

Computer Organization Homework #10

Due: December 5, 2008 (F)

Translate the following quick sort algorithm to MIPS assembly language. **(YOU ARE TO USE THE MIPS CALLING CONVENTIONS WHEN IMPLEMENTING THE QUICK SORT, PARTITION, AND SWAP SUBPROGRAMS!!!)**

```
main( )
begin
    quickSort(array, 0, n-1)
end main
```

```
quickSort( start address of array with n integers,
           integer low, high)
```

```
local variable
    integer pivotPosition
```

```
begin
    if (low < high) then
        pivotPosition = partition(array, low, high)
        quickSort(array, low, pivotPosition-1)
        quickSort(array, pivotPosition+1, high)
    end if
end quickSort
```

(swap should exchange the values at the addresses passed it)

```
swap( address of integer operand1,
      address of integer operand2)
```

```
local variable
    integer temp
```

```
begin
    temp = operand1
    operand1 = operand2
    operand2 = temp
end swap
```

```
function partition( start address of array with n integers,
                   integer low, high ) returns an integer
```

```
local variables
    integer left, right, pivot
```

```
begin
    left = low
    right = high
    pivot = array[low]
    while left < right do
        while array[right] > pivot do
            right = right - 1
        end while
        while (left < right) and (array[left] <= pivot) do
            left = left + 1
        end while
        if left < right then
            swap(array[left], array[right]) //pass addresses
                                           // of these elts.
        end if
    end while
    array[low] = array[right]
    array[right] = pivot
    return right
end partition
```

Turn in the following:

- 1) The MIPS assembly language program -- include comments describing which registers are being used for parameters and local variables
- 2) The "output" of the MIPS program using the following 11 integers for your initial data:

8 2 7 10 5 6 3 4 9 3 1

(I want a screen shot of the MIPS simulator including the sorted array in memory **after** the program has run).

Extra Credit: Use the PCSpin I/O system calls described on page 7 of the MIPS Guide to allow the user to interactively enter the elements to be sorted and then display the sorted results.

General Directions (on back): (See hw #8 description for detailed directions:

<http://www.cs.uni.edu/~fienup/cs041f08/homework/hw8.pdf>)

General Directions:

- 1) Write your assembly language program on paper first! I will not help anyone debug their program without your handwritten program.
- 2) Type in your program using WordPad. Remember to save it as a **Text Document** on a USB flash memory stick.
- 3) Debug your MIPS assembly language program.
- 4) When it is correct, **run it to completion** and copy to the Window's clipboard a snapshot of the PCSpim window by using the <Alt> and <Print Screen> keys together.
- 5) Open up new Word document and set its page layout to Landscape by File | Page Setup | Paper Size and then select Landscape.
- 6) Paste the snapshot of the PCSpim Debugger window into the Word document. Resize the snapshot to the margins and print a copy to turn in.
- 7) Print a copy of the assembly language program to turn in too.
- 8) Hand in a copy of your assembly language program **and** the snapshot of the PCSpim window showing the resulting sorted memory.