|  |  |
| --- | --- |
|  | **命令** |
|  | tar <操作> [选项] 归档名文件列表    操作：     * -c   创建一个新的归档文件 * -r    向归档文件末尾追加文件 * -u   更新归档文件 * -x   从归档文件中解出文件 * -t    列出归档文件中的文件 * --delete       从归档文件中删除一个文件 * -A   合并两个规档文件     选项：     * -C  指定一个解档目录 * -f    指定一个归档文件/设备进行操作 * -z   调用gzip来压缩/解压缩文件 |

|  |  |
| --- | --- |
|  | **注意** |
|  | * 归档文件的“.tar”后缀名不是必须的，但是一般我们都会加上这个后缀，以告诉用户这个文件是一个归档文件 * -f选项后必须紧跟归档文件名 * tar不会压缩文件 |

|  |  |
| --- | --- |
|  | **命令** |
|  | gzip [选项] 压缩（解压缩）的文件名     * -c        将输出写到标准输出上，并保留原有文件。 * -d        将压缩文件解压。 * -r         递归式压缩/解压缩指定目录中的所有文件。 * -num   用指定的数字num调整压缩的速度和压缩比。 * -v         对每一个压缩和解压的文件显示文件名和压缩比. * -t         测试，检查压缩文件是否完整。 |

|  |  |
| --- | --- |
|  | **注意** |
|  | * gzip 在压缩文件过程中，会自动为文件添加一个“.gz”的后缀名，并且将原文件删除。 * 如果要保留原文件，就必须使用-c 选项和Shell的输出重定向机制（在后面会进行详细讨论） |
|  | **命令** | |
|  | 用法：find  <path> <-option> [-print | -ls]  [-exec|-ok <command> {} \;]    在<path>指定的目录结构中搜索文件，并执行指定的操作。      find需要根据某个条件在特定的目录进行查找，有以下常用的查找选项：           -print    将查找到的文件输出到标准输出         -exec command {} \;   将查到的文件执行command操作,{} 和 \;之间有空格         -fprint  <filename> 将查找到的文件输出到指定文件         -ok       和-exec相同，只不过在操作前要询问用户         -name <filename>  : 查找名为filename的文件，即按文件名查找。         -iname <filename>    : 与-name相同,只是忽略大小写(-i);         -perm                      : 按执行权限来查找         -user <username>     : 按文件属主来查找         -group <groupname>   : 按组来查找         -user <username>     : 按文件属主来查找         -amin  <-n|+n>       : 按文件访问时间来查找文件,-n指n分钟以内,+n指n分钟以前         -atime <-n|+n>       : 按文件访问时间来查找文件,-n指n天以内,+n指n天以前         -cmin  <-n|+n>       : 按文件创建时间来查找文件,-n指n分钟以内,+n指n分钟以前         -ctime <-n|+n>       : 按文件创建时间来查找文件,-n指n天以内,+n指n天以前         -mmin  <-n|+n>       : 按文件更改时间来查找文件,-n指n分钟以内,+n指n分钟以前         -mtime <-n|+n>       : 按文件更改时间来查找文件,-n指n天以内,+n指n天以前         -empty               : 空白文件、空白文件夹、没有子目录的文件夹         -newer <f1 !f2>      : 查更改时间比f1新但比f2旧的文件         -type <b|d|c|p|l|f>  : 查是块设备/目录/字符设备/管道/符号链接/普通文件         -prune               : 忽略某个目录         -size    <+n|-n>    :根据文件大小查找文件，+n表示超过n大小的文件，-n表示小于n大小的文件 | |

|  |  |
| --- | --- |
|  | **注意** |
|  |        -size <n[cwbkMG]>，其中：n表示文件大小的数值，b表示512字节的块，c表示字节，k表示K字节，M表示M字节（1024\*1024字节=1024k），G表示G字节（1024\*1024\*1024字节=1024M）           -type <  [bcdpfls]>，其中b表示块设备文件，c表示字符设备文件，d表示目录，p表示命名管道文件，f表示普通文件，l表示符号链接文件，s表示socket文件 |

|  |  |
| --- | --- |
|  | **注意** |
|  |        find命令的所谓“命令动作（Actions）”,表示在查找到文件后，要对这些文件所做的操作，我们在这里只介绍其中三个常用的命令动作：    1.       –print：find命令的默认动作，将查找到的文件名输出  2.       –ls：将将查找到的文件名以ls –dils格式输出  3.       -exec command ; ：对查找到的文件名执行指定命令，命令command由用户指定，用{} 代表查找到的文件，用\;结束。  4.       -ok command ; ：与-exec非常类同，只不过在对文件执行指定命令之前会询问用户是否执行 |

|  |  |
| --- | --- |
|  | **注意** |
|  |        在Linux中与文件相关的时间属性有三个：    1.       atime：文件最后被访问时间(Time when file data  was last accessed.)，atime  是在读取文件或者执行文件时更改的。  2.       mtime：文件内容最后被更改的时间(Time when data was  last modified.)，mtime  是在写入文件时随文件内容的更改而更改的  3.       ctime：文件状态（属性或者内容）最后被更改的时间(Time when file  status was last changed)，ctime  是在写入文件、更改所有者、权限或链接设置时随 Inode 的内容更改而更改的。           常常在一些文章和资料中提到的文件时间戳(Timestamp)指的是文章的mtime。 |

|  |  |
| --- | --- |
|  | **注意** |
|  |        在find命令支持用逻辑运算符来操作单个逻辑条件，或者将逻辑条件连接起来：    1.       –a 表示与  2.       –o 表示或  3.       ！表示取反           注意:使用多个条件时,这些条件要用圆括号括起来,左右圆括号还要使用反斜杠"\"来进行转义,以防止shell的解释; |

3.3

某个目录下中创建的所有文本文件的修改时间，而不需要列出分散的文件。尽管这很容易在小目录中实现，但是对于大型文件系统则非常困难。

要解决这个问题，可以使用 bash shell 中内置的通配符(wildcard)支持。这种支持也称为“globbing”（因为它最初被实现为一个名为 /etc/glob 的程序），让你能够使用通配符模式指定多个文件。

包含任何 '?'、'\*' 或 '[' 字符的字符串就是一个通配符模式。Globbing 是指 shell（或另一个程序）将这些模式扩展为一组匹配该模式的参数的过程。这种匹配按照如下方式完成：

                    表1 Linux中常用通配符

|  |  |  |
| --- | --- | --- |
| 字符 | 含义 | 实例 |
| \* | 匹配 0 或多个字符 | a\*b   a与b之间可以有任意长度的任意字符, 也可以一个也没有, 如aabcb,  axyzb, a012b, ab。 |
| ? | 匹配任意一个字符 | a?b  a与b之间必须也只能有一个字符, 可以是任意字符, 如aab, abb,  acb, a0b。 |
| [list] | 匹配 list 中的任意单一字符 | a[xyz]b   a与b之间必须也只能有一个字符,但只能是 x 或 y 或 z, 如: axb,  ayb, azb。 |
| [!list] | 匹配除list 中的任意单一字符 | a[!0-9]b  a与b之间必须也只能有一个字符,但不能是阿拉伯数字, 如axb, aab,  a-b。 |
| [c1-c2] | 匹配 c1-c2 中的任意单一字符如：[0-9] [a-z] | a[0-9]b  0与9之间必须也只能有一个字符如a0b,  a1b... a9b。 |
| {string1,string2,...} | 匹配 sring1 或 string2 (或更多)其一字符串 | a{abc,xyz,123}b     a与b之间只能是abc或xyz或123这三个字符串之一。 |

我们常常在 ls、cp、mv 或 rm 命令中需要操作多个文件或目录名时使用通配符。

|  |  |
| --- | --- |
|  | **注意** |
|  | '\*' 和  '?' 字符与它们自身匹配。如果在文件名中使用这些字符，那么需要注意适当的引用或转义。    两个字符之间的 '-' 字符表示一个范围，包括这两个字符和排序序列中介于这两个字符之间的所有字符。例如，[0-9a-fA-F] 表示任何大写或小写十六进制数位。你可以通过将 '-' 放到一个范围的首位或末位来匹配它。    '!' 在Shell中是有特殊含义的，因此需要小心地对它进行适当的转义。    文件通配符被单独应用到路径名的每个组成中。但我们无法匹配'/'。 |

|  |  |
| --- | --- |
|  | **小心** |
|  | 命令中的任何通配符都可以被 shell 扩展，这将导致意外的结果。在使用 cp、mv 或 rm 出现异常行为之前，使用 ls 检查你的通配符模式。 |

3.4

|  |  |
| --- | --- |
|  | **命令** |
|  | 用法：cat [选项]... [文件]...    将[文件]或标准输入组合输出到标准输出。     * -b对非空输出行编号 * -E在每行的未尾添加$符号，便于确认每一行，因为有些行内容太长了，在显示时会在显示器上以多行表示。 * -n对输出的所有行编号 * -s将相连的多个空行用一个空行来代替 * -T将Tab字符显示为^I     如果没有指定文件，或者文件为"-"，则从标准输入读取。 |

|  |  |
| --- | --- |
|  | **命令** |
|  | 用法：less [选项]... [文件]...    对文件或其它输出进行分页显示。     * -n对输出的所有行编号 * -M  输出打开的文件名，当前显示的行数，总行数和光标当前所处文档位置 * -s将相连的多个空行用一个空行来代替 * -S  截断过长的行而不换行显示     在less中，可以使用如下命令来控制文本的查看：                                          导航命令回车键，e向下移动一行 y,k向上移动一行空格键，f向下滚动一屏b向上滚动一屏 d   向下滚动半屏   u向上滚动半屏   g  跳到第一行   G跳到最后一行p n％跳到n％。如50％，表示从整个文档的50％处开始显示  搜索命令/pattern从光标处向下搜索与pattern匹配的字符串，例如/ftpuser，表示从光标处向下文件中搜索单词ftpuser?pattern从光标处向上搜索与pattern匹配的字符串 n跳到下一个匹配的文本  N跳到上一个匹配的文本其他命令R,r刷新屏幕   q退出  :e [文件]打开一个新文件:n如打开多个文件，则跳转到打开的下一个文件:p如打开多个文件，则跳转到打开的上一个文件:d如打开多个文件，则关闭当前查看的文件!command调用Shell命令，例如使用“！ls”，表示列出当前目录下的所有文件 |

|  |  |
| --- | --- |
|  | **命令** |
|  | 用法：head [选项]... [文件]...    将每个指定文件的头10 行显示到标准输出。    如果指定了多于一个文件，在每一段输出前会给出文件名作为文件头。如果不指定文件，或者文件为"-"，则从标准输入读取数据。     * -c  K 显示每个文件的前K 字节内容；如果附加"-"参数，则除了每个文件的最后K字节数据外显示剩余全部内容 * -n  K 显示文件的前K 行内容. |

|  |  |
| --- | --- |
|  | **命令** |
|  | 用法：tail [选项]... [文件]...    显示每个指定文件的最后10 行到标准输出。    若指定了多于一个文件，程序会在每段输出的开始添加相应文件名作为头。如果不指定文件或文件为"-"  ，则从标准输入读取数据。      -c K     输出最后K 字节；另外，使用-c +K 从每个文件第K 字节输出    -n K     显示文件的最后K 行内容. |