RUNNING YOUR C CODE ON OSU SERVER MACHINES

CS 261

STEP 1: SETTING UP YOUR SYSTEM

Contents

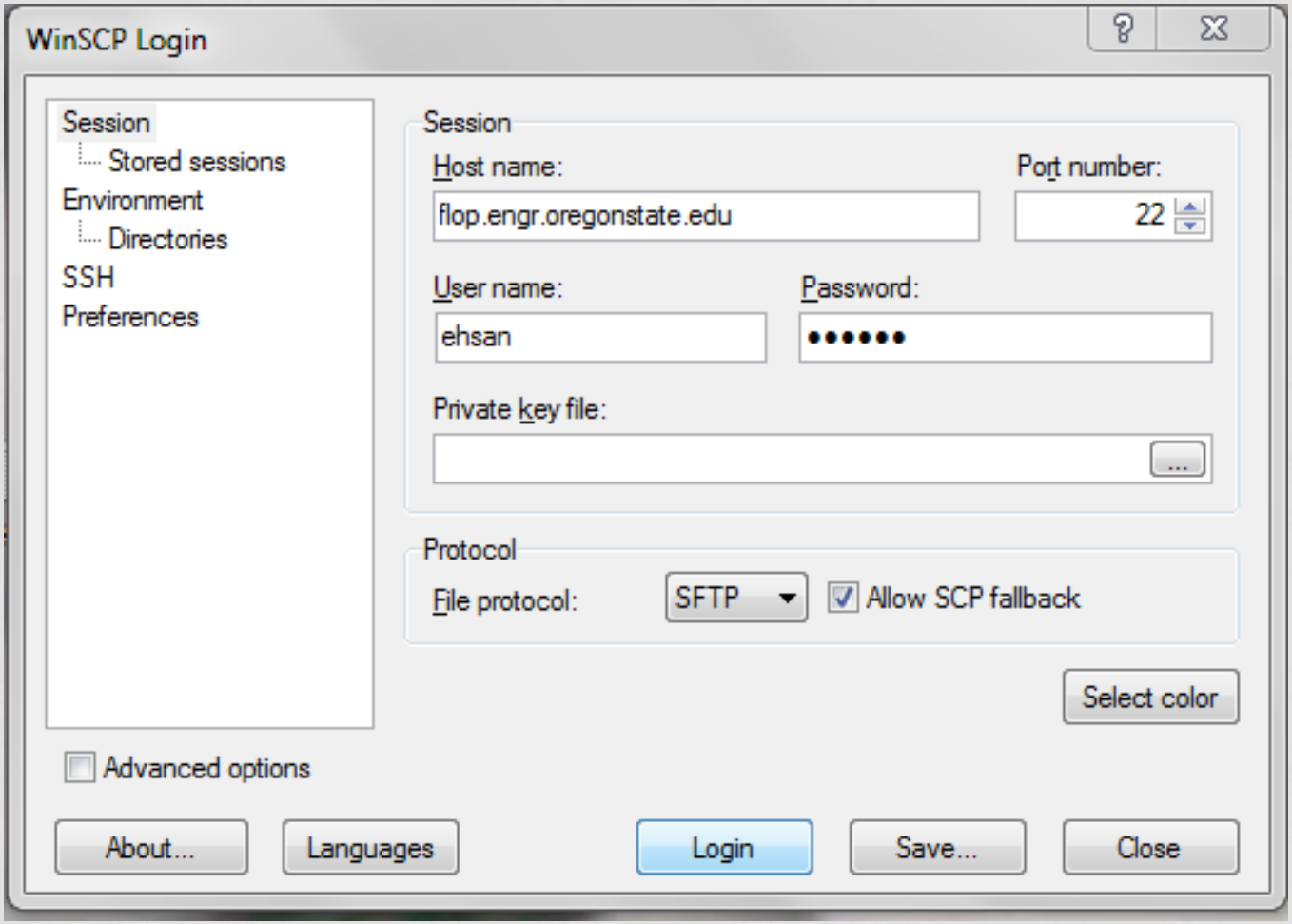
- Flip and flop servers at OSU
- Using WinSCP (ForWindows) (http://winscp.net/eng/download.php)
- Using Putty (http://www.putty.org/)
- Using ssh, scp from Linux terminal

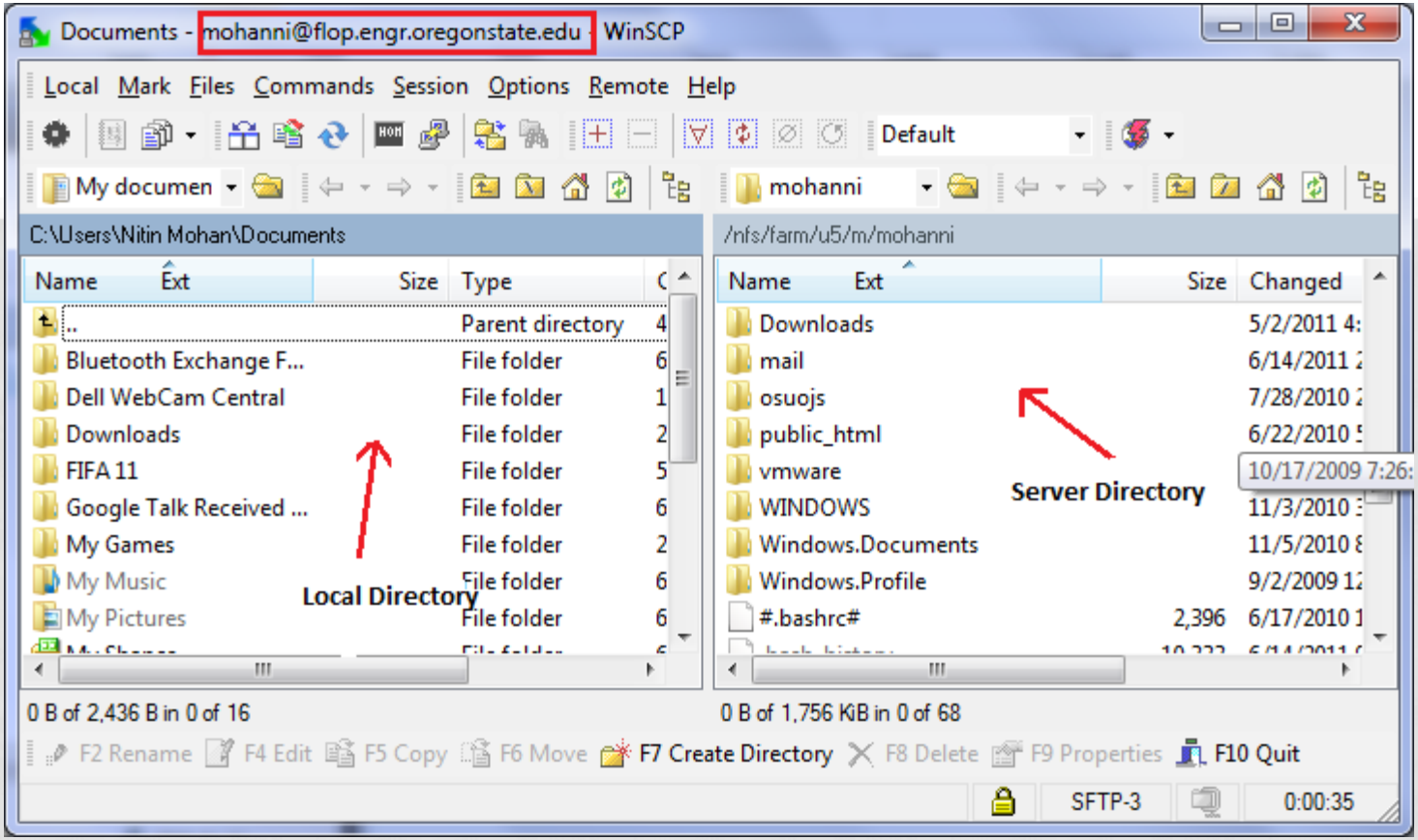
FLIP AND FLOP SERVERS

- Flip (flip.engr.oregonstate.edu) – a Linux terminal server at OSU.
  - Accessible anywhere **on-campus** using a SSH client.
- Flop (flop.engr.oregonstate.edu) – Linux terminal server at OSU.
  - Accessible from anywhere **off- campus** using a SSH client.
- Other servers at OSU
  - access.engr.oregonstate.edu (both flip and flop connect to this)
  - Shell.onid.oregonstate.edu
- All these servers uses your ENGR credentials except for shell.onid.

USING WINSCP

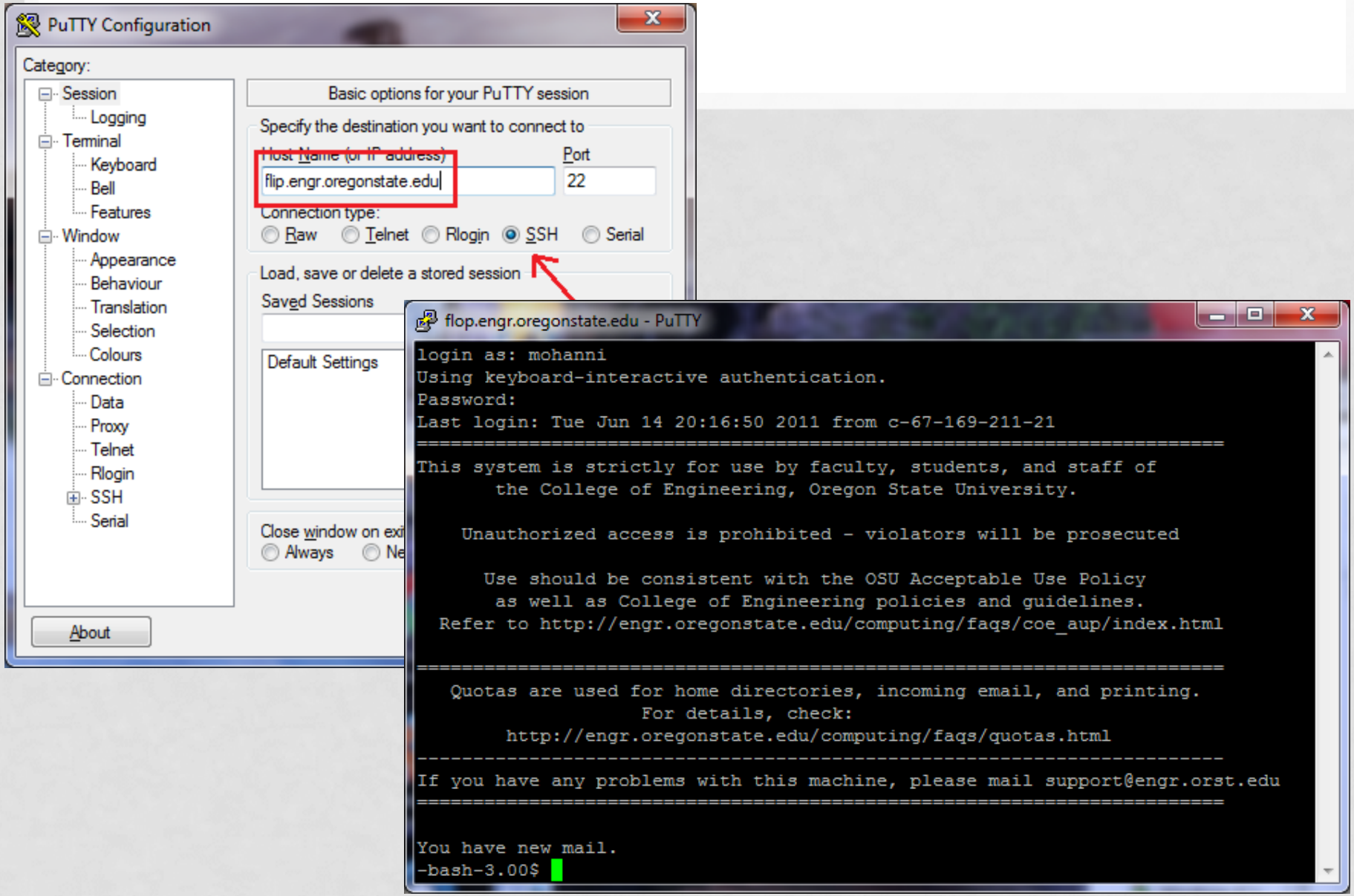
- Free software (only for Windows) for file transfer via SCP, FTP, SFTP.
- To transfer files to the flop server





- Files can be transferred by drag and drop.
- Software also features a basic linux terminal which executes the commands in the server.

USING PUTTY


PuTTY Configuration
Category:
Session
Logging
Terminal
Keyboard
Bell
Features
Window
Appearance
Behaviour
Translation
Selection
Colours
Connection
Data
Proxy
Telnet
Rlogin
SSH
Serial
Basic options for your PuTTY session
Specify the destination you want to connect to
Host Name (or IP address)
Port
flip.engr.oregonstate.edu
22
Connection type:
Raw
Telnet
Rlogin
SSH
Serial
Load, save or delete a stored session
Saved Sessions
Default Settings
Close window on exi
Always
Ne
About
flop.engr.oregonstate.edu - PuTTY
login as: mohanni
Using keyboard-interactive authentication.
Password:
Last login: Tue Jun 14 20:16:50 2011 from c-67-169-211-21
This system is strictly for use by faculty, students, and staff of
the College of Engineering, Oregon State University.
Unauthorized access is prohibited - violators will be prosecuted
Use should be consistent with the OSU Acceptable Use Policy
as well as College of Engineering policies and guidelines.
Refer to http://engr.oregonstate.edu/computing/faqs/coe_aup/index.html
Quotas are used for home directories, incoming email, and printing.
For details, check:
http://engr.oregonstate.edu/computing/faqs/quotas.html
If you have any problems with this machine, please mail support@engr.orst.edu
You have new mail.
-bash-3.00$

USING SSH FROM TERMINAL

- ssh command is in-built in the linux terminal
- It can be used to connect to a server remotely via terminal
- Usage: ssh username@serveraddr
- Example: fickerd@flop.engr.oregonstate.edu

USING SSH ON A MAC.

- The best way to do this is through your terminal. You'll find it in:
  -Macintosh HD->Application->Utilities->Terminal
- After its opened, go to "Shell (or File)" and click on "New Remote Connection".
- Then click on "Secure Shell (ssh). There technically shouldn't be anything in the right window that just flushed onto your screen, under the "server" part side.

USING SCP

- Scp is to securely copy files from local machine to server directory and vice versa using command line interface.
- Usage:

From the host to your computer: scp username@hostname:~/remotepath localpath

From your computer to the host: scp localpath username@hostname:~/remotepath

- Example:
- To move a file from my desktop to my web space I would use -

scp ~/Desktop ehsan@flop.engr.oregonstate.edu:~/public_html

STEP 2: RUNNING YOUR CODE USING GCC

- The GNU Compiler Collection (usually shortened to GCC) is a command line compiler system produced by the GNU Project, supporting various programming languages (includes C, C++).

- Compiling with GCC:

  gcc <list of options> sourcefile.c

- e.g.: gcc -Wall -ansi -pedantic-errors main.c -o main

- Output:
  - Compiling the code converts it into object files (*.o)
  - Linking the code uses the information from the object code to build executable.

RUNNING YOUR CODE USING GCC

- Compile multiple files:
  1. To stop the process till compilation step :
     gcc –c code1.c code2.c code3.c

  2. To link the individual '.o' files to generate the executable :
     gcc –o executor code1.o code2.o code3.o

  The same can be done in a single step :
  gcc –o executor code1.c code2.c code3.c