How to Pipe Command Output to Other Commands in Linux

Aaron Kili | Last Updated: July 14, 2023 | Read Time: 3 mins | Linux Commands | 4 Comments

While using the command line, you can directly pass the output of one program (for example a tool that generates some system information or statistics) as input for another program (such as text-filtering or pattern-searching tools like grep, sed, or awk, for further processing), using a pipeline.

[You might also like: Learn The Basics of How Linux I/O (Input/Output) Redirection Works]

Two of the most important command line utilities that can be used with pipelines to build command lines are:

- xargs – reads streams of data from standard input, then generates and executes command lines.
- tee – reads from standard input and writes simultaneously to standard output and one or many files. It's more of a redirection command.

Sending Command Output to Another Command in Linux

In this simple article, we will describe how to build and execute multiple commands from standard input using **pipes**, **tee**, and **xargs** commands in Linux.

The simplest syntax for using a **pipe**, which you might have already seen in commands in many of our Linux tutorials, is as follows. But you can build a longer command line with several Linux commands.

```
$ command1 args | command2 args
OR
# command1 args | command2 args | command3 args ...
```

Below is an example of using a pipeline to pass the output of the dmesg command to the head command.

```
$ dmesg | head
```

```
aaronkilik@tecmint ~ $ dmesg | head
[    0.000000] Linux version 4.12.0-041200-generic (kernel@gloin) (gcc version 6.3.
0 20170618 (Ubuntu 6.3.0-19ubuntu1) ) #201707022031 SMP Mon Jul 3 00:32:52 UTC 2017
[    0.000000] Command line: BOOT_IMAGE=/boot/vmlinuz-4.12.0-041200-generic root=UU
ID=0e5c45dc-28dc-49aa-a796-3ff14a7dff71 ro quiet splash vt.handoff=7
[    0.000000] KERNEL supported cpus:
[    0.000000]   Intel GenuineIntel
[    0.000000]   AMD AuthenticAMD
[    0.000000]   Centaur CentaurHauls
[    0.000000] x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point registe
rs'
[    0.000000] x86/fpu: Supporting XSAVE feature 0x002: 'SSE registers'
[    0.000000] x86/fpu: Supporting XSAVE feature 0x004: 'AVX registers'
[    0.000000] x86/fpu: xstate_offset[2]:  576, xstate_sizes[2]:  256
aaronkilik@tecmint ~ $
```

Pass Command Output to Another Command

Xargs – Pass Command Output to Other Command

In this example, the ls command output will pass to another command called xargs that concatenate multiple lines of output to one line as shown.

```
$ ls -1 *.sh
$ ls -1 *.sh | xargs
```

```
aaronkilik@tecmint ~ $ ls -1 *.sh
backup.sh
diskusage.sh
enable_lst_debain_repo.sh
mkdirs.sh
update.sh
aaronkilik@tecmint ~ $ ls -1 *.sh | xargs
backup.sh diskusage.sh enable_lst_debain_repo.sh mkdirs.sh update.sh
aaronkilik@tecmint ~ $
```

Run Commands Using Xargs

To count the number of lines/words/characters in each file in a list, use the commands below.

```
$ ls *.sh | xargs wc -l     #count number of lines in each file
$ ls *.sh | xargs wc -w     #count number of words in each file
$ ls *.sh | xargs wc -c     #count number of characters in each file
$ ls *.sh | xargs wc        #count lines, words, and characters in each fil
```

```
aaronkilik@tecmint ~ $ ls *.sh | xargs wc
  0   0    0 backup.sh
  3   3   20 diskusage.sh
 45 243 1849 enable_lst_debain_repo.sh
 19  41  579 mkdirs.sh
  0   0    0 update.sh
 67 287 2448 total
aaronkilik@tecmint ~ $
```

Count File Words Using Xargs

The command below finds and recursively deletes the directory named `All` in the current directory.

```
$ find . -name "All" -type d -print0 | xargs -0 /bin/rm -rf "{}"
```

The find command with the option `-print0` action enables printing of the full directory path on the standard output, followed by a null character and `-0 xargs` flag deals with space in filenames and an rm -rf command to delete a directory.

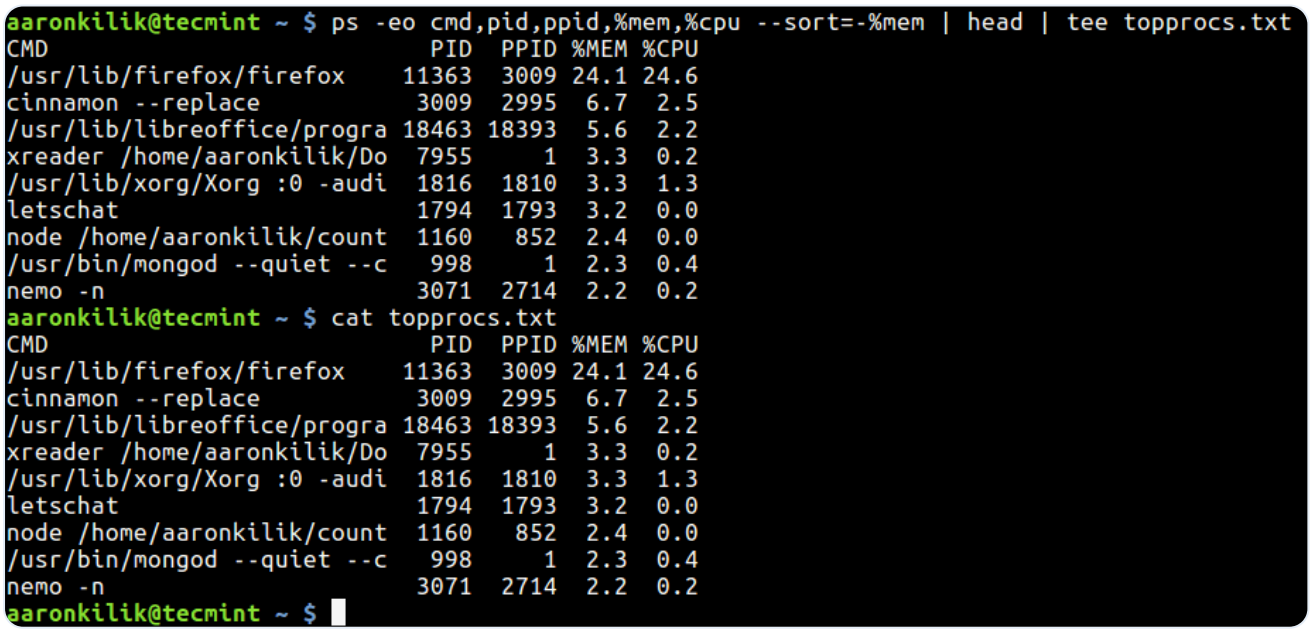
You can find other practical **xargs command** usage examples in these articles:

- *How to Copy a File to Multiple Directories in Linux*
- *Rename All Files and Directory Names to Lowercase in Linux*
- *4 Ways to Batch Convert Your PNG to JPG and Vice-Versa*
- *3 Ways to Delete All Files in a Directory Except One or Few Files with Extensions*

Tee – Send Command Output to Other Command and Save to File

This example shows how to send command output to standard output and save it to a file; the command below allows you to view the top running processes by highest memory and CPU usage in Linux.

```
$ ps -eo cmd,pid,ppid,%mem,%cpu --sort=-%mem | head | tee topprocs.txt
$ cat  topprocs.txt
```



Save Command Output to File

To append data in an existing file(s), pass the `-a` flag.

```
$ ps -eo cmd,pid,ppid,%mem,%cpu --sort=-%mem | head | tee -a topprocs.txt
```

You can find more information on the **tee** and **xargs** man pages.

```
$ man xargs
$ man tee
```

That's all! Do not forget to check out our special article: A – Z Linux Commands – Overview with Examples.

In this article, we described how to generate command lines using **pipelines**; **xargs**, and **tee** commands. You can ask any questions or share any thoughts via the feedback form below.

Linux Tricks

Hey TecMint readers,

Exciting news! Every month, our top blog commenters will have the chance to win fantastic rewards, like free Linux eBooks such as RHCE, RHCSA, LFCS, Learn Linux, and Awk, each worth $20!

Learn more about the contest and stand a chance to win by sharing your thoughts below!

PREVIOUS ARTICLE:

How to Run Linux Commands in Background and Detach in Terminal

NEXT ARTICLE:

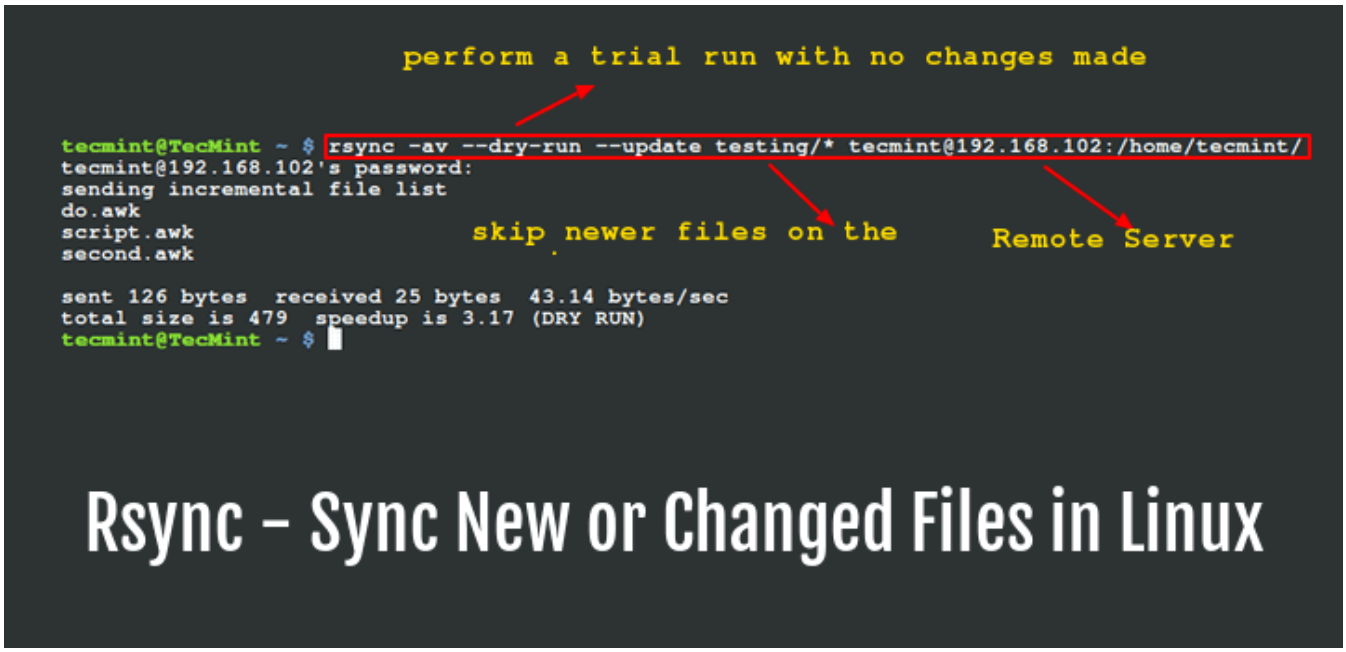
Fd – The Best Alternative to 'Find' Command for Quick File Searching

Aaron Kili

Aaron Kili is a Linux and F.O.S.S enthusiast, an upcoming Linux SysAdmin, web developer, and currently a content creator for TecMint who loves working with computers and strongly believes in sharing knowledge.

*Each tutorial at **TecMint** is created by a team of experienced Linux system administrators* so that it meets our high-quality *standards*.

Related Posts



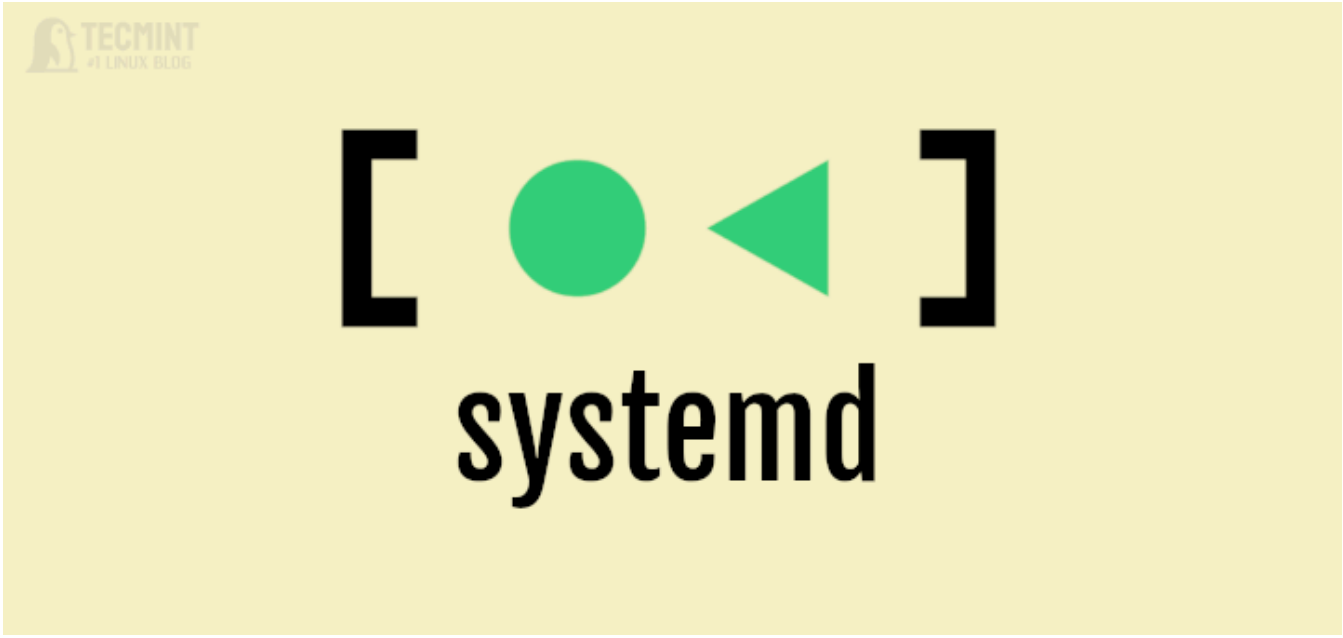
How to Sync New and Changed Files Using ‘rsync’ Command

How to Search Files by Name or Extension Using find Command

10 Lesser Known Linux Commands – Part 2

11 Lesser Known Useful Linux Commands

26 Security Hardening Tips for Modern Linux Servers



How to Remove Systemd Services on Linux

4 Comments

Leave a Reply

David Webb

July 30, 2024 at 6:44 pm

I see a lot of articles explaining the tee command the way you did, but why use tee like this?

Your code example:

```
ps -eo cmd,pid,ppid,%mem,%cpu --sort=-%mem | head | tee topprocs.txt
```

could just as easily use redirection instead of tee:

```
ps -eo cmd,pid,ppid,%mem,%cpu --sort=-%mem | head > topprocs.txt
```

Use `>>` for appending.

In my opinion, tee is useful if you need to send the same output to multiple files simultaneously:

```
ps -eo cmd,pid,ppid,%mem,%cpu --sort=-%mem | head | tee topprocs1.txt topprocs2.txt
```

and so on.

Reply

Admin

Ravi Saive

July 31, 2024 at 10:27 am

@David,

You're absolutely right!

The tee command is particularly useful when you want to direct output to multiple files at once or also display it on the terminal while saving it to a file.

For a simple case where you only need to save output to one file, redirection with `>` or `>>` works just fine.

Reply

Pavel Vachek

June 12, 2023 at 12:47 am

"Tree – Send Command Output to Other Command and Save to File"

I think that the first word in the title should read "Tee", not "Tree".

Reply

Admin

Ravi Saive

June 12, 2023 at 9:51 am

@Pavel,

Thanks, corrected the command in the article...

Reply

Got Something to Say? Join the Discussion...

Thank you for taking the time to share your thoughts with us. We appreciate your decision to leave a comment and value your contribution to the discussion. It's

important to note that we moderate all comments in accordance with our comment policy to ensure a respectful and constructive conversation.

Rest assured that your email address will remain private and will not be published or shared with anyone. We prioritize the privacy and security of our users.

Name *

Email *

Save my name, email, and website in this browser for the next time I comment.

Post Comment

Search...

Do You Enjoy My Blog?

Support from readers like YOU keeps this blog running. Buying me a cup of coffee is a simple and affordable way to show your appreciation and help keep the posts coming!

Buy Me a Coffee

Linux Commands and Tools

6 Useful apt-cache Command Examples for Ubuntu

How to Calculate IP Subnet Address with ipcalc Tool

How to Save Command Output to a File in Linux

How to Use IP Command in Linux [24 Useful Examples]

12 Ping Command Examples to Test Your Network

How to Find Recent or Today's Modified Files in Linux

Linux Server Monitoring Tools

How to Add Windows Host to Nagios Monitoring Server

How to Monitor System Usage, Outages and Troubleshoot Linux Servers – Part 9

10 Strace Commands for Troubleshooting and Debugging Linux Processes

BpyTop – Resource Monitoring Tool for Linux

Pyinotify – Monitor Filesystem Changes in Real-Time in Linux

GoAccess (A Real-Time Apache and Nginx) Web Server Log Analyzer

Learn Linux Tricks & Tips

How to Add a New Disk to an Existing Linux Server

How to Count Number of Files and Subdirectories inside a Given Directory

How to Create Hard and Symbolic Links in Linux

Lolcat – Display Text in Rainbow Colors in Linux Terminal

How to Check Bad Sectors or Bad Blocks on Hard Disk in Linux

How to Add a New Disk Larger Than 2TB to An Existing Linux

Best Linux Tools

5 Best Skype Alternatives for Linux Desktop

10 Most Popular Download Managers for Linux in 2023

5 Best PDF to Word Converters for Linux

6 Best Whiteboard Applications for Your Linux Systems

8 Best PowerPoint Alternatives for Linux

5 Best Open-Source eLearning Platforms for Linux in 2024