

POSIX Compliance

Google Summer of Code Program 2017 Project Proposal

Aditya Upadhyay
aadit0402@gmail.com
IIT(ISM) Dhanbad
+91-9128811628

Project Abstract

POSIX Compliance allows developers the maximum flexibility in creating application software that can be ported on to the other execution environments. Before POSIX, application written on one version can not be used on other version. Solution to this problem, POSIX was introduced. It is an interface standard defined by IEEE. It is an Open API, for this we need to implement header files and methods related to the header files. It includes a porting of function from FreeBSD as well as NetBSD for various file system including NFS and FAT file System. It pertains the OS, not the file system specifically, but a part of this API is used to access file system.

Project Description

POSIX Compliance allows us to port the source code that is running on one machine, can be run on other machine. This Project deals with the implementation of a subset of functions that is defined in POSIX 1003_1 standard, Due to physical resource constraints some real time systems like small embedded system needs limited set of operating system functionality. For these type of system it is necessary that the standards allow implementation to support only a particular subset of POSIX functions. Real Time Working Group addressed these subset of function.

POSIX profiles defined in FACE technical Standards, designed for avionics system versus POSIX in RTEMS where some functions are missing. These missing function need to be implemented. For the same I will use FreeBSD, NetBSD, POSIX 1003_1 as reference and IEEE std 1003.1™-2008. I will integrate these implemented function to Newlib.

Header files and related methods that are not yet implemented :

Fenv.h - In this file signature of all floating point environment related APIs will declared. Math.h - All 35 methods, sys/select.h - I will define pselect method and will make a signature of the same in sys/select.h header file.

Unistd.h, trace.h, complex.h, some missing function and flags from signal.h : sigaction()

Inttypes.h - Some missing functions, I will need to implement.

These are the functions that are missing in newlib as compare to FACE general purpose POSIX profile. Some constant are also there to implement with these POSIX methods.

Project Deliverables

- GSoC Timeline : <https://developers.google.com/open-source/gsoc/timeline>
- May 30 : (Coding Begins) - I will have RTEMS setup environment for sparc/erc32, arm/xilinx_zynq_a9_qemu and also have completed Hello World on these two architectures. In the initial phase, I will go through the IEEE std 1003.1™-2008 and develop understanding of changes to Base definition, changes to System Interface, Changes to Shell and Utilities, and Changes to Rationale and sort out the previously implemented, unimplemented (yet to

implement) and unimplementable APIs. I will have FreeBSD and NetBSD repository and try to gain much more implementation details about the functions, macros, and constants. I will get well versed with RTEMS coding conventions. Consequently, will fork the necessary repositories on github where my mentor can track my progress.

- June 26-30 (Phase 1 Evaluation) - Before evaluation of this phase, as priority suggested by Joel Sherrill, I am planning to successfully implement library support for long double complex methods declared inside `complex.h`, library support for Floating point environment along with specific types. I will implement 37 interfaces from API Extra section from RTEMS POSIX API Status doc, prepared by Joel, where some of the functions are already implemented in RTEMS and ENOSYS, so i will have to port only those functions to newlib. I will stay in contact with my mentors and other experts for review of my work and for mentor approval, I will be creating patch for each and every function for submission.
- July 24 - 28 (Phase 2 Evaluation) - In this phase, Work will be on library support for mathematical operations and header files. I will implement some missing methods from `inttypes.h` : `strtoumax`, `strtoimax`, `wcstoumax`, `wcstoimax` and related constants and macros. Then check for support in RTEMS, newlib libc, newlib libm. I will keep updating my patches and submit them for merging with newlib.
- August 29 - September 5(Phase 3 Evaluation) - In this phase, I will work on mandatory missing APIs as Joel have listed out in RTEMS POSIX API status, related functions, constants, macros. (`sys/mman.h`), `sys/select.h` for `timeval` structure and `sys/socket.h`. Gedare is already working on memory management related methods, Instead of `mman.h`, I will implement `sigaction` function and its related flags : `SA_RESETHAND`, `SA_RESTART`, `SA_SIGINFO`, `SA_NODEFER`. I will implement `confstr()` declared inside `unistd.h` header file for getting and setting configuration string value. I will work on `ftw.h`. I will implement `nftw()` function signature declared inside this header file as it is mandatory API, missing in newlib. `ftw()` is also missing API from newlib. It is not a mandatory API. I will check and resolve the header file dependencies. I will test examples related to these header files, if it will work then i will request my mentors and other experts to track my progress.
- September 6 (Final Result Announce) - Before final result announcement, I will refine my code. I will consult with my mentors and request for review. I will create patch for each implemented function and merge with the mainline RTEMS repositories and newlib. By this point of time, I will maintain my works in a proper document as well as I will update my blog and share my experiences with others.
- Post GSoC - Honestly, i would be very happy to work in open source community as i always wanted to work and contribute in open source. I will continue with my work in RTEMS to gain more knowledge. I will help others user/developer in fixing bugs and try to resolve any issues related to RTEMS.

Project Schedule

March 20 - april 30 (Application Period)

I will go through the RTEMS wiki and set up my RTEMS environment for `sparc/erc32`, `arm/xilinx_zynq_a9_qemu` with `libbsd` and `waf` configured. I will need to set up the other toolset and BSPs at the time of `fenv.h` implementation for testing purpose. By this point of time I will have successfully completed hello world application. I will join devel and user mailing list. I will set up my github account and fork RTEMS repository. I will write proposal draft and request my mentor and other experts to review the same. I will do further refinement on proposal draft according to requirements and suggestions.

April 30 -May 4 (Acceptance Waiting Period)

By this point of time, I will go through the RTEMS Coding convention, Previous works related to POSIX. I will understand RTEMS tool chains for sparc/erc32 and arm/xilinx_zynq_a9_qemu, other BSPs and other related directories. I will have a rough idea of POSIX profiles defined in the FACE Technical standard, designed for avionics system and NetBSD, FreeBSD and Newlib and get information about header files, methods defined inside these header files, macros and constants. I will be in touch with my mentors, previous GSoC students who have worked for this project to get better understandings.

May 4 - May 29 (Community Bonding Period)

I will continuously communicate with my mentors and other experts. I will make myself capable to have good understanding of git commands, RTEMS tools. I will understand the process of committing the code, submitting patches for merging in future for code submission. By this point of time, i will have implemented some function, macros and constants and integrate these successfully implemented function to newlib.

May 30 - June 30 (Phase 1)

Methods, macros and constants related to complex.h and fenv.h will be implemented. I will need to set up other toolset and BSPs for testing the examples of fenv.h library. It is architecture specific implementation. For the same I will go through the FreeBSD and NetBSD and port the code to newlib. I will check dependencies, if there will be dependencies in between header files and function, I will try to minimize it. In complex.h, I will first declare the signatures for every methods related to long double complex method in header file. I will port the code for complex.h from FreeBSD. I will list out the mandatory missing methods related to floating point environment and will implement those methods with the help of FreeBSD and NetBSD. As Joel have prepared a spreadsheet and categorize the different APIs. In this phase I will port these real time extensions in newlib. I will test examples related to these libraries on arm architectures and other required architectures. I will send patch for every methods for reviewing purpose. If i run into unexpected issues, I will contact to my mentors and other experts.

June 30 - July 29 (Phase 2)

In this phase, I will implement methods, macros and constants related to mathematical operations and libraries related to inttypes.h. In inttypes.h, Some methods are missing like imaxdiv(), strtoumax(), wcstoumax(). I will implement some mandatory missing APIs defined inside the trace.h and the remaining methods would have completed in next phase. I will test these libraries with possible number of examples. I will send the patches for review, if there would be any issues in code, I will refine the code.

July 29 - August 29 (Phase 3)

APIs related to socket.h is also in RTEMS libbsd and need to port implemented APIs to newlib. I will implement remaining APIs from trace.h and will continue the implementation after second phase. I will check out dependencies between files and try to minimize it. Methods, macros, constants, flags will be defined for every header files and I will make signature for each and every function in header files. I will define timeval structure that will contain at least two attributes as time_t and suseconds_t. These two will be defined in <sys/types.h> and other structures will be retrieved from following header files : <signal.h> and <time.h>. So first i will investigate these header files, Methods, macros, constant defined inside the header files. I will implement declared function signature inside the header files. I will test examples

related to these libraries. I will get reviewed code. If refinement and modification is needed, I will do the same. By this point of time, I will prepare my patches and merge to git repository. I will maintain my document and update my blog related to work done by me.

Future Improvements

POSIX Compliance deals with several number of functions, macros, constants, flags and it's difficult to categories feasible methods. After completion of GSoC, I would love to contribute regularly where i have interest in fixing bugs related to POSIX APIs and also test the examples related to these libraries on different architectures.

Continued Involvement

I always wanted to contribute in open source communities related to embedded system. I will continuously contribute code for POSIX Compliance and try to understand other areas of RTEMS like BSPs, Tracing, and incoming architectures. I will regularly maintain my blog and documents so that future students may take reference. I would love to solve challenging problems.

Conflict of Interest or Commitment

There is no conflict of interest in coding phase. During the community bonding phase, I will have my final year Thesis submission, scheduled between may 20-30. If there will any sudden changes in university schedule, I will discuss with the mentors. I will able to put full time effort when the coding phase begins.

Major Challenges Foreseen

- Understandings of FreeBSD and NetBSD.
- Understandings of Dependency tree among header files.
- Categorization of feasible functions and priority
- Time Management

Reference

- <https://lists.freebsd.org/pipermail/freebsd-ports>
- man7.org/linux/man-pages
- <https://devel.rtems.org/wiki/GSoC/GettingStarted>
- <https://github.com/freebsd/freebsd>
- <https://github.com/IJ-NetBSD/netbsd-src>
- <https://docs.google.com/spreadsheets/d/16k7UAc5JPQ7yWNe9sNozGVXtDmflRxxv9V94KUXJrm0/edit?ts=58dfd5a2#gid=21>
- Single unix specification
- FACE Consortium
- <http://www.opengroup.org/testing/downloads.html>

Relevant Background Experience

- Sound in C coding
- Beginner in python coding

- Firm grasp of operating system and computer architecture theoretically

Personal

I am doing M.Tech in Computer Science & Engg. at Indian Institute of Technology (ISM) Dhanbad. One of my friend told me about GSoC, who got selected for last year GSoC, was working in RTEMS. I have gained knowledge about operating system and computer architecture theoretically and always wanted to understand these things practically. RTEMS is a platform where i can learn these core parts of real time os and different architectures and explore myself in embedded system.

Language Skill Set and Experience

- C,CPP - Comfortable.
- Python - Comfortable.
- Familiar with assembly language.
- Bash scripting - Beginner.