

LAB 4: Introduction to MASM and File-Based Assembly Programming

Course: Microprocessor and Interfacing

Lab Number: 4

Duration: 2 Hours

Tools: MASM, DOSBox, Notepad++

Objective: Transition from interactive DOSBox coding to file-based MASM programming.

Overview

In previous labs, you worked directly inside DOSBox and implemented basic instructions, loops, and string operations. In this lab, you will shift to a proper MASM workflow by:

- Writing `.asm` files using Notepad++
- Assembling using MASM
- Linking object files
- Executing generated `.exe` programs

General Instructions

1. All programs must follow the `.model small` format.
2. Initialize the data segment explicitly.
3. Use DOS interrupt `INT 21h` for input/output.
4. Do not use macros or external libraries.
5. Assume all inputs are valid unless stated otherwise.

Compilation Steps

1. Write your program in Notepad++ and save as required.
2. Place the file inside MASM BIN directory.
3. Assemble:

```
masm Qx.asm
```

4. Link:

```
link Qx.obj
```

5. Run:

```
Qx.exe
```

Question 1: Digit Frequency Counter

Write a MASM program that:

- a) Accepts a string of digits (0–9) from the user (maximum length: 20).
- b) Counts how many times each digit appears.
- c) Prints the frequency of every digit from 0 to 9 in ascending order.

Constraints

- Input consists only of numeric characters.
- Use buffered input (INT 21h, AH=0Ah).
- Store counts in an array of size 10.
- Convert ASCII to numeric values manually.
- Output must be space-separated.

Sample

Input:

1123401

Output:

1 3 1 1 1 0 0 0 0 0

Input:

01234567899876543210

Output:

2 2 2 2 2 2 2 2 2 2

Question 2: Running Mean (Streaming Input)

Write a MASM program that continuously:

- a) Accepts single-digit numeric inputs from the user.
- b) Updates the running mean after each new digit.
- c) Prints the current mean.
- d) Continues until a specific termination character (X) is entered.

After termination, the program exits.

Constraints

- Input must be taken character-by-character.
- Maintain cumulative sum and count.
- Mean must be computed using integer division.
- Only print the first digit of the mean found
- Program must wait for the next digit after displaying each mean.
- Exit only when X is encountered.

Sample Interaction

Input:

2
3
4
X

Output:

2
2
3

Submission

- Rename files as:
 - Q1_<BITSID>.asm
 - Q2_<BITSID>.asm
- Upload files on Quanta.