Oregon State University

School of Electrical Engineering and Computer Science

CS 261 – Recitation 1
Compiling C on UNIX

Fall 2013

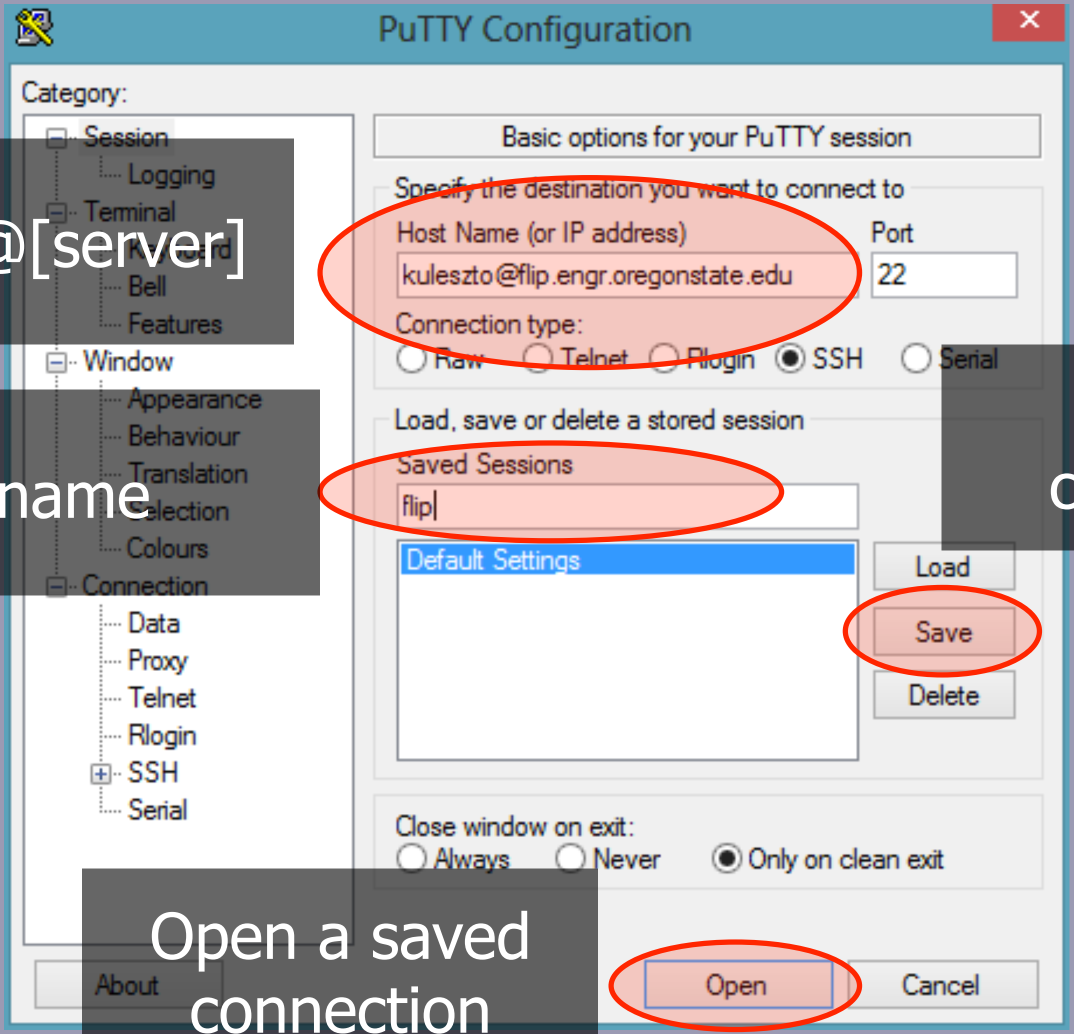
Outline

- **Secure Shell**
- Basic UNIX commands
- Editing text
- The GNU Compiler Collection (gcc)
- Setting up your IDE
- Basic debugging

Secure Shell Tools

- Mac OS X / Linux
  - Open a terminal
  - Type “ssh [username]@[server]”
    - *ssh kuleszto@flip.engr.oregonstate.edu*
  - scp [file] [username]@[server]
    - Transmit, Cyberduck, Filezilla, etc. (if you want a GUI)
- Windows
  - Putty (http://engineering.oregonstate.edu/computing/fileaccess/putty_ssh/)
  - WinSCP (http://winscp.net/)

Putty Configuration



Specify the destination you want to connect to

Host Name (or IP address): kuleszto@flip.engr.oregonstate.edu

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Saved Sessions: flip

Close window on exit: ○ Always ○ Never ● Only on clean exit

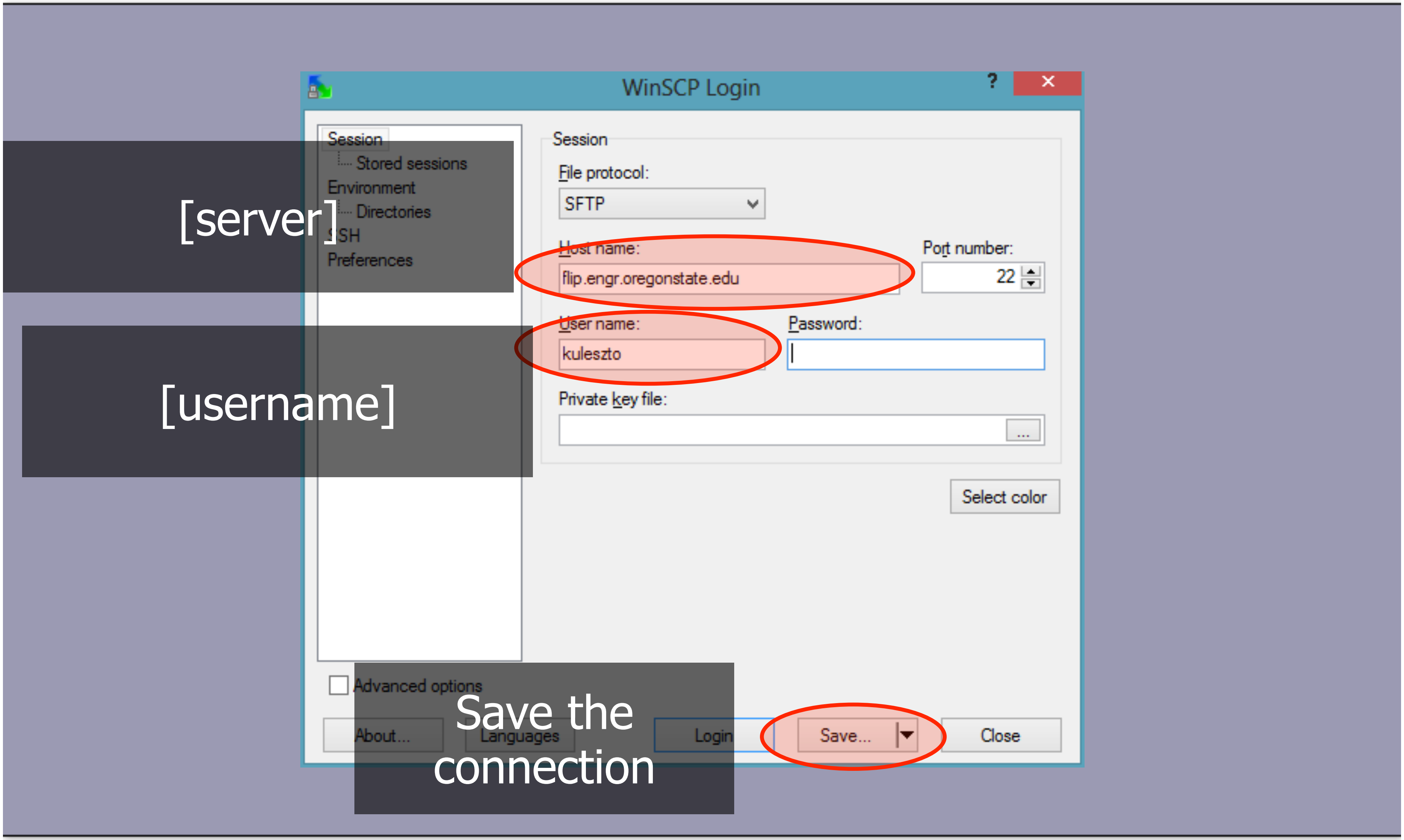
[username]@[server]

Server name

Save the connection

Open a saved connection

WinSCP Configuration



Session

File protocol: SFTP

Host name: flip.engr.oregonstate.edu

Port number: 22

User name: kuleszto

Password:

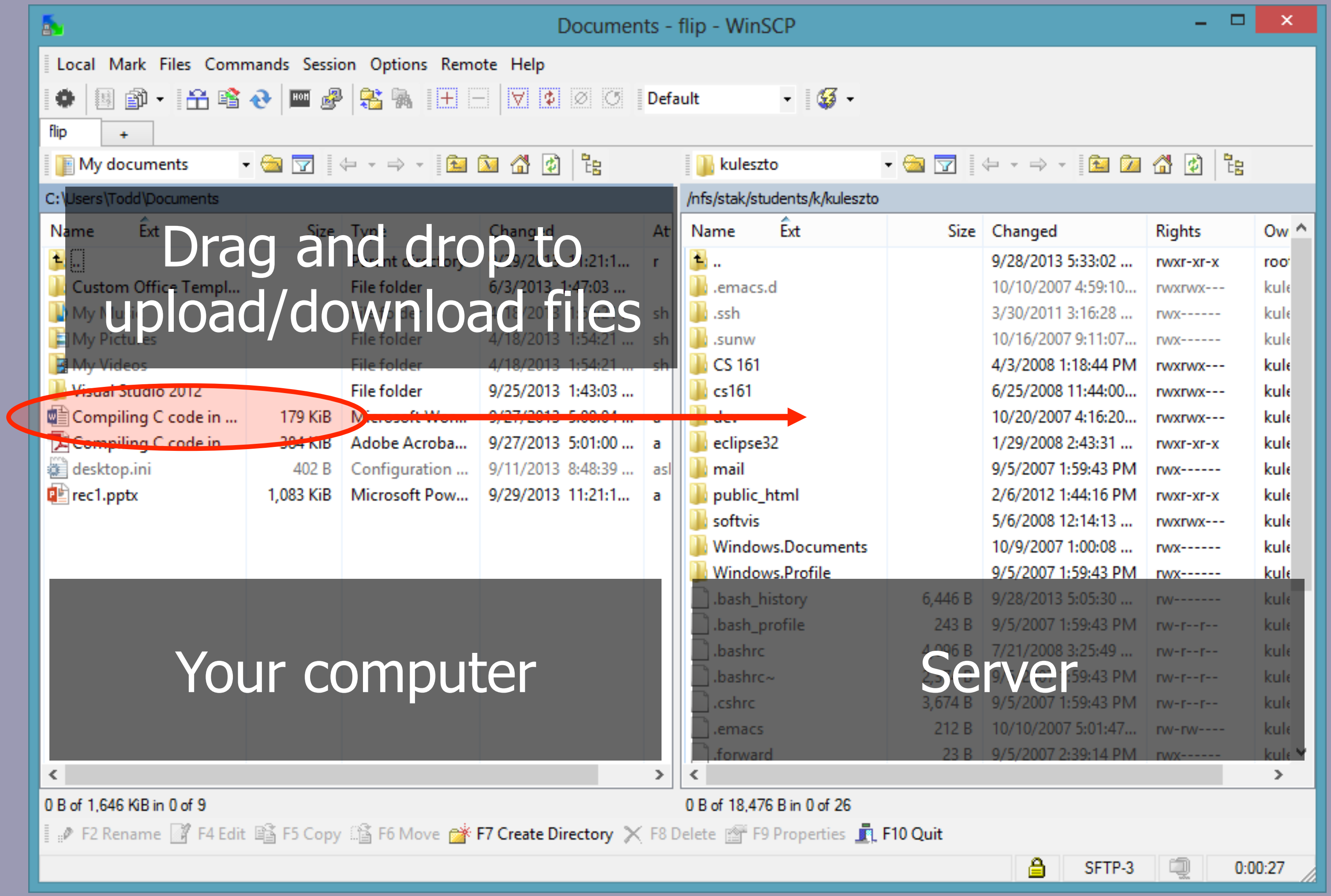
Private key file:

[server]

[username]

Save the connection

WinSCP Usage



Documents - flip - WinSCP
Local Mark Files Commands Session Options Remote Help
Default
flip
My documents
kuleszto
F2 Rename F4 Edit F5 Copy F6 Move F7 Create Directory F8 Delete F9 Properties F10 Quit
SFTP-3
0:00:27

Drag and drop to upload/download files

Your computer

Server

C:\Users\Todd\Documents

Name	Ext	Size	Type	Changed	At
..			Parent directory	9/29/2013 11:21:1...	r
Custom Office Templ...			File folder	6/3/2013 1:47:03 ...	
My Mu			File folder	2013	sh
My Pictures			File folder	4/18/2013 1:54:21 ...	sh
My Videos			File folder	4/18/2013 1:54:21 ...	sh
Visual Studio 2012			File folder	9/25/2013 1:43:03 ...	
Compiling C code in ...		179 KiB	Microsoft Wor...	9/27/2013 5:00:04 ...	a
Compiling C code in ...		384 KiB	Adobe Acroba...	9/27/2013 5:01:00 ...	a
desktop.ini		402 B	Configuration ...	9/11/2013 8:48:39 ...	ash
rec1.pptx		1,083 KiB	Microsoft Pow...	9/29/2013 11:21:1...	a

0 B of 1,646 KiB in 0 of 9

/nfs/stak/students/k/kuleszto

Name	Ext	Size	Changed	Rights	Ow
..			9/28/2013 5:33:02 ...	rwxr-xr-x	roo
.emacs.d			10/10/2007 4:59:10...	rwxrwx---	kule
.ssh			3/30/2011 3:16:28 ...	rwx------	kule
.sunw			10/16/2007 9:11:07...	rwx------	kule
CS 161			4/3/2008 1:18:44 PM	rwxrwx---	kule
cs161			6/25/2008 11:44:00...	rwxrwx---	kule
dev			10/20/2007 4:16:20...	rwxrwx---	kule
eclipse32			1/29/2008 2:43:31 ...	rwxr-xr-x	kule
mail			9/5/2007 1:59:43 PM	rwx------	kule
public_html			2/6/2012 1:44:16 PM	rwxr-xr-x	kule
softvis			5/6/2008 12:14:13 ...	rwxrwx---	kule
Windows.Documents			10/9/2007 1:00:08 ...	rwx------	kule
Windows.Profile			9/5/2007 1:59:43 PM	rwx------	kule
.bash_history		6,446 B	9/28/2013 5:05:30 ...	rw-------	kule
.bash_profile		243 B	9/5/2007 1:59:43 PM	rw-r--r--	kule
.bashrc		4,096 B	7/21/2008 3:25:49 ...	rw-r--r--	kule
.bashrc~			9/5/2007 1:59:43 PM	rw-r--r--	kule
.cshrc		3,674 B	9/5/2007 1:59:43 PM	rw-r--r--	kule
.emacs		212 B	10/10/2007 5:01:47...	rw-rw----	kule
.forward		23 B	9/5/2007 2:39:14 PM	rwx------	kule

0 B of 18,476 B in 0 of 26

UNIX Servers @ OSU

- ENGR
  - flip.oregonstate.edu (RHEL 6.4)
  - flop.oregonstate.edu (RHEL 6.4)
- ONID
  - shell.onid.oregonstate.edu (Debian 6.0)

Outline

- Secure Shell
- **Basic UNIX commands**
- Editing text
- The GNU Compiler Collection (gcc)
- Setting up your IDE
- Basic debugging

Basic UNIX Commands

Command	Description
`ls`	Lists files and folders (use “ls -l” for a “long” listing)
`cd [dir name]`	Change directory (“cd ..” will go up a directory)
`pwd`	Print name of current directory
`mkdir [dir name]`	Make a directory
`rmdir [dir name]`	Remove a directory
`cp [file] [new file]`	Copy a file (use “cp –r” to copy a directory)
`mv [file] [new file]`	Move (or rename) a file or directory
`rm [file]`	Remove a file
`cat [file]`	Show file contents
`exit`	Close connection to server
`ctrl-c`	Kill the current process

Basic UNIX Commands

- For more info on a command, use the manual page:

```
$  man ls
```

- Tutorial: http://www.ee.surrey.ac.uk/Teaching/Unix/

Outline

- Secure Shell
- Basic UNIX commands
- **Editing text**
- The GNU Compiler Collection (gcc)
- Setting up your IDE
- Basic debugging

UNIX Text Editors

- Nano
  - Easiest choice
  - Command: nano [filename]
  - *ctrl-o* to save file
  - *ctrl-x* to exit
- Emacs
  - Amazingly powerful, steep learning curve
  - Command: emacs [filename]
  - *ctrl-x ctrl-s* to save file
  - *ctrl-x ctrl-o* to exit
  - http://www2.lib.uchicago.edu/keith/tcl-course/emacs-tutorial.html

Practice

1. Use any Secure Shell Client to connect flip.engr.oregonstate.edu
2. Create a folder named CS261
3. Change working directory to CS261
4. Create a folder named Rec1
5. Create a folder named Temp
6. Remove folder Temp
7. Change working directory to Rec1
8. Using nano, emacs, or vim, create and file named info.txt and input your name.
9. Use cat to show content of info.txt
10. Copy info.txt to the CS261 folder
11. Delete both info.txt files.

Outline

- Secure Shell
- Basic UNIX commands
- Editing text
- **The GNU Compiler Collection (gcc)**
- Setting up your IDE
- Basic debugging

GCC

- GCC is the standard UNIX/Linux/Mac OS X compiler for about a dozen languages (e.g., C, C++, Objective C, Fortran, Java)
- Compiling with GCC:

```
gcc <list of options> sourcefile.c
```

e.g.:

```
gcc –Wall –std=c99 -o test test.c
```

- Compiling multiple files:

```
gcc <list of options> <list of source files>
```

e.g.:

```
gcc –Wall –std=c99 -o test test1.c test2.c test3.c
```

Makefile Example

A makefile is like a script for the compiler. We'll provide makefiles for many of your assignments so you'll be able to compile by typing 'make'.

Contents of a makefile:

```
default:main

main: main.c
  gcc -Wall -std=c99 main.c -o main
clean:
  rm main main.o
```

Executed when you type "make"

Executed when you type "make clean"

Make Tutorial

http://mrbook.org/tutorials/make/

Outline

- Secure Shell
- Basic UNIX commands
- Editing text
- The GNU Compiler Collection (gcc)
- **Setting up your IDE**
- Basic debugging

C Programming IDEs

- Mac/Linux/Windows: Eclipse CDT
- Mac only: Xcode
- Windows only: Visual Studio

Note: Eclipse CDT does not come with a C/C++ compiler and Windows does not have a built-in C/C++ compiler either (Linux and Mac usually come with GCC installed). Therefore, using Eclipse CDT on Windows requires installation of a compiler first, for example MinGW. See http://www.eclipse.org/downloads/moreinfo/c.php for more info.

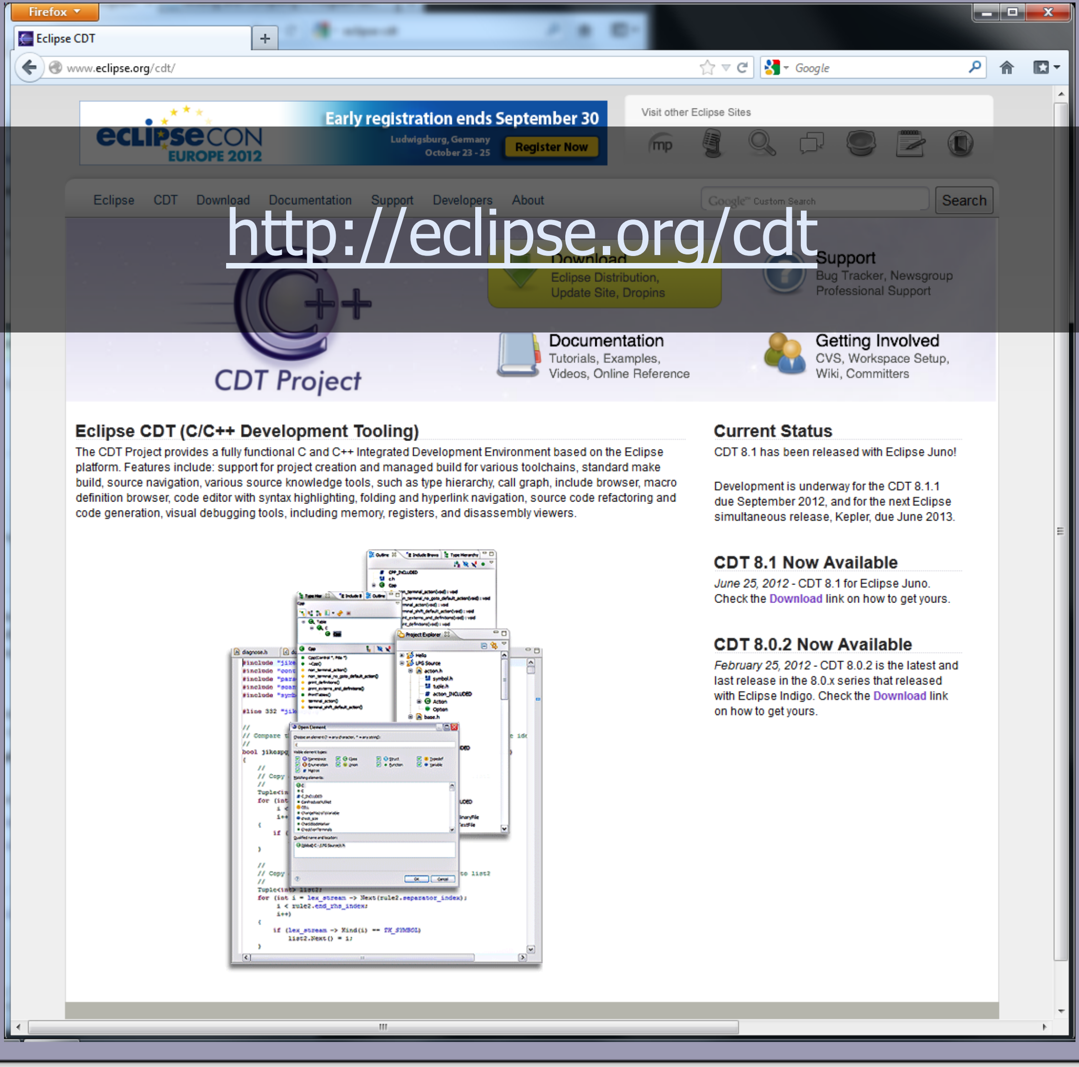
Always Test on UNIX!

You can use any IDE to develop and test your C application before submitting. However, UNIX is the environment in which the program will be graded.

Make sure your program will **compile and run without errors or warnings** using GCC on '***flip.engr.oregonstate.edu***'.

Downloading Eclipse CDT

http://eclipse.org/cdt



Eclipse CDT (C/C++ Development Tooling)

The CDT Project provides a fully functional C and C++ Integrated Development Environment based on the Eclipse platform. Features include: support for project creation and managed build for various toolchains, standard make build, source navigation, various source knowledge tools, such as type hierarchy, call graph, include browser, macro definition browser, code editor with syntax highlighting, folding and hyperlink navigation, source code refactoring and code generation, visual debugging tools, including memory, registers, and disassembly viewers.

Current Status

CDT 8.1 has been released with Eclipse Juno!

Development is underway for the CDT 8.1.1 due September 2012, and for the next Eclipse simultaneous release, Kepler, due June 2013.

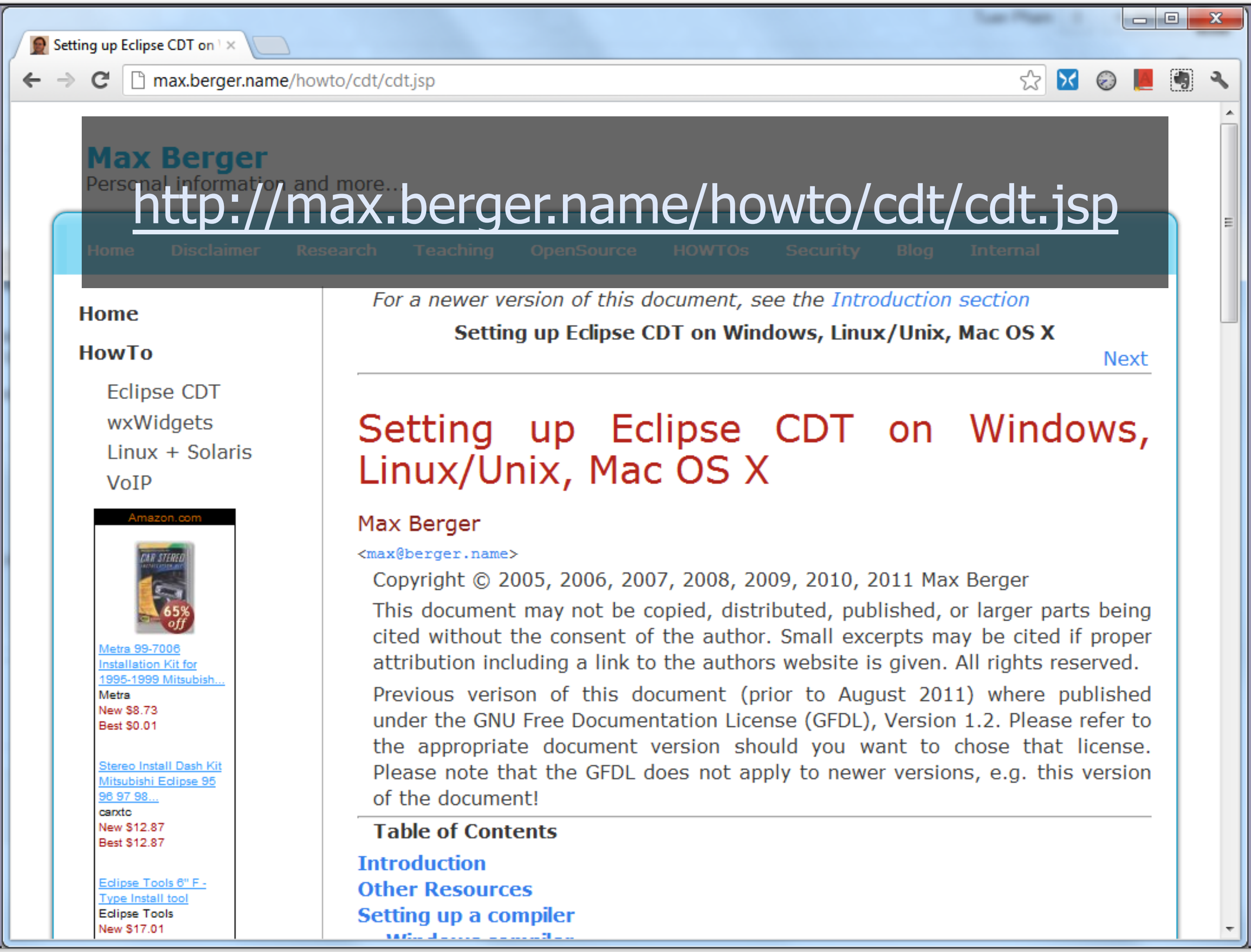
CDT 8.1 Now Available

June 25, 2012 - CDT 8.1 for Eclipse Juno. Check the Download link on how to get yours.

CDT 8.0.2 Now Available

February 25, 2012 - CDT 8.0.2 is the latest and last release in the 8.0.x series that released with Eclipse Indigo. Check the Download link on how to get yours.

Setting up Eclipse CDT



http://max.berger.name/howto/cdt/cdt.jsp

Max Berger

Personal information and more...

For a newer version of this document, see the Introduction section

Setting up Eclipse CDT on Windows, Linux/Unix, Mac OS X

Next

Setting up Eclipse CDT on Windows, Linux/Unix, Mac OS X

Max Berger

<max@berger.name>

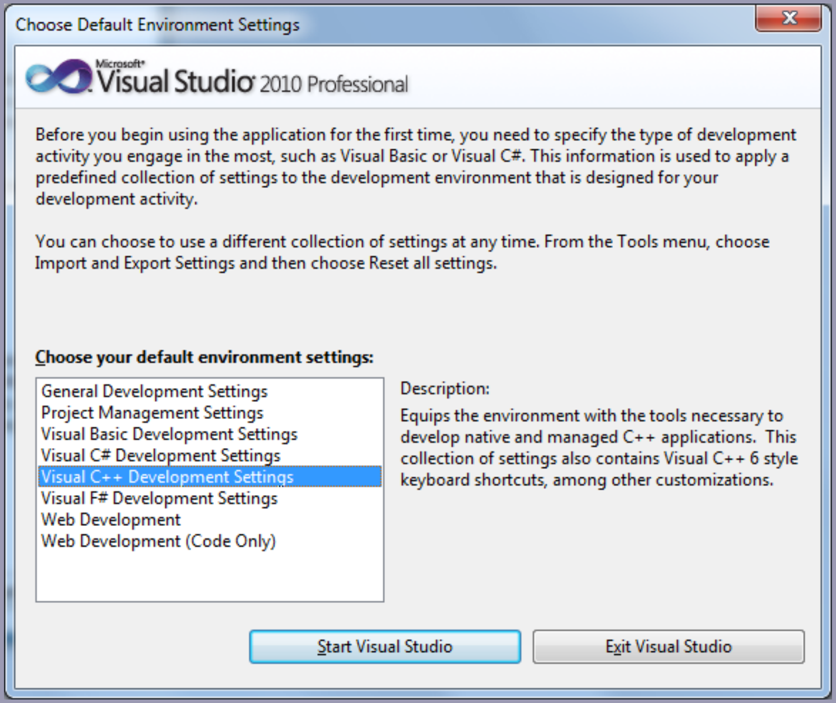
Table of Contents

Microsoft Visual Studio 2012

- Download it from https://secure.engr.oregonstate.edu:8000/teach.php?type=want_auth
  - Click on “Microsoft Dreamspark Login”
  - Visual Studio is under “Development Tools”
    - A pre-release of Visual Studio 2013 is also available. It probably works just like 2012, but we haven’t tested it and it’s beta software that may contain [more] bugs.

Setting up Visual Studio

- First time you start MSVS
  - The first time you start visual studio it will ask what environment settings to use.
  - Select "Visual C++"
  - On EECS lab machines you may wait a loooong time.



Choose Default Environment Settings

Microsoft Visual Studio 2010 Professional

Before you begin using the application for the first time, you need to specify the type of development activity you engage in the most, such as Visual Basic or Visual C#. This information is used to apply a predefined collection of settings to the development environment that is designed for your development activity.

You can choose to use a different collection of settings at any time. From the Tools menu, choose Import and Export Settings and then choose Reset all settings.

Choose your default environment settings:

- General Development Settings
- Project Management Settings
- Visual Basic Development Settings
- Visual C# Development Settings
- Visual C++ Development Settings
- Visual F# Development Settings
- Web Development
- Web Development (Code Only)

Description:

Equips the environment with the tools necessary to develop native and managed C++ applications. This collection of settings also contains Visual C++ 6 style keyboard shortcuts, among other customizations.

Setting up your IDE

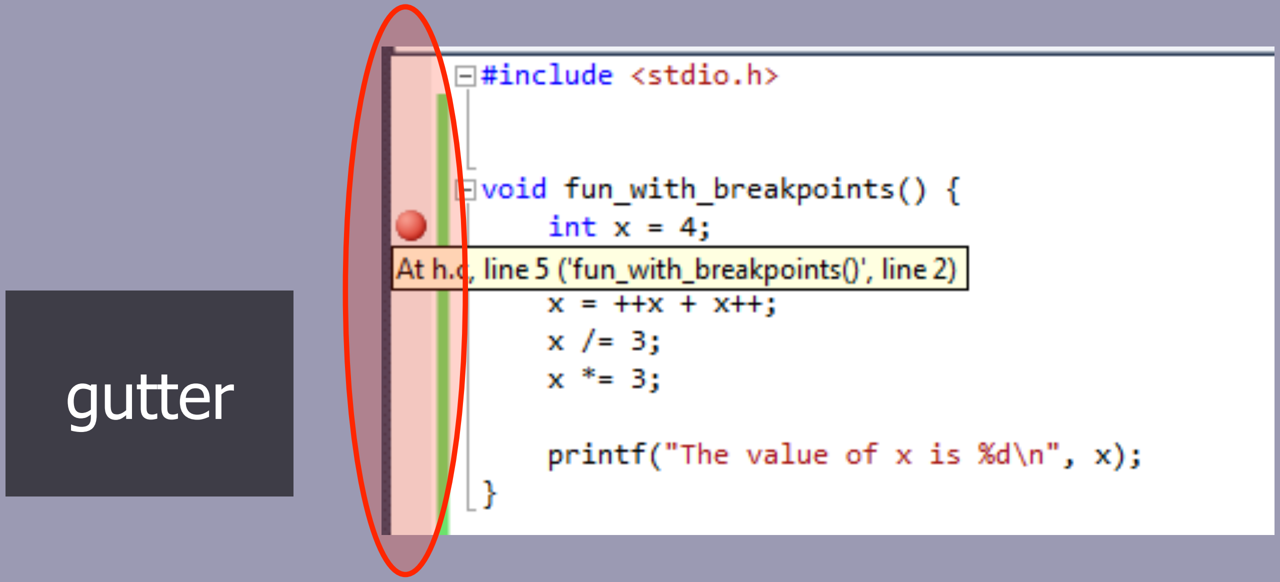
See our handouts on Visual Studio, Eclipse, and Xcode

Outline

- Secure Shell
- Basic UNIX commands
- Editing text
- The GNU Compiler Collection (gcc)
- Setting up your IDE
- **Basic debugging**

Debugging in Visual Studio

- **Break points** tell the IDE to stop execution at specific places. This lets you examine what the code is doing.
- Insert a break point by clicking in the "gutter":



```
#include <stdio.h>


void fun_with_breakpoints() {
    int x = 4;
At h.c, line 5 ('fun_with_breakpoints()', line 2)
    x = ++x + x++;
    x /= 3;
    x *= 3;

    printf("The value of x is %d\n", x);
}
```

- Eclipse and Xcode work in a similar manner.

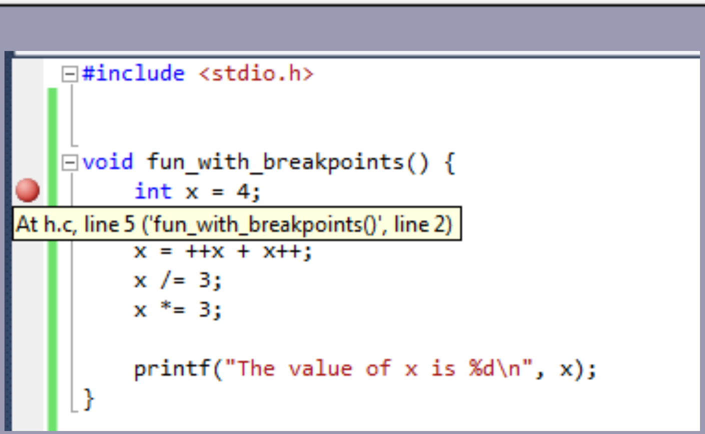
What is the value of ‘x’ after each line?

```
void fun_with_breakpoints() {
   int x = 4;
   x = ++x + x++;
   x /= 3;
   x *= 3;
   printf("The value of x is %d\n", x);
}
```

Ugh, let’s not compute this by hand.
We’ll let the computer do the heavy lifting.

Step-by-step execution

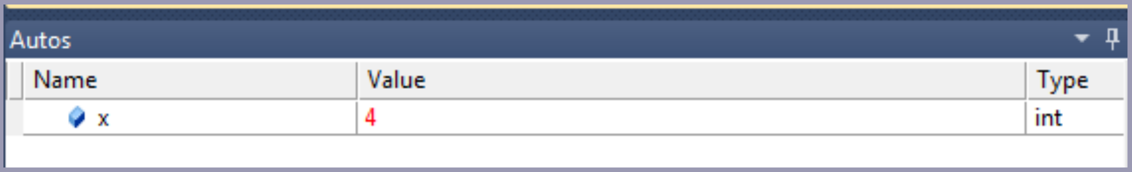
- Step 1: Set the break point



```
#include <stdio.h>


void fun_with_breakpoints() {
    int x = 4;

    x = ++x + x++;
    x /= 3;
    x *= 3;

    printf("The value of x is %d\n", x);
}
```

At h.c, line 5 ('fun_with_breakpoints()', line 2)

- Step 2: Start debugging: F5
- Step 3: Step through the program with F10 or F11.
  - The value of each variable will display at the bottom of the screen. Or, hover over a variable in the text editor to see its current value.



Autos

Name	Value	Type
x	4	int

That's all for today!