

## Тема № 2. Регулярные выражения и управление заданиями

Составитель: Фёдоров С. А.

Редакция от: 18.03.2017

### Общее задание «Составление базы электронных адресов»

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

1. Создайте текстовые файлы с *литературным текстом*, содержащим адреса электронной почты. Приведите там допустимые и не допустимые адреса.

Пример допустимых:

- name1@ya.ru
- na-m.e2@ya.ru («-» и «.» несколько раз, но не идут подряд)
- name3@ya1.ya-2.ru (несколько под-доменов)

Пример не допустимых:

- 1not@ya.ru (начинается с цифры)
- po.-t2@ya.ru («-» и «.» идут подряд)
- #abc@cde.ef& (стечение обстоятельств; адреса должны иметь правильный начальный и конечный символы)

Допустимым начальным символом адреса являются: пробел, табуляция, начало строки, «(», «:». Допустимыми конечными символами адреса являются: пробел, табуляция, конец строки, «)».

2. Напишите команду find, которая используя grep, найдёт с помощью регулярных выражений электронные адреса в текстовых файлах.
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.

4. Переведите все символы в нижний регистр (tr) и отсортируйте с *уникальностью* (sort).
5. Перенаправьте вывод всей команды в файл base.txt.
6. Шаблон команды:  
find ~ -name \*\*\* -exec grep -E?? "\*\*\*\*" \*\*\* | grep -E?? "\*\*\*\*" | tr [\*\*\*] [\*\*\*] | sort -? > base.txt
7. В grep понадобятся, по крайней мере, ещё две опции: оставлять только совпадение, а не всю строку с совпадением, и игнорировать регистр букв.
8. Отладьте и поместите эту команду в файл задания user.job.
9. Поставьте на выполнение это задание сразу или через одну минуту (at).

### \* Дополнительное задание

Решите поставленную задачу с использованием одного вызова grep.

## Индивидуальные задания

### Задание 1

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

Примеры:

2012/09/18 12:10 — Да

2012/09/32 12:10 — Нет (32 день)

2013/09/09 09:09 — Нет (после 2012)

### Задание 2

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

Примеры:

*\*this is italic\** => *<em>this is italic</em>*

**\*\*bold text (not italic)\*\*** => **\*\*bold text (not italic)\*\***

### Задание 3

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

Примеры:

abcdefghijkl — Да

abbc — Нет

### Задание 4

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

Примеры:

Extra      spaces => Extra spaces

Sentence.      Sentence. => Sentence. Sentence.