

COMPUTER ORGANIZATION (CDA – 3103)

SPRING 2005

Lab # 8: Implementation of Sorting Problem on MIPS

1	C++ CODE FOR SORTING 4 INTEGERS	1
2	MIPS CODE FOR SORTING 4 INTEGERS.....	2

For reference use:

1. *Previous SPIM Lab Handouts (available on Course Website)*
2. *Appendix A of your text book.*

1 C++ CODE FOR SORTING 4 INTEGERS

```
#include <iostream.h>

void main()
{
    int t0 , t1 , t2 , t3;
    int a1 , a2 , a3;

    cout<<"First Number = ";
    cin>>t0;
    cout<<"Second Number = ";
    cin>>t1;
    cout<<"Third Number = ";
    cin>>t2;
    cout<<"Fourth Number = ";
    cin>>t3;

    a3 = 3;

    for(a1 = 1 ; a1 <= a3 ; a1++)
    {
        // swap them t0 and t1 if required
        if(t0 > t1)
        {
            a2 = t1;
            t1 = t0;
            t0 = a2;
        }

        // swap them t1 and t2 if required
        if(t1 > t2)
        {
            a2 = t2;
            t2 = t1;
            t1 = a2;
        }

        // swap them t2 and t3 if required
        if(t2 > t3)
        {
            a2 = t3;
            t3 = t2;
            t2 = a2;
        }
    }

    cout<<endl<<t0 <<" , " <<t1 <<" , " <<t2 <<" , " <<t3 <<endl<<endl;
}
```

2 MIPS CODE FOR SORTING 4 INTEGERS

main:

data area where we can declare static string

.data

```
FIRST_NUM:      .asciiz "First Number = "  
SECOND_NUM:     .asciiz "Second Number = "  
THIRD_NUM:      .asciiz "Third Number = "  
FOURTH_NUM:     .asciiz "Fourth Number = "  
COMMA:          .asciiz ", "
```

.text

```
#-----  
# read first number from user and save in reg t0  
  
li $v0, 4  
la $a0, FIRST_NUM # address of string to print  
syscall           # print the string  
  
li $v0, 5  
syscall  
move $t0, $v0  
  
#-----  
# read second number from user and save in reg t1  
  
li $v0, 4  
la $a0, SECOND_NUM # address of string to print  
syscall           # print the string  
  
li $v0, 5  
syscall  
move $t1, $v0  
  
#-----  
# read third number from user and save in reg t2  
  
li $v0, 4  
la $a0, THIRD_NUM # address of string to print  
syscall           # print the string  
  
li $v0, 5  
syscall  
move $t2, $v0  
  
#-----  
# read fourth number from user and save in reg t3  
  
li $v0, 4  
la $a0, FOURTH_NUM # address of string to print  
syscall           # print the string  
  
li $v0, 5  
syscall  
move $t3, $v0
```

```
#-----  
#-----  
# Loop for sorting  
#-----  
#-----
```

```
# a3 contains the total number of iteration of loop  
li $a3, 3 # a3 = 3
```

```
# a1 contains the current iteration of loop  
li $a1, 0 # a1 = 0
```

LOOP:

```
add $a1, $a1, 1 # a1=a1+1
```

```

COMPARE_1_2:
    # compare and swap if required
    bge $t1 , $t0 , COMPARE_2_3
    move $a2, $t1
    move $t1, $t0
    move $t0, $a2

COMPARE_2_3:
    # compare and swap if required
    bge $t2 , $t1 , COMPARE_3_4
    move $a2, $t2
    move $t2, $t1
    move $t1, $a2

COMPARE_3_4:
    # compare and swap if required
    bge $t3 , $t2 , COMPARE_DONE
    move $a2, $t3
    move $t3, $t2
    move $t2, $a2

```

```

COMPARE_DONE:

```

```

    bne $a1, $a3, LOOP    # if a1!=a3, then jump back to loop

```

```

#-----
#-----
# Print Sorted Number
#-----
#-----

```

```

    li $v0, 1                # system call code for print_int
    move $a0 , $t0           # print int
    syscall

    li $v0, 4                # system call code for print_str
    la $a0, COMMA            # address of string to print
    syscall                  # print the string

    li $v0, 1                # system call code for print_int
    move $a0 , $t1           # print int
    syscall

    li $v0, 4                # system call code for print_str
    la $a0, COMMA            # address of string to print
    syscall                  # print the string

    li $v0, 1                # system call code for print_int
    move $a0 , $t2           # print int
    syscall

    li $v0, 4                # system call code for print_str
    la $a0, COMMA            # address of string to print
    syscall                  # print the string

    li $v0, 1                # system call code for print_int
    move $a0 , $t3           # print int
    syscall

    li $v0, 10               # exit
    syscall

```