2nd Edition

AN INTRODUCTION TO C & GUI PROGRAMMING

UPDATED FOR GTK 3

Simon Long

Raspberry Pi PRESS

First published in 2019 by Raspberry Pi Trading Ltd, Maurice Wilkes Building, St. John's Innovation Park, Cowley Road, Cambridge, CB4 0DS

Publishing Director: Russell Barnes • Editor: Phil King • Sub Editor: Nicola King
Author: Simon Long • Design: Critical Media
CEO: Eben Upton

Welcome to **An Introduction to C & GUI Programming**

The C programming language was invented in the early 1970s, and since then has become one of the most popular and widely used general-purpose languages. C can be used to create simple command-line programs, or embedded code to operate the tiny microcontrollers in toasters and watches. At the other extreme, it can be used to create rich graphical desktop applications – in fact, most of Linux (and Raspberry Pi OS itself) is written in it. It can give you control over the smallest details of how a processor operates, but is still simple to learn and read. The first part of this book is an introduction to programming in C for absolute beginners; the second part shows how to use C to create desktop applications for Raspberry Pi OS. You don't need any programming experience, and a Raspberry Pi running Raspberry Pi OS is all you need to get started.

About the Author

Simon Long is an engineer working for Raspberry Pi. He is responsible for the Raspberry Pi Desktop and its associated applications. Before joining Raspberry Pi, he worked for Broadcom, where he first met Eben Upton, and before that spent ten years working as a software engineer and user interface designer for a major consultancy firm. In his spare time, he enjoys solving those really hard crosswords without any black squares.

"Dedicated to the memory of Pythagoras, a very special cat, and my devoted companion while writing this book."

Contents

Chapter 1

Getting started

C is one of the most widely used programming languages – learn how to use it to program the Raspberry Pi!

What's so great about C?

C is a very versatile and widely used programming language. It has been used to write pretty much everything, from low-level routines to control the hardware in embedded microcontrollers to complete operating systems like Linux with graphical user interfaces. In spite of this huge flexibility, it is also relatively simple – the language only has about 20 or so keywords, but there are huge libraries of additional functions that you can call on when you need them. In the first part of this book, we are going to concentrate on learning about the keywords, with a few of the more commonly used library functions; the second part of the book shows how to use the GTK library to make it easy to write graphical interfaces in C.

Many of the languages that you may have seen, such as Python, are what are called *interpreted languages*. This means that the code you write is run directly: each line of code is read in and interpreted as you run it. C is different: it's a *compiled language*. This means that the code you write, known as the *source code*, is never run directly. The source code is passed through a program called a *compiler*, which converts it into a machine-readable version called an *executable* or a *binary*; you then run the resulting executable.

This may seem complex, but it has a few big advantages. First, it means that you don't need to have a copy of C itself on every computer you want to run your program on; once compiled, the executable is standalone and self-contained. Second, the compilation process will find a lot of errors before you even run the program (but it won't usually find all of them). Most importantly, the compilation process means that the time-consuming translation of human-readable code into machine-readable instructions has already happened, which means that compiled code generally runs many times faster than interpreted code would.

CHOOSE YOUR EDITOR

You can use whatever editor you like to enter code, as long as it saves it as plain text. The Geany editor included in Raspberry Pi OS is a good choice, but you can also use Leafpad, nano, or any others that you prefer.