Biography on Self-study of Computer Science

Eddie Wu, Shanghai

March 28, 2009

Abstract. I started to self-study computer science from the autumn of 1999, till now, almost ten years has passed. During this period, I have learned a lot of subjects on computer sciene, including Programming Language, Operating System & Kernel, Oracle Database, Algorithm & Algorithm Analysis, Network Security, Computer Architecure, Regular Expression, etc. I try to wrote down this kind of experience, just for records and for fun.

1. Programming Language.

1.1 Java.

Java is the first programming language I meet during my self-study. I still remember the first Java book I bought is the Chinese Translation of "Java How to Program". especially I remember the confusion I have gotten when I see the self-reference structure in Java, such as the linked node list, I thought it is kind of recursive definition, because at that time I do not know that at the low level, reference is just a pointer, a word of random access memory, it does not have to have any type information.

Before that, I already I have gotten some ideas about programming from college's Visuan Basic course, so it not a problem for me to understand $a = a + 1$.

After that I also get the oppotunity to develope J2EE application and learn it very quickly.

1.1.1 Core Java.

Here the core Java means the basic Java related technology, incluging the Java Language Grammar, Basic package, including tt java.lang.*, java.util.*, and also the capability to apply Java in real life problems.

1.1.1.1 Java Basic(12/1999 – 05/2001).

At that time, I am interested on playing GO, which is a ancient board game oriented from Chins and is popular in China, Japan, and Korea. The very first non-trivial program I have ever developered is a program used to play Go.

Basically, I need to translate the rule of GO into the Java language, I also need to consider the data structure to represent the board and status of Board.

I was a wonderful experience for me, I become familiar with the process of solving problem with computer.

1.1.1.2 Chess Game(10/2003-02/2004).

One time when I try to get a job, I went to Pudong for interview. during the interview, I show them the GO game I developed. they are interested on it. they did not offer me the job but they offer me a part time job—to co-develope an online Chess game.

For sure I am very familiar with the Chess and it's rule although I am not good at playing. I finishe the game very quickly and delivered the prototype in one week. but we enocunter some problem in integeration. The overall performance of the game is not good. It is majorly caused by the network related

1.1.1.3 SCJP.

After I start to work as a full time Java developer, I decided to persure the SCJP, which I have heard of since I know Java. I spend about two weeks to go throught the reference book and test my understanding with small program. This two weeks' system learning make a solid foundation of my Java capabilities.

1.1.1.4 SCJD & SCJA.

I also consider to persue SCJD & SCJA. but I decided not to take them. For SCJD, which is majorly about GUI programming, but I do not have oppotunity to use it in my full time job. For JSCA, it is majorly based on EJB technology, buf in my point of view, in most of the cases, light-weight technology, such as Spring will be more pragamtic.

1.1.2 Java Enterprise Edition (J2EE).

My J2EE experience start from 3/15/2004 when I get my first full time IT job. several days later, I received another offer from another company which develope network related software, such as firewall. I did not consider that offer since I just started my new work.

1.2 C.

I start to learn C when I decide to take the National Programmer Exam. I reference book I used is "C How to program". The most imressive small program I have written during the learing is a program to find the Horse Transversal in ChessBoard with Backtracking Algorithm.

Another impression I have is that Pointer in C is actually easy to understand compared to other technology or concept.

After i start to work on j2EE platform, I still spend some time to read an old book called "Secret in C" during my leisure time. the most impressive thing i leared is the difference between Pointer and Array, they are different, but so closely related that they can be exchanged in most of the case.

1.3 C++.

After I have master the Java programming language, I start to have a look on the C++ programming language. since I do not have opportunity to use it in my daily work, I focus on the theory and concept of C++ although I did verify the concept with small C++ programs. The impressive books includes "More Effective C++", "C++ Design and Envolution", "Accelarated C++".

Another reason to learn C++ is to understand the performance difference between Java and C++. But finally I realized that it is hard to compare the performance of two programming language even though you can have the same domain context and same algorithm.

2. Operating System & Kernel.

My first touch with Operating System is during my preparation of Graduate Entry Exam at the end of 2003. The textbook we use is Minix version 2 developed by Tanebum.

Later I also read the book "The Design and Implementation of Unix", which is based on Unix V6.

Starting from the autumn of 2006, I become interested in Linux Kernel, I have read the following books. "Linux Kernel Scenario Analysis", which cover 2.4.0 "Understanding the Linux Kernel", "Understanding the Virturl Memory", "The Design and Implementation of Linux Kernel", Robert love.

I like to know the details such how the synchronization are implemented in low level. Actually this kind of knownledge help me understand java concurrency programming better.

I also tried to build kernel by myself, the book "Linux Kernel In a Nutsheel" helps. In order to understand the build process, i learned GNU Make, but I have not gone too far in that direction till now.

3. Oracle Database.

During my training period of my first IT job at Zhongxin, I have the oppotunity to install Oracle 9i database and play with it for a while. Becuase of this previous experience, when I start my second IT job at BLEUM, I am able to take the oppotunity of database maintenance.

Starting from April of 2006, I start to pursue Oracle Certified Professional on Database Administrator. and I got the OCP at July. During this period, I have systematically learned the "Oracle 9i Concept" and "Oracle Programming—Expert on one one", and "Effective Oracle by Design". I am grateful that I have the oppotunity to apply my Oracle Knowldge and Experience in Performance tuning of the J2EE WEB applicaion developed by our team.

The most helpful knowledge about Oracle is the internal mechanism of how all kinds of features are implemented. For example, How is the SQL statement get executed. without such kind of knowledge, you can not tell the difference between a good SQL statement and a bad one, because they may be equivalent from logical point of view. Another tips is about practice and testing. Oracle provides lots of tools to help you get the trace information about low level details. using them to collect information will help you make the right decision.

4. Network Security.

I have security concern in my mind when I first time started to surf the internet, majorly because the rumor about virus and cracker. So I bought a book about the security.

Later when I start to develop WEB applicaiton on J2EE Platform, I had the oppotunity to look deeper into the security mechanism. I master the concept like symmetric encryption. computational infeasibility. In order to understand them, I also need to know more mathematics, especially the number theory.

5. Computer Architecure.

From June of 2007, after I have finished a few important project in tight schedule. I got some time to read the MMIX—The RISC Computer for the new Millennium which I borrowed from Shanghai Library. It is the first time for me to have a close look at the computer architecture of a mordern RISC computer. Honestly I like to such kind of hardware details because I believed in solid fundation. I also bought volume 1 of TAOCP(*The Art of Computer Programming*) but I did not have enough time to finish the reading because I start to prepare for PMP training and Exam from March, 2008.

After I pass the PMP Exam on September 2008, I reread TAOCP Volume 1, Fascicle 1 and I was even inspired to implemnent a simulator for it. Knuth already provided a perfect simulator for it, I just want to practice my Java programming skill by reimplementing it in Java. Till now, I have finished all the non-floating instruction. In order to implement the complex mechanism such as "Register Stack" I have also borrowed "MMIXWare — " and read it carefully to get all the details.

But I have not implemented floating point instruction. Although Java announce it is IEEE-754 compatible, it is actually not easy to implement MMIX's point instruction with Java. I have read Knuth's C code of MMIX simulator, I know how the floating point instruction can be simulated with Integer Arithmatic. Maybe I could have converted the C code to Java, but it sounds boring to me, so the Java Implementation of floating point MMIX instruction is still pending.

Late, I also read "Computer Architecure" and "Hardware Software Interface" to know more about the pipeling, cache, multi-issue.

I also read the book like "See MIPS Run", which

These book mentioned how hardware provides support to morder OS concepts, such memory mapping (Virtual Memory).

6. Algorithm and Algorithm Analysis.

Actually I have spend many effort in learing Algorithm, the first Algorithm book I have read from cover to cover is "Algorithm, Data Structure, and Applicaion—Described in C++". It is a good text book for me as a self-studier, some algorithm, such as LZ compresion algorithm, is so beautiful and impressive that I can still remember them today.

I become interested in Donald E. Knuth's "The Art of Computer Programming" since I first time heard of it on 2002. but I never have the time to go deep into it until June, 2007.

From May 2007, after I come back from San Francisco, one team mates recommends the MIT course "Algorithm Introduction" to me, which can be downloaded from MIT freely. so I can take the course offline and practice my English skill.

I also bought the Fascicle 2, 3, 4 of Volume 4 and downloaded Pre-Fascicle 0a, 0b, 0c, 1a,and 1b. all of them are about "Combinatorial Searching". During the reading of TAOCP, I have written my program to verify my understanding of the algorithm in the book. It provides so many ideas for you to explore, there are many more ideas in the exercises.

7. Regular Expression.

Regular expression is a handy powerful tool. I have the time to go deep into it on May 2008. I learn it as one way to relax my mind when I am prepareing for PMP exam. I majorly refer to "Master Regular Expression", I practiced most of examples described in the book. The most important thing is that I not only start to use it in my daily work, but also help my teammated use it in their work. I also realized that most of tool actually have good build in support of regular expression, they are just there waiting for us to use.

I also have the pleasure to find that my team mates improve their efficiency by using regular expression in their daily work.

8. Summary

I have start to learn all kinds of different things. they are interesting and I enjoy the course of learing. but till now I still did not reveal the way to take effect of those kinds of broad knowledge.