

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
    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 ПРИМЕНЕНИЯ 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 внутри 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 внутри 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

```
~/vMost/hw11_04
$ bash HomeWork11_04.sh tmp
Исходная файловая структура в текущей папке

.
├── generateFileStructure.sh
└── HomeWork11_04.sh

0 directories, 2 files
Создана папка: tmp
Сгенерирована случайная файловая структура в папке: tmp
Сгенерирована случайная файловая структура в папке: /home/vdellinubun/vMost/hw11_04/tmp/folder1
Сгенерирована случайная файловая структура в папке: /home/vdellinubun/vMost/hw11_04/tmp/folder2
```

Скорректированная файловая структура после случайной генерации

```
.
├── generateFileStructure.sh
├── HomeWork11_04.sh
└── tmp
    ├── file1.txt
    ├── file2.tmp
    ├── file3.backup
    ├── file4.backup
    ├── file5.tmp
    ├── folder1
    │   ├── file1.backup
    │   ├── file2.tmp
    │   ├── file3.backup
    │   ├── folder1
    │   ├── folder10
    │   ├── folder2
    │   ├── folder3
    │   ├── folder4
    │   ├── folder5
    │   ├── folder6
    │   ├── folder7
    │   ├── folder8
    │   └── folder9
    └── folder2
        ├── file1.tmp
        ├── file2.txt
        ├── file3.tmp
        ├── file4.bak
        ├── file5.bak
        ├── file6.tmp
        ├── folder1
        └── folder2
```

15 directories, 16 files

Удаление файлов с расширениями .bak, .tmp и .backup в папке /home/vdellinubun/vMost/hw11_04/tmp.

rm: невозможно удалить '*.bak': Нет такого файла или каталога

Удаление файлов с расширениями .bak, .tmp и .backup в папке /home/vdellinubun/vMost/hw11_04/tmp/folder1.

rm: невозможно удалить '*.bak': Нет такого файла или каталога

Удаление файлов с расширениями .bak, .tmp и .backup в папке /home/vdellinubun/vMost/hw11_04/tmp/folder2.

rm: невозможно удалить '*.backup': Нет такого файла или каталога

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

```
.
├── generateFileStructure.sh
├── HomeWork11_04.sh
├── tmp
│   ├── file1.txt
│   ├── folder1
│   │   ├── folder1
│   │   ├── folder10
│   │   ├── folder2
│   │   ├── folder3
│   │   ├── folder4
│   │   ├── folder5
│   │   ├── folder6
│   │   ├── folder7
│   │   ├── folder8
│   │   └── folder9
│   └── folder2
│       ├── file2.txt
│       ├── folder1
│       └── folder2
```

15 directories, 4 files

Удаление следов работы скрипта домашней работы.

Файловая структура в текущей папке на данный момент

```
.
├── generateFileStructure.sh
└── HomeWork11_04.sh
```

0 directories, 2 files