

# Alexander Fenster

fenster@fenster.name, (425) 435-8386

skype: fenster\_home, gtalk: alexander.fenster

Live in Bellevue, WA; eligible to work in the United States

## Employment history

- 2014–present** **Microsoft**, Bellevue, WA (previously Moscow, Russia). Software engineer II, Bing. I work on ObjectStore, our internal distributed key-value storage solution. The most important features I worked on were: for backend, a set of new REST APIs that enabled easier access to the data for some partners; the concept and implementation of pluggable storage engines that allowed us to use open source RocksDB as an alternative storage engine for our solution. For frontend, I implemented a bunch of debugging tools allowing our users to examine their data and simplifying development process for them.
- 2012–2014** **Parallels**, Moscow, Russia. Senior software engineer, third-line support team. My responsibilities included resolving issues related to the company's Linux kernel-based virtualization products (container virtualization software: Virtuozzo and OpenVZ), distributed storage solution (Parallels Cloud Storage) and userspace management tools (both Linux and Windows). Given a Linux kernel crash dump or a customer's bug report, I looked for the possible reasons of the crash and either provided a quick fix or reported the issue to the kernel team. My most used tools were `gdb` and `crash` utility.
- 2006–2012** **Intel**, Novosibirsk, Russia. Senior software engineer, Intel Compiler and Languages lab. On this position I implemented and maintained the software that performed nightly builds and testing of Intel compiler on various architectures. Most of those tools were written in Perl, and one of the most challenging parts of the job was to make them work on Linux, Windows, and OS X with little or no changes to the code. My responsibilities also included analysis of compiler build failures, and it was another challenging but exciting part: compiler build has several stages, including a self-build (compiler compiled by the same compiler), and finding a reason of the build failure may take some time.
- 2003–2006** **SWsoft**, Novosibirsk, Russia. Support engineer and trainer. I supported Linux hosting automation and virtualization solutions (Plesk, Virtuozzo): fixed customers' problems bringing their production servers back to life, developed workarounds for software issues. Also, I provided live trainings for customers and for other team members about both our software and Linux in general.

## Education

- 1999–2004** **Novosibirsk State University**, Mechanics and Mathematics department.

## Skills and Experience

I have fourteen years of experience in software engineering and technical support. Software development is something I like doing not only in the office but also in my free time. Just like any person pretending to be a good software engineer, I can (and like to) write in C and C++, but of course I can use a bunch of other programming languages (just to list some: loved Perl in the good old days when it was popular, used Python for small scripting, can write some JavaScript, both client- and server-side, used C# a lot after I moved to Windows world). I understand and like functional programming, and it sometimes helps to write better code in other languages.

Learning new languages is fun, so if I need to write a program in some language, I go ahead and write the code. Languages, operating systems, and development environments are not that important, and it's always cool to change the way how you work and to learn new things—that's why I started working for Microsoft after many years in Linux. Last but not least, I lectured on various programming subjects (mostly related to C programming and algorithms) in Novosibirsk university: teaching students is exciting and makes you *really* understand all the things you are talking about. It also makes you learn  $\text{\TeX}$ .

I like debugging, and I like to understand how programs work. To fix any complicated problem with software you need to be familiar with the underlying world, that's why—just a few examples—I know how network works, can act both as a client and a server (using netcat) for most major protocols (HTTP, SMTP, POP3, whatever), can write SQL queries, understand how compilers do their job, and so on.

I'm looking for a really challenging job that will allow me to apply what I already know about software engineering, and to learn even more cool new things.