**CSE Grid Tutorial**

**Introduction:**

CSE Grid (or csegrid) is a Linux server which is physically located in North Classroom Building. It runs Linux operating system. Linux is an operating system like Windows or Mac OS X. You will be using Linux server to complete your assignments in CSCI 2312 (Object Oriented Programming), CSCI 2421 (Data Structure), CSCI 3453 (Operating Systems) and other courses which uses C++ as programming language.

**Goals:**

The goal of this tutorial is to learn how to access csegrid and use it for completing your C++ assignments. In this tutorial we will also learn how to compile and run our C++ program. We will also look at a simple make file which can be used to compile C++ program with multiple files.

**Installing and Using:**

CSE Grid (or csegrid) is physically located in North Classroom Building. CSE Grid can only be accessed if we are connected to the UC Denver Network. On campus we are connected to UC Denver Network if we are using UC Denver WiFi (not the UC Denver Guest WiFi). Off campus we must use an encrypted tunnel to connect to the UC Denver Network. Instructions for installing appropriate version of VPN are given on the following web page:

<https://www.ucdenver.edu/offices/office-of-information-technology/software/how-do-i-use/vpn-and-remote-access>

Follow the instructions which are given under “Option1. Virtual Private Network (VPN)” section of the web page. Note that the exact process will depend on the operating system that you are using. We will also have to download and install Duo application for our smart phone. Instructions for installing Duo are given on the following web page:

<https://www.ucdenver.edu/offices/office-of-information-technology/software/how-do-i-use/getting-started-with-multi-factor-authentication>

After installing VPN software and Duo app follow the instructions given in the following pdf file to connect to UC Denver Network:  
<https://www1.ucdenver.edu/docs/default-source/offices-oit-documents/vpn-client-software/multifactor-vpn-user-guide.pdf?sfvrsn=54b49eb8_2>

**Creating and Editing C++ Files**

We can create and edit our C++ files on csegrid. We will discuss this later in this tutorial. To get started we will create the following four files on our computer using a text editor (Notepad on MS Windows and TextEdit on Mac OS X): main.cpp, convert.h, convert.cpp, and data.txt. Make sure that you save the files with appropriate extension.

// file: main.cpp

#include <iostream>

#include <string>

#include <fstream>

#include "convert.h"

using namespace std;

int main()

{

double fahren, celsius;

int value1 = 25;

string city;

ifstream infile("data.txt");

if (!infile) {

cout << "File not found\n";

}

while (getline(infile, city))

{

infile >> fahren;

infile.ignore();

//between stream operators and getline you must have a .ignore()

celsius = fahrenToCel(fahren);

cout << "The celsius temp in " << city << " is " << celsius <<endl;

}

return 0;

}

// file: convert.h

#ifndef CONVERT\_H

#define CONVERT\_H

//function prototypes belong in .h file

double fahrenToCel(double fahren);

#endif

// File: convert.cpp

#include "convert.h"

double fahrenToCel(double fahren)

{

double result = (fahren - 32) \* (5.0 / 9.0);

return result;

}

// File data.txt (Do not include these top two lines in the file. Also make  
// sure that there are no blank lines in the file)

Denver

32

Miami

75

Houston

90

Boston

22

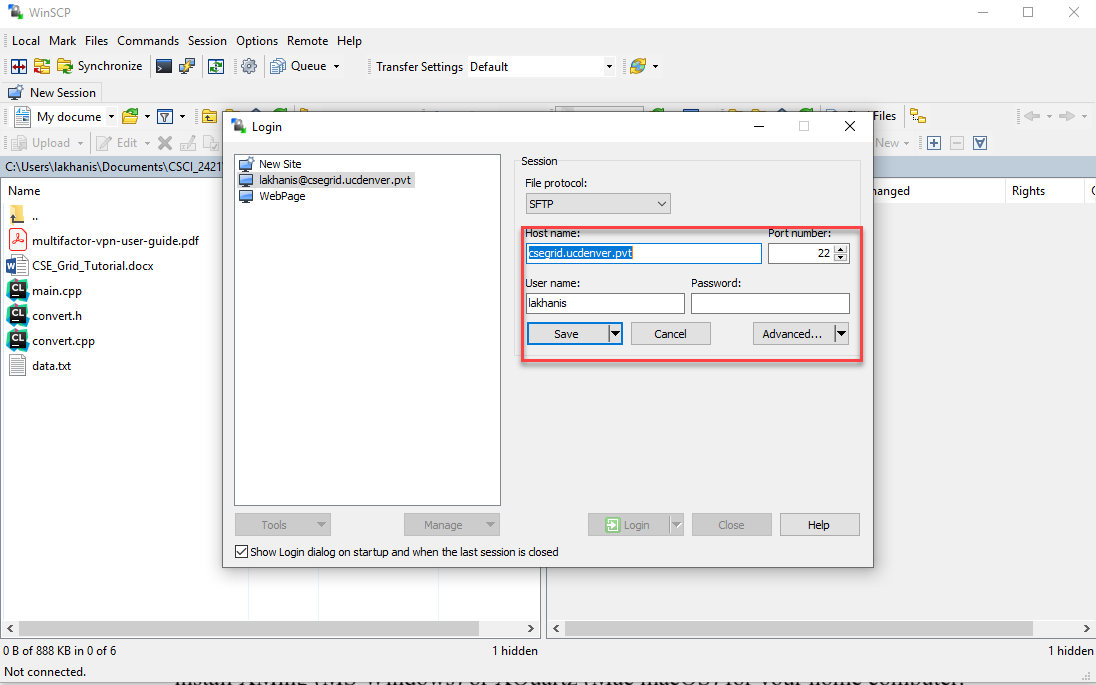
**Installing and Using WinSCP (MS Windows)**

We can use WinSCP to transfer files from our computer to CSE Grid. It is an open source software. It can be downloaded from the following web site:

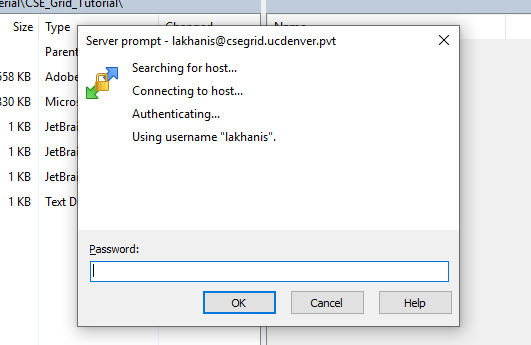
<https://winscp.net/eng/index.php>

Using WinSCP to transfer C++ file from our computer to CSE Grid

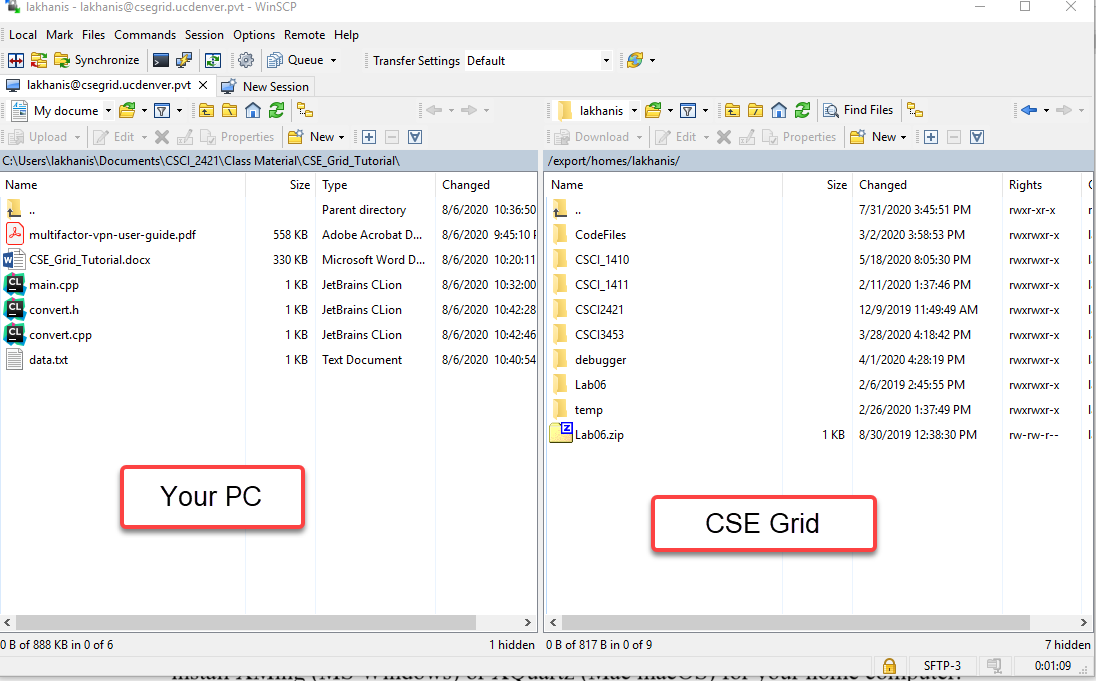
* Connect to UC Denver Network using VPN.
* Start WinSCP.



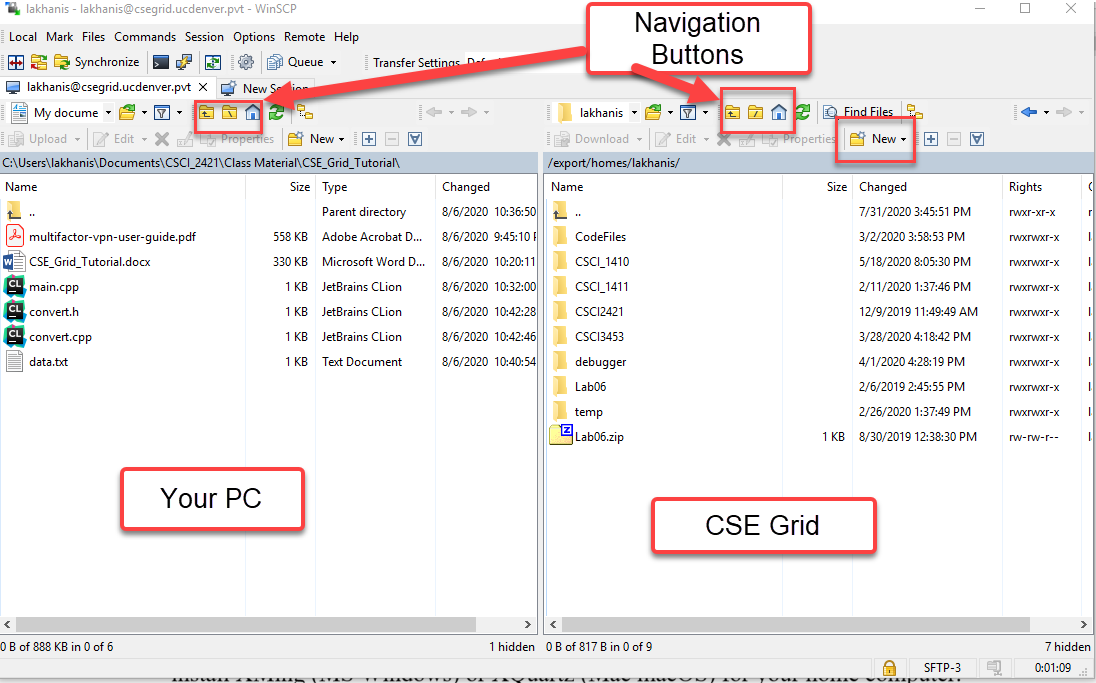
* In the Host name text box type csegrid.ucdenver.pvt and make sure that Port number is set to 22.
* Type user name in User name text box and click on Save button. Now click on Login.



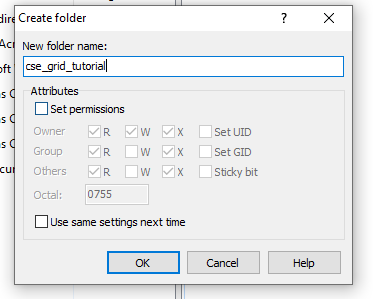
* Type your password and click OK.
* Note: Use the User Name and Password that you use to get into your Canvas account.
* You will see two windows. Your PC is on the left side and the csegrid is on the right.



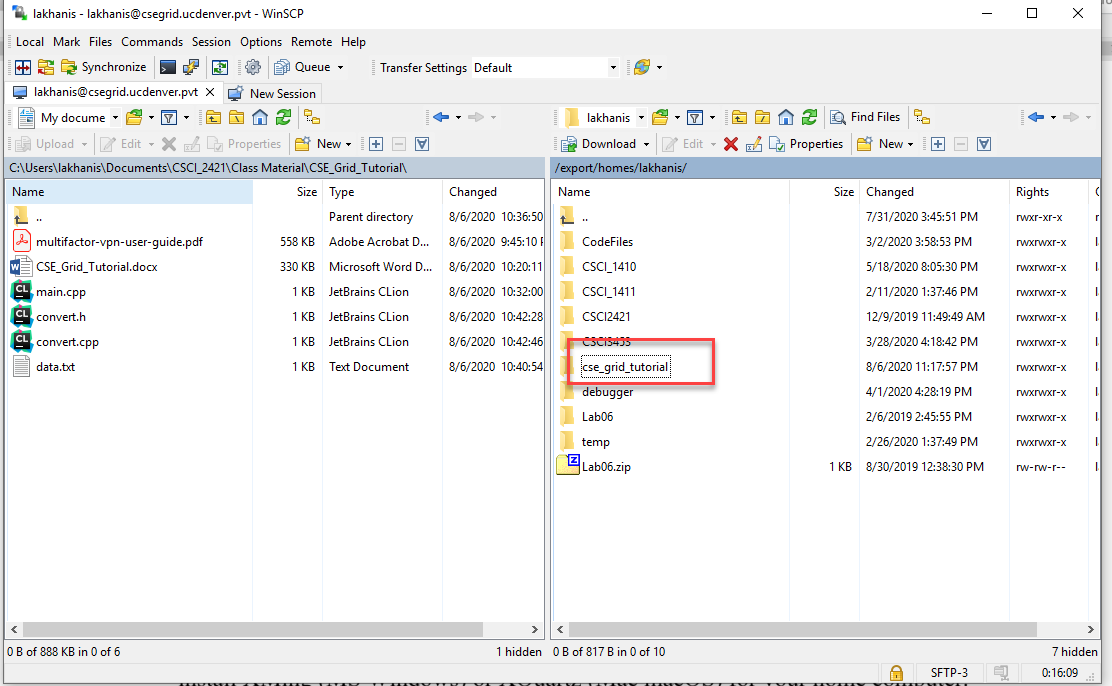
* On the left hand side, find the directory which contains C++ files. You can use the navigation buttons to find the directory which contains your C++ files.



* On the right hand side, click the Home Icon (Control + H), then click on New and select Directory from the drop down box (or click F7 key).



* Type cse\_grid\_tutorial in the New folder name text box and click on OK. It will appear in the right window.



* Double click on cse\_grid\_tutorial directory to open it.
* Drag and drop the code file from the left window into the right window. This will upload your files to csegrid server. There are four files that you will have to upload. These files are: main.cpp, data.txt, convert.h and convert.cpp.
* Close the WinSCP window to close the connection.

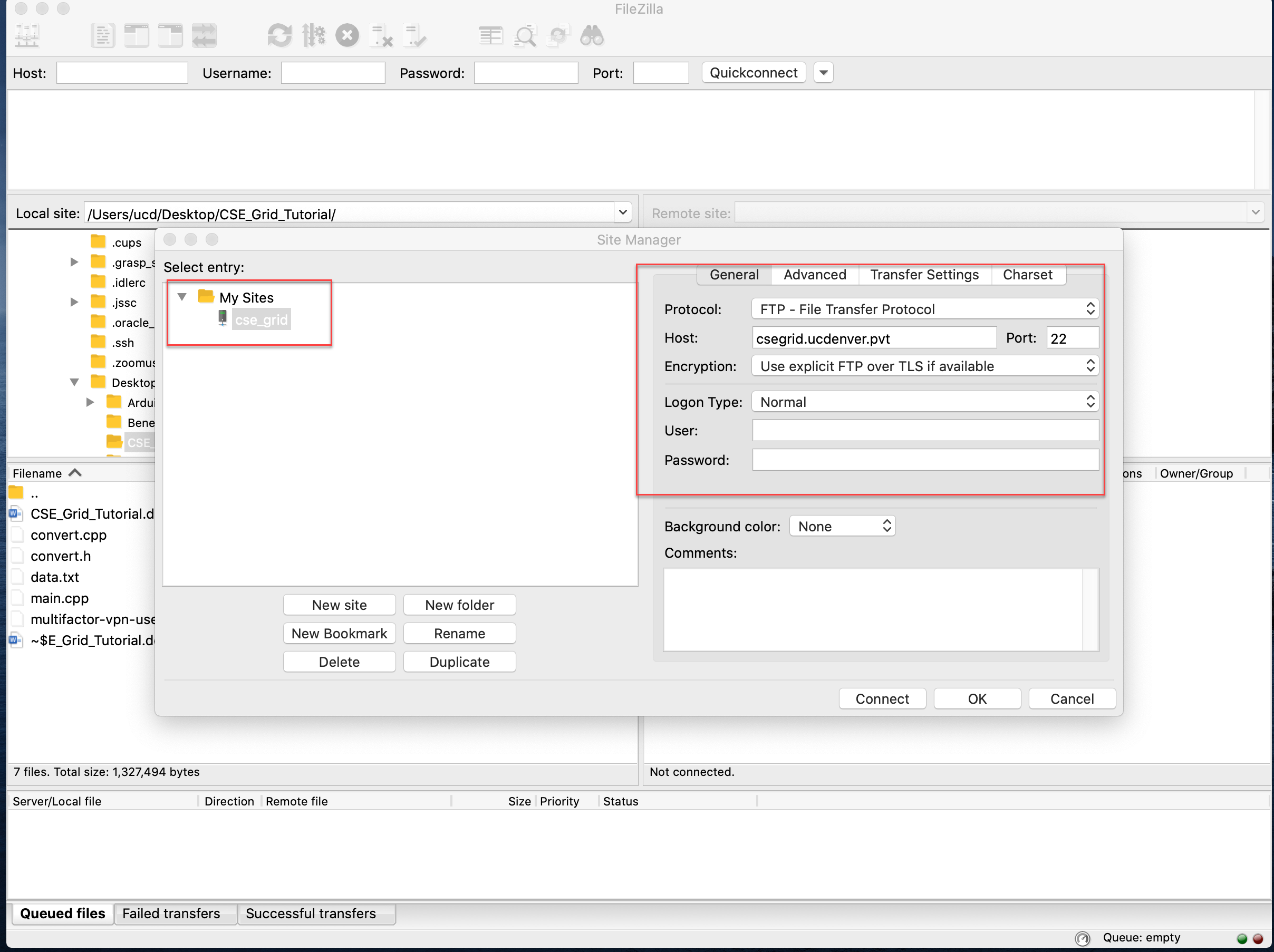
**Installing and Using FileZilla (Mac OS X and MS Windows)**

We can use FileZilla to transfer files from our computer to csegrid. It is an open source software. It can be downloaded from the following web site:

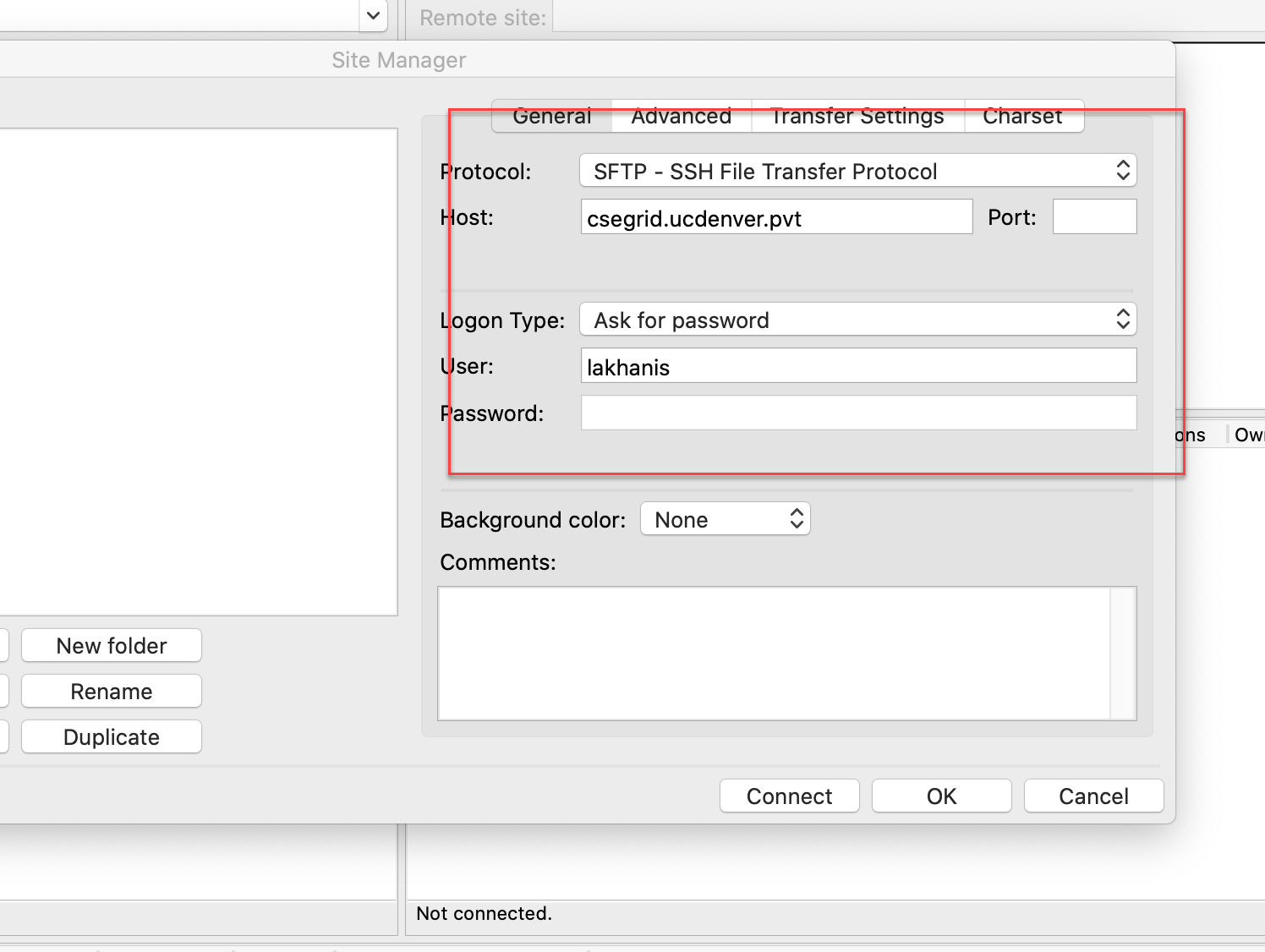
<https://filezilla-project.org/>

Using FileZilla to transfer C++ file from our computer to csegrid

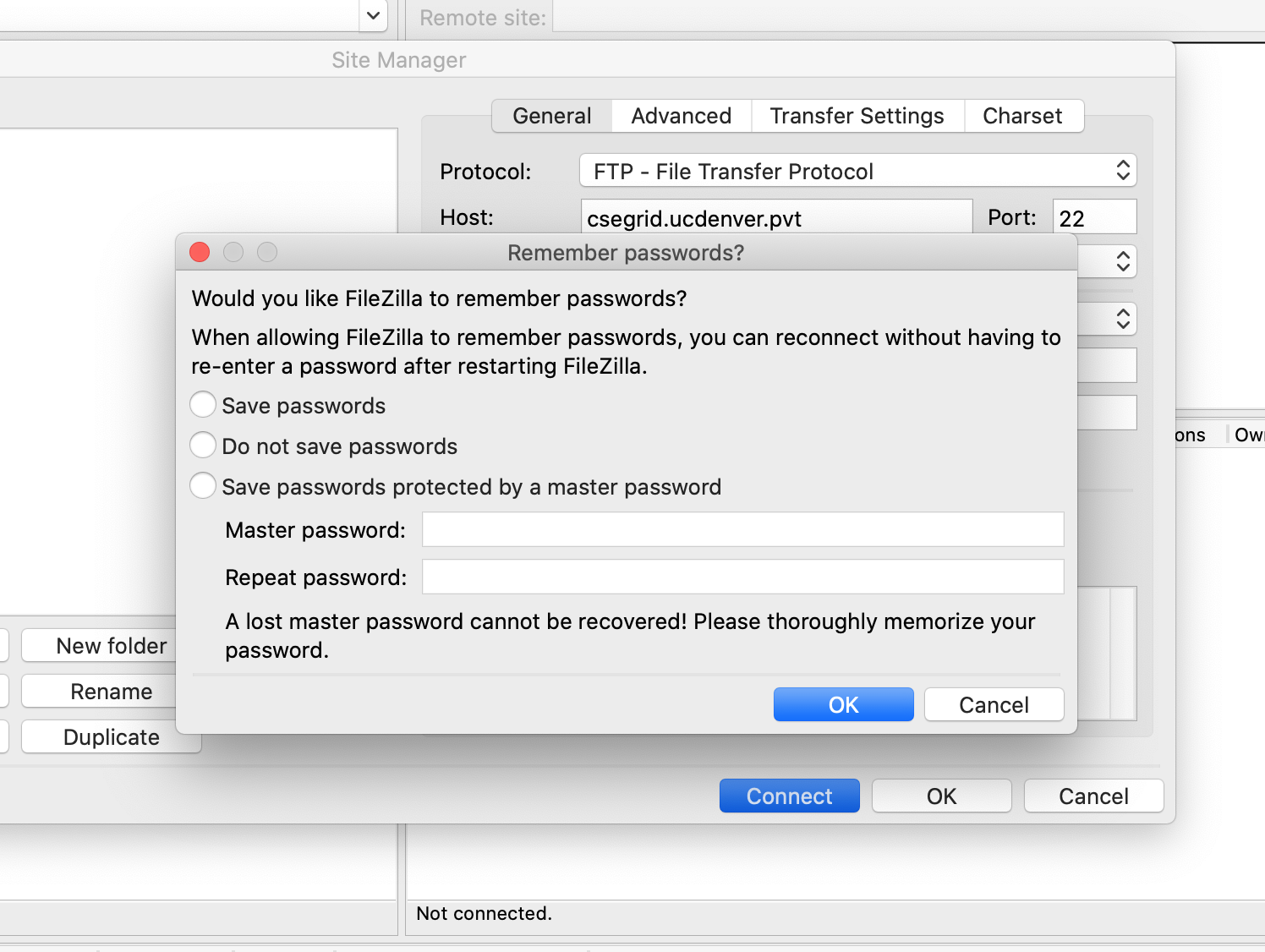
* Connect to UC Denver Network using VPN.
* Start FileZila. Click on File menu and select Site Manager. Click on New Site.



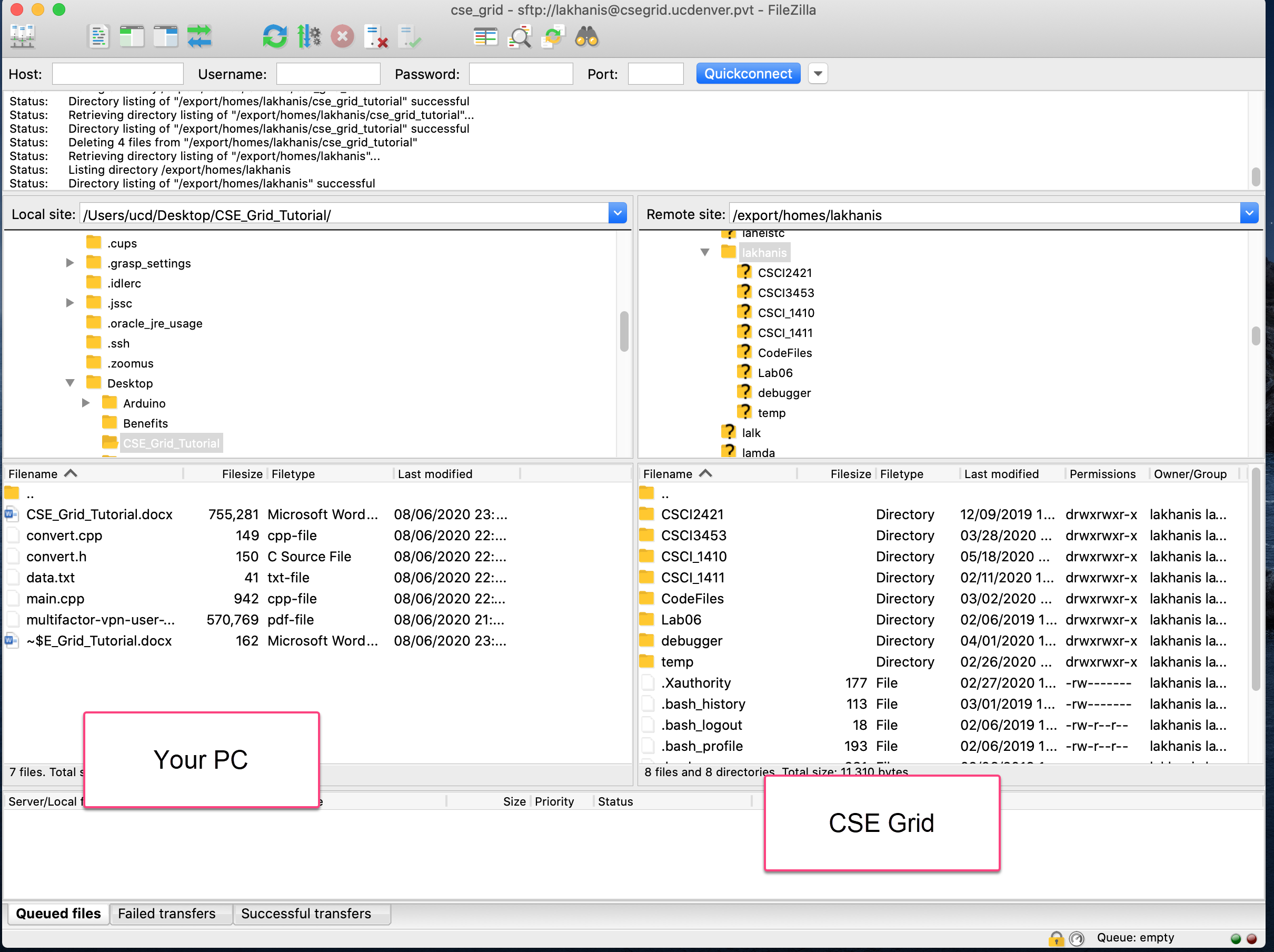
* Give a name to your site. In the Host text box type csegrid.ucdenver.pvt and 22 in text box for Port
* Make sure that Protocol is set to SFTP – SSH File Transfer Protocol and Logon Type is set to Normal. Type your user name and password in the User text box and Password text box. Click on the Connect button.



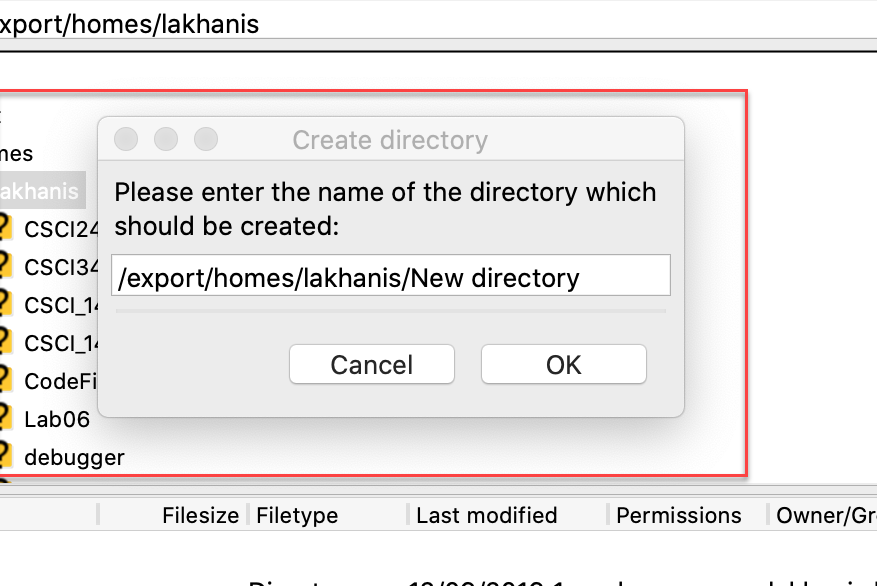
* Note: Use the User Name and Password that you use to get into your Canvas account.
* If you want FileZilla can remember your password.

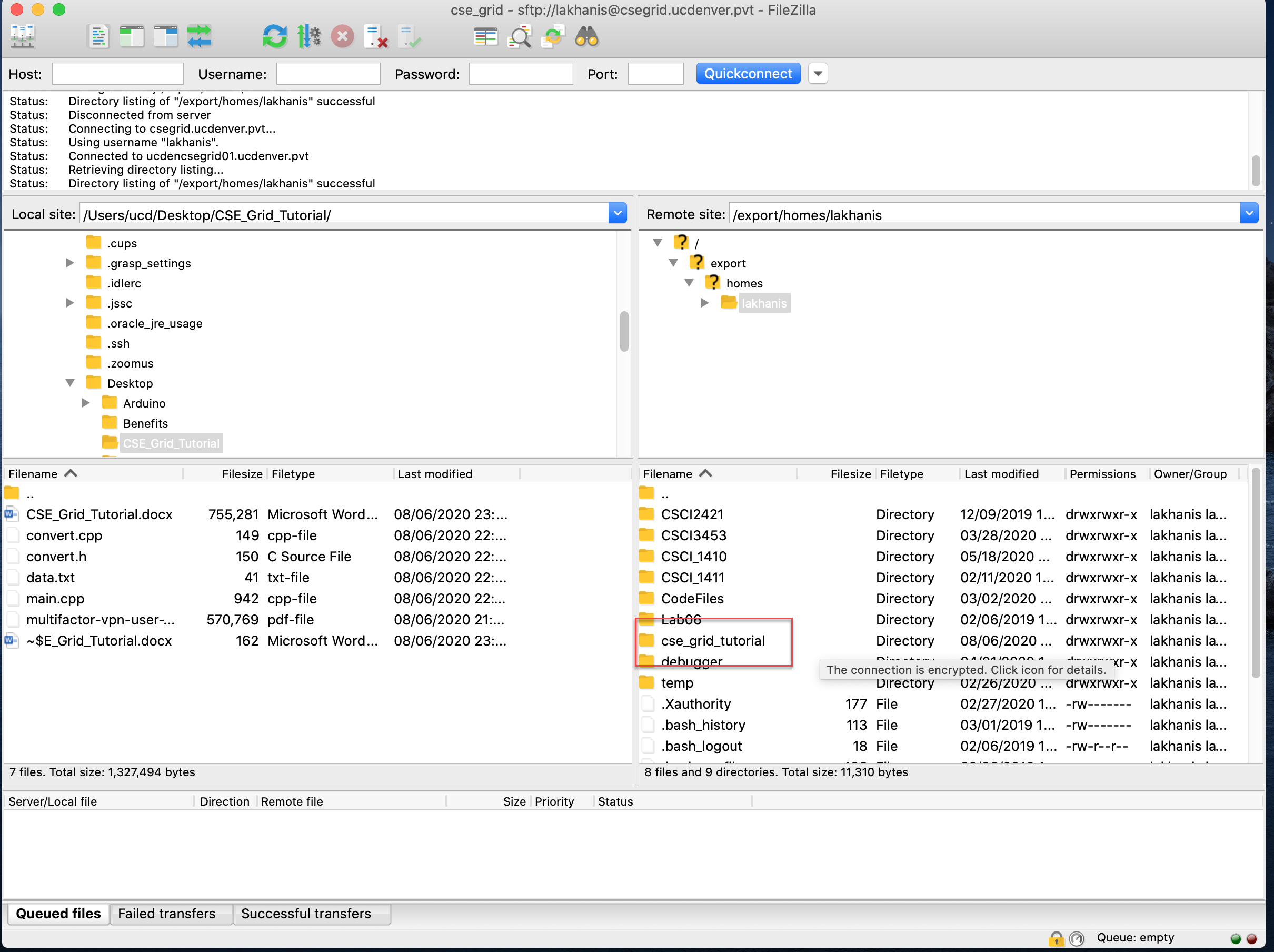


* You will see two windows. Your PC is on the left side and the csegrid is on the right.



* On the left hand side, find the directory which contains C++ files.
* On the right hand side right click on the name of your home folder (this will be same as your user name) and select directory from the quick menu. Change “New Directory” to “cse\_grid\_tutorial” and press OK key. It will appear in the right window.





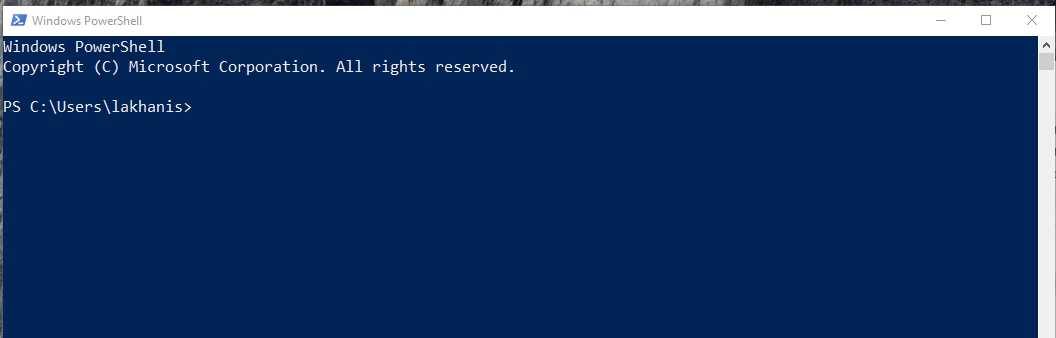
* Double click on cse\_grid\_tutorial to open it.
* Drag and drop the code file from the left window into the right window. This will upload your files to csegrid server. There are four files that you will have to upload. These files are: main.cpp, data.txt, convert.h and convert.cpp.
* Quit the FileZilla window to close the connection.

**Using SSH**

SSH can be used to start a session on CSE Grid. We need to start the session to edit, compile, and run our program on CSE Grid.

Using SSH

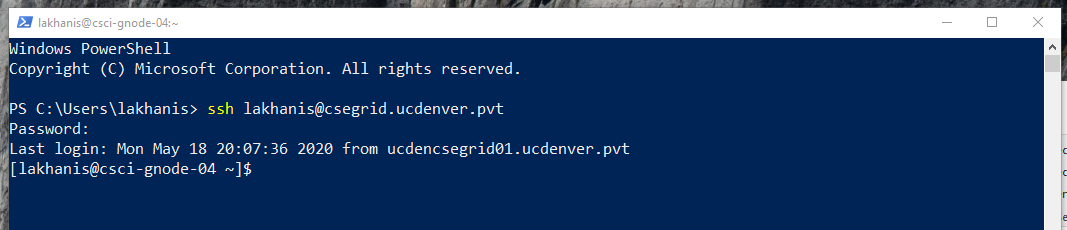
* Connect to UC Denver Network using VPN.
* Open Command Prompt (or cmd on MS Windows), Windows Power Shell (MS Windows) or Terminal (Mac OS X). It will open a window where we can type command to perform different tasks.



* Type ssh [username@csegrid.ucdenver.pvt](mailto:username@csegrid.ucdenver.pvt) (where username is your username) in the window and press enter. It will prompt you for password. Type your password and press enter.



* This will start the session on CSE Grid.
* If you entered correct username and password then you will see a line with $ at the end. This is called the prompt or command prompt, since it prompts you for a response.



* Once you are at the command prompt, you will have to type in commands to determine what directory you are currently in, make directories (which you can also do using WinSCP), change directories, list out what files are in a directory, edit code, compile it and run it.
* You should get good at these procedures. If you are going on to other Computer Science courses, you will see that all programming assignments must compile and run on the CSE Grid. Also, if you get a job or internship in Computer Science you may have to use similar environment. A good Linux command line cheat sheet is available at:  
  <https://cheatography.com/davechild/cheat-sheets/linux-command-line/>
* At the dollar sign ($) prompt type the following command followed by the Enter key:
  + clear clears the screen
  + pwd Print Working Directory…tells you what directory you are in
  + ls lists out what files are in the directory
  + cd cse\_grid\_tutorial change directory to cse\_grid\_tutorial
  + pwd you should see you are one more level deep
  + ls you should see the 4 files that we uploaded using WinSCP or FileZilla.
* Nano is a good editor that we can use to edit files on the csegrid. It provides c++ highlighting for comments, statements, etc.

**Nano**

Type nano main.cpp to start nano editor. Note: We cannot use our mouse, so we will have to use the arrow keys to move around. We can use the following commands:

* Use Control + X to exit (you will be prompted to save your file if you have not).
* Use Control + O to save
* Use Control + G to get help – this will display all the possible commands that you can use. Linux command cheat sheet also includes Nano shortcuts.
* You can cut and paste but it is a little awkward without a mouse
* Exit out without saving

**Compiling and Running Programs**

* On the CSE Grid there are no build and run button, so we have to type commands.
* We are going to go back to your home directory, move to the cse\_grid\_tutorial directory, compile the program and then run it. At the $ prompt type these commands (one at a time) and hit the Enter key after each command:
  + cd
  + cd cse\_grid\_tutorial
  + ls
  + g++ -o app main.cpp convert.cpp
* Notes:
  + g++ is the name of the compiler which is used to compile cpp code.
  + -o is used to name the output file which is app in our case. This is an optional parameter. If we omit –o app in the above command then it will create an output file with default name. Default name is a.out.
  + We need to list all our .cpp files to compile them. We do not have to list any .h files, input files or output files.
* If there are errors then you will need to edit the appropriate file and repeat the g++ command. You can either edit the file using nano on CSE Grid or you can edit it on your own computer using a text editor. In case if you edit the file on your own computer then you will have to upload the file to CSE Grid using WinSCP or FileZilla.
* We can run the compiled code using the following command  
  ./app

**Using Make file**

Makefiles are a simple way to organize the process of compiling your program. We can use make utility to compile our program. Make utility will look into the make file to get the name of the output file as well as name of the files to compile to create the output file. Makefile can be used to compile program in any programming language. A sample make file is as follows:

#Place this makefile in the same directory as your

#all .cpp and .h files

#at the command prompt

#make

#./target (or whatever you call the output file)

#for use with c++ files

.SUFFIXES : .cpp

#Compile with g++

CC = g++

#use -g for gnu debugger and -std= for c++11 compiling

CXXFLAGS = -g -std=c++11

######Change to match all .cpp files. Do not include .h files####

OBJS = list all .o files

######Change from target to name of output file

TARGET = target

######Do NOT change this...Tabs Matter!#########

$(TARGET): $(OBJS)

$(CC) -o $@ $(OBJS)

.cpp.o:

$(CC) -c $(CXXFLAGS) $(INCDIR) $<

#In order to get rid of all .o files create, at the command prompt

#make clean

clean:

rm -f $(OBJS) $(TARGET) core

The format of the make file is important. Make sure that you do not change any spacing or tabs. Use the attached make file. We need to change only the highlighted line. On the first highlighted line we will list all .o file corresponding to our .cpp files. In our example we have two cpp files: main.cpp and convert.cpp. The corresponding .o files are main.o and convert.o. Next we need to specify the name of the output file. We will change target to whatever name we want to use for our output file. Makefile with the above updates will be as follows:

#Place this makefile in the same directory as your

#all .cpp and .h files

#at the command prompt

#make

#./target (or whatever you call the output file)

#for use with c++ files

.SUFFIXES : .cpp

#Compile with g++

CC = g++

#use -g for gnu debugger and -std= for c++11 compiling

CXXFLAGS = -g -std=c++11

######Change to match all .cpp files. Do not include .h files####

OBJS = main.o convert.o

######Change from target to name of output file

TARGET = app

######Do NOT change this...Tabs Matter!#########

$(TARGET): $(OBJS)

$(CC) -o $@ $(OBJS)

.cpp.o:

$(CC) -c $(CXXFLAGS) $(INCDIR) $<

#In order to get rid of all .o files create, at the command prompt

#make clean

clean:

rm -f $(OBJS) $(TARGET) core

Place the makefile in the same directory as all our program files and execute the make command to compile our C++ program.

**Using Debugger**

Debugging a program on CSE Grid is a little tricky. We can use gdb to debug our C++ program. See the following video for a tutorial:

<https://www.youtube.com/watch?v=bWH-nL7v5F4>

Cheat Sheet for gdb debugger: <https://darkdust.net/files/GDB%20Cheat%20Sheet.pdf>