Lab for C Programming Basics

Ritu Arora

Texas Advanced Computing Center

March 19th, 2012

TACC

THE UNIVERSITY OF TEXAS AT AUSTIN

Introduction

- You will learn
  - How to write C code
  - How to compile and execute C code
- What will you do
  - Modify the code for the exercises to embed logic in it
  - Compile and execute the code for the programs discussed in the lecture and exercises
- Assumption
  - You have access to or have already installed the C compiler

Download Sample Code and Exercises

- Download the compressed files from the following link
  - `C_Training_March_2012.zip` for Windows or
  - `C_Training_March_2012.tgz` for Linux and OS/X) :

  http://www.tacc.utexas.edu/user-services/training/course-materials

- Save the file in the directory/folder of your choice and unzip/untar it
  - On Windows: for unzipping, right-click on the file, click on “Extract All”, and follow the instructions in the wizard
  - On Linux or OS/X:

    ```
    tar -xvzf C_Training_March_2012.tgz
    ```

Take a Peek at the Downloaded Code: Windows Users

- Bring up the command prompt – Windows 7 users type “cmd” in the text box that says “Search Programs and Files”

- Change directory to your unzipped folder, for example:

```
C:\Users\ritu> cd C:\Users\ritu\Desktop\trainingC
```

- Open any code file, for example, `mathExample.c` with an editor of your choice (e.g., Notepad or Textpad). You can also type “`edit mathExample.c`” at the command prompt to open the file:

```
C:\Users\ritu\Desktop\trainingC>edit mathExample.c
```

TACC

THE UNIVERSITY OF TEXAS AT AUSTIN

Take a Peek at the Downloaded Code: Linux Users

- Change directory to where you extracted the contents of the tar file:

```
login4$ cd trainingC
```

- Open any code file, for example, `mathExample.c` with an editor of your choice (e.g., vi and nano):

```
login4$ vi mathExample.c
```

Take a Peek at the Downloaded Code: OS/X Users

- You can either open/edit the file using Xcode or via the Terminal

- If you want to use the terminal/console
  - You can bring up a terminal and change directory to where you extracted the contents of the tar file
  - Open any code file, for example, `mathExample.c` with an editor of your choice

- If you want to use Xcode, please follow the instructions at the following link:

http://dyba.wordpress.com/2008/04/13/developing-c-programs-on-mac-os-x-using-xcode-30/

Exercise 1: mathExample.c

- Objective: Learn to compile and link with the gcc compiler
- Change directory to "`Exercise`" in the parent directory "`trainingC`"
- To Compile: **`gcc -o myMathEx2 mathExample.c`**

  If you are on TACC resources and see error messages like ,

```
login4$ gcc -o mathExample mathExample.c
/tmp/cczLt3nd.o: In function `main':
mathExample.c:(.text+0x2b): undefined reference to `sqrt'
mathExample.c:(.text+0x74): undefined reference to `pow'
collect2: ld returned 1 exit status
```

  then try: **`gcc -o myMathEx2 mathExample.c -lm`**

To explicitly load the Math ("m") library with the gcc compiler.

- To Run at the command prompt type: **`myMathEx2`**

TACC

THE UNIVERSITY OF TEXAS AT AUSTIN

Exercise 2: celToFar.c

- Modify the program for converting the temperature in Fahrenheit to Celsius
  - Formula for converting Fahrenheit to Celsius: C = (F – 32)*5/9
- Follow the instructions in the comments of CelToFar.c
- Compile and run the code:

```
gcc –o celtoFar celToFar.c
./celToFar
```

- Now add the code for converting Celsius to Fahrenheit
  - Formula for converting Celsius to Fahrenheit: F = (C*9)/5 + 32

Exercise 3: circle.c

- Calculate the circumference and area of the circle
  - Circumference of a circle: 2*PI*R
  - Area of a circle: PI*R*R

    PI = 3.14159265

    R = radius of the circle

- Modify the code in circle.c

- Save the file, compile and run

```
gcc -o circle circle.c
./circle
```

Exercise 4: prime.c

- Write a program to find all the prime numbers between 1 and N, N included

- Modify the code in prime.c
  - Read N from input
  - Write the code for determining the prime number in function named **`prime`**
  - Note that the function is returning an integer value
- Save, compile and run the code

```
gcc –o prime prime.c
./prime
```

Exercises 5 and 6

- If time permits modify the code in files `readInput2.c` and `readInput3.c`
- Desired modifications are described in the comments
- These exercises will give you some practice on I/O related functions