**David A.D. Morano**

**Current location: Malden, MA 02148-2752**

**Availability to join project: 1week notice**

**Interview Availability: 1day notice**

**EXPERIENCE WITH TOP SKILLS REQUIRED:**  
**Experience as C/C++ Developer**

* David has over 15+ year experience in Software architecture and development, enterprise or real-time embedded.
* Distributed or parallel computer system research and development, software and hardware. Embedded system design and development, software and hardware.
* C/C++/UNIX/Linux development of software components (all mutlithread-safe): library, programs, specialized loadable objects.
* Experience in development and analysis correctness verification analysis on C and C++ source languages.
* Analysis outputs optionally consist of fault source line location and counts of the various element openings and closings.
* Experience in designing and developing much low-level code in the embedded space: - power-up diagnostic code, peripheral device discovery and initialization, other processor discovery (for multiprocessors)
* Worked on Synchronization, system initialization code (memory, MMU, processor traps and interrupts, et cetera), coded for both single processors and symmetric multiprocessors

**Code Library**

* ***A fairly large sample of code (over 1-million NCSL) can be found on GitHub at***

***http://GitHub.com/DavidMorano/RightcoreMainBase***

**PROFESSIONAL WORK EXPERIENCE**

**RightCore Network Services, Malden, MA May 2007 – present**

**C/C++/UNIX/Linux developer**

**Responsibilities:**

* C/C++/UNIX/Linux development of software components (all mutlithread-safe): library, programs, specialized loadable objects.
* Container library objects: vectors, strings, numerous queues and FIFOs (interlocked, multi-threaded), sets, maps, arrays (fixed and variable).
* Utility objects: filesystem related, buffer related, numerous storage managers, string table creation and use, message queues, random variables.
* Mail related objects: mailbox, mail-msg, mail-attachment, and numerous sub-components.
* UNIX-related objects: signal, object-file.
* Utility components: numerous string creation-testing-searching, random variables, marshaling-serialization.
* UNIX system related components: account, network database path manipulation.
* UNIX filesystem middleware.
* UNIX shell development of over 100 built-in commands.
* Other: OOA/OOD, extensive code reuse, high performance multi-thread execution, very deep software stacks, dynamically loadable shared objects, user-mode system call emulation, numerous constant databases, sockets, XNET, TLI, numerous networking utilities.
* Large portfolio of software available for perusal.
* Specialties: no bugs, no memory leaks, no deadlocks, no other leaks (FDs, background threads).
* Dispatched sub-servers can be of several varieties:
  + An already running program which accepts a passed file descriptor a dynamically loaded shared object command, otherwise looking like an
  + independent program
  + an independent program (typical)
  + a special built-in service

**PROJECT DETAILS:**

**Computer language program correctness verification utilities**

**Responsibilities :**

* These utilities (either singly or grouped) provide correctness verification analysis on C and C++ source languages. Analysis outputs optionally consist of fault source line location and counts of the various element openings and closings. These utilities consist of:
  + balanced bracket analysis
  + balanced parenthesis analysis
  + balanced braces analysis
  + balanced comment (opening and closing) analysis
  + balanced single-quote analysis
  + balanced double-quote analysis

**Source documentation formatting helper utilities**

**Responsibilities**

* These utilities serve as helper programs for the typeset processing of source document formatted text files (generally sourced in TROFF, TROFF-MM, or associated source input formats). These utilities are often embedded within printer utilities or printer facilities.

**Network de-multiplexing server family**

**Responsibilities:**

* This project consists of a family of server programs that feature a de-multiplexing front-end and which dispatches a resulting named service using a service-dispatch database.

**Name service facility**

**Responsibilities:**

* This is a project which functions to provide real names, organization names, and sub-organization names (or project names) for software program clients where only some type of user identifier is initially available.

**Enhanced UNIX login facility and management**

**Responsibilities:**

* This project enhances the normal UNIX login facility by providing system configured login environments for users who log into the system using normal login (itself depending on PAM) facilities.

**Enhanced Secure Login Server**

**Responsibilities:**

* The first added feature is the use of more than one possible host or user cryptographic keys to validate an incoming secure connection.
* The second major enhancement to the server is the allowance for additional process environment to be provided for spawned programs or logins

**UNIX adaptation layer software:**

* This project provides a software adaptation layer for all (or almost all) of the standard UNIX section-2 kernel and section-3 (all standard varieties) system kernel calls and library calls.

**Skills:** C/C++, UNIX, Shell, AWK, HTML, CSS, JavaScript, CGI, Git.

***A fairly large sample of code (over 1-million NCSL) can be found on GitHub at***

***http://GitHub.com/DavidMorano/RightcoreMainBase***

**Adros Energy, Amherst, NH Jan 2009 - Sep 2010**

**Analyst**

**Responsibilities:**

* Provided analysis and evaluation of power generation and conversion equipment and technologies.
* Principal work mostly related to solar power generation systems.
* Engineer custom power generation system solutions for customers.

**Philips Consumer Communications, Holmdel, NJ Jan 1997 - Oct 1997**

**Member of the Technical Staff)**

**Responsibilities:**

* Performed research into future cell phone designs. Worked on minimizing product cost, space, and power consumption.
* Performed software architecture work to port and rewrite the existing code base to a new computer system design.
* Researched novel CDMA, TDMA, and analog radio designs for miniaturization and multi-mode operation.
* Evaluated new processors for cell phones, resulted in the choice of using the ARM processor.
* Skills: C, UNIX, Ksh, MC68x11 asm., DSP-16 asm., VHDL, numerous documents, presentations.

**Lucent Technologies, Holmdel, NJ Oct 1995 - Dec 1996**

**Member of the Technical Staff**

**Responsibilities**

* Researched and developed custom CMOS ICs for switching systems (voice and data); patented.
* Researched various data-switching technologies, both custom in-house and ATM, for future data switching products.
* Performed software architecture work for implementing the software switching application on our switch-system hardware.
* Developed CAD and other tool software.
* Skills: C, C++, UNIX, Ksh, SPICE, ADVICE, RTOS, numerous documents, presentations.

**Northeastern University, Boston, MA ,**

**Graduated Aug 2007**

**Doctor of Philosophy (PhD)**

* Major: Electrical Engineering w/ concentration in Computer Engineering. Minors (two): Computer Science, Control Theory.
* Researched computer microarchitectures to facilitate very large-scale instruction level parallelism (ILP); four patents.
* Dissertation: Exploring Instruction Level Parallelism Using Resource Flow Execution. •Major research contributions: multipath speculative execution, and a novel framework for tracking very large-scale speculative execution results.
* worked on four major microarchitectural simulators: SimpleSim, LevoSim, FastLevo, OptiFlow. •Developed several research and analysis tools. •Software highly object-oriented (OO) and often used dynamic plugin object components.
* Skills: C, UNIX, Java, MIPS, Alpha, SPICE, MATLAB, HTML, XML, and AMPL, Ksh, Bash. •Available for perusal: 387k+ commented LOC, 253k+ NCSL C, publications, presentations (available).
* SimpleSim: simple execution-based sim. of MIPS machine w/ emulated OS, for behavioral analysis.
* LevoSim: complex execution-based sim. of MIPS w/ emulated OS, for full hardware component microarchitectural simulation; OO design: hierarchically nested software objects directly correspond to the hierarchically nested hardware components.
* FastLevo: medium complexity trace-based machine simulator for evaluating machine MIPS microarchitecture.
* OptiFlow: complex execution-based sim. of Alpha w/ emulated OS, for full hardware component microarchitectural simulation; fully OO w/ hierarchical software objects as w/ LevoSim.
* Tools: designed several software programs, OO w/ run-time pluggable shared objects; format conversions, development of trace storage technology, multitrace comparison and analysis, evaluation of static and dynamic target program behavior and characteristics.