Contracts in OpenBSD

Supervisor Dr. Joseph Kiniry
Subject Area Software Engineering
Pre-requisite Good knowledge of C and operating systems
Co-requisite (things you must learn along the way) Basic knowledge of compilers, standardized APIs (e.g., POSIX APIs, the OpenBSD kernel API, the standard C library, etc.) and static analysis tools/frameworks for C.
Subject Coverage Software Engineering, Formal Specifications, Documentation, Operating Systems
Project Type Reviewing existing documentation (i.e., standards documents, man pages, and code comments) and writing formal specification of (portions of) critical, key C APIs
Hardware/Software: Any machine running OpenBSD.

Description

OpenBSD is widely regarded as the world's most secure operating system. It represents the combined effort of hundreds of dedicated individuals, most of whom work in their spare time for free, over nearly a decade.
OpenBSD developers try very hard to write code that is well-designed, clean, maintainable, but above all, *secure*. A number of techniques are used for such, but most of them are relatively *ad hoc* when compared to the kind of program development that KindSoftware research group regularly participates in.
One way to further improve the quality and security of OpenBSD code is to use various types of static analysis to reason about the code to (perhaps automatically) ensure that it conforms to some form of (perhaps lightweight, or entirely implicit) specification.
Unfortunately but predictably, the vast majority of OpenBSD is written in C. Historically, API functions in C are documented via standards (like POSIX), manual pages, and code comments. And while some of these "informal specifications" were written by (possibly large) groups of (possibly very smart) people, they are frequently imprecise, incorrect, difficult to maintain, etc.
The goal of this project is to learn more about writing useful documentation, primarily in the form of contracts, for C program code in OpenBSD. A small number of functions will be chosen for examination based upon their size, complexity, and utility. These functions will come from various C libraries and the OpenBSD kernel.
Each of these functions will have several specifications written: one based upon the standard, one based upon the manual page, one based upon the program code, and finally one based upon our derived "correct model" for the function. By comparing these specifications we will determine if/where each informal specification is incorrect and propose revisions to improve clarity and correctness. We will also attempt to get our formal contracts incorporated into the source code and manual pages for OpenBSD.

Mandatory

Gain familiarity with OpenBSD.
Gain familiarity with lightweight static checkers for C like the Splint tool.
Gain familiarity with using the Frama-C framework and its Jessie and Value plug-ins.

Gain familiarity with the popular languages used to annotated C code (e.g., ACSL, Microsoft's SAL, Split's annotation language, etc.) and compare/contrast these annotation languages.
Choose at least a half dozen functions with help from Joe from the latest production code within the tool limitations (e.g. Frama-C) for the OpenBSD community to evaluate.
Write contracts for these functions based upon the relevant standards, manual pages, and program code.
Reconsider the all documentation in the light of the new contracts.
Provide patches back to the OpenBSD community that contain the proposed revised documentation and the formal contracts.

Discretionary

Attempt to use one or more static analysis tools on the annotated functions to check their correctness and the code's correctness with respect to those specifications.
Coauthor, submit, and publish a paper on this work.
Use what we have learned about the work on the first few functions to choose another set of functions to analyse and describe the critical factors that inform these choices.
Chosen function set for analysis can possibly be from a run-able sub-component of OpenBSD.

Sources of information and preparatory reading

The OpenBSD web site
Download, install, and play around with OpenBSD.
Get familiar with chosen OpenBSD source code.
Learn about contracts
SAL
Splint
Frama-C
Jessie
ACSL
CIL.