

Homework #7 Computer Organization Nov. 12, 2016 (Saturday) by 11:59 PM

Write a MIPS assembly language program to repeatedly search for target values interactively entered by the user. You must search for each target by calling a binary search function which returns the target value's array index if it is found. If the target value is not in the array, the non-existent index of -1 should be returned and used to report an unsuccessful search. The main's algorithm should be:

```
while (True) do // an infinite loop that we'll break out of
    target = input("Enter a value to search for (or 0 to exit): ")
    if target == 0 then // use 0 as the sentinel value to stop the program
        break
    end if
    index = binarySearch(array, length, target)
    if index == -1 then
        print target, " not found"
    else
        print target, " found at index ", index
    end if
end while
```

Your .data section should be as shown below:

```
.data
array: .word 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 60, 70, 80, 85, 95
length: .word 15 # number of items actually in the array

.text
.globl main

main:
    # MIPS Assembly language program here

    li $v0, 10 # system call to exit the program
    syscall
```

The high-level language algorithm for binarySearch that I want you to implement uses a recursive binarySearchHelper. Both functions are given below:

```
int binarySearch(int array[], length, int target) {
    return binarySearchHelper(array, 0, length-1, target);
} // end binarySearch

int binarySearchHelper(int array[], int start, int end, int target) {
    int middle;
    if (start > end) {
        return -1; // indicates item not found
    } else {
        middle = (start + end) // 2; (integer division by 2 use srl $t0, $t1, 1
        if (array[middle] == target) {
            return middle;
        } else if (array[middle] < target) {
            return binarySearchHelper(array, middle+1, end, target);
        } else {
            return binarySearchHelper(array, start, middle-1, target);
        } // end if
    } // end if
} // end binarySearchHelper
```

For this assignment, you'll need to **FOLLOW THE MIPS REGISTER CONVENTIONS** and use the PCSpim I/O syscalls like the "calculate powers example from pp. 7-10 of the MIPS Assembly Language Guide.

Recall that you can download the MIPS simulator at: <http://sourceforge.net/projects/spimsimulator/files/>
For Window's OS, you will want the QtSpim_9.1.17_Windows.exe and for MAC I'm told you want the QtSpim_9.1.17.mac.mpkg.zip

You should submit your homework via the Internet by following the directions at:

<http://www.cs.uni.edu/~fienu/cs1410f16/homework/submissionDirections.htm>

Basically, you put the following files in a hw7 folder and zip the folder to create a hw7.zip file (made on Windows by right-clicking on the hw7 folder and selecting `Send to|Compressed (zipped) folder`) containing:

- the MIPS assembly language program, e.g., hw7.s from any text-editor (e.g., WordPad)
- a window capture of the QtSpim simulator **Console window after running** your assembly language program and searching for the values: 5, 20, 85, 95 and 0. You can capture this window by (1) right-clicking anywhere in the window to make it the "currently active" window, (2) while holding down the <Alt> key, press the <PrtScn> key to capture the window into the Window's clipboard, and (3) open some word processor (Word, Open Office, etc.) and paste the image into the document. Add your name to this document before saving it.

On the top of the "directions" web-page, is a link to the submission tool

(https://www.cs.uni.edu/~schafer/submit/which_course.cgi). You'll need to enter your CatID username and password when requested.