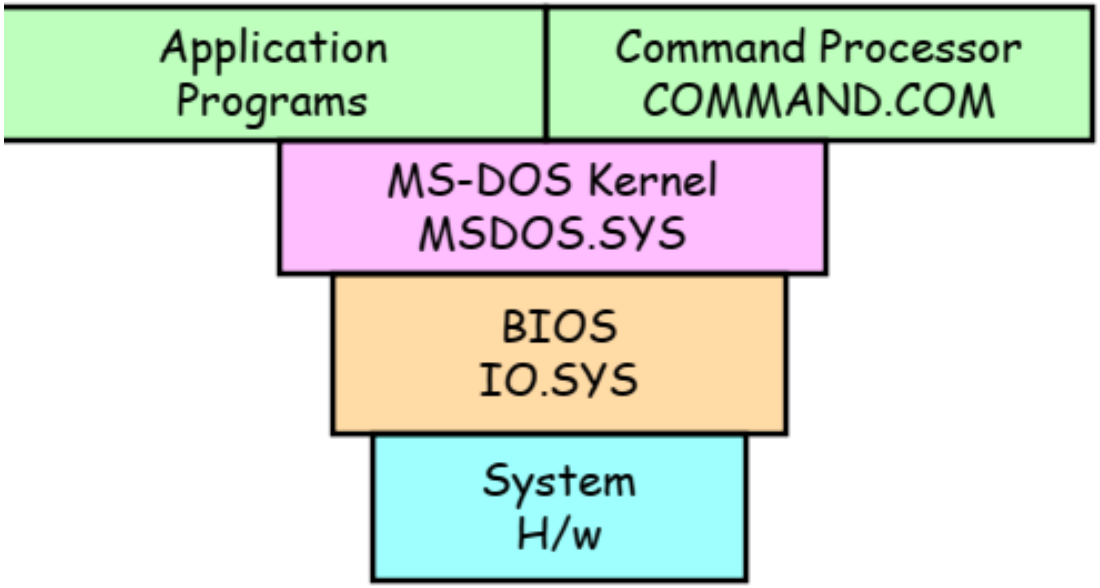
Software Interrupts

Dr. Vinay Chamola
Department of Electrical and Electronics

BITS Pilani
Pilani Campus

MS-DOS FUNCTIONS & BIOS CALLS

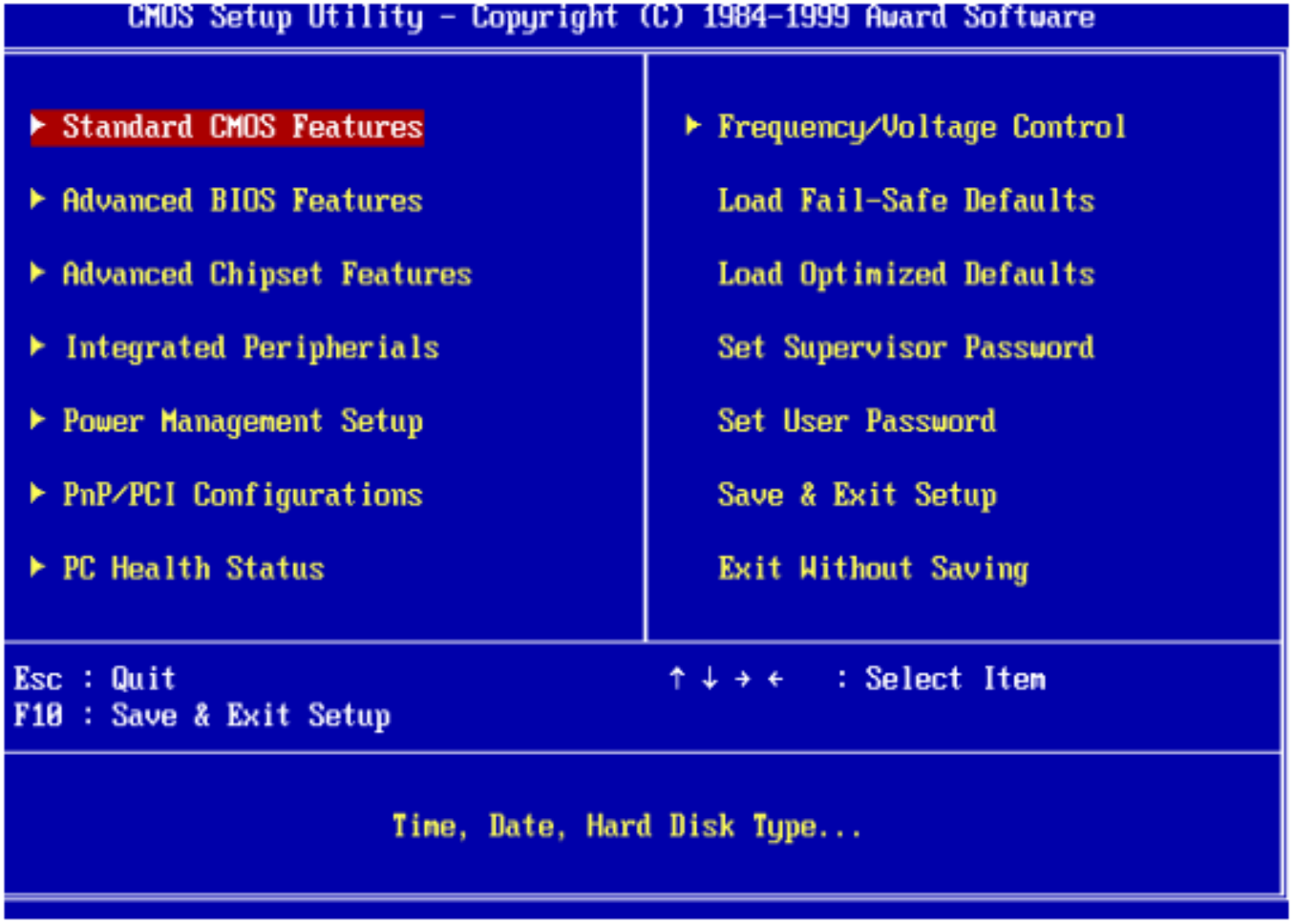


BIOS

- Hardware Specific
- Provided by manufacturer
- Two parts
  - Resident (BIOS)
  - Non-Resident (IO.SYS)
- Offers convenient way to add funcs to BIOS

BIOS

- Test and Initialize I/O
- Load Boot Loader/ OS


CMOS Setup Utility - Copyright (C) 1984-1999 Award Software
Standard CMOS Features
Advanced BIOS Features
Advanced Chipset Features
Integrated Peripherials
Power Management Setup
PnP/PCI Configurations
PC Health Status
Frequency/Voltage Control
Load Fail-Safe Defaults
Load Optimized Defaults
Set Supervisor Password
Set User Password
Save & Exit Setup
Exit Without Saving
Esc : Quit
F10 : Save & Exit Setup
↑ ↓ → ← : Select Item
Time, Date, Hard Disk Type...

INT NN_H

- INT 21_H - DOS INTERRUPT
- INT 10_H – BIOS INTERRUPT

INT

- Operates similar to call
  - Processor first pushes the Flags
  - TF, IF cleared
  - Next the processor pushes the current CS contents on to the stack
  - Next IP is pushed

Sequence of events for INT 08_H

CS:IP – $0100:0200_H$

Flag – $0F81_H$

				O	D	I	T	S	Z		A		P		C

SP Initial – 0050_H

$004F_H$	SP-1	$0F_H$
$004E_H$	SP-2	81_H
$004D_H$	SP-3	01_H
$004C_H$	SP-4	00_H
$004B_H$	SP-5	02_H
$004A_H$	SP-6	00_H

$08 * 04 = 20_H$ (32)

Memory/ISR table	
00020_H	20_H
00021_H	00_H
00022_H	50_H
00023_H	08_H

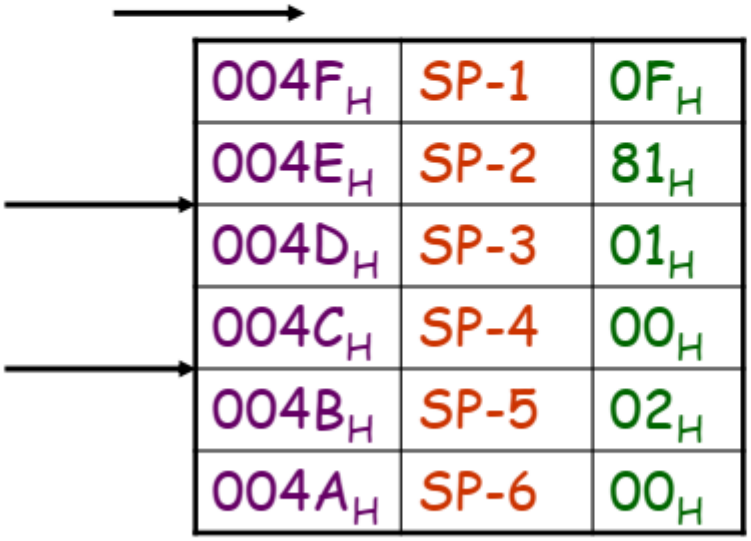
$0850_H:0020_H$

IRET

Used at end of ISR

For stack handling

SP Final - 0050_H



$004F_H$	SP-1	$0F_H$
$004E_H$	SP-2	81_H
$004D_H$	SP-3	01_H
$004C_H$	SP-4	00_H
$004B_H$	SP-5	02_H
$004A_H$	SP-6	00_H

Flag - $0F81_H$

0100_H

0200_H

SP Initial - $004A_H$

E.g. for a DOS interrupt

To read a key pressed

```
MOV     AH,01H
INT     21H
```

-Reads key pressed and stores the ASCII equivalent in AL

-Key pressed will be echoed on screen

AH	Description	AH	Description
01	Read character from STDIN	02	Write character to STDOUT
05	Write character to printer	06	Console Input/Output
07	Direct char read (STDIN), no echo	08	Char read from STDIN, no echo
09	Write string to STDOUT	0A	Buffered input
0B	Get STDIN status	0C	Flush buffer for STDIN
0D	Disk reset	0E	Select default drive
19	Get current default drive	25	Set interrupt vector
2A	Get system date	2B	Set system date
2C	Get system time	2D	Set system time
2E	Set verify flag	30	Get DOS version
35	Get Interrupt vector		
36	Get free disk space	39	Create subdirectory
3A	Remove subdirectory	3B	Set working directory
3C	Create file	3D	Open file
3E	Close file	3F	Read file
40	Write file	41	Delete file
42	Seek file	43	Get/Set file attributes
47	Get current directory	4C	Exit program
4D	Get return code	54	Get verify flag
56	Rename file	57	Get/Set file date

Examples

Purpose: Input a character from keyboard (STDIN) with Echo

Program Segment

```
MOV     AH, 01h                    ; AH -01 parameter for INT 21h
INT     21h
```

You have to run the program using debug/debug32 and the ASCII key you press will be stored in AL register.

Purpose: Input a character from keyboard (STDIN) without Echo

Program Segment

```
MOV     AH, 08h                    ; AH -08 parameter for INT 21h
INT     21h
```

You have to run the program using debug/debug32 and the ASCII key you press will be stored in AL register. The difference is the character will not be visible on the screen when you type it

Purpose: Input a string from keyboard (STDIN)

Program

```
.data
max1    db      32          ; 32 is max no. of chars that a user can type in (max possible – 255)
act1    db      ?           ; actual count of keys that user types will be stored here after int has
                            ; executed (Note this cannot exceed the value specified in max1 –
                            ; actual  keys you enter will 31 as the 32nd will be Enter key)
inp1    db      32 dup(0)   ; Reserve 32 locations for input string
.code
.startup
        LEA     DX,max1
        MOV     AH, 0Ah
        INT     21h
```

After the interrupt, act 1 will contain the number of characters read, and the characters themselves will start at inp1 The characters will be terminated by a carriage return (ASCII code 0Dh), although this will not be included in the count (Note: this will not be included in the ACT1 but you have to count Enter also when you are specifying it in max1)

Purpose: Output a character to display (STD OUT)

Program Segment

```
MOV    DL, 'A'
MOV    AH, 02h
INT    21h
```

After Interrupt is executed character 'A' will be displayed on the screen.

Purpose: Output a string on display (STDOUT)

Program

```
.data
str1    db      'HELLO $'       ; all strings must terminate with '$' ASCII value (24h)
.code
.startup
        lea     dx, str1
        mov     ah, 09h
        int     21h
```

When interrupt is executed the string "HELLO" will be displayed on screen.

Remove the '$' sign. What happens?

Task1: Write an ALP that will take in a string of maximum 20 characters from user and display it on the next line on the screen (ASCII equivalent for newline is 0Dh (enter) followed by 0Ah (next line))

Task2: Write an ALP that does the following

(1) Display the string "Enter User Name" and goes to the next line
(2) Takes in the user entered string compares with user name value already stored in memory
(3) If there is no match it should exit.
(4) If there is a match it should display the string "Enter Password" and goes to next line
(5) Takes in password entered by the user and compares with password already stored in memory
(6) If there is no match it should exit'
(7) If there is a match it should display "Hello *Username*"

Note:

While the username is being entered it can be displayed but when password is being entered user pressed key should be displayed instead it should display "*" for every key pressed.

The user name size is fixed to 10 characters and password to 8 characters.

Thank You