

DBSys Lab2 - LRU-K

Ching-Yao Huang, Jixiong LIU

1. FIFO

I reused the code from LRU.java since most of the functions are similar. First, about the "pick victim" part, when the buffer isn't full yet, evaluate and pin the frame one by one. After the buffer is full, use a loop to run through the frames looking for the first frame which is unpinned, pin it and update the frames. Since we pin a frame from the beginning of the array, the earlier it comes, the index of it is smaller. Therefore, a simple for loop start from index zero can find the frame comes earliest. Second, the "update" part. After picking a victim, move it to the end of the array since it's the newest one.

2. LIFO

"LIFO" is similar to "FIFO", so I reuse the code above but modify it slightly. First, for the "pick victim", when the buffer isn't full, we can pin it one by one, too. But after pinning one, we need to update once to move the frame to the beginning of the array. Then when the buffer is full, we can use a for loop starting from index zero to find a unpin frame. Since we keep moving the pinned frame to the beginning, the frame we found is the latest coming one. Second, the "update" part, it's similar to FIFO but moves the frame to the beginning instead of the end.

3. LRU-K

In this module, the core mainly revolves around two dictionaries HIST and Last, where HIST records the entire history, and Last records the most recent access time. Update and victim functions are mainly developed with reference to the pseudo-code of the paper involved in the Lab paper, in which we firstly consider whether the target page is in the buffer, where the parameters PageId and InBuffer are assigned by modifying the pin() function in the bufmgr class. In the development process, I encountered a bug that the key was lost due to repeated unpin. Finally, the return value was set to default -1.