

Google Summer of Code 2009

Proposal for RTEMS – By Aanjhan Ranganathan

MMU Support for RTEMS

Abstract

Most modern processors have Memory Management Unit Hardware built into the processor whose main functions are virtual address translation, memory protection and cache control. With RTEMS primarily focused on Embedded Real Time applications, making use of these MMU features especially Memory Protection is important to meet the needs of those applications that requires such support. RTEMS currently does not have MMU support and this project proposes to add this as part of the Google Summer of Code program.

The main objective of adding MMU support would be to limit operations on valid memory and uniformly catch invalid / faulty memory accesses. MMU support features like prohibiting unauthorised memory accesses, cacheable/uncacheable read/write support for I/O peripherals to name a few are proposed as part of the Summer Project.

Since RTEMS is a single process, multi-threaded system, the MMU features related to interprocess memory sharing is not relevant. Rather the MMU shall be used just to have memory protection.

Project Deliverables

- June 15th - An API Design Document for review by community, POSIX Stubs.
- July 10th - API compliant tests for the stubs, Reviewed final version of API design document. (Mid term evaluation)
- August 10th - Completed implementation for the reference architecture and successful test of the same. (For Final Evaluation)
- Chapter additon to POSIX Guide on shm* and mm* functions
- Additions to Porting Guide - indicates the fact that the MMU implementation is Architecture dependent and not BSP dependent.

Proposed Schedule

April 20 - June 15th 2009 (Requirements Analysis and Design Phase)

Getting to know the RTEMS code base by writing minor test programs / patches, reading literature on MMU architecture and coming up with an API Design document for the next phase of implementation. Verifying PSIM's MMU support will also be performed at this stage. It might be required that there are multiple levels in the API - like user-level, cpu level and BSP level. the design / flow of these API levels is also mentioned in the draft of the API design document.

June 15th - July 10th (Implementation Phase 1)

The defined APIs would be prototyped and tests for checking the API compliance committed and

run. The tests would be performed majorly for error checking without the underlying architecture specific implementation. The *reviewed* API design document with the changes is frozen.

July 11th - August 10th (Phase 2)

The APIs will be implemented for the reference architecture's MMU and psim simulator would be used to validate/debug the implementation. Successful run of API tests with the implementation of the reference architecture. QEMU is suggested as a backup emulator if PSIM does not end up providing good support for MMU. Debugging and improving/optimizing code-base and fixing any blocker issues also forms a major part of this period.

August 10th - August 21st 2009

Documentation, code clean ups for final deliverable submission.

Future Improvements

Here, some future improvements on this project are proposed. This would be done after the GSoC program time line or as Bonus implementation during the program itself if things go better than expected.

- Implementing Cache Memory Support for unsupported architectures.
- MMU support implementation and verification for a second architecture.

Major Challenges foreseen

- A lot of this support inclusion is dependent on the MMU hardware implementation and getting this sorted out in a nice way to reduce the MMU hardware architecture dependency is one major challenge.
- Analysing a lot of CPU hardware architectures and coming up with a generic API definition would also be a brain cracker.

References

- [FreeScale PowerPC MMU Architecture](#)
- [PowerPC Book E MMU architecture](#)
- [Enhancing Simulator Support for MMU \(PowerPC architecture\)](#)
- [Single UNIX Specification for mman.h](#)
- [Single UNIX Specification for shm.h](#)

Relevant Background Experience

- Being a GNUSim8085 upstream developer and maintainer, along with contributing in small quantities to several upstream/downstream projects, I am quite comfortable in using version control systems like CVS (which is used in RTEMS project), SVN, .. etc.
- Also comfortable with gdb, gcc usage.
- Having worked as an Embedded Engineer for over 3 years , and currently doing research in computer architecture, I am used to using CPU simulators like Simple Scalar, compiler teaching tools like TriMaran and also some emulators like QEMU (which I use sometime to test distros of my interest, and run some application for other platforms).
- RTEMS code base is mostly C based and I believe that I am pretty fluent in the language.

Personal

I am currently doing my Masters in Electronics at **École Polytechnique Fédérale de Lausanne (EPFL)** situated in Lausanne, Switzerland. I hold a Bachelors degree in Electronics and Communication from Anna University, Chennai, India and have worked for 3 years as Senior Embedded Engineer for the German MNC, Robert Bosch as part of their Car Multimedia Group popularly known as "Blaupunkt". My research interests have always (right from my Bachelor University days) revolved around Computer Architecture, Microprocessors, Controllers.

I have known RTEMS for a few weeks now, when a friend at school suggested it as a possible OS port project for the OpenRISC based multiprocessor system being developed at my Lab in EPFL and I feel the Google Summer of Code as a nice opportunity for me to learn the nitty gritty parts of the Operating System and get myself involved in RTEMS development in the future. I assure a dedication of atleast 30 hrs a week during the Summer Project. I hope to have a good learning experience from this programme and looking forward for a long relationship with the RTEMS community even after GSoC.

Experience

Free Software Experience/Contributions:

- [GNUSim8085](#) , an 8085 simulator's upstream developer and maintainer.
- Fedora Electronics Laboratory Contributor.
- [Conducted workshops](#) on using Free and Open Source Electronics CAD tools at schools and colleges.
- Official Ubuntu Member and Active member of the [Ubuntu India Admin team](#) which manages resources on the Local Community server and IRC Channel.
- Member of the Ubuntu Science Team which helps in adding new packages into Ubuntu and keeping the packages in sync with the upstream. I mostly look into Electronics related packages.

Language Skill Set : C, Verilog, VHDL, Perl and Python (beginner level)

Related Research and Work Experience :

- Filesystems : [Mounting of Version Control Repositories: RepoFS](#) , Poster presented at IEEE High Performance Computing Conference.
- Operating System Evaluation: Linux and BSD feasibility study for Automotive Embedded Software, presented at SAE Conference in Pune. (SAE Paper No. 2007-26-057)
- Current Research : Snoopy Based Cache Coherency Protocols for Multiprocessor based Embedded Systems.

Reference Links and Web URLs:

Personal homepage: <http://www.tuxmaniac.com>

Personal Blog : <http://blog.tuxmaniac.com>

Ubuntu Wikipage : <http://wiki.ubuntu.com/Aanjhan>