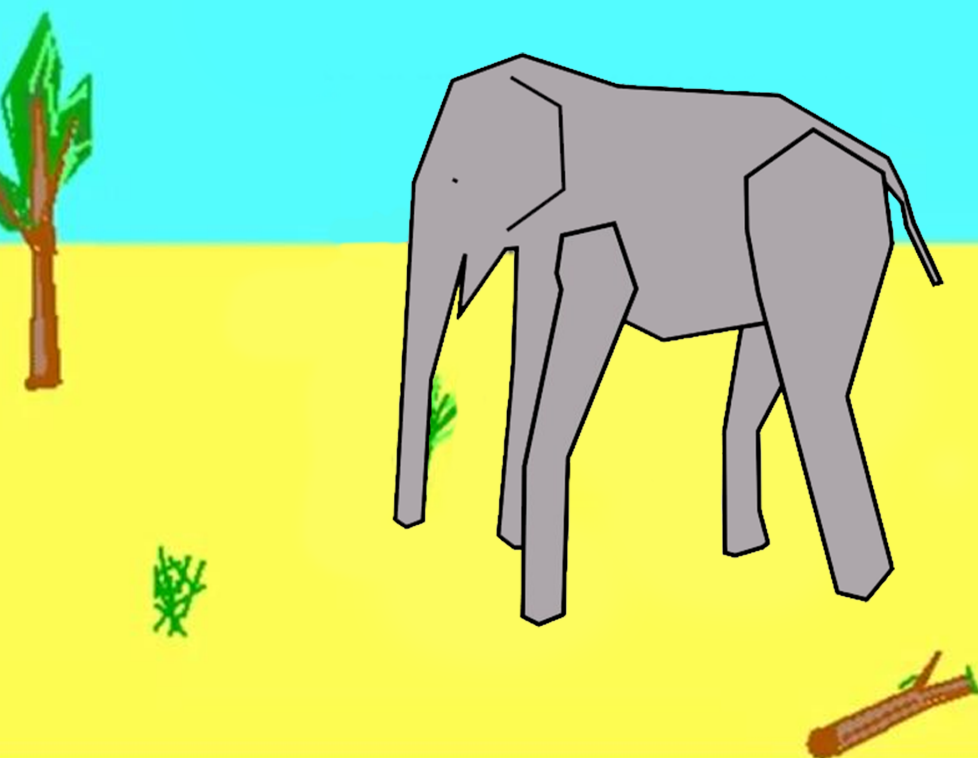


# The Ultimate TempleOS Handbook

---

Austin Eccentrici  
Terrence Andrew Davis



For Terry, the most humble and wholesome King.

# Acknowledgements

WRITING a book is surely not a simple task, especially when writing a book on programming, let alone a whole operating system. Thus it would be nearly impossible to write this book solely by oneself, therefore an acknowledgement is necessary to credit the people who have helped this book to exist.

- **Professional Necrophiliac:** For fixing the author's atrocious and mediocre vocabulary insufficient to write a book, and for doing it for free
- **Someone:** for reason
- **Terrence Andrew Davis:** this would not be much of a "TempleOS" book if this book does not commemorate the man who inspired this book, the creator of TempleOS

# Contents

<b>1</b>	<b>Acknowledgements</b>	<b>3</b>
<b>2</b>	<b>Preface</b>	<b>5</b>
	<b>PART ONE</b>	<b>7</b>
<b>3</b>	<b>The Temple</b>	<b>8</b>

# Preface

ADVANCES in the operating system development world has led us into an era of general-purpose computing and the accessibility of computers to the general public(which is often referred to as **niggercattle** by Terry Davis for their cattle-like obedient behavior), but has also resulted in a madness of unnecessary features, excess telemetry nobody asked for, and bloat which has been mostly unfixable due to the enterprise and proprietary nature of operating system development, since designing and programming a whole general-purpose operating system, complete with the userspace and kernel is not a task a single man can easily undergo.

TempleOS aims to be a single-user hobbyist operating system for the joy of programming which seems to have been forgotten since the rise of general-purpose operating systems. Terry Davis set a limit on the Temple Operating System's line of source code to keep its simplicity and codebase disciplined.

This book is the first detailed and coherent account of Terry Davis and the Temple Operating System. It does not document about unorthodox forks of the operating system. It is divided into 4 major parts:

- Part One - an overview of the key concepts of the operating system

- Part Two - a detailed technical description of the TempleOS ecosystem, involving HolyC and the TempleOS graphics library
- Part Three - various annotated and illustrated examples of HolyC programs
- Part Four - a detailed log of Terry Davis' life and philosophy which was the key to developing TempleOS

This book assumes the user has a basic understanding of:

- What an operating system is
- How to boot up a virtual machine
- What the command line is
- The C programming language
- Terry Davis' views and philosophy

The knowledge of these items do not necessarily have to be deep, however if the reader is not able to write a **for loop** or does not have an understanding of what a **pointer** is in the C programming language, he will have a very hard time understanding this book.

This book draws inspiration from **Smalltalk-80: The Language And Its Implementation**: the most well-organized and professional programming book ever written. The author would like to add special thanks to Adele Goldberg, David Robson and the rest of Xerox PARC for such a wonderful book.

# PART ONE

Part One of this book provides an overview of the core concepts and aspects of the Temple Operating System. Chapter 3 introduces the basic concepts and vocabulary of the Temple Operating System. Chapter 4 describes the general usage and aspects of TempleOS which is radically different from mainstream operating systems.

# The Temple

THE Temple Operating System is a ring-zero, multicore x86\_64 operating system that is a result of Terry Davis(1969 - 2018)' decades worth of work. It is not an easy task to explain all aspects of the very intricate network of features, quirks and surprises that is TempleOS, but the most profound and widely known are:

- The ability to talk to God using a random number generator which uses a timer as the seed for its algorithm
- A programming language which is an improved version of the C programming language called HolyC
- The operating system's shell(the software used to communicate with the operating system and launch other programs) which uses HolyC instead of a scripting language such as Bash/POSIX sh
- A JIT compiler which compiles the whole operating system's userspace on boot

There is much more to this but they will later be elaborated while Part One advances.