Embedded Systems Course

# Laboratory Practical 0:
# C Programming Practice

| | |
|---|---|
| TEAM: | Two |
| DURATION: | 2 hours |
| DOC REF: | Prac0.docx |

## 1 Introduction

The purpose of this practical is to allow you the opportunity to practice your C programming skills before starting on the main laboratory practicals for this course. This practical is voluntary and does not count for marks. You do not need to submit anything for this assignment, but if you do want to, you can do so using the Vula site – please upload just your main.c file, not any of the additional executable or binary files generated by the compiler.

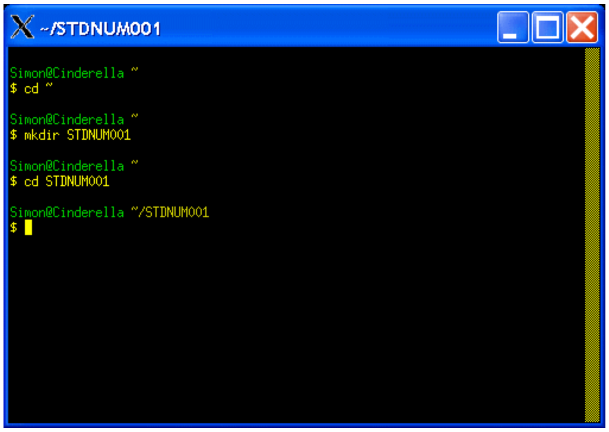
## 2 Starting out

Start by logging in to Ubuntu (or whatever Linux you are using), if you haven't already. The username and password are both **bluelabuser**. When you have logged in, opening a new Linux console on your computer. This should give you the equivalent of a command prompt in Microsoft Windows.

At the command line, create a subdirectory named according to your student number in the home directory. For example if your student number was STDNUM001, you would do as follows:

```
cd ~
mkdir STDNUM001
cd STDNUM001
```

The display would likely look something similar to that shown on the right.



```
Simon@Cinderella ~
$ cd ~

Simon@Cinderella ~
$ mkdir STDNUM001

Simon@Cinderella ~
$ cd STDNUM001

Simon@Cinderella ~/STDNUM001
$ 
```

Now download the Prac0.zip file that is provided with this document in the VULA Assignmet Prac0. Save the file into the home directory.

From the command line (within the STDNUM001 subdirectory), unzip the file as follows:

```
Unzip ~/Prac01.zip
```

List the files that were extracted using the following command:

```
ls -l
```

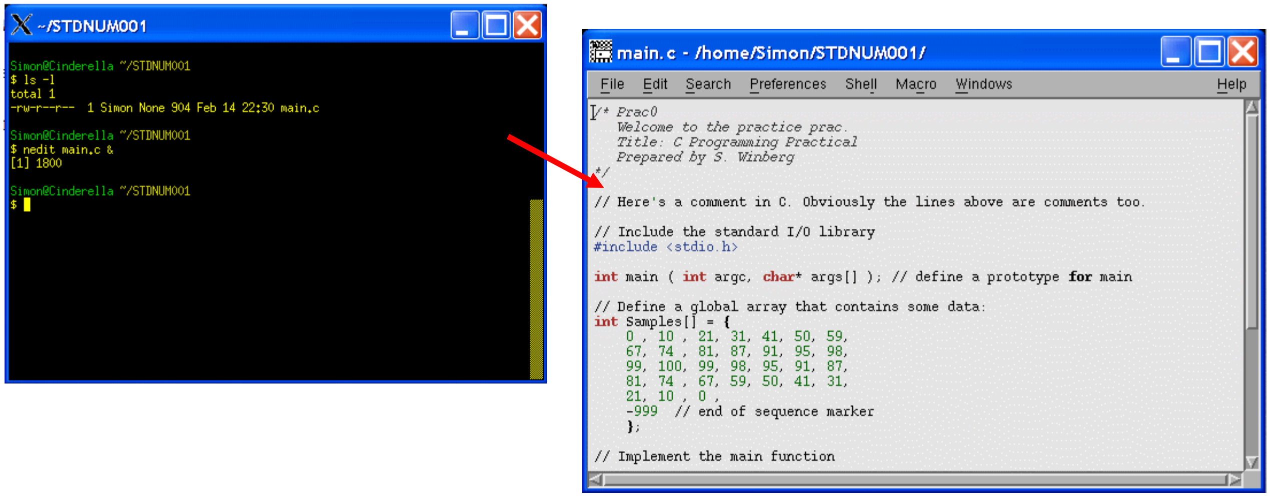
You should get a list something like the following:

```
$ ls -l
total 1
-rw-r--r--  1 Simon None 904 Feb 14 22:30 main.c
```

The –l option to the ls (i.e., 'list') command shows details of the size and date information of the files ('ls' on its own doesn't include this information).

Now open the main.c file. You could use vi if you are brave, otherwise use something like *gedit*, *nedit* or *kate*.

See example below…



```
Simon@Cinderella ~/STDNUM001
$ ls -l
total 1
-rw-r--r--  1 Simon None 904 Feb 14 22:30 main.c

Simon@Cinderella ~/STDNUM001
$ nedit main.c &
[1] 1800

Simon@Cinderella ~/STDNUM001
$ 
```

main.c - /home/Simon/STDNUM001/
File Edit Search Preferences Shell Macro Windows Help

```
/* Prac0
   Welcome to the practice prac.
   Title: C Programming Practical
   Prepared by S. Winberg
*/

// Here's a comment in C. Obviously the lines above are comments too.

// Include the standard I/O library
#include <stdio.h>

int main ( int argc, char* args[] ); // define a prototype for main

// Define a global array that contains some data:
int Samples[] = {
    0 , 10 , 21, 31, 41, 50, 59,
    67, 74 , 81, 87, 91, 95, 98,
    99, 100, 99, 98, 95, 91, 87,
    81, 74 , 67, 59, 50, 41, 31,
    21, 10 , 0 ,
    -999  // end of sequence marker
    };

// Implement the main function
```

In the editor, take note of the first lines showing how to write comments in C.

In the editor search for the text "PUT YOUR NAMES HERE!!". Change this text so that it shows your names (i.e., if you are working with a project partner, put both names in the text).

Now let's compile and run the application. This can be done as follows:

```
$ gcc -o test1 main.c
```

The command above with compile and link the program, generating the executable file called 'test1' (note in Cygwin this will be called test1.exe).

Now run the generate file as follows:

```
$ ./test1
```

This will give output that looks like:

```
$ ./test1.exe
Simple Signal Processor
Programmed by:  Your name here
RESULTS:
```

Now for the code you need to develop…

## 3 Activities

The *Samples* integer array (defined on line 15 of main.c) defines a number of integers. Note that the negative number, namely -999, indicates the end of the array. No size information is specified.

Complete the following. Your final program needs to do each of these steps in sequence:

1. Use a for-loop to determine the number of integers in the array (i.e., from index 0 to the index one before the -999 value). [5 marks]
2. Use a while loop that asks the user to input a number and then searches the Sample array to determine if that number is found in the array (if found, display the message FOUND, otherwise NOT FOUND). End the loop when the user enters a negative number. [5 marks]
3. Calculate the average (i.e., mean) value for all the integers in the array. This mean should be a float value, display the average to accuracy of two decimals (i.e. +- 0.01 accuracy). [10 marks]
4. Display the minimum and maximum values in the Sample array (note: you could do this calculation as part of step 3 if you want an optimized solution). [5 marks]

Quality of comments: [5 marks]

Note mark allocations are shown as a guideline only as this prac doesn't count for marks.

*end of assignment*