

sed

sed — это потоковый текстовый редактор. В отличие от текстового редактора `vim`, с которым мы уже сталкивались, `sed` неинтерактивный.

Работа с `sed` строится по следующей схеме: у нас есть некоторые инструкции, которые нужно применить к потоку данных, то есть к файлу, который передан в качестве параметра, либо к тексту, который передан на стандартный ввод:
`sed` -е инструкция файл.

Мы будем пользоваться следующими операциями `sed`. Это `p` — печать (на `stdout`), `d` — удаление и `s` — замена подстрок, `c` — замена строки под каким-то номером.

Давайте посмотрим на примере.

Начнём с команды печати. Давайте попробуем вывести содержимое файла.

```
sed 'p' /etc/passwd
```

Итак, мы видим содержимое файла, но строки повторяются по два раза. Происходит это из-за того, что `sed` автоматически выводит файл. Чтобы этого не происходило, отключим вывод:

```
sed -n 'p' /etc/passwd
```

`sed` оперирует понятием «адресное пространство» для определения, над какими строками необходимо произвести какие-то операции. Если адресное пространство не задано, то для обработки принимаются все строки. Адресное пространство может задаваться номером строки или шаблоном.

Номер позволяет указать номер строки, в которой надо выполнять команду;

- первая~шаг — команда будет выполняться для указанной в первой части строки, а затем для всех с указанным шагом;
- \$ — последняя строка в файле;
- /регулярное_выражение/ — любая строка, которая подходит по регулярному выражению. Модификатор `l` указывает, что регулярное выражение должно быть нечувствительным к регистру;
- номер, номер — начиная от строки из первой части и заканчивая строкой из второй части;
- номер, /регулярное_выражение/ — начиная от строки из первой части и до строки, которая будет соответствовать регулярному выражению;
- номер, +количество — начиная от номера строки, указанного в первой части, и ещё плюс количество строк после него;
- номер, ~число — начиная от номера строки и до строки, номер которой будет кратным числу.

Вот так мы можем вывести только первую строку:

```
sed -n '1p' BSD  
sed -n '1,+4p'
```

Предположим, что нам необходимо удалить комментарии из файла. Это можно сделать при помощи команды `d`. Команда не удалит строки из текста, только из вывода файла на поток стандартного вывода.

```
sed '/^#/ d' file
```

Чтобы изменения записались в файл, требуется использовать ключ `-i`; символ `~` здесь обозначает начало строки.

```
sed -i '/^#/ d' file
```

Также при помощи команды `d` можно удалить пустые строки:

```
$ sed '/^$/d' file
```

Самая популярная возможность `sed` — это замена одного слова на другое. Предположим, у нас есть список названий поддоменов (третьего уровня) в файле в столбик:

```
fire
candle
ice
danger
```

Все они принадлежат домену — предположим, `example.com` (нужно уточнить, что это специальный домен для тестовых и учебных задач). И мы хотели бы сформировать список полных доменных имён, то есть `fire.example.com`.

```
sed -e 's/$/.example.com/' domains.txt
```

Давайте посмотрим на текстовый файл. В нём отрывки разделены словом `Quote`.

Давайте заменим это на что-то другое, например тройную звёздочку:

```
sed -e 's/Quote/***/' pale_fire
```

Такая конструкция действительно работает. Есть одна проблема: если слово встречается в строке больше одного раза, заменено будет только первое вхождение слова. Для того чтобы это работало для любого количества совпадений в строке, добавим `g`:

```
sed -e 's/Quote/***/g' pale_fire
```

Мы можем заменить не только совпадение шаблона, но и целую строку. Например, давайте заменим содержимое третьей строки в файле:

```
sed '3c\This is a modified line.' pale_fire
```

`sed` позволяет однострочником не только отредактировать файл, но и предварительно создать его резервную копию с расширением `.bak`:

```
sed -i.bak '3c\This is a modified line.' pale_fire
```

Для того чтобы определить, сколько места на жёстком диске занято, есть полезная команда `df`. Чаще всего она используется с ключом `-h` для того, чтобы сделать вывод более понятным для человека:

```
df -h
```

Рассмотрим устройство `/dev/sda2`, на которое указывает слеш, корень нашей файловой системы. У вас имя устройства может отличаться. Давайте получим только его из всего вывода команды.

```
df -h | grep "/dev/sda2" | awk '{print $5}'
```

Сейчас занят 61% диска. Если мы захотим наблюдать за этим числом, например написать скрипт, который бы нам говорил, что занято больше 85%, это число нам будет не очень полезно, ведь с ним нельзя сделать арифметическую операцию. Так как тут присутствует посторонний знак, это уже какая-то непонятная строка для `bash`. Так что давайте при помощи команды `sed` это исправим:

```
sed -e 's/%//g'
```