

8 Linux TR Command Examples

by LAKSHMANAN GANAPATHY on DECEMBER 19, 2012

`tr` is an UNIX utility for translating, or deleting, or squeezing repeated characters. It will read from STDIN and write to STDOUT.

`tr` stands for translate.

Syntax

The syntax of `tr` command is:

```
$ tr [OPTION] SET1 [SET2]
```

Translation

If both the `SET1` and `SET2` are specified and `-d` `OPTION` is not specified, then `tr` command will replace each characters in `SET1` with each character in same position in `SET2`.

1. Convert lower case to upper case

The following `tr` command is used to convert the lower case to upper case

```
$ tr abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ  
thegeekstuff  
THEGEEKSTUFF
```

The following command will also convert lower case to upper case

```
$ tr [:lower:] [:upper:]  
thegeekstuff  
THEGEEKSTUFF
```

You can also use ranges in `tr`. The following command uses ranges to convert lower to upper case.

```
$ tr a-z A-Z  
thegeekstuff  
THEGEEKSTUFF
```

2. Translate braces into parenthesis

You can also translate from and to a file. In this example we will translate braces in a file with parenthesis.

```
$ tr '{}' '()' < inputfile > outputfile
```

The above command will read each character from “inputfile”, translate if it is a brace, and write the output in “outputfile”.

3. Translate white-space to tabs

The following command will translate all the white-space to tabs

```
$ echo "This is for testing" | tr [:space:] '\t'  
This    is    for testing
```

4. Squeeze repetition of characters using -s

In Example 3, we see how to translate space with tabs. But if there are two or more spaces present continuously, then the previous command will translate each space to a tab as follows.

```
$ echo "This  is  for testing" | tr [:space:] '\t'  
This      is      for testing
```

We can use `-s` option to squeeze the repetition of characters.

```
$ echo "This  is  for testing" | tr -s [:space:] '\t'  
This    is  for testing
```

Similarly you can convert multiple continuous spaces with a single space

```
$ echo "This  is  for testing" | tr -s [:space:] ' '  
This is for testing
```

5. Delete specified characters using `-d` option

`tr` can also be used to remove particular characters using `-d` option.

```
$ echo "the geek stuff" | tr -d 't'  
he geek suff
```

To remove all the digits from the string, use

```
$ echo "my username is 432234" | tr -d [:digit:]  
my username is
```

Also, if you like to delete lines from file, you can use [sed d command](#).

6. Complement the sets using `-c` option

You can complement the SET1 using `-c` option. For example, to remove all characters except digits, you can use the following.

```
$ echo "my username is 432234" | tr -cd [:digit:]  
432234
```

7. Remove all non-printable character from a file

The following command can be used to remove all non-printable characters from a file.

```
$ tr -cd [:print:] < file.txt
```

8. Join all the lines in a file into a single line

The below command will translate all newlines into spaces and make the result as a single line.

```
$ tr -s '\n' ' ' < file.txt
```