# ls

显示目录内容列表

## 补充说明

**ls命令** 就是list的缩写，用来显示目标列表，在Linux中是使用率较高的命令。ls命令的输出信息可以进行彩色加亮显示，以分区不同类型的文件。

### 语法

ls [选项] [文件名...]  
 [-1abcdfgiklmnopqrstuxABCDFGLNQRSUX] [-w cols] [-T cols] [-I pattern] [--full-time]   
 [--format={long,verbose,commas,across,vertical,single-col‐umn}]   
 [--sort={none,time,size,extension}] [--time={atime,access,use,ctime,status}]   
 [--color[={none,auto,always}]] [--help] [--version] [--]

### 选项

-C # 多列输出，纵向排序。  
-F # 每个目录名加 "/" 后缀，每个 FIFO 名加 "|" 后缀， 每个可运行名加“ \* ”后缀。  
-R # 递归列出遇到的子目录。  
-a # 列出所有文件，包括以 "." 开头的隐含文件。  
-c # 使用“状态改变时间”代替“文件修改时间”为依据来排序（使用“-t”选项时）或列出（使用“-l”选项时）。  
-d # 将目录名像其它文件一样列出，而不是列出它们的内容。  
-i # 输出文件前先输出文件系列号（即 i 节点号: i-node number）。 -l 列出（以单列格式）文件模式  
 # （file mode），文件的链接数，所有者名，组名，文件大小（以字节为单位），时间信息，及文件名。  
 # 缺省时，时间信息显示最近修改时间；可以以选项“-c”和“-u”选择显示其它两种时间信息。对于设备文件，  
 # 原先显示文件大小的区域通常显示的是主要和次要的信号（majorand minor device numbers）。  
-q # 将文件名中的非打印字符输出为问号。（对于到终端的输出这是缺省的。）  
-r # 逆序排列。  
-t # 按时间信息排序。  
-u # 使用最近访问时间代替最近修改时间为依据来排序（使用“-t”选项时）或列出（使用“-l”选项时）。  
-1 # 单列输出。  
-1, --format=single-column # 一行输出一个文件（单列输出）。如标准输出不是到终端，此选项就是缺省选项。  
-a, --all # 列出目录中所有文件，包括以“.”开头的文件。  
-b, --escape # 把文件名中不可输出的字符用反斜杠加字符编号(就像在 C 语言里一样)的形式列出。  
-c, --time=ctime, --time=status  
 # 按文件状态改变时间（i节点中的ctime）排序并输出目录内  
 # 容。如采用长格式输出（选项“-l”），使用文件的状态改  
 # 变时间取代文件修改时间。【译注：所谓文件状态改变（i节  
 # 点中以ctime标志），既包括文件被修改，又包括文件属性（ 如所有者、组、链接数等等）的变化】  
-d, --directory  
 # 将目录名像其它文件一样列出，而不是列出它们的内容。  
-f # 不排序目录内容；按它们在磁盘上存储的顺序列出。同时启 动“ -a ”选项，如果在“ -f ”之前存在“ -l”、  
 # “ - -color ”或“ -s ”，则禁止它们。  
-g # 忽略，为兼容UNIX用。  
-i, --inode  
 # 在每个文件左边打印 i 节点号（也叫文件序列号和索引号: file serial number and index num‐  
 # ber）。i节点号在每个特定的文件系统中是唯一的。  
-k, --kilobytes  
 # 如列出文件大小，则以千字节KB为单位。  
-l, --format=long, --format=verbose  
 # 输出的信息从左到右依次包括文件名、文件类型、权限、硬链接数、所有者名、组名、大小（byte）  
 # 、及时间信息（如未指明是其它时间即指修改时间）。对于6个月以上的文件或超出未来  
 # 1小时的文件，时间信息中的时分将被年代取代。  
 # 每个目录列出前，有一行“总块数”显示目录下全部文件所占的磁盘空间。块默认是1024字节；  
 # 如果设置了 POSIXLY\_CORRECT 的环境变量，除非用“-k”选项，则默认块大小是 512 字节。  
 # 每一个硬链接都计入总块数（因此可能重复计数），这无 疑是个缺点。  
  
# 列出的权限类似于以符号表示（文件）模式的规范。但是 ls  
 # 在每套权限的第三个字符中结合了多位（ multiple bits ） 的信息，如下： s 如果设置了 setuid  
 # 位或 setgid 位，而且也设置了相应的可执行位。 S 如果设置了 setuid 位或 setgid  
 # 位，但是没有设置相应的可执行位。 t 如果设置了 sticky 位，而且也设置了相应的可执行位。 T  
 # 如果设置了 sticky 位，但是没有设置相应的可执行位。 x  
 # 如果仅仅设置了可执行位而非以上四种情况。 - 其它情况（即可执行位未设置）。  
-m, --format=commas  
 # 水平列出文件，每行尽可能多，相互用逗号和一个空格分隔。  
-n, --numeric-uid-gid  
 # 列出数字化的 UID 和 GID 而不是用户名和组名。  
-o # 以长格式列出目录内容，但是不显示组信息。等于使用“ --format=long --no-group  
 # ”选项。提供此选项是为了与其它版本的 ls 兼容。  
-p # 在每个文件名后附上一个字符以说明该文件的类型。类似“ -F ”选项但是不 标示可执行文件。  
-q, --hide-control-chars  
 # 用问号代替文件名中非打印的字符。这是缺省选项。  
-r, --reverse  
 # 逆序排列目录内容。  
-s, --size  
 # 在每个文件名左侧输出该文件的大小，以 1024 字节的块为单位。如果设置了 POSIXLY\_CORRECT  
 # 的环境变量，除非用“ -k ”选项，块大小是 512 字节。  
-t, --sort=time  
 # 按文件最近修改时间（ i 节点中的 mtime ）而不是按文件名字典序排序，新文件 靠前。  
-u, --time=atime, --time=access, --time=use  
 # 类似选项“ -t ”，但是用文件最近访问时间（ i 节点中的 atime ）取代文件修  
 # 改时间。如果使用长格式列出，打印的时间是最近访问时间。  
-w, --width cols  
 # 假定屏幕宽度是 cols （ cols 以实际数字取代）列。如未用此选项，缺省值是这  
 # 样获得的：如可能先尝试取自终端驱动，否则尝试取自环境变量 COLUMNS （如果设  
 # 置了的话），都不行则取 80 。  
  
-x, --format=across, --format=horizontal  
 # 多列输出，横向排序。  
  
-A, --almost-all  
 # 显示除 "." 和 ".." 外的所有文件。  
  
-B, --ignore-backups  
 # 不输出以“ ~ ”结尾的备份文件，除非已经在命令行中给出。  
  
-C, --format=vertical  
 # 多列输出，纵向排序。当标准输出是终端时这是缺省项。使用命令名 dir 和 d 时， 则总是缺省的。  
  
-D, --dired  
 # 当采用长格式（“-l”选项）输出时，在主要输出后，额外打印一行： //DIRED// BEG1 END1 BEG2  
 # END2 ...  
  
# BEGn 和 ENDn 是无符号整数，记录每个文件名的起始、结束位置在输出中的位置（  
# 字节偏移量）。这使得 Emacs 易于找到文件名，即使文件名包含空格或换行等非正  
# 常字符也无需特异的搜索。  
#   
# 如果目录是递归列出的（“ -R ”选项），每个子目录后列出类似一行：  
 # //SUBDIRED// BEG1 END1 ... 【译注：我测试了 TurboLinux4.0 和 RedHat6.1 ，发现它们都是在 “  
 # //DIRED// BEG1... ”之后列出“ //SUBDIRED// BEG1 ... ”，也即只有一个  
 # 而不是在每个子目录后都有。而且“ //SUBDIRED// BEG1 ... ”列出的是各个子目 录名的偏移。】  
  
-F, --classify, --file-type  
 # 在每个文件名后附上一个字符以说明该文件的类型。“ \* ”表示普通的可执行文件； “ / ”表示目录；“  
 # @ ”表示符号链接；“ | ”表示FIFOs；“ = ”表示套接字 (sockets) ；什么也没有则表示普通文件。  
  
-G, --no-group  
 # 以长格式列目录时不显示组信息。  
  
-I, --ignorepattern  
 # 除非在命令行中给定，不要列出匹配shell文件名匹配式（pattern ，不是指一般  
 # 表达式）的文件。在shell中，文件名以"."起始的不与在文件名匹配式(pattern)  
 # 开头的通配符匹配。  
  
-L, --dereference  
 # 列出符号链接指向的文件的信息，而不是符号链接本身。  
  
-N, --literal  
 # 不要用引号引起文件名。  
  
-Q, --quote-name  
 # 用双引号引起文件名，非打印字符以 C 语言的方法表示。  
  
-R, --recursive  
 # 递归列出全部目录的内容。  
  
-S, --sort=size  
 # 按文件大小而不是字典序排序目录内容，大文件靠前。  
  
-T, --tabsize cols  
 # 假定每个制表符宽度是 cols 。缺省为 8。为求效率， ls 可能在输出中使用制表符。 若 cols 为  
 0，则不使用制表符。  
  
-U, --sort=none  
 # 不排序目录内容；按它们在磁盘上存储的顺序列出。（选项“-U”和“-f”的不  
 # 同是前者不启动或禁止相关的选项。）这在列很大的目录时特别有用，因为不加排序  
 # 能显著地加快速度。  
  
-X, --sort=extension  
 # 按文件扩展名（由最后的 "." 之后的字符组成）的字典序排序。没有扩展名的先列 出。  
  
--color[=when]  
 # 指定是否使用颜色区别文件类别。环境变量 LS\_COLORS 指定使用的颜色。如何设置 这个变量见 dir‐  
 # colors(1) 。 when 可以被省略，或是以下几项之一：  
none # 不使用颜色，这是缺省项。  
 # auto 仅当标准输出是终端时使用。 always 总是使用颜色。指定 --color 而且省略 when 时就等同于  
 # --color=always 。  
  
--full-time  
 # 列出完整的时间，而不是使用标准的缩写。格式如同 date(1) 的缺省格式；此格式  
 # 是不能改变的，但是你可以用 cut(1) 取出其中的日期字串并将结果送至命令 “ date -d ”。  
  
# 输出的时间包括秒是非常有用的。（ Unix 文件系统储存文件的时间信息精确到秒，  
 # 因此这个选项已经给出了系统所知的全部信息。）例如，当你有一个 Makefile 文件  
 # 不能恰当地生成文件时，这个选项会提供帮助。

### 参数

目录：指定要显示列表的目录，也可以是具体的文件。

### 实例

$ ls # 仅列出当前目录可见文件  
$ ls -l # 列出当前目录可见文件详细信息  
$ ls -hl # 列出详细信息并以可读大小显示文件大小  
$ ls -al # 列出所有文件（包括隐藏）的详细信息  
$ ls --human-readable --size -1 -S --classify # 按文件大小排序  
$ du -sh \* | sort -h # 按文件大小排序(同上)

显示当前目录下包括隐藏文件在内的所有文件列表

[root@localhost ~]# ls -a  
. anaconda-ks.cfg .bash\_logout .bashrc install.log .mysql\_history satools .tcshrc .vimrc  
.. .bash\_history .bash\_profile .cshrc install.log.syslog .rnd .ssh .viminfo

输出长格式列表

[root@localhost ~]# ls -1  
  
anaconda-ks.cfg  
install.log  
install.log.syslog  
satools

显示文件的inode信息

索引节点（index inode简称为“inode”）是Linux中一个特殊的概念，具有相同的索引节点号的两个文本本质上是同一个文件（除文件名不同外）。

[root@localhost ~]# ls -i -l anaconda-ks.cfg install.log  
2345481 -rw------- 1 root root 859 Jun 11 22:49 anaconda-ks.cfg  
2345474 -rw-r--r-- 1 root root 13837 Jun 11 22:49 install.log

水平输出文件列表

[root@localhost /]# ls -m  
  
bin, boot, data, dev, etc, home, lib, lost+found, media, misc, mnt, opt, proc, root, sbin, selinux, srv, sys, tmp, usr, var

修改最后一次编辑的文件

最近修改的文件显示在最上面。

[root@localhost /]# ls -t  
  
tmp root etc dev lib boot sys proc data home bin sbin usr var lost+found media mnt opt selinux srv misc

显示递归文件

[root@localhost ~]# ls -R  
.:  
anaconda-ks.cfg install.log install.log.syslog satools  
  
./satools:  
black.txt freemem.sh iptables.sh lnmp.sh mysql php502\_check.sh ssh\_safe.sh

打印文件的UID和GID

[root@localhost /]# ls -n  
  
total 254  
drwxr-xr-x 2 0 0 4096 Jun 12 04:03 bin  
drwxr-xr-x 4 0 0 1024 Jun 15 14:45 boot  
drwxr-xr-x 6 0 0 4096 Jun 12 10:26 data  
drwxr-xr-x 10 0 0 3520 Sep 26 15:38 dev  
drwxr-xr-x 75 0 0 4096 Oct 16 04:02 etc  
drwxr-xr-x 4 0 0 4096 Jun 12 10:26 home  
drwxr-xr-x 14 0 0 12288 Jun 16 04:02 lib  
drwx------ 2 0 0 16384 Jun 11 22:46 lost+found  
drwxr-xr-x 2 0 0 4096 May 11 2011 media  
drwxr-xr-x 2 0 0 4096 Nov 8 2010 misc  
drwxr-xr-x 2 0 0 4096 May 11 2011 mnt  
drwxr-xr-x 2 0 0 4096 May 11 2011 opt  
dr-xr-xr-x 232 0 0 0 Jun 15 11:04 proc  
drwxr-x--- 4 0 0 4096 Oct 15 14:43 root  
drwxr-xr-x 2 0 0 12288 Jun 12 04:03 sbin  
drwxr-xr-x 2 0 0 4096 May 11 2011 selinux  
drwxr-xr-x 2 0 0 4096 May 11 2011 srv  
drwxr-xr-x 11 0 0 0 Jun 15 11:04 sys  
drwxrwxrwt 3 0 0 98304 Oct 16 08:45 tmp  
drwxr-xr-x 13 0 0 4096 Jun 11 23:38 usr  
drwxr-xr-x 19 0 0 4096 Jun 11 23:38 var

列出文件和文件夹的详细信息

[root@localhost /]# ls -l  
  
total 254  
drwxr-xr-x 2 root root 4096 Jun 12 04:03 bin  
drwxr-xr-x 4 root root 1024 Jun 15 14:45 boot  
drwxr-xr-x 6 root root 4096 Jun 12 10:26 data  
drwxr-xr-x 10 root root 3520 Sep 26 15:38 dev  
drwxr-xr-x 75 root root 4096 Oct 16 04:02 etc  
drwxr-xr-x 4 root root 4096 Jun 12 10:26 home  
drwxr-xr-x 14 root root 12288 Jun 16 04:02 lib  
drwx------ 2 root root 16384 Jun 11 22:46 lost+found  
drwxr-xr-x 2 root root 4096 May 11 2011 media  
drwxr-xr-x 2 root root 4096 Nov 8 2010 misc  
drwxr-xr-x 2 root root 4096 May 11 2011 mnt  
drwxr-xr-x 2 root root 4096 May 11 2011 opt  
dr-xr-xr-x 232 root root 0 Jun 15 11:04 proc  
drwxr-x--- 4 root root 4096 Oct 15 14:43 root  
drwxr-xr-x 2 root root 12288 Jun 12 04:03 sbin  
drwxr-xr-x 2 root root 4096 May 11 2011 selinux  
drwxr-xr-x 2 root root 4096 May 11 2011 srv  
drwxr-xr-x 11 root root 0 Jun 15 11:04 sys  
drwxrwxrwt 3 root root 98304 Oct 16 08:48 tmp  
drwxr-xr-x 13 root root 4096 Jun 11 23:38 usr  
drwxr-xr-x 19 root root 4096 Jun 11 23:38 var

列出可读文件和文件夹详细信息

[root@localhost /]# ls -lh  
  
total 254K  
drwxr-xr-x 2 root root 4.0K Jun 12 04:03 bin  
drwxr-xr-x 4 root root 1.0K Jun 15 14:45 boot  
drwxr-xr-x 6 root root 4.0K Jun 12 10:26 data  
drwxr-xr-x 10 root root 3.5K Sep 26 15:38 dev  
drwxr-xr-x 75 root root 4.0K Oct 16 04:02 etc  
drwxr-xr-x 4 root root 4.0K Jun 12 10:26 home  
drwxr-xr-x 14 root root 12K Jun 16 04:02 lib  
drwx------ 2 root root 16K Jun 11 22:46 lost+found  
drwxr-xr-x 2 root root 4.0K May 11 2011 media  
drwxr-xr-x 2 root root 4.0K Nov 8 2010 misc  
drwxr-xr-x 2 root root 4.0K May 11 2011 mnt  
drwxr-xr-x 2 root root 4.0K May 11 2011 opt  
dr-xr-xr-x 235 root root 0 Jun 15 11:04 proc  
drwxr-x--- 4 root root 4.0K Oct 15 14:43 root  
drwxr-xr-x 2 root root 12K Jun 12 04:03 sbin  
drwxr-xr-x 2 root root 4.0K May 11 2011 selinux  
drwxr-xr-x 2 root root 4.0K May 11 2011 srv  
drwxr-xr-x 11 root root 0 Jun 15 11:04 sys  
drwxrwxrwt 3 root root 96K Oct 16 08:49 tmp  
drwxr-xr-x 13 root root 4.0K Jun 11 23:38 usr  
drwxr-xr-x 19 root root 4.0K Jun 11 23:38 var

显示文件夹信息

[root@localhost /]# ls -ld /etc/  
  
drwxr-xr-x 75 root root 4096 Oct 16 04:02 /etc/

按时间列出文件和文件夹详细信息

[root@localhost /]# ls -lt  
  
total 254  
drwxrwxrwt 3 root root 98304 Oct 16 08:53 tmp  
drwxr-xr-x 75 root root 4096 Oct 16 04:02 etc  
drwxr-x--- 4 root root 4096 Oct 15 14:43 root  
drwxr-xr-x 10 root root 3520 Sep 26 15:38 dev  
drwxr-xr-x 14 root root 12288 Jun 16 04:02 lib  
drwxr-xr-x 4 root root 1024 Jun 15 14:45 boot  
drwxr-xr-x 11 root root 0 Jun 15 11:04 sys  
dr-xr-xr-x 232 root root 0 Jun 15 11:04 proc  
drwxr-xr-x 6 root root 4096 Jun 12 10:26 data  
drwxr-xr-x 4 root root 4096 Jun 12 10:26 home  
drwxr-xr-x 2 root root 4096 Jun 12 04:03 bin  
drwxr-xr-x 2 root root 12288 Jun 12 04:03 sbin  
drwxr-xr-x 13 root root 4096 Jun 11 23:38 usr  
drwxr-xr-x 19 root root 4096 Jun 11 23:38 var  
drwx------ 2 root root 16384 Jun 11 22:46 lost+found  
drwxr-xr-x 2 root root 4096 May 11 2011 media  
drwxr-xr-x 2 root root 4096 May 11 2011 mnt  
drwxr-xr-x 2 root root 4096 May 11 2011 opt  
drwxr-xr-x 2 root root 4096 May 11 2011 selinux  
drwxr-xr-x 2 root root 4096 May 11 2011 srv  
drwxr-xr-x 2 root root 4096 Nov 8 2010 misc

按修改时间列出文件和文件夹详细信息

[root@localhost /]# ls -ltr  
  
total 254  
drwxr-xr-x 2 root root 4096 Nov 8 2010 misc  
drwxr-xr-x 2 root root 4096 May 11 2011 srv  
drwxr-xr-x 2 root root 4096 May 11 2011 selinux  
drwxr-xr-x 2 root root 4096 May 11 2011 opt  
drwxr-xr-x 2 root root 4096 May 11 2011 mnt  
drwxr-xr-x 2 root root 4096 May 11 2011 media  
drwx------ 2 root root 16384 Jun 11 22:46 lost+found  
drwxr-xr-x 19 root root 4096 Jun 11 23:38 var  
drwxr-xr-x 13 root root 4096 Jun 11 23:38 usr  
drwxr-xr-x 2 root root 12288 Jun 12 04:03 sbin  
drwxr-xr-x 2 root root 4096 Jun 12 04:03 bin  
drwxr-xr-x 4 root root 4096 Jun 12 10:26 home  
drwxr-xr-x 6 root root 4096 Jun 12 10:26 data  
dr-xr-xr-x 232 root root 0 Jun 15 11:04 proc  
drwxr-xr-x 11 root root 0 Jun 15 11:04 sys  
drwxr-xr-x 4 root root 1024 Jun 15 14:45 boot  
drwxr-xr-x 14 root root 12288 Jun 16 04:02 lib  
drwxr-xr-x 10 root root 3520 Sep 26 15:38 dev  
drwxr-x--- 4 root root 4096 Oct 15 14:43 root  
drwxr-xr-x 75 root root 4096 Oct 16 04:02 etc  
drwxrwxrwt 3 root root 98304 Oct 16 08:54 tmp

按照特殊字符对文件进行分类

[root@localhost nginx-1.2.1]# ls -F  
  
auto/ CHANGES CHANGES.ru conf/ configure\* contrib/ html/ LICENSE Makefile man/ objs/ README src/

列出文件并标记颜色分类

[root@localhost nginx-1.2.1]# ls --color=auto  
  
auto CHANGES CHANGES.ru conf configure contrib html LICENSE Makefile man objs README src

## 扩展知识

### 不同颜色代表的文件类型

* 蓝色：目录
* 绿色：可执行文件
* 白色：一般性文件，如文本文件，配置文件等
* 红色：压缩文件或归档文件
* 浅蓝色：链接文件
* 红色闪烁：链接文件存在问题
* 黄色：设备文件
* 青黄色：管道文件