**Microprocessor and Assembly Programming Laboratory**

**B.Tech. V Semester**



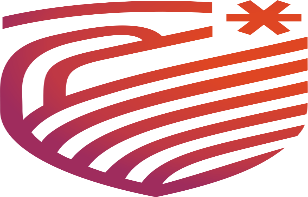
**Name : Harshit Kumar**

**Roll Number : 21ETMC412011**

**Department : Computer Science and Engineering**

**Faculty of Engineering & Technology**

**Ramaiah University of Applied Sciences**



**Ramaiah University of Applied Sciences**

Private University Established in Karnataka State by Act No. 15 of 2013

|  |  |
| --- | --- |
| Faculty | Engineering & Technology |
| Programme | B. Tech. in Computer Science and Engineering |
| Year/Semester | 2021/3rd Semester |
| Name of the Laboratory | Microprocessor and Assembly Programming Laboratory |
| Laboratory Code | 21CSL201A |

# Laboratory 1

Title of the Laboratory Exercise: Data transfer operations

1. Introduction and Purpose of Experiment

Students will be able to define data of different data types and perform data transfer operations on the data

1. Aim and Objectives

Aim

To develop assembly language program to perform data transfer operations on different data.

Objectives

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

* + Define data of different data types
  + Perform data transfer operations
  + Create a simple assembly language program
  + Use GAS assembler
  + Understand GNU debugger

1. 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 a laboratory report documenting the work

1. Questions

1. Perform the following data transfer operations

|  |  |
| --- | --- |
| 1. 32 bit integer data to a | General Purpose register  Segment Register  Memory |
| 2. 16 bit integer data to a | General Purpose register  Segment Register  Memory |
| 3. 8 bit integer data to a | General Purpose register  Segment Register  Memory |
| 4. 32 bit integer data from a General purpose register to a  *(Repeat the same for 16 bit integer data and 8 bit integer data)* | General Purpose register  Segment Register  Memory |
| 5. 32 bit integer data from memory to a  *(Repeat the same for 16 bit integer data and 8 bit integer data)* | General Purpose register  Segment Register  Memory |
| 6. 32 bit integer data from memory to | Memory region |

1. Calculations/Computations/Algorithms

.section **.data**

    int32:

        .int 0

    int16:

        .octa 0

    int8:

        .quad 0

.section **.text**

**.global** \_start

\_start:

    # moving 32 bit integers

    movl $128, %eax     # constant to register

    movl $125, int32    # constant to memory

    movl %eax, %ebx     # register to register

    movl %eax, int32    # register to memory

    movl int32, %ecx    # memory to register

    # moving 16 bit integers

    movw $128, %ax      # constant to register

    movw $125, int16    # constant to memory

    movw %ax, %bx       # register to register

    movw %ax, int16     # register to memory

    movw int16, %cx     # memory to register

    # moving 8 bit integers

    movb $48, %al      # constant to register

    movb $45, int8    # constant to memory

    movb %al, %bl       # register to register

    movb %al, int16     # register to memory

    movb int16, %cl     # memory to register

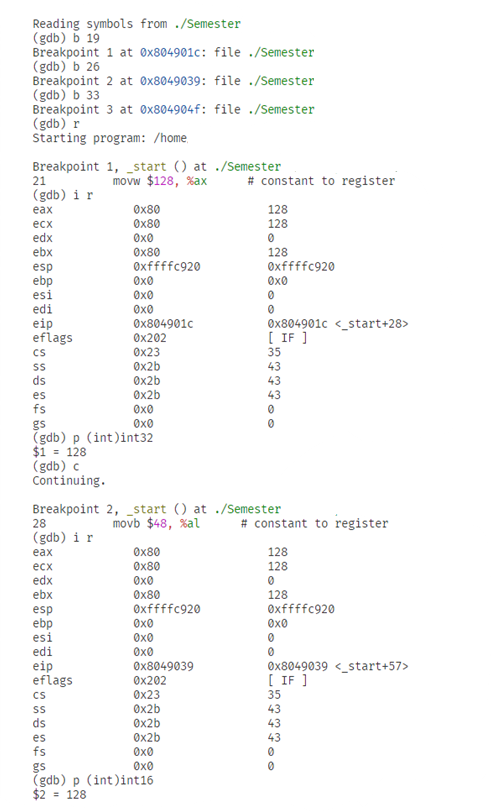
    # exit

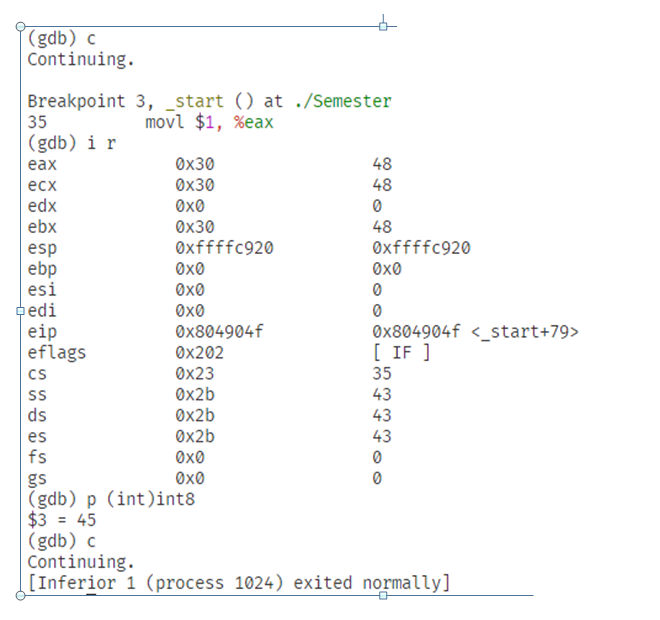
    movl $1, %eax

    movl $0, %ebx

    int $0x80

1. Presentation of Results





1. Conclusions

The programme executed successfully.

1. Comments

1. Learning happened

Learnt to copy data from one location to another

Signature and date Marks