unix 的 grep 命令

一、grep、egrep、fgrep 命令

本文中主要介绍了 linux 系统下 grep egrep fgrep 命令和正则表达式的基本参数和使用格式、方法。（注释：文中 fg 代表例子，）

1.1、基本定义：

grep（global search regular RE ） and print out the line,全面搜索正则表达式并把行打印出来）是一种强大的文本搜索工具，它只能使用基本的正则表达式来搜索文本，并把匹配的行打印出来。

grep 是很常见也很常用的命令，它的主要功能是进行字符串数据的比较，然后符合用户需求的字符串打印出来，但是主意，grep 在数据中查找一个字符串时，是以“整行”为单位进行数据筛选的。

egrep 命令等同于 grep -E，利用此命令可以使用扩展的正则表达式对文本进行搜索，并把符合用户需求的字符串打印出来。

fgrep 命令等同于 grep -F，它利用固定的字符串来对文本进行搜索，但不支持正则表达式的引用，所以此命令的执行速度也最快。

1.2、命令基本用法

grep [option] ’搜索字符串’ filename

grep 常用选项：

-a :在二进制文件中，以文本文件的方式搜索数据

-c :计算找到’搜索字符串’的次数

-i :忽略大小写

-v :反向查找，即显示没有’搜索字符串’内容的那行

-o :只显示被模式匹配的字符串

-n :输出行号

--colour（color）:颜色显示

-A：显示匹配到字符那行的后面 n 行

-B：显示匹配到字符那行的前面 n 行

-C：显示匹配到字符那行的前后 n 行

二、正则表达式

2.1、基本定义：

正则表达使用单个字符串来描述、匹配一系列符合某个句法规则的字符串。在很多文本编辑器里，正则表达式通常被用来检索、替换那些符合某个模式的文本。简而言之，正则表达式就是处理字符串的方法，以行为单位进行字符串的处理，通过一些特殊符号的辅助，可以让用户轻松搜索/替换某特定的字符串。

正则表达式分为两类：基本的正则表达式和扩展的正则表达式。

2.2、正则表达式详细介绍

2.2.1、基本的正则表达式：

（1）元字符：

. :匹配任意单个字符

fg：查找包含 student 且 student 后面带一个字符的行

grep ‘student.’ /etc/passwd （模式可以用单引号和双引号，如果模式中要做变量替换时则必须用双引）

[] :匹配指定范围内的任意单个字符,[abc],[a-z],[0-9],[a-zA-Z]

fg：查找带有数字的行

grep ‘[0-9]’ /etc/passwd

[^] :匹配指定范围外的任意单个字符

fg：查找没有小写字母的行。

grep ‘[^a-z]’ /etc/inittab

[:space:]:表示空白字符

[:punct:]:表示所有标点符号的集合

[:lower:]:表示所有的小写字母

[:upper:]:表示所有的大写字母

[:alpha:]:表示大小写字母

[:digit:]:表示数子

[:alnum:]:表示数字和大小写字母-----使用格式[[:alnum:]]等

（2）次数匹配：

*　:匹配其前面的字符任意次

fg：查找 root 出现 0 次或 0 次以上的行

grep ‘root*’ /etc/passwd

.* :任意字符

fg：查找包含 root 的行

grep 'root.*' /etc/passwd

\?：匹配其前面的字符 1 次或 0 次

\{m,n\} :匹配其前字符最少 m，最多 n 次）

(3) 字符锚定：

^:锚定行首，此字符后面的任意内容必须出现在行首

fg：查找行首以#开头的行

grep '^#' /etc/inittab

$:锚定行尾，此字符前面的任意内容必须出现在行尾

fg：查找行首以 root 结尾的行

grep 'root$' /etc/inittab

^$:锚定空白行，可以统计空白行

\<或者\b:锚定词首，其后面的任意字符必须做为单词首部出现

fg:查找 root 且 root 前面不包含任何字符的行

grep '\

\>或者\b:锚定词尾，其前面的任意字符必须做为单词尾部出

现 fg：\ 查找 root 单词 grep "\" =grep "\broot\b"

2.2.2、扩展的正则表达式：

扩展的正则表达只是在基本的正则表达上作出了小小的一点修改，其修改如下：

在扩展的正则表达中把 写成()、\{ \} 写成{ }，另外加入了+：次数匹配，匹配其前面的字符至少出现一次，无上限、|: 或者(二取一），其余的都一样， 基本正则表达式，使用( ) { } . ? |都需要转义,在扩展正则表达中不需要加\，其详细信息如下：

(1) 字符匹配的命令和用法与基本正则表达式的用法相同，这里不再重复阐述。

(2) 次数匹配：

* :匹配其前面字符的任意次

? :匹配其前面字符的 0 此或着 1 此

+ :匹配其前面字符至少 1 此

fg：至少一个空白符： '[[:space:]]+'

{m,n} :匹配其前面字符 m 到 n 次

(3) 字符锚定的用法和基本正则表达式的用法相同，在此不再阐述。

（4）特殊字符：

| : 代表或者的意思。

fg：grep -E 'c|cat' file：表示在文件 file 内查找包含 c 或者 cat

\.:\表示转义字符，此表示符号.

三、grep 命令利用小实例

(1)显示/etc/inittab 中以#开头，且后面跟一个或者多个空白符，而后又跟了任意非空白符的行

grep '#[[:space:]]*[^[:space:]]' /etc/inittab

(2) 输出不是数字开关的行 grep '^[^0-9]'

/etc/passwd

(3)输出行首是 1 或 2

grep '^1||21‖2' /etc/inittab

或

grep -E '^(1|2)' /etc/inittab

(4)查找前面是 rc 中间接任意字符而后跟/rc

grep '.*rcrc.*\/\1.*' /etc/inittab

(5）取出当前电脑上的 IP

ifconfig |grep -A 1 "^eth0" |grep "\<[0-9.]\{1,\} |cut -d: -f2

(6)查找当前系统上名字为 student（必须出现在行首）的用户账户的相关信息，文件为/etc/passwd

grep "^student" /etc/passwd