Microprocessor and Assembly Programming Laboratory

B.Tech. III Semester

Name : DEEPAK R

Roll Number : 18ETCS002041

Department : Computer Science and Engineering

Faculty of Engineering & Technology

Ramaiah University of Applied Sciences

Name: **DEEPAK**

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	2018/3rd Semester
Name of the Laboratory	Microprocessor and Assembly Programming Laboratory
Laboratory Code	

List of Experiments

1. Data transfer operations	
2. Arithmetic operations	
3. Logical operations	
4. Controlling execution flow using conditional instructions	
5. String manipulation	
6. Searching an element in an array	
7. Sorting an array	
8. Interfacing	
9. Interfacing	

Name: **DEEPAK R** Registration Number: **18ETCS002041**

No.	Lab Experiment	Viva (6)	Results (7)	Documentation (7)	Total Marks (20)
1	Data transfer operations				
2	Arithmetic operations				
3	Logical operations				
4	Controlling execution flow using conditional instructions				
5	String manipulation				
6	Searching an element in an array				
7	Sorting an array				
8	Interfacing				
9	Interfacing				
10	Lab Internal Test conducted along the lines of SEE and valued for 50 Marks and reduced for 20 Marks				
	Total Marks				

Name: **DEEPAK R** Registration Number: **18ETCS002041**

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

2. 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

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 a laboratory report documenting the work

4. 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

5. Calculations/Computations/Algorithms

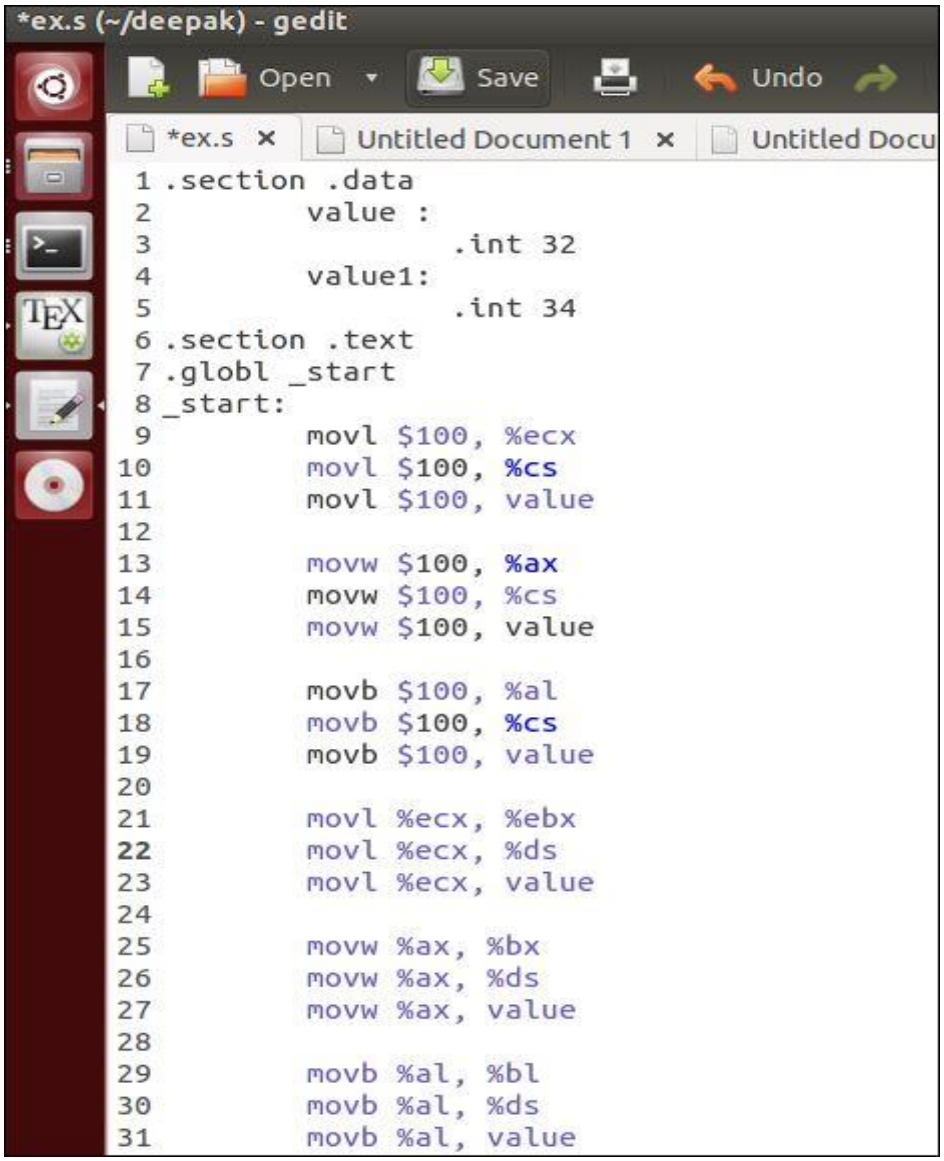


FIG 1 SHOWING CODES FOR DATA TRANSFER

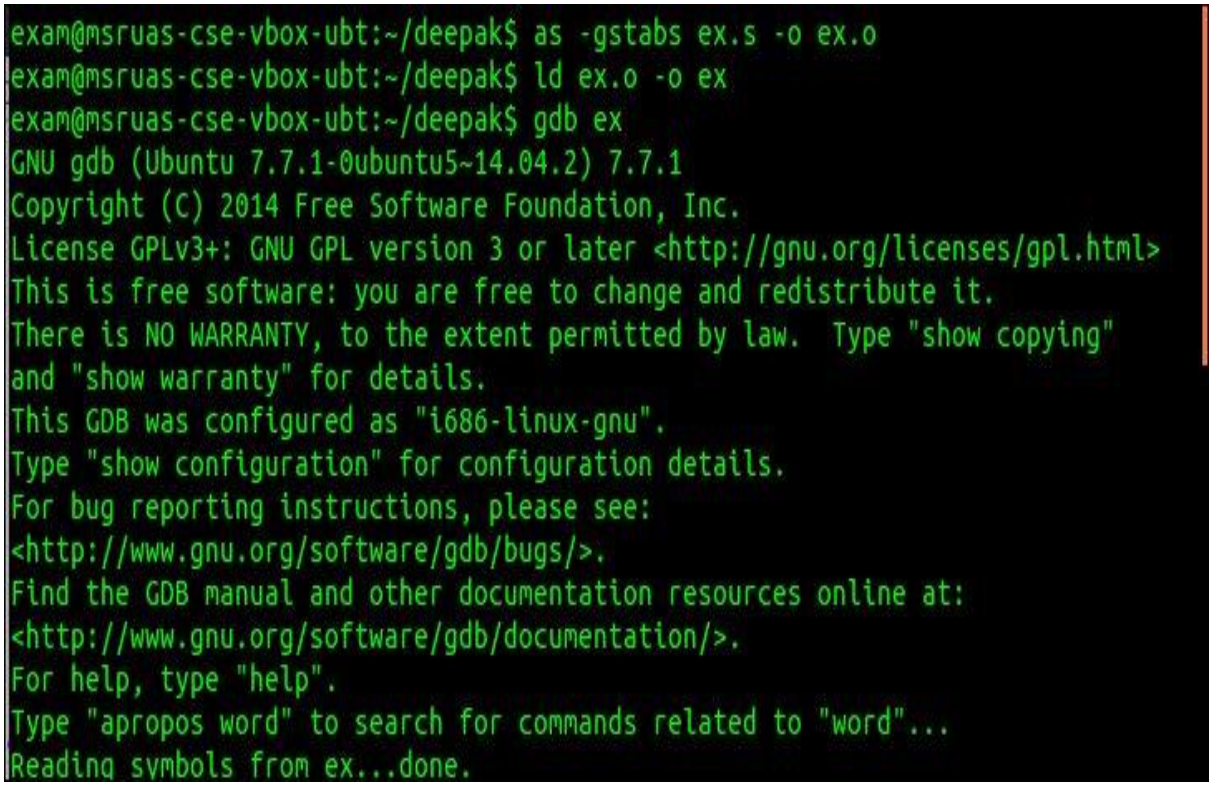


FIG2 SHOWING WRITTEN INPUT IN TERMINAL

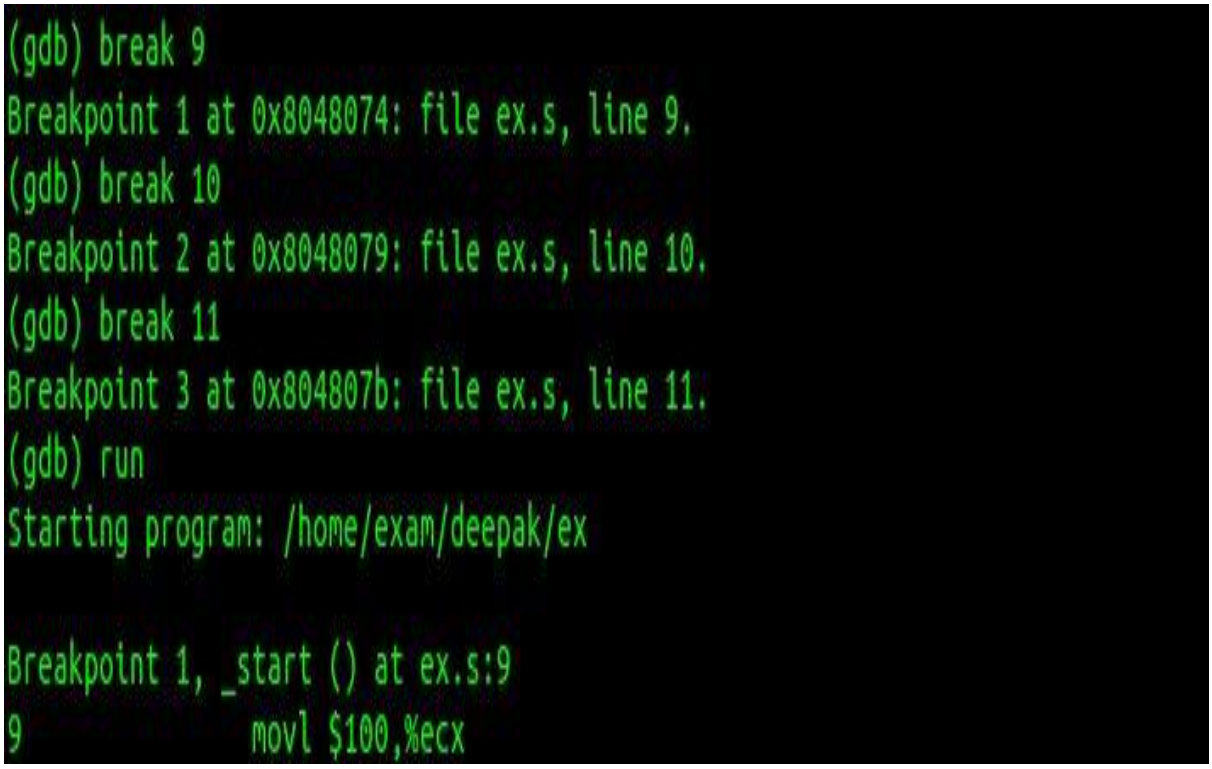


FIG 3 SHOWING GIVEN COMMANDS TO BREAKPOINT EACH LINE

6. PRESENTATION OF RESULTS

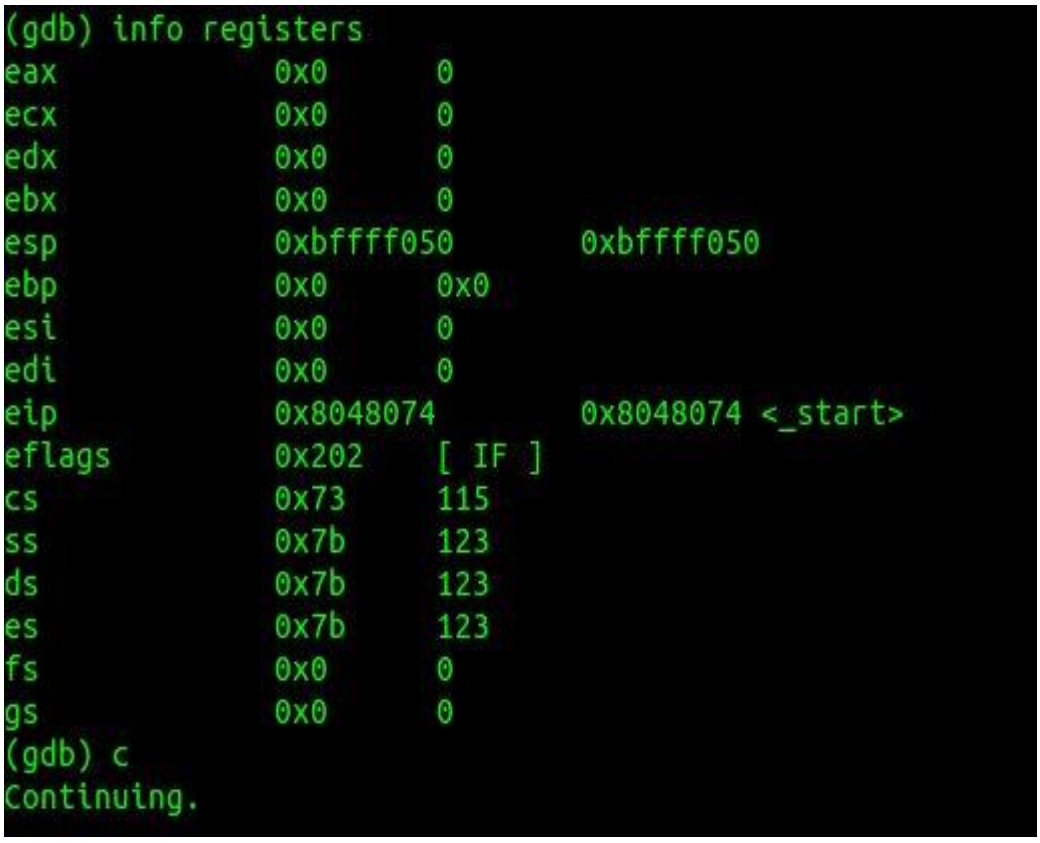


FIG 4 SHOWING INFO OF REGISTERS BEFORE DATA TRANSFER

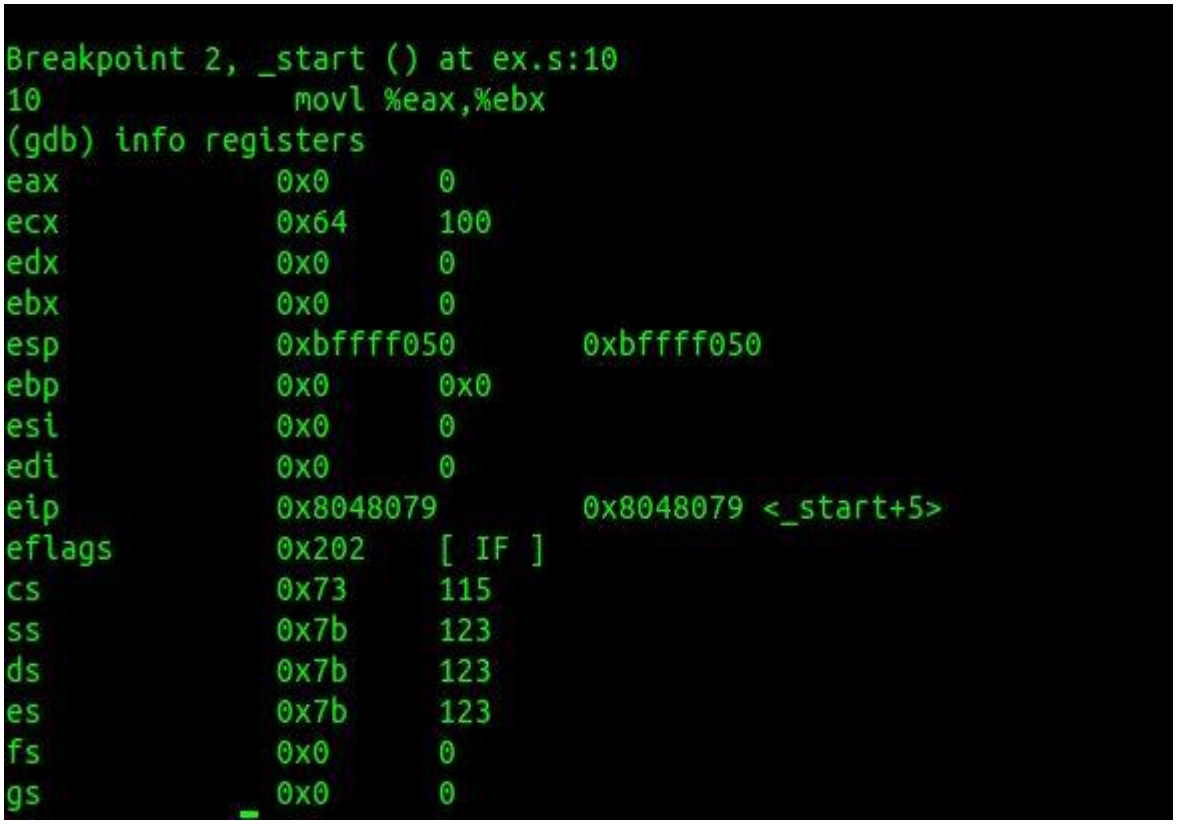


FIG 5 SHOWING INFO OF REGISTERS AFTER DATA TRANSFER

7. Analysis and Discussions

Assembly language is a low level language and dependent on the system architecture but in case of C, it can be ported to various platforms

8. Conclusions

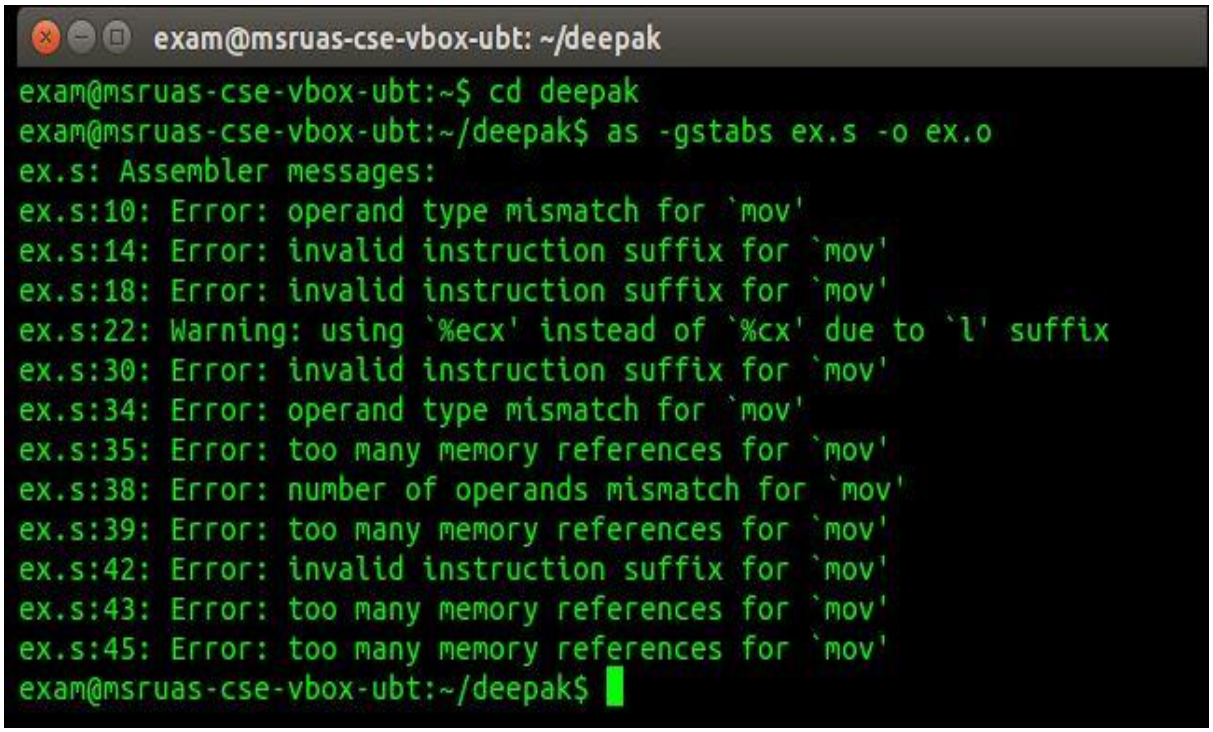
The assembly code for one architecture can not port to another architecture with out radical rewrite of the code

9. Comments

1. Limitations of Experiments

The certain programs cannot be performed using above method certain example are shown in the below picture .

2. Limitations of Results



Name: **DEEPAK R** Registration Number: **18ETCS002041**

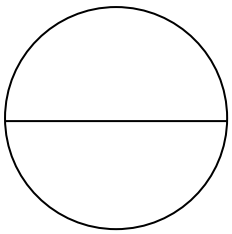
3. Learning happened

To develop assembly language program to perform data transfer operations on different data. Performing data transfer operation using Immediate, Register and Direct addressing modes

4. Recommendations

Assembly language is a human readable translation of the machine code. There is a direct one to one translation from assembly instruction to processor instruction. Assembly language is used when you need to guarantee certain things happen in a certain order, or a certain number of cycles.Assembly programming is used a lot when writing operating systems or drivers where there is close interaction with hardware needed. Often it is the only way to access specialised instruction that a processor may provide.So it can be used in many language compilers and interpreters.

Signature and date **Marks**



****

Name: **DEEPAK R** Registration Number: **18ETCS002041**