

Введение

Командная строка в Linux

Основы навигации в Linux

Файлы в Linux

Руководство в Linux

Права доступа в Linux

Шаблоны подстановки в Linux

Работа с файлами в Linux

Фильтры в Linux

Конвейеры и перенаправление ввода-вывода в Linux

Управление процессами в Linux

Команда grep и регулярные выражения в Linux

# Шаблоны подстановки в Linux

В статье «[Работа с файлами](#)» вы познакомились с несколькими командами, которые позволяют работать с файлами. Проблема в том, что в том материале мы обрабатывали только один файл. Теперь рассмотрим способ работы сразу с набором файлов. Для этого мы будем использовать шаблоны подстановки (wildcards).

## Что такое шаблоны подстановки

**Шаблоны подстановки (wildcards)** позволяют настраивать фильтры отбора имён файлов и путей. В интернете вы можно встретить и такие названия: *джокеры, шаблоны поиска, маски* (хотя на самом деле это немного другое).

Как вы помните, когда мы обращаемся к файлу или каталогу в командной строке, мы фактически ссылаемся на его путь. Поэтому когда мы указываем путь, можно использовать так называемые символы подстановки в этом пути, чтобы превратить его в набор файлов или каталогов.

Вот основной набор **символов подстановки**:

- `*` — для замены нескольких символов (в том числе `0`);
- `?` — для замены одиночного символа;
- `[ ]` — для замены определенного набора символов.

Символы подстановки также называют *символами-джокерами*.

## Символ подстановки `*`

Давайте рассмотрим несколько примеров, которые продемонстрируют, как работает символ подстановки `*` (**звёздочка**). Он представляет собой несколько символов (в том числе `0`).

**Примечание.** Для всех приведенных представьте, что мы находимся в каталоге `linuxtutorialwork`, в котором находятся такие файлы:





скриптов в  
Linux

- firstfile, secondfile, thirdfile;
- foo1, foo2, foo3;
- video.mpeg.

## Пример 1. Выводим названия файлов, начинающихся с b

В качестве первого примера попробуем вывести все файлы, чьё название начинается с b.

1. **user@bash:** pwd
2. /home/karpaff/linuxtutorialwork
3. **user@bash:**
4. **user@bash:** ls b\*
5. barry.txt blah.txt bob
6. **user@bash:**

## Как это работает

На первый взгляд может показаться, что приведенная выше команда **ls** получает аргумент **b\***, а затем переводит его в требуемые совпадения.

На самом деле это bash (программа, предоставляющая интерфейс командной строки) выполняет перевод за нас. Когда мы вводим эту команду, bash видит, что мы использовали символы подстановки, и поэтому перед запуском команды (в данном случае **ls**) заменяет шаблон каждым файлом или каталогом (то есть путем), который соответствует этому шаблону.

Вот, как это происходит:

1. Мы вводим команду: **ls b\***
2. Система переводит эту команду в такой вид: **ls barry.txt blah.txt bob**
3. Система выполняет команду из пункта 2.

Сама команда не видит символы подстановки и не знает, что мы их применяли. Это означает, что мы их в командной строке когда угодно.





`ls` в примерах просто потому, что это удобный способ проиллюстрировать работу подстановки. Символы подстановки можно использовать с любой командой.

## Пример 2. Выводим все файлы с расширением `.txt`

Давайте попробуем вывести все файлы с расширением `.txt`. В этом примере мы используем абсолютный путь. Обратите внимание, что шаблон подстановки работает одинаково, и с абсолютными, и с относительными путями.

1. `user@bash: ls /home/karpaff/linuxtutorialwork/*.txt`
2. `/home/karpaff/linuxtutorialwork/barry.txt /home/karp`
3. `user@bash:`



## Символ подстановки `?`

Теперь давайте рассмотрим, как работает символ `?` (знак вопроса). Он представляет собой 1 любой символ.

## Пример 3. Выводим все файлы с именем, вторая символ которого – `i`

В примере ниже мы выведем все файлы с именем, второй символ равен `i`. Как видите, шаблон можно строить с помощью нескольких символов подстановки.

1. `user@bash: ls ?i*`
2. `firstfile video.mpeg`
3. `user@bash:`

## Пример 4. Выводим все файлы с расширением из 3 букв

Теперь выведем все файлы с расширением из трех букв. Обратите внимание, что `video.mpeg` не подходит, поскольку имя пути должно точно соответствовать

1. `user@bash: ls *.???`
2. `barry.txt blah.txt example.png frog.png`





## Символ подстановки []

И, наконец, символ подстановки **[]** (квадратные скобки) – для диапазона. В отличие от двух предыдущих символов подстановки, которые позволяли задать несколько символов, [] позволяет задать набор символов.

Давайте выведем все файлы, имя которых начинается с s или v.

Пример 5. Выводим все файлы, имя которых начинается с s или v

```
1. user@bash: ls [sv]*
2. secondfile video.mpeg
3. user@bash:
```

В диапазон можно включить готовые наборы символов, используя дефис. Например, если мы хотим вывести каждый файл, имя которого включает в себя цифру, можно использовать диапазон **[0-9]**.

Пример 6. Выводим все файлы, в имени которых есть хотя бы 1 цифра

```
1. user@bash: ls *[0-9]*
2. foo1 foo2 foo3
3. user@bash:
```

## Инвертированный диапазон

Мы также можем использовать инвертированный диапазон, используя знак **^** (каретка). Он позволяет искать символы, которые не входят в заданный диапазон.

```
1. user@bash: ls [^a-k]*
2. secondfile thirdfile video.mpeg
3. user@bash:
```

Примеры из реальной жизни





повсюду, поэтому вот несколько примеров, которые покажут вам, что еще можно делать с шаблонами.

Помните, что это лишь небольшие кейсы, в которых могут пригодиться wildcards. На самом деле, шаблоны удобно использовать в практически любой ситуации.

### Пример 8. Выводим тип файла для каждого файла или папки в каталоге

```
1. user@bash: file /home/karpaff/*
2. bin: directory
3. Documents: directory
4. frog.png: PNG image data
5. public_html: directory
6. user@bash:
```

### Пример 9. Перемещаем все файлы типа jpg или png в другой каталог.

```
1. user@bash: mv public_html/*.??g public_html/images/
2. user@bash:
```

### Пример 10. Узнаем размер и время изменения файла .bash\_history в домашнем каталоге пользователя

*.bash\_history* – это файл в домашнем каталоге пользователя, который хранит историю команд, введенных пользователем в командной строке. Это скрытый файл.

Как вы можете видеть в этом примере, мы можем использовать подстановочные знаки в любой точке пути.

```
1. user@bash: ls -lh /home/*.bash_history
2. -rw----- 1 harry users 2.7K Jan 4 07:32 /home/karp
3. -rw----- 1 ryanusers 3.1K Jun 12 21:16 /home/andre
4. user@bash:
```

Что нужно запомнить





Символы подстановки могут использоваться в любой части пути.

### Можно использовать везде, где есть путь

Поскольку подстановка символов выполняется системой, а не командой, шаблоны подстановки могут использоваться везде, где используется путь.

## Практические задания

Хороший каталог, на котором можно попрактиковаться – это `/etc`, который содержит файлы конфигурации системы. Как обычный пользователь, вы можете просматривать файлы, но не можете вносить какие-либо изменения, поэтому случайно повредить данный каталог не получится.

Выведите содержимое каталога и посмотрите, что там находится. Затем выберите несколько файлов и подумайте, можно ли создать шаблон для выбора только этих файлов.

1. Выведите список файлов с определенным расширением.
2. Выведите файлы, расширение которых состоит только из трех букв.
3. Выведите файлы, имя которых содержит заглавную букву (подсказка: используйте конструкцию `[: upper:]`)
4. Выведите файлы, имя которых состоит из четырех символов.



[CodeChick.io](https://codechick.io) - простой и  
эффективный способ  
изучения  
программирования.  
2024 © 000 "Алгоритмы и  
практика"

