

Расширение имени пути (глобализация)

Общая информация

В отличие от других платформ, которые вы, возможно, видели, в UNIX® оболочка отвечает за интерпретацию и расширение глобусов ("подстановочных знаков имени файла"). Вызываемая программа никогда не увидит сам глобус; она будет видеть только расширенные имена файлов в качестве своих аргументов (здесь все совпадающие имена файлов) `*.log` :

```
grep "изменения:" *.log
```

Базовым синтаксисом для расширения имени пути является синтаксис сопоставления с образцом. Шаблон, который вы описываете, сопоставляется со всеми существующими именами файлов, и соответствующие имена заменяются. Поскольку эта замена происходит **после разделения слов**, все результирующие имена файлов являются буквальными и обрабатываются как отдельные слова, независимо от того, сколько пробелов или других IFS символов они содержат.

Нормальное поведение

- с помощью команды `set (-f , noglob)` вы можете полностью отключить расширение имени пути
- при сопоставлении имени пути символ косой черты (`/`) всегда должен совпадать явно
- точка в начале имени файла должна быть сопоставлена явно (также точка, следующая `/` за `a` в глобусе)
- глобус, который не соответствует имени файла, не изменяется и остается тем, чем он является

Настройка

- когда установлен параметр оболочки `nullglob`, несоответствующие глобусы удаляются, а не сохраняются
- когда установлен параметр оболочки `failglob`, несоответствующие глобусы выдают сообщение об ошибке, и текущая команда не выполняется

- когда установлен параметр оболочки `nocaseglob`, сопоставление выполняется без учета регистра
- когда установлен параметр оболочки `dotglob`, подстановочные знаки могут совпадать с точкой в начале имени файла
- когда установлен параметр оболочки `dirspell`, Bash выполняет исправления орфографии при сопоставлении имен каталогов
- когда установлен параметр оболочки `globstar`, `**` глобус будет рекурсивно сопоставлять все файлы и каталоги. Этот глобус не является "настраиваемым", т. Е. Вы не **можете** сделать что-то вроде `**.` с рекурсивного получения всех `.*` с имен файлов.
- когда установлен параметр оболочки `globasciiranges`, глобусы с диапазоном скобок (например `[A-Z]`) используют порядок локали C, а не порядок настроенной локали (т. Е. `ABC...abc...` Вместо, Например `AaBbCc...`), - начиная с 4.3-alpha
- переменная `GLOBIGNORE` может быть установлена в список шаблонов, разделенных двоеточием, которые необходимо удалить из списка, прежде чем он будет возвращен

nullglob

Обычно, когда ни один из указанных глобусов не соответствует существующему имени файла, расширение имени пути не выполняется, и глобусы **не** удаляются:

```
$ echo "Текстовые файлы здесь:" *.txt
Текстовые файлы здесь: *.txt
```

В этом примере ни один файл не соответствовал шаблону, поэтому глобус остался нетронутым (буквенная звездочка, за которой следует точка-txt).

Это может быть очень раздражающим, например, когда вы запускаете цикл `for`, используя расширение имени пути:

```
для имени файла в *.txt; выполните
echo "=== BEGIN: $filename ==="
cat "$filename"
echo "=== END: $filename ==="
Готово
```

Когда имя файла не совпадает с глобусом, цикл не только выведет глупый текст (`" BEGIN: *.txt "`), но также приведет к `cat` сбою команды -с ошибкой, поскольку файл с именем `*.txt` не существует.

Теперь, когда установлен параметр оболочки, Bash удалит весь глобус из командной строки. `nullglob` В случае цикла `for` здесь не будет выполнено даже одной итерации. Он просто не будет запускаться.

Итак, в нашем первом примере:

```
$ shopt -s nullglob
$ echo "Текстовые файлы здесь:" *.txt
Текстовые файлы здесь:
```

и глобус исчез.

Символы глобуса

- `*` - означает "сопоставление любого количества символов". `/` не соответствует (и в зависимости от ваших настроек могут совпадать или не совпадать такие вещи, как `.`, см. Выше)
- `?` - означает "сопоставить любой отдельный символ"
- `[abc]` - соответствует любому из перечисленных символов. Этот синтаксис также поддерживает диапазоны, такие как `[0-9]`

Например, для сопоставления чего-либо, начинающегося с `'S'` или `'K'`, за которым следуют две цифры, за которыми следуют по крайней мере еще 3 символа:

```
[SK][0-9][0-9]???
```

Смотрите также

- Введение в расширение и замену
- синтаксис сопоставления с образцом
- встроенная команда `set`
- встроенная команда `short`
- список параметров оболочки

Обсуждение

Альтаир IV, [2011/12/27 16:12 \(\)](#), [2011/12/28 00:55 \(\)](#)

*globstar не "настраивается", т. Е. Вы не можете сделать что-то вроде `**.`, чтобы рекурсивно получить все имена файлов `.*`.*

Сам `globstar` не настраивается, но вы можете использовать его в сочетании с другими глобусами для этого.

```
echo ./**/*.c
```

Этот сайт поддерживается Performing Databases - вашими
экспертами по администрированию баз данных

Bash Hackers Wiki



Except where otherwise noted, content on this wiki is licensed under the following license:
GNU Free Documentation License 1.3