

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:

```
task3.sh u
task3.sh
1  #!/bin/bash
2
3  TARGET="ключевое_слово"
4
5  # Альтернатива чтобы сделать скрипт переиспользуемым:
6  # read TARGET
7  # echo $TARGET
8
9  find . -type f -exec grep -l "$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$ █

```

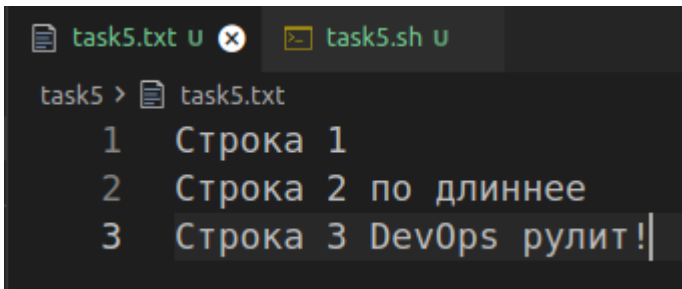
4. Архивирование и распаковка:

```

● 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:



```

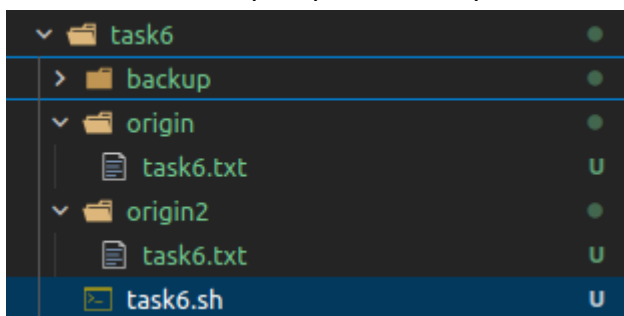
task5 > task5.txt
1 Строка 1
2 Строка 2 по длиннее
3 Строка 3 DevOps рулит!

```

```
task5.sh U task5.txt U
task5 > task5.sh
1  #!/bin/bash
2
3  TARGET="task5.txt"
4
5  # Альтернатива чтобы сделать скрипт переиспользуемым:
6  # read TARGET
7
8  # Дополнительная логика для проверки наличия файла.
9  if [ ! -f "$TARGET" ]; then
10     echo "Ошибка: Файл $TARGET не найден."
11     exit 1
12 fi
13
14 # Если я правильно понял задачу, то будет выведена каждая строка в изначальной последовательности,
15 # однако каждая строка будет выведена на экран в обратном порядке по символам.
16 awk '{ print | "rev" }' "$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
ееннилд оп 2 акортС
!тилур sp0veD 3 акортС
○ user@user:~/Documents/github/dev_ops_hw/task5$
```

6. Автоматизация резервного копирования:

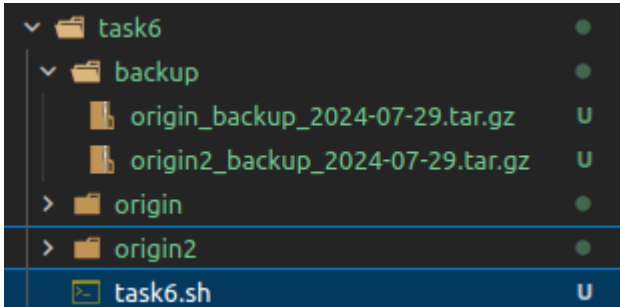


```
task6.sh U
task6 > task6.sh
1  #!/bin/bash
2
3  DIRECTORIES="./origin ./origin2"
4
5  BACKUP_DEST="./backup"
6
7  for DIR in $DIRECTORIES; do
8     BASENAME=$(basename "$DIR")
9
10     DATE=$(date +%Y-%m-%d)
11
12     BACKUP_FILE="${BACKUP_DEST}/${BASENAME}_backup_${DATE}.tar.gz"
13
14     tar -czf "$BACKUP_FILE" "$DIR" # проще все упаковать в архив и хранить.
15
16     echo "Резервная копия $DIR сохранена: $BACKUP_FILE."
17 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$ █

```



```

● 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:

```

input.txt x task7.sh u
task7 > input.txt
1
2 В данном файле хранится ровно 7 слов.

```

```
task7.sh u x input.txt u
task7 > task7.sh
1  #!/bin/bash
2
3  read FILE
4
5  if [ ! -f "$FILE" ]; then
6      echo "Ошибка: $FILE не доступен."
7      exit 1
8  fi
9
10 WORD_COUNT=$(wc -w < "$FILE")
11
12 echo "Результат $FILE: $WORD_COUNT"
13
```

```
ТЕРМИНАЛ  КОНСОЛЬ ОТЛАДКИ
● 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:

```
task8.sh U x password.txt U
task8 > task8.sh
1  #!/bin/bash
2
3  generate_password() {
4      local length="$1"
5      < /dev/urandom tr -dc 'A-Za-z0-9_@#%&*+-' | head -c "$length"
6  }
7
8  echo "Длина пароля: "
9  read PASSWORD_LENGTH
10
11 echo "Файл: "
12 read TARGET_FILE
13
14 # Проверка, нельзя принимать отрицательные числа.
15 if ! [[ "$PASSWORD_LENGTH" =~ ^[0-9]+$ ]] || [ "$PASSWORD_LENGTH" -le 0 ]; then
16     echo "Ошибка: Длина пароля должна быть положительным числом."
17     exit 1
18 fi
19
20 generate_password "$PASSWORD_LENGTH" > "$TARGET_FILE"
21
22 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$
```

```
password.txt U x task8.sh U
task8 > password.txt
1  AvYA4
```

9. Подсчет файлов:

```

task9.sh u
task9 > task9.sh
1  #!/bin/bash
2
3  FILE_COUNT=0
4  DIR_COUNT=0
5
6  for item in *; do
7      if [ -f "$item" ]; then
8          FILE_COUNT=$((FILE_COUNT + 1))
9      elif [ -d "$item" ]; then
10         DIR_COUNT=$((DIR_COUNT + 1))
11     fi
12 done
13
14 echo "Коль-во файлов: $FILE_COUNT"
15 echo "Коль-во директорий: $DIR_COUNT"
16

```

```

● 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:

```

task10.sh U
task10 > task10.sh
1  #!/bin/bash
2
3  if [ "$(id -u)" -ne 0 ]; then
4      echo "Ошибка: этот скрипт может быть выполнен только пользователем с супер правами."
5      exit 1
6  fi
7
8  echo "Проверка наличия обновлений..."
9  apt-get update
10
11  UPDATES=$(apt-get -s upgrade | grep -P '^d+ upgraded,' | awk '{print $1}')
12
13  if [ "$UPDATES" -gt 0 ]; then
14      echo "Доступны обновления: $UPDATES обновлений."
15      echo "Начинаем установку обновлений..."
16
17      apt-get upgrade -y
18
19      echo "Очистка ненужных пакетов..."
20      apt-get autoremove -y
21
22      echo "Обновления успешно установлены."
23  else
24      echo "Нет доступных обновлений."
25  fi
26

```

```

user@user:~/Documents/github/dev_ops_hw/task10$ chmod +x task10.sh
user@user:~/Documents/github/dev_ops_hw/task10$ ./task10.sh
[sudo] пароль для user:
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-1386/Packages», так как репозиторий «https://apt.postgresql.org/pub/repos/apt jammy-pgdg InRelease» не поддерживает архитектуру «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$

```