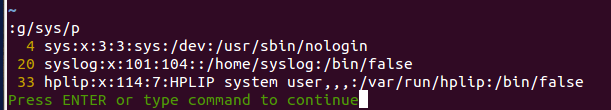
shell 脚本

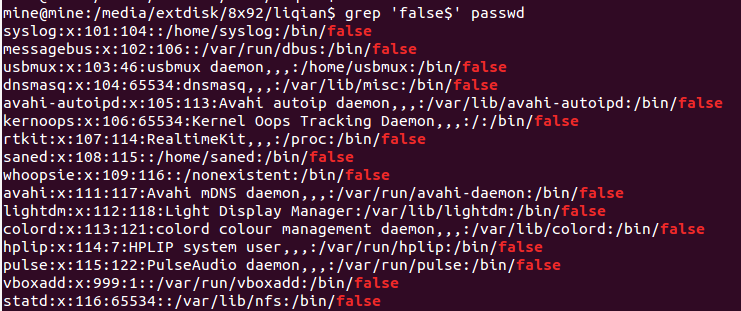
grep

#vim编辑器中

:□g/pattern/p



#terminal grep 正则表达式

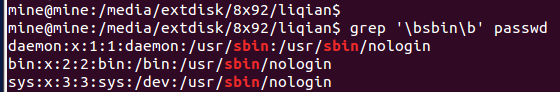


#grep keyword path/filename



#grep 匹配单词





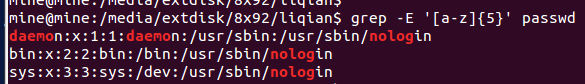
**# \b 单词分割符号 \b**word**\b** 匹配单词

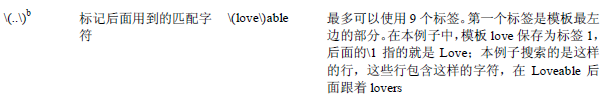
**\<** 匹配单词开始 **\>**匹配单词结尾

【标记匹配到的字符串】

扩展元字符 () {} 必须使用反斜杠 \

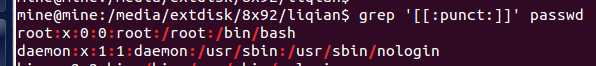
{} grep –E 不必使用 \





POSIX字符类

#grep使用:digit: :punct:



grep

查看查找到的结果

echo $?

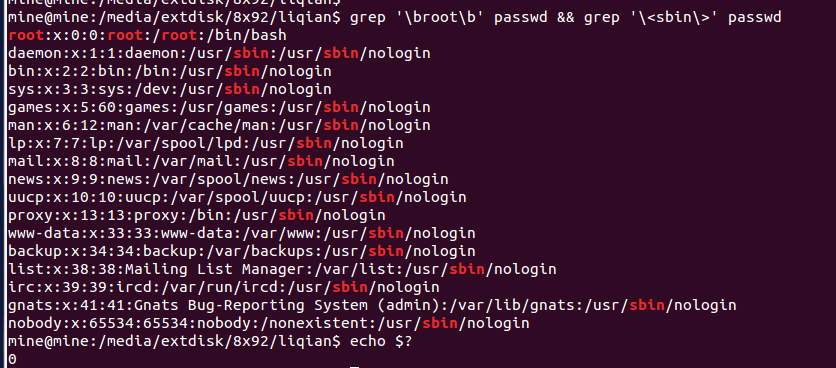
0 ： successed

1 ：fail

2 ：file not found

#greo 逻辑表达式、echo $?

grep && or



扩展grep（grep-E或者egrep）

grep 常用选项

从多个文件中匹配

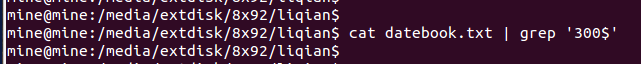
grep keyword keyword1 keyword2 keyword3

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  | -E | 使用正则表达式【重要】 |
|  | -o -v | 打印匹配（不匹配项） |
|  | -c | 只统计行数 |
|  | -i | 忽略大小写 |
|  | -e | 匹配多个字符串 |
|  | -q | 静默模式 |

P74 linux shell实例精解

grep –E ‘regrex’ filename

【疑惑】



#sed基本命令

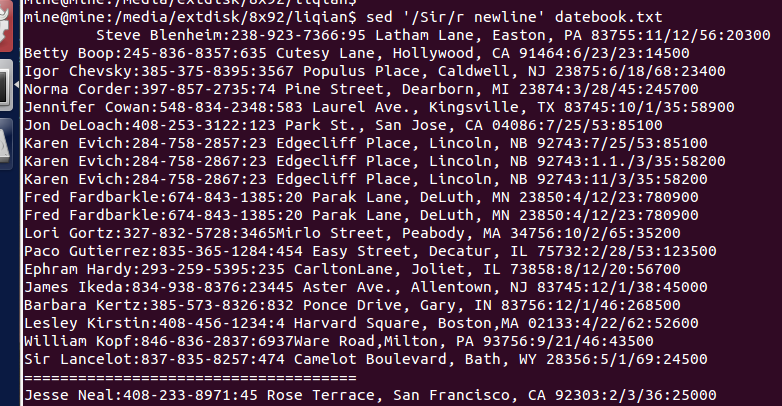
|  |  |  |
| --- | --- | --- |
|  | d | 从模板块指定位置删除 |
|  | D |  |
|  | r | 从file中读取 |
|  | a |  |
|  | i |  |
|  | c | 用新的文本改变或者替代本行的文本 |
|  | p |  |
|  | n | 静默输出 |
|  | e | 多点编辑 |
|  | q | 退出 |

sed ‘m ,n/search /replace /g, p, d, i, a’

sed ‘/Keyword / {s/ / / ;p } /’

sed ‘/pattern/ , / pattern / s/ / / ;p’

r filename 从指定文件读取文件的内容



sed –n ‘/keyword/w newfilename’ datafile

读取包含关键字keyword的行，将datafile插入其中；

C:\Users\liqian1\Pictures\李前_工牌\Linux命令.png

awk

awk 不支持正则表达式 {m, n}