

How to write and run C programs in Linux

C is one of the oldest, most widely used programming languages in the world. It has been used to develop countless applications, from operating systems to embedded devices.

C is one of the oldest, most widely used programming languages in the world. It has been used to develop countless applications, from operating systems to embedded devices. Even today, many developers still rely on C for its versatility and reliability as a programming language.

Install C Compiler (GCC) on Linux

The steps and commands mentioned in this article are on Ubuntu 22.04 LTS system, but it also works the same on other versions like Ubuntu 20.04 or Debian 11.

To compile a simple C program, we use the Linux command line tool, Terminal. To open the terminal, you can use Ubuntu Dash or the key combination Ctrl+Alt+T.

Install the necessary build packages

To compile and execute a C program, you need to install essential packages on your system. Enter the following command as root in your Linux Terminal:

```
$ sudo apt install build-essential
```

```

administrator@server1:~$ sudo apt install build-essential
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  cpp cpp-11 dpkg-dev fakeroot fontconfig-config fonts-dejavu-core g++ g++-11 gcc gcc-11 gcc-11-base
  libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan6 libatomic1 libc-dev-bin
  libc-devtools libc6-dev libcc1-0 libcrypt-dev libdeflate0 libdpkg-perl libfakeroot libfile-fcntllock-perl
  libfontconfig1 libgcc-11-dev libgd3 libgomp1 libisl23 libitm1 libjbig0 libjpeg-turbo8 libjpeg8 liblsan0
  libmpc3 libnsl-dev libquadmath0 libstdc++-11-dev libtiff5 libtirpc-dev libtsan0 libubsan1 libwebp7 libxpm4
  linux-libc-dev lto-disabled-list make manpages manpages-dev patch rpcsvc-proto
Suggested packages:
  cpp-doc gcc-11-locales debian-keyring g++-multilib g++-11-multilib gcc-11-doc gcc-multilib autoconf automake
  libtool flex bison gdb gcc-doc gcc-11-multilib libstdc++-11-doc gfortran-doc g++-11-doc libstdc++-11-doc
  man-browser ed diffutils-doc
The following NEW packages will be installed:
  build-essential cpp cpp-11 dpkg-dev fakeroot fontconfig-config fonts-dejavu-core g++ g++-11 gcc gcc-11
  gcc-11-base libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan6 libatomic1
  libc-dev-bin libc-devtools libc6-dev libcc1-0 libcrypt-dev libdeflate0 libdpkg-perl libfakeroot
  libfile-fcntllock-perl libfontconfig1 libgcc-11-dev libgd3 libgomp1 libisl23 libitm1 libjbig0 libjpeg-turbo8
  libjpeg8 liblsan0 libmpc3 libnsl-dev libquadmath0 libstdc++-11-dev libtiff5 libtirpc-dev libtsan0 libubsan1
  libwebp7 libxpm4 linux-libc-dev lto-disabled-list make manpages manpages-dev patch rpcsvc-proto
0 upgraded, 53 newly installed, 0 to remove and 1 not upgraded.
Need to get 65.1 MB of archives.

```

You will be asked to enter the root password; The installation process will then begin. Please make sure you are connected to the Internet.

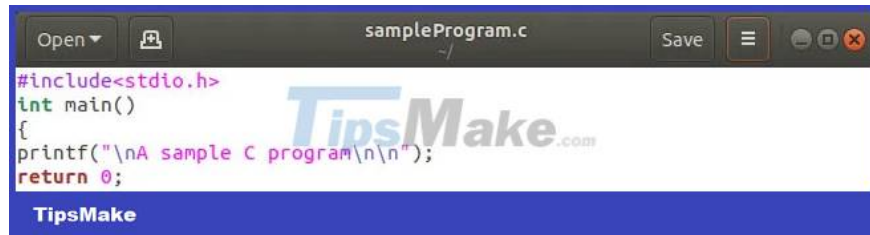
Write a simple C program

After installing the necessary packages, write a simple C program.

Open Ubuntu's graphical Text Editor and write or copy the following sample program into it:

```
#include <stdio.h>
int main() { printf("\nA sample C program\n"); return 0; }
```

Then save the file with the extension `.c`. This example names its C program `sampleProgram.c`



Additionally, you can write a C program through Terminal in `gedit` as follows:

```
$ gedit sampleProgram.c
```

This will create a `.c` file to write and save the program.

Compile C programs with GCC Compiler

In your Terminal, enter the following command to create an executable version of the program you wrote:

Syntax:

```
$ gcc [programName].c -o programName
```

For example:

```
$ gcc sampleProgram.c -o sampleProgram
```



Make sure your program is placed in the Home folder. If not, you will need to specify the appropriate paths in this command.

Run the program

The final step is to run the compiled C program. Use the following syntax to do so:

```
$ ./programName
```

For example:

```
$ ./sampleProgram
```

```
administrator@server1:~$ ./sampleProgram
A sample C program
TipsMake.com
TipsMake
```

You can see how the program is executed in the example above, displaying the written text to print through.

This article shows you how to write, compile, and run a simple C program in Linux. All you need are the necessary packages and skills to turn you into a programming expert in Linux!

You finished reading the article "**How to write and run C programs in Linux**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.