CS 366: Programming Assignment 3

Merge Sort

Due: 02 Nov 2005, 11:59 AM

Merge Sort Implement a recursive form of Merge sort. That is implement a function of the form:

```
MergeSort(int *array, int length) {

  ...

  /* This does the division of the array */
  if (length > 1) {
    MergeSort(array, length/2);
    MergeSort(array + length/2, length-length/2);
  }

  /* Do the merge here */
  ...
}
```

The function accepts two arguments viz., a pointer to an integer array (`int *array`) and the length or number of elements in this array (`int length`).

Initialization The program accepts just one input: the size of the array to be sorted. Space for this array may be allocated on the stack (like a local variable for `main()`) and pass the address as an argument in the call to MergeSort(). [MIPS also has a way of allocating storage on the heap, refer section 6.7 in the book]

It would be ideal to use a random number generator like the one shown below or the one given in section 9.7 of the book, to generate random numbers to populate the array. The C equivalent of the MIPS code shown below is `unsigned int fastrand (unsigned int s);`

```
fastrand:

move       $2,  $4          # R2 = S (arg passed in R4)
li         $8,  33614       # R8 = 2      onstant A

multu      $2,  $8          # HI, LO = A
mflo       $9               # R9 = Q = bits 00..31 of A
srl        $9,  $9, 1
mfhi       $10              # R10 = P = bits 32..63 of A
addu       $2,  $9, $10     # R2 = S' = P + Q
bltz       $2,  overflow    # handle overflow (rare)
```

```
j          $31             # return (result in R2)

overflow:

sll        $2, $2, 1       # zero bit 31 of S’
srl        $2, $2, 1
addiu      $2, 1           # increment S’
j          $31             # return (result in R2
```

Debug This is not a required option, but one that might be very helpful to see the sequence of function calls. That is for every invocation of MergeSort() print the address (`int *`) and `length`. You could also have this be selectively turned on/off based on an input from the user.

Output The program is should print the unsorted array before calling MergeSort() and the sorted array after the return.

Turnin The command for the electronic turnin is:

```
turnin -c cs366 -p merge ⟨files⟩
```

The hard-copy should contain a listing of your program. Also have a pseudo-code (or some high-level language) listing for the algorithm you used, the testing done and the status of your implementation (what works and what doesn't). This is due in class.