# Compiling calcurse 4.8.0 from Source on Mint 21.1

By `kalwisti`

Feb. 2023

## Contents

## 1 Introduction

If you are familiar with calcurse, the text-based calendar / scheduling application, you probably know that it has experimental CalDAV support which allows you to synchronize calcurse with your mobile devices, as well as with Google Calendar and Apple Calendar. However, in order to configure this, you need the most recent version of calcurse.[1] The version in the official Mint 21 repositories is somewhat old (4.6.0, released in March 2020). The current version of the program is 4.8.0 (released in April 2022). To the best of my knowledge, there is no PPA or Flatpak available, so if you want to run the most recent version, you must compile it yourself from source.

The process is not as intimidating as it sounds; I was pleasantly surprised that I was able to compile calcurse 4.8.0 without difficulty on Mint 21.1 (both the Cinnamon and MATE edition). If you are interested in trying, here is a summary of the steps that I followed.

---

[1]The `calcurse-caldav` script is part of the latest release. To set up CalDAV synchronization, you should read the project's documentation at: https://www.calcurse.org/files/calcurse-caldav.html

> ✏️ **Caveats:**
>
> I should mention that by compiling your own version of calcurse, the program will not be upgraded automatically.
>
> It is also possible that the functionality of your compiled calcurse package might break when its dependencies and software are upgraded during your normal update process. If this happens, you might need to recompile. Using self-compiled software means that you are responsible for maintaining it, as well as the libraries that it depends on.
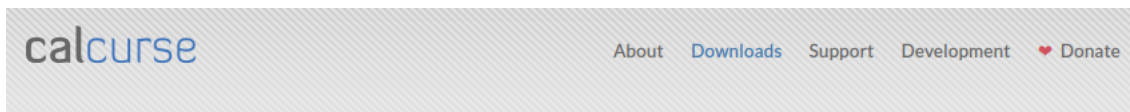
That said, compiling calcurse from source is safe and unlikely to break your system. It will give you a sense of accomplishment when you check your schedule with version 4.8.0 …

## 2  Download the Source Files

Go to the calcurse website

https://www.calcurse.org/downloads/

and download the `calcurse-4.8.0.tar.gz` file (716 kB in size).



## 3  Move File to `/home` Directory and Extract It

Move the tarball (i.e., the compressed calcurse .tar file) that you just downloaded to your /home folder.

Then, use Archive Manager — or a similar utility — to extract the compressed files. The contents of your decompressed calcurse folder will look like this:
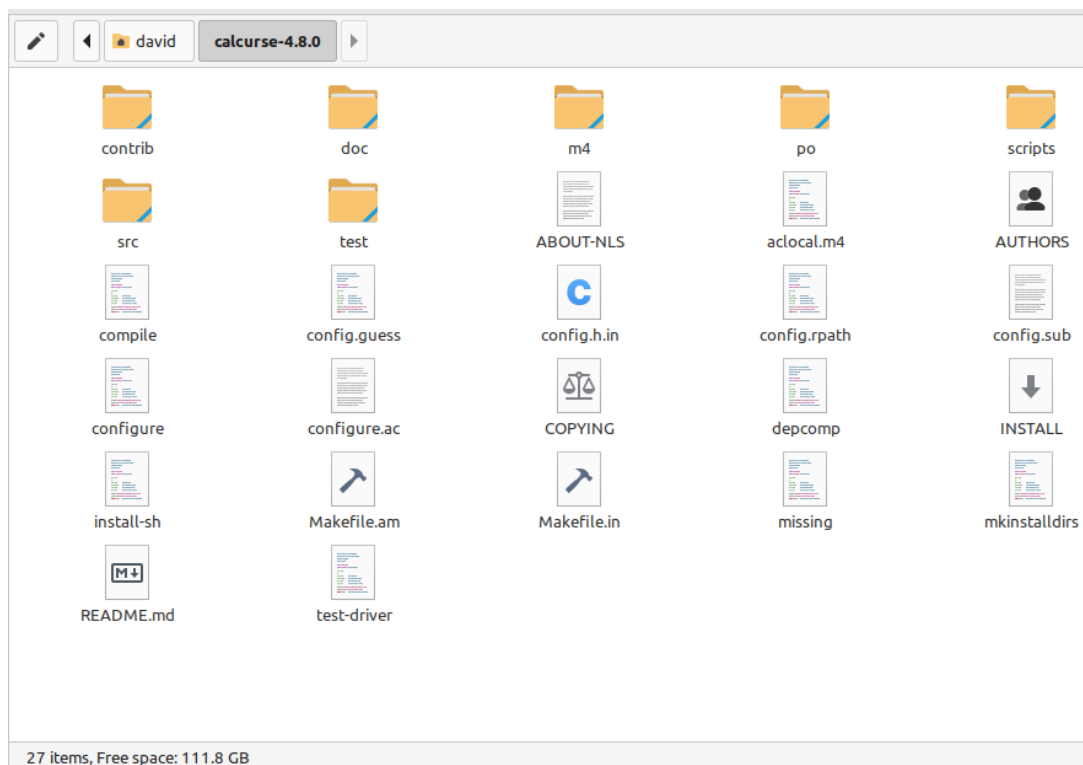


## 4   Install Required Packages and Dependencies

Use the Synaptic Package Manager to install the following packages, if they are not already present on your system. (The list below is not exhaustive, as some of these packages will pull in dependencies of their own):

| Package Name | Version No. |
|---|---|
| build-essential | 12.9ubuntu3 |
| checkinstall | 1.6.2+git20170426… |
| gcc-11 | 11.3.0-1ubuntu1 |
| autoconf-archive | 20210219-2.1 |
| autoconf | 2.71-2 |
| automake | 1:1.16.5-1.3 |
| autopoint | 0.21-4ubuntu4 |
| gettext | 0.21-4ubuntu4 |
| ncurses-base | 6.3-2 |
| ncurses-bin | 6.3-2 |
| libncurses5 | 6.3-2 |
| libncurses5-dev | 6.3-2 |
| libncursesw5-dev | 6.3-2 |

**Caution:**

The README file in calcurse's directory mentions the possibility of also installing the asciidoc package. However, I chose *not* to install it when Synaptic indicated that asciidoc would pull in 115 dependencies—including many of the huge texlive (TEX Live 2021) packages. Leaving out asciidoc does not seem to have harmed calcurse, so I recommend omitting this package.

## 5   Compile the Source Code

Now you are ready to compile calcurse's source code.

Open a Terminal and navigate to the extracted calcurse folder in your /home directory:

```
$ cd calcurse-4.8.0
```

Type:

```
$ sudo ./configure
```

(Remember to include the initial period and slash (" ./ ") before the `configure` command.)

This step will perform checks / tasks that will allow your system to compile the software successfully. If the configure script detects a missing package, it will notify you of what to do at the end of the script. You will need to install any missing packages that the configure script has detected.

When that process finishes, you will see a message similar to the one below:

```
=======================================================================
calcurse is configured as follows.
Please check that this configuration matches your expectations.
Also give a look at the config.h file to check for preprocessor symbols.

Host system type   : x86_64-pc-linux-gnu

Options used to compile and link:
  PREFIX            = /usr/local
  VERSION           = 4.8.0
  CC                = gcc
  CFLAGS            = -g -O2 -Wall
  CPPFLAGS          =
  DEFS              = -DHAVE_CONFIG_H
  LD                = /usr/bin/ld
  LDFLAGS           =
  LIBS              =  -lncursesw -ltinfo -pthread -lm
=======================================================================
```

Next, issue the command:

```
$ sudo make
```

The make command will actually begin compiling calcurse. Since it is a small program, the compilation will only take a couple of minutes.

The final step is to install your now-compiled calcurse package with the command:

```
$ sudo checkinstall
```

Linux Mint Forums user xenopeek explains the advantage of using "sudo checkinstall" instead of the more common "sudo make install" command:

"[It] will make the software and its files known to your package managers (e.g., Software Manager and apt) so that you can easily remove the software or replace it with a newer version."[2]

As checkinstall runs, follow its on-screen instructions and prompts. I was initially asked about the creation of a set of package documents:

```
The package documentation directory ./doc-pak does not exist.
Should I create a default set of package docs?  [y]:
```

I answered "y" [yes] to this question.

You will also be asked to fill out some requested information fields. (I left the majority of the fields blank; however, it did not prevent the installation from completing nor has it adversely affected calcurse's performance).

Sample output is pasted below:

```
Preparing package documentation...OK

Please write a description for the package.
End your description with an empty line or EOF.
>>
```

I typed "Text-based (ncurses) calendar app" and pressed Enter.

```
*****************************************
**** Debian package creation selected ***
*****************************************

This package will be built according to these values:

0 -  Maintainer: [ root@david-ThinkCentre-M800 ]
1 -  Summary: [ Text-based (ncurses) calendar app ]
2 -  Name:    [ calcurse ]
3 -  Version: [ 4.8.0 ]
4 -  Release: [ 1 ]
5 -  License: [ GPL ]
```

[2]https://forums.linuxmint.com/viewtopic.php?t=230142. 20 Sept. 2016.

```
6 -  Group:    [ checkinstall ]
7 -  Architecture: [ amd64 ]
8 -  Source location: [ calcurse-4.8.0 ]
9 -  Alternate source location: [   ]
10 - Requires: [   ]
11 - Recommends: [   ]
12 - Suggests: [   ]
13 - Provides: [ calcurse ]
14 - Conflicts: [   ]
15 - Replaces: [   ]


Enter a number to change any of them or press ENTER to continue:
```

There were two additional prompts which needed a response. The first was:

```
Some of the files created by the installation are inside the home directory: /home

You probably don't want them to be included in the package.
Do you want me to list them?  [n]:
```

Answer "n" [no] to this question.

The final question appeared:

```
Should I exclude them from the package? (Saying yes is a good idea)  [n]:
```

As the prompt suggests, you should answer "yes" here.

After you press the Enter key, a wall of text will scroll rapidly by. When the checkinstall process completed, the Terminal displayed the following message:

```
Done. The new package has been installed and saved to

 /home/david/calcurse-4.8.0/calcurse_4.8.0-1_amd64.deb

 You can remove it from your system anytime using:

   dpkg -r calcurse
```
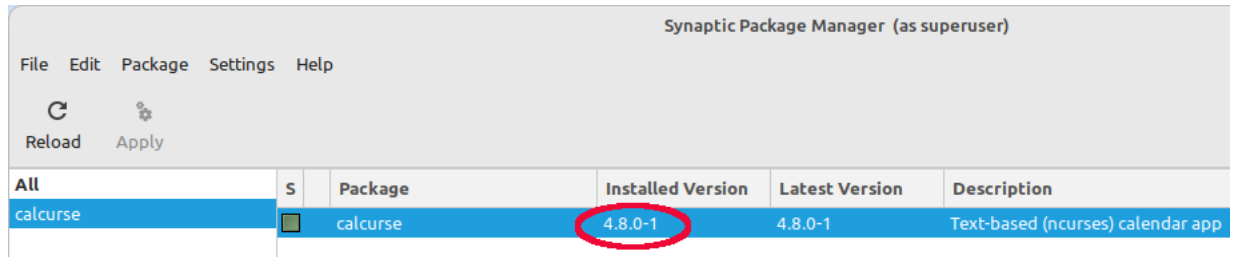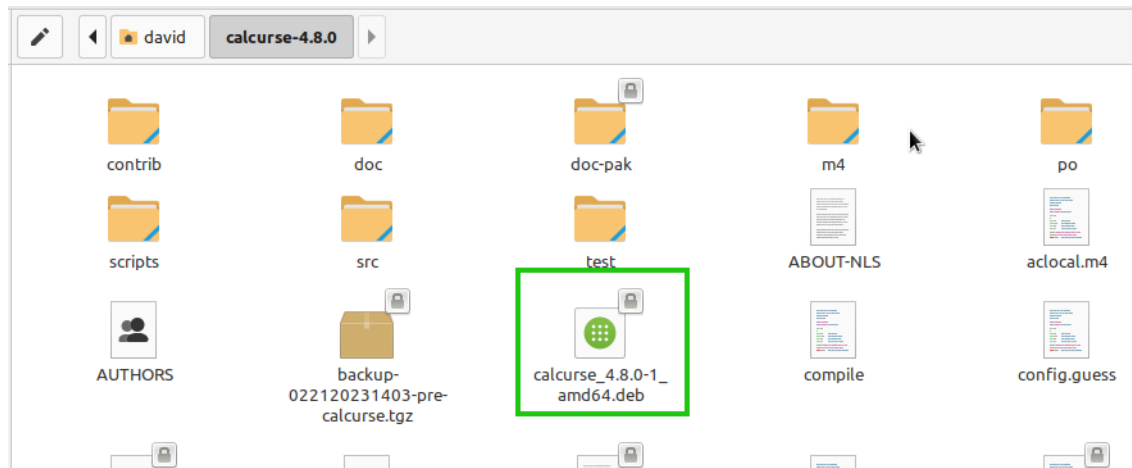
As xenopeek mentioned, Synaptic is aware of the new calcurse package and displays that it has been installed:

In addition, a 64-bit .deb package has been saved to my `calcurse-4.8.0` source folder:

Good luck compiling and Have fun with calcurse!