

tr(1)

НАЗВАНИЕ

tr - преобразование символов

СИНТАКСИС

```
/usr/bin/tr [ -cs ] строка1 строка2
/usr/bin/tr -s | -d [ -c ] строка1
/usr/bin/tr -ds [ -c ] строка1 строка2
/usr/xpg4/bin/tr [ -cs ] строка1 строка2
/usr/xpg4/bin/tr -s | -d [ -c ] строка1
/usr/bin/xpg4/tr -ds [ -c ] строка1 строка2
```

ОПИСАНИЕ

Утилита **tr** копирует стандартный входной поток в стандартный выходной, подставляя или удаляя некоторые символы. Заданные опции и операнды **строка1** и **строка2** управляют происходящим преобразованием по ходу копирования отдельных символов и последовательностей символов.

ОПЦИИ

Поддерживаются следующие опции:

- | | |
|-----------|--|
| -c | Дополняет набор символов, задаваемый строкой 1 . |
| -d | Удаляет все вхождения символов, указанных в строке1 . |
| -s | Заменяет повторяющиеся вхождения символа одним символом. |

Когда опция **-d** не указана:

- Каждый входной символ, найденный в массиве, задаваемом **строкой1**, заменяется символом в той же относительной позиции из массива, задаваемого **строкой2**. Когда массив, задаваемый **строкой2**, короче, чем задаваемый **строкой1**, результат не определен.
- Если указана опция **-c**, символы, дополняющие указанные в **строке1** (набор всех символов текущей кодовой страницы, определяемый по текущему значению переменной среды **LC_STYPE**, за исключением явно указанных в операнде **string1**) помещаются в массив в порядке возрастания в соответствии с последовательностью сортировки, задаваемой текущим значением переменной среды **LC_COLLATE**.
- Поскольку порядок, в котором идут символы, задаваемые классом символов или классом эквивалентности, не определен, такие выражения должны использоваться только если предполагается сопоставить несколько символов одному. Исключение - преобразование регистра символов, как будет описано далее.

Когда опция **-d** указана:

- Входные символы, находящиеся в массиве, задаваемом **строкой1**, удаляются.
- Если вместе с опцией **-d** указана опция **-c**, будут удалены все символы, кроме указанных в **строке1**. **Строка2** при этом игнорируется, если только не указана также опция **-s**.
- Одну и ту же строку нельзя использовать для опций **-d** и **-s**; если заданы обе опции, надо задавать две строки, **строку1** (используется для удаления) и **строку2** (используется для сжатия).

Когда задана опция **-s**, после любых удалений или преобразований последовательные многократные вхождения символа заменяются одним вхождением, если символ находится в массиве, заданном последним операндом. Если последний операнд содержит класс символов, как в следующем примере

```
tr -s '[:space:]'
```

то массив символов последнего операнда будет содержать все символы соответствующего класса. Однако в случае преобразования регистра символов, например

```
tr -s '[:upper:]' '[:lower:]'
```

массив символов последнего операнда будет содержать только символы, являющиеся парными при соответствующем преобразовании регистра. (См. **toupper(3C)** и **tolower(3C)**).

Если **строка1** или **строка2** - пустая, результат не определен.

ОПЕРАНДЫ

Поддерживаются следующие операнды:

строка1
строка2

Строки, управляющие преобразованием. Каждая строка представляет набор символов, по которому строится массив, используемый при преобразовании.

Операнды **строка1** и **строка2** (если заданы) определяют два массива символов. Для задания символов или односимвольных элементов можно использовать конструкции из следующего списка. Если любая из конструкций породит многосимвольные элементы, утилита **tr** без предупреждения исключит эти многосимвольные элементы из результирующего массива.

символ

Любой символ, не относящийся к описанным далее, представляет сам себя.

\восьмеричное_число

Восьмеричные значения можно использовать для представления символов с соответствующими кодами. Для этого необходимо ввести обратную косую и одно-, двух- или трехзначную последовательность восьмеричных цифр (**01234567**). В результате, в массив подставляется символ с соответствующим восьмеричным кодом. Многобайтовые символы требуют ввода нескольких управляющих последовательностей этого типа подряд, включая начальный символ **** для каждого байта.

\символ

Поддерживаются управляющие последовательности, начинающиеся с обратной косой, \a, \b, \f, \n, \r, \t и \v. Результат использования после обратной косой любого другого символа, кроме восьмеричной цифры, не определен.

/usr/xpg4/bin/tr

c-c

/usr/bin/tr

[c-c]

Представляет диапазон символов между указанными, включительно, из последовательности сортировки, определяемой переменной среды **LC_COLLATE**. Начальный символ должен предшествовать завершающему в текущей последовательности сортировки. Символы в указанном диапазоне помещаются в массив отсортированными в порядке возрастания.

[:class:]

Представляет все символы, принадлежащие к заданному классу символов в соответствии с текущим значением переменной среды **LC_CTYPE**. При задании **строки1** можно использовать следующие классы символов:

alnum blank digit lower punct upper
alpha cntrl graph print space xdigit

Кроме того, распознаются классы символов вида **[:имя:]** в тех локалях, где с ключевым словом **имя** связано определение класса символов в категории **LC_CTYPE**.

Когда указаны обе опции, **-d** и **-s**, в **строке2** можно указывать любое имя класса символов. В противном случае, в **строке2** можно задавать только класс **lower** или **upper**, причем только если соответствующий класс символов, **upper** или **lower**, указан в той же относительной позиции в **строке1**. Такая спецификация интерпретируется как запрос преобразования регистра символов. Когда в **строке1** встречается класс **[:lower:]**, а в **строке2** - **[:upper:]toupper** в категории **LC_CTYPE** текущей локали. Когда в **строке1** указан класс **[:upper:]**, а в **строке2** указан класс **[:lower:]**, массивы будут содержать символы из соответствия **tolower** в категории **LC_CTYPE** текущей локали. Первый символ из каждой пары соответствия будет в массиве для **строки1**, а второй - в массиве для **строки2** в той же относительной позиции.

За исключением преобразования регистра символов, задаваемые при раскрытии класса символы помещаются в массив в непредсказуемом порядке.

Если указанное в качестве имени класса имя не соответствует одному из допустимых в соответствии с текущей локалью классов символов, поведение утилиты не определено.

[=equiv=]

Представляет все символы, принадлежащие к тому же классу эквивалентности, что и **equiv**, в соответствии с текущим значением переменной среды **LC_COLLATE**. Указывать класс эквивалентности можно только в **строке1**, или в **строке2**, когда она используется с опциями **-d** и **-s** одновременно. Символы, принадлежащие к указанному классу эквивалентности, помещаются в массив в непредсказуемом порядке.

[x*n]

Представляет **n** вхождений символа **x** подряд. Поскольку это выражение используется для сопоставления многих символов с одним, допускается его включение только в **строку2**. Если **n** не указано или равно **0**, это интерпретируется как "достаточной длины для расширения массива на основе **строки2** до длины массива, построенного на основе **строки1**". Если значение **n** начинается с **0**, значение интерпретируется как восьмеричное. В противном случае, оно интерпретируется как десятичное.

ИСПОЛЬЗОВАНИЕ

Описание особенностей работы утилиты **tr** с файлами размером более 2 Гбайтов (2**31 байтов) см. на странице справочного руководства **largefile(5)**.

ПРИМЕРЫ

Пример 1: Создание списка слов

В следующем примере создается список всех слов в файле **file1**, который помещается, по одному слову в строке, в файл **file2**. При этом совом считается максимально длинная последовательность букв.

```
tr -cs "[:alpha:]" "\n*" <file1 >file2
```

Пример 2: Преобразование символов

В этом примере все символы нижнего регистра в файле **file1** переводятся в верхний регистр и результат записывается в стандартный выходной поток.

```
tr "[:lower:]" "[:upper:]" <file1
```

Обратите внимание, что нюанс, выявляющийся в соответствующем примере в XPG3, больше не проявляется. Такое преобразование регистра теперь рассматривается как специальный случай, использующий классификацию символов функций **tolower** и **toupper**, гарантирующую правильное сопоставление (если правильно задана локаль).

Пример 3: Идентификация эквивалентных символов

В этом примере используется класс эквивалентности для выявления акцентированных вариантов базового символа **e** в файле **file1**, из которых удаляются диактрические знаки, и результат записывается в файл **file2**.

```
tr "[=e]" e <file1 >file2
```

ПЕРЕМЕННЫЕ СРЕДЫ

Описание следующих переменных среды: **LC_COLLATE**, **LC_CTYPE**, **LC_MESSAGES** и **NLSPATH**, влияющих на работу команды **tr**, см. на странице справочного руководства **environ(5)**.

СТАТУС ВЫХОДА

Команда завершается со следующими статусами выхода:

0	Все входные данные успешно обработаны.
>0	Произошла ошибка.

АТРИБУТЫ

Описание следующих атрибутов см. на странице справочного руководства [attributes\(5\)](#):

/usr/bin/tr

ТИП АТРИБУТА	ЗНАЧЕНИЕ АТРИБУТА
Доступен в пакете	SUNWcsu
CSI	Включено

/usr/xpg4/bin/tr

ТИП АТРИБУТА	ЗНАЧЕНИЕ АТРИБУТА
Доступен в пакете	SUNWxcu4
CSI	Включено

ССЫЛКИ

[ed\(1\)](#), [sed\(1\)](#), [sh\(1\)](#), [tolower\(3C\)](#), [toupper\(3C\)](#), [ascii\(5\)](#), [attributes\(5\)](#), [environ\(5\)](#), [largefile\(5\)](#), [XPG4\(5\)](#)

ПРИМЕЧАНИЯ

В отличие от некоторых прежних версий, программа **/usr/xpg4/bin/tr** корректно обрабатывает символы **NUL** во входном потоке. Символы **NUL** можно удалить с помощью команды **tr -d '\000'**.

Последнее изменение: 18 марта 1997 года

Copyright © 2002 В. Кравчук, OpenXS Initiative, перевод на русский язык