

**COMPUTER ENGINEERING DEPARTMENT
TERM 052**

COE 205

Computer Organization & Assembly Language Programming

Lab Project

Due in the last lab 4 Weeks

I. Introduction

The objective of this project is to see how much knowledge in assembly language programming the student has gained during the whole term. However, the main objectives intended by this project are the following.

1. Review of most assembly language instructions,
2. Structured programming using procedures & Macros,
3. File I/O,
4. Use of interrupt service routines (ISR)
5. Use of graphics and mouse handling instructions.

II. References

1. Textbook, lecture notes, lab manual, and online assembly book.
2. *Assembly Language Programming & Organization of the IBM PC*, Ytha Yu and Charles Marut, McGraw Hill, 1992.
3. Randall Hyde, "The Art of Assembly Language Programming",

<http://webster.cs.ucr.edu/AoA/DOS/AoADosIndex.html>

4. A very good list of interrupts, <http://www.ctyme.com/intr/int.htm>

III. Project Description

This project is about developing a program for processing text files. The program has the following main parts:

1. Developing a TSR for handling Function keys
2. Designing a GUI for displaying Histograms

The main menu should be interactive and easy to use. The use of the mouse as a pointing device is appreciable and recommended.

The following options should be present on the main menu (atleast):

1. Enter the name of the file for processing
2. Exit

Task 1:

You are required to use INT 61h for handling function keys. The ISR should perform the following operations on pressing the function keys.

F1: View the contents of the text file

F2: Count the number of words in the file

F3: Display a histogram showing the number of occurrences of each alphabet (developed in Task 2)

Task 2:

You have to design a graphical histogram showing the frequency with which different alphabets are used in the file. The percentage should be mentioned alongwith the graph.

IV. Guidelines

Make your program well structured, use procedures and MACROs that you should put in a separate file and include at the beginning of your program. Use comments as much as you think this will make your program understandable, and separate your procedures by such comments. Make your program interactive, in the sense that the user can use it easily. The use of different colors for different options and different windows is appreciated.

At the end of your work submit a hard copy and a soft copy of your program, including both source and executable files on a floppy disc or flash memory. *A demo is necessary for your instructor to be able to assess your work.*

V. Hints

To be able to write such a program use a simple approach, and fix yourself very simple targets that you should first achieve. Start by making a simple ISR which just recognizes different function keys. Then write a program which can read text files and display them.

Build upon the program you have written, and keep all the modified versions of your program by giving them names that should be meaningful, like pgm1ver1.asm or p11.asm for program 1 version 1 Submit only the last version that contains the finalized work, and any attached files like the MACRO file or the procedure file.

VI. Guidelines

1. You can team up with only one of your friends.
2. Come up with a task breakdown in which you indicate the responsibilities of each one of you.
3. The work should be divided equally among team members. This division of work must be included in the final report.
4. Start with a text-based menu. Then, after you finish the core program, develop the GUI.
5. Each member of a team should be able to compile his work separately.

VII. Mark Distribution

The project accounts for 8 points out of 20 in the lab

Item	Weight	Criteria	Mark
Report	5	Content & Presentation	5
Program	10	Clarity & Presentation	2.5
		Use of Procedures	2.5
		Use of Macros / Include File	2.5
		Comments	2.5
Interactivity	10	Menu Presentation	5
		Error Handling	5
Program Working	65	Input	5
		Task 1	30
		Task 2	30
Teamwork	10	Task Breakdown	10
Bonus	20	Use of Mouse	10
		Integration	10
Overall Mark	120		120

Policy on Academic Dishonesty:

Almost all computer programs contain many ideas borrowed from elsewhere; however, copying code from other people or from books won't teach *you* how to program. To learn programming, you must learn to solve your own programming problems.

- You are *not allowed* to copy code from other students (or former students). Possession of source code belonging to another student is an academic offence, with penalties for everyone involved, both the author and the recipients.
- Do not share your disks or listings with other students.
- *Share your ideas, not your source code.*

Your instructor will be sympathetic to the students who try to accomplish the tasks, but in the end are not able to furnish a working program; however, **you will be given a ZERO in the project if any dishonesty is found on your part.**