Getting Started with C++

Part 2

Linux

Getting Started on Linux

- Now we will look at Linux.

- See how to copy files between Windows and Linux
- Compile and run the “Hello, World” program on Linux.
- Try out a very simple editor on Linux.

About Circe

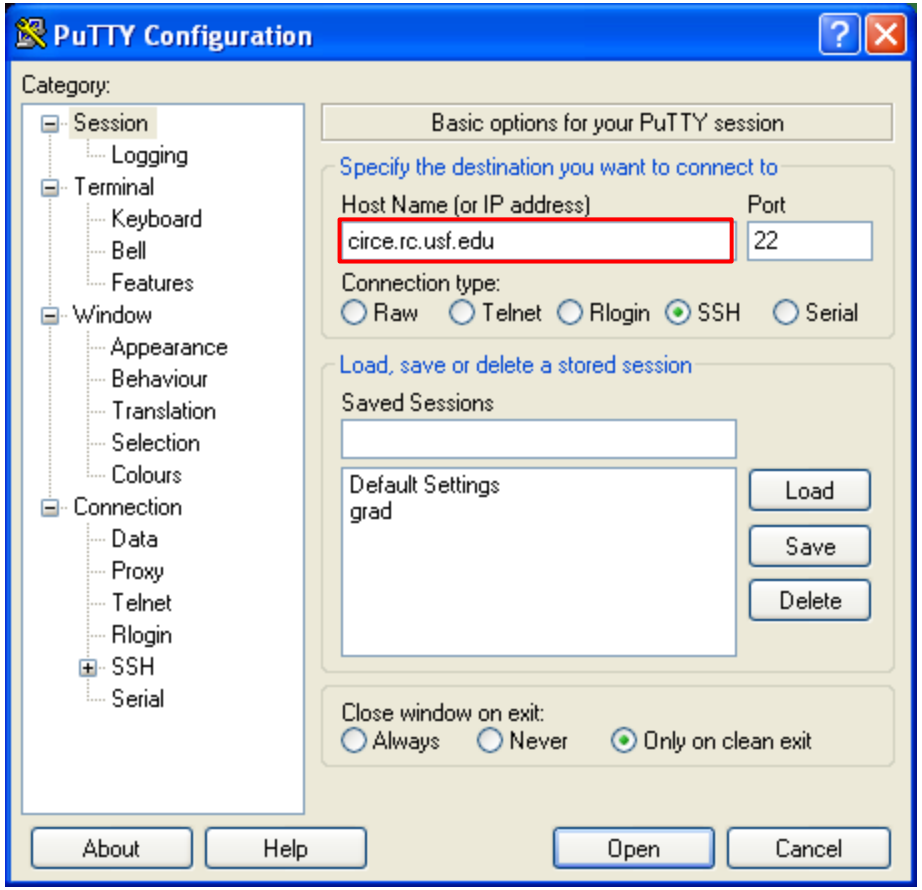
- All USF students have access to a Linux system known as Circe.
- Try logging in with your USF NetID.
  - (Details follow.)
- If unsuccessful, you should be able to get access at rc.usf.edu
  - If unsuccessful there, please contact the USF IT Help Desk. 974-1222

Connecting to Circe

- Use an SSH terminal client program to connect to Circe.
- Recommended client program is PuTTY
  - Can download from http://www.putty.org/

- You will also need an SSH file transfer program.
- Recommended client program is WinSCP
  - Can download from https://winscp.net/eng/download.php

Connecting with PuTTY



Click Open

Using Circe

```
turnerr@login0:~
login as: turnerr   Your Net ID

      University of South Florida Central Instructional and Research
                      Computing Environment (CIRCE)

     WARNING: UNAUTHORIZED ACCESS TO THIS SYSTEM IS PROHIBITED BY LAW

turnerr@circe.rc.usf.edu's password:        Your Net ID Password
Last login: Sun Nov 28 20:51:29 2010 from 72.187.18.171

             Documentation:  https://rc.usf.edu/trac/doc
                   Support:  help@usf.edu
                     Phone:  813-974-1222

   To change your password or to recover a lost password, please visit
                        https://netid.usf.edu

   Please report any and all problems to the Helpdesk at help@usf.edu so we
   can track your incident!

[turnerr@login0 ~]$ Command Prompt
```

Create a Test Directory

turnerr@login1:~/test

```
[turnerr@login1 ~]$
[turnerr@login1 ~]$
[turnerr@login1 ~]$
[turnerr@login1 ~]$
[turnerr@login1 ~]$
[turnerr@login1 ~]$
[turnerr@login1 ~]$
[turnerr@login1 ~]$
[turnerr@login1 ~]$
[turnerr@login1 ~]$
[turnerr@login1 ~]$
[turnerr@login1 ~]$
[turnerr@login1 ~]$ mkdir test
[turnerr@login1 ~]$ cd test
[turnerr@login1 test]$ ls
[turnerr@login1 test]$
[turnerr@login1 test]$
[turnerr@login1 test]$
```

Copying a File to Circe

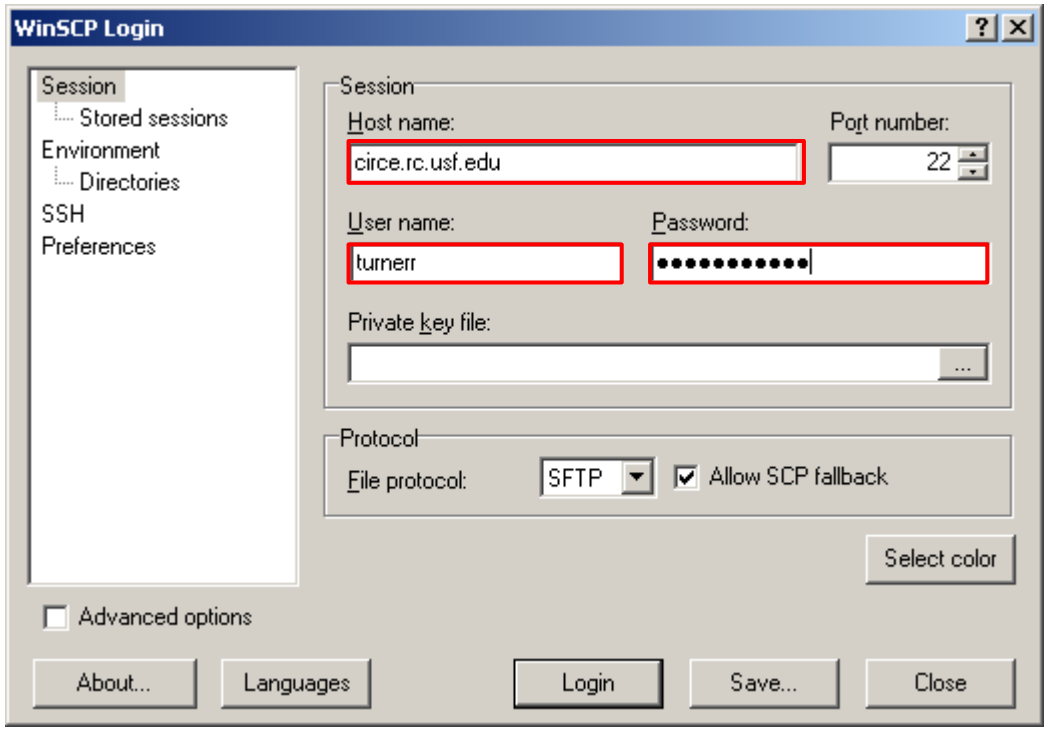
- Use an SSH file transfer program to copy the C++ source file to your test directory on Circe.
  - Recommended program: WinSCP

WinSCP

- WinSCP gives you a window on your desktop that looks and acts like a normal Windows folder.

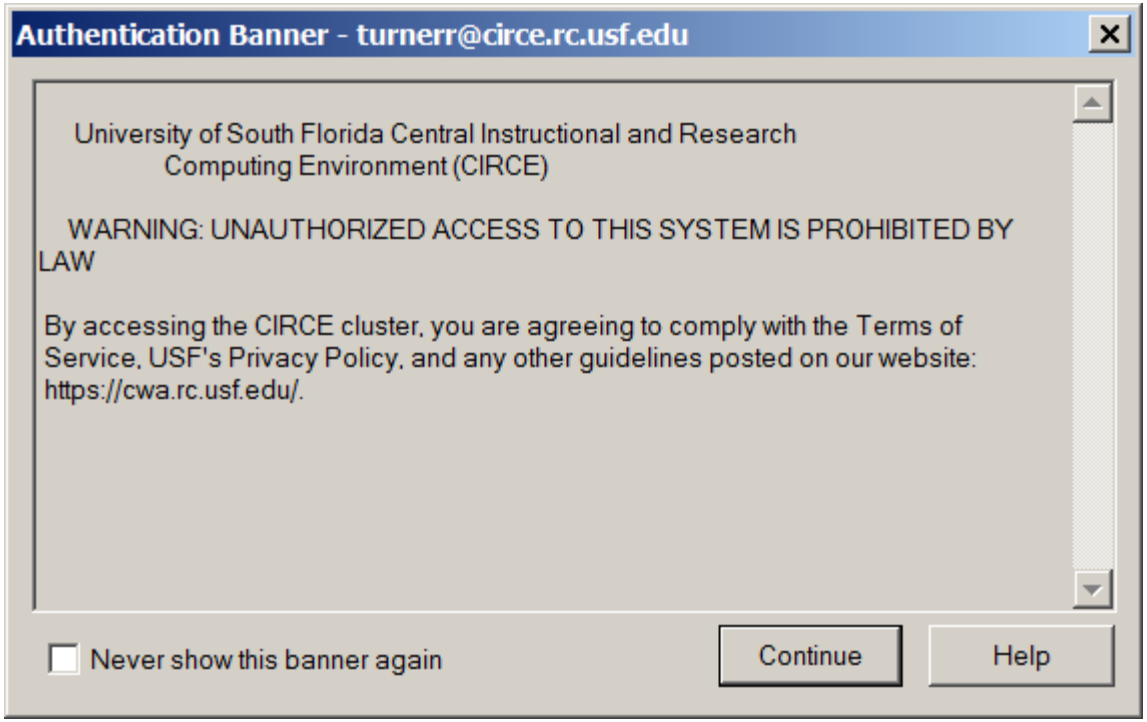
- You can drag and drop between the WinSCP window and a Windows folder
  - in either direction.

Connecting to Circe with WinSCP

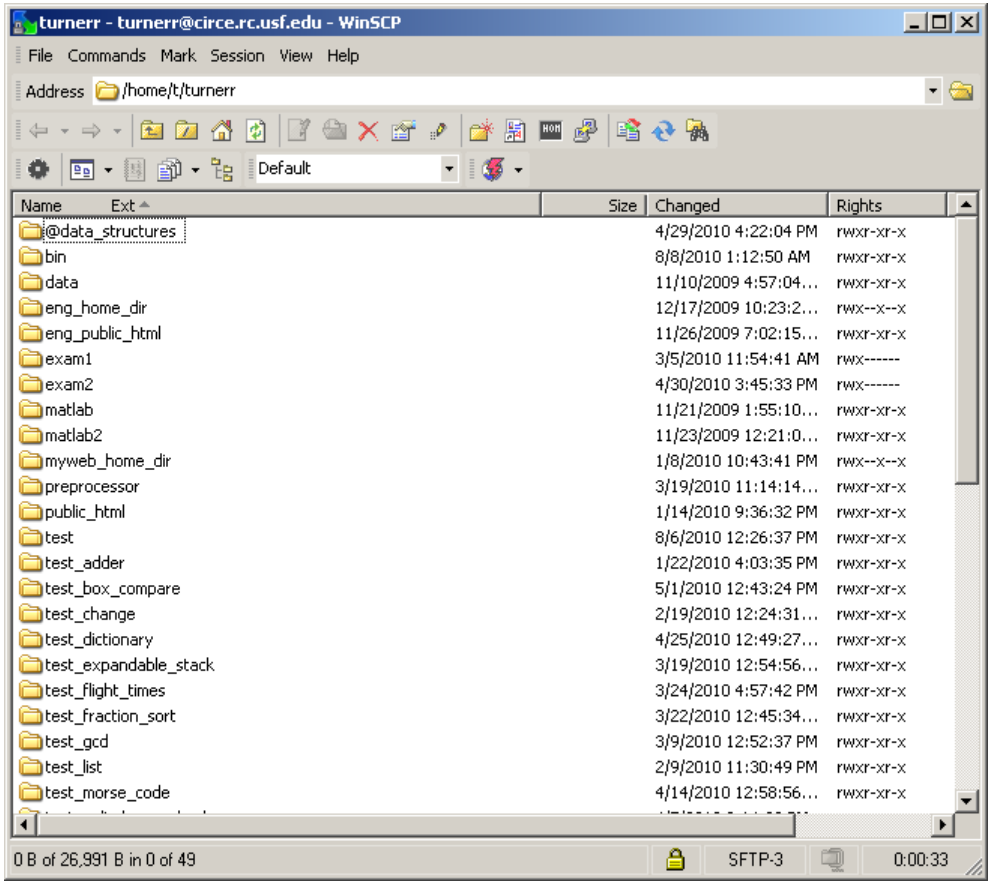


Click Login

Connected to Circe

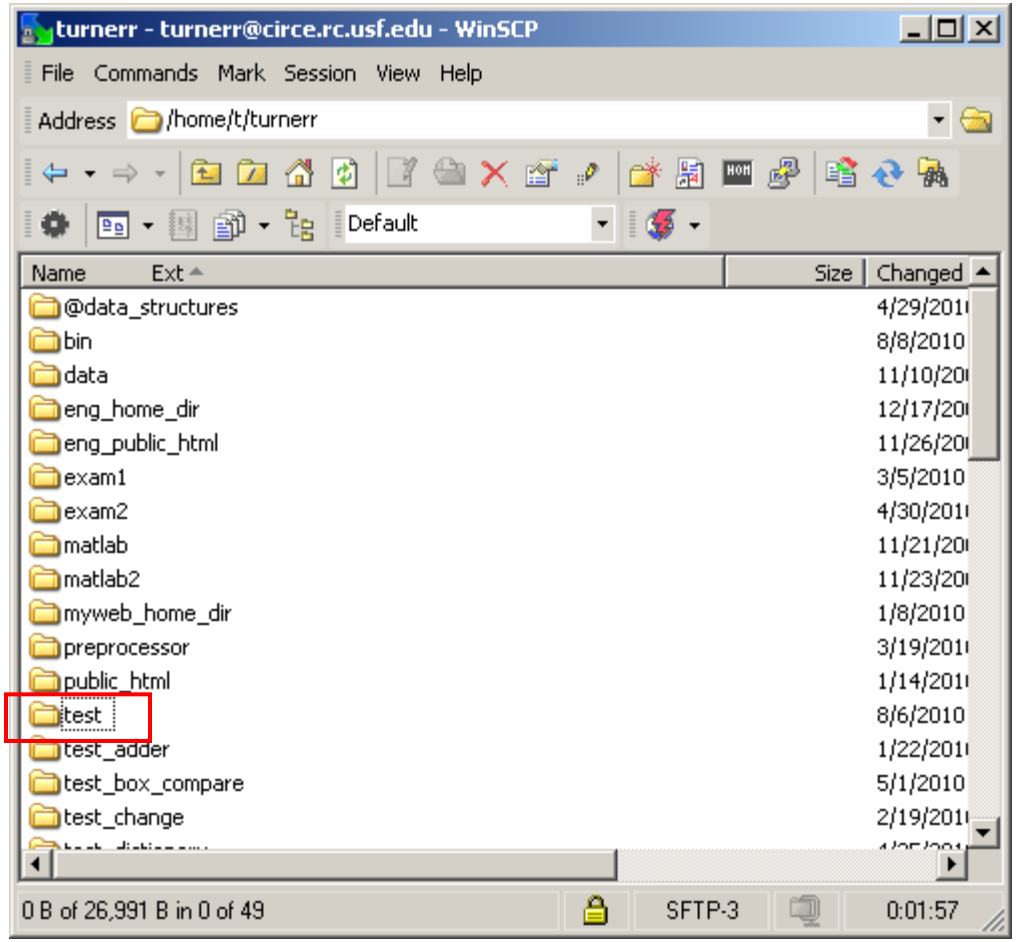


Connected to Circe



Initially at your home directory.
Your contents will be different.

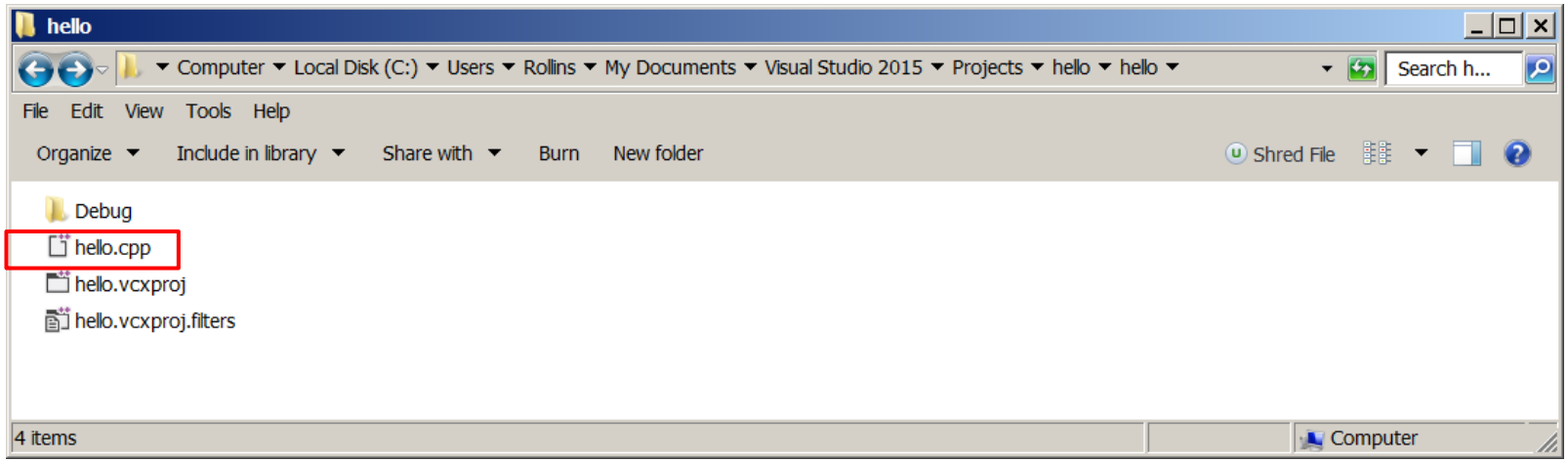
Open your test directory



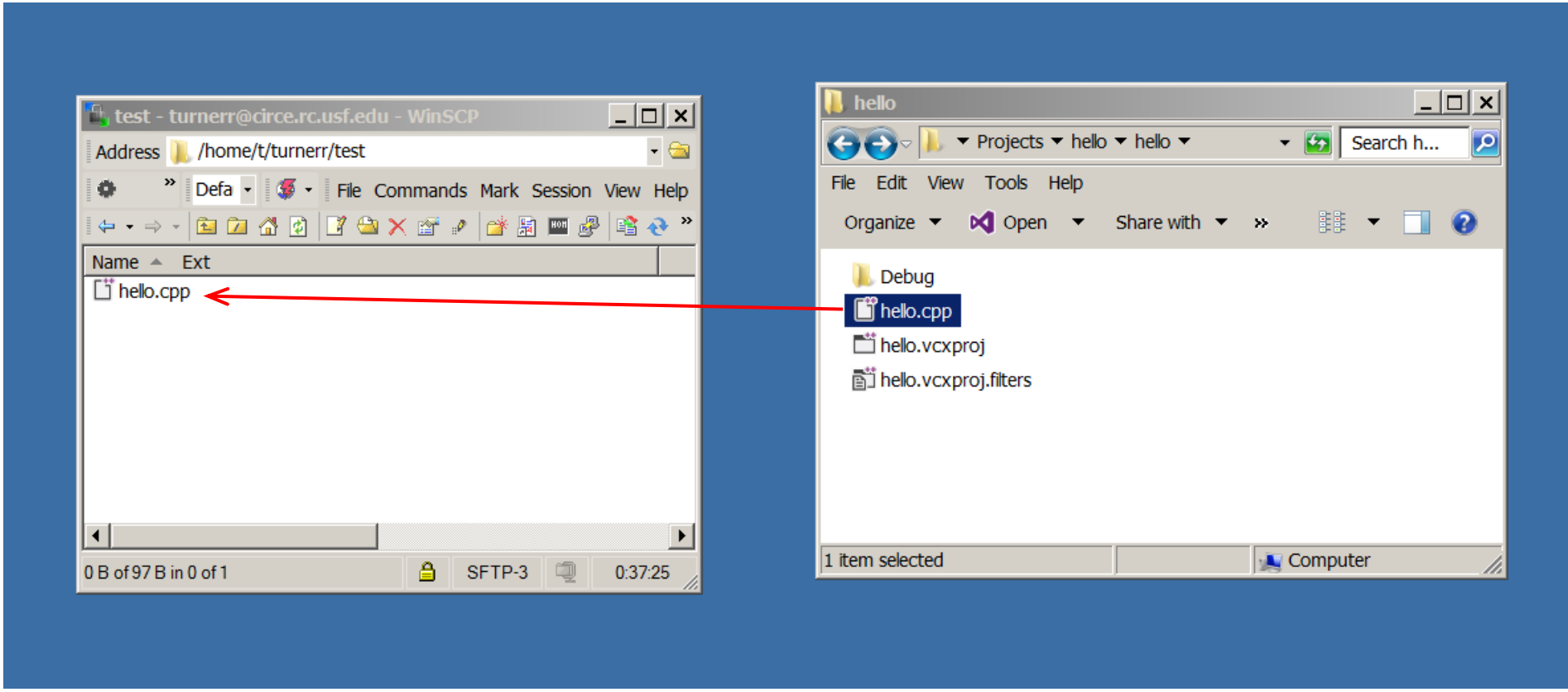
Double click on your test directory to open it.

Open your project directory

- Locate and open your Visual Studio project directory
- By default it is in your "My Documents" folder under Visual Studio 2015\Projects
  - C:\Users\Rollins\Documents\Visual Studio 2015\Projects\hello
- Drill down to hello.cpp

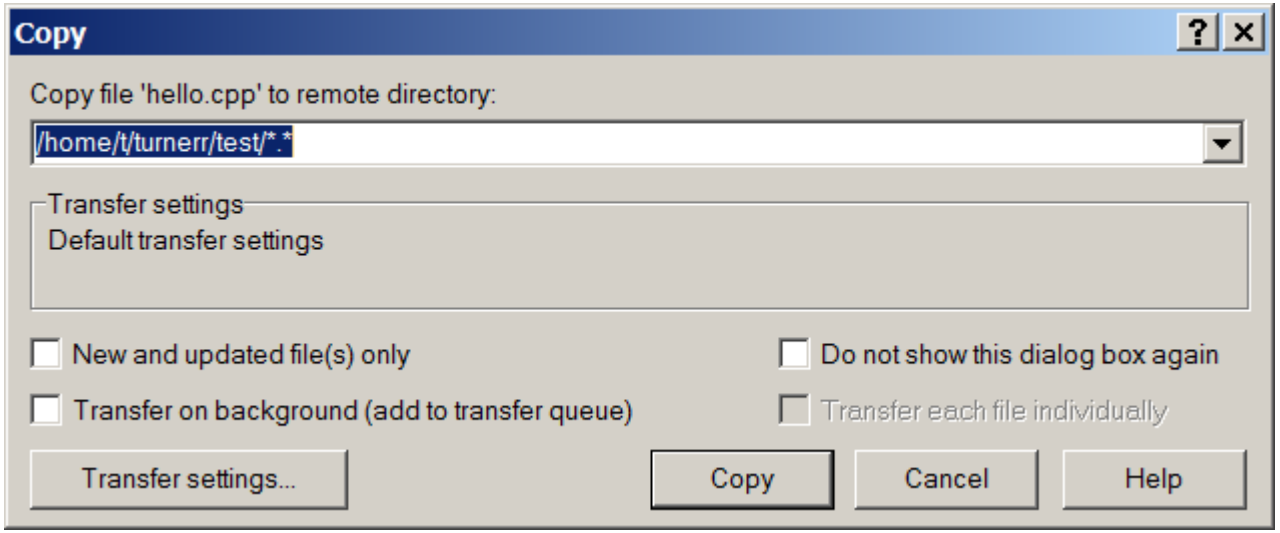


Copy hello.cpp to Circe



Drag hello.cpp from your Visual Studio project directory window into the WinSCP window and drop it.

Copy hello.cpp to Circe



In the PuTTY Terminal Window

```
turnerr@login0:~/test
[turnerr@login0 test]$
[turnerr@login0 test]$ cat hello.cpp
#include <iostream>

int main()
{
    std::cout << "Hello, World!\n";

    std::cin.get();     // Keep the window open.

    return 0;
}

[turnerr@login0 test]$ g++ hello.cpp
[turnerr@login0 test]$
[turnerr@login0 test]$
```

Look at the file

Compile and link

No response from g++ means that it was successful.
Your executable file is named a.out

Run It

turnerr@login0:~/test

```
int main()
{
    std::cout << "Hello, World!\n";

    std::cin.get();     // Keep the window open.

    return 0;
}

[turnerr@login0 test]$ g++ hello.cpp
[turnerr@login0 test]$
[turnerr@login0 test]$ ./a.out
Hello, World!

[turnerr@login0 test]$
```

Creating a Source File on Circe

Delete existing files. We will start from scratch.

```
turnerr@login0:~/test

[turnerr@login0 test]$ g++ hello.cpp
[turnerr@login0 test]$
[turnerr@login0 test]$ ./a.out
Hello, World!

[turnerr@login0 test]$
[turnerr@login0 test]$
[turnerr@login0 test]$
[turnerr@login0 test]$ ls
a.out  hello.cpp
[turnerr@login0 test]$ rm *
[turnerr@login0 test]$
[turnerr@login0 test]$ ls
[turnerr@login0 test]$
[turnerr@login0 test]$ nano hello.cpp
```

nano is a very simple text editor.

Creating a Source File on Circe

```
turnerr@login7:~/test
  GNU nano 2.0.9                File: hello.cpp

^G Get Help    ^O WriteOut    ^R Read File   ^Y Prev Page   ^K Cut Text    ^C Cur Pos
^X Exit        ^J Justify     ^W Where Is    ^V Next Page   ^U UnCut Text  ^T To Spell
```

Whatever you type into this window will be written to the output file.

Creating a Source File on Circe

```
turnerr@login7:~/test
  GNU nano 2.0.9                File: hello.cpp                          Modified

#include <iostream>

int main()
{
        std::cout << "Hello, World!\n";

        return 0;
}

^G Get Help    ^O WriteOut    ^R Read File   ^Y Prev Page   ^K Cut Text    ^C Cur Pos
^X Exit        ^J Justify     ^W Where Is    ^V Next Page   ^U UnCut Text  ^T To Spell
```

Press Ctrl-o to write out the file.

Creating a Source File on Circe

```
turnerr@login7:~/test
  GNU nano 2.0.9                File: hello.cpp                        Modified

#include <iostream>

int main()
{
        std::cout << "Hello, World!\n";

        return 0;
}



File Name to Write: hello.cpp
^G Get Help          ^T To Files          M-M Mac Format       M-P Prepend
^C Cancel            M-D DOS Format       M-A Append           M-B Backup File
```

Press Enter to write contents of window to hello.cpp

Creating a Source File on Circe

```
turnerr@login7:~/test
  GNU nano 2.0.9                File: hello.cpp

#include <iostream>

int main()
{
        std::cout << "Hello, World!\n";

        return 0;
}

                                [ Wrote 9 lines ]
^G Get Help   ^O WriteOut   ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit       ^J Justify    ^W Where Is   ^V Next Page  ^U UnCut Text ^T To Spell
```

Press Ctrl-x to exit from nano.

Exit from the Editor

```
turnerr@login0:~/test
    return 0;
}

[turnerr@login0 test]$ g++ hello.cpp
[turnerr@login0 test]$
[turnerr@login0 test]$ ./a.out
Hello, World!

[turnerr@login0 test]$
[turnerr@login0 test]$
[turnerr@login0 test]$
[turnerr@login0 test]$ ls
a.out  hello.cpp
[turnerr@login0 test]$ rm *
[turnerr@login0 test]$
[turnerr@login0 test]$ ls
[turnerr@login0 test]$
[turnerr@login0 test]$ nano hello.cpp
[turnerr@login0 test]$
```

View the Source File

```
turnerr@login0:~/test
[turnerr@login0 test]$ rm *
[turnerr@login0 test]$
[turnerr@login0 test]$ ls
[turnerr@login0 test]$
[turnerr@login0 test]$ nano hello.cpp
[turnerr@login0 test]$
[turnerr@login0 test]$
[turnerr@login0 test]$ cat hello.cpp
#include <iostream>

int main()
{
    std::cout << "Hello, World!\n";

    return 0;
}


[turnerr@login0 test]$
```

Compile and Run

```
turnerr@login0:~/test
[turnerr@login0 test]$
[turnerr@login0 test]$
[turnerr@login0 test]$ cat hello.cpp
#include <iostream>

int main()
{
    std::cout << "Hello, World!\n";

    return 0;
}


[turnerr@login0 test]$ g++ hello.cpp
[turnerr@login0 test]$
[turnerr@login0 test]$ ./a.out
Hello, World!
[turnerr@login0 test]$
[turnerr@login0 test]$
```

The Manipulator `endl`

```
#include <iostream>

int main( void )
{
    std::cout << "Hello, World!";

    std::cout << std::endl;

    return 0;
}
```

`std::endl` is not a *character* like **`'\n'`**

It is an instruction to cout to start a new line.

Referred to as a *manipulator*.

Multiple outputs with cout

```
#include <iostream>

int main( void )
{
    std::cout << "Hello, World!" << std::endl;

    std::cin.get();
    return 0;
}
```

<< operators can be cascaded as many times as you wish.

Input from the Keyboard

```
#include <iostream>

int main( void )
{
   int a;
   int b;
   std::cout << "Enter two integers to compute their sum:" << std::endl;

   std::cin >> a;
   std::cin >> b;

   std::cout << "The sum of " << a << " and " << b  << " is ";
   std::cout << a + b << std::endl;

   return 0;
}
```

Input from the Keyboard

```
turnerr@login0:~/test
[turnerr@login0 test]$ cat ./add.cpp
#include <iostream>

int main(void)
{
   int a;
   int b;
   std::cout << "Enter two integers to compute their sum:" << std::endl;

   std::cin >> a;
   std::cin >> b;

   std::cout << "The sum of " << a << " and " << b  << " is ";
   std::cout << a + b << std::endl;

   return 0;
}

[turnerr@login0 test]$
[turnerr@login0 test]$ g++ add.cpp
[turnerr@login0 test]$ ./a.out
Enter two integers to compute their sum:
5
7
The sum of 5 and 7 is 12
[turnerr@login0 test]$
[turnerr@login0 test]$
```

Avoiding all those "std::"s

```
#include <iostream>

using namespace std;

int main( void )
{
    int a;
    int b;
    cout << "Enter two integers to computer their sum:" << endl;

    cin >> a;
    cin >> b;
    cout << "The sum of " << a << " and " << b  << " is ";
    cout << a + b << endl;

    return 0;
}
```

Being More Selective

```
#include <iostream>

using std::cin;
using std::cout;
using std::endl;

int main( void )
{
    int a;
    int b;
    cout << "Enter two integers to computer their sum:" << endl;

    cin >> a;
    cin >> b;
    cout << "The sum of " << a << " and " << b  << " is ";
    cout << a + b << endl;
    ...
    return 0;
}
```

This is generally considered better practice.

Assignment

Before next class

- Be sure you can connect to Circe and log in using your USF NetID.
- Do today's examples for yourself
  if you didn't do them in class.
- Read Chapters 1 and 2.

End of Presentation