

**Lab Project****Design of a PC Test-Bench****Introduction:**

The primary objective of this project is to review most of the assembly language tools learned during the whole term. In this project you are required to write a complete program to test the different components of your PC.

- 1- BIOS access through software interrupts.
- 2- Structured programming using procedures, Macros and Interrupts.
- 3- Use of TSR programs
- 4- Use of the Mouse and the Video functions

**Project Work:**

The student is required to search for ways to find characteristics of the hardware components of the PC. This can be done through the web or the library. Any useful information will be put on the following link: <http://www.ccse.kfupm.edu.sa/~kamel/COE205/Project/Project.htm> or on the WebCT.

In this project, you are required to write a program that identifies the main hardware components of your PC by reading the BIOS and the different peripheral devices attached to your PC. The program should generate a complete report that has to be saved in a file as text.

The program starts by displaying a menu that contains the following options:

1. Identify Hardware components
2. Processor
3. Hard drive
4. Memory
5. Mouse
6. Serial Port
7. Parallel Port
8. Exit.

Design and presentation of the menu are left for the programmer. Such menu should be interactive and easy to use. The use of a graphics menu and the mouse as a pointing and interfacing device is appreciable. The last Lab experiments related to interrupts, video and mouse can be of some help to you. The choices in the menu should more or less be similar the following:

**Choice 1:**

The program displays a list of all components attached to your PC; such as processor, memory, mouse, graphics card, ..etc.

**Choice 2:**

The program identifies the type of processor in your PC.

Eg: Processor: PIII  
 Manufacturer: Intel  
 Speed: 500 MHz  
 Cache: 256 MB  
 ...

**Choice 3:**

The program identifies the type of hard drive in your PC.

Hard drive type:  
 Manufacturer:  
 Size:

**Choice 4:**

The program identifies the memory size of your PC.

**Choice 5:**

The program identifies the type of mouse attached to your PC.

Mouse attached: Y/N  
 Mouse Type: 2 or 3 buttons  
 Manufacturer: If Possible

**Choice 6:**

The program gives you the number and hardware addresses of serial ports in your PC.

Number of Serial Ports: 2  
 Addresses: com1: XXX com2: YYY

**Choice 7:**

The program gives you the number and hardware addresses of parallel ports in your PC.

Number Parallel Port: 2  
 Addresses: lpt1: 278H, lpt2: 27AH

**Choice 8:**

Let the program write all the information about your PC in a file as a complete report and finally saves it.

**Choice 9:**

The program simply exits to DOS.

**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.

To be able to write such a program start using a simple approach, and fix yourself very simple targets that you should achieve first. Then 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.

**Bonus:**

Extra marks are given for a program that tests the memory access time. This time is estimated by repeatedly writing into a block of memory then reading it. The read/write time is related to the amount of time, the size of the block to be read/written number and the number of times the operation is repeated. To have an estimate of the time the programmer should use the internal PC timer.

Another bonus is given if the whole program is made as a TSR and activated by a hot key sequence. For any new device tested (scanner, USB ports ...) extra marks are given.

**Use of a High Level Language:**

You can use any high level language for the design of the interface (GUI). However, most of the program routines and functions have to be written in assembly.

**References:**

1. Textbook, lecture notes and lab manual.
2. **Randall Hyde, "The Art of Assembly Language Programming",**  
[http://webster.cs.ucr.edu/Page\\_asm/ArtofAssembly/ArtofAsm.html](http://webster.cs.ucr.edu/Page_asm/ArtofAssembly/ArtofAsm.html)

**Mark Distribution:**

The project counts for **50% of the lab**, i.e. **10% of the course**, and is marked as follows:

	<b>Section</b>	<b>Mark</b>
<b>Program /15</b>	Clarity and Presentation	<b>2.5</b>
	Use of Procedures	<b>5</b>
	Use of Macros/Include File	<b>5</b>
	Comments	<b>2.5</b>
<b>Interactivity /20</b>	Menu	<b>5</b>
	Graphics	<b>5</b>
	Use of the Mouse	<b>5</b>
	Ease of Use	<b>5</b>
<b>Working Program /65</b>	Identify Hardware Components	<b>5</b>
	Processor	<b>10</b>
	Hard drive	<b>8</b>
	Memory	<b>8</b>
	Mouse	<b>8</b>
	Serial Ports	<b>8</b>
	Parallel Ports	<b>8</b>
	File Report Generation	<b>10</b>
<b>Bonus</b>	Interface to a HLL	<b>10</b>
	Program running as a TSR	
	Extra Devices	
<b>Overall Mark</b>		<b>100</b>

**Notes:**

1. If the program, does not work a partial mark will be given as explained in part 1 of the above table. However, if the program does not assemble and generates three or more errors a zero (0) will be assigned in the project.
2. Similar copies are automatically given 0.
3. The instructor will ask each student on a specific part of the program to check that the submitted work is effectively done by the student.
4. Each student has to submit a soft copy of the complete source and any macro or procedure files, an executable file and an example of a report file. The source file should assemble and generate the right executable code file.
5. Last day for students to submit the work is during the last week of classes before the finals: **25/12/04 to 29/12/04.**