# Design Document

### CMPS 111, Winter 2019

## Goal

We want to understand how page replacement is implemented in FreeBSD. Then, we want to implement a different algorithm (FIFO) in which the oldest page in the queue gets replaced first.

## Assumptions

We assume that all the data structures from FreeBSD’s page replacement algorithm are correctly implemented because we are building our FIFO algorithm on top of the existing data structures.

## Design

Since the FIFO algorithm doesn’t rely on active and inactive queues, we began by using a single queue, by making the references to active and inactive queues point at the same queue. References play an important role in the original FreeBSD algorithm but are not necessary for our implementation, so we looked through and removed code where pages were affected by referencing. Removing the references from the code and combining the queues gives us our desired FIFO algorithm. We then added the attribute time\_t birth to the vm\_page struct to keep track of when a page enters a queue.

## Benchmark

For our benchmark, we stressed the system by calling calloc() multiple times in order to induce page replacement. In order to test the benchmark, we need to keep track of the age of each page, which is done by calculating how long each page has been in the FIFO queue for with timestamps. We also need to set the interval for running the pageout scan algorithm to once every 5 seconds in order to observe the results.

## Bugs/Unfinished Work

We are trying to keep track of the created time of every page as an attribute. We added this to the vm\_page struct and it was completely created by us. However, for some reason, when we print this value, the time is printed as 0 sometimes even though we do not set the creation time to 0 anywhere in the program. Due to this issue, our code may not be working entirely as intended, and as a result we can’t verify this. However, in theory and from what we can examine it is working as intended.