Programming Windows

Terry Marris January 2013

1 Hello World

The intention here is to simply, and perhaps naively, program Windows applications using the Win32 Application Programming Interface (API) libraries and the C programming language.

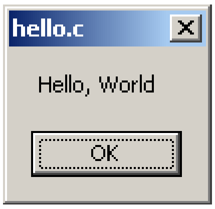
These notes are intended as a primer, to get you started.

Unicode represents most characters in most languages of the world. ASCII is just a small subset of Unicode, sufficient to represent most characters used in the English language. To keep things simple, consideration of Unicode will be left to the serious and professional Windows programmers, and not considered here.

The reader is assumed to be comfortable with ANSI C.

1.1 Hello World

The first program displays *Hello, World* in a message box.



hello.c

Hello, World

```
/* hello.c - displays a greeting */

#include <windows.h>

int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance,
                   PSTR cmdLine, int cmdShow)
{
   MessageBox(NULL, "Hello, World", "hello.c", 0);
   return 0;
}
```

windows.h is the standard header file that contains specifications for Windows Win32 API functions, macros and constants.

WINAPI represents the calling convention used by most Win32 API functions.

WinMain is the program entry point. Program execution starts with *WinMain*. *WinMain* always has four parameters.

HINSTANCE stands for *a handle to an instance*. A handle is just an integer that identifies an object. *hInstance* uniquely identifies the program when it is running under Windows. You could have several copies of the same program running at the same time. But each copy would have its own *hInstance* value. And each copy is known as an *instance*.

hPreviousInstance is now obsolete.

PSTR stands for *pointer to a string*. C programmers would normally write *char* * to represent the same thing.

cmdLine stores any arguments provided by the user when the program is loaded. *cmdLine* will be used in a later chapter.

cmdShow is an integer that indicates whether the program is to be launched in a normal sized window, or minimised. Its argument values are usually supplied by another program.

MessageBox has the format

MessageBox(owner, text, title, icons)

Here

```
   MessageBox(NULL, "Hello, World", "hello.c", 0);
```

the text box has no owner window, the text to be displayed is *Hello, World*, the text that appears in the title bar is *hello.c*, and there are no icons. More will be said about MessageBoxes in a later chapter.

1.3 Resources

We sit at the feet of giants. Kernighan and Ritche and their book The C Programming Language, and Charles Petzold and his books Programming Windows 95 and Programming Windows 5th Edition provide the inspiration and source, updated by the often anonymous guys at Microsoft *http://msdn.microsoft.com/en-gb/library/windows/desktop/*

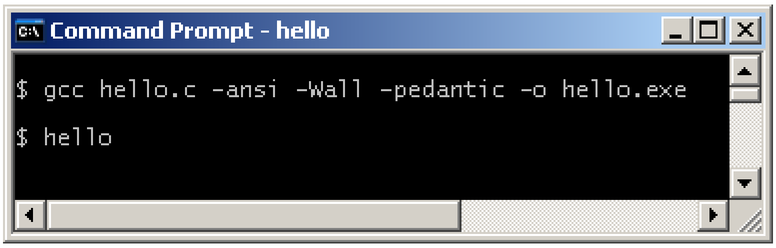
And to the guys who built and maintain the GNU C and resource compilers: respect and thanks.

The C and resource compilers and run-time libraries, are from the GNU Compiler Collection at *http://gcc.gnu.org*.

The editor used is the vanilla Windows text editor, Notepad. And the programming environment is Microsoft's Command Prompt Window.

1.4 Program Development Environment

The program development environment used is the Windows Command Prompt window.



```
$ gcc hello.c -ansi -Wall -pedantic -o hello.exe
$ hello
```

$ is the command line prompt (set by entering *prompt $$*).

gcc loads the C compiler.

hello.c is the source file to be compiled.

-ansi, *-Wall* and *-pedantic* are all switches to the compiler requesting most errors and warnings to be displayed.

-*o* is for output.

-*o hello.exe* puts the output of the compiler and linker into *hello.exe*.

hello.exe is the executable program, run by entering *hello* at the command line prompt.

You could navigate to the directory containing the source code before compiling. Use

dir to list the contents of the current directory
cd <dirName> to change to a child directory
.. to change to a parent directory

You may prefer to use another program development environment, compiler and editor.