

Writing C programs as a McGill student

This guide enumerates the many ways a student can write C programs at McGill in order to do assignments for the numerous classes that require C. This is not an exhaustive list of all possible methods to write C programs but it will get you started as quickly as possible on your own computer. Just choose the method that suites you the best from the list below.

Table of Contents

Using Linux machine at Trottier	2
Using Scite at McConnel.....	2
SSH Into a SOCs machine	2
Linux and Mac: Terminal and SCP.....	2
More Info	2
Windows: Putty and winscp	3
Writing C programs on a Linux Environment.....	3
Compiler	3
Editor.....	4
Running the code.....	5
Writing C Programs on a Mac.....	6
Compile using the Terminal.....	6
Writing C Programs on a windows Machine	6
IDE.....	6
Scite	6
Pelles C	6
VirtualBox	7
http://www.ubuntu.com/download/desktop/install-desktop-long-term-support	7
Installing VirtualBox	7
Creating a Linux Virtual machine	8
Creating a shared folder between windows and Linux.....	8

Using Linux machine at Trottier

You may choose to do your assignments in school using the computer labs on the third floor of Trottier. Go to a machine, choose Ubuntu, and login as you normally would. Then follow the steps under “running the code” in Linux.

Using Scite at McConnel

You can also go to the computer labs in McConnel which have scite pre-installed. Check your applications and you will find scite.

SSH Into a SOCS machine

Using SSH (secure shell) is a good method to test if your code will run properly on your professor's or grader's machine. If it works there it will work on their machine. This is especially important if you are using system calls. You may, if you wish, write your code after you've opened an SSH connection to the SOCS machine.

In order to SSH into a SOCS machine, you must make an account. To do so go to the following website and follow the steps:

<https://newuser.cs.mcgill.ca/>

Once you've made your account there are two ways to access the account.

Linux and Mac: Terminal and SCP

Open up a terminal and check if you have ssh using:

```
ssh -v
```

you should see the version of your ssh program. Otherwise you can get it using:

```
sudo apt-get install ssh
```

Once you have ssh you can log in to your account using:

```
ssh -R 2222:localhost:22 SID@linux.cs.mcgill.ca
```

Where SID stands for your McGill short id. You will then be prompted for your password. Once you've typed your password you will be logged into the SOCS machine.

In order to transfer your files into your SOCS account you need to use SCP (secure copy). Copy your files using the following command:

```
scp -p yourfile.c SID@linux.cs.mcgill.ca
```

Then log back into your SOCS account using ssh to check if your files have transferred.

More Info

You can read more about using SSH to tunnel into McGill here:

<http://socsinfo.cs.mcgill.ca/wiki/SSH>

You can read about how to SSH without typing your password here:

<http://www.howtoforge.com/set-up-ssh-with-public-key-authentication-debian-etch>

Windows: Putty and winscp

On windows, you may get a terminal such as mingw or gitbash and follow the previous instructions, otherwise you will need to get putty to login to your SOCS machine and winscp to transfer your files.

Download putty from

<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>

Then run putty.exe. Under the category tab, type in:

```
SID@linux.cs.mcgill.ca
```

And choose port 22 then click open. After a password prompt, you should be logged into your SOCS machine.

To transfer your files, download winscp from

<http://winscp.net/eng/download.php>

When running winscp choose the following options:

```
File protocol: SFTP
Hostname: linux.cs.mcgill.ca
Port: 22
User Name: yourMcGillShortID
Password: yourPassword
```

You will then see a box with two panels. On the left are your local computer's files and on the right are the files in your SOCS machine. Simply drag and drop the files you want to transfer. You are now set up to use the SOCS machine remotely.

Writing C programs on a Linux Environment

If you've chosen to do your assignments on Linux you have definitely chosen well. Linux gives you the most fine grain control over your options, however it can be easy to get a little lost.

Compiler

To start programming in Linux you need a compiler. Open up terminal and check if you have one, type:

```
gcc -v
```

You should see some text appear that says something like:

```
Gcc version 4.4.3
```

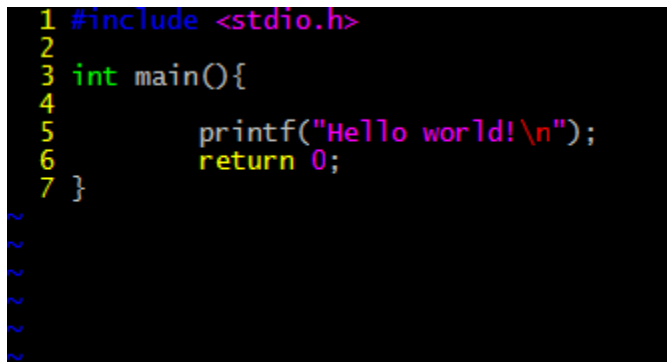
This means you may go to the next steps otherwise perform:

```
sudo apt-get update
```

```
sudo apt-get install build-essential
```

Editor

To write code you need an editor. If you want to use the terminal exclusively consider using vim or nano. Simply type vim or nano to get started.



```
1 #include <stdio.h>
2
3 int main(){
4     printf("Hello world!\n");
5     return 0;
6 }
7
```

Figure 1 A screenshot of programming in VIM

If you want to use graphical editors consider using gedit. Simply type gedit in the terminal and an instance of gedit will open. Otherwise install it using the command:

```
sudo apt-get install gedit
```

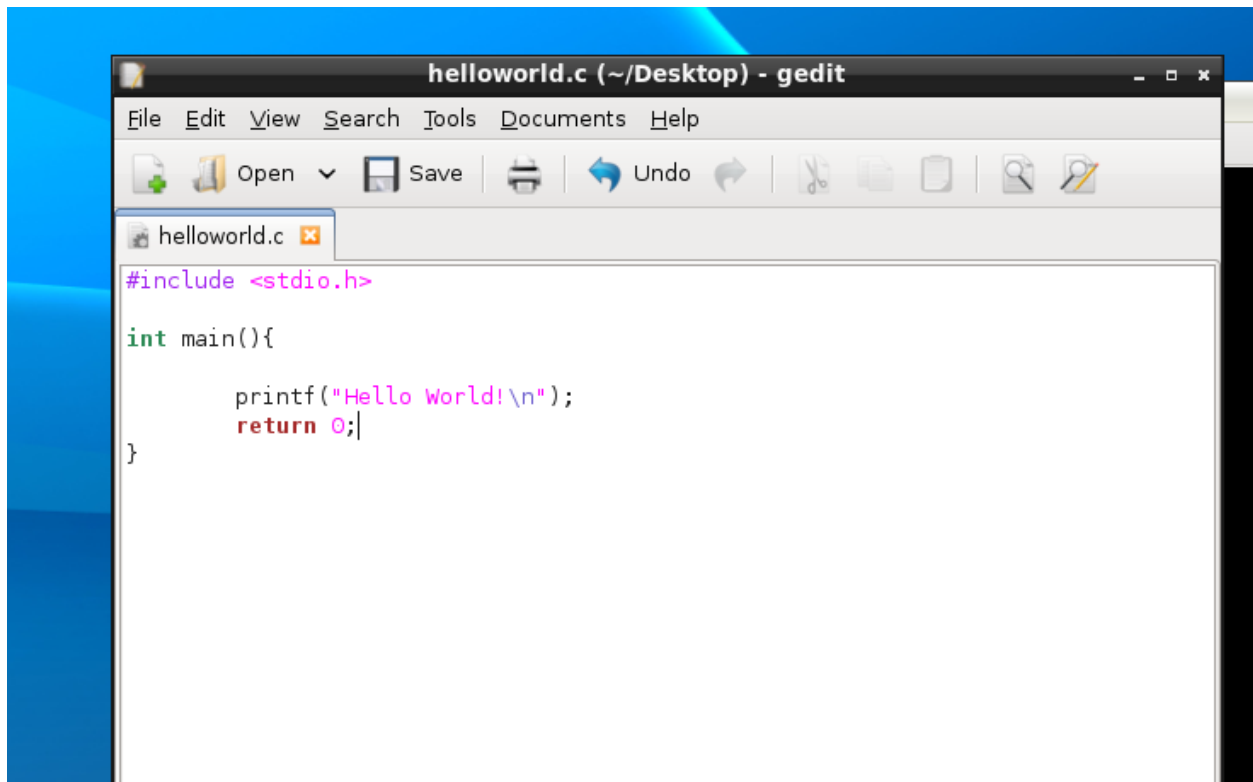


Figure 2 A screenshot of Gedit

Then start typing your code.

Running the code

To run your code open a terminal and change to the directory in which you wrote your c file is placed. You will need to compile your code into a runnable executable file. For example, consider if your file was called helloworld.c then you would use gcc as follows:

```
gcc -o helloworld helloworld.c
```

If your code was able to compile you would then see a file named 'helloworld' appear. To run the file, type the command:

```
./helloworld
```

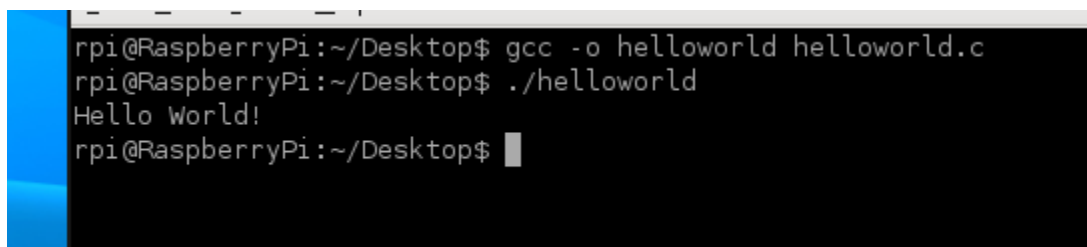


Figure 3 Running a program on the terminal

If everything worked you will then see the output of your program on screen.

Writing C Programs on a Mac

The recommended ide for creating C programs on a Mac is to use Xcode. This software is both an editor and a compiler that will get your programs off the ground very quickly.

For more information about programming using Xcode:

<http://www.cprogramming.com/xcode.html>

Compile using the Terminal

To open the terminal go to finder>applications>utilities> terminal

Go to the directory of your code and compile as if you would using a Linux machine.

```
gcc -o helloworld helloworld.c
```

If your code was able to compile you would then see a file named 'helloworld' appear. To run the file, type the command:

```
./helloworld
```

Writing C Programs on a windows Machine

If you choose to use a windows machine to write your C codes you have two good options. You may either get an Integrated Development Environment (IDE) such as Scite or get Virtualbox and run a flavour of Linux.

IDE

If you are just getting started, it is advised to use an IDE since you will be able to write and run your code quickly.

Scite

You can download Scite from the following website (choose the zip file):

<http://www.scintilla.org/SciTEDownload.html>

For scite to compile, you also need to get mingw which you can find here:


<http://sourceforge.net/projects/mingw/files/Installer/>

Pelles C

Pelles C is a great one stop shop for both editor with compiler and it even has code completion. You can download Pelles C from the following website:

<http://www.christian-heffner.de/index.php?page=download&lang=en>

Ignore all warnings and follow the onscreen instructions to install pelles C.

Once you have it working write up your program and hit the execute button  to run the program.

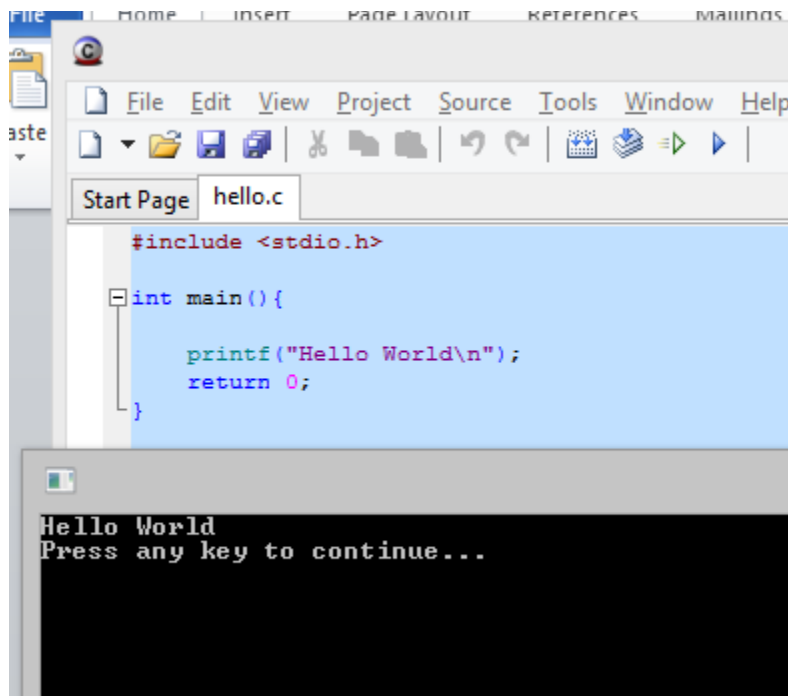


Figure 4 Running a program using Pelles C

VirtualBox

The following is a guide on installing VirtualBox on your windows pc and then running a virtual Instance of Linux. Using Virtualbox is definitely the best option in that gives you the best of two worlds. You can have comfortably write your code using your favourite IDE on your windows machine and using a shared folder you can compile and run the same exact files on your Linux virtual machine. You can even work on your code within the virtual machine and have then synced to your dropbox via the shared folder.

You can download Virtualbox from the following link:

<https://www.virtualbox.org/wiki/Downloads>

You can then download a copy of Linux from the following link

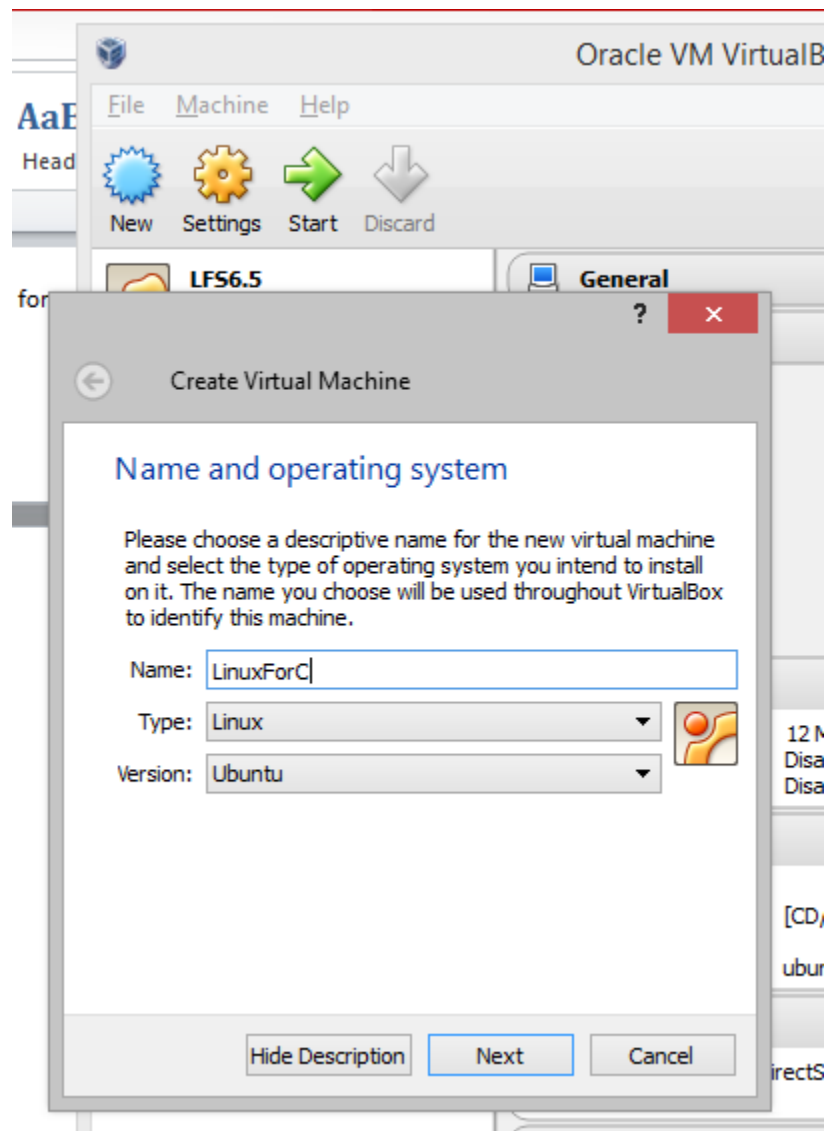
<http://www.ubuntu.com/download/desktop/install-desktop-long-term-support>

Installing VirtualBox

To install VirtualBox, run the executable and follow the instructions on screen. It should take to no more than a few minutes.

Creating a Linux Virtual machine

To create a linux virtual machine, click the 'new' button. Then select a name for your new machine and choose the version of Linux you downloaded.



Then hit next a few times. And you should see an icon appear with the name you've chosen. Choose the virtual machine you've made and hit the start button. You will be guided through the normal installation of Linux.

Creating a shared folder between windows and Linux

Right click on your Linux machine and choose settings. Go under the shared folder tab and click the button to add a new shared folder.

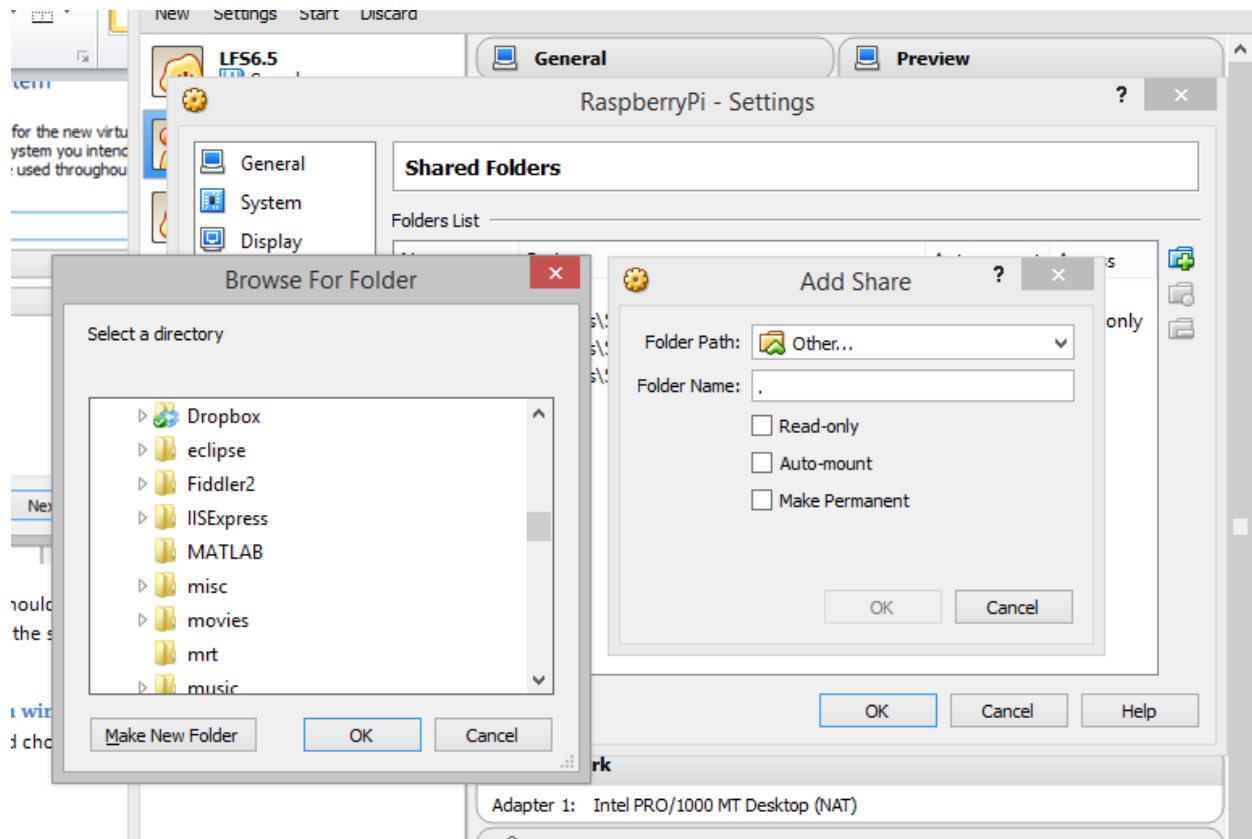


Figure 5 Adding a shared folder

Choose a folder you want to share and check the boxes that say auto-mount and make permanent. Protip: choose a folder that is synced with dropbox, this way anytime a change is made on the virtual machine it will also be synced. Once you've chosen the folder you want to share, remember the name you gave it and go inside your virtual machine and mount the folder. Open up a terminal and create a directory with the same name as the folder

```
mkdir yourfolder
```

Then mount the folder using the command:

```
Sudo mount -t vboxsf yourfoldername ~/yourfolder
```

For more instructions see the link below:

<https://help.ubuntu.com/community/VirtualBox/SharedFolders>