**Урок 4. Скрипты Bash**

**Задание.**

Написать скрипт очистки директорий. На вход принимает путь к директории. Если директория существует, то удаляет в ней все файлы с расширениями .bak, .tmp, .backup. Если директории нет, то выводит ошибку.

\* Создать скрипт ownersort.sh, который в заданной папке копирует файлы в директории, названные по имени владельца каждого файла. Учтите, что файл должен принадлежать соответствующему владельцу.

**Решение.**

В директории **HomeWork11\_04** создал 2 файла:  
- основной скрипт домашнего задания - **HomeWork11\_04.sh**;

- дополнительный скрипт, генерирующий случайную файловую структуру - **generateFileStructure.sh** (его запуск вызывается из основного скрипта).

**Код основного скрипта.**

#!/bin/bash

# Функция удаления

del\_files() {

# Проверяем, существует ли папка

if [ -d "." ]; then

echo "Удаление файлов с расширениями .bak, .tmp и .backup в папке $(pwd)."

# Удаляем файлы с расширениями .bak, .tmp и .backup  
# внутри папки rm \*.bak \*.tmp \*.backup

rm \*.bak \*.tmp \*.backup

else

echo "Папка $(pwd) не существует."

fi

}

### КОД СКРИПТА ###

# Показ исходной файловой структуры

echo "Исходная файловая структура в текущей папке"

tree

# Создание случайной файловой системы и ее показ

targetFolder="$1"

bash generateFileStructure.sh "$targetFolder"

echo "Скорректированная файловая структура после случайной генерации"

tree

# Проверка существования папки

if [ -z "$targetFolder" ]; then

echo "Не указан путь к папке."

exit 1

fi

# Заходим в папку targetFolder

cd "$targetFolder" || exit 1

# Удаляем файлы

del\_files

# Получаем список папок внутри targetFolder

subfolders=$(ls -d \*/)

# Цикл для обхода папок

for folder in $subfolders

do

# Удаляем символ "/" из имени папки

folder=${folder%/}

# Переходим в папку

cd "$folder" || continue

# Удаляем необходимые файлы текущей папки

del\_files

# Возвращаемся обратно в папку targetFolder

cd ..

done

# Возвращаемся обратно на уровень вверх

cd ..

# Показ файловой системы после удаления необходимых файлов

echo "Скорректированная файловая структура после удаления файлов с расширениями .bak, .tmp и .backup"

tree

# Удаление всего созданного в домашке и показ файловой системы

echo "Удаление следов работы скрипта домашней работы."

rm -r "$targetFolder"

# Показ файловой системы после удаления следов работы скрипта

echo "Файловая структура в текущей папке на данный момент"

tree

**Код скрипта генерации случайной файловой системы.**

#!/bin/bash

### ФУНКЦИИ ###

# Функция для создания заданного количества папок

create\_folders() {

local num\_folders=$1

for ((i=1; i<=num\_folders; i++))

do

folder\_name="folder$i"

mkdir "$folder\_name"

done

}

# Функция для создания заданного количества файлов

create\_files() {

local num\_files=$1

for ((i=1; i<=num\_files; i++))

do

file\_name="file$i"

# ТРИ КОМАНДы НИЖЕ ТОЛЬКО ДЛЯ РЕШЕНИЯ ЗАДАЧИ 4

# ДЛЯ УНИВЕРСАЛЬНОГО ПРМЕНЕНИЯ УДАЛИТЬ

random\_number=$(shuf -i 1-4 -n 1)

if [ "$random\_number" -eq 1 ]; then

extension=".bak"

elif [ "$random\_number" -eq 2 ]; then

extension=".tmp"

elif [ "$random\_number" -eq 3 ]; then

extension=".backup"

else

extension=".txt"

fi

file\_name="file$i$extension"

touch "$file\_name"

done

}

# Функция для создания случайной файловой системы

generateFoldersFiles() {

# Генерируем случайное количество папок и файлов (от 1 до 10)

local num\_folders=$(( (RANDOM % 10) + 1 ))

local num\_files=$(( (RANDOM % 10) + 1 ))

# Создаем папки в директории

create\_folders "$num\_folders"

# Создаем файлы в директории

create\_files "$num\_files"

}

### КОД СКРИПТА ###

targetFolder=$1

# Проверяем, было ли передано название папки

if [ -z "$targetFolder" ]; then

echo "Не указано название папки. Используйте: ./script.sh <название\_папки>"

exit 1

fi

# Создаем папку с указанным названием

mkdir "$targetFolder"

echo "Создана папка: $targetFolder"

# Переходим в папку targetFolder

cd "$targetFolder" || exit

# Генерируем случайную файловую структуру внути текущей папки

generateFoldersFiles

echo "Сгенерирована случайная файловая структура в папке: $targetFolder"

# Получаем список папок внутри targetFolder

subfolders=$(ls -d \*/)

# Цикл для обхода папок

for folder in $subfolders

do

# Удаляем символ "/" из имени папки

folder=${folder%/}

# Переходим в папку

cd "$folder" || continue

# Генерируем случайную файловую систему внути текущей папки

generateFoldersFiles

echo "Сгенерирована случайная файловая структура в папке: $(pwd)"

# Возвращаемся обратно в папку targetFolder

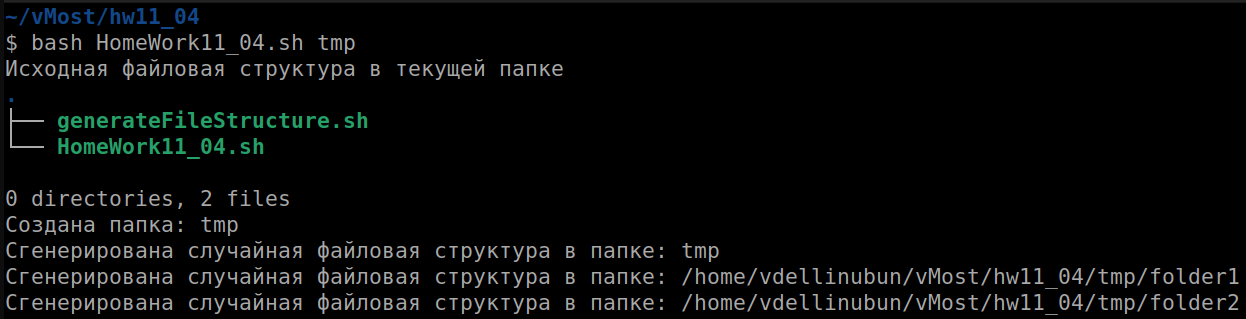
cd ..

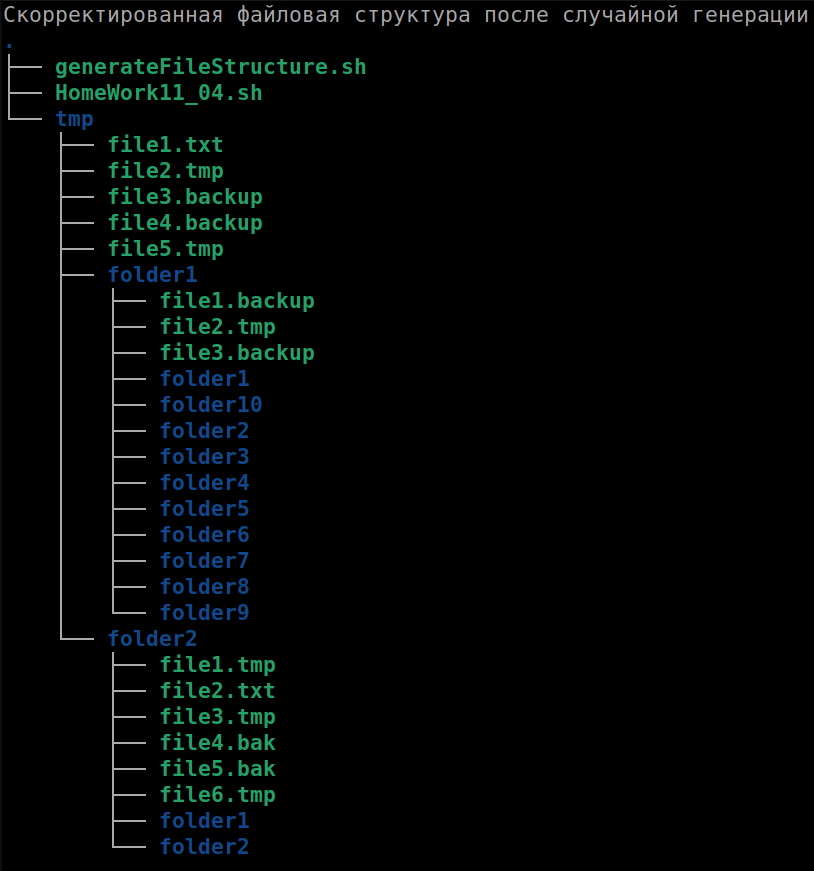
done

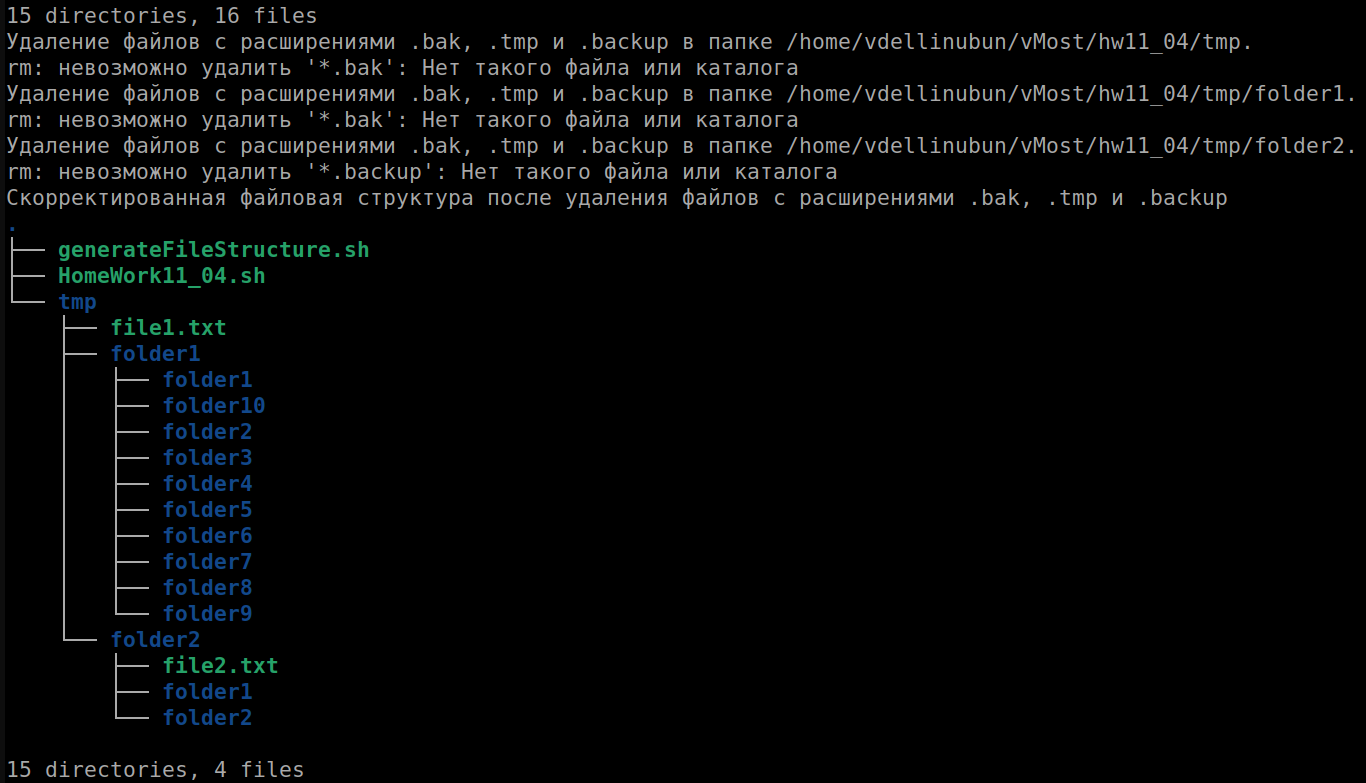
# Возвращаемся обратно из папки targetFolder

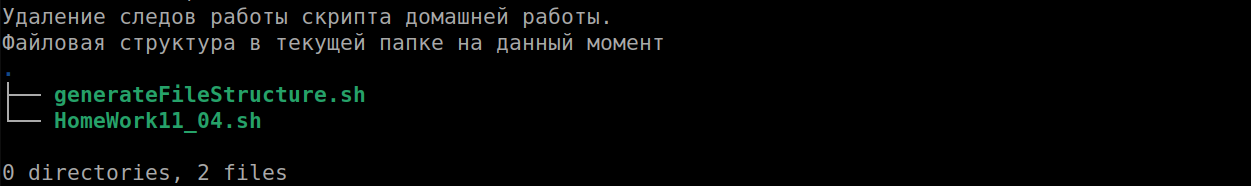
cd ..

**Скриншоты работы скрипта**

****

****

****

****