

Josh Colletti & Vikramjeet Singh

-SegFaultHandler function:-

The function segfaultHandler handles all of the segmentation fault and gets all parameter needed for the address from the parameters to handle it.

SegfaultHandler functions handles the core of the workflow and checks with call to another function called valid address to see if the address is between the start and end of the limit.

-validaddress function basically just sees if the given address is greater than starting address and less than the last possible address(by adding page size to the starting address).

Then it checks if we want to have a new page for which we call handleNewPage function if we need to create new one.

-handleNewPage:- it handles a new page in a virtual memory system. If the page is being written to, the log data is updated, the page's reference and write back values are set, and the page's memory protection is changed to allow for read and write access. If the page is being read, the log data is updated, the page's only read value is set, and the page's memory protection is changed to allow for read access only.

Otherwise, if there is already existing one we found, we call

-handleExistingPage

HandleExistingPage:-. If the page is marked as "onlyRead" but the write flag is true, it sets the page to be writable, updates the reference and write back bits, and changes the page protection to allow reading and writing. If the

write flag is true, it sets the reference and write back bits and changes the page protection to allow reading and writing. Otherwise, it sets the reference bit and changes the page protection to allow reading only. It also logs the cause of the page access.

For searching of a page, we use a helper function called searchpage which simply just iterates over the entire list and see if it can find something matching.

If there is some physical frame available then we need to have no paging replacement in works as we do not need to remove/swap anything.

But if there is no physical frames available we call either handleFIFO which is type =1 or HandleThirdreplacement which is type=2.

- handleFIFO: we in this simply just go through the entire linked list for pages and simply give the first physical frame with the negative frame which is just -1. then , after we assigned the physical frame, we have to reset everything. So, we call resetpage.

Resetpage:- it just simply set every values to starting values.

In other case, if page is new page we use addtolist function which simply adds it to tail/end of the list otherwise we put it back to the queue.

-addTolist: simply add to the end of the list

-removeFromList: it removes from the head of the list and once our algorithm have been done it is called as eviction/ removing to remove it from head of the list.

-handleThirdReplacement:- here first thing is that what is the page address of page where it is stopped. Then, it goes through the list of our pages and have some protection as per them to set reference bit.

So, major difference from second chance to third chance have is that second chance have reference bit to give chance but third have another bit called dirty bit or modified bit.

So, we keep removing bit as we go. First bit removed is from reference bit , second bit removed is from dirty/modified bit. If there are no more chances then, we simply remove the page. At the end we give this address of physical frame to the new page

-Other logistics:

So, I (vikramjeet singh) did the coding and conceptual parts of the project and Josh colletti helped with the concepts and debugging in the project.

Overall, we have 139 test cases out of total 144 test cases. We got all test cases working FIFO and we did not get 5 test cases in third replacement, it looks like some edge cases . I tried working on it but it looked weird as the only very small part was not matching and it was just in wrong order which preference is given to something else due to some reason in 5 of those but overall, we have worked lots of hours everyday since day1 and have learned throughout the process of project.