

31 Syracuse Road
Nashua NH 03064-1752
603-943-0330 (cell)
603-880-0482 (office)
burt.janz@gmail.com
bhjanz@ccsneinc.com
<http://bhjanz.ccsneinc.com>

Burton H. Janz

Overview

A principal-level software engineer with mission-critical architecture design and coding skills, and successful team leadership experience.

Current Skills (at or approaching expert level)

- C; bash/sh
- Linux drivers/kernel internals; embedded Linux
- NXP LX family; ARM Cortex-A15; Xilinx Zynq; Intel; Freescale I.MX5/6; TI OMAP
- Yocto; custom build tools
- GIT (porcelain); Jenkins; Agile (Atlassian Jira)

Additional Skills (within past 3 years)

- C++; STL; BOOST; Python; Perl
- TCP/IP (client/server)
- GIT (plumbing); CVS/SVN
- IBM/Freescale PPC (various); Intel/AMD x86
- Redmine; Ansible
- Windows
- Wordpress (beginner)

Previous Experience (longer than 3 years ago)

- VXWorks; Microsoft Window internals; VMS; RSX; other UNIX
- UDP protocols; RPC; XML; HTML; SOAP; NFS/SMB; COM/DCOM; other networking systems/protocols
- UNIX (SYSV, BSD) drivers/internals; Microsoft drivers; RAID; SCSI; other device drivers and kernel internals
- Android; Java/J2ME/MIDP; Microsoft Windows CE Embedded; MetroWerks for Palm; other embedded OS and IDEs
- ClearCase; RCS
- SQL; BASIC/VB; other languages
- DOORS; Bugzilla; other bug tracking systems
- Omnivision 5xxx
- Apache; Samba; other networking services

Work History

August 2017 – Present: BAE Systems, Nashua NH

Software Development

Principal Software Engineer

Program 1:

- Created a core Yocto build environment and a GIT repository for a new program consisting of 3 separate hardware configurations in a “common boot” configuration
- Developed/tested multiple Yocto meta layers, one for each of 3 separate hardware configurations
- Wrote/tested numerous Yocto recipes to generate multiple MACHINES and IMAGES for each hardware

configuration (swdebug, hwdebug, ship, etc)

- Developed/tested multiple u-boot configurations to allow building 3 separate hardware configurations – all changes driven by Yocto recipes to prevent modifications to u-boot source code (re GPL2)
- Developed cross-boot mechanism for multiple-board embedded system (3 boards boot from the 4th board)
- Developed/tested u-boot secure boot and cyber protection code for one of the hardware configurations
- Developed/tested kernel configurations for each of the hardware configurations
- Developed/tested “drop-in” build mechanism to allow building appropriately in open and closed labs
- Provided Jenkins-compatible build mechanisms using Yocto
- Wrote numerous WIKI pages for new user “handholding” when developing under Yocto and to describe cyber protection mechanisms and their implementation
- Answered numerous questions about Linux: bash scripts, boot configurations, Python code, etc

Program 2:

- Assisted with development of seL4-based software environment
- Debugged boot problems with 3rd-party hardware platform
- Wrote/rewrote scripts to configure and build the seL4-based product, including kernel and Buildroot modifications

Program3:

- Investigated 3rd-party memory devices for new ARM 32-bit system – documented memory interfaces
- Modified u-boot for ARM 32-bit RDB to use external programs for startup auto-test code
- Added interfaces for custom hardware, modified device tree accordingly.
- Investigated memory and Yocto build processes for ARM 64-bit system
- Modified Yocto build to eliminate requirement for Ubuntu “docker” (non-DFARS operating system)
- Modified u-boot for ARM 64-bit RDB to add external programs for startup auto-test
- Added interfaces for custom hardware, modified device tree code accordingly.
- Wrote documentation in Confluence documenting builds, interfaces, and test methods for both processors

May 2017 – August 2017: Harris Corporation, Rochester NY

Software Development / Toolsmith (embedded)

Contractor

- Created a Yocto build environment and appropriate GIT repositories for a new custom hardware platform.
- Configured u-boot for non-reference design platform-specific memory and eMMC configurations.
- Modified i.MX manufacturing mode configuration to allow loading eMMC from external exported file system (no flash on system – no way to load eMMC except via i.MX chip-level commands).
- Ported existing Harris code into Yocto-generated file system via new recipes and machine descriptions (all builds were from GIT).
- Wrote extensive software requirements and implementation documentation.
- Worked offsite (most work done in home lab – 1 wk/mo in Rochester NY)

October 2016 – April 2017: Arris Corporation, Lowell MA

Software Development / Toolsmith (embedded)

Contractor

- First hire in new set-top box (STB) group.
- Tailored new software distribution from Broadcom for new hardware.
- Wrote Python, Yocto recipes, and Jenkins scripts to create, maintain, and upgrade build environments.
- Wrote manifests for “repo” tool and established new GIT repository layouts.
- Developed Yocto “patch autogeneration” class and methodology to create GIT software distribution patches on-the-fly for outside partner’s consumption.
- Assisted with EMMC partition design and consulted on kernel build methods.
- Mentored other team members in Yocto, build methodologies, GIT usage, Linux configuration and boot methods.

August 2015 – September 2016: Harris Corporation, Nashua NH

Principal Embedded Engineer

IST (secret clearance required)

Development

- Made changes to Zynq 7020 u-boot sources to load a pre-boot FPGA bitstream to allow access to a GPIO in PS space and access to the Zynq's device DNA.
- Rewrote sections of the Zynq FSBL to permit a timeout-based two-stage auto-switch boot mechanism similar to the two-stage boot selector built into the OMAP.
- Wrote BOOST "coverage" tests for existing application software (not for testing on target platform).
- Proposed implementation of ANSIBLE-based unit test procedures for embedded platform.
- Extended implementation of existing Python-based black-box product test code.
- Designed and wire-wrapped board to remotely enable/disable USB d+/d- lines to simulate insertion/removal of USB drive.
- Reconfigured installation boot media code to use customized version of syslinux.
- Investigated network connection issues – worked with IT on resolution.
- (Classified aspects of work are not described here.)

May 2014 – July 2015: MIT Lincoln Laboratory, Lexington MA

Software Development / Toolsmith (non-embedded)

Contractor

Cyber Security Division – Group 59: Cyber Assessments (Secret clearance required)

- Created and documented classifications for types of cyber security invasive/non-invasive situations.
- Wrote the multi-threaded multi-tasking OS-agnostic Performer Emulation Module to demonstrate cyber security classifications to future BAA performers.
- Created scripts to allow Ansible to fully control the PEM. Goal: prove to performers that automated assessment is possible and provide a demonstration system for their use.
- Wrote tools and ported/enhanced existing open-source applications for use in analyzing existing software to find weakness in algorithmic design.

Exposure to llvm, mcsema, klee, clang, and other similar tools.

(Classified aspects of work are not described here.)

March 2014 – April 2014: Vapotherm, Exeter MA

Device Driver Work / System Engineering Work

Contractor (short term project while waiting for approval for long term work)

Two projects, both are medical devices used for oxygen therapy.

- Investigated PIC-based system to fully document debug interface for use as a system control protocol.
- Debugged resistive touchscreen driver for medical device and added variation between touch and release to reduce false positive / false negative detection.
- Debugged screen color issues on ARM9-based Linux device (color model required BGR, not RGB).
- Modified u-boot and kernel to allow use of EXT3 filesystem to reduce filesystem damage on power loss.
- Began conversion of IAR-based ARM7 code to non-proprietary environment (not completed – company ran out of funds).

November 2013 – February 2014: Vislink, Billerica MA

Device Driver Work / Configuration Management Work

Contractor (contract-to-hire – project terminated 2/14/14)

Project is a system for reception and retransmission of video for broadcast systems.

- Began writing HDMI driver for ADV7611 to be tested on Xilinx Zynq ARM Linux 3.10 kernel and backported to Ubuntu x86-based 3.6 kernel. (Work specified by manager.)
- Rewrote GPIO multiplexer code.
- Rewrote touchscreen driver to use interrupts from GPIO multiplexer via I2C bus expander.
- Created VirtualBox VM as development platform to augment company's only build machine.
- Created and tested nightly build scripts on both build machine and VM.
- Ported external SVN repository to SVN to validate move from external repo to local repo.
- Supported company efforts by allowing external Vislink personnel to download VM from personal ftp site (using

system in my home office)

November 2010 – October 2013: L3 (Warrior Systems – Insight), Londonderry NH

Embedded Linux Systems Development (kernel, system, drivers, and applications)

Senior Software Engineer (laid off 10/21/13)

Secret clearance granted January 2011

Project 1: Hired to take control of project that had reached a dead end. Led a team of 3.

- Built custom development environment for embedded Linux using Kconfig, Makefiles, PERL and Bash scripts that built JFFS2/YAFFS containers as well as supporting NFS boot.
- Ported project from TimeSys to local environment cutting build from 6 hours to 20 minutes.
- Wrote GPIO driver for proprietary hardware interface to process button events.
- Wrote interrupt-driven controls for other proprietary hardware (information restricted).
- Rewrote Omnivision camera driver to support multiple resolutions.
- Wrote software function to rotate video frames 90 degrees off-axis (VRFB unavailable).
- Ported GTK components from LTIB to local environment.
- Wrote flash file system manipulation tools to support application-based persistent storage.
- Implemented u-boot scripts to support automatic system update via SD card,.
- Directed other members of team and assigned tasks.

Project 2: Rewrite Project 1 to eliminate GTK, speed up graphics and hardware interface.

- Rewrote all software routines to eliminate use of GTK.
- Wrote camera capture code to place captured video frames in a shared memory.
- Wrote graphics library based on DirectFB providing icon, shape, label object support.
- Rewrote button manager using separate process using semaphore signaling.

Project 3: Implemented new Linux application architecture. Led a team of 2.

- Designed and wrote a Linux-based embedded application execution environment to emulate a multiple-micro controller system (MPA).
- Created scripts to convert spreadsheets into state machine C source code.
- Development/prototyping is performed on a Linux workstation.
- Prototyped Project 2 application on desktop and demonstrated on Phase 1 embedded hardware.

Prototyped Project 3 non-GUI application on desktop and i.MX6 embedded system.

- Incorporated project from a different group into development environment.
- Built and reconfigured u-boot and 3.0 kernel for use on i.MX6 prototyping board.
- Wrote Android demo program for new board using Android SDK (simple “hello world”).
- Began of Android GUI “slave” interface to control Android GUI via system messages.
- Directed other member(s) of the team and assigned tasks.

January 2010 – October 2010: Ingersoll Rand Security Systems, Hartford CT

Embedded Linux Systems Development (kernel, system, drivers, and applications)

Contractor (contract-to-hire - building closed 10/30/10)

Embedded Linux Kernel Development

- Ported existing SuSE/LxNET embedded development environment into VirtualBox.
- Implemented NFS boot via modifications to u-boot.
- Configured new development toolchain with modifications for IRCO systems.
- Reconfigured NET-ARM kernel to eliminate unused device drivers.
- Rewrote IRCO-specific 2.6.12 device drivers for use under 2.6.32.
- Modified existing 2.6.32 drivers (i2c, gpio, eeprom) to support IRCO devices.
- Rewrote NET-ARM BSP for 9750 with mods/conversions for 9360.
- Ported IRCO applications from LxNET environment to new environment
- Wrote parent process to control IRCO application startup/shutdown and auto-reboot.
- Created Subversion source code repository, including web and svn: access methods.

- Directed other member(s) of team and assigned tasks.

September 2009 – December 2009: Crossbeam, Boxborough MA

Embedded Linux Systems Development (boot systems and applications)

Contractor (contract-to-hire – project terminated 12/2009))

New Installation Device

- Analyzed existing NFS installation server (tftpboot w/ NFS root file systems)
- Wrote functional document describing a USB flash drive installation device to replace NFS "install server"
- Coded initialization scripts, building development environment, and porting kernels and platform tools for use within a USB-based boot/install device.

March 2007- April 2009: Cisco, Boxborough MA

Linux Systems Development (kernel, system, drivers, and applications)

Software Engineer IV (laid off 4/2009)

New Platform Port

- Analyzed VMWare ESX for port of Cisco Security Agent to virtual system host console.
- Modified VMWare ESX initrd to support loading CSA kernel modules without affecting ESX "ownership" of system hardware.
- Built network frame with multiple HP blades (via IPKVM) for development of new port.

Existing Platform Maintenance

- Rewrote existing CSA Linux kernel module interrupt handler to convert from per-network-packet hardware interrupt handler to software interrupt handler.
- Rewrote C code network packet processing handlers to use Linux NAPI "filter".
- Created new filter insertion routines to perform modification of L2/L3 packet contents.
- Modified and improved multi-platform (Linux, Solaris, Windows) library/executable code.
- Ported some library routines between Linux and Solaris, and Linux and Windows.
- Began port of existing CSA product to RHEL5..

Level 3 Support

- Miscellaneous bug chasing in CSA C code for Linux, Solaris UNIX, and MS Windows.
- Assigned as adjunct to Q/A group to perform fix "verification".

May 2006 – February 2007: Sanmina-SCI, Huntsville AL

Embedded Linux Systems Development (kernel, system, drivers, and applications)

Contractor

Embedded Linux development

- Assisted with hardware design for multiple 802.11 VOIP systems based around Xilinx Spartan-3E processor (MicroBlaze core).
- Assisted with hardware design for a new VOIP system based around Xilinx Virtex-4 processor (PPC405 core).
- Ported 2.4 and 2.6 Linux kernels to Spartan-3E and Virtex-4 platforms.
- Ported u-boot to Virtex-4 platform.
- Extended development environment to allow simultaneous development of Spartan-3E system (2.4 kernel) and Virtex-4 system (2.6 kernel), uClib versions 2.26 and 2.27.
- Generated design and implementation documentation for both platforms.
- Evaluated multiple embedded Java Virtual Machines for use in embedded environments.
- Implemented scripts and procedures for management of x509-encoded ssl certificates
- Directed other members of team and assigned tasks.

December 2005 – March 2006: Sepaton, Marlboro MA

Linux Kernel Modules

Contractor

- Analyzed command requirements for CLI and developed skeleton "proof of concept" code. Tasks included writing specifications and developing C skeleton code for intermediate code library (callable from CLI, SNMP, and other non-interactive interfaces), writing specifications and developing C skeleton code for platform-specific code

libraries (callable from intermediate code library)

- Re-designed the architecture and re-implemented a tape library simulation compression driver subsystem to allow addition of hardware devices as an adjunct/replacement for software devices. Tasks included: rewriting queue and thread management C code; implementing callback-based device management to allow insertion/removal of compression device modules on-the-fly; implementing VTL simulator to permit development of compression driver modules and allow test/timing of compression devices in a VTL-emulated manner.

Feb. 2005 – Oct. 2005: Siemens Corp., Chelmsford MA

Embedded Linux Systems Development (kernel, system, drivers, and applications)

Contractor

- Designed and implemented u-boot 1.1.2 class/device IIC driver for proprietary PPC440GX daughter board
- Ported Linux 2.4 kernel (MontaVista) to proprietary PPC440GX-based daughter board (TLB changes, PHY changes, flash/RAM changes, networking modifications, etc.)
- Modified early versions of u-boot 1.1.2 to include methods, environment, and other C code to permit automated NFS boot for group-wide testing.
- Designed and implemented multiple changes to Linux build (Makefiles, Config.in files, etc.) to generate platform-specific boot kernels for NFS test.
- Implemented NFS root file system group-wide boot methods for Linux on PPC440GX-based daughter board: created “template” file system, created scripts to create per-user file systems, created TFTP boot images, etc.
- Modified Linux kernel to use RAM-based file system, and assisted in modifications to u-boot to allow memory-only TFTP boot (non-flash).
- Implemented Compact Flash driver in Linux kernel, and implemented multiple changes to in-house environment to support single-image RAM-only boot from CF, TFTP, or NFS.
- Wrote init and other scripts to properly start embedded daughter board when used in multiple base cards, as well as writing other tools and assisting other developers working on other base cards.
- Directed one other member of team and assigned tasks (creation of boot CF card, testing code, modification of u-boot to add status LED flashes, etc.)

May 2004 – September 2004: Optovia Corp., Acton MA

Embedded Linux Systems Development (kernel, system, drivers, and applications)

Contractor

- Designed and implemented interrupt-driven SPI (Serial Peripheral Interface) driver, GPIO driver (signal control and sense for external devices), I2C “bit-bang” driver (using general-purpose timer pins), and I2C multi-master multi-channel interrupt-driven device driver for MPC5200
- Designed and implemented test programs to verify device driver functionality, and worked with hardware engineers to expand/enhance test programs as required
- Analyzed requirements to attach Philips PCA9564 I2C controller to MPC8270 (chip select, interrupt handling, etc.)
- Implemented configuration macros, TCL script modifications, and network interface modifications to allow single kernel source support for multiple platforms (Motorola MPC8260 and MPC5200)
- Modified u-boot for 5200 to include additional environment variables, changes to network interface for customized backplane, support/removal of other devices.
- Analyzed requirements to add non-standard I2C device drivers, flash memory, and IP address auto-configuration to u-boot and Linux kernel for multiple devices (MPC 8260/8270, MPC5200, PCA9564)
- Generated documentation (including data flows) for planned use of OpenIPMI via SMBus (I2C)
- Patched development kernel w/ RTAI extensions and began suitability tests

October 1996 – March 2004: Multiple Employers

Embedded and Enterprise Linux Systems Development (kernel, system, drivers, and applications)

Contractor

- Polaroid (4 years): Principal engineer / project leader for Indiana and West Virginia Digital Driver License Programs (wrote central image server code, implemented new network protocols, wrote C++ server interface class for Windows, OS/2, Solaris, and Linux, and wrote central image server system code that compiled and operated under Solaris Sparc, Solaris Intel, Linux Intel, and Linux Alpha – led group of 6 subordinates and assigned tasks, both in development and support)

- Ported Java “midlet” cell phone game to applet and added network code to support multiplayer mode using MIDP, JNLP, BREW (C++)
- Designed and wrote multiuser networked vehicle dispatch management system using ODBC, C++, and MySQL for both MS Windows and Linux
- Debugged problems and implemented fixes in the Compaq/HP Tru64 UNIX kernel CAM (SCSI) device driver subsystem
- Wrote vehicle rental management package in C++ under Linux and MS Windows
- Wrote sales management software package in C++ in MS Windows
- Wrote multithreaded sendmail “milter” in C and C++ to capture and process email
- Wrote new OS/2 RAID driver for Adaptec PCI-based controller card

Home Lab

Home lab systems are networked via gigabit ethernet and/or WIFI 802.11a/b/g/n/ac. Connection to the Internet is via fiber, with router/firewall/SELinux protection.

The following operating systems are active/available in home lab:

- CentOS, Ubuntu (kernels from 2.6.x thru 5.x)
- Microsoft Windows 10, 8.
- Apple OS X 10.x
- Android

Hardware platforms include:

- Centos-7 Internet-facing system (web/email)
- 16TB RAID-5 and 6TB RAID (striped) NAS
- Multiple Intel desktops and laptops (Windows, Chrome OS, and Linux)
- Apple MacBook Air 11”
- Multiple tablets and phablets (Windows and Linux)
- Other miscellaneous networking equipment (hubs, routers, etc.)
- Embedded development boards (i.MX6, Raspberry PI, ARM, Xilinx, etc.)

Education and Activities

- Reviewer for Pakt Publishing “Yocto for Raspberry PI” (ISBN 978-1-78528-195-2)
- Reviewer for Pakt Publishing “Embedded Linux Projects Using Yocto Project Cookbook” (ISBN 978-1-78439-518-6)
- GCS Adjunct Instructor, Daniel Webster College, Nashua, 1999 to 2011 (curriculum included UNIX/Linux device drivers; networking concepts and protocol basics; system management; TCP/IP and sockets; C/C++ programming; OOP methodology and techniques; Artificial Intelligence)
- Appears in product manuals for Yggdrasil Linux, 1994 and 1995.
- Published in MIPS, Personal Workstation, and Leisure Times Magazines.
- BSCS, Franklin Pierce College, 1988, with honors.
- Honorable Discharge USN, 1982, while stationed aboard USNS Rigel as LPO of 23-man detachment.
- BOD member, Gift of Life New England Inc., a 501(c)3 organization that arranges for no-cost open-heart surgery for children from 3rd world countries (www.golne.org).
- VFW Post 483 (Nashua) Trustee (officer)