MAREK MARECKI, inż. (BEng)

PERSONAL INFORMATION

birthplace and date	*Poland*, 22 March 1995
education	inżynier (Graduated software engineer)
email	contact@marecki.me
web	http://www.marecki.me
	https://www.linkedin.com/in/marekjm/
	https://github.com/marekjm
phone	+43.66499846260

GOAL

Deliver robust, reliable, and resilient durable software on LINUX, POSIX, *nix-like*, and real-time platforms.

STRENGTHS

Software design, development, and reengineering. Concurrent software based on either threads or processes. Technical and documentation writing.

WORK EXPERIENCE

Feb 2021—Present   Senior Software Engineer, R&D   Vienna, Austria

ADVA Optical Networking

Prepared a Docker-based environment for the network simulator to run in (and wrote an accompanying documentation). Extended an internal tool used to manage a source tree comprised of multiple Git repositories. Continued maintenance and development of the simulator system.
Key tech: Docker, Linux, GNU Make, C++17, Tcl, Python 3

usecase: migrated the codebase from C++11 to C++17 to take advantage of newer language features and improved standard library (eg, `std::variant`)

usecase: created a Docker image to be used as a platform for running the simulator, based on a root filesystem supplied in a tar archive

usecase: extended an internal tool used to manage a source tree consisting of multiple Git and SVN repositories, to allow automated status reports and command execution

Apr 2018—Jan 2021   Software Engineer, R&D   Gdynia, Poland

ADVA Optical Networking

Modernised and maintained a network simulation system, delivering bug fixes and features across a range of releases. Wrote user and technical documentation for the system from scratch.
Key tech: Git, GDB, Linux, C++11, Tcl, C99, Python 3

usecase: migrated the codebase from C++98 to C++11 to take advantage of newer language features and improved standard library (eg, `std::unique_ptr`)

usecase: reengineered startup and shutdown procedures of the simulator and simulated network nodes using Linux's `PR_SET_CHILD_SUBREAPER` feature of the `prctl(2)` system call to enforce a strictly hierarchical process life cycle

usecase: added a "distress signal" mechanism (using POSIX signals) to processes running as part of a simulated network node to avoid simulation lockups due to differences in behaviour between native (PowerPC) and simulation (x86) platforms

usecase: implemented automatic process resurrection to increase resilience (through self-healing by restart) of the simulation in presence of software errors

usecase: enhanced monitoring capabilities of the simulation controller to detect simulation lockups and node failures

usecase: rewritten simulator's core I/O loop in a non-blocking style, allowing the communication protocol to include expedited messages to maintain communication during simulation lockups

usecase: separated the simulator into server (a daemon running the core simulation) and client (running user interfaces) processes to increase fault tolerance and allow user interface reloads without disturbing the simulation

usecase: reengineered the simulator to allow mutiple simultaneous, concurrent client connections to the core daemon, and allow the clients to connect and disconnect dynamically

usecase: prepared user and technical documentation for the simulator

Oct 2019—Present Instructor, academic teacher Gdańsk, Poland

Polish-Japansese Academy of Information Technology
Delivering "Programming fundamentals (in C++)" and "Users' introduction to Linux" courses. I delivered lectures and workshops, and coordinated work of other tearchers.
Key tech: Git, Linux, BASH, C++14

Dec 2017—Mar 2018 C++ developer Gdynia, Poland

OBR CTM S.A.
Implemented software layers of the STANAG 5066 military HF radio communications stack, based on the protocol specification supplied by the employer. The hardware was designed by a separate team and was exposed through a standard non-blocking file descriptor interface.
Key tech: Git, Linux, C++11

usecase: analysed the protocol description to design a suitable software architecture, with emphasis on robustness

usecase: implemented various parts of the protocol in software, in cooperation with a dedicated testing team, and delegated some of the easier tasks to junior programmers

Jan 2015—Nov 2017 Jack-of-all-trades Gdynia, Poland

Posbit.pl
Implemented various parts of a POS (Point-of-Sale) software running on an embedded Linux platform.
Key tech: Git, Linux, Maria DB, Rabbit MQ, Perl 5.22, Python 3, C++11

usecase: reengineered code responsible for communication with a fiscal terminal to allow handling arbitrary amount of errors and restarts during a single fiscal transaction

usecase: designed a daemon process exposing TCP-based interface for communication with First Data payment terminal, and supervised a junior programmer implementing it

usecase: implemented a tool for static analysis of SQL schemas to automatically infer relationships between tables and verify their correctness to avoid data loss; the tool was also capable of generating Perl code to interface with the database

usecase: implemented code processing invoice data from distributed POS devices and Web interface to maintain a FIFO warehouse for accurate profit calculation

usecase: implemented federation of invoice data between distributed POS devices and an optional central server

usecase: implemented a tool for automatic setup, teardown, and management (including database migrations) of Web interface instances

NOTABLE OPEN SOURCE EXPERIENCE

Dec 2014—Present VM engineer

Viua VM
Design and implemention of Viua VM, a custom-ISA virtual machine.
Website: https://viuavm.org/
Key tech: Git, GNU Make, GDB, Valgrind (Memcheck), Linux, C++17, Viua VM assembly language, Python 3

usecase: designed an instruction set architecture, execution regime and asynchronous I/O model for the virtual machine

usecase: implemented various virtual machine subsystems to run in parallel, theoretically increasing its throughput

usecase: implemented a preemptive process scheduler for virtual processes running on the virtual machine, and an offloading mechanism to avoid foreign function calls blocking bytecode execution

usecase: designed an assembly language for the virual machine, and implemented an assembler and disassembler capable of converting assembly source code to bytecode and back

usecase: implemented a linker for bytecode modules to facilitate their separate compilation

usecase: implemented a static analysis and type inference engine for the assembly source code to aid programmers in writing reliable software

usecase: implemented a test suite for the VM using Python

Nov 2018—Present   Language designer, compiler engineer

Viua VM

Design and implemention of Viuact, a high-level programming language for Viua VM.
Website: https://git.sr.ht/~maelkum/viuact
Key tech: Git, Python 3, Viua VM assembly language

usecase: designed Viuact language's syntax (based on Lisp family of languages) and semantics (based on Erlang and OCaml languages)

usecase: implemented lexer, parser, and compiler for the Viuact language

Oct 2014—Present   Python programmer

Issue, issue tracker

Design and implemention a distributed issue tracking tool.
Website: https://github.com/marekjm/issue
Key tech: Git, Python 3, SSH

usecase: designed a set of primitives required to facilitate distributed issue tracking, and a data format to serialise them to for storage and transfer

usecase: implemented a way to exchange the issues between users of the system, and a command line user interface

Mar 2013—Mar 2018   API designer

diaspy

Maintained and extended the semi-official Python API for Diaspora* social network.
Website: https://github.com/marekjm/diaspy
Key tech: Git, Python 3, HTTP

usecase: participated in reverse engineering the data format and HTTP methods used by Diaspora* network nodes to communicate

usecase: implemented Python code to act as a user agent for communication with Diaspora* network

TECH SKILLS

Revision control

Git · highly proficient in day-to-day work

SVN · basic use (checkout, commit, merge)

Linux

BASH, awk, sed, xargs, SSH · proficient in day-to-day usage on a workstation

userspace debugging · some knowledge of /proc tree; some ability to use strace(1) and dmesg(1) to debug userspace issues

networking · basic ip(1) usage

distributions used · CentOS, Arch Linux

POSIX

experienced in using signals for inter-process communication (using POSIX syscalls or Linux-specific signalfd(2)

experienced in parallel programming using either processes or threads

experienced in child process management in complex software systems

proficient with different I/O models (blocking, non-blocking, asynchronous)

C++ and C compilers	GCC, Clang (x86) · able to quickly discern meaning of error messages, knowledgeable about diagnostic flags, some experience in using sanitisers (ASan, UBSan, etc.)
C++ and C debuggers	GDB (x86) · able to *ad hoc* inspect stack of multithreaded software and spot common error signs; able to analyse core dump files to look for common error sources; limited ability to work with stripped executables; limited ability to inspect assembly code
	Valgrind (x86) · able to understand Memcheck reports
Build systems	GNU Make · able to write Makefiles from scratch, or modify existing ones
Technical writing	LaTeX· able to typeset complex documents
	man · able to write simple manual pages
Software architecture	able to design, explain, and defend system architecture on multiple levels (from data type, class, and API, through process separation, to distributed systems)
	able to work with multiple programming paradigms and models (procedural, object-oriented, functional, actor, event-driven)

EDUCATION

Polish-Japansese Academy of Information Technology

| *2015-2019* | Engineer's Degree | Gdańsk, Poland |

Thesis: *Viua VM in action. Implementation of a high-level programming language and a simple application*
Description: I designed and implemented a high-level programming language to run on Viua VM platform. I have also implemented the Websocket protocol, which was required for the second part of the thesis (an Internet chat).
Key tech: Git, GNU Make, Linux, C++, Python 3, Websocket
Co-author: inż. Krzysztof FRANEK
Advisor: prof. dr. hab. Marek BEDNARCZYK
Link: http://www.marecki.me/Marek_Marecki-engineers_thesis-pl_PL.pdf *(in Polish)*

Languages

POLISH · Mothertongue

ENGLISH · Proficient (CPE)

GERMAN · Basic (simple words and phrases only)

TALKS, PUBLICATIONS, CONFERENCE PAPERS AND PRESENTATIONS

| *Nov 2019* | IT Fascinations | Gdańsk, Poland |

Title: *SUV 4x4. Four subsystems of Viua VM*
In-depth presentation of inner workings and architecture of Viua VM, and their role in facilitating development of robust systems.
Link: https://viuavm.org/media/talks/2019/suv4x4.pdf *(slides in Polish)*

| *April 2018* | NUUG | Oslo, Norway |

Title: *Virtual machines and where to find them*
Technical overview of Viua virtual machine, and discussion of impact of Free Software development on one's academic and business relations.
Link: http://www.nuug.no/aktiviteter/20180410-virt-viua/

| *April 2017* | ⟨*Programming*⟩ 2017 | Brussels, Belgium |

Title: *Moving beyond single-threaded concurrency*
Overview of the internals of the Viua virtual machine, and the primitives it provides to achieve its design goals.
Link: https://2017.programming-conference.org/details/MoreVMs-2017-papers/5/Moving-beyond-single-threaded-concurrency

GDPR

I hereby give consent for my personal data included in my application to be processed for the purposes of the recruitment process under the European Parliament's and Council of the European Union Regulation on the Protection of Natural Persons as of 27 April 2016, with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (Data Protection Directive)

November 26, 2021