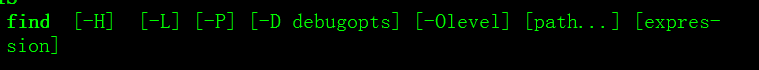
# find命令探究

## 参数：

**Find命令的格式是：find 搜索路径 搜索目录模式 搜索文件名模式 执行动作**



**默认的，搜索路径是当前路径。搜索目录模式使用-path参数，搜索文件名模式使用-name参数，执行动作默认是把搜索结果输出到标准流中，即使不满足，只要搜索到都会打印到标准输出中，使用-print来打印满足条件的文本；**

**-D是调试模式，这个模式后面加debugopts参数，查看使用find时会遇到的问题；（留作以后讨论）**

**-0level 是不同的等级模式，会对应不同的效果（没看懂，留作后面讨论）**

**-path 表示搜索的路径**

**Expression: 这个参数包括三个选项：options,tests,action 分隔符**

Options **总是返回true，它影响的是几乎所有的操作，并不只是某个具体文件的处理；option用来处理命令行的解析；而test并不会做任何事，直到文件被解析完成；所以，option会影响所有的tests参数；它被放在表达式的开头，用来设定一些基本的筛选参数；以及一些find命令的版本信息等。**

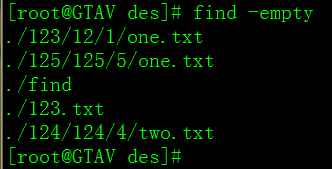
**Options的参数有以下几个：**

**-depth : 用来处理目录里面的内容；**

**-maxdepth levels : 用来设置遍历树的最大深度。Level必须是一个非负整数，如果是0，则依赖于test和action的值；**

Tests： **返回布尔值，这个参数用来将所筛选到的文件和测试文件进行对比，然后筛选出相应的结果。参数如下：**

**-empty：寻找空文件和空目录**

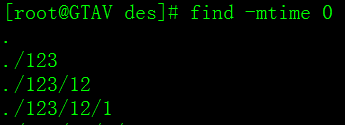


-gid n : 依据文件的组id号查找

-group gname：根据文件的组名查找

-mmin n ：寻找在n分钟以前修改过的文件

-mtime n : 寻找在n天以前修改过的文件



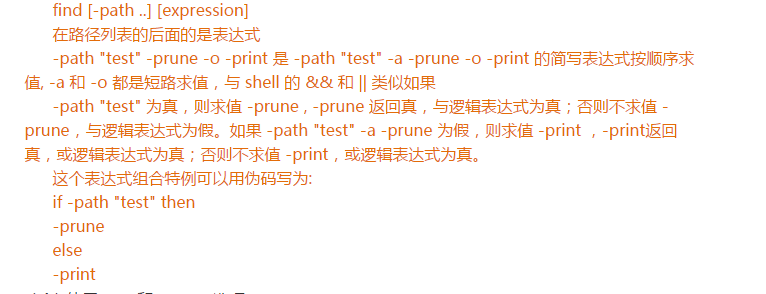
-name ：以文件名来查询，文件名可以是正则表达式

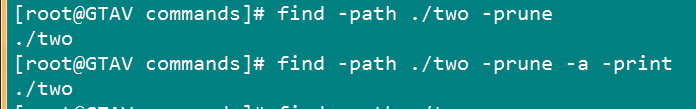
-path ：以目录名来查询，如果要忽略某个目录，使用-prune参数；查询目录时，目录必须使用路径，不能只是用目录名



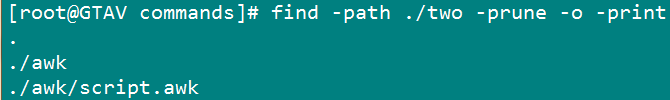
**使用-prune参数和path参数结合使用，可以忽略某个目录及其子目录，这适用于当查找一个目录或文件时，已经确定某个目录下没有要找的文件，则可以选择跳过，而不是再去遍历。下面的例子显示了这种用法**



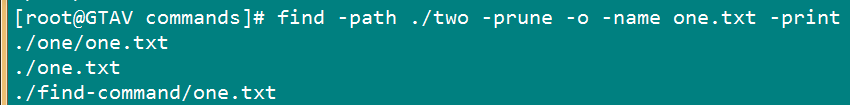
****



上面两个命令是等价的，因为不指定-a，相当于默认是匹配到two目录后直接输出来匹配到的结果；

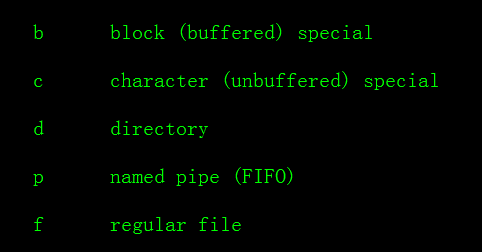


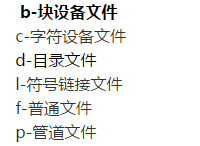
如果使用的是-o，表示如果匹配到目录two，则忽略掉，然后输出目录下的其他文件；



如果-o后面加了其他命令，则表示，如果匹配到two目录，则忽略这个目录，然后执行-o后面的命令；否则，直接执行-o后面的命令；

-type c ：指定文件类型，c可以是以下的任意一种

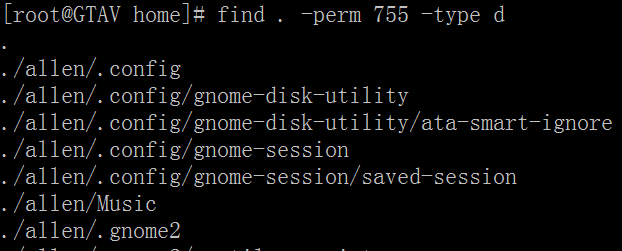


****

-uid n : 使用用户的uid来查找

-user name ： 使用文件的所有者来查询；、

**使用-perm选项，按照文件权限模式来查找，文件权限使用八进制模式，下面是例子**

****

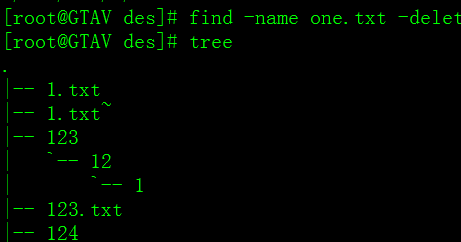
Action：**返回布尔值，会对操作有部分影响；**

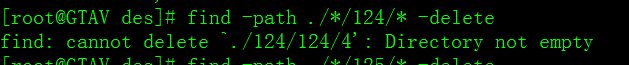
**分隔符：默认的分隔符是-and；**

**如果表达式不包含action，（除了-prune之外），在表达式必须返回真的情况下，-print会是默认的action；**

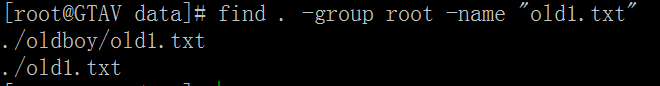
**常用参数包括：**

**-delete ： 删除文件，如果删除成功则返回true，否则报错。使用这个选项会自动打开-depth参数；而且，不要把这个命令放在最前面，因为它会从你指定挂载的地方开始删除，那样会删除所有东西，一定要详细指定删除的位置，因为它默认使用了-depth选项，而且，这个选项不可以和-prune选项同时使用**





**它不能删除非空目录；但可以删除非空文件；**

****

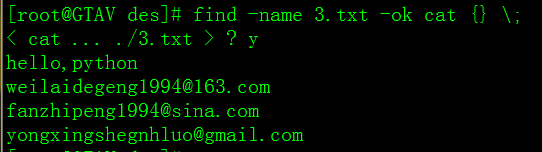
**-exec ： 查找一个匹配文件，并执行该参数给定的shell命令，相应的命令形式为“command {} \”,注意{}和\之间有空格**

**格式为：find + options - exec command {} \;**

**这种方式适合小容量的查找，因为它是一次性把筛选到的内容都放在大括号中，所以效率很低，更好的解决方案是使用xargs，后者适合大容量的文件查找；**

**-OK ： 和-exec的作用相同，只是以一种更安全的方式来执行，每次执行操作都会询问用户是否继续；**





**Xargs command : 这是一种更好方式的传递命令，格式为：**

**Find + options + action | xargs command**

**它把find到的结果分步提交给xargs，因为是管道，所以处理效率会更好，适合大容量的文件。**

**使用find命令删除**

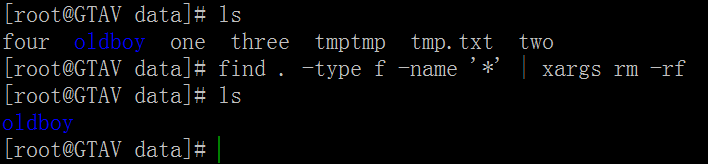
**第一种方案：**

**Find /root/data -type f -exec rm -f {} \;**

**使用-exec，必须使用;来结束，由于shell对分号有歧义，所以要转义；{}用来存放find找到的数据，交给rm来处理**

**更好的方式：**

**Find /root/data -type f | xargs rm -f**

****

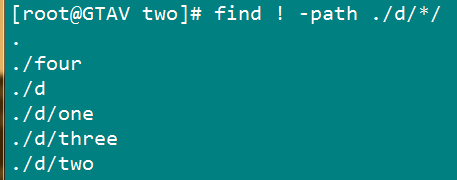
**Note： 因为第一种方式是直接把所有find到的文件全部一次性放到rm的{}当中，所以有可能有参数太多的错误**

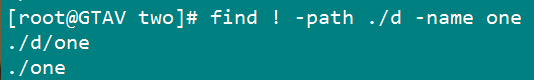
**而第二种方式会分批次的将文件通过管道进入rm命令，不会报错**

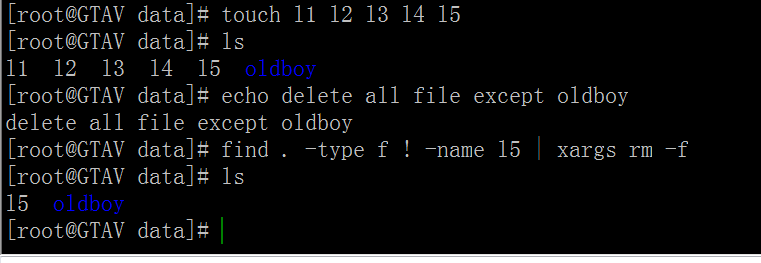
**一个经典问题：**

**在一个目录里，删除所有文件，但保留一个指定文件**

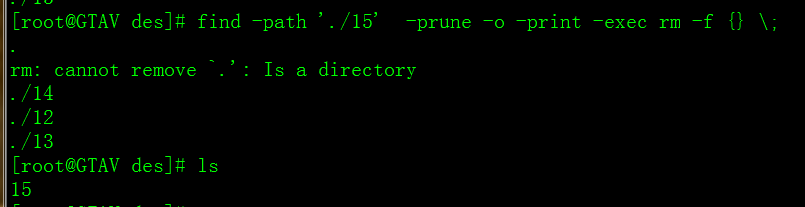
**使用！符号来完成，感叹号相当于“非”，在它之后的第一个参数的结果会被忽略**





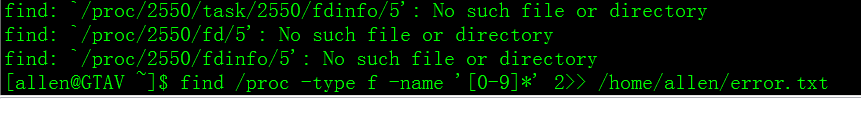
****

**另一种方法**



当需要查找的文件出错时，可以利用标准错误输出把错误文件写到一个文件中

就像下面这样



**这个方法是把查找错误提示转移到特定的目录中去。系统执行这个命令后，遇到错误的信息就直接输送到error.txt 中，error.txt就是表明系统将把错误信息输送到stderrstream ，这样查询到的错误信息将被转移了，不会再显示了**

**还有一些特殊的函数用来查找**

