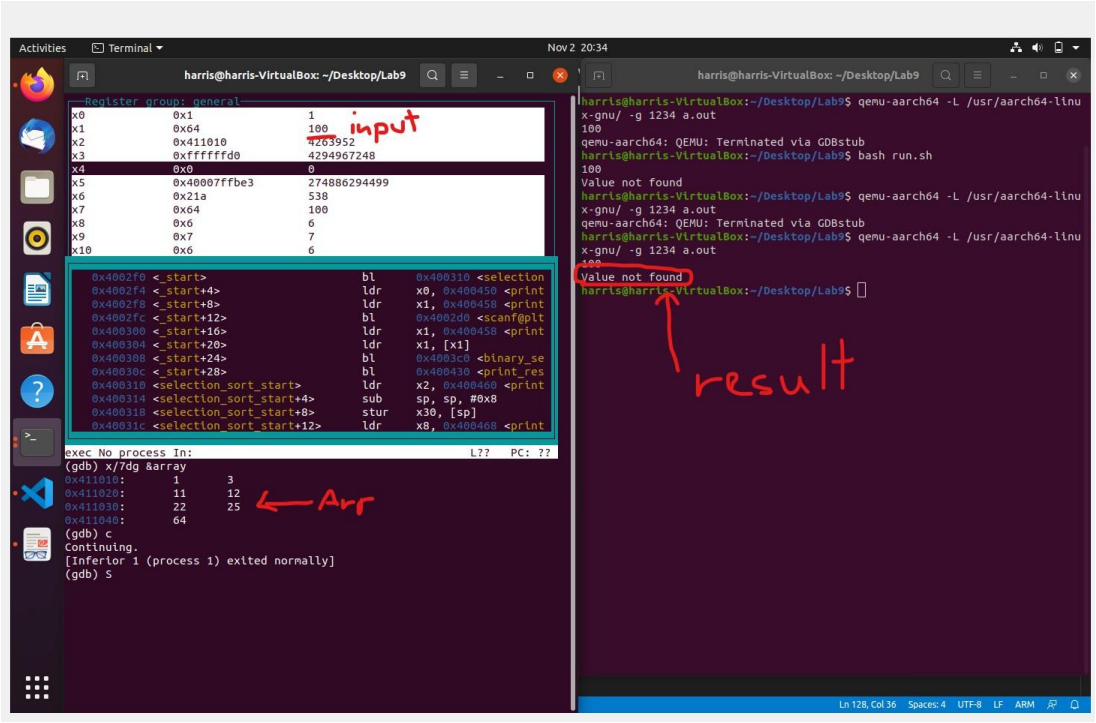
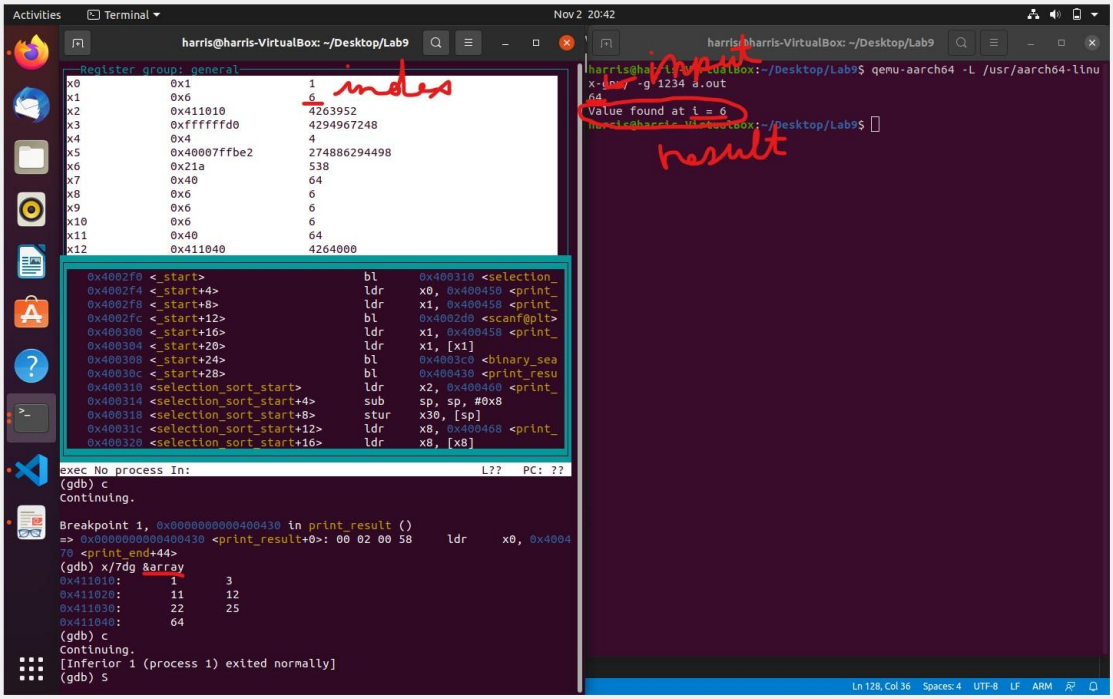
Name: Harris Spahic
Pledge: “I pledge my honor I have abided by the Stevens Honor System.”

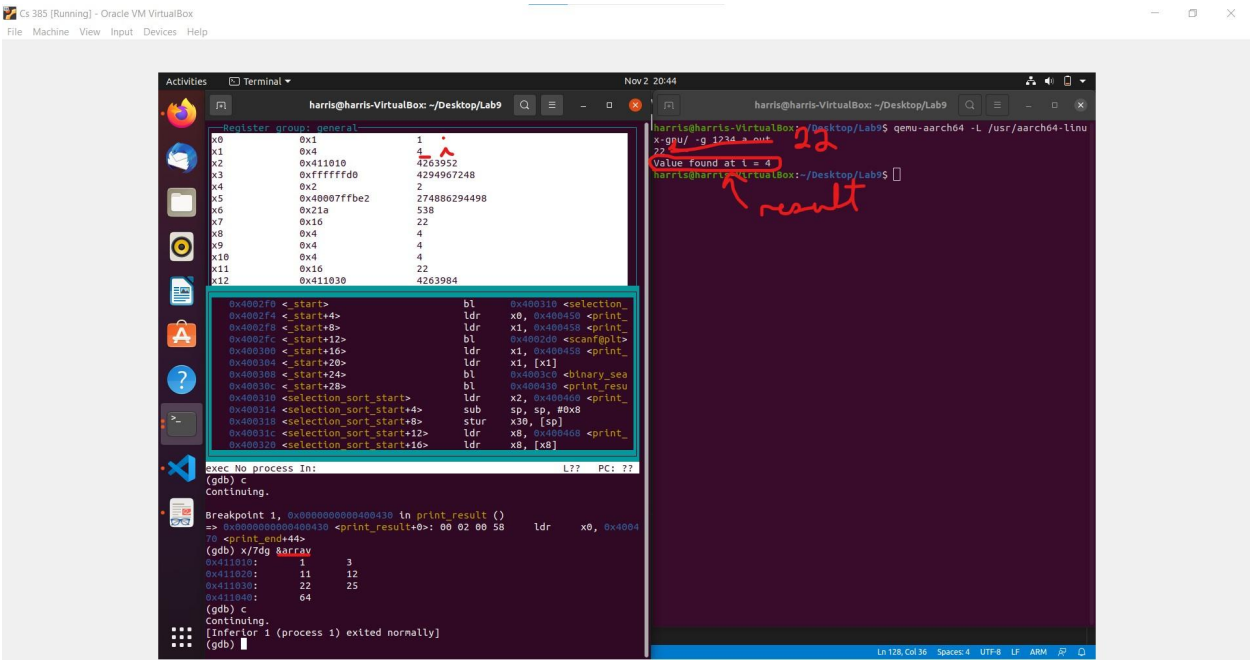
Writeup:



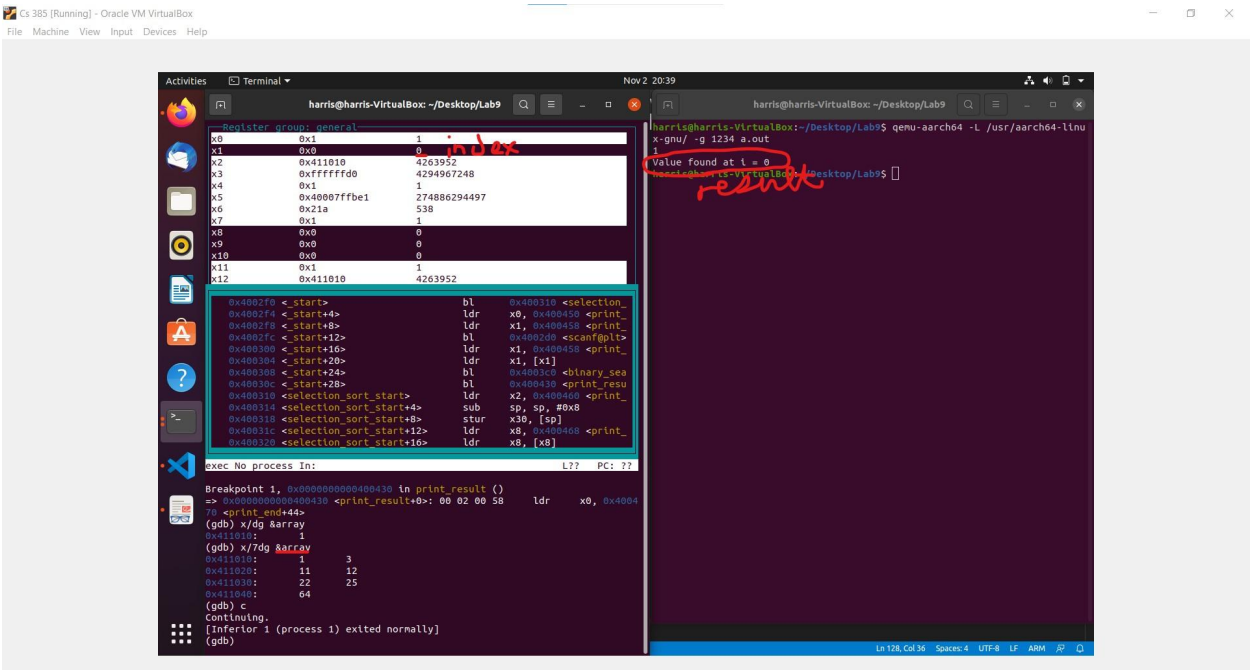
Case 1: In this case, I give an input that is not in my array (100). My program runs through binary search, has the left pointer edge surpass the right edge of the array and branches off to the “print_error” branch since the value was not found. Hence, x1 does not change to the index of (100) in my array since it was never found.



Case 2: Input is the last element of my sorted array. In this case binary_search repeatedly takes the right half of the array until it finds the first element. The index of the said element is stored into x1. Then binary_search branches to print_result, which prints that the “value was found at i = 6”. Since my size is 7, that is correct.



Case 3: Input is the middle element of my sorted array. I chose 22, the 5th element in my array. In this case binary_search alternates accordingly for splitting the left and right half of the array. The index of the said element is stored into x1. Then binary_search branches to print_result, which prints that the “value was found at i = 4”. Since i = 0 is the first element this is correct.



Case 3: Input is the first element of my sorted array. In this case binary_search takes the right half of the array repeatedly until it reaches the end. The index of the said element is stored into x1. Then binary_search branches to print_result, which prints that the “value was found at i = 0”. Since i = 0 is the first element this is correct.