

6. MS-DOS

6.1 MS-DOS 호출 방법

6.2 주요 MS-DOS 기능

6.1 MS-DOS 호출 방법

▶ MS-DOS 호출 방법

- 'INT N' 명령은 소프트웨어로 인터럽트를 호출하는 명령어이다. N은 1 바이트 수로 0부터 255까지의 수이다. 8086 또는 펜티엄의 실제모드에서 'INT N' 명령은 다음과 같이 동작한다.
- 1. 코드 세그먼트 레지스터(CS)와 명령어 포인터(IP) 및 플래그 레지스터(PSW)를 스택에 PUSH(저장)한다.
- 2. 'N*4' 번지에서 16비트를 읽어서 IP에 저장하고, 'N*4 + 2' 번지에서 16비트를 읽어서 CS에 저장한다. 즉, 'N*4' 번지에 저장된 번지로 분기한다.
- 3. 'N*4' 번지에 저장된 번지에는 해당 인터럽트가 발생하였을 경우에 수행되는 프로그램이 기억되어 있다. 이 프로그램을 인터럽트 서비스 루틴이라고 부른다.
- 4. 인터럽트 서비스 루틴의 수행을 마치면, 'IRET' 명령을 수행한다. IRET 명령은 코드 세그먼트 레지스터(CS)와 명령어 포인터(IP) 및 플래그 레지스터(PSW)를 스택에서 POP(인출)한다. 즉, 'INT N' 명령어 다음에 위치한 명령어로 복귀한다.

6.1 MS-DOS 호출 방법

- IBM-PC에서는 'INT N' 명령어를 사용하여 BIOS에서 제공하는 입출력 기능을 사용하거나 MS-DOS를 호출한다. MS-DOS는 'INT 21H'로 호출한다. MS-DOS는 많은 기능(function)을 제공하는데, AH 레지스터에 기능 번호를 지정하여 호출한다.
- 예를 들어 키보드에서 문자를 입력하는 MS-DOS 기능 번호는 '1'이다. MS-DOS를 호출하여 키보드로부터 문자를 입력받는 프로그램은 다음과 같다.
- MOV AH, 01H ; 키보드 입력 기능 번호를 ah에 저장
- INT 21H ; MS-DOS 호출.
- 위의 예제는 표준 입력장치(키보드)로부터 입력이 있을 때까지 기다린 후 데이터가 입력되면 입력 문자를 표준 출력장치(모니터)에 표시하고 AL에 저장한다.

6.2 주요 MS-DOS 기능

▶ 주요 MS-DOS 기능

- 화면출력: 기능번호 02H.
- 기능: DL 레지스터에 있는 ASCII 문자를 표준 출력장치(화면)로 보낸다.
- 입력 : AH 레지스터에 기능번호 02h를 입력, 화면에 출력할 8비트 아스키 코드 값을 DL 레지스터에 저장한다.
- 출력 : DL 레지스터에 저장된 한 문자를 화면에 출력한다. 레지스터로 리턴 되는 값은 없다.
- 사용 예 :
 - `mov dl, 'A'` ; 화면에 'A' 문자 출력
 - `mov ah, 02h`
 - `int 21h.`

6.2 주요 MS-DOS 기능

- 프린터 출력: 기능번호 05H.
- 기능: 한 문자를 프린터에 출력한다.
- 입력 : AH 레지스터에 기능번호 05h를 입력, 프린터로 출력할 8비트 아스키 코드 값을 DL 레지스터에 저장한다.
- 출력 : DL 레지스터에 저장된 한 문자를 프린터로 출력한다. 레지스터로 리턴 되는 값은 없다.
- 사용 예 :
 - `mov dl, 'C' ; C를 프린터로 출력`
 - `mov ah, 05h`
 - `int 21h.`

6.2 주요 MS-DOS 기능

- 키보드 입력: 기능번호 08H.
- 기능: 키보드로부터 한 문자 입력을 기다린 후 입력된 8비트 문자 코드를 AL 레지스터에 리턴 한다. 화면에 입력된 문자의 출력은 없다.
- 입력 : AH 레지스터에 기능번호 08h를 입력.
- 출력 : 화면 출력은 없고, 입력된 8비트 문자 코드는 AL 레지스터로 리턴 된다.
- 사용 예 :
 - `mov ah, 08h`
 - `int 21h.`

6.2 주요 MS-DOS 기능

- 문자열 출력: 기능번호 09H.
- 기능: \$(ASCII 24h)로 끝나는 문자열을 표준출력장치에 출력한다. 출력할 문자열의 주소는 DS:DX에 저장한다.
- 입력 : AH 레지스터에 기능번호 09h를 입력, 출력할 문자열의 세그먼트: 오프셋을 DS:DX 레지스터에 저장한다.
- 출력 : DS:DX에 저장된 문자열을 화면에 출력한다. 레지스터로 리턴 되는 값은 없다.
- 사용 예 :
 - segment data
 - char db "A"
 - prompt db " input : \$"
 - segment code
 - ..start: ...
 - mov ax, data
 - mov ds, ax
 - mov dx, prompt
 - mov ah, 09h
 - int 21h
 - ...

6.2 주요 MS-DOS 기능

- 버퍼된 키보드 데이터 입력: 기능번호 0AH.
- 기능: 키보드로부터 문자를 읽어 지정된 버퍼에 저장한다. 입력버퍼의 주소는 DS:DX이며, 형식은 다음과 같다. 버퍼의 첫 바이트는 버퍼의 크기를 나타내며, 최소 1에서 최대 255 바이트까지 가능하다. 두 번째 바이트는 실제 입력된 문자수이며 세 번째 바이트부터는 실제로 입력된 문자를 저장하는 공간이다. 문자 입력은 CR(0DH: 엔터 키)이 입력될 때까지 버퍼에 입력된다. 그리고 실제로 입력된 문자수에는 CR이 포함되지 않는다. 버퍼가 '최대길이-1'까지 채워졌을 경우 이 다음부터 입력되는 문자들은 CR이 입력될 때까지 무시된다. 입력된 문자들은 백 스페이스로 수정 가능하다.
- 입력 : AH 레지스터에 기능번호 0Ah를 입력, 입력 버퍼의 주소를 DS:DX 레지스터에 저장 한다.
- 출력 : 레지스터로 리턴 되는 값은 없다.

6.2 주요 MS-DOS 기능

- 사용 예:
- `segment data`
- `cin resb 25 ; 라벨 정의. 입력될 최대 문자수`
- `ach resb 0 ; 실제로 입력된 문자수`
- `rch resb 25 ; 실제로 입력될 문자 배열`
- `segment code`
- `..start:`
- `mov ax, data`
- `mov ds, ax`
- `mov dx, cin`
- `mov ah, 0ah`
- `int 21h`
- `...`
- 실제로 위 코드에서 "string"을 키보드로 입력하면 버퍼는 다음과 같다.

19h	0	s	t	r	i	n	g	0dh
-----	---	---	---	---	---	---	---	-----	-----	-----	-----	-----	-----

6.2 주요 MS-DOS 기능

- 날짜 알아내기: 기능번호 2AH.
- 기능: DOS의 현재 날짜에 대해 알려 준다.
- 입력 : AH 레지스터에 기능번호 2AH를 입력.
- 출력 : CX 레지스터에 년도 출력(1980 - 2099까지), DH 레지스터에 월이 출력(1 - 12까지), DL 레지스터에 일이 출력(1 - 31까지), AL 레지스터에 요일이 출력(0 - 6까지)된다.
- 사용 예 :
- segment data
- year dw 0 ; 연도 변수
- month db 0 ; 월 변수
- day db 0 ; 일 변수
- week db 0 ; 요일 변수
- segment code
- ..start:
- mov ah, 2ah
- int 21h
- mov [year], cx ; 2004년이면, 07d4h가 year에 복사
- mov [month], dh ; 01h ~ 0ch까지
- mov [day], dl ; 01h ~ 1fh까지
- mov [week], al ; 0 = 일, 1 = 월, ... , 6 = 토

...

6.2 주요 MS-DOS 기능

- 날짜 설정: 기능번호 2BH.
- 기능: DOS에 날짜를 설정한다.
- 입력 : AH 레지스터에 기능번호 2BH를 입력, CX 레지스터에 년도 (1980 - 2099)를 입력, DH 레지스터에 월(1 -12)을 입력, DL 레지스터에 일(1 - 31)을 입력한다.
- 출력 : AL 레지스터에 유효한 날짜이면 00H(valid date)를, 부적합한 날짜 지정이면 FFH를 리턴 한다.
- 사용 예 :
- `mov ah, 2bh`
- `mov cx, 07d4h ; 년도 설정`
- `mov dh, 09h ; 월 설정`
- `mov dl, 17h ; 일 설정`
- `int 21h`

6.2 주요 MS-DOS 기능

- 시간 알아내기: 기능번호 2CH.
- 기능: DOS의 현재 시간을 알려준다.
- 입력 : AH 레지스터에 기능번호 2CH를 입력.
- 출력 : CH 레지스터에 시간을 출력(0 - 23까지), CL 레지스터에 분을 출력(0 - 59까지), DH 레지스터에 초를 출력(0 - 59까지), DL 레지스터에 1/100초 출력(0 - 99까지)한다. 시스템에 따라 1/100초는 사용하지 않는 경우도 있으며, 이 때 DL 레지스터는 0 이 리턴 된다.
- 사용 예 :
- `mov ah, 2ch`
- `int 21h 21h`

6.2 주요 MS-DOS 기능

- 시간 설정: 기능번호 2DH.
- 기능: DOS에 시간을 설정한다.
- 입력 : AH 레지스터에 기능번호 2DH를 입력, CH 레지스터에 시간을 입력, CL 레지스터에 분을 입력, DH 레지스터에 초를 입력, DL 레지스터에 1/100초를 입력한다.
- 출력 : AL 레지스터에 유효한 시간이면 00H(valid time)를 부적합한 시간 지정이면 FFH를 리턴 한다.

◦ 사용 예 :

```
◦ segment      data
◦ hour          db          14
◦ min           db          13
◦ sec           db          33
◦ hsec          db          0
◦ segment      code
◦ ..start
◦ mov          ch,          [hour]
◦ mov          cl,          [min]
◦ mov          dh,          [sec]
◦ mov          dl,          [hsec]
◦ mov          ah,          2dh
◦ int          21h
```

6.2 주요 MS-DOS 기능

- 메모리 할당: 기능번호 48H.
- 기능: 처리 프로그램에 지정된 패러그래프 양의 메모리를 할당한다.
- 입력 : AH 레지스터에 기능번호 48H를 입력, BX 레지스터에 할당할 메모리 패러그래프 수를 입력한다.
- 출력 : 에러가 없으면 AX 레지스터에 할당된 메모리의 세그먼트 주소가 리턴, 에러가 있으면 carry flag가 1로 셋팅 되고, AX 레지스터에 에러 코드(07H = 메모리 제어블록이 파괴, 08H = 메모리 부족)와 BX 레지스터에 사용 가능한 최대 메모리 패러그래프 수가 리턴 된다.
- 사용 예 :
 - `mov bx, (바이트_수 + 15) ; 할당 받을 바이트 수`
 - `shr bx, 4`
 - `mov ah, 48h`
 - `int 21h`

6.2 주요 MS-DOS 기능

- 할당된 메모리 회수: 기능번호 49H.
- 기능: 할당된 메모리를 회수하여 이용 가능영역으로 한다.
- 입력: AH 레지스터에 기능번호 49H를 입력, ES 레지스터에 회수할 메모리의 세그먼트 주소를 입력 한다.
- 출력 : 에러가 없으면 레지스터에 리턴 값이 없고, 에러가 있으면 carry flag가 1로 셋팅 되며, AX 레지스터에 에러 코드(07H = 메모리 제어 블록 파괴, 09H = ES로 지정된 메모리는 할당되지 않았음)가 리턴 된다.
- 사용 예 :
 - mov ax, segaddr ; 세그먼트 주소를 ax에 저장
 - mov es, ax
 - mov ah, 49h
 - int 21h

6.2 주요 MS-DOS 기능

- 처리 종료: 기능번호 4CH.
- 기능: 처리를 종료하고 DOS로 복귀한다.
- 입력: AH 레지스터에 기능번호 4CH를 입력, AL 레지스터에 복귀 코드(00H = 정상적인 종료, 01H = Control-C에 의한 종료, 02H = 장치에러에 의한 종료, 04H = 기능번호 31H에 의한 종료)를 입력 한다.
- 출력 : 레지스터에 리턴 되는 값은 없다.
- 사용 예 :
- `mov ax, 4c00h`
- `int 21h`

6.2 주요 MS-DOS 기능

- PSP 알아내기: 기능번호 62H.
- 기능: 현재 처리중인 프로그램의 PSP(Program Segment Prefix)를 알려 준다.
- 입력: AH 레지스터에 기능번호 62H를 입력.
- 출력 : BX 레지스터에 PSP의 세그먼트 주소를 리턴 한다.
- 사용 예 :
- `mov ah, 62h`
- `int 21h`