LAB WORK NO. 10
USING SYSTEM FUNCTIONS IN ASSEMBLY LANGUAGE

1. Object of laboratory

Study of OS's functions (BIOS and DOS functions) and their use in assembly language. An example for using the terminal is presented.

2. Theoretical considerations

The OS is a collection of routines (procedures) useful for efficiently manage system's resources. These routines are divided in 2 categories: BIOS routines and DOS routines. For IBM-PC compatible systems a series of system routines are available for the user. They are written as procedures accessed through software interrupts.

BIOS functions are written to allow access for the user to system resources (hardware) DOS functions are written for OS's specific purposes. DOS functions facilitate usage of the file system, the user doesn't need to have a full knowledge of the hardware of the file system in order to create a file.

Access to BIOS and DOS functions from user programs is done through software interrupts (INT instruction).

The main groups of BIOS functions available for the user are:

INT 10h - use of video terminal
INT 11h - determine system's configuration
INT 12h - determine RAM's capacity
INT 13h - access to HDD and FDD
INT 14h - use of serial interface
INT 15h - APM
INT 16h - use of keyboard
INT 17h - use of parallel interface
INT 19h - loader of resident system on disk
INT 1Ah - real time clock controller(RTC)

During a subroutine call (by INT) other functions can be specified. The function is specified "by convention", placing it's number in AH

register. The call for a certain BIOS function is possible through a generic sequence:

```
MOV  AH, function_nr        ; specify function
INT  int_nr                 ; specify interruption
```

According to a function's complexity a series of parameters can be specified by following it's pattern.

DOS functions are mainly referring to files, but there is a large scale of functions. All DOS functions are called by INT 21h and by specifying the desired function (eventually the parameters) in AH register.

Some BIOS functions will be presented:

INT 10h

This function facilitates the use of video terminal. Inside INT 10h function there are many sub-functions which allow character posting and use of various graphic modes. For graphic modes with bigger resolutions this interrupt it's not recommended because it's slow; it is recommended to write directly into video memory. For example posting a character on screen: calling 10h interrupt implies a big amount of code to be executed (interpreting parameters transferred into registers, setting sub-function etc), while for writing directly into video memory only one MOV instruction is needed.

Still, this interrupt is very practical for programs that are not posting big amount of information on screen at certain moment. By using this interrupt the programmer doesn't have to calculate video memory addresses in which to write every character.

(AH)	Function	Input parameters	Output parameters
00h	Select functioning mode for graphic terminal	(AL) =0 alphanumeric 40 col. x25 lin. b/w =1 alphanumeric 40 col. x25 lin. Col =2 alphanumeric 80 col. x25lin. b/w =3 alphanumeric	

		80 col. x25lin. col. =4 graphic 320 colx200 lin col. =5 graphic 320 colx200 lin b/w =6 graphic 640 colx200 lin b/w	
01h	Select shape and size for cursor	(CH) 0-4 bites for cursor's start line (CH) bites 5-7=0 (CL) 0-4 bites for cursor's finish line (CL) bites 5-7=0	
02h	Cursor positioning	(DH, DL) lin., col. (0, 0) –left upper corner (BH) – page nr. =0 for graphic mode	
03h	Read cursor's coordinates	(BH) – page nr =0 for graphic mode	(DH, DL) lin. , col. (CH, CL) shape
04h	Read position for optic indicator		(AH) =0 light pen inactive =1 light pen active (DH, DL) lin., col. Cursor (CH) line pixel (0-199) (BX) col pixel (0-319636)
05h	Select active display pages	(AL) – page nr. 0-7 for mode 0 and 1 0-3 for mode 2 and 3	
06h	Execute operation "scroll up"	(AL) – nr. of lines (AL) =0 delete	

		window (CH, CL) – lin., col. for left upper corner (DH, DL) – lin., col. for right down corner (BH) – the attribute of a white line	
07h	Execute operation "scroll down"	(AL) – nr. of lines (AL) =0 delete window (CH, CL) – lin. , col. for left upper corner (DH, DL) – lin. , col. for right down corner (BH) – the attribute of a white line	
08h	Read character from screen and determine it's attribute (the character is read from cursor's current position).	(BH) – referred number of page (only for alphanumeric mode)	(AL) – read character (AH) – character's attribute
09h	Show character on screen.	(BH) – referred number of page (only for alphanumeric mode) (BL) –character's attribute (for alphanumeric mode) - color (for graphic mode) (CX) – nr. of	

		characters to show (AL) – character	
0Ah	Replace characters on screen maintaining color characteristics	(BH) –the number for reference page (CX) – nr. of characters to show (AL) (AL) – character	
0Bh	Set color characteristics(only for 320x200 pixels graphic mode)	(BH) – palette's index (conf doc IBM-PC) (BL)-value of color within palette	
0Ch	Show point on screen (for graphic mode)	(DX) – line's number (CX) – column's number (AL) –color's value (0, 1, 2, 3)	
0Dh	Read point's color (for graphic mode)	(DX) – line's number (CX) – column's number	(AL) – read color
0Eh	Show character on screen and update cursor's position	(AL) – character to show (BL) –background color (for graphic mode) (BH) – page (for alphanumeric mode)	
0Fh	Read characteristics for current mode		(AL) – current module (AH) – character's number of columns (BH) – page number

INT 13h

Direct access to HDD and FDD is possible through this interrupt. INT 13 allows writing and reading disk sectors directly without considering the existent system of files on disk. That's why this function is not recommended when working with files, for these operations DOS functions are more appropriate. But there are situations when these functions are the only solution: reading a disk with a different file system, other than FAT.

INT 14h

This function facilitates access to system's serial interface. There are 4 available functions:

AH	the function
00h	initialize port (4 ports are supported)
01h	send one character
02h	read one character
03h	check port's status

INT 16h

This function is used both for reading characters from keyboard and for obtaining keyboard's current state (Caps Lock, Ctrl, Shift etc.) . Function 00h (in AH) is used to read a character from the keyboard, the character will be returned in AL.

INT 19h

After POST the processor executes the code for this interruption by trying to read a code named bootstrap from the floppy or from the hard disk. So, this interruption loads to memory at address 0000:7C00h the first sector from floppy or hard disk and makes a JMP to this address in this way the bootstrap gets the control. The execution of this code determines which partition is active and then loads into memory and executes the boot sector from this partition. Because of this we can have multiple operating systems on a single hard disk and we can choose one of them at start up.

INT 1Ah

This function allows access to system's clock, for reading and setting the time.

AH function
00h read time
01h set time

The hour is represented as units; one unit has 55ms. When the system is on the number of units increases every 55ms.

For AT class computers through this interruption BIOS gives access to system's RTC. The RTC runs even when the computer is off because of the battery on mainboard It uses a memory segment from CMOS to store the time. This memory segment is rewritten every 55ms without using the microprocessor.

DOS functions are mainly referring to files but there is a large variety of functions. Every DOS function is called with INT 21h and by specifying the desired function (and other parameters) in AH register.

DOS – INT21h functions, for keyboard and monitor.

(AH)	Function	Input parameters	Output parameters
00h	End program's execution		
01h	Read character from keyboard and send it in echo to screen. If CTRL-BREAK are pressed INT 23h executes		(AL) –inserted character
02h	Show character on screen. If CTRL-BREAK are pressed INT 23h executes	(DL) –the character	
05h	Print character	(DL) – the character	
06h	Direct read/write - from keyboard - to screen	(DL) = 0FFh (DL) – the character	(AL) – the character (AL) = 0 (no character)
07h	Read character from keyboard without echo and without interpretation		(AL) – the character
08h	Read character from keyboard without		(AL) – the

	echo If CTRL-BREAK are pressed INT 23h executes		character
09h	Show a row from memory ending with $ (24h)	(DS:DX) – row address	
0Ah	Read from keyboard and place into a memory buffer a row of characters, until <CR> AKA ENTER is pressed Buffer structure see below	(DS:DX) – buffer's address	
0Bh	Determines keyboard's situation. If CTRL-BREAK are pressed INT 23h executes		(AL) =Off one character is available (AL) =0 no character
0Ch	Initialize keyboard's for buffer and then call a function. The system waits for a character.	(AL) – requested function (01h, 06h, 07h, 08h, 0Ah) .	

Read_buff db max_char,?, max_char DUP (?)

Where max_char is the maximum number of characters that are read including enter,
On return the function sets the second byte to the effective number of characters read, the following bytes contain the characters read including enter.

Observations and recommendations for using these functions:

1. BIOS functions save registers CS, SS, DS, ES, BX, CX, DX, and destroy the others, so the user must save them and restore them.
2. Before modifying the monitor's functioning regime it is recommended to save the current attribute and reset it at the end.
3. To erase the screen (window) it is recommended the use of function 06h with 0 displacement (in AL) and not 25!
4. For 0Eh function (write in teletype mode) the attribute is "inherited" from the previous character. But this is a slow process. It can be accomplished much faster by MOVS repetition!
5. DOS functions offer less facilities than BIOS functions when working with the screen.

3. Lab tasks:

1. Study DOS and BIOS functions for using the screen.
2. Read two 3 digit decimal numbers and display the sum on the screen
   Use loop and use DOS function 01 or no loop and DOS function 0AH
3. Write a program which draws on the graphic screen a sequence of squares of constant dimensions but with different colors and positioning.
4. Further documentation here http://ftp.utcluj.ro/pub/users/cemil/ng/