ICS 233, Term 142

Computer Architecture & Assembly Language

Programming Assignment# 1
Due date: Tuesday, March 10, 2015

Q.1. Write a MIPS assembly program to implement the procedure, **BinarySearch** to search an array which has been previously sorted in an ascending order. Each element in the array is a 32-bit signed integer. Three parameters should be passed as procedure arguments in $a0, $a and $a2 registers: the address of the array to be searched, the size (number of elements) of the array, and the number to be searched. If the number is found then **BinarySearch** returns in the $v0 register the position of the number in the array. Otherwise, -1 is returned in $v0. Your procedure should preserve all necessary registers on the stack.

The pseudocode for the **BinarySearch** procedure is given below:

```
BinarySearch (array, size, number) {
    lower = 0;
    upper = size-1;
    while (lower <= upper) {
        middle = (lower + upper)/2;
        if (number == array[middle])
            return middle;
        else if (number < array[middle])
            upper = middle–1;
        else
            lower = middle+1;
    }
    return -1;
}
```

Write a complete program to use the procedure **BinarySearch** to search for the number **5** in the sorted array given below:

Array: .word 1, 3, 4, 5, 9, 11, 20, 29

Note that the size of the array in this case is 8 and the **BinarySerach** procedure should return the position of number 5 as 3.

Clearly indicate in your assembly code where each pseudo code statement is translated. Also clearly indicate what registers are used to store the variables.

The solution should be well organized and your program should be well documented. Submit a soft copy of your solution in a zip file. Your solution should be submitted in a ***word file*** *that contains the following items:*

i. *Your name and ID*

ii. *Assignment number*

iii. *Problem statement*

iv. *Your solution along with the code*

v. *Discussion of what worked and what did not work in your program. Include snapshots that demonstrate the working parts of your program. If things did not work and you attempted to solve them, mention that and write about the difficulty that you have faced.*