|  |  |  |  |
| --- | --- | --- | --- |
| LINUX常用操作命令 | | | |
| 计划学时 | 12 | 授课时间 | 第 周 |
| 步骤 | 教学内容 | | |
| 明确任务 | 1、学会文件目录操作命令  2、学会文本文件操作命令  3、学会打包和压缩命令  4、学会信息显示命令  5、学会文本编辑器vi/vim  6、学会使用shell | | |
| 项目引导 | 在linux中我们应该如何来查看文件？如何对文件进行相应的操作？ | | |
| **1、文件目录操作命令**  **（1）常用的文件目录操作命令**  下表中列出了一些常用的文件目录操作命令。   |  |  |  |  | | --- | --- | --- | --- | | 命令 | 功能 | 命令 | 功 能 | | ls | 显示文件和目录列表 | pwd | 显示当前工作目录 | | touch | 生成一个空文件或更改文件的时间 | cd | 切换目录 | | cp | 复制文件或目录 | find | 在文件系统中查找指定的文件 | | mv | 移动文件或目录、文件或目录改名 | mkdir | 创建目录 | | rm | 删除文件或目录 | rmdir | 删除空目录 | | ln | 建立链接文件 | tree | 显示目录树 |   **(2)文件目录命令操作举例**   |  |  | | --- | --- | | 命 令 | 说明 | | ls | 列表显示当前月录下的文件和目录 | | ls -a | 列表显示当前目录下的文件和目录(包括隐含文件和目录) | | ls -l | 以长格式列表显示结果 | | ls -R | 递归显示当前目录及其子目录下的文件和目录 | | ls -dl /usr/share/ | 仅显示/usr/share/目录本身，而非/usr/share/目录中的内容 | | pwd | 显示当前所在的工作路径 | | mkdir /home/osmond/mybin | 以绝对路径创建一个空目录 | | mkdir -p mydoc/FAQ | 以相对路径创建一个空目录树 | | mkdir -p /srv/{abc,bcd}/html | 创建srv/abc/html和/srv/bcd/html目录(注意{}的使用) | | touch abc bcd | 创建两个0字节文件 | | touch oldfile | 修改已存在文件的时间为当前时间 | | touch -r oldfile newfile | 参考oldfile文件的时问属性设置newfile文件的时间 | | cd | 切换到私有月录 | | cd mybin | 进入mybin目录 | | cd ~ | 切换到上一次使用cd命令前的目录 | | cd .. | 返回当前目录的上一级目录 | | cd ../.. | 返回当前目录的，上两级目录 | | tree | 显示当前目录下的目录结构 | | tree -L 3 /usr/ | 显示/usr目录下的三级目录树 | | cp /bin/?sh. | 使用“?”通配符复制多个文件到当前目录(.) | | cp http.conf{,.orig} | 将当前目录下的http.conf复制为http.conf.orig | | cp /bin/cpio mybin | 复制单个文件/bin/cpio到mybin目录 | | cp abc bcd mydoc | 将两个指定的文件复制到mydoc目录下 | | cp abc bcd ~mydoc | 将两个指定的文件复制到自己目录的mydoc子目录下 | | cp /usr/bin/[yz]\*. | 使用通配符“[]”和“\*”复制多个文件到当前目录(.) | | cp -r /etc/skel. | 将/etc/skel目录及其下面的所有内容复制到当前目录(.) | | mv FAQ bash-FAQ | 将当前目录下的FAQ文件或目录改名为bash-FAQ | | mv [yz]\* myusr/ | 将使用通配符“[]"和“\*”指定的多个文件移动到myusr目录下 | | rm myfile | 删除指定的文件 | | rm .\* | 删除当前目录下的所有隐含文件(隐含文件的文件名均以“.”开头) | | rm -f file{1,3,5} | 强制删除文件filel、file3和files | | rm -r myusr/ | 删除myusr目录及其内容(有删除提示) | | rm -rf myusr/ | 刪除myusr目录及其内容(强制利除，无删除提示) | | rmdir abc | 删除空目录abc | | In cpio edit1 | 建立cpio的硬链接文件editl | | In -s cpio edits1 | 建立cpio的符号链接文件edis1 | | In -s mydoc/FAQ/ FAQ | 对指定的目录mydoc/FAQ/创建符号链接文件FAQ | | find .-name 'my\*' | 从当前目录下开始查找以my开头的文件 | | find /home -user "osmond" | 从home目录下开始查找用户属主为osmond的文件 | | find .-type d -exec chmod 755{}\; | 将当前目录及其子目录下所有目录的权限改为755(目录属主可读可写可进入，同组人 和其他人员可读可进入) | | find .-type f -exec chmod 644{}\; | 将当前目录及其子目录下所有文件的权限改为644(文件属主可读可写可执行，同组人 和其他人员可读可执行) |   **注意：**  GNU/Linux的文件有如下3种类型的时间戳。   1. mtime: 最后修改时间(ls - lt)。 2. ctime: 状态改变时间(ls -lc). 3. atime: 最后访问时间ls -lu)。   **说明:**  (1) ctime并非文件创建时间。  (2)覆盖一个文件会改变所有3类时间: mtime、 ctime和atime.  (3)改变文件的访问权限或拥有者会改变文件的ctime和atime.  (4)读文件会改变文件的atime.  **详解：**  (1)ls  命令格式：ls [-alrtAFR][name…]  参数：  -a 显示所有文件及目录 (ls内定将文件名或目录名称开头为"."的视为隐藏档，不会列出)  -l 除文件名称外，亦将文件型态、权限、拥有者、文件大小等资讯详细列出  -r 将文件以相反次序显示(原定依英文字母次序)  -t 将文件依建立时间之先后次序列出  -A 同 -a ，但不列出 "." (目前目录) 及 ".." (父目录)  -F 在列出的文件名称后加一符号；例如可执行档则加 "\*", 目录则加 "/"  -R 若目录下有文件，则以下之文件亦皆依序列出  实例：  ls / 列出根目录下的所有目录    ls -ltr s\* 列出目前工作目录下所有名称是s开头的文件，越新的排越后面    ls -lR /bin      (2)pwd 用于显示工作目录，执行pwd指令可立刻得知您目前所在的工作目录的绝对路径名称  命令格式：pwd [-L][-R]  pwd 查看默认工作目录的完整路径    pwd查看指定文件夹    目录连接链接时，pwd -P 显示出实际路径，而非使用连接（link）路径；pwd显示地是连接路径      /bin/pwd [选项]  -L 目录连接链接时，输出连接路径  -R 输出物理路径    输出版本信息    (3) mkdir mkdir命令用于建立名称为 dirName 之子目录  命令格式：mkdir [-p] dirName  -p 确保目录名称存在，不存在就创建一个新的。  在工作目录下，建立一个名为ruanjian的子目录。    在工作目录下的ruanjian目录中，建立一个名为Test的子目录，若ruanjian目录原本不存在，则建立一个。（如不加-p，且原来ruanjian目录不存在，则产生错误）    不加-p的情况    (4) touch 用于修改文件或者目录的时间属性，包括存取时间和更改时间。若文件不存在，系统会建立一个新的文件。ls -l 可以显示档案的时间记录。  命令格式：touch [-acfm] [-d<日期时间>][-r<参考文件或目录>][-t<日期时间>][--help][--version][文件或目录…]  参数说明：  -a 改变档案的读取时间记录。  -m 改变档案的修改时间记录。  -c 假如目的档案不存在，不会建立新的档案。与 --no-create 的效果一样。  -f 不使用，是为了与其他 unix 系统的相容性而保留。  -r 使用参考档的时间记录，与 --file 的效果一样。  -d 设定时间与日期，可以使用各种不同的格式。  -t 设定档案的时间记录，格式与 date 指令相同。  --no-create 不会建立新档案。  --help 列出指令格式。  --version 列出版本讯息。  在工作目录下，创建mylinux文件：    使用touch命令一次创建多个文件:rzpt1,rzpt2,rzpt3    如果当前目录ruanjian已经存在，touch ruanjian会把ruanjian的last access time 改成当前时间。 强制避免使用touch命令创建新文件  更改文件的访问和修改时间 通过touch命令创建了一个名为“linuxmy”的文件，并通过stat命令查看了它的统计信息：    其中 touch -a linuxmy 命令仅更改此文件的访问时间。  通过此文件上的touch命令使用'-m'选项仅更改此文件的修改时间：  $ touch -m linuxmy  stat命令的输出现在显示修改时间已更改为使用'-m'选项运行touch命令的时间：    将访问和修改时间从一个文件复制到另一个文件    如果要将linuxa的访问和修改时间更改为linuxb的访问和修改时间，可以使用如下touch命令：  $ touch linuxa -r linuxb   使用指定的时间戳创建新文件 要创建具有指定时间戳的新空文件而不是创建它的实际时间，可以使用touch命令的以下语法：  $ touch -t YYMMDDHHMM.SS “filename”  以下示例显示了linuxa上的stat命令如何显示其访问和修改时间是基于通过touch命令创建时提供的时间戳：  创建具有指定时间戳的新文件  touch -t 2006151230.30 linuxa 将文件的时间戳更改为其他时间 您可以使用touch命令的以下语法将现有文件的时间戳更改为其他时间：  $ touch -c -t YYMMDDHHMM.SS “filename”  touch -c -t 2008191130.30 linuxa  在下面的示例中，通过touch命令更改了现有文件的时间戳，然后通过该示例文件上的stat命令验证了更改：    (5) cp  cp 命令主要用于复制文件或目录。即用来将一个或多个源文件或者目录复制到指定的目的文件或目录。  cp 命令可以将单个源文件复制成一个指定文件名的具体的文件或一个已经存在的目录下。  cp 命令支持同时复制多个文件，当一次复制多个文件时，目标文件参数必须是一个已经存在的目录，否则将出现错误。  命令格式：  cp [-options] source\_file dest\_file  source\_file：源文件，制定源文件列表。cp 命令默认情况下不能复制目录，除非使用 -r/-R选项。  dest\_file：目标文件，指定目标文件。当“源文件”为多个文件时，要求“目标文件”为指定的目录。  参数说明：  -a ：等于 “dpR” 参数组合，通常在复制目录时保留链接、文件属性，并复制目录下的所有内容。  -b：覆盖已存在的文件目标前将目标文件备份；  -d ：复制时保留符号链接。  -f ：强行复制文件或目录，覆盖已经存在的目标文件也给出提示。  -i ：与-f选项相反，在覆盖目标文件之前给出提示，要求用户确认是否覆盖。  -l ：不复制文件，只是生成源文件的硬链接文件。  -p ：复制文件时保留源文件或目录的属性。  -r ：递归复制目录，即复制该目录及其所有的子目录和文件。  -R ：同 -r 选项参数。  -s：对源文件建立符号连接，而非复制文件；  -S：在备份文件时，用指定的后缀“SUFFIX”代替文件的默认后缀；  -u：同步复制源文件文件，只在源文件更新时或目标文件不存在时复制。  -v：详细显示命令执行的操作。  复制文件：格式：cp source\_file dest\_file  cp 命令复制文件到目标文件，同名则会覆盖目标文件的内容，该目标文件的内容将被破坏。  cp 命令复制文件到目标文件，不同名则使用目标文件名，但目标文件内容变为源文件内容。  文件路径可以是绝对路径名，也可以是相对路径名。通常会用到当前目录.或上一级目录..的形式。    将家目录下的.bashrc复制到/tmp下，并更名为bashrc    如果/tmp下已经存在bashrc了，加上-i参数，则在覆盖前会询问使用者是否确定。    将/var/log/wtmp复制到/tmp下    注意：在不加任何参数的情况下，档案的所属者会改变，连权限也改变了。  如果要将档案的所有特性一起复制过来，可以加上-a    复制/etc/这个目录下的所有内容到/tmp下    注意：/etc是目录，不能直接复制，要加上-r参数  将bashrc建立一个连结档（symbolic link）    注意：那个bashrc\_ slink 是由-s 的参数造成的，建立的是一个[快捷方式J;所以我们会看到在档案的最右边，会显示这个档案是[连结]到哪里去的! 建立了这个档案之后，bashrc与bashrc\_ \_hlink所有的参数都一样，只是，第二栏的link数改变成为2了~而不是原本的1。  将bashrc\_slink复制成为bashrc\_slink\_2    将家目录的.bashrc及.bash\_history复制到/tmp下      （6）rmdir（remove directory）命令   可以使用rmdir 命令删除一个目录。必须离开目录，并且目录必须为空目录，不然提示删除失败。  命令格式：rmdir [-p] dirName  参数：  -p :是当子目录被删除后使它也成为空目录的话，则顺便一并删除。          在工作目录下的 LINUX目录中，删除名为LX的子目录。若 LX删除后，LINUX目录成为空目录，则 LINUX亦予删除。    同时删除两个目录：    (7)rm命令：删除文件或者目录  命令格式：rm [options] name…  参数：  -i 删除前逐一询问确认。  -f 即使原档案属性设为唯读，亦直接删除，无需逐一确认。  -r 将目录及以下之档案亦逐一删除。  删除文件可以直接使用rm命令，若删除目录则必须配合选项"-r"，例如：    删除当前目录下的所有文件及目录，命令行为：    文件一旦通过rm命令删除，则无法恢复，所以必须格外小心地使用该命令。  （8）mv命令：用来为文件或目录改名、或将文件或目录移入其它位置。  命令格式：  mv [options] source dest  mv [options] source… directory  参数说明：  -b ：若需覆盖文件，则覆盖前先行备份。  -f ：force 强制的意思，如果目标文件已经存在，不会询问而直接覆盖； -i ：若目标文件 (destination) 已经存在时，就会询问是否覆盖！ -u ：若目标文件已经存在，且 source 比较新，才会更新(update) -t  ： --target-directory=DIRECTORY move all SOURCE arguments into DIRECTORY，即指定mv的目标目录，该选项适用于移动多个源文件到一个目录的情况，此时目标目录在前，源文件在后。   |  |  | | --- | --- | | **命令格式** | **运行结果** | | mv 文件名 文件名 | 将源文件名改为目标文件名 | | mv 文件名 目录名 | 将文件移动到目标目录 | | mv 目录名 目录名 | 目标目录已存在，将源目录移动到目标目录；目标目录不存在则改名 | | mv 目录名 文件名 | 出错 |   比如：  例1：文件改名：mv file1 file2    例2：移动文件：mv file1 file2      例3：将多个文件移到到目录中去：mv filename dir1        例4：将testlinux.txt移动到rzpt目录中去：mv -t dirname filename      例5：将文件file1改名为file2，如果file2已经存在，则询问是否覆盖  mv -i file1 file2    例6：将文件file1改名为file2，如果file2已经存在，直接覆盖  mv -f file1 file2    例7：目录移动：mv dir1 dir2      例8：移动当前文件夹下的所有文件到上一级目录 mv \* ../    例9：把当前目录的一个子目录里的文件移动到另一个子目录里  mv test3/\*.txt test5    例10：文件被覆盖前做简单备份，前面加参数-b  mv log1.txt -b log2.txt    （9）find命令  find命令用来在指定目录下查找文件。任何位于参数之前的字符串都将被视为欲查找的目录名。如果使用该命令时，不设置任何参数，则find命令将在当前目录下查找子目录与文件。并且将查找到的子目录和文件全部进行显示。  命令格式：  find path -option [-print][-exec -ok command] {} \;  参数说明：  find 根据下列规则判断 path 和 expression，在命令列上第一个 - ( ) , ! 之前的部份为 path，之后的是 expression。如果 path 是空字串则使用目前路径，如果 expression 是空字串则使用 -print 为预设 expression。  expression 中可使用的选项有二三十个之多，在此只介绍最常用的部份。  -mount, -xdev : 只检查和指定目录在同一个文件系统下的文件，避免列出其它文件系统中的文件  -amin n : 在过去 n 分钟内被读取过  -anewer file : 比文件 file 更晚被读取过的文件  -atime n : 在过去n天内被读取过的文件  -cmin n : 在过去 n 分钟内被修改过  -cnewer file :比文件 file 更新的文件  -ctime n : 在过去n天内被修改过的文件  -empty : 空的文件-gid n or -group name : gid 是 n 或是 group 名称是 name  -ipath p, -path p : 路径名称符合 p 的文件，ipath 会忽略大小写  -name name, -iname name : 文件名称符合 name 的文件。iname 会忽略大小写  -size n : 文件大小 是 n 单位，b 代表 512 位元组的区块，c 表示字元数，k 表示 kilo bytes，w 是二个位元组。-type c : 文件类型是 c 的文件。  d: 目录  c: 字型装置文件  b: 区块装置文件  p: 具名贮列  f: 一般文件  l: 符号连结  s: socket  -pid n : process id 是 n 的文件  你可以使用 ( ) 将运算式分隔，并使用下列运算。  exp1 -and exp2  ! expr  -not expr  exp1 -or exp2  exp1, exp2  例1：将目前目录及其子目录下所有延伸档名是txt的文件列出来。  find . -name “\*.txt”    例2：将目前目录其其下子目录中所有一般文件列出。find . -type f    例3：将目前目录及其子目录下所有最近 20 天内更新过的文件列出。  find . -ctime -20    例4：查找/var/log目录中更改时间在7日以前的普通文件，并在删除之前询问它们：  find /var/log -type f -mtime +7 -ok rm {} \;    例5：查找前目录中文件属主具有读、写权限，并且文件所属组的用户和其他用户具有读权限的文件：  find . -type f -perm 644 -exec ls -l {} \;    例6：为了查找系统中所有文件长度为0的普通文件，并列出它们的完整路径：  find . -type f -size 0 -exec ls -l {} \;    （10）ln命令  ln命令是一个非常重要命令，它的功能是为某一个文件在另外一个位置建立一个同步的链接。  当我们需要在不同的目录，用到相同的文件时，我们不需要在每一个需要的目录下都放一个必须相同的文件，我们只要在某个固定的目录，放上该文件，然后在 其它的目录下用ln命令链接（link）它就可以，不必重复的占用磁盘空间。  命令格式：  ln [参数][源文件或目录][目标文件或目录]  参数格式为：[-bdfinsvF] [-S backup-suffix] [-V {numbered,existing,simple}]  [--help] [--version] [--]  命令功能:  Linux文件系统中，有所谓的链接(link)，我们可以将其视为档案的别名，而链接又可分为两种 : 硬链接(hard link)与软链接(symbolic link)，硬链接的意思是一个档案可以有多个名称，而软链接的方式则是产生一个特殊的档案，该档案的内容是指向另一个档案的位置。硬链接是存在同一个文件系统中，而软链接却可以跨越不同的文件系统。  不论是硬链接或软链接都不会将原本的档案复制一份，只会占用非常少量的磁碟空间。  **软链接**：  1.软链接，以路径的形式存在。类似于Windows操作系统中的快捷方式  2.软链接可以 跨文件系统 ，硬链接不可以  3.软链接可以对一个不存在的文件名进行链接  4.软链接可以对目录进行链接  **硬链接**：  1.硬链接，以文件副本的形式存在。但不占用实际空间。  2.不允许给目录创建硬链接  3.硬链接只有在同一个文件系统中才能创建 命令参数 **必要参数**：  -b 删除，覆盖以前建立的链接  -d 允许超级用户制作目录的硬链接  -f 强制执行  -i 交互模式，文件存在则提示用户是否覆盖  -n 把符号链接视为一般目录  -s 软链接(符号链接)  -v 显示详细的处理过程  **选择参数**：  -S "-S<字尾备份字符串> "或 "--suffix=<字尾备份字符串>"  -V "-V<备份方式>"或"--version-control=<备份方式>"  --help 显示帮助信息  --version 显示版本信息  例1：给文件创建软件链接，为log1.log文件创建软链接linklog1，如果log1.log丢失，linklog1将失效：  ln -s log1.log linklog1      例2：给文件创建硬链接，为log2013.log创建硬链接ln2013，log2013.log与ln2013的各项属性相同。  ln log1.log ln2020    （11）man命令  **man命令**是Linux下的帮助指令，通过man指令可以查看Linux中的指令帮助、配置文件帮助和编程帮助等信息。  命令格式：  man [选项][参数]  选项：  -a：在所有的man帮助手中搜索  -f：等价于whatis指令，显示给定关键字的简短描述信息  -P：指定内容时使用分布程序  -M：指定man手册搜索的路径  参数：  数字：指定从哪本man手册中搜索帮助  关键字：指定要搜索帮助的关键字  例1：man ls 在最左上角显示“LS（1）”，在这里，“LS”表示手册名称，而“（1）”表示该手册位于第一节章。  man ifconfig它会在最左上角显示“IFCONFIG（8）”。也可以这样输入命令：“man [章节号] 手册名称”。  例2：man是按照手册的章节号的顺序进行搜索的。  man sleep  man 3 sleep  （12）tree命令  Linux tree命令用于以树状图列出目录的内容。执行tree指令，它会列出指定目录下的所有文件，包括子目录里的文件。  命令格式：  tree [-aACdDfFgilnNpqstux][-I <范本样式>][-P <范本样式>][目录...]  参数说明：  -a 显示所有文件和目录。  -A 使用ASNI绘图字符显示树状图而非以ASCII字符组合。  -C 在文件和目录清单加上色彩，便于区分各种类型。  -d 显示目录名称而非内容。  -D 列出文件或目录的更改时间。  -f 在每个文件或目录之前，显示完整的相对路径名称。  -F 在执行文件，目录，Socket，符号连接，管道名称名称，各自加上"\*","/","=","@","|"号。  -g 列出文件或目录的所属群组名称，没有对应的名称时，则显示群组识别码。  -i 不以阶梯状列出文件或目录名称。  -I<范本样式> 不显示符合范本样式的文件或目录名称。  -l 如遇到性质为符号连接的目录，直接列出该连接所指向的原始目录。  -n 不在文件和目录清单加上色彩。  -N 直接列出文件和目录名称，包括控制字符。  -p 列出权限标示。  -P<范本样式> 只显示符合范本样式的文件或目录名称。  -q 用"?"号取代控制字符，列出文件和目录名称。  -s 列出文件或目录大小。  -t 用文件和目录的更改时间排序。  -u 列出文件或目录的拥有者名称，没有对应的名称时，则显示用户识别码。  -x 将范围局限在现行的文件系统中，若指定目录下的某些子目录，其存放于另一个文件系统上，则将该子目录予以排除在寻找范围外。  注：安装tree命令：yum -y install tree  **2、文本文件操作命令**  **（1）常用的文本文件操作命令**  下表中列出了一些常用的文本文件操作命令。   |  |  |  |  | | --- | --- | --- | --- | | 命 令 | 功 能 | 命 令 | 功 能 | | cat、tac | 显示文本文件内容 | diff | 显示两个文本文件的差异 | | more、less | 分页显示文本文件内容 | expand | 将文件中的制表符转换为空格 | | head、tail | 显示文本文件的前若千行或后若干行 | unexpand | 将文件中的空格转换为制表符 | | cut | 纵向切割出文本指定的部分 | dos2unix | 将DOS格式的文本转换成UNIX格式 | | paste | 纵向合并多个文本 | unix2dos | 将UNIX格式的文本转换成DOS格式 | | grep | 按关键字抽取匹配的行 | iconv | 将文本从--种编码转换成另--种编码 | | wc | 文本数据统计 | tr | 转换字符 | | sort | 以行为单位对文本文件接序 | sed | 流编辑器，通常用于非交互式的字符串替换 | | uniq | 删除文本文件中连续重复的行 | awk | awk是-种用于处理文本的编程语言工具，通常用于处理有格式的文本 |  1. cat命令用于连接文件并打印到标准输出设备上   命令格式：cat [-AbeEnsTuv] [--help][--version] filename  参数说明：  -n 或 --number：由 1 开始对所有输出的行数编号。  -b 或 --number-nonblank：和 -n 相似，只不过对于空白行不编号。  -s 或 --squeeze-blank：当遇到有连续两行以上的空白行，就代换为一行的空白行。  -v 或 --show-nonprinting：使用 ^ 和 M- 符号，除了 LFD 和 TAB 之外。  -E 或 --show-ends : 在每行结束处显示 $。  -T 或 --show-tabs: 将 TAB 字符显示为 ^I。  -A, --show-all：等价于 -vET。  -e：等价于"-vE"选项；  -t：等价于"-vT"选项；  例1：把textfile1的文档内容加上行号后输入到textfile2这个文档里。    例2：把 textfile1 和 textfile2 的文档内容加上行号（空白行不加）之后将内容附加到 textfile3 文档里：    例3：清空 /etc/test.txt 文档内容    例4：cat 也可以用来制作镜像文件。例如要制作软盘的镜像文件，将软盘放好后输入：    例5：相反的，如果想把 image file 写到软盘，输入     1. tac命令：查看文本文件，逆序显示      1. more命令   类似 cat ，不过会以一页一页的形式显示，更方便使用者逐页阅读，而最基本的指令就是按空白键（space）就往下一页显示，按 b 键就会往回（back）一页显示，而且还有搜寻字串的功能（与 vi 相似），使用中的说明文件，请按 h 。  命令格式：  more [-dlfpcsu] [-num][\*/pattern][+linenum][fileNames…]  参数说明：  -num 一次显示的行数  -d 提示使用者，在画面下方显示 [Press space to continue, 'q' to quit.] ，如果使用者按错键，则会显示 [Press 'h' for instructions.] 而不是 '哔' 声  -l 取消遇见特殊字元 ^L（送纸字元）时会暂停的功能  -f 计算行数时，以实际上的行数，而非自动换行过后的行数（有些单行字数太长的会被扩展为两行或两行以上）  -p 不以卷动的方式显示每一页，而是先清除萤幕后再显示内容  -c 跟 -p 相似，不同的是先显示内容再清除其他旧资料  -s 当遇到有连续两行以上的空白行，就代换为一行的空白行  -u 不显示下引号 （根据环境变数 TERM 指定的 terminal 而有所不同）  +/pattern 在每个文档显示前搜寻该字串（pattern），然后从该字串之后开始显示  +num 从第 num 行开始显示  fileNames 欲显示内容的文档，可为复数个数  例1：逐页显示 testfile 文档内容，如有连续两行以上空白行则以一行空白行显示。    例2：从第10行开始显示文档内容    **常用操作命令**  Enter 向下n行，需要定义。默认为1行  Ctrl+F 向下滚动一屏  空格键 向下滚动一屏  Ctrl+B 返回上一屏  = 输出当前行的行号  ：f 输出文件名和当前行的行号  V 调用vi编辑器  !命令 调用Shell，并执行命令  q 退出more   1. less命令   less 与 more 类似，但使用 less 可以随意浏览文件，而 more 仅能向前移动，却不能向后移动，而且 less 在查看之前不会加载整个文件。  命令格式：  less [参数] filename  参数说明：  -b <缓冲区大小> 设置缓冲区的大小  -e 当文件显示结束后，自动离开  -f 强迫打开特殊文件，例如外围设备代号、目录和二进制文件  -g 只标志最后搜索的关键词  -i 忽略搜索时的大小写  -m 显示类似more命令的百分比  -N 显示每行的行号  -o <文件名> 将less 输出的内容在指定文件中保存起来  -Q 不使用警告音  -s 显示连续空行为一行  -S 行过长时间将超出部分舍弃  -x <数字> 将"tab"键显示为规定的数字空格  /字符串：向下搜索"字符串"的功能  ?字符串：向上搜索"字符串"的功能  n：重复前一个搜索（与 / 或 ? 有关）  N：反向重复前一个搜索（与 / 或 ? 有关）  b 向上翻一页  d 向后翻半页  h 显示帮助界面  Q 退出less 命令  u 向前滚动半页  y 向前滚动一行  空格键 滚动一页  回车键 滚动一行  [pagedown]： 向下翻动一页  [pageup]： 向上翻动一页  例1：查看文件    例2：ps查看进程信息并通过less分页显示      例3：查看命令历史使用记录并通过less分页显示      例4：浏览多个文件    说明： 输入 ：n后，切换到 log2014.log 输入 ：p 后，切换到log2013.log  **附加备注**  全屏导航  ctrl + F - 向前移动一屏  ctrl + B - 向后移动一屏  ctrl + D - 向前移动半屏  ctrl + U - 向后移动半屏  单行导航  j - 向前移动一行  k - 向后移动一行  其它导航  G - 移动到最后一行  g - 移动到第一行  q / ZZ - 退出 less 命令  其它有用的命令  v - 使用配置的编辑器编辑当前文件  h - 显示 less 的帮助文档  &pattern - 仅显示匹配模式的行，而不是整个文件  标记导航  当使用 less 查看大文件时，可以在任何一个位置作标记，可以通过命令导航到标有特定标记的文本位置：  ma - 使用 a 标记文本的当前位置  'a - 导航到标记 a 处   1. head命令   head命令用于显示文件文字区块  命令格式：  Head [参数][文件]  参数说明：  -q 隐藏文件名  -v 显示文件名  -c<字节> 显示字节数  -n<行数> 显示的行数  例1：显示文件前5行文字    例2：显示文件前5个字节     1. tail命令   tail命令从指定点开始将文件写到标准输出。  命令格式：  tail [必要参数][选择参数][文件]  参数说明：  -f 循环读取  -q 不显示处理信息  -v 显示详细的处理信息  -c<数目> 显示的字节数  -n<行数> 显示行数  --pid=PID 与-f合用,表示在进程ID,PID死掉之后结束.  -q, --quiet, --silent 从不输出给出文件名的首部  -s, --sleep-interval=S 与-f合用,表示在每次反复的间隔休眠S秒  例1：显示文件末尾内容    例2：循环查看文件内容    例3：从第5行开始显示文件     1. cut命令   cut命令用于显示每行从开头算起 num1 到 num2 的文字。  命令格式：  cut [-bncdf][file]  参数说明：  -b ：以字节为单位进行分割。这些字节位置将忽略多字节字符边界，除非也指定了 -n 标志。  -c ：以字符为单位进行分割。  -d ：自定义分隔符，默认为制表符。  -f ：与-d一起使用，指定显示哪个区域。  -n ：取消分割多字节字符。仅和 -b 标志一起使用。如果字符的最后一个字节落在由 -b 标志的 List 参数指示的范围之内，该字符将被写出；否则，该字符将被排除  例1：取文件中的第2列的值。        因为1个汉字点3个字节，所以我们得不到第2个字节。    例2：取第几个字符（最后的列可以使用范围值）    例3：第1个字符到第n个字符（包含这个字符）    例4：第n个字符到最后一个字符（包含第n个字符）    例5：第1个字符到第n个字符及第n个字符到最后一个字符（第n个字符不重复）    例6：颠倒写，cut命令自动进行排序    例7：重复写，不重复执行    例8：用e做作为分隔符，然后在-f参数后面选择第1部分还是第2部分    例9：执行who命令，想提取每一行的第3个字节。       1. paste命令   paste 指令会把每个文件以列对列的方式，一列列地加以合并。  命令格式：  paste [-s][-d<间隔字符>][--help][--version][文件…]  参数说明：  -d<间隔字符>或--delimiters=<间隔字符> 　用指定的间隔字符取代跳格字符。  -s或--serial 　串列进行而非平行处理。  --help 　在线帮助。  --version 　显示帮助信息。  [文件…] 指定操作的文件路径  例1：   1. grep命令    grep 命令用于查找文件里符合条件的字符串。  命令格式：  grep [-abcEFGhHilLnqrsvVwxy] [-A<显示列数>][-B<显示列数>][-C<显示列数>][-d<进行动作>][-e<范本样式>][-f<范本文件>][--help][范本样式][文件或目录...]  参数说明：  **-a 或 --text** : 不要忽略二进制的数据。  **-A<显示行数> 或 --after-context=<显示行数>** : 除了显示符合范本样式的那一列之外，并显示该行之后的内容。  **-b 或 --byte-offset** : 在显示符合样式的那一行之前，标示出该行第一个字符的编号。  **-B<显示行数> 或 --before-context=<显示行数>** : 除了显示符合样式的那一行之外，并显示该行之前的内容。  **-c 或 --count** : 计算符合样式的列数。  **-C<显示行数> 或 --context=<显示行数>或-<显示行数>** : 除了显示符合样式的那一行之外，并显示该行之前后的内容。  **-d <动作> 或 --directories=<动作>** : 当指定要查找的是目录而非文件时，必须使用这项参数，否则grep指令将回报信息并停止动作。  **-e<范本样式> 或 --regexp=<范本样式>** : 指定字符串做为查找文件内容的样式。  **-E 或 --extended-regexp** : 将样式为延伸的正则表达式来使用。  **-f<规则文件> 或 --file=<规则文件>** : 指定规则文件，其内容含有一个或多个规则样式，让grep查找符合规则条件的文件内容，格式为每行一个规则样式。  **-F 或 --fixed-regexp** : 将样式视为固定字符串的列表。  **-G 或 --basic-regexp** : 将样式视为普通的表示法来使用。  **-h 或 --no-filename** : 在显示符合样式的那一行之前，不标示该行所属的文件名称。  **-H 或 --with-filename** : 在显示符合样式的那一行之前，表示该行所属的文件名称。  **-i 或 --ignore-case** : 忽略字符大小写的差别。  **-l 或 --file-with-matches** : 列出文件内容符合指定的样式的文件名称。  **-L 或 --files-without-match** : 列出文件内容不符合指定的样式的文件名称。  **-n 或 --line-number** : 在显示符合样式的那一行之前，标示出该行的列数编号。  **-o 或 --only-matching** : 只显示匹配PATTERN 部分。  **-q 或 --quiet或--silent** : 不显示任何信息。  **-r 或 --recursive** : 此参数的效果和指定"-d recurse"参数相同。  **-s 或 --no-messages** : 不显示错误信息。  **-v 或 --revert-match** : 显示不包含匹配文本的所有行。  **-V 或 --version** : 显示版本信息。  **-w 或 --word-regexp** : 只显示全字符合的列。  **-x --line-regexp** : 只显示全列符合的列。  **-y**: 此参数的效果和指定"-i"参数相同。   1. wc命令   wc命令用于计算字数。  命令格式：  wc [-clw][--help][--version][文件…]  参数说明：  -c或--bytes或--chars 只显示Bytes数。  -l或--lines 只显示行数。  -w或--words 只显示字数。  --help 在线帮助。  --version 显示版本信息。   1. uniq命令   uniq 命令用于检查及删除文本文件中重复出现的行列，一般与 sort 命令结合使用。  命令格式：  uniq [-cdu][-f<栏位>][-s<字符位置>][-w<字符位置>][--help][--version][输入文件][输出文件]  参数说明：  -c或--count 在每列旁边显示该行重复出现的次数。  -d或--repeated 仅显示重复出现的行列。  -f<栏位>或--skip-fields=<栏位> 忽略比较指定的栏位。  -s<字符位置>或--skip-chars=<字符位置> 忽略比较指定的字符。  -u或--unique 仅显示出一次的行列。  -w<字符位置>或--check-chars=<字符位置> 指定要比较的字符。  --help 显示帮助。  --version 显示版本信息。  [输入文件] 指定已排序好的文本文件。如果不指定此项，则从标准读取数据；  [输出文件] 指定输出的文件。如果不指定此选项，则将内容显示到标准输出设备（显示终端）。   1. diff命令   diff以逐行的方式，比较文本文件的异同处。如果指定要比较目录，则diff会比较目录中相同文件名的文件，但不会比较其中子目录。  命令格式：  diff [-abBcdefHilnNpPqrstTuvwy][-<行数>][-C <行数>][-D <巨集名称>][-I <字符或字符串>][-S <文件>][-W <宽度>][-x <文件或目录>][-X <文件>][--help][--left-column][--suppress-common-line][文件或目录1][文件或目录2]  参数说明：   -<行数> 　指定要显示多少行的文本。此参数必须与-c或-u参数一并使用。   -a或--text 　diff预设只会逐行比较文本文件。   -b或--ignore-space-change 　不检查空格字符的不同。  -B或--ignore-blank-lines 　不检查空白行。  -c 　显示全部内文，并标出不同之处。  -C<行数>或--context<行数> 　与执行"-c-<行数>"指令相同。  -d或--minimal 　使用不同的演算法，以较小的单位来做比较。  -D<巨集名称>或ifdef<巨集名称> 　此参数的输出格式可用于前置处理器巨集。  -e或--ed 　此参数的输出格式可用于ed的script文件。  -f或-forward-ed 　输出的格式类似ed的script文件，但按照原来文件的顺序来显示不同处。  -H或--speed-large-files 　比较大文件时，可加快速度。  -l<字符或字符串>或--ignore-matching-lines<字符或字符串> 　若两个文件在某几行有所不同，而这几行同时都包含了选项中指定的字符或字符串，则不显示这两个文件的差异。  -i或--ignore-case 　不检查大小写的不同。  -l或--paginate 　将结果交由pr程序来分页。  -n或--rcs 　将比较结果以RCS的格式来显示。  -N或--new-file 　在比较目录时，若文件A仅出现在某个目录中，预设会显示：  Only in目录：文件A若使用-N参数，则diff会将文件A与一个空白的文件比较。  -p 　若比较的文件为C语言的程序码文件时，显示差异所在的函数名称。  -P或--unidirectional-new-file 　与-N类似，但只有当第二个目录包含了一个第一个目录所没有的文件时，才会将这个文件与空白的文件做比较。  -q或--brief 　仅显示有无差异，不显示详细的信息。  -r或--recursive 　比较子目录中的文件。  -s或--report-identical-files 　若没有发现任何差异，仍然显示信息。  -S<文件>或--starting-file<文件> 　在比较目录时，从指定的文件开始比较。  -t或--expand-tabs 　在输出时，将tab字符展开。  -T或--initial-tab 　在每行前面加上tab字符以便对齐。  -u,-U<列数>或--unified=<列数> 　以合并的方式来显示文件内容的不同。  -v或--version 　显示版本信息。  -w或--ignore-all-space 　忽略全部的空格字符。  -W<宽度>或--width<宽度> 　在使用-y参数时，指定栏宽。  -x<文件名或目录>或--exclude<文件名或目录> 　不比较选项中所指定的文件或目录。  -X<文件>或--exclude-from<文件> 　您可以将文件或目录类型存成文本文件，然后在=<文件>中指定此文本文件。  -y或--side-by-side 　以并列的方式显示文件的异同之处。  --help 　显示帮助。  --left-column 　在使用-y参数时，若两个文件某一行内容相同，则仅在左侧的栏位显示该行内容。  --suppress-common-lines 　在使用-y参数时，仅显示不同之处。   1. expand命令 2. unexpand命令 3. dos2unix命令 4. unix2dos命令 5. iconv命令 6. tr命令 7. sed命令 8. awk命令   awk是一种处理文本文件的语言，是一个强大的文本分析工具。  命令格式：  awk [选项参数] `script` var=value file(s)  awk [选项参数]-f scriptfile var=value file(s)  参数说明：  -F fs or --field-separator fs 指定输入文件折分隔符，fs是一个字符串或者是一个正则表达式，如-F:。  -v var=value or --asign var=value 赋值一个用户定义变量。  -f scripfile or --file scriptfile 从脚本文件中读取awk命令。  -mf nnn and -mr nnn 对nnn值设置内在限制，-mf选项限制分配给nnn的最大块数目；-mr选项限制记录的最大数目。这两个功能是Bell实验室版awk的扩展功能，在标准awk中不适用。  -W compact or --compat, -W traditional or --traditional 在兼容模式下运行awk。所以gawk的行为和标准的awk完全一样，所有的awk扩展都被忽略。  -W copyleft or --copyleft, -W copyright or --copyright 打印简短的版权信息。  -W help or --help, -W usage or --usage 打印全部awk选项和每个选项的简短说明。  -W lint or --lint 打印不能向传统unix平台移植的结构的警告。  -W lint-old or --lint-old 打印关于不能向传统unix平台移植的结构的警告。  -W posix 打开兼容模式。但有以下限制，不识别：/x、函数关键字、func、换码序列以及当fs是一个空格时，将新行作为一个域分隔符；操作符\*\*和\*\*=不能代替^和^=；fflush无效。  -W re-interval or --re-inerval 允许间隔正则表达式的使用，参考(grep中的Posix字符类)，如括号表达式[[:alpha:]]。  -W source program-text or --source program-text 使用program-text作为源代码，可与-f命令混用。  -W version or --version 打印bug报告信息的版本。  **（2）正则表达式**  在许多文本处理工具(如grep、sed、awk、vi 等)中都可以使用正则表达式。正则表达式是使用某种模式(pattern)来匹配(matching) 一类字符串的一个公式。通常使用正则表达式进行查找、替换等操作。虽然复杂的正则表达式对于初学者来说晦涩难懂，但对于Linux使用者来说，学会使用正则表达式是非常必要的。在适当的情况下使用正则表达式可以极大地提高工作效率。POSIX 风格的正则表达式有两种:基本的正则表达式( Basic Regular  Expression, BRE)和扩展的正则表达式(Extended Regular Expression，ERE)。  正则表达式由一些普通字符和一些元字符(metacharacters) 组成。普通字符包括大小写的字母、数字(即所有非元字符),而元字符则具有特殊的含义。下面两表中分别列出了POSIX RE的元字符及其含义。   |  |  |  |  |  | | --- | --- | --- | --- | --- | | POSEXRE用于方括号之外的元字符 | | | | | | 特殊  字符 | 含义 | 类型 | 举例 | 说明 | | ^ | 匹配首字符 | BRE | ^x | 以字符x开始的字符串 | | $ | 匹配尾字符 | BRE | x$ | 以x字符结尾的字符串 | | · | 匹配任意一个字符 | BRE | l..e | love,life,live… | | ? | 匹配任意-一个可选字符 | ERE | xy? | x,xy | | \* | 匹配零次或多次重复 | BRE | xy\* | xy,xyy,xyyy… | | + | 匹配-次或多次重复 | ERE | xy+ | xy,xyy,xyyy… | | [...] | 匹配任意-一个字符 | BRE | [xyz] | x,y,z | | ( ) | 对正则表达式分组 | ERE | (xy)+ | xy,xyxy,xyxyxy, | | \{n\} | 匹配n次 | BRE. | co\{2\}gle | coogle | | \{n,\} | 匹配最少n次 | BRE | co\{2,\}gle | coogle,cooogle,coooogle… | | \{n,m\} | 匹配n~m次 | BRE | co\{2,4\)}gle | coogle,cooogle,coooogle | | {n} | 匹配n次 | ERE | co{2}gle | coogle | | {n,} | 匹配最少n次 | ERE | co{2,}g]e | coogle,cooogle,coooogle… | | {n,m} | 匹配n~m次 | ERE | c{2,4}gle | coogle,cooogle,coooogle | | | | 以或逻辑连接多个匹配 | ERE | good|bon | 匹配good或bon | | \ | 转义字符 | BRE | \\* | \* |  |  |  |  |  |  | | --- | --- | --- | --- | --- | | POSIX RE用于方括号之内的元字符 | | | | | | 特殊字符 | 含义 | 类型 | 举例 | 说明 | | ^ | 非（仅用于起始字符） | BRE | [^xyz] | 匹配xyz之外的任意一个字符 | | - | 用于指明字符范围（不能是首字符和尾字符） | BRE | [a-zA-Z] | 匹配任意一个字母 | | \ | 转义字符 | BRE | [\.] | . |   **（3）常用的文本文件操作命令举例**  下表列出了一些常用的文件目录操作命令的使用举例。   |  |  | | --- | --- | | 常用的文件目录操作命令使用举例 | | | 命 令 | 说 明 | | cat /etc/passwd | 滚屏显示文/et/passwd的内容 | | cat -n /etc/passwd | 滚屏显示文件/etc/passwd的内容并显示行号(等同于nl/et/passwd) | | more /etc/passwd | 分屏显示文件/etc/passwd的内容(注意<Space>键、<Enter>键和q的使用) | | more +10 /etc/passwd | 从第10行起分屏显示文/etc/passwd的内容 | | less /etc/passwd | 分屏显示文件epasswd的内容(注意<Space>键、(Enter)键、<PgDn>健、<PgUp>键和q的使用) | | head -4 /etc/passwd | 显示文件/etc/passwd前4行的内容 | | tail -4 /etc/passwd | 显示文件/etc/passwd后4行的内容 | | tail -n+10/te/passwd | 显示文件/epasswd从10行开始到文件尾的内容 | | tail -f/var/log/messages | 跟踪显示不断增长的文件结尾内容(通常用于显示日志文件) | | cut -f1,3-5 -d:/etc/passd | 以冒号作为问隔符显示/etc/passwd的第1、3、4、5列 | | paste mytxt.en mytxt.cn | 纵向合并文件mytxt.en和mytxt.cn | | wc myalllist | 统计指定文本文件的行数、字数、字符数 | | wc -l myalllist | 统计指定文本文件的行数 | | tr 'A-Z''a~z' mytxt | 将mytxt文件中的所有大写字母转换为小写字母显示在屏幕上 | | sort mytxt | 以行为单位对文本文件mytxt排序(以ASCII码顺序), | | sort -u mytxt | 以行为单位对文本文件mytxt排序(对相同的行只输出一-行) | | sort -r mytxt | 以行为单位对文本文件mytxt排序(以ASCII码逆序) | | sort -n mytxt | 以行为单位对文本文件mytxt排序(根据字符串的数值进行排序) | | grep my mybxt | 在文件mytxt中查找字符串my | | grep -i my mylist myalllist | 在多个指定的文件中查找字符串my(忽略大小写) | | grep -v "^#" /etc/grub.conf | 显示文件/etc/grub.conf除了以#开始行 | | grep -l root /etc/\* | 列出/etc目录下所有的内容包含字符串root的文件名 | | grep-lr root /etc/\* | 列出/etc目录包括子目录下所有的内容包含字符串root的文件名 | | diff httpd.conf httpd.conf.bak | 比较文件httpd.conf和httpd.conf.bak的差异 | | dos2unix -k \*.txt | 将当前目录下所有后缴为txt的文件转换为UNIX格式(不改变时间戳) | | dos2unix -k -n dosfile linuxfile | 将DOS格式的dosfile文本文件转化成UNIX格式的linuxfile | | iconv -f GB2312 -t UTF-8 -o outputfile inputfile | 将编码为GB2312的inputfile文件转化为UTTF-8编码的ouputfile | | sed 's/Windows/Linux/g' myfile | 将myfile中的所有Windows替换成Linux | | sed 's/cc\*/c/g' myfle | 将myfile中所有连续出现的c都压缩成单个的c | | sed 's^[\t]\*// ' myfile | 删除myfile中每一行前导的连续“空白字符”(空格，制表符) | | sed 's/ \*$/' myfile | 删除 myfile中每行结尾的所有空格 | | sed 's/'^>/' myfile | 在每一行开头加上一个尖括号和空格（引用信息） | | sed 's/'^>//' myfile | 将每一行开头处的尖括号和空格删除(解除引用) | | sed 's/.\*\∨// myfile | 删除路径前缀 | | sed '/^$/d' myfile | 删除所有空白行 | | awkF{print$1,S5}'/etc/passwd | 以分号为间隔符，列出/etc/passwd的第1列和第5列 |   **3、打包和压缩命令**  **（1）常用的打包（归档）和压缩命令**  用户经常需要把一-组文件 存储成-一个文件以便备份或传输到另"一个目录甚至另一台计算机。有时还需要把文件压缩成-一个文件，使得其占用少量的磁盘空间并能更快地通过互联网传输。下表中列出了一些常用的打包和压缩命令。   |  |  |  |  | | --- | --- | --- | --- | | 与打包和压缩相关的常用命令 | | | | | 命 令 | 功 能 | 命 令 | 功 能 | | gzip | 压缩(解压)文件或目录，压缩文件后级为好gz | zcat/zmore/  zless/zgrep | 不解压直接显示.gz文件的内容  不解压直接在.gz文件中查找指定的字符串 | | bzip2 | 压缩(解压)文件或目录，压缩文件后级为bz2 | bzcatbzmore/  bzless/bzgrep | 不解压直接显示.bz2文件的内容  不解压直接在bz2文件中查找指定的字符串 | | xz | 压缩(解压)文件或目录，压缩文件后缀为xz | xzcat/xzmore/xzless/xzgrep | 不解压直接显示xz文件的内容  不解压直接在.xz文件中查找指定的字符串 | | tar | 文件、目录打(解)包 |  |  |   **注意：**  **正确使用zcat/zmore/zless/zgrep 、bzcat/bzmore/bzless/bzgrep 、xzcat/xzmore/xzlss/xzgrep命令的前提是压缩前的文件是纯文本文件。**  **（2）常用的打包和压缩命令操作举例**   |  |  | | --- | --- | | 常用的压缩命令操作举例 | | | 命 令 | 说 明 | | gzip myfile | 压缩myfile文件生成myfile.gz并删除原始文件myfile | | gzip -l myfile.gz | 显示压缩文件myfile.gz的压缩信息 | | zless myfile.gz | 若myfile是文本文件，直接显示myfile.gz的内容 | | zgrep STRING myfile.gz | 在文件myfile.gz中查找包含STRING的行 | | gzip -d myfile.gz | 解压缩myfile.gz文件为myfile并删除压缩文件myfile.gz | | bzip2 myfile | 压缩myfile文件生成myfile.bz2并删除原始文件myfile | | bzip2 -k myfile | 压缩myfile文件生成myfile.bz2并保留原始文件myfile | | bzless myfile.b22 | 若myfile是文本文件，直接显示myfile.b22的内容 | | bzgrep STRING myfile.bz2 | 在文件myfle.bz2中查找包含STRING的行 | | bzip2 -d myfile.bz2 | 解压缩myfile.bz2文件为myfile并删除压缩文件myfile.bz2 | | bzip2 -dk myfile.bz2 | 解压缩myfile.bz22文件为myfile并保留压缩文件myfile.bcZ2 | | xz myfile | 压缩myfile文件生成myfile.xz并删除原始文件myfile | | xz -k myfile | 压缩myfile文件生成myfile.xz并保留原始文件myfile | | xz -l myfile.xz | 显示压缩文件myfile.xz的压缩信息 | | xzless myfile.xz | 若myfile是文本文件，直接显示myfile.xz的内容 | | xzgrep STRING myfile.xz | 在文件myfile.x中查找包含STRING的行 | | xz -d myfile.xz | 解压缩myfile.xz文件为myfile并删除压缩文件myfile.xz | | xz -dk myfile.xz | 解压缩myfile.xz文件为myfile并保留压缩文件myfile.xz |   在Linux环境下，通常使用GNU的tar 命令调用各个压缩软件实现打包压缩和解压缩。下表中列出了tar命令和压缩命令的操作举例。   |  |  | | --- | --- | | tar打包与压缩命令操作举例 | | | 命 令 | 说 明 | | tar -cvf myball.tar mydir | 将mydir目录打包为myball.tar文件 | | tar -tf myball.tar | 查看mybal.tar包中的内容 | | tar -xvf myball.tar | 将mybal,tar在当前目录下解包 | | tar -zcvf myball.tar.gz mydir | 将mydir目录打包后压缩(调用gzip压缩工具) | | tar -ztf myball.tar.gz | 查看myball.tar.gz包中的内容 | | tar -zxvf myball.tar.gz | 解压缩(调用gzip压缩工具) | | tar -jcvf myball.tar.bz2 mydir | .将mydir目录打包后压缩(调用bzip2压缩工具) | | tar -jtf mybal.tar.bz2 | 查看myball,tarbz2包中的内容 | | tar -jxvf myball.tar.bz2 | 解压缩(调用bzjp2压缩工具) | | tar -Jcvf myball.tar.xz mydir | 将mydir目录打包后压缩(调用xz压缩工具) | | tar -Jtf mybal.tar.xz | 查看mybal.tar.xz包中的内容 | | tar -Jxvf myball.tar.xz | 解压缩(调用xz压缩工具) |   **4、信息显示命令**  **（1）常用的信息显示命令**  下表列出了一些常用的信息显示命令。   |  |  |  |  | | --- | --- | --- | --- | | 命 令 | 功 能 | 命 令 | 功 能 | | stat | 显示指定文件的相关信息 | ps | 显示进程 | | file | 显示指定文件的类型 | pstree | 显示进程树 | | whereis | 查找系统文件所在路径 | top | 显示当前系统中耗费资源最多的进程 | | locale | 显示当前的语言环境 | uptime | 显示系统运行时间、用户数、平均负载 | | locate | 在updatedb库中查找文件名 | free | 显示当前内存和交换空间的使用情况 | | find | 在文件系统中查找匹配的文件 | du | 显示指定的文件或目录已占用的磁盘空间 | | who | 显示在线的登录用户 | df | 显示文件系统磁盘空间的使用情况 | | whoami | 显示用户自己的身份 | ifconfg | 显示网络接口信息 | | tty | 显示用户当前使用的终端 | route | 显示系统路由表 | | id | 显示当前用户的ID信息 | netstat | 显示网络状态信息 | | groups | 显示当前用户属于哪些组 | date | 显示当前日期 | | env | 显示当前用户可用的环境变量 | cal | 显示日历 |   **（2）常用的信息显示命令操作举例**  下表中列出了一些常用的信息显示命令的操作举例。大多数的信息显示命令无须使用任何参数，下表中仅举例说明需要带参数的命令。   |  |  | | --- | --- | | 常用的信息显示命令操作举例 | | | 命 令 | 说 明 | | whereis ls | 查找程序ls的位置 | | file /etc/passwd/bin/bash/dev/console | 显示指定文件的类型 | | stat /etc/passwd | 显示文件/etc/passwd的相关属性信息 | | ps aux | 查看所有进程(ps -ef) | | du -sh | 显示当前目录总的使用量(不显示目录中每个文件的使用量) | | date +%F\_%H-%M | 以特定格式显示日期时间(2015-04-01\_20-50) | | TZ=Europe/Paris date | 显示巴黎的当前时间 | | date -d "@2147483647" | 将UNIX纪元时间(从1970年1月1日开始的秒数)转换为date命令默认的输出格式 | | cal 2015; | 显示2015年的日历 | | cal 9 1752 | 显示1752年9月的日历(新历换旧历,有11天被去除) | | ifconfig eth0 | 显示网络接口eth0的信息 | | netstat -luntp | 查看所有监听端口 | | netstat -antp | 查看所有已经建立的连接 | | netstat -s | 查看网络统计信息 | | netstat -n | 显示内核路由表信息(route -n) | | cut -d : -fl /etc/passwd | 查看系统所有用户(awk-:'{print$1}'/ecpasswd) | | cut -d:- fl /etc/group | 查看系统所有组(awkF:'{print$I}'/etc/group) |   **5、文本编辑器vi/vim**  **（1）vi的简介**  Vi是Visual interface 的简称，可以执行输出、删除、查找、替换、块操作等众多文本操作，而且用户可以根据自己的需要对其进行定制，这是其他编辑程序所没有的功能。  Vi不是一个排版程序，不像MS Word 或WPS那样可以对字体、格式、段落等其他属性进行编排，它只是-一个文本编辑程序。  Vim是从 vi 发展出来的一个文本编辑器。代码补完、编译及错误跳转等方便编程的功能特别丰富，在程序员中被广泛使用。  简单的来说， vi 是老式的字处理器，不过功能已经很齐全了，但是还是有可以进步的地方。 vim 则可以说是程序开发者的一项很好用的工具。  vim 的官方网站 (http://www.vim.org) 自己也说vim 是一个程序开发工具而不是文字处理软件。  **（2）进入vi**  下表列出了进入vi文本编辑器的方式及说明。   |  |  | | --- | --- | | 进入Vi文本编辑器的方式 | | | 方 式 | 说 明 | | vi | 进入Vi的默认模式 | | vi filename | 打开新建文件filename，并将光标置于第一行首 | | vi +n filename | 打开文件flename，并将光标置于第n行首 | | vi +filename | 打开文件filename，并将光标置于最后一行首 | | vi +/pattern filename | 打开文件filename，并将光标置于第一个与pattern匹配的串处 | | vi -r filename | 打开上次用Vi编排时发生系统崩溃的文件flename,并恢复它 |   **(3)vi/vim的3种运行模式**  Vi有3种基本工作模式:普通(normal)模式、插入(insert) 模式和命令行(command-line 或Cmdline)模式，如下图所示。    进入Vi之后，首先进入的就是普通模式。进入普通模式后Vi 等待编辑命令输入而不是文本输入，也就是说这时输入的字母都将作为命令来解释。在普通(normal)模式里，可以输入所有的普通编辑命令。普通模式亦称为命令(command) 模式。  进入普通模式后光标停在屏幕第一行首位上(用\_表示)，其余各行的行首均有一个“~”符号，表示该行为空行。最后一行是状态行，显示出当前正在编辑的文件名及其状态。如果是[New File]， 则表示该文件是一个新建的文件。如果输入Vi之后带有文件名参数，文件已在系统中存在，则在屏幕上显示出该文件的内容，并且光标停在第一行的首 位，在状态行显示出该文件的文件名、行数和字符数。  在普通模式下输入插入命令i、附加命令a、打开命令0、修改命令C、取代命令r或替换命令s都可以进入插入模式。在插入模式下，用户输入的任何字符都被Vi当作文件内容保存起来，并将其显示在屏幕上。在文本输入过程中，若想回到命令行模式下，按<Esc> 键即可。  在普通模式下，执行ex命令使用“:",查找使用“?”和“/”,调用Shell命令使用“!”。多数文件管理命令都是在此模式下执行的。末行命令执行完后，Vi 自动回到普通模式。  若在命令行模式下输入命令过程中改变了主意，可用<Back space>键将输入的命令全部删除，之后再按一下(Back space) 键，即可使Vi回到普通模式。  **（4）普通模式下的操作**  下面的各个表分别列出了普通模式下的几类命令。   |  |  | | --- | --- | | 普通模式下进入插入模式 | | | 命令 | 说 明 | | i | 从光标所在位置前开始插入文本 | | I | 将光标移到当前行的行首，然后在其前插入文本 | | a | 用于在光标当前所在位置之后追加新文本 | | A | 将光标挪到所在行的行尾，从那里开始插入新文本 | | o | 将在光标所在行的下面新开一行,并将光标置于该行的行首,等待输入文本 | | O | 在光标所在行的上而插入一行,并将光标竖于该行的行首,等待输入文本 |  |  |  |  |  | | --- | --- | --- | --- | | 普通模式下的光标定位 | | | | | 命令 | 说 明 | 命令 | 说 明 | | G | 将光标移至最后一行行首 | $ | 移动到光标所在行的行尾 | | nG | 光标移至第n行首 | ^. | 移动到光标所在行的第-一个字符(非空格) | | n+ | 光标下移n行 | h,j,k,l | 分别用于光标左移、下移、上移、右移一个字符 | | n- | 光标上移n行 | H | 将光标移至当前屏幕首行的行首(即左上角) | | n$ | 光标移至第n行尾 | M | 将光标移至屏幕显示文件的中间行的行首 | | 0 | 移动到光标所在行的行首 | L | 将光标移至当前屏幕的最底行的行首 |  |  |  |  |  | | --- | --- | --- | --- | | 普通模式下的替换和删除 | | | | | 命令 | 说明 | 命令 | 说明 | | rc | 用字符c替换光标所指向的当前字符 | nrc | 用字符c替换光标所指向的前n个字符 | | x | 删除光标处的字符 | nx | 删除从光标位置开始向右的n个字符 | | dw | 删除一个单词 | ndw | 删除n个指定的单词 | | db | 删除光标所在位置之前的一个词 | ndb | 删除光标所在位置之前的n个词 | | dd | 删除光标所在的整行 | ndd | 删除当前行及其后n-l行的内容 | | dG | 删除光标位置到最后一行的所有内容 | dIG | 删除光标所在位置到第一行的所有内容 | | d$ | 删除光标位置到当前行的末尾的内容 | d0 | 删除光标位置到当前行的开始的内容 |  |  |  | | --- | --- | | **普通模式下的复制和粘贴** | | | 命令 | 说明 | | yy | 将当前行的内容复制到缓冲区 | | nyy | 将当前开始的n行内容复制到缓冲区 | | yG | 将当前光标位置到最后--行的所有内容复制到缓冲区 | | yIG | 将当前光标位置到第一行的所有内容复制到缓冲区 | | y$ | 将当前光标位置到当前行末尾的内容复制到缓冲区 | | y0 | 将当前光标位置到当前行开始的内容复制到缓冲区 | | P | 将缓冲区的内容写出到光标所在的位置 |  |  |  |  |  | | --- | --- | --- | --- | | 普通模式下的字符串搜索 | | | | | 命令 | 说明 | 命令 | 说明 | | /str | 往右移动到有str的地方 | n | 向相同的方向移动到有str的地方 | | ?str | 往左移动到有str的地方 | N | 向相反的方向移动到有str的地方 |  |  |  |  |  | | --- | --- | --- | --- | | **普通模式下的撤销和重复** | | | | | 命令 | 说 明 | | | | u | 取消前一次的误操作或不合适的操作对文件造成的影响，使之恢复到这种误操作或不合适操作被执行之前的状态 | | | | . | 再执行一次前面刚完成的某个命令 | | | | 在普通模式下退出Vi | | | | | 命令 | 说 明 | 命令 | 说 明 | | ZZ | 存盘退出 | ZQ | 不保存退出 |   **（5）命令模式下的操作**  下列各表中列出了在命令模式下的常用命令。   |  |  | | --- | --- | | 在命令模式下跳行 | | | 命令 | 说 明 | | :n | 直接输入要移动到的行号即可实现跳行 |  |  |  | | --- | --- | | 在命令模式下搜索和替换字符串 | | | 命 令 | 说 明 | | :/str/ | 从当前光标开始往右移动到有sr的地方 | | ?str? | 从当前光标开始往左移动到有str的地方 | | :/str/w file | 将包含有str的行写到文件file中 | | :/str1/str2/w file | 将从strl开始到st2结束的内容写入f.le文件中 | | :s/str1/str2 | 将找到的第1个str1替换为str2 | | :s/str1/str2/g | 将找到的所有str1替换为str2 | | :n1,n2s/str1/st2g | 将从n1行到n2行找到的所有的srl替换为str2 | | :1,s/str1/str2/g | 将从第I行到当前位置的所有的srl替换为st2 | | :sSs/str1/str2/g | 将从当前位置到结尾的所有的str1替换为stur2 | | :1,$s/str1/str2/gc | 将从第1行到最后一行的所有的str1替换为stur2,并在替换前询问 |  |  |  | | --- | --- | | 在命令模式下复制、移动和删除文件行(块) | | | 命 令 | 说 明 | | :nl,n2 co n3 | 将从n1开始到n2为止的所有内容复制到n3后面 | | :n1,n2 m n3 | 将从n1开始到n2为止的所有内容移动到n3后面 | | :d | 删除当前行 | | :nd | 删除从当前行开始的n行 | | :n1,n2d | 删除从n1开始到n2为止的所有内容 | | :.,$d | 删除从当前行到结尾的所有内容 | | :/str1/,/str2/d | 删除从str1开始到str2为止的所有内容 |  |  |  | | --- | --- | |  | | | 在命令模式下的文件相关命令 | | | 命 令 | 说 明 | | :w | 将当前编辑的内容存盘 | | :w file | 将当前编辑的内容写到file文件中 | | :n1,n2w file | 将从n1开始到n2结束的行写到file文件中 | | :nw file | 将第n行写到file文件中 | | :1,.w file | 将从第1行起到光标当前位置的所有内容写到file文件中 | | :.,$w file | 将从光标当前位置起到文件结尾的所有内容写到file文件中 | | :r file | 打开另一个文件file | | :e file | 新建file文件 | | :f file | 把当前文件改名为file文件 |  |  |  | | --- | --- | | 在命令模式下执行Shell命令 | | | 命 令 | 说 明 | | :!Cmd | 运行Shell命令Cmd | | :n1,n2 w ! Cmd | 将n1到n2行的内容作为Cmd命令的输入，如果不指定n1和n2,则将整个文件的内容作为命令Cmd的输入 | | :r! Cmd | 将命令运行的结果写入当前行位置 |  |  |  | | --- | --- | | 在命令模式下设置Vi环境 | | | 命令 | 说 明 | | :set autoindent | 缩进每一行，使之与前一行相同。常用于程序的编写 | | :set noautoindent | 取消缩进 | | :set number | 在编辑文件时显示行号 | | :set nonumber | 不显示行号 | | :set ruler | 在屏幕底部显示光标所在的行、列位置 | | :set noruler | 不显示光标所在的行、列位置 | | :set tabsto=value | 设置显示制表符的空格字符个数 | | set wrapmargin=value | 设置显示器的右页边。当输入进入所设置的页边时，编辑器自动回车换行 | | :set | 显示设置的所有选项 | | :set all | 显示所有可以设置的选项 |  |  |  | | --- | --- | | 在命令模式下退出Vi | | | 命 令 | 说 明 | | :q | 退出Vi | | :wq | 保存退出Vi | | :q! | 不保存退出Vi | | | | |
| 知识深化 |  | | |
| 归纳总结 |  | | |
| 课堂作业 | 1. 浏览并熟悉Linux目录结构 2. 学会使用命令帮助 3. 熟悉各种常用命令的使用 4. 熟悉文本编辑器VIM的使用 | | |
| 课后作业 | (1)简述文件的类型。硬链接和软链接有何区别?  (2)简述Linux的标准目录结构及其存放内容。  (3) Linux 的基本命令格式如何? Linux下经常使用的通配符有哪些?  (4)在Linux下如何使用设备?常用的设备名有哪些?  (5)常用的文件和目录操作命令有哪些?各自的功能是什么?  (6)常用的信息显示命令有哪些?各自的功能是什么?  (7)打包和压缩有何不同?常用的打包和压缩命令有哪些?  (8)比较下面命令的含义   |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | | cd | CP | m | touch | alias | logout | whoami | cat | locale | find | | cd- | In | rm -r | mkdir | history | exit | who am i |  | locate | grep | | cd... | In -s | rmdir |  |  |  | who | less |  |  | | cd~ |  |  |  |  |  | uname | head |  |  | | pwd |  |  |  |  |  | hostname | tail |  |  | | | |
| 教学体会 |  | | |