SOFE 3950U / CSCI 3020U: Operating Systems

Lab #1: Linux Setup Version 2.0

Objectives

- To install a Linux-based operating system on your computer
- To install a C compiler and code editor in the Linux environment
- To create a GitHub account and learn how to publish projects to GitHub

Important Notes

- Work in groups of **three** students
  - All reports must be submitted as a PDF on blackboard, if source code is included submit everything as an archive (e.g. zip, tar.gz)
- Save the file as <lab_number>_<first student's id>.pdf (e.g. lab1_100123456.pdf)

If you cannot submit the document on Blackboard then please contact the TA (Neil Seward) via neil.seward@uoit.ca

Deliverables

This lab is merely a setup lab to ensure that everyone is ready to begin work on future assignments. For full marks you must demonstrate that...

1. You have a working Linux environment.
2. You can edit and compile C code.
3. You can publish your C project and commits to GitHub.

Please show the lab TA your setup before leaving. There is no report for this lab.

Note that you do not have to use the exact software recommended in this manual. You may use any Linux distribution you want, installed in any fashion you prefer, with your favorite text editor/IDE and compiler.

Linux Setup

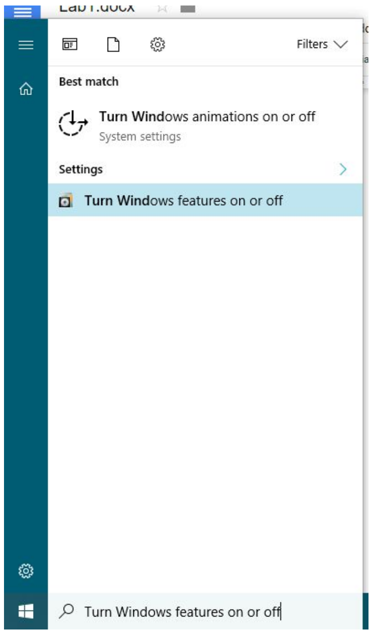
OPTIONAL: If you would prefer to dual boot Linux rather than use it within a virtual machine environment, then please download the following latest version of Xubuntu. It is recommended that you follow the instructions in the guide in the link below before attempting to install Xubuntu as the install process can **erase all of your existing data** if done incorrectly. If you are not comfortable with installing Linux then it is recommended that you **do not try to dual boot Linux**, and rather follow the instructions below using Linux in a virtual machine.

http://mirror.us.leaseweb.net/ubuntu-cdimage/xubuntu/releases/16.04/release/xubuntu-16.04.1-desktop-amd64.iso
http://www.pcsteps.com/961-install-ubuntu-linux-windows/

If you are able to install Linux as a dual boot then please see final step of the virtual machine instructions to install all of the necessary software (clang, lldb, etc.) required for the course.

Enable Virtualization on Windows Instructions

1. Turn off Hyper-V virtualization on windows.


Filters

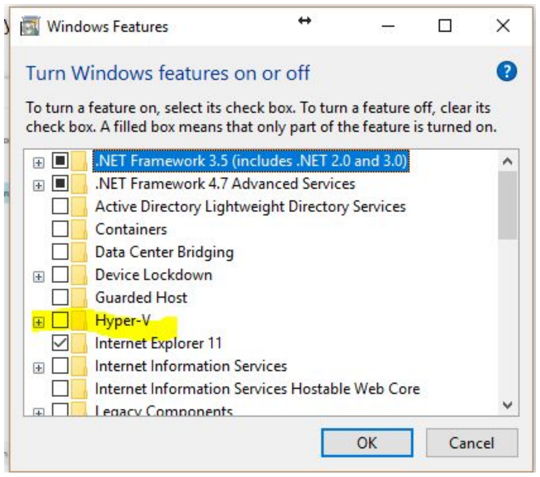
Best match

Turn Windows animations on or off
System settings

Settings

Turn Windows features on or off

Turn Windows features on or off


Windows Features

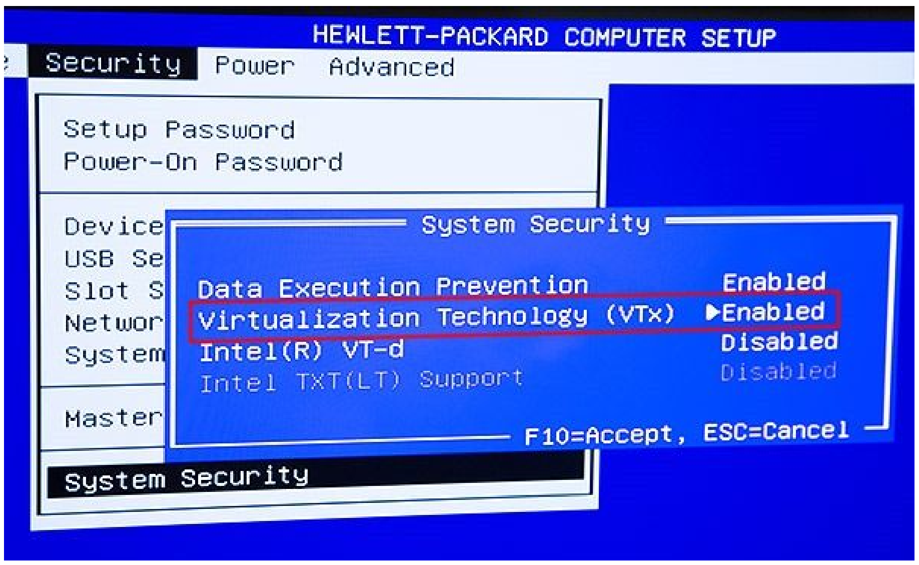
Turn Windows features on or off

To turn a feature on, select its check box. To turn a feature off, clear its check box. A filled box means that only part of the feature is turned on.

- ■ .NET Framework 3.5 (includes .NET 2.0 and 3.0)
- ■ .NET Framework 4.7 Advanced Services
- ☐ Active Directory Lightweight Directory Services
- ☐ Containers
- ☐ Data Center Bridging
- ☐ Device Lockdown
- ☐ Guarded Host
- ☐ Hyper-V
- ☑ Internet Explorer 11
- ☐ Internet Information Services
- ☐ Internet Information Services Hostable Web Core
- ☐ Legacy Components

OK Cancel

2. Enable virtualization on BIOS on startup.


HEWLETT-PACKARD COMPUTER SETUP

Security Power Advanced

Setup Password
Power-On Password

System Security

```
Data Execution Prevention          Enabled
Virtualization Technology (VTx)   ►Enabled
Intel(R) VT-d                      Disabled
Intel TXT(LT) Support              Disabled
```

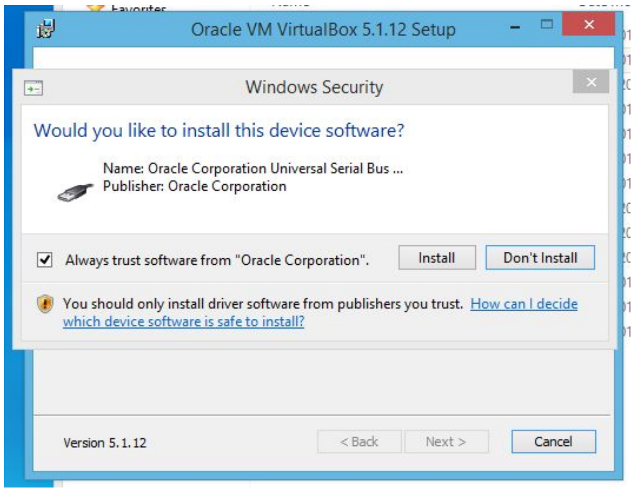
F10=Accept, ESC=Cancel

Virtual Machine Setup Instructions

1. Download VirtualBox for Windows or other host that you are running.

2. Download the 64 bit version of Ubuntu.

3. Install virtualbox just with the default installation options, make sure when prompted that you install the drivers.



Oracle VM VirtualBox 5.1.12 Setup

Windows Security

Would you like to install this device software?

Name: Oracle Corporation Universal Serial Bus ...
Publisher: Oracle Corporation

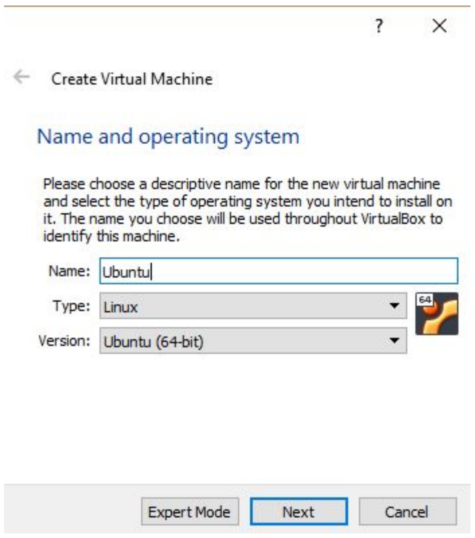
☑ Always trust software from "Oracle Corporation".

You should only install driver software from publishers you trust. How can I decide which device software is safe to install?

Version 5.1.12

4. Extract the virtualbox Xubuntu image using 7zip, use the right click context menu in Windows on the file to bring up 7zip option to extract the file.

5. Follow the instructions below to configure and use the ubuntu image. Begin by pressing the “New” icon at the top menu in Virtualbox to start the process of creating a new machine. Make the new machine named Ubuntu and select for type **Linux** and for version **Ubuntu (64-bit)**.



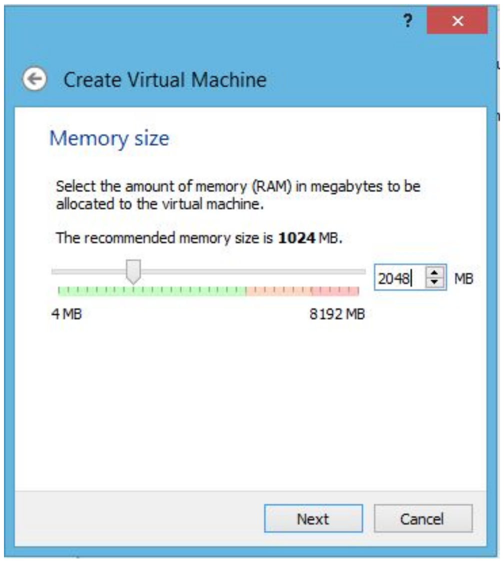
Create Virtual Machine

Name and operating system

Please choose a descriptive name for the new virtual machine and select the type of operating system you intend to install on it. The name you choose will be used throughout VirtualBox to identify this machine.

Name: Ubuntu
Type: Linux
Version: Ubuntu (64-bit)

6. Next, for memory size make sure you set the memory size to **at least 2048 MB** (4096 MB recommended).



Create Virtual Machine

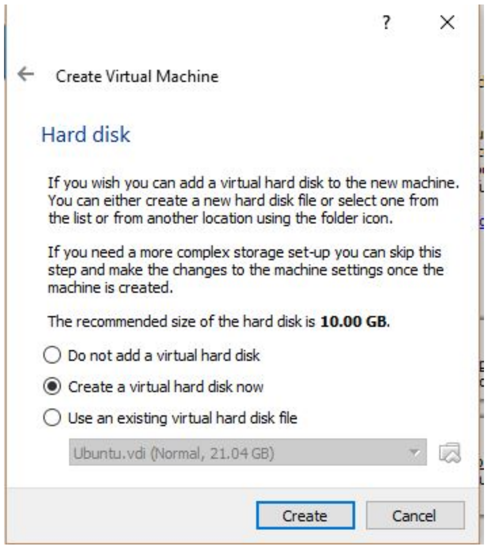
Memory size

Select the amount of memory (RAM) in megabytes to be allocated to the virtual machine.

The recommended memory size is **1024** MB.

2048 MB

7. Next, at the hard disk menu, select the final option “create a virtual disk” and then click the folder icon to load the virtual machine image that you extracted previously.



Hard disk

If you wish you can add a virtual hard disk to the new machine. You can either create a new hard disk file or select one from the list or from another location using the folder icon.

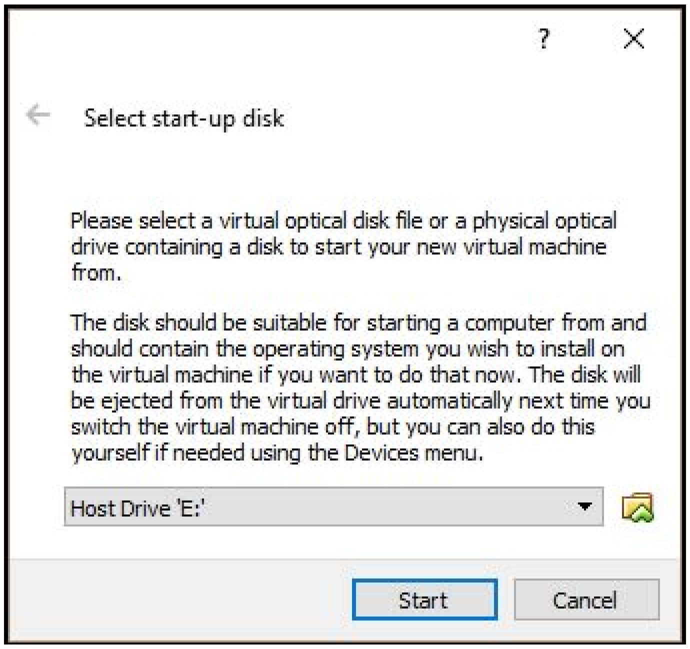
If you need a more complex storage set-up you can skip this step and make the changes to the machine settings once the machine is created.

The recommended size of the hard disk is **10.00 GB**.

○ Do not add a virtual hard disk
● Create a virtual hard disk now
○ Use an existing virtual hard disk file

Ubuntu.vdi (Normal, 21.04 GB)

8. Finally, press start to launch your virtual machine, you should be prompted to point to the Ubuntu image(iso) file. Select the image file and follow the instruction setup.

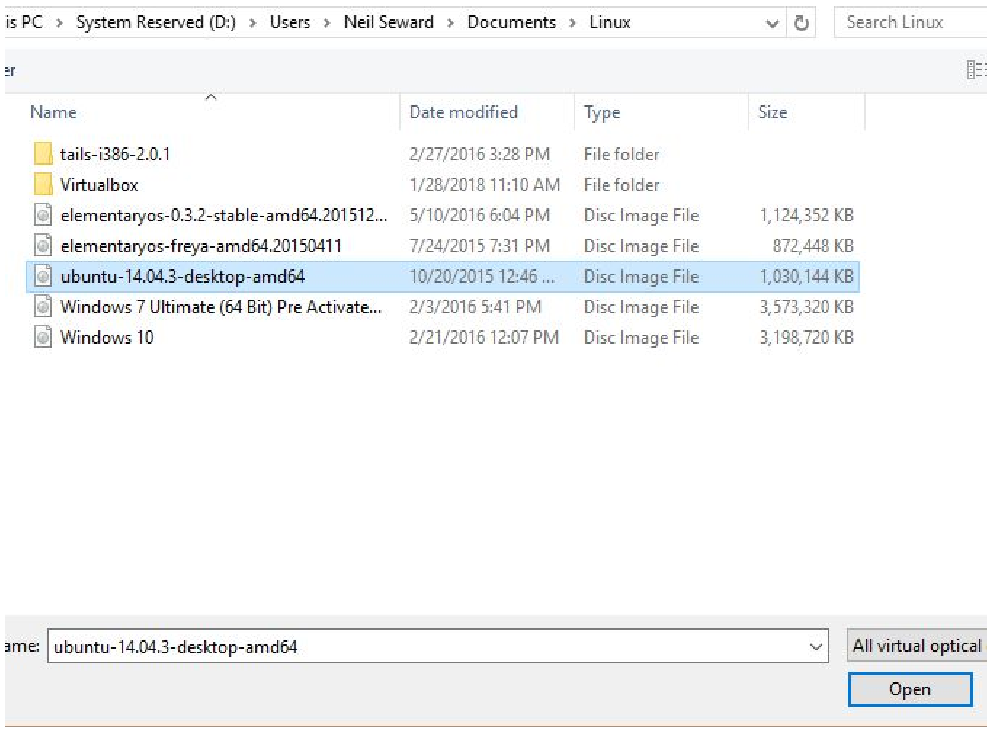


Select start-up disk

Please select a virtual optical disk file or a physical optical drive containing a disk to start your new virtual machine from.

The disk should be suitable for starting a computer from and should contain the operating system you wish to install on the virtual machine if you want to do that now. The disk will be ejected from the virtual drive automatically next time you switch the virtual machine off, but you can also do this yourself if needed using the Devices menu.

Host Drive 'E:'


is PC › System Reserved (D:) › Users › Neil Seward › Documents › Linux

Name	Date modified	Type	Size
tails-i386-2.0.1	2/27/2016 3:28 PM	File folder	
Virtualbox	1/28/2018 11:10 AM	File folder	
elementaryos-0.3.2-stable-amd64.201512...	5/10/2016 6:04 PM	Disc Image File	1,124,352 KB
elementaryos-freya-amd64.20150411	7/24/2015 7:31 PM	Disc Image File	872,448 KB
ubuntu-14.04.3-desktop-amd64	10/20/2015 12:46 ...	Disc Image File	1,030,144 KB
Windows 7 Ultimate (64 Bit) Pre Activate...	2/3/2016 5:41 PM	Disc Image File	3,573,320 KB
Windows 10	2/21/2016 12:07 PM	Disc Image File	3,198,720 KB

ame: ubuntu-14.04.3-desktop-amd64

9. After logging into Ubuntu, next install the guest additions from the menu **"Devices"** then **"Insert Guest Additions CD Image"**. The Virtualbox guest additions make it much easier to work back and forth between the virtual machine and Windows by allowing you to share content that is copy and pasted. In Ubuntu, if you insert Guest Additions, you will be asked to enter in the admin password to run. If you are not prompted for an admin password, follow steps 10-11. Skip 10-11 if Guest Additions ran on insert.

10. You should now see that it has loaded a CD on the virtual machine, navigate to the contents of the CD directory and right click, from the menu select the option **"Open in terminal**" which will open a terminal window directly in the CD folder.

11. To install the Virtualbox guest additions run the following command from the terminal, when prompted for the password enter the same password as for login**.** After the installer has finished it will prompt you to restart your machine for the changes to take effect, after you restart the virtual machine the guest additions should now be installed and working.

```
sudo ./VBoxLinuxAdditions.run
```

12. To enable the clipboard so you can use copy and paste back and forth between the virtual machine and your desktop from the main menu select “Devices” then for “Shared Clipboard” and “Drag and Drop” select the option “Bidirectional”.

13. Open a terminal by running the Terminal Emulator program from the programs menu, then enter the following commands to install all of the software you will need for the rest of the lab (you can copy and paste it all as one command):

```
sudo apt-get update
sudo apt-get install build-essential clang valgrind htop
git
```

GitHub Setup

1. For the labs and assignments in this course it is recommended that you become familiar with version control systems, in particular Git as it will be very beneficial for your collaborative work in this course and other courses.

2. If you haven't already create an account at https://github.com **make sure to use your uoit.net account.** After creating your account you will receive an email from GitHub to verify your email to complete the signup.

3. **Optional:** After you've created your GitHub account and verified your email apply for the student pack, which gives you access to a **plethora of software and services all for free!** (github premium account, domain names, $100 in free hosting on digital ocean, etc.).

https://education.github.com/discount_requests/new

4. Next, go through the following tutorial in order to familiarize yourself with GitHub and git.

https://try.github.io

5. **Optional:** If you would like to learn more about git the following are all excellent resources, git immersion is great if you wish to become more familiar with the commands.

http://git-scm.com/docs/gittutorial
http://gitref.org/
http://gitimmersion.com/
https://help.github.com/articles/set-up-git/#platform-linux

Git Setup for Repository

1. Before you are able to commit anything with Git, you will need to tell Git the name and email address you would like it to sign your commits with. Run the following commands in a terminal, replacing "John Doe" with your name:

   ```
   git config --global user.name "John Doe"
   git config --global user.email "john.doe@uoit.net"
   ```

2. Next you will need to set up a key-pair to be able to push commits to GitHub. Open a terminal and enter the following command:

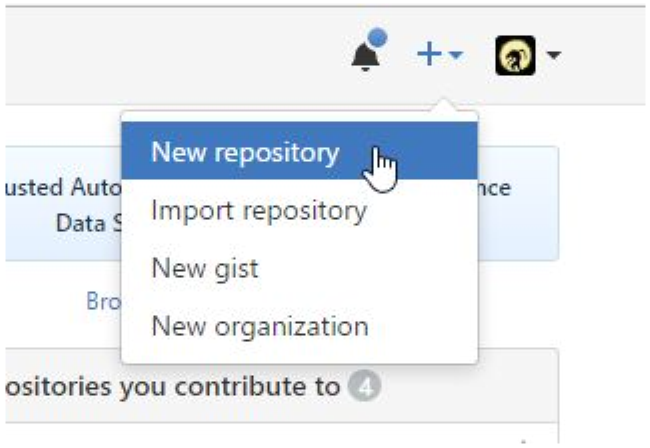
   ```
   ssh-keygen -t rsa -b 4096
   ```

   Press enter for each prompt **without entering anything** (i.e. **do not** enter a path or a password). After this dump your public key to the terminal by entering:

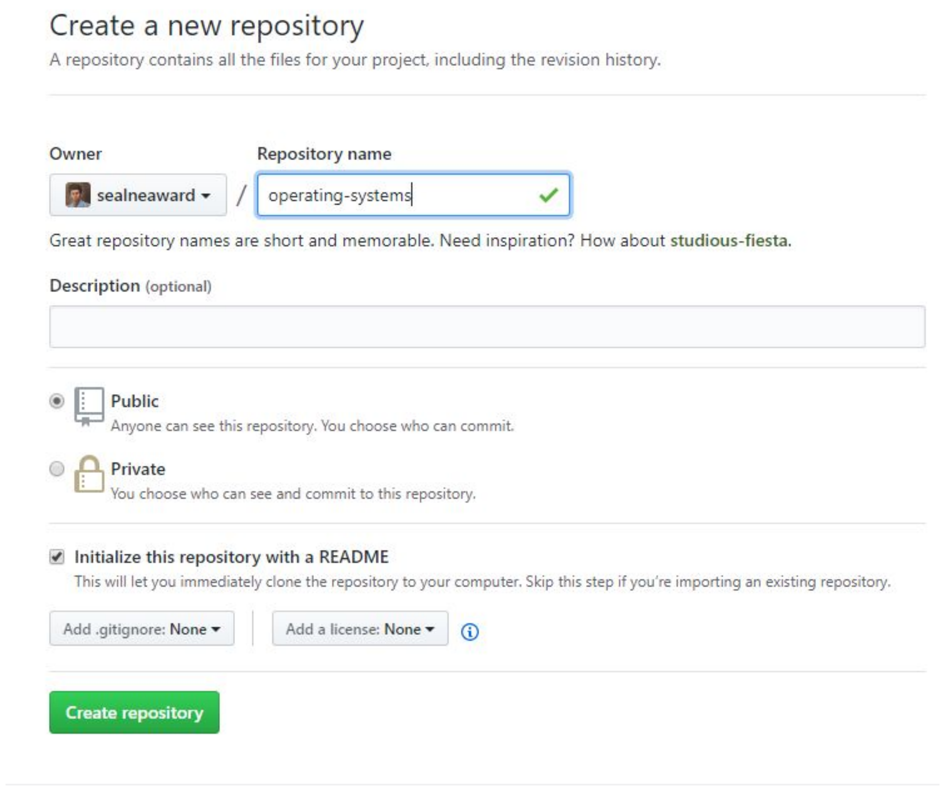
   ```
   cat ~/.ssh/id_rsa.pub
   ```

   Copy the key from the terminal. Go to https://github.com/settings/keys (GitHub login required) and click the green "New SSH key" button in the top-right. Enter a title for the key (e.g. "Xubuntu VM") and paste the key in the "Key" box. Then click the green "Add SSH key" button to add the key to your GitHub account.

3. Next create a new repo by opening https://github.com and clicking the **+** button in the upper-right corner of the page, and then selecting the “New repository” option. Give your repository a name and ensure that the “Initialize this repository with a README” option is checked.



Your repo will be initialized to a placeholder page. **Leave this open** as you will need it a few steps later.



Create a new repository

A repository contains all the files for your project, including the revision history.

Owner: sealneaward / Repository name: operating-systems

Great repository names are short and memorable. Need inspiration? How about studious-fiesta.

Description (optional)

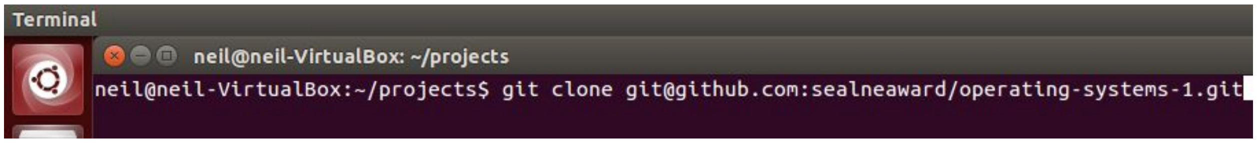
● Public
Anyone can see this repository. You choose who can commit.

○ Private
You choose who can see and commit to this repository.

[✓] Initialize this repository with a README
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: None | Add a license: None

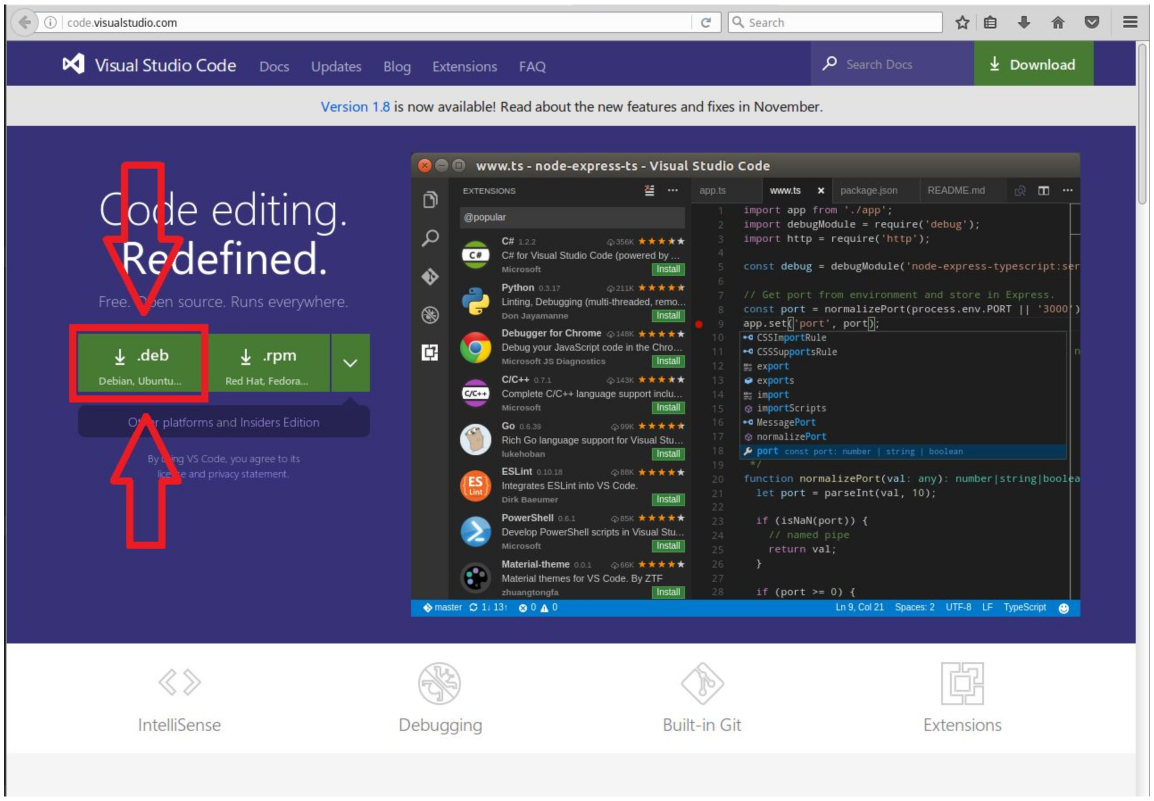
4. Clone the repository on your machine. Remember where this is located.



```
neil@neil-VirtualBox:~/projects$ git clone git@github.com:sealneaward/operating-systems-1.git
```

IDE Installation (Visual Studio Code)

1. Download the .deb version of Visual Studio Code by going to https://code.visualstudio.com/ in the browser of your Linux virtual machine and clicking the .deb download button.



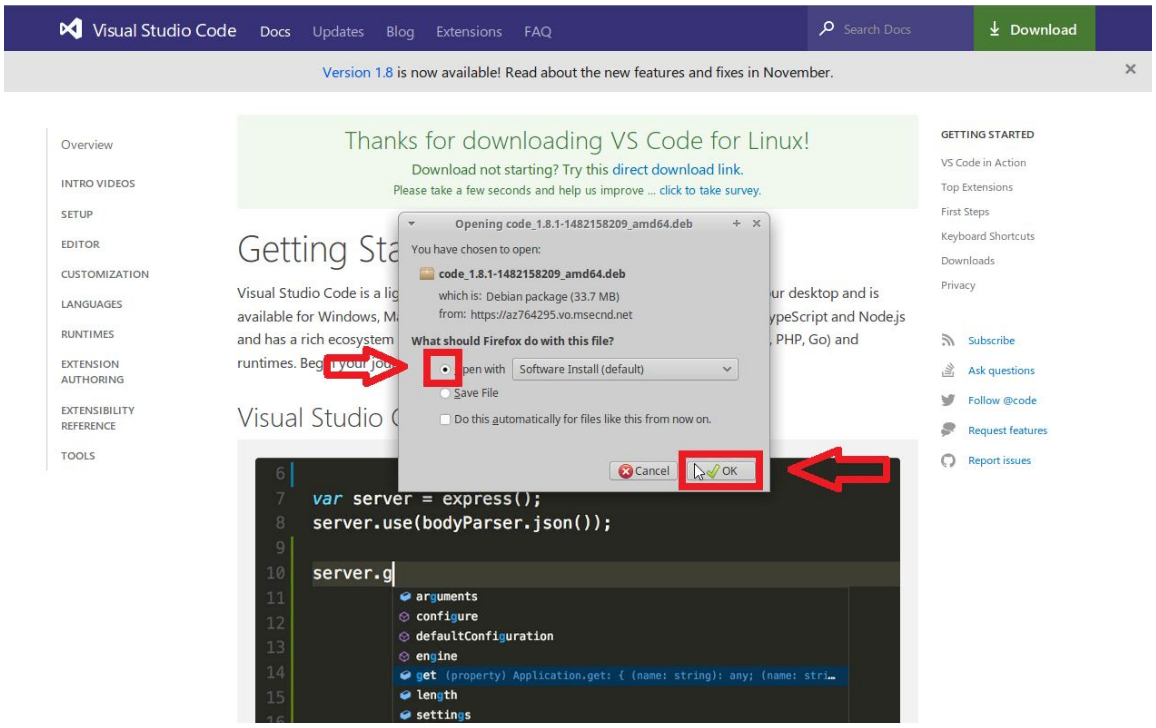
Version 1.8 is now available! Read about the new features and fixes in November.

By using VS Code, you agree to its license and privacy statement.

```
import app from './app';
import debugModule = require('debug');
import http = require('http');

const debug = debugModule('node-express-typescript:ser

// Get port from environment and store in Express.
const port = normalizePort(process.env.PORT || '3000')
app.set('port', port);
```

```
function normalizePort(val: any): number|string|boolea
  let port = parseInt(val, 10);

  if (isNaN(port)) {
    // named pipe
    return val;
  }

  if (port >= 0) {
```

2. Choose to open the .deb with Software Install and click OK to start the download.



Version 1.8 is now available! Read about the new features and fixes in November.

Thanks for downloading VS Code for Linux!

Download not starting? Try this direct download link.

Please take a few seconds and help us improve ... click to take survey.

Opening code_1.8.1-1482158209_amd64.deb

You have chosen to open:

code_1.8.1-1482158209_amd64.deb

which is: Debian package (33.7 MB)

from: https://az764295.vo.msecnd.net

What should Firefox do with this file?

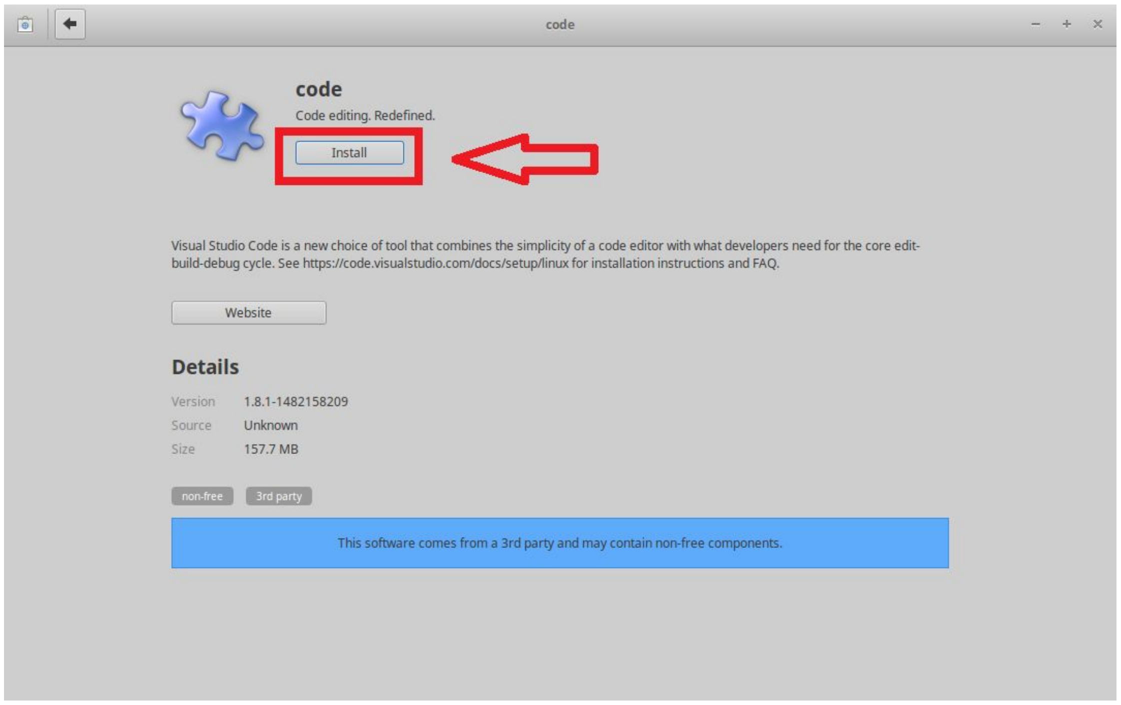
[✓ Open with] Software Install (default)

Save File

Do this automatically for files like this from now on.

```
var server = express();
server.use(bodyParser.json());

server.g
        arguments
        configure
        defaultConfiguration
        engine
        get (property) Application.get: { (name: string): any; (name: stri…
        length
        settings
```

3. Click the Install button once the download is complete and the Software Install program opens.



code

Code editing. Redefined.

Visual Studio Code is a new choice of tool that combines the simplicity of a code editor with what developers need for the core edit-build-debug cycle. See https://code.visualstudio.com/docs/setup/linux for installation instructions and FAQ.

Details

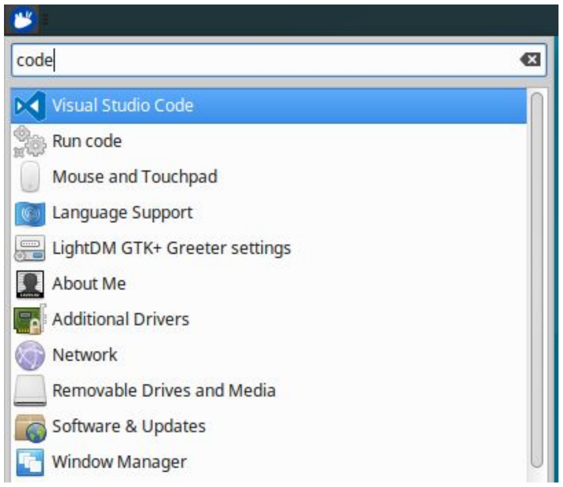
Version	1.8.1-1482158209
Source	Unknown
Size	157.7 MB

This software comes from a 3rd party and may contain non-free components.

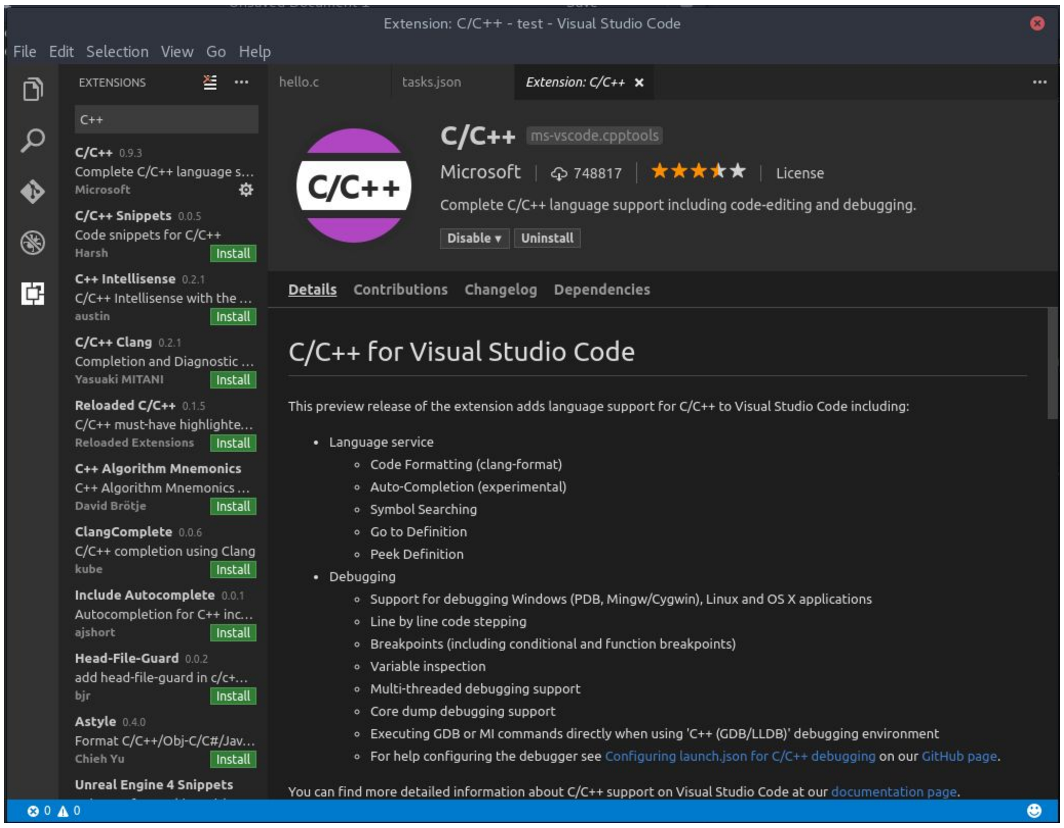
4. **If the previous step failed:** Download the Visual Studio Code .deb file again but instead of opening it save it to your desktop. Right-click your desktop and click the "Open Terminal" menu option. In your terminal enter:

```
sudo dpkg -i code_*.deb
sudo apt-get install -f
```

5. **Visual Studio Code is now installed.** It can be launched from the program launcher via the search tool or within the Development category.



6. Lastly, install support for C/C++ in Visual Studio Code by selecting the final left navigation option (CTRL+SHIFT+X) to install extensions. In the search enter **C++** and install the C/C++ language extension published by Microsoft.


Extension: C/C++ - test - Visual Studio Code
File Edit Selection View Go Help
EXTENSIONS
hello.c
tasks.json
Extension: C/C++
C++

C/C++ 0.9.3
Complete C/C++ language s…
Microsoft

C/C++ Snippets 0.0.5
Code snippets for C/C++
Harsh Install

C++ Intellisense 0.2.1
C/C++ Intellisense with the …
austin Install

C/C++ Clang 0.2.1
Completion and Diagnostic …
Yasuaki MITANI Install

Reloaded C/C++ 0.1.5
C/C++ must-have highlighte…
Reloaded Extensions Install

C++ Algorithm Mnemonics
C++ Algorithm Mnemonics …
David Brötje Install

ClangComplete 0.0.6
C/C++ completion using Clang
kube Install

Include Autocomplete 0.0.1
Autocompletion for C++ inc…
ajshort Install

Head-File-Guard 0.0.2
add head-file-guard in c/c+…
bjr Install

Astyle 0.4.0
Format C/C++/Obj-C/C#/Jav…
Chieh Yu Install

Unreal Engine 4 Snippets

C/C++ ms-vscode.cpptools
Microsoft | 748817 | ★★★★☆ | License
Complete C/C++ language support including code-editing and debugging.
Disable ▾ Uninstall

Details Contributions Changelog Dependencies

C/C++ for Visual Studio Code

This preview release of the extension adds language support for C/C++ to Visual Studio Code including:

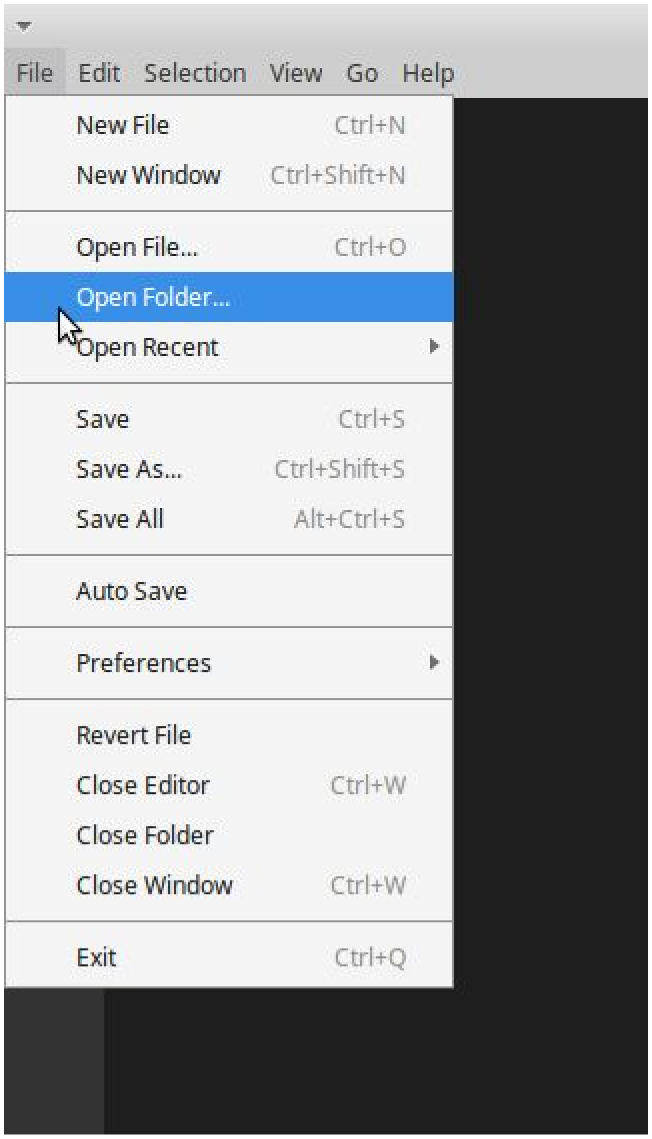
- Language service
  - Code Formatting (clang-format)
  - Auto-Completion (experimental)
  - Symbol Searching
  - Go to Definition
  - Peek Definition
- Debugging
  - Support for debugging Windows (PDB, Mingw/Cygwin), Linux and OS X applications
  - Line by line code stepping
  - Breakpoints (including conditional and function breakpoints)
  - Variable inspection
  - Multi-threaded debugging support
  - Core dump debugging support
  - Executing GDB or MI commands directly when using 'C++ (GDB/LLDB)' debugging environment
  - For help configuring the debugger see Configuring launch.json for C/C++ debugging on our GitHub page.

You can find more detailed information about C/C++ support on Visual Studio Code at our documentation page.

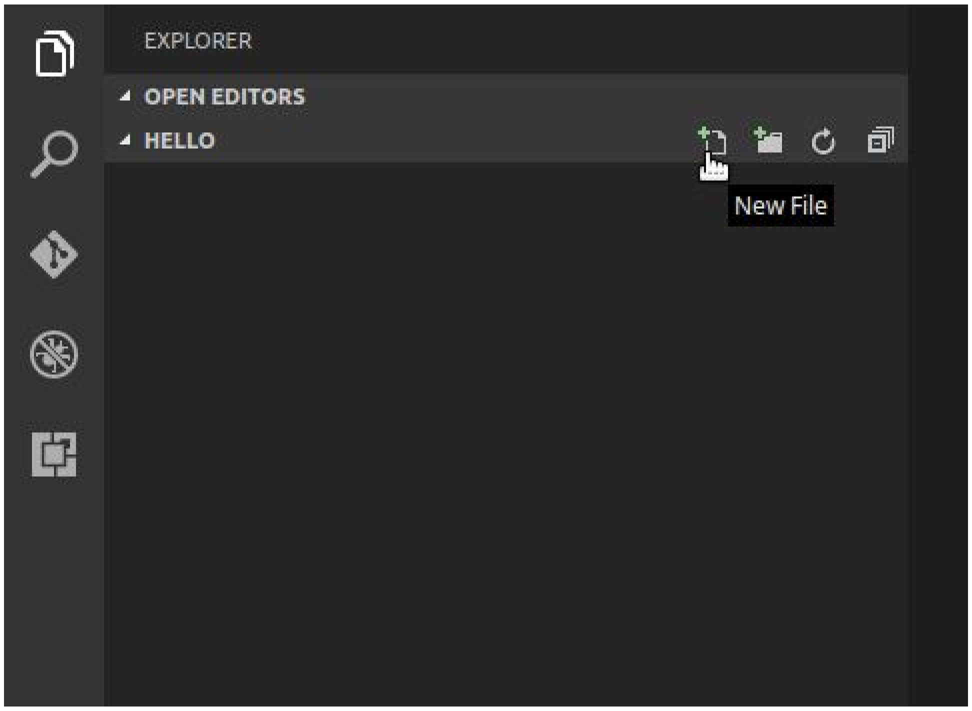
C Project Setup

To setup a new C language project with Visual Studio Code to allow for building and debugging within the IDE please perform the following steps:

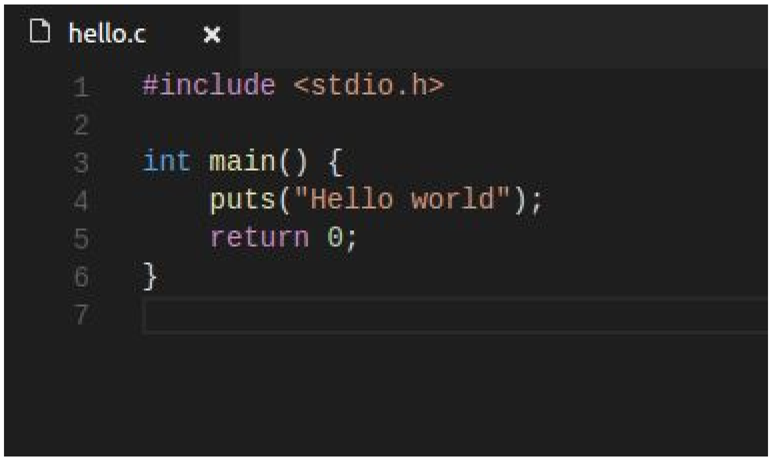
1. Open the folder that will contain your project’s files in VS Code by choosing the “Open Folder…” option from the “File” menu. (If you haven’t created a directory for your project yet you can create one with the “Create Folder” button in the “Open Folder…” dialog box.)



2. Click the “New File” button to create a new C file. Name this file “hello.c” for this lab session.



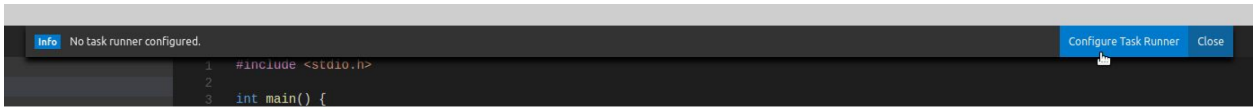
3. Enter the following code into the editor window for hello.c:



```c
#include <stdio.h>

int main() {
    puts("Hello world");
    return 0;
}

```

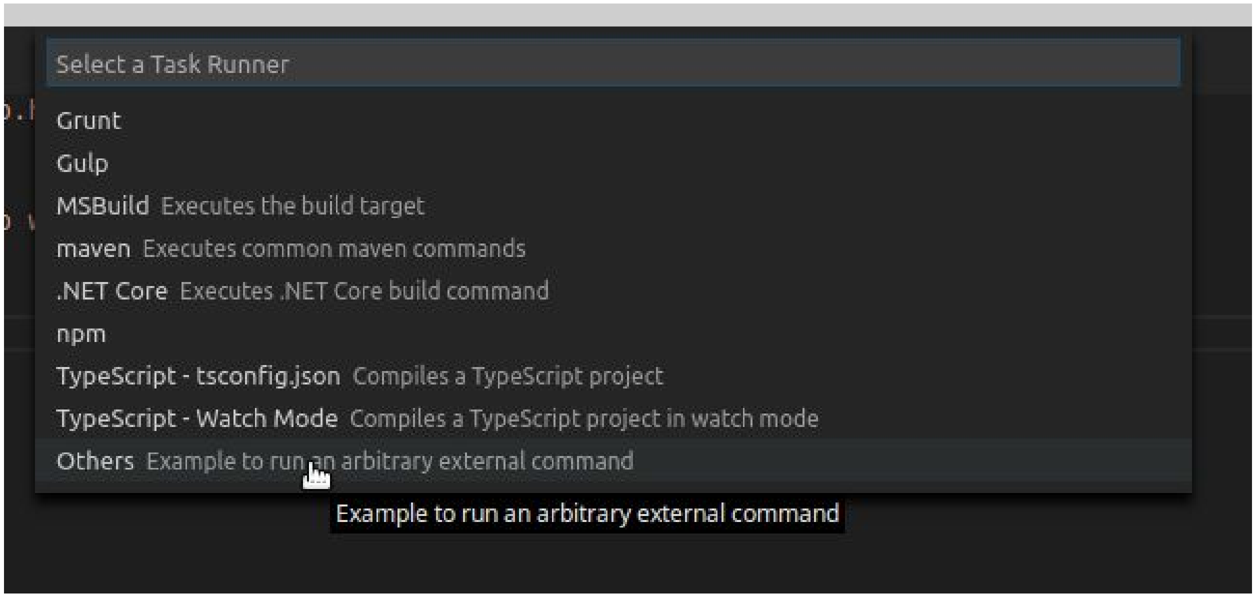
make: clang: Command not found

4. Press the "Run Build Task" key combo (Ctrl-Shift-B) to build the code. Since no build command is configured a pop-up will appear. Press the "Configure Task Runner" button and choose "Others" from the menu that appears.

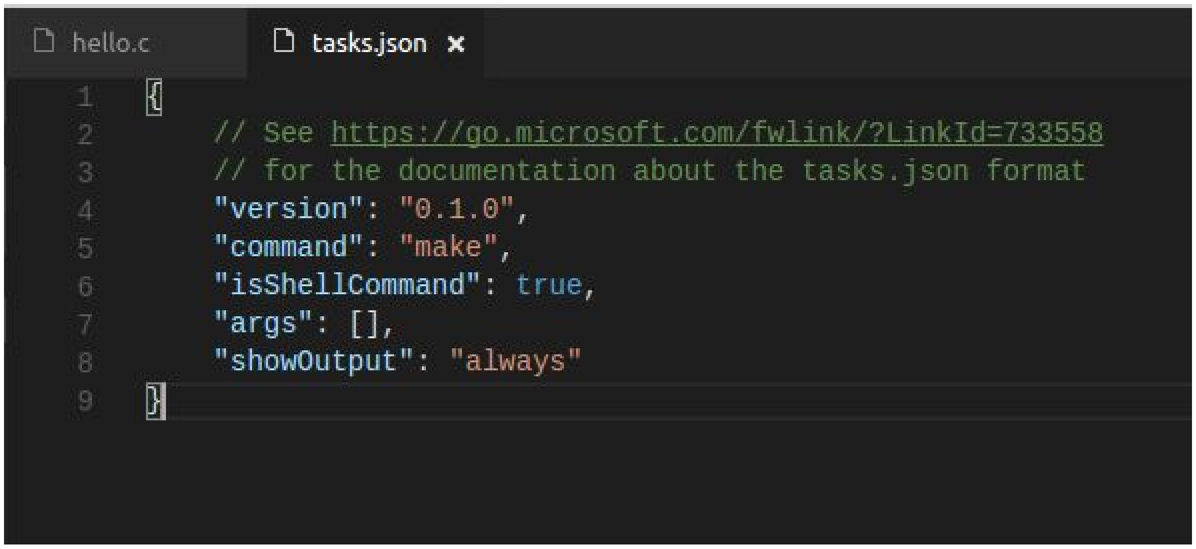


Info No task runner configured.

```c
#include <stdio.h>

int main() {
```

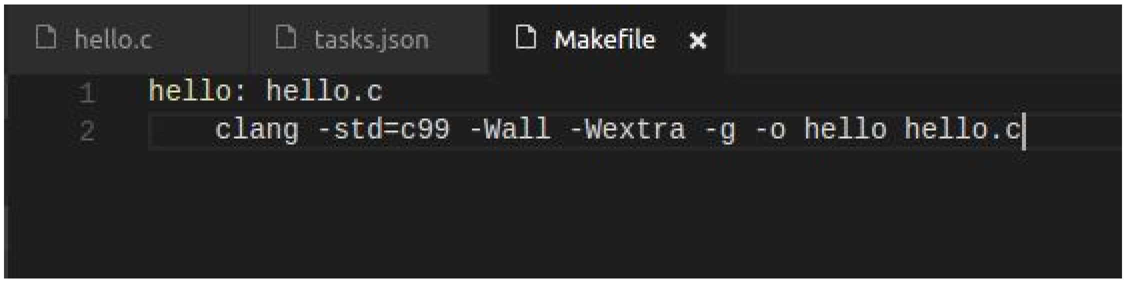

Select a Task Runner
Grunt
Gulp
MSBuild Executes the build target
maven Executes common maven commands
.NET Core Executes .NET Core build command
npm
TypeScript - tsconfig.json Compiles a TypeScript project
TypeScript - Watch Mode Compiles a TypeScript project in watch mode
Others Example to run an arbitrary external command
Example to run an arbitrary external command

5. Change the command to “make” and remove any arguments from the “args” array, then save tasks.json.



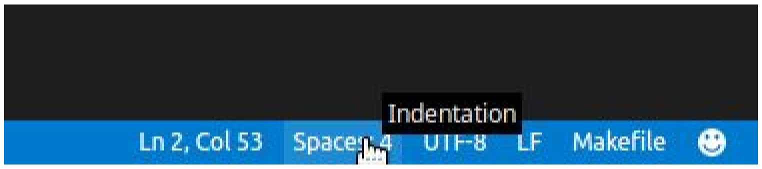
```
{
    // See https://go.microsoft.com/fwlink/?LinkId=733558
    // for the documentation about the tasks.json format
    "version": "0.1.0",
    "command": "make",
    "isShellCommand": true,
    "args": [],
    "showOutput": "always"
}
```

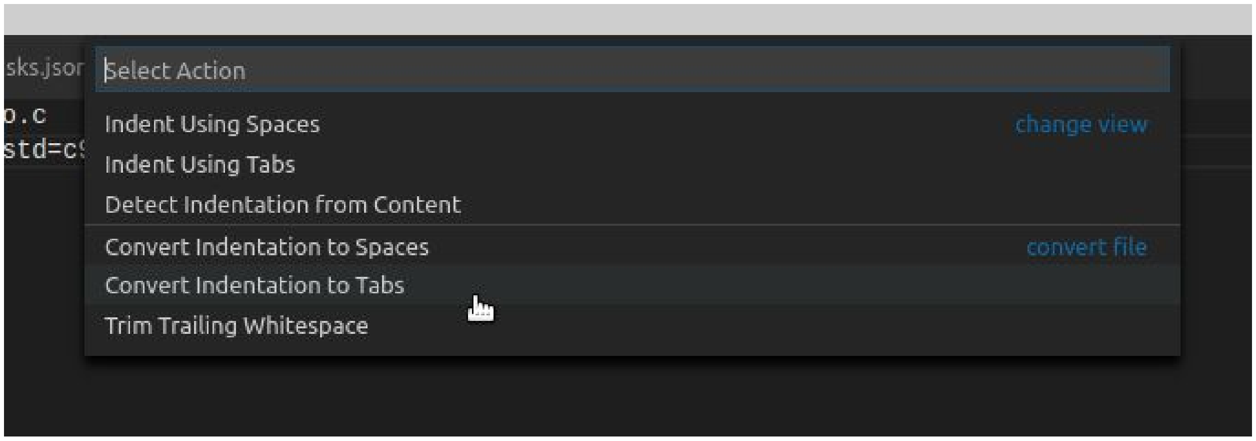
6. Create a new file called “Makefile” in the root of the project folder and enter the following into the file:



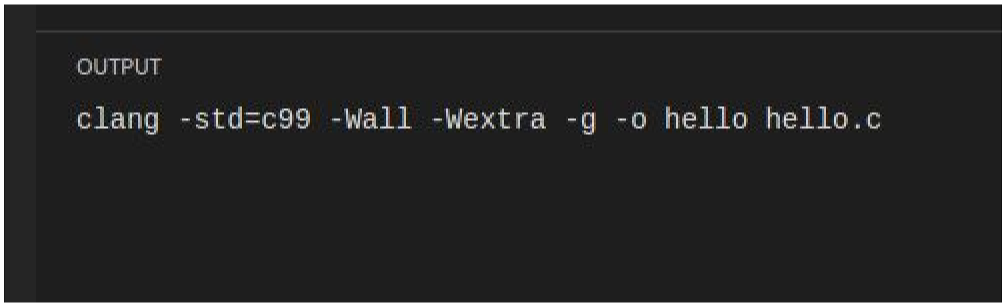
```
hello: hello.c
    clang -std=c99 -Wall -Wextra -g -o hello hello.c
```

7. **Make is very picky about indentation!** The second line should be indented by exactly one tab character. Make sure that there are only tabs in your Makefile by clicking the indentation settings button in the bottom status bar and choosing the "Convert Indentation to Tabs" option from the drop-down menu that appears. *Remember to save the changes to Makefile after you do this.*

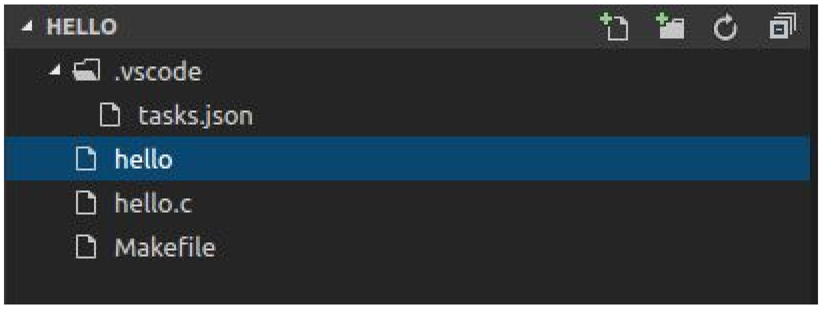




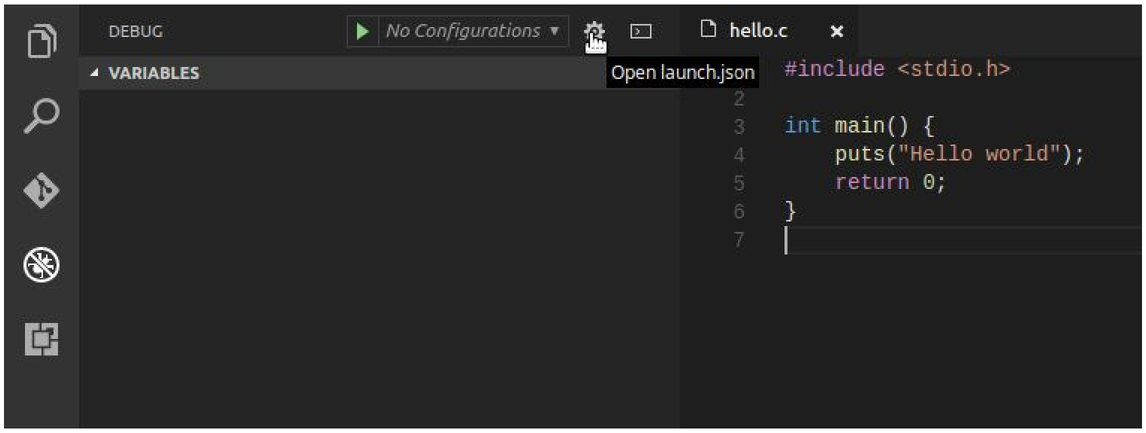
8. Press the "Run Build Task" key command (Ctrl-Shift-B) again to build the program. The "Output" window should show the command that was executed by Make and a file named "hello" should appear in your project.


OUTPUT

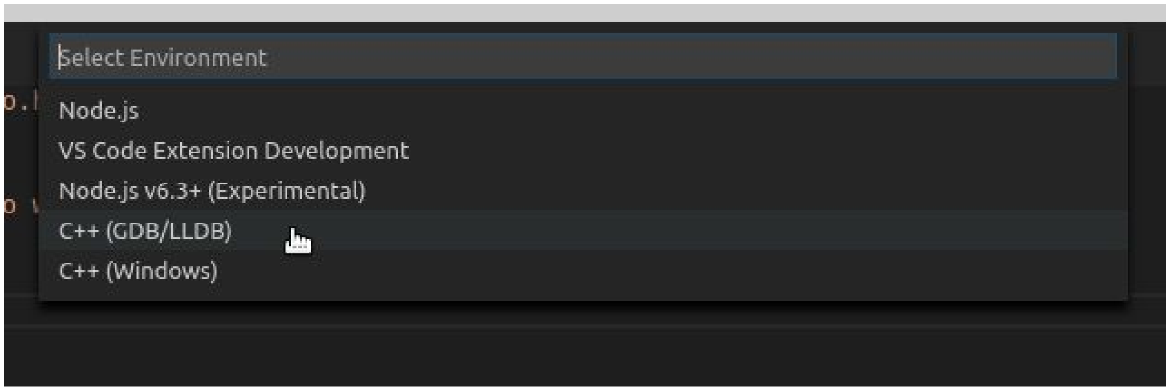
```
clang -std=c99 -Wall -Wextra -g -o hello hello.c
```



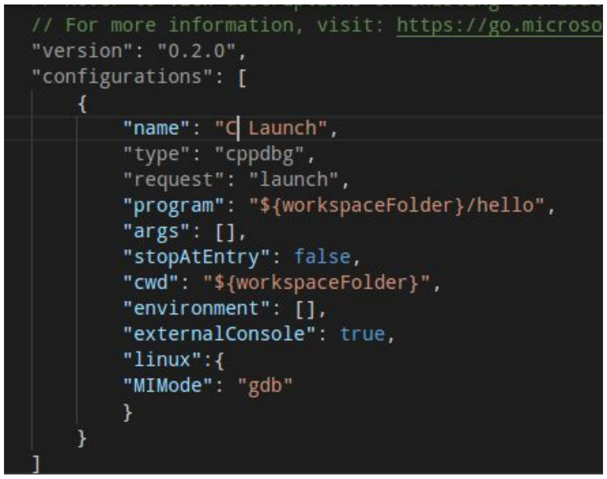
9. Open the “Debug” panel by clicking the fourth bugdbtton in the toolbar on the left side of VS Code. Press the “Open launch.json” button and choose the “C++ (GDB/LLDB)” option from the drop-down menu to generate a launch.json file.



```
#include <stdio.h>

int main() {
    puts("Hello world");
    return 0;
}
```

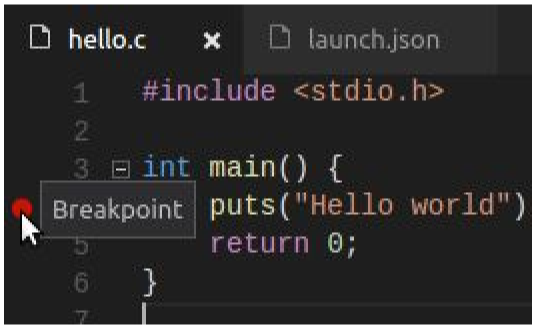


10. Modify the "`MIMode`" option in the “linux” section of the "`C++ Launch`" configurations to be "`gdb`" and remove the "`setupCommands`" part. Change the "`program`" option of each configuration to "`${workspaceRoot}/hello`".



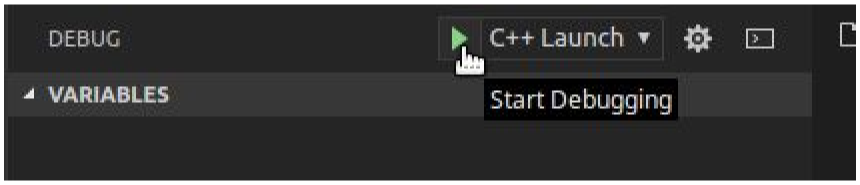
```
// For more information, visit: https://go.microso
"version": "0.2.0",
"configurations": [
    {
        "name": "C Launch",
        "type": "cppdbg",
        "request": "launch",
        "program": "${workspaceFolder}/hello",
        "args": [],
        "stopAtEntry": false,
        "cwd": "${workspaceFolder}",
        "environment": [],
        "externalConsole": true,
        "linux":{
        "MIMode": "gdb"
        }
    }
]
```

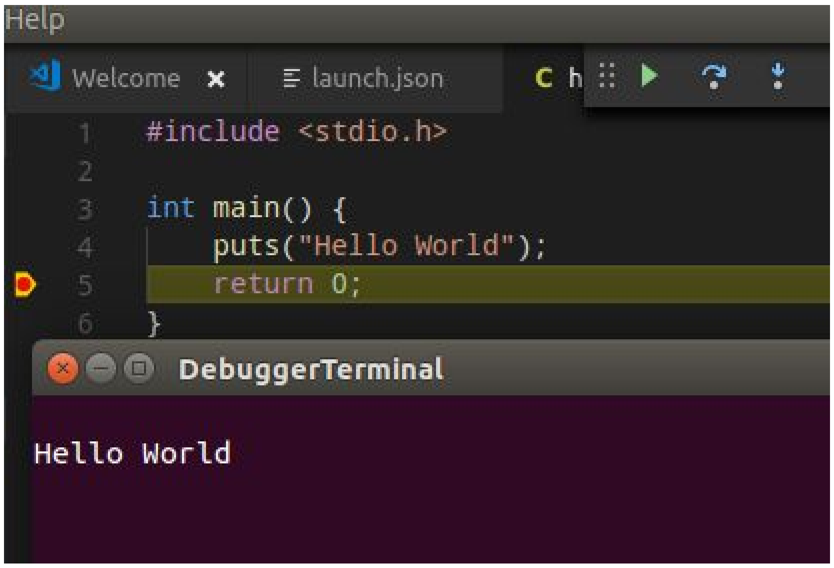
11. Set a breakpoint in the first line of the main function in hello.c by clicking the empty space in the margin to the left of the line number. A red circle will appear.



```
#include <stdio.h>

int main() {
    puts("Hello world")
    return 0;
}
```

12. Run the program in gdb by clicking the green play button in the “Debug” panel. The program will stop at the breakpoint. You can now use the controls at the top of the editor to step through the program line by line.



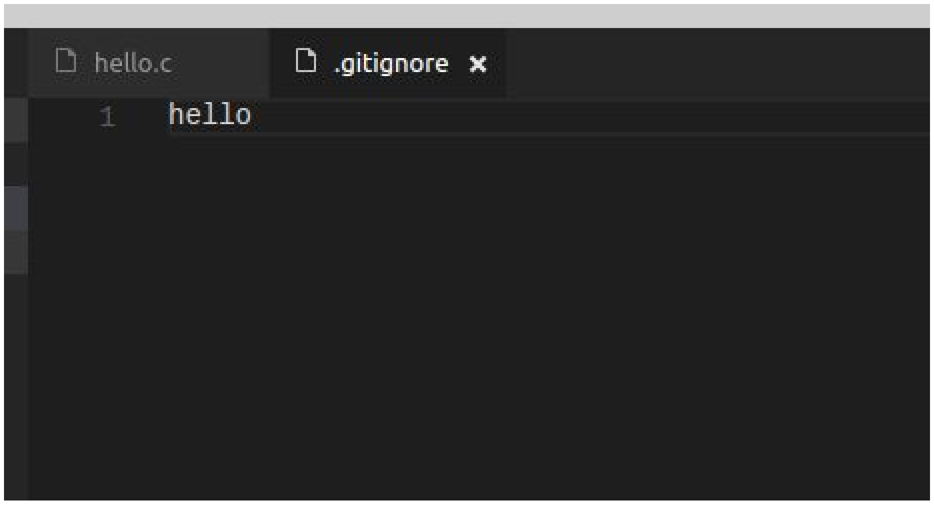


```
#include <stdio.h>

int main() {
    puts("Hello World");
    return 0;
}
```

```
Hello World
```

Git Publishing Your Project

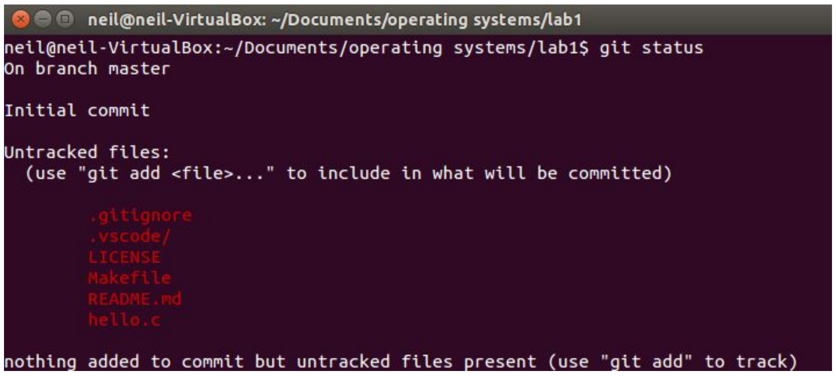
5. Add a file named .gitignore to the project. This file tells Git which files of your project it should not bother tracking for changes or including in commits (e.g. files generated by the compiler). Put the following in your .gitignore:

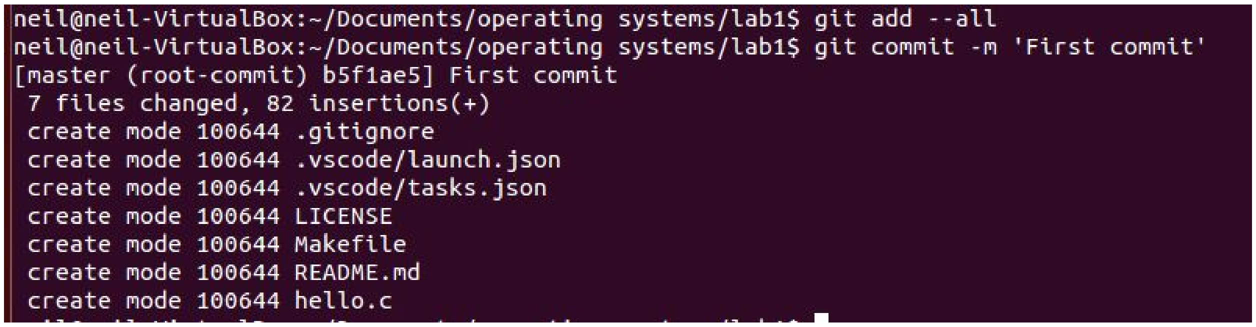
   ```
   hello
   ```

   This will prevent Git from tracking the hello program generated by the C compiler.

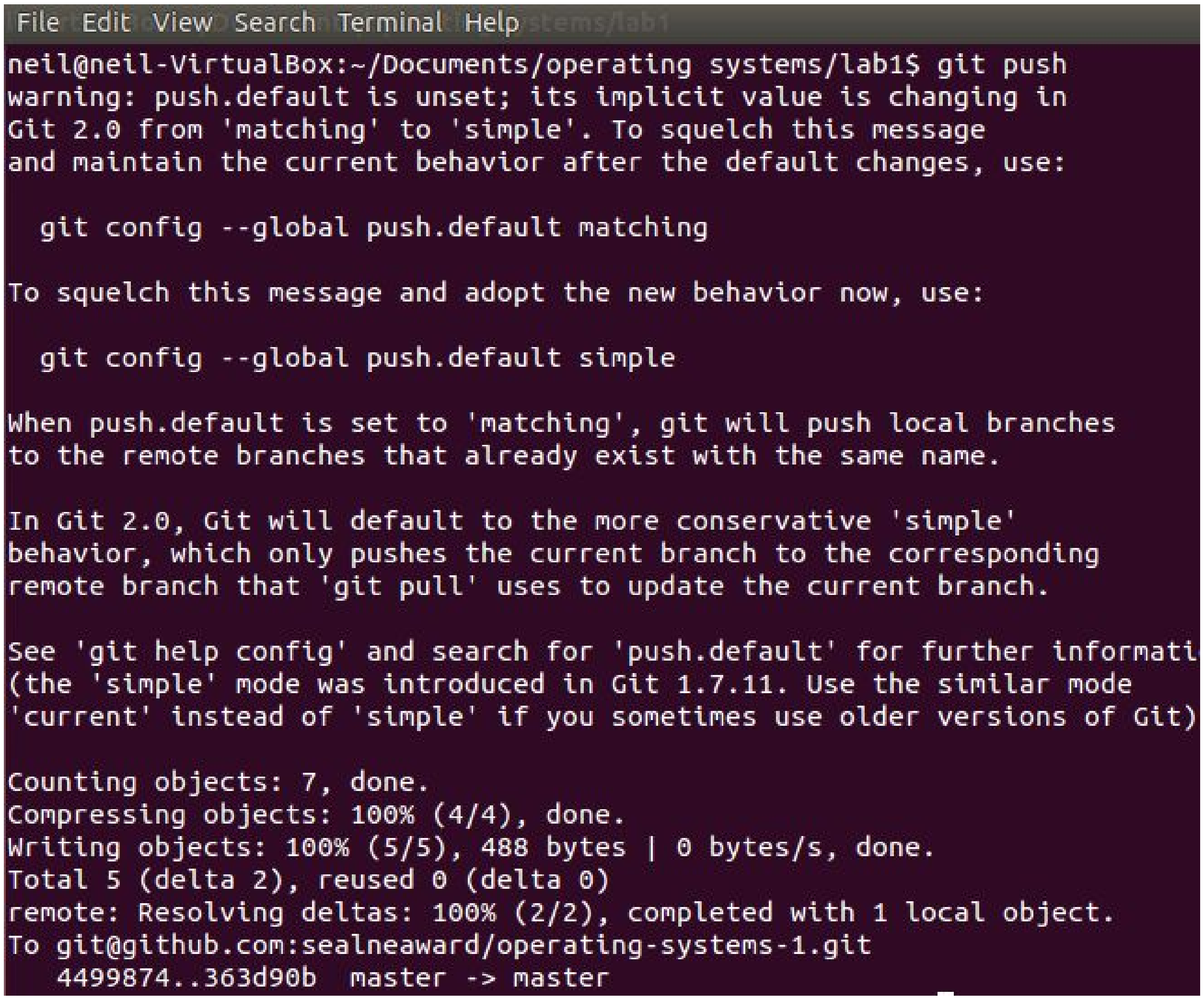
6. You can publish it to your GitHub repo by running the commands specified in the repo's placeholder page inside the terminal you set aside in the previous step.



```
neil@neil-VirtualBox:~/Documents/operating systems/lab1$ git status
On branch master

Initial commit

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        .gitignore
        .vscode/
        LICENSE
        Makefile
        README.md
        hello.c

nothing added to commit but untracked files present (use "git add" to track)
```



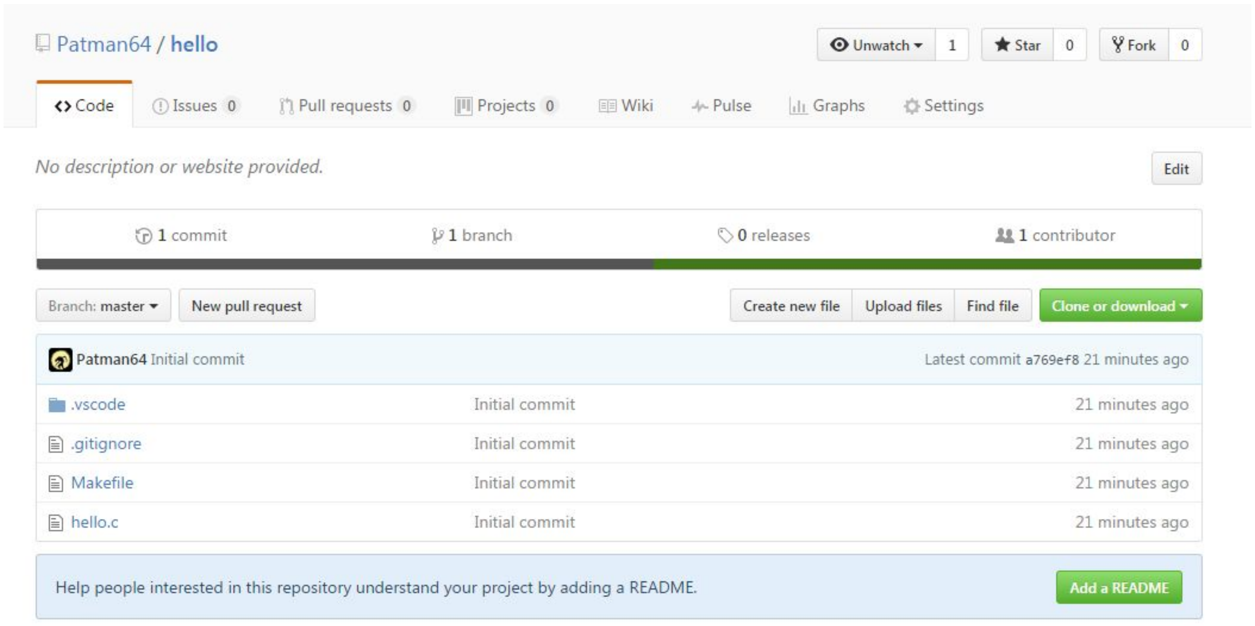
```
neil@neil-VirtualBox:~/Documents/operating systems/lab1$ git add --all
neil@neil-VirtualBox:~/Documents/operating systems/lab1$ git commit -m 'First commit'
[master (root-commit) b5f1ae5] First commit
 7 files changed, 82 insertions(+)
 create mode 100644 .gitignore
 create mode 100644 .vscode/launch.json
 create mode 100644 .vscode/tasks.json
 create mode 100644 LICENSE
 create mode 100644 Makefile
 create mode 100644 README.md
 create mode 100644 hello.c
```



```
neil@neil-VirtualBox:~/Documents/operating systems/lab1$ git push
warning: push.default is unset; its implicit value is changing in
Git 2.0 from 'matching' to 'simple'. To squelch this message
and maintain the current behavior after the default changes, use:

  git config --global push.default matching

To squelch this message and adopt the new behavior now, use:

  git config --global push.default simple

When push.default is set to 'matching', git will push local branches
to the remote branches that already exist with the same name.

In Git 2.0, Git will default to the more conservative 'simple'
behavior, which only pushes the current branch to the corresponding
remote branch that 'git pull' uses to update the current branch.

See 'git help config' and search for 'push.default' for further informati
(the 'simple' mode was introduced in Git 1.7.11. Use the similar mode
'current' instead of 'simple' if you sometimes use older versions of Git)

Counting objects: 7, done.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 488 bytes | 0 bytes/s, done.
Total 5 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To git@github.com:sealneaward/operating-systems-1.git
   4499874..363d90b  master -> master
```

These commands will be slightly different for your repo! Do not type in what you see in the image above. **NOTE:** Make sure that the "SSH" button is clicked in the "Quick setup" area so that the URL is of the form "git@github.com:<your GitHub name>/<project name>.git".

7. After typing in the commands you can refresh your GitHub repo's page to see the contents of the repo.



No description or website provided.

1 commit · 1 branch · 0 releases · 1 contributor

Patman64 Initial commit — Latest commit a769ef8 21 minutes ago

.vscode	Initial commit	21 minutes ago
.gitignore	Initial commit	21 minutes ago
Makefile	Initial commit	21 minutes ago
hello.c	Initial commit	21 minutes ago

Help people interested in this repository understand your project by adding a README.

You have now published your project on GitHub.

If you are reading this and have done all of the above, you have completed the lab. Please show the T.A. your progress before you leave.