Christopher MacGregor

chris@cybermato.com
206-650-0686

Objective

An opportunity to design and implement solid systems
in a Unix/Linux or embedded environment.

Summary of Value

- Highly motivated, very experienced senior-level software engineer
- Emphasizes robustness, quality, extensibility, reusability, and maintainability
- Excellent written and verbal communication skills
- Detail-oriented
- Generalist with a broad base of knowledge and significant depth in many areas
- Fast-learning self-starter
- Works well independently and/or in a team

Summary of Skills

- Design and implementation of large and small software systems
- C/C++/Perl/etc. in a Linux/Unix or Microsoft environment
- Embedded and real-time systems and software:
  - Large-scale embedded systems (e.g., client/server networked systems using MontaVista Linux on 32-bit RISC architectures)
  - Small-scale embedded systems (e.g., H8S/300 – 16-bit microcontroller, 16k RAM)
  - Kernels and drivers
- Development tools and infrastructure:
  - Compilers (front and back ends)
  - Instruction-level simulators
  - Libraries, middleware, runtime environments
  - Linkers, debuggers, etc.
  - Code generation tools
- API design
- Mentor/resource for other engineers

Experience

High-level languages:

Advanced C++ (including templates, STL, & exceptions), ANSI C, Perl (4 & 5), SQL, HTML, XML, PHP, Python, Java, Javascript, bash, sh, awk, Lisp, PL/I (Subset G), Modula-2, Pascal, BASIC, PostScript, Forth

Assembly Languages:

ARM, Thumb, TriCore, Hitachi H8S/300 & SH-3, some MIPS, some PowerPC, x86 (32-bit), MC680x0, MC88100, i80860, PA-RISC 1.1, i8086, MC6809, Z80, VAX/MACRO, MACRO/11

Operating Systems, Environments, etc.:

Linux (Redhat, Fedora, MontaVista), BSD Unix, Solaris, Unix System V r3 & r4, other Unix, Windows NT/95/98/2000/XP, Stratus VOS, VMS, MS-DOS, TOPS-20, AOS/VS, OS9, RT-11, X Windows, GNOME, GTK+, Qt, Iptables, Apache, Postfix, Cyrus IMAP, BIND (DNS), MySQL, TCP/IP, SIP (Session Initiation Protocol), YACC, Bison, Lex, GNU Emacs, Perforce, SVN, CVS, RCS

CYBERMATO CONSULTING (Seattle, WA): April 2003 – present
Principal Engineer – part time/occasional consulting – see cybermato.com/projects for more detail

UW Oceanography Dept. (4/2003 – present)

- Designed & implemented numerous significant new features and enhancements for an open source video editing program to adapt it to video analysis of real-world biological behaviors
- Designed & implemented a Linux interface to a velocimeter (3D water current measurement device)
- Designed & implemented a Linux interface to a stepper motor controller
- Current projects:
  - Designing and implementing a Linux interface directly to two stepper motors and some sensors
  - Optimizing a program that solves specific Partial Differentiation Equations

Local stealth-mode startup (12/2007 – present)

- Various tasks including porting Windows-based system to Linux, introducing portability layer and refactoring/restructuring along the way

DIGEO (Kirkland, WA): May 2005 – January 2008
Senior Software Engineer

Tools Group (8/2006 – 1/2008; Technical Lead)

- Designed, implemented, tested, and documented a fully automated, highly configurable system to build from scratch a complete set of GNU-based cross-development tools, plus Digeo's entire custom Linux distro, including constructing a full-blown Linux system from nothing. This system is deployed and working for ARM, MIPS, and x86 targets.
- Acted as a consulting resource to Engineering on various topics (advanced C++, Linux, etc.).

Exploratory Porting Group (5/2005 – 10/2006; Technical Lead)

- Investigated porting Digeo's Moxi software to the Motorola 6412P3 set-top box
- Designed and co-implemented an OpenGL subset to facilitate porting the Moxi software, including optimized blit operations incorporating rotating, scaling, and flipping in a single operation, using C++ templates to make the blits fast, portable, and configurable
- Designed and co-implemented a system to capture a trace of the OpenGL operations performed by Moxi, and to replay the trace using our OpenGL subset implementation – this provided an easily-ported but accurately representative graphics performance test for potential porting target platforms
- Investigated porting Moxi to an Amino AmiNET110 IPTV set-top box; reverse-engineered direct access to the framebuffer to significantly improve graphics performance, and built custom gcc 3.4-based cross-compilers because Amino's gcc 2.95.x couldn't compile Moxi
- Investigated porting Moxi to a Scientific Atlanta 8300 set-top box
- Investigated porting the Real Networks (RealPlayer) Helix client to Digeo's environment; extensively modified Broadcom's Linux device drivers to improve performance and implement missing features
- Acted as a consulting resource to Engineering on various topics (advanced C++, Linux, etc.).

TIGERWAVE NETWORKS (Seattle, WA): March 2003 – May 2007
Chief Technical Officer

Co-owned and handled all technical aspects of small web & email hosting company:

- Designed and partly implemented a customer-facing web-based control panel (Perl & MySQL)
- Implemented a text-based configuration generation system to automatically generate configuration files for the web, DNS, email, and other server systems from centralized customer information
- Managed all servers, networking equipment, software installation & configuration, system admin., etc.
- Postfix (& Sendmail), Bind, Apache, Cyrus-IMAPd, SpamAssassin, MySQL, RedHat, Fedora

- Invented a way to run FrontPage without the security holes (no setuid! modified suexec & FrontPage)
- Also did much of the sales and most of the technical support

SONY ELECTRONICS (Mountlake Terrace, WA): February 2001 – April 2005
Senior Staff Software Engineer

Embedded Software (4/2003 – 4/2005; member of Advanced Technology Group)

- Designed, implemented, tested, documented, and deployed several large new software components for advanced research projects and prototypes, using C++ in a Linux-based embedded environment, with Redhat/Fedora Linux as a development platform. Examples:
  - Client/server-structured configuration information subsystem for LAN-based multiple-head media system
  - Infinitely multi-zoned flexible debug/trace output management system built on the configuration information subsystem
  - SIP-based VoIP+video communication management system
- Acted as a consulting resource to the group on numerous topics (advanced C++, Linux, drivers, software development process issues, design for reuse, etc.).

LCD Monitor Firmware (4/2002 – 5/2003; Technical Lead for second phase of project)

- Extensively modified and enhanced to existing software to adapt it to different requirements for a new product and to dramatically improve maintainability and reliability, in part by replacing several painful and error-prone manual processes with automated systems.
  This was an H8S/300 16-bit microcontroller environment.

Automated Television Testing System (7/2001 – 4/2002; Technical Lead)

- Designed, implemented, tested, documented, and deployed fully automated system for testing various aspects of TV functionality, such as compliance with V-Chip requirements (EIA-608B).
- Designed and presented training for the group to whom the system was delivered.

Digital Television Code Review (2/2001 – 3/2001; contractor)

- Made major contributions to a code review of software for a Digital TV product.
- Proposed and prototyped several significant enhancements to functionality and performance.

BSQUARE CORPORATION (Bellevue, WA): November 1994 – May 2001
Senior Software Engineer, Technical Lead

Linux Investigation/Experimentation (4/2001 – 5/2001; Technical Lead)

- Technical lead for very small team investigating and prototyping ports of Bsquare's proprietary technology to a Linux platform.

TriCore Compiler Port (10/1999 - 1/2001; Technical Lead)

- Technical lead for team porting Microsoft's retargetable back end to target the TriCore architecture.
- Designed creative solution to maintain backward source-level compatibility despite incompatible function calling convention (floating point args passed in floating point registers – bad for varargs)

ARM/Thumb Compiler Port (1/1997 - 12/1999; Senior Software Engineer)

- Part of a team porting Microsoft's retargetable back end to target ARM (architecture 4), and later Thumb. Was second-in-command, providing technical leadership to most of the team.
- Designed and implemented special peephole optimizer to take advantage of unusual aspects of the ARM architecture.
- Designed and implemented ARM disassembler.
- Enhanced ARM's simulator to make some use of Microsoft debugging information.

SH-3 Compiler Port (11/1994 - 3/1997; Senior Software Engineer)

- Part of a team writing and rewriting a back end targeting the Hitachi SH-3 processor in the Microsoft environment (MS front end, COFF, linker, etc.). My work focused on code generation, low-level optimization, and bit assembly.
- Designed, implemented, tested, and deployed an assembler for the SH-3, designed to be compatible with Hitachi's assembler, producing MS-COFF (including some debug info).

Miscellaneous (1994 - 2001)

- Worked with various customers, primarily Infineon, Advanced RISC Machines (ARM), Hitachi, and Microsoft.
- Initiated a relationship with the engineers at Microsoft who maintain their compiler back end, linker, and disassembler. Acted as the primary interface in that relationship while continuing to build and broaden it. This included delicate handling of some difficult personalities. The majority of the interaction was via email, with occasional phone calls and extremely rare face-to-face meetings.
- Enhanced C runtime in various ways.
- Debugged and fixed various WinCE kernel bugs (mostly related to exception handling, plus a few originally thought to be compiler bugs).
- Proposed & implemented numerous initiatives to improve our development environment, including many new tools.
- Wrote a sophisticated set of customizable, extensible build scripts for the development tools (in Perl 5).
- Acted as a resource for other engineers.

GTE NETWORK MANAGEMENT OPERATIONS (Bothell, WA): Nov. 1993 - Dec. 1994
*Software Engineer / Systems Consultant (contract through **FDSI**)*

Configuration Management Tool Development

- Designed, implemented, tested, and documented powerful new CM tool using Perl 4, RCS, and Sybase (SQL Server).
- Maintained and supported existing CM tool while new version was under development.

STRATUS COMPUTER, INC. (Marlboro, MA): May 1987 - August 1993
Software Engineer

Source Control & Configuration Management Tools

- Developed and maintained a toolset to support and automate source control/configuration management, build management, and regression testing. This is still in use today.
- Ported new Stratus proprietary source control system to VOS (Stratus' proprietary operating system).
- Developed and maintained powerful make front end (in Perl) and makefile framework to support local build environments. This is still in use today.

GNU Assembler

- Enhanced the GNU assembler (gas) to produce object modules for VOS for a PA-RISC based system under development; added features; fixed pre-existing bugs; ported gas to run native on VOS.

GNU and Unix Tools; Unix Compatibility Library

- Ported many GNU and other standard Unix tools to VOS; maintained and enhanced those ports.
- Modified GNU Make, GNU Emacs, gcc, Perl, Bison, RCS, and many others.
- Co-developed and maintained a tcsh-like shell for VOS.
- Developed large library of C functions and header files to: provide or emulate standard Unix system calls and runtime functionality missing from VOS; provide ANSI C and POSIX compliance; ease porting of standard tools and other software. This library is still in use today.

Instruction-Level Simulators

- Designed, implemented, and tested an i80860 instruction-level simulator (ILS) with support for branch-level (C1) code coverage analysis, using GDB as a front end.
- Maintained an MC88100 ILS used for all software development for Stratus MC88100-based systems.
- Enhanced MC88100 ILS, on own initiative, to support:
  - Full-screen and symbolic debugging.
  - Branch-level (C1) code coverage analysis.
  - Fully automated regression testing of assembly language code, including a test description language with byte-level read/write memory access limiting.

MC88100 Assembly Language

- Designed, implemented, and exhaustively tested high-performance language runtime support routines for Stratus MC88100-based systems.
- Implemented high-speed variable-alignment data moves and compares, varying-length string operations, sophisticated (PL/I) data type conversions, and many more.

GCC on VOS (MC680x0 and MC88100)

- Enhanced GCC (the GNU C compiler) to support VOS calling conventions and stackframe layout.
- Modified GCC back ends for MC680x0 and MC88100 targets to produce code compatible with VOS assemblers.
- Ported GCC to run native on VOS.

Resource and Consultant

Acted as a general resource and consultant to Stratus Software Engineering on various topics:

- VOS internals
- Unix internals
- GNU and other standard Unix tools
- Porting issues (Unix vs. VOS, Unix vs. Unix)
- Unix systems programming
- Source control, make, and build issues

COMPLECS ENGINEERING (Self): 1987 - 1990

Designed and implemented a sophisticated programmable concert and theatrical lighting effect control system, including:

- Text-window-based user interface with mouse support (in Turbo C on MS-DOS)
- Interrupt-driven realtime kernel
- Lighting control language:
  - Compiled by YACC (Bison)-based compiler
  - Compiler output interpreted by realtime kernel

SOFTWARE INTERPHASE, INC. (Providence, RI): 1984 - 1986

- Designed and implemented a powerful multiuser electronic BBS in Modula-2.
- Designed and implemented Modula-2 libraries for dynamic string handling, file I/O & manipulation, and low-level system functions.

Education

Bachelor of Science in Computer Science, with distinction: May, 1990
Worcester Polytechnic Institute (Worcester, MA)

Community Involvement

Seattle Pro Musica – non-profit semi-professional choral group

- Board member 1999-2006
- Volunteer 1997-2006
- Technology go-to guy 1999-2007
- Wrote auction management database application, in use 2000 - present
- Wrote web-based ticket/CD sales system, in use 2002 - present

USArefs.com – volleyball referee association

- Wrote web-based schedule display system

On behalf of Seattle public school community and my kids

- To determine the real truth of how the Seattle Public Schools school-assignment system works, negotiated access to the School District's source code for the assignment system, reviewed it, and reported the result to the community – see www.cybermato.com/projects/school-assignments

Hobbies

Various, including **stunt computing**:

- Using gdb to change the current directory of a running process to release a locked NFS inode
- Using mount --bind and chroot to swap out /bin, /lib, etc. on a running Linux system without rebooting)
- Using gdb to bypass the password protection on a running screen session

Other Skills

- Basic digital electronics
- Basic electrical (residential) wiring; network & phone wiring

References

See LinkedIn or www.cybermato.com/chris/references;
Interactive references can be arranged on request

Experience & Skill Summary

The skill levels shown indicate my evaluation of my proficiency as a programmer, and as a user and/or system administrator (if applicable), in the listed environments, at the time when I was using them. The range is from 1 (novice or just learning) to 5 (thoroughly familiar with and comfortable in the environment). The Rust Factor is an indication of how "rusty" the skill has become, on a scale of 0 (very fresh) to 5 (it's been a while). (For instance, it's been a long time since I used BASIC or Modula-2, and I use PostScript only sporadically.) I started programming in BASIC in 1976, then Z80 assembly language around 1979, & 6809 assembly language in the early 1980's.

Languages:	**Years Used**	**Skill** 1-5	**Rust Factor** 0-5
C	19+	5	0
C++ (including templates, exceptions, etc.)	15	5	0
Perl	18	5	0
SQL (MySQL, Sybase, Oracle, MS Access)	8	5	1
HTML	7	4	1
XML	1	2	3
PHP	1	2	2
Python	1	1	2
Java/Javascript	1	1	3
bash, csh, awk, etc.	10+	5	1
Lisp	4	4	2
PL/I (Subset G)	6+	5	4
Modula-2	2+	5	5
Pascal	4	5	5
BASIC (from TRS-80 to Visual Basic)	13+	5	5
PostScript	3	4	3
Forth	1	3	3
YACC, Bison, Lex	~5	5	3

Assembly Languages:	**Years Used**	**Skill** 1-5	**Rust Factor** 0-5
ARM & Thumb	4	5	4
MIPS	1	1	4
PowerPC	< 1	1	5
Hitachi H8S/300	1.5	3	4
Hitachi SH-3	3	5	4
TriCore	2	5	4
PA-RISC 1.1	< 1	3	4
MC680x0	6+	4	4
MC88100	2	5	4
i80386	2	3	4
i80860	2	4	5
i8086	6+	4	5
6809	3+	5	5
Z80	6+	5	5
VAX/MACRO	1	4	5
MACRO/11 (PDP-11/23+, etc.)	1	5	5

Experience & Skill Summary

(continued from previous page)

Operating Systems, Environments, etc.	Years Used	Prog. Skill 1-5	User/Admin Skill 1-5	Rust Factor 0-5
Linux (Red Hat, Fedora, MontaVista, etc.)	7+	5	5	0
BSD Unix (various, including SunOS)	6	5	5	3
Unix System V r4 (various flavors)	2	4	4	5
Unix System V r3	1	4	3	5
Windows NT/95/98/2000/XP	13	4	5	3
Stratus VOS	6+	5	5	5
VMS	1	2	3	5
MS-DOS	8+	5	5	4
TOPS-20 (DEC-20)	3	2	5	5
AOS/VS	2	2	5	5
OS9	1+	3	5	5
RT-11 (PDP-11/23+)	1	2	4	5
TRS-DOS (various flavors)	~6	5	5	5
GNU software	21+	5	5	0
X Windows (R3-R6), GNOME	12+	4	5	1
GTK+	1+	3	NA	0
Qt	< 1	1	NA	1
Emacs (TECO, GNU, Micro, Stratus)	22	3	5	0
Iptables	5	0	5	1
Apache, Postfix, Cyrus IMAP, BIND (DNS)	5	0	5	1
MySQL	3+	4	4	1
Microsoft Access & Visual Basic	2+	4	4	5
TCP/IP	~3	4	NA	2
SIP (network protocol)	1+	4	4	3
Perforce	3+	1	4	0
Subversion (svn)	2+	0	4	0
CVS	5	0	5	2
RCS	4+	5	5	5
Various other SC/CM systems	3+	0	3	5