

## Практическая работа «Регулярные выражения»

Регулярные выражения - это шаблоны, интерпретируемые определенными командами. Шаблоны заключены в одинарные кавычки, чтобы оболочка не интерпретировала их. Регулярные выражения отличаются от глобализации, которая является функцией, выполняемой оболочкой, и поэтому не связана напрямую ни с какой командой.

Команда **grep** используется для демонстрации использования регулярных выражений. Более подробная информация об этой команде приведена ниже в этом разделе.

Существует два типа регулярных выражений - базовое и расширенное. В то время как основные выражения интерпретируются большинством команд, расширенные выражения могут использоваться вместе с опцией в командах, которые поддерживают их интерпретацию.

Чтобы отобразить имена файлов в каталоге **/etc**, которые соответствуют регулярному выражению **'rc [3-6] \*\d'**, выполните следующую команду:

```
ls /etc | grep 'rc[3-6]*\d'
```

Регулярное выражение, используемое в приведенном выше примере, интерпретируется как совпадение с буквенными символами «**rc**», за которыми следуют цифры в диапазоне **3–6**, повторяемые ноль или более раз, литерал «**.**» и буква «**d**».

Чтобы отобразить имена файлов в каталоге **/etc**, которые соответствуют регулярному выражению **'rc [3-6]+\d'**, выполните следующую команду:

```
ls /etc | grep -E 'rc[3-6]+\d'
```

Регулярное выражение, показанное в примере выше, является примером *расширенного регулярного* выражения. Он интерпретируется как соответствие буквенным символам «**rc**», за которыми следуют один или несколько цифровых символов в диапазоне **3–6**, литерал «**.**» и буква «**d**».

Команда **egrep** является альтернативой команде **grep -E**. Чтобы соответствовать тому же расширенному регулярному выражению, которое использовалось в предыдущем примере, выполните следующую команду:

```
ls /etc | egrep 'rc[3-6]+\d'
```

Чтобы найти вхождения шаблона **'root'** в начале строки в файле **/etc/passwd**, выполните следующую команду:

```
grep '^root' /etc/passwd
```

Чтобы найти вхождения шаблона **'bash'** в конце строки в файле **/etc/passwd**, выполните следующую команду:

```
grep 'bash$' /etc/passwd
```

Символ **."** используется для соответствия любому символу, кроме символа новой строки. Чтобы сопоставить шаблон «**r.t**» (т. е. любой шаблон, содержащий «**r**», за которым следуют ровно два символа, а затем «**t**») в файле **/etc/passwd**, выполните следующую команду:

```
grep 'r..t' /etc/passwd
root:x:0:0:root:/root:/bin/bash
operator:x:1000:37::/root:
```

Обозначение скобки **[]**, которое используется при глобализации файлов, имеет аналогичную реализацию в регулярных выражениях:

```
grep ':[01234]:' /etc/passwd
```

Результат:

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
```

Размещение в начале нотации **[]** символа «**^**» обозначает отрицание сопоставляемого шаблона. Следующий шаблон будет соответствовать одному символу между символами **":"**, который не является **0, 1, 2, 3, 4** или **x**:

```
grep ':[^01234x]:' /etc/passwd
```

Результат:

```
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
```

```
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
```

Символ «\*» используется для соответствия «нуль или более» исходящего значения. Выполните следующую команду, чтобы сопоставить двоеточие, за которым следует ноль или более чисел, а затем двоеточие:

```
tail /etc/passwd | grep ':[0-9]*:'
```

Результат:

```
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
libuuid:x:100:101::/var/lib/libuuid:
syslog:x:101:104::/home/syslog:/bin/false
bind:x:102:106::/var/cache/bind:/bin/false
sshd:x:103:65534::/var/run/sshd:/usr/sbin/nologin
```

Вывод в приведенном выше примере показывает, как иногда использование символа \* приводит к «плохим совпадениям». В последних строках вывода есть строки, которые просто совпадают с "::". В большинстве случаев лучше сопоставить «один или несколько», а не «ноль или более».

Символ «+» используется для соответствия «одному или нескольким» искомым значениям. Выполните следующую команду, чтобы найти двоеточие, за которым следуют одно или несколько чисел и двоеточие:

```
tail /etc/passwd | grep -E ':[0-9]+:'
```

```
[root@R-FW ~]# tail /etc/passwd | grep -E ':[0-9]+:'
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
```

Символ "?", который является частью символов расширенного регулярного выражения, используется для того, чтобы сделать предыдущий символ необязательным в шаблоне. Например, чтобы сопоставить два разных способа написания слова «старение» (age или age) в файле словаря, выполните следующую команду:

```
grep -E '^age?ing$' /usr/share/dict/words
```

```
root@L-FW:~# grep -E '^age?ing$' /usr/share/dict/words
ageing
aging
```

Символ "|", который является частью символов расширенного регулярного выражения, используется для сопоставления двух или более элементов. Например, чтобы соответствовать слову «root» или «games» в файле /etc/passwd, выполните следующую команду:

```
grep -E 'root|games' /etc/passwd
```

```
root@L-FW:~# grep -E 'root|games' /etc/passwd
root:x:0:0:root:/root:/bin/bash
games:x:5:60:games:/usr/games:/usr/sbin/nologin
```

Запись фигурных скобок {} используется для сопоставления повторяющихся символов и шаблонов определенное количество раз. Чтобы найти все вхождения буквы «o» ровно дважды в файле /etc/passwd, выполните следующую команду:

```
grep -E "o{2}" /etc/passwd
```

```
root@L-FW:~# grep -E "o{2}" /etc/passwd
root:x:0:0:root:/root:/bin/bash
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
```

Скобки используются для ссылки на то, что ранее было сопоставлено, путем обращения к совпадению, соответствующему первому набору скобок, как «\1», а второму - как «\2». Например, чтобы изменить порядок последних двух полей в /etc/passwd, выполните следующую команду:

```
head /etc/passwd
```

```
head /etc/passwd | sed -r 's/([a-z/]+):([a-z/]+)$/\\2:\\1/'
```

Шаблоны границ слов могут быть указаны при поиске шаблона с помощью последовательности с обратной косой чертой: `'\b'`. Чтобы заменить слово «is» на «was» в файле `/etc/wgetrc`, выполните следующую команду:

```
sed 's/\\broot\\b/toor/' /etc/passwd | head -n -5 | tail -n -1
```

Регулярные выражения используют определенные последовательности обратной косой черты, которые интерпретируются по-разному при вычислении выражения. Например, «`\b`» обозначает границу слова, «`\s`» обозначает символ пробела, а «`\w`» обозначает символ слова.

Специальные символы, такие как «`*`» и «`+`», необходимо экранировать, используя символ «`\`», чтобы избежать их интерпретации в качестве специального символа во время вычисления выражения. Чтобы найти все вхождения шаблона «`*`» в файле `/etc/rsyslog.conf`, выполните следующую команду:

```
grep "\\*" /etc/rsyslog.conf
```

```
root@L-FW:~# grep "\\*" /etc/rsyslog.conf
$IncludeConfig /etc/rsyslog.d/*.conf
auth,authpriv.* /var/log/auth.log
*.*;auth,authpriv.none -/var/log/syslog
#cron.* /var/log/cron.log
```

Альтернативой использованию «`\`» для экранирования каждого специального символа является использование команды **fgrep**, которая всегда обрабатывает свой шаблон как буквенные символы.

Чтобы сопоставить шаблон «`*`» в файле `/etc/rsyslog.conf`, аналогично предыдущему примеру, выполните следующую команду:

```
fgrep "*" /etc/rsyslog.conf
```

```
root@L-FW:~# fgrep "*" /etc/rsyslog.conf
$IncludeConfig /etc/rsyslog.d/*.conf
auth,authpriv.* /var/log/auth.log
*.*;auth,authpriv.none -/var/log/syslog
```

Команда **grep** предоставляет опцию `-i`, чтобы сделать сопоставление с образцом нечувствительным к регистру. Чтобы сопоставить «`abid`» в файле `/usr/share/dict/words` для прописных и строчных букв, выполните следующую команду:

```
grep -i "abid" /usr/share/dict/words
```

Опция `-v` для команды **grep** приведет к отображению всех строк, которые не соответствуют шаблону. Чтобы отобразить все строки, которые не содержат «`local`» в файле `/etc/hosts`, выполните следующие команды:

```
more /etc/hosts
```

```
grep -v "local" /etc/hosts
```

```
root@L-FW:~# grep -v "local" /etc/hosts
127.0.1.1    debian

# The following lines are desirable for IPv6 capable hosts
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

В приведенных выше примерах команда **grep** использовалась для поиска шаблонов в одном файле. Команда **grep** также может использоваться для поиска в нескольких файлах. Чтобы сопоставить «`test`» в файлах «`m*`» в каталоге `/etc`, выполните следующую команду:

```
grep "test" /etc/m*
```

```
root@L-FW:~# grep "test" /etc/m*
/etc/mailcap:application/x-troff-man; /usr/bin/man -X100 -l '%s'; test=test -n "$DISPLAY"
/bin/gxditview; description=Man page
/etc/mailcap:text/troff; /usr/bin/man -X100 -l '%s'; test=test -n "$DISPLAY" -a -e /usr/bi
w; description=Man page
/etc/mailcap:text/plain; view %s; edit=vim %s; compose=vim %s; test=test -x /usr/bin/vim;
nal
/etc/mailcap:text/*; view %s; edit=vim %s; compose=vim %s; test=test -x /usr/bin/vim; need
```

Чтобы просмотреть только имена файлов (вместо каждой совпадающей строки) в примере выше, выполните следующую команду:

```
grep -l "script" /etc/mime*
```

Чтобы сопоставить «DelegateSystem» путем рекурсивного поиска всех файлов в каталоге **/etc** и просмотра совпадающих имен файлов, выполните следующую команду:

**grep -r -l "delegateSystem" /etc/\* 2>/dev/null**

```
root@L-FW:~# grep -r -l "delegateSystem" /etc/* 2>/dev/null
/etc/xml/catalog
/etc/xml/xml-core.xml.old
/etc/xml/catalog.old
/etc/xml/xml-core.xml
```