

Samir Ribić
Trg ZAVNOBiHa 14
71000 Sarajevo

Tel. 033-543174
061-314779

E-mail
mezaribi@epn.ba
samir.ribic@etf.unsa.ba

Elektrotehnički fakultet u Sarajevu
Zmaja od Bosne b.b.
Sarajevo

Naučno-nastavnom vijeću
putem Vijeća Odsjeka za računarstvo i informatiku

Predmet: Zahtjev za ocjenu podobnosti kandidata i teme za izradu doktorske disertacije

Molim da mi se odobri izrada doktorske disertacije pod radnim naslovom
“Koncept i implementacija programskog jezika i prevodioca za embedded sisteme na bazi dekompilacije mašinskog koda i ekvivalencije izvornog i izvršnog koda”

OBRAZLOŽENJE

1. Opis oblasti istraživanja

Prevodioci viših programskih jezika su najčešće realizovani u obliku čistih kompajlera, interpretera, pseudokompajlera i hibridnih prevodioca.

Čisti kompajleri prevode iz višeg programskog jezika u mašinski ili asemblerski jezik. Od ova četiri koncepta, ovo je koncept koji generiše najbrži izvršni kod i stoga se najčešće koristi na većim sistemima. On je pogodan i za embedded sisteme ukoliko se prevođenje koda obavlja na drugom računaru, ali za prevođenje na istom embedded sistemu na kome se program i izvršava nije najpogodniji. Razlog tome je potreba i za izvornim i za izvršnim kodom, što zahtijeva veću količinu memorije (koja je na ovim sistemima često ograničena), te vremena potrebnog za kompilaciju.

Interpreteri su znatno sporiji od kompajlera, ali ne prave razliku između izvornog i izvršnog koda. Kod ovog koncepta se izvorni kod analizira i na bazi njega pozivaju odgovarajući podprogrami. Interpreteri nemaju potrebe za prevođenjem cijelog programa

svaki put kada se promijeni samo jedna linija. Mana koncepta interpretera je sporiji rad, potreba za bibliotekom rutina i interpreterskom petljom.

Pseudokompajleri se ponašaju slično kompajlerima. Oni prevode viši programski jezik u pseudomašinski kod. Ovaj pseudomašinski kod zahtijeva specijalni interpreter, a potrebni su pored njega i izvorni i izvršni (pseudo) kod. Alternativa specijalnom interpreteru su dodatni prevodioci koji ovaj program prevode u prirodni mašinski kod, bilo na zahtjev korisnika, bilo u trenutku izvršavanja (just in time). Iz pseudokoda se ponekad može rekonstruisati i izvorni kod, iako to nije konceptualno ugrađeno u sam jezik.

Hibridni prevodioci su realizovani ili projektovani na više načina. Najpoznatiji je prevodilac za programski jezik FORTH. On je uveo koncept prevođenja podprograma (koji se u ovom jeziku zovu riječi) odmah nakon njihovog unosa u kompaktan kod iz koga se rekonstruiše izvorni kod kada se riječ treba editovati. No, format ovog koda nije čisti mašinski kod, nego se i dalje interpretira i koristi biblioteke. Osim toga, FORTH je jezik dosta niskog nivoa zbog koncepta steka i obrnute poljske notacije.

Proces suprotan kompajliranju je dekompajliranje, odnosno prevođenje iz mašinskog jezika ili pseudomašinskog koda. Ova oblast je daleko manje razvijena od oblasti kompajliranja. Uspješnost procesa mnogo varira od računarske arhitekture, jezika u kome je pisan izvorni kod, formata prevedenog koda, stepena optimizacije mašinskog ili pseudomašinskog koda itd.

Kompilacija, editovanje i dekompilacija su najčešće odvojeni procesi. Dekompilacija u gotovo svim implementacijama programskih jezika nije sastavni dio prevodioca, a i ako jeste to obično nije dekompilacija mašinskog koda nego pseudomašinskog ili nekog drugog kompaktnog formata.

U ovoj tezi bi se istražila mogućnost realizacije prevodioca višeg programskog jezika na novoj osnovi, koji bi objedinio neke dobre osobine interpreterskih i kompajlerskih jezika. Te osobine bi bile: jedinstveni kod u glavnoj ili spoljnoj memoriji (a ne i izvorni i izvršni), ukidanje potrebe za prevođenjem cijelog programa, brzina izvršavanja mašinskog jezika i neovisnost o interpreteru i ugrađenim bibliotekama. Programski jezik treba da ima standardne elemente proceduralnih programskih jezika (izraze u algebarskoj notaciji sa zagradama i prioritetima, uslove, nizove, petlje, podprogramme, različite tipove podataka...).

Većinu uloge prevodioca ili interpretera ovdje preuzima specijalizovani editor. Svaka unesena naredba jezika se prevodi u mašinski kod nakon njenog unosa, brinući o lokaciji u memoriji, skokovima i alociranim varijablama. Kada se želi editovati neka linija, ona se iz mašinskog programa dekompajlira u liniju izvornog koda. Linija postoji u ASCII obliku samo u vrijeme editovanja, nakon čega se ponovo pretvara u mašinski kod ili prikazivanja, nakon čega se briše iz memorije.

2. Stanje u oblasti

a) Embedded sistemi sačinjavaju najširu kategoriju kompjuterskih sistema. Ova kategorija uključuje sve osim kompjutera opšte namjene. Primjeri embedded sistema su: bankomati, avionski uređaji, celularni telefoni, ruteri, printeri, fotokopiri, diskovi, kontrolori kočnica u automobilima, termostati, kalkulatori, mikrovalne pećnice, DVD plejeri, televizori, programabilni logički kontroleri za industriju, medicinska oprema, ručni satovi, PDA digitalni asistenti, napredni mobilni telefoni, konzole za video igre. U ovim uređajima mikroprocesori variraju od četvorobitnih do šezdeset četvorobitnih, a količina memorije od reda bajtova do reda gigabajta. Neki od uređaja se mogu i programirati od strane korisnika (najčešće sa ugrađenim interpreterskim jezikom), dok su neki preprogramirani već ranije. S obzirom na širinu područja koju embedded sistemi obuhvataju, a to je gotovo cijela moderna elektronska industrija, u pitanju je oblast u koju se mnogo ulaže i koja će u budućnosti davati još više rezultata.

b) Oblast kompajliranja je široko poznata oblast u računarskim naukama. Počela se razvijati još od 1957 godine sa radovima Grace Hopper na jeziku A-0 i Johna Backusa na jeziku FORTRAN. Teoretski principi kompajlera su razvijeni u toku šezdesetih godina dvadesetog vijeka.

Kompajleri se pišu u implementacijskom jeziku. U početku je to bio asemblerski jezik, a od ranih sedamdesetih godina i u višim programskim jezicima. Čest je primjer da se kompajler za neki programski jezik napiše u njemu samom.

Od uspostavljanja kompajlerske teorije, kompajleri najčešće imaju sedam osnovnih elemenata, od kojih je svaki tema velikog broja istraživanja i publikacija. To su:

1. Leksička analiza – pretvaranje izvornog koda u manje dijelove, tokene ili terminalne simbole.
2. Sintaksna analiza ili parsiranje – identificiranje sintaksne strukture izvornog koda
3. Semantička analiza – prepoznavanje značenja programskog koda, tipova i priprema izlaza.
4. Međureprezentacija – ekvivalentna predstava u formi stabla ili grafa odnosno nekog međujezika.
5. Kompajlerska analiza – Dobijanje informacija o programu iz međureprezentacije o zavisnostima između varijabli, podataka, dijelova koda itd. Ova faza je osnova za optimizaciju koda.
6. Optimizacija – transformacija međureprezentacije u ekvivalentnu ali bržu ili kraću predstavu.
7. Generisanje koda – transformacija međureprezentacije u mašinski kod.

Iako je oblast dosta razvijena, istraživanja na temu kompajlera su i dalje česta, posebno u oblastima kompajlerske analize, međureprezentacije i optimizacije. Paralelno se poboljšavaju i komercijalni kompajleri. Detaljan spisak trenutnih istraživanja i istraživača u ovoj oblasti može se naći na adresi:

<http://www.compilerconnection.com/research/research.htm>

c) Oblast dekompajliranja je daleko manje istražena od oblasti kompajliranja. Razlog tome je gubitak informacija koji se dešava prilikom prevođenja iz višeg programskog jezika u mašinski jezik. Za razliku od tehnika kompajliranja koje su tema standardnih kurseva na većinama univerziteta, tehnike dekompajliranja se uglavnom publikuju u doktorskim tezama i pojedinačnim programima. Oblast je ustanovio Maurice Halstead tokom šezdesetih i sedamdestih godina kada su korišteni za portiranje softvera sa mašina druge na mašine treće generacije, tokom osamdesetih za rekreiranje izgubljenog izvornog koda i modifikaciju postojećeg binarnog koda, a devedesetih su postali alatke za inverzni inženjering, detekciju malicioznog koda, razumijevanje rada biblioteka i prenos mašinskih programa.

Popis najznačajnijih radova iz oblasti dekompile:

1960, 1962	D-Neliac decompiler	1993	Alpha AXP Migration Tools
1963-7	The Lockheed Neliac decompiler	1993	Source/PROM Comparator
1966	W.Sassaman	1994	The dcc decompiler
1967	Autocoder to Cobol translator	1995	DECLER Decompiler
1972-6	The Inverse Compiler	1997-2001	University of Queensland Binary
1973	C.R.Hollander PhD thesis	Translator	
1973	B.C.Housel PhD thesis	1999	A. Mycroft's Type Reconstruction for
1974	The Piler System	Decompilation	
1974	F.L.Friedman PhD thesis	2000	University of London's Asm21toc reverse
1974	Ultrasystems	compiler	
1974	V.Schneider and G. Winiger	2001	Proof-Directed De-compilation of Low-
1977, 1981, 1988	Decompilation of Polish code	Level Code	
1978	G.L.Hopwood PhD thesis	2001	Computer Security Analysis through
1978	D.A.Workman	Decompilation and High-Level Debugging	
1981	Zebra	2001	Type Propagation in IDA Pro
1982	Decompilation of DML programs	Disassembler	
1982, 1984	Forth Decompiler	2001	Java decompiler
1984	Dataflex Decompilers	2001	DisC, by Satish Kumar
1985	Software transport system	2002	ndcc decompiler
1988	J.Reuter's Decomp VAX BSD decompiler	2002	The Anatomizer Decompiler
1989	FermaT	2002	Analysis of Virtual Method Invocation for
1990	Austin Code Works' exe2c DOS	Binary Translation	
decompiler		2002	Boomerang
1991	PLM-90 decompiler	2002	Desquirr
1991	O'Gorman's PhD thesis	2004	Analyzing Memory Accesses in x86
1991-94	Decompiler compiler	Executables	
1991-93	8086 C Decompiling System	2004	R. Falke's Type Analysis for Decompilers

Detalji o ovim istraživanjima mogu se naći na Web stranici

<http://www.program-transformation.org/Transform/HistoryOfDecompilation1>

d) Istraživanja u konceptima prevodilaca programskih jezika koji objedinjuju osobine interpretera i kompajlera vođena su na više polja.

Najpoznatiji je interpreter za programski jezik Forth koji je inicirao Chuck Moore. Okruženje za ovaj jezik se sastoji od stek-orijentisanog, interaktivnog inkrementalnog interpretera i kompajlera za virtuelnu mašinu.

Koncept inkrementalnog kompajlera, koji prevodi samo jednu liniju čuvanu u virtuelnoj memoriji, patentiran je od strane McKeeman, William M.; Aki, Shota;

Michael P. Plezbert je definisao koncept kontinualnog kompajlerskog modela u kome se kompajler i interpreter programskog jezika izvršavaju istovremeno tako da interpreter omogućava startovanje programa odmah, dok kompajler u međuvremenu prevodi podprograme.

3. Pravci istraživanja u okviru doktorske disertacije i očekivani rezultati

Istraživanje će se obavljati u pravcima dizajna jezika, programskih biblioteka, određivanja strukture memorije, generisanja koda i njegove optimizacije te pronalaženja algoritama za unos, editovanje i prikaz programa.

U toku istraživanja potrebno je analizirati različite računarske sisteme, mikroprocesore i mikrokontrolere da se za svaki od njih pronađe obostrano jednoznačno preslikavanje između izvornog i izvršnog koda. Potrebno je istražiti pogodnu sintaksu ovakvog jezika koja bi omogućila realizaciju editor-prevodioca uz eliminaciju višeznačnosti pri dekompajliranju. Za određene sisteme će se editor-prevodilac i realizovati, u formi ROM-a za uređaj ili eksternog programa.

Dekompilabilni kod zahtijeva nešto redundantnih informacija kako bi se saznala imena varijabli i funkcija i generisanje koda koji nije optimizovan, jer optimizacije uništavaju obostranu jednoznačnost. Ali, očekuje se da će rezultati i dalje biti bolji od onih koje pružaju interpreterski jezici, a i zauzeće memorije na embedded sistemima (pogotovo onim bez diskova) mnogo manje od onog koje imaju kompajlerski jezici. Neki ciljevi istraživanja su da se smanji količina redundantnih informacija.

Prvi rezultati istraživanja su pokazali da je ovakav koncept editor-prevodioca programskog jezika gdje su izvorni i mašinski kod identični ostvariv. Realizovan je mali editor-prevodilac za programski jezik koji ima većinu standardnih operatora u aritmetičkim izrazima, cjelobrojne varijable, uslove, podprograme i petlje, čime su postignuti uslovi teorema o struktuiranom programiranju. Format programa je obična COM datoteka sa mašinskim kodom za 8086.

Uprkos čestom mišljenju, sistemi sa malo memorije nisu nestali, samo je promijenjena njihova oblast primjene. Tako je, recimo, tipična memorija moćnijih komercijalnih programabilnih kalkulatora oko 32 K, dok specijalizovani mikrokontroleri na čipu često imaju i manji RAM. Editor-prevodilac programskog jezika koji nema odvojeni izvorni i izvršni kod može postići značajnu uštedu memorije na ovakvim uređajima u odnosu na klasične kompajlere i povećanje brzine izvršavanja u odnosu na interpretere.

Rezultati ovog istraživanja mogu biti korisni ne samo za programiranje embedded sistema sa malo memorije, nego i za programiranje personalnih računara. Tu se prije svega misli na oblast zakrpa softvera, open source inicijative te uštede u vremenu

razvoja. Open source licence zahtijevaju isporuku i izvornog i izvršnog koda, što često dovodi do distribucijskih problema, jer se moraju održavati dvije verzije programa, izvorna i izvršna. Programi napisani u jeziku čiji je mašinski kod ujedno i izvorni kod, automatski zadovoljavaju ovakve licence, uz svođenje distribucije na samo jednu verziju. Također, manje zakrpe softvera pisanog ovakvim programskim jezikom je moguće obavljati na licu mjesta, bez posezanja za izvornim kodom u klasičnom smislu ili distribucijom čitave izvršne verzije programa. Ne treba zanemariti ni uštede u vremenu kompajliranja u usporedbi sa klasičnim kompajlerima, odnosno uštede u vremenu izvršavanja u usporedbi s klasičnim interpreterima.

4. Pregled relevantne literature za oblast istraživanja

Ovdje je naveden dio literature koji će se koristiti prilikom izrade doktorske teze.

1. B.S. Baker. "An algorithm for structuring flowgraphs". Journal of the ACM, 24(1):98-120, January 1977.
2. C. Cifuentes. "Reverse Compilation Techniques". PhD Dissertation. Queensland University of Technology, Department of Computing Science, 1994.
3. D. Balbinot and L. Petrone. "Decompilation of Polish code in Basic". Rivista di Informatica, 9(4):329-335, October 1979.
4. D.A. Workman. "Language design using decompilation". Technical report, University of Central Florida, December 1978.
5. David Gelernter, Alexandru Nicolau and David Padua. "Languages and Compilers for Parallel Computing", MIT Press 1990
6. G.L. Hopwood. "Decompilation". PhD dissertation, University of California, Irvine, Computer Science, 1978.
7. Grupa autora: AMD Elan microprocessor
8. Grupa autora: IA-32 Intel® Architecture Optimization Reference Manual
9. Grupa autora: IA-32 Intel® Architecture Software Developer's Manual
10. Grupa autora: M68000 Programmers reference manual
11. Keith Cooper, Linda Torczon, "Engineering a Compiler", Morgan Kaufmann (September, 2003)
12. Kulenović Adnan "Osnove projektovanja kompajlera", Svjetlost 1988
13. M.N. Bert and L. Petrone. "Decompiling context-free languages from their Polish-like representations". pages 35-57, 1981.
14. Peter Lee: "Realistic Compiler Generation", MIT Press 1989
15. Ronald Mak "Writing Compilers and Interpreters", Wiley; 2 edition (August 10, 1996)
16. Samir Ribić: "Spiralni pristup u razvoju programskog jezika i prevodioca", magistrski rad, ETF Sarajevo, 2003
17. Steven Muchnick : "Advanced Compiler Design and Implementation", Morgan Kaufmann; 1st edition (August 1, 1997)
18. V. Schneider and G. Winiger. "Translation grammars for compilation and decompilation". BIT, 14:78-86, 1974.
19. W. May. "A simple decompiler". Dr.Dobb's Journal, pages 50-52, June 1988.

5. Biografija kandidata

Rođen sam 4.11.1968. godine u Brčkom, a osnovnu školu sam završio u Sarajevu 1983. godine sa prosjekom 5,0. Već na kraju osnovne škole pokazujem veliko interesovanje za računarske nauke, te upisujem Srednje usmjereno obrazovanje, zanimanje programer , te stičem taj stepen obrazovanja 1987, sa prosjekom ocjena 4,6. Pored redovnog obrazovanja u srednjoj školi pohađao sam i dodatne kurseve BASICA, Z80 asemblera i LISPa, te držao kurseve upotrebe računara za nastavnike osnovnih škola. Svoje prvo samostalno predavanje održao sam 1986 pod nazivom “Računar kao sredstvo učenja esperanta”.

1987 objavljujem svoj prvi stručni rad na temu “Grafika u mašinskom jeziku” u listu “Svet kompjutera”. Ukupno je objavljeno oko 60 radova, iz oblasti mašinskog jezika, sistemskog softvera, informatičke publicistike i aplikacionog softvera, čiji je djelomični spisak dat u prilogu. Za vrijeme služenja vojnog roka bavio sam se, između ostalog, računarskim simulacijama raketa. Elektrotehnički fakultet u Sarajevu započeo sam studirati 1988 pokazavši poseban uspjeh u predmetima iz oblasti računarskih nauka (ukupni prosjek ocjena je 8,6, a prosjek ocjena viših godina je 9,5). Za vrijeme rata u BiH učestvovao sam u razvoju informacionih sistema u Armiji BiH, te na digitalnim vezama. U periodu 1993-1999 učestvovao sam u realizaciji dva emulatora Z80 mikroprocesora, Warajevo za Intel 80286, na PC računarima (u suradnji s mr Željkom Jurićem) i Tezxas za Motorolu 68000 na TI 89 i TI92+ kalkulatorima(samostalno). Emulator Warajevo poslužio je kao baza za diplomski rad na temu “Simulacija mikroracunarskog sistema ZX Spectrum na personalnim računarima tipa PC-AT”. Rad je odbranjen u martu 1996, i stečeno zvanje diplomiranog inženjera elektrotehnike. Warajevo je 1998 po mišljenju nekih kompjuterskih časopisa širom svijeta ocijenjen kao najbolji emulator ZX Spectruma, a Tezxas je prvi u svijetu realizovani emulator kućnog kompjutera na džepnom kalkulatoru.

Po završetku rata diplomirao sam i radio u OSCE kao saradnik za informacione tehnologije, zatim Hermes SoftLab Ljubljana kao razvojni inženjer na projektu OmniBack (program za back-up podataka koga prodaje firma Hewlett Packard). Od septembra 1996 prelazim u stalni radni odnos u firmu Hermes SoftLab Sarajevo najprije kao razvojni inženjer, a zatim kao direktor za tehnologiju. U toj firmi sam držao edukaciju iz oblasti operativnih sistema i mreža, obavljao superviziju manjih projekata (“Finansije - Eko stil”, “Studentska služba - Fakultet za fizičku kulturu”, “Poljoprivredni fakultet – obrada anketa”), realizovao Internet konekciju i lokalnu mrežu, same firme te Državne bolnice i Energokomerca, te učestvovao u projektu “Modernizacija platnog prometa”, koji je izuzetno složena distribuirana aplikacija.

1998 postajem asistent na Elektrotehničkom fakultetu na predmetima “Programska organizacija računara i operativni sistemi” za odsjeke za RI i TK, te na predmetu “Digitalni računari I” na odsjeku za TK. Od 2003 sam predavač na predmetu “Projektovanje sistemskog softvera” na Elektrotehničkom fakultetu.

U oktobru 1999 prešao sam u firmu Alem sistem, u kojoj sam radio do februara 2005 kao direktor za razvoj i istraživanje. U ovoj firmi učestvovao sam na projektima: “Modernizacija platnog prometa” (edukacija kadra i objašnjenje internog rada aplikacije), “Zdravstveni finansijski sistem”, “Informacioni sistem apoteka” , “Informacioni sistem Natron Maglaj”, “Reforma knjigovodstva Federacije BiH”, “Reforma knjigovodstva Države BiH”, “Reforma knjigovodstva kantona”.

Od februara do juna 2005 radio sam kao asistent na Sarajevo school of science and technology na predmetima “Computer organization and basic networking”, “Programming and problem solving I” , “Programming and problem solving II” , “Protocols and software organization” a od juna 2005 prešao sam u stalni radni odnos na Elektrotehnički fakultet na predmetima “Programska organizacija računara i operativni sistemi” i “Projektovanje sistemskog softvera”.

Pohađao sam stručne kurseve “ICRC – konfiguracija Cisco rutera”, “Oracle DBMS 8 administration”, “Oracle PL SQL”, “Oracle financials”, “Oracle procurement”, “Oracle order entry”, “Oracle financials administration” i “Oracle financials Case Study”. Položio stručni ispit “Windows NT 4.0 Server” i stekao zvanje “Microsoft Certified Professional” a zatim i sljedeće Brainbench ispite sa odgovarajućim certifikatima: Assembly Language, Cisco Network Support, C, DOS 6.2, Oracle Financials Rel 11, Programmer/Analyst Aptitude, SQL (ANSI), Windows Application Program Interface, Internet Concepts, MS Windows 98 Administration, Linux Administration (General), MS Windows 95 Navigation, E-Commerce Concepts, MS Windows NT 4.0 Administration, Project Management, Delphi 3.0, Oracle 8.0 Administration, Computer Technical Support, MS Windows 2000, Network Technical Support, Java 1, Human Resources Concepts (U.S.), Delphi 6.0, General Sciences, C++ Fundamentals, Project Management (2000), Programming Concepts, Data Entry Numeric, Written English, Cisco Network Design, CGI Scripting, Listening Skills, Business Communication, Information Technology Terminology, Networking Concepts, Disaster Recovery and Planning, Server Administration, Business Writing, Managing People (U.S.), Customer Requirements Analysis, Change Management (U.S.), Technical Writing, Information Technology Security Fundamentals, Coaching, RDBMS Concepts, Data Modeling Concepts, Data Modeling Concepts, Oracle PL/SQL, XML, Software Configuration Management, Software Testing

Od 1997 sam član Tehničkog komiteta za informatiku u sastavu Zavoda za standardizaciju, mjeriteljstvo i patente i evaluirao sam oko 150 standarda za proglašavanje.

Postdiplomski magistarski studij sam započeo krajem 1999 , u okviru postdiplomskog studija realizovao aplikacije “TCP/IP protokol za programabilni kalkulator TI-89” i “Web baza podataka za praćenje epizoda serije ‘Zvezdane staze’”, a zatim započeo sa razvojem vlastitog programskog jezika i kompajlera, koristeći pojednostavljenu metodu, pogodnu za nastavu, što je poslužilo kao baza za magistarski

rad pod nazivom “Spiralni pristup u razvoju programskog jezika i prevodioca”. Rad je odbranjen u junu 2003, čime je stečeno zvanje magistra elektrotehničkih nauka.

U toku 2004 godine realizovao sam “DEDAFORGEN” automatski generator aplikacija za programski jezik Delphi, koji iz strukture podataka generiše izvorni kod za unosne i pretražne forme, izvještaje i uputstva.

Od 2005 član sam Međunarodne akademije nauka San Marino (Akademio Internacia de la Sciencoj) koja je članica IAUPL (International Association of University Professor & Lectures).

Objavio sam tri rada na konferencijama, navedena u referencama.

Pored maternjeg jezika dobro znam engleski i esperanto, a služim se i francuskim i slovenačkim. Član sam komiteta Svjetskog saveza za esperanto iz Bosne i Hercegovine i učestvovao sam na oko 30 međunarodnih kongresa, skupova i susreta.

6. Reference kandidata

Iz oblasti dekompajliranja i parsiranja objavljena su tri rada

“Racunaj kako piše 1” Svet kompjutera 6/90
“Racunaj kako piše 2” Svet kompjutera 7-8/90
“Analiticko diferenciranje” Svet kompjutera 11/90

Iz oblasti kompajlera imam objavljena dva rada na konferencijama, magistarski rad i nastavnu skriptu:

“A simplified method of the compiler development” (ITI Cavtat 2005)
“An example of the compiler explainable in a short time” (ICAT Sarajevo, 2005),
magistarski rad “Spiralni pristup u razvoju programskog jezika i prevodioca”, 2003.
“Projektovanje sistemskog softvera”, skripta, 2003.

Iz oblasti računarskih arhitektura i programiranja u mašinskom jeziku objavljen je jedan rad na konferenciji i 22 rada u stručnim časopisima.

“Hipotetski procesor HP-16”, IKT Sarajevo 2005. (zajedno s Idrizom Fazlićem)
“Grafika u mašinskom jeziku”, Svet kompjutera 1/87
“Kako raditi u mašincu”, Svet kompjutera 2/87
“Izvinite zbog prekida programa”, Svet kompjutera 5/87
“Ne ulazi bez poziva”, Svet kompjutera 7-8/87
“Nedovršena PC simfonija” Svet kompjutera 9/87
“Paralele bez merdijana” Svet kompjutera 10/87
“Dobrodošli u mašincima” Svet kompjutera 11/87
“Gospodarenje memorijom” Svet kompjutera 12/87

“Instrukcije prenosa” Svet kompjutera 1/88
“Sabiranje i oduzimanje” Svet kompjutera 2/88
“Matematika nije bauk” Svet kompjutera 3/88
“Razmišljajmo logicki” Svet kompjutera 4/88
“Šiftovanje, rotiranje i stek” Svet kompjutera 5/88
“Podprogrami i skokovi” Svet kompjutera 6/88
“Uslovni skokovi i petlje” Svet kompjutera 7-8/88
“Pod tackom razno” Svet kompjutera 9/88
“A sad adio” Svet kompjutera 10/88
“Visoka rezolucija na niskom nivou” Svet kompjutera 1/89
“Kompresujmo ekran” Svet kompjutera 1/89
“Još o PC casovniku” Svet kompjutera 12/89
“Decimalni editor za mašinski kod” Svet kompjutera 4/90
“Video režimi”, Svet kompjutera, 10/92
“Warajevo ZX Spectrum emulator” Moj mikro 7-8/96

Iz oblasti operativnih sistema i organizacije memorije objavljena su tri rada:

“Program segment prefix” Svet kompjutera 3/93
“COM programi” Svet kompjutera 4/93
“EXE programi” Svet kompjutera 5/93

Iz ostalih oblasti računarstva objavljeni su sljedeći radovi:

“Turbo ili GW”, Svet kompjutera 2/89
“Lisp – generalne osobina”, Svet kompjutera 5/89
“Funkcije Lisp-a”, Svet kompjutera 6/89
“Iris MUV”, Svet kompjutera 6/89
“Lisp: Demonstracija”, Svet kompjutera 7-8/89
“Lisp: Borba protiv grešaka”, Svet kompjutera 9/89
“Sycerop DB: Program pravi programe”, Svet kompjutera 10/91
“Chi Writer, Citizen 180D i YU slova”, Svet kompjutera 10/91
“PSPICE”, Svet kompjutera 2/92
“Borland Delphi 4 Standard”, Svet kompjutera 11/98
16 informativnih članaka, (izvještaji sa računarskih sajмова i događaja)

Prema dosadašnjem toku istraživanja i problematici kojom se rad bavi, kao mentora predlažem red. prof. dr. Adnana Salihbegovića.

U nadi da ćete pozitivno odgovoriti na moju molbu, unaprijed se zahvaljujem.

Sarajevo, _05.01.2006

Podnosilac
Mr. Samir Ribić, dipl. el. ing