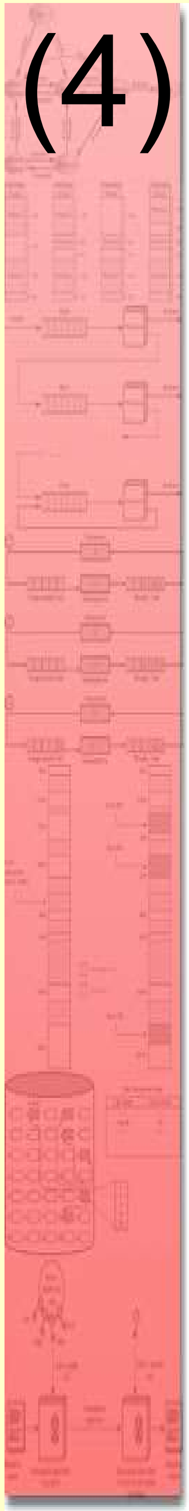
(4)



Escuela Politécnica Superior

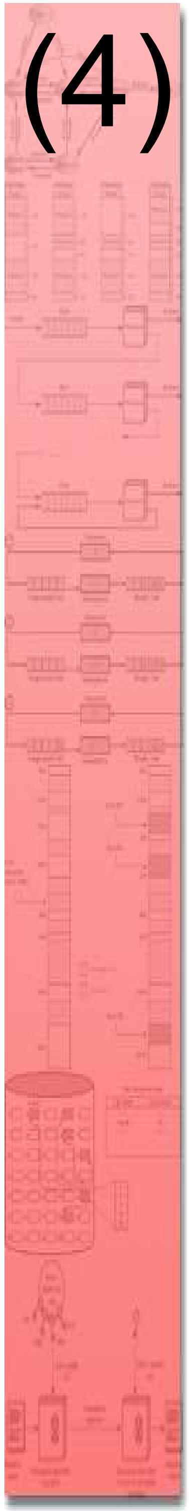
Universidad Autónoma
de Madrid

Unit 4
Programming Resources

MICROPROCESSOR-BASED SYSTEMS

Degree in Computer Science Engineering
Double Degree in Computer Engineering and Mathematics

EPS - UAM

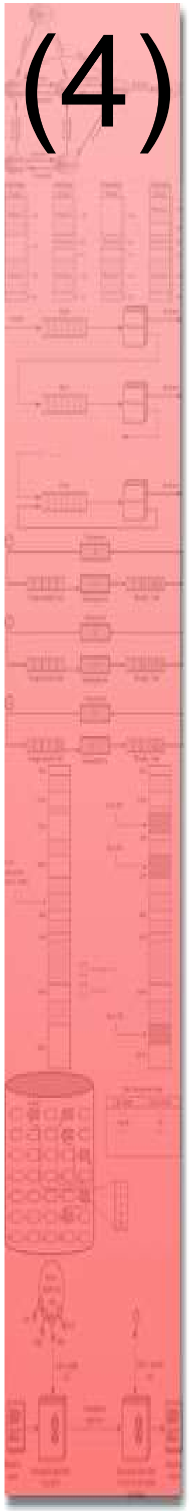


(4)

Index

4. Programming resources.

- 4.1. BIOS interrupts.
- 4.2. DOS interrupts.
- 4.3. Program execution under DOS.
- 4.4. PSP (Program Segment Prefix).
- 4.5. Types of programs: EXE, COM and resident (TSR).

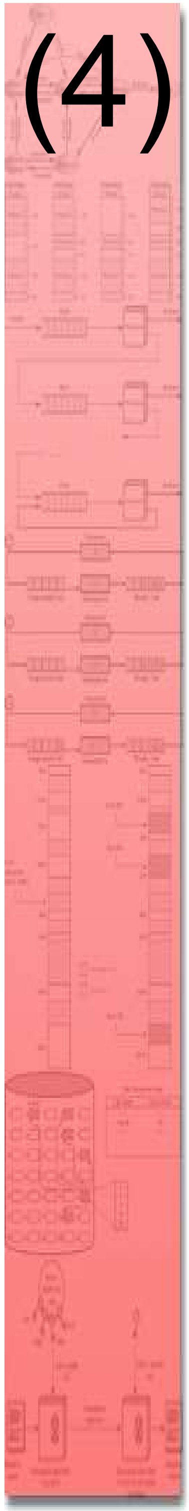


(4)

4.1. BIOS interrupts (I)

- **BIOS** (*Basic Input/Output System*): Basic *firmware* installed in the motherboard.
- It provides basic routines for accessing the hardware.
- Five broad groups:
  - Interrupts associated with the CPU (INT 0 to INT 7)
  - Interrupts associated with the 8259 interrupt controller (INT 8 to INT 0Fh)
  - BIOS services (INT 10h to INT 1Ah and INT 40h)
  - User routines (INT 1Bh and INT 1Ch)
  - Pointers to data tables (INT 1Dh to INT 1Fh and INT 41h)
- Ralf Brown’s interrupt list:

http://www.ctyme.com/rbrown.htm



4.1. BIOS interrupts (II)

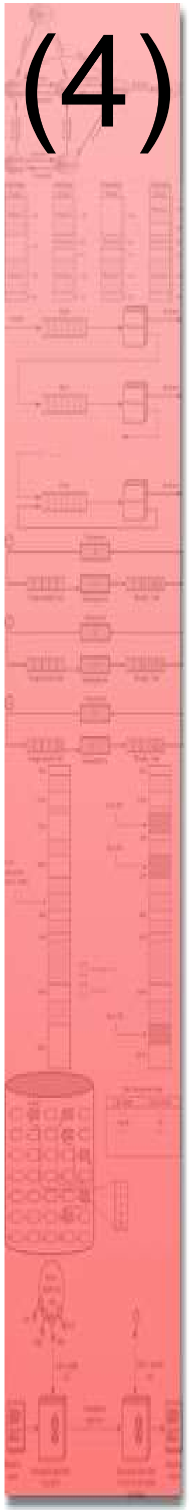
Associated with the CPU

- INT 0: Division by zero
  - Generated by the CPU when the quotient of a division (**DIV** or **IDIV**) is too big to be stored in **AL** or **AX.**
  - It prints ***"Divide overflow"*** on the console and returns to DOS.
- INT 1: Step-by-step execution
  - Activated when the trace flag (**TF**) is **1** and the CPU has executed any instruction.
  - DOS initializes the interrupt vector with an address that contains instruction **IRET**.
  - Debuggers (**DEBUG**, **SYMDEB**, **TD**, …) change the interrupt vector to a service routine that allows the step-by-step execution of programs.

4.1. BIOS interrupts (III)

Associated with the CPU

- INT 2: Non-maskable
  - Activated with a rising edge in the NMI pin of the CPU. The pin is connected to the RAM’s parity detector.
  - It prints ***"Parity Check 1"*** on the console and halts the CPU.
- INT 3: Breakpoint
  - Activated when an instruction with code CCh is executed.
  - Used by debuggers: It allows the execution of a program until that instruction is reached.
  - DOS initializes the interrupt vector with an address that contains instruction **IRET**.

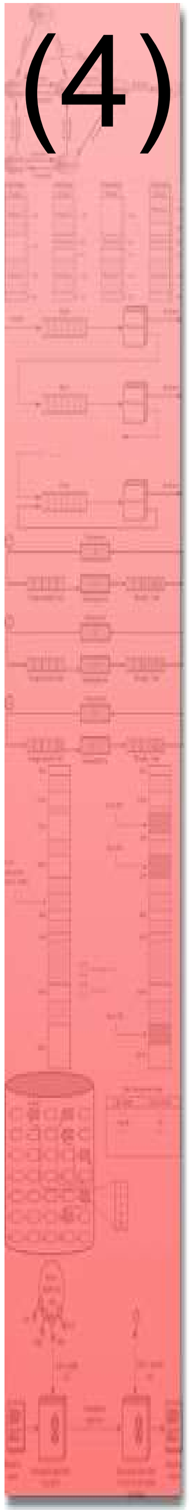


(4)

4.1. BIOS interrupts (IV)

Associated with the CPU

- INT 4: Overflow
  - Activated through instruction **INTO**.
  - It generates an INT 4 provided flag **O**=1.
  - DOS initializes the interrupt vector with an address that contains instruction **IRET**.
- INT 5: Print screen
  - Print the text shown on the screen.
  - It can be activated by pressing key ***Print Scrn***.
- INT 6, INT 7 (Not used)



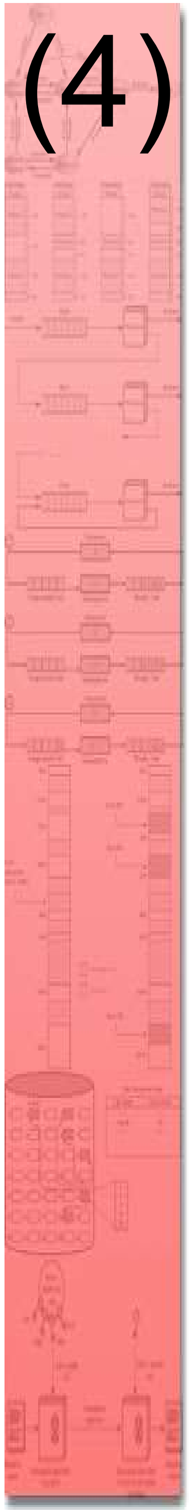
(4)

4.1. BIOS interrupts (V)

Associated with the interrupt controller

- Interrupts 8 to 15 (0Fh) are associated with the hardware interrupt controller (8259A). They are activated with an edge in its inputs IRQ0 to IRQ7.
- INT 8: Timer
  - The system timer (8253) activates this interrupt 18.2 times per second (every 55 ms).
  - The service routine increments by one the 32-bit counter located in the following BIOS addresses (and resets it every 24 hours):

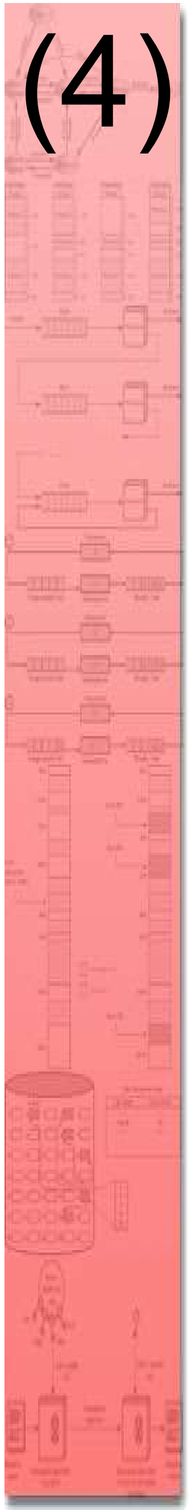
    0040h:006Ch (low word)
    0040h:006Eh (high word)
  - The service routine also executes **INT 1Ch**.



4.1. BIOS interrupts (VI)

Associated with the interrupt controller

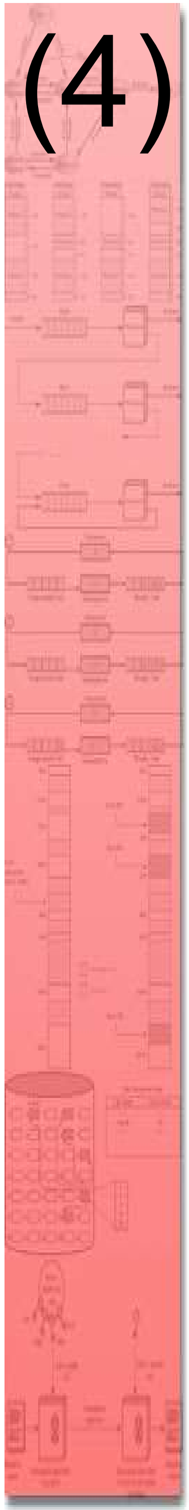
- INT 9: Keyboard
  - Activated every time a key is pressed or released.
  - The service routine stores the key code into the keyboard buffer.
- INT 0Ah (Not used)
- INT 0Bh: Serial port 1
- INT 0Ch: Serial port 2
- INT 0Dh: Hard disk (XT) or parallel port 2 (AT)
- INT 0Eh: Floppy
- INT 0Fh: Parallel port 1



4.1. BIOS interrupts (VII)

BIOS services

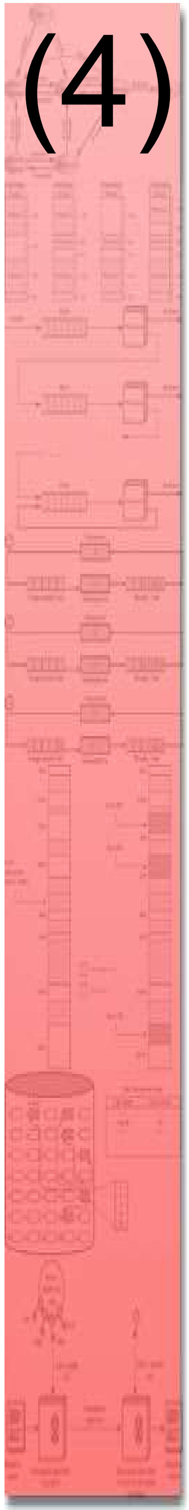
- INT 10h: Video input/output
  - Several functions related to the video output according to the value of **AH**.
- INT 11h: Physical equipment check
  - A description of the installed hardware (memory banks, number of serial and parallel ports, …) is returned in **AX**.
- INT 12h: Memory size
  - The number of 1 KB blocks of installed RAM memory is returned in **AX**.
- INT 13h: Disk access
  - Several functions related to the access to the floppy or hard disk at the sector or track levels according to the value of **AH**.



4.1. BIOS interrupts (VIII)

BIOS services

- INT 14h: Serial port RS-232 access
- INT 15h: Cassette access
- INT 16h: Keyboard input/output
  - Several functions related to the keyboard according to the value of **AH**.
- INT 17h: Printer input/output
- INT 18h: Execution of BASIC
- INT 19h: System restart
  - It reads sector 1 from track 0 of the boot disk and executes the DOS boot program.
- INT 1Ah: Time of day
  - Access to the timer’s 32-bit counter (**INT 8**).

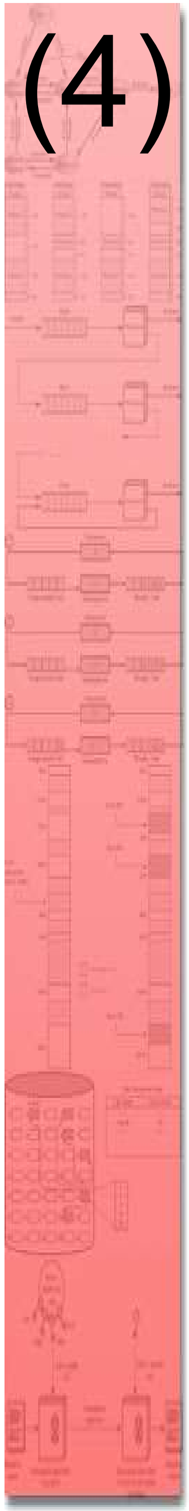


(4)

4.1. BIOS interrupts (IX)

User routines

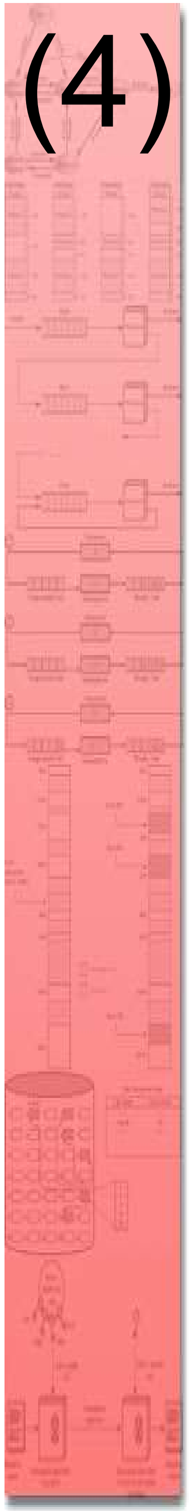
- INT 1Bh: Keyboard break
  - Activated by the service routine of **INT 9** (keyboard) when **Ctrl-C** (Ctrl-Break) is detected.
  - BIOS initializes the interrupt vector with an address that contains instruction **IRET**.
  - DOS changes the interrupt vector to a routine that sets an internal flag. DOS periodically checks that flag and calls **INT 23h** (Ctrl-Break service routine) when it is set.
- INT 1Ch: Timer tic
  - Activated by the service routine of **INT 8** (timer).
  - BIOS initializes the interrupt vector with an address that contains instruction **IRET**.


(4)

4.1. BIOS interrupts (X)

Pointers to data tables

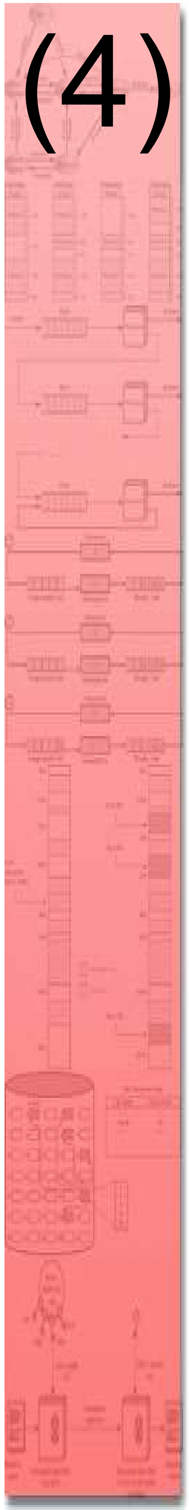
- Interrupts **1Dh** to **1Fh** and **41h** are actually addresses of parameter tables used by the video and disk services of BIOS.
- INT 1Dh: Video parameters
- INT 1Eh: Floppy parameters
- INT 1Fh: Table of graphical characters
- INT 41h: Hard disk parameters



(4)

4.2. DOS interrupts (I)

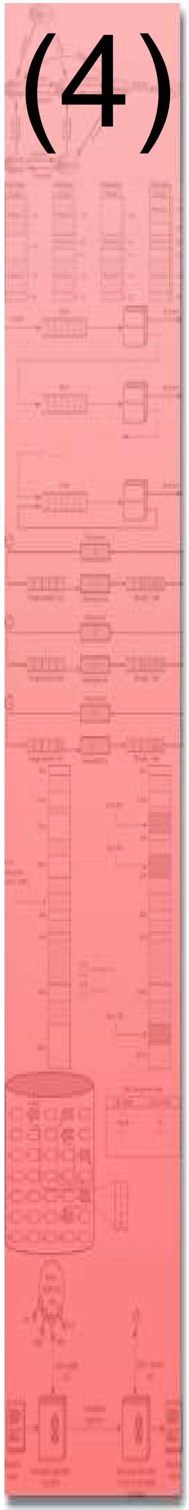
- INT 20h: Terminate program
  - Terminate program execution returning to the command interpreter. Microsoft recommends using **INT 21h** with **AH**=4Ch (terminate program, closing files and releasing memory) instead.
- INT 21h: DOS dispatcher
  - Execute the different DOS services according to **AH**.
- INT 22h: Termination address
  - Address of the routine executed when the program terminates. Not to be called directly.
- INT 23h: CTRL-Break service routine
  - Called by DOS when CTRL-C (CTRL-Break) is detected. Not to be called directly.



(4)

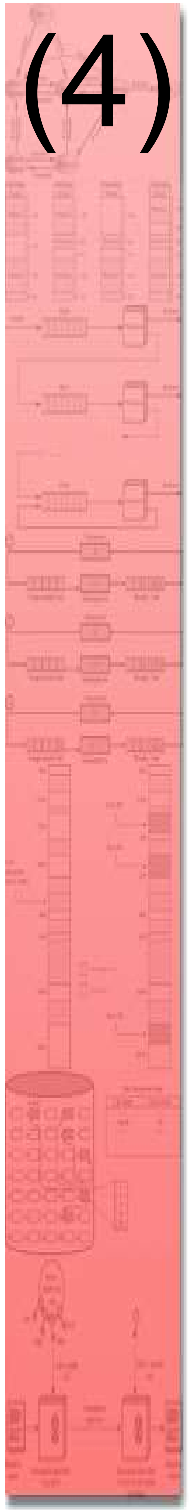
4.2. DOS interrupts (II)

- INT 24h: Critical error handler
  - Called by DOS when a critical error in accessing a hardware device is produced (disk, printer, …)
- INT 27h: Terminate program and leave resident
  - Terminate execution of a .COM program (*driver*) leaving it resident into memory.
  - Alternatively, use **INT 21h** with **AH**=31h to terminate and leave a .EXE program resident.



4.3. Program execution under DOS

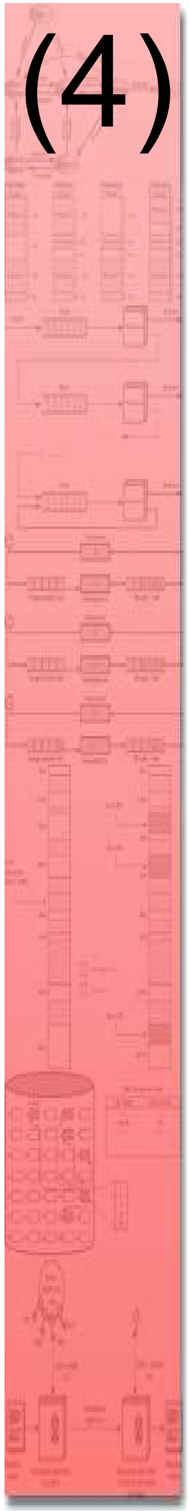
- Machine code programs are stored on disk as executable files.
- When a program is executed, the command interpreter loads the contents of its executable file into a free area reserved in RAM memory.
- As part of the load, an area of 256 bytes that contains data related to the program is added (Program Segment Prefix, PSP)
- Executable files can be in format .EXE or .COM, with their execution having a slightly different behavior.
- When a program terminates, control is returned to the DOS command interpreter. The memory it occupies is released unless it is resident.



(4)

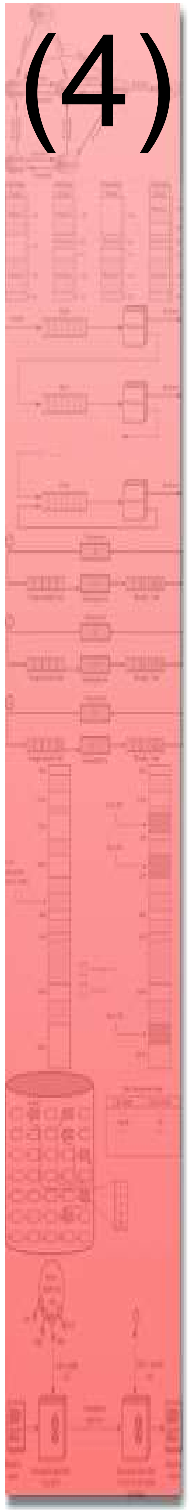
4.4. PSP (Program Segment Prefix) (I)

- Data zone of 256 bytes that heads .EXE and .COM programs once they are loaded into RAM memory for their execution.
- Generated by DOS through the command interpreter (COMMAND.COM).
- PSP's main fields:
  - Offsets 0 and 1 (2 bytes)
    - Instruction **INT** 20h.
    - Possible to terminate the program by jumping to offset 0 (not recommended).
  - Offsets 0Ah to 0Dh (4 bytes)
    - Original vector of the service routine of **INT** 22h (address of program termination)
    - When a program terminates, that vector is copied to the interrupt vector table and the CPU jumps to that address.



4.4. PSP (Program Segment Prefix) (II)

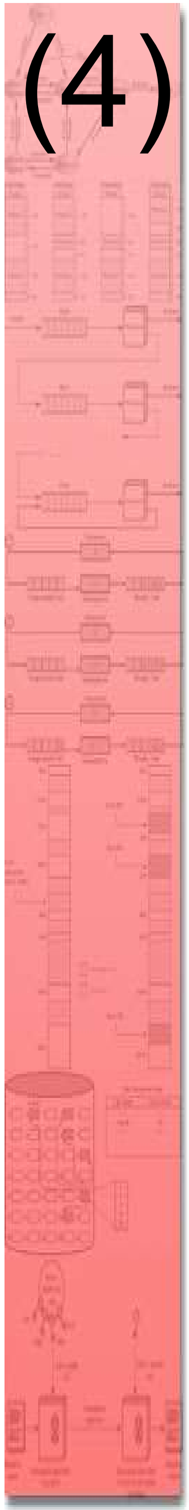
- Offsets 0Eh to 11h (4 bytes)
  - Original vector of the service routine of **INT** 23h (Ctrl-Break)
  - The program can change the service routine of that interrupt for capturing Ctrl-C/Ctrl-Break.
  - When the program terminates, the original routine is restored by copying its address from this field to the interrupt vector table.
- Offsets 12h to 15h (4 bytes)
  - Original vector of the service routine of **INT** 24h (critical error handler)
  - The program can change the service routine of that interrupt for capturing critical errors.
  - When the program terminates, the original routine is restored by copying its address from this field to the interrupt vector table.



(4)

4.4. PSP (Program Segment Prefix) (III)

- Offsets 2Ch and 2Dh (2 bytes)
  - Physical segment that contains a copy of the DOS environment variables.
  - It allows the program to access those variables.
- Offset 80h (1 byte)
  - Size in bytes of the program parameters in the command line.
- Offsets 81h to FFh (127 bytes)
  - ASCII codes of the program parameters in the command line. It ends with code 13 (carriage return).
  - It allows the program to access the parameters indicated in the command line.



(4)

4.4. PSP (Program Segment Prefix) (IV)

Example

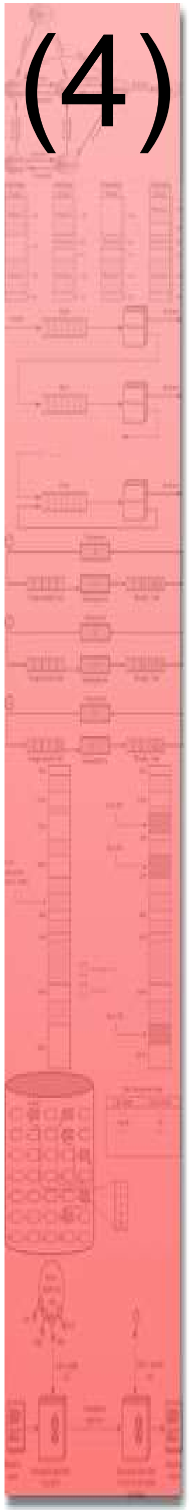
- Given the following environment variables (DOS command **SET**):

```
COMSPEC=C:\DOS60\COMMAND.COM
PROMPT=$P$G
TEMP=C:\TEMP
PATH=C:\TD;C:\TASM
```

- If executable file PROGRAM is run with parameters /D and C:\DISCO:

```
C:\> PROGRAM /D C:\DISCO
```

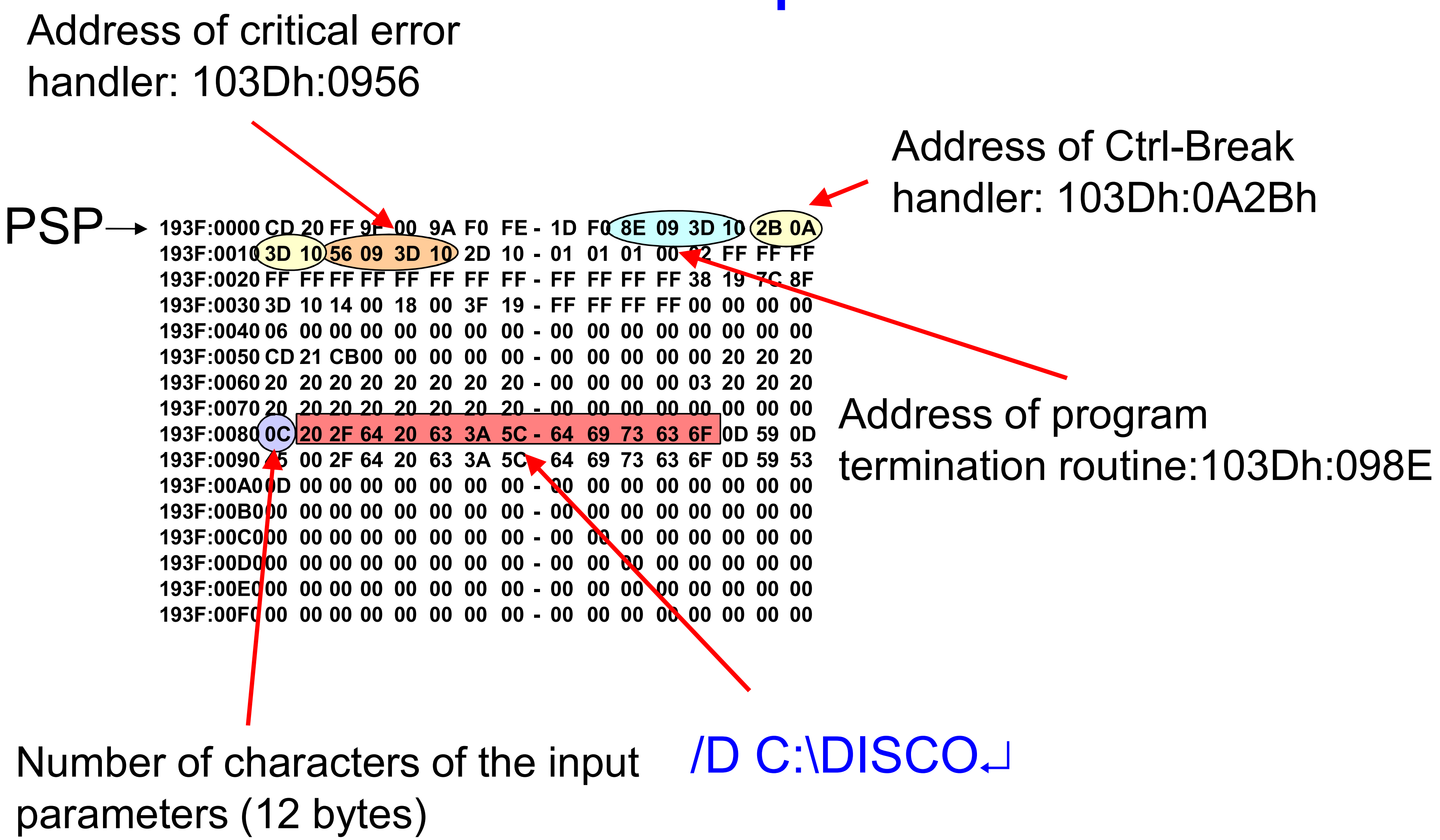
- The PSP would have the following values:



(4)

4.4. PSP (Program Segment Prefix) (V)

Example



Address of critical error handler: 103Dh:0956

Address of Ctrl-Break handler: 103Dh:0A2Bh

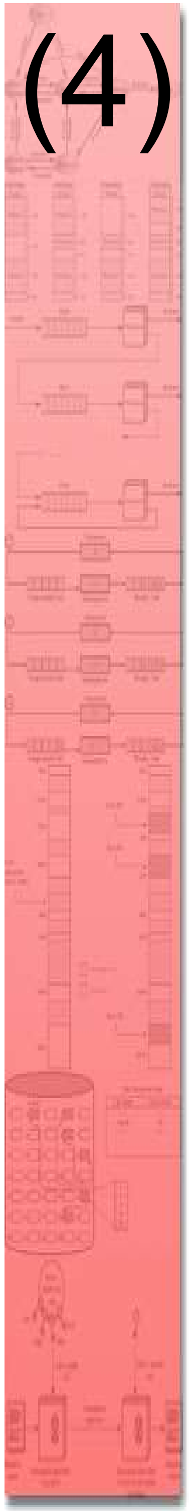
PSP→

```
193F:0000 CD 20 FF 9F 00  9A  F0  FE - 1D  F0  8E  09  3D  10  2B  0A
193F:0010 3D  10  56 09  3D  10  2D  10 - 01  01  01  00  02  FF  FF  FF
193F:0020 FF  FF FF FF  FF  FF  FF  FF - FF  FF  FF  FF 38  19  7C  8F
193F:0030 3D  10 14 00  18  00  3F  19 - FF  FF  FF  FF 00  00  00  00
193F:0040 06  00 00 00  00  00  00  00 - 00  00  00  00 00  00  00  00
193F:0050 CD 21 CB00  00  00  00  00 - 00  00  00  00 00  20  20  20
193F:0060 20  20 20 20  20  20  20  20 - 00  00  00  00 03  20  20  20
193F:0070 20  20 20 20  20  20  20  20 - 00  00  00  00 00  00  00  00
193F:0080 0C  20 2F 64  20  63  3A  5C - 64  69  73  63 6F  0D  59  0D
193F:0090 45  00 2F 64  20  63  3A  5C - 64  69  73  63 6F  0D  59  53
193F:00A00D  00 00 00  00  00  00  00 - 00  00  00  00 00  00  00  00
193F:00B000  00 00 00  00  00  00  00 - 00  00  00  00 00  00  00  00
193F:00C000  00 00 00  00  00  00  00 - 00  00  00  00 00  00  00  00
193F:00D000  00 00 00  00  00  00  00 - 00  00  00  00 00  00  00  00
193F:00E000  00 00 00  00  00  00  00 - 00  00  00  00 00  00  00  00
193F:00F000  00 00 00  00  00  00  00 - 00  00  00  00 00  00  00  00
```

Address of program termination routine:103Dh:098E

Number of characters of the input parameters (12 bytes)

/D C:\DISCO↵



4.4. PSP (Program Segment Prefix) (VI)

Example

PSP→

```
193F:0000 CD 20 FF 9F 00 9A F0 FE - 1D F0 8E 09 3D 10 2B 0A
193F:0010 3D 10 56 09 3D 10 2D 10 - 01 01 01 00 02 FF FF FF
193F:0020 FF FF FF FF FF FF FF FF - FF FF FF FF 38 19 7C 8F
193F:0030 3D 10 14 00 18 00 3F 19 - FF FF FF FF 00 00 00 00
193F:0040 06 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00
193F:0050 CD 21 CB 00 00 00 00 00 - 00 00 00 00 00 20 20 20
193F:0060 20 20 20 20 20 20 20 20 - 00 00 00 00 03 20 20 20
193F:0070 20 20 20 20 20 20 20 20 - 00 00 00 00 00 00 00 00
193F:0080 0C 20 2F 64 20 63 3A 5C - 64 69 73 63 6F 0D 59 0D
193F:0090 45 00 2F 64 20 63 3A 5C - 64 69 73 63 6F 0D 59 53
193F:00A0 0D 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00
193F:00B0 00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00
193F:00C0 00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00
193F:00D0 00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00
193F:00E0 00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00
193F:00F0 00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00
```

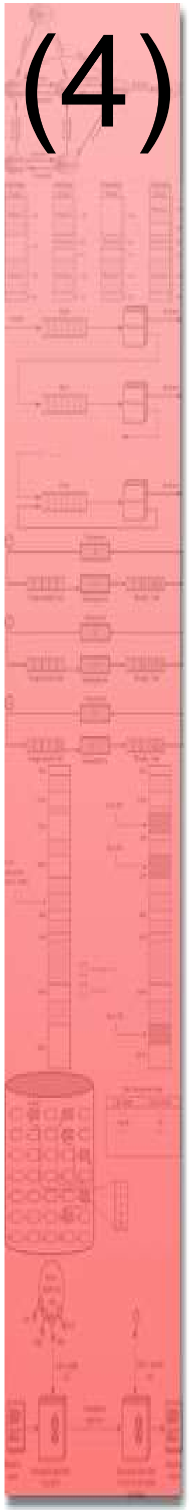
Number of segment with a copy of the DOS environment variables: 1938h

```
1938:0000 43 4F 4D 53 50 45 43 3D - 43 3A 5C 44 4F 53 36 30
1938:0010 5C 43 4F 4D 4D 41 4E 44 - 2E 43 4F 4D 00 50 52 4F
1938:0020 4D 50 54 3D 24 70 24 67 - 00 54 45 4D 50 3D 43 3A
1938:0030 5C 54 45 4D 50 00 50 41 - 54 48 3D 43 3A 5C 54 44
1938:0040 3B 43 3A 5C 54 41 53 4D - 00 00 01 00 43 3A 5C 41
```

COMSPEC=C:\DOS60
\COMMAND.COM.PRO
MPT=$P$G.TEMP=C:
\TEMP.PATH=C:\TD
;C:\TASM....C:\A

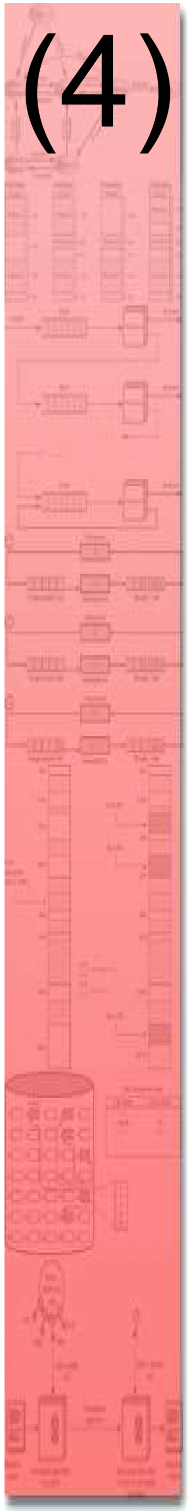
4.5.Types of programs: EXE, COM and resident (I)

- Three types of executable files in DOS:
- **.BAT**
  - Sequences of DOS commands (not machine code)
- **.EXE**
  - Machine code programs.
  - Generated by a linker from one or several object files generated by a compiler or an assembler.
- **.COM**
  - Machine code programs.
  - The program occupies a single 64 KB physical segment with code, data and stack.
  - The first executable instruction is at address 256 (100h) with respect to the segment’s origin. Directive **ORG** 256 must be used prior to the first assembly instruction.
  - Created from a .EXE with command **EXE2BIN** or directly with option **/t** of the linker (TLINK).



4.5.Types of programs: EXE, COM and resident (II)

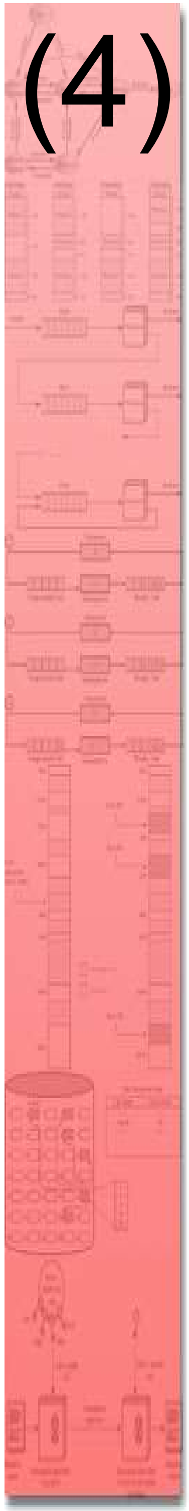
- Execution of **.EXE** programs:
  - **CS** and **SS** initialized by DOS.
  - **DS** and **ES** point to PSP.
  - **IP** initialized with address indicated in the **END** directive.
  - **SP** initialized with the largest value of the stack segment.
  - **AL** indicates if the disk unit (C, D, …) of the first file is valid (**AL**= 0 is valid).
  - **AH** indicates if the disk unit (C, D, …) of the second file is valid (**AH**= 0 is valid).
  - When the program terminates, control is returned to the operating system (command interpreter) and the memory area where the program was loaded is released.



(4)

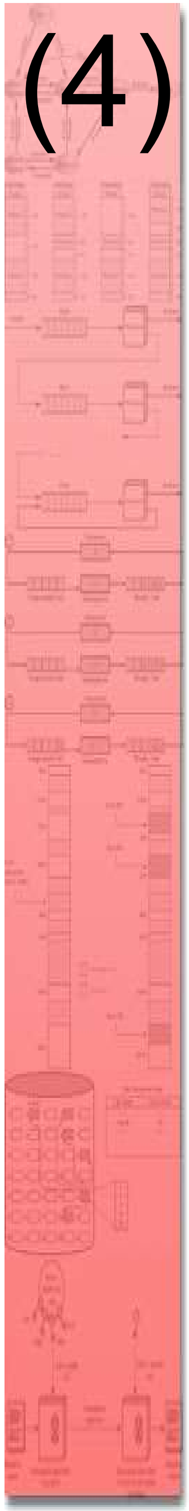
4.5.Types of programs: EXE, COM and resident (III)

- Execution of **.COM** programs:
  - **CS**, **DS**, **ES** and **SS** point to PSP.
  - **IP** initialized to 256 (position after PSP).
  - **SP** initialized to 0FFFEh.
  - **AL** indicates if the disk unit (C, D, …) of the first file is valid (**AL**= 0 is valid).
  - **AH** indicates if the disk unit (C, D, …) of the second file is valid (**AH**= 0 is valid).
  - When the program terminates, control is returned to the operating system (command interpreter) and the memory area where the program was loaded is released.



4.5.Types of programs: EXE, COM and resident (IV)

- Resident program (*Terminate & Stay Resident, TSR)*
  - Programs .COM or .EXE that end their execution not releasing part of the memory they occupy.
  - Their memory position is usually stored as an interrupt vector.
  - They can be called from other programs or interrupt service routines.
- Resident programs .COM (*installation*)
  - They terminate with **INT** 27h.
  - **DX** must contain the offset of the position right after the last byte that must stay resident.
  - They consist of two parts:
    - The information (code, variables, …) that is left resident.
    - The code that installs the information that is left resident.
  - Installation example of an interrupt service routine for interrupt 40h:

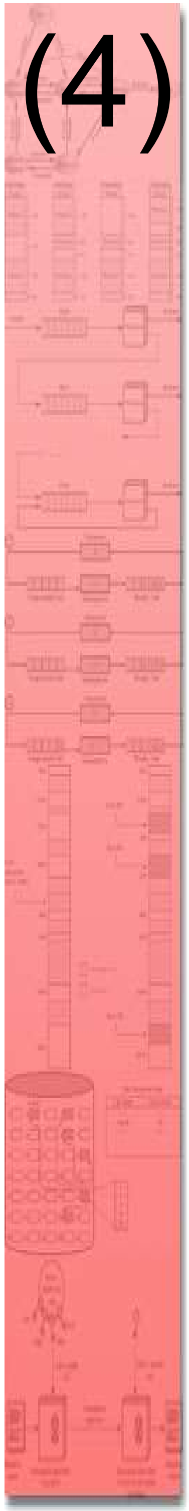


(4)

4.5.Types of programs: EXE, COM and resident (V)

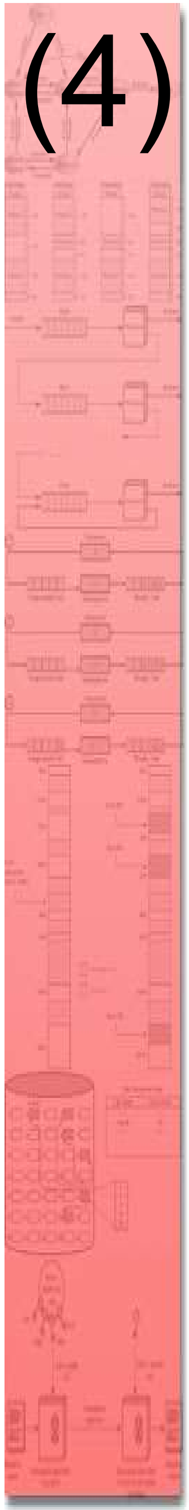
```
code SEGMENT
        ASSUME cs : code

        ORG 256

start:   jmp installer

; Global variables
table   DB   "abcdf "
flag     DW  0

; Interrupt service routine
isr PROC FAR
        ; Save modified registers
        push ...
        ; Routine instructions
        ...

        ; Restore modified registers
        pop ...
        iret
isr ENDP
...
```

```
...
installer PROC
            mov ax, 0
            mov es, ax
            mov ax, OFFSET isr
            mov bx, cs
            cli
            mov es:[ 40h*4 ], ax
            mov es:[ 40h*4+2 ], bx
            sti
            mov dx, OFFSET installer
            int 27h ;  Terminate and stay resident
                         ;  PSP, variables, isr routine.
installer ENDP

code ENDS
END start
```



4.5.Types of programs: EXE, COM and resident (VI)

- Resident program .COM (*uninstallation*)
  - A program or routine (uninstaller) must be executed to release the memory that was left resident.
  - A physical memory segment is released through **INT** 21h with **AH**=49h and **ES**=segment number.
  - Two physical segments must be released:
    - Code segment of the resident program (usually stored in some interrupt vector).
    - Segment of environment variables (offset 2Ch of the PSP).
  - Before releasing a program, it is convenient to verify that it is really installed:
    - Interrupt vector different from zero.
    - First bytes of the service routine belong to the program that is to be uninstalled (program’s digital signature).
  - Example of uninstallation of the service routine of interrupt 40h:



4.5.Types of programs: EXE, COM and resident (VII)

```
uninstall_40h PROC                    ; Uninstall ISR of INT 40h
      push ax bx cx ds es

      mov cx, 0
      mov ds, cx                      ;  Segment of interrupt vectors
      mov es, ds:[ 40h*4+2 ]          ;  Read ISR segment
      mov bx, es:[ 2Ch ]              ;  Read segment of environment from ISR's PSP.

      mov ah, 49h
      int 21h           ;  Release ISR segment (es)
      mov es, bx
      int 21h           ;  Release segment of environment variables of ISR

      ;  Set vector of interrupt 40h to zero
      cli
      mov ds:[ 40h*4 ], cx            ;  cx = 0
      mov ds:[ 40h*4+2 ], cx
      sti

      pop es ds cx bx ax
      ret
uninstall_40h ENDP
```