Chad Dugie

David Trinh

CSS 430

11-29-15

Program 4 Report

# SPECIFICATION AND DESIGN

**Cache.java**

The cache class is designed to be used as a cache as part of ThreadOS. It uses the enhanced second chance algorithm using the reference bit and dirty bit that is stored with the private class Entry. Entry also has a block ID stored inside. The page table is created as an array of Entry classes.

When reading in Cache, the read method searches the page table for the block id to be read. When found, reference bit is set to true. Otherwise the method will continue searching for a free page to store the block id. If there is no free page then the method will look for a page to replace.

The same process occurs for writing, except the dirty bit is set to true when the block id has been written over.

# pERFORMANCE

*For full command prompt and results, please look at the TestResults.txt.*

**Random Access**

With cache disabled: 7781msec

With cache enabled: 6438msec

**Localized Access**

With cache disabled: 15116msec

With cache enabled: 7132msec

**Mixed Access**

With cache disabled: 5287msec

With cache enabled: 3087msec

**Adversary Access**

With cache disabled: 8064msec

With cache enabled: 7701msec

# Conclusion

Based on the results from testing each method, we have found that using cache is always going to give better performance in any of the four cases. Also, localizing 90% of the disk operation and having the rest of the 10% be random access gives the best performance out of the all cases.