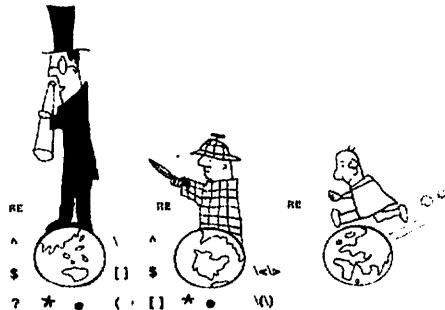


chapter

4

grep 家族



grep 家族由命令 grep、egrep 和 fgrep 组成。grep 命令在文件中全局查找指定的正则表达式，并且打印所有包含该表达式的行。egrep 和 fgrep 都只是 grep 的变体。egrep 命令是扩展的 grep，支持更多的正则表达式元字符。fgrep 命令被称为固定 grep(fixed grep)，有时也被称作快速 grep(fast grep)，它按字面解释所有的字符，也就是说，正则表达式元字符不会被特殊处理，它们只匹配自己。自由软件基金会提供了 grep 的免费版本，称作 GNU grep。Linux 系统上使用的就是这种版本的 grep。在 Sun 公司的 Solaris 操作系统上也可以找到 GNU 版的 grep，所在目录是 /usr/xpg4/bin。GNU 版的 grep 扩展了基本的正则表达式元字符集，增加了与 POSIX 的一致性，并且包括很多新的命令行选项。它们还提供了一个名为 rgrep 的递归式 grep，用于逐级搜索整个目录树。

4.1 grep 命令

4.1.1 grep 的含义

grep 这个名字的由来可以追溯到 ex 编辑器。如果启动 ex 编辑器来查找某个字符串，需要在 ex 的命令提示符后键入：

```
:/pattern/p
```

字符“p”表示打印(print)命令，所以包含字符串 pattern 的第一行的内容会被打印出来。如果希望打印所有包含 pattern 的行，可以输入：

```
:g/pattern/p
```

当 g 出现在 pattern 前面时，其含义是“文件中的所有行”，或“执行全局替换”。

被查找的模式称作正则表达式(regular expression)，因此我们可以用 RE 来替换 pattern，于是上面这条命令就变成了：

```
:g/RE/p
```

说到这，您就应该清楚 grep 的含义和它名称的来源了。它表示“全局查找正则表达式(RE)并且打印结果行。”使用 grep 的好处在于，不需要启动编辑器就可以执行查找操作，也用不着把正则表达式括在正斜杠里。使用 grep 比使用 ex 或 vi 快得多。

4.1.2 grep 如何工作

grep 命令在一个或多个文件中查找某个字符模式。如果这个模式中包含空格，就必须用引号把它括起来。grep 命令中，模式可以是一个被引号括起来的字符串，也可以是单个词^①，位于模式之后的所有单词都被视为文件名。grep 将输出发送到屏幕，它不会对输入文件进行任何修改或变化。

命令格式

```
grep word filename filename
```

范例 4-1

```
grep Tom /etc/passwd
```

说明

grep 将在文件/etc/passwd 中查找模式 Tom。如果查找成功，文件中的相应行会显示在屏幕上，如果没有找到指定的模式，就不会有任何输出，如果所指定的文件不是一个合法文件，屏幕上就会显示报错信息。如果发现了要查找的模式，grep 返回的退出状态为 0，表示成功，如果没找到，返回的退出状态就是 1，而找不到指定的文件时，退出状态将是 2。

grep 程序的输入可以来自标准输入或管道，而不仅仅是文件。如果忘了指定文件，grep 会以为你要它从标准输入(即键盘)获取输入，于是停下来等着你键入一些字符。如果输入来自管道，就会有另一条命令的输出通过管道变成 grep 命令的输入，如果匹配到要查找的模式，grep 会把输出打印在屏幕上。

范例 4-2

```
ps -ef | grep root
```

说明

ps 命令的输出(ps -ef 显示正在系统上运行的所有进程)被送到 grep，然后所有包含 root 的行被打印在屏幕上。

4.1.3 元字符

元字符也是一种字符，但它表达的含义不同于字符本身的字面含义。例如，^和\$就是元字符。

grep 命令支持很多正则表达式元字符(参见表 4-1)，以使用户更精确地定义要查找的模式。它还提供了很多命令选项(参见表 4-2)用于调整执行查找或显示结果的方式。例如，可以通过指定选项来关闭大小写敏感、要求显示行号，或者只显示报错信息等。

^① 词(word)也称为标记(token)。

范例 4-3

```
grep -n '^jack:' /etc/passwd
```

说明

grep 在文件/etc/passwd 中查找 jack，如果 jack 出现在某行的行首，grep 就打印出该行的行号和内容。

表 4-1 grep 使用的正则表达式元字符

元字符	功 能	示 例	匹 配 对 象
^	行首定位符	'^love'	匹配所有 love 开头的行
\$	行尾定位符	'love\$'	匹配所有以 love 结尾的行
.	匹配一个字符	'.e'	匹配包含一个 l，后跟两个字符，再跟一个 e 的行
*	匹配零个或多个前导字符	'*love'	匹配包含跟在 0 或多个空格后的模式 love 的行
[]	匹配一组字符中任一个	'[Ll]ove'	匹配 love 或包含 Love 的行
[^]	匹配不在指定字符组内的字符	'[^A-K]ove'	匹配包含一个不在 A 至 K 之间的字符，并且该字符后紧跟着 ove 的行
<	词首定位符	'<love'	匹配包含以 love 开头的词的行
>	词尾定位符	'love>'	匹配包含以 love 结尾的词的行
\(...)	标记匹配到的字符	'\(\love\)\ing'	标记寄存器中的一段字符，被记作 1 号寄存器。如果之后要引用这段字符，可以用 \1 来重复该模式。最多可以设置 9 个标签，从左边开始编号，最左边的是第一号。左边这个例子中，模式 love 被保存在 1 号寄存器里，之后可以用 \1 来引用它
x\{m\} 或 x\{m,\} 或 x\{m,n\} ^②	字符 x 的重复出现次数： m 次、至少 m 次、至少 m 次但不超过 n 次	'o\{5\}' 'o\{5,\}' 'o\{5,10\}'	匹配连续出现 5 个 o、至少 5 个 o 或 5~10 个 o 的行

表 4-2 grep 的选项

选 项	功 能
-b	在每一行前面加上其所在的块号，根据上下文定位磁盘块时可能会用到
-c	显示匹配到的行的数目，而不显示行的内容
-h	不显示文件名
-i	比较字符时忽略大小写的区别(即认为字母的大小写相等)
-l	只列出匹配行所在文件的文件名(每个文件名只列一次)，文件名之间用换行符分隔

② 并非所有版本的 UNIX 或所有的模式匹配工具都支持元字符 \{ \}，但 vi 和 grep 通常都支持它们。

(续表)

选 项	功 能
-n	在每一行前面加上它在文件中的相对行号
-s	无声操作，即只显示报错信息，以检查退出状态
-v	反向查找，只显示不匹配的行
-w	把表达式作为词来查找，就好像它被\<和\>所包含一样。只适用于 grep(并非所有版本的 grep 都支持这一功能，譬如，SCO UNIX 就不支持)

4.1.4 grep 的退出状态

grep 在 shell 脚本中很有用，因为它总会返回一个退出状态，以说明能否定位到要查找的模式或文件。如果找到了模式，grep 返回的退出状态为 0，表示成功，如果找不到模式，grep 返回 1 作为退出状态，而当找不到要搜索的文件时，grep 返回的退出状态是 2(其他查找模式的 UNIX 工具，例如 sed 和 awk，不使用退出状态来说明查找模式成功与否，它们只在命令中出现语法错误时才报告失败)。

下面这个范例中，在文件/etc/passwd 中未找到 john。

范例 4-4

```

1 % grep 'john' /etc/passwd      # john is not in the passwd file
2 % echo $status      (csh)
1
或者
2 $ echo $?      (sh, ksh)
1

```

说明

1. grep 在文件/etc/passwd 中查找 john，如果成功，grep 以状态 0 退出。若未在该文件中找到 john，grep 以状态 1 退出。如果没找到文件/etc/passwd，返回的退出状态是 2。
2. C shell 的变量 status 和 Bourne/Korn shell 的变量“?”的值都是上一条命令执行后的退出状态。

4.2 使用正则表达式的 grep 实例

下面这些例子所使用的文件的名字是 datafile。为了方便您的阅读，将会在必要时重复地出现。

% cat datafile							
northwest	NW	Charles Main	3.0	.98	3	34	
western	WE	Sharon Gray	5.3	.97	5	23	
southwest	SW	Lewis Dalsass	2.7	.8	2	18	
southern	SO	Suan Chin	5.1	.95	4	15	
southeast	SE	Patricia Hemenway	4.0	.7	4	17	
eastern	EA	TB Savage	4.4	.84	5	20	
northeast	NE	AM Main Jr.	5.1	.94	3	13	
north	NO	Margot Weber	4.5	.89	5	9	
central	CT	Ann Stephens	5.7	.94	5	13	

范例 4-5**% grep NW datafile**

northwest NW Charles Main 3.0 .98 3 34

说明

打印文件 datafile 中所有包含正则表达式 NW 的行。

范例 4-6**% grep NW d***datafile: northwest NW Charles Main 3.0 .98 3 34
db: northwest NW Joel Craig 30 40 5 123**说明**

打印所有名字以 d 开头的文件中，包含正则表达式 NW 的所有行。shell 把 d* 扩展为所有名字以 d 开头的文件，这个例子中，所包括的文件是 db 和 datafile。

范例 4-7**% grep '^n' datafile**northwest NW Charles Main 3.0 .98 3 34
northeast NE AM Main Jr. 5.1 .94 3 13
north NO Margot Weber 4.5 .89 5 9**说明**

打印所有以字母 n 开头的行。脱字符(^\)是句首定位符。

范例 4-8**% grep '4\$' datafile**

northwest NW Charles Main 3.0 .98 3 34

说明

打印所有以数字 4 结尾的行。美元符(\$)是行尾定位符。

范例 4-9**% grep TB Savage datafile**

```
grep: Savage: No such file or directory
datafile: eastern EA TB Savage
```

4.4 .84 5 20

说明

第一个参数是模式，其余所有参数都是文件名，grep 将在 Savage 和 datafile 这两个文件中查找 TB。如果要查找 TB Savage，请看下一个范例。

% cat datafile

northwest	NW	Charles Main	3.0	.98	3	34
western	WE	Sharon Gray	5.3	.97	5	23
southwest	SW	Lewis Dalsass	2.7	.8	2	18
southern	SO	Suan Chin	5.1	.95	4	15
southeast	SE	Patricia Hermenway	4.0	.7	4	17
eastern	EA	TB Savage	4.4	.84	5	20
northeast	NE	AM Main Jr.	5.1	.94	3	13
north	NO	Margot Weber	4.5	.89	5	9
central	CT	Ann Stephens	5.7	.94	5	13

范例 4-10

```
% grep 'TB Savage' datafile
eastern EA TB Savage 4.4 .84 5 20
```

说明

打印所有包含模式 TB Savage 的行。如果不加引号(这个例子中，使用单引号或双引号都可以)，TB 和 Savage 之间的空格将导致 grep 在文件 Savage 和 datafile 中查找 TB，就像前面那个示例一样。

范例 4-11

```
% grep '5\..' datafile
western WE Sharon Gray 5.3 .97 5 23
southern SO Suan Chin 5.1 .95 4 15
northeast NE AM Main Jr. 5.1 .94 3 13
central CT Ann Stephens 5.7 .94 5 13
```

说明

打印所有包含数字 5，后跟一个句点，再跟一个任意字符的行。句点这个元字符通常代表单个字符，除非它被反斜杠转义。句点被转义后就不再是特殊的元字符，而只代表本身，即一个句号。

范例 4-12

```
% grep '\.5' datafile
north NO Margot Weber 4.5 .89 5 9 6
```

说明

打印所有包含表达式 “.5” 的行。

范例 4-13

```
% grep '^*[w]' datafile
western      WE    Sharon Gray      5.3   .97   5   23
eastern      EA    TB Savage      4.4   .84   5   20
```

说明

打印所有以字母 w 或 e 开头的行。脱字符(^)是句首定位符，方括号中任何一个字符都可以被匹配。

范例 4-14

```
% grep '[^0-9]' datafile
northwest    NW    Charles Main    3.0   .98   3   34
western      WE    Sharon Gray    5.3   .97   5   23
southwest    SW    Lewis Dalsass  2.7   .8    2   18
southern     SO    Suan Chin     5.1   .95   4   15
southeast    SE    Patricia Hemenway 4.0   .7    4   17
eastern      EA    TB Savage     4.4   .84   5   20
northeast    NE    AM Main Jr.   5.1   .94   3   13
north       NO    Margot Weber   4.5   .89   5   9
central      CT    Ann Stephens  5.7   .94   5   13
```

说明

打印包含非数字字符的所有行。由于每一行都至少有一个非数字字符，因此所有行都被打印(参见表 4-2 中的-v 选项)。

范例 4-15

```
% grep '[A-Z][A-Z] [A-Z]' datafile
eastern      EA    TB Savage     4.4   .84   5   20
northeast    NE    AM Main Jr.  5.1   .94   3   13
```

说明

打印所有包含两个大写字符、后跟一个空格和一个大写字符的行，例如 TB Savage 和 AM Main。

范例 4-16

```
% grep 'ss*' datafile
northwest    NW    Charles Main    3.0   .98   3   34
southwest    SW    Lewis Dalsass  2.7   .8    2   18
```

说明

打印所有包含一个 s、后跟 0 个或多个连着的 s 和一个空格的文本行。这个例子中，grep 找到了 Charles 和 Dalsass。

% cat datafile						
northwest	NW	Charles Main	3.0	.98	3	34
western	WE	Sharon Gray	5.3	.97	5	23
southwest	SW	Lewis Dalsass	2.7	.8	2	18
southern	SO	Suan Chin	5.1	.95	4	15
southeast	SE	Patricia Hemenway	4.0	.7	4	17
eastern	EA	TB Savage	4.4	.84	5	20
northeast	NE	AM Main Jr.	5.1	.94	3	13
north	NO	Margot Weber	4.5	.89	5	9
central	CT	Ann Stephens	5.7	.94	5	13

范例 4-17

```
% grep '[a-z]\{9\}' datafile
northwest      NW      Charles Main      3.0  .98  3   34
southwest     SW      Lewis Dalsass    2.7   .8   2   18
southeast      SE      Patricia Hemenway 4.0   .7   4   17
northeast     NE      AM Main Jr.    5.1   .94  3   13
```

说明

打印所有出现至少 9 个小写字母连在一起的行，例如，northwest、southwest、southeast 和 northeast。

范例 4-18

```
% grep '\(3\)\.[0-9]*\1' datafile
northwest      NW      Charles Main      3.0  .98  3   34
```

说明

如果某个文本行包含一个 3 后面跟一个句点和一个数字，再任意多个字符(*)，然后跟一个 3 和任意多个制表符，再接一个 3，则打印该行。由于数字 3 被括在圆括号中，\(\$1\$)，以后就可以用\\$1 来引用它。\\$1 代表被\(\$1\$)标记的第一个表达式。

范例 4-19

```
% grep '\<north\>' datafile
northwest      NW      Charles Main      3.0  .98  3   34
northeast     NE      AM Main Jr.    5.1   .94  3   13
north        NO      Margot Weber    4.5   .89  5   9
```

说明

打印所有包含以 north 开头的单词的行。“\<”是词首定位符。

范例 4-20

```
% grep '\<north\>' datafile
north        NO      Margot Weber    4.5   .89  5   9
```

说明

打印所有含单词 north 的行。“<”是词首定位符，“>”则是词尾定位符。

范例 4-21

```
% grep '\<[a-z].*n\>' datafile
northwest      NW      Charles Main      3.0  .98   3   34
western        WE      Sharon Gray       5.3  .97   5   23
southern        SO      Suan Chin        5.1  .95   4   15
eastern         EA      TB Savage        4.4  .84   5   20
northeast       NE      AM Main Jr.     5.1  .94   3   13
central         CT      Ann Stephens     5.7  .94   5   13
```

说明

打印所有包含以小写字母开头，以 n 结尾，中间由任意多个字符组成的单词的行。注意符号.*，它代表任意字符，包括空格。

4.3 grep 的选项

grep 提供了多个可用来控制其行为的选项。各种 UNIX 版本上 grep 的选项并不完全一样，因此一定要在手册中查找完整的 grep 选项清单。

本节中的示例使用下面的 datafile，为了阅读方便，该文件将会重复地出现。

% cat datafile						
northwest	NW	Charles Main	3.0	.98	3	34
western	WE	Sharon Gray	5.3	.97	5	23
southwest	SW	Lewis Dalsass	2.7	.8	2	18
southern	SO	Suan Chin	5.1	.95	4	15
southeast	SE	Patricia Hemenway	4.0	.7	4	17
eastern	EA	TB Savage	4.4	.84	5	20
northeast	NE	AM Main Jr.	5.1	.94	3	13
north	NO	Margot Weber	4.5	.89	5	9
central	CT	Ann Stephens	5.7	.94	5	13

范例 4-22

```
% grep -n '^south' datafile
3:southwest      SW      Lewis Dalsass      2.7  .8   2   18
4:southern        SO      Suan Chin        5.1  .95  4   15
5:southeast       SE      Patricia Hemenway  4.0  .7   4   17
```

说明

选项-n 在找到指定模式的行前面加上其行号再一并输出。

```
% cat datafile
```

northwest	NW	Charles Main	3.0	.98	3	34
western	WE	Sharon Gray	5.3	.97	5	23
southwest	SW	Lewis Dalsass	2.7	.8	2	18
southern	SO	Suan Chin	5.1	.95	4	15
southeast	SE	Patricia Hemenway	4.0	.7	4	17
eastern	EA	TB Savage	4.4	.84	5	20
northeast	NE	AM Main Jr.	5.1	.94	3	13
north	NO	Margot Weber	4.5	.89	5	9
central	CT	Ann Stephens	5.7	.94	5	13

范例 4-23

```
% grep -i 'pat' datafile
southeast      SE      Patricia Hemenway      4.0  .7  4  17
```

说明

选项-i 关闭大小写敏感性。表达式 pat 包含任意大小写的组合都符合。

范例 4-24

```
% grep -v 'Suan Chin' datafile
northwest      NW      Charles Main      3.0  .98  3  34
western        WE      Sharon Gray      5.3  .97  5  23
southwest      SW      Lewis Dalsass    2.7  .8   2  18
southeast       SE      Patricia Hemenway 4.0  .7   4  17
eastern         EA      TB Savage       4.4  .84  5  20
northeast       NE      AM Main Jr.     5.1  .94  3  13
north          NO      Margot Weber    4.5  .89  5  9
central         CT      Ann Stephens    5.7  .94  5  13
```

说明

这个示例中，选项-v 打印所有不含模式 Suan Chin 的行。选项-v 可用来删除输入文件中的特定条目。如果要真正删除这些条目，就要把 grep 的输出重定向到一个临时文件，然后把这个临时文件的名字改成原始文件的名字，就像这样：

```
grep -v 'Suan Chin' datafile > temp
mv temp datafile
```

记住，必须使用临时文件来重定向源自 datafile 的输出。如果您从 datafile 重定向到 datafile，shell 就会“摧毁” datafile(请参见 1.6.6 节中的“重定向”)。

范例 4-25

```
% grep -l 'SE' *
datafile
datebook
```

说明

选项-l 使 grep 只输出包含模式的文件名，而不输出文本行。

范例 4-26

```
% grep -c 'west' datafile
3
```

说明

选项-c 让 grep 打印出含有模式的行的数目。这个数字并不代表模式的出现次数。例如，即使 west 在某行中出现了 3 次，这行也只计一次。

范例 4-27

```
% grep -w 'north' datafile
north      NO      Margot Weber      4.5 .89   5   9
```

说明

选项-w 使 grep 只查找作为一个词，而不是词的一部分出现的模式^③。这条命令只打印包含词 north 的行，而不打印那些在 northwest、northeast 等中出现 north 的行。

范例 4-28

```
% echo $LOGNAME
lewis
% grep -i "$LOGNAME" datafile
southwest      SW      Lewis Dalsass      2.7 .8   2   18
```

说明

打印 shell 的环境变量 LOGNAME 的值，这个值记录了用户的登录名。即使变量被括在双引号之间，也会被 shell 展开，而且，如果有变量的值中有多个词时，用于分隔它们的空白符也可免受 shell 的解释。如果用的是单括号，就不会发生变量替换，也就是说，打印结果是\$LOGNAME。

4.4 grep 与管道

grep 的输入并不一定都是文件，它也常常从管道读取输入。

范例 4-29

```
% ls -l
drwxrwxrwx  2 ellie  2441 Jan  6 12:34  dir1
-rw-r--r--  1 ellie  1538 Jan  2 15:50  file1
-rw-r--r--  1 ellie  1539 Jan  3 13:36  file2
drwxrwxrwx  2 ellie  2341 Jan  6 12:34  grades
```

^③ 这里所说的词是指一个字母或数字字符序列，它始于行首或紧跟在空白符后，以空白符、标点或换行符结束。

```
% ls -l | grep '^d'
drwxrwxrwx 2 ellie 2441 Jan 6 12:34 dir1
drwxrwxrwx 2 ellie 2341 Jan 6 12:34 grades
```

说明

ls 命令的输出通过管道传给 grep。输出结果中以字母 d 开头的所有行都被打印出来，也就是说，所有目录被打印出来。

grep 回顾

表 4-3 中给出了 grep 命令的一些示例，并描述了它们执行的操作。

表 4-3 grep 回顾

grep 命令	命令执行的操作
grep '<Tom>' file	打印包含词 Tom 的行
grep 'Tom Savage' file	打印包含 Tom Savage 的行
grep '^Tommy' file	打印以 Tommy 开头的行
grep '\.bak\$' file	打印以.bak 结尾的行。单引号保护了美元符号(\$)，使之不被 shell 解释
grep '[Pp]yramid' *	打印当前工作目录下所有文件中包含 pyramid 或 Pyramid 的行
grep '[A-Z]' file	打印所有至少包含一个大写字母的行
grep '[0-9]' file	打印所有至少包含一个数字的行
grep '[A-Z]...[0-9]' file	打印包含这样一个模式的行，该模式以大写字母开头、数字结尾、共有五个字符
grep -w '[tT]est' files	打印包含词 Test 或 test 的行
grep -s 'Mark Todd' file	查找包含 Mark Todd 的行，但不打印找到的行。可用于检查 grep 的退出状态
grep -v 'Mary' file	打印所有不含 Mary 的行
grep -i 'sam' file	打印所有包含 sam 的行，sam 的各种大小写形式都可以(如：SAM、sam、Sam、sAm)
grep -l 'Dear Boss' *	列出所有包含 Dear Boss 的文件名
grep -n 'Tom' file	在每个匹配行前面加上行号
grep "\$name" file	展开变量 name 的值，打印包含该值的所有行。必须用双引号
grep '\$5' file	打印包含\$5 的行。必须用单引号
ps -ef grep '^ *user1'	把 ps -ef 的输出经管道发给 grep，在行首查找 user1，即使 user1 前面有多个空格也行

4.5 egrep(扩展的 grep)

使用 egrep 的主要好处是它在 grep 提供的正则表达式元字符集的基础上增加了更多的

元字符(参见表 4-4)。但是, egrep 不允许使用\()和\{\}(如果使用的是 Linux 系统, 请参考 GNU 的 grep -E 命令)。

表 4-4 egrep 使用的正则表达式元字符

元字符	功 能	示 例	匹 配 对 象
^	行首定位符	'^love'	匹配所有以 love 开头的行
\$	行尾定位符	'love\$'	匹配所有以 love 结尾的行
.	匹配一个字符	'l.e'	匹配包含一个 l, 后跟两个字符, 再跟一个 e 的行
*	匹配零个或多个前导字符	'*love'	匹配包含跟在零个或多个空格后的模式 love 的行
[]	匹配一组字符中任一个	'[L]ove'	匹配包含 love 或 Love 的行
[^]	匹配不在指定字符组内的字符	'[^A-KM-Z]ove'	匹配包含 ove、但 ove 前面那个字符既不在 A 到 K 之间、也不在 M 到 Z 之间的行
egrep 新增的元字符			
+	匹配一个或多个加号前的字符	'[a-z]+ove'	匹配一个或多个小写字母后跟 ove 的字符串。将找出 move、approve、love、behoove 等
?	匹配零个或一个前导字符	'lo?ve'	匹配 l 后跟一个或零个字母 o 以及 ve 的字符串。将找到 love 或 lve
a b	匹配 a 或 b	'love hate'	匹配 love 和 hate 这两个表达式之一
()	字符组	'love(able ly)(ov)+'	匹配 lovable 或 lovely 匹配 ov 的一次或多次出现

4.5.1 egrep 示例

下面这些示例只介绍 egrep 如何使用扩展集中那部分新的正则表达式元字符。之前给出的 grep 示例已经说明了标准元字符的用法, 和在 egrep 中的用法相同。egrep 所用的命令行参数也和 grep 的一样。

范例 4-30

```
% egrep 'NW|EA' datafile
```

northwest	NW	Charles Main	3.0	.98	3	34
eastern	EA	TB Savage	4.4	.84	5	20

说明

打印包含表达式 NW 或 EA 的行。

范例 4-31

```
% egrep '3+' datafile
```

northwest	NW	Charles Main	3.0	.98	3	34
western	WE	Sharon Gray	5.3	.97	5	23
northeast	NE	AM Main Jr.	5.1	.94	3	13
central	CT	Ann Stephens	5.7	.94	5	13

% cat datafile						
northwest	NW	Charles Main	3.0	.98	3	34
western	WE	Sharon Gray	5.3	.97	5	23
southwest	SW	Lewis Dalsass	2.7	.8	2	18
southern	SO	Suan Chin	5.1	.95	4	15
southeast	SE	Patricia Hemenway	4.0	.7	4	17
eastern	EA	TB Savage	4.4	.84	5	20
northeast	NE	AM Main Jr.	5.1	.94	3	13
north	NO	Margot Weber	4.5	.89	5	9
central	CT	Ann Stephens	5.7	.94	5	13

说明

打印所有包含一个或多个数字 3 的行。

范例 4-32

```
% egrep '2\.?[0-9]' datafile
```

western	WE	Sharon Gray	5.3	.97	5	23
southwest	SW	Lewis Dalsass	2.7	.8	2	18
eastern	EA	TB Savage	4.4	.84	5	20

说明

打印所有包含数字 2，后面跟零个或一个句点，再跟一个数字的行，将匹配 2.5、25、29、2.3 等。

范例 4-33

```
% egrep '(no)+' datafile
```

northwest	NW	Charles Main	3.0	.98	3	34
northeast	NE	AM Main Jr.	5.1	.94	3	13
north	NO	Margot Weber	4.5	.89	5	9

说明

打印连续出现一个或多个模式组 no 的行，将匹配 no、nono、nonononono 等。

范例 4-34

```
% egrep 'S(h|u)' datafile
```

western	WE	Sharon Gray	5.3	.97	5	23
southern	SO	Suan Chin	5.1	.95	4	15

说明

打印所有包含字母 S，后跟 h 或 u 的行，将匹配 Sharon 和 Suan。

范例 4-35

```
% egrep 'Sh|u' datafile
```

western	WE	Sharon Gray	5.3	.97	5	23
---------	----	-------------	-----	-----	---	----

southern	SO	Suan Chin	5.1	.95	4	15
southwest	SW	Lewis Dalsass	2.7	.8	2	18
southeast	SE	Patricia Hemenway	4.0	.7	4	17

说明

打印所有包含表达式 Sh 或 u 的行。将匹配 Sharon 或 southern。

4.5.2 egrep 回顾

表 4-5 中给出了 egrep 命令的示例，并说明了它们所执行的操作。

表 4-5 egrep 回顾

命 令	命令执行的操作
egrep '^ +' file	打印以一个或多个空格开头的行
egrep '^ *' file	打印以零个或多个空格开头的行 ^④
egrep '(Tom Dan) Savage' file	打印包含 Tom Savage 或 Dan Savage 的行
egrep '(ab)+' file	打印出现一个或多个 ab 的行
egrep '^X[0-9]?' file	打印以 X 开头， X 后面跟零个或一个数字的行
egrep 'fun.\$' *	打印当前工作目录下所有文件中以 fun. 结尾的行 ^④
egrep '[A-Z]+-' file	打印包含一个或多个大写字母的行
egrep '[0-9]' file	打印包含数字的行 ^④
egrep '[A-Z]...[0-9]' file	打印包含这样一个模式的行，该模式以大写字母开头、数字结尾、中间 3 个任意字符，共计 5 个字符 ^④
egrep '[tT]est' files	打印包含 Test 或 test 的行 ^④
egrep '(Susan Jean)Doe' file	打印包含 Susan Doe 或 Jean Doe 的行 ^④
egrep -v 'Mary' file	打印所有不含 Mary 的行 ^④
egrep -i 'sam' file	打印所有包含 sam 的行， sam 的各种大小写形式都可以(如： SAM、 sam、 Sam、 sAm) ^④
egrep -l 'Dear Boss' *	列出所有包含 Dear Boss 的文件名 ^④
egrep -n 'Tom' file	在每个匹配行前面加上行号 ^④
egrep -s '\$name' file	展开变量 name 的值，查找它。但是不打印结果。可用来检查 egrep 的退出状态 ^④

4.6 fgrep(固定的 grep 或快速的 grep)

fgrep 命令的运行方式与 grep 类似，但它不对任何正则表达式元字符做特殊处理。所

④ egrep 和 grep 对该模式的处理方式相同。

有字符都只代表它们自己：脱字符就是脱字符，美元符就是美元符，全部如此(如果使用的是 Linux 系统，请参考 GNU 的 grep -F 命令)。

范例 4-36

```
fgrep '[A-Z]****[0-9]..$5.00' file
```

说明

查找文件中所有包含字符串[A-Z]****[0-9]..\$5.00 的行。所有字符都代表它们本身，没有什么特殊含义。

4.7 Linux 与 GNU grep

Linux 使用 GNU 版本的 grep，其功能大部分与 grep 相同，只是有些方面做得更好。除 POSIX 字符(参见表 4-7 和表 4-8)外，还包含了很多新的选项，如-G、-E、-F 和-P 等使用普通 grep 的选项，另外还有 egrep 和 fgrep^⑤的功能。



基本正则表达式与扩展正则表达式

GNU grep 命令支持与 UNIX grep 相同的正则表达式元字符(参见表 4-7)。同时，修改了部分元字符(参见表 4-8)搜索与行显示方式。例如，可以提供选项来关闭大小写敏感、显示行号、显示文件名等。

有两种版本的正则表达式元字符：基本元字符和扩展元字符。标准版本的 GNU grep(grep -G)使用基本集(参见表 4-7)，egrep(或 grep -E)使用扩展集(参见表 4-8)。对 GNU grep，两种版本均可用。基本集包括^, \$, ., *, [], [^], < >，和\().

另外，GNU grep 识别\b、\w 和\W 以及一种新类别：POSIX 元字符(参见表 4-9)。

以-E 选项使用 GNU grep，则扩展集(egrep)可用。即使没有-E 选项，默认设置的标准 grep 也可以使用扩展集中的元字符。仅需要对这些元字符前置一个反斜线^⑥。例如，扩展集元字符为

?，+，{ }，|，()

扩展集中的元字符对标准 grep 来讲没有任何特殊意义，除非在它们前面以如下方式加上一个反斜线：

\?，\+，\{ }，\|，\(\)

GNU grep 的使用格式参见表 4-6。

⑤ 递归使用 grep，参见附录 A 中关于 GNU grep 和 xargs 的相关内容。

⑥ 在任何版本的 grep 中，可以使用反斜线来引用元字符以关闭它的特殊含义。

表 4-6 GNU grep

格 式	含 义
grep 'pattern' filename(s)	基本的 RE 元字符(默认)
grep -G 'pattern' filename(s)	含义同上(默认)
grep -E 'pattern' filename(s)	扩展 RE 元字符
grep -F 'pattern' filename	非 RE 元字符
grep -P 'pattern' filename	将模式解释为 Perl RE

表 4-7 GNU 版本 grep 的正则表达式元字符基本集

元字符	功 能	示例	匹 配 内 容
^	行首定位符	^love	匹配所有以 love 开头的行
\$	行尾定位符	love\$	匹配所有以 love 结尾的行
.	匹配单个字符	l.e	匹配包含一个 l, 后接两个字符, 再接一个 e 的行
*	匹配零个或多个字符	*love	匹配以零个或多个空格开始, 后跟 love 模式的行
[]	匹配集合中的一个字符	[L]ove	匹配包含 love 或 Love 的行
[^]	匹配非集合中的一个字符	[^A-K]ove	匹配包含不以 A 至 K 之间的某个字符开头, 后接 ove 的行
\< ^⑦	词首定位符	\<love	匹配包含以 love 开头的词的行
\>	词尾定位符	love\>	匹配包含以 love 结尾的词的行
\(..) ^⑧	标签匹配字符	\(love)able	寄存器中以标签标记的部分, 以数字 1 记录。将来引用时, 用\1 重复该模式。最多可以使用九个标签, 模式最左侧部分为第一个标签。例如, 模式 love 保存在寄存器 1 中, 将来以\1 进行引用。
x\{m\}	字符 x 重复 m 次	o\{5\}	匹配字母 o 出现 5 次的行
x\{m,\}	字符 x 重复至少 m 次	o\{5,\}	匹配字母 o 出现至少 5 次的行
x\{m,n\} ^⑨	字符 x 重复 m 到 n 次	o\{5,10\}	匹配字母 o 出现 5~10 次的行
\w	所有字母与数字, 称为字符[a-zA-Z0-9_]	\w*e	匹配一个 \w 后跟零个或多个字符, 最后接一个 e
\W	所有字母与数字之外的字符, 称为非字符[^a-zA-Z0-9_]	love\W+	匹配 love 后接一个或多个非字符(., ? 等)
\b	词边界	\blove\b	仅匹配 love 这个单词

⑦ 除非使用反斜线, 否则即使是使用 grep -E 和 GNU egrep 也不会工作。而在 UNIX egrep 上无论如何也不会工作。

⑧ 这些元字符确属扩展集的一部分。之所以放在这里是因为它们在使用反斜线的情况下能够在 UNIX grep 和 GNU regular grep 上工作。它们根本不会在 UNIX egrep 上工作。

⑨ 所有版本的 UNIX 及所有模式匹配工具均不支持元字符\{ \}, 通常在 vi 和 grep 中使用它们。它们根本不能在 UNIX egrep 上工作。

表 4-8 egrep 与 grep -E 使用的扩展集

元字符	功 能	实 例	匹 配 内 容
+	匹配一个或多个前导字符	[a-z]+ove	匹配一个或多个小写字母，后跟 ove。如 move, approve, love, behoove 等
?	匹配零个或一个前导字符	lo?ve	匹配一个 l 后跟一个或零个字母 o 的模式，如 love 或 lve
a b c	匹配 a 或 b 或 c	love hate	匹配其中的一个表达式，love 或 hate
()	组字符	love(able rs) (ov)+	匹配 loveable 或 lovers 匹配一个或多个连续的 ov
(..) (...) \l ②	标签匹配字符	\(love\)\ing	寄存器中以标签标记的部分，以数字 l 记录。将来引用时，用 \l 重复该模式。最多可以使用 9 个标签，模式最左侧部分为第一个标签。例如，模式 love 保存在寄存器 1 中，将来以 \l 进行引用
x{m}	字符 x 重复 m 次	o\{5\}	匹配字母 o 出现 5 次的行
x{m,}	字符 x 重复至少 m 次	o\{5,\}	匹配字母 o 出现至少 5 次的行
x{m,n} ③	字符 x 重复 m 到 n 次	o\{5,10\}	匹配字母 o 出现 5~10 次的行

POSIX 类 POSIX(可移植操作系统接口, the Portable Operating System Interface)是一个保证程序能够跨操作系统移植的工业标准。为了实现可移植性, POSIX 认可不同的国家和地区在字符编码、货币表示以及时间和日期的表示方式上的不同。为处理不同类型的字符, POSIX 在基本正则表达式与扩展正则表达式的基础上加入了表 4-9 所示的括号字符类的字符。

这种字符类, 例如, [:alnum:] 是 A-Za-z0-9 的另一种表达方式。为使用这种字符类, 它必须使用另外一对括号进行引用以将其标识为一个正则表达式。例如, A-Za-z0-9 本身并不是正则表达式, 但[A-Za-z0-9]是。同样地, [:alnum:] 应写作 [[:alnum:]]。使用第一种形式[A-Za-z0-9]与使用括号形式的 [[:alnum:]] 之间的差别在于第一种形式依赖于 ASCII 字符编码, 而第二种形式可以在该类中表示来自其他语言的字符, 如瑞典语中的 ring 字符和德语中的元音变音字符(umlaut)。

表 4-9 括号字符类

括 号 类	含 义
[:alnum:]	字母与数字两种字符
[:alpha:]	字母字符
[:cntrl:]	控制字符

⑩ 这种标签与引用在 UNIX egrep 上不能工作。

⑪ 所有版本的 UNIX 及所有模式匹配工具均不支持元字符{\}; 通常在 vi 和 grep 中使用它们。它们根本不能在 UNIX egrep 上工作。

(续表)

括号类	含 义
[:digit:]	数字字符
[:graph:]	非空字符(不包含空格、控制字符等)
[:lower:]	小写字母
[:print:]	与[:graph:]类似，但包含空格字符
[:punct:]	标点字符
[:space:]	所有的空白字符(换行符，空格符，制表符)
[:upper:]	大写字母
[:xdigit:]	十六进制数字字符(0-9a-fA-F)

范例 4-37

```

1 % grep '[[[:space:]]\.[[:digit:]][:space:]]' datafile
southwest SW Lewis Dalsass      2.7   .8   2   18
southeast  SE Patricia Hemenway  4.0   .7   4   17

2 % grep -E '[[[:space:]]\.[[:digit:]][:space:]]' datafile
southwest SW Lewis Dalsass      2.7   .8   2   18
southeast  SE Patricia Hemenway  4.0   .7   4   17

3 % egrep '[[[:space:]]\.[[:digit:]][:space:]]' datafile
southwest SW Lewis Dalsass      2.7   .8   2   18
southeast  SE Patricia Hemenway  4.0   .7   4   17

```

说明

1, 2, 3 适用于所有 grep(而非 fgrep)的 Linux 变体，支持 POSIX 括号字符类集合。每个例子中，grep 搜索一个空格字符、一个句点、一个数字和另一个空格字符。

4.8 带正则表达式的 GNU 基本 grep(grep -G)

基本 grep 将它的模式解释为基本正则表达式。本节中所有的 UNIX 基本 grep 的例子同样也适用于 GNU 版本的基本 grep，以及 grep -G 或 grep '--basic-regexp'。

下面范例中使用的元字符不能在基本 UNIX grep 中找到。本节中的例子使用下面的数据文件。

范例 4-38

```
% grep NW datafile          或
% grep -G NW datafile
```

northwest	NW	Charles Main	3.0	.98	3	34
-----------	----	--------------	-----	-----	---	----

% cat datafile						
northwest	NW	Charles Main	3.0	.98	3	34
western	WE	Sharon Gray	5.3	.97	5	23
southwest	SW	Lewis Dalsass	2.7	.8	2	18
southern	SO	Suan Chin	5.1	.95	4	15
southeast	SE	Patricia Hemenway	4.0	.7	4	17
eastern	EA	TB Savage	4.4	.84	5	20
northeast	NE	AM Main Jr.	5.1	.94	3	13
north	NO	Margot Weber	4.5	.89	5	9
central	CT	Ann Stephens	5.7	.94	5	13

说明

打印文件 datafile 中所有包含正则表达式 NW 的行。

范例 4-39

```
% grep '^n\w*\W' datafile
northwest      NW      Charles Main      3.0  .98  3  34
northeast      NE      AM Main Jr.      5.1  .94  3  13
```

说明

打印以一个 n 开始，后面跟零个或多个的字母数字混合的字符[a-zA-Z0-9_]，后面跟一个非字母数字的字符[^a-zA-Z0-9_]。 \w 和 \W 是 grep 的 GNU 变体中的标准单词元字符。

范例 4-40

```
% grep '\bnorth\b' datafile
north      NO      Margot Weber      4.5  .89  5  9
```

说明

打印包含词 north 的行。 \b 是一个词分界符。在所有 grep 的 GNU 变体上它可以用来代替词定位符(<>)。

4.9 grep -E 或 egrep(GNU 扩展 grep)

使用扩展 grep 的主要优势是有附加的正则表达式元字符(参见表 4-10)加入到基本集中。通过-E 扩展，GNU grep 可以使用这些新的元字符。

表 4-10 egrep 的正则表达式元字符

元字符	功 能	实 例	匹 配 内 容
^	行首定位符	^love	匹配所有以 love 开头的行
\$	行尾定位符	love\$	匹配所有以 love 结尾的行
.	匹配单个字符	l..e	匹配包含一个 l, 后接两个字符, 再接一个 e 的行
*	匹配零个或多个字符	*love	匹配以零个或多个空格开始, 后跟 love 模式的行
[]	匹配集合中的一个字符	[Ll]ove	匹配包含 love 或 Love 的行
[^]	匹配非集合中的一个字符	[^A-KM-Z]ove	匹配不包含以 A 至 K 或 M 至 Z 之间的某个字符开头, 后接 ove 的行

grep -E 或 egrep 新增元字符

+	匹配一个或多个前导字符	[a-z]+ove	匹配一个或多个小写字母, 后跟 ove。如 move, approve, love, behoove 等
?	匹配零个或一个前导字符	lo?ve	匹配一个 l 后跟一个或零个字母 o, 如 love 或 lve
a b	匹配 a 或 b	love hate	匹配其中的一个表达式, love 或 hate
()	组字符	love(able ly) (ov)+	匹配 loveable 或 lovely。 匹配一个或多个 ov 模式
x\{m\}	字符 x 重复 m 次	o\{5\}	匹配字母 o 出现 5 次的行
x\{m,n\}	字符 x 重复至少 m 次	o\{5,\}	匹配字母 o 出现至少 5 次的行
x\{m,n\} ^⑫	字符 x 重复 m~n 次	o\{5,10\}	匹配字母 o 出现 5~10 次的行
\w	所有字母与数字, 称为词字符[a-zA-Z0-9_]	\w*w	匹配一个 l 后跟零个或多个词字符, 最后是一个 e
\W	所有字母与数字之外的字符, 称为非词字符[^a-zA-Z0-9]	\W\w*	匹配一个非词字符(\W), 后跟零个或多个词字符(\w)
\b	词边界	\blove\b	仅匹配词 love

4.9.1 grep -E 和 egrep 实例

以下的例子示意了 grep -E 和 egrep 使用扩展集正则表达式元字符的方式。首先以一个 grep 实例示意标准元字符的用法, 这些标准元字符同样适用于 egrep。基本 GNU grep(grep -G) 可以使用任何附加的元字符, 只需在特殊元字符前加上一个反斜线。

^⑫ 所有版本的 UNIX 及所有模式匹配工具均不支持元字符{}; 通常在 vi 和 grep 中使用它们。它们根本不能在 UNIX egrep 上工作。

以下的范例显示了 grep 三种变体是如何完成同样的任务的。

本节所有的范例均使用下面的 datafile，为方便查阅，它将周期地重复出现。

% cat datafile						
northwest	NW	Charles Main	3.0	.98	3	34
western	WE	Sharon Gray	5.3	.97	5	23
southwest	SW	Lewis Dalsass	2.7	.8	2	18
southern	SO	Suan Chin	5.1	.95	4	15
southeast	SE	Patricia Hemenway	4.0	.7	4	17
eastern	EA	TB Savage	4.4	.84	5	20
northeast	NE	AM Main Jr.	5.1	.94	3	13
north	NO	Margot Weber	4.5	.89	5	9
central	CT	Ann Stephens	5.7	.94	5	13

范例 4-41

```

1 % egrep 'NW|EA' datafile
  northwest      NW      Charles Main      3.0  .98   3   34
  eastern        EA      TB Savage       4.4  .84   5   20

2 % grep -E 'NW|EA' datafile
  northwest      NW      Charles Main      3.0  .98   3   34
  eastern        EA      TB Savage       4.4  .84   5   20

3 % grep 'NW|EA' datafile

4 % grep 'NW\|EA' datafile
  northwest      NW      Charles Main      3.0  .98   3   34
  eastern        EA      TB Savage       4.4  .84   5   20

```

说明

1. 打印包含表达式 NW 或 EA 的行。本例中使用的是 egrep。如果没有 GNU 版本的 grep，则用 egrep 代替。
2. 本例中，使用了带-E 选项的 GNU grep 以将扩展元字符包括在内。与 egrep 相同。
3. 标准的 grep 通常并不支持扩展正则表达式；竖线是扩展正则表达式用作间隔的元字符。标准 grep 不能识别它，因此搜索显式模式'NW|EA'，没有匹配，没有打印输出。
4. 如果在 GNU 标准 grep(grep -G)的元字符前加上一个反斜线，则与 egrep 和 grep -E 一样，该元字符将被解释为一个扩展正则表达式。

范例 4-42

```

% egrep '3+' datafile
% grep -E '3+' datafile
% grep '3\+' datafile
  northwest      NW      Charles Main      3.0  .98   3   34
  western        WE      Sharon Gray      5.3  .97   5   23
  northeast      NE      AM Main Jr.     5.1  .94   3   13
  central        CT      Ann Stephens    5.7  .94   5   13

```

% cat datafile						
northwest	NW	Charles Main	3.0	.98	3	34
western	WE	Sharon Gray	53	.97	5	23
southwest	SW	Lewis Dalsass	2.7	.8	2	18
southern	SO	Suan Chin	5.1	.95	4	15
southeast	SE	Patricia Hemenway	4.0	.7	4	17
eastern	EA	TB Savage	4.4	.84	5	20
northeast	NE	AM Main Jr.	5.1	.94	3	13
north	NO	Margot Weber	4.5	.89	5	9
central	CT	Ann Stephens	5.7	.94	5	13

说明

打印所有包含一个或多个 3 的行。

范例 4-43

```
% egrep '2\.\?[0-9]' datafile
% grep -E '2\.\?[0-9]' datafile
% grep '2\.\?\d' datafile
western          WE    Sharon Gray      5.3   .97   5   23
southwest        SW    Lewis Dalsass   2.7   .8    2   18
eastern          EA    TB Savage       4.4   .84   5   20
```

说明

打印所有包含一个 2，后跟 0 个或 1 个句点，再接一个 0~9 之间的数的行。

范例 4-44

```
% egrep '(no)+' datafile
% grep -E '(no)+' datafile
% grep '\(no\)\+' datafile
northwest        NW    Charles Main    3.0   .98   3   34
northeast        NE    AM Main Jr.   5.1   .94   3   13
north            NO    Margot Weber  4.5   .89   5   9
```

说明

打印包含一个或多个模式组 no 的行。

范例 4-45

```
% grep -E '\w+\W+[ABC]' datafile
northwest        NW    Charles Main    3.0   .98   3   34
southern         SO    Suan Chin     5.1   .95   4   15
northeast        NE    AM Main Jr.   5.1   .94   3   13
central          CT    Ann Stephens   5.7   .94   5   13
```

说明

打印所有包含一个或多个字母数字词字符(\w+), 后跟一个或多个非字母数字词字符(\W+), 再接上集合 ABC 中一个字母的行。

**范例 4-46**

```
% egrep 'S(h|u)' datafile
% grep -E 'S(h|u)' datafile
% grep 'S\h\|u\)' datafile
western          WE      Sharon Gray      5.3   .97   5   23
southern         SO      Suan Chin       5.1   .95   4   15
```

说明

打印所有包含一个 S 后跟一个 h 或 u(如 Sh 或 Su)的行。

范例 4-47

```
% egrep 'Sh|u' datafile
% grep -E 'Sh|u' datafile
% grep 'Sh\|u' datafile
western          WE      Sharon Gray      5.3   .97   5   23
southern         SO      Suan Chin       5.1   .95   4   15
southwest        SW      Lewis Dalsass    2.7   .8    2   18
southeast        SE      Patricia Hemenway 4.0   .7    4   17
```

说明

打印所有包含表达式 Sh 或 u 的行。

4.9.2 grep 变体的不规则形式

Linux 支持的 GNU grep 变体几乎等同于它们在 UNIX 上的同名 grep，但也不是完全相同。例如，Solaris 或 BSD UNIX 上的 egrep 版本不支持 3 种元字符：用于重复的{\}、标签字符(\)和词定位符<>。在 Linux 系统上，grep 和 grep -E 可以识别这 3 种元字符，但 egrep 不能识别<>。下面的范例示意了它们之间的差别，假定是在 UNIX 系统而不是 Linux 系统上运行 bash 或 tcsh，并在 shell 脚本中使用 grep 及 grep 族。

本节所有的范例使用下面的 datafile，为方便，它将周期地重复出现。

% cat datafile							
northwest	NW	Charles Main	3.0	.98	3	34	
western	WE	Sharon Gray	53	.97	5	23	
southwest	SW	Lewis Dalsass	2.7	.8	2	18	
southern	SO	Suan Chin	5.1	.95	4	15	
southeast	SE	Patricia Hemenway	4.0	.7	4	17	
eastern	EA	TB Savage	4.4	.84	5	20	
northeast	NE	AM Main Jr.	5.1	.94	3	13	
north	NO	Margot Weber	4.5	.89	5	9	
central	CT	Ann Stephens	5.7	.94	5	13	

范例 4-48

```
(Linux GNU grep)
1 % grep '<north>' datafile # 必须使用反斜杠
2 % grep '\<north\>' datafile
  north      NO      Margot Weber      4.5 .89 5 9
3 % grep -E '\<north\>' datafile
  north      NO      Margot Weber      4.5 .89 5 9
4 % egrep '\<north\>' datafile
  north      NO      Margot Weber      4.5 .89 5 9
(Solaris egrep)
5 % egrep '\<north\>' datafile
<no output; not recognized>
```

说明

1. 无论使用哪种 grep 变体，都必须在词定位符元字符前加上一个反斜线。
2. 这一次，grep 搜索一个以 north 开头并以之结尾的词。<代表词首定位符，>代表词尾定位符。
3. 带-E 选项的 grep 也能够识别词定位符。
4. GNU 版本的 egrep 可以识别词定位符。
5. 当使用 Solaris(SVR4)时，egrep 不能将词定位符识别为正则表达式元字符。

范例 4-49

```
(Linux GNU grep)
1 % grep 'w(es)t.*\1' datafile
  grep: Invalid back reference
2 % grep 'w\es\t.*\1' datafile
  northwest    NW    Charles Main      3.0 .98 3 34
3 % grep -E 'w(es)t.*\1' datafile
  northwest    NW    Charles Main      3.0 .98 3 34
4 % egrep 'w(es)t.*\1' datafile
  northwest    NW    Charles Main      3.0 .98 3 34
(Solaris egrep)
5 % egrep 'w(es)t.*\1' datafile
<no output; not recognized>
```

说明

1. 当使用标准 grep 时，扩展元字符()前必须加上反斜线，否则会出错。
2. 如果正则表达式 w\es\t 匹配成功，则模式 es 被存储在内存寄存器 1 中。该表达式含义是：如果找到了 west，则标记并保存模式 es，接着，搜索任意数目的字符后再次匹配 es(\1)的行，并打印该行。Charles 中的 es 在后向引用时被匹配成功。
3. 与前一个例子相同，不同之处在于带-E 选项的 grep 不用在()前加上一个反斜线。
4. GNU egrep 也使用不带反斜线的扩展元字符()。
5. 在 Solaris 系统上，egrep 不能识别任何形式的标签及后向引用。

% cat datafile						
northwest	NW	Charles Main	3.0	.98	3	34
western	WE	Sharon Gray	5.3	.97	5	23
southwest	SW	Lewis Dalsass	2.7	.8	2	18
southern	SO	Suan Chin	5.1	.95	4	15
southeast	SE	Patricia Hemenway	4.0	.7	4	17
eastern	EA	TB Savage	4.4	.84	5	20
northeast	NE	AM Main Jr.	5.1	.94	3	13
north	NO	Margot Weber	4.5	.89	5	9
central	CT	Ann Stephens	5.7	.94	5	13

范例 4-50

(Linux GNU grep)

```

1 % grep '\.[0-9]{2}\[^0-9]' datafile
  northwest    NW    Charles Main   3.0   .98   3   34
  western      WE    Sharon Gray   5.3   .97   5   23
  southern     SO    Suan Chin    5.1   .95   4   15
  eastern       EA   TB Savage    4.4   .84   5   20
  northeast    NE   AM Main Jr.   5.1   .94   3   13
  north        NO   Margot Weber  4.5   .89   5   9
  central       CT   Ann Stephens  5.7   .94   5   13

2 % grep -E '\.[0-9]{2}\[^0-9]' datafile
  northwest    NW    Charles Main   3.0   .98   3   34
  western      WE    Sharon Gray   5.3   .97   5   23
  southern     SO    Suan Chin    5.1   .95   4   15
  eastern       EA   TB Savage    4.4   .84   5   20
  northeast    NE   AM Main Jr.   5.1   .94   3   13
  north        NO   Margot Weber  4.5   .89   5   9
  central       CT   Ann Stephens  5.7   .94   5   13

3 % egrep '\.[0-9]{2}\[^0-9]' datafile
  northwest    NW    Charles Main   3.0   .98   3   34
  western      WE    Sharon Gray   5.3   .97   5   23
  southern     SO    Suan Chin    5.1   .95   4   15
  eastern       EA   TB Savage    4.4   .84   5   20
  northeast    NE   AM Main Jr.   5.1   .94   3   13
  north        NO   Margot Weber  4.5   .89   5   9
  central       CT   Ann Stephens  5.7   .94   5   13

  (Solaris egrep)
4 % egrep '\.[0-9]{2}\[^0-9]' datafile
<no output; not recognized with or without backslashes>

```

说明

- 扩展元字符{}用于重复。GNU 与 UNIX 版本的标准 grep 并不对该扩展元字符集求

值，除非括号前加上反斜线。整个表达式含义是：搜索一个句点`\.`，后跟一个 0~9 之间的数字`[0~9]`，如果此模式恰好重复两次`\{2\}`，后跟一个非数字`[^0~9]`。

- 2. 扩展 grep, grep -E 使用重复元字符`\{2\}`，不必像上面例子那样在前面加一个反斜线。
- 3. 因为 GNU egrep 和 grep-E 功能相同，所以该命令产生的输出与前一个例子相同。
- 4. 这是标准的 UNIX 版本的 egrep。无论是否加上反斜线，它都不能将花括号识别为一个扩展元字符。

4.10 固定的 grep(grep -F 和 fgrep)

fgrep 命令的行为与 grep 的类似，但它不能够识别任何正则表达式元字符的特殊意义。所有字符仅能代表它们自身。脱字号就是脱字号，美元符号就是美元符号，以此类推。带 -F 选项的 GNU grep 的行为与 fgrep 的完全一致。

范例 4-51

```
% fgrep '[A-Z]****[0-9]..$5.00' file 或  
% grep -F '[A-Z]****[0-9]..$5.00' file
```

说明

查找文件中所有包含字符串`[A-Z]****[0-9]..$5.00`的行。所有字符只代表它们本身，不表示任何特殊含义。

4.11 递归的 grep(rgrep,grep -R)

与 grep 族的成员不同，Linux 上的 rgrep 可以沿一个目录树递归而下。rgrep 有许多命令行选项并支持与标准 grep(grep -R)相同的元字符。附录 A 给出了 rgrep 的完整描述，也可键入 rgrep?-以获得在线帮助(UNIX 标准版本上不支持)。

范例 4-52

```
% grep -r 'Tom' ./dir  
% rgrep 'Tom' ./dir
```

说明

递归地搜索./dir 目录下包含字符串 Tom 的所有文件。

4.12 带选项的 GNU grep

grep 命令有许多可以控制其行为的选项。GNU 版本的 grep 又加入了许多新选项，同时对原来的一些选项提供了另外的选择方式。GNU grep 选项在 grep 所有不同的变体上都

能够工作，包括 grep -G, -E 和 -F，如表 4-11 所示。

表 4-11 所有变体(-G,-E 和-F)均可用的 GNU grep 选项

选 项	作 用
-# (# 是一个用来代表整数值的符号)	将匹配行前后#行的内容一同打印出来；也就是说，grep -2 pattern filename 将导致 grep 打印匹配行及匹配行的前两行和后两行
-A #, --after-context=#	打印匹配行后面#行的内容；也就是说，匹配行及它后面指定的#行内容
-B #, --before-context=#	打印匹配行前面#行的内容；也就是说，匹配行及它前面指定的#行内容
-C #, --context=#	等价于 -2 选项。打印匹配行的前两行和后两行
-V, --version	打印 grep 版本信息，版本信息应当包含在所有的 bug 报告中
-a, --text, --binary-files=text	将二进制文件当作文本文件处理
-b, --byte-offset	在输出的每行前显示偏移字节数
-c, --count	为每个输入文件打印成功匹配的行数。 -v 则打印一些未匹配的行数
-D action, --devices=action	如果输入文件为一个设备，如套接字或管道。则 action 默认从该设备读，就如同读一个普通文件一样。如果 action 为 skip，则该设备被忽略
-e PATTERN, --regexp=PATTERN	使用字面 PATTERN 作为模式；这对保护以-开头的模式非常有帮助
-f FILE, --file=FILE	从 FILE 中获得模式，每行一个。空文件包含 0 个模式，因此什么也不能匹配
--help--	显示有关 grep 命令行选项及错误报告地址的帮助信息，然后退出
-h, --no-filename	当搜索多个文件时，禁止输出文件名前缀
-i, --ignore-case	忽略模式和输入文件的大小写区别
-L, --files-without-match	仅打印所有未能匹配模式的文件名
-l, --files-with-matches	仅打印所有正确匹配模式的文件名
-m #, --max-count=#	如果文件是标准输入或正规文件，在找到指定数量(#)的匹配行后停止读文件
-n, --line-number	在匹配成功的输出行前加上行号作为前缀
-q, --quiet	禁止正规输出。可用来替代 -n
-r, -R, --recursive, --directories=recurse	对列出的目录，递归地读并处理这些目录中的所有文件；也就是指该目录下的所有目录
-s, --silent	禁止显示文件不存在或文件不可读的错误信息
-v, --revert-match	转换匹配性质，选择非匹配行

(续表)

选 项	作 用
-w, --word-regexp	仅选择包含词匹配的行。匹配词边界上包含字母、数字和下划线的字符串
-x, --line-regexp	仅选择精确匹配整行的那些匹配
-y	与已废除的-i 同义
-U, --binary	将文件作为二进制文件处理。仅有 MS-DOS 和 MS-Windows 支持该选项
-u, --unix-byte-offsets	报告 UNIX 风格的字节偏移。这个选项仅在同时使用-b 选项的情况下才有效；仅有 MS-DOS 和 MS-Windows 支持该选项
-Z, --null	在文件名的末尾放上 ASCII 空字符以取代换行符

4.13 带选项的 grep(UNIX 和 GNU)

grep 有许多可以控制其行为的选项。并非所有版本的 UNIX 都支持完全相同的选项，所以最好检查帮助手册以得到一个完整的列表。

本节所有的范例均使用下面的 **datafile**，为方便查阅，它将周期地重复出现。

% cat datafile						
northwest	NW	Charles Main	3.0	.98	3	34
western	WE	Sharon Gray	5.3	.97	5	23
southwest	SW	Lewis Dalsass	2.7	.8	2	18
southern	SO	Suan Chin	5.1	.95	4	15
southeast	SE	Patricia Hemenway	4.0	.7	4	17
eastern	EA	TB Savage	4.4	.84	5	20
northeast	NE	AM Main Jr.	5.1	.94	3	13
north	NO	Margot Weber	4.5	.89	5	9
central	CT	Ann Stephens	5.7	.94	5	13

范例 4-53

```
% grep -n '^south' datafile
3:southwest      SW      Lewis Dalsass      2.7  .8  2  18
4:southern       SO      Suan Chin       5.1  .95 4  15
5:southeast      SE      Patricia Hemenway 4.0  .7  4  17
```

说明

-n 选项在匹配模式成功的行前面加上该行的行号。

范例 4-54

```
% grep -i 'pat' datafile
southeast      SE      Patricia Hemenway  4.0  .7  4  17
```

说明

-i 选项关闭大小写敏感性。表达式 pat 中包含任意的大小写组合都没有关系。

范例 4-55

```
% grep -v 'Suan Chin' datafile
northwest      NW      Charles Main       3.0  .98   3  34
western        WE      Sharon Gray        5.3  .97   5  23
southwest      SW      Lewis Dalsass     2.7  .8    2  18
southeast       SE      Patricia Hemenway  4.0  .7    4  17
eastern         EA      TB Savage        4.4  .84   5  20
northeast       NE      AM Main Jr.      5.1  .94   3  13
north          NO      Margot Weber     4.5  .89   5  9
central         CT      Ann Stephens     5.7  .94   5  13
```

说明

在这里，-v 选项打印所有不包含模式 Suan Chin 的行。这个选项用于从输入文件中删除指定的一个入口。要实际删除此入口，应该将 grep 的输出重定向到一个临时文件，然后将临时文件的名字改回最初文件的名字，如下所示：

```
grep -v 'Suan Chin' datafile > temp
mv temp datafile
```

记得在将 datafile 的输出重定向时一定要使用临时文件。如果直接由 datafile 重定向到 datafile，shell 将摧毁该 datafile(参见 1.6.6 节中的“重定向”)。

范例 4-56

```
% grep -l 'SE' *
datafile
datebook
```

说明

-l 选项导致 grep 仅打印成功匹配模式的文件名，而不打印原文中的行。

范例 4-57

```
% grep -c 'west' datafile
3
```

说明

-c 选项导致 grep 打印成功匹配模式的行数。这并不是指该模式出现的次数。例如，如果一行中 west 出现了 3 次，则该行仅计数一次。

% cat datafile						
northwest	NW	Charles Main	3.0	.98	3	34
western	WE	Sharon Gray	5.3	.97	5	23
southwest	SW	Lewis Dalsass	2.7	.8	2	18
southern	SO	Suan Chin	5.1	.95	4	15
southeast	SE	Patricia Hemenway	4.0	.7	4	17
eastern	EA	TB Savage	4.4	.84	5	20
northeast	NE	AM Main Jr.	5.1	.94	3	13
north	NO	Margot Weber	4.5	.89	5	9
central	CT	Ann Stephens	5.7	.94	5	13

范例 4-58

```
% grep -w 'north' datafile
north          NO      Margot Weber    4.5 .89 5 9
```

说明

-w 选项导致 grep 查找作为一个词^⑬而非词的一部分存在的模式。仅打印包含词 north 的行，而不打印包含 northwest,northeast 等词行。

范例 4-59

```
% echo $LOGNAME
lewis
% grep -i "$LOGNAME" datafile
southwest     SW      Lewis Dalsass   2.7 .8 2 18
```

说明

打印了 shell 环境变量 LOGNAME 的值，它包含用户登录名。如果该变量被双引号引用，它还将被 shell 扩展。假如给该变量赋的值多于一个词，则 shell 进行解释时将屏蔽空白。如果是被单引号引用，则不会进行变量替换，也就是说，将会打印\$LOGNAME。

GNU grep 选项实例

除 UNIX grep 提供的选项外，GNU 版本提供了进一步精炼模式搜索输出结果的选项。本节所有的例子均使用下面的 datafile，为方便查阅，它将周期地重复出现。

% cat datafile						
northwest	NW	Charles Main	3.0	.98	3	34
western	WE	Sharon Gray	5.3	.97	5	23
southwest	SW	Lewis Dalsass	2.7	.8	2	18
southern	SO	Suan Chin	5.1	.95	4	15
southeast	SE	Patricia Hemenway	4.0	.7	4	17

^⑬ 词是一个字母数字字符序列，从行首开始或以空白为前导字符，以空白、标点符号或换行符结束。

TopSage.com
 (续表)

eastern	EA	TB Savage	4.4	.84	5	20
northeast	NE	AM Main Jr.	5.1	.94	3	13
north	NO	Margot Weber	4.5	.89	5	9
central	CT	Ann Stephens	5.7	.94	5	13

范例 4-60

```
% grep -V
grep (GNU grep) 2.2
```

Copyright (C) 1988, 92, 93, 94, 95, 96, 97 Free Software Foundation, Inc.
 This is free software; see the source for copying conditions. There is NO warranty;
 not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

说明

-v 选项列出了 grep 的版本和版权信息。任何发往 GNU 基金会的 bug 报告都应该包含版本信息。

范例 4-61

```
1 % grep -2 Patricia datafile
  southwest SW Lewis Dalsass 2.7 .8 2 18
  southern SO Suan Chin 5.1 .95 4 15
  southeast SE Patricia Hemenway 4.0 .7 4 17
  eastern EA TB Savage 4.4 .84 5 20
  northeast NE AM Main Jr. 5.1 .94 3 13
  2 % grep -C Patricia datafile
  southwest SW Lewis Dalsass 2.7 .8 2 18
  southern SO Suan Chin 5.1 .95 4 15
  southeast SE Patricia Hemenway 4.0 .7 4 17
  eastern EA TB Savage 4.4 .84 5 20
  northeast NE AM Main Jr. 5.1 .94 3 13
```

说明

- 在查找到匹配 Patricia 的行后, grep 显示该行及该行的前两行和后两行。
- C 选项与-2 相同。

```
% cat datafile
```

northwest	NW	Charles Main	3.0	.98	3	34
western	WE	Sharon Gray	5.3	.97	5	23
southwest	SW	Lewis Dalsass	2.7	.8	2	18
southern	SO	Suan Chin	5.1	.95	4	15
southeast	SE	Patricia Hemenway	4.0	.7	4	17
eastern	EA	TB Savage	4.4	.84	5	20
northeast	NE	AM Main Jr.	5.1	.94	3	13
north	NO	Margot Weber	4.5	.89	5	9
central	CT	Ann Stephens	5.7	.94	5	13

范例 4-62

```
% grep -A 2 Patricia datafile
southeast      SE    Patricia Hemenway   4.0   .7   4   17
eastern        EA    TB Savage          4.4   .84  5   20
northeast      NE    AM Main Jr.       5.1   .94  3   13
```

说明

在查找到匹配 Patricia 的行后，grep 显示该行及该行的后两行。

范例 4-63

```
% grep -B 2 Patricia datafile
southwest     SW    Lewis Dalsass    2.7   .8   2   18
southern       SO    Suan Chin       5.1   .95  4   15
southeast      SE    Patricia Hemenway 4.0   .7   4   17
```

说明

在查找到匹配 Patricia 的行后，grep 显示该行及该行的前两行。

范例 4-64

```
% grep -b '[abc]' datafile
0:northwest   NW    Charles Main     3.0   .98  3   34
39:western    WE    Sharon Gray      5.3   .97  5   23
76:southwest  SW    Lewis Dalsass   2.7   .8   2   18
115:southern   SO    Suan Chin       5.1   .95  4   15
150:southeast  SE    Patricia Hemenway 4.0   .7   4   17
193:eastern    EA    TB Savage       4.4   .84  5   20
228:northeast  NE    AM Main Jr.    5.1   .94  3   13
266:north     NO    Margot Weber    4.5   .89  5   9
301:central    CT    Ann Stephens   5.7   .94  5   13
```

说明

使用-b 选项，grep 在输出的每行前打印该行在输入文件中的偏移字节数

下面的两个范例不再使用 datafile，而是使用文件 negative 以示范-e 和-x 选项的用法。

```
% cat negative
```

-40 is cold.

This is line 1.

This is line 2.5

-alF are options to the ls command

范例 4-65

```
1 % grep -e '-alF' negative
-alF are options to the ls command
2 % grep --regexp=-40 negative
-40 is cold.
```

**说明**

1. 使用-e 选项, grep 将模式中所有的字符同等对待, 因此第一个长划线将不会被误当作选项。
2. 表示-e 的另一种方式是--regexp=pattern, 其中 pattern 是正则表达式; 在本例中, 正则表达式为-40。

范例 4-66

```
% grep -x -e '-40 is cold.' negative
-40 is cold.
```

说明

使用-x 选项, 除非搜索模式与整行内容完全一致, 否则 grep 不会匹配该行。使用-e 以将一个长划线作为搜索字符串的第一个字符。

本节余下的范例使用下面的 datafile 文件。

```
% cat datafile
```

northwest	NW	Charles Main	3.0	.98	3	34
western	WE	Sharon Gray	5.3	.97	5	23
southwest	SW	Lewis Dalsass	2.7	.8	2	18
southern	SO	Suan Chin	5.1	.95	4	15
southeast	SE	Patricia Hemenway	4.0	.7	4	17
eastern	EA	TB Savage	4.4	.84	5	20
northeast	NE	AM Main Jr.	5.1	.94	3	13
north	NO	Margot Weber	4.5	.89	5	9
central	CT	Ann Stephens	5.7	.94	5	13

范例 4-67

```
1 % cat repatterns
western
north
2 % grep -f repatterns datafile
northwest      NW      Charles Main      3.0      .98      3      34
western        WE      Sharon Gray       5.3      .97      5      23
northeast      NE      AM Main Jr.     5.1      .94      3      13
north          NO      Margot Weber    4.5      .89      5      9
```

说明

1. 显示文件 repatterns。它包含了 grep 的搜索模式, 这些搜索模式将与输入文件中的行进行匹配。western 和 north 是 grep 使用的搜索模式。

2. 在-f 选项后跟上一个文件名(本例中为 repatterns), grep 将从该文件获取搜索模式并与 datafile 中的行进行匹配。grep 搜索并打印所有包含模式 western 和 north 的行。

范例 4-68

```

1 % grep '[0-9]' datafile db
datafile:northwest NW      Charles Main       3.0 .98 3 34
datafile:western WE        Sharon Gray        5.3 .97 5 23
datafile:southwest SW      Lewis Dalsass     2.7 .8 2 18
datafile:southern SO       Suan Chin         5.1 .95 4 15
datafile:southeast SE      Patricia Hemenway 4.0 .7 4 17
datafile:eastern EA        TB Savage          4.4 .84 5 20
datafile:northeast NE      AM Main Jr.      5.1 .94 3 13
datafile:north NO          Margot Weber      4.5 .89 5 9
datafile:central CT        Ann Stephens     5.7 .94 5 13
db:123

2 % grep -h '[0-9]' datafile db
northwest      NW      Charles Main       3.0 .98 3 34
western        WE      Sharon Gray        5.3 .97 5 23
southwest      SW      Lewis Dalsass     2.7 .8 2 18
southern        SO      Suan Chin         5.1 .95 4 15
southeast       SE      Patricia Hemenway 4.0 .7 4 17
eastern         EA      TB Savage          4.4 .84 5 20
northeast       NE      AM Main Jr.      5.1 .94 3 13
north           NO      Margot Weber      4.5 .89 5 9
central         CT      Ann Stephens     5.7 .94 5 13
123

```

说明

- 如果列出的文件多于一个, grep 在输出每行内容之前先输出文件名。文件名为 datafile 和 db。
- 使用-h 选项, grep 禁止头部信息。也就是说, 不打印文件名。

范例 4-69

```

% grep -q Charles datafile      或
% grep --quiet Charles datafile
% echo $status
0

```

说明

quiet 选项禁止从 grep 输出。它用于退出状态恰为所需要的值时。当退出状态为 0 时, 说明 grep 查找到匹配的模式。

习题 1: grep 练习

(参考从本书合作站点下载的文件中名为 databook 的文件)。

Steve Blenheim:238-923-7366:95 Latham Lane, Easton, PA 83755:11/12/56:20300

Betty Boop:245-836-8357:635 Cutesy Lane, Hollywood, CA 91464:6/23/23:14500

Igor Chevsky:385-375-8395:3567 Populus Place, Caldwell, NJ 23875:6/18/68:23400

Norma Corder:397-857-2735:74 Pine Street, Dearborn, MI 23874:3/28/45:245700

Jennifer Cowan:548-834-2348:583 Laurel Ave., Kingsville, TX 83745:10/1/35:58900

Jon DeLoach:408-253-3122:123 Park St., San Jose, CA 04086:7/25/53:85100
 Karen Evich:284-758-2857:23 Edgecliff Place, Lincoln, NB 92743:7/25/53:85100
 Karen Evich:284-758-2867:23 Edgecliff Place, Lincoln, NB 92743:11/3/35:58200
 Karen Evich:284-758-2867:23 Edgecliff Place, Lincoln, NB 92743:11/3/35:58200
 Fred Fardbarkle:674-843-1385:20 Parak Lane, Duluth, MN 23850:4/12/23:780900
 Fred Fardbarkle:674-843-1385:20 Parak Lane, Duluth, MN 23850:4/12/23:780900
 Lori Gortz:327-832-5728:3465 Mirlo Street, Peabody, MA 34756:10/2/65:35200
 Paco Gutierrez:835-365-1284:454 Easy Street, Decatur, IL 75732:2/28/53:123500
 Ephram Hardy:293-259-5395:235 CarltonLane, Joliet, IL 73858:8/12/20:56700
 James Ikeda:834-938-8376:23445 Aster Ave., Allentown, NJ 83745:12/1/38:45000
 Barbara Kertz:385-573-8326:832 Ponce Drive, Gzary, IN 83756:12/1/46:268500
 Lesley Kirstin:408-456-1234:4 Harvard Square, Boston, MA 02133:4/22/62:52600
 William Kopf:846-836-2837:6937 Ware Road, Milton, PA 93756:9/21/46:43500
 Sir Lancelot:837-835-8257:474 Camelot Boulevard, Bath, WY 28356:5/13/69:24500
 Jesse Neal:408-233-8971:45 Rose Terrace, San Francisco, CA 92303:2/3/36:25000
 Zippy Pinhead:834-823-8319:2356 Bizarro Ave., Farmount, IL 84357:1/1/67:89500
 Arthur Putie:923-835-8745:23 Wimp Lane, Kensington, DL 38758:8/31/69:126000
 Popeye Sailor:156-454-3322:945 Bluto Street, Anywhere, USA 29358:3/19/35:22350
 Jose Santiago:385-898-8357:38 Fife Way, Abilene, TX 39673:1/5/58:95600
 Tommy Savage:408-724-0140:1222 Oxbow Court, Sunnyvale, CA 94087:5/19/66:34200
 Yukio Takeshida:387-827-1095:13 Uno Lane, Ashville, NC 23556:7/1/29:57000
 Vinh Tranh:438-910-7449:8235 Maple Street, Wilmington, VM 29085:9/23/63:68900

1. 打印所有包含字符串 San 的行。
2. 打印所有名字以 J 开头的行。
3. 打印所有以 700 结尾的行。
4. 打印所有不包含 834 的行。
5. 打印所有生日在 December 的行。
6. 打印所有电话号码区号为 408 的行。
7. 打印所有包含一个大写字母，后跟 4 个小写字母，一个逗号，一个空格和一个大写字母的行。
8. 打印所有最后一个名字以 K 或 k 开始的行。
9. 打印所有薪水为 6 位数字的行，前导是一个行号。
10. 打印包含 Lincoln 或 lincoln(注意，grep 不区分大小写)的行。