

COE 205
Computer Organization & Assembly Language Programming

Lab Project

Due on 25 May 2004 5 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. Understand the structure of the floating-point unit and how it works,
5. Use of floating-point instructions, and
6. Use of graphics and mouse handling instructions.

II. References

1. Textbooks, lecture notes, lab manual, and handouts.
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/Page_asm/ArtofAssembly/ArtofAsm.htm

Any useful information regarding the project will be put on the following link:
http://faculty.kfupm.edu.sa/COE/yosais/coe_205.htm

III. Project Description

This project is about developing a program for processing text files that contain names of students, their ID numbers, and their marks in three exams. Marks are floating-point numbers. The program has two main parts:

1. Graphical-User Interface (GUI), and
2. Code for handling user inputs and processing files.

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

When your program is run, it displays a menu with the following options:

1. Find a student using his last name
2. Find a student using his ID number
3. Display the marks of a student
4. Add information of a new student
5. Display information of all students
6. Save information to a file
7. Display the distribution of marks for each major exam
8. Display the average for each exam
9. Exit

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. *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 reading a file and displaying the contents of it. Make the file contain simple values (byte values only: 0 .. 255). Once done, try to read a floating-point number and display it. Then, write a program that reads two floating-point numbers, adds them, and displays the result on the screen.

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.

VII. 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. Start with the core program. You need first to understand floating-point arithmetic.
4. Start with a text-based menu. Then, after you finish the core program, develop the GUI.

VIII. Mark Distribution

The project counts for 50% of the lab; i.e., 10% of the course. It is marked as follows.

Item	Weight	Criteria	Mark
Report	5	Content & Presentation	5
Program	15	Clarity & Presentation	3
		Use of Procedures	5
		Use of Macros / Include File	5
		Comments	2
Interactivity	30	Menu Presentation	5
		Graphics	25
Program Working	50	Input	5
		Output (Display)	5
		All Options Implemented	40
Bonus	50	Use of Mouse	25
		TSR / Interrupts	15
		More options / Functions	10
Overall Mark	150	Total Mark (student gets)	100

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.**