RTEMS 4.10 and Beyond

Joel Sherrill, Ph.D.
Director of R&D
OAR Corporation

December 2010

Outline

- Highlights of RTEMS history
- RTEMS features
- What’s new in RTEMS 4.10 releases
- Wish List for 4.11 releases and beyond

History – Part I

- 1988 – OAR initiates development under contact to U.S. Army Missile Command (now U.S. AMCOM)
- 1992 – Superconducting Super Collider (SSC) is first non-Army organization to receive RTEMS. Evaluated easily and favorably against pSOS+.
- 1992 – Project begins using GNU tools
- 1994 – ESA sponsors OAR development of SPARC port
- 1994 – Publicly available via anonymous ftp from U.S. Army
- 1995 – Oldest date in RTEMS CVS 11 May 1995
- 1996 – rtems.com domain registered
- 1997 – GNAT/RTEMS passes Ada95 ACVC
- 1999 – First GNU/Linux RPMs

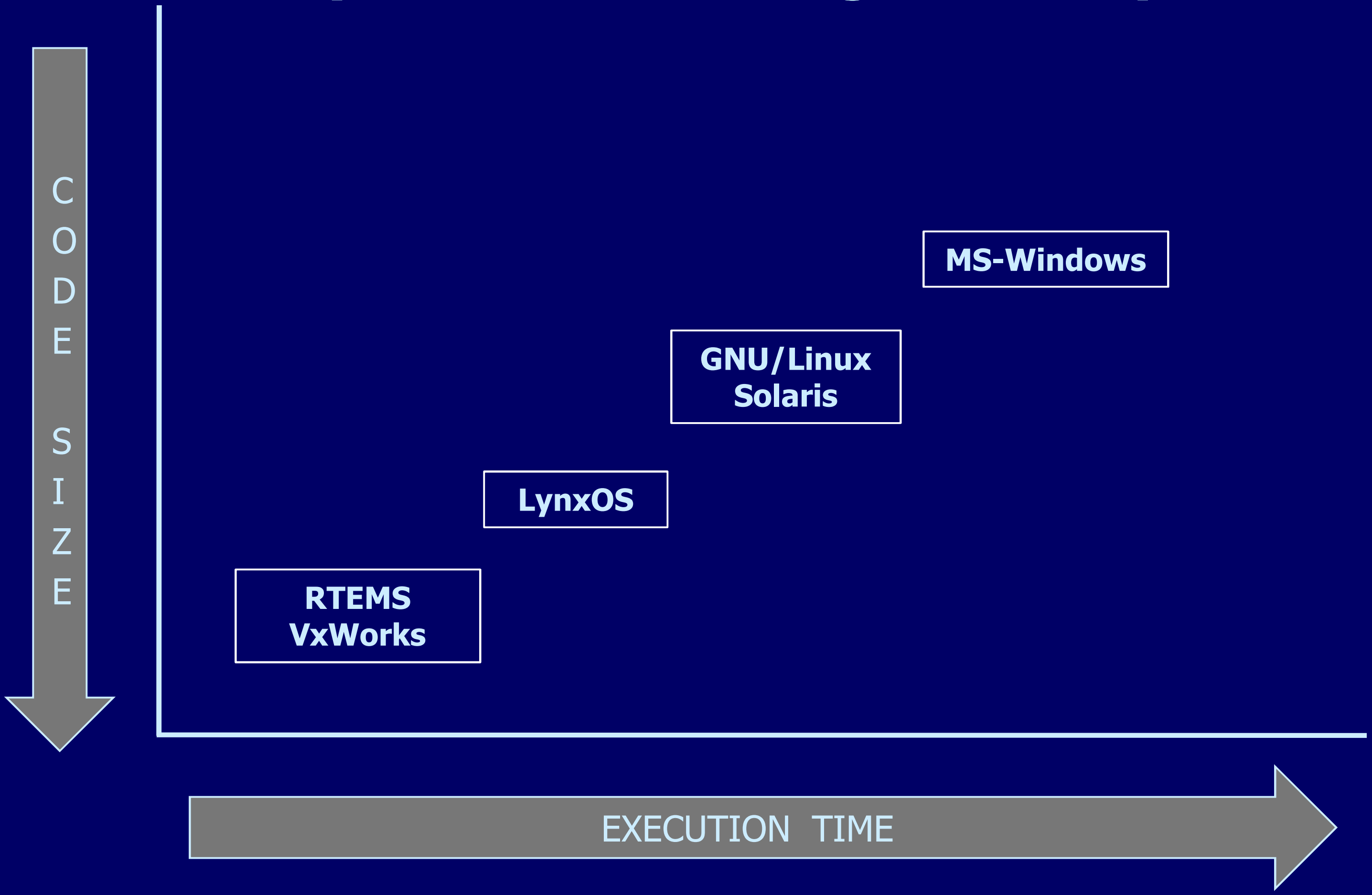
History – Part II

- 2001 – Steering Committee formed
- 2001 – Public problem tracking database available
- 2001 – Ported to NASA space hardened MIPS
- 2004 – Wiki started
- 2006 – Circles Venus and Mars
- 2007 – Launched to the asteroid belt with Dawn
- 2008 – Launched with Herschel/Plank missions
- 2008 – First year as Google Summer of Code project
- 2010 – Launched with Solar Dynamic Observatory
- 2010 – RTEMS Filesystem added
- 2010 – One of 20 organizations in Google Code In

RTEMS vs Other OSes

- Designed and supported to be open source
  - we can answer “Why was it done this way?”
  - openness includes test suite and documentation
- Single process, multi-threaded
  - comparable to VxWorks, Nucleus, pSOS+, ThreadX
- POSIX Profiles PSE51 and PSE52
  - no processes or MMU means no exec(), shm*(), etc.
  - lighter than LynxOS or real-time GNU/Linux

Graph of RTEMS vs other Oses (Order of Magnitude)



CODE SIZE

MS-Windows

GNU/Linux
Solaris

LynxOS

RTEMS
VxWorks

EXECUTION TIME

RTEMS Features

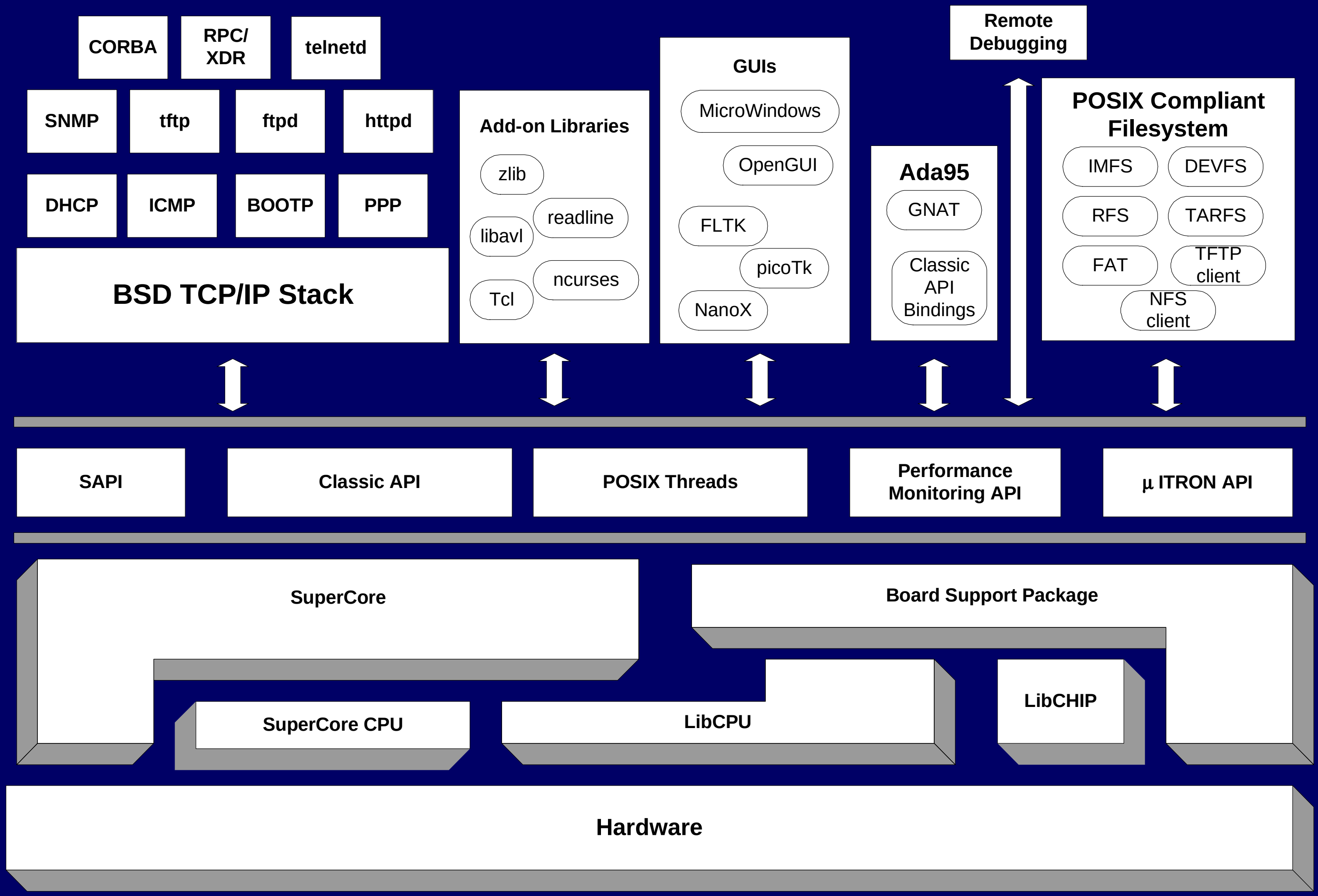
- RTEMS is a Commercial Grade Real-Time Operating System
- Truly free in price, liberty, and end user requirements
  - All source code for OS, support components, tests, documentation, development environment, and project website is provided
- High performance with deterministic behavior
- Low overhead with predictable resource consumption
  - Full executables currently as small as 16K
- Highly configurable with unused features left out by linker

RTEMS Features

- Multitasking
- Highly Portable
- Standards Based
- Multiprocessing
- User Configurable
- Object-oriented
- Interrupt Processing/Management
- Development Environment
- Remote Debug Server
- Shell
- Eclipse plugin available

- Communication/Synchronization
- Memory Management
- I/O Drivers
- Various File Systems
  - FAT, RFS, RAM, NFS, YAFFS, etc.
- User Extensible
- Networking with Servers
- File System Infrastructure
- C, C++, Ada, Go, and Java
- Libchip – Peripheral Controller Library

RTEMS Architecture



Processors Supported by RTEMS

Architecture	4.6	4.7	4.8	4.9	4.10	CVS
Altera NIOS II	No	No	Yes	Yes	Yes	Yes
ADI Blackfin	No	No	Yes	Yes	Yes	Yes
ARM with many CPU models	Yes	Yes	Yes	Yes	Yes	Yes
ARM/Thumb	No	No	No	Yes	Yes	Yes
Atmel AVR	No	Partial	Partial	Partial	Partial	Partial
AMD A29K	Yes	No	No	No	No	No
HP PA-RISC	Yes	No	No	No	No	No
Intel/AMD x86 (i386 and above)	Yes	Yes	Yes	Yes	Yes	Yes
Intel i960	Yes	No	No	No	No	No
Lattice Semiconductor Micro32	No	No	No	No	No	Yes
MIPS including multiple ISA levels	Yes	Yes	Yes	Yes	Yes	Yes

Processors Supported by RTEMS

Architecture	**4.6**	**4.7**	**4.8**	**4.9**	**4.10**	**CVS**
Freescale MC680xx and MC683xx	Yes	Yes	Yes	Yes	Yes	Yes
Freescale Coldfire	Yes	Yes	Yes	Yes	Yes	Yes
OpenCores OR32	Yes	No	No	No	No	No
PowerPC including 4xx, 5xx, 6xx, 7xx, 8xx, 52xx, and 74xx	Yes	Yes	Yes	Yes	Yes	Yes
Renesas H8/300	Yes	Yes	Yes	Yes	Yes	Yes
Renesas M32C	No	No	No	No	Partial	Partial
Renesas M32R	No	No	No	No	Partial	Partial
Renesas SuperH (SH1 through SH4)	Yes	Yes	Yes	Yes	Yes	Yes
SPARC including ERC32 and LEON	Yes	Yes	Yes	Yes	Yes	Yes
SPARC64	No	No	No	No	No	Yes
TI C3x/C4x	Yes	No	No	No	No	No

What’s New in 4.10? Updated Tools

- Latest version of all tools in the RTEMS Development Environment
  - autoconf 2.68
  - automake 1.11.1
  - binutils 2.20.1
  - gcc 4.4.5 for C/C++ with newlib 1.18.0
  - gdb 7.1
- Tool repositories are mirrored at three sites

What's New in 4.10?
File System Related

- POSIX pipe support
- RTEMS File System (RFS)
- libblock now supports variable sized buffers
- Miscellaneous improvements
  - more compliant chmod() behavior
  - statvfs() support added
- Mount operation now appears more like UNIX
- New "device only file system" which is very light

What's New in 4.10?
Code and Data Space Reduction

- Configurable maximum number of priorities
- Configurable minimum task stack size
- File system support can be disabled completely
- Idle task can be used to perform system initialization

What’s New in 4.10? Coverage Analysis

- Significant improvements over the past few releases
- Profile tracked is “RTEMS proper” at –Os
  - RTEMS proper: score, rtems, posix, and sapi
- Coverage is tracked for 10 BSPs across 6 architectures

Release	% Covered	Uncovered Ranges	Uncovered Bytes	Total Bytes
4.7	77.51%	454	17508	77840
4.8	76.37%	538	21772	92140
4.9	96.41%	167	2532	70564
4.10	100%	0	0	70480

http://www.rtems.org/ftp/pub/rtems/people/joel/coverage/

What's New in 4.10? New Ports and BSPs

- New target architectures
  - Lattice Micro 32 (LM32)
- New Board Support Packages
  - ARM lpc24xx with variants
  - ARM lpc32xx with variants
  - ARM Gumstix
  - ARM GDB simulator with variants
  - ARM CSB637 and Kit637
  - Coldfire Cobra5475
  - LM32 LM32_EVR
  - PowerPC Beatnik
  - PowerPC gwlcfm (MPC55xx variant)
  - PowerPC QEMU Simulator
- Many BSPs updated and enhanced

What’s New in 4.10?
Important Odds and Ends

- Configure separate or unified RTEMS Workspace and C Program Heap (e.g. malloc)
  - Separate: historical behavior – used in hard resource apps
  - Unified: similar to VxWorks – used by more dynamic apps
- More use of C99 types
- Optional use of 64-bit nanoseconds since epoch as internal time representation
- Warnings across all BSPs reduced
- BSP command line argument support added

4.10 and Beyond – The RTEMS Roadmap

RTEMS Requirements Sources

- Standards
  - The OpenGroup Single UNIX Specification
  - ISO/ANSI language and library standards
  - GNU Standards
- Space community
  - NASA, ESA, and their contractors
- Scientific Community
  - EPICS is most visible organization
- Commercial users
- Academic Community
- Good ideas
  - Research
  - User suggestions

Already in CVS for 4.11

- Changes for issues identified by static analyzers
  - Coverity Scan
  - clang scan-build

- Coverage improvements
  - Thanks to GSOC student, development profile coverage improved from ~83% in March 2010 to ~98%

- Initial support for POSIX Threading Timing Tests

Already in CVS for 4.11

- SuperCore Scheduler Refactoring
  - User can configure alternative scheduler
  - Prerequisite for SMP aware scheduler
- Per CPU Data Structure
  - Heir, executing, etc. tracked on per-CPU core basis
  - Prerequisite for SMP
- RTEMS Scheduler Simulator (close)
  - Allows one to evaluate new scheduling algorithms on host system

Community Interaction Wish List

- User application requirements
  - Please sponsor improvements which make RTEMS the perfect RTOS for your applications
- User testing and V&V
  - RTEMS artifacts and procedures should eventually dovetail into those of users
  - Make sure WE do want you need in this area
- Tell us about your application
  - Project needs more flyers on your success stories

Community Interaction Wish List – continued

- User configuration management requirements
  - Would periodic DVDs of tools and matching RTEMS tools be useful?
  - What else would help make RTEMS easier for your CM?
- Legacy support requirements
  - Free project maintains 2 branches
  - OAR provides support for versions as long as required
  - What does the space community need in this area?

Feature Wish List

- USB support
  - Port BSD USB stack to RTEMS
  - Refactor TCP/IP port to have shared “BSD porting layer”
  - Working port in community; requires effort to merge
- TCP/IP stack update
  - Update TCP/IP stack code to a late model FreeBSD version
  - Key Goald: IPV6 and ensure future updates are easy
  - No active work in this area

TASKS ARE INTER-RELATED AND WILL REQUIRE SPONSHORSHIP TO COMPLETE!!

Feature Wish List – continued

- Symmetric Multiprocessing Support (SMP)
  - some infrastructure and refactoring work done
  - scheduling, locks, and testing are primary activities left
  - needs engineering hands-on to finish
- Enhanced Eclipse support
  - RTEMS Eclipse plug-in available
  - Need Eclipse expertise to highlight more capabilities

Feature Wish List – continued

- Event Capturing and Visualization
  - Log events from RTEMS target and analyze on host
  - Chris Johns' capture engine is a baseline
  - GSOC students have worked on target side of this
- Free CANbus Stack
  - There are open source implementations
  - No active work in this area

Why Does RTEMS Exist?

For YOU!

Because of YOU!

YOU are the Community!

YOU are the Source For Improvements!

Contact Information

Joel Sherrill

Director of Research and Development

OAR Corporation

7047 Old Madison Pike Suite 320
Huntsville AL 35806

Voice: (256) 722-9985
Fax: (256) 722-0985
Email: joel.sherrill@oarcorp.com

RTEMS Project URL

http://www.rtems.org

RTEMS Wiki

http://www.rtems.org/wiki