

BLM19302E COMPUTER ORGANIZATION AND ARCHITECTURE

PROJECT 1 REPORT

Furkan Gündoğan 1821221005

09.05.2021

MAIN OBJECTIVES

By using ARM Assembly on Keil;

- Generating a minHeap structure by given array.
- Writing a find procedure that checks existence of a given number.
- Writing a sort procedure for the heap structure.

IMPLEMENTATIONS OF OBJECTIVES

1. Generating a minHeap by given Array

- At the beginning of this code, i did basic assignments to start creating a minHeap.
- Then I created a loop that continues until it see -1 in the array.
- During this loop, the index of the parent and current node are kept as variables.
- During loop every node is written into given heapAdress(0x40000100) + index.
- If number of nodes is $= (\text{parent} * 2) + 3$, then parent index is increase by one .
- Now I have both parent and current node indexes, so I can easily make parent-current comparisons and swaps in every stage.
- In this way, the loop continues and the minHeap structure is created.

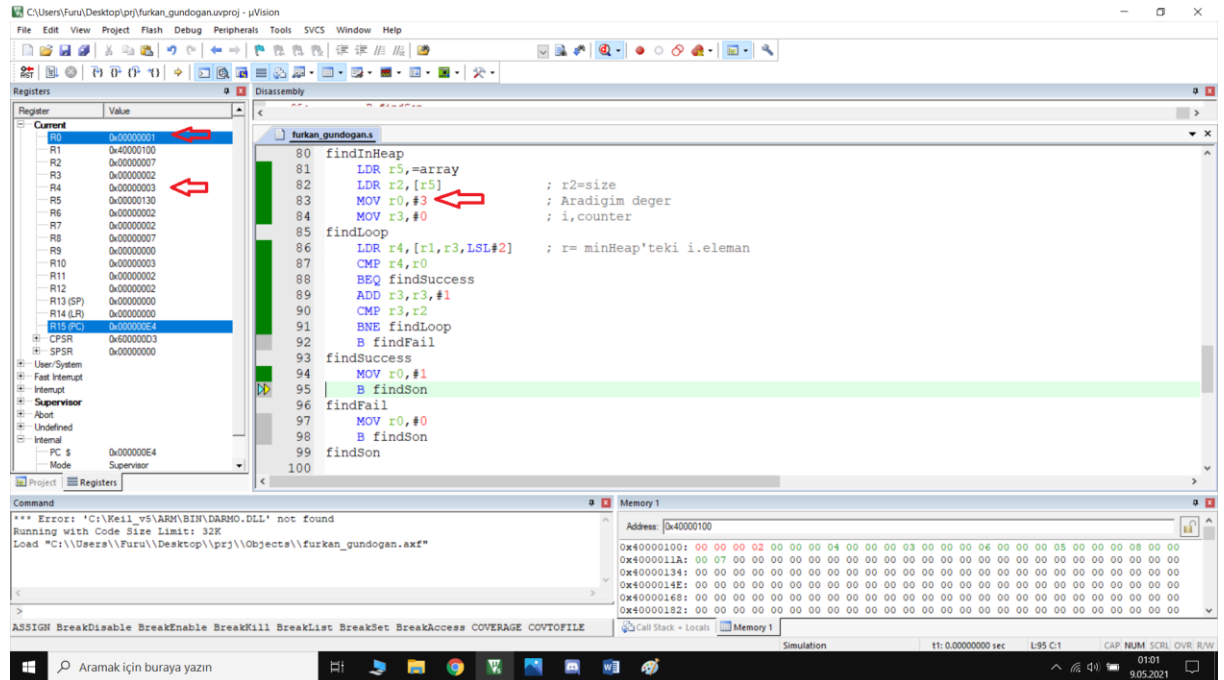
Array : 7,5,6,7,4,3,8,2,-1

Note : I'm giving read-write-execute permissions at 0x40000100- 0x40000500 on debugging

The screenshot displays the Keil uVision IDE interface. The main window shows ARM assembly code for a project named 'furkan_gundogan'. The code includes an 'AREA' directive for memory allocation, an 'ENTRY' point, and a loop that processes an array of values (7, 5, 6, 7, 4, 3, 8, 2, -1) to build a minHeap. Comments in Turkish explain the steps: initializing the array pointer, setting a counter, pushing the parent index, and calculating the current node index. A large blue arrow points from the code to a binary tree diagram on the right. The diagram shows a root node '2' with children '4' and '3'. Node '4' has children '6' and '5', and node '3' has children '8' and '7'. The bottom of the IDE shows the 'Registers' window with 'R1' at 0x40000100 and 'R2' at 0xFFFFFFFF. The 'Command' window shows an error message: 'Error: 'C:\Keil\v5\ARM\BIN\DARMO.DLL' not found'. The 'Memory' window shows the memory layout starting at 0x40000100.

2. Finding a value in Heap

- From the heap address, I looped through heap size. If I can find the value I was looking for, I assigned 1 to r0. I can't, I assigned 0 to r0.



3. Sorting Heap

- Heap structure is compatible with arrays. I did a sort operation by using this advantage.
- I assigned the last two indexes of the heap to the variables.
- Then I compared them and swapped if necessary.
- I reduced these indexes by one until I see the first element.
- Finally, the minHeap structure became sorted. (2345678)

