Green University of Bangladesh
Department of Computer Science and Engineering (CSE)

Faculty of Sciences and Engineering

Semester: (Spring, Year:2024), B.Sc. in CSE (Day)

Lab Report NO: 01

Course Title:

Microprocessors and Microcontrollers

Course Code: CSE 304 Section: 231_D4

Lab Experiment Name: Introduction to the assembly program structure and various arithmetic operations.

Student Details

	Name	ID
1.	Promod Chandra Das	231002005

Lab Date: 21.02.2025

Submission Date: 24 .02.2025

Course Teacher's Name: Md.Romzan Alom

Lab Report Status

Marks: ……

Comments:.....

Signature:.....

Date:.....

➢ TITLE OF THE LAB REPORT EXPERIMENT

Introduction to assembly program structure and various arithmetic operations.

➢ OBJECTIVES/AIM

The primary objective of this lab is to introduce the basic structure of an assembly language program and to demonstrate the implementation of various arithmetic operations such as addition, subtraction, multiplication, and division. By the end of this lab, students will be able to:

1. Understand the structure of an assembly program.
2. Write and execute simple assembly programs.
3. Perform basic arithmetic operations using assembly language.

➢ Introduction

Assembly language is a low-level programming language that is closely related to machine code. It provides a direct way to interact with the hardware of a computer. Each assembly language is specific to a particular computer architecture. In this lab, we will focus on the x86 architecture. An assembly program typically consists of the following sections:

1. Data Section: Used to declare variables and constants.
2. Code Section: Contains the actual instructions to be executed.
3. Stack Section: Used for managing function calls and local variables.

Arithmetic operations in assembly language are performed using specific instructions such as ADD, SUB, MUL, and DIV**.**

➢ Materials and Tools

1.Assembler (e.g., NASM, MASM)
2. Text Editor (e.g., Notepad++, VS Code)
3. Debugger (e.g., GDB, OllyDbg)
4. Operating System (e.g., Windows, Linux)

➢ IMPLEMENTATION

Problem: 01 & Problem: 02 & Problem 03

1. Take a double-digit number input from the user.

2. Convert 260°C to Fahrenheit using the following expression and store in a F variable: °F = °C×9/5 + 32 - 1

3. Convert 1000 °F (Fahrenheit) to °C (Celsius) using the following expression and store in a C variable: °C = (°F - 32)× 5/9 + 1

Answer To The Q No: 01 & 02 & 03

```
    section .data
  prompt db "Enter a double-digit number: ", 0
  len_prompt equ $ - prompt

  celsius dd 260      ; 260°C
  fahrenheit dd 1000  ; 1000°F

  F dd 0              ; Variable to store °F result
  C dd 0              ; Variable to store °C result

  buffer db 3         ; Buffer to store user input
  len_buffer equ 3

section .bss
  num resb 2          ; Reserve space for user input

section .text
  global _start

_start:
  ; Display prompt to user
  mov eax, 4          ; sys_write system call
  mov ebx, 1          ; file descriptor (stdout)
  mov ecx, prompt     ; message to write
  mov edx, len_prompt ; message length
  int 0x80            ; call kernel

  ; Read user input
  mov eax, 3          ; sys_read system call
  mov ebx, 0          ; file descriptor (stdin)
  mov ecx, num        ; buffer to store input
  mov edx, 2          ; number of bytes to read
  int 0x80            ; call kernel
```

```
; Convert 260°C to Fahrenheit
mov eax, [celsius]   ; Load 260 into eax
mov ebx, 9           ; Load 9 into ebx
imul eax, ebx        ; Multiply eax by 9 (260 * 9)
mov ebx, 5           ; Load 5 into ebx
idiv ebx             ; Divide eax by 5 ((260 * 9) / 5)
add eax, 32          ; Add 32 to eax
mov [F], eax         ; Store result in F

; Convert 1000°F to Celsius
mov eax, [fahrenheit] ; Load 1000 into eax
sub eax, 32          ; Subtract 32 from eax (1000 - 32)
mov ebx, 5           ; Load 5 into ebx
imul eax, ebx        ; Multiply eax by 5 ((1000 - 32) * 5)
mov ebx, 9           ; Load 9 into ebx
idiv ebx             ; Divide eax by 9 (((1000 - 32) * 5) / 9)
add eax, 1           ; Add 1 to eax
mov [C], eax         ; Store result in C

; Exit the program
mov eax, 1           ; sys_exit system call
xor ebx, ebx         ; exit code 0
int 0x80             ; call kernel
```

- **Results**

1. User Input: The program successfully accepts a double-digit number from the user.
2. Temperature Conversions:
    - 260°C to Fahrenheit:
      °F = 260 × 9/5 + 32 = 500
      The result is stored in the variable F.
    - 1000°F to Celsius:
      °C = (1000 - 32) × 5/9 + 1 ≈ 538
      The result is stored in the variable C.

➢ ANALYSIS AND DISCUSSION

This lab demonstrated how to perform arithmetic operations in assembly language, including multiplication, division, addition, and subtraction. The program successfully converted temperatures between Celsius and Fahrenheit using the given formulas. Handling user input in assembly requires careful management of system calls and buffers.

Through this lab, we learned how to:

- ✓ Accept user input in assembly language.
- ✓ Perform arithmetic operations to convert temperatures.
- ✓ Store results in variables for further use.

This exercise provided a practical understanding of assembly programming and its applications in performing calculations.