

CWR 2020

Working under Ubuntu

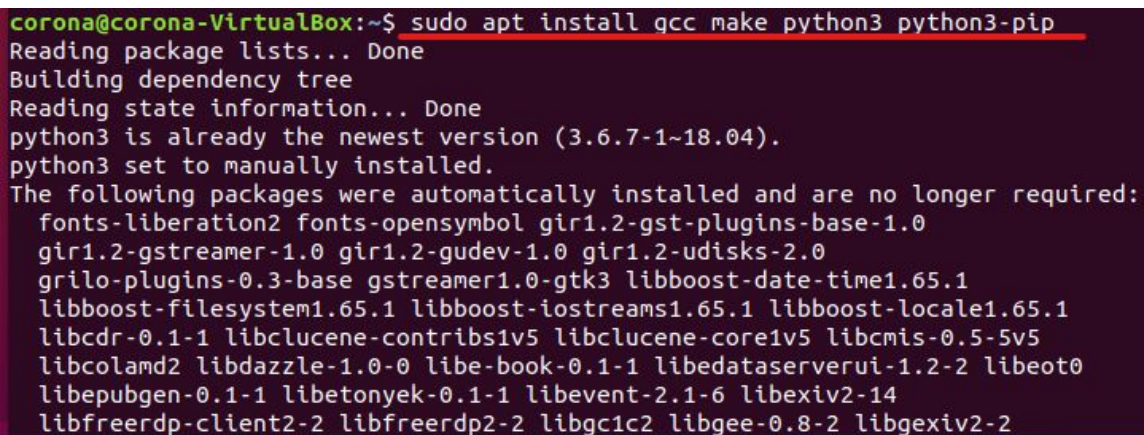
Here we will demonstrate how you can get a C compiler and Python running under a **fresh minimal install** of Ubuntu. We will install all essential tools that you'll need throughout this semester's CWR course. You should be able to adapt this to other Debian based systems.

Installing Compiler and Python

Open a terminal window (for example with CTRL+ALT+T). Install gcc, make, python and pip using the command

```
sudo apt install gcc make python3 python3-pip
```

You'll have to enter your password.



```
corona@corona-VirtualBox:~$ sudo apt install gcc make python3 python3-pip
Reading package lists... Done
Building dependency tree
Reading state information... Done
python3 is already the newest version (3.6.7-1~18.04).
python3 set to manually installed.
The following packages were automatically installed and are no longer required:
 fonts-liberation2 fonts-opensymbol gir1.2-gst-plugins-base-1.0
 gir1.2-gstreamer-1.0 gir1.2-gudev-1.0 gir1.2-udisks-2.0
 grilo-plugins-0.3-base gstreamer1.0-gtk3 libboost-date-time1.65.1
 libboost-filesystem1.65.1 libboost-iostreams1.65.1 libboost-locale1.65.1
 libcdr-0.1-1 libclucene-contribs1v5 libclucene-core1v5 libcmis-0.5-5v5
 libcolamd2 libdazzle-1.0-0 libe-book-0.1-1 libdataserverui-1.2-2 libeot0
 libepubgen-0.1-1 libetonyek-0.1-1 libevent-2.1-6 libexiv2-14
 libfreerdp-client2-2 libfreerdp2-2 libgc1c2 libgee-0.8-2 libgexiv2-2
```

After the installation is complete, enter the command

```
gcc -v
```

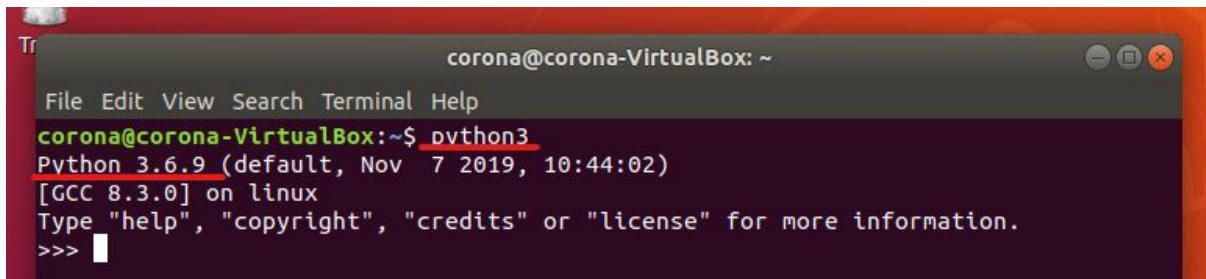
To check that the C compiler is installed correctly.

```
corona@corona-VirtualBox:~$ gcc -v
Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/lib/gcc/x86_64-linux-gnu/7/lto-wrapper
OFFLOAD_TARGET_NAMES=nvptx-none
OFFLOAD_TARGET_DEFAULT=1
Target: x86_64-linux-gnu
Configured with: ../src/configure -v --with-pkgversion='Ubuntu 7.5.0-3ubuntu1~18.04' --with-bugurl=file:///usr/share/doc/gcc-7/README.Bugs --enable-languages=c,ada,c++,go,brig,d,fortran,objc,obj-c++ --prefix=/usr --with-gcc-major-version-only --program-suffix=-7 --program-prefix=x86_64-linux-gnu- --enable-shared --enable-linker-build-id --libexecdir=/usr/lib --without-included-gettext --enable-threads=posix --libdir=/usr/lib --enable-nls --enable-bootstrap --enable-clocale=gnu --enable-libstdcxx-debug --enable-libstdcxx-time=yes --with-default-libstdcxx-abi=new --enable-gnu-unique-object --disable-vtable-verify --enable-libmpx --enable-plugin --enable-default-pie --with-system-zlib --with-target-system-zlib --enable-objc-gc=auto --enable-multiarch --disable-werror --with-arch-32=i686 --with-abi=m64 --with-multilib-list=m32,m64,mx32 --enable-multilib --with-tune=generic --enable-offload-targets=nvptx-none --without-cuda-driver --enable-checking=release --build=x86_64-linux-gnu --host=x86_64-linux-gnu --target=x86_64-linux-gnu
Thread model: posix
gcc version 7.5.0 (Ubuntu 7.5.0-3ubuntu1~18.04)
corona@corona-VirtualBox:~$
```

Start Python using the command

```
python3
```

to check if it is installed correctly.



```
corona@corona-VirtualBox: ~
File Edit View Search Terminal Help
corona@corona-VirtualBox:~$ python3
Python 3.6.9 (default, Nov 7 2019, 10:44:02)
[GCC 8.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Exit python with the command

```
exit()
```

Install the GNU Scientific Library with

```
sudo apt install libgsl-dev
```

Install NumPy, SciPy and Matplotlib with

```
pip3 install numpy scipy matplotlib
```

Checking your Installation

Create a new folder “test” in any location you like using the command

```
mkdir test
```

Create a new source file using the command

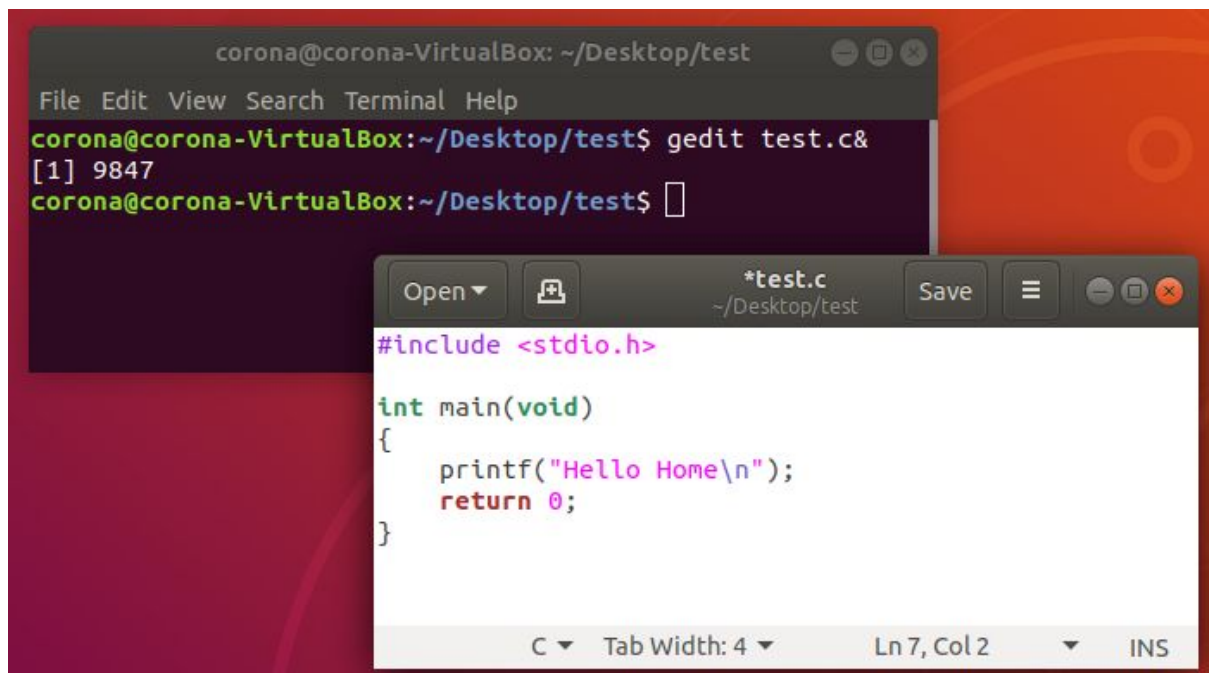
```
touch test.c
```

and open it with

```
gedit test.c&
```

The ampersand (“&”) will detach your bash input from the gedit process, so you can still enter commands.

Write the following code into the source file and save it:



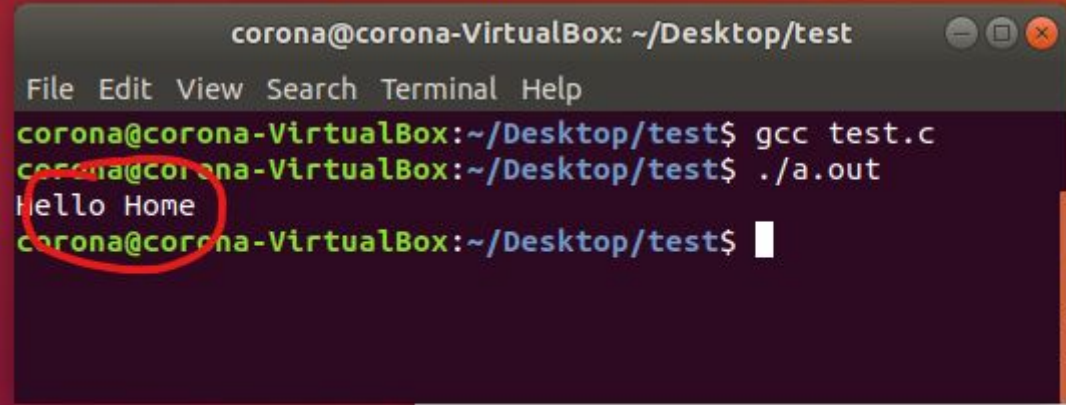
Then, enter the command

```
gcc test.c
```

in the terminal window to compile the source into the binary “a.out”. Enter the command

`./a.out`

to execute it. It should output the words "Hello Home" to the terminal.



The screenshot shows a terminal window titled "corona@corona-VirtualBox: ~/Desktop/test". The terminal has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The command prompt shows the user has compiled a program with `gcc test.c` and then executed it with `./a.out`. The output of the program is "Hello Home", which is circled in red. The prompt then returns to `corona@corona-VirtualBox:~/Desktop/test$`. Below the terminal window, a snippet of the C source code is visible, showing the `main` function using `printf` to output "Hello Home\n".

```
corona@corona-VirtualBox: ~/Desktop/test
File Edit View Search Terminal Help
corona@corona-VirtualBox:~/Desktop/test$ gcc test.c
corona@corona-VirtualBox:~/Desktop/test$ ./a.out
Hello Home
corona@corona-VirtualBox:~/Desktop/test$
```

```
int main(void)
{
    printf("Hello Home\n");
    return 0;
}
```

Exchange the source code for the following code to test GSL:

```
#include <stdio.h>
#include <gsl/gsl_sf_bessel.h>

int main (void)
{
    double x = 5.0;
    double y = gsl_sf_bessel_J0 (x);
    printf ("J0(%g) = %.18e\n", x, y);
    return 0;
}
```

Recompile the code using the command

```
gcc test.c -lgsl -lgslcblas
```

and run the program. Check that it outputs the correct value for the Bessel function:


```
File Edit View Search Terminal Help
corona@corona-VirtualBox:~/Desktop/test$ gcc test.c -lgsl -lgslcblas
corona@corona-VirtualBox:~/Desktop/test$ ./a.out
J0(5) = -1.775967713143382642e-01
corona@corona-VirtualBox:~/Desktop/test$
```

Open

test.c
~/Desktop/test

Save

```
#include <stdio.h>
#include <gsl/gsl_sf_bessel.h>

int main(void)
{
```

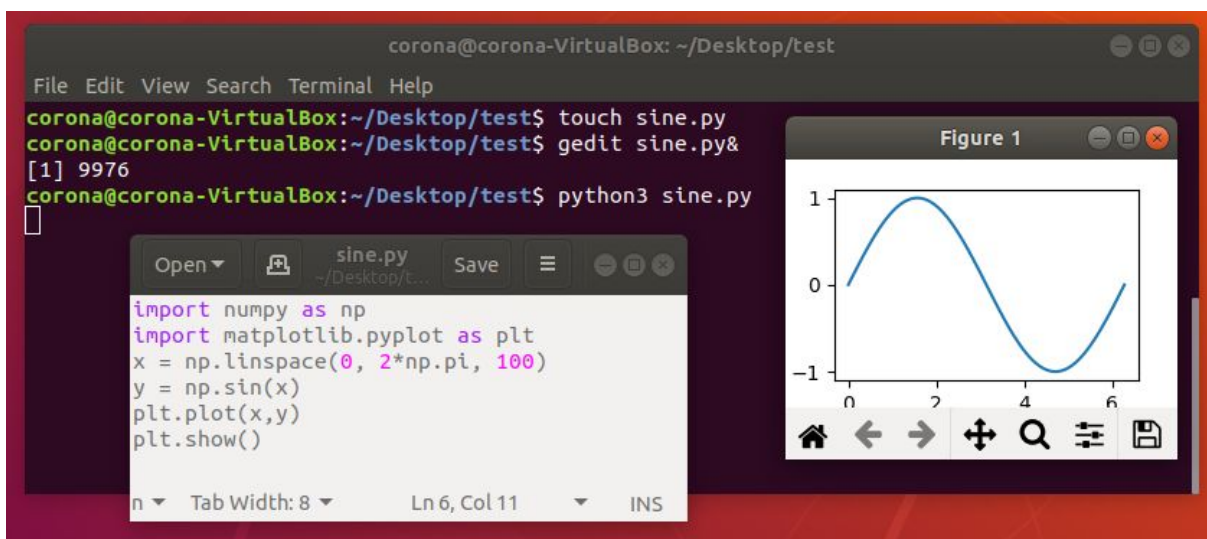
Create a new file called "sine.py" and insert the following code:

```
import numpy as np
import matplotlib.pyplot as plt
x = np.linspace(0, 2*np.pi, 100)
y = np.sin(x)
plt.plot(x,y)
plt.show()
```

Run the script using the command

```
python3 sine.py
```

and make sure that it draws a sine curve.



Congratulations, you are ready for the course :-) Good luck and stay safe!