

Урок 6

1. Написать скрипт, который удаляет из текстового файла пустые строки и заменяет маленькие символы на большие. Воспользуйтесь tr или SED.

```
evgeniy:~/geekbrains/homework/lesson5$ man sed | head -n 10 > sed.txt
evgeniy:~/geekbrains/homework/lesson5$ cat sed.txt
SED(1)
    User Commands
        SED(1)

NAME
    sed - stream editor for filtering and transforming text

SYNOPSIS
    sed [OPTION]... {script-only-if-no-other-script} [input-file]...

DESCRIPTION
    Sed is a stream editor. A stream editor is used to perform basic text tran-
    sformations on an input stream (a file or input from a pipeline). While in some ways
    similar to an editor which
evgeniy:~/geekbrains/homework/lesson5$ sed -e '/^$/d; y/abcdefghijklmnopqrstuvwxyz/AB
CDEFGHIJKLMNOPQRSTUVWXYZ/' sed.txt
SED(1)
    USER COMMANDS
        SED(1)

NAME
    SED - STREAM EDITOR FOR FILTERING AND TRANSFORMING TEXT

SYNOPSIS
    SED [OPTION]... {SCRIPT-ONLY-IF-NO-OTHER-SCRIPT} [INPUT-FILE]...

DESCRIPTION
    SED IS A STREAM EDITOR. A STREAM EDITOR IS USED TO PERFORM BASIC TEXT TRAN-
    SFORMATIONS ON AN INPUT STREAM (A FILE OR INPUT FROM A PIPELINE). WHILE IN SOME WAYS
    SIMILAR TO AN EDITOR WHICH
evgeniy:~/geekbrains/homework/lesson5$
```

```
evgeniy:~/geekbrains/homework/lesson5$ cat sed.txt | tr -s "\n" | tr [:lower:] [:upper:]
SED(1)
    USER COMMANDS
        SED(1)

NAME
    SED - STREAM EDITOR FOR FILTERING AND TRANSFORMING TEXT

SYNOPSIS
    SED [OPTION]... {SCRIPT-ONLY-IF-NO-OTHER-SCRIPT} [INPUT-FILE]...

DESCRIPTION
    SED IS A STREAM EDITOR. A STREAM EDITOR IS USED TO PERFORM BASIC TEXT TRANSFOR-
    MATIONS ON AN INPUT STREAM (A FILE OR INPUT FROM A PIPELINE). WHILE IN SOME WAYS SIMILAR
    TO AN EDITOR WHICH
evgeniy:~/geekbrains/homework/lesson5$
```

2. Создать однострочный скрипт, который создаст директории для нескольких годов (2010–2017), в них — поддиректории для месяцев (от 01 до 12), и в каждый из них запишет несколько файлов с произвольными записями. Например, 001.txt, содержащий текст «Файл 001», 002.txt с текстом «Файл 002» и т. д.

```
evgeniy:~/geekbrains/homework/lesson5$ if [ -d $PWD/years ]; then rm -rf $
PWD/years; fi; mkdir $PWD/years; year=2010; while [ $year -ne 2018 ]; do m
kdir $PWD/years/$year; for month in 01 02 03 04 05 06 07 08 09 10 11 12; d
o mkdir $PWD/years/$year/$month; let "countFiles=$RANDOM%5+1"; while [ $co
untFiles -ne 0 ]; do let "countChar=$RANDOM%200+1"; head -c $countChar /de
v/urandom | base64 > $PWD/years/$year/$month/file00$countFiles.txt; let "c
ountFiles -= 1"; done; done; let "year += 1";done;
evgeniy:~/geekbrains/homework/lesson5$ tree years
```

Думаю, не стоит всё сюда вываливать:

```
├── file002.txt
├── 09
│   ├── file001.txt
│   ├── file002.txt
│   ├── file003.txt
│   ├── file004.txt
│   └── file005.txt
├── 10
│   ├── file001.txt
│   ├── file002.txt
│   └── file003.txt
├── 11
│   └── file001.txt
└── 12
    └── file001.txt

104 directories, 282 files
evgeniy:~/geekbrains/homework/lesson5$
```

Чтобы проще было проверять:

```
#!/bin/bash

if [ -d $PWD/years ] #Чтобы в ручную каждый раз папки не удалять
then
    rm -rf $PWD/years;
fi

mkdir $PWD/years;

year=2010;
while [ $year -ne 2018 ];
do
    mkdir $PWD/years/$year;
    for month in 01 02 03 04 05 06 07 08 09 10 11 12;
    do
        mkdir $PWD/years/$year/$month;
        let "countFiles=$RANDOM%5+1"; #Случайное количество файлов от 1 до 5 включительно
        while [ $countFiles -ne 0 ];
        do
            let "countchar=$RANDOM%300+1"; #Случайное количество символов в тексте от 1 до 300 включительно
            head -c $countchar /dev/urandom | base64 > $PWD/years/$year/$month/file00$countFiles.txt; #Случайный текст в каждом файле
            let "countFiles -= 1";
        done;
    done;
    let "year += 1";
done;
```


3. Использовать команду AWK на вывод длинного списка каталога, чтобы отобразить только права доступа к файлам. Затем отправить в конвейере этот вывод на sort и uniq, чтобы отфильтровать все повторяющиеся строки.

```
evgeniy:~$ ls -l /etc | grep -v того | awk '{ print substr($1, 2)}' | sort | uniq -c
  1  r--r-----
  1  r--r--r--
  5  rw-r-----
 82  rw-r--r--
  3  rw-rw-r--
  1  rwxr-s---
  6  rwxrwxrwx
130  rwxr-xr-x
evgeniy:~$
```

4. Используя grep, проанализировать файл /var/log/syslog, отобрав события на своё усмотрение.

```
evgeniy:~$ cat /var/log/syslog | grep Login
Oct 19 09:51:52 test03 systemd[1]: Started Login Service.
Oct 19 12:24:41 test03 systemd[1]: Stopped target Login Prompts.
Oct 19 12:24:59 test03 systemd[1]: Reached target Login Prompts.
Oct 19 12:24:59 test03 systemd[1]: Starting Login Service...
Oct 19 12:24:59 test03 systemd[1]: Started Login Service.
evgeniy:~$
```

5. Создать разовое задание на перезагрузку операционной системы, используя at. Если не от имени root – говорит, отказано в доступе L:

```
evgeniy:~$ echo "reboot" | sudo at 14:50 10/21/2020
warning: commands will be executed using /bin/sh
job 47 at Wed Oct 21 14:50:00 2020
evgeniy:~$ sudo atq
47          Wed Oct 21 14:50:00 2020 a root
evgeniy:~$
```

6. Написать скрипт, делающий архивную копию каталога etc, и прописать задание в crontab.

```
evgeniy:~$ sudo crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
31 14 * * * /home/evgeniy/script.sh
evgeniy:~$
```

```
evgeniy:~$ cat script.sh
#!/bin/bash
if [ -d "/back_etc" ]
then
    rm -rf /back_etc;
fi

mkdir /back_etc;
cp -r /etc /back_etc/etc;
evgeniy:~$
```