Name: KAUSHAL VASHISTH Roll no.: 18ETCS002147

Laboratory 6

Title of the Laboratory Exercise: Sorting

1. Introduction and Purpose of Experiment

   Students will create assembly code with sorting techniques and nested loops

2. Aim and Objectives

   Aim

   To develop assembly language program to perform sorting using nested loop structures

   Objectives

   At the end of this lab, the student will be able to

   - use nested loops in assembly
   - perform sorting in ascending/ descending order
   - Build complex looping logic in assembly language

3. Experimental Procedure

   1. Write algorithm to solve the given problem
   2. Translate the algorithm to assembly language code
   3. Run the assembly code in GNU assembler
   4. Create laboratory report documenting the work

4. Questions

   Develop an assembly language program to perform the following

   1. To design calculator to perform all arithmetic operations based on input given by user.
   2. To perform SWAP operation using Logical instructions
   3. To compute factorial of a number.
   4. To find second smallest number in an unsorted array.

5. Calculations/Computations/Algorithms

Program for calculator:

```
.section .data
add:
        .int 1
sub:
        .int 2
mul:
        .int 3
div:
        .int 4
user_input:
        .int 3
.section .text
.globl _start
_start:
        cmpl $1,user_input
        JE add_op
        cmpl $2,user_input
        JE sub_op
        cmpl $3,user_input
        JE mul_op
        cmpl $4,user_input
        JE div_op

     add_op:
        movl $10,%eax
        movl $20,%ebx
        addl %eax,%ebx
     sub_op:
        movl $10,%eax
        movl $20,%ebx
        subl %eax,%ebx
     mul_op:
        movl $10,%eax
        movl $20,%ebx
        mull %ebx
```

```
div_op:
    movl $10,%eax
    movl $20,%ebx
    divl %ebx

    movl $1,%eax
    movl $0,%ebx
    int $0x80
```

Program to swap two numbers using logical operations:

```
.section.data
.section    .text
.globl  _start
_start:

    movl $100, %eax
    movl $200, %ebx
    xorl %ebx, %eax
    xorl %eax, %ebx
    xorl %ebx, %eax

    movl $1, %eax
    movl $0, %ebx
    int $0x80
```

Program to find the factorial of a number:

```
.section .data

.section .text
.globl _start
_start:

    movl $5,%eax
    movl $4,%ebx

    loop:
        mull %ebx
        subl $1,%ebx
        cmpl $1,%ebx
        JNE loop

    mov $1,%eax
    movl $0,%ebx
    int $0x80
```

Name: KAUSHAL VASHISTH Roll no.: 18ETCS002147

Program to find the second smallest integer in a unsorted array:

```
.section .data
array:
        .int 4,2,6,9,10,7
array1:
        .int 0,0,0,0,0,0
second_smallest:
        .int 0
.section .text
.globl _start
_start:
        movl $0,%ebx
        movl $0,%ecx
        movl $0,%edx
loop1:
        movl array( ,%ecx,4),%eax
        cmpl %ebx,%eax
        je loop2
inc:
        addl $1,%ecx
        cmpl $6,%ecx
        jne loop1
        addl $1,%ebx
        movl $0,%ecx
        cmpl $16,%eax
        jne loop1

loop2:
        movl %eax,array1( ,%edx,4)
        addl $1,%edx
        cmpl $6,%edx
        jne inc
        movl $1,%edx
        movl array1( ,%edx,4),%eax
        movl %eax,second_smallest

        movl $1,%eax
        movl $0,%ebx
        int $0x80
```

6. Presentation of Results

Output for calculator:

```
Breakpoint 22, div_op () at ex6.s:37
37               movl $10,%eax
(gdb) info registers
eax            0xc8     200
ecx            0x0      0
edx            0x0      0
ebx            0x14     20
esp            0xbffff050       0xbffff050
ebp            0x0      0x0
esi            0x0      0
edi            0x0      0
eip            0x80480bc        0x80480bc <div_op>
eflags         0x202    [ IF ]
cs             0x73     115
ss             0x7b     123
ds             0x7b     123
es             0x7b     123
fs             0x0      0
gs             0x0      0
```

Output for swapping two numbers:

```
eax            0xc8     200
ecx            0x0      0
edx            0x0      0
ebx            0x64     100
esp            0xbffff040       0xbffff040
ebp            0x0      0x0
esi            0x0      0
edi            0x0      0
eip            0x8048064        0x8048064 <_start+16>
eflags         0x202    [ IF ]
cs             0x73     115
ss             0x7b     123
ds             0x7b     123
es             0x7b     123
fs             0x0      0
gs             0x0      0
```

Output for finding the factorial:

```
Breakpoint 3, loop () at lab6_3.s:14
14                      JNE loop
(gdb) info registers
eax            0x78     120
ecx            0x0      0
edx            0x0      0
ebx            0x1      1
esp            0xbffff5b0       0xbffff5b0
ebp            0x0      0x0
esi            0x0      0
edi            0x0      0
eip            0x8048066        0x8048066 <loop+8>
eflags         0x246    [ PF ZF IF ]
cs             0x73     115
ss             0x7b     123
ds             0x7b     123
es             0x7b     123
fs             0x0      0
gs             0x0      0
(gdb) -

[2]+  Stopped                 gdb lab6_3
```

Output for finding the second smallest integer in an unsorted array:

```
Breakpoint 1, loop2 () at exe.s:35
35               movl $1,%eax
(gdb) info registers
eax            0x4      4
ecx            0x4      4
edx            0x1      1
ebx            0xa      10
esp            0xbffff050       0xbffff050
ebp            0x0      0x0
esi            0x0      0
edi            0x0      0
eip            0x80480c3        0x80480c3 <loop2+32>
eflags         0x246    [ PF ZF IF ]
cs             0x73     115
ss             0x7b     123
ds             0x7b     123
es             0x7b     123
fs             0x0      0
gs             0x0      0
```

7. Analysis and Discussions:

   To complete the given problems we have made use of looping statements, compare statements and XOR operation to swap the numbers.

Signature and date