lab2 - Регулярные выражения и управление заданиями

#linux #bash #regex

grep - конспект

```
# -o - ввод совпадений (без - строчек)
# -i - case insensitive
# -E - extended re (расширенные regexp, без экранирования)
# -h - отключить вывод имён файлов
grep -hoiE "[a-z][a-z0-9]+@[a-z]+\.[a-z]{2,10}"

# | - канал (передать вывод-ввод)
grep ... | sort | uniq | tr | ...

# [:upper:] == [A-Z], [:lower:] == [a-z]

# перенаправление вывода-ввода (> / <)
# > filename - создаст файл с именем filename
# >> - без перезаписи содержимого
grep ... > filename
```

Задача

Цель работы – автоматизированное составление упорядоченного списка уникальных адресов электронной почты.

1. Создайте текстовые файлы с литературным текстом, содержащим адреса электронной почты. Включите в них допустимые и не допустимые адреса.

```
Внезапно name1@ya.ru оказался в пустоте.
Но затем na-me2@ya.ru появился из ниоткуда.
В письме было указано: name3@ya1.ya-2.ru
( spb@edu.ru ) - это контакты университета.
Кто-то сказал a@b.cd и все замолчали.
Адрес для ответа: first.last@sub.domain.co.uk
Он оставил email123@test-mail.com однако не ответил.
x@example.info был на экране.
(other.email-with-dash@example.com) - еще один вариант.

А вот недопустимые: привет1not@ya.rumир и словоno.-t2@ya.rueще.
Пишитенаabc@cde.efсразу если что.
А этоnot@.comнеправильно.
Сообщение-not-allowed@example.comне доходит.
В not@allowed..comслучае ошибки.
Для not@-allowed.comтеста.
Not@allowed_.com это не работает.
Abc.example.com просто текст.
Not@allowedдомен не существует.
```

2. Напишите команду `find`, которая используя grep, найдет с помощью регулярных выражений электронные адреса в текстовых файлах.[1]

```
find ~ -name "*.txt" -exec grep -hEio "(^| |\:|\()[a-z]([-._][a-z0-9]|[a-z0-9])*@[a-z0-9]+(\.[a-z0-9]+)*([-._][a-z0-9]|[a-z0-9])*\.[a-z]{2,}($| |\:|\)|\!)" '{}' \; | grep -hEio "[a-z].+[a-z]" | tr [:upper:] [:lower:] | sort
```

3. Возможно, вам понадобится передавать результат по каналу в следующий grep, чтобы убрать из списка допустимый начальный и конечные символы у адресов:

```
(mail1@ya.ru)
mail1@ya.ru
:mail2@ya.ru.
mail2@ya.ru
```

> Второй grep должен просто брать всё от первого символа до последнего, разрешая между любой символ.

```
# 2й grep (простой)
find ~ -name "*.txt" -exec grep -hEio "(^| |\:|\()[a-z]([-._][a-z0-9]|[a-z0-9])*@[a-z0-9]+(\.[a-z0-9]+)*([-._][a-z0-9]|[a-z0-9])*\.[a-z]{2,}($| |\:|\)|\!)" '{}' \; | grep -hEio "[a-z].+[a-z]"

# Perl-compatible regular expressions
find ~ -name "*.txt" -exec grep -Pio "(?<=^| |\:|\()[a-z]([-._][a-z0-9]|[a-z0-9])*@[a-z0-9]+(\.[a-z0-9]+)*([-._][a-z0-9]|[a-z0-9])*\.[a-z]{2,}(?=$| |\:|\)|\!)" '{}' \;
```

4. Переведите все символы в нижний регистр (`tr`) и отсортируйте с уникальностью (`sort`).

```
# sort -u
find ~ -name "*.txt" -exec grep -hEio "(^| |\:|\()[a-z]([-._][a-z0-9]|[a-z0-9])*@[a-z0-9]+(\.[a-z0-9]+)*([-._][a-z0-9]|[a-z0-9])*\.[a-z]{2,}($| |\:|\)|\!)" '{}' \; | grep -hEio "[a-z].+[a-z]" | tr [:upper:] [:lower:] | sort -u

# uniq
... | sort | uniq
```

5. Перенаправьте вывод всей команды в файл `base.txt`.

```
... > base.txt
```

Отложенный запуск команды в файле `user.job`

```
at -f user.job now + 1 minutes
```

Индивидуальные задания

1. Выбрать существующие даты между 1000 и 2012 годом. Секунды могут быть опущены. В каждом месяце 30 дней.[2]

```
grep -E "(20[01][012]|1[0-9]{3})\/(1[012]|0[1-9])\/(0[1-9]|1[0-9]|2[1-9]|30) (([01][0-9]|2[0-3]):
[0-5][0-9])"
```

2. Преобразовать текст, обрамленный в звездочки, в *курсив*. Не трогать текст в двойных звездочках (**жирный**). Использовать команду sed.

```
# -i - in-place editing (заменяет исходный файл)
# -r - расширенные regexp (-E на macOS)
# s - substitude; g - global (все вхождения)
sed -E 's/(^|[^*])\*([a-z ]+)\*[^*]/<em>\2<\/em>/g' file.txt
```

3. Выбрать последовательность неповторяющихся символов в алфавитном порядке. Пробелы нужно игнорировать.

```
# -d - удалить из строки
cat indt3.txt | tr -d ' ' | grep -Eio '(a?b?c?d?e?f?g?h?i?j?k?l?m?n?o?p?q?r?s?t?u?v?w?x?y?z?)'
```

4. Убрать повторяющиеся пробелы и знаки табуляции, оставить по одному пробелу между словами и по два между предложениями. Использовать команду sed.

```
# -e - добавть команду для выполнения
sed -E -e 's/[[:space:]]+/ /g' -e 's/([.!?]) /\1  /g' file.txt
```

---

1. Проверить выражение на сайте regex101 ↩
2. Проверить выражение на сайте regex101 ↩