# Лабораторная работа 8. Регулярные выражения в Linux

**Аннотация:** Цель работы: изучить синтаксис регулярных выражений в Linux; научиться использовать регулярные выражения для поиска данных.

**Задание 1.** Изучите синтаксис регулярных выражений в Linux.

Ход выполнения:

1. Ознакомьтесь с понятием «регулярное выражение».

*Регулярное выражение* – образец, состоящий из символов. Регулярные выражения используются тогда, когда нужно проанализировать входные данные, имеющие произвольный формат.

Для сокращения размеров регулярных выражений используются специальные символы, некоторые из которых могут быть использованы только в расширенных версиях соответствующих команд (например, при запуске *grep* с ключом **-E**)..

**Символ Описание**

. Соответствует любому символу (за исключением символа новой строки)

(...) Группирует последовательность элементов (атом – шаблон,

заключенный в скобки)

+ Удовлетворяет предыдущему образцу один или

большее количество раз

? Удовлетворяет образцу нуль или один раз

\* Соответствует образцу нуль или любое количество раз

[...] Соответствует символу из заданного множества, допустимы диапазоны типа **а-z**; возможно объединение диапазонов, например **[a-z0-9]**

[^…] Соответствует символу из множества, полученного отрицанием

(...|...|...) Соответствует одной из альтернатив

^ Соответствует началу строки

$ Соответствует образцу в конце строки

{n, m} Соответствует образцу от n до m раз

{n} Соответствует образцу точно n раз

{n,} Соответствует образцу минимум n раз

{,n} Соответствует образцу максимум n раз \n\t etc. Соответствует знаку новой линии, символу табуляции и т. д.

\b Соответствует на границе слова

\В Соответствует внутри границ слова

\d Соответствует цифре

\D Соответствует не цифре

\s Соответствует пробелу

\S Соответствует не пробелу

\w Соответствует букве или цифре

\W Соответствует символу, не являющемуся ни буквой, ни цифрой

2. Ознакомьтесь с описанием команды *grep*.

*grep* – поиск по шаблону, заданному ограниченным регулярным выражением

*grep [-b] [-c] [-i] [-l] [-n] [-s] [-v] ограниченное\_регулярное\_выражение [файл ...]*

Команда *grep* сопоставляет строки исходных файлов с шаблоном, заданным ограниченным\_регулярным\_выражением. Если файлы не указаны, используется стандартный ввод. Обычно каждая успешно сопоставленная строка копируется на стандартный вывод; если исходных файлов несколько, перед найденной строкой выдается имя файла.

Для экранирования символов $, \*, [ ], ^, |, ( ), и \ от интерпретации shell'ом проще всего заключать *ограниченное\_регулярное\_выражение* в одинарные кавычки.

В командной строке могут задаваться следующие опции:

**-b** Перед каждой строкой ставить номер блока, в котором она находится. Используется для поиска блока по контексту (блоки нумеруются с нуля).

**-c** Выдавать только количество успешно сопоставленных строк.

**-i** При сопоставлении не различать большие и малые буквы (регистр).

**-l** Выдавать только имена файлов, в которых есть успешно сопоставленные строки, разделяя имена переводами строк.

**-n** Перед каждой строкой ставить ее номер в файле (строки нумеруются с 1).

**-s** Подавить выдачу диагностических сообщений о несуществующих и недоступных для чтения файлах.

**-v** Выдавать только строки, не удовлетворяющие шаблон

**--color** (обязательно два дефиса) – подсветка найденных совпадений при выводе на экран

**--only-matching** изменяет поведение grep так, что команда выводит не целиком строки, в которых нашлись совпадения с шаблоном регулярного выражения, а только сами совпадения.

Использование команды *egrep* аналогично команде *grep*, кроме того, добавлена возможность использование метасимволов (+ и т.д.).

**Задание 2.** Используйте регулярные выражения для поиска данных.

Ход выполнения:

1. Рекурсивно пройдите весь каталог /etc, найдите все файлы, содержащие строчку *test.* Используйте для этого команду

*grep –r “test” /etc*

2. Найдите строчку *abc* без учета регистра во всех файлах текущего каталога и выведите на экран с подсветкой найденные совпадения.

*grep –i --color “abc” \**

3. Найдите в файле /etc/passwd строки, содержащие имя вашего пользователя.

4. Найдите в файле, содержащем информацию о зарегистрированных в системе группах, строку с названием группы вашего пользователя.

**Задание 3.** Используйте регулярные выражения для поиска данных в текстовом файле по заданным критериям поиска.

Ход выполнения:

1. Пусть вы имеете список телефонов компании, и он выглядит подобно этому:

*Phone Name ID*

*3412 Bob 123*

*3834 Jonny 333*

*1248 Kate 634*

*1423 Tony 567*

*2567 Peter 435*

*3567 Alice 535*

*1548 Kerry 534*

Предположим, это компания более чем из 500 человек. Они хранят данные в простом текстовом файле. Человек с первой цифрой телефонного номера, равной 1, работает в строении под номером 1.

2. Создайте соответствующий файл с именем phonelist.txt.

3. Введите в терминале регулярные выражения, дающие ответ на вопрос: кто работает в строении 1?

*grep '^1' phonelist.txt*

Это значит: искать все строки, которые начинаются с единицы.

"^" соответствует началу строки. Благодаря этому символу выражение соответствует только тем строкам, которые начинаются с единицы.

4. Составьте регулярное выражение для поиска тех строк телефонной книги, в которых человек работает в строении под номером *3* и имя начинается с буквы *A*.

5. Выведите имена системных пользователей, которые начинаются на *s* или *d*.

6. Выберите всех системных пользователей, без оболочки (записи о таких пользователях заканчиваются на *false*).

**Задание 4.** Создайте текстовый файл и выполните задания самостоятельно.

Ход выполнения:

1. Сформируйте файл с любым текстом таким образом, чтобы в файл содержались повторяющиеся слова и пустые строки.

2. Сохраните файл под именем data.

3. Найти в файле data пустые строки с помощью регулярных выражений.

4. В файл data введите адрес электронной почты и составьте регулярное выражение для поиска электронной почты.

5. В файл data введите ip-адрес какого-либо сайта (вида 127.0.0.1) и составьте регулярное выражение для поиска ip-адресов.

6. Добавьте в файл data слово YES с разными вариантами (yes, Yes, YES, YeS и т.д.). Составьте регулярное выражение для поиска слова YES независимо от регистра каждого символа данного слова.

7. В файл data впишите выражение abbaxabaxabbax и напишите регулярное выражение для нахождения слов, содержащих abbax не менее двух раз.

8. В файл data добавьте строку The code is yyXGDH99 – have you got my 1999 sub?. Составьте регулярное выражение, которое позволяет находить выводить код (в данном случае yyXGDH99).