I was unable to get this OSP project completely working as of yet. My wife and I just moved over the weekend, we have no internet access at home and I’ve been struggling to keep up with all of my classes over the last two weeks. I was able to successfully implement many of the functions, including: do\_acquire(), do\_release(), and much of deadlock\_recovery(), I merely got lost in the details attempting to finish deadlock and giveupresources(). I understand the concept is just like our work, allocation, etc. algorithms we learned in class.

In do\_acquire, we first obtain the current thread based on the PTBR and the current task running. If the quantity requested + the quantity allocated are greater than the total, we return null. Else, we calculate in a hash table if we can allocate the request by utilizing do\_grant() to grant this request based on several parameters. In do\_release(), we are setting the available amount to a new total that is total-allocated. We then iterate over the hash table granting access to any thread that now has the available resources to complete their task now.

Apologies that I just haven’t had the time to commit to finish this with group projects due in multiple other classes, including CSCE590 (Mobile App Development) and CSCE567 Visualization Tools. I hope to have much more free time available to fully implement the final OSP thread project, but I feel I have made at least some good progress on this one.