Linux Cat Command Usage with Examples

Definition:

- The cat command (short for “concatenate “) is one of the most frequently used command in Linux/Unix, Apple Mac OS X operating systems.
- cat command allows us to create single or multiple files, view contain of file, concatenate files and redirect output in terminal or files.
- It is a standard Unix program used to concatenate and display files.
- The cat command display file contents to a screen. Cat command concatenate FILE(s), or standard input, to standard output.
- With no FILE, or when FILE is -, it reads standard input. Also, you can use cat command for quickly creating a file.
- The cat command can read and write data from standard input and output devices.
- It has three main functions related to manipulating text files:

    1.creating them,

    2. displaying them

    3. combining them.

Syntax:

The basic syntax of **cat** command is as follows:

$ cat filename

OR

$ cat > filename

OR

$ cat [options] filename

Options

Tag	Description
-A, --show-all	equivalent to -vET
-b, --number-nonblank	number nonblank output lines
-e	equivalent to -vE
-E, --show-ends	display $ at end of each line
-n, --number	number all output lines
-s, --squeeze-blank	never more than one single blank line
-t	equivalent to -vT
-T,--show-tabs	display TAB characters as ^I
-u	(ignored)
-v, --show-nonprinting	use ^ and M- notation, except for LFD and TAB. display this help and exit
--help	display this help and exit
--version	output version information and exit

Examples

1. Display Contents of File

In the below example, it will show contents of /etc/passwd file.

cat /etc/passwd

root:x:0:0:root:/root:/bin/bash

bin:x:1:1:bin:/bin:/sbin/nologin

narad:x:500:500::/home/narad:/bin/bash

2. View Contents of Multiple Files in terminal

In below example, it will display contents of test and test1 file in terminal.

 # cat test test1

Hello everybody

Hi world,

3. Create a File with Cat Command

We will create a file called test2 file with below command.

 # cat >test2

Awaits input from user, type desired text and press CTRL+D (hold down Ctrl Key and type ‘d‘) to exit. The text will be written in test2 file. You can see content of file with following cat command.

 # cat test2

hello everyone, how do you do?

4. Use Cat Command with More & Less Options

If file having large number of content that won’t fit in output terminal and screen scrolls up very fast, we can use parameters more and less with cat command as show above.

 # cat song.txt | more

 # cat song.txt | less

5. Display Line Numbers in File

With -n option you could see the line numbers of a file song.txt in the output terminal.

cat -n song.txt

1 "Heal The World"

2 There's A Place In

3 Your Heart

4 And I Know That It Is Love

5 And This Place Could

6 Be Much

7 Brighter Than Tomorrow

8 And If You Really Try

9 You'll Find There's No Need

10 To Cry

11 In This Place You'll Feel

12 There's No Hurt Or Sorrow

6. Display $ at the End of File

In the below, you can see with -e option that '$' is shows at the end of line and also in space showing '$' if there is any gap between paragraphs. This options is useful to squeeze multiple lines in a single line.

 # cat -e test

hello everyone, how do you do?$

$

Hey, am fine.$

How's your training going on?$

$

7. Display Tab separated Lines in File

In the below output, we could see TAB space is filled up with '^I' character.

 # cat -T test

hello ^Ieveryone, how do you do?

Hey, ^Iam fine.

^I^IHow's your training ^Igoing on?

Let's do ^Isome practice in Linux.

8. Display Multiple Files at Once

In the below example we have three files test, test1 and test2 and able to view the contents of those file as shown above. We need to separate each file with ; (semi colon).

 # cat test; cat test1; cat test2

This is test file

This is test1 file.

This is test2 file.

9. Use Standard Output with Redirection Operator

We can redirect standard output of a file into a new file else existing file with '>' (greater than) symbol. Careful, existing contents of test1 will be overwritten by contents of test file.

 # cat test > test1

10. Appending Standard Output with Redirection Operator

Appends in existing file with '>>' (double greater than) symbol. Here, contents of test file will be appended at the end of test1 file.

 # cat test >> test1

11. Redirecting Standard Input with Redirection Operator

When you use the redirect with standard input '<' (less than symbol), it use file name test2 as a input for a command and output will be shown in a terminal.

 # cat < test2

This is test2 file.

12. Redirecting Multiple Files Contain in a Single File

This will create a file called test3 and all output will be redirected in a newly created file.

 # cat test test1 test2 > test3

13. Sorting Contents of Multiple Files in a Single File

This will create a file test4 and output of cat command is piped to sort and result will be redirected in a newly created file.

 # cat test test1 test2 test3 | sort > test4

Usage

The Single Unix Specification defines the operation of cat to read files in the sequence given in its arguments, writing their contents to the standard output in the same sequence. The specification mandates the support of one option flag, u for unbuffered output, meaning that each byte is written after it has been read. Some operating systems, like the ones using GNU Core Utilities, do this by default and ignore .

If one of the input filenames is specified as a single hyphen (-), then cat reads from standard input at that point in the sequence. If no files are specified, cat reads from standard input only.

THE GREP COMMAND

Grep is an acronym that stands for **G**lobal **R**egular **E**xpression **P**rint.

- The grep filter searches a file for a particular pattern of characters, and displays all lines that contain that pattern.
- The pattern that is searched in the file is referred to as the regular expression (grep stands for globally search for regular expression and print out).

INSTALLING GREP

Grep comes installed in most of the Linux distributions. However in case, it is missing from your system, you can install it using the following method in Terminal:

$ sudo apt-get install grep

Syntax:

grep [options] pattern [files]

Options	Description
-c :	This prints only a count of the lines that match a pattern
-h :	Display the matched lines, but do not display the filenames.
-i :	Ignores, case for matching
-l :	Displays list of a filenames only.
-n :	Display the matched lines and their line numbers.
-v :	This prints out all the lines that do not matches the pattern
-e exp :	Specifies expression with this option. Can use multiple times.
-f file :	Takes patterns from file, one per line.

-E : Treats pattern as an extended regular expression (ERE)

-w : Match whole word

-o : Print only the matched parts of a matching line.

Examples

Consider the below file as an input.

$cat > file.txt

OUTPUT:

unix is great os. unix is opensource. unix is free os.

learn operating system.

Unix linux which one you choose.

uNix is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful.

1. Case insensitive search :

The -i option enables to search for a string case insensitively in the give file. It matches the words like “UNIX”, “Unix”, “unix”.

$grep -i "UNix" file.txt

Output:

unix is great os. unix is opensource. unix is free os.

Unix linux which one you choose.

uNix is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful.

2. Displaying the count of number of matches :

We can find the number of lines that matches the given string/pattern

```
$grep -c "unix" file.txt
```

Output:

2

3. Display the file names that matches the pattern :

We can just display the files that contains the given string/pattern.

```
$grep -l "unix" *
```

or

```
$grep -l "unix" f1.txt f2.txt f3.xt f4.txt
```

Output:

file.txt

4. Checking for the whole words in a file :

By default, grep matches the given string/pattern even if it found as a substring in a file. The -w option to grep makes it match only the whole words.

```
$ grep -w "unix" file.txt
```

Output:

unix is great os. unix is opensource. unix is free os.

uNix is easy to learn.unix is a multiuser

5. Displaying only the matched pattern :

By default, grep displays the entire line which has the matched string. We can make the grep to display only the matched string by using the -o option.

```
$ grep -o "unix" file.txt
```

Output:

unix

unix

unix

unix

unix

6. Show line number while displaying the output using grep -n : To show the line number of file with the line matched.

```
$ grep -n "unix" file.txt
```

Output:

1:unix is great os. unix is opensource. unix is free os.

4:uNix is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful.

7. Inverting the pattern match : You can display the lines that are not matched with the specified search sting pattern using the -v option.

$ grep -v "unix" file.txt

Output:

learn operating system.

Unix linux which one you choose.

8. Matching the lines that start with a string :

The ^ regular expression pattern specifies the start of a line. This can be used in grep to match the lines which start with the given string or pattern.

$ grep "^unix" file.txt

Output:

unix is great os. unix is opensource. unix is free os.

9. Matching the lines that end with a string :

The $ regular expression pattern specifies the end of a line. This can be used in grep to match the lines which end with the given string or pattern.

$ grep "os$" file.txt

10.Specifies expression with -e option.:

Can use multiple times

$grep –e "Agarwal" –e "Aggarwal" –e "Agrawal" file.txt

11. -f file option Takes patterns from file, one per line.

$cat pattern.txt

Output:

Agarwal

Aggarwal

Agrawal

$grep –f pattern.txt file.txt

USE OF GREP

- By default, grep displays the matching lines.
- Use grep to search for lines of text that match one or many regular expressions, and outputs only the matching lines.
- Grep is considered to be one of the most useful commands on Linux and Unix-like operating systems.
- grep is a powerful file pattern searcher in Linux.
- If it is not installed on your system, you can easily install it via your package manager (apt-get on Debian/Ubuntu and yum on RHEL/CentOS/Fedora)
- grep searches the named input FILEs (or standard input if no files are named, or if a single hyphen-minus (-) is given as file name) for lines containing a match to the given PATTERN.

- By default, grep prints the matching lines. In addition, three variant programs egrep, fgrep and rgrep are available.
- egrep is the same as grep -E. fgrep is the same as grep -F. rgrep is the same as grep -r.
- Direct invocation as either egrep or fgrep is deprecated, but is provided to allow historical applications that rely on them to run unmodified.
- The grep has no limits on input line length other than available memory, and it can match arbitrary characters within a line.
- If the final byte of an input file is not a newline, grep silently supplies one. Since newline is also a separator for the list of patterns, there is no way to match newline characters in a text.