RTEMS Development Roadmap

Joel Sherrill, Ph.D.
OAR Corporation

October 2011
Outline

• Community Driven Focus
• Project Process Improvement Efforts
• 4.9 and 4.10 Updates
• Development Head (4.11) Improvements
• Wish List for Future Improvements
• OAR Legacy Support Releases
• OAR Support Subscriptions
• Conclusion

http://www.rtems.org
Community Driven Focus

• Without users, project has no reason to exist!
• Users drive requirements
  – Please let us know what you need
• Users provide or fund many improvements
  – Again those reflect your requirements
• Most bug reports are from users

RTEMS Evolves to Meet Your Needs
Participate in Student Programs

• Attracts next generation of developers
  – As users and developers of RTEMS

• Promotes RTEMS to students
  – For use in projects and research

• Forces RTEMS Project to evaluate
  – Ease of getting started for new users
  – Approachability of email list and IRC channel
RTEMS Project Participation

• Google Summer of Code (2008-2011)
  – Almost thirty students over the four years

• Google Code-In (2010)
  – High school students did ~100 tasks for RTEMS
  – Included only twenty FOSS projects

• ESA Summer of Code In Space (2011)
  – First year and only twenty FOSS projects involved
Project Process Improvement Efforts

• New website in Drupal
  – Old site used MetaHTML which is now unmaintained
  – Now looks fresher and is easier to update

• Build bot for regression testing
  – Each check in will be tested for regressions
  – Automated test of all configurations impacted tested

• Transition from CVS to git
  – Driven by build bot which needed atomic change sets
Establish Tool Evaluation Process

• RTEMS Project is over twenty years old
  – Very long time in computing terms
• New approaches and tools introduced
  – Doxygen did not exist when project began
• New tools available to address old needs
  – For example, RCS -> CVS -> SVN -> git, etc.
• Tools may fall out of favor or unmaintained

• Unplanned transitions are PAINFUL
  – GNATS -> Bugzilla was too late
Evaluation Process Goals

• Recognize when a
  – tool is becoming or has become unmaintained
  – superior tool becomes available

• Transitions need to be planned not forced

• Goal is to produce a high quality real-time operating system – not support tools
  – We need to stay in the “sweet spot”

http://www.rtems.org
4.9 and 4.10 Updates

• 4.9 Release Branch Updates
  – 4.9.5 and 4.9.6 released addressing ~30 PRs

• 4.10 Release Branch Updates
  – 4.10.0 and 4.10.1 released
  – 4.10.1 addressed ~30 PRs
    • Added TLL6527M BSP
    • Included updates on other BSPs

• RAD750 BSP available for 4.10 but under ITARS
  – If interested, contact OAR for details
Development Head (4.11) Changes

• Pluggable Scheduler Infrastructure
  – Multiple single-core Schedulers now in tree
    • Deterministic Priority Scheduler
    • Simple Priority Scheduler
    • Earliest Deadline First (EDF)
    • Constant Block Scheduler (CBS)
  – RTEMS Scheduler Simulator to aid in writing one

• Initial SMP Support
  – Tested up to 4 cores on LEON3 and pc386
  – Plenty of room for improvement
More 4.11 Changes

• Sixty-Four Bit Internal Time
  – Represents nanoseconds since POSIX epoch
  – 584.55 years before overflow

• ARM Cortex-M3 Support

• SPARC64 Port Added

• Improved NIOS II Support
  – External interrupt controller support

http://www.rtems.org
New BSPs in 4.11

• ARM
  – Thorn TLI800
  – Stellaris LM3S6965
• LM32
  – Milkymist
• SPARC64
  – Niagara
  – UltraSPARC III

• PowerPC
  – DP2
  – Phytec MPC5554
  – MPC5674 EVB
  – MPC8309 SOM
  – QorIQ
  – Qemu PREP
  – Qemu Altivec

http://www.rtems.org
Wish List for Future Improvements

- Coverage
- Capture Engine
- Run-Time Loader
- TCP/IP Stack Upgrade
- USB Stack
- Multi I/O API
Coverage Status

• Instruction and branch level coverage reports for
  – Ten BSPs covering six architectures

• Initial focus was RTEMS “Proper” at -Os
  – score, rtems, posix, and sapi
  – Instruction coverage only – hovers near 100% now

• Current effort is broader
  – Branch taken/not taken coverage of RTEMS “Proper”
  – CPU Kit except networking and shell

• Reports now included with releases

http://www.rtems.org
Coverage Wish List

• Output from `covoar` in `gcov` format
  – Enables use of tools that support `gcov` format

• Output from `covoar` in `gperf` format

• Assistance from domain experts to
  – Get DC and MC/DC coverage reports from `covoar`
  – Produce reports in formats useful to RTEMS users

• Help from any user to improve coverage
  – Submit test cases... please

Coverage is a long journey with many small steps
Capture Engine

- Generation – Need tool to assist in trace configuration
- Capture – Need feedback from real use cases
- Analysis – Need GUI tool for visualization of traces

All will require user support to accomplish!

http://www.rtems.org
Run-Time Loader

• Standards based using the dlopen family of calls.
• Implemented as a run-time link editor (RTL). Locates and fixes up ELF object files in the target.
• Rejected the shared library model found on Unix due to performance and complexity. RTEMS is a ‘single process’ and the sharing code support offers nothing.
• Developed for RTEMS with a small target footprint. Designed to avoid the heap fragmenting when loading a number of object files. Minimal resident overhead.

http://www.rtems.org
Run-Time Loader

• Object files can be loaded from archives or as separate files from the file system
• The link editor loads and locates ELF object files resolving symbols and loading the symbol table with public symbols. There is a small amount of architecture specific code.
• Exported base image symbols can be linked to the base image using a 2 pass link process or loaded via an object file created post the base image linking.
Run-Time Loader To Do

• Dependency processing is incomplete
• Need to support C++
• Only supports a few architectures
• Documentation is lacking
• Host side tool to link could have more features
  – Identify unused methods in executable
• Feedback and support from users needs

http://www.rtems.org
TCP/IP Stack Upgrade

• Based on a current FreeBSD kernel.
  – The latest kernels support threading to aid SMP.
• Provide a adaption layer for FreeBSD kernel services such as threads, mutex and condition variables.
• Large project with numerous complex issues and needing user funding to pull all the pieces together.
USB Stack

• Based on a current FreeBSD kernel.
• Attempt to NOT modify the FreeBSD sources to improve the ability to import updates.
• Support via Bus Space API for FreeBSD drivers such as EHCI.
• Port of class layers such as mass storage.
• Recent port available in Bugzilla
  – However Score support for a Bus Space API is needed before being accepted into the source tree.
RTEMS Multi I/O API

- Want standard RTEMS API for boards with
  - Discrete I/O, DACs, and ADCs

- Current status
  - multiio CVS module available with shell commands and support for two boards

- Next steps
  - Feedback from users on API and features
    - May require more development
  - Once API is settled, then must be documented

http://www.rtems.org
OAR Legacy Support Releases

• Community supports
  – Development head (e.g. 4.11), and
  – Two release branches (currently 4.9 and 4.10)

• OAR offers legacy support for older branches
  – Currently available for 4.6, 4.7, and 4.8
    • 4.6.9 includes 54 fixes not in 4.6.6
    • 4.7.5 includes 34 fixes not in 4.7.3
    • 4.8.4 includes 22 fixes not in 4.8.2
  – Branch supported as long as needed by users
OAR Support Subscriptions

• Bring RTEMS into your project faster and simpler
• Direct Contact with the RTEMS Engineers
• Timely Responses from the RTEMS Engineers
  – normally less than 24 hours
  – always within 2 business days
• Rapid Problem Response
  – solutions as quickly as technically possible
• Assistance with the RTEMS Environment

Now includes subscription to easy to install DVD with the RTEMS Development Environment
How to Get Your Feature Merged

• RTEMS is free software
  – You are free to modify or add to it
• No guarantee anything will get merged
  – Must meet coding and design standards
• Improve your situation!!!
  – Get a core developer involved
  – Support them implementing your feature
• Recent user funded projects
  – RFS, Run-Time Loader, and SMP
Conclusion

• RTEMS Project is driven by user needs

• Continual effort to improve our processes

• Improvements will continue with user support

• Current and legacy release support available

Please Support RTEMS Enhancement Activities

http://www.rtems.org