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.

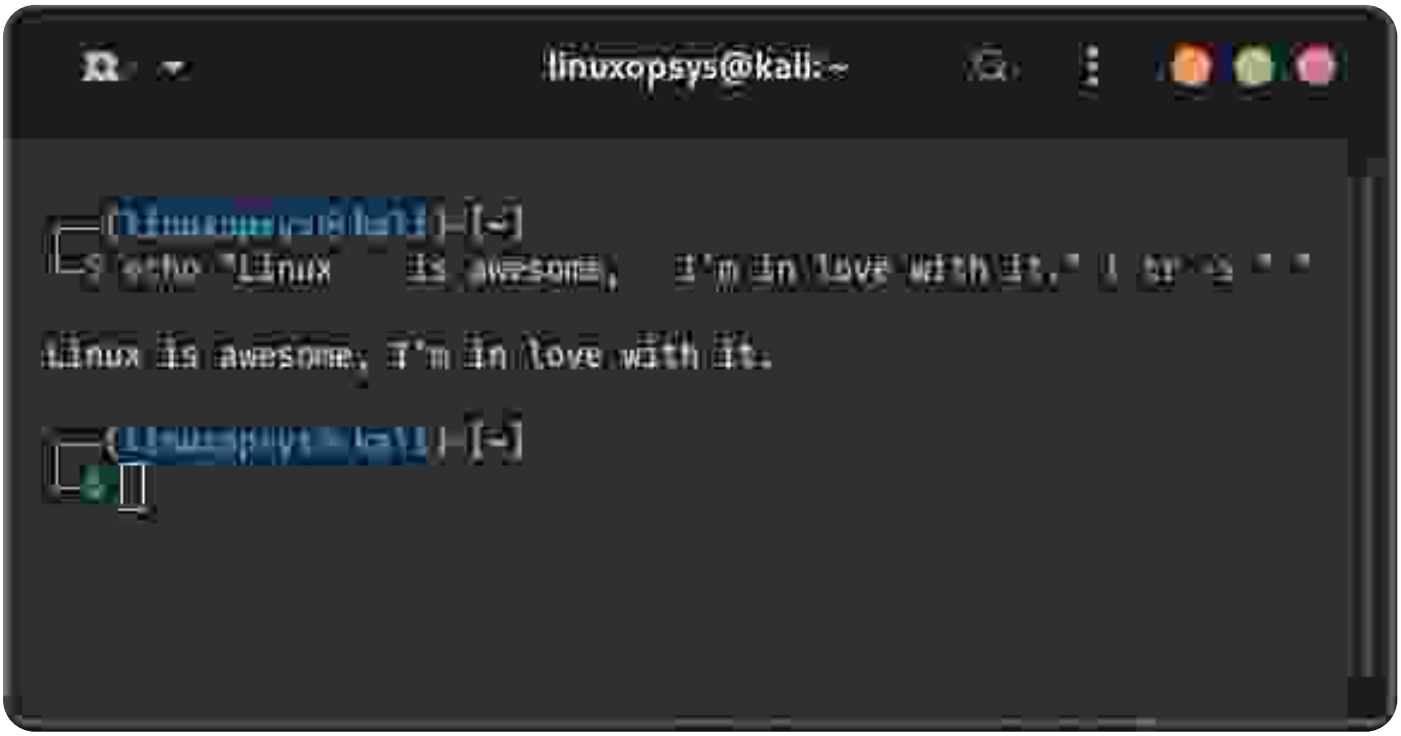
It is usually used in conjunction with other commands via piping.

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.

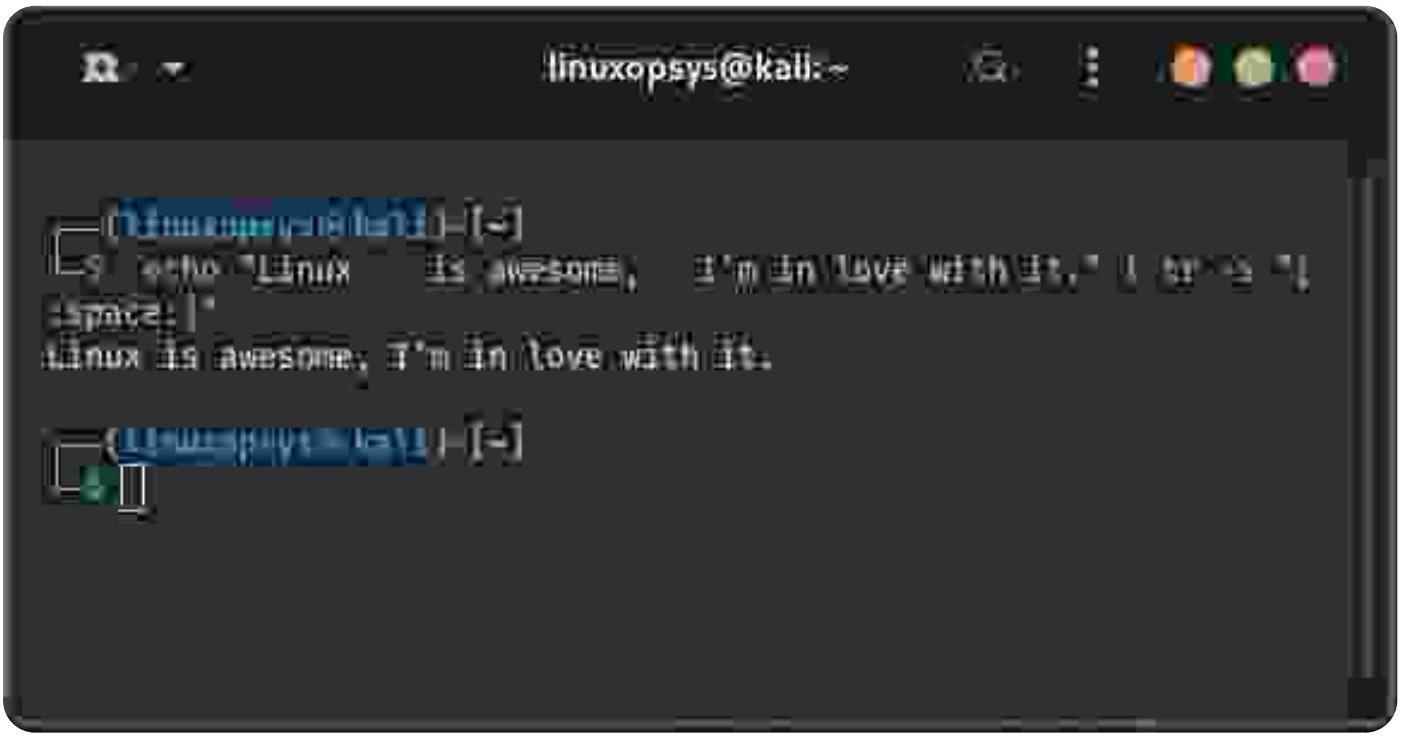
$ echo "Linux is awesome, I'm in love with it." | tr -s " "



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.

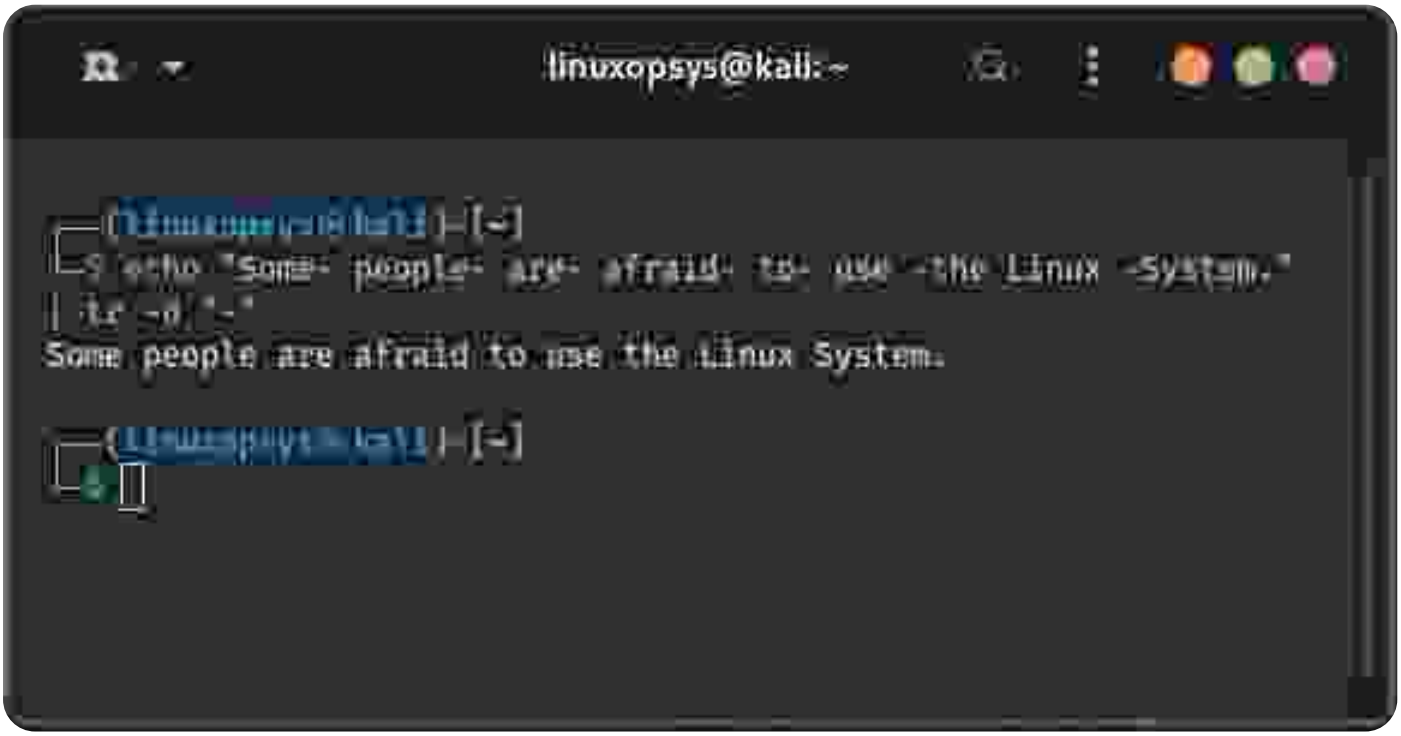
$ echo "Linux is awesome, I'm in love with it." | tr -s "[:space:]"



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:

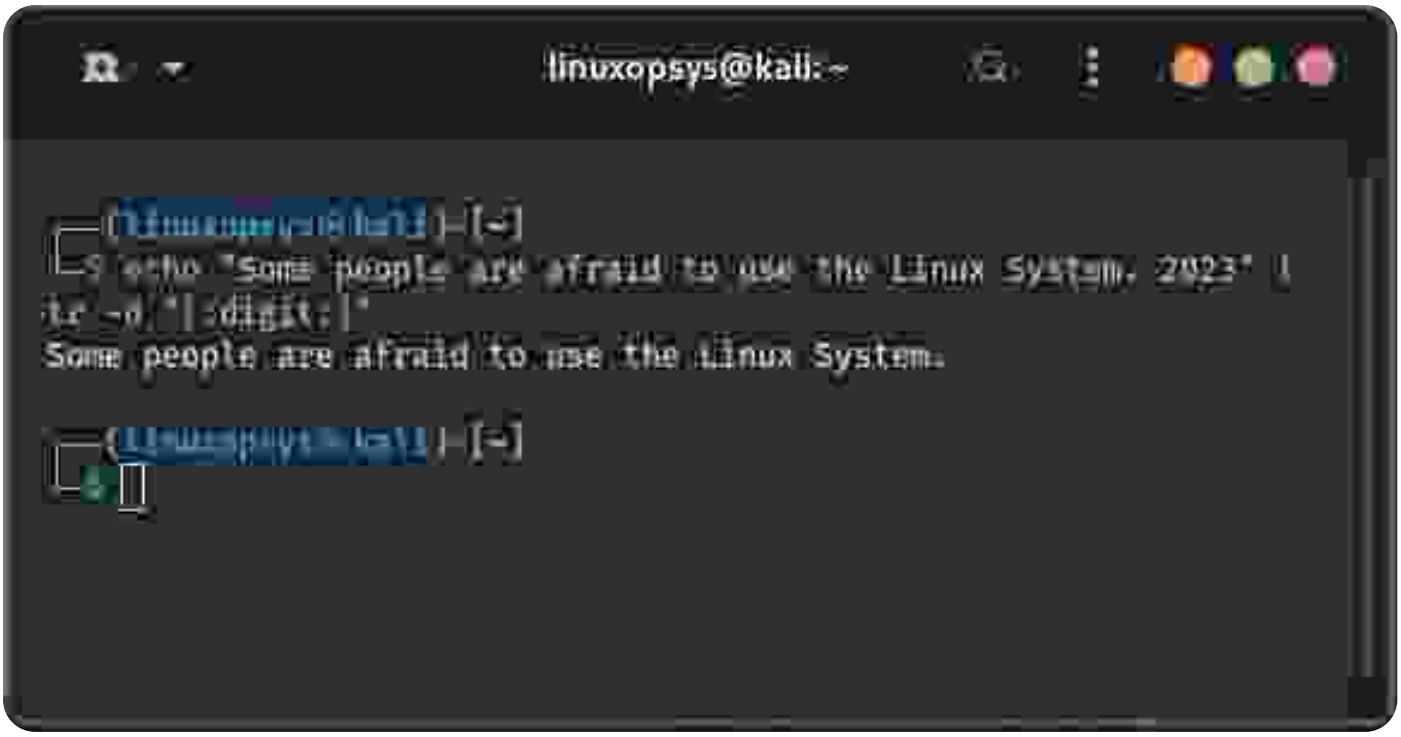
$ echo "Some- people- are- afraid- to- use -the Linux -System." | tr -d "-"



3. Delete all the digits

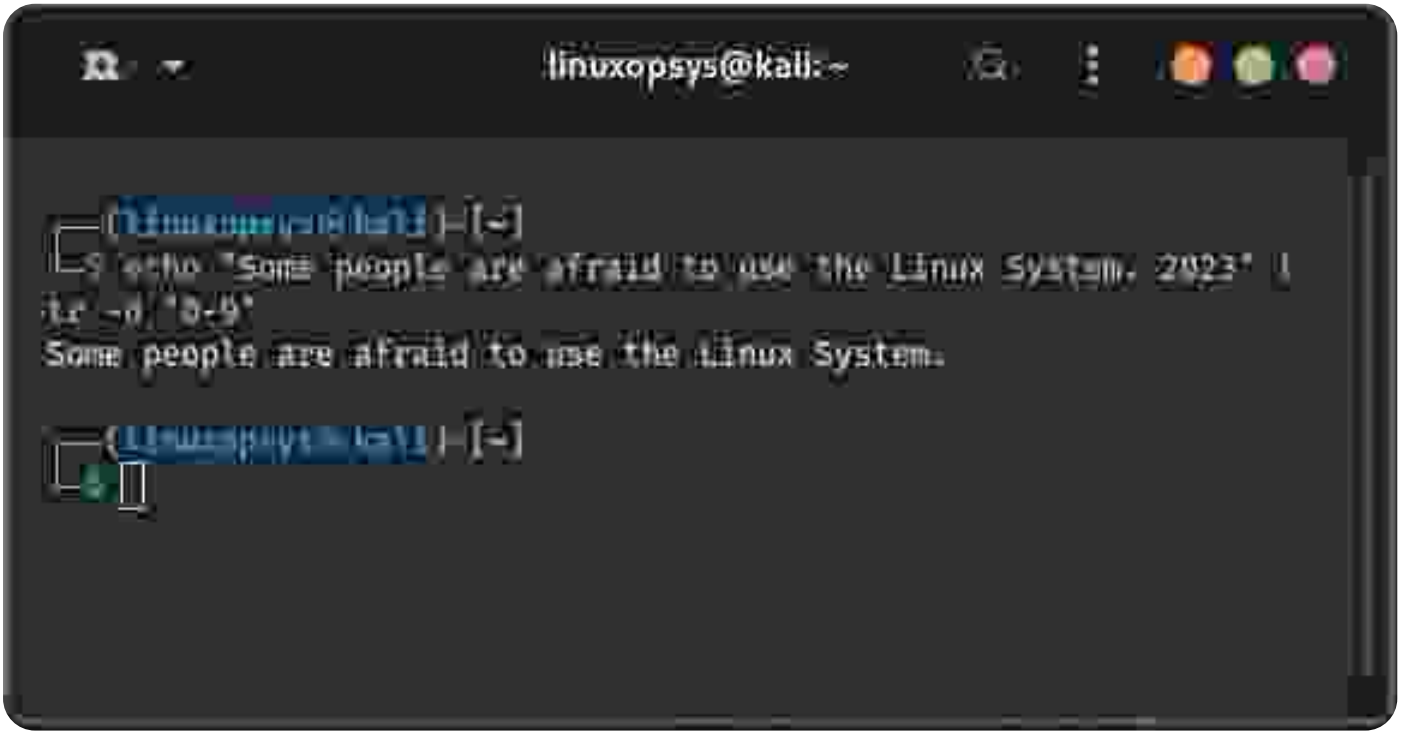
You can also use the -d option to remove any numbers or digits from your text.

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



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"

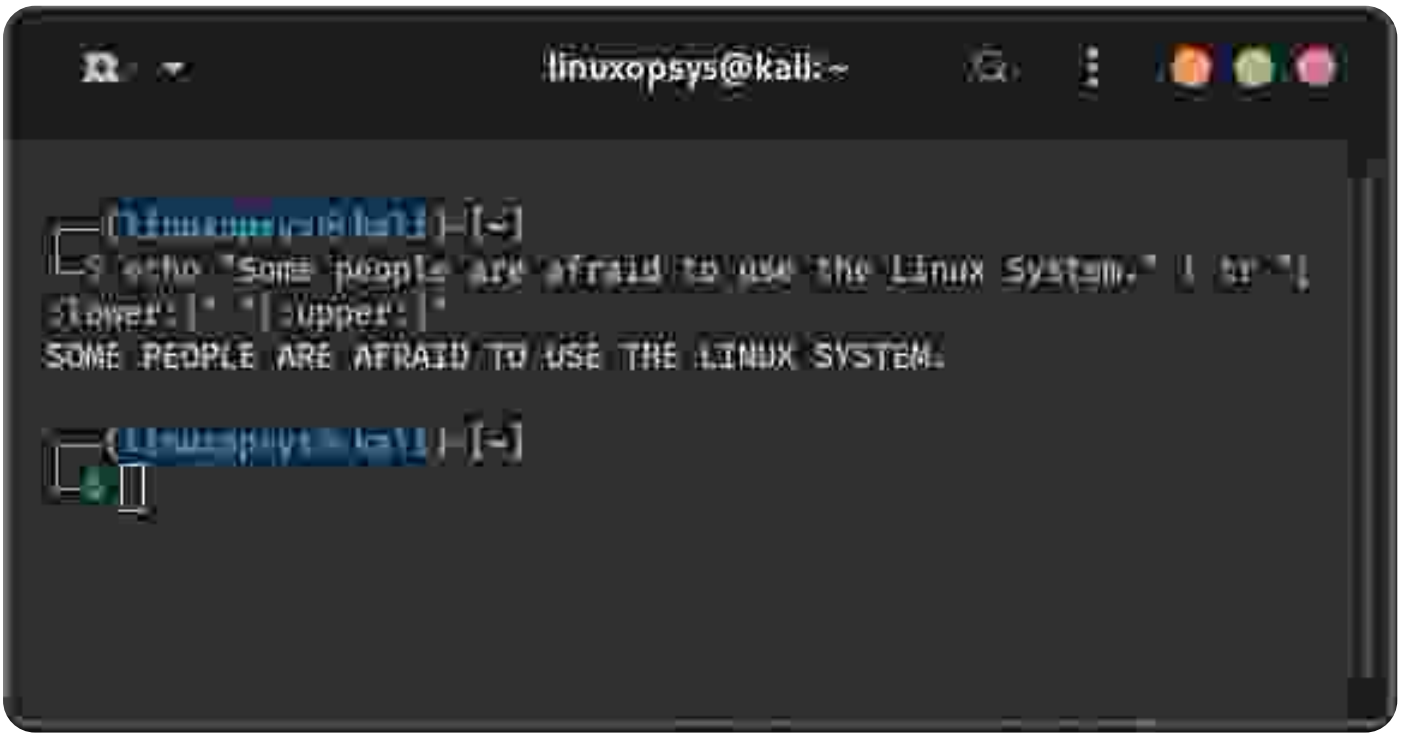


4. Case conversion

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.

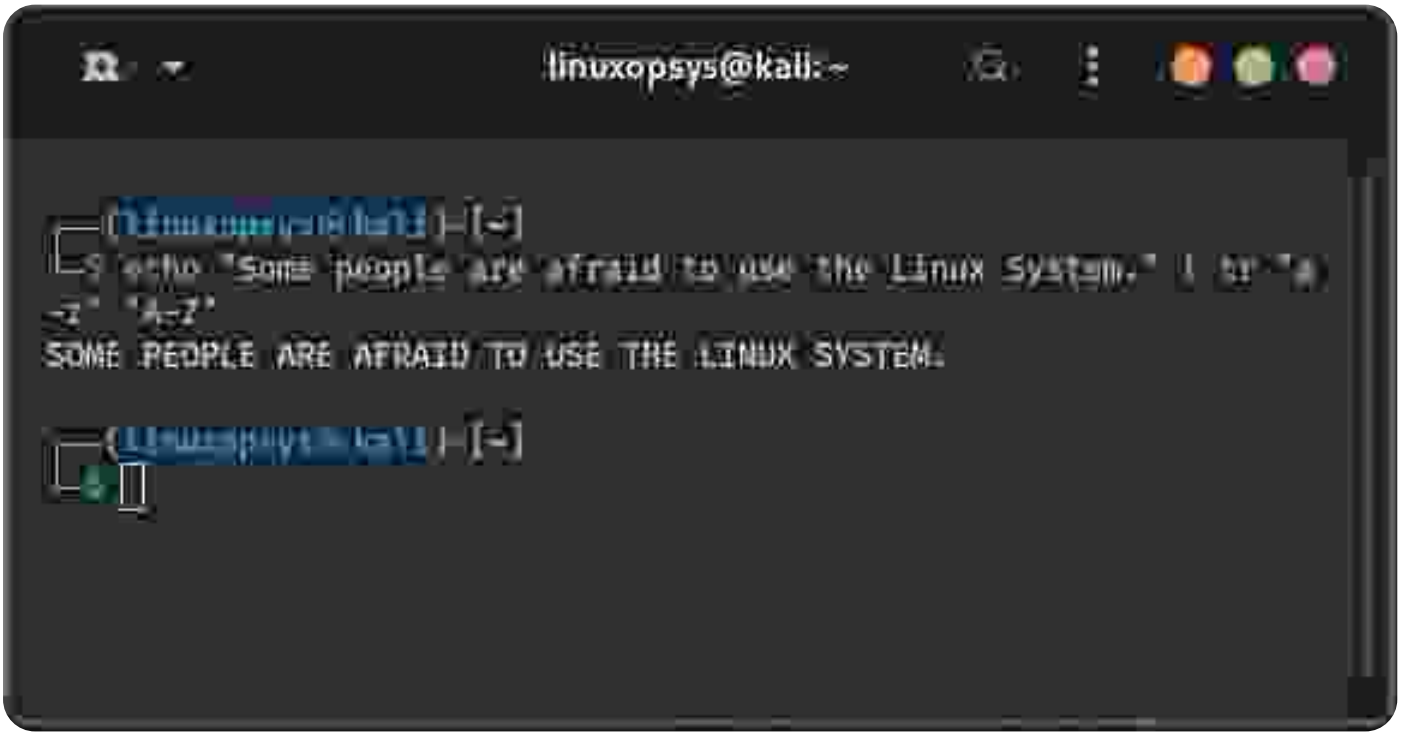
The following will transform characters from upper case to lower case.

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



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"

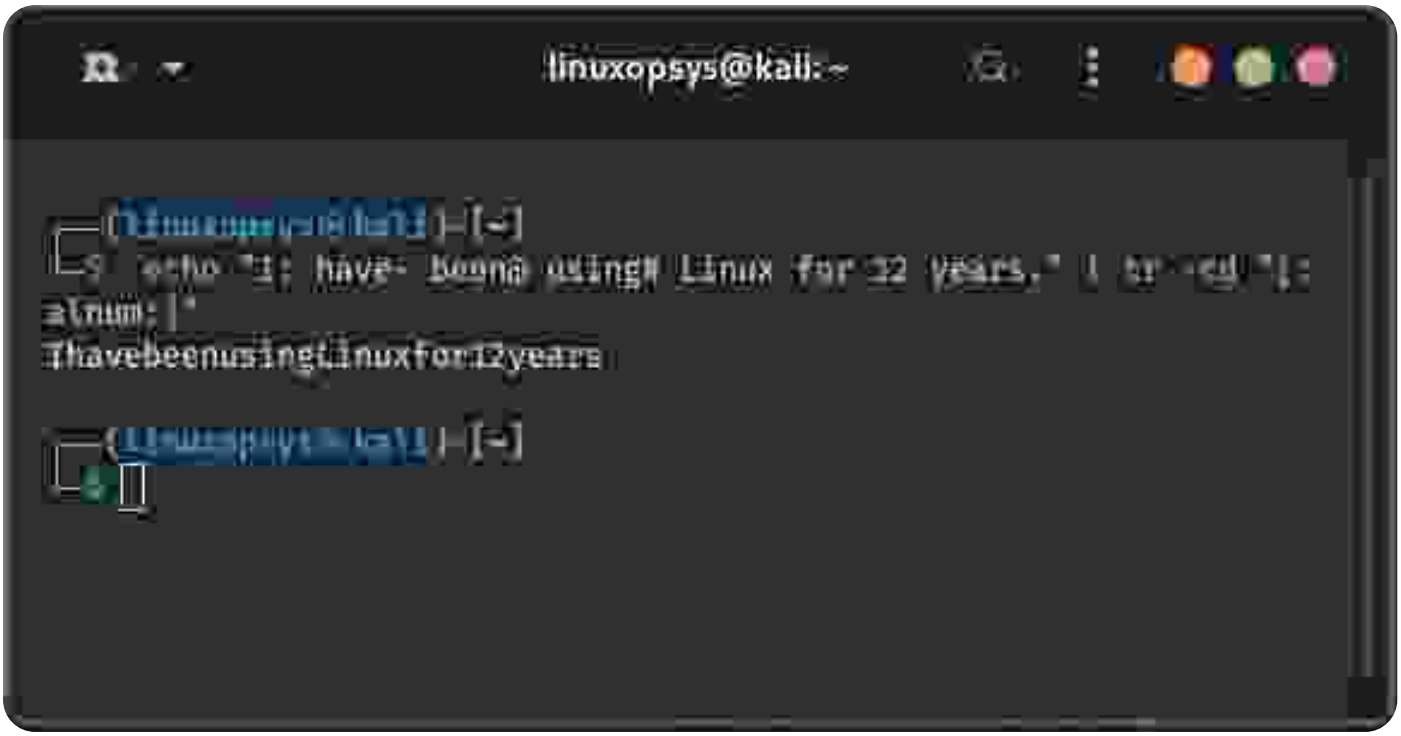


5. Removing non alphanumerical □haracters

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

The following command will delete all non-alphanumerical characters.

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

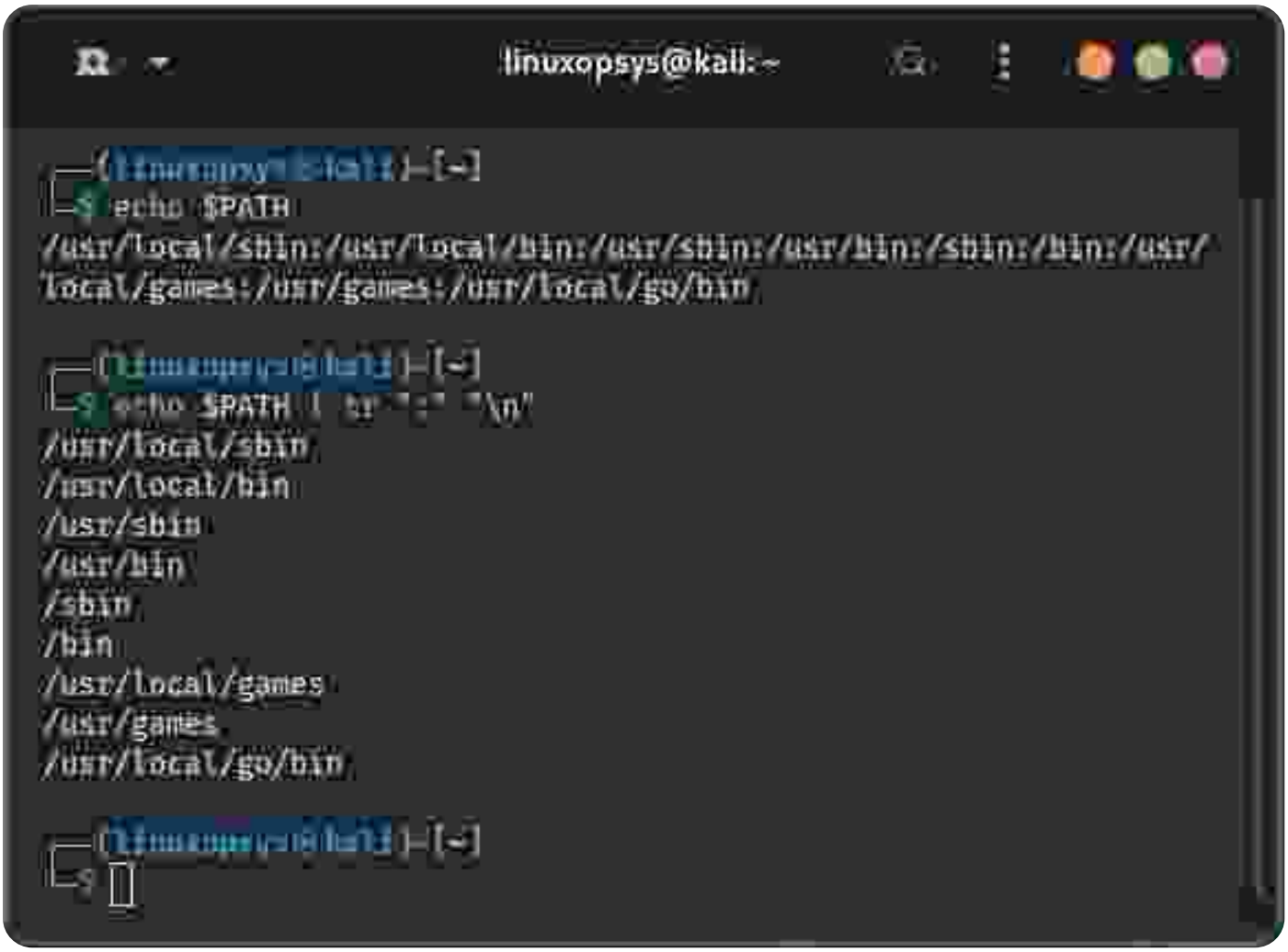


6. Print each word on a single □ □ine

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"



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

7. Remove all non-numeric □ □haracters

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:]"

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

8. Remove Newline Characters

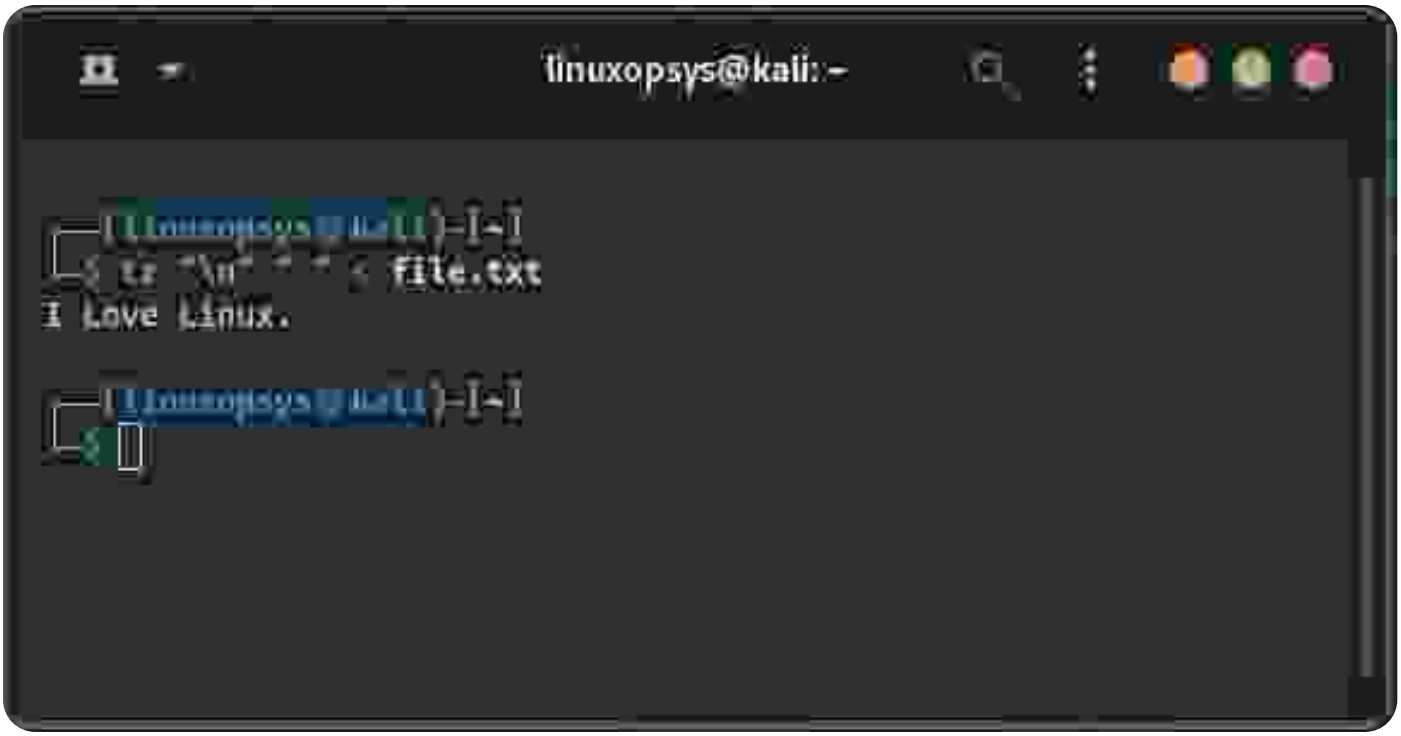
Assume you have a text file containing data that looks like this, and you want to remove the newlines and put the words on a single line separated by spaces.

$ cat file.txt

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.



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.

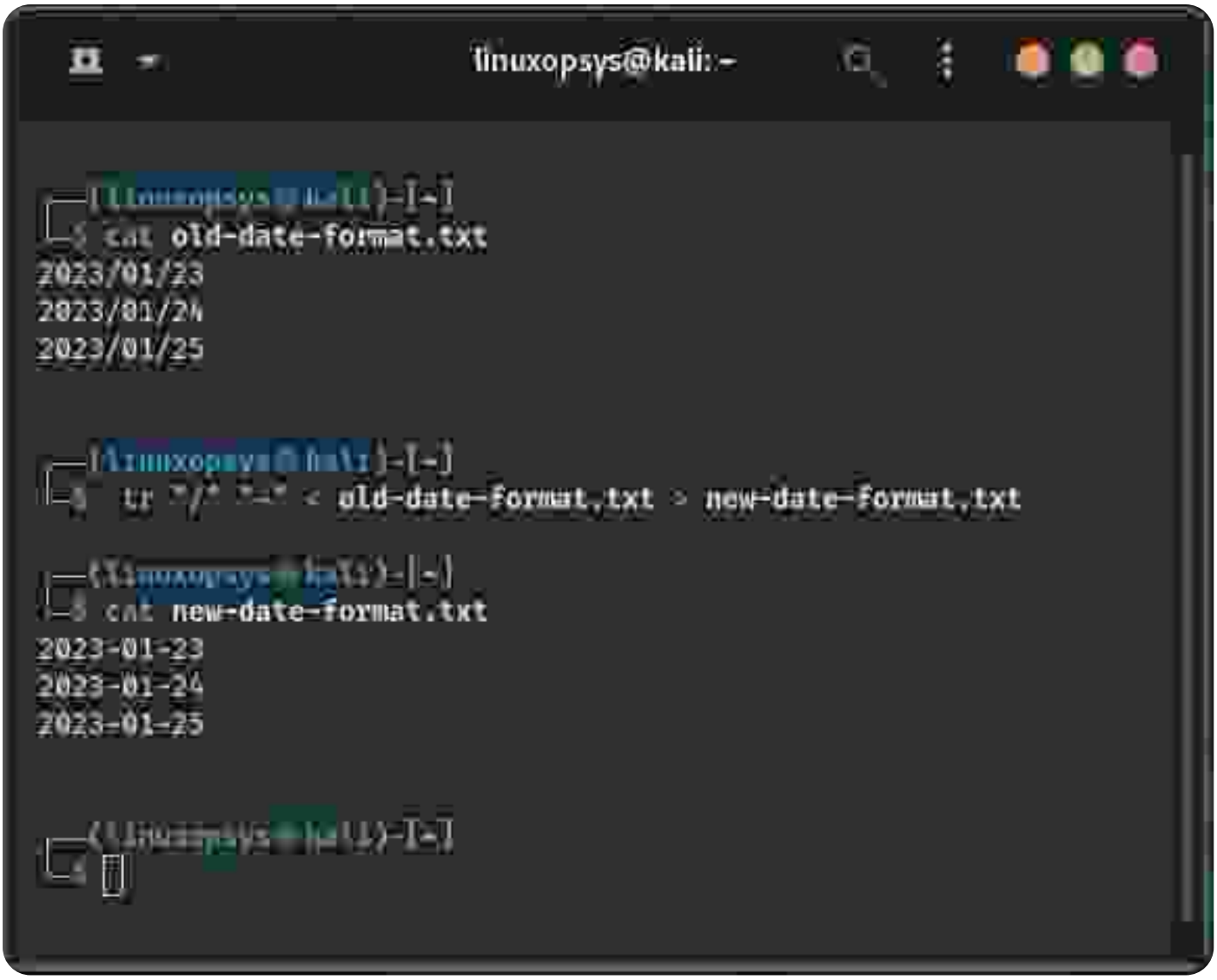
$ echo "john james kay fredrick george" | tr "[:space:]" "\ n"

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



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.

of any law resulting from the contents of this publication.