

## Computer Organization Homework #9

Due: May 2, 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).

**General Directions:** (This assume you are using PCSpim, but xspim is the labs under Linux too)

1) Log on the Wright lab using your AD-ITS (same account you use for the Library, Union, Dorm, and various hallway computers on campus. If you need your password reset, you must go to ITT 36 for ITS User Support. CNS Support CANNOT reset passwords for student accounts.)

- 2) Write your assembly language program on paper first! I will not help anyone debug their program without your handwritten program.
- 3) Type in your program using WordPad and save it on a USB flash memory stick. Remember to use quotes around the file name "hw9.s" (Details below)
- 4) Debug your MIPS assembly language program.
- 5) 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.
- 6) Open up new Word document and set its page layout to Landscape by File | Page Setup | Paper Size and then select Landscape. (Details below)
- 7) 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.
- 8) Print a copy of the assembly language program to turn in too.
- 9) Hand in a copy of your assembly language program **and** the snapshot of the PCSpim window showing the resulting sorted memory.

### **Entering the Program using WordPad:**

- 1) Start | Programs | Accessories | WordPad
- 2) Type in the program in a new file
- 3) File | Save As  
Save in file name: "hw9.s" (USE DOUBLE-QUOTES AROUND FILE NAME)
- 4) After debugging, print a copy of the program to hand in

### **Running the Program using PCSpim:**

- 1) Start | Programs | Programming | PCSpim
- 2) Maximize the window by clicking on the 2nd icon in the upper right of the window
- 3) Load the program by File | Open
- 4) Observe the *initial* DATA values of the array in memory (as hexadecimals) before the program runs.
- 5) Observe the *initial* register are all 0's before the program runs
- 6) Run the program by Simulator | Go, then Click "OK" in the Run Parameters window
- 7) Observe the *resulting* DATA values for the "array" in memory (as hex) after the program runs. **The elements in the array should be sorted!**
- 8) Observe the *resulting* register values after the program runs
- 9) Copy the PCSpim window to the Window's Clipboard using <Alt><Print Screen> keys

### **Printing the PCSpim window using Word:**

- 1) Start | Programs | Microsoft Office | Microsoft Office Word 2003
- 2) Type your name at the top of the blank page and hit <enter>
- 3) Paste in the PCSpim window from the Clipboard by Edit | Paste
- 4) Change to landscape by File | Page Setup and then click "Landscape" button
- 5) Resize the frame containing the PCSpim picture by clicking on the frame drag the lower-right corner to make the picture bigger
- 6) Print this picture of the PCSpim window to hand in

### **Downloading MIPS simulator (Spim) for your home computer:**

The Spim web-site is <http://pages.cs.wisc.edu/~larus/spim.html>

Download PCspim if you are using Microsoft Windows (Windows NT, 2000, XP). (Will probably run under Vista versions -- let me know if you try it!)

To run spim or xspim on a Unix or Linux system, copy either the compressed tar file

(<http://www.cs.wisc.edu/~larus/SPIM/spim.tar.Z>) or the gzip'ed tar file

(<http://www.cs.wisc.edu/~larus/SPIM/spim.tar.gz>). Both files contains source code and must be compiled.