

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

Opening a Terminal

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

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

Opening a terminal is not so hard. There are several similarities and differences between various operating systems:

- If you're a Mac user then you can go to Applications -> Utilities -> Terminal.
- If you're a Linux user then you will find it in Applications -> System or Applications -> Utilities. (Or, just 'right-click' on the desktop and click 'Open in terminal').
- If you're a Windows user and want to remotely connect to another machine then you should get an SSH client. We recommend you use Putty which is free and easy to use. Note that Putty is already installed on all CS Windows systems.

Using GitHub Codespaces

- Go to github.com
- In the top right corner, there are two options:
 - Sign in or Sign up if you need to create one
 - ** As a student you can get GitHub pro regardless of a new or old account; Visit education.github.com/pack
- Click the "+" icon in the top-right corner and select "New repository."
- Name your repository (name does not really matter but please make it intuitive like <blazerid> CS332/532)
 - Readme is not required for the overall repository
- Repository page should pop up automatically
 - If not click on your profile in the top right corner
 - Go to "My Repositories"
- Click "Create a Codespace" in the main page of your repository

- Click "Create a new codespace" on the new page
- Wait for Codespace initialization
- If there is a README created for you:
 - Make a small change to the document (the change you make does not matter)
- If your codespace is empty, create a text file just so that you can do your initial commit
 - Make sure you add something to the file so the commit will go through
- Once you have a file to commit, follow these steps in the terminal:
 - Open the terminal if not open already
 - 3 lines in the top left corner has an option for the terminal
 - Create a new one
 - Once open:
 - Type the following commands:
 - "git add <file name>"
 - "git commit -m <some meaningful message>"
 - "git push"
- Go back to your repository home page in a new browser tab
- Double check to see that the file you pushed is now in the repository
- **Once you are done with your codespace, you need to stop your codespace**
 - In the bottom left corner of the screen, you should see a button that states: "Codespaces: <randomly generated name>"
 - Click this button and click stop codespace in the pop up menu
 - This can also be done from the main repository page by:
 - Click the green "Code" button
 - Click the codespaces tab
 - Click the three dots on your codespace
 - Stop the codespace

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 either 32-bit or 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 don't have to type in moat.cs.uab.edu everything to use Putty. You can also customize the font size, windows size, etc. and save these settings.

Mac and Linux users

Since you already have the terminal you don't 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 very basic commands to use in the terminal.

pwd : displays the full pathname of the current directory (present working directory) ; the character / is used to separate directories

mkdir CS332 : make a new directory (folder) CS332 inside the current directory;

~ the home directory; for example, my home directory is */home/UAB/unan* therefore, *~/CS332* is equivalent to */home/UAB/unan/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/unan/CS332*

cd .. move one level up (. the present directory and .. the parent directory)

ls : list the contents of the current directory

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)

mv file1 anyDirectory: move the file *file1* to the directory *anyDirectory*; the directory may be specified by a full pathname or relative path

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 inputted command.

For example; man cd

touch: make a new empty file

locate: find a file in the Linux OS

clear: Clear the screen

NOTE: Linux is case-sensitive

Lab Exercise

- Open a terminal
- Connect to "moat.cs.uab.edu"
- Clear the screen
- Create a directory called CS332
- Change to the CS332 directory
- Create a new directory inside CS332 and name it Lab1
- Change to the Lab1 directory
- Find the current directory using the command "pwd"
- Create a new file using the touch command by entering "touch file1"
- List the contents of the Lab1 directory
- Rename file to new_file1
- List the contents of the Lab1 directory
- Change to the CS332 directory
- Try to delete the directory Lab1. Do you get an error? What is the error message?
- Delete the file new_file1 in the directory Lab1
- Try to delete the directory Lab1. Do you still get an error?
- Use the "man" command to find out more about the touch command

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 users 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.cis.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 have to execute the following command on your local computer:

scp *blazerid*@moat.cis.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 a 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 the 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 document - **Windows Subsystem for Linux Installation Guide for Windows.pdf**

If you are using WSL and like to copy files between the WSL and the CS Linux systems follow the instructions here - **SCP commands for File Transfer.pdf**

Writing a Simple C Program

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;
}
```

Compiling and executing a C Program

To compile use: gcc -o hello hello.c

To execute use: ./hello

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

Write a C program to check whether given number is prime or not 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 *given_number* is prime or not prime.
4. Print the final output as *The number is prime* or *The number is not prime*.

Compile and test the program on CS Linux systems and upload the C source code in the lab assignment submission section. You can use the sample Java and C programs provided as examples: **Palindrome.java** and **palindrome.c**

Submission

Upload the C source file (.c file) to Canvas as part of this lab submission. Make sure to test the program on the CS Linux systems and include instructions to compile and run the program in the comments section of your program. Please do not upload executables.