

Lab Assignment 0

Jack Fenton

fenton.j@husky.neu.edu

Poorva Jitendra Patil

patil.ji@husky.neu.edu

Submit date: 1/16/2020

Due Date: 1/16/2020

Abstract

This laboratory experiments served as an introduction to the embedded systems that will be used in the course. The experiment was broken up into three parts, each giving a step by step introduction to the embedded system: Zedboard and MobaXterm, the computing device and shell; VIM, the editor; and C++, the programming language.

Introduction

This first lab session is devised to enable students to get familiarised with the lab equipment used throughout the semester. The ZedBoard, an embedded computing device based on an ARM processor and Linux operating system, was chosen as the platform to be built upon in the course. The desktop PCs serve as peripherals to the ZedBoard, allowing students to access them through a program called MobaXterm using a secure shell, SSH. This shell can be used to create directories and files on the Zedboard. Files are written in C++ using an editor called Vim that works inside the shell. The shell also has the ability to make and run executable files from the C++ source files. Anything written in the shell can be copied from the ZedBoard to the desktop PC using an SFTP shell. In this lab, a directory was created containing two C++ files and an executable file for each. The first program was a “hello world” program. The second sorted a random list of numbers in ascending order.

Lab Discussion

Lab 0.0

The ZedBoard was plugged into a power source and connected to the computer using an Ethernet cable. Using the MobaXterm program, a secure shell(SSH) connection was made between the host computer and the ZedBoard. The ZedBoards were configured to be accessible at the local IP address, 192.168.1.10, with a default SSH port of 22. Once connected and logged in, the user’s initial working directory was `home/user###`.

Lab 0.1

A simple hello world program was written in C++ on the ARM processor of the ZedBoard. In a new directory `lab1`, The Vim editor was used to create, edit, and save the program `hello.cpp`. The completed file gets compiled to generate an executable file, `hello.out`.

Lab 0.2

As a more complex assignment, students are required to write a C++ program that generates an array of 10 random integer numbers between 0 and 99, prints the original array; sorts the array in ascending order, and prints the sorted array. In the same directory `lab1`, the file `array.cpp` has been edited to perform the array sorting. Then the file is compiled to generate an executable.

All commands were run using UNIX commands.

Results and Analysis

Lab 0.1


First, the students were required to create a new directory using the command

```
mkdir lab1
```

and then changing the directory to lab1 to start working on C++ programs

```
cd lab1
```

hello.cpp was created and edited with the vim editor.



```
user217@localhost:~/lab1$ vim hello.cpp
user217@localhost:~/lab1$ g++ hello.cpp -o hello.out
user217@localhost:~/lab1$ ./hello.out
hello world
user217@localhost:~/lab1$ ls
hello.cpp  hello.out
user217@localhost:~/lab1$
```

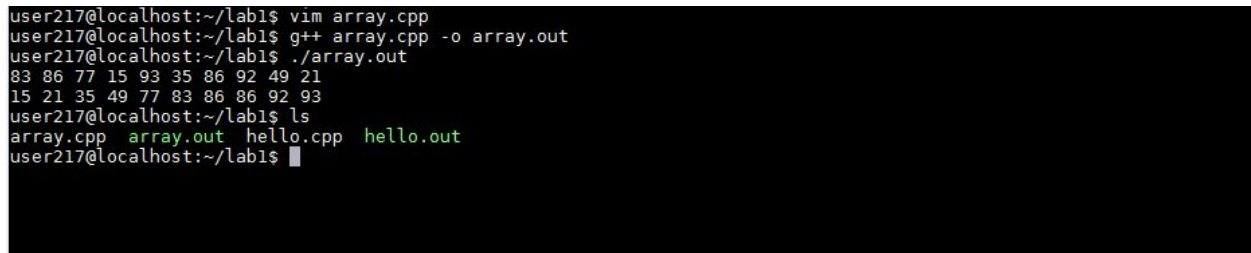
The above screenshot of the MobaXterm terminal shows the commands used to edit the code, compile the file as an executable, and run the executable. The output is “hello world”.

The command `ls` lists all the files in the current directory.

Refer to appendix 0.1 for the source code

Lab 0.2

In the same directory, a new file named `array.cpp` was created and edited.



```
user217@localhost:~/lab1$ vim array.cpp
user217@localhost:~/lab1$ g++ array.cpp -o array.out
user217@localhost:~/lab1$ ./array.out
83 86 77 15 93 35 86 92 49 21
15 21 35 49 77 83 86 86 92 93
user217@localhost:~/lab1$ ls
array.cpp  array.out  hello.cpp  hello.out
user217@localhost:~/lab1$
```

The above screenshot of the MobaXterm terminal shows the commands used to edit the code, compile the file as an executable, and run the executable. The program returns a two line output. The first line shows the randomly initialized array and the second line shows the array sorted in ascending order.

The command `ls` lists all the files in the current directory. This includes the two C++ files, `hello.cpp` and `array.cpp`, and their corresponding executables, `hello.out` and `array.out`.

Refer to appendix 0.2 for the source code

Conclusion

The majority of the work done for this experiment was designed to increase knowledge of and comfortability with the programs used. Learning the basic commands to control MobaXterm and Vim was not enough to become proficient on these platforms as they are less dynamic than many

EECE2160	Embedded Design: Enabling Robotics Lab Assignment 0
----------	--

others used today. The results of this lab are minimal on their own, but they will be used to build upon for all future labs in this course.

Appendix

0.1: The source code for hello.cpp:

```
#include<iostream>

using namespace std;

int main()
{
    //prints out hello
    cout << "hello world" << endl;
}
```

0.2: The source code for array.cpp

```
#include <iostream>
#include <stdlib.h>

using namespace std;

//Prints array in a single line
void PrintArray( int v[], int size)
{
    for(int i=0;i<size;i++) cout<< v[i] << " ";
    cout << endl;
}

//Initializes a random number 0-99 to each position in the array
void RandomArray(int v[], int size)
{
    for (int i=0; i<size;i++)
    {
        v[i]=rand()%100;
    }
}
```

```
}

//Sorts the values in the array
void SortArray(int v[], int size)
{
    int i, j, min_idx, temp;

    //starting from the left, keeps swapping so no values behind it are smaller
    for (i=0;i<size-1;i++)
    {
        min_idx=i;

        //compares all following values with the minimum index
        for (j=i+1;j<size;j++)
        {
            //makes new min index if smaller than i
            if (v[j]<v[min_idx])
            {
                min_idx=j;
            }
        }

        //swaps with new min index
        temp=v[min_idx];
        v[min_idx]=v[i];
        v[i]=temp;
    }
}

int main ()
{
    int size=10;
    int v[10];

    //unsorted array
    RandomArray(v,size);
```

```
PrintArray(v,size);  
  
//sorting array  
  
SortArray(v,size);  
  
PrintArray(v,size);  
  
}
```