

Shu-Chun Weng

51 Prospect Street
New Haven, CT, 06511

203-843-1528
scweng@gmail.com
<http://scw.tw/en>

Objective

To obtain a summer intern position in software development, especially focusing on programming languages (development tools, frameworks), program analysis, or security.

Education

Pursuing Ph.D. Dept. of Computer Science, Yale University 2009.9 – present
B.S., Double Major in Dept. of Computer Science and Dept. of Mathematics, National Taiwan University
- Major GPA: 4.0/4.0 Full GPA: 3.9/4.0 Rank: 15/142 2003.09 – 2007.06

Experiences

Having part-time worked in several institutions, I always strive to maintain readable and best quality for web services and products of frequent launch and ship schedule.

Engineer Intern, Google Inc. 2010.05 – 2010.08
- Programed in C++ and Python.

Research Assistant under Prof. K. Chen, National Chungchi U. 2006.02 – 2009.08
<http://www.cs.nccu.edu.tw/~chenk>
- Work done mostly in Haskell and Java.
- My project was to implement AspectFun, which is an aspect-oriented functional programming language compiler. In particular I worked on components that perform Cflow (control flow) transformation, monadification, and analysis.
- My research involved developing the theoretical foundation for type-directed program transformation and detection of memory leaks and potential security problems, which resulted in several publications.

System Administrator, PTT Bulletin-Board System 2003.08 – 2009.08
<http://ptt.cc>
- PTT is the largest telnet-based BBS in Taiwan, with more than 15,000 concurrent users online during peak hours every day. The system is built using C/C++ and Perl. I added new features (including an instant message-based human-to-human user service and an online chess playing, viewing, replaying framework supporting go, Chinese chess, five in a row, and reversi), fixed bugs and security problems including information leaking via buffer overflows and bypass intended access restrictions, and reduced memory usage by a factor of 2 by shrinking the memory usage in Linux kernel.
- Persistent hardware, operating system, and library maintenance – we had upgraded the system, found the libraries broken, and patched them more than once.

Part-time Engineer, Miniworld 2007.02 – 2008.12
<http://www.miniworld.com.tw> and <http://www.godgame.com.tw>
- Programed in PHP, Perl, MySQL and shell scripts.
- Maintained and added feature to a legacy codebase, and built from scratch a server-side engine for the new web-based game business.
- Job also involved maintaining and tuning MySQL database and hardware.

Part-time Engineer, Institute for Information Industry 2005.01 – 2007.09
<http://www.iii.org.tw>

- Developed a real-time natural hazard monitoring system using Microsoft .NET framework 1.1, C# and COM.
- Independently completed a deployment system akin to Java Web Start technology and shipped to several clients.

Assistant coach of ACM contest teams

2006 – 2007

Projects

Pugs, an open source project implementing Perl 6 compiler/interpreter

2005-2006

<http://pugscode.org>

- Implemented in Haskell and Perl 5/6.
- Worked on parsing and regular expression support.

C-subset compiler (course project)

2006.06

- The project (about 5,000 lines of code) was written in flex, bison, and C++, and made use of design patterns extensively. I employed a test-driven development model and independently implemented the entire system.
- Compiler supports functions, multi-level pointers, multi-dimensional arrays, array-to-pointer collapsing, type checking, and basic optimizations. Also includes an interpreter (on type-checked and optimized AST) and a three-address assembly generator.

Mini-Pascal compiler (course project)

2007.06

- Written in pure C, lex, and yacc.
- Approximately 2,500 lines of code by a team of two.
- Supports pascal-style nested functions, multi-dimensional arrays, and basic optimizations.
- Targets C--, a three-way assembly language.

MIPS CPU simulator and toolchain (course project)

2005.06

- Written in C++ employing many design patterns.
- Approximately 4,000 lines of code by a team of two.
- Consist of an assembler and a five-stage pipelined CPU simulator, supporting all of the standard MIPS instructions.

Five-stage pipelined CPU in VHDL

2006.01

- Written in VHDL and is able to run on FPGA boards, a small assembler written in Perl included.
- Approximately 1,300 lines of code all by myself.
- A five-stage pipelined CPU with RISC-like instruction set but with some non-standard dual-write instructions. Those complex ones are decoded into reduced ones and takes several cycles to complete – CISC way.

JPEG and MPEG video decoder

2007.06

- Written in C++, using Win32 SDK and multimedia timers, employing a lot of performance tweaks and preprocessor tricks to achieve real-time decoding.
- Approximately 1,200 and 2,600 lines of code respectively. I was the sole developer in both projects.
- Not fully implemented the specifications, but are powerful and efficient enough for real usage.
- Algorithmic optimizations include fast IDCT (Arai et al., 1988) and Huffman decoding (Hashemian, 2004).

Publications

- K. Chen, J.-Y. Lin, **S.-C. Weng**, and S.-C. Khoo. “Designing Aspects for Side-Effect Localization”. In Proceedings of the 2009 ACM SIGPLAN Workshop on Partial Evaluation and Program Manipulation. PEPM '09. ACM, New York, NY, 189-198.

- K. Chen, **S.-C. Weng**, M. Wang, S.-C. Khoo, and C.-H. Chen. “A Compilation Model for Aspect-Oriented Polymorphically Typed Functional Languages”, Symposium of Static Analysis 2007: 34-51.
- K. Chen, **S.-C. Weng**, M. Wang, S.-C. Khoo, and C.-H. Chen. “A compilation model for AspectFun”. Technical report, TR-03-07, National Chengchi University, Taiwan, March 2007.
<http://www.cs.nccu.edu.tw/~chenk/AspectFun/AspectFun-TR.pdf>
- K. Chen, **S.-C. Weng**, M. Wang, S.-C. Khoo, and C.-H. Chen. “Type-Directed Weaving of Aspects for Polymorphically Typed Functional Languages”, Science of Computer Programming. To appear.
- K. Chen, J.-Y. Lin, **S.-C. Weng**, M. Wang, and S.-C. Khoo. “Designing Aspects for Side-Effect Localization”, Higher-Order and Symbolic Computation. To appear.
- A. Aviram, **S.-C. Weng**, S. Hu, B. Ford. “Efficient System-Enforced Deterministic Parallelism”. In The 9th USENIX Symposium on Operating Systems Design and Implementation (OSDI '10). To appear.
- **S.-C. Weng**. “Comparison and Analysis to Extending a Existing Programming Language to Support Generics”. Course report, June 2005. Unpublished.
http://scw.tw/works/hw_report/java-generics.pdf

Honors and Activities

Presidential Award, National Taiwan University	
- 2 times (out of 8), top 5% in class	
Succeeded in Google Summer of Code 2006	2006.5 - 2006.10
- Works on Perl 6 Self-hosting Bootstrap From Perl 5 and Rules	
First Place, National College Software Designing Contest	2004.11
(with C.-H. Hsiao and C.-Y. Lee)	
Third Place, ACM Computer Programming Contest Asia	2003.10
(with C.-H. Hsiao and C.-Y. Lee)	
Silver Award, International Olympiad in Informatics	2002.08
Headman of school swimming team	2005.6 – 2006.6

Skills

- Programming languages: C/C++, Java, C#, Perl, Python, PHP, shell scripts, functional languages (Haskell, Erlang, etc.), Prolog, and can master new languages very quickly.
- Disciplined programmer, relying on version control systems (CVS, SVN, git) heavily, taking documentation and coding style seriously, and paying much attention to writing readable, reliable, and efficient code.
- Strong mathematics background, especially in discrete math and algebra.
- Languages: Chinese, English, and a little Japanese (reading).