Computer Engineering Department

COE 205

Computer Organization & Assembly Language Programming

Lab Project (Term-043)

Due on:

Sec. 51 Sat. 2:10PM 20th August 2005

Sec. 52 Sun. 2:10PM 21st August 2005

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 graphics and mouse handling instructions.

II. References

1. Textbooks, lecture notes, online material, 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”,

Any useful information regarding the project will be put on the following link:
http://www.ccse.kfupm.edu.sa/~masud/text/annou205.htm

III. Project Description

This project is about developing a program for performing DOS utilities. 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 runs, it should display a menu with the following options:

1. Search for a file in the current directory
2. Create a directory
3. Remove a directory
4. Copy a source file to destination
5. Read and display the contents of a text file
6. Append some text to a file
7. Delete a file
8. Search for a word in a given file
9. Count number of words in a file
10. 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 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 simple approach, and fix yourself for very simple targets that you should achieve first. Start by reading a file and displaying the contents of it. Make the file contain simple values (byte values only: 0 ... 255).

Build upon the program you have written, and keep all the modified versions of your program by giving names that should be meaningful, like pgm1ver1.asm or p11.asm for program 1 version 1 or p12aug3pm.asm …. Submit only the last working 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 the basics of file handling.
4. Start with a text-based menu. Then, after you finish the core program, develop the GUI.

VIII. Marks Distribution

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

Item	Criteria	Marks	Weight
Report	Content & Presentation	5	5
Program	Clarity & Use of Comments	5	15
	Use of Procedures	5	
	Use of Macros / Include File	5	
Interactivity	Menu Presentation	5	25
	Graphics	20	
Program Working	Input /Output	5	55
	Error Handling	5	
	All Options Implemented	45	
Bonus	Use of Mouse	20	30
	Interrupts	10	
Overall Mark	Total Marks	100	130

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 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.**