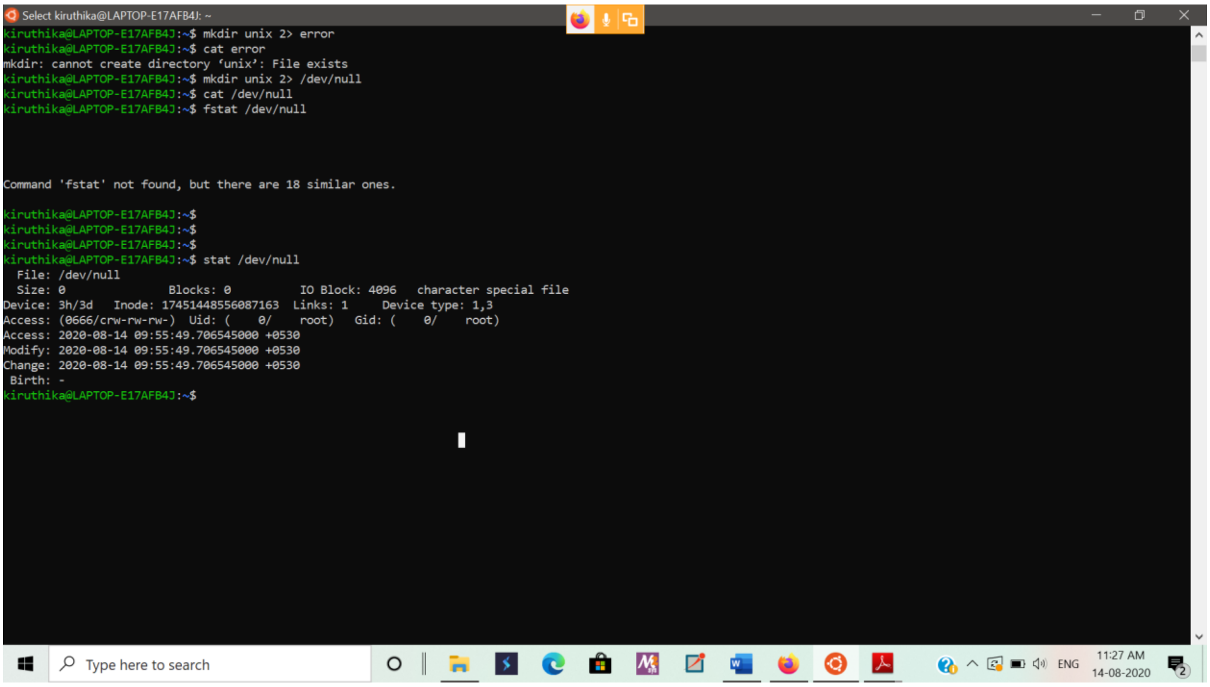
/dev/null file

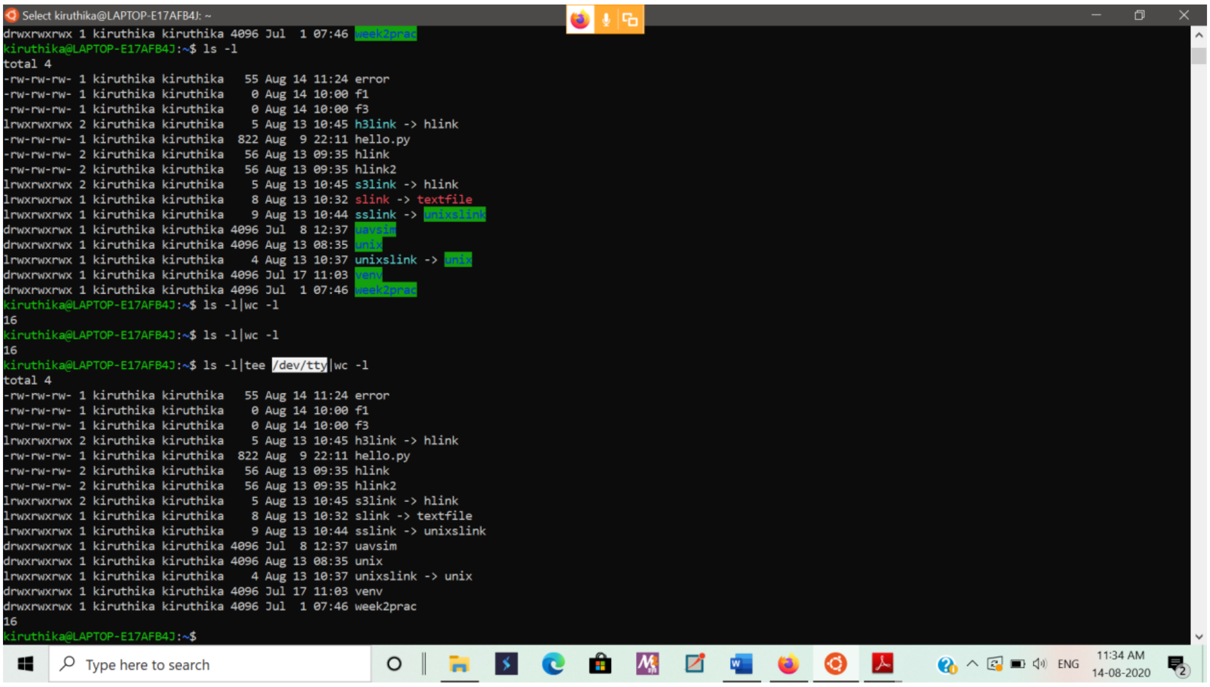
To begin, /dev/null is a special file called the **null device** in Unix systems. Colloquially it is also called the **bit-bucket** or the **blackhole** because it immediately discards anything written to it and only returns an end-of-file EOF when read.



```
kiruthika@LAPTOP-E17AFB4J:~$ mkdir unix 2> error
kiruthika@LAPTOP-E17AFB4J:~$ cat error
mkdir: cannot create directory ‘unix’: File exists
kiruthika@LAPTOP-E17AFB4J:~$ mkdir unix 2> /dev/null
kiruthika@LAPTOP-E17AFB4J:~$ cat /dev/null
kiruthika@LAPTOP-E17AFB4J:~$ fstat /dev/null




Command 'fstat' not found, but there are 18 similar ones.

kiruthika@LAPTOP-E17AFB4J:~$
kiruthika@LAPTOP-E17AFB4J:~$
kiruthika@LAPTOP-E17AFB4J:~$
kiruthika@LAPTOP-E17AFB4J:~$ stat /dev/null
  File: /dev/null
  Size: 0               Blocks: 0          IO Block: 4096   character special file
Device: 3h/3d   Inode: 17451448556087163  Links: 1     Device type: 1,3
Access: (0666/crw-rw-rw-)  Uid: (    0/    root)   Gid: (    0/    root)
Access: 2020-08-14 09:55:49.706545000 +0530
Modify: 2020-08-14 09:55:49.706545000 +0530
Change: 2020-08-14 09:55:49.706545000 +0530
 Birth: -
kiruthika@LAPTOP-E17AFB4J:~$
```

/dev/tty file

/dev/tty is a special file, representing the terminal for the current process.



```
drwxrwxrwx 1 kiruthika kiruthika 4096 Jul  1 07:46 week2prac
kiruthika@LAPTOP-E17AFB4J:~$ ls -l
total 4
-rw-rw-rw- 1 kiruthika kiruthika   55 Aug 14 11:24 error
-rw-rw-rw- 1 kiruthika kiruthika    0 Aug 14 10:00 f1
-rw-rw-rw- 1 kiruthika kiruthika    0 Aug 14 10:00 f3
lrwxrwxrwx 2 kiruthika kiruthika    5 Aug 13 10:45 h3link -> hlink
-rw-rw-rw- 1 kiruthika kiruthika  822 Aug  9 22:11 hello.py
-rw-rw-rw- 2 kiruthika kiruthika   56 Aug 13 09:35 hlink
-rw-rw-rw- 2 kiruthika kiruthika   56 Aug 13 09:35 hlink2
lrwxrwxrwx 2 kiruthika kiruthika    5 Aug 13 10:45 s3link -> hlink
lrwxrwxrwx 1 kiruthika kiruthika    8 Aug 13 10:32 slink -> textfile
lrwxrwxrwx 1 kiruthika kiruthika    9 Aug 13 10:44 sslink -> unixslink
drwxrwxrwx 1 kiruthika kiruthika 4096 Jul  8 12:37 uavsim
drwxrwxrwx 1 kiruthika kiruthika 4096 Aug 13 08:35 unix
lrwxrwxrwx 1 kiruthika kiruthika    4 Aug 13 10:37 unixslink -> unix
drwxrwxrwx 1 kiruthika kiruthika 4096 Jul 17 11:03 venv
drwxrwxrwx 1 kiruthika kiruthika 4096 Jul  1 07:46 week2prac
kiruthika@LAPTOP-E17AFB4J:~$ ls -l|wc -l
16
kiruthika@LAPTOP-E17AFB4J:~$ ls -l|wc -l
16
kiruthika@LAPTOP-E17AFB4J:~$ ls -l|tee /dev/tty|wc -l
total 4
-rw-rw-rw- 1 kiruthika kiruthika   55 Aug 14 11:24 error
-rw-rw-rw- 1 kiruthika kiruthika    0 Aug 14 10:00 f1
-rw-rw-rw- 1 kiruthika kiruthika    0 Aug 14 10:00 f3
lrwxrwxrwx 2 kiruthika kiruthika    5 Aug 13 10:45 h3link -> hlink
-rw-rw-rw- 1 kiruthika kiruthika  822 Aug  9 22:11 hello.py
-rw-rw-rw- 2 kiruthika kiruthika   56 Aug 13 09:35 hlink
-rw-rw-rw- 2 kiruthika kiruthika   56 Aug 13 09:35 hlink2
lrwxrwxrwx 2 kiruthika kiruthika    5 Aug 13 10:45 s3link -> hlink
lrwxrwxrwx 1 kiruthika kiruthika    8 Aug 13 10:32 slink -> textfile
lrwxrwxrwx 1 kiruthika kiruthika    9 Aug 13 10:44 sslink -> unixslink
drwxrwxrwx 1 kiruthika kiruthika 4096 Jul  8 12:37 uavsim
drwxrwxrwx 1 kiruthika kiruthika 4096 Aug 13 08:35 unix
lrwxrwxrwx 1 kiruthika kiruthika    4 Aug 13 10:37 unixslink -> unix
drwxrwxrwx 1 kiruthika kiruthika 4096 Jul 17 11:03 venv
drwxrwxrwx 1 kiruthika kiruthika 4096 Jul  1 07:46 week2prac
16
kiruthika@LAPTOP-E17AFB4J:~$
```

tee command

The tee command reads from the standard input and writes to both standard output and one or more files at the same time. tee is mostly used in combination with other commands through piping. (tee command used in above screenshot).

Shell Wildcards

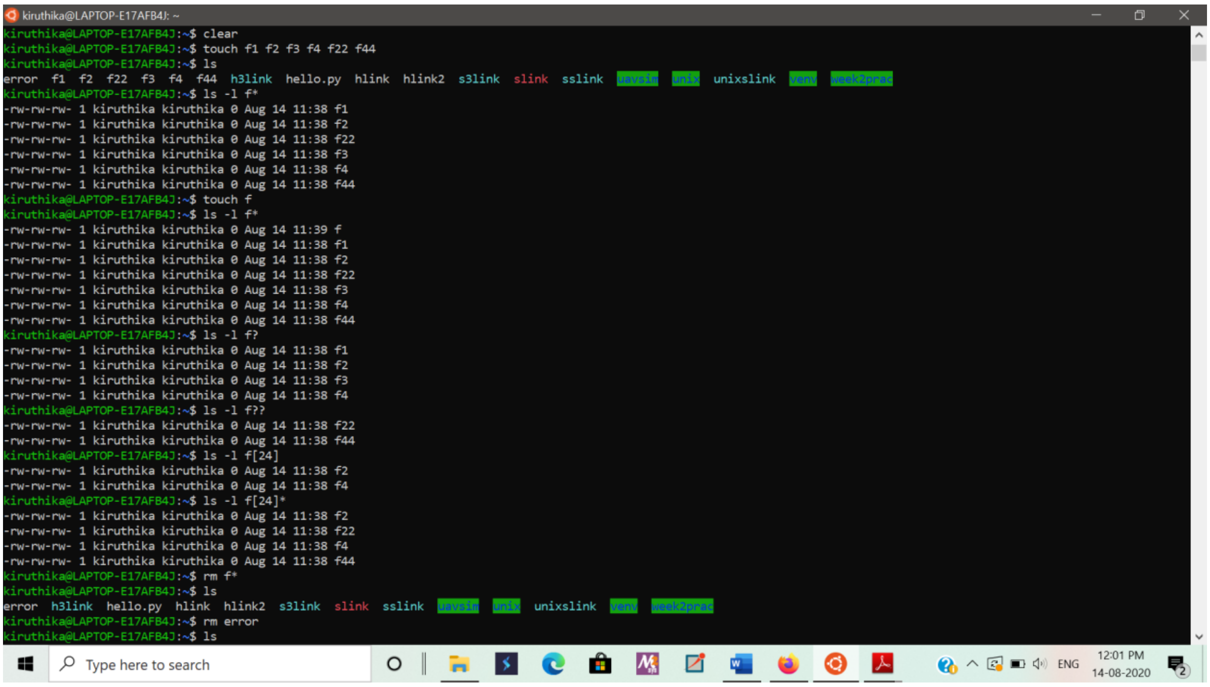
Wildcards (also referred to as meta characters) are symbols or special characters that represent other characters. You can use them with any command such as ls command or **rm command** to list or remove files matching a given criteria, receptively.

Also it can be used in **regular expressions**, popular with many other commands and popular for use with text searching and manipulation.

These wildcards are interpreted by the shell and the results are returned to the command you run. There are three main wildcards in Linux:

- An asterisk (*) – matches one or more occurrences of any character, including no character.
- Question mark (?) – represents or matches a single occurrence of any character.
- Bracketed characters ([ ]) – matches any occurrence of character enclosed in the square brackets. It is possible to use different types of characters (alphanumeric characters): numbers, letters, other special characters etc.

We need to carefully choose which wildcard to use to match correct filenames: it is also possible to combine all of them in one operation.



```
kiruthika@LAPTOP-E17AFB4J:~$ clear
kiruthika@LAPTOP-E17AFB4J:~$ touch f1 f2 f3 f4 f22 f44
kiruthika@LAPTOP-E17AFB4J:~$ ls
error  f1  f2  f22  f3  f4  f44  h3link  hello.py  hlink  hlink2  s3link  slink  sslink  uavsim  unix  unixslink  venv  week2prac
kiruthika@LAPTOP-E17AFB4J:~$ ls -l f*
-rw-rw-rw- 1 kiruthika kiruthika 0 Aug 14 11:38 f1
-rw-rw-rw- 1 kiruthika kiruthika 0 Aug 14 11:38 f2
-rw-rw-rw- 1 kiruthika kiruthika 0 Aug 14 11:38 f22
-rw-rw-rw- 1 kiruthika kiruthika 0 Aug 14 11:38 f3
-rw-rw-rw- 1 kiruthika kiruthika 0 Aug 14 11:38 f4
-rw-rw-rw- 1 kiruthika kiruthika 0 Aug 14 11:38 f44
kiruthika@LAPTOP-E17AFB4J:~$ touch f
kiruthika@LAPTOP-E17AFB4J:~$ ls -l f*
-rw-rw-rw- 1 kiruthika kiruthika 0 Aug 14 11:39 f
-rw-rw-rw- 1 kiruthika kiruthika 0 Aug 14 11:38 f1
-rw-rw-rw- 1 kiruthika kiruthika 0 Aug 14 11:38 f2
-rw-rw-rw- 1 kiruthika kiruthika 0 Aug 14 11:38 f22
-rw-rw-rw- 1 kiruthika kiruthika 0 Aug 14 11:38 f3
-rw-rw-rw- 1 kiruthika kiruthika 0 Aug 14 11:38 f4
-rw-rw-rw- 1 kiruthika kiruthika 0 Aug 14 11:38 f44
kiruthika@LAPTOP-E17AFB4J:~$ ls -l f?
-rw-rw-rw- 1 kiruthika kiruthika 0 Aug 14 11:38 f1
-rw-rw-rw- 1 kiruthika kiruthika 0 Aug 14 11:38 f2
-rw-rw-rw- 1 kiruthika kiruthika 0 Aug 14 11:38 f3
-rw-rw-rw- 1 kiruthika kiruthika 0 Aug 14 11:38 f4
kiruthika@LAPTOP-E17AFB4J:~$ ls -l f??
-rw-rw-rw- 1 kiruthika kiruthika 0 Aug 14 11:38 f22
-rw-rw-rw- 1 kiruthika kiruthika 0 Aug 14 11:38 f44
kiruthika@LAPTOP-E17AFB4J:~$ ls -l f[24]
-rw-rw-rw- 1 kiruthika kiruthika 0 Aug 14 11:38 f2
-rw-rw-rw- 1 kiruthika kiruthika 0 Aug 14 11:38 f4
kiruthika@LAPTOP-E17AFB4J:~$ ls -l f[24]*
-rw-rw-rw- 1 kiruthika kiruthika 0 Aug 14 11:38 f2
-rw-rw-rw- 1 kiruthika kiruthika 0 Aug 14 11:38 f22
-rw-rw-rw- 1 kiruthika kiruthika 0 Aug 14 11:38 f4
-rw-rw-rw- 1 kiruthika kiruthika 0 Aug 14 11:38 f44
kiruthika@LAPTOP-E17AFB4J:~$ rm f*
kiruthika@LAPTOP-E17AFB4J:~$ ls
error  h3link  hello.py  hlink  hlink2  s3link  slink  sslink  uavsim  unix  unixslink  venv  week2prac
kiruthika@LAPTOP-E17AFB4J:~$ rm error
kiruthika@LAPTOP-E17AFB4J:~$ ls
```

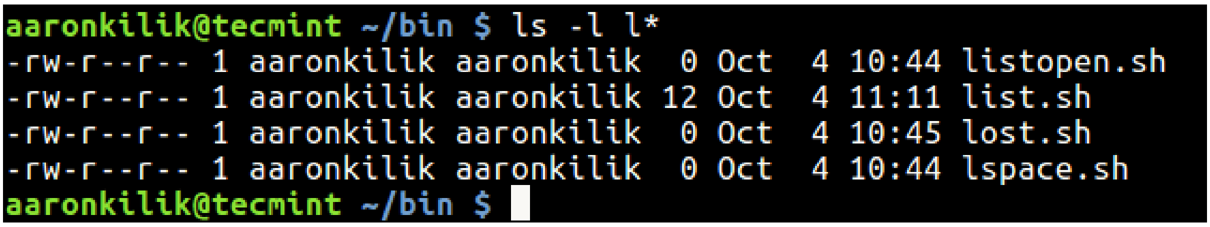
How to Match Filenames Using Wildcards

For the purpose of demonstration, create only below files in a folder and make it your current working directory.

```
createbackup.sh  list.sh  lspace.sh       speaker.sh
listopen.sh      lost.sh  rename-files.sh  topprocs.sh
```

1. This command matches all files with names starting with l (which is the prefix) and ending with one or more occurrences of any character.

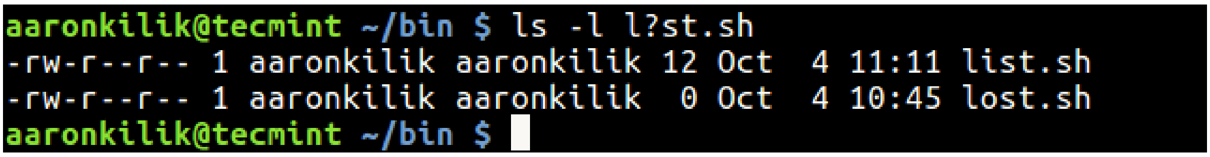
```
$ ls -l l*
```



```
aaronkilik@tecmint ~/bin $ ls -l l*
-rw-r--r-- 1 aaronkilik aaronkilik  0 Oct  4 10:44 listopen.sh
-rw-r--r-- 1 aaronkilik aaronkilik 12 Oct  4 11:11 list.sh
-rw-r--r-- 1 aaronkilik aaronkilik  0 Oct  4 10:45 lost.sh
-rw-r--r-- 1 aaronkilik aaronkilik  0 Oct  4 10:44 lspace.sh
aaronkilik@tecmint ~/bin $ 
```

2. The following command matches all files with names beginning with l followed by any single character and ending with st.sh (which is the suffix).

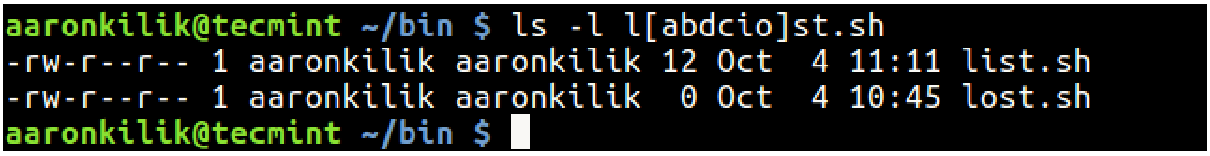
```
$ ls l?st.sh
```



```
aaronkilik@tecmint ~/bin $ ls -l l?st.sh
-rw-r--r-- 1 aaronkilik aaronkilik 12 Oct  4 11:11 list.sh
-rw-r--r-- 1 aaronkilik aaronkilik  0 Oct  4 10:45 lost.sh
aaronkilik@tecmint ~/bin $ 
```

3. The command below matches all files with names starting with l followed by any of the characters in the square bracket but ending with st.sh.

```
$ ls l[abdcio]st.sh
```



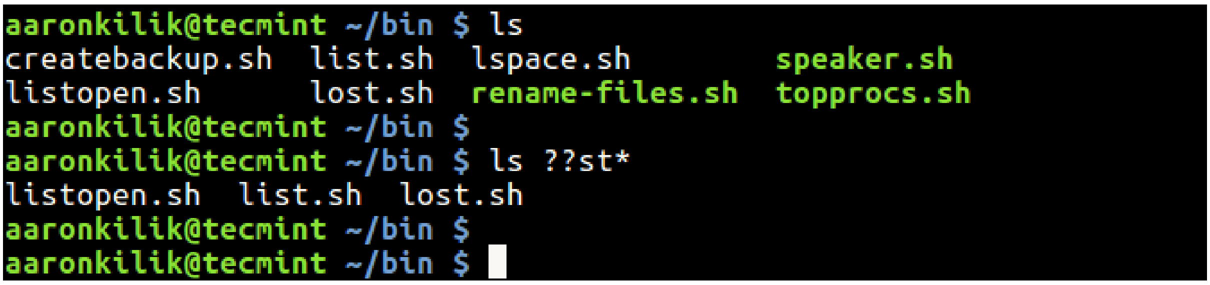
```
aaronkilik@tecmint ~/bin $ ls -l l[abdcio]st.sh
-rw-r--r-- 1 aaronkilik aaronkilik 12 Oct  4 11:11 list.sh
-rw-r--r-- 1 aaronkilik aaronkilik  0 Oct  4 10:45 lost.sh
aaronkilik@tecmint ~/bin $ 
```

How to Combine Wildcards to Match Filenames

You can combine wildcards to build a complex filename matching criteria as described in the following examples.

4. This command will match all filenames prefixed with any two characters followed by st but ending with one or more occurrence of any character.

```
$ ls
$ ls ??st*
```

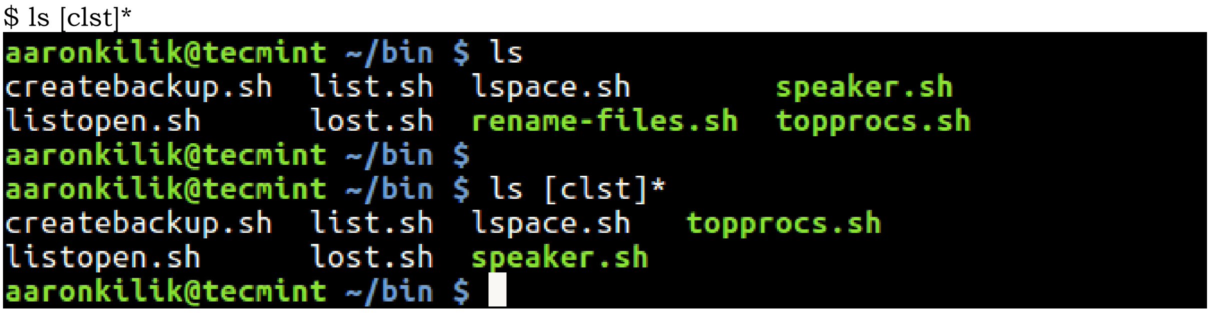


```
aaronkilik@tecmint ~/bin $ ls
createbackup.sh  list.sh  lspace.sh        speaker.sh
listopen.sh      lost.sh  rename-files.sh  topprocs.sh
aaronkilik@tecmint ~/bin $
aaronkilik@tecmint ~/bin $ ls ??st*
listopen.sh  list.sh  lost.sh
aaronkilik@tecmint ~/bin $
aaronkilik@tecmint ~/bin $ 
```

Match File Names with Prefix

5. This example matches filenames starting with any of these characters [clst] and ending with one or more occurrence of any character.

$ ls
$ ls [clst]*

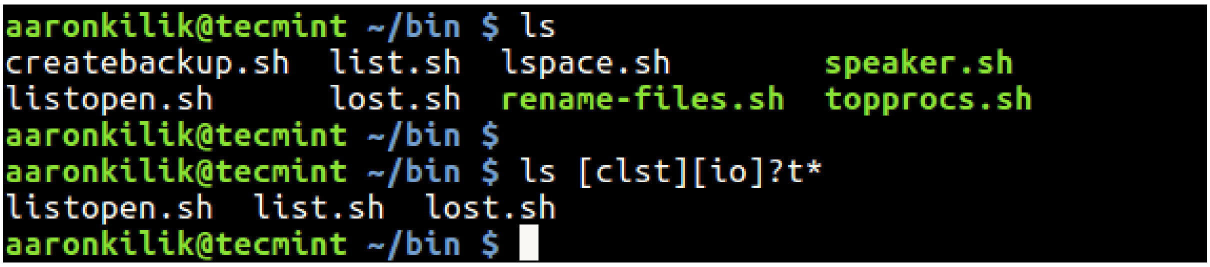


```
aaronkilik@tecmint ~/bin $ ls
createbackup.sh  list.sh  lspace.sh        speaker.sh
listopen.sh      lost.sh  rename-files.sh  topprocs.sh
aaronkilik@tecmint ~/bin $
aaronkilik@tecmint ~/bin $ ls [clst]*
createbackup.sh  list.sh  lspace.sh   topprocs.sh
listopen.sh      lost.sh  speaker.sh
aaronkilik@tecmint ~/bin $ 
```

Match Files with Characters

6. In this example, only filenames starting with any of these characters [clst] followed by one of these [io] and then any single character, followed by a t and lastly, one or more occurrence of any character will be listed.

$ ls
$ ls [clst][io]?t*

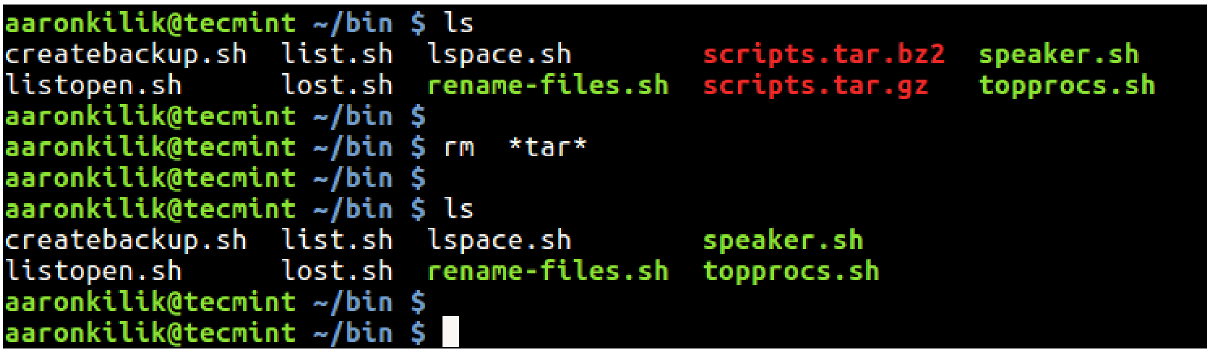


```
aaronkilik@tecmint ~/bin $ ls
createbackup.sh  list.sh  lspace.sh        speaker.sh
listopen.sh      lost.sh  rename-files.sh  topprocs.sh
aaronkilik@tecmint ~/bin $
aaronkilik@tecmint ~/bin $ ls [clst][io]?t*
listopen.sh  list.sh  lost.sh
aaronkilik@tecmint ~/bin $ 
```

List Files with Multiple Characters

7. Here, filenames prefixed with one or more occurrence of any character, followed by the letters tar and ending with one or more occurrence of any character will be removed.

$ ls
$ rm *tar*
$ ls



```
aaronkilik@tecmint ~/bin $ ls
createbackup.sh  list.sh  lspace.sh        scripts.tar.bz2  speaker.sh
listopen.sh      lost.sh  rename-files.sh  scripts.tar.gz   topprocs.sh
aaronkilik@tecmint ~/bin $
aaronkilik@tecmint ~/bin $ rm  *tar*
aaronkilik@tecmint ~/bin $
aaronkilik@tecmint ~/bin $ ls
createbackup.sh  list.sh  lspace.sh        speaker.sh
listopen.sh      lost.sh  rename-files.sh  topprocs.sh
aaronkilik@tecmint ~/bin $
aaronkilik@tecmint ~/bin $
```

8. How to Match Characters Set

For the purpose of demonstration, create only below files in another folder and make it your current working directory.

```
$ ls
```

```
users-111.list  users-1AA.list  users-22A.list  users-2aB.txt   users-2ba.txt
users-111.txt   users-1AA.txt   users-22A.txt   users-2AB.txt   users-2bA.txt
users-11A.txt   users-1AB.list  users-2aA.txt   users-2ba.list
users-12A.txt   users-1AB.txt   users-2AB.list  users-2bA.list
```

This below command will match all files whose name starts with users-, followed by a number, a lower case letter or number, then a number and ends with one or more occurrences of any character.

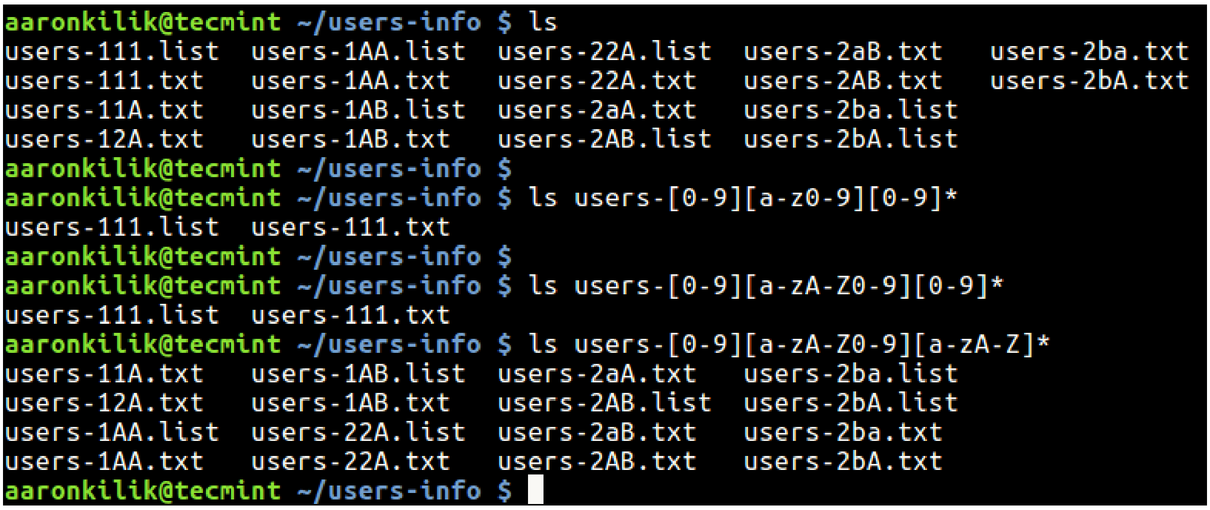
```
$ ls users-[0-9][a-z0-9][0-9]*
```

The below command matches filenames beginning with users-, followed by a number, a lower or upper case letter or number, then a number and ends with one or more occurrences of any character.

```
$ ls users-[0-9][a-zA-Z0-9][0-9]*
```

This below command will match all filenames beginning with users-, followed by a number, a lower or upper case letter or number, then a lower or upper case letter and ends with one or more occurrences of any character.

```
$ ls users-[0-9][a-zA-Z0-9][a-zA-Z]*
```



```
aaronkilik@tecmint ~/users-info $ ls
users-111.list  users-1AA.list  users-22A.list  users-2aB.txt   users-2ba.txt
users-111.txt   users-1AA.txt   users-22A.txt   users-2AB.txt   users-2bA.txt
users-11A.txt   users-1AB.list  users-2aA.txt   users-2ba.list
users-12A.txt   users-1AB.txt   users-2AB.list  users-2bA.list
aaronkilik@tecmint ~/users-info $
aaronkilik@tecmint ~/users-info $ ls users-[0-9][a-z0-9][0-9]*
users-111.list  users-111.txt
aaronkilik@tecmint ~/users-info $
aaronkilik@tecmint ~/users-info $ ls users-[0-9][a-zA-Z0-9][0-9]*
users-111.list  users-111.txt
aaronkilik@tecmint ~/users-info $ ls users-[0-9][a-zA-Z0-9][a-zA-Z]*
users-11A.txt   users-1AB.list  users-2aA.txt   users-2ba.list
users-12A.txt   users-1AB.txt   users-2AB.list  users-2bA.list
users-1AA.list  users-22A.list  users-2aB.txt   users-2ba.txt
users-1AA.txt   users-22A.txt   users-2AB.txt   users-2bA.txt
aaronkilik@tecmint ~/users-info $
```

How to Negate a Set of Characters in Linux

9. You can as well negate a set of characters using the ! symbol. The following command lists all filenames starting with users-i, followed by a number, any valid file naming character apart from a number, then a lower or upper case letter and ends with one or more occurrences of any character.

```
$ ls users-[0-9][!0-9][a-zA-Z]*
```