Cut Command in Linux

Cut command in unix (or linux) is used to select sections of text from each line of files. You can use the cut command to select fields or columns from a line by specifying a delimiter or you can select a portion of text by specifying the range or characters. Basically the cut command slices a line and extracts the text.

Unix Cut Command Example

We will see the usage of cut command by considering the below text file as an example

```
> cat file.txt
unix or linux os
is unix good os
is linux good os
```

1. Write a unix/linux cut command to print characters by position?

The cut command can be used to print characters in a line by specifying the position of the characters. To print the characters in a line, use the -c option in cut command

```
cut -c4 file.txt
x
u
l
```

The above cut command prints the fourth character in each line of the file. You can print more than one character at a time by specifying the character positions in a comma separated list as shown in the below example

```
cut -c4,6 file.txt
xo
ui
ln
```

This command prints the fourth and sixth character in each line.

2.Write a unix/linux cut command to print characters by range?

You can print a range of characters in a line by specifying the start and end position of the characters.

```
cut -c4-7 file.txt
x or
unix
linu
```

The above cut command prints the characters from fourth position to the seventh position in each line. To print the first six characters in a line, omit the start position and specify only the end position.

```
cut -c-6 file.txt
unix o
is uni
is lin
```

To print the characters from tenth position to the end, specify only the start position and omit the end position.

```
cut -c10- file.txt
inux os
ood os
good os
```

If you omit the start and end positions, then the cut command prints the entire line.

```
cut -c- file.txt
```

3.Write a unix/linux cut command to print the fields using the delimiter?

You can use the cut command just as awk command to extract the fields in a file using a delimiter. The -d option in cut command can be used to specify the delimiter and -f option is used to specify the field position.

```
cut -d' ' -f2 file.txt
or
unix
linux
```

This command prints the second field in each line by treating the space as delimiter. You can print more than one field by specifying the position of the fields in a comma delimited list.

```
cut -d' ' -f2,3 file.txt
or linux
unix good
linux good
```

The above command prints the second and third field in each line.

Note: If the delimiter you specified is not exists in the line, then the cut command prints the entire line. To suppress these lines use the -s option in cut command.

4. Write a unix/linux cut command to display range of fields?

You can print a range of fields by specifying the start and end position.

```
cut -d' ' -f1-3 file.txt
```

The above command prints the first, second and third fields. To print the first three fields, you can ignore the start position and specify only the end position.

```
cut -d' ' -f-3 file.txt
```

To print the fields from second fields to last field, you can omit the last field position.

```
cut -d' ' -f2- file.txt
```

5. Write a unix/linux cut command to display the first field from /etc/passwd file?

The /etc/passwd is a delimited file and the delimiter is a colon (:). The cut command to display the first field in /etc/passwd file is

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

6. The input file contains the below text

```
> cat filenames.txt
logfile.dat
sum.pl
add_int.sh
```

Using the cut command extract the portion after the dot.

First reverse the text in each line and then apply the command on it.

```
rev filenames.txt | cut -d'.' -f1
```

Paste Command in Linux

In this article, we will see how to use the **paste** command with some examples. paste command, by definition of man page, is used to merge lines of files. It is very useful for merging a single file and also for merging set of files as well. This article is divided into 2 parts:

- paste command examples for single file handling
- paste command examples for multiple files handling

Let us consider a file with the sample contents as below:

```
$ cat file1
Linux
Unix
Solaris
HPUX
AIX
```

paste command with a single file:

1. paste command without any options is as good as the cat command when operated on a single file.

```
$ paste file1
Linux
Unix
Solaris
HPUX
AIX
```

2. Join all lines in a file:

```
$ paste -s file1
Linux   Unix    Solaris HPUX    AIX
```

-s option of paste joins all the lines in a file. Since no delimiter is specified, default delimiter tab is used to separate the columns.

3. Join all lines using the comma delimiter:

```
$ paste -d, -s file1
Linux,Unix,Solaris,HPUX,AIX
```

-d option is used to specify the delimiter. Using this -d and -s combination, all the lines in the file get merged into a single line.

4. Merge a file by pasting the data into 2 columns:

```
$ paste - - < file1
Linux   Unix
Solaris HPUX
AIX
```

The '-' reads a line from the standard input. Two '-' reads 2 lines and pastes them side by side.

5.Merge a file by pasting the data into 2 columns using a colon separator:

```
$ paste -d':' - - < file1
Linux:Unix
Solaris:HPUX
AIX:
```

This is same as joining every 2 lines in a file.

6. Merge a file by pasting the file contents into 3 columns:

```
$ paste - - - < file1
Linux   Unix    Solaris
HPUX    AIX
```

7. Merge a file into 3 columns using 2 different delimiters:

```
$ paste -d ':,' - - - < file1
Linux:Unix,Solaris
HPUX:AIX,
```

The -d option can take multiple de-limiters. The 1st and 2nd columns is separated by ':', whereas the 2nd and 3rd are separated by a ','.

paste command with multiple files:

Let us consider a file, file2, with the following contents:

```
$ cat file2
Suse
Fedora
CentOS
OEL
Ubuntu
```

8. paste contents of 2 files side by side.

```
$ paste file1 file2
Linux   Suse
Unix    Fedora
Solaris CentOS
HPUX    OEL
AIX     Ubuntu
```

paste command is used in scenarios to merge multiple files side by side. As shown above, the file contents are pasted side by side.

9. paste contents of 2 files side by side with a comma separator:

```
$ paste -d, file1 file2
Linux,Suse
Unix,Fedora
Solaris,CentOS
HPUX,OEL
AIX,Ubuntu
```

10. paste command can take standard input in case of multiple files too:

```
$ cat file2 | paste -d, file1 -
Linux,Suse
Unix,Fedora
Solaris,CentOS
HPUX,OEL
AIX,Ubuntu
```

Like this as well:

```
$ cat file1 | paste -d, - file2
Linux,Suse
Unix,Fedora
Solaris,CentOS
HPUX,OEL
AIX,Ubuntu
```

One more:

```
$ cat file1 file2 | paste -d, - -
Linux,Unix
Solaris,HPUX
AIX,Suse
Fedora,CentOS
OEL,Ubuntu
```

11. Read lines in both the files alternatively:

```
$ paste -d'\n' file1 file2
Linux
Suse
Unix
Fedora
Solaris
CentOS
HPUX
OEL
AIX
Ubuntu
```

Using the newline character as the delimiter, we can read 2 files line by line alternatively.