OZ of Programming

Shiv S Dayal

Dedicated to my family and
Free Software Community.

Copyright ©Shiv Shankar Dayal. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation. The GNU FDL is there in the same directory as this file with name “LICENSE”.
Wikipedia images are under their LICENSE.

Contents

2 2

Tables

Figures

Preface

Please let me introduce myself. As you know my name is Shiv Shankar Dayal if you have taken care of reading author's name on the first page which I expect you have unless you were too busy and having your attention anywhere else. I am a programmer by choice and was so by profession as well till recent past when I decided to quit job because of personal problems. So I decided to write this book to pass my time as I am at home full time. There are many books on C, however, for a person like me who has no source of income is forced to rely on open-source books. I agree that there are tons of excellent free (free as in freedom, not free beer) software, but few free books. Again, there is a problem that software can be created and copied and maintained easily but such is not the case with books if you publish it. Therefore, I have decided to keep under GNU FDL and distribute only soft copies. If some publisher publishes it even then soft copy will remian free.

There is another reason for writing a book on C. First, I think I know it well. Just to brag I have been programming for 11 years, however, my knowledge is very limited and inaccurate when it comes to programming. On the contrary I write programs very slow. They are never commented and complex and difficult to maintain. Second, C is kind of greatest common denominator of many major software. Almost all Unix-like systems and also Windows have been developed in C. Programming languages like C++, Java, Perl, Python, Ruby, Lua etc have all there roots somewhere in C. Again, I am not sure facts please cross-check about names. I am just guessing. All the languages have C bindings. Backward compatibility with C is a very important reason for the success of C++. However, that is again a matter of entirely different discussion. I am just saying what I think.

This is a book on C as you already know by its name. However, this book will delve into topics not ventured together in many books. The most authoritative source on this subject is ofcourse K&R's "The C Programming Language". This book is not a replacement of the above book. It is a complement and supplement for the K&R's (henceforth I will refer it as The C Book) book. However, C has undergone revision and latest standard being C99 which is not part of The C Book. Therefore, until a revision is made in the contents it will not reflect the changes in the standard. Several compilers are already supporting this standard. `gcc` already supports most of standard which is used in preparing this book. This book attempts to fill the gap between standard, gcc and The C Book.

I believe in practical programs so sometimes code examples may span several files and pages. Please skip them if you are not interested. Sometimes toy examples are not simple.

My thoughts jump from one topic to another and I capture them immediately so the book may seem rather wayward. I have tried my best to organize the topics but still the ordering may be not so good. Please use index. After all, that is what indices are meant for! :-)

Let us come back to the title. As you have seen it is "Learn Correct Programming". There are various ways to program. However, programming paradigms associated with language does not enforce correctness in the sense of correct way. Two programming paradigms which are generic try to do it. One of them is "Literate Programming" and

"Pair Programming". Literate Programming is baby of greatest computer scientist alive in my humble opinion Donald Erwin Knuth. I am a big fan of Literate Programming, however, I can not enforce documentation or learning TEX on a normal programmer. Pair programming is not always feasible and not particularly so in enterprise environment where the sole aim is to make money. However, if you can please use both of them as they will help you learn programming immensely. Programming requires discipline. Discipline in choosing correct language for the task, correct selection of algorithm, correct implementation and never to mention documentation and testing.

As you may notice there is something called Part I C in this book. There is a reason that once this part will be finished I will put more parts probably C++ or some library uses like pthreads.

Prerequisite

As such there are not many prerequisite. This book is well suited for beginners as well as advanced programmers. Since I have tried to follow C99 standard the terminology may be a bit different. I expect the reader to be at least having high-school level knowledge of Mathematics. If the reader does not have this prerequisite then I strongly recommend him to study it as some of the sections may be difficult for him to comprehend. Programmers from any language will be able to make rapid advances while studyting the text and may even skip some sections or chapters entirely.

As such this book is meant for anyone having more than casual interest in C programming language, beginners may face some problem and may also need guidance while studying it. Since I have tried to follow the specificaiton at times there are forward references.

Tools

You must choose your tools carefully as it may determine the direction your career is going in. If you want to Windows programmer choose MS Visual C++ Express Edition if that suits you else get cygwin or MSYS and MINGW. However, you can also choose GNU/Linux (free comes in many variations), MacOS X (not free and supposed to run only on Mac hardware but internally is Unix), OpenSolaris but at this moment it is lagging as only 2009 release is there.

As compiler I use `gcc`, `gdb` as debugger, `emacs` along with viper-mode, cedet, Emacs class browser as my editor and ide. Emacs is fantastic because I can compile and debug right there in editor. Also, viewing manuals in Emacs is pretty easy for quick reference. But be warned Emacs has a steep learning curve and may be daunting who has only used Windows. It will feel better if someone entirely new to computers uses it. Also, Emacs has an in build shell, browser, and many other things which I do not even know. Best part is that if you want some functionality then it can be extended using Emacs-Lisp which is another programming language of its own type. I also use `valgrind` whenever

I suspect a suble memory problem or thread problem. All these tools are licensed and cost nothing.

Acknowledgements

The time when I am writing this book is very tough on my family and even then they have supported me. I can not show enough gratitude to my wife, parents and siblings. They have stood by me and my decisions to quit job which is a big change in our lives. I can not state how much the life of my parents have effected me.

I thank my teachers Yogendra Yadav, Hriday Narayan Singh, Satyanand Satyarthi, Gopal, Shilesh Kumar, Praveen Kumar, Prod. T K Basu, Prof. S Sen, and Prof. S Banerjee for imparting their knowledge.

At professional level I learnt a lot from two persons. First is Anurag Johri and second is Vibhav Saluja. They taught me several technical things and a lot more about life.

On the programming side there are far too many people I mena whole Free Software Community, however, I will take some names. Donald Erwin Knuth, creator of TEX, which has been used in preparation of this book, who is also author of many books like ``The Art of Computer Programming'' is first among them. Richard M Stallman is second but his contributions are more to life than software, however, that does not mean that he has done any less contribution on software side. He contributed to `gcc`, `gdb`, `emacs`, TEX`Info`, `ELisp` and may more things. His biggest contribution is free software movement. I also want to thank Hans Hagen who helped me with ConTEXt which I used to finally process the contents of this book. I also used `vim` syntax highlighting module from Aditya Mahajan to highlight the source code given in this book. I am most grateful to all these nice people who believe in sharing the knowledge at source code level and not at binary level.

English is not my native language and even though I have run spell check over buffers there are chances that grammatical and factual errrors are there. The reason is that it has not been proof-read and edited. All I can guarantee is that source code given runs on my machine. Please mail me your suggestions, errors which you have found to me at **shivshankar.dayal@gmail.com**. I will try to give reply to each of your mail.

Shiv Shankar Dayal

Part I

Maths, Physics and Chemistry

1 Mathematics, Physics and Chemistry

Let me reiterate here that I am taking a botom-up approach for studying this book to the reader. If you already know this or do not wnat to see the basics of how a computer works please skip to Part II and again in II if you are not interested in specification directltу skip to the part where programming sub-part begins and continue doing so.

When these three things combine only then a computer is born else there is no computer. This is going to be a very big chapter and only one chapter covering all there as this books is meant for programming not science. We will srudy Chemistry first then Physics and Mathematics as reuquired.

As you know almost all the chips and electronic cricuitry is made up of Silicon so let us first study about it. When images will come from **http://en.wikipedia.org/** it will be written in footnote.

1.1 Silicon

We will first study the Chemistry of Silicon whose symbol is Si in perodic table. Most of the images are from wikipedia so their licenses apply for images. Silicon's atomic number is 14 which means it is a group IV element and has got a valency of four that is it can make four covalent bonds. You may also know its distribution in shells and subshells and that $1s^1\ 2s^2\ 2p^6\ 3s^2\ 3p^2$. As you can see second orbital has full octet so it should have a valency of two as the 3s subshell is also filled. But when it reacts the two electrons mover from 3s to 3p and then there are 4 valence electrons available for reaction. There are certain reasons for it which involve going deep in Chemistry which I would like to avoid.

The very pure form of this element is very important in compueter or rather semi-conductor electronic industry, Silicon is the ssecond most abundant material element by weight on out planet (mother earth). It has got 27,200 ppm (parts per million) and relative abundance is 2 only after Oxygen which has relative abundance of 1. So it seems god wants us just to breath and program! :-)

Tons and tons or rather millions of tons of Silicon is produced every year across the globe just to meet the requiement of computer and electronic industries. The very requirement of Silicon from computer and semiconductor inductries are very strict in nature as they need very pure form of Silicon. Only 1 out of 10^9 parts may be impurity. That means 1ppb (part per billion is permitted.)!!!

1.1.1 Phsical and Chemical Properties

For the curious one :-) I would have liked to give a table but a list is more suitable in my opinion. I would have been given a table if more elements would have been there to compare side by side.

1. **Overview**

i. Atomic Number: 14

ii. Group: 14

iii. Period: 3

iv. Series: Metalloids(non-metal)

2. **Atomic Structure**

i. Atomic Radius: 1.46 Å

ii. Atomic Volume: 12.1 cm^3/mol

iii. Covalent Radius: 1.11 Å

iv. Cross Section: (Thermal Neutron Capture) σ_a/barns:0.171

v. Crystal Structure: Cubic Face Centered

vi. Electronic Configuration: $1s^2\ 2s^2\ 2p^2\ 3s^2\ 3p^2$

vii. Electrons per enerfy level: 2 8 4

viii. Shell Model:

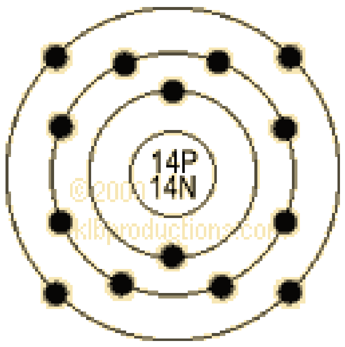

Figure 1.1 Shell Model Stucture of Silicon[1]

ix. Ionic Radius: 0.4 Å

x. Filling Orbital: $3p^2$

[1] http://envirochem.us/images/periodic/shellmodel/Si.gif

xi. No.of Electron (with no charge): 14

xii. Number of Neutrons (most common/stable nuclide): 14

xiii. Number of Protons: 14

xiv. Oxidations: 4

xv. Valence Electron: $3s^2$ $3p^2$ Electtron Dot Model

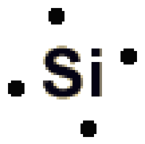

Figure 1.2 Electron Dot Model of Si[2]

i. **Chemical Properties**

ii. Electrochemical Equivalent: 0.26197 g/amp-hr

The most commons source of Silicon exist as Silicon sioxide $[SiO_2]$ which is found as sand and quartz. As you may be knowing in $CO_2]$ there are two pi bonds and two sigma bonds that is $p\pi - p\pi$ bonds and fills its ocet. Because of this reason CO_2 can form a discrete molecule and hence is in gaseous form at room tempreature. The bonds in SiO_2 are all σ bonds.It has a 3-dimensional infinte crystal structure like diamon. Given below is one such diagram for SiO_2. You will be able to clearly see how Silicon and Oxygen atoms form an infinte 3-dimensional crystal in space. SiO_2 has been found to exist in at-least 12 different forms.. The most common ones are quartz, trydymite and cristobalite. Each of these have different strutures at different temperatures.

α-Quartz is the is most common and is a major constituent of granite and sandstone. Pure Silicon dioxide is colorless but impurities may give it as color. Birthstones are colored because of these impurities.

1.1.2 Sources

Silicon

[2] http://envirochem.us/images/periodic/ElectronDot/Si.gif
[3] http://www.utas.edu.au/sciencelinks/chemincon/files/s1_grow/s1_soils/bonding/images/network.gif

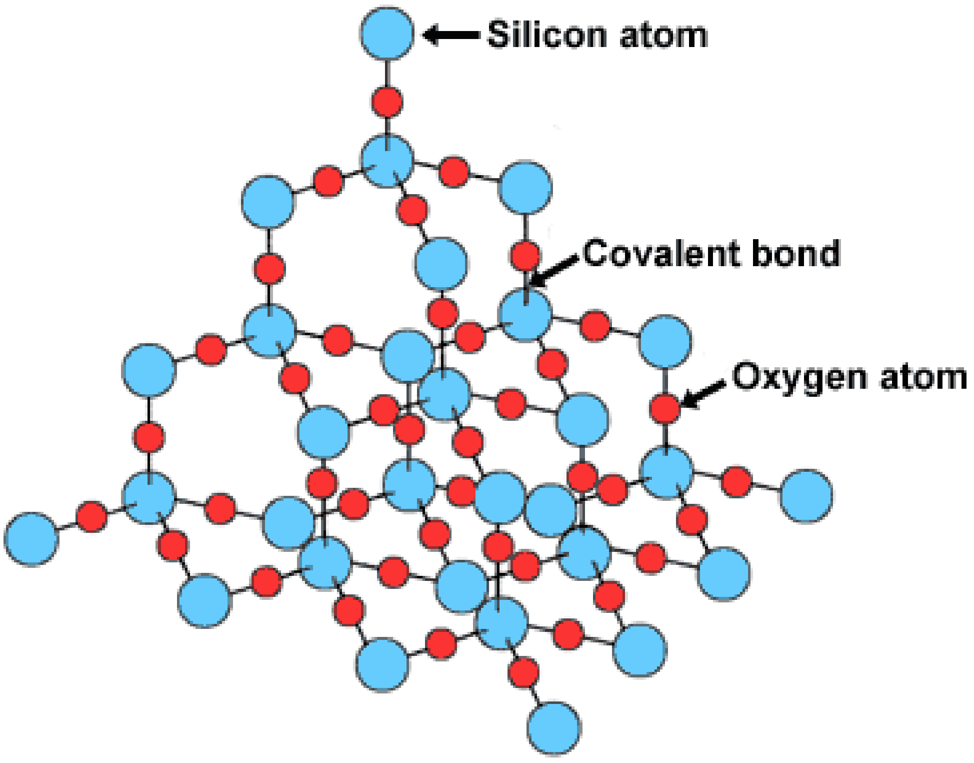

Figure 1.3 Crystalline Stucture of Silicon dioxide[3]