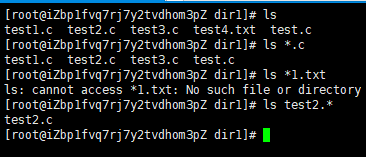
##### 前节只知识回顾

1. 创建目录的命令是mkdir，创建普通文件的命令是touch
2. 删除删除空白文件夹的命令是rmdir，删除文件的命令是rm
3. 删除带有内容的文件夹的命令（当然也能删除空白的文件夹）rm -r

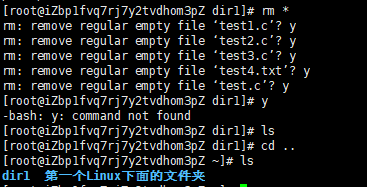
##### \*

//这个 \* 是通配符的意思，也就是类似于模糊查询的意思

1. 使用\*查找相关的文件

ps：类似于模糊查询，什么都能匹配上

1. 使用\*删除相关文件（或者文件夹）

ps：可以配上-f，这样就可以全部没有提示的进行删除

##### man指令

//这个指令就是一个帮助手册，但是是英文版的（可汉化）。具体的使用方式都是类似的，如下所示。 man里面有8个手册

1. man ls man rm -f 都是这样的方式
2. 甚至这个man还能查询c语言中的函数关键字怎么用
3. 使用完这个手册可以直接输入q退出；可以ctrl+z挂到后台，输入指令fg返回手册，也可以输入bg命令放在后台运行
4. 想要从指定的手册里面进行查询的时候可以这样 man 1 ls :从指定的1号手册查询ls指令怎么用

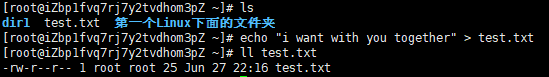
##### echo 指令

//echo ：回声

//这个指令能够将后面跟随的字符串打印出来

没有次选项的情况下默认向屏幕上面输出

1. echo 字符串 > 文件名字 ：向指定的文件输入进去这个字符串



明显的能够看出来输入了25字节的内容到这个文本文件中

ps：请一定要带上这个 > 才能写入到这个文本文件中去。这个 > 叫做输出重定向符号

1. 如果想要写入的这个文件已经存在内容了，那么会将原先的内容全部删除，然后将后面的内容进行写入文件

//ps: 也就是说，在写入新的内容之前会将原来的内容先清空，简单的理解就是覆盖。

3.对于键盘和显示器，可以将显示器和键盘也看做文件，既然是文件就有这读和写的方式

##### cat 指令

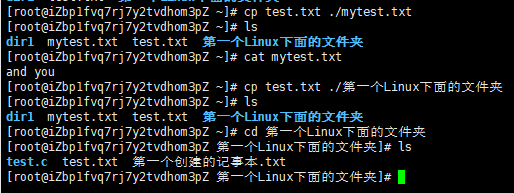
//concatenate：连接 这个指令会将文件中的内容打印出来

1.常见的使用方式就是 cat 带输出的文件名字

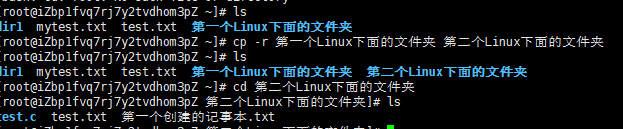
##### cp指令

//这个指令用来对文件或者文件夹进行拷贝，拷贝有不同的使用方法  
 1.cp 待拷贝的文件 空白文件/文件夹路径/空白文件路径 -----如果目标文件如果已经存在内容，那么待拷贝文件中的内容就会将目标文件中的内容覆盖

//待拷贝文件和目标文件:是将待拷贝文件中的内容复制到目标文件中 ps:这个是为了自己区分两个文件的描述,不是官方的定义

ps:只有一个 cp 是只能复制单个的文件

1. cp -r :这个选项是拷贝一个文件夹到一个新的或者老的文件夹中去 --------r选项一般都是递归的意思，也就是挨个进行.



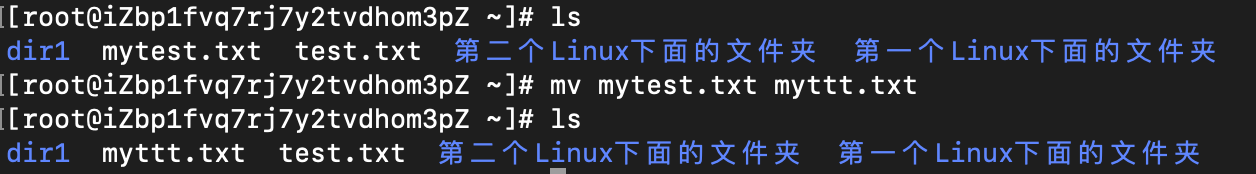
3.cp -i : 这个选项是针对普通用户的，普通用户直接进行拷贝是会进行提示的，带上这个就不会进行提示了，可以和上面的-i进行合并

##### mv 指令

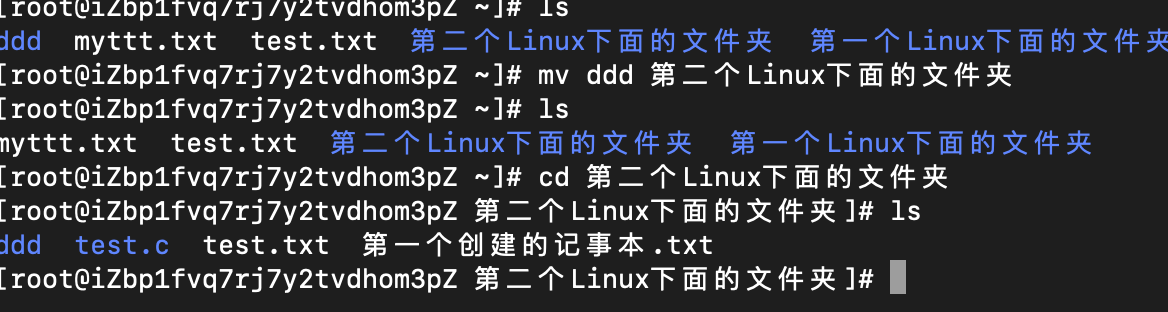
//这个执行是用来进行移动文件、文件夹或者对文件、文件夹重命名的 ----剪切也就是相当于移动

//mv--move:移动

1.mv ：直接使用的时候能够对文件或者文件夹进行重命名 ---这是mv指令的第一种使用的方法



2. mv :如果对两个目录进行操作的情况下，会将前面的目录放入到后面的目录当中去 ----mv指令的第二种用法，等价于win系统中的剪切或者移动。

ps:也可以使用相对或者绝对路径来进行各种剪切的操作。

3.mv 文件 路径 --这种就是将这个文件剪切到指定的路径 mv 文件 路径/文件 ----这种就是剪切并且重命名（进行两个操作）

ps ： 其实可以将所有的mv的行为看作剪切并重命名 也就是 mv 文件 ./文件 还是当前的相同的文件

4.mv 目录 上级目录：这种情况比较特殊，如果想要剪切到上级目录当中并且重新命名，但是上级目录中已经存在想要取的名字，那么就会将这个文件夹放入到指定名字的文件夹内而非重命名。

##### 制作一个简单的Linux下面的垃圾桶（这里只是简单的提一下，掌握了足够的知识就可以着手进行）

//前面的知识讲过了，Linux下面没有所谓的回收站

//回收站简单的来说就是一个个目录，win上面的垃圾桶就是将目录进行了可视化

//基本的原理就是将rm这个删除的命令进行重命名成为一个mv，只是使用了重命名的mv只会将删除的文件夹移动到指定的文件夹里面（而这个指定的文件夹的名字就可以叫做垃圾桶）

##### 怎么看待指令

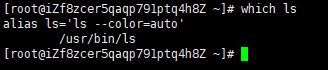
1. 指令本质上就是文件。只是这个文件是可执行的。这个文件只是安装到了指定的目录下了
2. 在输入一个指令以后，系统首先会查找这个指令是不是存在的
3. 这个指令肯定会有存放的位置，这个存放的位置就是/usr/bin这个路径下面 这个usr目录的等级是和root以及home 是同一个等级的。home文件夹里面保存的是用户的信息。



1. 既然是可执行程序，自己编写的源文件编译之后也是可执行程序，同样可以进行重命名，将这个编译好的可执行文件放在这个目录的下面，就可以像使用普通指令一样打开这个可执行程序。（这样就是安装软件的本质） ps：但是不要随意的将自己的可执行程序移动到/usr/bin这个路径下面。可能会出现意料之外的错误。在这个bin目录下删除某些文件（就是软件卸载的本质）

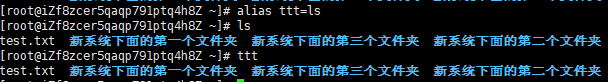
##### which 指令

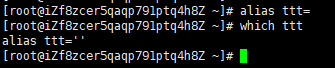
//which指令能够查看指令所在的目录

 //ps: alias：别名

##### alias 指令

//这个指令可以用来进行重命名。

1.alias 自己定义的名字=原本系统的指令 //这样就能进行系统指令的命名  
 

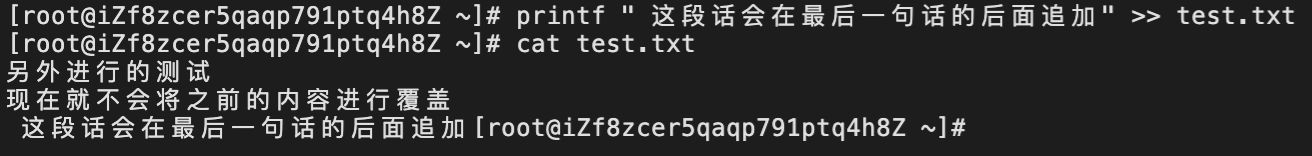
2.alias 刚才自己定义的名字=空格 //这样就再输入刚才的指令就是无效的  
 

3.这样对指令的重命名只是暂时的,只要退出系统就会重新恢复。想要永久的进行重命名，后续会学习

##### 输出重定向的其他用法 1.使用 >> 就不会讲之前的文件中的内容进行覆盖 ps:这样的重定向就叫做追加重定向

ps：无论是输出重定向还是追加重定向，如果后面的文件名不存在，会创建出来一个这个名字的文件，也可以利用这个特点 >+文件名来创建文件

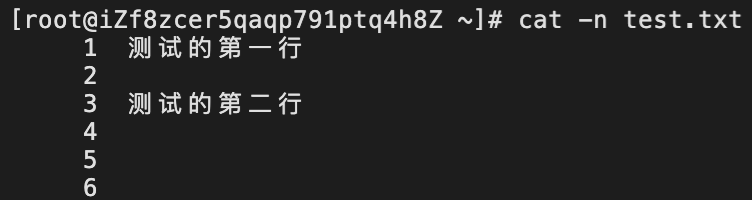
2.echo将内容进行写入的时候会自动进行换行，使用 printf 进行写入就不会出现换行的情况。



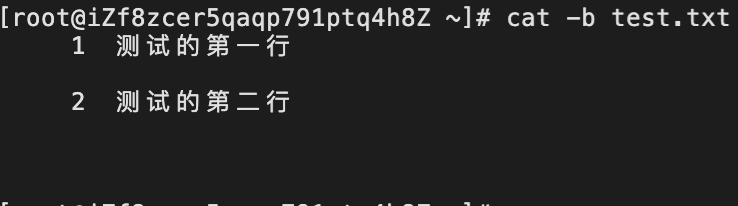
##### cat 命令的补充

1.如果使用 cat 命令的时候后面不指定文件名字，则默认是从键盘进行读取，默认是从显示器输出。

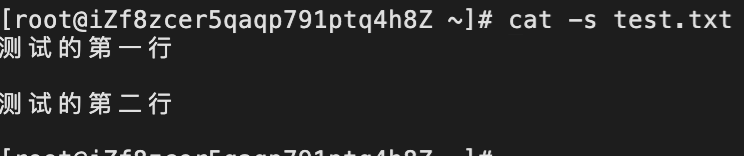
2. cat -n :这个指令能够将每行的行号标出来，即使是空行（回车）也会进行编号。 ps：这个是进行编号的指令



3. cat -b :只会对不是空行的进行编号，但是依旧会将所有的空行进行输出。 ps: 这个也是进行编号的指令



4.cat -s :输出的内容如果包括多个空行，那么只会输出一个空行。 ps：这个是只管输出的，并不进行编号

ps:这些指令同样能够进行简单的组合 sb nb

##### 输入重定向的用法

1。暂时就一个简单的例子。后面的学习会将补充 < :输入重定向的符号



##### more 指令

//这个指令和cat指令是类似的，都是将文件中的内容输出到屏幕上面，并且这个指令是针对文件较大的情况。

1.cat指令的缺点是不能查看较多行的文件，如果行比较多，输出的时候，查看具体的某一行是比较麻烦的。

2.直接使用more指令的时候会先输出一部分，输出了多少下面会有百分比进行提示.按回车可以进行下翻查看。 ps：退出的时候按q

3.more -数字 ：屏幕上面显示的就是截止到指定行数，就会停止，想要查看指定行下面的还可以按回车进行下翻查看

4.在使用more -数字 输出的时候，在下面的行输入 /指定行数 会直接跳转到相应的行数。

5.more指令只能向下翻页

##### less 指令 ps：需要对照课件进行补充

//基本的使用方法和more类似，但是可以使用pageup pagedown向上向下翻页，也可以使用查询的功能，查询是搜索关键字的，关键字可以是数字，可以是英文，也可以是汉字

1. / :向下搜索出现的字符串
2. ？ ：向上搜索出现过的字符串
3. -i :忽略搜索时候的大小写 //ps:这两个是搭配/ ？ 来进行搜索的
4. -N :显示每行的行号
5. n ：重复前一个搜索
6. N ：反向重复前一个搜索
7. q ：搜索之后按下q键退出

##### head 指令

//这个指令只会将文件的前n行打印出来（一般情况下是10行），

1. head -数字 文件名字：只显示出来指定的这么多行，数字为8就显示8行，为100，就只显示100行。

##### tail 指令

//tail：尾巴，末尾的意思。 使用这个指令只会显示末尾的几行。

1.tail -数字 文件名字 ： 数字是多少就只显示最后的几行

//ps:可以使用tail指令和head指令进行截取代码的片段

##### 管道文件 | 的使用

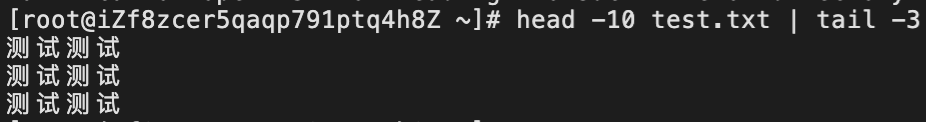
// | 就是一个符号，也就代表着这回成为管道文件 ps：管道文件并不是真正在磁盘上存储的，管道文件是在内存中存储的。

1.比如右边的这条指令 head -100 test.txt | tail 20 : |左边读取出来的内容会交个|这个符号（这个符号就是管道文件），然后右侧的文件会从管道文件中读取文件。

2.关于上面的这条指令的解释

（1）管道，顾名思义就是有头有尾链接两端的一个桥梁

（2）| 左侧的head -100 test.txt的内容会从这个管道的头进去，然后从另一端流出来，将前面的内容的后20行进行输出到屏幕上面



3。wc -l :这条指令使用来统计文本行的个数的



4. | 在一条指令当中可以有多个 xxxx|xxxx|xxxx|xxxx

5.管道的大小是一定的。

6. yum install -y nano 安装nano指令