

## Kjell Winblad, Ph.D.

web: <https://winsh.me>, (Personal), <https://dupwin.se> (Company),

mail: [kjellwinblad@gmail.com](mailto:kjellwinblad@gmail.com) tel: (0046) 73 053 50 89

### Summary

---



Kjell is a software engineer with over 15 years of experience in software development and research. With a Ph.D. focusing on low-level synchronization primitives and concurrent data structures, along with professional experience working with the Erlang VM team at Ericsson, he has acquired deep expertise in low-level programming in C and multicore optimizations. Kjell also has substantial professional experience in high-level system design and programming, using languages such as Python, Erlang, Java, and JavaScript.

Through professional roles in various fields within the IT industry, he has gained broad knowledge across multiple frameworks and programming paradigms, including low-level systems programming in C, object-oriented programming in Python, Java, Scala, and C++, functional programming in Erlang, database systems, and testing methodologies. He has made significant contributions to the Erlang virtual machine (implemented in C), worked on large-scale telecom and IoT products (using Erlang, Java, and C), conducted embedded systems development (Zephyr), and served as a full-stack web developer (Python, Java, and JavaScript). In recent years, he has also become proficient in leveraging newly developed AI tools to enhance productivity.

As a person, Kjell is responsible, curious, and persevering. He enjoys challenging himself, learning new things, and gaining a deep understanding of how systems function. He became fascinated by programming during his teenage years and has continuously strived to improve his coding and software design skills ever since.

Interesting future customer project areas are:

- System Programming
- System Design and Architecture
- Embedded Systems
- Multicore Optimization
- Performance Analysis and Improvements
- AI-tools and APIs
- Tools and Services
- Full-Stack Web Development
- Data Visualization
- Data Science
- Compilers
- Game Programming
- End-to-end software design and implementation

### AREA OF COMPETENCE

---

<b>Specialty Area</b>	<ul style="list-style-type: none"><li>• System Design and Architecture</li><li>• Profency in many programming languages and frameworks</li><li>• Parallel programming</li><li>• Multicore optimizations</li><li>• Debugging of C and C++ programs (gdb, rr, memory/address sanitizer)</li><li>• System Performance Analysis and Optimization</li><li>• AI-tools and APIs</li></ul>
<b>Operating systems</b>	Linux, MacOS and Windows
<b>Languages, frameworks &amp; Databases</b>	Python, C, Erlang, Java, Scala, Golang, JavaScript (React etc), C++, NodeJS, SQL, MongoDB, Sandard ML, Flask, SQL
<b>CI/CD &amp; Cloud</b>	Docker, CI/CD – Jenkins
<b>Tools and Environments</b>	Linux, Zephyr, Git, Emacs, Vim, PyCharm, Cursor, Visual Studio Code, Eclipse, Slack, Jira, Jenkins, Microsoft Teams etc
<b>Software processes</b>	Agile, SCRUM
<b>Architectures</b>	Micro-services Architecture, Actor Based Systems, REST APIs, Object-oriented
<b>Methodologies</b>	Test Driven Develoment, Property Based Testing
<b>Machine Learning</b>	Arificial Neural Networks, Suport Vector Machines, Natural Language Processing
<b>Learning on-going</b>	Electronics, Large Language Models and other AI tools

---

---

## PROFESSIONAL EXPERIENCE

---

2025 April -  
2025 Oct.

(6 months  
contract)

### Software Engineer ([Xylem Water Solutions](#))

**Role:** Software Engineer, maintaining and improving pump control software

**Technology stack:** C, Python, Zephyr, Yocto, CMake

**Tasks and Achievements:**

- Improved build times and development tools
- Improved developer documentation
- Refactorings in C code for better maintainability (new LED handling API)
- Automated Software Bill of Materials (SBOM) generation
- Fixed numerous bugs in C and Python codebase

2024 Sep. -  
2025 April

### Founder and Software Engineer ([PingisKompis.se](#) / [PingFriend.com](#))

**Role:** Developer of a new platform to make it easy for table tennis players to connect

**Technology stack:** Python, Flask, MongoDB, JavaScript, React, AI APIs and tools

**Tasks and Achievements:**

- Developed and launched a web-based platform that connects table tennis players and clubs from scratch:
  - Player and club discovery features:
    - Location based advertisements
    - Messaging system
  - Multilingual support:
    - Automatic translation for 25 languages (easily extendable)
    - Forum with dedicated sections for all supported languages
  - Email-based and OAuth authentication
  - Integrated Stripe for future monetization and payment processing
  - SEO and digital marketing to drive traffic and user engagement
  - Infrastructure for automatic multi-node deployment using Docker Swarm

2022 Mars -  
2024 August

(Contract)

### Software Engineer at [EMQX](#)

**Role:** Software Engineer with focus on MQTT, Erlang and NIF extensions written in C

**Technology stack:** Erlang, C, Rebar3, the MQTT protocol, EMQX Messaging Platform

**Tasks and Achievements:**

- [Improvements](#) to Erlang's NIF library (C and Erlang programming) for executing jq (<https://stedolan.github.io>) filter programs
- [An LRU Cache for jq filter programs](#) (C programming)
- [SASL Authentication for Erlang's Kafka library Brod](#) (C and Erlang programming)
- [Many bug fixes and improvements to EMQX's MQTT message broker](#)
- [Added support for MongoDB 5.1+ in an Erlang MongoDB driver](#)
- [Large refactoring of EMQX bridges](#) (to databases and streaming software like Apache Kafka, AWS Kinesis, RabbitMQ, Oracle, MySQL, PostgreSQL, MongoDB, Cassandra, and many others) so they are split into an action component and a connector component (that can be shared between multiple actions)
- [Tracing support for EMQX's rule engine](#)
- [Documentation writing](#)

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:** C, Erlang, Unix, Make

**Tasks and Achievements:**

- New scalable back-end data structure for ETS ordreded\_set (C programming). See [PR-1952](#) and [blog post](#).
  - Scalable counters for ETS tables. See [PR-2190](#), [PR-2229](#) and [blog post](#) (C programming).
-

- Design and implementation of source-to-source transformation tool for automatically creating yielding C code (C programming). 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 (C programming). See [PR-5020](#) and [blog post](#).
- Test driven development of new features and functionalities (Erlang and C programming)
- 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:** C, Python, Java, Scala, Docker, Erlang, 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 (C programming).
- 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 (C programming).
- Co-invented, implemented and analyzed a new synchronization technique called [QD Locking](#) (C programming). 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
  - 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:**

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

- 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)

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

(Part Time)

**Taks 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:  
Object-Oriented Programming with Java, Fundamentals of Computer Science and Functional Programming

## EDUCATION

---

2004-2007	Bachelor Science, Computer Science – Umeå University, Sweden <a href="#">Bachelor Thesis</a> , <a href="#">Courses and grades</a>
2008-2010	Master of Science, Computer Science – Uppsala University, Sweden <a href="#">Master Thesis</a> , <a href="#">Courses and grades</a>
2012-2018	Doctor of Philosophy, Computer Science – Uppsala University, Sweden <a href="#">Ph.D. Thesis</a>

## PERSONAL INFORMATION

---

<b>Citizenship</b>	Swedish
<b>Date of Birth</b>	1985-10-25
<b>Languages</b>	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;

C++;

Cloud Computing;

Concurrency;

Data Mining;

Data Modelling;

Databases;

Distributed computing;

Erlang;

Express Web Framework;

Docker, Docker Swarm;

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;  
XML;  
Zephyr;  
Yocto Linux;  
Flask;  
React;  
PyTest;

---