所以用户user2对于user1创建的文件“test1\_bag”为其他用户。普通用户的遮罩码为0002，所以文件“test\_bag”对user2仅开放“r”权限。   
因为普通用户的遮罩码为0002，所以user1创建的文件的属组权限是有“w”权限的。所以如果目录“bag”开放SGID权限，则文件“test1\_bag”的属组就会是“test\_group”，从而使user2对其有“w”权限。二者就可互相修改对方创建的文件了[9](https://blog.csdn.net/wangzhenyu177/article/details/52080031#fn:11)。

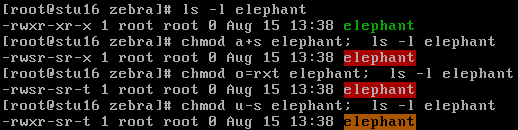
* SBIT权限   
  Sticky Bit，简称SBIT。仅对目录有效。若目录有此权限，目录的其他用户的“x”权限位置变为“t”。

效果：具有SBIT权限的目录A，其中有文件B，则仅目录A的属主、文件B的属主和root可以对其重命名、剪切、删除等。

   
如上，目录有SBIT权限时也会着背景色。目录car具有SBIT权限。用户user2在该目录创建了文件“by\_user2”，但user1不能删除、重命名，即使目录car对所有用户都开放了“w”权限。

* 修改文件的特殊权限   
  仍使用命令“chmod”，同样有8进制数和符号两种方式：
  + 使用8进制数修改   
    修改“rwx”权限时，可直接使用3位8进制数，在前面增加1位8进制数，可用来表示特殊权限。类似于“r、w、x”分别对应“4、2、1”，“SUID、SGID、SBIT”也分别对应“4、2、1”。   
       
    如上，先开放了文件的SUID权限，然后取消SUID，开放SGID、SBIT权限（SBIT不会有任何效果，因为这个文件不是目录，这里只是说明如何修改权限）。
  + 使用符号修改

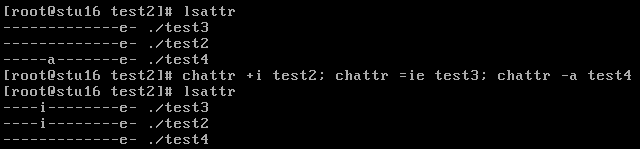
|  | **修改谁的权限** | **添加、减去、指定** | **权限** |  |
| --- | --- | --- | --- | --- |
| chmod | u、g、o、a | +、-、= | s、t | 文件 |

* + 如下所示，使用符号修改文件的特殊权限   
    
* 特殊情况：   
  这里写图片描述   
  如上，文件开放SUID、SGID权限，其属主、属组成员的“x”权限位置会变为“s”，但若其本就没开放属主或属组成员的“x”权限，则SUID、SGID会用“S”表示。
  + 显然SUID在这种情况下不会再有效果，因为属主本身不具备“x”权限了，执行者当然无法以文件属主的权限运行文件。
  + SGID对于二进制文件的情况类似于SUID，即如果文件属组用户并没有开放“x”权限，那么执行者以文件属组用户的权限执行文件并没有意义。
  + 但若是目录开放SGID，即便属组用户不开放“x”权限，SGID依然会影响到属主和其他用户，即若属主或其他用户可以在该目录下创建文件，则它们创建的文件的属组同该目录的属组。

这里写图片描述   
如上，目录开放SBIT权限，其他用户的“x”权限位置变为“t”，但若其他用户的“x”权限本就未开放，则使用“T”表示SBIT。SBIT的效果不受影响。

## 2.5 文件隐藏权限及修改

文件隐藏权限有多个，对于数据安全较为重要。

* **命令lsattr**，用于显示文件隐藏权限。   
  这里写图片描述   
  上图中，除权限“e”，其他均未开放。“e”开放表示文件使用的是ext文件系统。在红帽6中使用的是ext文件系统，所以新建的文件都有“e”权限，且无法使用“chattr”删除该权限[10](https://blog.csdn.net/wangzhenyu177/article/details/52080031#fn:5)。
* **命令chattr**，用于修改文件隐藏权限。
  + 命令“chattr”仅可在ext文件系统使用。
  + 普通命令也可使用该命令，但某些隐藏权限仅root可设置。
  + 该命令可使用“=”“+”“-”三种方式，分别表示指定、添加、减去权限。   
       
    如上图分别表示文件test2添加“i”权限；test3指定为“i、e”权限[11](https://blog.csdn.net/wangzhenyu177/article/details/52080031#fn:6)；test4去掉“a”权限。
* 两个常用权限
  + 权限“i”，仅root可设置。表示文件不可被修改、删除、重命名、设置链接（即使root也不能）。完全变为只读文件，对系统安全有很大帮助。
  + 权限“a”，仅root可设置。表示文件仅可追加数据，文件不可被修改、删除[12](https://blog.csdn.net/wangzhenyu177/article/details/52080031#fn:7)。可用于日志文件，使其可以增加内容，旧数据不能被修改和删除。

## 2.6 facl

facl：file access control list，文件访问控制列表。

对于普通用户来说，他所创建的文件，默认其他用户都是没有w和x权限的。如果想开放某个其他用户对其创建的文件有相关权限，就必须开放所有其他用户的权限。   
facl机制作为文件的额外的赋权机制（即除了u、g、o方式的权限），可解决这个问题。

**命令setfacl**可设置某用户或组对指定文件有何权限，选项“**-m**”用于赋予权限，选项“**-x**”用于收回权限。   
**命令getfacl**可查看指定文件对哪些用户和组开放了权限。

[root@localhost tmp]# setfacl -m u:hadoop:w by\_root

[root@localhost tmp]# getfacl by\_root

# file: by\_root

# owner: root

# group: root

user::rw-

user:hadoop:-w-

group::r--

mask::rw-

other::r--

# 显示文件权限的结果中，显示用户hadoop对该文件有w权限。

[root@localhost tmp]# ll by\_root

-rw-rw-r--+ 1 root root 0 Aug 27 06:59 by\_root

# 文件的详细信息中，权限位的右侧也会出现符号“+”，表示该文件有额外开放的权限。

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17

也可对指定用户组开放权限，比如仍对文件by\_root，令用户组user1对其有rw权限：

[root@localhost tmp]# setfacl -m g:user1:rw by\_root

[root@localhost tmp]# getfacl by\_root

# file: by\_root

# owner: root

# group: root

user::rw-

user:hadoop:-w-

group::r--

group:user1:rw-

mask::rw-

other::r--

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11

撤销上述赋予的权限：

[root@localhost tmp]# setfacl -x u:hadoop by\_root

[root@localhost tmp]# setfacl -x g:user1 by\_root

[root@localhost tmp]# getfacl by\_root

# file: by\_root

# owner: root

# group: root

user::rw-

group::r--

mask::r-- # mask指什么，哪位高手给解释下

other::r--

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10

## 2.7 文件权限检查次序

一个用户发起的进程访问（或写、执行）某文件，系统检查其权限过程如下：

用户访问某文件该用户是否为文件属主使用该文件属主的权限文件访问控制列表是否设置了该用户的权限使用访问控制列表中指定给该用户的权限用户是否属于文件属组使用文件属组的权限文件访问控制列表是否设置了该用户所属的组的权限使用文件访问控制列表中设置的用户所属的组的权限使用文件的其他用户权限yesnoyesnoyesnoyesno

比如，在文件by\_root的访问控制列表中，设置用户gentoo无任何权限。可验证上述的文件权限检查次序：

[root@localhost tmp]$ls -l by\_root

-rw-r--r--. 1 root root 0 Aug 27 10:06 by\_root # 用户gentoo作为该文件的其他用户是有r权限的。

[root@localhost tmp]$setfacl -m u:gentoo:--- by\_root # 访问控制列表中设置用户gentoo对该文件无任何权限

[root@localhost tmp]$su - gentoo

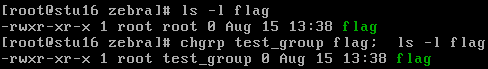
[gentoo@localhost ~]$cat /tmp/by\_root

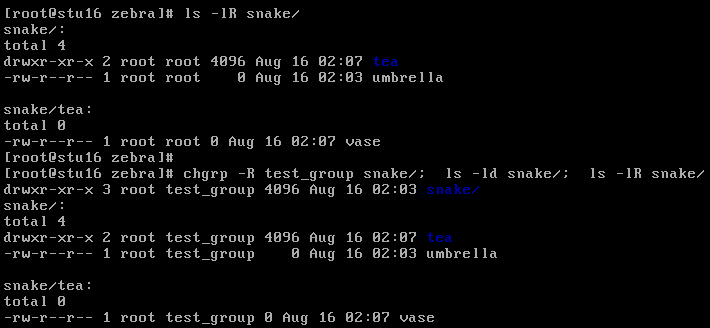
cat: /tmp/by\_root: Permission denied #读取该文件提示无权限，可以看出是文件访问控制列表生效，而没有使用文件的其他用户权限。

* 1
* 2
* 3
* 4
* 5
* 6

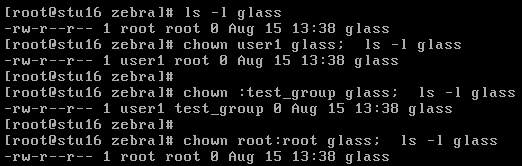
# 3.文件属主、属组修改

* **管理员命令chgrp**   
  用于修改文件属组，修改的属组必须是“/etc/group”中已存在的。
  + “-R”选项，用于递归修改指定目录下的所有文件的属组





* **管理员命令chown**   
  用于修改用户属主、属组。修改的属主、属组必须是“/etc/passwd、/etc/group”中已存在的。

默认修改属主，使用格式：“chown user:group file”[13](https://blog.csdn.net/wangzhenyu177/article/details/52080031#fn:12)可同时修改属主属组，该格式若不写用户，则仅修改属组。如下所示：   


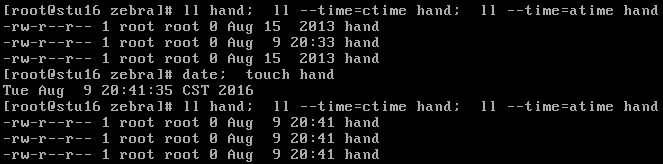
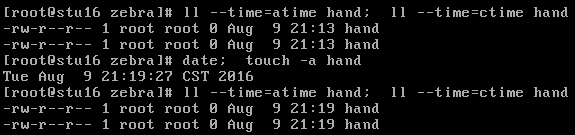
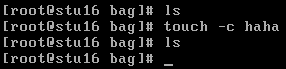
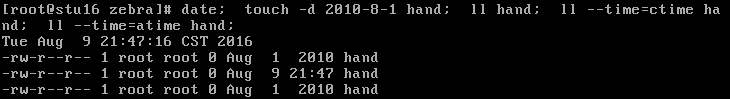
* + chown的“-R”选项用于递归修改文件属主、属组，效果类似上述命令“chgrp”，不再赘述。

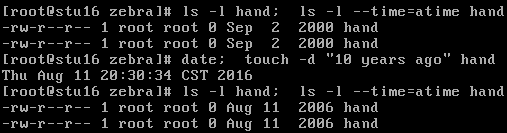
# 4.文件时间戳

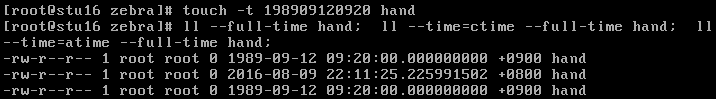
文件时间戳包括：modification time（mtime）、status time（ctime）、access time（atime）。

* mtime   
  mtime表示文件内容最近被修改（modify）的时间。   
  当文件的数据、内容被修改时，mtime会更新。数据、内容不包括文件属性、权限。   
  命令“ls -l”显示的时间默认是mtime。
* ctime   
  ctime表示文件状态（status）最近被改变（change）的时间。   
  当文件的状态被修改时，ctime会更新。比如文件的权限、属性被改变。
* atime   
  atime表示文件最近被访问（access）的时间。比如“cat”了某文件，其atime会更新。

**命令“touch”**，用于更新文件的时间戳为现在的时刻（若文件不存在，则用于创建文件）。

* 默认三个时间都更新。   
  
* 选项“**-a**”：仅更新atime。   
     
  由上操作截图可看出，仅更新atime时，ctime也会被更新。因为“最近被访问时间”也是文件属性之一。
* 选项“**-m**”：仅更新mtime。   
  与atime同理，，更新mtime时，ctime也会被更新。
* 选项“**-c**”：仅更新文件时间，若文件不存在，也不创建文件。   
  
* 选项“**-d**”：同“**–date=**”，表示更改为指定日期，而非当前日期。   
     
  由上述操作也可看出，ctime是无法被指定的[14](https://blog.csdn.net/wangzhenyu177/article/details/52080031#fn:13)，它被更新成了当前时间。   
  日期格式也可为“YYYY/MM/DD、YYYYMMDD”。

“-d”选项也可将atime、mtime更新为当前时间的某段时间前，不论文件原来的时间是多少：   
   
注意上述操作是把atime、mtime更新为当前时间的十年前，而非文件原atime、mtime的十年前（虽然这好像更合逻辑）。

* 选项“**-t**”：更改为指定时间而非当前时间，时间格式为“YYMMDDhhmm”。   
  

（完）

1. linux一切皆文件，为方便叙述，把目录文件直接称为目录。 [↩](https://blog.csdn.net/wangzhenyu177/article/details/52080031#fnref:2)
2. linux文件类型：   
   1、普通文件（-），普通文件又可按内容划分为文本文件、二进制文件、数据库文件等；   
   2、目录文件（d）；   
   3、链接文件（l），确切的说是符号链接或软链接；   
   4、字符设备文件（c）；   
   5、块设备文件（b）；   
   6、管道文件（p）；   
   7、套接字文件（s）。 [↩](https://blog.csdn.net/wangzhenyu177/article/details/52080031#fnref:1)
3. 注意，如果是目录文件，该字段显示的也仅是目录文件本身的大小，不包括目录所指向的各文件的大小。这个要和windows的文件夹大小区分开。目录也是文件并不是容器。 [↩](https://blog.csdn.net/wangzhenyu177/article/details/52080031#fnref:14)
4. 能“看到”文件内容，才能复制文件内容。剪切与复制不同，剪切并不访问文件内容，仅改变目录的文件名列表，所以剪切操作不需文件自身开放“r”权限。 [↩](https://blog.csdn.net/wangzhenyu177/article/details/52080031#fnref:3)
5. 能“进入”目录，才能在其中创建文件；   
   能“进入”目录，“看到”目录下的各文件，才能操作它们。 [↩](https://blog.csdn.net/wangzhenyu177/article/details/52080031#fnref:4)
6. shell脚本不是二进制程序，它是很多命令（二进制程序）的堆砌。 [↩](https://blog.csdn.net/wangzhenyu177/article/details/52080031#fnref:9)
7. 只有root可为文本文件“/etc/shadow”添加“w”权限，而后写入。 [↩](https://blog.csdn.net/wangzhenyu177/article/details/52080031#fnref:8)
8. 两个或以上的用户共同在同一目录下开发，这些用户应有共同的用户组A（主要是为了指派权限，不一定作为当前有效用户组），且目录的属组应为A，然后该目录需把属组权限开放为“rwx”才能保证这些用户都能进入该目录并能创建文件；为防止干扰，该目录的其他用户权限均关闭即可。 [↩](https://blog.csdn.net/wangzhenyu177/article/details/52080031#fnref:10)
9. 当然可以修改umask，使user1创建的文件对其他用户也开放“w”，反正只有属于用户组“test\_group”的才能进目录“bag”；或者user1、user2在创建文件前都“newgrp”把当前有效组变为“test\_group”，使创建的属组为“test\_group”，也可达到互相修改对方创建的文件的目的。不过显然都不如设置SGID权限，对其他文件无影响且操作简单。 [↩](https://blog.csdn.net/wangzhenyu177/article/details/52080031#fnref:11)
10. 显然，文件系统的变更，不可能通过修改某个文件的“e”权限来实现。 [↩](https://blog.csdn.net/wangzhenyu177/article/details/52080031#fnref:5)
11. 显然，在ext文件系统，使用“=”指定特殊权限，必须有“e”权限。 [↩](https://blog.csdn.net/wangzhenyu177/article/details/52080031#fnref:6)
12. 开放权限“a”的文件，不能通过“vim”编辑也不能使用“>”输入重定向增加内容，**即使该文件本来是空的**。这应该是因为“vim”和“>”输入重定向被视作修改文件内容。而使用“>>”追加重定向增加文件内容则不会报错。 [↩](https://blog.csdn.net/wangzhenyu177/article/details/52080031#fnref:7)
13. 该格式中的“：”替换为“.”效果一样，但一般使用“：”，因为用户名中是允许出现“.”的，这样会造成系统的误判；而“：”是不允许出现在用户名中的。 [↩](https://blog.csdn.net/wangzhenyu177/article/details/52080031#fnref:12)
14. ctime记录的是文件的状态最近被改变的时间，文件的属性、权限变化时它才会跟着变（具有客观性）。所以ctime无法被人为指定为某时间，也无法单独被更新为当前时间。“touch 文件”也可看做是atime、mtime更新所以ctime 更新。 [↩](https://blog.csdn.net/wangzhenyu177/article/details/52080031#fnref:13)

版权声明：本文为博主原创文章，未经博主允许不得转载。 https://blog.csdn.net/wangzhenyu177/article/details/52080031