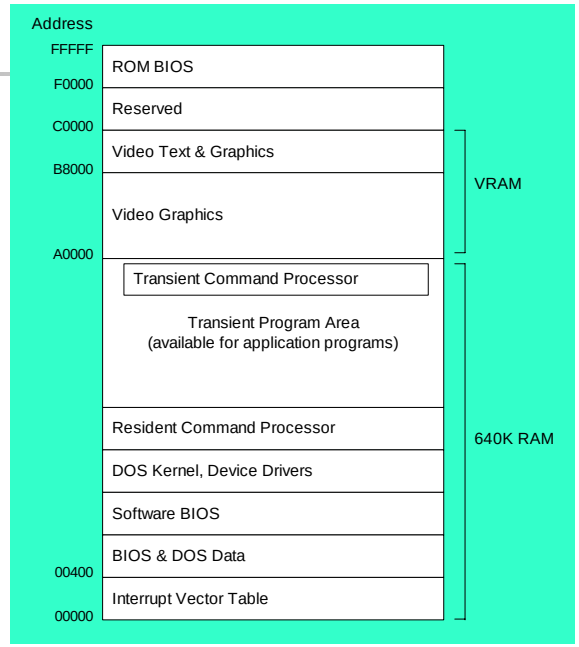
Chapter 13:
16-Bit MS-DOS Programming

Real-Address Mode

- Real-address mode (16-bit mode) program의 특징
  - 최대 1 MB RAM
  - Single tasking
  - 16 bit Offsets → 세그먼트 크기 최대 64KB
  - No memory boundary protection
- MS-DOS
  - Digital Research CP/M을 기초로 하여 Microsoft에서 개발됨
  - IBM PC (8088 CPU 사용)를 위한 운영체제로 출발 – PC-DOS
  - 후에 MS-DOS로 이름을 바꿈

MS-DOS Memory Organization

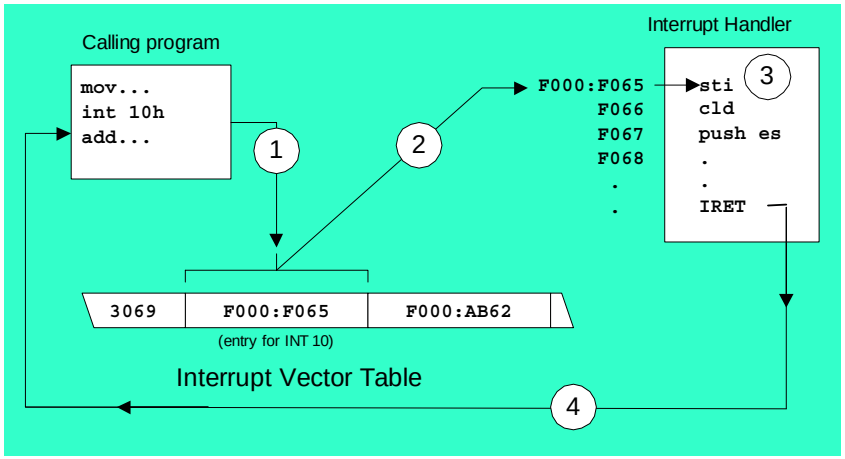
- IBM-PC의 Memory Map
  - 00000 – 9FFFF (640K) : RAM
  - A0000 – BFFFF (128K) : Video memory (text, graphics)
  - C0000 – EFFFF (192K) : reserved (for device controller)
  - F0000 – FFFFF (64K) : ROM BIOS
- RAM의 Memory map
  - Interrupt Vector Table (00000 – 003FF)
  - BIOS & DOS data
  - Software BIOS
  - MS-DOS kernel
  - Resident command processor
  - Transient programs



INT Instruction

- INT number
  - software interrupt 수행
  - number에 따라서 지정되는 interrupt handler를 수행함
    - number: 00– FF (0 – 255)
- Interrupt Vector Table(IVT)
  - interrupt handler의 시작주소를 보관하는 table
  - real 모드에서는
    - table은 주소 000 – 3FF (1KB)에 위치함
    - interrupt handler 주소는 segment와 offset으로 구성 (32 bit)
  - protected 모드에서는
    - table의 위치는 OS에 의해서 지정될 수 있음
    - interrupt handler 주소 이외에도 protection, privilege level의 정보가 함께 보관됨
- 운영체제의 service는 대부분 software interrupt를 통해서 제공됨

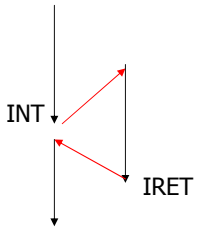
Interrupt Vectoring Process

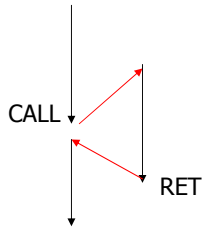


int 10h을 위한 entry: 4 x 10h = 40h 번지 (각 entry당 4B)

IRET instruction

- IRET
  - return from interrupt handler
- INT-IRET와 CALL-RET의 비교



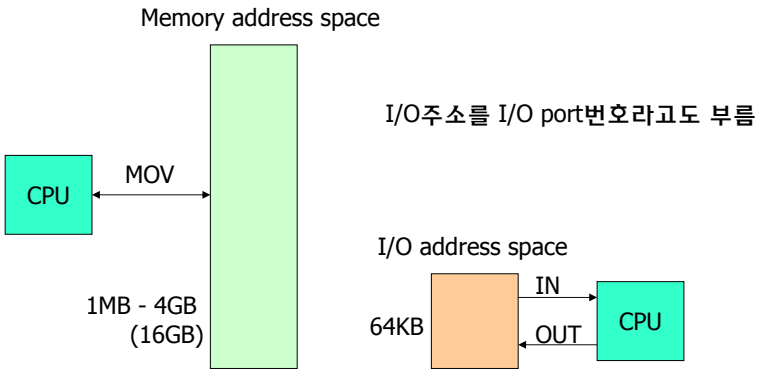


INT: push flags, CS, IP
RET: pop IP, CS, flags

CALL: push (CS,) IP ; return addr
RET: pop IP (,CS)

Memory address와 I/O address space

- I/O address space
  - I/O 장치를 위한 주소 공간
  - 80x86 family는 64KB의 I/O address space를 제공함



IN, OUT Instructions

- IN Acc, imm8 ; imm8: 8비트 I/O주소 보관
  IN Acc, DX ; DX: 16비트 I/O주소 보관
  - Acc ← I/O port ; Acc는 EAX, AX, AL
- OUT imm8, Acc
  OUT DX, Acc
  - I/O port ← Acc

```
IN AL, 20          ; AL ← Port(20)

MOV DX, 3F0
IN AL, DX          ; AL ← Port(3F0)
```

```
OUT 80, AX         ; Port(80) ← AX

MOV DX, CC00
IN AX, DX          ; Port(CC00) ← AX
```


Common Interrupt Services

- ROM BIOS services
  - INT 10h Video Services
  - INT 16h Keyboard Services
  - INT 17h Printer Services
  - INT 1Ah Time of Day
  - INT 1Ch User Timer Interrupt
- MS-DOS services
  - INT 21h MS-DOS Services
    (AH : function number - service 종류를 지정)

MS-DOS Function Calls (INT 21h)

- AH=4Ch : terminate Process

```
mov ah, 4Ch        ; terminate process
mov al, 0          ; return code
int 21h
```

↕ 같음

```
.Exit 0            ; macro call
```

- ASCII control characters
  - 08h - Backspace (Ctrl+H)
  - 09h - Horizontal tab (Ctrl+I)
  - 0Ah - Line feed (Ctrl+J)
  - 0Ch - Form feed (Ctrl+L)
  - 0Dh - Carriage return (Ctrl+M)
  - 1Bh - Escape character

Selected Output Functions

- 02h, 06h - Write character to standard output
- 05h - Write character to default printer
- 09h - Write string to standard output ($ terminated string)
- 40h - Write string to file or device

Function Calls – character output functions

- 02h, 06h - Write character to standard output

```
mov ah,02h
mov dl,'A'
int 21h
```

```
mov ah,06h
mov dl,08h    ; BS
int 21h
```

do not filter control character

- 05h - Write character to default printer

```
mov ah,05h
mov dl,41h    ; 'A'
int 21h
```

```
mov ah,05h
mov dl,09h     ; tab
int 21h
```


Function Calls – string output functions

- 09h - Write string to standard output ($ terminated string)

```
.data
string BYTE "This is a string$"
.code
    mov  ah,9
    mov  dx,OFFSET string
    int  21h
```

- 40h - Write string to file or device

```
.data
message "Writing a string to the console"
bytesWritten WORD ?
.code
    mov ah,40h
    mov bx,1                ; device(1:console)
    mov cx,LENGTHOF message
    mov dx,OFFSET message
    int 21h
    mov bytesWritten,ax
```


Selected Input Functions

- 01h, 06h - Read character from standard input
- 0Ah - Read array of buffered characters from standard input
- 0Bh - Get status of the standard input buffer
- 3Fh - Read from file or device

Function Calls – character input functions

- 01h - Read character from standard input (with echo)
  - AL = read character
  - waits for input if buffer is empty
  - check Ctrl-Break(Ctrl-C)
- 06h - Read character from standard input (without echo)
  - AL = read character, DL=FFh
  - Does not wait for input
  - use ZF(zero flag) to check for an input character

```
.data
char BYTE ?
.code
L1: mov  ah,06h          ; keyboard input
    mov  dl,0FFh         ; don't wait for input
    int  21h
    jz   L1              ; no character? repeat loop
    mov  char,al         ; character pressed: save it
```

Function Calls – input functions

- 0Ah - Read array of buffered characters from standard input

```
KEYBOARD STRUCT
    maxInput BYTE count
    inputCount BYTE ?
    buffer BYTE 80 DUP (?)
KEYBOARD ENDS

.data
kybdData KEYBOARD <>
.code
    mov ah,0Ah
    mov dx,OFFSET kybdData
    int 21h
```


Example: String Encryption

- 동작: 표준입력을 암호화 하여 표준출력으로 보냄
  - 암호화는 XOR 연산을 이용함

```
XORVAL = 239              ; any value between 0-255
.code
main PROC
     mov  ax,@data
     mov  ds,ax
L1:  mov  ah,6            ; direct console input
     mov  dl,0FFh         ; don't wait for character
     int  21h             ; AL = character
     jz   L2              ; quit if ZF = 1 (EOF)
     xor  al,XORVAL
     mov  ah,6            ; write to output
     mov  dl,al
     int  21h
     jmp  L1              ; repeat the loop
L2:  exit
```


Date/Time Functions

- 2Ah - Get system date
- 2Bh - Set system date *
  - CX=year, DH=month, DL=day, AL=weekday(0:Sunday)
- 2Ch - Get system time
- 2Dh - Set system time *
  - CH=hour, CL=min, DH=sec, DL=0.01sec

Standard MS-DOS File I/O Services

- 716Ch (AX) - Create or open file
  - BX = access mode (0 = read, 1 = write, 2 = read/write)
  - CX = attributes (0 = normal, 1 = read only, 2 = hidden, 3 = system, 8 = volume ID, 20h = archive)
  - DX = action (1 = open, 2 = truncate, 10h = create)
  - DS:SI = segment/offset of filename
  - DI = alias hint (optional)
- 3Eh - Close file handle
- 42h - Move file pointer
- 5706h (AX) - Get file creation date and time

Function 716Ch - Create/Open a New File

```
mov  ax,716Ch              ; extended open/create
mov  bx,2                  ; read-write
mov  cx,0                  ; normal attribute
mov  dx,10h + 02h          ; create + truncate
mov  si,OFFSET Filename
int  21h
jc   failed
mov  handle,ax             ; file handle
mov  actionTaken,cx        ; action taken to open file
```

```
mov  ax,716Ch               ; extended open/create
mov  bx,0                   ; read-only
mov  cx,0                   ; normal attribute
mov  dx,1                   ; open existing file
mov  si,OFFSET Filename
int  21h
jc   failed
mov  handle,ax              ; file handle
mov  actionTaken,cx         ; action taken to open file
```


Function 3Eh - Close file handle

```
.data
filehandle WORD ?
.code
    mov  ah,3Eh
    mov  bx,filehandle
    int  21h
    jc   failed
```


Move file pointer

- file에 대한 random access 허용

```
mov   ah,42h
mov   al,0          ; offset from beginning
mov   bx,handle
mov   cx,offsetHi
mov   dx,offsetLo
int   21h
```

  - AL : pointer offset 계산방법
    - 0: Offset from the beginning of the file
    - 1: Offset from the current pointer location
    - 2: Offset from the end of the file

Function 5706h - Get file creation date and time

```
mov ax,5706h
mov bx,handle        ; handle of open file
int 21h
jc  error
mov date,dx
mov time,cx
mov milliseconds,si
```

- file 생성일시가 file 수정 또는 접근 일시와 같지 않을 수 있음