Практична робота № 12. Використання регулярних виразів у Linux

Теоретичні відомості

Регулярним виразом є шаблон, за допомогою якого описується набір рядків. Регулярні вирази будуються аналогічно арифметичним виразам з використанням різних операторів, які об'єднуються в невеликі вирази.

Основними конструктивними елементами є регулярні вирази, які відповідають одиничному символу. Більшість символів, включаючи всі літери і цифри, є регулярними виразами, які відповідають самі собі. Можна скасувати спеціальне значення будь-якого метасимвола, якщо додати перед ним зворотний слеш.

Метасимволи регулярних виразів

За будь-яким регулярним виразом може слідувати один з поданих операторів повторення (метасимвол):

Таблиця 1.

Оператори регулярних виразів

Оператор	*Дія*
`.`	Відповідає будь-якому одиночному символу
`?`	Попередній елемент необов'язковий і може бути присутнім не більше одного разу
`*`	Попередній елемент може бути присутнім нуль або більше число разів
`+`	Попередній елемент може бути присутнім один або більше число разів
`{N}`	Попередній елемент присутній рівно ***N*** разів
`{N,}`	Попередній елемент може бути присутнім ***N*** або більше число разів
`{N, M}`	Попередній елемент може бути присутнім принаймні ***N*** раз, але не більше ***M*** разів
`-`	За допомогою цього символу задається діапазон, якщо це не перший і не останній елемент у списку і не завершальне значення діапазону
`^`	Відповідає порожньому рядку на початку рядка введення, також представляє символи, що не підпадають в діапазон, зазначений у списку
`$`	Відповідає порожньому рядку в кінці рядка
`\b`	Відповідає порожньому рядку на межі слова

Закінчення таблиці

\B	Відповідає порожньому рядку не на межі слова
\<	Відповідає порожньому рядку на початку слова
\>	Відповідає порожньому рядку в кінці слова

Можна виконувати конкатенацію двох регулярних виразів; результуючий регулярний вираз буде відповідати будь-якому рядку, сформованому конкатенацією двох підрядків, які відповідно представляють з'єднані підвирази.

Два регулярних вирази можуть бути з'єднані бінарним оператором «**|**»; результуючий регулярний вираз буде відповідати будь-якому рядку, який відповідає якому-небудь одному зі з'єднаних підвиразів.

Операція повторення має більший пріоритет, ніж конкатенація, яка, у свою чергу, має більший пріоритет, ніж операція вибору варіанта. Підвираз можна укладати в дужки для того, щоб змінювати ці правила пріоритету.

У базових регулярних виразах символи «**?**» «**+**», «**{**», «**|**», «**(**» і «**)**» не є метасимволами; замість них використовуйте варіанти зі зворотним слешем «**\?**», «**\+**», «**\{**», «**\|**», «**\(**» та «**\)**».

Перевірте в документації до всієї системи, чи є команди, що дозволяють використовувати розширені регулярні вирази.

Команда grep і регулярні вирази

Команда ***grep*** використовується для пошуку у вхідних рядках відповідностей, що визначаються лише за вибраними шаблонами. Коли команда знаходить в рядку відповідність, цей рядок копіюється в стандартне виведення (дія, визначена за замовчуванням), або в будь-який інший вихідний потік, який ви можете задати.

Наведемо декілька прикладів:

```
$ grep root /etc/passwd
root:x:0:0:root:/root:/bin/bash
operator:x:11:0:operator:/root:/sbin/nologin

$ grep -n root /etc/passwd
1:root:x:0:0:root:/root:/bin/bash
12:operator:x:11:0:operator:/root:/sbin/nologin
```

У першій команді відображаються рядки з файлу ***/etc/passwd***, в яких є рядок ***root***.

У другому прикладі відображаються, крім того, номери рядків, в яких є шуканий рядок.

Тепер відобразимо тільки ті рядки, які починаються з рядка «***root***»:

```
$ grep ^root /etc/passwd
root:x:0:0:root:/root:/bin/bash
```

Якщо ми хочемо побачити, в яких облікових записах командна оболонка взагалі не використовувалася, ми шукаємо рядки, що закінчуються символом «***:***»:

```
$ grep :$ /etc/passwd
news:x:9:13:news:/var/spool/news:
```

Щоб перевірити, чи експортується у файлі ***~/.bashrc*** змінна ***PATH***, спочатку потрібно вибрати рядки з «***export***», а потім знайти рядки, що починаються з рядка «***PATH***»; в такому разі не будуть відображатися ***MANPATH*** та інші можливі шляхи:

```
$ grep export ~/.bashrc | grep '\<PATH'
 export
PATH="/bin:/usr/lib/mh:/lib:/usr/bin:/usr/local/bin
:/ usr/ucb:/usr/dbin:$PATH"
```

Символьні класи

Виразом у квадратних дужках є список символів, укладених усередині символів «***[***» і «***]***». Він відповідає будь-якому одиночному символу, вказаному в цьому списку; якщо першим символом списку є «**^**», то він відповідає будь-якому символу, який відстутній у списку. Наприклад, регулярний вираз «***[0123456789]***» відповідає будь-якій одиночній цифрі.

Усередині виразу в квадратних дужках можна вказувати діапазон, що складається з двох символів, розділених дефісом. Тоді вираз відповідає будь-якому одиночному, який згідно з правилами сортування потрапляє всередину цих двох символів, включаючи і ці два символи; при цьому враховується послідовність упорядкування і набір символів, зазначених в локалі. Наприклад, коли за замовчуванням вказана локаль **C**, вираз «***[a-d]***» еквівалентний вислову «***[abcd]***». Є багато локалей, в яких сортування виконується в словниковому порядку, і в цих локалях «***[a-d]***», як правило, не еквівалентне «***[abcd]***», в них, наприклад, воно може бути еквівалентним вислову «***[aBbCcDd]***». Щоб використовувати традиційну інтерпретацію виразів, зазначених вище в квадратних дужках, ви можете скористатися локаллю **C**, встановивши для цього в змінній оточення ***LC_ALL*** значення «***C***».

Нарешті, є певним чином зазначені символьні класи, які вказуються всередині виразів у квадратних дужках. Додаткову інформацію про ці зумовлених виразах можна подивитись на сторінках ***man*** або в документації команди ***grep***.

```
$ grep [yf] /etc/group
sys:x:3:root,bin,adm
tty:x:5:
mail:x:12:mail,postfix
ftp:x:50:
nobody:x:99:
floppy:x:19:
xfs:x:43:
nfsnobody:x:65534:
postfix:x:89:
```

У прикладі відображаються всі рядки, що містять або символ «***y***», або символ «***f***».

Універсальні символи (метасимволи)

Можна використати «**.**» для пошуку відповідності будь-якому одиночному символу. Якщо ви хочете отримати список всіх англійських слів, взятих зі словника, що містять п'ять символів, що починаються з «***c***» і закінчуються «***h***»:

```
$ grep '\<c...h\>' /usr/share/dict/words
catch
clash
cloth
coach
couch
cough
crash
crush
```

Якщо ми хочемо відобразити рядки, в яких є символ точки у вигляді літерала, то потрібно вказати в команді ***grep*** параметр ***-F***.

Щоб подібним чином знайти слова, в яких між «***c***» і «***h***» може бути будь-яке число символів, потрібно використати зірочку. У наведеному нижче прикладі з системного словника вибираються всі слова, що починаються з «***c***» і закінчуються символом «***h***»:

```
$ grep '\<c.*h\>' /usr/share/dict/words
caliph
cash
```

```
catch
cheesecloth
cheetah
```

Якщо ви хочете знайти у файлі або в вихідному потоці літеральний символ «зірочка», використовуйте для цього одинарні лапки. Користувач в наведеному нижче прикладі спочатку намагається в файлі **/etc/profile** знайти «зірочку» без використання лапок, в результаті чого нічого не знаходиться. Коли використовуються лапки, у вихідний потік видається результат:

```
$ grep * /etc/profile

$ grep '*' /etc/profile
for i in /etc/profile.d/*.sh ; do
```

Символьні діапазони

Крім команди **grep** і регулярних виразів, у командній оболонці безпосередньо є ряд способів пошуку відповідності шаблоном без використання зовнішніх програм.

Як ви вже знаєте, зірочка (*****) і знак питання (**?**) відповідає будь-якому рядку або будь-якому одному символу, відповідно. Якщо укласти ці спеціальні символи в лапки, то під час пошуку відповідності буде розглядатися їх літеральне значення:

```
$ touch "*"

$ ls "*"
*
```

Але також можна використовувати квадратні дужки, щоб знайти відповідність будь-якого укладеного в них символу або діапазону символів, якщо пара символів розділяється дефісом. Наприклад:

```
$ ls -ld [a-cx-z]*
drwxr-xr-x 2 user user 4096 Jul 20 2014 app-defaults/
drwxrwxr-x 4 user user 4096 May 25 2014 arabic/
drwxr-xr-x 7 user user 4096 Sep 2 2015 crossover/
drwxrwxr-x 3 user user 4096 Mar 22 2014 xml/
```

У цьому списку відображаються всі файли, які починаються із символів «**a**», «**b**», «**c**», «**x**», «**y**» або «**z**» і розташовані в домашньому директорії користувача **user**.

Якщо першим символом у квадратних дужках буде «**!**» або «**^**», то шукається відповідність будь-якому символу, який не вказаний

всередині дужок. Якщо потрібно знайти відповідність символу дефіса («**-**»), його потрібно вказати як перший або останній символ. Правила сортування залежать від поточної локалі і від значення змінної ***LC_COLLATE***, якщо вона встановлена. Якщо ви хочете бути впевненими, що використовується традиційна інтерпретація діапазонів, явно задайте саме таку інтерпретацію, присвоївши для цього значення «***C***» змінним ***LC_COLLATE*** або ***LC_ALL***.

Символьні класи

Усередині квадратних дужок можна вказувати символьні класи, використавши формат ***[:CLASS:]***, де ***CLASS*** визначається стандартом POSIX і має одне з таких значень:

«***alnum***», «***alpha***», «***ascii***», «***blank***», «***cntrl***», «***digit***», «***graph***», «***lower***», «***print***», «***punct***», «***space***», «***upper***», «***word***» або «***xdigit***».

Декілька прикладів:

```
$ ls -ld [[:digit:]]*
drwxrwxr-x 2 user user 4096 Apr 20  13:45  2/

$ ls -ld [[:upper:]]*
drwxrwxr-- 3 user user 4096 Sep 30 2014 Nautilus/
drwxrwxr-x  4  user  user  4096  Jul  11  2002
OpenOffice.org1.0/
-rw-rw-r--  1  cathy  cathy  997376  Apr  18  15:39
Schedule.sdc
```

Завдання

1. Покажіть список всіх користувачів у вашій системі, які під час входу в систему за замовчуванням користуються командною оболонкою ***bash***.
2. Відобразіть у файлі ***/etc/group*** всі рядки, що починаються з рядка «***daemon***».
3. Відкрийте всі рядки з тих же файлів, в яких немає цього рядка.
4. Покажіть інформацію про ***localhost*** з файлу ***/etc/hosts***, покажіть номери рядків, в яких знайдено відповідність, і порахуйте кількість входжень шуканого рядка.
5. Покажіть список підкаталогів каталогу ***/usr/share/doc***, в яких міститься інформація про командні оболонки.
6. Підрахуйте, скільки файлів ***README*** знаходиться в цих директоріях.

7. За допомогою команди ***grep*** створіть список файлів з вашого домашнього каталогу, які були змінені менше 10 годин тому, не враховуйте при цьому підкаталоги.
8. Помістіть ці команди у сценарій, за допомогою якого буде створюватися зрозумілий вихідний потік.
9. Створіть сценарій, за допомогою якого можна перевіряти, чи існує користувач у файлі ***/etc/passwd***.
10. Покажіть конфігураційні файли, що знаходяться у каталозі ***/etc***, в іменах яких присутні числа.

Контрольні питання

1. Які символи відносяться до метасимволів для їх застосування у регулярних виразах?
2. Що означають вирази у квадратних дужках у регулярних виразах?
3. Що означає символ «**^**» у регулярних виразах?
4. Що означає символ «***$***» у регулярних виразах?
5. Які символьні класи POSIX можна застосувати у регулярних виразах?
6. Який символ у розширених регулярних виразах одначає збіг 0 або 1 раз?
7. Який символ у розширених регулярних виразах одначає збіг 0 або більше разів?
8. Який символ у розширених регулярних виразах одначає збіг з елементом один або більше разів?