



Linuxopsys

@linuxopsys

29 Tweets • 2023-02-03 • [See on Twitter](#)

rattibha.com

10 Linux tr command practical examples you should know as a system administrator (bookmark this):

The tr command short for translate, is one of the most useful command for manipulating text on the command line.

It allows you to perform useful operations such as converting lowercase characters to uppercase characters, uppercase characters to lowercase characters, character replacing, and deleting characters.

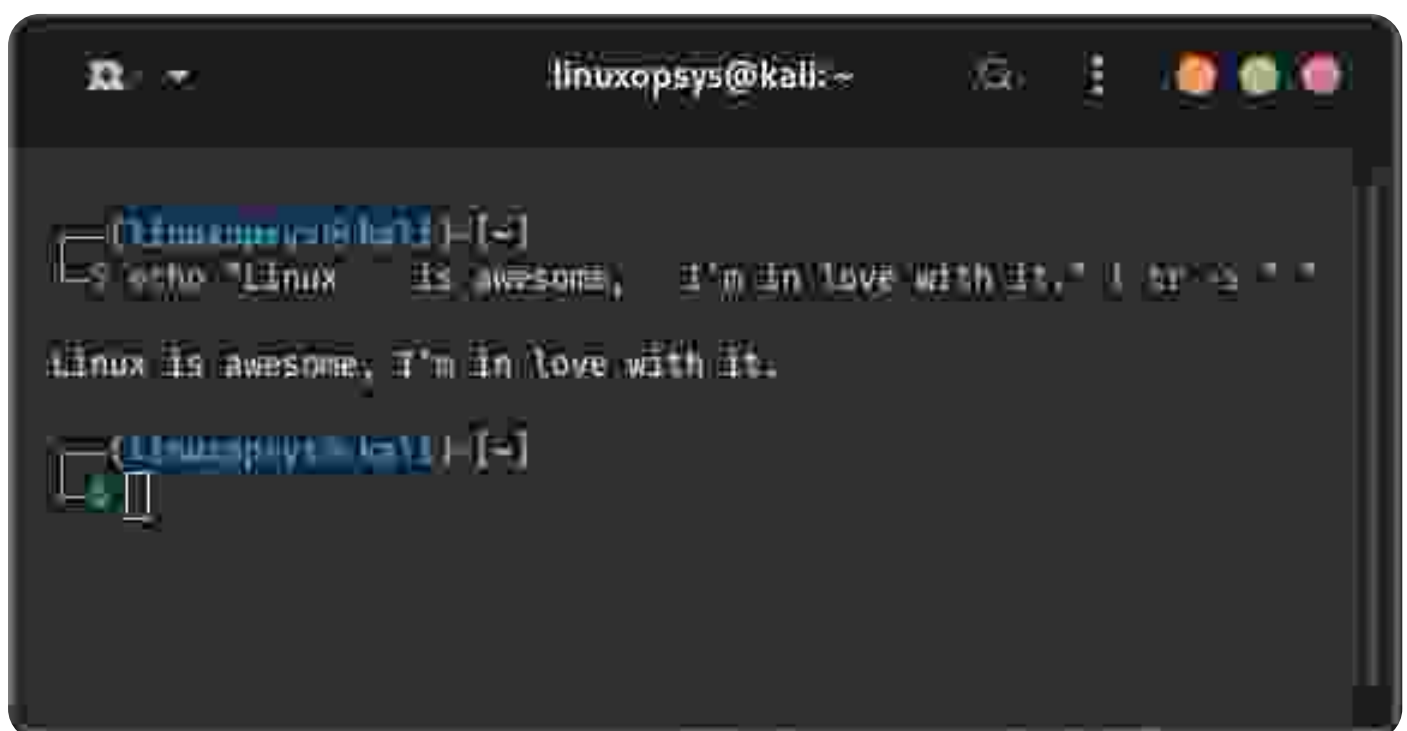
It is usually used in conjunction with other commands via piping.

In this thread, I will show you some of the most examples of the `tr` command on Linux.

1. Deleting repeated characters

You can use the `-s` option to squeeze a character that is repeating to make it a single character. This option is especially useful when you want to squeeze multiple continuous spaces characters into a single character.

```
$ echo "Linux   is awesome,  I'm in love with it." | tr -s  
" "
```

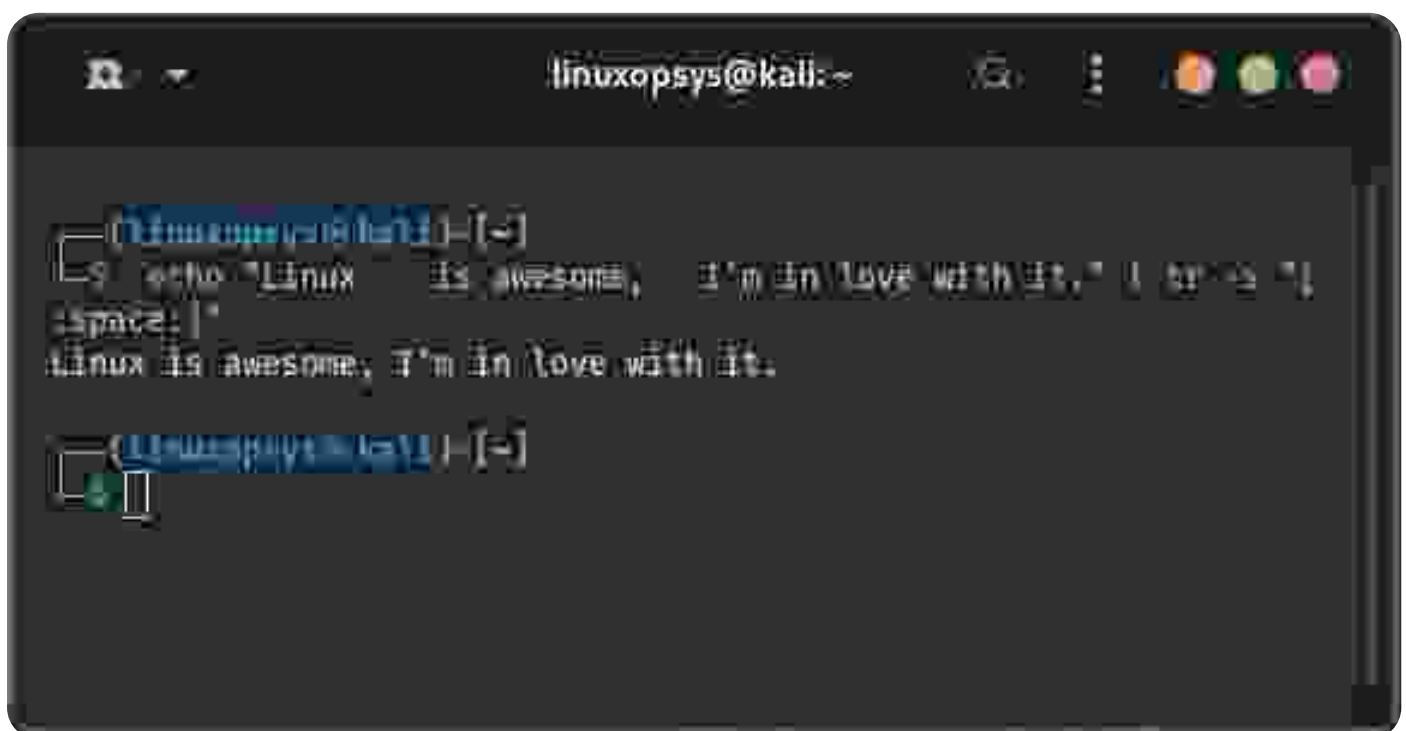
A terminal window with a dark background and light-colored text. The window title bar shows 'linuxopsys@kali:~'. The terminal content shows a command being entered and executed. The command is 'echo "Linux is awesome, I'm in love with it." | tr -s " "'. The output of the command is 'linux is awesome, I'm in love with it.', where the multiple spaces between 'Linux' and 'is' have been reduced to a single space.

```
linuxopsys@kali:~  
$ echo "Linux   is awesome,  I'm in love with it." | tr -s  
" "  
linux is awesome, I'm in love with it.  
$
```

From the above example you can clearly see the multiple spaces have been translated to single space.

Instead of using the space character in our character SET for squeezing repeating characters, we can also make use of character classes and still get the same results.

```
$ echo "Linux   is awesome,  I'm in love with it." | tr -s  
"[:space:]"
```

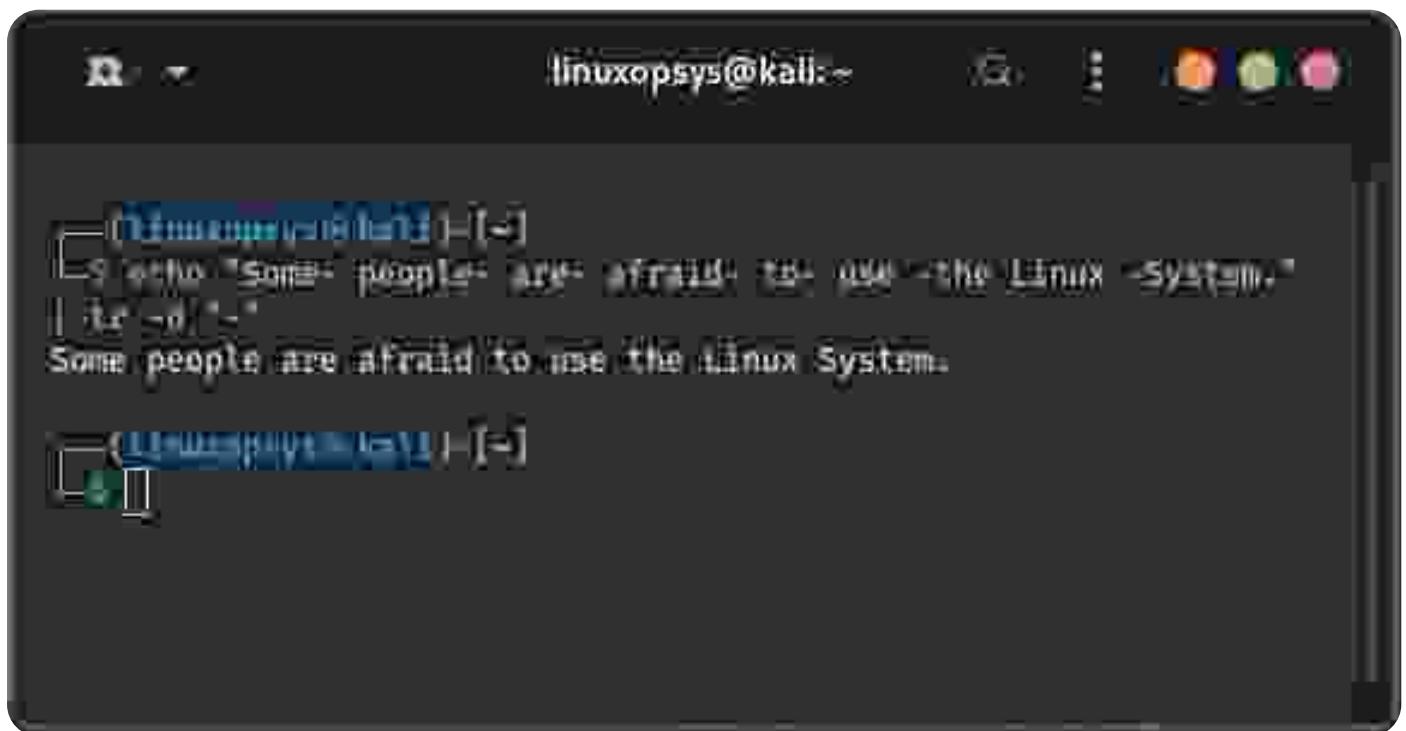
A terminal window with a dark background and light-colored text. The window title is 'linuxopsys@kali:~'. The prompt is a root symbol. The command entered is 'echo "Linux is awesome, I'm in love with it." | tr -s "[:space:]"'. The output is 'linux is awesome, I'm in love with it.'.

```
linuxopsys@kali:~  
$ echo "Linux   is awesome,  I'm in love with it." | tr -s  
"[:space:]"  
linux is awesome, I'm in love with it.  
$
```

2. Delete specific characters

Using the -d option, you can delete characters you specify. tr command deletes every instance of the "-" character in the following example:

```
$ echo "Some- people- are- afraid- to- use -the Linux -  
System." | tr -d "-"
```

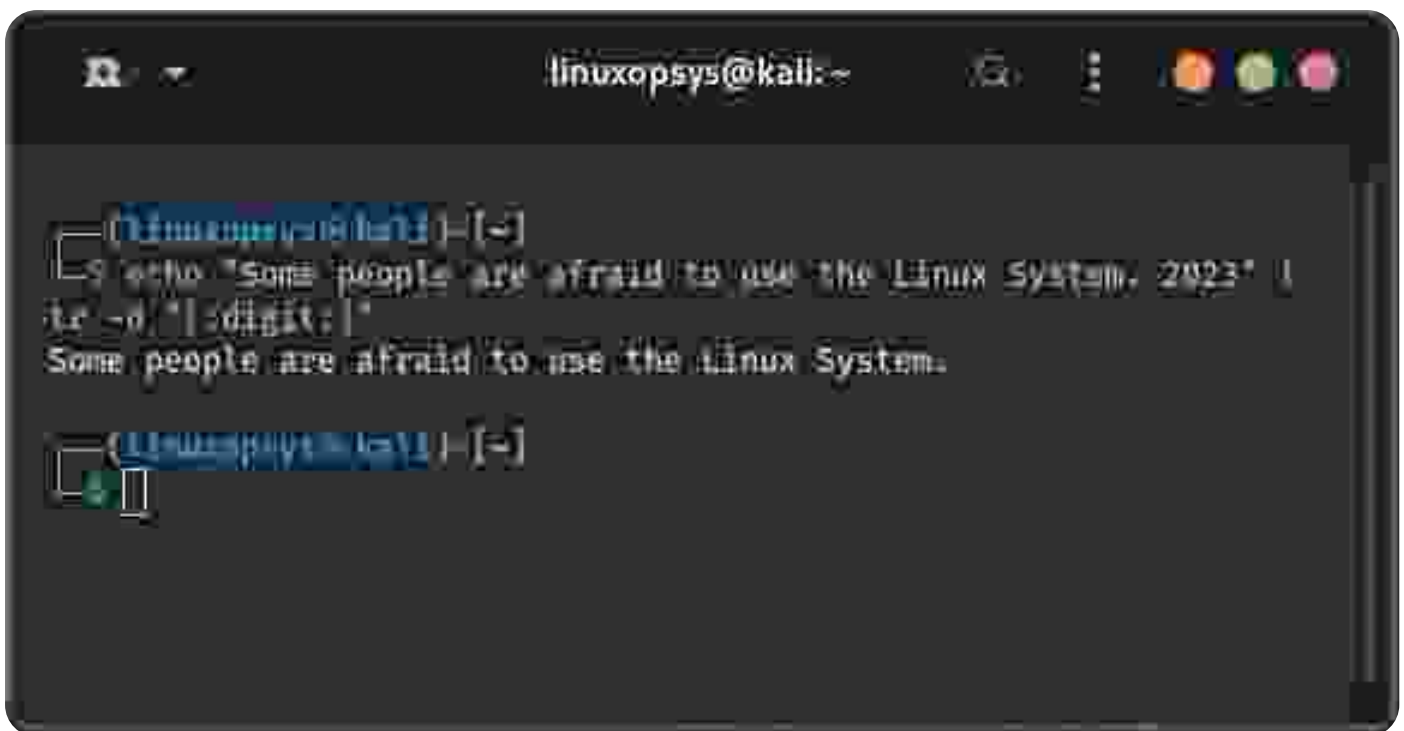
A terminal window with a dark background and light-colored text. The window title bar shows 'linuxopsys@kali:~' and standard window control buttons. The terminal shows a command prompt followed by the command 'echo "Some- people- are- afraid- to- use -the Linux -System." | tr -d "-"'. The output of the command is 'Some people are afraid to use the Linux System.'.

```
linuxopsys@kali:~  
$ echo "Some- people- are- afraid- to- use -the Linux -System."  
| tr -d "-"  
Some people are afraid to use the Linux System.  
$
```

3. Delete all the digits

You can also use the `-d` option to remove any numbers or digits from your text.

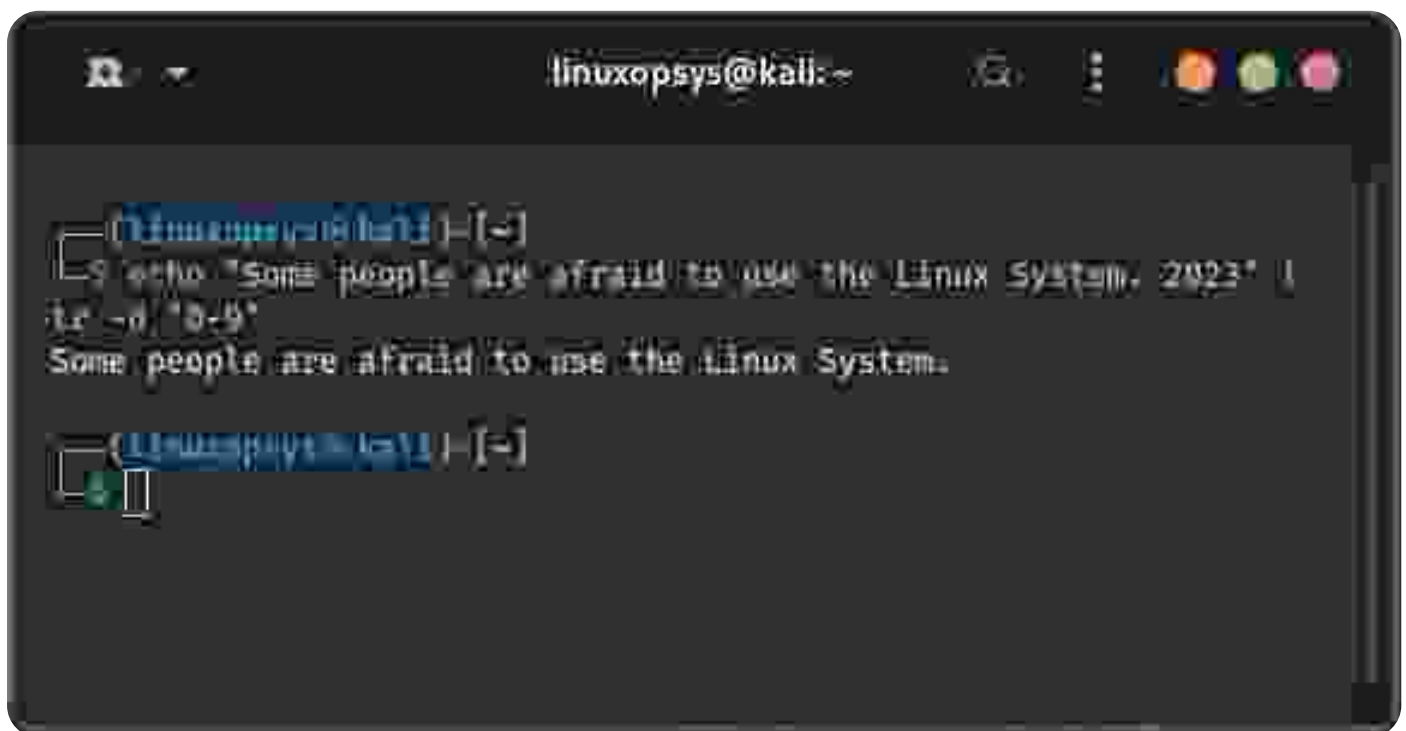
```
$ echo "Some people are afraid to use the Linux System. 2023" | tr -d "[:digit:]"
```

A terminal window with a dark background and light-colored text. The window title bar shows 'linuxopsys@kali:~'. The terminal content shows a command being entered and executed, resulting in the removal of digits from a string.

```
linuxopsys@kali:~$ echo "Some people are afraid to use the Linux System. 2023" | tr -d "[:digit:]"
Some people are afraid to use the Linux System.
```

Instead of using character classes you can use number character range and still get the same results.

```
$ echo "Some people are afraid to use the Linux System. 2023" | tr -d "0-9"
```

A screenshot of a Linux terminal window. The window title is 'linuxopsys@kali:~'. The terminal shows a command being entered: 'echo "Some people are afraid to use the Linux System. 2023" | tr -d "0-9"'. The output of the command is 'Some people are afraid to use the Linux System.'. The prompt character is a green square. The terminal has a dark background and standard window controls at the top.

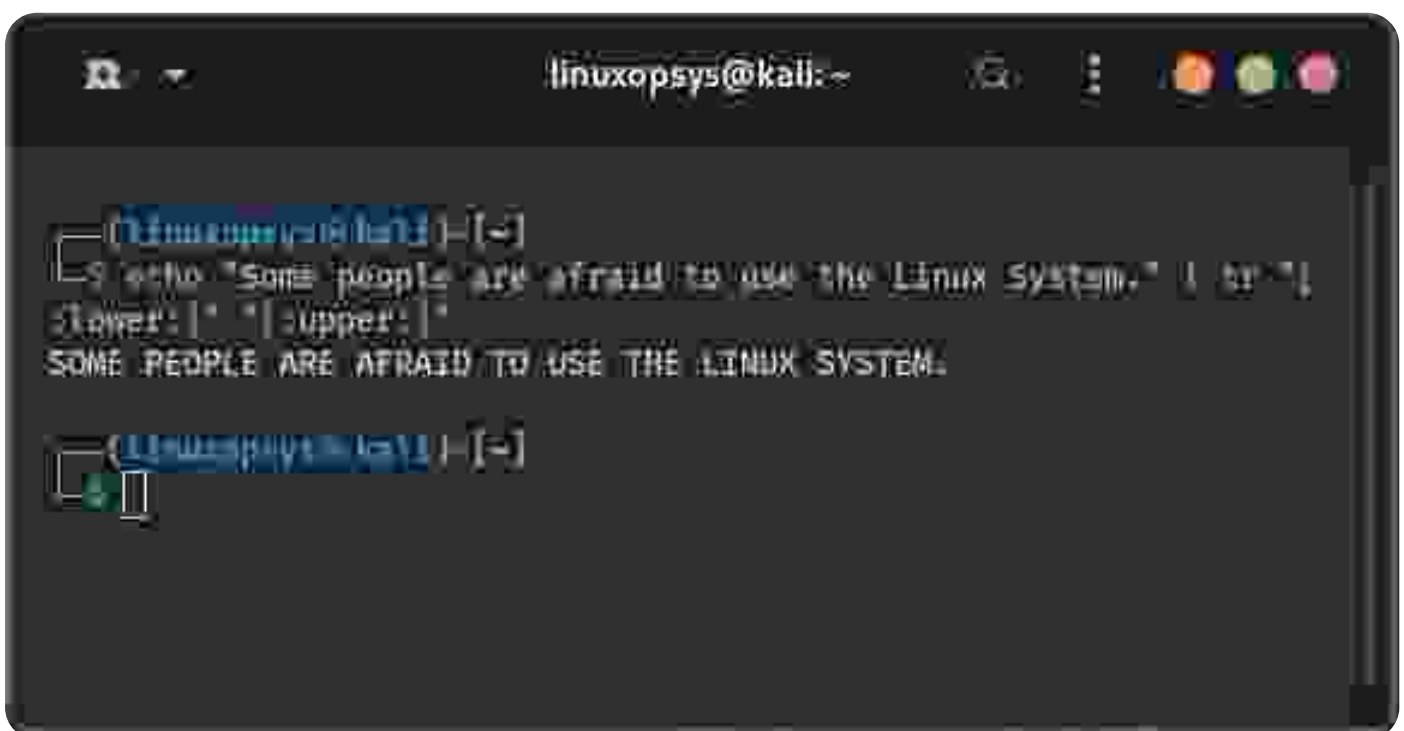
```
linuxopsys@kali:~  
$ echo "Some people are afraid to use the Linux System. 2023" | tr -d "0-9"  
Some people are afraid to use the Linux System.  
$
```

4. Case conversion

The `tr` command is frequently used to convert lowercase letters to uppercase letters or the opposite.. The character class `[:lower:]` matches all lowercase characters, while the character class `[:upper:]` matches all uppercase characters.

The following will transform characters from upper case to lower case.

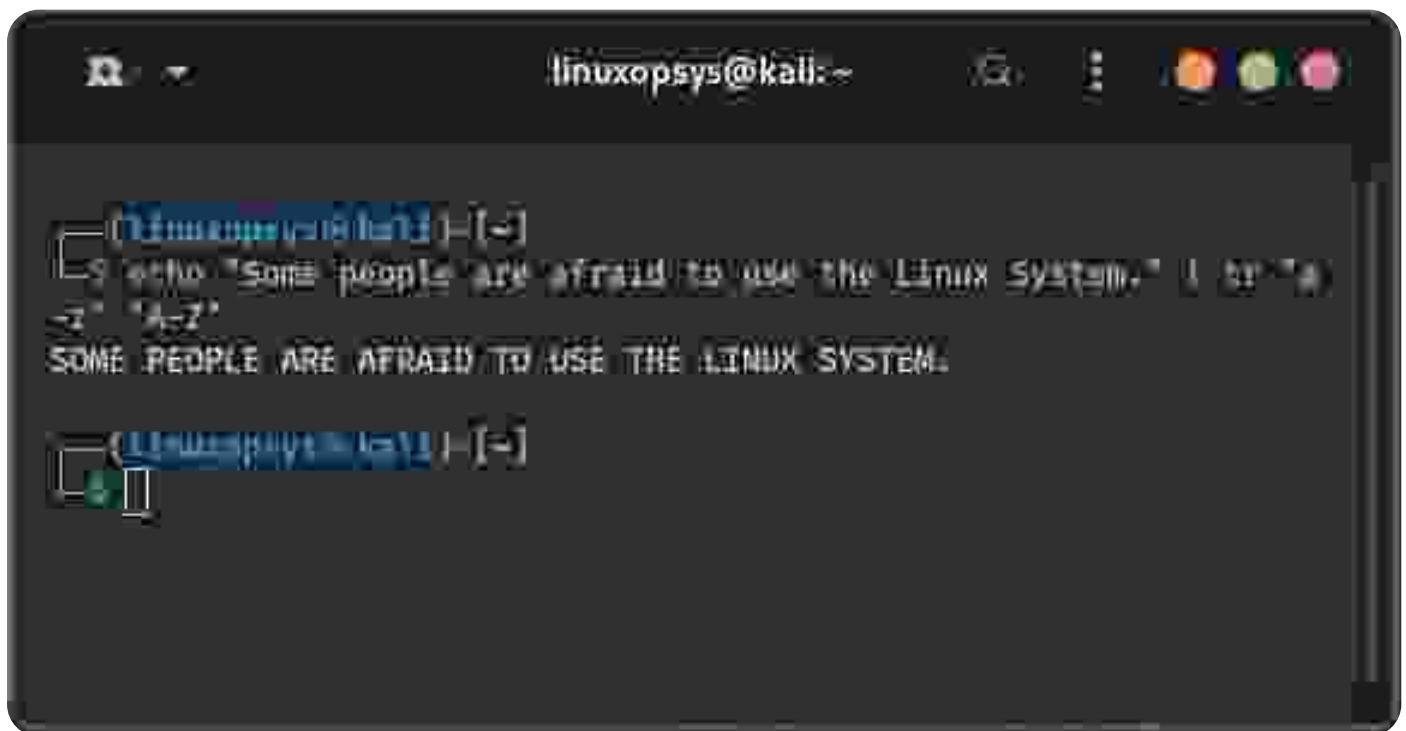
```
$ echo "Some people are afraid to use the Linux System." | tr "[:lower:]" "[:upper:]"
```

A terminal window with a dark background and light-colored text. The window title bar shows the username 'linuxopsys@kali' and some window control icons. The terminal content shows a command being entered and executed. The command is `$ echo "Some people are afraid to use the Linux System." | tr "[:lower:]" "[:upper:]"`. The output of the command is `SOME PEOPLE ARE AFRAID TO USE THE LINUX SYSTEM.` displayed on the next line. The prompt character is a green square.

```
linuxopsys@kali:~$ echo "Some people are afraid to use the Linux System." | tr "[:lower:]" "[:upper:]"  
SOME PEOPLE ARE AFRAID TO USE THE LINUX SYSTEM.  
linuxopsys@kali:~$
```

Alternatively, you can also use character range (regular expression) in place of the character classes.

```
$ echo "Some people are afraid to use the Linux System." | tr "a-z" "A-Z"
```

A terminal window with a dark background and light-colored text. The window title bar shows 'linuxopsys@kali:~'. The terminal displays the command `echo "Some people are afraid to use the Linux System." | tr "a-z" "A-Z"` and its output `SOME PEOPLE ARE AFRAID TO USE THE LINUX SYSTEM.`.

```
linuxopsys@kali:~  
$ echo "Some people are afraid to use the Linux System." | tr "a-z" "A-Z"  
SOME PEOPLE ARE AFRAID TO USE THE LINUX SYSTEM.  
$
```

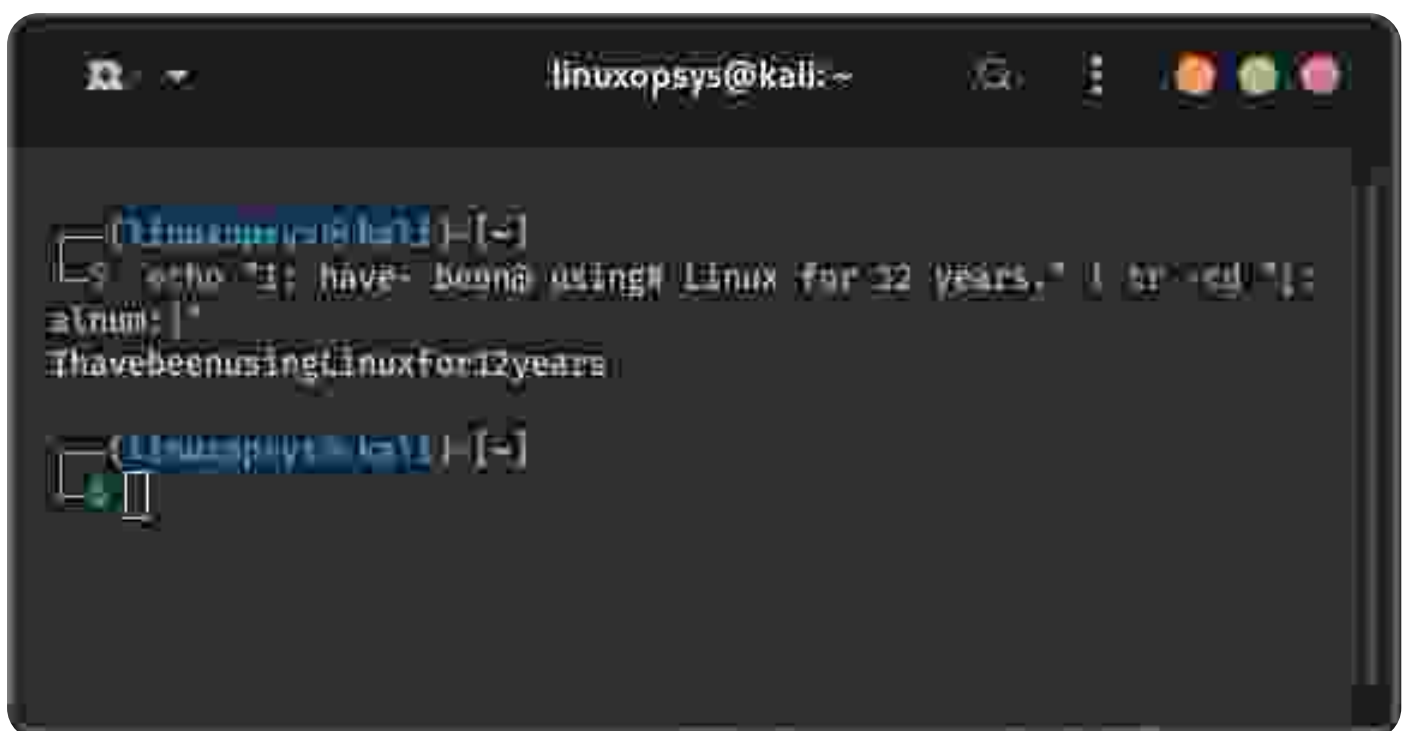

5. Removing non alphanumerical ☐

☐haracters

We can combine the complement option (-c) with the delete option (-d) to delete all non-alphanumeric characters.

The following command will delete all non-alphanumeric characters.

```
$ echo "I: have- been@ using# Linux for 12 years." | tr -cd "[:alnum:]"
```

A terminal window with a dark background. The title bar shows 'linuxopsys@kali:~'. The command 'echo "I: have- been@ using# Linux for 12 years." | tr -cd "[:alnum:]"' is entered and executed. The output 'Ihavebeenusinglinuxfor12years' is displayed on the next line. The prompt character is a green square.

```
linuxopsys@kali:~  
$ echo "I: have- been@ using# Linux for 12 years." | tr -cd "[:alnum:]"  
Ihavebeenusinglinuxfor12years  
$
```

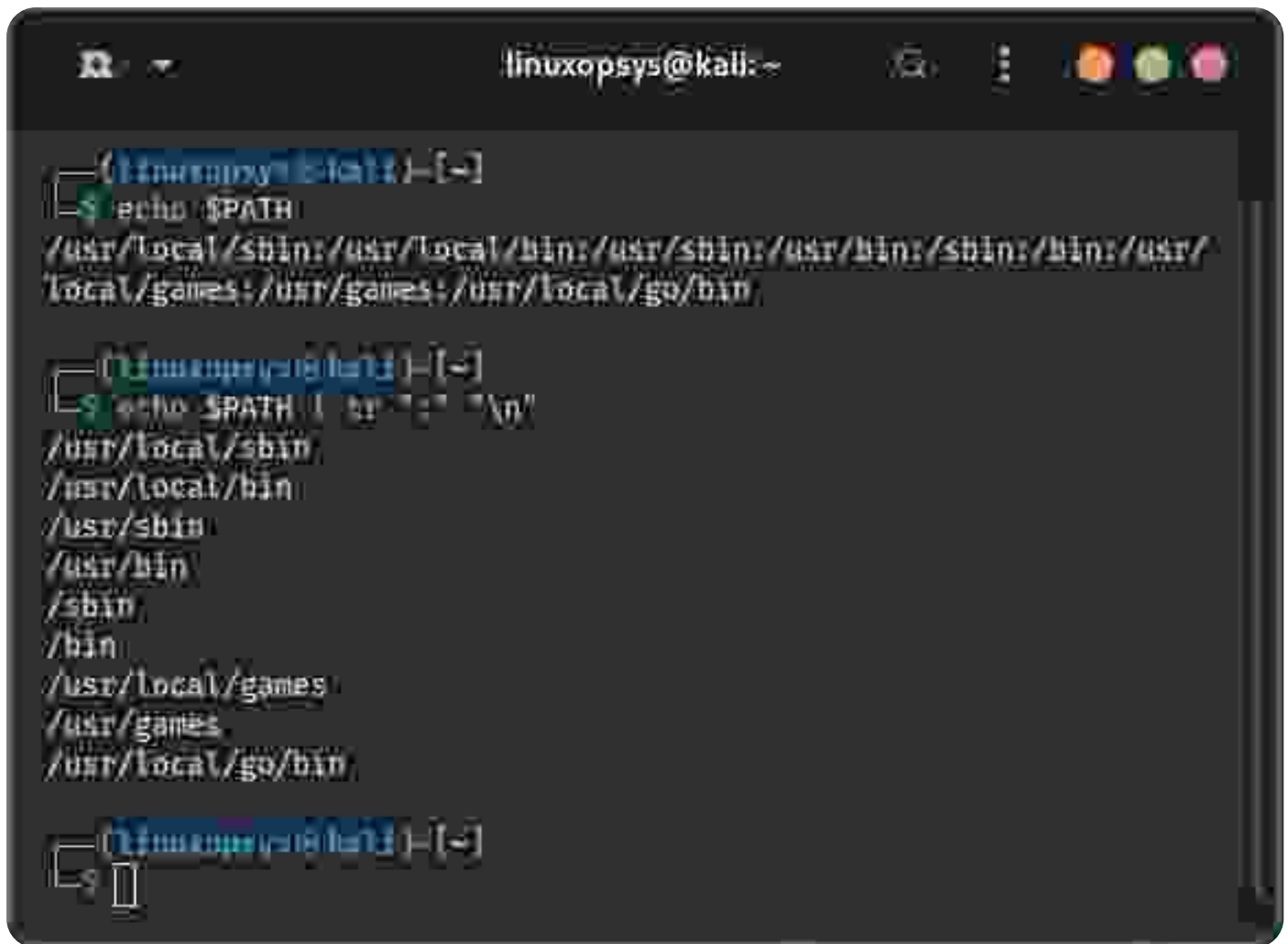
6. Print each word on a single ☐

☐ine

The \$PATH variable is an environmental variable that contains a list of directories separated by a colon that instructs the shell where to look for executable files when you type a command.

So if you're having trouble reading the directories in the \$PATH variable, you can make use the tr command to replace the colons with the newline characters so each directory is displayed on a single line.

```
$ echo $PATH | tr ":" "\n"
```

A terminal window with a dark background and light-colored text. The window title bar shows the user 'linuxopsys@kali' and the home directory '~'. The terminal shows three commands being executed. The first command is 'echo \$PATH', which outputs a long string of directory paths separated by semicolons. The second command is 'echo \$PATH | tr ":" "\n"', which outputs the same paths, but each path is on a new line, with the semicolons replaced by newline characters. The third command is a prompt character, and the cursor is visible on the line below it.

```
linuxopsys@kali:~$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/
local/games:/usr/games:/usr/local/go/bin

linuxopsys@kali:~$ echo $PATH | tr ":" "\n"
/usr/local/sbin
/usr/local/bin
/usr/sbin
/usr/bin
/sbin
/bin
/usr/local/games
/usr/games
/usr/local/go/bin

linuxopsys@kali:~$
```

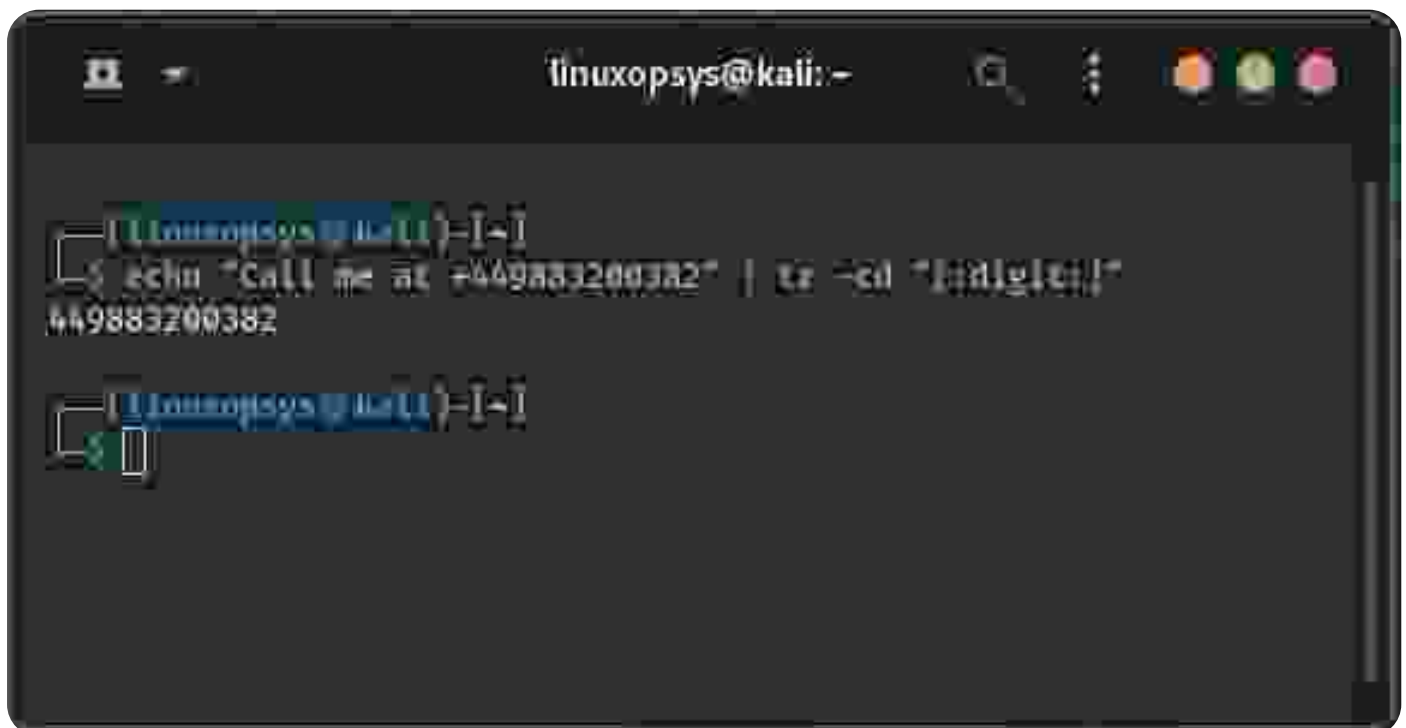
The preceding example will replace all the semicolon in our path variable with the newline characters.

7. Remove all non-numeric

characters

The `-c` option instructs `tr` to use the complement in the SET given. In this example, we want to remove all of the letters and only keep the phone number.

```
$ echo "Call me at +449883200382" | tr -cd "[:digit:]"
```

A terminal window with a dark background and light-colored text. The window title is 'linuxopsys@kali:~'. The prompt is 'linuxopsys@kali:~'. The command entered is 'echo "Call me at +449883200382" | tr -cd "[:digit:]"'. The output is '449883200382'.

```
linuxopsys@kali:~$ echo "Call me at +449883200382" | tr -cd "[:digit:]"
449883200382
linuxopsys@kali:~$
```

This is very useful if you want to extract phone numbers or employee IDs from text files.

8. Remove Newline Characters

Assume you have a text file containing data that looks like this, and you want to remove the newlines and put the words on a single line separated by spaces.

```
$ cat file.txt
```

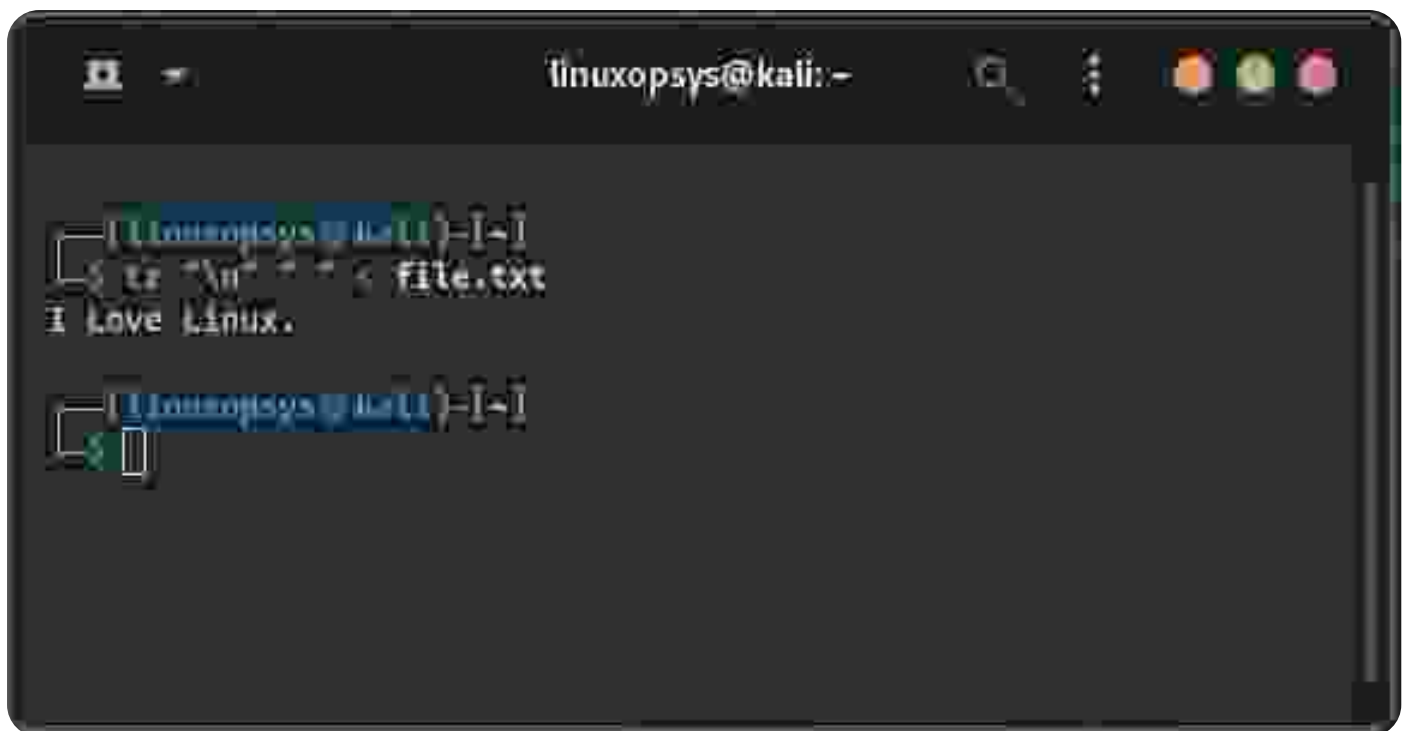
A terminal window with a dark background and light-colored text. The window title is 'linuxopsys@kali:~'. The prompt is '\$'. The command 'cat file.txt' has been entered and executed. The output is 'Love Linux,'. The terminal shows the command prompt, the command, and the output on separate lines.

```
linuxopsys@kali:~  
$ cat file.txt  
Love  
Linux,  
$
```

To achieve that, you can redirect you file contents to the tr command as shown on the below command.

```
$ tr "\n" " " < file.txt
```

This will replace each newline character in a text file with a space.

A terminal window with a dark background and light-colored text. The window title is 'linuxopsys@kali:~'. The prompt is '\$'. The command 'tr "\n" " " < file.txt' has been entered and executed. The output of the command is 'I Love Linux.' on a single line. The prompt '\$' is visible again on the next line, indicating the command has finished.

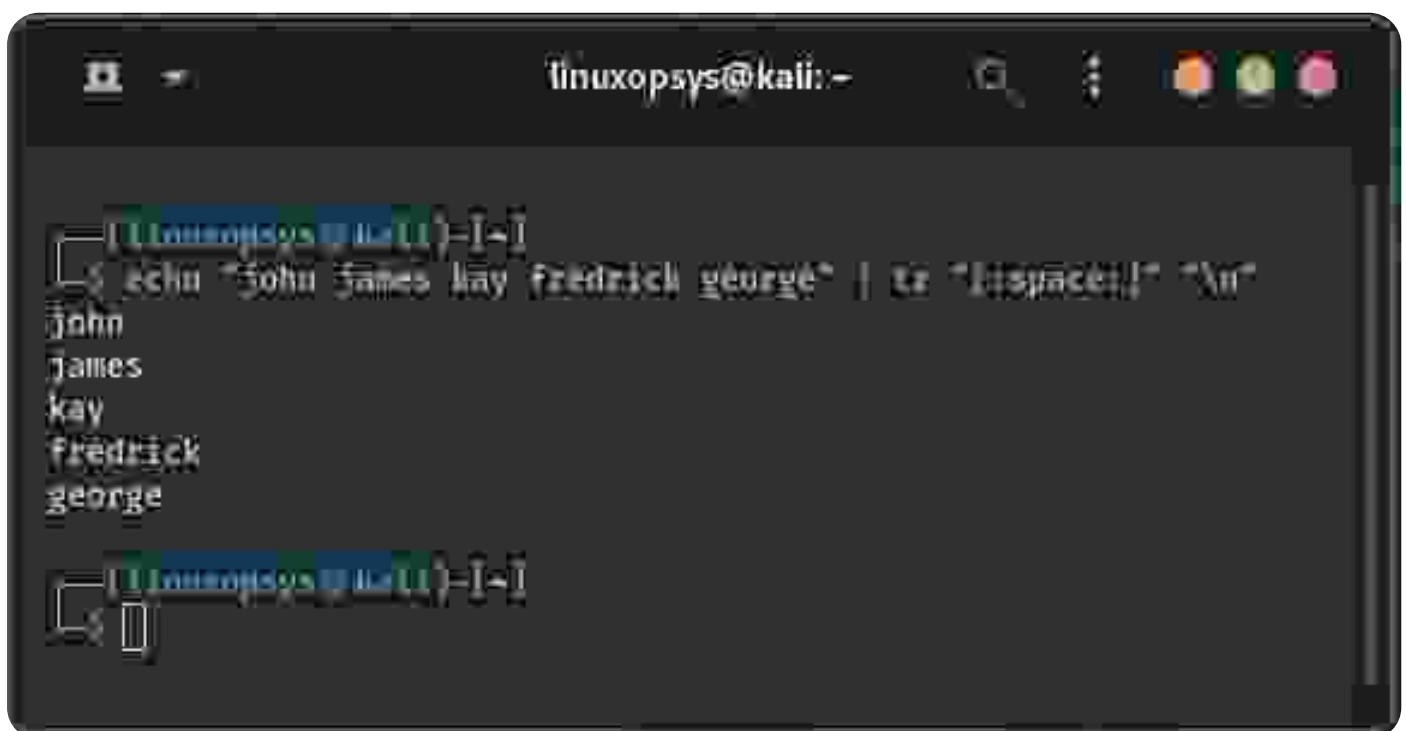
```
linuxopsys@kali:~  
$ tr "\n" " " < file.txt  
I Love Linux.  
$
```

9. Put each word in a new line

As a system administrator, you may be given employee names on single lines separated by spaces, and you may want to put each name on a single line for easy readability.

The command below will assist you in accomplishing this by dividing a sentence into multiple lines, with each word on its own line.

```
$ echo "john james kay fredrick george" | tr "[:space:]"  
"\n"
```

A terminal window titled 'linuxopsys@kali:~' showing the execution of the command `echo "john james kay fredrick george" | tr "[:space:]" "\n"`. The output displays the words 'john', 'james', 'kay', 'fredrick', and 'george' on separate lines. The terminal has a dark background with syntax highlighting for the command and its output.

```
linuxopsys@kali:~  
$ echo "john james kay fredrick george" | tr "[:space:]"  
"\n"  
john  
james  
kay  
fredrick  
george  
$
```

10. Convert a forward slash (/) to ☐ hyphens (-).

As a system administrator, you may be tasked with changing the date formats from yyyy/mm/dd to the new format yyyy-mm-dd.

Here's an example of converting a forward slash (/) to a hyphen (-) and then appending the data to a file for storage.

```
$ tr "/" "-" < old-date-format.txt > new-date-format.txt
```



```
linuxopsys@kali:~$ cat old-date-format.txt
2023/01/23
2023/01/24
2023/01/25

linuxopsys@kali:~$ tr -d '/' < old-date-format.txt > new-date-format.txt

linuxopsys@kali:~$ cat new-date-format.txt
2023-01-23
2023-01-24
2023-01-25

linuxopsys@kali:~$
```

Summing Up!

This thread demonstrated how to use the `tr` command with practical examples and its some available options for various text transformations.

If you get stuck with this command, feel free to refer to the man pages or the command help menu.

That's all!

Hope you learn something new from this thread? If so, please let us know by replying in the comments.

If you're new here, do toss us a follow us (@linuxopsys) for more threads, tips and resources on Linux.

These pages were created and arranged by Rattibha services (<https://www.rattibha.com>)

The contents of these pages, including all images, videos, attachments and external links published (collectively referred to as "this publication"), were created at the request of a user (s) from Twitter. Rattibha provides an automated service, without human intervention, to copy the contents of tweets from Twitter and publish them in an article style, and create PDF pages that can be printed and shared, at the request of Twitter user (s). Please note that the views and all contents in this publication are those of the author and do not necessarily represent the views of Rattibha. Rattibha assumes no responsibility for any damage or breaches

of any law resulting from the contents of this publication.