MACROS

MASM

MACROS

- A macro is a group of instructions that perform a task, just as procedure performs one task

- All the instructions defined in the macro are inserted in the program at the point of usage

- Name of macro and any parameters associated with it are typed , and the assembler then inserts them into the program.

- Macro is faster than procedure because no CALL and RET

```
MOVE        MACRO       A, B
            PUSH        AX
            MOV         AX, B
            MOV         A, AX
            POP         AX
            ENDM
```

```
MOV          CX,100

MOVE         VAR1, VAR2
PUSH         AX
MOV          AX, VAR2
MOV          VAR1, AX
POP          AX

MOV          BX,VAR1

MOVE         VAR3,   VAR4
PUSH         AX
MOV          AX,  VAR4
MOV          VAR3, AX
POP          AX

MOV          DX,VAR3
ADD          BX,DX
DEC          CX
```
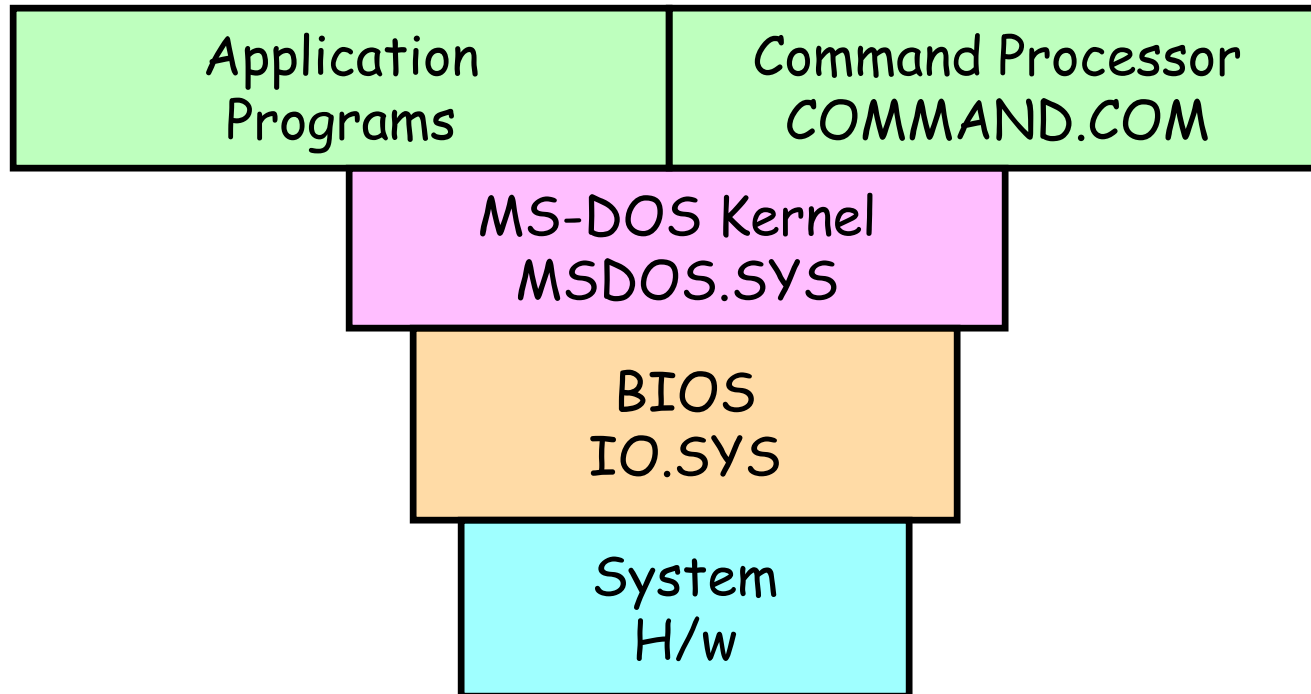
SOFTWARE INTERRUPTS

MS-DOS FUNCTIONS & BIOS CALLS

Application Programs	Command Processor COMMAND.COM

MS-DOS Kernel MSDOS.SYS

BIOS IO.SYS

System H/w

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          ▶ Frequency/Voltage Control

  ▶ Advanced BIOS Features            Load Fail-Safe Defaults

  ▶ Advanced Chipset Features         Load Optimized Defaults

  ▶ Integrated Peripherials           Set Supervisor Password

  ▶ Power Management Setup            Set User Password

  ▶ PnP/PCI Configurations            Save & Exit Setup

  ▶ PC Health Status                  Exit Without Saving

  Esc : Quit                    ↑ ↓ → ←   : Select Item
  F10 : Save & Exit Setup

                 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)

00020_H – IP

00022_H – CS

Memory/ISR table	
00020_H	20_H
00021_H	00_H
00022_H	50_H
00023_H	08_H
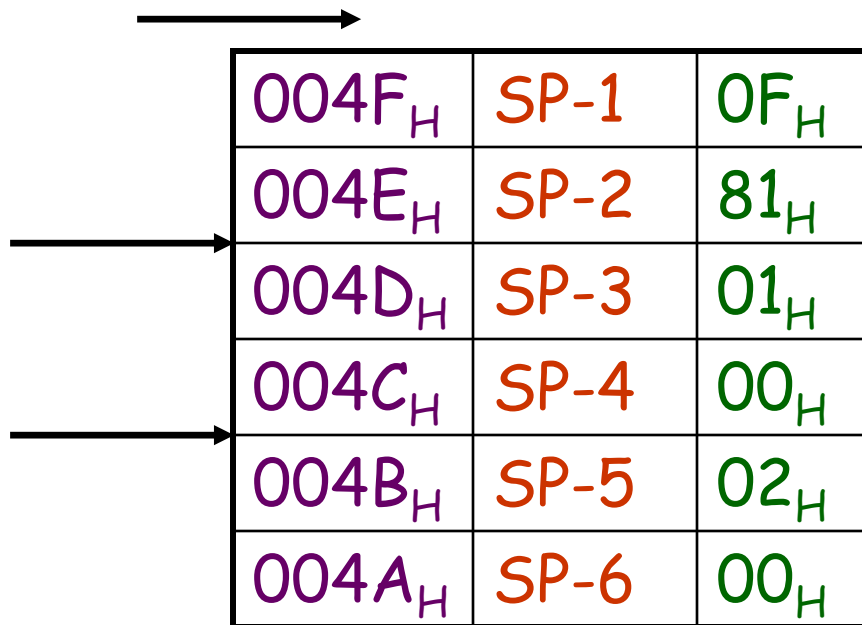
$0850_H:0020_H$

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,01_H

      INT 21_H

-Reads key pressed and stores the ASCII equivalent in AL

-Key pressed will be echoed on screen

MISCELLANEOUS INSTRUCTIONS

ADDITIONAL INSTRUCTIONS

- SAHF- Store AH to Flag Register (Lower)

- LAHF- Load Flag into AH Register (Lower)

- CMC- Complement Carry Flag

- STI

- CLI

- XCHG DSR,SRC

ADDITIONAL INSTRUCTIONS

- CWDE

- CDQ

ADDITIONAL INSTRUCTIONS

- MOVSX DST, SRC

- MOVZX DST,SRC

ADDITIONAL INSTRUCTIONS

- SHLD DST, SRC, Count

- SHRD DST, SRC, Count

ADDITIONAL INSTRUCTIONS

- CMPXCHG
  - CMPXCHG DST,SRC

ADDITIONAL INSTRUCTIONS

- BSWAP REG32

ADDITIONAL INSTRUCTIONS

- XADD

  - XADD REG1,REG2