UNIVERSITY OF ARKANSAS

Linux Tutorial #1

Introduction

The Linux operating system is now over 20 years old, and is widely used in industry and universities because it is fast, flexible and free. Because Linux is open source, millions of researchers and software developers are continuously updating and extending the operating system and creating new applications. One major area of development has been in creating tools to create software, in particular text editors, program compilers, and debugging tools. In this lab, we will use a number of Linux applications to create, compile and run a simple C++ program.

Login to a remote Linux machine

In this class, we will be using "turing" our department's main Linux machine for all program development. This will give you a chance to learn more about Linux without having to go through the work to install Linux on your local machine. It will also guarantee that all of your programs are stored in a central location that is backed up daily and can be accessed from anywhere.

To login to turing, you need to do the following:

- Open a "terminal" application on your Mac (the black square icon).
- You should see a command prompt at the top of the window.
- Type in the command "ssh user_name@turing.csce.uark.edu" where you replace "user_name" with your UofA user name.
- The computer should prompt you for your UofA password.
- Congratulations, you are now logged into a Linux machine and ready to start programming.

If have a Mac laptop, you can use the "terminal" application to SSH into turing from home or anywhere with a good internet connection. If you have a PC, you will need to install a SSH client program like "PuTTY" or "SSH Cryptonaut" from www.ssh.com and use this application to SSH into turing.

Using nano to create and edit C++ programs

A wide range of text editors has been written for Linux over the years. The earliest editors "vi" and "emacs" were written so long ago that the mouse had not been invented yet. These editors are fine for Linux power users, but to start off we will be using the "nano" program because it has a simple user interface.

To create a new program, type “nano hello.cpp” at the command prompt and hit return. This will create an empty document. Type in the following C++ program exactly as shown (or use copy/paste):

```
// Include statements
#include < iostream >
using namespace std;

// Main function
int main()
{
  // Print message
  cout << "Hello, world!\n";
  return 0 ;
}
```

At the bottom of your terminal window, you should see a menu of nano commands. To save the file you need to use ^O (which stands for control O). You will be prompted for the “File Name to Write:”, and if you just hit return the file will be saved in hello.cpp. Now type ^X (control X) to exit the nano program. Congratulations, you have created and saved your first program.

Using g++ to compile C++ programs

Your next task is to compile your C++ program. This will translate the C++ commands you typed into the "hello.cpp" file into machine language that can be executed on the computer. The "g++" compiler the widely used C++ compiler for Linux. It was developed as part of the GNU compiler collection developed by Richard Stallman, one of the pioneers of the open software movement.

Before we can compile our program, we must open up a Linux terminal window. To do this, go to the "Applications" menu and select "Terminal" from the pull down menu. This will open up a window with a command prompt. Next we must navigate to the Documents folder. To do this, type "cd Documents" and then type "ls". This will show you a list of all files in the Documents folder.

To compile your program, type "g++ -Wall hello.cpp -o hello.exe". There are three parts to this command. The "-Wall" part of the command tells g++ to print out all possible warning messages. The "hello.cpp" part tells g++ the name of the program to compile. The "-o hello.exe" tells g++ the name of the output file. If you leave the third part off, the default output file name is "a.out".

If you have no errors, you will just get the Linux prompt again. Otherwise g++ will print out a list of error messages. For example, if you get the following messages

```
hello.cpp:8:22: error: iostream : No such file or directory
hello.cpp: In function 'int main()':
hello.cpp:15: error: 'cout' was not declared in this scope
```

it is because there should NOT be spaces before and after the word "iostream" in the program above. To correct this, run “nano hello.cpp” again, and modify the program to remove the extra spaces and save the file again. Now type "g++ -Wall hello.cpp -o hello.exe" to recompile your program. The error messages should be gone now.

Run and extend the C++ program

To run the program you just compiled type in "./hello.exe". You should see "Hello, world!" and then the Linux prompt again. Congratulations, you just ran your first C++ program!

Before we can extend this program lets look at the code in more detail. At the top of the file, you can see the include statements. This is where we tell the C++ compiler what built-in commands we plan to use in the program. In this case, the "iostream" library contains the "cout <<" command for printing data to the screen and the "cin >>" command for reading data from the keyboard and storing it in variables.

Edit your program using nano again and add the following three lines to your program just before the "return 0;" line:

```
int number;
cin >> number;
cout << number << endl;
```

The first line declares an integer variable called "number". The second line stops the program and waits for the user to type in an integer on the keyboard, and then stores this value in the variable number. The third line prints the value of the number variable out on the screen followed by a carriage return.

Now type "g++ -Wall hello.cpp -o hello.exe" to compile your modified program. To run your new program type in "./hello.exe". This time you should see "Hello, world!" and but no Linux prompt because the program is waiting for input. Now type in the integer "42" and hit enter.

Your program should now print this value and end. Run your program several more times and type in some different integer values. Your program should simply print out the values you type in. What happens if you type in something other than an integer?

Logout of Linux

It is always a good idea to logout of ALL machines you are using when you have completed your task (or if you leave the room). To logout of turing, simply type the “logout” command. Then you can close your “terminal” application. When you SSH onto turing again next time, you will see all of the files you created today.

Summary

Command	**Meaning**
ssh	start using Linux machine
nano	program for editing C++ programs
terminal	program that lets you execute Linux commands
cd	program to change directories
ls	program to list all files in a directory
g++	program to compile C++ programs
./program.exe	how we execute a C++ program
logout	finish using Linux machine

Author: jgauch@uark.edu, January 2015