

MIHAI MATEI

Personal details

Birth Date	September 3, 1983
Address	18 Latea Gheorghe Street, Bucharest
Phone	+4 0723.696.966
E-mail	matmih2002@yahoo.com
Status	not married

Objective

Developing a career in IT.

Improving myself as a system programmer, becoming a senior engineer and architect.

Experience

August 2004 – February 2005:	Telelogix Inc Software developer(freelancer)
I developed software for a family of recording devices and several COM add-ons for Microsoft Outlook(2000, 2002, XP) and ACT(an ERP) for managing contacts and phone records made with them.	
<ul style="list-style-type: none">• Programming languages: Visual Basic, C++• Development environments: Visual Studio 6.0, Borland C++ Builder 6• Technologies:COM, MFC, WinAPI, VCL• Operating systems: Windows 98, Me, 2000, XP	
November 2007 – April 2009:	Freescale Semiconductor Software engineer
I was part of the team that is developing the CodeWarrior IDE debugger, based on Eclipse CDT framework, for PowerArchitecture family processors. Most notably, I developed a debug agent for the virtualization solution on Freescale's e500mc 8 core processor(<i>Google Search</i> : "P4080 hypervisor" – the first hardware assisted hypervisor for embedded PowerPC architectures), that enabled a user to debug a guest application(AMP, SMP, raw binary or OS). I was also responsible for integrating the agent into the CodeWarrior's debug engine, on the host Windows/Linux side.	
<ul style="list-style-type: none">• Programming languages: Java, C++, C, PowerPC gcc asm• Development environments: Eclipse, CodeWarrior• Technologies:Eclipse CDT, SWT, CORBA, JNI, COM, ATL• Operating systems: Windows XP, Linux RedHat 4	

May 2009 – October 2009: Rack-Soft SRL
Software engineer

I was part of the team that is developing the 4PSA Voipnow product, a PBX software based on Asterisk and OpenSER (Kamailio). My main tasks were related to Kamailio product integration with the Voipnow architecture, including improving overall system's performance for high server SIP workloads, as well as a number of SIP bug fixing. I've also done an Asterisk application and some php Zend engine modules.

- Programming languages: C, php
- Development environments: Eclipse, vim
- Technologies: Asterisk PBX, Kamailio (OpenSER), SIP, Posix
- Operating systems: Windows XP, RedHat 4, Centos 5

Education

2007 – present	Polytechnic University of Bucharest Faculty of Automation Control and Computer Science Master of Computer Science in “Advanced Systems for Internet Applications”
2002 – 2007	Polytechnic University of Bucharest Faculty of Automation Control and Computer Science Specialization: C3 - “Core Systems and Applications” Bachelor of Computer Science 8.32 Graduation Average, 9.25 on Graduation Thesis
1998 – 2002	“Tudor Vianu” National College of Informatics, Bucharest

Additional Certificates

- Cambridge Certificate in Advanced English (2002)
- Junior programmer certificate, National College of Informatics “Tudor Vianu” (2002)
- “Proficiency in English” certificate, Polytechnic University of Bucharest (2004)
- Oracle Academic Initiative certificate (2006)

Skills

Personal skills

- Quick learner
- Team-worker
- Ambitious
- Perfectionist
- Highly adaptable

Technical skills

Category	Item	No. of years	Last year used	Skill level	Remarks
Systems [OS]	Linux(2.4.x, 2.6.x)	6	2009	4	Application and kernel programming, administration
	Windows	12	2009	3	Application and kernel programming, administration
	Solaris	1	2007	2	application programming
Programming languages	C/C++	10	2009	5	along with parallel libraries (MPI, OpenMP)
	Java	6	2009	5	along with J2EE (JSP/servlet, JMX, EJB, JSF, JDBC) and RMI
	Visual Basic	2	2006	2	COM programming and WinAPI
	SQL and PL/SQL scripting	3	2007	4	Oracle Academic Initiative certificate
	C#	2	2006	2	
	Python	1	2006	3	
	ASP	1	2006	2	ASP.Net Oracle project for the Oracle Academic Certificate
	XML family	2	2008	3	DTD, XSchema, XPath, XQuery, XSLT, SAX
	Intel x86 assembler	3	2006	2	
PowerPC assembler	2	2009	3		
Technologies	J2EE	2	2007	3	
	JSF/JSP/JavaScript	2	2007	4	
	MFC/WinAPI	1	2006	2	
	.Net Framework	2	2006	2	
	POSIX threads	5	2009	4	
	COM	2	2009	2	
	MPI	2	2007	4	
	OpenMP	2	2007	4	
	Corba	2	2009	2	
	RPC	1	2006	2	
	OpenGL	2	2006	3	
Tools/methods/other	Microsoft Visual Studio 6, .Net IDE	4	2006	3	Developing experience using this IDE
	Eclipse IDE	5	2009	4	Developing experience using this IDE
	Apache Tomcat, Axis, Jboss	2	2007	4	Deployment of applications, administration
	CVS	4	2009	3	
	Gcc, cl	5	2009	4	
	Gdb, Valgrind	3	2009	3	Used for debugging applications under Linux

Skill level: 1 – Theoretical knowledge; 2 - Junior; 3 – Confirmed experience; 4 – Advanced experienced; 5–Senior expert

Known languages

English	Advanced level
French	Average level

Hobby

Computers & Internet, Tennis, Skiing, Music, Travel

Projects

I will briefly describe some of the projects I am particularly proud of. For further information do not hesitate to contact me.

Multithreading library for POSIX systems and WinNT (2004)

This project was supposed to offer a common API to developing multithreaded applications on different systems. It implemented in a Java like pattern different objects to offer the user access to creation of threads, concurrency primitives and methods, signals and so on.

Responsibilities:

- Design the library OO architecture
- Porting the architecture on different systems
- Writing documentation and tests

Used technologies:

- Programming languages: C++
- Development environment: Visual Studio 6.0, Kate
- Operating System: Windows 2000, XP, Linux 2.4, Linux 2.6, Solaris

Com Add-on for Microsoft Outlook and ACT(2004-2005)

This project added new functionality to different versions of Microsoft Outlook and Act allowing a user to add phone records and to associate them with already existing contacts. The user had the possibility to modify the data, to listen to phone records, or to call a 3rd party application to edit the voice records.

Responsibilities:

- Design the Database Architecture
- Design the UI and user cases
- Testing and porting the application on different versions of the programs
- Modifying the Visual Basic Installer to deploy the add on on clients system,

Used technologies:

- Programming languages: Visual Basic
- Technologies: COM API for Ms Outlook 2000, 2002, XP, COM API for ACT, WinAPI, ActiveX, MS

Access.

- Development environment: Visual Studio 6.0
- Operating System: Windows 2000, Me, XP

Direct Connect Client for Linux (2005)

This project wanted to extend the functionality and the success of direct connect applications (such as DC++) to the Linux System. It implemented several features such as connecting

and authenticate to a hub, getting the list of users, sending and responding to search queries, fetching the file list of a user, downloading a file from a user, uploading a requested file and so on. Also a user had the choice to be either active or passive as with the specification of the DC protocol.

Responsibilities:

- Implementing the multithreading module
- Implementing the networking layer
- Writing a parser for the DC protocol
- Writing an event based architecture based on callbacks
- Writing an XML framework for parsing settings and user file lists
- Developing the GUI using Qt

Used technologies:

- Programming languages: C/C++
- Technologies: POSIX threads, Socket Api, Qt, libxml
- Development environment: Kate, KDevelop
- Operating System: Linux 2.4.x, 2.6.x

Interpreter and Code Generator for a Pascal based language (2006)

This was the first part of making a compiler for a Pascal based language as an assignment for the Compiler Course I took while in school. I developed the syntactic and semantic analysis of the language. The compiler could generate the Abstract Syntax Tree of a pascal code, it could interpret the code, printing the results, or it could generate asm like code. The asm code, which was similar to x86 asm, could then be interpreted by a simulator.

Responsibilities:

- Writing the grammar for the language
- Generating the AST
- Interpreting the code based on AST
- Generating asm like code from the AST

Used technologies:

- Programming languages: C/C++
- Technologies: Lex, Yacc
- Development environment: Kate
- Operating System: Linux 2.6.x

Source Cross Referencer for C/C++ (2007 – Thesis Project)

This project is meant to be a platform for storing and indexing source code for different languages, even if now it only supports C/C++. It is implemented as a three-tier web application using Java, Java Server Faces for the web and business tier and JDBC for data tier. The application has many features such as: the ability to log into the system, to create a new project, to upload source code(or archives containing more sources). The system will store the source code files in a binary format in the database, indexing the source, identifying certain language tokens and types. Also the user can browse a project tree, can view a source code nicely formatted according to a specific language, and can click on a type to see properties for it (like where it was defined, in what other sources in the project it is used, and how many times it appears in that source). The application is very similar from this point of view with Cross Referencing Linux project. Also the application was design to allow new modules to be installed(such as a plug-in for a new language). The project has internalization, having an English and a Romanian version.

Responsibilities:

- Developing the framework of the application
- Designing the look of the web application
- Developing the fuzzy parsing algorithm to identify types for C/C++
- Implementing the indexing system for C/C++

Used technologies:

- Programming languages: Java
- Technologies: JSF, JSP/servlets, JDBC, JAXP, JCE, JavaCC
- Development environment: Eclipse with Exadel Studio Add on

HyperTRK debug agent for P4080 hypervisor (2008 – 2009)

The purpose of any internal debug agent is to offer debug services over the higher level applications that are running over the current run level to any external or internal client. One such example is the ptrace system call service offered by the Linux kernel that enables userspace debug clients, such as gdb, to control the state of the targeted application. This is the same for hyperTRK, except that the current level of execution is not in supervisor mode, such as the Linux kernel, but in the real operating mode that hypervisor software is running. Offering debug services such as run control, register and memory access it can be used by any debug client, implementing Freescale's proprietary TRK protocol, an example being CodeWarrior for PowerApplication Eclipse IDE. The hyperTRK debug agent is practically offering debug services on the virtual cores that the hypervisor is exposing to the guest application. Tasks that involved developing the agent included integrating with the internal byte channel mechanism exposed by the hypervisor for communication, accessing the emulated guest registers, adding the temporary TLB mappings from physical guest supervisor to virtual hypervisor. The programming environment was the SMP based application that the hypervisor software is, at different interrupt levels and context that the software is implementing.

Responsibilities:

- Developing the agent and integrating it with hypervisor software
- Developing the host debugger Protocol Plugin layer, and integrating it in the debugger engine
- Developing GUI Eclipse CDT CodeWarrior TRK debug configuration

Used technologies:

- Programming languages: C, PowerPC asm, C++, Java
- Technologies: virtualization, Eclipse SWT
- Development environment: Ganymede, CodeWarrior, gcc, make

Kamailio Custom Slab Allocator (2009)

Any custom allocator must be able to replace the default system's libc's allocator. On Linux systems this is possible in a generic way due to the weak reference of the malloc's specific functions. Because of the specific nature of the Kamailio application, being a multi-process non threaded, the thread optimised libc's malloc had some performance penalties, especially when allocating large amounts of objects of equal size. To improve on Kamailio's allocation requirements a custom Slab allocator was implemented for both private memory and shared memory allocation. The memory mapping scheme was done as sequential chunks of memory mapped into the process address space using posix mmap's API. The shared memory allocator used a lazy on demand scheme to defer mapping the shared memory into each of Kamailio process (it only synchronized the memory layout when the SIGSEGV signal was raised). The Valgrind custom API was implemented for this allocator to enable application debugging using Valgrind's engine and implemented tools (like memcheck).

Responsibilities:

- Developing the the allocation logic and SLAB architecture
- Implementing, testing and profiling for high workloads using Oprofile

Used technologies:

- Programming languages: C
- Technologies: posix memory API
- Development environment: RedHat4, Centos 5