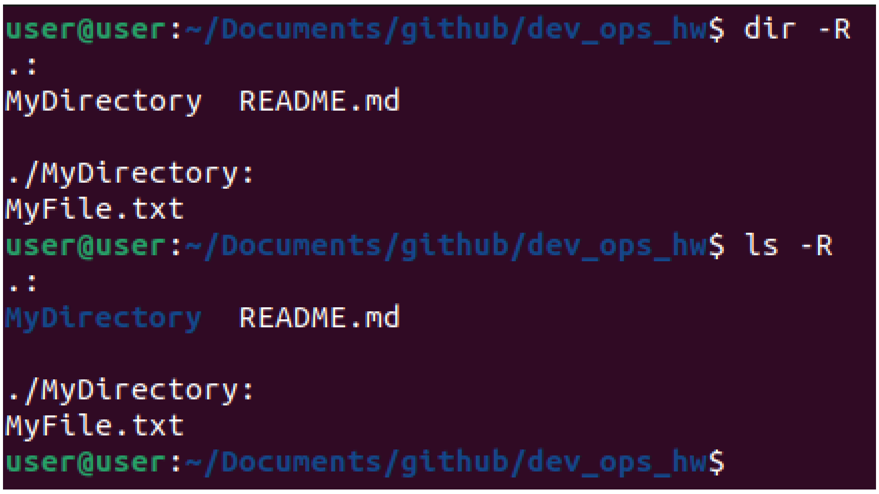
Домашнее задание

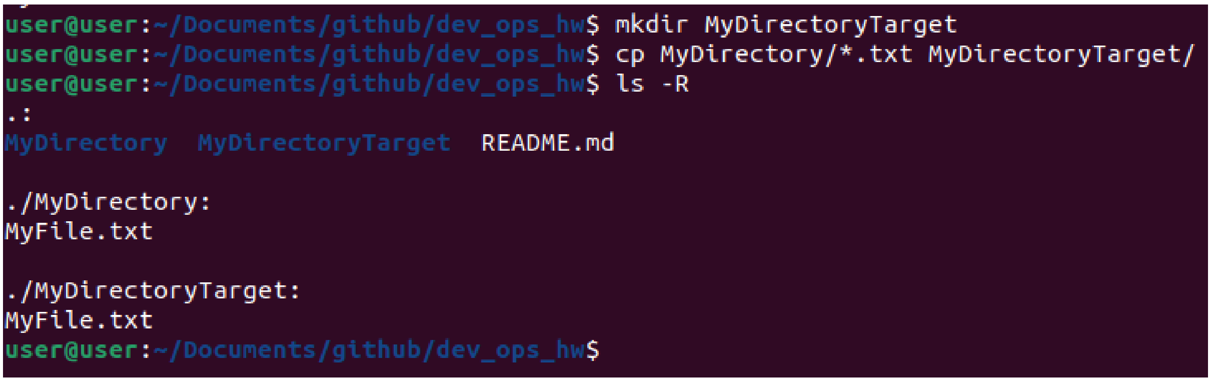
Выполнил: Ilyassov Olzhas (Ильясов Олжас)
Группа: Интенсив DevOps for Tech Orda 2024

1. Создание директории и файла:



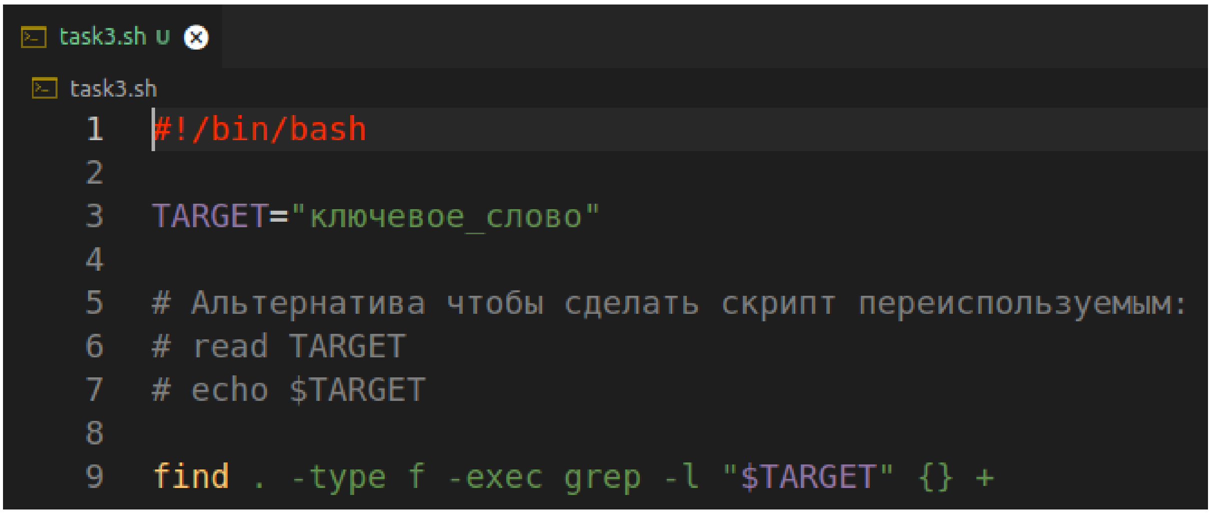
```
user@user:~/Documents/github/dev_ops_hw$ dir -R
.:
MyDirectory  README.md

./MyDirectory:
MyFile.txt
user@user:~/Documents/github/dev_ops_hw$ ls -R
.:
MyDirectory  README.md

./MyDirectory:
MyFile.txt
user@user:~/Documents/github/dev_ops_hw$
```

2. Копирование файлов:

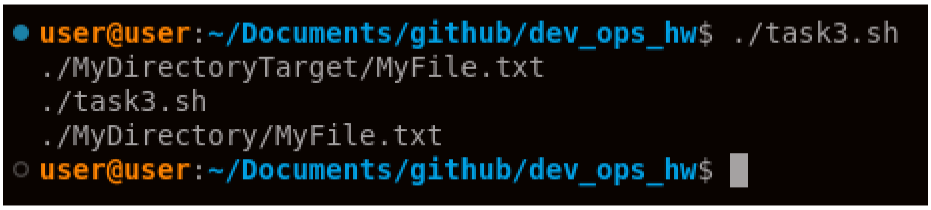


```
user@user:~/Documents/github/dev_ops_hw$ mkdir MyDirectoryTarget
user@user:~/Documents/github/dev_ops_hw$ cp MyDirectory/*.txt MyDirectoryTarget/
user@user:~/Documents/github/dev_ops_hw$ ls -R
.:
MyDirectory  MyDirectoryTarget  README.md

./MyDirectory:
MyFile.txt

./MyDirectoryTarget:
MyFile.txt
user@user:~/Documents/github/dev_ops_hw$
```

3. Поиск слова:

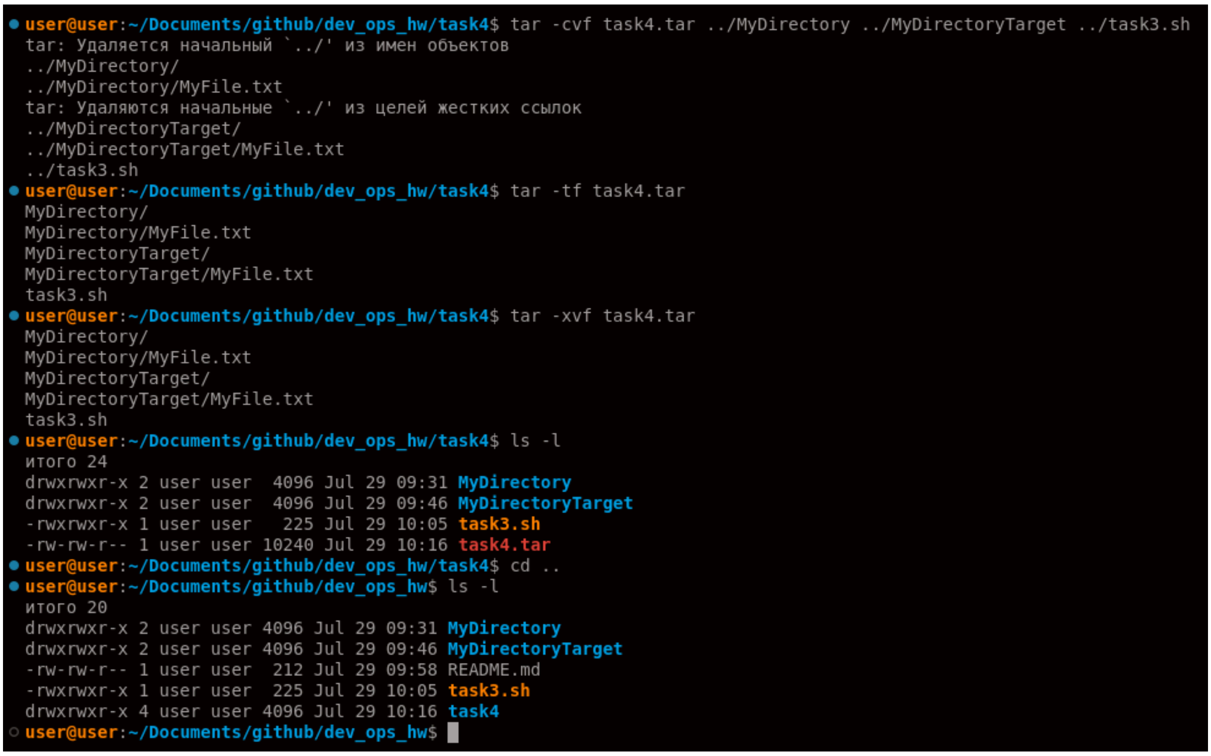


```
#!/bin/bash

TARGET="ключевое_слово"

# Альтернатива чтобы сделать скрипт переиспользуемым:
# read TARGET
# echo $TARGET

find . -type f -exec grep -l "$TARGET" {} +
```



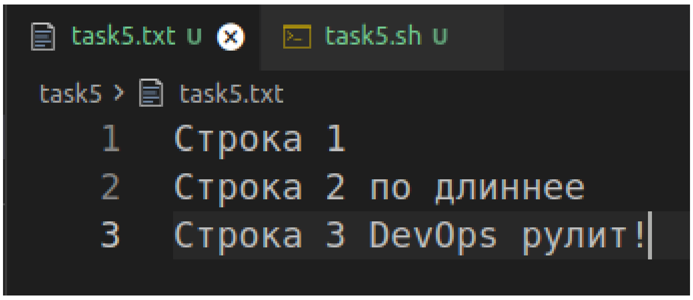
```
user@user:~/Documents/github/dev_ops_hw$ ./task3.sh
./MyDirectoryTarget/MyFile.txt
./task3.sh
./MyDirectory/MyFile.txt
user@user:~/Documents/github/dev_ops_hw$
```

4. Архивирование и распаковка:

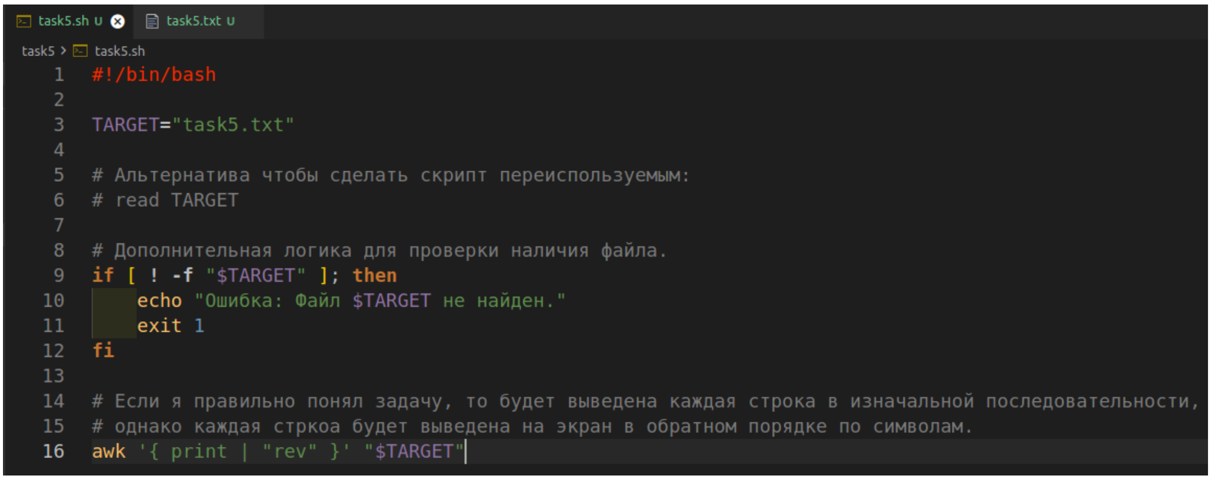


```
user@user:~/Documents/github/dev_ops_hw/task4$ tar -cvf task4.tar ../MyDirectory ../MyDirectoryTarget ../task3.sh
tar: Удаляется начальный `../' из имен объектов
../MyDirectory/
../MyDirectory/MyFile.txt
tar: Удаляются начальные `../' из целей жестких ссылок
../MyDirectoryTarget/
../MyDirectoryTarget/MyFile.txt
../task3.sh
user@user:~/Documents/github/dev_ops_hw/task4$ tar -tf task4.tar
MyDirectory/
MyDirectory/MyFile.txt
MyDirectoryTarget/
MyDirectoryTarget/MyFile.txt
task3.sh
user@user:~/Documents/github/dev_ops_hw/task4$ tar -xvf task4.tar
MyDirectory/
MyDirectory/MyFile.txt
MyDirectoryTarget/
MyDirectoryTarget/MyFile.txt
task3.sh
user@user:~/Documents/github/dev_ops_hw/task4$ ls -l
итого 24
drwxrwxr-x 2 user user  4096 Jul 29 09:31 MyDirectory
drwxrwxr-x 2 user user  4096 Jul 29 09:46 MyDirectoryTarget
-rwxrwxr-x 1 user user   225 Jul 29 10:05 task3.sh
-rw-rw-r-- 1 user user 10240 Jul 29 10:16 task4.tar
user@user:~/Documents/github/dev_ops_hw/task4$ cd ..
user@user:~/Documents/github/dev_ops_hw$ ls -l
итого 20
drwxrwxr-x 2 user user 4096 Jul 29 09:31 MyDirectory
drwxrwxr-x 2 user user 4096 Jul 29 09:46 MyDirectoryTarget
-rw-rw-r-- 1 user user  212 Jul 29 09:58 README.md
-rwxrwxr-x 1 user user  225 Jul 29 10:05 task3.sh
drwxrwxr-x 4 user user 4096 Jul 29 10:16 task4
user@user:~/Documents/github/dev_ops_hw$
```

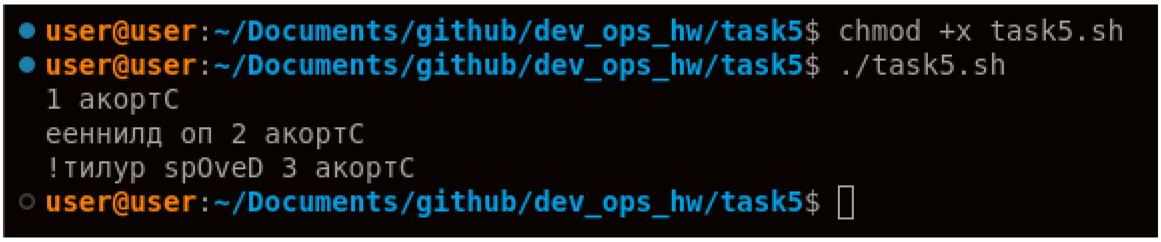
5. Обработка текстового файла:



```
Строка 1
Строка 2 по длиннее
Строка 3 DevOps рулит!
```

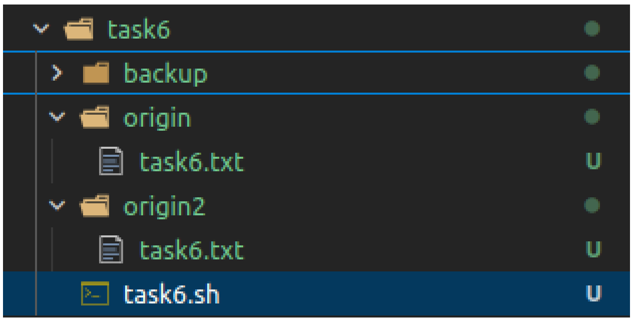


```bash
#!/bin/bash

TARGET="task5.txt"

# Альтернатива чтобы сделать скрипт переиспользуемым:
# read TARGET

# Дополнительная логика для проверки наличия файла.
if [ ! -f "$TARGET" ]; then
    echo "Ошибка: Файл $TARGET не найден."
    exit 1
fi

# Если я правильно понял задачу, то будет выведена каждая строка в изначальной последовательности,
# однако каждая стркоа будет выведена на экран в обратном порядке по символам.
awk '{ print | "rev" }' "$TARGET"
```

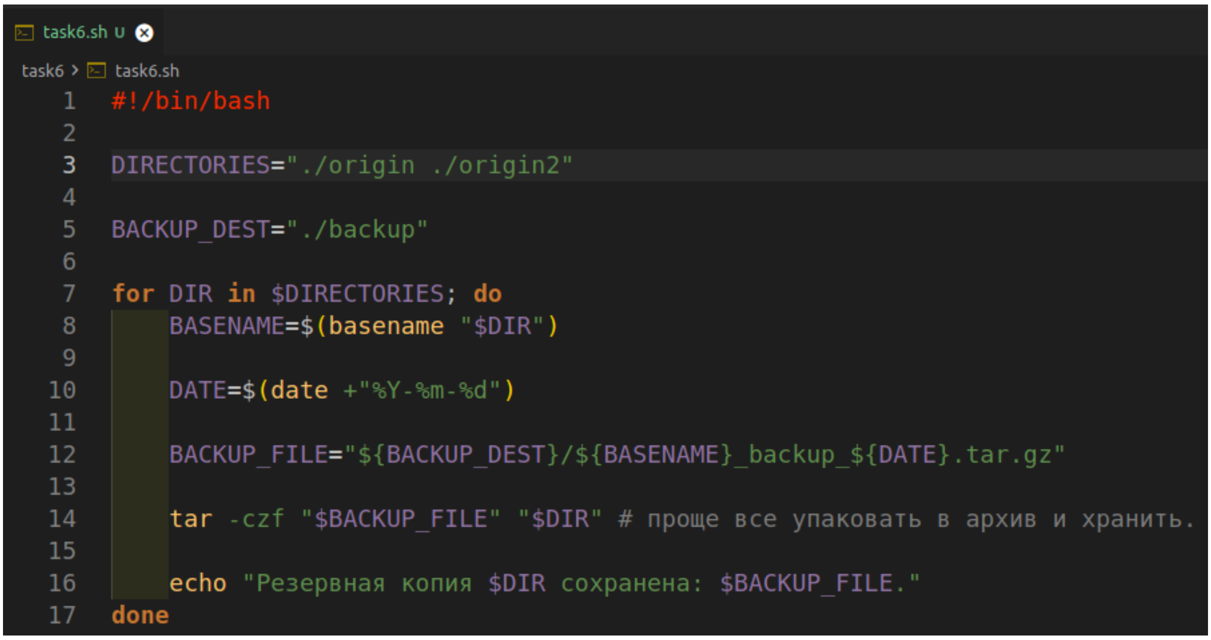


```
● user@user:~/Documents/github/dev_ops_hw/task5$ chmod +x task5.sh
● user@user:~/Documents/github/dev_ops_hw/task5$ ./task5.sh
  1 акортС
  еннилд оп 2 акортС
  !тилур sp0veD 3 акортС
○ user@user:~/Documents/github/dev_ops_hw/task5$ 
```

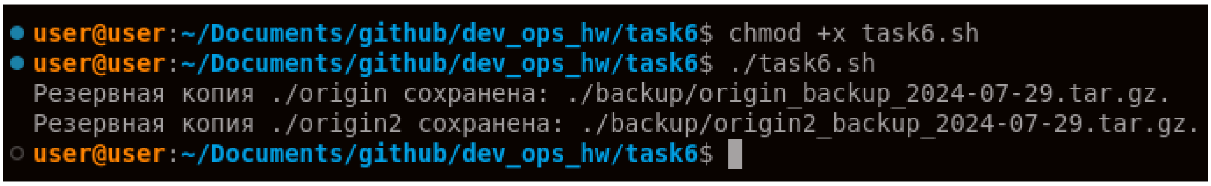
6. Автоматизация резервного копирования:



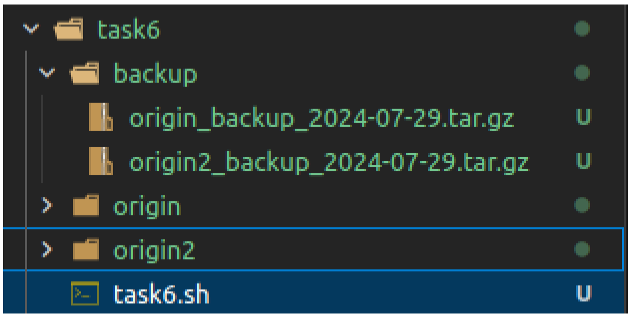
```
task6
  backup
  origin
    task6.txt      U
  origin2
    task6.txt      U
  task6.sh         U
```



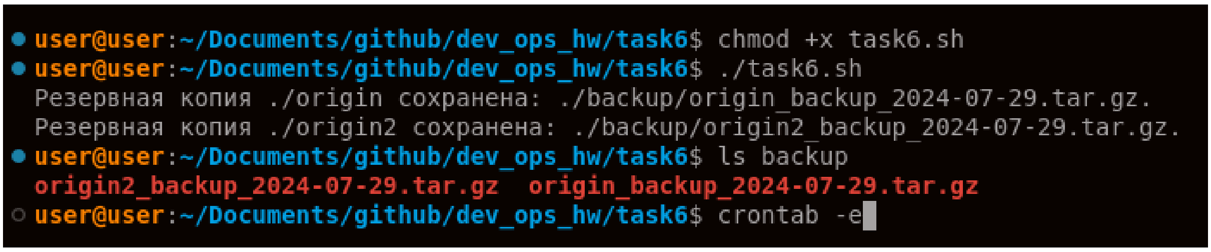
```bash
#!/bin/bash

DIRECTORIES="./origin ./origin2"

BACKUP_DEST="./backup"

for DIR in $DIRECTORIES; do
    BASENAME=$(basename "$DIR")

    DATE=$(date +"%Y-%m-%d")

    BACKUP_FILE="${BACKUP_DEST}/${BASENAME}_backup_${DATE}.tar.gz"

    tar -czf "$BACKUP_FILE" "$DIR" # проще все упаковать в архив и хранить.

    echo "Резервная копия $DIR сохранена: $BACKUP_FILE."
done
```



```
user@user:~/Documents/github/dev_ops_hw/task6$ chmod +x task6.sh
user@user:~/Documents/github/dev_ops_hw/task6$ ./task6.sh
Резервная копия ./origin сохранена: ./backup/origin_backup_2024-07-29.tar.gz.
Резервная копия ./origin2 сохранена: ./backup/origin2_backup_2024-07-29.tar.gz.
user@user:~/Documents/github/dev_ops_hw/task6$ 
```

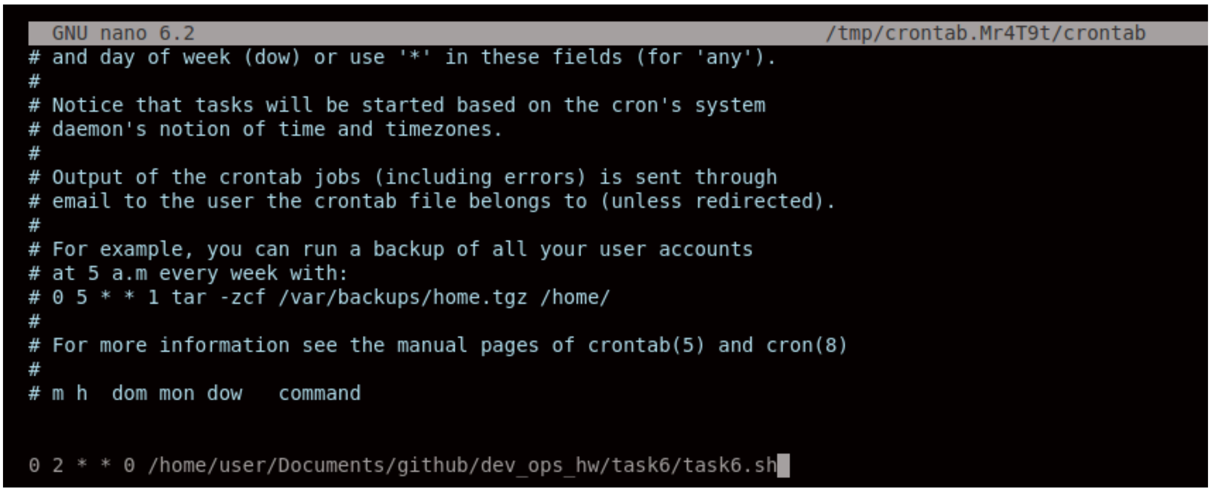


```
task6
  backup
    origin_backup_2024-07-29.tar.gz   U
    origin2_backup_2024-07-29.tar.gz  U
  origin
  origin2
  task6.sh                            U
```



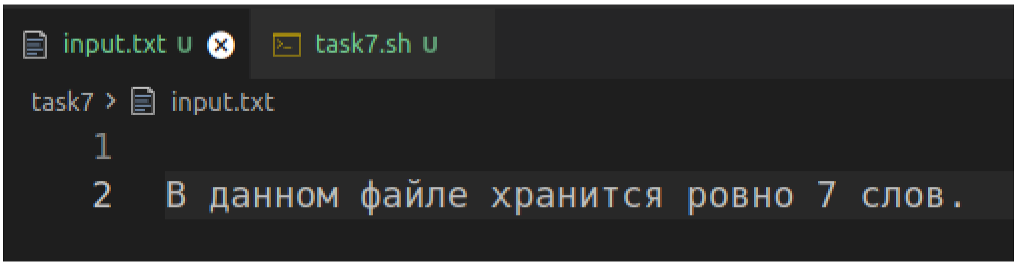
```
user@user:~/Documents/github/dev_ops_hw/task6$ chmod +x task6.sh
user@user:~/Documents/github/dev_ops_hw/task6$ ./task6.sh
Резервная копия ./origin сохранена: ./backup/origin_backup_2024-07-29.tar.gz.
Резервная копия ./origin2 сохранена: ./backup/origin2_backup_2024-07-29.tar.gz.
user@user:~/Documents/github/dev_ops_hw/task6$ ls backup
origin2_backup_2024-07-29.tar.gz  origin_backup_2024-07-29.tar.gz
user@user:~/Documents/github/dev_ops_hw/task6$ crontab -e
```

А для того, чтобы автоматизировать скрипт и вызывать срабатывание каждую неделю, использовал инструмент cron.

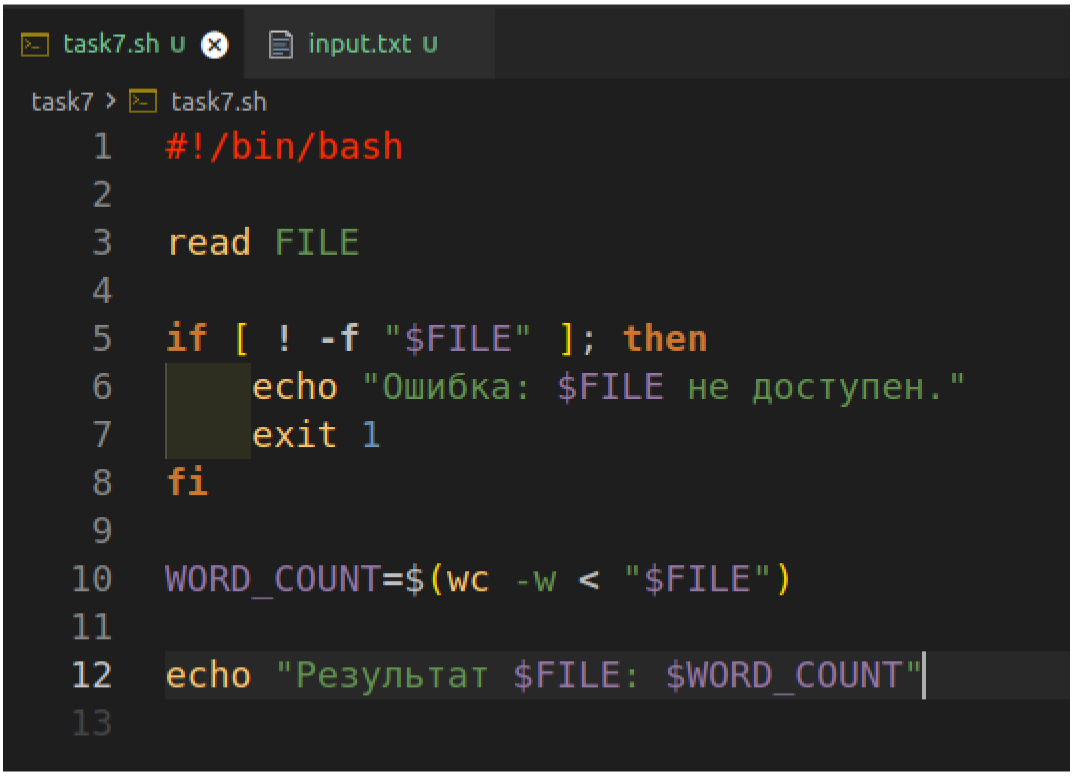


```
  GNU nano 6.2                                          /tmp/crontab.Mr4T9t/crontab
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) 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


0 2 * * 0 /home/user/Documents/github/dev_ops_hw/task6/task6.sh
```

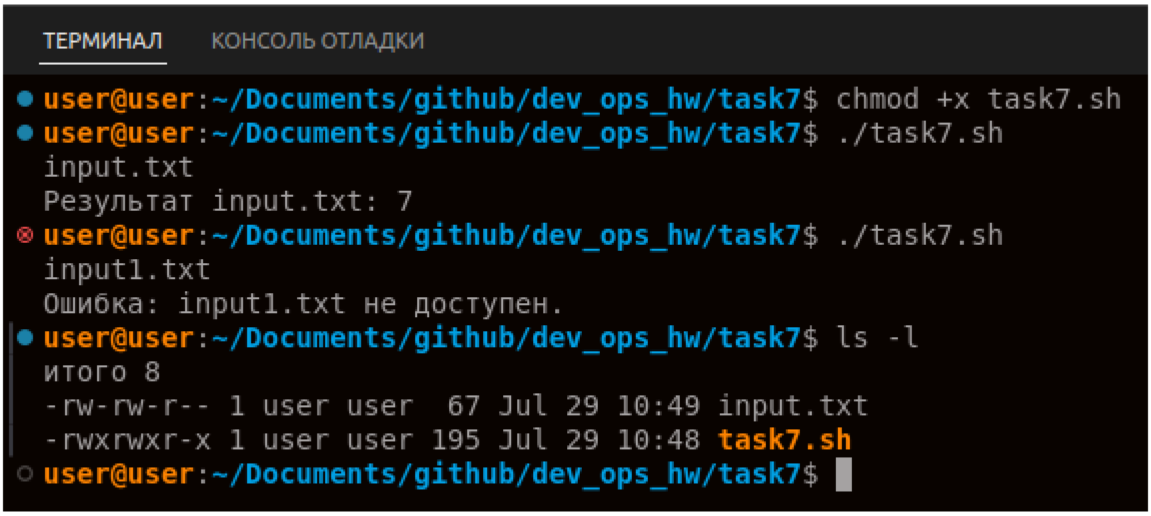
7. Подсчет количества слов:



```

В данном файле хранится ровно 7 слов.
```

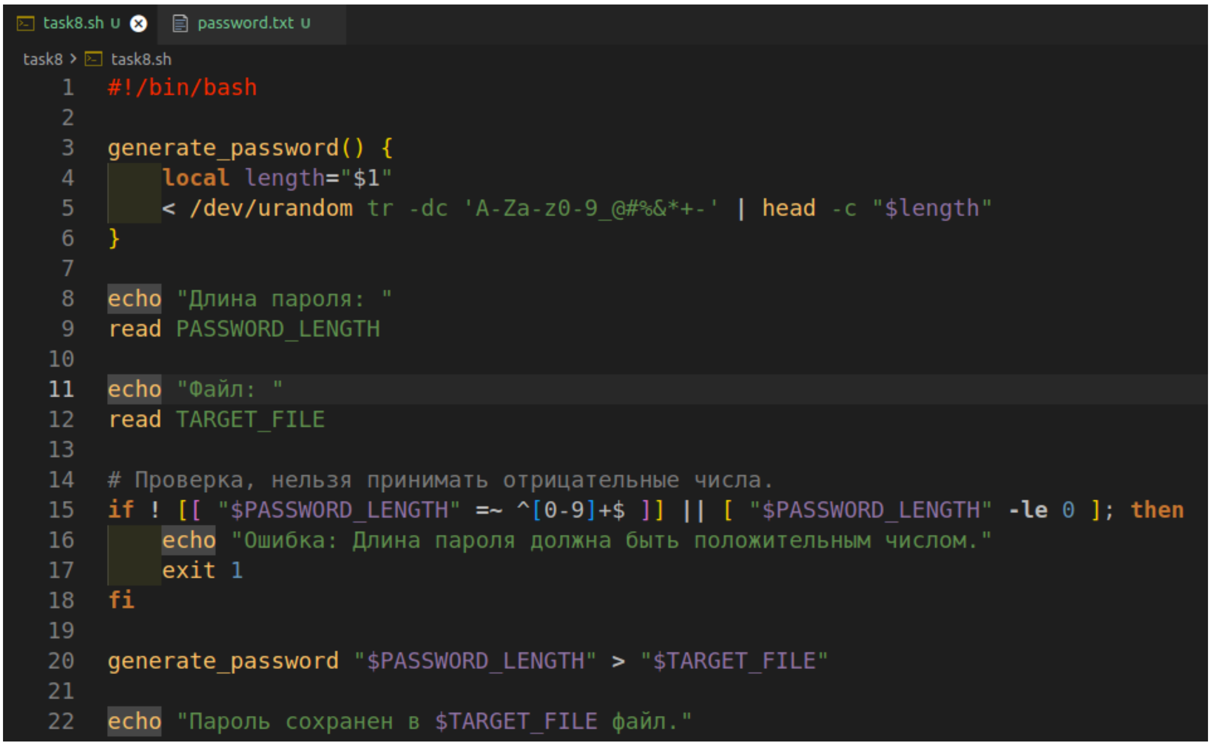


```bash
#!/bin/bash

read FILE

if [ ! -f "$FILE" ]; then
    echo "Ошибка: $FILE не доступен."
    exit 1
fi

WORD_COUNT=$(wc -w < "$FILE")

echo "Результат $FILE: $WORD_COUNT"
```

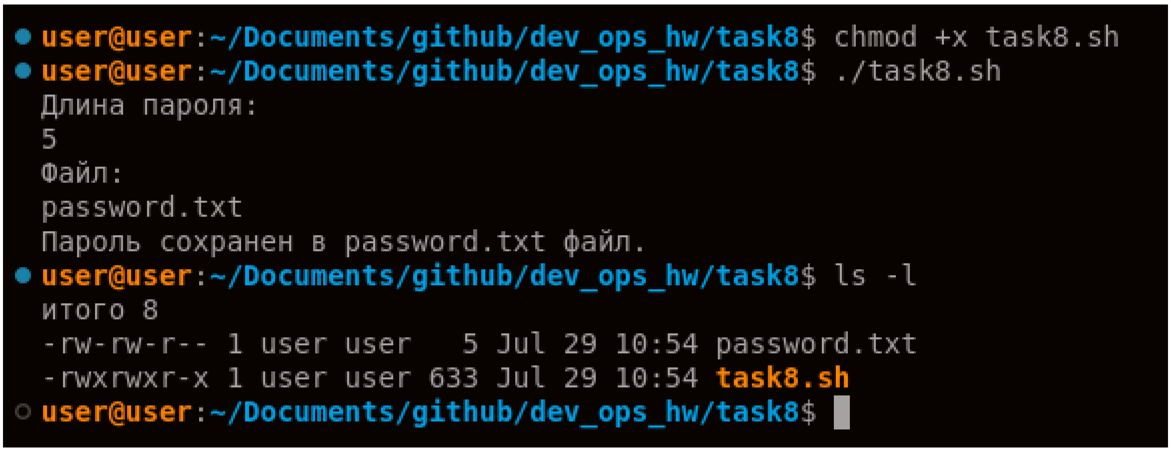


```
user@user:~/Documents/github/dev_ops_hw/task7$ chmod +x task7.sh
user@user:~/Documents/github/dev_ops_hw/task7$ ./task7.sh
input.txt
Результат input.txt: 7
user@user:~/Documents/github/dev_ops_hw/task7$ ./task7.sh
input1.txt
Ошибка: input1.txt не доступен.
user@user:~/Documents/github/dev_ops_hw/task7$ ls -l
итого 8
-rw-rw-r-- 1 user user  67 Jul 29 10:49 input.txt
-rwxrwxr-x 1 user user 195 Jul 29 10:48 task7.sh
user@user:~/Documents/github/dev_ops_hw/task7$ 
```

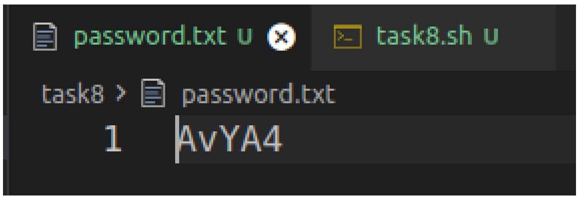
8. Создание случайных паролей:



```bash
#!/bin/bash

generate_password() {
    local length="$1"
    < /dev/urandom tr -dc 'A-Za-z0-9_@#%&*+-' | head -c "$length"
}

echo "Длина пароля: "
read PASSWORD_LENGTH

echo "Файл: "
read TARGET_FILE

# Проверка, нельзя принимать отрицательные числа.
if ! [[ "$PASSWORD_LENGTH" =~ ^[0-9]+$ ]] || [ "$PASSWORD_LENGTH" -le 0 ]; then
    echo "Ошибка: Длина пароля должна быть положительным числом."
    exit 1
fi

generate_password "$PASSWORD_LENGTH" > "$TARGET_FILE"

echo "Пароль сохранен в $TARGET_FILE файл."
```

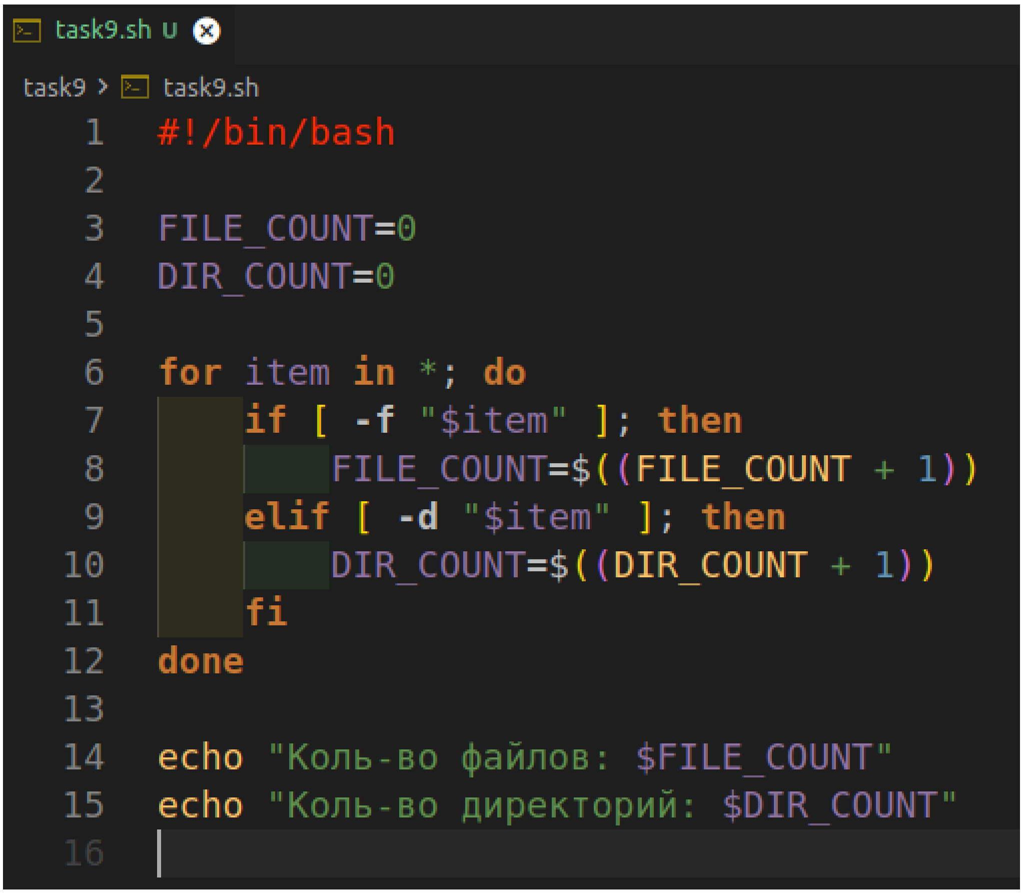


```
user@user:~/Documents/github/dev_ops_hw/task8$ chmod +x task8.sh
user@user:~/Documents/github/dev_ops_hw/task8$ ./task8.sh
Длина пароля:
5
Файл:
password.txt
Пароль сохранен в password.txt файл.
user@user:~/Documents/github/dev_ops_hw/task8$ ls -l
итого 8
-rw-rw-r-- 1 user user   5 Jul 29 10:54 password.txt
-rwxrwxr-x 1 user user 633 Jul 29 10:54 task8.sh
user@user:~/Documents/github/dev_ops_hw/task8$ 
```

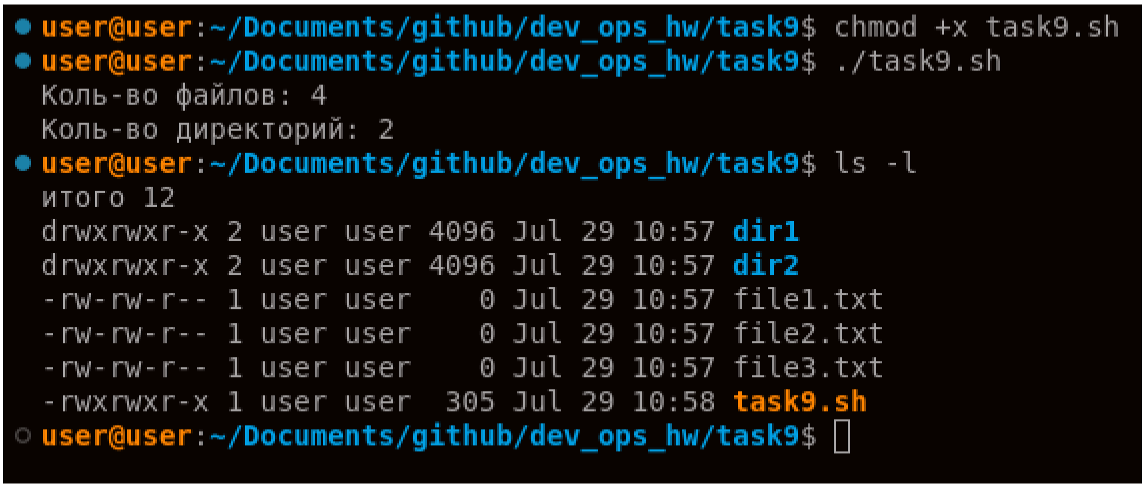


```
AvYA4
```

9. Подсчет файлов:

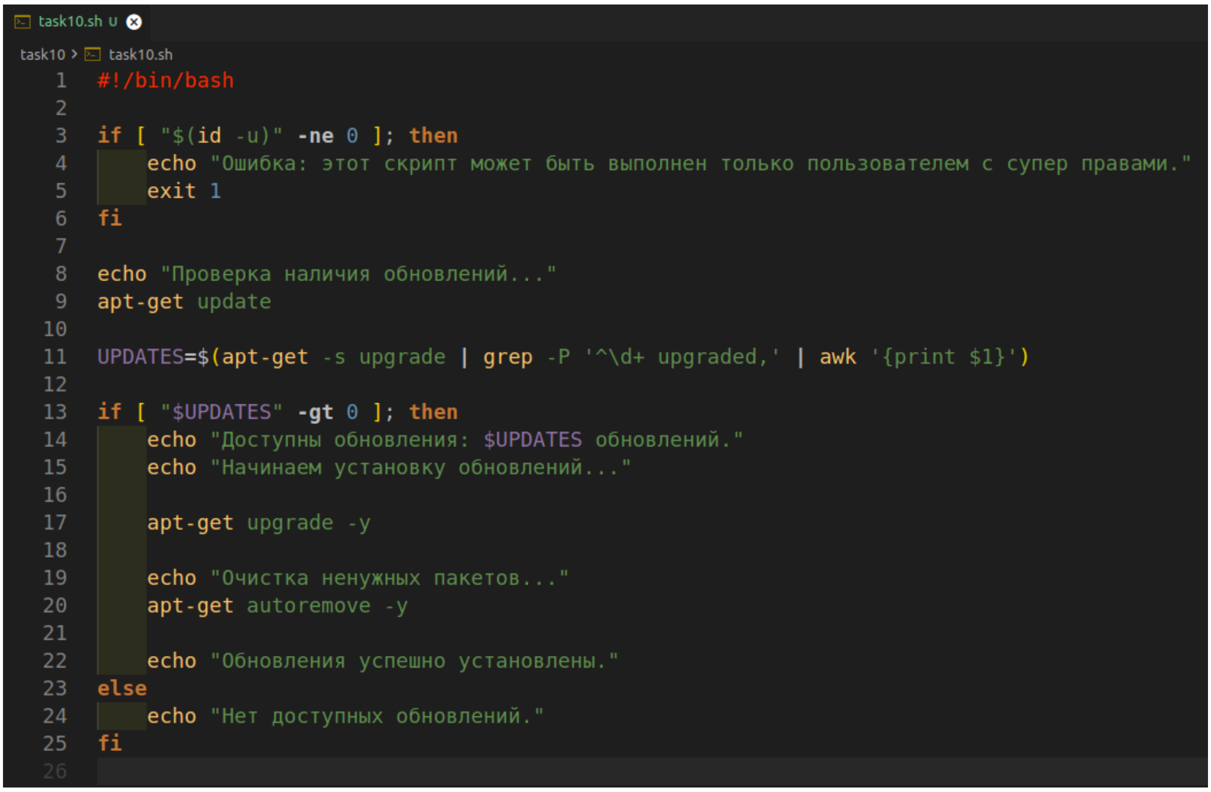


```bash
#!/bin/bash

FILE_COUNT=0
DIR_COUNT=0

for item in *; do
    if [ -f "$item" ]; then
        FILE_COUNT=$((FILE_COUNT + 1))
    elif [ -d "$item" ]; then
        DIR_COUNT=$((DIR_COUNT + 1))
    fi
done

echo "Коль-во файлов: $FILE_COUNT"
echo "Коль-во директорий: $DIR_COUNT"

```

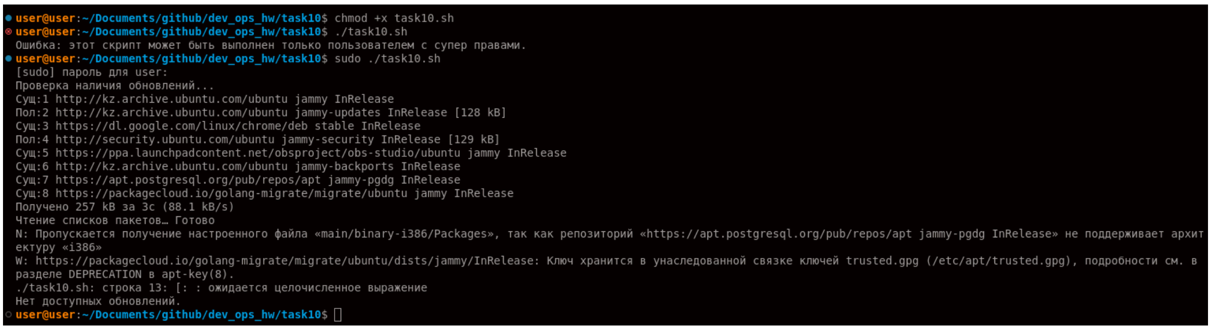


```
user@user:~/Documents/github/dev_ops_hw/task9$ chmod +x task9.sh
user@user:~/Documents/github/dev_ops_hw/task9$ ./task9.sh
Коль-во файлов: 4
Коль-во директорий: 2
user@user:~/Documents/github/dev_ops_hw/task9$ ls -l
итого 12
drwxrwxr-x 2 user user 4096 Jul 29 10:57 dir1
drwxrwxr-x 2 user user 4096 Jul 29 10:57 dir2
-rw-rw-r-- 1 user user    0 Jul 29 10:57 file1.txt
-rw-rw-r-- 1 user user    0 Jul 29 10:57 file2.txt
-rw-rw-r-- 1 user user    0 Jul 29 10:57 file3.txt
-rwxrwxr-x 1 user user  305 Jul 29 10:58 task9.sh
user@user:~/Documents/github/dev_ops_hw/task9$ 
```

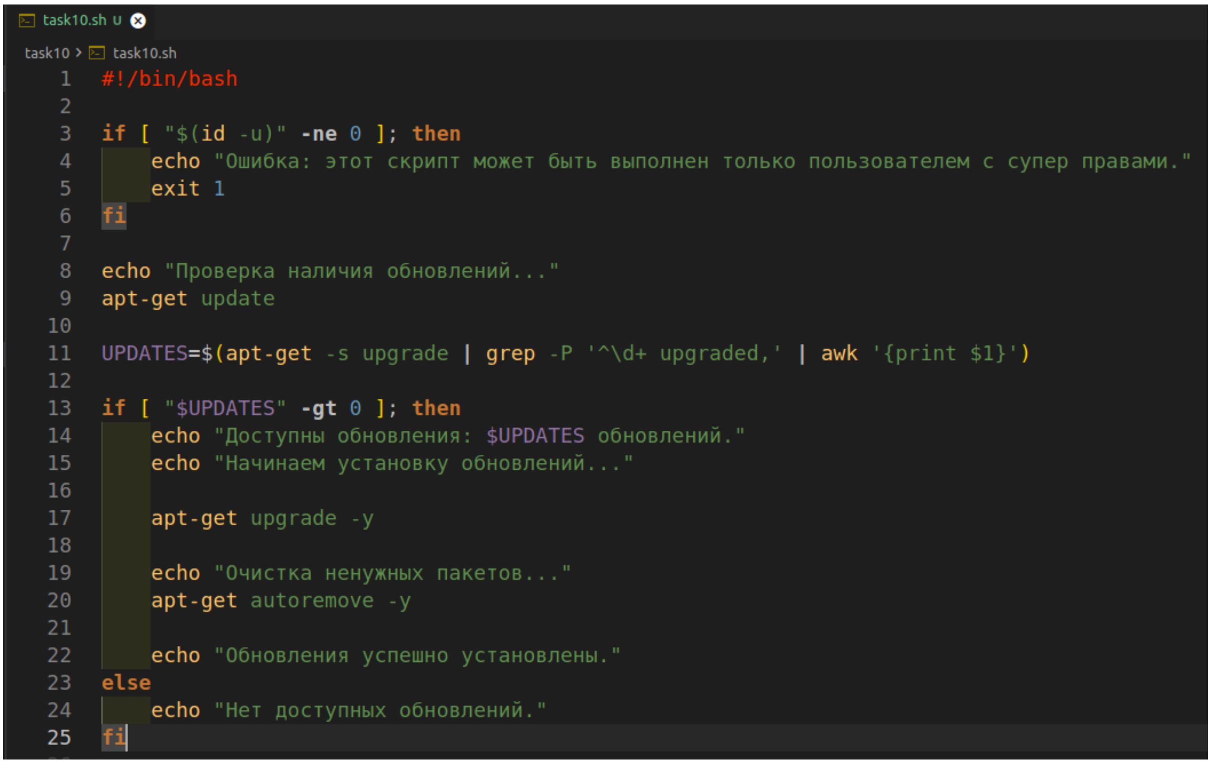
10. Автоматизация задачи обновления системы:



```bash
#!/bin/bash

if [ "$(id -u)" -ne 0 ]; then
    echo "Ошибка: этот скрипт может быть выполнен только пользователем с супер правами."
    exit 1
fi

echo "Проверка наличия обновлений..."
apt-get update

UPDATES=$(apt-get -s upgrade | grep -P '^\d+ upgraded,' | awk '{print $1}')

if [ "$UPDATES" -gt 0 ]; then
    echo "Доступны обновления: $UPDATES обновлений."
    echo "Начинаем установку обновлений..."

    apt-get upgrade -y

    echo "Очистка ненужных пакетов..."
    apt-get autoremove -y

    echo "Обновления успешно установлены."
else
    echo "Нет доступных обновлений."
fi

```



```
user@user:~/Documents/github/dev_ops_hw/task10$ chmod +x task10.sh
user@user:~/Documents/github/dev_ops_hw/task10$ ./task10.sh
Ошибка: этот скрипт может быть выполнен только пользователем с супер правами.
user@user:~/Documents/github/dev_ops_hw/task10$ sudo ./task10.sh
[sudo] пароль для user:
Проверка наличия обновлений...
Сущ:1 http://kz.archive.ubuntu.com/ubuntu jammy InRelease
Пол:2 http://kz.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Сущ:3 https://dl.google.com/linux/chrome/deb stable InRelease
Пол:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Сущ:5 https://ppa.launchpadcontent.net/obsproject/obs-studio/ubuntu jammy InRelease
Сущ:6 http://kz.archive.ubuntu.com/ubuntu jammy-backports InRelease
Сущ:7 https://apt.postgresql.org/pub/repos/apt jammy-pgdg InRelease
Сущ:8 https://packagecloud.io/golang-migrate/migrate/ubuntu jammy InRelease
Получено 257 kB за 3с (88.1 kB/s)
Чтение списков пакетов… Готово
N: Пропускается получение настроенного файла «main/binary-i386/Packages», так как репозиторий «https://apt.postgresql.org/pub/repos/apt jammy-pgdg InRelease» не поддерживает архит
ектуру «i386»
W: https://packagecloud.io/golang-migrate/migrate/ubuntu/dists/jammy/InRelease: Ключ хранится в унаследованной связке ключей trusted.gpg (/etc/apt/trusted.gpg), подробности см. в
разделе DEPRECATION в apt-key(8).
./task10.sh: строка 13: [: : ожидается целочисленное выражение
Нет доступных обновлений.
user@user:~/Documents/github/dev_ops_hw/task10$ 
```



```bash
#!/bin/bash

if [ "$(id -u)" -ne 0 ]; then
    echo "Ошибка: этот скрипт может быть выполнен только пользователем с супер правами."
    exit 1
fi

echo "Проверка наличия обновлений..."
apt-get update

UPDATES=$(apt-get -s upgrade | grep -P '^\d+ upgraded,' | awk '{print $1}')

if [ "$UPDATES" -gt 0 ]; then
    echo "Доступны обновления: $UPDATES обновлений."
    echo "Начинаем установку обновлений..."

    apt-get upgrade -y

    echo "Очистка ненужных пакетов..."
    apt-get autoremove -y

    echo "Обновления успешно установлены."
else
    echo "Нет доступных обновлений."
fi
```