

Bash: Как сделать автоматическую замену различных фрагментов текста на однообразный фрагмент во всех подкаталогах и файлах?

Category: bash & sh, bash программирование, find, GNU/Linux, sed
2026-08-13

Задача.

Имеется каталог web-сайта или несколько каталогов с разнообразным содержимым. Внутри каталогов есть еще каталоги и так далее... Целое дерево каталогов.

```
/tmp/test-replacement-folder/  
├── test-folder-1  
│   ├── test_file_1.txt  
│   ├── test_file_2.txt  
│   └── test_file_3.txt  
├── test-folder-2  
│   ├── test_file_1.txt  
│   ├── test_file_2.txt  
│   └── test_file_3.txt  
└── test-folder-3  
    ├── test_file_1.txt  
    ├── test_file_2.txt  
    └── test_file_3.txt
```

Требуется сделать автоматическую замену фрагмента текста на другой фрагмент текста во всех файлах внутри дерева каталогов.

```
/tmp/test-replacement-folder/test-folder-1/test_file_1.txt  
10.25.36.25  
10.25.36.26  
10.25.36.27  
10.25.36.28  
10.25.36.29  
10.25.36.30  
█
```

Фрагменты, которые требуется заменить:

- 10.25.36.27
- 10.25.36.28

Фрагмент, на который требуется заменить:

- 192.57.36.15

Создание тестового окружения.

Создадим и заполним скрипт `test_environment.sh` создания нужных для автозамены каталогов и файлов:

```
$ cd ~
$ touch test_environment.sh
$ mcedit test_environment.sh
```

Способ 1: Цикл `for`.

```
#!/bin/bash
# создание дерева каталогов
mkdir -p /tmp/test-replacement-folder/{test-folder-1,test-
folder-2,test-folder-3}
# создание файлов
touch /tmp/test-replacement-folder/{test-folder-1,test-
folder-2,test-
folder-3}/{test_file_1.txt,test_file_2.txt,test_file_3.txt}
# заполнение файлов информацией
for file in /tmp/test-replacement-folder/{test-folder-1,test-
folder-2,test-
folder-3}/{test_file_1.txt,test_file_2.txt,test_file_3.txt}; do
cat << EOF > "$file"
10.25.36.25
10.25.36.26
10.25.36.27
10.25.36.28
10.25.36.29
10.25.36.30
EOF
done
```

Способ 2: Использование `tee`.

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

```
# создание дерева каталогов
mkdir -p /tmp/test-replacement-folder/{test-folder-1,test-
folder-2,test-folder-3}
# создание файлов
touch /tmp/test-replacement-folder/{test-folder-1,test-
folder-2,test-
folder-3}/{test_file_1.txt,test_file_2.txt,test_file_3.txt}
# заполнение файлов информацией
echo -e
"10.25.36.25\n10.25.36.26\n10.25.36.27\n10.25.36.28\n10.25.36.29\n10
.25.36.30" | \
tee /tmp/test-replacement-folder/{test-folder-1,test-folder-2,test-
folder-3}/{test_file_1.txt,test_file_2.txt,test_file_3.txt} >
/dev/null
```

Способ 3: Использование printf с циклом.

```
#!/bin/bash
# создание дерева каталогов
mkdir -p /tmp/test-replacement-folder/{test-folder-1,test-
folder-2,test-folder-3}
# создание файлов
touch /tmp/test-replacement-folder/{test-folder-1,test-
folder-2,test-
folder-3}/{test_file_1.txt,test_file_2.txt,test_file_3.txt}
# заполнение файлов информацией
for file in /tmp/test-replacement-folder/{test-folder-1,test-
folder-2,test-
folder-3}/{test_file_1.txt,test_file_2.txt,test_file_3.txt}; do
printf "%s\n" "10.25.36.25" "10.25.36.26" "10.25.36.27"
"10.25.36.28" "10.25.36.29" "10.25.36.30" > "$file"
done
```

Запустим его:

```
$ bash test_environment.sh
```

Проверим факт того, что всё создано корректно:

```
$ tree /tmp/test-replacement-folder/
```

Ответ:

```
aleksandr [REDACTED]@ [REDACTED]:/tmp$ tree /tmp/test-replacement-folder/  
/tmp/test-replacement-folder/  
├── test-folder-1  
│   ├── test_file_1.txt  
│   ├── test_file_2.txt  
│   └── test_file_3.txt  
├── test-folder-2  
│   ├── test_file_1.txt  
│   ├── test_file_2.txt  
│   └── test_file_3.txt  
└── test-folder-3  
    ├── test_file_1.txt  
    ├── test_file_2.txt  
    └── test_file_3.txt
```

Удаление тестового окружения:

```
$ rm -fr /tmp/test-replacement-folder/
```

Выполнение задачи.

В арсенале GNU\Linux есть пара интересных утилит: утилита **find** и утилита **sed**.

Утилита **find** — для поиска файлов и каталогов на основе специальных условий. Ее можно использовать в различных обстоятельствах, например, для поиска файлов по разрешениям, владельцам, группам, типу, размеру и другим подобным критериям.

Утилита **sed** — это потоковый редактор текста, работающий по принципу замены. Его можно использовать для поиска, вставки, замены и удаления фрагментов в файле. С помощью этой утилиты вы можете редактировать файлы не открывая их. Будет намного быстрее если вы напишите что и на что надо заменить, чем вы будете открывать редактор **mcedit**, искать нужную строку и вручную всё заменять.

Команда в одну строку.

Общий функционал команды легко уместится в одну строку:

```
# find '/mesto-poiska/' -type f -name '*.txt' -exec sed -i -r  
's/THIS-IS/TO-THAT/g' {} \;
```

где:

- **/mesto-poiska/** — каталог, в котором будет осуществлён поиск;
- ***** — все файлы подряд, опционально можно добавить маску файлов

по ситуации, например, *.txt;

- **THIS-IS** — то, что ищем и хотим заменить;
- **TO-THAT** — то, на что мы хотим заменить, то, что ищем.

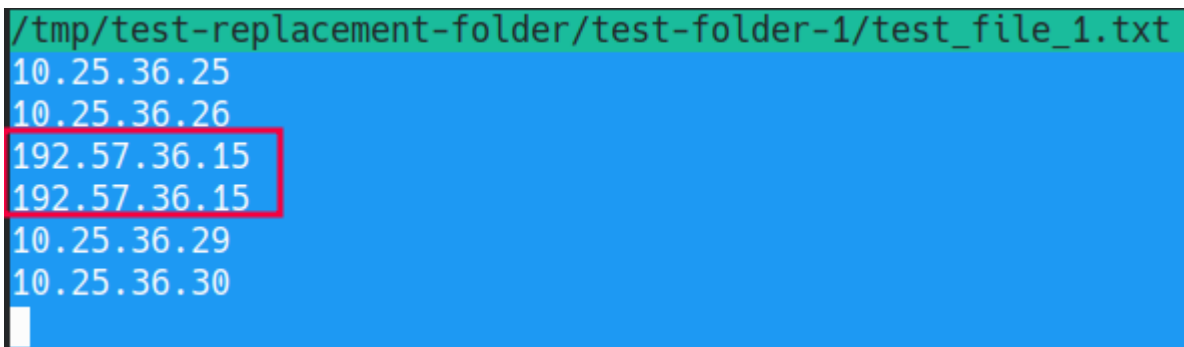
Попробуем вводом каждого факта замены поштучно:

```
# find '/tmp/test-replacement-folder/' -type f -name '*.txt' -exec  
sed -i -r 's/10.25.36.27/192.57.36.15/g' {} \;  
# find '/tmp/test-replacement-folder/' -type f -name '*.txt' -exec  
sed -i -r 's/10.25.36.28/192.57.36.15/g' {} \;
```

Проверяем факты замены:

```
$ cat /tmp/test-replacement-folder/{test-folder-1,test-  
folder-2,test-  
folder-3}/{test_file_1.txt,test_file_2.txt,test_file_3.txt}
```

Ответ (во всех файлах будет такое содержимое):



```
/tmp/test-replacement-folder/test-folder-1/test_file_1.txt  
10.25.36.25  
10.25.36.26  
192.57.36.15  
192.57.36.15  
10.25.36.29  
10.25.36.30  
█
```

Автоматизация процесса.

Каждый раз использовать данную команду, конечно, хорошо, но не особо удобно. Удобнее написать автоматизированный скрипт и вносить в него то, что нам требуется.

Автоматизируем однострочную команду из предыдущего пункта.

- Сделаем в скрипте переменные, в которые запрограммируем наши фрагменты для автозамены.
- Приделаем систему журналирования действий и результатов.
- В цикл поместим список каталогов, по которым требуется пройти и заменить в них фрагменты.

Пересоздадим окружение:

```
$ rm -fr /tmp/test-replacement-folder/ && bash ~/test_environment.sh
```

Проверим факт корректного создания:

```
$ cat /tmp/test-replacement-folder/{test-folder-1,test-  
folder-2,test-  
folder-3}/{test_file_1.txt,test_file_2.txt,test_file_3.txt}
```

Ответ (во всех файлах будет такое содержимое):

```
/tmp/test-replacement-folder/test-folder-1/test_file_1.txt  
10.25.36.25  
10.25.36.26  
10.25.36.27  
10.25.36.28  
10.25.36.29  
10.25.36.30  
█
```

Для примера сделаем скрипт поиска и замены пары старых **IP-адресов** на один новый **IP-адрес**:

```
$ cd ~  
$ touch test_replacement.sh  
$ mcedit test_replacement.sh
```

Разверните для просмотра...

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

```
## Задаем переменные.
```

```
# Фрагменты, которые требуется заменить.
```

```
word_off_1='10.25.36.27'
```

```
word_off_2='10.25.36.28'
```

```
# Фрагмент, на который требуется заменить.
```

```
word_on='192.57.36.15'
```

```
# Лог-файл.
```

```
log_path='/tmp/find.log'
```

```
## Каталоги для операций с заменой могут быть любые, ищет насквозь.
```

```
srv_paths=(
```

```
    /tmp/test-replacement-folder/test-folder-1
```

```
    /tmp/test-replacement-folder/test-folder-2
```

```
    /tmp/test-replacement-folder/test-folder-3
```

```
)
```

```
# Сброс содержимого log-файла вывода результатов работы скрипта
```

```

перед новым циклом работы скрипта.
> $log_path;

echo "`date +%H:%M` Скрипт стартовал" >> $log_path;
echo "-----" >> $log_path;

# Действия по замене IP-адресов.
echo -e "\nСкрипт автоматической замены успешно стартовал!";
echo -e "\nВремя запуска: `date +%H:%M`";
echo -e "\nПожалуйста, дождитесь окончания работы скрипта...";

# Счетчик подсчета затраченного времени на автоматическую замену.
TIMEFORMAT="Общее время выполнения скрипта составило %lR"
time {

    # Фрагмент работы автоматического поиска и замены фрагментов
    текста.
    cd /;
    for line in ${srv_paths[@]};
    do

        echo -e "\n----- '$line' -----";

        # WORD_1
        echo "Автоматическая замена $word_off_1 на $word_on" в
каталоге $line;
        find $line -type f -name '*.txt' -exec sed -i -r
's/'$word_off_1'/'$word_on'/g' {} \;
        # Проверка успеха выполненной операции, если "0", то скрипт
отработал без ошибок.
        echo "Status '$?' - All OK, if status return 0";
        # Визуализация переменных для автоматическая замена.
        echo "'$word_off_1' --> '$word_on' = $line";
        # Контрольная запись о результатах этапа в log-файл.
        echo "`date +%H:%M` '$word_off_1' --> '$word_on' = $line"
>> $log_path;
        echo "Этап пройден в `date +%H:%M`";
        echo -e "";

        # WORD_2
        echo "Автоматическая замена $word_off_2 на $word_on" в
каталоге $line;
        find $line -type f -name '*.txt' -exec sed -i -r
's/'$word_off_2'/'$word_on'/g' {} \;
        # Проверка успеха выполненной операции, если "0", то скрипт
отработал без ошибок.
        echo "Status '$?' - All OK, if status return 0";

```

```

# Визуализация переменных для автоматическая замена
echo "'$word_off_2' --> '$word_on' = $line";
# Контрольная запись о результатах этапа в log-файл.
echo "`date +%H:%M` '$word_off_2' --> '$word_on' = $line"
>> $log_path;
echo "Этап пройден в `date +%H:%M`";
echo -e "";

echo "-----"
----";

# Перегородка между этапами замены слов по каталогам в log-
файле.
echo "-----" >> $log_path;

done
}

echo -e "\nВремя окончания: `date +%H:%M`";
echo -e "\nСкрипт успешно закончил работу!";
echo -e "";

echo "`date +%H:%M` Скрипт закончил работу" >> $log_path;

# После работы скрипта, выход с нулевым статусом.
exit 0

```

Запустим его:

```
$ bash test_replacement.sh
```

Ответ:

Скрипт автоматической замены успешно стартовал!

Время запуска: 14:55

Пожалуйста, дождитесь окончания работы скрипта...

```

----- '/tmp/test-replacement-folder/test-folder-1' -----
Автоматическая замена 10.25.36.27 на 192.57.36.15 в каталоге
/tmp/test-replacement-folder/test-folder-1
Status '0' - All OK, if status return 0
'10.25.36.27' --> '192.57.36.15' = /tmp/test-replacement-
folder/test-folder-1

```

Этап пройден в 14:55

Автоматическая замена 10.25.36.28 на 192.57.36.15 в каталоге
/tmp/test-replacement-folder/test-folder-1

Status '0' - All OK, if status return 0

'10.25.36.28' --> '192.57.36.15' = /tmp/test-replacement-
folder/test-folder-1

Этап пройден в 14:55

----- '/tmp/test-replacement-folder/test-folder-2' -----

Автоматическая замена 10.25.36.27 на 192.57.36.15 в каталоге
/tmp/test-replacement-folder/test-folder-2

Status '0' - All OK, if status return 0

'10.25.36.27' --> '192.57.36.15' = /tmp/test-replacement-
folder/test-folder-2

Этап пройден в 14:55

Автоматическая замена 10.25.36.28 на 192.57.36.15 в каталоге
/tmp/test-replacement-folder/test-folder-2

Status '0' - All OK, if status return 0

'10.25.36.28' --> '192.57.36.15' = /tmp/test-replacement-
folder/test-folder-2

Этап пройден в 14:55

----- '/tmp/test-replacement-folder/test-folder-3' -----

Автоматическая замена 10.25.36.27 на 192.57.36.15 в каталоге
/tmp/test-replacement-folder/test-folder-3

Status '0' - All OK, if status return 0

'10.25.36.27' --> '192.57.36.15' = /tmp/test-replacement-
folder/test-folder-3

Этап пройден в 14:55

Автоматическая замена 10.25.36.28 на 192.57.36.15 в каталоге
/tmp/test-replacement-folder/test-folder-3

Status '0' - All OK, if status return 0

'10.25.36.28' --> '192.57.36.15' = /tmp/test-replacement-
folder/test-folder-3

Этап пройден в 14:55

Общее время выполнения скрипта составило 0m0,090s

Время окончания: 14:55

Скрипт успешно закончил работу!

Содержимое файла журналирования find.log работы скрипта:

```
/tmp/find.log [----] 0 L: [ 1+12 13/ 13] *(659 / 659b) <EOF>
15:01 Скрипт стартовал
-----
15:01 '10.25.36.27' --> '192.57.36.15' = /tmp/test-replacement-folder/test-folder-1
15:01 '10.25.36.28' --> '192.57.36.15' = /tmp/test-replacement-folder/test-folder-1
-----
15:01 '10.25.36.27' --> '192.57.36.15' = /tmp/test-replacement-folder/test-folder-2
15:01 '10.25.36.28' --> '192.57.36.15' = /tmp/test-replacement-folder/test-folder-2
-----
15:01 '10.25.36.27' --> '192.57.36.15' = /tmp/test-replacement-folder/test-folder-3
15:01 '10.25.36.28' --> '192.57.36.15' = /tmp/test-replacement-folder/test-folder-3
-----
15:01 Скрипт закончил работу
```

Проверяем целевые файлы на факт замены:

```
$ cat /tmp/test-replacement-folder/{test-folder-1,test-
folder-2,test-
folder-3}/{test_file_1.txt,test_file_2.txt,test_file_3.txt}
```

Ответ (во всех файлах будет такое содержимое):

```
/tmp/test-replacement-folder/test-folder-1/test_file_1.txt
10.25.36.25
10.25.36.26
192.57.36.15
192.57.36.15
10.25.36.29
10.25.36.30
```

Как видно замена прошла успешно!

Опционально вы можете сделать ввод фрагментов текста с захватом с клавиатуры, расширить список фрагментов для замены, приделать отправку сообщения на почту и так далее.

Оригиналы источников информации.

1. losst.ru «Команда find в Linux.»
2. losst.ru «Команда sed в Linux.»
3. novall.net «Простые способы автозамены текста в Linux через SSH.»