

시스템프로그래밍 HW2

20101432 조운지

주제 : 시험 성적 계산하기

- 32bit x86 어셈블리어 프로그램
 - 자신의 학번과 이름을 화면에 출력한 뒤 프로그램을 시작
 - 시험점수를 정수로 입력받아, 그에 해당하는 성적등급을 화면에 출력
 - 시험점수를 입력하고 성적등급을 출력하는 과정을 반복적으로 수행
 - 음수 입력시 반복을 마치고 프로그램 종료
 - 잘못된 입력에 대해서는 오류메시지 출력
 - MASM과 Irvine32 라이브러리를 이용하여 프로그램 작성 및 어셈블링 수행
- 프로그램 구조 및 소스 코드에 대한 설명

```
1  INCLUDE C:\Irvine32\Irvine32.inc    ;사용할 라이브러리 추가
2
3  .386      ; this as a 32-bit program
4  .model flat, stdcall    ; program's memory model (flat), calling convention (named stdcall) for procedures
5  .stack 4096    ; 4096 bytes of storage for the runtime stack
6
7  .data
8  MyInfo BYTE "20101432 조운지", 0    ; 학번과 이름을 저장한 null-terminated string
9  InputScore SBYTE "성적을 입력하세요 (0~100 사이의 정수) : ", 0
10 Maxscore SBYTE 100;
11 scoreLimitArray SBYTE 90, 80, 70, 60
12 scoreLimitArraySize = ($-scoreLimitArray)/TYPE scoreLimitArray;
13 gradeArray BYTE "A", "B", "C", "D", "F"
14 grade BYTE ?
15
16 PrintGrade BYTE "학점 : ", 0
17 PrintErrorMsg BYTE "범위를 벗어났습니다.", 0
18
```

(1) 자신의 학번과 이름을 화면에 출력

```
20  .code
21
22  main PROC    ; 메인 프로시저 시작
23
24  call Cclrscr ; 라이브러리에서 지원하는 프로시저. 콘솔창을 clear
25
26  mov edx, OFFSET MyInfo
27  call WriteString    ; "20101432 조운지"를 출력
28
29
```

다음 과정을 반복적으로 수행,

(2-1) 시험점수를 정수로 입력받음. 음수 입력시 반복을 마치고 프로그램 종료하기 위해 @NegativeInput으로 이동

```
30  | @beginwhile:
31
32  | call Crlf
33  | call Crlf
34
35  | mov edx, OFFSET InputScore
36  | call WriteString ; "성적을 입력하세요(0~100 사이의 정수) :" 출력
37
38  | call ReadInt ; 점수 입력받아서 al 레지스터에 저장
39  | test al, 80h ; 점수가 음수라면
40  | jnz @NegativeInput ; @NegativeInput(exit 바로 전 줄)로 감.
41
```

(2-2) 잘못된 입력에 대해서는 오류메시지 출력, 올바른 입력이면 @VaildInput으로 이동

```
42  | cmp al, Maxscore ; 점수와 100을 비교
43  | jle @VaildInput ; 점수가 100보다 작거나 같으면 @VaildInput으로 이동
44  | mov edx, OFFSET PrintErrorMsg
45  | call WriteString ; 아니면 에러메시지 "범위를 벗어났습니다." 출력하고
46  | jmp @beginwhile ; 다음 while문 시작으로 감
```

(2-3) 입력에 해당하는 성적 등급을 화면에 출력

@Grade : gradeArray에서 해당하는 문자(A, B, C, D, F)를 출력하고 while문 시작으로 돌아간다.

@Compare : 입력을 scoreLimitArray에 있는 숫자(90, 80, 70, 60)와 비교하여 만약 해당 숫자보다 입력한 점수가 크거나 같으면 해당 등급(A, B, C, D)을 출력하는 @Grade로 이동한다. 그렇지 않으면 esi 값(배열의 인덱스이자 count와 비교하는 값) 1 증가하여 L1으로 돌아간다.

L1에서 esi와 ecx가 같으면 @Grade로 점프하여 gradeArray의 마지막 값인 F가 출력된다.

```
48 @VaildInput:
49 mov esi,0;
50 mov ecx, scoreLimitArraySize
51 jmp L1
52
53 L1:
54 cmp esi, ecx
55 jl @Compare
56 jmp @Grade
57
58 @Compare:
59 cmp al, scoreLimitArray[esi]
60 jge @Grade
61 inc esi
62 jmp L1
63
64 @Grade:
65 mov edx, OFFSET PrintGrade
66 call WriteString
67 mov al, gradeArray[esi]
68 call WriteChar
69 jmp @beginwhile
```

(3) 프로그램 종료

```
71 @NegativeInput:
72
73 exit ; 라이브러리에서 지원하는 프로시저. invoke ExitProcess, 0 대신 사용
74 main ENDP ; 메인 프로시저 종료
75 END main ; 프로그램 종료
```

- 프로그램 실행 결과 화면 첨부

*이어지는 화면입니다.



The screenshot shows the Microsoft Visual Studio debugger console with the following text:

```
Microsoft Visual Studio 디버그 × + v
20101432 조윤지
성적을 입력하세요(0~100 사이의 정수) : 120
범위를 벗어났습니다.
성적을 입력하세요(0~100 사이의 정수) : 100
학점 : A
성적을 입력하세요(0~100 사이의 정수) : 90
학점 : A
성적을 입력하세요(0~100 사이의 정수) : 80
학점 : B
성적을 입력하세요(0~100 사이의 정수) : 70
학점 : C
성적을 입력하세요(0~100 사이의 정수) : 60
학점 : D
성적을 입력하세요(0~100 사이의 정수) : 50
학점 : F
성적을 입력하세요(0~100 사이의 정수) : 5
학점 : F
성적을 입력하세요(0~100 사이의 정수) : 59
학점 : F
성적을 입력하세요(0~100 사이의 정수) : 69
학점 : D
성적을 입력하세요(0~100 사이의 정수) : 79
학점 : C
성적을 입력하세요(0~100 사이의 정수) : 89
학점 : B
성적을 입력하세요(0~100 사이의 정수) : 99
학점 : A
성적을 입력하세요(0~100 사이의 정수) : -20
C:\Users\USER\source\repos\MASM_TEST\Debug\
디버깅이 중지될 때 콘솔을 자동으로 닫으려
하도록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요...
```