# find命令探究

* 1. 使用name选项：文件名选项是最常用的选项，按照名字来查找，

格式为：

Find+ 搜查目录+ ‘-name’ + 搜寻文件 -print

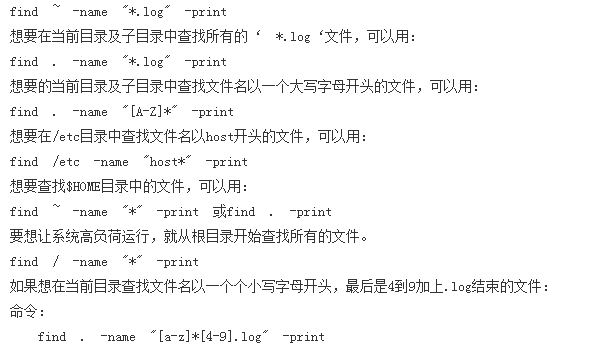
-print 表示将匹配到的结果显示到标准输出中；

搜查目录可以是任意目录，.表示当前目录，/表示根目录；

-name 用来添加文件名称；

搜寻的文件名可以是普通文件名，也可以是由正则表达式构成的文件名，匹配指定的文件。

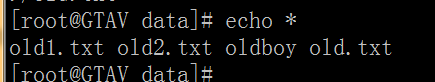
正则表达式的一般实例在下图中有用法展示；



错误解决：

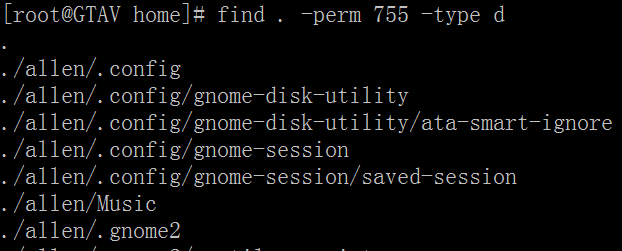
find: paths must precede expression  
Usage: find [-H] [-L] [-P] [path...] [expression]

解决方法：使用\*的时候必须要加单引号或者对\*进行转义，使用\\*，因为默认\*会匹配所有的内容，它会展开目录下的所有文件，就像这样：

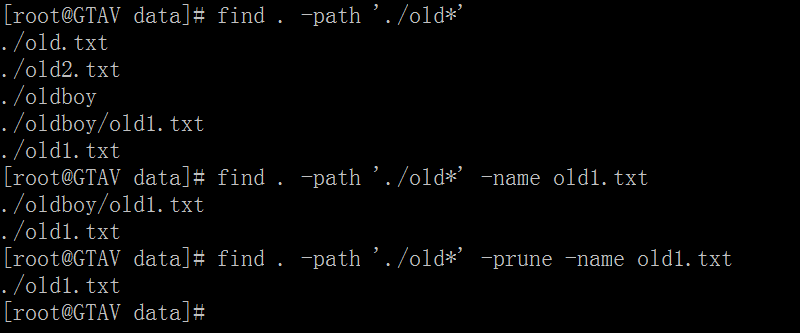


所以要使用转义或者引号将\*进行约束，使它能够匹配指定的表达式。

b) 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



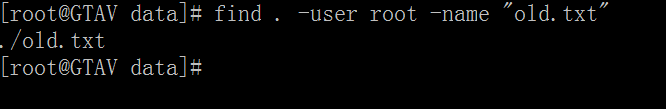
-prune : 使用这个参数用来忽略某个文件，它和-path参数一起使用。用来忽略指定的目录



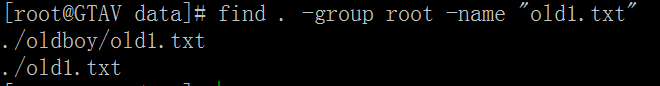
Note： -path 参数 表示匹配的路径，第一行命令表示在当前目录下匹配以old开头的目录。因为是多文件匹配，所以要加单引号。

1. 命令：添加了-name，会在当前目录里匹配old开头的目录，然后再匹配old1.txt文件名。
2. 命令：添加了-prune ，会在当前目录里匹配old1.txt，因为使用了prume参数，所以不会在-path路径里进行查找
3. ：顺序很重要，如果命令的顺序弄反，则不会出现如上的筛选结果

使用-user：通过文件所属的用户来筛选文件



使用-group：通过文件所属的用户组来筛选文件



Find命令格式：find pathname -options [-print -exec -ok ....]

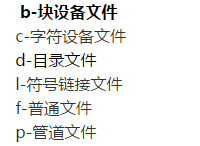
Pathname : 表示find文件查找的搜索路径；

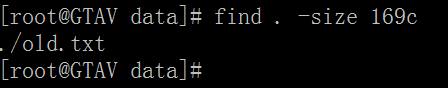
-print ： 将查找后匹配到的文件输出到标准输出上

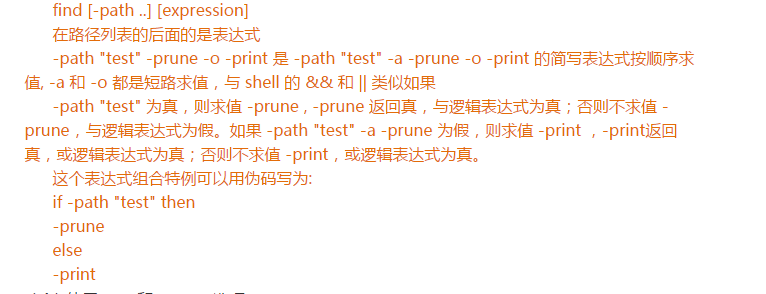
-exec ： 查找一个匹配文件，并执行该参数给定的shell命令，相应的命令形式为“command {} \”,注意{}和\之间有空格

-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；

命令选项：

1. -name : 按照文件名来查找
2. -perm : 按照文件权限来查找
3. -prune : 使用这个选项可以让find忽略某个目录
4. -user : 按照文件所属的用户来查找
5. -group : 按照文件所属的用户组来查找
6. -mtime -n/+n: -n表示n天以前，+n表示n天之后。这个参数表示文件的修改时间。还有ctime和atime，作用类似
7. -type：按照文件类型查找
8. 
9. -size n:(c) 查找文件时查找文件大小为n块的文件，带有c时表示以字节计算。



1. -depth： 表示查找文件时，先在当前目录查找，然后在子目录里查找。
2. Find -path [添加表达式]
3. 

使用find命令删除

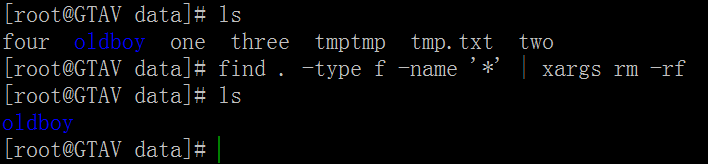
第一种方案：

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

