

## CS1050 – Lab 1

Fall 2021

### Concepts to Practice

- Connecting to the Teaching Cluster (the server; aka tc.rnet.missouri.edu)
- Editing a file using vim
- Compiling a program
- File Transfer

### Submission Information (you will do this at the end of your lab)

Submit this assignment by following the instructions given by your TA. SUBMIT ONLY the .c file (no a.out or executable file is required). All of the lab assignments must be submitted before the end of the lab using the lab code given by the TA.

Use the following submit command:

```
mucs submit <class> <assignment_type> <filename>
```

For example:

```
mucs submit 1050 lab x-lab1.c
```

Filenames must be: *sectionletter-lab1.c* (include your respective lab section, e.g., a-lab1.c).

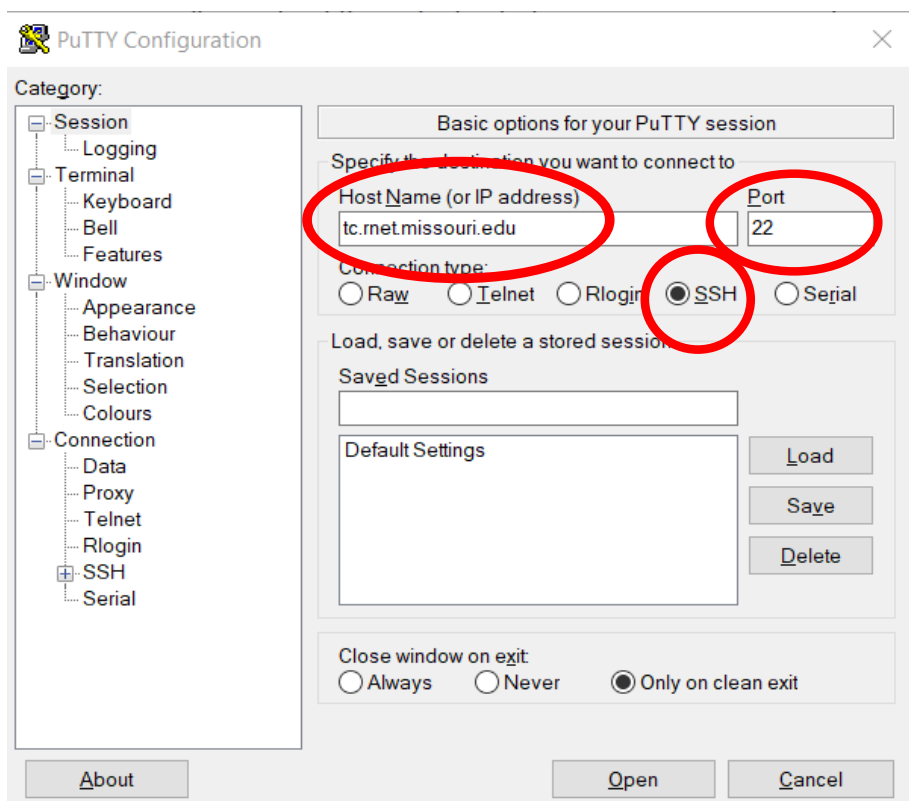
### Description

For this course we will be coding in C and will be programming in a Unix environment (if you do not know what that means, don't worry). To connect to the server that our code will be hosted on, we will use PuTTY – this software is available on Mizzou lab computers but you will want to download it to your device if you are using a Windows computer for labs. It is a free piece of software. If you have a Mac, you will not need PuTTY and can use the built-in Terminal program. Here is the link to download PuTTY if you need it: <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>

## Connecting to the Server

### Using Windows

First, open PuTTY by hitting the windows key (⊞) and then typing “putty” to search for the program. After the program opens, type in tc.rnet.missouri.edu for the Host Name (make sure that SSH is selected and the port is 22 (picture below)).



**NOTE:** When prompted if you trust the source you are connecting to hit yes.

After connecting you should be prompted with text that shows “login as: ” – type your pawprint (your pawprint is a string 5-6 with digits and characters – NOT your student number) and hit enter. It will then ask you for your password (this will be the same password that you use for MyZou). Note that when you are typing your password it will not show anything – but you are still entering it – you just can’t see it.

### Using Mac

Using a Mac is very similar to using PuTTY on Windows. However, Mac already has a built-in program called “Terminal”, so PuTTY is not needed. Just run Terminal. From the prompt, type in:

```
ssh pawprint@tc.rnet.missouri.edu
```

Note that “pawprint” in the above line should be replaced with your actual pawprint ID. You will be asked for your password as shown above under the “Using Windows” section. Again, you will not see your password as you type it (for security).

## Basic UNIX Commands

Here are the basic Unix commands you will need to know in order to create directories, navigate through them, and create files/code.

List of commands:

- **mkdir directoryname** – “mkdir” stands for “make directory” and will create a new directory (think of it like a folder on your computer) with the name that you give it.
- **ls** – “ls” stands for “list” and will show all of the files/directories in the directory you are currently in.
- **cd directoryname** – “cd” stands for “change directory” and will move you into the directory that you specify. If you want to move back a directory you would type “cd ..”
- **pwd** – “pwd” stands for “print working directory” and will show you the current “path” of directories you are in.
- **vim filename.c** – “vim” is a text editor and will be the primary way that we will edit our code. By typing “vim lab1.c” it will create a file called “lab1.c” and will automatically take us into that file. To re-enter a file that currently exists you would still type “vim lab1.c” to enter that file.

## VIM Editor Tips

If you do the command “vim .vimrc”, and type out (cutting and pasting might cause problems so if you want to cut and paste use the vimrc.txt file provided on Canvas) the following code and save it, it’ll make working in vim a bit easier:

```
set number
set mouse=a
set background=dark
colorscheme evening
set expandtab
set smarttab
set shiftwidth=4
set tabstop=4
set lbr
set tw=500
set ai “Auto indent
set si “Smart indent
set wrap “Wrap lines
```

To save and quit, hit esc :wq

## *Getting Started*

For some labs this semester, we will use the `cs1050start` command to get starter code and files that will help us begin the lab. For this lab, you will probably want to create a subdirectory (e.g., “`mkdir cs1050labs`”) and go into that directory (e.g., “`cd cs1050labs`”). At this point, you can get the starter code by issuing this command:

```
cs1050start lab1
```

Now you can go to the `lab1` subdirectory (“`cd lab1`”) and look around (“`ls`”). You should see something like this:

```
Makefile README.md hello.c
```

This means that your new `lab1` directory has 3 files in it named `Makefile`, `README.md`, and `hello.c` respectively. If you want to see what the “Hello World” program should look like you can issue these commands:

```
make hello
```

```
./hello
```

You should see the program output this message:

```
Hello world!
```

Now it is your turn to write your own “Hello World” program.

### *Steps to creating our Hello World! program:*

1. Type vim lab1.c (make sure that the only space that you have is in-between vim and lab1.c – also make sure that you have the “.c” at the end of lab1 – otherwise your code won’t work).
2. From here you will need to hit the letter “i” on your keyboard to be able to type any code. In the bottom right of your screen you should see “-- INSERT --” this shows that you are now able to start typing.
3. From here, type the code shown below exactly as it appears – capitalization matters and but for the most part, spacing does not matter (for now).

```
#include<stdio.h>

int main(void)
{
    printf("Hello World!\n");
}
```

#### **Explanation of the code:**

#include<stdio.h> – will “include” a pre-built library of different functions that we can use – this is not the only library that exists but it’s the only one we need for this program. Libraries are incredibly useful as they save us a ton of time since we don’t need to code everything ourselves thanks to them.

int main(void) – think of this as the “driver” portion of our code – when we compile the code it will know to look for a function called “main” in order to know where to start the program.

printf(“Hello World!\n”); – printf will print whatever is in the quotation marks. \n tells the code to then add a new line wherever we put it.

4. After typing the code, in order to compile it we will need to exit the code and return to our terminal screen. To do that, hit the ESC key – after hitting the key you should no longer see “-- INSERT --” in the bottom left hand corner of your screen. From here, type :wq this stands for write, quit which will save the code and then exit it. If you ever horribly mess up your code and want to revert to your last save you can type :edit! and it will revert all of your changes to the last save of your code – there is no way to get your changes back after typing that so only do that if you know for sure you want to revert your code.
5. From here, we should be back on our terminal screen (where we were prior to typing vim lab1.c). In order to compile our code, type the following: compile lab1.c – this will make our code “readable” to the computer.
6. After doing that if you type “ls” you should now see an executable called “a.out” – there are ways to rename our executable, but for this course we’ll leave it alone. From here, type ./a.out – this will execute our code and you should see Hello World! displayed if everything worked properly.

7. Now you're ready to submit your code! In order to submit the code, type the following command exactly as shown.

**mucs submit 1050 lab lab1.c**

**NOTE:** if you named your file something other than lab1.c you will need to change "lab1.c" in the above command to whatever you named your file.

8. After submitting you should receive text saying the submission was successful!

## Using a File Transfer Protocol (FTP) client

For this lab, we will be using Filezilla as our FTP client. You may choose to use SecureFX or any other FTP client if you prefer. Filezilla is a free download available here:

<https://filezilla-project.org/download.php?type=client>

You likely want to download the standard edition with the “Download” button on the far left, as that edition is free:

### Please select your edition of FileZilla Client

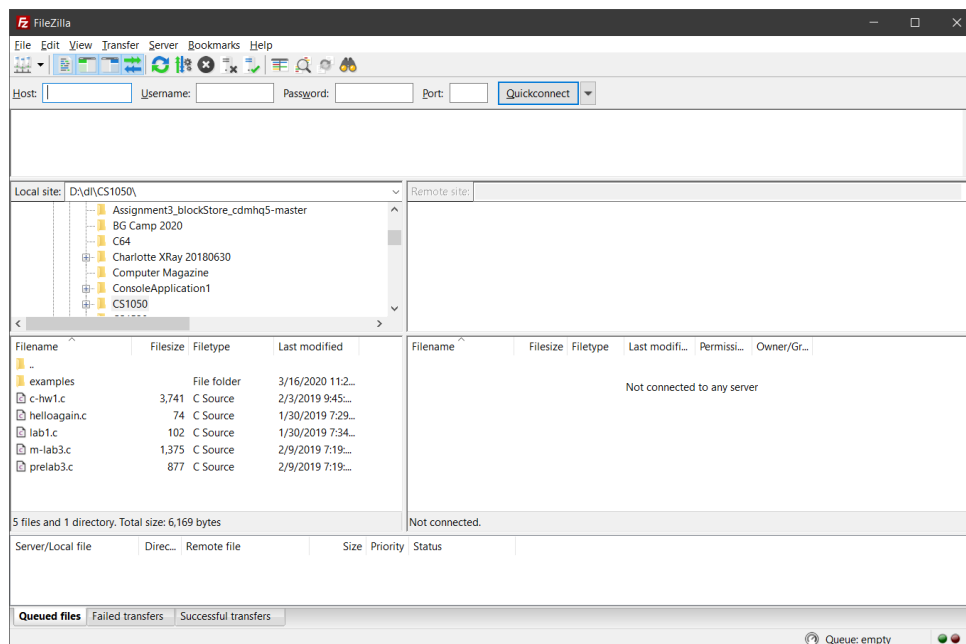
	FileZilla	FileZilla with manual	FileZilla Pro
Standard FTP	Yes	Yes	Yes
FTP over TLS	Yes	Yes	Yes
SFTP	Yes	Yes	Yes
Comprehensive PDF manual	-	Yes	Yes
Amazon S3	-	-	Yes
Backblaze B2	-	-	Yes
Dropbox	-	-	Yes
Microsoft OneDrive	-	-	Yes
Google Drive	-	-	Yes
Google Cloud Storage	-	-	Yes
Microsoft Azure Blob and File Storage	-	-	Yes
WebDAV	-	-	Yes
OpenStack Swift	-	-	Yes
Box	-	-	Yes
Site Manager synchronization	-	-	Yes

Download

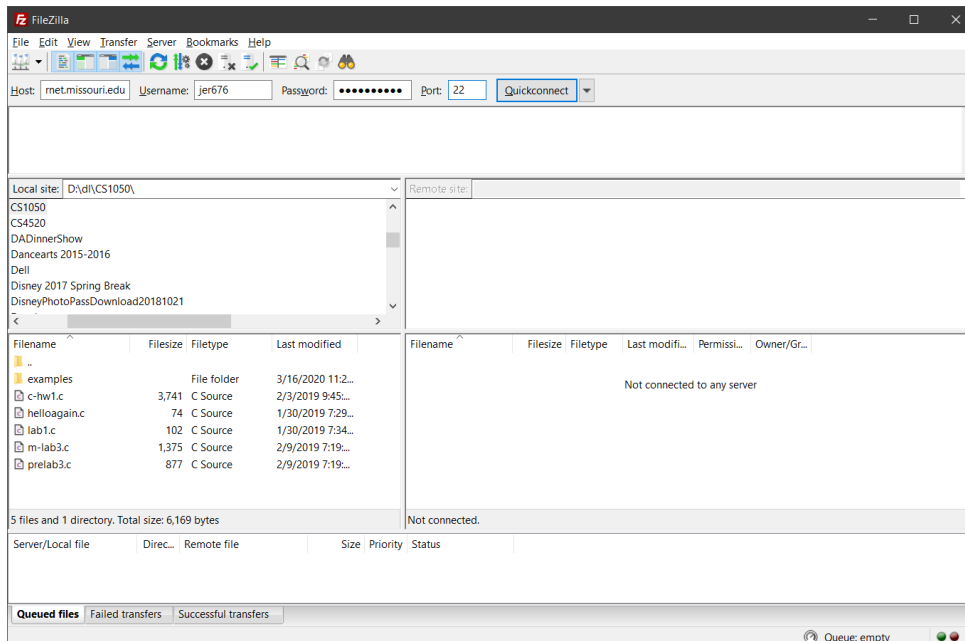
Select

Select

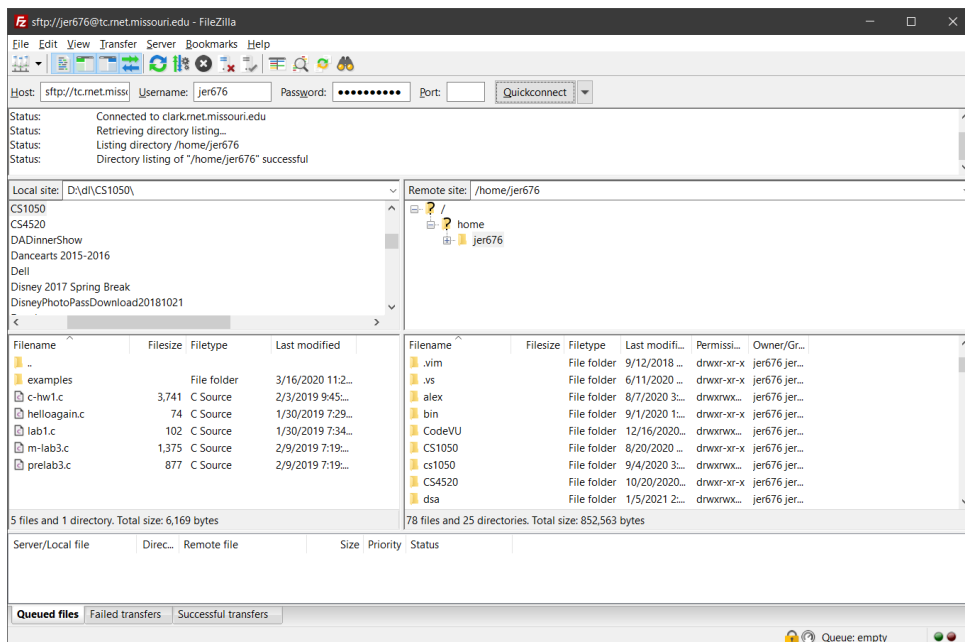
After opening the program, your screen should look something like this:



At the top, next to “Host”, type in tc.rnet.missouri.edu and then your pawprint next to “Username” and then your password next to “Password”. The port should be 22.



Click the “Quickconnect” button. Now your screen should look similar to this:



The files on the left side are what’s on your computer and the files on the right are what are on your personal portion of tc.rnet – to transfer a file simply click and drag that file from one side to the other!

For the purposes of CS 1050 that’s about all you’ll need to know for how an FTP client works!



## **Guidelines for Grading Lab 1**

### **10 Points Possible**

#### **General**

This section of the lab handout will tell you how we plan to grade your program. For this one, you really should get all 10 points as long as you typed in the program and submitted it properly. However, in the future, there may be parts of your program that work and parts that do not work. This “rubric” will help you understand which parts of your program are most important to get right so you can get as many points as possible. Also, in future labs, we will ask you to have a comment header at the top of your file and we will also ask that your code is nicely indented and formatted.