Mommy, I found it! — 15 Practical Linux Find Command Examples

by SathiyaMoorthy on March 4, 2009

Tweet

Photo courtesy of Oole Pejorian

Apart from the basic operation of looking for files under a directory structure, you can also perform several practical operations using find command that will make your command line journey easy.

In this article, let us review 15 practical examples of Linux find command that will be very useful to both newbies and experts.

First, create the following sample empty files under your home directory to try some of the find command examples mentioned below.

```
# vim create_sample_files.sh
touch MybashProgram.sh
touch mycprogram.c
touch MyCProgram.c
touch Program.c

mkdir backup
cd backup

touch MybashProgram.sh
touch mycprogram.c
touch MyCProgram.c
touch Program.c

# chmod +x create_sample_files.sh

# ./create_sample_files.sh

# ls -R
.:
backup                  MybashProgram.sh  MyCProgram.c
create_sample_files.sh  mycprogram.c      Program.c

./backup:
MybashProgram.sh  mycprogram.c  MyCProgram.c  Program.c
```

1. Find Files Using Name

This is a basic usage of the find command. This example finds all files with name — MyCProgram.c in the current directory and all its sub-directories.

```
# find -name "MyCProgram.c"
./backup/MyCProgram.c
./MyCProgram.c
```

2. Find Files Using Name and Ignoring Case

This is a basic usage of the find command. This example finds all files with name — MyCProgram.c (ignoring the case) in the current directory and all its sub-directories.

```
# find -iname "MyCProgram.c"
./mycprogram.c
./backup/mycprogram.c
./backup/MyCProgram.c
./MyCProgram.c
```

3. Limit Search To Specific Directory Level Using mindepth and maxdepth

Find the passwd file under all sub-directories starting from root directory.

```
# find / -name passwd
./usr/share/doc/nss_ldap-253/pam.d/passwd
./usr/bin/passwd
./etc/pam.d/passwd
./etc/passwd
```

Find the passwd file under root and one level down. (i.e root — level 1, and one sub-directory — level 2)

```
# find -maxdepth 2 -name passwd
./etc/passwd
```

Find the passwd file under root and two levels down. (i.e root — level 1, and two sub-directories — level 2 and 3)

```
# find / -maxdepth 3 -name passwd
./usr/bin/passwd
./etc/pam.d/passwd
./etc/passwd
```

Find the password file between sub-directory level 2 and 4.

```
# find -mindepth 3 -maxdepth 5 -name passwd
./usr/bin/passwd
./etc/pam.d/passwd
```

4. Executing Commands on the Files Found by the Find Command.

In the example below, the find command calculates the md5sum of all the files with the name MyCProgram.c (ignoring case). {} is replaced by the current file name.

```
# find -iname "MyCProgram.c" -exec md5sum {} \;
d41d8cd98f00b204e9800998ecf8427e  ./mycprogram.c
d41d8cd98f00b204e9800998ecf8427e  ./backup/mycprogram.c
d41d8cd98f00b204e9800998ecf8427e  ./backup/MyCProgram.c
d41d8cd98f00b204e9800998ecf8427e  ./MyCProgram.c
```

5. Inverting the match.

Shows the files or directories whose name are not MyCProgram.c .Since the maxdepth is 1, this will look only under current directory.

```
# find -maxdepth 1 -not -iname "MyCProgram.c"
.
./MybashProgram.sh
./create_sample_files.sh
./backup
./Program.c
```

6. Finding Files by its inode Number.

Every file has an unique inode number, using that we can identify that file. Create two files with similar name. i.e one file with a space at the end.

```
# touch "test-file-name"

# touch "test-file-name "
[Note: There is a space at the end]

# ls -1 test*
test-file-name
test-file-name
```

From the ls output, you cannot identify which file has the space at the end. Using option -i, you can view the inode number of the file, which will be different for these two files.

```
# ls -i1 test*
16187429 test-file-name
16187430 test-file-name
```

You can specify inode number on a find command as shown below. In this example, find command renames a file using the inode number.

```
# find -inum 16187430 -exec mv {} new-test-file-name \;
# ls -i1 *test*
16187430 new-test-file-name
16187429 test-file-name
```

You can use this technique when you want to do some operation with the files which are named poorly as shown in the example below. For example, the file with name — file?.txt has a special character in it. If you try to execute "rm file?.txt", all the following three files will get removed. So, follow the steps below to delete only the "file?.txt" file.

```
# ls
file1.txt  file2.txt  file?.txt
```

Find the inode numbers of each file.

```
# ls -i1
804178 file1.txt
804179 file2.txt
804180 file?.txt
```

Use the inode number to remove the file that had special character in it as shown below.

```
# find -inum 804180 -exec rm {} \;

# ls
file1.txt  file2.txt
[Note: The file with name "file?.txt" is now removed]
```

7. Find file based on the File-Permissions

Following operations are possible.

- Find files that match exact permission
- Check whether the given permission matches, irrespective of other permission bits
- Search by giving octal / symbolic representation

For this example, let us assume that the directory contains the following files. Please note that the file-permissions on these files are different.

```
# ls -l
total 0
-rwxrwxrwx 1 root root 0 2009-02-19 20:31 all_for_all
-rw-r--r-- 1 root root 0 2009-02-19 20:30 everybody_read
---------- 1 root root 0 2009-02-19 20:31 no_for_all
-rw------- 1 root root 0 2009-02-19 20:29 ordinary_file
-rw-r----- 1 root root 0 2009-02-19 20:27 others_can_also_read
----r----- 1 root root 0 2009-02-19 20:27 others_can_only_read
```

Find files which has read permission to group. Use the following command to find all files that are readable by the world in your home directory, irrespective of other permissions for that file.

```
# find . -perm -g=r -type f -exec ls -l {} \;
-rw-r--r-- 1 root root 0 2009-02-19 20:30 ./everybody_read
-rwxrwxrwx 1 root root 0 2009-02-19 20:31 ./all_for_all
----r----- 1 root root 0 2009-02-19 20:27 ./others_can_only_read
-rw-r----- 1 root root 0 2009-02-19 20:27 ./others_can_also_read
```

Find files which has read permission only to group.

```
# find . -perm g=r -type f -exec ls -l {} \;
----r----- 1 root root 0 2009-02-19 20:27 ./others_can_only_read
```

Find files which has read permission only to group [search by octal]

```
# find . -perm 040 -type f -exec ls -l {} \;
----r----- 1 root root 0 2009-02-19 20:27 ./others_can_only_read
```

8. Find all empty files (zero byte file) in your home directory and its subdirectory

Most files of the following command output will be lock-files and place holders created by other applications.

```
# find ~ -empty
```

List all the empty files only in your home directory.

```
# find . -maxdepth 1 -empty
```

List only the non-hidden empty files only in the current directory.

```
# find . -maxdepth 1 -empty -not -name ".*"
```

9. Finding the Top 5 Big Files

The following command will display the top 5 largest file in the current directory and its subdirectory. This may take a while to execute depending on the total number of files the command has to process.

```
# find . -type f -exec ls -s {} \; | sort -n -r | head -5
```

10. Finding the Top 5 Small Files

Technique is same as finding the bigger files, but the only difference the sort is ascending order.

```
# find . -type f -exec ls -s {} \; | sort -n  | head -5
```

In the above command, most probably you will get to see only the ZERO byte files (empty files). So, you can use the following command to list the smaller files other than the ZERO byte files.

```
# find . -not -empty -type f -exec ls -s {} \; | sort -n  | head -5
```

11. Find Files Based on file-type using option -type

Find only the socket files.

```
# find . -type s
```

Find all directories

```
# find . -type d
```

Find only the normal files

```
# find . -type f
```

Find all the hidden files

```
# find . -type f -name ".*"
```

Find all the hidden directories

```
# find -type d -name ".*"
```

12. Find files by comparing with the modification time of other file.

Show files which are modified after the specified file. The following find command displays all the files that are created/modified after ordinary_file.

```
# ls -lrt
total 0
-rw-r----- 1 root root 0 2009-02-19 20:27 others_can_also_read
----r----- 1 root root 0 2009-02-19 20:27 others_can_only_read
-rw------- 1 root root 0 2009-02-19 20:29 ordinary_file
-rw-r--r-- 1 root root 0 2009-02-19 20:30 everybody_read
-rwxrwxrwx 1 root root 0 2009-02-19 20:31 all_for_all
---------- 1 root root 0 2009-02-19 20:31 no_for_all

# find -newer ordinary_file
.
./everybody_read
./all_for_all
./no_for_all
```

13. Find Files by Size

Using the -size option you can find files by size.

Find files bigger than the given size

```
# find ~ -size +100M
```

Find files smaller than the given size

```
# find ~ -size -100M
```

Find files that matches the exact given size

```
# find ~ -size 100M
```

Note: – means less than the give size, + means more than the given size, and no symbol means exact given size.

14. Create Alias for Frequent Find Operations

If you find some thing as pretty useful, then you can make it as an alias. And execute it whenever you want.

Remove the files named a.out frequently.

```
# alias rmao="find . -iname a.out -exec rm {} \;"
# rmao
```

Remove the core files generated by c program.

```
# alias rmc="find . -iname core -exec rm {} \;"
# rmc
```

15. Remove big archive files using find command

The following command removes *.zip files that are over 100M.

```
# find / -type f -name *.zip -size +100M -exec rm -i {} \;"
```

Remove all *.tar file that are over 100M using the alias rm100m (Remove 100M). Use the similar concepts and create alias like rm1g, rm2g, rm5g to remove file size greater than 1G, 2G and 5G respectively.

```
# alias rm100m="find / -type f -name *.tar -size +100M -exec rm -i {} \;"
# alias rm1g="find / -type f -name *.tar -size +1G -exec rm -i {} \;"
# alias rm2g="find / -type f -name *.tar -size +2G -exec rm -i {} \;"
# alias rm5g="find / -type f -name *.tar -size +5G -exec rm -i {} \;"

# rm100m
# rm1g
# rm2g
# rm5g
```

Find Command Examples Second Part

If you liked this Mommy article on find command, don’t forget to check-out the Daddy article of the find command — Daddy, I found it!, 15 Awesome Linux Find Command Examples (Part2)

Awesome Linux Articles

Following are few awesome **15 examples** articles that you might find helpful.

- 15 Examples To Master Linux Command Line History
- Unix LS Command: 15 Practical Examples
- Get a Grip on the Grep! – 15 Practical Grep Command Examples
- Linux Crontab: 15 Awesome Cron Job Examples

Tweet > Add your comment

If you enjoyed this article, you might also like..

1. 50 Linux Sysadmin Tutorials
2. 50 Most Frequently Used Linux Commands (With Examples)
3. Top 25 Best Linux Performance Monitoring and Debugging Tools
4. Mommy, I found it! – 15 Practical Linux Find Command Examples
5. Linux 101 Hacks 2nd Edition eBook Free

- Awk Introduction – 7 Awk Print Examples
- Advanced Sed Substitution Examples
- 8 Essential Vim Editor Navigation Fundamentals
- 25 Most Frequently Used Linux IPTables Rules Examples
- Turbocharge PuTTY with 12 Powerful Add-Ons

Tags: find, Find Big Files, Find Command, Find Command Examples, Find Command Operators, Find Command RegEx, Find Small Files, Linux Find Command, Linux Search Tool, Locate Files, Search Directories, Search Files, Search Sub-Directories

{ 48 comments... read them below or add one }

1 Mathias March 4, 2009 at 12:12 pm

Really nice overview, thanks. As an addition a very handy combination of find and grep I use almost every day: find a string inside a filename.

find . -type f -exec grep -qi "foo" {} \; -print

2 ejjp March 4, 2009 at 12:12 pm

Hello,

good examples! Some was unknows for me.

could you explain some examples to find files by date range (creation, modification,..etc)?

3 Tim March 4, 2009 at 2:24 pm

Hello,

It was a long time I wanted to really know how to use this command. Now I know, Thanks.

4 myselfhimself March 4, 2009 at 7:25 pm

you can use find for setting permissions
this is useful for server side scripts for example:

find /var/www/ -name "*.php*" -exec chmod 644 {} \; #changes permissions of all php php4 php5.. files in /var/www/ to 644 (user:rw group:r other:r)

also note that the -exec thing can be done by piping to xargs:
find -name "*~" | xargs rm #is equivalent to rm -r "*~" ; and this removes all files ending with ~ as a last chacter, under the present folder. Usually files ending with ~ are temporary files created by text editors such as kate.
or
find -name "*~" | xargs -i echo file {} is maybe useless ; #using the -i option allows you to tell where the parameter received by xargs over stdin will go in the command given to xargs for executing.

5 find command user March 5, 2009 at 3:45 am

find . -type f -print | xargs grep -ni "mystring"

6 Thanh Dat March 5, 2009 at 6:44 am

Very nice post, thanks !!!
More details can be found in Unix Power Tools (Oreilly) which contains many other super-cool tools ^^

7 Aleš Friedl March 5, 2009 at 8:11 am

Just small note – there is a much more straightforward way to remove files with wildchars. Just

rm "test?.txt"

8 Aleš Friedl March 5, 2009 at 8:22 am

Mathias:

It is useful to distinguish

find . -type f -iname "*foo*"

where you can use patterns with wildcards vs. more complicated but more powerful

find . -type f -exec grep -qi "…regexp…" {} \; -print

as people often do not need regular expressions and the first is more easy to remember and type.

9 Eric Wendelin March 5, 2009 at 3:05 pm

You can also find files changed in the last day:
find . -ctime -1 -type f

also, "find . -print0 | xargs -0 cmd" is safer than just using xargs.

10 Bala March 7, 2009 at 1:48 pm

I was looking for something like for a long time.

11 Mit March 9, 2009 at 7:42 am

Excellent work buddy! Thanks.

12 Mike J. March 12, 2009 at 11:40 am

Great post. Find combined with -exec is one of the most powerful and useful utilities.
Re: #5 Inverting the match- at least on some flavors you can use \! as a shortcut for -not -iname…

```
# find -not -iname "MyCProgram.c"
# find 1 \! "MyCProgram.c"
```

13 Mike J. March 12, 2009 at 5:58 pm

Sorry my previous comment example was in correct.
It should read…

```
# find \! -iname "MyCProgram.c"
# find -not -iname "MyCProgram.c"
```

14 Daemon March 24, 2009 at 3:03 am

Gr8, thanks.

How to find files that are newly created 5 or 10 minutes before ?

15 Hari April 7, 2009 at 6:39 am

We can find the latest access/changed files using following command,

```
$ find -cmin
```

16 peter May 12, 2009 at 2:19 am

I'm using macosx tiger and if I use the command for example:

```
find -iname "MyCProgram.c"
```

then:

```
find: illegal option — i
find: illegal option — n
find: illegal option — a
find: illegal option — m
find: illegal option — e
```

Why?

17 Speed Racer May 12, 2009 at 6:13 am

@peter

try this:

```
find . -iname "MyCProgram.c"
```

18 James May 12, 2009 at 4:32 pm

@peter: install Ubuntu and try again. 😀

19 Mike May 13, 2009 at 3:34 am

```
# alias rmc="find . -iname core -exec rm {} \;"
# rmc
```

find has a -delete option so that the upper can simply look like:

```
# alias rmc="find . -iname core -delete"
# rmc
```

20 peter May 24, 2009 at 2:56 am

You may also get a speed up using:

```
find … -exec cmd '{}' +
```

(note: this is also an alternative to using xargs -0)

21 shellbasher May 25, 2009 at 6:22 pm

Late comment for peter, he should install an alias in his .profile or .bashrc like this:
alias find=/sw/bin/find
because there is another find command which are preceding the right one in the default path string.

22 Komal July 6, 2009 at 5:43 am

When I am doing find -inum , I am not getting any output for some files inode numbers.Why this is happening?

23 Ramesh Natarajan July 8, 2009 at 11:01 pm

@Tim, @Bala, @Mit, Thanks for your comment. I'm very glad that you found this article helpful.

@ejjp, @Daemon, To find files based on date, refer to our Find Command Examples (Part 2) article.

@Mathias, @myselfhimself, @find command user, @Aleš Friedl, @Eric Wendelin, @Mike J, @Hari, @Mike, @Peter, @shellbasher, @Geoff, Thanks for sharing your awesome examples on find command. I appreciate it.

@Thanh Dat, Yes. I agree. Unix Power Tools, Third Edition is an excellent book with lot of practical examples.

@Speed Racer, @James, Thanks for helping out Peter. 😀

@Komal, Can you give us an example of exactly what command you tried and what was your output? Did you give the correct inode number in the find command?

24 Gabriel July 23, 2009 at 1:52 am

Thank You Ramesh for the useful examples, especialy for the commands that remove the big files.

25 Ramesh Natarajan July 25, 2009 at 12:25 am

@Gabriel,

Thanks for the comments. I'm very glad that you found this article helpful in removing the big files on your linux system.

26 chandan March 31, 2010 at 11:34 pm

Hi,

I am unable to find out the boot error which had occured some where in last week while rebooting the server.

Can someone help me out finding that error.

Thanks

27 raj August 27, 2010 at 1:08 am

Really it is helping me a lot,…

28 charles October 25, 2010 at 4:57 am

Hai ,
find has an inbuilt -ls option , which greatly reduces the fork time incured when we use -exec ls . you may use
find /opt -type f -size +10000000c -ls | sort -nk 7 | tail

29 Lew January 8, 2011 at 1:48 pm

I know # is for using comments in Linux, so I don't understand your use of it in some lines like eg these:
# vim create_sample_files.sh
# chmod +x create_sample_files.sh
# ./create_sample_files.sh
Am I supposed to add the "#"? Are these comments and shouldn't I type 'em at all?
Since the tutorial is also for newbies, as you wrote, could you explain please?
Thanks 🙂

30 Venkat Subramanian January 20, 2011 at 7:30 pm

Hey Ramesh !! You Rock man.. I love this website and ur free books.. Thanks for all the work you are doing!! I am proud of you!! 🙂

31 lokesh grover March 22, 2011 at 11:06 am

I m student of iacm for hardware networking . I m gain the extra knowldge of linux and networking .

32 chamu May 3, 2011 at 11:48 am

Hi Ramesh,
I like these commands and examples. Really of a help even for an experienced software developer who does not have hands on shell scripting. Keep up the servi

33 ionh May 16, 2011 at 3:32 pm

I've been trying to figure out how to "find" a set of files, with *both* inclusive and exclusive name filters, and then execute a command on the results – conceptually, something like:

find . -iname "*.txt" -exec grep -v blob ; cp {} MyDir \;

Of course, this is broken, but should indicate what i'm trying to do, namely,
find all "*.txt" files; reject any that are "*blob*"; and copy the rest to the MyDir directory.

Anyone know how to do this?

34 Peng July 9, 2011 at 10:22 pm

hi ionh,

step 1, find all .txt files in the current directory and all of its subdirectories
$ find . -type f -name "*.txt"
step 2, pipe the result to grep, find those files containing 'blob'
$ find . -type f -name "*.txt" | grep 'blob'
step 3, add '-v' (invert-match), find those files not containing 'blob'
$ find . -type f -name "*.txt" | grep -v 'blob'
step 4, pipe the file list to xargs, which breaks the list into sublists and deal with each one of them, note here we have to use '–target-directory' to clarify our intention.
$ find . -type f -name "*.txt" | grep -v 'blob'| xargs cp –target-directory=newDir

now you got all the files you want in the newDir. Mission completed! hope this helps.

Peng

note the '–target-directory' above has 2 hyphen-minuses, instead of a dash, this website made it barely recognizable. you'll know what i am talking about if you 'cp –help'.

35 shubham July 13, 2011 at 3:24 am

Wonderful!!

36 Vimal Roy October 13, 2011 at 3:21 am

great cmds and example. thank u very much. need more……

37 debasis November 13, 2011 at 3:39 am

can anyone please tell me
the use of "prune" in linux.
please explain it with example.

thanx alot

38 Roberto January 27, 2012 at 4:06 am

Hello Ramesh! Many thanks for all this useful information. Im currently preparing to take my LPI-C 1 exam and while looking for more info in regard find command I found this web site. However, Im still have one initial question unaswered, what is the {} \; for?
{} is replaced by the current file name.(I got this)
then you have that backslash there and and semi-colon. I guessed this is related to the shell environment. Any chance that you or other internauts of this web site could explain a little bit further?
Kind Regards

39 Dan January 31, 2012 at 9:50 am

I just came across a few articles on thegeekstuff.com, this being one of them. All I can say is, with writing like this there is hope that "normal people" will learn Linux. I've been in the industry for 7 years now (programming degree, etc) and this is some of the best, if not THE best written Linux quickref doco I've seen. Normal geeks (like me, as opposed to command line junkies) don't read man pages because they're written like scientific notation and contain virtually no examples. And if they do contain examples, they're for rare cases that I virtually never use. This stuff is written well, to the point without lots of unnecessary verbiage, and has examples I can actually use. Great work guys!

40 Abhijit February 15, 2012 at 12:47 am

Awesome Work !!!!. Please keep enlightening all of us ….

41 mr March 15, 2012 at 7:13 am

RE: 6. Finding Files by its inode Number.
RE: find command renames a file using the inode number.
RE: # find -inum 16187430 -exec mv {} new-test-file-name \;

Warning: Running the above command will erase any duplicate hard links to inode 16187430. "new-test-file-name" will still point to inode 16187430 afterward, but duplicate hard links to inode 16187430 using other file names or paths will be gone.

While odds are low duplicate hard links exist, a safer method would check for duplicate hard links to inode 16187430:
find -inum 16187430 -printf "%n %p\n"

The 1st column in the output displays the hard link count and any duplicate files hard linking to inode 16187430. If the hard link count is 1, no hard links to inode 16187430 exist and the top command is safe to use.

And yes, the article is old but none of the comments referenced this.

42 priya March 25, 2012 at 11:16 pm

Could you please explain the usage of mindepth and maxdepth…and the above mentioned example of finiding passwd file in between level 2 and 4…..

43 ramanathan July 14, 2012 at 3:01 pm

Hi all,
i get the following error when performing 3rd one on above!!

```
`/proc/16918/task/16918/fd': Permission denied
find: `/proc/16918/task/16918/fdinfo': Permission denied
find: `/proc/16918/fd': Permission denied
find: `/proc/16918/fdinfo': Permission denied
find: `/proc/16935/task/16935/fd': Permission denied
find: `/proc/16935/task/16935/fdinfo': Permission denied
find: `/proc/16935/fd': Permission denied
find: `/proc/16935/fdinfo': Permission denied
find: `/sys/kernel/debug': Permission denied
/home/ram/ramapala/bo
find: `/lost+found': Permission denied
find: `/tmp/vmware-root': Permis
```

44 dhandayuthabani August 14, 2012 at 2:51 am

awesome explanation for find command thanks a lot….do it for all the unix commands

45 Peter November 3, 2012 at 4:29 pm

Hi.

I am looking for a FIND IF REMOVE command. I know it can be done but don't know how to do it. I believe this is referred to a conditional expression. I am out of my depth here.

I want to remove the smallest sized file IF a larger version exists of the same file (because it will be a better quality data). Both files will have the same date but the smaller files will have slightly different names, e:g cooking1-photograph100px.jpg as opposed to cooking1-photograph200px.jpg

There are thousands of files on my system in dozens of different directories.

How can I use the FIND and REMOVE IF commands if criteria are met so as to remove multiple files using a wild card to represent the consistent parts of the file name so that it will remove all smaller sized versions of a file, ONLY if a larger version exist on my system?

eg: Find and remove the file cooking1-photograph100px.jpg IF cooking1-photograph200px.jpg exists, but to leave cooking1-photograph100.jpg if no other other file named "cooking1-photograph"*.jpg exists?

Obviously I need to use a wild card in place of, for example "cooking1-photograph" so that the command will find all files listed not just as "cooking1-photograph" but also "cooking2-photograph" and "cooking3-photograph" and then apply the conditional criteria looking for "100″, then deleting the file *100.jpg IF file *200.jpg is found.

I have thousands of files that I need to sort through and cull.

Can this be done? Please tell me how.

Thank you
Peter

46 Aishwarya April 26, 2013 at 3:47 am

For finding big/small files sorting is being done on the basis of block size. I have tried sorting based on the actual size. Here it is :

9. Finding the Top 5 Big Files
find . -type f -ls | awk '{print $7″\t"$11}' | sort -nr | head -5

10. Finding the Top 5 Small Files
find . -type f -ls | awk '{print $7″\t"$11}' | sort -n | head -5

47 pranil June 18, 2013 at 1:21 pm

Hi,
I have 100 files. and out of those in 80 files there is a particular string 'john'.
Now my objective is to find all those files which contains the string 'john' and remove them. can any one please help me to do that..

48 Aishwarya June 19, 2013 at 1:12 am

@pranil

# go to the directory in which 100 files are present
# and try out the instructions below on your prompt
ls | while read fileName
do
sed s:John::g $fileName -i
done

Leave a Comment

Name

E-mail

Website

☐ Notify me of followup comments via e-mail

Submit

Previous post: 3 Powerful Musketeers Of Vim Editor — Macro, Mark and Map

Next post: 8 Essential Vim Editor Navigation Fundamentals

- Search

  95012 readers BY FEEDBURNER >Email >RSS >Twitter >Facebook

- **COURSE**
  - Linux Sysadmin CentOS 6 Course - Master the Tools, Configure it Right, and be Lazy
- **EBOOKS**
  - Free Linux 101 Hacks 2nd Edition eBook - Practical Examples to Build a Strong Foundation in Linux
  - Bash 101 Hacks eBook - Take Control of Your Bash Command Line and Shell Scripting

- **POPULAR POSTS**
  - 12 Amazing and Essential Linux Books To Enrich Your Brain and Library
  - 50 UNIX / Linux Sysadmin Tutorials
  - 50 Most Frequently Used UNIX / Linux Commands (With Examples)
  - How To Be Productive and Get Things Done Using GTD
  - 30 Things To Do When you are Bored and have a Computer
  - Linux Directory Structure (File System Structure) Explained with Examples
  - Linux Crontab: 15 Awesome Cron Job Examples
  - Get a Grip on the Grep! – 15 Practical Grep Command Examples
  - Unix LS Command: 15 Practical Examples
  - 15 Examples To Master Linux Command Line History
  - Top 10 Open Source Bug Tracking System
  - Vi and Vim Macro Tutorial: How To Record and Play
  - Mommy, I found it! -- 15 Practical Linux Find Command Examples
  - 15 Awesome Gmail Tips and Tricks
  - 15 Awesome Google Search Tips and Tricks
  - RAID 0, RAID 1, RAID 5, RAID 10 Explained with Diagrams
  - Can You Top This? 15 Practical Linux Top Command Examples
  - Top 5 Best System Monitoring Tools
  - Top 5 Best Linux OS Distributions
  - How To Monitor Remote Linux Host using Nagios 3.0
  - Awk Introduction Tutorial – 7 Awk Print Examples
  - How to Backup Linux? 15 rsync Command Examples
  - The Ultimate Wget Download Guide With 15 Awesome Examples
  - Top 5 Best Linux Text Editors
  - Packet Analyzer: 15 TCPDUMP Command Examples
  - The Ultimate Bash Array Tutorial with 15 Examples
  - 3 Steps to Perform SSH Login Without Password Using ssh-keygen & ssh-copy-id
  - Unix Sed Tutorial: Advanced Sed Substitution Examples
  - UNIX / Linux: 10 Netstat Command Examples
  - The Ultimate Guide for Creating Strong Passwords
  - 6 Steps to Secure Your Home Wireless Network
  - Turbocharge PuTTY with 12 Powerful Add-Ons
- **CATEGORIES**
  - Linux
  - Vim
  - Sed
  - Awk
  - Bash
  - Nagios
  - OpenSSH
  - IPTables
  - Apache
  - MySQL
  - Perl
  - Google
  - Ubuntu
  - PostgreSQL
  - Hello World
  - C Programming
  - C++ Programming
  - DELL
  - Oracle
  - VMware
- **About The Geek Stuff**

My name is **Ramesh Natarajan**. I will be posting instruction guides, how-to, troubleshooting tips and tricks on Linux, database, hardware, security and web. My focus is to write articles that will either teach you or help you resolve a problem. Read more about Ramesh Natarajan and the blog.

- **Support Us**

Nagios Core 3 eBook

- **Contact Us**

Email Me : Use this Contact Form to get in touch me with your comments, questions or suggestions about this site. You can also simply drop me a line to say hello!.

Follow us on Twitter

Become a fan on Facebook