HW1 report

Register mapping table for each function.

1. main function
$sp -4 used to make room in the stack for return address
$ra used to store the return address
$a0 and $a1 used by mips pseudo instruction la to store the address of A and Buf in the algorithm
$a2 is used to load the length of the array which is 20 according to the input of the algorithm
$a3 is used to set the left=o
$t0 used to temporarily store the address of unsorted array
$t1 used to temporarily store the length of the array
$v0 used to system call for the print string
$a0 also used to print info0 or newline or space later it is used in la pseudo instruction to point the address of the string to print
$t4 used to set the i to 0 and used in print unsorted function the print the unsorted item from 0 to 20

2.merge_sort function
$sp -20 used to make room in the stack for return address
$ra used to save the return address
$s1 used to store the address of buf
$s2 used to save right
$s3 used to save left
$s4 used to save mid
$t3 if left<right then $t3=1 else if $t3=0 then the algorithm finished
$s4 used to store (left+right)/2
$a2 used to store the left argument
$a3 used to store the mid argument
Then mergesort(a, left, mid)
$t4 store the mid+1
Then $a3 is set to equal to $t4, assign mid+1 to left
$a2 didn’t change, it is right
Then merge(a,mid+1,right)
After sort begin to merge
$a1 reassign the buf address to $a1
$a2 reassign right to $a2
$a3 reassign left to $a3
$a4 reassign mid to $a4

3.merge function

$sp -20 used to make room in the stack for return address
$ra used to save the return address
$s1 used to store the address of buf
$s2 used to save right
$s3 used to save left
$s4 used to save mid
$t1 corresponds to left_i
$t2 corresponds to right_i
$t3 corresponds to i
$t4 if mid<left_i then $t4=1 , if $t4 != 0 go to while 3 (left_i > mid)
$t5 $t5=1 if right<right_i go to while 2 if right_i > right
$t6 $t6 = address buf[left_i++]
$s5 =buf[left_i++]
$t7 = address of buf[right_i++]
$s6 = buf[right_i++]
$t4 if buf[left_i] < buf[right_i] then $t4=1 and if $t4=0 then exit(buf[left_i] >= buf[right_i])
$t8 = address of A[i]

In the begin function
$t1 is used to reassign left to left_i

4.for1tst
$t5 if left_i< i then $t5=1 and if $t5=0 then branch to finish
$s7 $s7 = A[left_i]
Others gets the same usage as main

5. print_unsorted_array

$t6 While (i< length) $t6=0 then prepare mergesort
$t0 used to store i*4
$t6 used to load buf[i]
Others gets the same usage as main

6. prepare_mergesort
$sp make room on the stack
$ra store the return address
$a1 save address of buf
$a2=right-1 index
$a3=left

6.print_sorted
$t7 used to set I to 0

Others gets the same usage as main

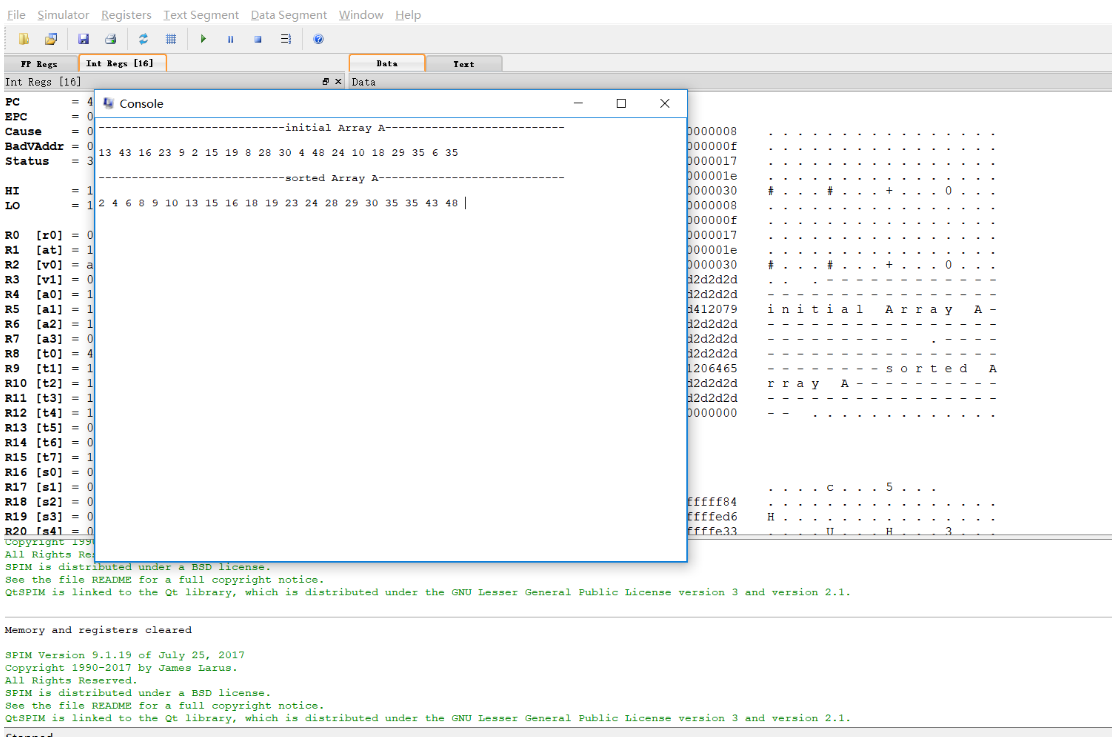
7. print_sorted_array
$t6 used to determine if exit the loop
$t7=i
$t0=4*i
Others gets the same usage as main

8.terminate
The registers in terminate function gets the same usage as main.

Brief Explanation on the implementation

The total implementation consists of two parts : algorithm part and print part
In the algorithm part merge_sort used to divide and conquer. Separate the array by mid and recursively call the mergesort. Finally merge the two part separated by mid.
The merge part used to make an order of the integer in the array.
Compare numbers and make an right order. And copy the right ordered value to A[] array and end when finished.

In the print part, it print the unsorted array first and then print the sorted array. And it also has the ability to prepare the mergesort.



```
----------------------------initial Array A---------------------------

13 43 16 23 9 2 15 19 8 28 30 4 48 24 10 18 29 35 6 35

----------------------------sorted Array A----------------------------

2 4 6 8 9 10 13 15 16 18 19 23 24 28 29 30 35 35 43 48
```

```
SPIM is distributed under a BSD license.
See the file README for a full copyright notice.
QtSPIM is linked to the Qt library, which is distributed under the GNU Lesser General Public License version 3 and version 2.1.

Memory and registers cleared

SPIM Version 9.1.19 of July 25, 2017
Copyright 1990-2017 by James Larus.
All Rights Reserved.
SPIM is distributed under a BSD license.
See the file README for a full copyright notice.
QtSPIM is linked to the Qt library, which is distributed under the GNU Lesser General Public License version 3 and version 2.1.
```