CSSE 332 -- Operating Systems

Lazy Page Allocation

Name: Solution Key

Question 1. (10 points) In class, we did end up running `lazy 10 1` to allocate 10 integers in an array and then print them out 1 by 1. We then also tried `lazy 1024 100` to allocate 1024 integers and print them out every 100 values.

In both those cases, we were able to print out all the value in the array without triggering any page faults. Based on your understanding of paging and lazy page allocation, explain why we did not see any page faults triggered when we attempt to access those integers.

Solution: Since 10 and 1024 integers can all fit within a single page, we do not get any new page faults triggering while printing the integers. We only need a single page fault to access all of our integers.

Question 2. (10 points) After those two trials runs, we again tried `lazy 2048 100` and `lazy 3500 100`, and we saw that the number of page faults encountered **while printing the array values** increased. Based on your understanding of paging and the lazy page allocation scheme, explain why the number of pages faults increases.

Solution: When we go above 1024 integers, we cross the boundary of a single page and need more pages to contain our integers. This is why we start encountering more page faults as we are printing the integers. Roughly, every 1024 integers, we would encounter a new page fault.