**10 Practical Linux Cut Command Examples to Select File Columns**

by Balakrishnan Mariyappan on June 6, 2013

Linux command cut is used for text processing. You can use this command to extract portion of text from a file by selecting columns.

This tutorial provides few practical examples of cut command that you can use in your day to day command line activities.

For most of the example, we’ll be using the following test file.

$ cat test.txt

cat command for file oriented operations.

cp command for copy files or directories.

ls command to list out files and directories with its attributes.

**1. Select Column of Characters**

To extract only a desired column from a file use -c option. The following example displays 2nd character from each line of a file test.txt

$ cut -c2 test.txt

a

p

s

As seen above, the characters a, p, s are the second character from each line of the test.txt file.

**2. Select Column of Characters using Range**

Range of characters can also be extracted from a file by specifying start and end position delimited with -. The following example extracts first 3 characters of each line from a file called test.txt

$ cut -c1-3 test.txt

cat

cp

ls

**3. Select Column of Characters using either Start or End Position**

Either start position or end position can be passed to cut command with -c option.

The following specifies only the start position before the ‘-’. This example extracts from 3rd character to end of each line from test.txt file.

$ cut -c3- test.txt

t command for file oriented operations.

command for copy files or directories.

command to list out files and directories with its attributes.

The following specifies only the end position after the ‘-’. This example extracts 8 characters from the beginning of each line from test.txt file.

$ cut -c-8 test.txt

cat comm

cp comma

ls comma

The entire line would get printed when you don’t specify a number before or after the ‘-’ as shown below.

$ cut -c- test.txt

cat command for file oriented operations.

cp command for copy files or directories.

ls command to list out files and directories with its attributes.

**4. Select a Specific Field from a File**

Instead of selecting x number of characters, if you like to extract a whole field, you can combine option -f and -d. The option -f specifies which field you want to extract, and the option -d specifies what is the field delimiter that is used in the input file.

The following example displays only first field of each lines from /etc/passwd file using the field delimiter : (colon). In this case, the 1st field is the username. The file

$ cut -d':' -f1 /etc/passwd

root

daemon

bin

sys

sync

games

bala

**5. Select Multiple Fields from a File**

You can also extract more than one fields from a file or stdout. Below example displays username and home directory of users who has the login shell as “/bin/bash”.

$ grep "/bin/bash" /etc/passwd | cut -d':' -f1,6

root:/root

bala:/home/bala

To display the range of fields specify start field and end field as shown below. In this example, we are selecting field 1 through 4, 6 and 7

$ grep "/bin/bash" /etc/passwd | cut -d':' -f1-4,6,7

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

bala:x:1000:1000:/home/bala:/bin/bash

**6. Select Fields Only When a Line Contains the Delimiter**

In our /etc/passwd example, if you pass a different delimiter other than : (colon), cut will just display the whole line.

In the following example, we’ve specified the delimiter as | (pipe), and cut command simply displays the whole line, even when it doesn’t find any line that has | (pipe) as delimiter.

$ grep "/bin/bash" /etc/passwd | cut -d'|' -f1

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

bala:x:1000:1000:bala,,,:/home/bala:/bin/bash

But, it is possible to filter and display only the lines that contains the specified delimiter using -s option.

The following example doesn’t display any output, as the cut command didn’t find any lines that has | (pipe) as delimiter in the /etc/passwd file.

$ grep "/bin/bash" /etc/passwd | cut -d'|' -s -f1

**7. Select All Fields Except the Specified Fields**

In order to complement the selection field list use option –complement.

The following example displays all the fields from /etc/passwd file except field 7

$ grep "/bin/bash" /etc/passwd | cut -d':' --complement -s -f7

root:x:0:0:root:/root

bala:x:1000:1000:bala,,,:/home/bala

**8. Change Output Delimiter for Display**

By default the output delimiter is same as input delimiter that we specify in the cut -d option.

To change the output delimiter use the option –output-delimiter as shown below. In this example, the input delimiter is : (colon), but the output delimiter is # (hash).

$ grep "/bin/bash" /etc/passwd | cut -d':' -s -f1,6,7 --output-delimiter='#'

root#/root#/bin/bash

bala#/home/bala#/bin/bash

**9. Change Output Delimiter to Newline**

In this example, each and every field of the cut command output is displayed in a separate line. We still used –output-delimiter, but the value is $’\n’ which indicates that we should add a newline as the output delimiter.

$ grep bala /etc/passwd | cut -d':' -f1,6,7 --output-delimiter=$'\n'

bala

/home/bala

/bin/bash

**10. Combine Cut with Other Unix Command Output**

The power of cut command can be realized when you combine it with the stdout of some other Unix command.

Once you master the basic usage of cut command that we’ve explained above, you can wisely use cut command to solve lot of your text manipulation requirements.

The following example indicates how you can extract only useful information from the [ps command](http://www.thegeekstuff.com/2011/04/ps-command-examples/) output. We also showed how we’ve filtered the output of ps command using grep and sed before the final output was given to cut command. Here, we’ve used cut option -d and -f which we’ve explained in the above examples.

$ ps axu | grep python | sed 's/\s\+/ /g' | cut -d' ' -f2,11-

2231 /usr/bin/python /usr/lib/unity-lens-video/unity-lens-video

2311 /usr/bin/python /usr/lib/unity-scope-video-remote/unity-scope-video-remote

2414 /usr/bin/python /usr/lib/ubuntuone-client/ubuntuone-syncdaemon

2463 /usr/bin/python /usr/lib/system-service/system-service-d

3274 grep --color=auto python