CSCE-231/2303 Spring 2020 Assignment 4: MIPS and x86 Programming

Assigned: Thursday, 5-3-2020

Due: Sunday, 15-3-2020

Delayed submission with penalty until Tuesday, 17-3-2020

Goals

This assignment is an individual assignment and you will work on it on your own. The goal of this assignment is to write modular assembly programs that utilizes function calls using the stack.

Details

You are required to write the following functions using: 1) MIPS and 2) x86:

1. **Bubble Sort**
   You are required to write a function that sorts an array of words using the bubble sort algorithm. The function should have two parameters, the address of the word array to be sorted and the number of words in the array. **You are required to pass all the parameters on the stack**

   Hint: You should use this pseudocode as a helper for your implementation

   ```
   void bubbleSort(int arr[], int n)
   {
     int i, j;
     for (i = 0; i < n-1; i++)
     for (j = 0; j < n-i-1; j++)
       if (arr[j] > arr[j+1])
         swap(&arr[j], &arr[j+1]);
   }
   ```

2. **Binary search**
   You are required to write a **recursive function** that can search in a sorted array. The function should utilize the divide-and-conquer binary search algorithm. The function should have three parameters, the address of the word array, the value to be searched for and the number of items in the array. The function should return the index of the item that matches the search in case of a successful search, or -1 in case of search failure. **You are required to pass your parameters on the stack.**

Hint: You should use this pseudocode as a helper for your implementation

```
BinarySearch(SortedArray, value, low, high) {
  if (high < low)
    return not_found
  mid = (low + high) / 2
  if (SortedArray [mid] > value)
    return BinarySearch(SortedArray, value, low, mid-1)
  else if (SortedArray [mid] < value)
    return BinarySearch(SortedArray, value, mid+1, high)
  else
    return mid
}
```

Important: you are required to perform all needed validations and generate/report errors whenever needed.

What to submit

1. Your full in-line documented source code; it should include a main program that has some test cases that shows how your functions work.
2. A PDF report that includes:
   a. Any assumptions you have made.
   b. Your design approach to implement these function and why do you think it is the most optimum approach.
3. Screenshots for your running programs.

How to submit:

Compress all your work: source code, report, readme file, and any extra information into a zip archive. You should name your archive in the specific format <Student_ID>_<Name>_Assignment4.zip. Finally, upload your code to blackboard.

Grade

This assignment is worth 4% of the overall course grade. The assignment will be graded on a 100% grade scale, and then will be scaled down to the 5% its worth. The grading of the assignment will be broken down as follows:

1. 10 % for just submitting a meaningful assignment before or on the due date. This 10% does not account for the correctness of your assignment but submitting an empty assignment without code will definitely results in loosing this 10% and consequently the whole grade of this assignment.
2. 65 % for the correctness and the quality of your code.
3. 25 % for the quality of your inline documentation and the readme file.

Delays

You have up to 2 working days of delay, after which the assignment will not be accepted and your grade in that case will be ZERO. For every day (of the 2 allowed days), a penalty of 10% will be deducted from the grade. And of course, you will lose the 10% mentioned in point 1 above under the “Grade” section.