cut command in Linux with examples

The cut command in UNIX is a command for cutting out the sections from each line of files and writing the result to standard output. It can be used to cut parts of a line by **byte position, character and field**. Basically the cut command slices a line and extracts the text. It is necessary to specify option with command otherwise it gives error. If more than one file name is provided then data from each file is **not precedes** by its file name.

Syntax:

```
cut OPTION... [FILE]...
```

Let us consider two files having name **state.txt** and **capital.txt** contains 5 names of the Indian states and capitals respectively.

```
$ cat state.txt
Andhra Pradesh
Arunachal Pradesh
Assam
Bihar
Chhattisgarh
```

Without any option specified it displays error.

```
$ cut state.txt
cut: you must specify a list of bytes, characters, or fields
Try 'cut --help' for more information.
```

Options and their Description with examples:

1. -b(byte): To extract the specific bytes, you need to follow -b option with the list of byte numbers separated by comma. Range of bytes can also be specified using the hyphen(-). It is necessary to specify list of byte numbers otherwise it gives error. **Tabs and backspaces** are treated like as a character of 1 byte.

```
List without ranges
$ cut -b 1,2,3 state.txt
And
Aru
Ass
Bih
Chh

List with ranges
$ cut -b 1-3,5-7 state.txt
Andra
Aruach
Assm
Bihr
Chhtti
```

It uses a special form for selecting bytes from beginning upto the end of the line:

```
In this, 1- indicate from 1st byte to end byte of a line
$ cut -b 1- state.txt
Andhra Pradesh
Arunachal Pradesh
Assam
Bihar
Chhattisgarh

In this, -3 indicate from 1st byte to 3rd byte of a line
$ cut -b -3 state.txt
And
```

```
Bih
Chh
```

2. -c (column): To cut by character use the -c option. This selects the characters given to the -c option. This can be a list of numbers separated comma or a range of numbers separated by hyphen(-). **Tabs and backspaces** are treated as a character. It is necessary to specify list of character numbers otherwise it gives error with this option.

Syntax:

```
$cut -c [(k)-(n)/(k),(n)/(n)] filename
```

Here,**k** denotes the starting position of the character and **n** denotes the ending position of the character in each line, if *k* and *n* are separated by "-" otherwise they are only the position of character in each line from the file taken as an input.

```
$ cut -c 2,5,7 state.txt
nr
rah
sm
ir
hti
```

Above cut command prints second, fifth and seventh character from each line of the file.

```
$ cut -c 1-7 state.txt
Andhra
Arunach
Assam
Bihar
Chhatti
```

Above cut command prints first seven characters of each line from the file.

Cut uses a special form for selecting characters from beginning upto the end of the line:

```
$ cut -c 1- state.txt
Andhra Pradesh
Arunachal Pradesh
Assam
Bihar
```

```
Above command prints starting from first character to end. Here in command
only starting
position is specified and the ending position is omitted.
```

```
$ cut -c -5 state.txt
Andhr
Aruna
Assam
Bihar
Chhat
```

```
Above command prints starting position to the fifth character. Here the
starting position
is omitted and the ending position is specified.
```

3. -f (field): -c option is useful for fixed-length lines. Most unix files doesn't have fixed-length lines. To extract the useful information you need to cut by fields rather than columns. List of the fields number specified must be separated by comma. *Ranges are not described with -f option*. **cut** uses **tab** as a default field delimiter but can also work with other delimiter by using **-d** option.
Note: Space is not considered as delimiter in UNIX.

Syntax:

```
$cut -d "delimiter" -f (field number) file.txt
```

Like in the file **state.txt** fields are separated by space if -d option is not used then it prints whole line:

```
$ cut -f 1 state.txt
Andhra Pradesh
Arunachal Pradesh
Assam
Bihar
Chhattisgarh
```

If -d option is used then it considered space as a field separator or delimiter:

```
$ cut -d " " -f 1 state.txt
Andhra
```

```
Bihar
Chhattisgarh
```

Command prints field from first to fourth of each line from the file.

Command:

```
$ cut -d " " -f 1-4 state.txt
```

Output:

```
Andhra Pradesh
Arunachal Pradesh
Assam
Bihar
Chhattisgarh
```

4. –complement: As the name suggests it complement the output. This option can be used in the combination with other options either with **-f** or with **-c**.

```
$ cut --complement -d " " -f 1 state.txt
Pradesh
Pradesh
Assam
Bihar
Chhattisgarh

$ cut --complement -c 5 state.txt
Andha Pradesh
Arunchal Pradesh
Assa
Biha
Chhattisgarh
```

5. –output-delimiter: By default the output delimiter is same as input delimiter that we specify in the cut with **-d** option. To change the output delimiter use the option **–output-delimiter=”delimiter”**.

```
$ cut -d " " -f 1,2 state.txt --output-delimiter='%'
Andhra%Pradesh
Arunachal%Pradesh
Assam
Bihar
Chhattisgarh
```

Here cut command changes delimiter(%) in the standard output between the fields which is specified by using -f option .

6. –version: This option is used to display the version of cut which is currently running on your system.

```
$ cut --version
cut (GNU coreutils) 8.26
Packaged by Cygwin (8.26-2)
Copyright (C) 2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later .
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Written by David M. Ihnat, David MacKenzie, and Jim Meyering.
```

Applications of cut Command

1. How to use tail with pipes(|): The cut command can be piped with many other commands of the unix. In the following example output of the **cat** command is given as input to the **cut** command with **-f** option to sort the state names coming from file state.txt in the reverse order.

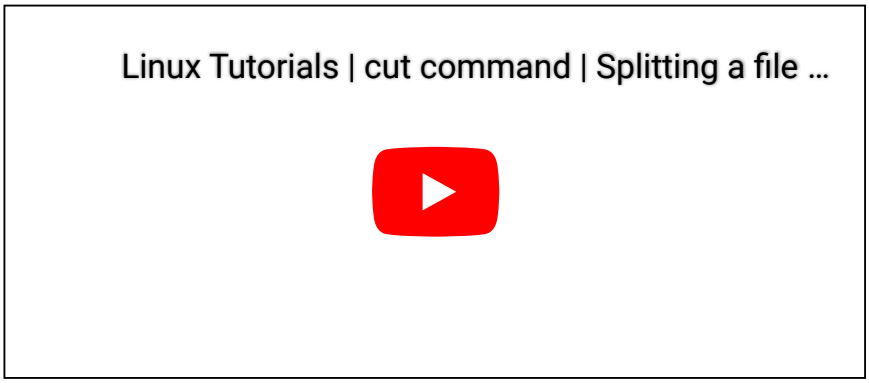
```
$ cat state.txt | cut -d ' ' -f 1 | sort -r
Chhattisgarh
Bihar
Assam
Arunachal
Andhra
```

It can also be piped with one or more filters for additional processing. Like in the following example, we are using cat, head and cut command and whose output is stored in the file name list.txt using directive(>).

```
$ cat state.txt | head -n 3 | cut -d ' ' -f 1 > list.txt

$ cat list.txt
Andhra
Arunachal
Assam
```

Thanks **Saloni Gupta** for providing more examples.



Last Updated : 02 Dec, 2022

52

Similar Reads

1. Ccat – Colorize Cat Command Output command in Linux with Examples
2. 'IPCS' command in Linux with examples
3. select command in Linux with examples
4. Sed Command in Linux/Unix with examples
5. ZIP command in Linux with examples
6. SORT command in Linux/Unix with examples
7. Cat command in Linux with examples
8. Head command in Linux with examples
9. Tail command in Linux with examples
10. wc command in Linux with examples

Previous Next

Article Contributed By :

GeeksforGeeks

Vote for difficulty

Current difficulty : Medium

Easy | Normal | Medium | Hard | Expert

Improved By : akshaysingh98088

Article Tags : linux-command, Linux-Shell-Commands, Linux-Unix

Improve Article | Report Issue

GeeksforGeeks
A-143, 9th Floor, Sovereign Corporate Tower,
Sector-136, Noida, Uttar Pradesh - 201305

feedback@geeksforgeeks.org

Graph

Backtracking

Web Development

HTML

CSS

JavaScript

Bootstrap

ReactJS

AngularJS

NodeJS

Write & Earn

Write an Article

Improve an Article

Pick Topics to Write

Write Interview Experience

Internships

Video Internship

Computer Science

GATE CS Notes

Operating Systems

Computer Network

Database Management System

Software Engineering

Digital Logic Design

Engineering Maths

Data Science & ML

Data Science With Python

Data Science For Beginner

Machine Learning Tutorial

Maths For Machine Learning

Pandas Tutorial

NumPy Tutorial

NLP Tutorial

Interview Corner

Company Preparation

Preparation for SDE

Company Interview Corner

Experienced Interview

Internship Interview

Competitive Programming

Aptitude

Python

Python Tutorial

Python Programming Examples

Django Tutorial

Python Projects

Python Tkinter

OpenCV Python Tutorial

GfG School

CBSE Notes for Class 8

CBSE Notes for Class 9

CBSE Notes for Class 10

CBSE Notes for Class 11

CBSE Notes for Class 12

English Grammar

UPSC/SSC/BANKING

SSC CGL Syllabus

SBI PO Syllabus

IBPS PO Syllabus

UPSC Ethics Notes

UPSC Economics Notes

UPSC History Notes