

Getting Started with Linux and C++ Embedded Design: Enabling Robotics EECE2160

Michael BRODSKIY

Brodskiy.M@Northeastern.edu

March 16, 2023

Date Performed:	March 2, 2023
Partner:	Dylan POWERS
Instructor:	Professor SHAZLI

Abstract

This laboratory experiment, most importantly, served as an introduction to two main course themes: C++ programming and basic bash commands. Using this, in tandem with a secure shell connection to the De1-SoC board, rudimentary C++ code was written on the board, allowing for ease of uploading.

KEYWORDS: C++, bash, De1-SoC

1 Equipment

Available equipment included:

- DE1-SoC board
- DE1-SoC Power Cable
- USB-A to USB-B Cable
- Computer
- MobaXTerm SSH Terminal
- USB-to-ethernet Adapter

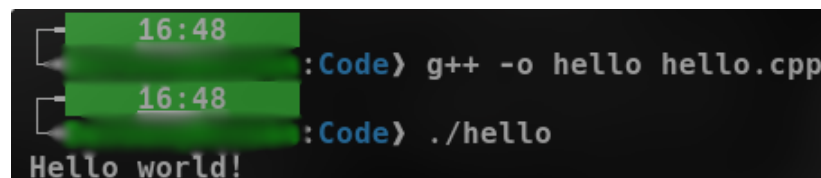
2 Introduction

The goal of this lab was to serve as an introduction to the DE1-SoC, an embedded computing device built around the Altera System-on-Chip (SoC) FPGA and an ARM processor, running an operating system with the GNU/Linux kernel. The lab began with the familiarization of the lab equipment and establishing a serial connection through a secure shell between the computer and the DE1-SoC board. After the connection was established, a simple `hello world` program was run on the ARM processor of the DE1-SoC. Following this program, functions to print, randomize, and sort arrays were created separately and then used in conjunction with one another.

3 Discussion & Analysis

3.1 Assignment 1

The goal of Assignment 1 was to run a simple `hello world` program on the ARM processor of the DE1-SoC. The `hello.cpp` C++ program is displayed in the Appendix, and the output from the MobaXterm terminal is shown in Figure 1.

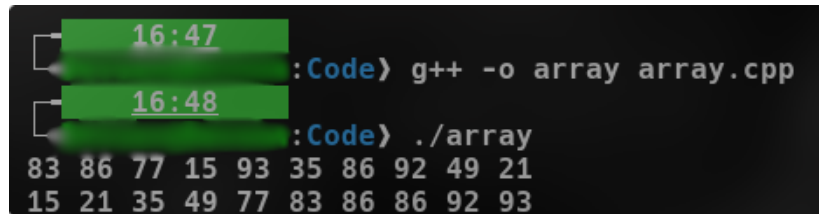
A screenshot of a MobaXterm terminal window with a dark background. It shows two command prompts, each preceded by a green box containing the time '16:48'. The first prompt is ':Code>' followed by the command 'g++ -o hello hello.cpp'. The second prompt is ':Code>' followed by the command './hello'. The output of the second command is 'Hello world!' displayed on the line below.

```
16:48 :Code> g++ -o hello hello.cpp
16:48 :Code> ./hello
Hello world!
```

Figure 1: Hello World Program Output

3.2 Assignment 2

The goal of Assignment 2 was to write a more complex C++ program that generates an array of 10 random integer numbers between 0 and 99 (line 63 in source code), prints the original array (line 64 in source code), sorts the array in ascending order (line 65 in source code), and prints the sorted array (line 66 in the source code). The source code, `array.cpp`, is attached in the Appendix and the output on the MobaXterm terminal after the execution of `array.cpp` is shown in Figure 2.



```
16:47  
:Code> g++ -o array array.cpp  
16:48  
:Code> ./array  
83 86 77 15 93 35 86 92 49 21  
15 21 35 49 77 83 86 86 92 93
```

Figure 2: Output for Array Program

3.3 Assignment 3

The goal of Assignment 3 was to write a C++ program that reads in 10 random strings from the console, sorts the random strings in alphabetical order, and prints the strings sorted alphabetically. The source code, `string_sort.cpp`, is included in the Appendix and the output on the MobaXterm terminal after the execution of the program is shown in Figure 3.



```
16:49  
:Code> ./string_sort  
Enter string #0: j  
Enter string #1: h  
Enter string #2: e  
Enter string #3: a  
Enter string #4: c  
Enter string #5: b  
Enter string #6: d  
Enter string #7: f  
Enter string #8: g  
Enter string #9: i  
a  
b  
c  
d  
e  
f  
g  
h  
i  
j
```

Figure 3: String Sort Program Output

4 Conclusion

Overall, this lab resulted in the creation of three C++ programs, each more complex than the last. First was a simple Hello World program. Next was a numerical array generator and sorter, and, finally, a string sorter. These programs allowed for a simple introduction to C++.

5 Appendix

5.1 Assignment 1

Listing 1: Hello World Source Code

```
1  /*
2  * =====
3  *
4  * Filename: hello.cpp
5  * Assignment: Lab 5
6  * Title: Print Hello World
7  *
8  * Author: M. Brodskiy & D. Powers
9  * =====
10 */
11
12 #include <iostream>
13
14 using namespace std;
15
16 int main() {
17
18     cout << "Hello world!\n";
19
20     return 0;
21 }
22
```

5.2 Assignment 2

Listing 2: Array Generator Source Code

```
1  /*
2  * =====
3  *
4  * Filename: array.cpp
5  * Assignment: Lab 5

```

```

6  * Title: Random array generator and sorter
7  *
8  * Author: M.Brodskiy & D.Powers
9  * =====
10 */
11
12 #include <iostream>
13 #include <cstdlib>
14
15 using namespace std;
16
17 void SortArray(int v[], int size) {
18     int smallest_index;
19
20     for (int i = 0; i < size; i++) {
21
22         smallest_index = i;
23
24         for (int j = i + 1; j < size; j++) {
25
26             if (v[j] < v[smallest_index])
27                 smallest_index = j;
28
29         }
30
31         swap(v[i], v[smallest_index]);
32
33     }
34 }
35
36 void RandomArray(int v[], int size) {
37
38     for (int i = 0; i < size; i++) {
39
40         v[i] = rand() % 100;
41
42     }
43
44 }
45
46 void PrintArray(int v[], int size) {
47
48     for (int i = 0; i < size; i++) {
49
50

```

```

51         cout << v[i] << " ";
52
53     }
54
55     cout << endl;
56
57 }
58
59 int main() {
60
61     int v[10];
62
63     RandomArray(v, 10);
64     PrintArray(v, 10);
65     SortArray(v, 10);
66     PrintArray(v, 10);
67
68     return 0;
69
70 }

```

5.3 Assignment 3

Listing 3: String Sorter Source Code

```

1  /*
2  * =====
3  *
4  * Filename: array.cpp
5  * Assignment: Lab 5
6  * Title: Random array generator and sorter
7  *
8  * Author: M.Brodskiy & D.Powers
9  * =====
10 */
11
12 #include <iostream>
13 #include <cstdlib>
14 #include <string>
15
16 using namespace std;
17
18 void SortArray(string v[], int size) {
19
20     int smallest_index;
21

```

```

22     for (int i = 0; i < size; i++) {
23
24         smallest_index = i;
25
26         for (int j = i + 1; j < size; j++) {
27
28             if (v[j] < v[smallest_index])
29                 smallest_index = j;
30
31         }
32
33         swap(v[i], v[smallest_index]);
34
35     }
36 }
37
38 void ReadArray(string v[], int size) {
39
40     for (int i = 0; i < size; i++) {
41
42         cout << "Enter string #" << i << ": ";
43         cin >> v[i];
44
45     }
46
47 }
48
49 void PrintArray(string v[], int size) {
50
51     for (int i = 0; i < size; i++) {
52
53         cout << v[i] << endl;
54
55     }
56
57 }
58
59 int main() {
60
61     string v[10];
62
63     ReadArray(v, 10);
64     SortArray(v, 10);
65     PrintArray(v, 10);
66

```



```
67     return 0;  
68  
69 }
```