BASH Programming

Bash Globbing Tutorial

5 years ago • by Fahmida Yesmin

Bash does not support native regular expressions like some other standard programming languages. The Bash shell feature that is used for matching or expanding specific types of patterns is called globbing. Globbing is mainly used to match filenames or searching for content in a file. Globbing uses wildcard characters to create the pattern. The most common wildcard characters that are used for creating globbing patterns are described below.

Question mark – (?)

'?' is used to match any single character. You can use **'?'** for multiple times for matching multiple characters.

Example-1:

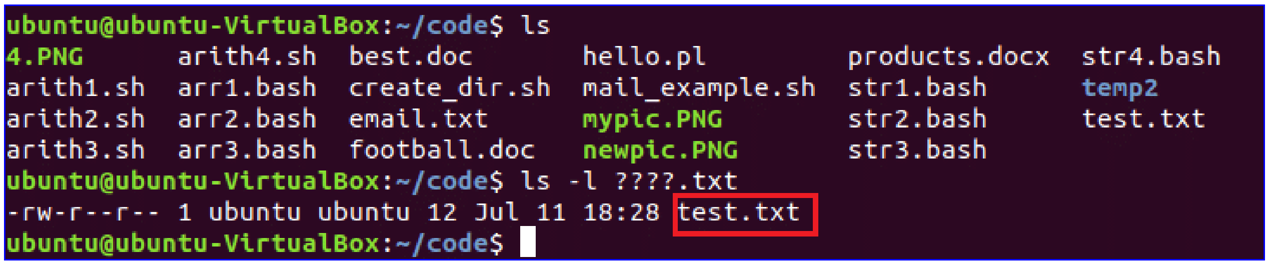
Suppose, you want to search those text filenames whose names are 4 characters long and extension is **.txt**. You can apply globbing pattern by using '**?**' four times to do this task.

Find out the list of all files and folder of the current directory.

```
$ ls -l
```

Run the following command search those files whose names are four characters long and unknown.

```
$ ls -l ????.txt
```

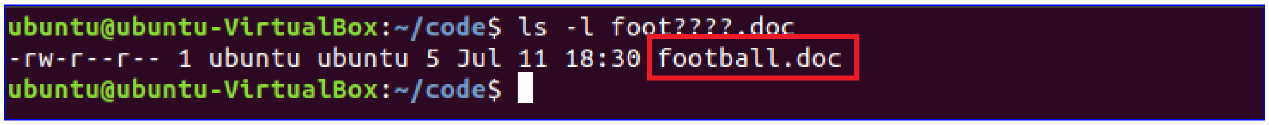


```
ubuntu@ubuntu-VirtualBox:~/code$ ls
4.PNG       arith4.sh  best.doc       hello.pl          products.docx  str4.bash
arith1.sh   arr1.bash  create_dir.sh  mail_example.sh   str1.bash      temp2
arith2.sh   arr2.bash  email.txt      mypic.PNG         str2.bash      test.txt
arith3.sh   arr3.bash  football.doc   newpic.PNG        str3.bash
ubuntu@ubuntu-VirtualBox:~/code$ ls -l ????.txt
-rw-r--r-- 1 ubuntu ubuntu 12 Jul 11 18:28 test.txt
ubuntu@ubuntu-VirtualBox:~/code$
```

Example-2:

Suppose, you want to search those document files whose names are 8 characters long, first 4 characters are **f, o, o** and **t** and extension is **doc**. Run the following command with globbing pattern to search the files.

```
$ ls -l foot????.doc
```

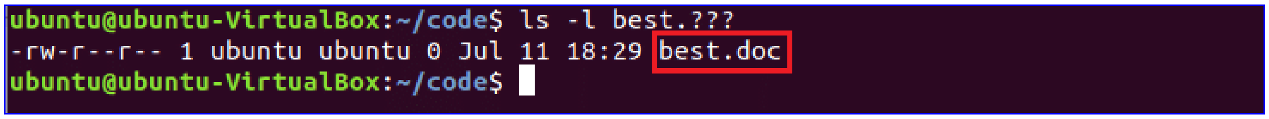


```
ubuntu@ubuntu-VirtualBox:~/code$ ls -l foot????.doc
-rw-r--r-- 1 ubuntu ubuntu 5 Jul 11 18:30 football.doc
ubuntu@ubuntu-VirtualBox:~/code$
```

Example-3:

Suppose, you know the filename is **'best'** and extension is 3 characters long, but don't know the extension. Run the following command by using **'?'** to search all files with the name 'test' having any extension of three characters long.

```
$ ls -l best.???
```



```
ubuntu@ubuntu-VirtualBox:~/code$ ls -l best.???
-rw-r--r-- 1 ubuntu ubuntu 0 Jul 11 18:29 best.doc
ubuntu@ubuntu-VirtualBox:~/code$
```

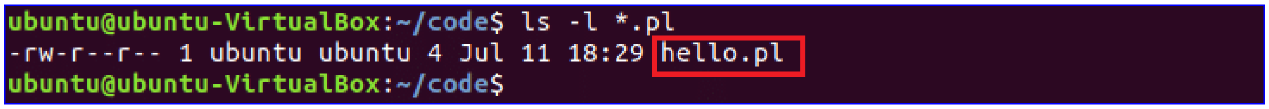
Asterisk – (*)

'*' is used to match zero or more characters. If you have less information to search any file or information then you can use **'*'** in globbing pattern.

Example -1:

Suppose, you want to search all files of **‘pl’** extension. Run the following command using **‘*’** to do that task.

```
$ ls -l *.pl
```

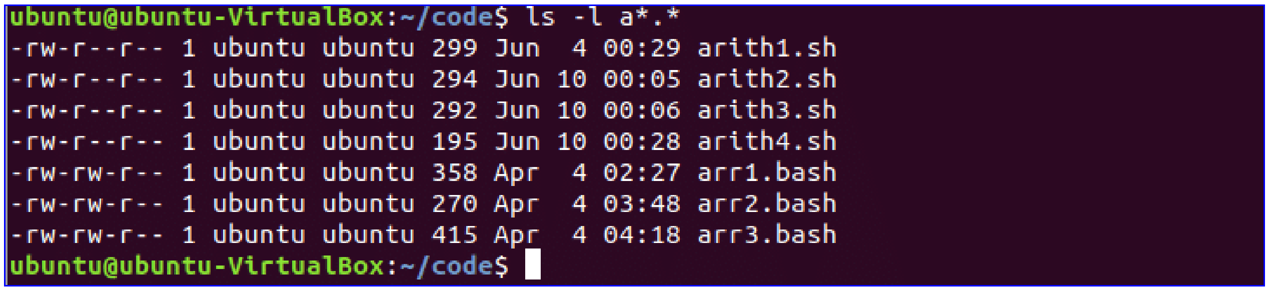


```
ubuntu@ubuntu-VirtualBox:~/code$ ls -l *.pl
-rw-r--r-- 1 ubuntu ubuntu 4 Jul 11 18:29 hello.pl
ubuntu@ubuntu-VirtualBox:~/code$
```

Example-2:

Suppose, you know the starting character of the filename only which is **‘a’**. Run the following command using ‘*’ to search all files of the current directory whose names are started with **‘a’**.

```
$ ls -l a*.*
```



```
ubuntu@ubuntu-VirtualBox:~/code$ ls -l a*.*
-rw-r--r-- 1 ubuntu ubuntu 299 Jun  4 00:29 arith1.sh
-rw-r--r-- 1 ubuntu ubuntu 294 Jun 10 00:05 arith2.sh
-rw-r--r-- 1 ubuntu ubuntu 292 Jun 10 00:06 arith3.sh
-rw-r--r-- 1 ubuntu ubuntu 195 Jun 10 00:28 arith4.sh
-rw-rw-r-- 1 ubuntu ubuntu 358 Apr  4 02:27 arr1.bash
-rw-rw-r-- 1 ubuntu ubuntu 270 Apr  4 03:48 arr2.bash
-rw-rw-r-- 1 ubuntu ubuntu 415 Apr  4 04:18 arr3.bash
ubuntu@ubuntu-VirtualBox:~/code$
```

Example-3:

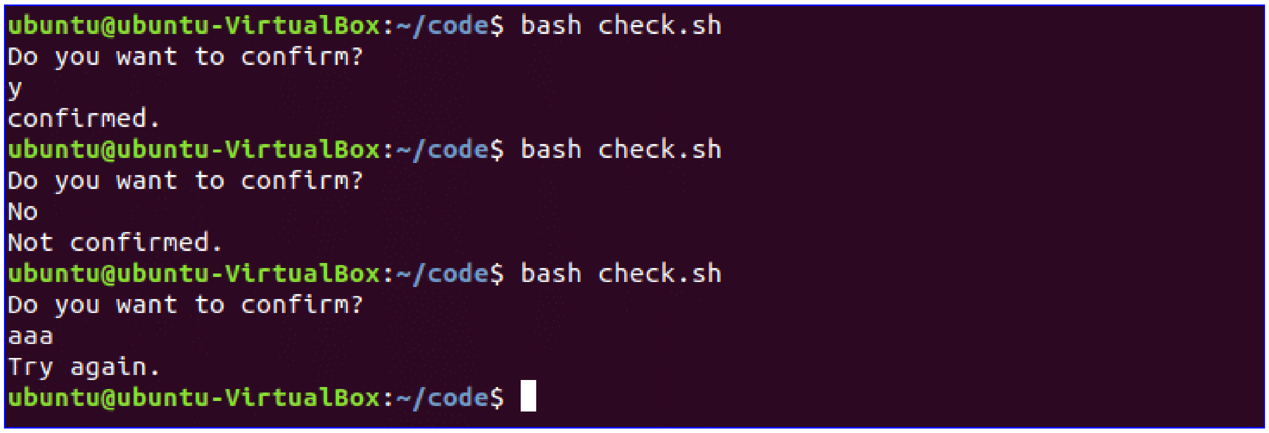
You can apply ‘*’ in bash script for various purposes without searching files. Create a bash file named **‘check.sh’** with the following script. Here, when the user will type ‘**y’ or ‘Y’ or ‘yes’ or ‘Yes’** then **‘confirmed’** will print and when the type will type **‘n’ or ‘N’ or ‘no’ or ‘No’** then **‘Not confirmed’** will print.

```
#!/bin/bash
echo "Do you want to confirm?"
read answer
case $answer in
[Yy]* )  echo "confirmed.";;
[Nn]* )  echo "Not confirmed.";;
```

```
*) echo "Try again.";;
esac
```

Run the script.

```
$ bash check.sh
```



```
ubuntu@ubuntu-VirtualBox:~/code$ bash check.sh
Do you want to confirm?
y
confirmed.
ubuntu@ubuntu-VirtualBox:~/code$ bash check.sh
Do you want to confirm?
No
Not confirmed.
ubuntu@ubuntu-VirtualBox:~/code$ bash check.sh
Do you want to confirm?
aaa
Try again.
ubuntu@ubuntu-VirtualBox:~/code$
```

Square Bracket – ([])

'[]' is used to match the character from the range. Some of the mostly used range declarations are mentioned below.

All uppercase alphabets are defined by the range as, [:upper:] or [A-Z] .

All lowercase alphabets are defined by the range as, [:lower:] or [a-z].

All numeric digits are defined by the range as, [:digit:] or [0-9].

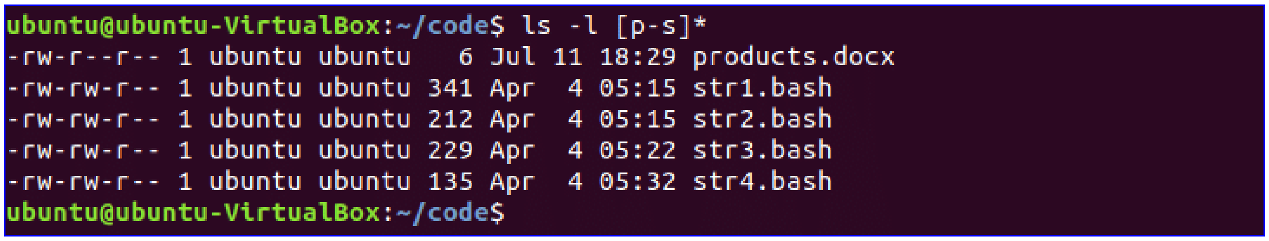
All uppercase and lower alphabets are defined by the range as, [:alpha:] or [a-zA-z].

All uppercase alphabets, lowercase alphabet and digits are defined by the range as, [:alnum:] or [a-zA-Z0-9]

Example -1:

Run the following command to search all files and folders whose name contains **p** or **q** or **r** or **s**.

```
$ ls -l [p-s]*
```

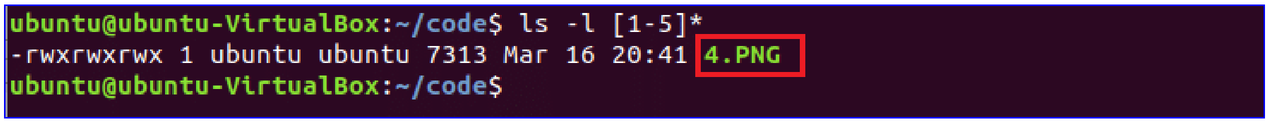


```
ubuntu@ubuntu-VirtualBox:~/code$ ls -l [p-s]*
-rw-r--r-- 1 ubuntu ubuntu   6 Jul 11 18:29 products.docx
-rw-rw-r-- 1 ubuntu ubuntu 341 Apr  4 05:15 str1.bash
-rw-rw-r-- 1 ubuntu ubuntu 212 Apr  4 05:15 str2.bash
-rw-rw-r-- 1 ubuntu ubuntu 229 Apr  4 05:22 str3.bash
-rw-rw-r-- 1 ubuntu ubuntu 135 Apr  4 05:32 str4.bash
ubuntu@ubuntu-VirtualBox:~/code$
```

Example-2:

Run the following command to search all files and folders whose name starts with any digit from 1 to 5.

```
$ ls -l [1-5]*
```



```
ubuntu@ubuntu-VirtualBox:~/code$ ls -l [1-5]*
-rwxrwxrwx 1 ubuntu ubuntu 7313 Mar 16 20:41 4.PNG
ubuntu@ubuntu-VirtualBox:~/code$
```

Caret – (^)

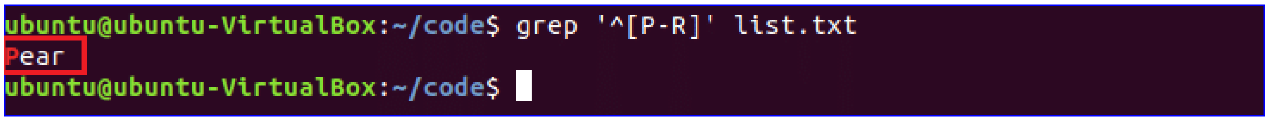
You can use **'^'** with square bracket to define globbing pattern more specifically. **'^'** can be used inside or outside of square bracket. '^' is used outside the square bracket to search those contents of the file that starts with a given range of characters. '^' is used inside the square bracket to show all content of the file by highlighting the lines start with a given range of characters . You can use different types of globbing patterns for searching particular content from a file. **'grep'** command is used for content searching in bash. Suppose, you have a text file named **'list.txt'** with the following content. Test the following examples for that file.

```
Apple
4000
Banana
700
Orange
850
Pear
9000
Jackdruit
```

Example – 1:

Run the following command to search those lines from **list.txt** file that starts with **P or Q or R**.

```
$ grep '^[P-R]' list.txt
```

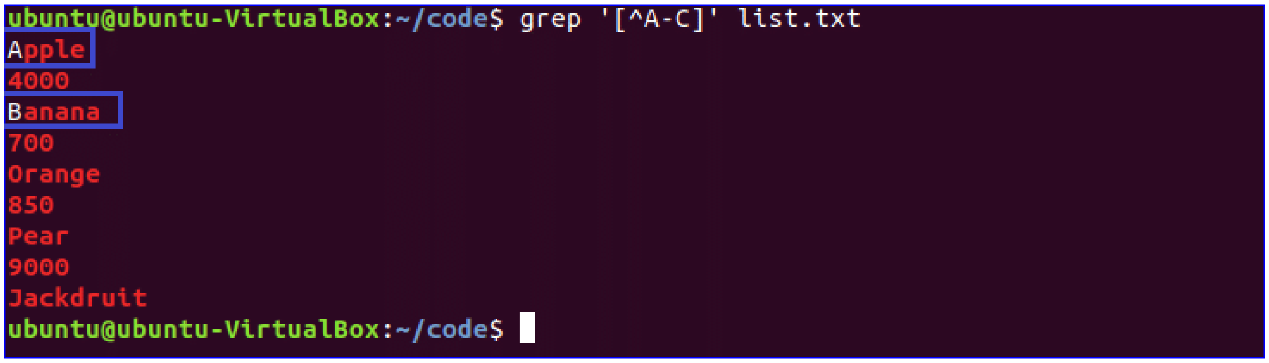


```
ubuntu@ubuntu-VirtualBox:~/code$ grep '^[P-R]' list.txt
Pear
ubuntu@ubuntu-VirtualBox:~/code$
```

Example – 2:

Run the following command to highlight those lines from **list.txt** file that starts with **A or B or C**.

```
$ grep '[^A-C]' list.txt
```



```
ubuntu@ubuntu-VirtualBox:~/code$ grep '[^A-C]' list.txt
Apple
4000
Banana
700
Orange
850
Pear
9000
Jackdruit
ubuntu@ubuntu-VirtualBox:~/code$
```

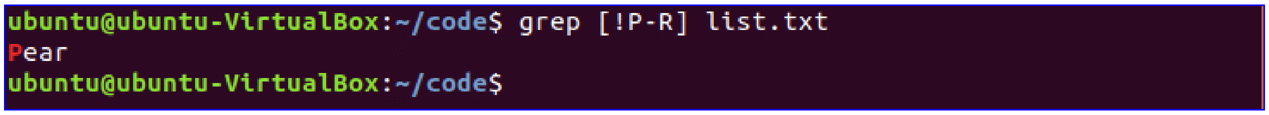
Exclamatory Sign – (!)

You can use **'!'** inside the range pattern. It works same as the use of '^' symbol outside the range pattern. Some examples of using **'!'** sign are given below.

Example – 1:

Run the following command to show those lines from **list.txt** file that starts with **'P' or Q or R**.

```
$ grep [!P-R] list.txt
```

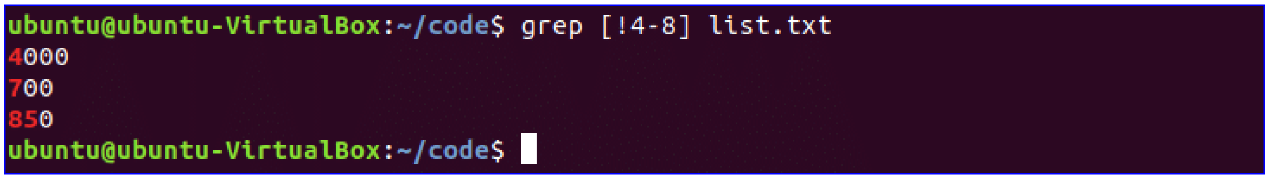


```
ubuntu@ubuntu-VirtualBox:~/code$ grep [!P-R] list.txt
Pear
ubuntu@ubuntu-VirtualBox:~/code$
```

Example – 2:

Run the following command to show those lines from **list.txt** file that starts with any digit from **4 to 8**.

```
$ grep [!4-8] list.txt
```



```
ubuntu@ubuntu-VirtualBox:~/code$ grep [!4-8] list.txt
4000
700
850
ubuntu@ubuntu-VirtualBox:~/code$
```

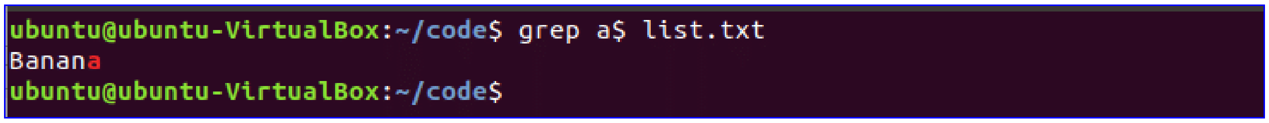
Dollar Sign – ($)

'$'** is used to define the ending character. If you know want to search information based on last character then you can use **'$' in globbing pattern.

Example – 1:

Run the following command to search those lines from **list.txt** file that ends with **'a'**.

```
$ grep a$ list.txt
```

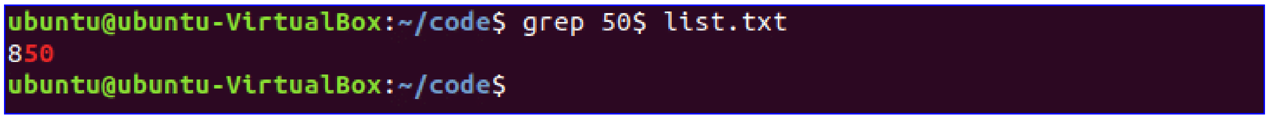


```
ubuntu@ubuntu-VirtualBox:~/code$ grep a$ list.txt
Banana
ubuntu@ubuntu-VirtualBox:~/code$
```

Example – 2:

Run the following command to search those lines from **list.txt** file that end with the number **50**.

```
$ grep 50$ list.txt
```



```
ubuntu@ubuntu-VirtualBox:~/code$ grep 50$ list.txt
850
ubuntu@ubuntu-VirtualBox:~/code$
```

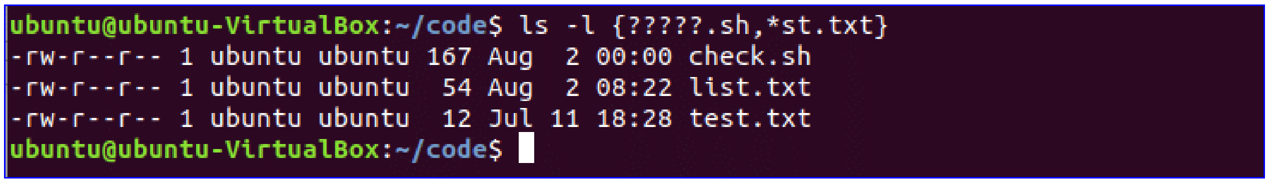
Curly bracket – ({})

'{}' can be used to match filenames with more than one globbing patterns. Each pattern is separated by ',' in curly bracket without any space. Some examples are given below.

Example – 1:

Run the following command to search those files whose names are 5 characters long and the extension is **'sh'** or the last two characters of the files are **'st'** and the extension is **'txt'**.

```
$ ls -l {?????.sh,*st.txt}
```

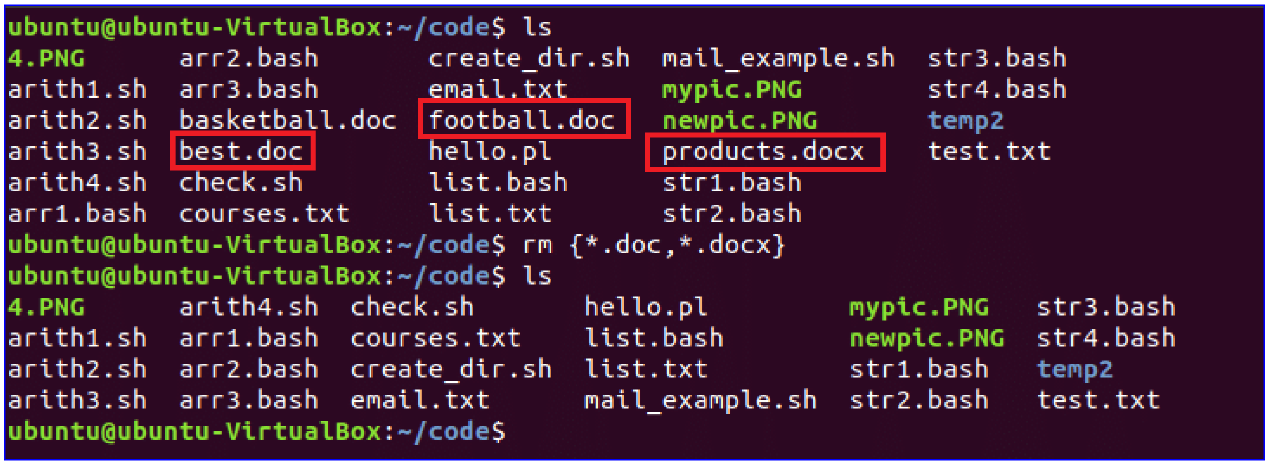


```
ubuntu@ubuntu-VirtualBox:~/code$ ls -l {?????.sh,*st.txt}
-rw-r--r-- 1 ubuntu ubuntu 167 Aug  2 00:00 check.sh
-rw-r--r-- 1 ubuntu ubuntu  54 Aug  2 08:22 list.txt
-rw-r--r-- 1 ubuntu ubuntu  12 Jul 11 18:28 test.txt
ubuntu@ubuntu-VirtualBox:~/code$
```

Example – 2:

Run the following command to delete all files whose extensions are **'doc' or 'docx'**.

```
$ rm {*.doc,*.docx}
```



```
ubuntu@ubuntu-VirtualBox:~/code$ ls
4.PNG      arr2.bash       create_dir.sh  mail_example.sh  str3.bash
arith1.sh  arr3.bash       email.txt      mypic.PNG        str4.bash
arith2.sh  basketball.doc  football.doc   newpic.PNG       temp2
arith3.sh  best.doc        hello.pl       products.docx    test.txt
arith4.sh  check.sh        list.bash      str1.bash
arr1.bash  courses.txt     list.txt       str2.bash
ubuntu@ubuntu-VirtualBox:~/code$ rm {*.doc,*.docx}
ubuntu@ubuntu-VirtualBox:~/code$ ls
4.PNG      arith4.sh  check.sh       hello.pl         mypic.PNG   str3.bash
arith1.sh  arr1.bash  courses.txt    list.bash        newpic.PNG  str4.bash
arith2.sh  arr2.bash  create_dir.sh  list.txt         str1.bash   temp2
arith3.sh  arr3.bash  email.txt      mail_example.sh  str2.bash   test.txt
ubuntu@ubuntu-VirtualBox:~/code$
```

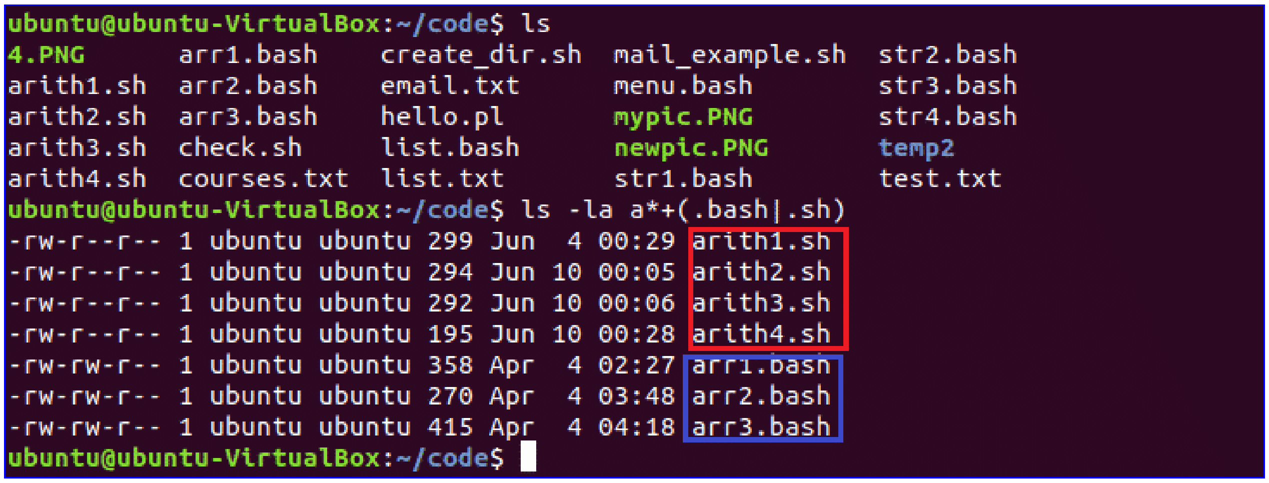
Pipe– (|)

'|' sign is also used for applying more than one condition on globbing pattern. Each pattern is separated by '|' symbol in the command.

Example – 1:

Run the following command to search those filenames which are starting with character ‘a’ and has the extension ‘bash’ or ‘sh’.

```
$ ls a*+(.bash|.sh)
```



```
ubuntu@ubuntu-VirtualBox:~/code$ ls
4.PNG       arr1.bash    create_dir.sh  mail_example.sh  str2.bash
arith1.sh   arr2.bash    email.txt      menu.bash        str3.bash
arith2.sh   arr3.bash    hello.pl       mypic.PNG        str4.bash
arith3.sh   check.sh     list.bash      newpic.PNG       temp2
arith4.sh   courses.txt  list.txt       str1.bash        test.txt
ubuntu@ubuntu-VirtualBox:~/code$ ls -la a*+(.bash|.sh)
-rw-r--r-- 1 ubuntu ubuntu 299 Jun  4 00:29 arith1.sh
-rw-r--r-- 1 ubuntu ubuntu 294 Jun 10 00:05 arith2.sh
-rw-r--r-- 1 ubuntu ubuntu 292 Jun 10 00:06 arith3.sh
-rw-r--r-- 1 ubuntu ubuntu 195 Jun 10 00:28 arith4.sh
-rw-rw-r-- 1 ubuntu ubuntu 358 Apr  4 02:27 arr1.bash
-rw-rw-r-- 1 ubuntu ubuntu 270 Apr  4 03:48 arr2.bash
-rw-rw-r-- 1 ubuntu ubuntu 415 Apr  4 04:18 arr3.bash
ubuntu@ubuntu-VirtualBox:~/code$
```

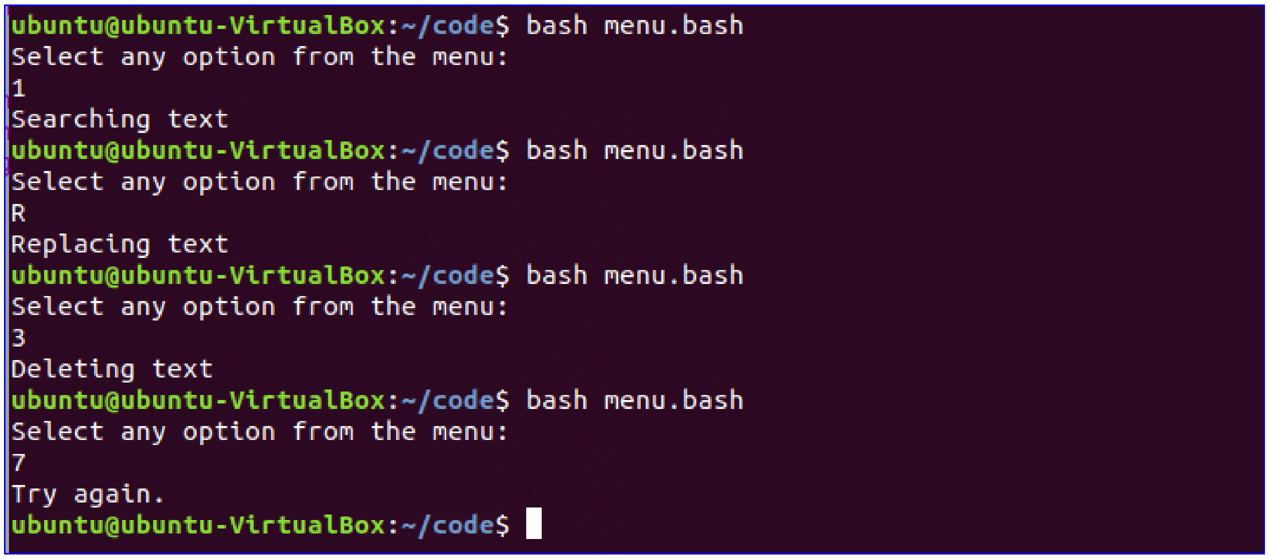
Example – 2:

Create a bash file named **‘menu.bash’** and add the following script. If the user type **1 or S** then it will print **“Searching text”**. If the user type **2 or R** then it will print “**Replacing text**”. If the user type **3 or D** then it will print **“Deleting text”**. It will print **“Try again”** for any other input.

```
#!/bin/bash
echo "Select any option from the menu:"
read answer
case $answer in
1 | S )  echo "Searching text";;
2 | R )  echo "Replacing text";;
3 | D )  echo "Deleting text";;
*) echo "Try again.";;
esac
```

Run the script.

```
$ bash menu.bash
```

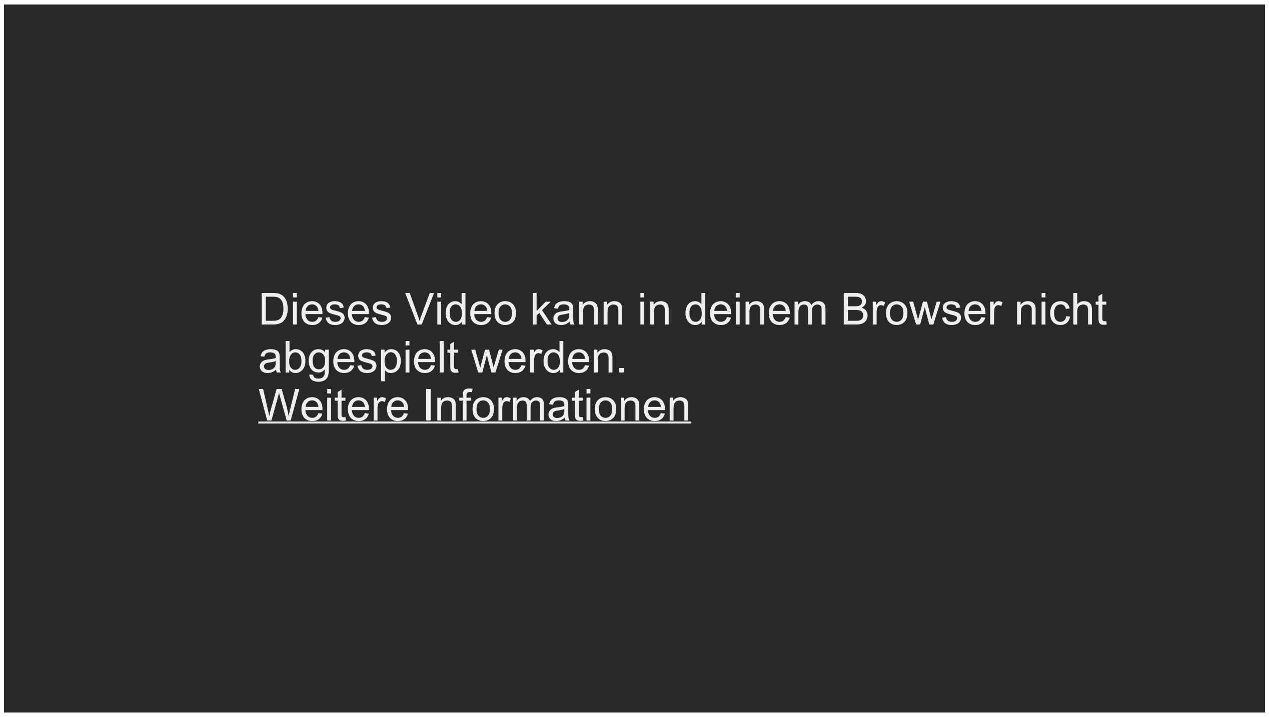


```
ubuntu@ubuntu-VirtualBox:~/code$ bash menu.bash
Select any option from the menu:
1
Searching text
ubuntu@ubuntu-VirtualBox:~/code$ bash menu.bash
Select any option from the menu:
R
Replacing text
ubuntu@ubuntu-VirtualBox:~/code$ bash menu.bash
Select any option from the menu:
3
Deleting text
ubuntu@ubuntu-VirtualBox:~/code$ bash menu.bash
Select any option from the menu:
7
Try again.
ubuntu@ubuntu-VirtualBox:~/code$
```

CONCLUSION

Some of the most commonly used globbing patterns are explained in this tutorial by using very simple examples. I hope after practicing the above examples, the concept of globbing will be clear to you and you will be able to apply it in bash commands and scripts successfully.

For more info check this video:



Dieses Video kann in deinem Browser nicht abgespielt werden.
Weitere Informationen

ABOUT THE AUTHOR

Fahmida Yesmin

I am a trainer of web programming courses. I like to write article or tutorial on various IT topics. I have a YouTube channel where many types of tutorials based on Ubuntu, Windows, Word, Excel, WordPress, Magento, Laravel etc. are published: Tutorials4u Help.

View all posts

RELATED LINUX HINT POSTS

Is There a TRY CATCH Command in Bash

How To Check Existence of Input Argument in a Bash Shell Script

How to Fix – bash: pip: command not found

How to Add a New Element to an Array Without Specifying the Index in Bash

How To Find the Length of an Array in Shell Script

How to Extract Part of a String Using Bash cut and split Commands

How to Count Number of Lines in Terminal Output in Bash