DOS Function Calls

01H	Read Keyboard with Echo
Entry	AH = 01h
Exit	AL = Character read from standard input device
Description:	This function will read a single character from the standard input device. If no character is waiting to be read, it will wait until a character becomes available.

09H	Display A Character String
Entry	AH = 09h DS:DX = Address of the string to point
Exit	None
Description:	This function will write a character string to the standard output device. On entry, DS:DX contains the address of the string. The string must be terminated by a dollar character '$'. All characters up to, but not including, the terminating dollar character will be written to the standard output device.

Experiment Name / No.: 2. INTRODUCTION TO MASM ASSEMBLER

Camlin / Page No. 9
Date 23 / 01 / 2020

Aim

To familiarize with the MASM assembler as well as various assembler directives.

Theory

The Microsoft macro assembler is an x86 high level assembler for DOS & Microsoft Windows. It supports wide varieties of macro facilities & structured programming idioms including high level functions for looping & procedures. A program called assembler is used to convert the mnemonics of instructions along with the data into the equivalent object code modules. This type of programming is called as ASSEMBLY LANGUAGE PROGRAMMING. The assembler converts an Assembly language source file to machine code.

An assembler like Microsoft Macro Assembler (MASM) provides a large number of features for assembly language programmers. Microsoft MASM version 6.11 contains updated software capable of processing pointing instructions. To assemble the file PROG1.ASM use this command.

```
MASM PROG.ASM
```

The MASM program will assemble the PROG.ASM file. To create PROG.EXE from PROG.OBJ, use this LINK command:

```
LINK PROG.OBJ
```

It converts the contents of PROG1.OBJ into PROG.EXE.

The following is a list of MASM reserved words:

1. ASSUME - assume definition
2. CODE - begin code segment
3. DATA - begin data segment
4. DB - define byte

Teacher's Signature:

02H	Write to Standard Output Device
Entry	AH = 02h DL = Characters to write to standard output device
Exit	None
Description:	This function will write the specified characters on the standard output device.

4CH	Terminate Program
Entry	AH = 4Ch AL = Program status code
Exit	Does not return
Description:	This function is used to terminate execution of a program. The value passed in AL is a status code that will be saved by DOS & can be queried by the parent program using function 4Dh.

5. DD - define double word
6. DQ - define quad word
7. DS - define storage
8. DUP - duplicate
9. DW - define word
10. END - end program
11. ENDM - end macro
12. ENDP - end procedure
13. ENDS - end segment
14. EQU - equate
15. FAR - far reference
16. MACRO - define macro
17. NEAR - near reference
18. OFFSET - offset
19. ORG - origin
20. PROC - define procedure
21. PUBLIC - public reference
22. SEGMENT - define segment

Assembler Directives: The limits are given to the assembler using some pre-defined alphabetical strings called Assembler Directives which help assembler to correctly understand the assembly language programs to prepare the codes.

1. DB - Define Byte: The directive DB defines a byte type variable.
2. DW - Define Word: The directive DW defines a word type variable.
3. DQ - Define Quad Word (4 words): This directive defines a quad word (4 words) type variable.
4. DT - Define Ten Bytes: The directive DT defines ten bytes variable.
5. ASSUME: This directive tells the assembler the name of a logical

Teacher's Signature:

segment, which is to be used for a specified segment.

6. END: The END directive marks the end of an ALP.
7. ENDP: End of the procedure.
8. ENDS: End of the segment.
9. EQU: This directive is used to assign name to some value.
10. OFFSET: Specifies offset address.
11. SEGMENT: The directive SEGMENT indicates the beginning of a logical segment.

Result

Familiarized with the MASM assembler as well as various assembler directives.

Teacher's Signature: