

Kjell Winblad, Ph.D. | <https://winsh.me> (Personal) <https://dupwin.se> (Company)
mail: kjellwinblad@gmail.com tel: (0046) 73 053 50 89

Summary



Kjell is a software architect and full-stack software engineer with over 15 years of experience in software development and research. From completing a Ph.D. degree focusing on synchronization primitives and concurrent data structures, he has got a deep understanding of low-level and multicore optimizations. He also loves high-level system design and programming in high-level languages like Scala, Python, Java, Erlang, JavaScript, and Golang.

From doing professional work in various fields within the IT industry, he has got a broad knowledge of many different kinds of frameworks and programming languages (low-level systems programming, object-oriented programming, functional programming, testing methods, etc.). He has been a full-stack web developer, worked on large telecom server products, and made significant contributions to the Erlang virtual machine (BEAM).

As a person, Kjell is responsible, curious, and persevering. He loves to challenge himself, learn new things, and get a deep understating of how things work. He got hooked on programming while in his teen years and has since then never stopped trying to improve his coding and software design abilities.

Interesting future customer project areas are:

- System Design and Architecture
- System Programming
- Multicore Optimization
- Performance Analysis and Improvements
- Tools and Services
- Full-Stack Web Development
- Big Data (Data Mining and Data Analytics)
- Data Visualization
- Data Science
- Compilers
- Game Programming
- End-to-end software design and implementation

AREA OF COMPETENCE

Specialty Area	<ul style="list-style-type: none">• System Design and Architecture• Profency in many programming languages and frameworks• Parallel programming• Multicore optimizations• Debugging of C and C++ programs (gdb, rr, memory/address sanitizer)• System Performance Analysis and Optimizations• Techincal Documentation
Operating systems	Linux, MacOS and Windows
Languages, frameworks & Databases	Python, Erlang, Java, C, C++, Scala, Golang, JavaScript (React, jQuery, Lodash), Objective-C, NodeJS, SQL, MongoDB, Sandard ML
CI/CD & Cloud	Docker, CI/CD - Jenkins
Tools and Environments	Unix, Git, Emacs, Visual Studio Code, Eclipse, Slack, Microsoft Teams
Software processes	Agile, SCRUM
Architectures	Micro-services Architecture, Actor Based Systems, REST APIs
Methodologies	Test Driven Develoment, Property Based Testing
Machine Learning	Arificial Neural Networks, Suport Vector Machines, Natural Language Processing
Learning on-going	Data science, Game programming

PROFESSIONAL EXPERIENCE

- 2022 Mars - Present**
(Contract)
- Software Engineer at EMQX**
Role: Software Engineer with focus on
Technology stack: Erlang, C, Rebar3, the MQTT protocol, EMQX Messaging Platform
Tasks and Achievements:
- [Improvements](https://stedolan.github.io) to Erlang NIF library for executing jq (<https://stedolan.github.io>) filter programs
 - [An LRU Cache for jq filter programs](#)
- 2021 Nov - 2022 Feb**
(3 months contract)
- Tools Team at Imsys AB**
Role: LLDB plugin developer and bug fixer
Technology stack: LLVM, LLDB, C++, Python, Robot Framework
Tasks and Achievements:
- Made most common debugging functionalities work (source level stepping, breakpoints and backtraces)
 - Implemented automated tests for the debugger with Robot Framework and LLDB's Python scripting support
 - Made a guide (including a video tutorial) describing how to debug programs running on Imsys hardware/simulator inside VS Code
- 2019 Jan - 2021 Nov**
(~6 months parental leave)
- Erlang/OTP team at Ericsson**
Role: Software Engineer in the Erlang/OTP Virtual Machine Team
Technology stack: Erlang, C, Unix, Make
Tasks and Achievements:
- New scalable back-end data structure for ETS ordreded_set. See [PR-1952](#) and [blog post](#).
 - Scalable counters for ETS tables. See [PR-2190](#), [PR-2229](#) and [blog post](#).
 - Design and implementation of source-to-source transformation tool for automatically creating yielding C code. This tool is used to make Erlang systems more responsive and to make the scheduling of Erlang processes more fare. See [PR-2396](#).
 - Design and implementation of a message sending optimizations that makes message sending between Erlang processes perform up to 300 times better than before when multiple processes send messages to a single process. Erlang is a concurrent programming language where the core communication primitive is message sending. See [PR-5020](#).
 - Test driven development of new features and functionalities
 - Handling support errands from our customers
 - Handling of pull requests and issue reports coming to our GitHub repository from open source Erlang users
 - Internal investigations of new features and processes
 - Internal presentations and discussions about new optimizations and language features
 - Usage of Docker for testing and debugging
- 2018 Sep -2019 Jan**
- Ericsson, Baseband Software Developer**
Role: Software Engineer
Technology Stack: C, Baseband telecom software
Tasks and Achievements:
- Implementation of automatic tests and experiments for new system feature
 - Ported a large code base to a new C compiler
 - Fixed issues reported by a new static analysis tool for C code
- 2012 Aug - 2018 jun**
(8 months parental leave)
- Multicore Synchronization and Concurrent Data Structure Researcher at Uppsala University (Ph.D. Candidate)**
Roles: Teacher, Researcher, Scientific Writer, Programmer of tools for experimentation, teaching and research
Technology Stack: Python, Java, Scala, Docker, Erlang, C, Standard ML and Matlab, NodeJS, Express, PHP
Tasks and Achievements:
- Invented, implemented and analyzed lock-based and lock-free concurrent data
-

structures for ordered sets and maps ([the contention adapting search tree](#) and [the lock-free contention adapting search tree](#)). The data structures performed better than the state-of-art in many scenarios at the time of publishing. One of the data structures has been integrated into Erlang's in-memory database ETS.

- Invented, implemented and analyzed a new concurrent adaptive priority queue called [the contention avoiding concurrent priority queue](#). The data structures performed better than the state of art in many scenarios at the time of publishing.
- Co-invented, implemented and analyzed a new synchronization technique called [QD Locking](#). The locking primitive performs extremely well under heavy load, is simple, and easy to extend with new functionalities such as NUMA-awareness and read-only critical sections.
- Teaching assistant for the courses:
 - Data Structures and Algorithms II (4 instances). Corrected assignments, handled computer lab sessions and implemented (NodeJS and Docker) an automatic web-based testing system that the students could use to check their assignments before submission.
 - Functional Programming (Standard ML, 4 instances). Helped the students in the computer lab, wrote tests for grading and corrected assignments.
 - Semantics of Programming Languages (3 instances). Corrected assignments and gave assignment feedback.
 - Machine Learning (1 instance). Corrected assignments and helped students in computer labs.
- Member of the group responsible for maintaining the IT-department's website (1 year). Programming and bug fixing in PHP.
- Presented publications in five different international conferences and did multiple internal presentations at Uppsala University.
- Main author of seven different peer-reviewed papers and co-author of three more.
- Completed graduate level courses about: testing of concurrent and parallel software, parallel and concurrent programming, concurrent data structures, advanced data structures, technical writing, compiler design and implementation (implemented a machine code generating compiler for a subset of C), and teaching.
- Successfully defended Ph.D. thesis titled [Dynamic Adaptations of Synchronization Granularity in Concurrent Data Structures](#)

2012 Jan-
2012 Aug

Tacton AB, Full Stack Developer

Role: Software Engineer and Consultant

Technology Stack: Java (Backend), JavaScript, jQuery (Frontend)

Tasks and Achievements:

- Implemented and tested customer specializations for Tacton's configuration and billing software.
- Java and JavaScript programming
- Design of user interface components (HTML, JavaScript)
- Bug fixing

2007 Aug-
2011 Aug

Digital Route AB, Java Software Engineer

Role: Java Software Engineer for Digital Route's mediation system

Technology Stack: Java, Java Swing, Python, Android SDK, Apache Web Services

Taks and Achievements:

- Bug fixes and maintenance
- New compression module for Digital Route's mediation system
- Significant contribution to t Digital Route's web services module
- New graphical user interface components (Swing)
- Performance analysis and improvements
- Android Application to monitor and interact with Digital Route's software
- Design and implementation of automated tests (Python)

(leave of
absence
between 2008
Jun and 2010
Jun for master
studies)

2009 Aug-
2011 may

Zyked, Software Consultant

Technology Stack: Objective-C, SQLite

Role: Implementation of an iPhone Running App called [Zyked](#) that was released on the

(Part Time) Apple iPhone app store. Zyked was a game where users could create tracks using the iPhone's GPS. The tracks could be uploaded to the server component so other users could compete on them.

Tasks and Achievements:

- Objective-C implementation of the app
- Interaction with the REST API for the server component
- Managing persistent data using SQLite

2005 Aug - 2006 Jun **Teaching Assistant at Umeå University, Sweden**
Role: Teaching assistant for three different courses while studying

Tasks and Achievements

- (20%)
- Corrected assignments and was responsible for computer lab sessions for the following courses
 - o Object-Oriented Programming with Java
 - o Fundamentals of Computer Science
 - o Functional Programming

EDUCATION

- 2004-2007 Bachelor FScience, Computer Science – Umeå University, Sweden
[Bachelor Thesis](#), [Courses and grades](#)
- 2008-2010 Master of Science, Computer Science – Uppsala University, Sweden
[Master Thesis](#), [Courses and grades](#)
- 2012-2018 Doctor of Philosophy, Computer Science – Uppsala University, Sweden
[Ph.D. Thesis](#)

PERSONAL INFORMATION

Citizenship Swedish
Date of Birth 1985-10-25
Languages Swedish(Native), English (Fluent)

Raw Enumeration of Skills/Known Tools, Frameworks

Architecture;

Artificial Intelligence;

Agile software development;

Algorithms and Data Structures;

Backend;

Bash shell, scripting;

Big Data;

C++;

Cloud Computing;

Concurrency;

Data Mining;

Data Modelling;

Databases;

Distributed computing;

Erlang;
Express Web Framework;
Docker;
Git, GitHub;
Golang;
Graphical User Interfaces;
HTML;
Java;
JavaScript;
Jenkins;
JSON;
Linux;
Microservices;
MongoDB;
Multithreading, Concurrency;
MySQL;
Network programming;
Node.js;
NoSQL;
Objective C;
OOP, Object-Oriented Programming;
OpenGL,
Parallel, multithreaded programming;
Python;
Scala;
Software Engineering;
SQL;
SQLite;
SSH secure shell;
System Security;
Test-Driven Software Development;
UML;
Unit Testing;
Web Services;

Web Sockets;

XML;

