Oregon State University School of Electrical Engineering and Computer Science

CS 261 – Recitation 1



Spring 2016

Outline

- Using Secure Shell Clients
- The GCC
- CIDE : Codeblocks / Visual Studio / terminal (or Eclipse or Xcode or etc..)
- Some Examples
- Intro to C

Get your terminals

- Windows people, get ssh <u>http://sils.unc.edu/it-services/servers/using-</u> <u>ssh</u>
 - It already has a build-in file transfer client

 Mac people, you have ssh built-in and you can use a file transfer client such as <u>Cyberduck</u>.

Using Secure Shell Clients

🗐 - default - SSH Secure Shell		
Eile Edit View Window Help		
🖬 🚑 🖪 🛢 🖻 🖻 🖨 🚔 🎒 🎒 🏀 🛠 🕅		
Quick Connect Connect		
SSH Secure Shell 3.2.0 (Build 267) Copyright (c) 2000-2002 SSH Communications Security Corp - htt	cp://www.ssh.com	۵/
This copy of SSH Secure Shell is a non-commercial version. This version does not include PKI and PKCS #11 functionality.		
1		
		-
Not connected - press Enter or Space to connect	80x24	

• Open 'Terminal' & type <username>@<host name> (for Mac)

Using Secure Shell Clients

List of available servers :

- flip.engr.oregonstate.edu
- flop.engr.oregonstate.edu off-campus

And more (see http://eecs.oregonstate.edu/it/)

Connec	t to Remote Host		
	Host Name:	flip.engr.oregonstate.edu	Connect
IS	<u>U</u> ser Name:	ledu	Cancel
	Port Number:	22	
	Authentication Method:	<profile settings=""></profile>	

🕘 flip.engr.oregonstate.edu - default -	- SSH Secure Shell	
Eile Edit View Window Help		
日 🍯 🖪 📕 🍠 🖪 🕾 😁	M 🙇 🕰 🦠 🛷 N?	
👔 Quick Connect 🦳 Profiles		
This copy of SSH Secure Shell i	is a non-commercial version.	1
This version does not include H	PKI and PKCS #11 functionality.	
Last login: Thu Sep 24 16:01:49	> 2009 from zermelo.eecs.oregonstate.edu	
This system is strictly for use	e by faculty, students, and staff of	
the College of Engineeri	ing, Oregon State University.	
Unauthorized access is prob	nibited - violators will be prosecuted	
Use should be consistent	with the OSU Acceptable Use Policy	
as well as College of Er	ngineering policies and guidelines.	
Refer to http://engr.oregonst	tate.edu/computing/faqs/coe_aup/index.html	
		-
Quotas are used for home din For details	cectories, incoming email, and printing. s, check:	
http://engr.oregonstate	e.edu/computing/faqs/quotas.html	
If you have any problems with t	chis machine, please mail support@engr.orst.	.edu
		5
Terminal type? [vt100]		
Connected to flip.engr.oregonstate.edu	SSH2 - aes128-cbc - hmac-md5 - none 80x24	21

Using Secure Shell Clients

•Other Secure shell clients available :

Remote connect: PuTTY

http://www.chiark.greenend.org.uk/~sgtatham/putty/



http://winscp.net/eng/index.php

Using Linux Shell

•For Mac and Unix users :

- Open the shell terminal and enter the following ssh engid@flip.engr.oregonstate.edu \rightarrow On campus ssh engid@flop.engr.oregonstate.edu \rightarrow Off campus

padidehdanaee — ssh danaeep@flip.engr.oregonstate.edu — 70×22
 [als2077-mac4:~ padidehdanaee\$ ssh danaeep@flip.engr.oregonstate.edu

danaeep@flip.engr.oregonstate.edu's password: 👔

Basic commands

• pwd

- Present working directory
- Is
 - list files and directories in current directory
 - % ls -la : 'el ' denotes long listing
 - 'a' including all hidden files

- cd
 - Change directory
- mkdir
 - make new directory
- ср
 - copy <srcFileName> <desFileName>
- mv
 - Moves / renames <srcFileName> <desFileName>
- rm
 - remove file
- cat
 - show file content
- exit
 - exit the session

Basic commands continued..

- . (dot)
 - Represents current directory
- .. (dot dot)
 - Represents parent directory

Text Editors

• Vim

- To start : vim <*filename*>
- 3 modes for text editing:
 - Insert (i) / Replace (r / R) /Browse (Esc)
- To save :w
- To quit :q
- dd delete 1 line
- ZZ save the file and quit vi

More information : <u>http://vimdoc.sourceforge.net/htmldoc/usr_toc.html</u>

- Emacs
 - To start : emacs <*filename*>
 - Common commands:
 - Ctrl X Ctrl S: Save
 - Ctrl X Ctrl C: Exit

More information: http://lowfatlinux.com/linux-editor-emacs.html

• You can use other GUI editors like Notepad++ which have syntax highlighting and is available on ENGR servers.

Useful vi Commands

- arrow keys and h,j,k,I move cursor
- **dd** delete 1 line
- :w save file
- :q! quit even if file is not saved
- **ZZ** save the file and quit vi

The 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 -o test test.c

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

Using the GCC compiler (cont.)

- 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

Examples

- Write a program to print "Hello World".
- Compile it using "make"
- Contents of "makefile"

```
default:main
main: main.c
  gcc main.c -o main
clean:
  rm main main.o
```

NOTE

You can use any IDE(Integrated Development Environment) to develop and test your C application before submitting. However, Linux is the environment in which the program will be graded. So make sure your program will **compile and run without errors or warnings** using GCC only on *'flop.engr.oregonstate.edu'*.

Intro 2 C

- Useful websites:
 - cplusplus.com
 - <u>http://www.cprogramming.com/</u>

Headers in C

• Essential header:

#include <stdio.h> :Includes the standard Input/output library. Without this statement the program will not be able to print/read data.

Other headers & including any files : (Next class..)

Variable/function declaration

All the variables/functions are required to be declared prior to its use in the ٠ program.

Eg.

int add (int , int); // function declaration /prototyping void main(){

```
int var1 = 10;
   printf("%d", var1);
                                  Should be placed before void main(){ ..}
   int var2=20;
   printf("%d", var2);
   int result = add(var1, var2);
int add(int a, int b){
   return (a+b);
```

}

}

Pointers

- <u>A tutorial on Pointers & Arrays in C</u>
- Example:

int var1=10;

int *pointertovar1;

pointertovar1=&var1; // & is called as 'ampersand'. It means 'address of'

*pointertovar1=20; // * denotes the 'thing pointed by'

printf("%d%", var1); //Now the value of var1 becomes 20.

Memory allocation and structures

- Memory has to be managed manually due to absence of 'Garbage collector'.
- Syntax :

datatype * varname = (datatype *) malloc (sizeof(datatype));
eg. struct record *Rec = (struct record *)malloc(sizeof(struct record))

- Structures are similar to classes.
 - struct record{

```
char name[20];
```

```
int id;
float GPA;
```

- }
- A structure stores only variables but no functions.
- Details of both these topics in the next session..

That's all for today. Please remember the earlier **NOTE**.