Module 2 – Getting Started

Tutorial Questions

Objectives

As an introduction to the language we will dissect a basic ANSI C program. By the end of this tutorial you should have a rudimentary understanding of C compiling techniques, program initialisation, the use of header files, primitive types and function calls.

Source Code

```
#include <stdlib.h>
#include <stdio.h>
int main (int argc, char *argv[])
{
     int a, b, c;
     char string[] = "ello world";
     char character;
     a = 2;
     b = 4;
     c = a * b;
     printf("%d * %d = %d\n", a, b, c);
     character = 'H';
     printf("%c%s\n", character, string);
     return EXIT_SUCCESS;
}
```

Questions

1. How do you compile and run a C program in a UNIX environment? What compile flags do we recommend that you to use? Why is it important to use these flags when compiling?

   gcc –ansi –Wall –pedantic file.c –o file

2. The first two lines of our example include two standard header files. What is the purpose of including these files? Can you name any other standard header files?

   Including these files tells the compiler that the functions and constants declared in those files should be available to the program. Other header files: time.h, string.h, ctype.h, …

3. What function is called at program start-up? What are the standard prototypes for this function?

   _start() is the first function called at startup however that is already written for us. The first function that we implement that is normally called inside a function is main(). There are two standard prototypes of this function:

   int main(void);
   int main(int argc, char* argv[]);

   For the first part of the course, the first prototype will be used.

4. The example uses two data types, what are they? Can you name any other primitive types?

   This program uses int, char and char[]. Other datatypes in c are float, double, long. We can also create our own types with enum and struct keywords.

5. Discuss the use of the format string in the two calls to printf. What is the output of each call?

   The format specifiers (%s, %d, %c) specify the type of the data that printf will expect (string, decimal integer, char).
   The output of this program is:

   2*4=8
   Hello world