

### Домашнее задание:

1. Написать скрипт, который удаляет из текстового файла пустые строки и заменяет маленькие символы на большие. Воспользуйтесь `tr` или `SED`.
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.
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.
4. Используя `grep`, проанализировать файл `/var/log/syslog`, отобразив события на своё усмотрение.
5. Создать разовое задание на перезагрузку операционной системы, используя `at`.
6. \* Написать скрипт, делающий архивную копию каталога `etc`, и прописать задание в `crontab`.

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

Для удобства, скопируем файл «`/etc/ssh/sshd_config`» в текущую папку для тестирования скрипта: (`cp /etc/ssh/sshd_config file1.txt`). Убедимся, что файл имеет пустые строки и текст в разном регистре.

Создадим скрипт при помощи команды:

```
cat > remove_empty_change_ucase.sh << "END"
```

```
#!/bin/bash
```

```
FILE=file1.txt
```

```
#FILE=$1
```

```
#FILE=$(cat)
```

```
if [[ -e "${FILE}" ]]; then
```

```
    echo "обрабатываем файл '${FILE}'"
```

```
    sed '/^$/d' "${FILE}" | sed 's/[a-z]/\U&/g' > file2.txt
```

```
else
```

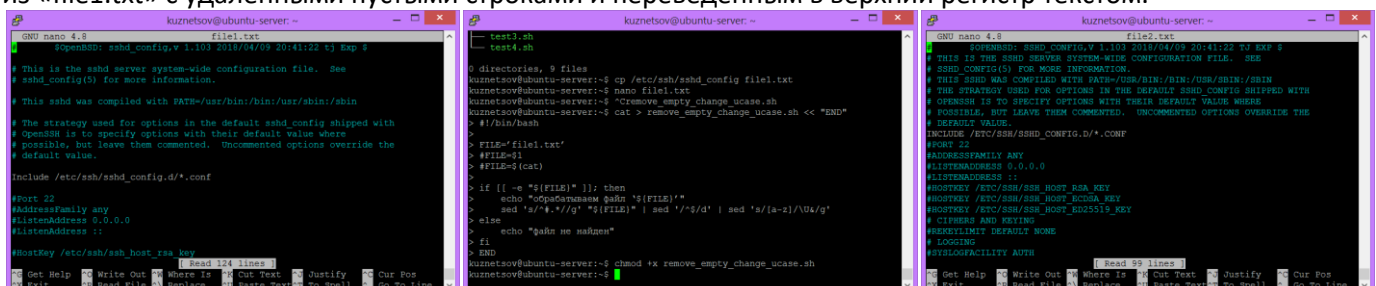
```
    echo "файл не найден"
```

```
fi
```

```
END
```

```
chmod +x remove_empty_change_ucase.sh
```

И запустим его на исполнение (`./remove_empty_change_ucase.sh`). Убедимся, что «`file2.txt`» содержит строки из «`file1.txt`» с удаленными пустыми строками и переведенным в верхний регистр текстом.



```
GNU nano 4.8 file1.txt
# This is the sshd server system-wide configuration file. See
# sshd_config(5) for more information.
#
# This sshd was compiled with PATH=/usr/bin:/bin:/usr/sbin:/sbin
#
# The strategy used for options is the default sshd_config shipped with
# OpenSSH. It is to specify options with their default value where
# possible, but leave them commented. Uncommented options override the
# default value.
#
# Include /etc/ssh/sshd_config.d/*.conf
#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::
#
#HostKey /etc/ssh/ssh_host_rsa_key
#
# Get Help: Write Out Where Is Cut Text Justify Cur Pos
# Exit Read File Replace Paste Text To Spell Go To Line

kuznetsov@ubuntu-server:~$ cat > remove_empty_change_ucase.sh << "END"
#!/bin/bash
FILE=file1.txt
#FILE=$1
#FILE=$(cat)

if [[ -e "${FILE}" ]]; then
    echo "обрабатываем файл '${FILE}'"
    sed '/^$/d' "${FILE}" | sed 's/[a-z]/\U&/g' > file2.txt
else
    echo "файл не найден"
fi
END
chmod +x remove_empty_change_ucase.sh
kuznetsov@ubuntu-server:~$ ./remove_empty_change_ucase.sh
обрабатываем файл 'file1.txt'
kuznetsov@ubuntu-server:~$ cat file2.txt
# This is the sshd server system-wide configuration file. See
# sshd_config(5) for more information.
#
# This sshd was compiled with PATH=/usr/bin:/bin:/usr/sbin:/sbin
#
# The strategy used for options is the default sshd_config shipped with
# OpenSSH. It is to specify options with their default value where
# possible, but leave them commented. Uncommented options override the
# default value.
#
# Include /etc/ssh/sshd_config.d/*.conf
#Port 22
#AddressFamily ANY
#ListenAddress 0.0.0.0
#ListenAddress ::
#
#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key
# Ciphers and keying
#LogLevel INFO
#X11Forwarding no
```

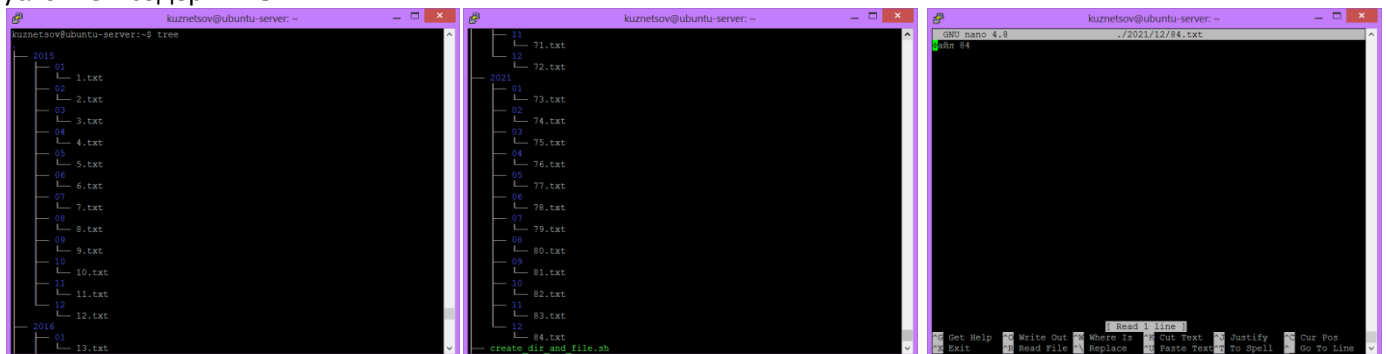
2. Создать однострочный скрипт, который создаст директории для нескольких годов (2010-2017), в них — поддиректории для месяцев (от 01 до 12), и в каждый из них запишет несколько файлов с произвольными записями. Например, 001.txt, содержащий текст «Файл 001», 002.txt с текстом «Файл 002» и т. д.

Создадим скрипт командой:

```
cat > create_dir_and_file.sh << "END"
#!/bin/bash

#mkdir -p {2015..2021}/{01..12}
count_file=0
for year in {2015..2021}
do
    mkdir ${year}
    for month in {01..12}
    do
        let count_file=count_file+1
        mkdir ${year}/${month}
        echo 'Файл '${count_file}' > ${year}/${month}/${count_file}.txt
    done
done
END
chmod +x create_dir_and_file.sh
```

Выполним скрипт и убедимся, что система каталогов создана и в каждом каталоге находится файл с заданным условием содержимым.



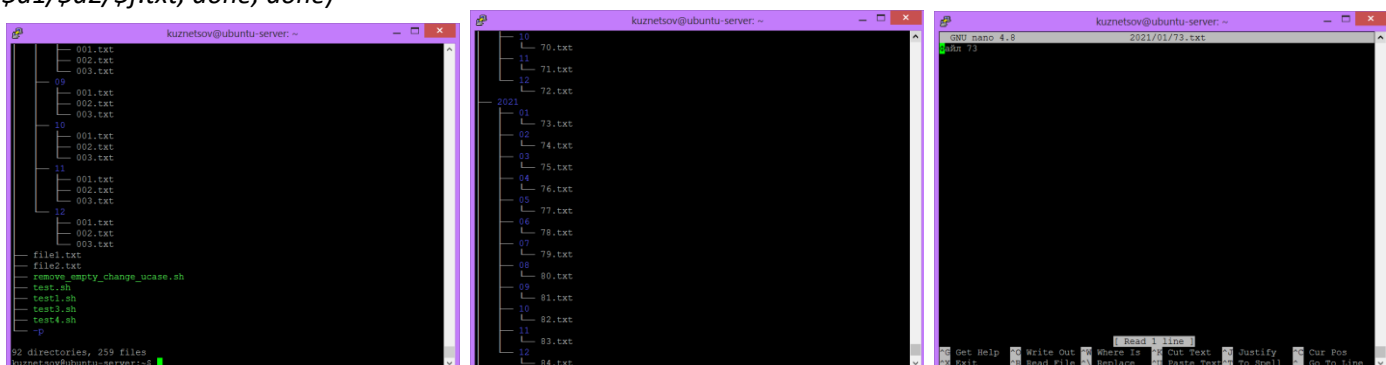
Учитывая то, что в задании имеется фраза «однострочный скрипт», перепишем наш «читабельный» и «понятный» в «однострочный»:

```
#!/bin/bash;count_file=0;for year in {2015..2021};do;mkdir ${year};for month in {01..12};do;let count_file=count_file+1;mkdir ${year}/${month};echo 'Файл '${count_file}' > ${year}/${month}/${count_file}.txt;done;done
```

Правда, в таком случае гораздо проще для создания системы каталогов использовать команду (`mkdir -p {2015..2021}/{01..12}; touch {2015..2021}/{01..12}/{001..003}.txt`)

Можно и с циклами:

```
(f=0; for d1 in {2015..2021}; do mkdir $d1; for d2 in {01..12}; do mkdir $d1/$d2; let f=$f+1; echo 'Файл '$f' > $d1/$d2/$f.txt; done; done)
```

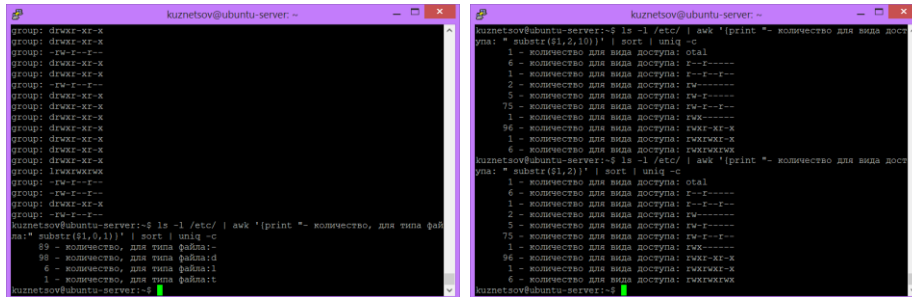


### 3. \* Использовать команду AWK на вывод длинного списка каталога, чтобы отобразить только права доступа к файлам. Затем отправить в конвейере этот вывод на sort и uniq, чтобы отфильтровать все повторяющиеся строки.

Чтобы посмотреть, сколько и каких видов файлов в выбранном каталоге «/etc/», достаточно вести команду (ls -l /etc/ | awk '{print "- количество, для типа файла:" substr(\$1,0,1)}' | sort | uniq -c)

Для вывода прав доступа строки доступа к файлам (каталоги мы также считаем «файлами»), используем команду (ls -l /etc/ | awk '{print "- количество для вида доступа: " substr(\$1,2)}' | sort | uniq -c)

Примечание: мы считаем, права доступа не включают первый символ указывающий на тип объекта.



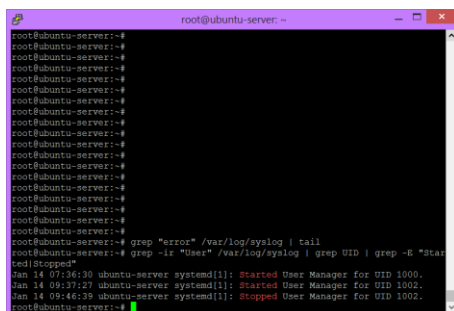
```
kuznetsov@ubuntu-server:~$ ls -l /etc/ | awk '{print "- количество, для типа файла:" substr($1,0,1)}' | sort | uniq -c
 93 - количество, для типа файла:
 98 - количество, для типа файла:d
 6 - количество, для типа файла:l
 1 - количество, для типа файла:t

kuznetsov@ubuntu-server:~$ ls -l /etc/ | awk '{print "- количество для вида доступа: " substr($1,2)}' | sort | uniq -c
 6 - количество для вида доступа: r--r--r--
 1 - количество для вида доступа: r--r--r--
 2 - количество для вида доступа: rw-r--r--
 5 - количество для вида доступа: rw-r--r--
 75 - количество для вида доступа: rw-r--r--
 1 - количество для вида доступа: rwxr-xr-x
 96 - количество для вида доступа: rwxr-xr-x
 1 - количество для вида доступа: rwxr-xr-x
 4 - количество для вида доступа: rwxr-xr-x
```

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

Проверим на наличие ошибок: (grep "error" /var/log/syslog | tail) – их нет.

Посмотрим, какие пользователи регистрировались в системе и выходили из нее: (grep -ir "User" /var/log/syslog | grep UID | grep -E "Started|Stopped")



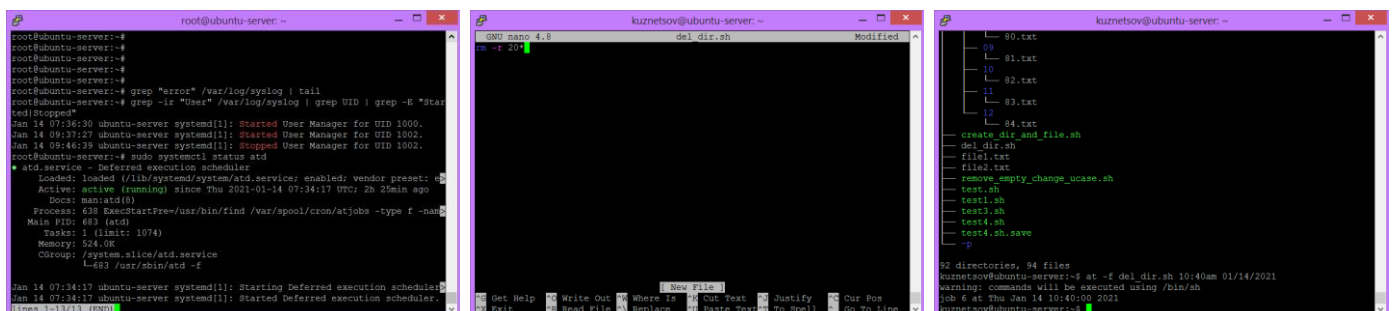
```
root@ubuntu-server:~$ grep "error" /var/log/syslog | tail
root@ubuntu-server:~$ grep -ir "User" /var/log/syslog | grep UID | grep -E "Started|Stopped"
Jan 14 07:36:30 ubuntu-server systemd[1]: Started User Manager for UID 1000.
Jan 14 09:37:27 ubuntu-server systemd[1]: Started User Manager for UID 1002.
Jan 14 09:46:39 ubuntu-server systemd[1]: Stopped User Manager for UID 1002.
root@ubuntu-server:~$
```

### 5. Создать разовое задание на перезагрузку операционной системы, используя at.

Убедимся, что служба планировщика atd запущена (sudo systemctl status atd) и не требует установки (sudo apt install at -y).

Создадим файл «del\_dir.sh» с командой на удаление системы каталогов созданных во 2-й главе (rm -r 20\*).

Запустим задачу (at -f del\_dir.sh 10:40am 01/14/2021) – ответ: «commands will be executed using /bin/sh job 5 at Thu Jan 14 10:40:00 2021»



```
root@ubuntu-server:~$ sudo systemctl status atd
* atd.service - Deferred execution scheduler
   Loaded: loaded (/lib/systemd/system/atd.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2021-01-14 07:34:17 UTC; 2h 25min ago
     Docs: man:atd(8)
  Process: 683 ExecStartPre=/usr/bin/find /var/spool/cron/atjobs -type f -not -name
    Main PID: 683 (atd)
       Tasks: 1 (limit: 1074)
      Memory: 524.0K
    CGroup: /system.slice/atd.service
            └─683 /usr/sbin/atd -f

Jan 14 07:34:17 ubuntu-server systemd[1]: Starting Deferred execution scheduler:
Jan 14 07:34:17 ubuntu-server systemd[1]: Started Deferred execution scheduler.
root@ubuntu-server:~$

kuznetsov@ubuntu-server:~$ cat del_dir.sh
rm -r 20*

kuznetsov@ubuntu-server:~$ at -f del_dir.sh 10:40am 01/14/2021
warning: commands will be executed using /bin/sh
job 6 at Thu Jan 14 10:40:00 2021
```

В нами запланированное время, система запустила на исполнение задачу. Проверка (tree) подтверждает удаление каталогов.

По аналогии с предыдущим вариантом, создадим файл «sudo shut\_down.sh» с командой «shutdown -r now». Запустим задачу (at -f shut\_down.sh 11:00am 01/14/2021). Цель достигнута.

Также, при помощи pipe команда отрабатывает в назначенное время (echo 'sudo /usr/sbin/shutdown -r now' | at 11:32am 01/14/2021).

The first terminal window shows the creation of a directory and several files: file2.txt, del\_dir.sh, file1.txt, file2.txt, remove\_empty\_change\_ucase.sh, test1.sh, test2.sh, test3.sh, test4.sh, and test4.sh.save. The second window shows the login process for kuznetsov on Ubuntu 20.04.1 LTS, followed by system updates and a message about self-healing high availability clusters. The third window shows the execution of an at job at 11:00am on 01/14/2021, which runs the command 'sudo /usr/sbin/shutdown -r now'.

По началу, несколько удивляла «неработоспособность» настройки отложенной перезагрузки. Ни через файл, ни через (echo 'shutdown -r now' | at 11:00am 01/14/2021). Пришлось установить (apt install postfix mailutils -y) и проанализировать почту (mail)...понять, что система ругается «Permission denied»... Не хватало прав – добавил «sudo».

The terminal window displays a mail log entry for a message sent to kuznetsov@ubuntu-server. The log shows the message was received by postfix and then failed to be delivered to the user's mailbox due to a 'Permission denied' error.

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

Создадим файл скрипта «create\_arh.sh» с командой архивирования «tar -czvf "\$(date +"%Y-%m-%d")".tar.gz /etc/\*»:

```
cat > create_arh.sh << "END"
#!/bin/bash
tar -czvf "$(date +"%Y-%m-%d")".tar.gz /etc/*
END
chmod +x create_arh.sh
```

Добавляем строчку к расписанию «cron» (crontab -e) «15 12 \* \* \* /home/kuznetsov/create\_arh.sh» (каждый день в 12:15 запускает скрипт, производящий архивирование)

Все отлично, архив «2021-01-14.tar.gz» создан автоматически.

The first terminal window shows the creation of the create\_arh.sh script and its execution. The second window shows the crontab being edited to add the task '15 12 \* \* \* /home/kuznetsov/create\_arh.sh'. The third window shows the contents of the /etc directory, which has been archived into the file 2021-01-14.tar.gz.