# Alexander Magola

**Location:** Russia, Novosibirsk | **Born:** 1978

[Email](mailto:magola.a.s@gmail.com) | [LinkedIn](https://www.linkedin.com/in/alexander-magola) | [GitHub](https://github.com/pustotnik) | [GitLab](https://gitlab.com/pustotnik) | [BitBucket](https://bitbucket.org/pustotnik/)

[pdf version](https://pustotnik.github.io/curriculum-vitae/alexander-magola.pdf) • [html version](https://pustotnik.github.io/curriculum-vitae/) • [docx version](https://pustotnik.github.io/curriculum-vitae/alexander-magola.docx)

## About me

A multi-skilled Linux software engineer and team leader with 20 years of experience. As a self-starter I always pay attention to the quality and performance (when necessary) of a code/product and prefer to use Linux both at work and home. I like (and have a lot of experience) designing/improving project architecture and conducting research. I prefer simple but flexible solutions when possible. And it is important for me to find and implement the right solution.

Interested in many things, including: Linux and relative technologies, distributed/networking/sync/etc technologies/systems/applications/etc, high load, virtual things, embedded systems, automation, open source, digital privacy, cyber security, etc.

## Languages

* **Russian**: Native speaker
* **English**: Intermediate

## Main Technical Skills (short list)

* **Operating Systems**: GNU/Linux, MS Windows
* **Programming languages**: C/C++, Python, bash (basic skill)
* **Version control**: Git, Mercurial
* **Development Tools**: GCC, Clang, KVM/QEMU, Docker, Waf, MSVC
* **Technologies**: Multithreading, IPC, Shared memory, Networking, RPC
* **Testing/CI**: GTest/GMock, GCC/Clang analyzers/sanitizers, Python unittest/pytest, Jenkins CI, TravisCI, GitHub Actions

## Technical Experience

Jun 2013 — present time

**Open Source Developer**

Developing some open source projects: [GitHub](https://github.com/pustotnik), [GitLab](https://gitlab.com/pustotnik)

Most active and big project is own build system [ZenMake](https://github.com/pustotnik/zenmake) for C/C++ and some other languages (Fortran, D) supporting GNU/Linux, MacOS and MS Windows. Worked on making easy to use and flexible config files without need to know some programming language. Made ready to use system with detailed [documentation](https://zenmake.readthedocs.io/) and many [examples](https://github.com/pustotnik/zenmake/tree/master/demos). However a lot of features were not implemented yet. The unit/functional tests are regularly run in CI and covers more than 80% of the code. This system is based on [Waf](https://waf.io/) but it is used as a framework/engine here.

* ***Programming languages***: Python, bash
* ***Tools and technologies***: GCC, CLang, MSVC, Waf, Python unittest/pytest, pylint, python coverage, KVM/QEMU, Docker, TravisCI, GitHub Actions, Markdown, reStructuredText, Sphinx (documentation generator)
* ***Version control***: Git, Mercurial
* ***Issue tracker***: GitHub Issues, GitLab Issue Tracker
* ***Operating Systems***: GNU/Linux, MS Windows (tests only), MacOS (only with TravisCI and GitHub Actions)

Apr 2012 — Feb 2017

**Linux Team Leader** at [Signatec](https://www.signatec.ru/) [Russia, Novosibirsk]

Managed a long-term Linux project of legal network surveillance with a team of 5 C/C++ developers and 1-2 QA testers. This project had strong requirements for performance and stability. Designed many parts of the architecture. Developed some important/critical code. Reviewed other code of the project. Introduced several things to improve the quality of the project: GTest, gcov, LLVM Clang static analyzer, cppcheck, AddressSanitizer/LeakSanitizer (GCC/Clang), -Werror. Helped the team improve some technical knowledge. Conducted daily meetings to review current state of the project. Made many important decisions including resolving problem of memory fragmentation.

Managed/improved my team workflow including:

* Jenkins CI server: installed/configured, made additional scripts (bash, python)
* migration from SVN to Mercurial:
  + installed/configured selected by myself SCM-Manager as a central repository server for internal use.
  + made short manuals for my team
* organizing small cluster of KVM/QEMU servers (Proxmox VE + simple backups of images) on 2 physical servers with roles: http proxy, SVN, hg/git server, Redmine, Jenkins CI (main + several agents)
* set of scripts to create base Linux firmware image based on Gentoo with ability to install on any computer

**Experience in brief**:

* ***Programming languages***: C++ (03, 11), Python, bash
* ***Tools and technologies***: GCC, CLang, Intel TBB (mostly flow graph, spin mutexes, malloc), MessagePack, ZeroMQ, Boost, GTest/GMock, Google Benchmark, Jenkins CI, Clang static analyzer, cppcheck, AddressSanitizer/LeakSanitizer (GCC/Clang), KVM/QEMU
* ***Databases***: LMDB
* ***Version control***: Mercurial, SVN
* ***Issue tracker***: Redmine
* ***Operating Systems***: GNU/Linux (Gentoo, Debian)

Dec 2009 — Apr 2012

**C++ Linux Software Engineer** at [Signatec](https://www.signatec.ru/) [Russia, Novosibirsk]

Was a main developer and one of the architects of a system of distributed services DSS (for C++ SOA solutions on Linux, similar to WCF in .NET). The system allowed different services to communicate with each other using TCP/UDP/Unix (Local IPC) sockets and Shared memory by configuring files in XML/JSON formats. (C++, Linux)

Managed sub project of web interface for DSS with one Python developer.

Was involved in development of some base Linux/POSIX C++ libraries for local needs like networking, date/time, filesystem, threading, etc. (C++, Linux)

Initiated and installed/configured Redmine as a issue tracker for use in our team. Helped the team learn how to use it. (Debian Linux, Redmine, Nginx)

Made useful build system BDS based on Waf for C/C++ projects on Linux. This system was successfully used for local C/C++ projects all the time I worked for this company. (Python, Waf, Linux)

* ***Programming languages***: C++, Python
* ***Tools and technologies***: GCC, Waf, TCP/UDP/Unix sockets, Shared memory
* ***Version control***: SVN
* ***Issue tracker***: Redmine
* ***Operating Systems***: GNU/Linux (Gentoo, Debian)

Nov 2007 — Nov 2009

**Project Manager/Team Leader** at [Internet Service](https://inetss.ru/) [Russia, Novosibirsk]

Designed architecture and developed an experimental real-time web search system. (Scala, Jabber/XMPP)

Made research with some experiments for potential project of a web storage for user photos/pictures. (Python, Java, Hadoop/Hbase/HDFS, Lucene, PostgreSQL, Thrift)

Designed architecture and developed part of the back-end of a distributed DNS system. (Python, Twisted, BIND, BerkeleyDB with replications, PostgreSQL, AMQP, RabbitMQ)

Managed web searching project Assista (similar to google search) with team of 2-5 front-end/back-end developers. Improved/developed the project core search engine solution based on Sphinx (C++ open source search engine). Carried out some sysadmin tasks for the project remote computer cluster with more than 100 servers (CentOS) using bash/ssh and own perl scripts. Improved the project architecture. (C++, Perl, MySQL, memcached, Sphinx, bash/ssh) This project was one of ambitious projects of the main customer of that company. It was my first acquaintance with distributed systems.

* ***Programming languages***: C, C++, Java, Perl, Python, Scala
* ***Tools and technologies***: MSVS, GCC, Sphinx (search engine), bash, ssh, Twisted (Python network framework), memcached, BIND, AMQP, RabbitMQ, Thrift
* ***Databases***: MySQL, PostgreSQL, BerkeleyDB
* ***Version control***: SVN
* ***Issue tracker***: company’s own internal web application
* ***Operating Systems***: MS Windows, GNU/Linux (Gentoo, CentOS)

Nov 2006 — Oct 2007

**Software Developer** at [Internet Service](https://inetss.ru/) [Russia, Novosibirsk]

Participated in startup project of drawing web application with ability of recognition of hand-drawn shapes. Made experimental module of recognition (C++, OpenCV, Linux).

Was in a team of development of multi chat client-server application Avago supported audio and video. Was involved in development as for client side (C++, WTL, FFmpeg, MS Windows) as for server side (Java, Red5, MySQL, Linux) of the application.

* ***Programming languages***: C++, Java
* ***Tools and technologies***: MS Visual C++, WTL, JVM/JRE/JDK, FFmpeg, Red5 (Java media server), Jetty, MySQL, OpenCV
* ***Version control***: SVN
* ***Issue tracker***: company’s own internal web application
* ***Operating Systems***: MS Windows, Gentoo Linux

Apr 2005 — Oct 2006

**Software Developer** at [Technodesign](www.technodesign.ru) [Russia, Komsomolsk-on-Amur]

Participated in improvement/expanding of web site for local billing system. (Perl, MySQL, Linux)

Made base of a project for the file storage system for web. Participated in the architecture design and implementation of the front-end part. (Perl, Catalyst, MySQL, Linux)

Made a client application for a HotSpot system. (C++, XML-RPC, MS Windows)

Developed a plug-in for FreeRADIUS for authentication, authorization and accounting internet sessions from pppd and VoIP calls from GnuGK for local billing system. (C, MySQL, Linux)

* ***Programming languages***: Perl, C, C++,
* ***Tools and technologies***: GCC, make, MySQL, H323, HTML, Catalyst (Perl MVC Web Framework), FreeRADIUS, VoIP (GnuGK), CVS, MS Visual C++, STL, WTL, XML-RPC
* ***Operating Systems***: Slackware Linux, MS Windows

Jan 2001 — Jun 2003

**Software Developer** at Komsomolsk-on-Amur State Technical University ([KnASTU](https://knastu.ru/))

I was a postgraduate in KnASTU and had a task to develop a calculation program of stress-strain state of solid materials with cracks based on boundary element method. This project was part of my PhD thesis.

* ***Tools and technologies***: MS Visual C++, MFC, STL, Win32 API

## Education

Sep 1995 — Jun 2000

Komsomolsk-on-Amur State Technical University ([KnASTU](https://knastu.ru/))

* ***Degree***: Master’s degree in applied mathematics (Specialist degree)
* ***Tools and technologies***: Turbo Pascal, Turbo C, Borland C++ Builder, MS VB/VBA, MS Visual C++, MFC, STL, Win32 API

Last modified: 2021-07-16