# **常规命令**

## cp 命令，复制文件和目录

### 参数

**cp -R dir1 dir2 =====使用-R来复制目录，-a相当于-dpR,功能更加强大，所以复制目录更多的用-a；**

**--p ====复制时连同属性一起复制**

**--f====强制复制，忽略目的文件是否存在，如果存在，则强制替换**

### 描述

**这个命令包括一些有趣的功能，比如在复制文件时可以使用backup参数来备份目的文件夹中的同名的文件；可以使用n参数来判断如果目的文件夹中有重名的文件，则不进行复制；可以使用f参数来强制复制，无论目的文件夹中是否有重名文件。**

### 核心内容

**这个命令的关键要素其一在复制过程中防止文件被重新覆盖，所以有一些参数来控制这种操作；**

**关键之二是复制目录时，是全部复制还是通过筛选来复制，只复制指定的文件，这个命令通过使用p参数和d参数等来判断是否要连带符号连接文件一起复制；**

**关键之三在于复制文件时是仅仅复制内容，还是连同文件属性（包括用户组，所有者，模式，文件时间）一同复制，或是仅仅复制部分，这些都可以通过参数--parents来指定**

# **Help 用来处理bash的内置命令**

## 参数

**Help 内置命令 #查看内置命令的帮助**

**比如，cd是内置命令，它的使用说明要用help cd 来完成，而不是man来查看，使用man只会跳转到所有内置命令的说明上；**

**Help \*string # 查询某个命令的说明，或者匹配多个命令的说明。**

## 描述

**使用help 内置命令 来查看内置命令的使用帮助**

**Linux命令分为内置命令和外部命令，内置命令又称为bash命令，它因为在系统加载过程中立即被存放在内存中，所以常驻内存，执行效率高；而外部命令需要从硬盘中读入；**

**内置命令包括像cd、exit，bg,command等类似的命令，它只能使用help来查看说明，不能使用man来查看**

## 核心内容

**常用的内置命令如下：**

**alias, bg, bind, break, builtin, caller, cd, command, compgen, complete,**

**compopt, continue, declare, dirs, disown, echo, enable, eval, exec, exit, export, false,fc, fg, getopts, hash, help, history, jobs, kill, let, local, logout, mapfile, popd, printf, pushd, pwd, read, readonly, return, set, shift, shopt, source, suspend, test, times, trap, true, type, typeset, ulimit, umask, unalias, unset, wait**

## 扩展内容

### Enable命令

**参数：**

Enable -n command #关闭指定的内建命令

Enable command # 打开

Enable -a #列出所有内建命令

### 描述：

**这个命令用来打开或者关闭shell内建命令，默认情况下，在命令被执行时，先从shell内建命令中寻找该命令，然后再从$path中寻找；**

### 核心内容：

**这个命令存在的关键在于，当用户想要自定义命令时，如果自定义的命令和内建命令重名，可以暂时关闭内建命令，以腾出命名空间，在调用自定义命令后再打开内建命令。**

# **rm命令**

## 参数

**Rm -f #来强制执行，不管文件是什么类型**

**Rm-r #删除目录**

## 描述

**这个命令用来删除文件或目录。删除文件时文件所在的目录是可以被恢复的，如果要删除目录并且不可恢复，请使用shred；**

**注意：使用rm命令前一定要先备份，谨慎使用这个命令**

**替代命令：使用mv来移动到临时文件中，或者使用find删除**

**用来删除目录时，默认是不会删除根目录（/），可以使用特殊参数指定忽略根目录,--no-preserve-root来忽略根目录；**

### 核心内容

**这个命令的关键在于它是一个非常危险的命令，在应用过程中尽可能多的指定参数，进行外科手术式的精确删除，否则后果很严重；而且这里使用的rm是GNU/LINUX下的rm，提供了非常丰富的参数来保证删除的安全性，但尽可能还是少用这个命令，使用上面提到的mv或者是find来进行更精确，安全的删除。**

# **cat命令的一些特殊用法**

## 参数：

**Cat >> 1.txt << EOF**

**STRING**

**STRING**

**...........**

**EOF**

**这个命令可以将1.txt打开进行编辑，编辑的内容是EOF标签之间的内容。因为cat>>file或者cat>file用来创建一个文件**

**Cat file1 - file 2 可以先输出文件1的内容，然后是标准输出，然后是file2的内容；（不知道如何结束标准输出）**

## 描述：

**这个命令用来查看文件中的内容，有几个有用的属性用来让内容显示更加标准和规范，比如添加行号，去掉空行，去掉tab字符等。还可以同时打开多个文件内容，顺序就按照文件的打开顺序。**

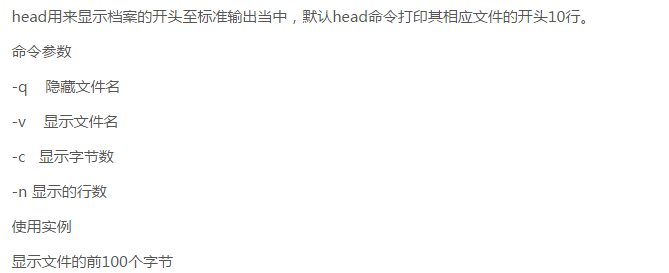
## 核心内容：

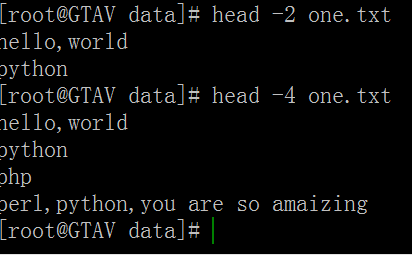
**这个命令的关键不是用来查看文件内容，而是作为标准输入的一部分，因为使用vi可以更有效的查看，而作为标准输入的一部分，使用这个命令更好，而且它提供了一系列参数用来显示更规范的输出。它更多的是和>>,>,<<,<结合，或者结合管道符号来完成后续的事情；**

# **一头一尾：head命令和tail命令**

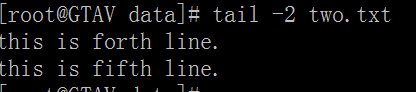
**这个命令用来读取文件的指定内容，默认取前10行的内容。**

**格式为：head n filename**

****

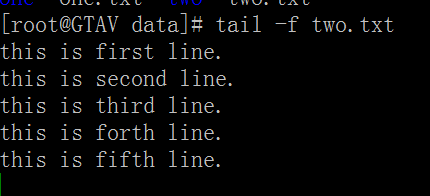
****

**Tail命令用来从文件的尾部进行读取，和head正好相反。**

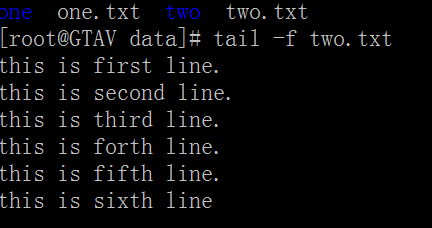
****

**它会跟踪一个文件尾部的变化，而且是即时的。使用-f参数可以观察到这种现象**

**下图的两个图片，当文件尾部改变时，它会实时跟踪这个文件。**

****

**当修改了这个文件后**

****

# **mkdir命令：**

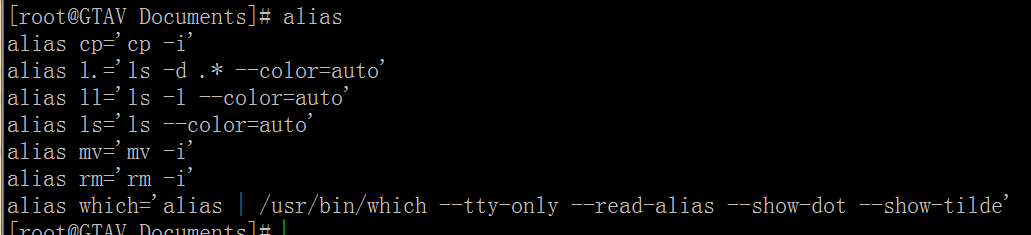
**创建目录，**

**使用-p参数，递归创建目录，即使目录不存在也会创建**

# **别名**

**复制一个文件到一个已存在的目录下，会弹出提示。如何在不提示的情况下直接执行程序，下面会讨论别名的概念和用法**

**Linux因为有时候命令很长，或者需要用到时经常会用到的是这个命令加上其他参数，所以用一个别名来替代这个较长的命令。Linux使用alies这个命令来查看别名，修改，停用别名**

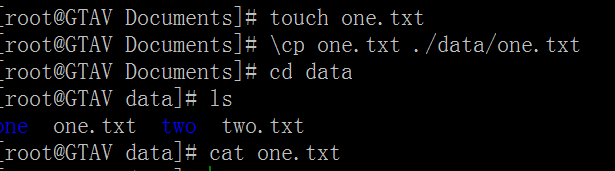
****

**这是默认命令行里存在的别名，比如cp，会调用cp -i 命令，即当复制的文件遇到重名时，会弹出提示。有两种方法来关闭这种提示：**

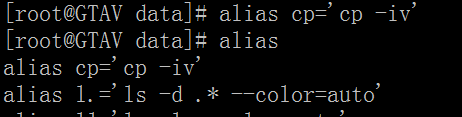
**一是使用unalias ，直接停用别名，这样系统的一切别名服务都会被停止，只在当前回话有效**

**但是不推荐这种方式，因为这本来就是系统为提高安全性而开发的特性，这会降低系统的安全性。**

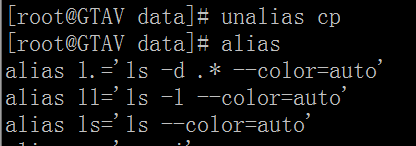
**二是使用反斜杠\,比如cp命令，在使用时在cp命令前添加\cp，即可临时关闭cp的别名，下图演示了这种用法**

****

**很多时候我们需要自定义别名，使用alias name=’string’来自定义别名，下图演示了这种方式**

****

**也可以取消别名，使用ualias name 来取消别名，如下图**

****

**但是以上的这些操作仅仅在当前会话有效，当关闭当前回话后，别名依然有用。如果想让别名一直生效，需要在全局文件中进行声明**

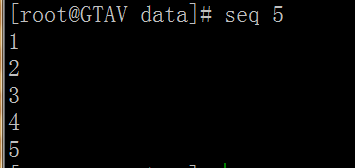
**Alias rm=’echo don’t use this command , due of danger’ 定义别名**

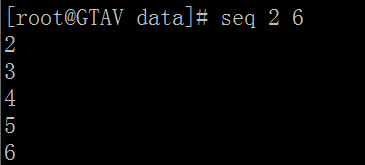
**别名生效的位置：/root/.bashrc中（针对root用户）**

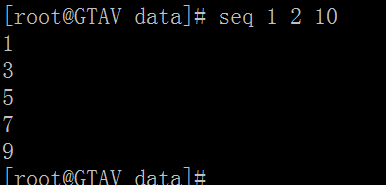
**如果希望所有用户生效：在etc/profile或者ext/bashrc中定义**

# **Seq**

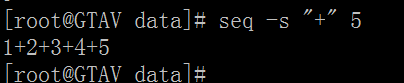
* 1. **序列命令，用来生成序列，可以有三个参数，**
  2. **Seq n 表示n个数，从1开始到n的正整数**
  3. **Seq n1,n2 表示从n1到n2的正整数**
  4. **Seq n1,n2,n3 表示步长为n2,从n1到n3的正整数，左右都包括**

****

****

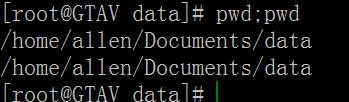
****

**使用-s参数来指定分隔符，然后横着输出，其实默认的是以换行符作为分割，下图以+作为分隔符**

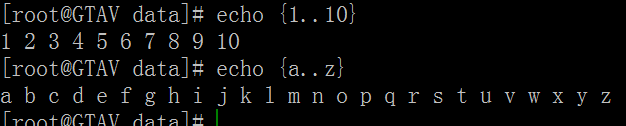
****

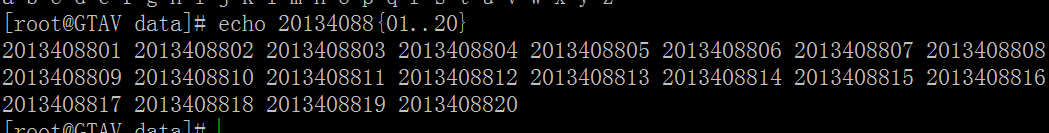
# **特殊符号**

**分号：分号用来在一行中执行多个命令之间的分隔符**

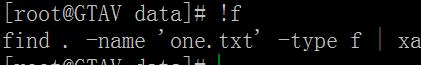
****

**大括号：{}用来生成序列，配合echo使用，在find中存放数据的集合，以..两个点号进行扩展**

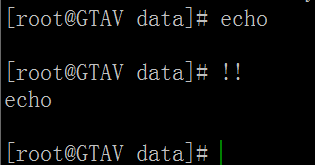
****

****

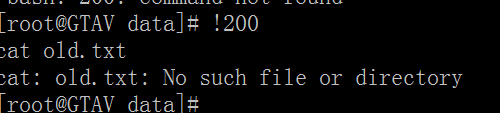
**感叹号：! + 字母 调出最近一次以此字母开头的命令；**

****

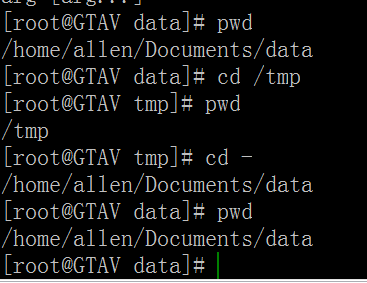
**双感叹号：!! 表示调出最后一次使用的命令**

****

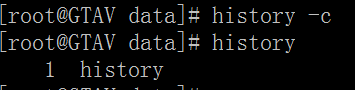
**感叹号+数字 调出历史第i条命令，中间没有空格**

****

**横线-： 进入上一次的目录，不是上一级的目录**

****

**History： 记录命令历史，使用-c来清除历史记录**

****

**History num 列出最近的num条命令**

# **Linux快捷键**

**Ctrl+a 让光标从任意地方移动到命令开头**

**Ctrl+e 让光标从任何地方移动到尾部**

**Ctrl+u 把光标之前的命令删掉（不用选中）**

**Ctrl+k 把光标之后的命令删掉（不用选中）**

**Ctrl+shift+c 复制 Ctrl+shift+v 粘贴**

**Ctrl+l 清屏**

**Ctrl+d 关闭当前链接**