Master the Red Hat Challenge: Your Ultimate Guide to RHCE Certification Success!

How to Install and Run Multiple glibc Libraries in Linux

Ravi Saive | June 13, 2024 | Read Time: 2 mins | Linux Commands | 3 Comments

The GNU C Library (glibc) is a fundamental component of the Linux operating system, providing essential functionality for various applications.

In some cases, you may need to install multiple versions of glibc to support different applications or to ensure compatibility with older software.

This guide will show you how to install and manage multiple glibc libraries on your Linux system using simple steps.

Understanding the Need for Multiple glibc Libraries

The primary reason for installing multiple glibc libraries is to support different applications that require different versions of the library.

This can occur when you need to run older software that is not compatible with the latest version of glibc. For instance, some older applications might rely on specific features or functions that were available in earlier versions of glibc.

Step 1: Installing the Required Packages

Before installing multiple glibc libraries, ensure that your system is up to date.

```
sudo apt update    [On Debian-based Systems]
sudo yum update    [On RedHat-based Systems]
```

Next, install the required necessary packages for building and installing glibc on Debian-based distributions.

```
sudo apt install build-essential libssl-dev libgdbm-dev libdb-dev libexpat-
```

Install Required Packages

For RedHat-based distributions, install the Development Tools package.

```
sudo yum groupinstall "Development Tools"
```

Step 2: Installing Multiple glibc Libraries

To install multiple glibc libraries, you will need to compile and install each version separately as shown.

First, download the required versions of glibc from the official glibc repository or a reliable source, or use the following wget command to download the glibc 2.39 and glibc 2.38 versions.

```
wget https://ftp.gnu.org/gnu/glibc/glibc-2.39.tar.xz
wget https://ftp.gnu.org/gnu/glibc/glibc-2.38.tar.bz2
```

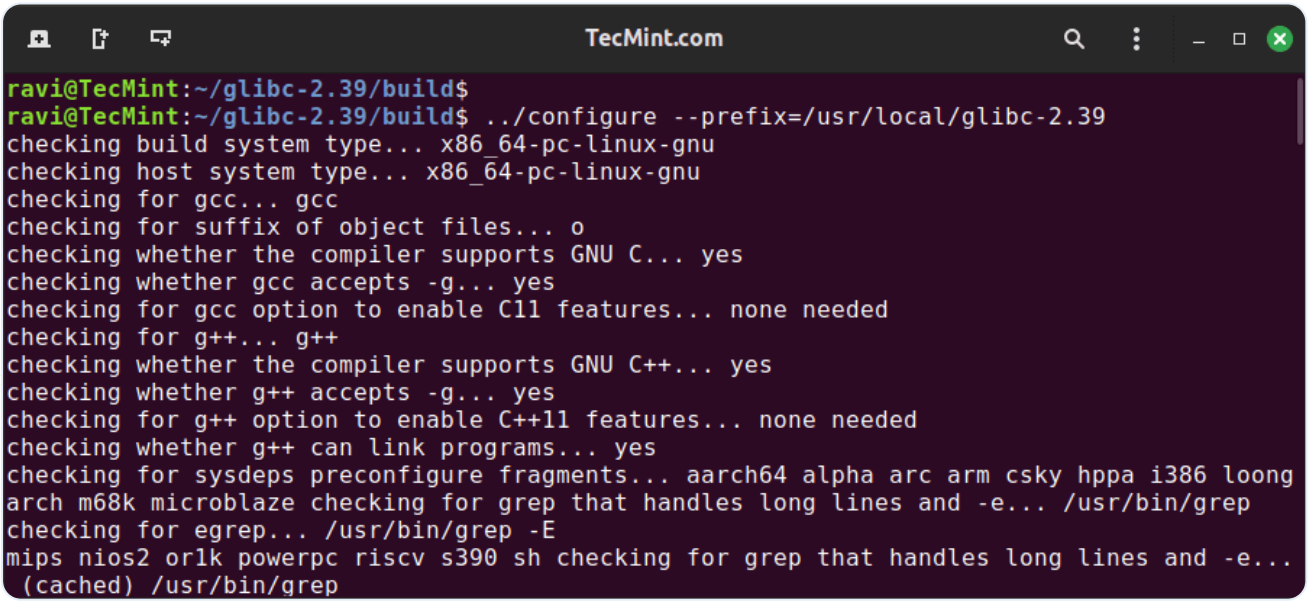
Next, extract the downloaded source code using the following tar command, which will create a new directory containing the **glibc** source code.

```
tar -xvf glibc-2.39.tar.xz
tar -xvf glibc-2.38.tar.xz
```

Navigate to the extracted directory and run the following command to configure the build and install **glibc**, which allows you to install multiple versions of **glibc** without conflicts.

For glibc 2.39:

```
cd glibc-2.39
mkdir build
cd build
../configure --prefix=/usr/local/glibc-2.39
make -j4
sudo make install
```



Build Glibc in Linux

For glibc 2.38:

```
cd glibc-2.38
mkdir build
cd build
../configure --prefix=/usr/local/glibc-2.38
make -j4
sudo make install
```

Managing Multiple glibc Libraries

After installing multiple **glibc** libraries, you need to manage them to ensure that the correct version is used by your applications.

To use a specific **glibc** version, you need to set the LD_LIBRARY_PATH environment variable.

For **glibc 2.39**:

```
export LD_LIBRARY_PATH=/usr/local/glibc-2.39/lib:$LD_LIBRARY_PATH
```

For **glibc 2.38**:

```
export LD_LIBRARY_PATH=/usr/local/glibc-2.38/lib:$LD_LIBRARY_PATH
```

You can add these lines to your `.bashrc` or `.profile` file to make the changes permanent.

Finally, check the **glibc** version to ensure it's correctly installed.

```
/usr/local/glibc-2.39/lib/ld-2.30.so --version
/usr/local/glibc-2.38/lib/ld-2.31.so --version
```

Run Applications with Specific glibc Versions

To run an application with a specific **glibc** version, use the `LD_PRELOAD` environment variable.

For example, to run an application with **glibc 2.39**, use.

```
LD_PRELOAD=/usr/local/glibc-2.39/lib/ld-2.30.so ./your_application
```

Conclusion

By following these steps, you can install and manage multiple **glibc** versions on your Linux system, which allows you to run applications that depend on different **glibc** versions without interfering with your system's default libraries.

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:

Firefox 127 on Linux: New Features, Install Guide & More

NEXT ARTICLE:

How to Install vTiger CRM on Ubuntu 24.04

Ravi Saive

I am an experienced GNU/Linux expert and a full-stack software developer with over a decade in the field of Linux and Open Source technologies

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

Join the TecMint Weekly Newsletter (More Than 156,129 Linux Enthusiasts Have Subscribed)

Was this article helpful? Please add a comment or buy me a coffee to show your appreciation.

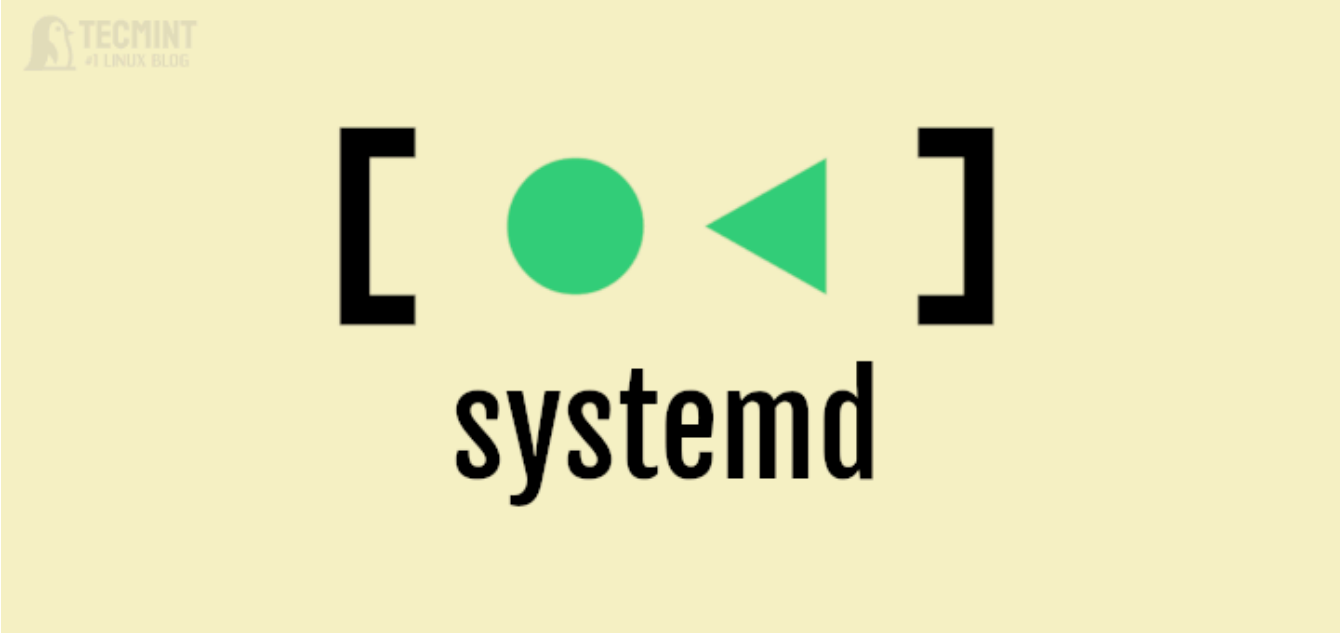
Related Posts

How to Search Files by Name or Extension Using find Command

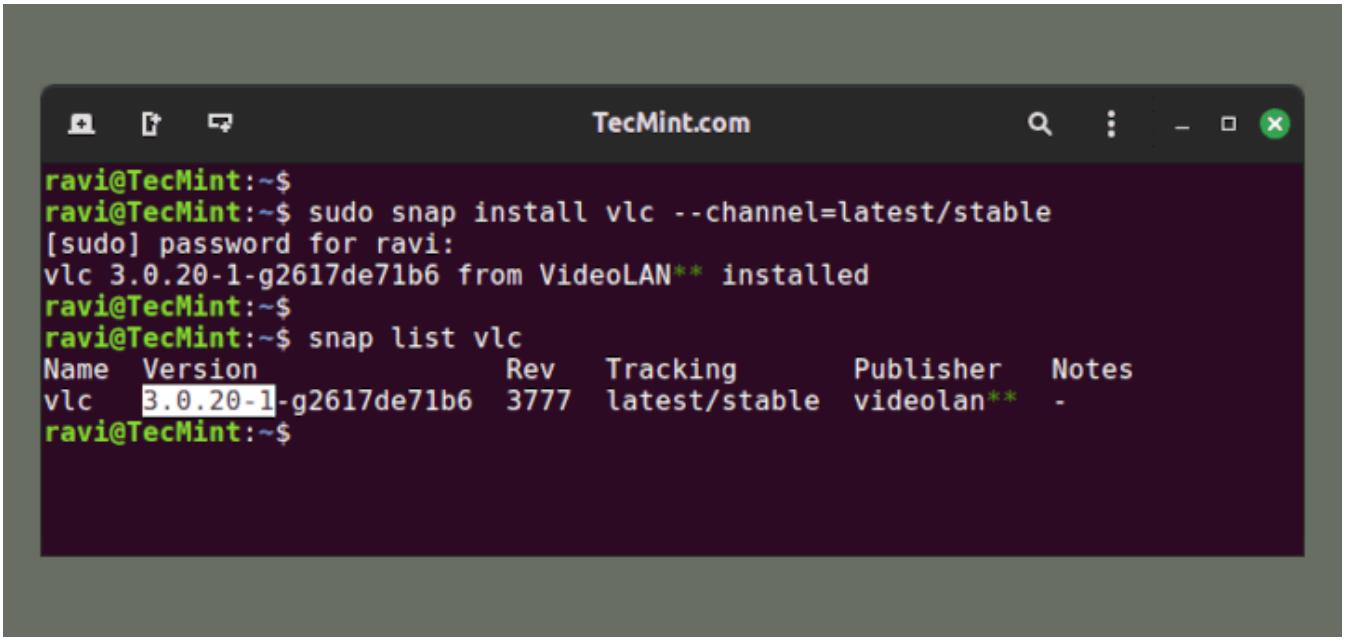
10 Lesser Known Linux Commands – Part 2

11 Lesser Known Useful Linux Commands

26 Security Hardening Tips for Modern Linux Servers



How to Remove Systemd Services on Linux



How to Install Particular Versions of Packages with Snap

3 Comments

Leave a Reply

Stefano Cacciatore

August 7, 2024 at 2:15 pm

Hi,

I followed the script and was able to install both **glibc-2.39** and **glibc-2.38**, but I do not have the file **ld-2.30.so**.

I tried to use the code with the file **ld-linux-x86-64.so.2**, but it failed.

```
LD_PRELOAD=/usr/local/glibc-2.38/lib/ld-linux-x86-64.so.2
/opt/python/${PYTHON_VERSION}/bin/pip install ipykernel
```

Segmentation fault (core dumped)

Reply

Author

Ravi Saive

August 8, 2024 at 8:00 am

@Stefano,

This could be due to compatibility issues between the versions of **glibc** and your **Python** installation.

Verify that the version of **ld-linux-x86-64.so.2** you are using is compatible with your installed glibc versions in the same directory.

```
ls /usr/local/glibc-2.38/lib/ld-linux-x86-64.so.2
```

You can try setting the `LD_LIBRARY_PATH` to include the **glibc** library path before running the **pip install** command.

```
export LD_LIBRARY_PATH=/usr/local/glibc-2.38/lib:$LD_LIBRARY_PATH
/opt/python/${PYTHON_VERSION}/bin/pip install ipykernel
```

Another approach is to create a symbolic link to the required version of ld-linux-x86-64.so.2 in the directory where your ld-2.30.so is expected.

```
ln -s /usr/local/glibc-2.38/lib/ld-linux-x86-64.so.2 /usr/local/glibc-2.38/lib/ld-2.30.so
```

If the segmentation fault persists, you may want to debug it using gdb.

```
gdb --args /opt/python/${PYTHON_VERSION}/bin/python -m pip install ipykernel
```

Reply

Jalal Hajigholamali
July 8, 2024 at 2:28 pm

Hi,
Thanks a lot
very useful 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

How to Find Files With SUID and SGID Permissions in Linux

Teleconsole – Share Your Linux Terminal with Your Friends

How to Convert Images to WebP Format in Linux

4 Useful Tools to Run Commands on Multiple Linux Servers

How to Add a New Disk to an Existing Linux Server

Understand Linux Shell and Basic Shell Scripting Tips – Part I

Linux Server Monitoring Tools

How to Monitor Apache Web Server Load and Page Statistics

How to Install Nagios Core in Rocky LInux and AlmaLinux

How to Install Cacti (Network Monitoring) Tool on RHEL Systems

10 Tips On How to Use Wireshark to Analyze Packets in Your Network

IPTraf-ng – A Console-Based Network Monitoring Tool

How to Boost Linux Server Internet Speed with TCP BBR

Learn Linux Tricks & Tips

An Easy Way to Hide Files and Directories in Linux

How to Restore Deleted /tmp Directory in Linux

5 Ways to Find a 'Binary Command' Description and Location on File System

Learn Why 'less' is Faster Than 'more' Command for Effective File Navigation

How to Configure Custom SSH Connections to Simplify Remote Access

4 Ways to Generate a Strong Pre-Shared Key (PSK) in Linux

Best Linux Tools

15 Best Free and Open Source Software (FOSS) Programs for Linux

10 Best File Comparison and Difference (Diff) Tools for Linux

4 Best Twitter Clients for Linux in 2024 (Updated)

Top 7 Apps to Install for Your Nextcloud Instance

16 Free and Open Source Video Players for Linux in 2024

8 Useful GUI Email Clients for Linux Desktop

Tecmint: Linux Howtos, Tutorials & Guides © 2024. All Rights Reserved.
The material in this site cannot be republished either online or offline, without our permission.

Hosting Sponsored by : **Linode Cloud Hosting**