sdmay18-09: Tool Support for Continuous Model-Based Verification of the Linux Kernel

Week 5 Report
October 14 - October 21

Team Members

Srinivas Dhanwada — *Team Lead*
Collin McIntyre — *Tool Integration Lead, Scribe*
Benjamin Weno — *Automation Lead*
Matthew Wall — *Web Lead*

Summary of Progress this Report

This reporting period was spent mostly fixing bugs. Our algorithm for mapping individual locking mechanisms from one version of the Linux kernel to another worked during the small-scale proof of concept phase, but some problems arose when trying to apply the algorithm to the entire kernel. The algorithm was originally tested on a case-insensitive file system which caused problems when trying to parse through files with the same name. We created some space to try again using a case-sensitive file system, but the algorithm is taking much longer to complete than expected, and we suspect that we may have encountered an infinite loop. The patch that we generated for version 4.13 of the kernel compiled, but didn't redirect locking mechanism function calls how they were supposed to. We're having trouble due to an "EXPORT_SYMBOL" macro call in the implementation of the locking mechanisms, and need to figure out how to modify the patch to eliminate this error. We've continued development of our website, integrating feedback into the design and fleshing out more details related to layout and functionality, searching, filtering results, etc. We've also developed a program that will check the RSS feed that monitors updates to the Linux kernel that we can adapt to begin the verification process automatically.

Pending Issues

We need to figure out whether or not our mapping algorithm is encountering an infinite loop or if our algorithm is just slow and needs to be optimized. We also need to figure out what the "EXPORT_SYMBOL" macro is and how it works so we can modify our patch to work properly.

Plans for Upcoming Reporting Period

We plan to finish fixing the bugs in our mapping algorithm and kernel patch. We also plan to each manually walk through our diff algorithm for five different instances of locking mechanisms. This will allow us to get a better feel for how to implement an automated version of our diff algorithm that incorporates our locking instance mapping algorithm.

Individual Contributions

Team Member	Contribution	Weekly Hours	Total Hours
Srinivas Dhanwada	Srinivas worked on fixing the bugs in the locking instance mapping algorithm. He created a case-sensitive file system but ran into merge conflicts that require manual merging of several key files.	8	33

Collin McIntyre	Collin worked on fixing the bugs in the kernel patch. He was able to fix the bugs related to Mutex locks, but still needs to fix the bugs related to Spin Locks.	8	27.5
Benjamin Weno	Ben created a program that will check the RSS feed that monitors new releases of the kernel that we can adapt to begin the automated verification process.	4	18
Matthew Wall	Matt continued creating rough designs for the website and has also begun considering website modularity and how it can be used to help display graphs and the results of testing different instances of locking mechanisms.	4	12.5