

- [Home](#)
- [Free eBook](#)
- [Contact](#)
- [About](#)
- [Start Here](#)

# The Ultimate Linux Soft and Hard Link Guide (10 Ln Command Examples)

by Balakrishnan Mariyappan on October 4, 2010

 11

Like  17

Tweet  2



There are two types of links available in Linux — **Soft Link and Hard Link.**

Linux ln command is used to create either soft or hard links.

This article explains how to create soft link, how to create hard link, and various link tips and tricks with 10 practical examples.

```
$ ls -l
total 4
lrwxrwxrwx 1 chris chris 10 2010-09-17 23:40 file1 -> sample.txt
-rw-r--r-- 1 chris chris 22 2010-09-17 23:36 sample.txt
```

The 1st character in each and every line of the [ls command](#) output indicates one of the following file types. If the 1st character is l (lower case L), then it is a link file.

- - regular file
- **l** link file
- **d** directory
- **p** pipe
- **c** character special device
- **b** block special device

## 1. What is Soft Link and Hard Link?

### Soft Link

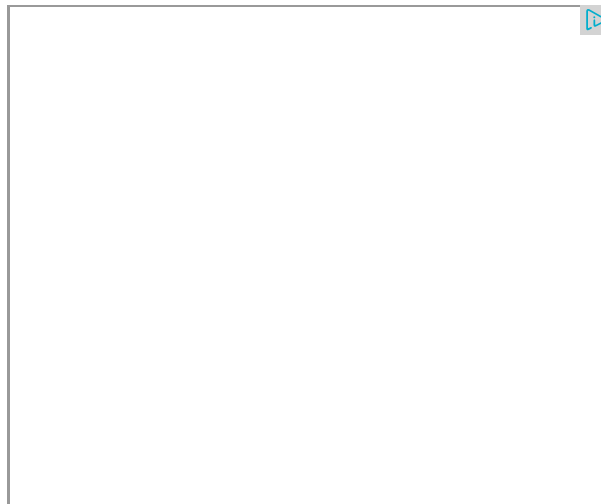
Linux OS recognizes the data part of this special file as a reference to another file path. The data in the original file can be accessed through the special file, which is called as Soft Link.

To create a soft link, do the following (ln command with -s option):

```
$ ln -s /full/path/of/original/file /full/path/of/soft/link/file
```

### Hard Link

With Hard Link, more than one file name reference the same inode number. Once you create a directory, you would see the hidden directories “.” and “..”. In this, “.” directory is hard linked to the current directory and the “..” is hard linked to the parent directory.



When you use link files, it helps us to reduce the disk space by having single copy of the original file and ease the administration tasks as the modification in original file reflects in other places.

To create a hard link, do the following (ln command with no option):

```
$ ln /full/path/of/original/file /full/path/of/hard/link/file
```

## 2. Create Symbolic Link for File or Directory

### Create a symbolic link for a File

The following examples creates a symbolic link library.so under /home/chris/lib, based on the library.so located under /home/chris/src/ directory.

```
$ cd /home/chris/lib

$ ln -s /home/chris/src/library.so library.so

$ ls -l library.so
lrwxrwxrwx 1 chris chris      21 2010-09-18 07:23 library.so -> /home/chris/src/library.so
```

## Create a symbolic link for a Directory

Just like file, you can create symbolic link for directories as shown below.

```
$ mkdir /home/chris/obj

$ cd tmp

$ ln -s /home/chris/obj objects

$ ls -l objects
lrwxrwxrwx 1 chris chris      6 2010-09-19 16:48 objects -> /home/chris/obj
```

**Note:** The inode of the original file/directory and the soft link should not be identical.

## 3. Create Hard Link for Files

The inode number for the hard linked files would be same. The hard link for files can be created as follows,

```
$ ln src_original.txt dst_link.txt

$ ls -i dst_link.txt
253564 dst_link.txt

$ ls -i src_original.txt
253564 src_original.txt
```

**Note:** Unix / Linux will not allow any user (even root) to create hard link for a directory.

## 4. Create Links Across Different Partitions

When you want to create the link across partitions, you are allowed to create only the symbolic links. Creating hard link across partitions is not allowed, as Unix can't create/maintain same inode numbers across partitions.

You would see the **“Invalid cross-device link”** error when you are trying to create a hard link file across partitions.

```
# mount /dev/sda5 /mnt

# cd /mnt

# ls
main.c Makefile

# ln Makefile /tmp/Makefile
ln: creating hard link `/tmp/Makefile' to `Makefile': Invalid cross-device link
```

And the symbolic link can be created in the same way as we did in the above.

## 5. Backup the Target Files If it Already Exists

When you create a new link (if another file exist already with the same name as the new link name), you can instruct ln command to take a backup of the original file before creating the new link using the `--backup` option as shown below.

```
$ ls
ex1.c  ex2.c

$ ln --backup -s ex1.c ex2.c

$ ls -lrt
total 8
-rw-r--r-- 1 chris chris 20 2010-09-19 16:57 ex1.c
-rw-r--r-- 1 chris chris 20 2010-09-19 16:57 ex2.c~
lrwxrwxrwx 1 chris chris  5 2010-09-19 17:02 ex2.c -> ex1.c
```

**Note:** If you don't want the backup and overwrite the existing file then use `-f` option.

## 6. Create Link Using “No-Deference” ln Command Option

While creating a new soft link, normally OS would de-reference the destination path before it creates the new soft link.

Sometimes you might not want ln command to create the new link, if the destination path is already a symbolic link that is pointing to a directory.

Following examples shows a normal way of creating soft link inside a directory.

```
$ cd ~

$ mkdir example

$ ln -s /etc/passwd example

$ cd example/

$ ls -l
total 0
lrwxrwxrwx 1 root root 16 2010-09-19 17:24 passwd -> /etc/passwd
```

In case the “example” directory in the above code-snippet is a symbolic link pointing to some other directory (for example second-dir), the ln command shown will still create the link under second-dir. If you don't want that to happen, use `ln -n` option as shown below.

```
$ cd ~

$ rm -rf example

$ mkdir second-dir

$ ln -s second-dir example

$ ln -n -s /etc/passwd example
ln: creating symbolic link `example': File exists
```

**Note:** In the above example, if you don't use the `-n` option, the link will be created under `~/second-dir` directory.

## 7. Create Link for Multiple Files at the Same Time

In the following example, there are two directories — first-dir and second-dir. The directory first-dir contains

couple of C program files. If you want to create soft links for these files in second-dir, you'll typically do it one by one. Instead, you can create soft list for multiple files together using -t option as shown below.

```
$ ls
first-dir second-dir

$ ls first-dir
ex1.c ex2.c

$ cd second-dir

$ ln -s ../first-dir/*.c -t .

$ ls -l
total 0
lrwxrwxrwx 1 chris chris 14 2010-09-19 15:20 ex1.c -> ../first-dir/ex1.c
lrwxrwxrwx 1 chris chris 14 2010-09-19 15:20 ex2.c -> ../first-dir/ex2.c
```

Keep in mind that whenever you are creating link files with -t option, it is better to go into target directory and perform the link creation process. Otherwise, you would face the broken link files as shown below.

```
$ cd first-dir

$ ln -s *.c /home/chris/second-dir

$ cd /home/chris/second-dir
$ ls -l
total 0
lrwxrwxrwx 1 chris chris 5 2010-09-19 15:26 ex1.c -> ex1.c
lrwxrwxrwx 1 chris chris 5 2010-09-19 15:26 ex2.c -> ex2.c
```

Instead, you might also use actual path for source files to create the link properly.

## 8. Removing the Original File When a Soft Link is pointing to it

When the original file referred by a soft-link is deleted, the soft link will be broken as shown below.

```
$ ln -s file.txt /tmp/link

$ ls -l /tmp/link
lrwxrwxrwx 1 chris chris 9 2010-09-19 15:38 /tmp/link -> file1.txt

$ rm file.txt

$ ls -l /tmp/link
lrwxrwxrwx 1 chris chris 9 2010-09-19 15:38 /tmp/link -> file1.txt
```

## 9. Links Help You to Increase the Partition Size Virtually

Let us assume that you have two partitions – 5GB and 20GB. The first partition does not have too much free space available in it. If a program located on the first partition needs more space (For example, for its log file), you can use some of the space from the second partition by creating a link for the log files as shown below.

Consider that partition1 is mounted on /, and partition2 is mounted to /mnt/. Let us assume that the logs that are located on partition1 is running out of space, and you've decided to move them to partition2. You can achieve this as shown below.

```
$ mkdir /mnt/logs

$ cd /logs
```

```
$ mv * /mnt/logs  
$ cd /; rmdir logs  
$ ln -s /mnt/logs logs
```

## 10. Removing the Hard Linked Files

When you delete a file that is hard linked, you would be still able to access the content of the file until you have the last file which is hard linked to it, as shown in the example below.

Create a sample file.

```
$ vim src_original.txt  
Created this file to test the hard link.
```

Create a hard link to the sample file.

```
$ ln src_original.txt dst_link.txt
```

Delete the original file.

```
$ rm src_original.txt
```

You can still access the original file content by using the hard link you created.

```
$ cat dst_link.txt  
Created this file to test the hard link.
```

 11

 2

 17

> [Add your comment](#)



Linux provides several powerful administrative tools and utilities which will help you to manage your systems effectively. If you don't know what these tools are and how to use them, you could be spending lot of time trying to perform even the basic administrative tasks. The focus of this course is to help you understand system administration tools, which will help you to become an effective Linux system administrator.

Get the [Linux Sysadmin Course](#) Now!

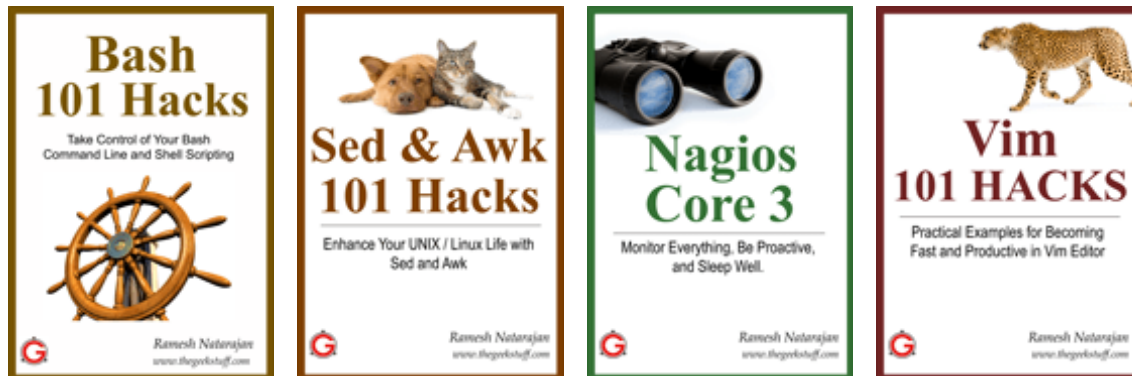
**If you enjoyed this article, you might also like..**

1. [50 Linux Sysadmin Tutorials](#)
2. [50 Most Frequently Used Linux Commands \(With Examples\)](#)
3. [Top 25 Best Linux Performance Monitoring and Debugging Tools](#)
- [Awk Introduction – 7 Awk Print Examples](#)
- [Advanced Sed Substitution Examples](#)
- [8 Essential Vim Editor Navigation Fundamentals](#)
- [25 Most Frequently Used Linux IPTables](#)

4. [Mommy, I found it! – 15 Practical Linux Find Command Examples](#)
5. [Linux 101 Hacks 2nd Edition eBook](#) **Free**

[Rules Examples](#)

- [Turbocharge PuTTY with 12 Powerful Add-Ons](#)



{ 12 comments... read them below or [add one](#) }

[1](#) al mic October 4, 2010 at 7:08 am

nice one!

do you also know how to list the links created to a file or directory?  
because sometimes i found directories in strange places (like logs in /mnt), but they could be linked from some other place, and i don't know from where

thanks

[2](#) Muddler October 4, 2010 at 7:12 am

I've been enjoying your guides for a little while now, and I want to say thank you for writing them. The section here on creating multiple links at the same time was a nice addition that I've not seen elsewhere.

[3](#) tshepo frans October 5, 2010 at 3:55 am

i am enjoying your guide its cool send us more

[4](#) SattaiLanfear October 5, 2010 at 3:14 pm

While ln can certainly do links fairly well, I've found "cp -l" and "cp -s" to work better. It can even recursively duplicate entire directory trees.

[5](#) Hank Hall October 13, 2010 at 1:53 pm

Hello,

How do you print or save the guide? I found the gudie very helpful and easy to understand.

Thanks for your help with this matter.

Hank

[6](#) eMancu October 14, 2010 at 10:25 pm

Hi dude! its a great tutorial but, I found 2 errors.

First, in item 2 “Create a symbolic link for a Directory”

You wrote

```
$ mkdir /home/chris/obj
$ cd tmp
$ ln -s objects /home/chris/obj
$ ls -l objects
```

but ln -s command should be “src/path” first and then dest/path

```
$ mkdir /home/chris/obj
$ cd tmp
$ ln -s /home/chris/obj objects
$ ls -l objects
```

---

The other mistake that I found, was on item 7, and its not a properly mistake maybe is more info to your post.

You can create multiple links with -t option without enter in a specific directory, you only need to be careful and use fixed path for every option.

A solution for your broken links could be:

```
$ ln -s /home/chris/first-dir/*.c /home/chris/second-dir/.
```

I hope this help!

[7](#) Ramesh Natarajan October 26, 2010 at 7:58 pm

@eMancu,

Regarding #2, thanks for pointing out the error. It is fixed now.

Regarding #7, yes I agree. We’ve already mentioned this in the item#7 – “Keep in mind that whenever you are creating link files with -t option, it is better to go into target directory and perform the link creation process. Otherwise, you would face the broken link files as shown below.”

[8](#) [Shantanu Oak](#) November 1, 2010 at 3:23 am

1) If the file already exists, you can forcefully create links.

```
ln -s /dev/null general.log
ln: creating symbolic link `general.log' to `/dev/null': File exists
```

In the case mentioned above I will have to use -sf option instead of -s to throw the MySQL general log to bit bucket. When I need the logs, I will remove the file using “rm” and flush logs.

2) If you are using MyISAM table types, it will have 3 files. tblName.frm, .myd and .myi If there is no enough space in the data directory, you can move the largest file (.MYD) to another location and symbolically link it.

[9](#) [Martin](#) January 19, 2011 at 4:04 am

Can’t help mentioning ..  
the ultimate link in shared root filesystems ....



In tru64 unix, exists the CLuster filesystem.

To be able to generate a link that depends on which member of the cluster you are on, engineering came up with CDSL's. Context Dependent Soft Link.

By having the variable "memb" defined, you can have for instance a log file writer writing to /var/cluster/members/{memb}/adm/messages

Other examples for location dependent configuration files:

/var/cluster/members/{memb}/etc/rc.config (network settings etc. etc.)

/var/cluster/members/{memb}/etc/sysconfigtab (Kernel params)

On a NON cluster node you always would have member0 defined, and still use the CDSL's  
HPUX today have an inherited implementation of the Cluster File system, but I think there is not a lot using it.

Regards an old DECie, working with TruClusters for a long time.

[10](#) Marcus Mota May 17, 2012 at 2:03 pm

This guide is very helpful.

Thanks!

[11](#) Linda Walsh November 15, 2013 at 10:18 pm

Even with "soft" links there can a type of invalid link: "forward links"

What happens when you put a softlink from /sbin/mount -> /usr/sbin/mount and /usr is a separate partition.

You get a system that won't boot.

I also have /usr/share being a link on /usr (it's actually a bind from /home/share) — /usr just got too full).

You'll find alot of these bad "forward" links now on systems switching to "systemd", due to it being unable to use the linux system path to look up binaries — it wanted them all in /usr. So they are moving boot related binaries off of /root on to /usr.

Their solution/requirement — you have to use and boot from a ramdisk first, and booting directly from your hard disk is no longer supported.

Wonderful!...

\*sigh\*

[12](#) Chris G. Sellers February 5, 2014 at 9:44 am

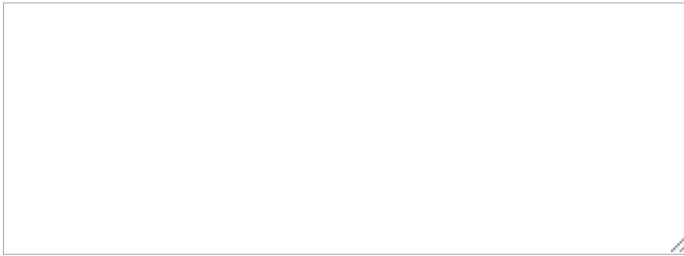
The term "Soft" link is a misnomer and the term is really Symbolic link. Your title should reflect that.  
Thank you for your efforts to help share knowledge.

Leave a Comment

Name

E-mail

Website



☐ Notify me of followup comments via e-mail

Submit

Previous post: [How To: 2 Methods To Change TimeZone in Linux](#)

Next post: [PostgreSQL Trigger Tutorial with EMP Table Examples](#)

- [RSS](#) | [Email](#) | [Twitter](#) | [Facebook](#) | [Google+](#)

Search



- 
- **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
  - [Sed and Awk 101 Hacks eBook](#) - Enhance Your UNIX / Linux Life with Sed and Awk
  - [Vim 101 Hacks eBook](#) - Practical Examples for Becoming Fast and Productive in Vim Editor
  - [Nagios Core 3 eBook](#) - Monitor Everything, Be Proactive, and Sleep Well



## • 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 Tutorials](#)
- [Vim Editor](#)
- [Sed Scripting](#)
- [Awk Scripting](#)
- [Bash Shell Scripting](#)
- [Nagios Monitoring](#)
- [OpenSSH](#)
- [IPTables Firewall](#)
- [Apache Web Server](#)
- [MySQL Database](#)
- [Perl Programming](#)
- [Google Tutorials](#)
- [Ubuntu Tutorials](#)
- [PostgreSQL DB](#)
- [Hello World Examples](#)
- [C Programming](#)
- [C++ Programming](#)
- [DELL Server Tutorials](#)
- [Oracle Database](#)
- [VMware Tutorials](#)

Ramesh Natarajan



Follow

- **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**

Support this blog by purchasing one of my ebooks.

[Bash 101 Hacks eBook](#)

[Sed and Awk 101 Hacks eBook](#)

[Vim 101 Hacks eBook](#)

[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 Google+](#)

[Follow us on Twitter](#)

[Become a fan on Facebook](#)

Copyright © 2008–2014 Ramesh Natarajan. All rights reserved | [Terms of Service](#)