

Joshua M. Kwan <joshua.m.kwan@gmail.com>

309 W 111th St, Apt 11, New York, NY 10026
(650) 539-5354

OBJECTIVE

A software engineering position with a focus in top-down design, systems programming, and open source collaboration.

EDUCATION

Bachelor of Science, Electrical Engineering and Computer Science
University of California, Berkeley, CA, December 2008

SKILLS

- *Programming Languages*: C, C++, Python, Perl, Autoconf, Makefile, Bash scripting, Ruby, C#, Java, MIPS assembly, Lisp/Scheme, SQL, HTML, JavaScript, CSS, PHP
- *Operating systems*: Proficient at using and developing on modern Unix (Linux, Solaris, FreeBSD, OS X) and Windows platforms
- *Frameworks, Libraries and Methodologies*: Glib, GTK+, Win32 API, POSIX, cURL, REST, Qt, pthreads, WiX/MSI, Core Foundation, J2EE
- *Source Control Systems*: Git, Perforce, Subversion, CVS
- *Spoken Languages*: Fluent spoken/written English and French, basic spoken/written Japanese.
- Strong system and network administration skills for Linux and a deep understanding of Linux distribution design from the ground up, from kernel to packaging system.

EXPERIENCE

Member of Technical Staff

January 2009 - present

VMware, Inc., Palo Alto, CA

AppBlast (ongoing): Building out MSI installers, Windows configuration utilities, RESTful APIs, and CDS infrastructure (see below) for a new HTML5, browser-based, pluginless, application remot-ing solution that works on all modern browsers, even those on tablet PCs and smartphones. Work involves extensive use of C++ and the Windows API in order to provide functionality.

ThinApp Factory: Developed solutions to automate creation of ThinApp packages (virtualized applications for Windows) on a large scale through use of RSS feeds and a work pool hosted on virtual infrastructure. Helped build a turnkey appliance based on Debian that allows IT administrators to hit the ground running with our software. Using optimizations involving linked-clone virtual machines and snapshots, an admin can create Office 2010 as a ThinApp package within 90 minutes of deployment.

Component Download Service (CDS): With my team, designed and wrote both client, server, and consumer API for a generic web updater framework for all VMware products, implemented in C using many open source libraries (cURL, c-ares, Glib, and OpenSSL.) The framework leverages platform-specific installer backends on Windows, Linux, and OS X, while providing a single unified API. This project made its debut in VMware Workstation 7.0 and VMware Fusion 3.0.

Network Manager

August 2007 - January 2008

Berkeley Student Cooperative, Berkeley, CA

Managed internal network infrastructure, firewall, routing software, Internet connection, and file servers for a house of 120 residents. Responsibilities include tasks from low-level (reinstalling broken Ethernet ports) to high-level (managing network QoS and traffic-shaping using Linux software.)

Intern
VMware, Inc., Palo Alto, CA

Summer 2007

Linux Easy Install: A feature for VMware products that enables people to simply insert an installation CD for a Linux distribution and instantly create a virtual machine containing that operating system without any hassle. Instead of forcing the user to download a pre-baked virtual machine image, the distribution's unattended installation mechanisms are leveraged to allow quick deployment. Ubuntu and Red Hat-based distributions were the focus of this effort, and the system now supports SuSE Linux as well.

Intern
Accenture Technology Labs, Palo Alto, CA

Summer 2006

Business Event Advisor: Project for reading RSS feeds, parsing them, and coming to conclusions using NLP algorithms applied to information in the feeds. Backend development in Python, native frontend development in Visual Basic / C# .NET, and development of a completely new portal-based, webservice-backed frontend in Java.

Intern
Sun Microsystems Labs, Menlo Park, CA

Summer 2005

The Magic Wall: Windows/Linux development in C++ to use data from a rotating laser to turn a 'projector wall' of networked Linux computers into a touchscreen.

Intern
Rackable Systems, Inc. (now SGI, Inc.), Milpitas, CA

Summer 2004

Designed and performed benchmark suites to measure the performance of high end server hardware on the Linux platform. Also wrote a database-backed system for tracking such results and correlating them to system specifications.

Volunteer Developer
The Debian Project (www.debian.org)

2003 - present

Putting together software packages for the Debian GNU/Linux distribution, involving knowledge of Make, shell, C, and Perl, and understanding OS infrastructure written in those languages. Also collaborated in writing the Debian Installer's network interface management code.

REFERENCES

Available upon request.