# Francis Litterio

774-573-9321
flitterio@gmail.com

I have four decades of software development experience with a variety of programming languages, operating systems, and (more recently) AI technologies. My career experience includes system software development, custom GPT development, TCP/IP networking, and compiler/debugger development. I learn quickly and possess excellent technical, interpersonal, and written communication skills. I can read, write, and debug assembly language. I keep up-to-date with modern AI technologies, using both frontier and local LLMs. My contributions to open source projects include implementing the regular expression operator (=~) in Bash and fixing bugs in GNU Emacs.

## Technical Summary

| | |
|---|---|
| Operating Systems: | Windows, Linux, Solaris, HP-UX, AIX, PRIMOS, TOPS-10, RSTS/E. |
| OS Emulators: | Cygwin, MSYS. |
| Virtualization: | VMware Workstation, VirtualBox. |
| Languages: | C++/C, C#, Python, PowerShell, Emacs-Lisp, various assembly languages (see *CPU Architectures*), UNIX shells (Bash, Bourne shell, C shell, Z shell). |
| Coding Assistants: | Cursor/Windsurf/Cline, Github Copilot. |
| APIs: | Windows (user-space and kernel), .NET, UNIX/Linux, Berkeley Sockets. |
| Protocols: | TCP/IP suite (IP, TCP, UDP, ICMP, DNS, ARP, etc.), HTTP, DNS. Experience with WireShark. |
| CPU Architectures: | Intel x64/x86, Sun SPARC, HP PA-RISC, Motorola 680x0 and 880x0, Prime 50 Series. |
| Revision Control: | Git, GitHub, Perforce, ClearCase, RCS, Subversion. |
| Publications: | *Supporting Unicode in C/C++*, *Git Overview* (available on request) |

## Employment Experience

**IntervalZero, Inc., Waltham, MA — October 2010 to August 2024**
Principle Software Engineer responsible for designing and implementing new features and improvements to RTX64, IntervalZero's flagship real-time operating system that co-exists with Windows on a single machine. Developed primarily in C++, C, and C#. Implemented new features and improvements to user-space libraries and the real-time kernel device driver, including RTAPI (the user-space and real-time APIs for RTX64 clients) and the RTX64 Managed Code Framework (a managed .NET assembly wrapping the native RTAPI APIs).

Designed, implemented, and maintained the RTX64 Monitoring Framework, a mix of user-space and kernel components that enable customers to monitor internal activity in the real-time kernel with minimal impact on real-time latency. Implemented a high-performance pipe to transfer monitoring data from RTX64 to Windows, supporting multiple concurrent real-time writer threads and a single Windows-side reader thread. The pipe uses no software locks and just one hardware lock. Worked closely with Percepio Software, the third-party company that makes Tracealyzer, a graphical tool to visualize monitoring data. Designed, implemented, and maintained user-space .NET APIs for configuration and control of the Monitoring Framework and for reading pre-recorded monitoring data.

Designed and implemented support for thread-local data in real-time binaries. This required reverse engineering how the Microsoft compiler, C runtime, and Windows kernel loader implement thread-local data, then implementing equivalent functionality in the RTX64 real-time kernel. This work was driven by the fact that Microsoft's C++ compiler and runtime library implicitly use thread-local data, even when the user's application code does not. Had this not been solved, we would have had to remove support for C++ in real-time code.

Helped implement a new licensing system based on the Reprise License Manager (RLM), which included a Windows service responsible for validating licenses in user-space and communicating license status to the real-time kernel. Designed and implemented a "stamping" utility to insert into real-time executables and shared libraries the status of RTX64 SDK licenses present at build-time. Modified the real-time kernel to validate the SDK license information in real-time binaries as they are loaded.

Implemented a custom GPT (based on OpenAI's GPT-4) to answer questions and give coding advice about the RTX64 real-time APIs. The custom GPT leveraged GPT-4's Web search functionality to obtain factual data from IntervalZero's online documentation, which worked far better than using Retrieval Augmented Generation (RAG) to search the documentation. The GPT was guard-railed against writing code for the customer and instead referred them to our published code samples.

**EMC Corporation, Hopkinton, MA — July 2008 to October 2010**
Principle Software Engineer responsible for designing and implementing Linux-based Common Information Model (CIM) management software for EMC's next generation mid-range storage platform, focusing on monitoring system health and transactional provisioning operations. Wrote Managed Object Format (MOF) interface definitions for our CIM object model, then implemented those MOF interfaces as C++ plugins for an existing CIM server. Assisted developers who were implementing SOAP client/server software using an open source SOAP framework. Provided C++ and Linux expertise to my group.

**Wind River Systems, Canton, MA — November 2006 to July 2008**
Member of Technical Staff responsible for designing and implementing improvements to Workbench (Wind River's Eclipse-based debugger IDE) and to Wind River's host-based debugging tools (targeting Windows XP, Red Hat Enterprise Linux, and Solaris). Designed and implemented an Eclipse-based ANSI terminal emulator and TELNET client to act as a host-side console for remote embedded devices. I was the resident Linux guru in a team consisting primarily of Windows developers.

**Atovia Networks, Westford, MA (startup) — August 2003 to November 2006**
Principal software engineer responsible for designing and implementing a Linux-based multithreaded distributed filesystem policy management system intended to load-balance NFS and CIFS traffic across filers. User Mode Linux was used as a simulation environment while hardware was being developed. Coding was done in Standard C++ with a combination of the C++ Standard Template Library (STL) and Berkeley DB as a local data store. Interprocess communication was performed using the Sun RPC protocol.

**Pyxsys Corporation, Sudbury, MA (startup) — October 2000 to August 2003**
Principal software engineer responsible for designing and implementing the software components of an intelligent disk controller for a Network Attached Storage (NAS) product: a proprietary board running Montavista's Hard Hat Linux on an IBM PowerPC 405GP processor attached to 12 TB of RAID storage. Designed and implemented a power-failure data protection mechanism triggered by a loss-of-power interrupt where volatile buffers were copied to battery-backed memory before the board lost power. Upon restoration of power, the driver restored the volatile buffers.

**PictureTel Corporation, Andover, MA — August 1997 to October 2000**
Principal software engineer responsible for designing and implementing the H.320 Call Manager and H.221 protocol device driver components of PictureTel's next-generation videoconferencing system. Challenges included learning the Microsoft Distributed Component Object Model (DCOM) programming framework and re-engineering the existing NT code base to use DCOM instead of a proprietary message-passing framework. Designed and implemented changes to NT kernel-mode device drivers for Live200, PictureTel's NT-based video-conferencing application. Challenges included coming up to speed rapidly with the wide array of ITU-T video-conferencing standards, including the core standards of H.320, H.242, H.221, and H.230.

**Aurora Systems, Inc., Acton, MA — June 1995 to August 1997**
Senior software engineer responsible for designing and implementing a Windows telephony API (TAPI) device control subsystem that allowed call-center agents to manipulate NORTEL digital phones via FastView, the company's computer telephony integration (CTI) product. Designed and implemented a TAPI-aware state-machine to manage time-delayed responses from the NORTEL TAPI service provider. Tracked a shifting set of functionality and defects in both the NORTEL TAPI service provider and in the NORTEL Meridian PBX via weekly conference calls with NORTEL developers. Responsible for updating software on an in-house NORTEL Meridian PBX and programming the PBX to configure new digital and analog lines. Advised other developers how to leverage the Win32 API and TCP/IP protocols in the 32-bit version of a 16-bit product.

**CenterLine Software, Inc., Cambridge, MA — November 1989 to June 1995**
Senior software engineer responsible for implementing improvements to ObjectCenter and CodeCenter, the company's C++ and C interactive development environments. Re-engineered ObjectCenter's dynamic linker to support dynamic loading and unloading of HP-UX object files on the HP 9000 Series 300/400 workstation. Designed and implemented the stack unwinding and interpreter/object-code procedure call subsystems for CodeCenter on the IBM RS/6000 workstation. Lead engineer for a three-engineer project to bring ObjectCenter to market on the HP Series 700/800 workstation, which included overseeing the implementation of the dynamic linker and implementing the debugger symbol table loader and static initialization/finalization support for C++ shared libraries. Consulted to another team about how to implement the object-code instrumentation subsystem of TestCenter, CenterLine's run-time error detection product, on the HP PA-RISC processor.

**Prime Computer, Inc., Framingham, MA — July 1983 to October 1989**
Software engineer responsible for implementing debugger symbol-table support in the common backend of a suite of UNIX C, Pascal, and Fortran compilers targeting a MIPS-based UNIX workstation running System V Release 3.2. Extended Prime's Emacs editor with Ada Language Mode that was Ada-syntax aware. This required implementing an Ada lexer/parser and integrating it with Emacs-Lisp code. Technical liaison between Prime Computer and Pacer Software, the ISV that provided the Prime C compiler and run-time library. Assisted other departments as they converted their source code from PL/I to C.

## Education

**Worcester Polytechnic Institute, Worcester, MA — Class of 1983**
Major: Computer Science