Andrew Gates

Professor Matthew Tolentino

TCES 420

October 21, 2016

Project #1 - Add Syscall to Linux on Raspberry Pi

Challenges: While making this project there were many challenges involved in getting a syscall working on the Raspberry Pi –

* There was an issue with the –j4 tag when trying to build the kernel, for some Pi’s people had no trouble doing this, but for others we found that –j4 caused it to run too fast and try to install files before their dependences were also installed, causing an error.
* Another issue that I had was trying to find the files that I needed to modify. While the files that needed to be changed were outlined in various sources on the internet, one such file was unable to be located. The unistd file was in the location that everyone else mentioned. But it did not look like any of the examples that various people on the internet had found. This caused issues in attempting to define my syscall offset.
* The last issue that I had was trying to figure out how to call my syscall. There were various functions that should work, but only one seemed to work correctly. So it took attempting multiple methods that did not work in order to finally find one that worked.

Another way that I found to communicate between kernel and user space is procfs. I read that procfs has the main purposes of providing information about the running system, information about various devices, and networking information. Procfs allows the kernel and user space to interact, but I found that it is deprecated and it is mainly used for exporting information now instead of communication.