Patrick Hamod
Dr. Rose
CSCE 311
13 Nov. 2014

Project 4 Memory Management

In this project I implemented the classes FrameTableEntry, PageTableEntry, MMU, PageTable, PageFaultHandler. Using these classes I implemented memory management for the OSP.

In FrameTableEntry it inherited all its class and functionality from its parent class and I created a long lastUse to keep track of the last time the frame had a page swapped in.

In PageTableEntry I implemented the methods do_unlock() and do_lock(). In do_lock the method implements the lock count of the frame that the page has and returns failure if the thread was killed during the process or the page has no frame. In do_unlock() I decremented the frames lock count as long as it was greater than 0.

In PageTable I implemented do_deallocate(). in do_deallocate I went through the frame table and removed the all the pages associated with the owner task from their frames.

In MMU I implemented do refer() and init(). The init() method fills the the FrameTable with instances of FrameTableEntries. The do_refer() method sets the refernce bit and dirty bit of the frame contained in the page baes the memory address. if the page has no frame then pagefault is necessary for the page to be put in memory.

PageFaultHandler implements do_handlePageFaults(). this method puts a page into memory by finding a free frame.If there is no free frames then find a frame with the oldest page if the dirty bit is 0 if that fails then find oldest page.

The first in first memory management algorithm preforms better compared to the demo because it spends less ticks on swap ins so pages stay in memory longer however there were more swap outs. My implementation utilized more of the cpu than the demo.jar and created more tasks and threads.