CS 332/532 – 1G- Systems Programming Lab 1

Objectives

1. Getting Started with the Linux Operating System (login, execute simple commands, edit file, transfer files).
2. Write a simple "Hello World" program in C, compile and execute the program.

Lab Assignment #1

Write a C program to check whether a given number is an odd or even number by performing following steps:

1. Define an integer variable as *given_number*.
2. Use the C function *scanf* to read the integer variable *given_number*.
3. Use the conditional statement to find out if *given_number* is an odd or even number.
4. Print the final output as "*The number is an odd number"* or "*The number is an even number"*.

Compile and test the program and upload the C source code in the lab assignment submission section. Examples have been provided in .c and .java for a similar Palindrome example.

Assignment Submission

The assignment submission in Canvas is required. No late submissions will be accepted.

Submission Checklist:

- Upload the C source file (.c file) to Canvas as part of this lab submission. Submissions through the Canvas "Comments" will not be accepted.
- Upload a README.md file which should include:
    - Instructions on how to compile your C source file into an executable.
    - How to run the executable program.
    - Any citation documentation.
    - A link to your GitHub repository.

Please do not upload executables or object files. Independent Completion Forms are not required for labs.

Lab Workbook

Opening a Terminal

Although Linux has a graphical user interface (GUI) and works like most other operating systems such as Windows and OS X, we will focus on a command-line interface running a bash shell (also known as a terminal).

A terminal is a text-based interface where you can enter commands and get feedback as text.

There are several similarities and differences between various operating systems to open a terminal:

- Mac: If you are a Mac user then you can go to Applications -> Utilities -> Terminal.
- Linux: If you are a Linux user then you will find it in Applications -> System or Applications -> Utilities. (Or 'right-click' on the desktop and click 'Open in terminal').
- Windows: Windows' built-in terminal program, PowerShell, does not include a native SSH client. To complete this lab and future assignments, you will need to remotely connect to another machine (the Vulcan service). We recommend you use Putty which is free and easy to use. Note that Putty is already installed on all CS Windows systems. There are additional alternatives such as "Remote-SSH" in VSCode or Posh-SSH module for PowerShell.

Remote Server Information

Host Name: moat.cs.uab.edu
Port: 22
Username: <Your BlazerID>
Password: <Your BlazerID password>

How to Install Putty (For Windows Users on non-CS systems)

- Go to https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html and click on the installer for 32-bit or for 64-bit, depending on your system.
- Save and run the file.
- Click "next" through the wizard and install Putty.
- Click "finish" and run Putty.
- You will see several categories and subcategories. For now, we will be using "Session".
- Go to the Host Name box and type in "moat.cs.uab.edu" and make sure the port is set to 22 and connection type is SSH.
- Click "open" and log in using your BlazerID. When prompted for your password, keep in mind that Linux does not show the number of characters you type for security reasons.

- You can also give a name to this connection and save it so that you do not have to type in "moat.cs.uab.edu" every time when using Putty. You can also customize the font size, windows size, etc. and save these settings.

Mac and Linux users

Since you already have an SSH-capable terminal you do not need to install an SSH client. Open the terminal and type "ssh *blazerid*@moat.cs.uab.edu" (make sure you replace *blazerid* with your actual BlazerID). You will be prompted to enter your password; enter the password and you will get a command prompt.

Basic Linux Commands

Let's start with basic commands to use in the terminal.

Command	Description
pwd	Displays the full pathname of the current directory (present working directory); the character / is used to separate directories.
mkdir CS322	Make a new directory (folder) *CS332* inside the current directory.
~	The home directory; for example, a home directory could be */home/UAB/blazerid* Therefore, *~/CS332* is equivalent to */home/UAB/blazerid/CS332*
cd	Change to the home directory (equivalent to *cd ~*)
cd CS332	Change to the directory *CS332*; you can also provide a full pathname as in *cd /home/UAB/blazerid/CS332*
cd ..	Move one level up ('.' is the present directory and '..' is the parent directory)

ls	List the contents of the current directory (lowercase L – s)
ls CS332	List the contents of the directory *CS332*
cp file1 file2	Copy the file 'file1' to the file 'file2'. If 'file2' already exists, this overwrites the present contents. 'file1' still exists.
cp file1 anyDirectory	Copy the file 'file1' into the directory 'anyDirectory'.
mv file1 file2	Rename the file 'file1' to 'file2'. Move it to a new file; 'file1' does not exist now.
rm file1	Remove the file 'file1' from the present directory. The file is lost forever, so be careful with the 'rm' command.
rmdir anyDirectory	Remove the directory 'anyDirectory' (which must be empty).
ls –l	Long listing; list more information about folders and files.
ls *.jpeg	List only the files with an extension of .jpeg.
rm	Delete files.
rmdir	Delete folder.
man	The manual command is used to show the manual of the input command.
touch	Make a new empty file.
locate	Find a file in the Linux OS.
clear	Clear the screen.

NOTE: Linux is case-sensitive.

The following resources may be useful to learn more Linux commands:

- http://linuxcommand.org
- https://www.tutorialspoint.com/unix/index.htm
- http://linuxcommand.org/tlcl.php

Editing text-based documents in Linux

As the terminal is text-based, editing documents in a terminal could be challenging. There are several well-known text editors in a Linux environment such as vi, emacs, nano, etc. We will use nano in this lab to create a new file, type some text, and save the file.

- Enter "nano file1" at the command prompt.
- Enter the following text (you can use the arrow keys to move backward/forward/up/down and backspace to delete a character):

Student Conduct Code

The Student Conduct Code promotes honesty, integrity, accountability, rights, and responsibilities expected of students consisted with the core missions of the University of Alabama at Birmingham. This Code describes the standards or behavior for all students, and outlines student's rights, responsibilities, and the campus processes for adjudicating alleged violations.

- Save the file by entering "Control-O". You will be prompted with "File Name to Write: file1", hit Enter.
- Exit the editor by entering "Control-X".
- You can view the contents of the file using one of the following commands: more, less, cat. What is the difference between the three commands?
- You can use the editor to make changes to the file you just created or create a new file like we just did.

Transferring files between your local computer and CS Linux Systems

Mac and Linux uses can use the 'scp' command to copy files from your local computer to the CS Linux systems using the following example:

```
scp mylocalfile.txt blazerid@moat.cs.uab.edu:CS332/Lab1
```

(replace *blazerid* with your actual BlazerID and execute this on your local computer)

This will copy the file mylocalfile.txt in the current directory of your local computer to the CS332/Lab1 directory on the CS Linux system. If you like to copy a file from the CS Linux system to your local computer, you must execute the following command on your local computer:

```
scp blazerid@moat.cs.uab.edu:CS332/Lab1/new_file1 .
```

(replace *blazerid* with your actual BlazerID and execute this on your local computer, note the space after *new_file1* and before '.')

This will copy the file new_file1 in the directory CS332/Lab1 on the CS Linux system to the current directory of your local computer.

Windows users will need an FTP/SCP client to transfer files. There are several FTP/SCP clients available for windows, you can use either WinSCP or FileZilla. WinSCP is already installed on CS Windows systems. It has a simple drag and drop interface that looks like Windows File Explorer.

Windows Subsystem for Linux (WSL)

If you have a Windows system, you can install Linux within your Windows environment using the instructions provided in this link - **https://learn.microsoft.com/en-us/windows/wsl/install**

Some modern versions of Windows already have some of this enabled by default.

Getting Started with GitHub and GitHub codespace

We will use and encourage your use of GitHub and its service GitHub codespace as another option for compiling and running your C code in this class.

1. Create a free Github.com account if you have not done it before.
2. Create a New Private repository called "CS332-532".
3. Create a new blank file to initialize the repository and "commit".
4. Create a codespace using that repository.
5. When the terminal becomes available in the bottom half of the screen, test it out by creating a new folder called "labs" to store your lab assignments (hint: mkdir labs).
6. For the labs, you can write your C program inside this folder. Save a test file with a ".c" extension.
7. Committing and pushing your changes: Once you have made a few changes, you can use the integrated terminal or the source view to publish your work to your new repository.
    a. In the Activity Bar on the left of codespaces, click the **Source Control** view. (Mouse over the icons to see the names if you are unsure.)
    b. To stage your changes, click next to the file or next to Changes if you have changed multiple files and you want to stage them all.
    c. To commit your staged changes, type a commit message describing the change you have made, then click **Commit**.

d. Click **Publish Branch** or **Sync Changes**. Note: If you have multiple repositories, you may see a "Repository Name" dropdown, type a name for your new repository, then select **Publish to GitHub private repository.**
e. The owner of the new repository will be the GitHub account with which you created the codespace.
f. In the pop-up that appears in the lower right corner of the editor, click **Open on GitHub** to view the new repository on GitHub.com. In the new repository, view the file you made changes to and check that the change you made in your codespace has been successfully pushed to the repository.

- When you are done with your codespace, you should delete it (not your source code in your repo). Free GitHub users only receive a certain number of hours of codespace time. Use it wisely.
- Take this opportunity to add your TA's GitHub username (in the Syllabus) as a contributor on your repository for this class.

Writing a Simple C Program Example

Open an editor and type in the following C program and save it as hello.c

```
#include <stdio.h>

int main(int argc, char** argv) {
   printf("Hello World!\n");
   return 0;
}
```

Note: If you have compiling errors when copying and pasting code snippets or examples, type them out instead as some Unicode characters can paste differently in different browsers and ssh applications.

Compiling and Executing a C Program Example

To compile use: `gcc –o hello hello.c`

To execute use: `./hello`

README Files

README files communicate to the coding community important information about your code or project. It includes the "who, what, when, where, and why". Your README files for this course should include the following items:

- Instructions on how to compile your C source file into an executable.
- How to run the executable program.
- Any citation documentation.
- A link to your GitHub repository that's been shared with your TA's GitHub username.

An example README file is provided, example_README.md.

VSCode

VSCode is a free code editor. It works on multiple platforms (Windows and Mac) and can support multiple coding languages. VSCode offers plug-ins to compile and run the GCC compiler and to pull and commit code to GitHub repositories.

It's highly suggested that you leverage this code editor for this class for consistency and easy to troubleshoot.

Getting Started - https://code.visualstudio.com/docs

C++/C How-To's - https://code.visualstudio.com/docs/cpp/introvideos-cpp

Windows - https://code.visualstudio.com/docs/cpp/config-mingw

Mac - https://code.visualstudio.com/docs/cpp/config-clang-mac

Working with GitHub in VS Code - https://code.visualstudio.com/docs/sourcecontrol/github

C Language Resources

The classic C reference book is:

C Programming Language 2nd Edition by Brian W. Kernighan and Dennis Ritchie. Prentice Hall. 1998.
Here are a few websites you can use to learn C (please be careful about clicking on ads):

- https://en.cppreference.com/w/c
- https://www.tutorialspoint.com/cprogramming/c_overview.htm
- https://www.learn-c.org

Lab Assignment #1 is on page 1.