

< 과제 6 >

2017253041_홍성우

1.

The screenshot shows a debugger window with assembly code on the left and a register window at the bottom. The assembly code is as follows:

```
1 include print.inc
2
3 .data
4 var dword ?
5
6 .code
7 main proc
8     mov var, 10
9     push 2
10    push 3
11    push var
12    call addmul
13    mov var, eax
14    call writehex
15
16    mov eax, 0 경과 시간 1ms 이하
17    call ExitProcess
18 main endp
19
20 addmul proc
21     push ebp
22     mov ebp, esp
23     sub esp, 8
24     mov eax, [ebp+8]
25     mul dword ptr [ebp+12]
26     mov dword ptr [ebp-8], eax
27     add eax, [ebp+16]
28     mov dword ptr [ebp-4], eax
29     mov esp, ebp
30     pop ebp
31     ret 12
32 addmul endp
33
34 end main
```

The register window at the bottom shows the following values:

레지스터	값
EAX	00000020
EBX	0024F000
ECX	0040101E
EDX	00000000
ESI	0040101E
EDI	0040101E
EIP	00401083
ESP	0019FF74
EBP	0019FF80
EFL	00000202

-전역변수 var를 ?로 설정해주고 C언어 프로그램에 따라 10을 대입하고 차례대로 push해준다. addmul을 call해주고 stack parameter와 local variable에 맞게 분류해서 C언어 프로그램대로 코딩해준다. 마지막엔 stack parameter를 callee에서 ret 12로 cleanup 해주고 ret을 해서 eax값을 var에 넣어주고 call writehex로 eax값을 출력해준다.

2.

(1)

```
1 include print.inc
2 .data
3 sum dword ?
4
5 .code
6 main proc
7     push 6
8     push 5
9     call addtwo
10    add esp, 8
11    mov sum, eax
12
13    call DumpRegs
14    mov eax, 0
15    call ExitProcess
16 main endp
17
18 addtwo proc
19     push ebp
20     mov ebp, esp
21     mov eax, [ebp+12] ; y
22     add eax, [ebp+8] ; x
23     pop ebp
24     ret
25 addtwo endp
26 end main
27
```

Microsoft Visual Studio 디버그 콘솔

EAX=0000000B EBX=00207000 ECX=00401019 EDX=00401019
ESI=00401019 EDI=00401019 EBP=0019FF80 ESP=0019FF74
EIP=00401076 EFL=00000216 CF=0 SF=0 ZF=0 OF=0 AF=1 PF=1

C:\Users\swsyj\source\repos\System Programming\lab3\Debug\lab3.exe (05개).
이 창을 닫으려면 아무 키나 누르세요...

(2)

```
main.asm addtwo.asm
1 include print.inc
2
3 addtwo proto
4
5 .data
6 sum dword ?
7
8 .code
9 main proc
10    push 6
11    push 5
12    call addtwo
13    add esp, 8
14    mov sum, eax
15
16    call DumpRegs
17    mov eax, 0
18    call ExitProcess
19 main endp
20
21 end main
22
```

Microsoft Visual Studio 디버그 콘솔

EAX=0000000B EBX=00234000 ECX=00401019 EDX=00401019
ESI=00401019 EDI=00401019 EBP=0019FF80 ESP=0019FF74
EIP=00401086 EFL=00000216 CF=0 SF=0 ZF=0 OF=0 AF=1 PF=1

C:\Users\swsyj\source\repos\System Programming\lab3\Debug\lab3.exe (05개).
이 창을 닫으려면 아무 키나 누르세요...

(3)

```
main.asm  addtwo.asm
1  include print.inc
2
3  ; addtwo proto
4
5  .data
6  sum dword ?
7

빌드 시작...
1>----- 빌드 시작: 프로젝트: lab3, 구성: Debug Win32 --
1>Assembling main.asm...
1>main.asm(12): error A2006: undefined symbol : addtwo
1>C:\Program Files (x86)\Microsoft Visual Studio\2019\Co
1>"lab3.vcxproj" 프로젝트를 빌드했습니다. - 실패
```

(4)

```
main2.asm  addtwo2.asm
1  include print.inc
2
3  addtwo proto
4
5  .data
6  sum dword ?
7
8  .code
9  main proc
10     push 6
11     push 5
12     call addtwo
13     ; add esp, 8
14     mov sum, eax
15
16     call DumpRegs
17     mov eax, 0
18     call ExitProcess
19 main endp
20 end main
21
```

Microsoft Visual Studio 디버그 콘솔

EAX=0000000B EBX=00259000 ECX=00401019 EDX=00401019
ESI=00401019 EDI=00401019 EBP=0019FF80 ESP=0019FF74
EIP=00401083 EFL=00000202 CF=0 SF=0 ZF=0 OF=0 AF=0 PF=0

C:\Users\swsyj\source\repos\System Programming\lab3\Debug\lab3.exe
(05개).
이 창을 닫으려면 아무 키나 누르세요...

-앞에서 작성한 프로그램과 다른 점은 stack parameter를 addtwo.asm에서는 caller가 add esp, 8로 clean up 해줬지만, addtwo2.asm에서는 callee가 ret 8로 clean up 해줬다.



3.

```
C:\Users\swsyj\source\repos\System Programming\lab3>notepad sum.c
C:\Users\swsyj\source\repos\System Programming\lab3>cl sum.c
x86용 Microsoft (R) C/C++ 최적화 컴파일러 버전 19.29.30146
Copyright (c) Microsoft Corporation. All rights reserved.

sum.c
Microsoft (R) Incremental Linker Version 14.29.30146.0
Copyright (C) Microsoft Corporation. All rights reserved.

/out:sum.exe
sum.obj

C:\Users\swsyj\source\repos\System Programming\lab3>cl /FA sum.c
x86용 Microsoft (R) C/C++ 최적화 컴파일러 버전 19.29.30146
Copyright (c) Microsoft Corporation. All rights reserved.

sum.c
Microsoft (R) Incremental Linker Version 14.29.30146.0
Copyright (C) Microsoft Corporation. All rights reserved.

/out:sum.exe
sum.obj
```

-VS 파일에 sum.c를 만들고 C컴파일러 옵션을 조절하여 sum.asm을 생성함



```

1 ; Listing generated by Microsoft (R) Optimizing Compiler Version 19.29.30146.0
2
3     TITLE C:\Users\swsyj\source\repos\System Programming\lab3\sum.obj
4     .686P
5     .XMM
6     include listing.inc
7     .model    flat
8
9     INCLUDELIB LIBCMT
10    INCLUDELIB OLDNAMES
11
12    PUBLIC    __local_stdio_printf_options
13    PUBLIC    _vfprintf_l
14    PUBLIC    _printf
15    PUBLIC    _sum
16    PUBLIC    _main
17    EXTRN    __acrt_iob_func:PROC
18    EXTRN    __stdio_common_vfprintf:PROC
19    _DATA    SEGMENT
20    COMM     ?_OptionsStorage@?1??_local_stdio_printf_options@@9@9:QWORD
21    _DATA    ENDS
22    _DATA    SEGMENT
23    $SG9253 DB '%d', 0aH, 00H
24    _DATA    ENDS
25    ; Function compile flags: /Odtp
26    _TEXT SEGMENT
27    _s$ = -4                ; size = 4
28    _main PROC
29    ; File C:\Users\swsyj\source\repos\System Programming\lab3\sum.c
30    ; Line 6
31        push    ebp
32        mov     ebp, esp
33        push    ecx
34    ; Line 9
35        push    10          ; 0000000aH
36        call    _sum
37        add     esp, 4
38        mov     DWORD PTR _s$[ebp], eax
39    ; Line 10
40        mov     eax, DWORD PTR _s$[ebp]
41        push    eax
42        push    OFFSET $SG9253
43        call    _printf
44        add     esp, 8

```

```

45 ; Line 12
46     xor     eax, eax
47 ; Line 13
48     mov     esp, ebp
49     pop     ebp
50     ret 0
51 _main ENDP
52 _TEXT ENDS
53 ; Function compile flags: /Odtp
54 _TEXT SEGMENT
55 _sum$ = -8           ; size = 4
56 _i$ = -4             ; size = 4
57 _n$ = 8              ; size = 4
58 _sum PROC
59 ; File C:\Users\swsyj\source\repos\System Programming\lab3\sum.c
60 ; Line 16
61     push    ebp
62     mov     ebp, esp
63     sub     esp, 8
64 ; Line 18
65     mov     DWORD PTR _sum$[ebp], 0
66 ; Line 19
67     mov     DWORD PTR _i$[ebp], 1
68     jmp     SHORT $LN4@sum
69 $LN2@sum:
70     mov     eax, DWORD PTR _i$[ebp]
71     add     eax, 1
72     mov     DWORD PTR _i$[ebp], eax
73 $LN4@sum:
74     mov     ecx, DWORD PTR _i$[ebp]
75     cmp     ecx, DWORD PTR _n$[ebp]
76     jg      SHORT $LN3@sum
77 ; Line 20
78     mov     edx, DWORD PTR _sum$[ebp]
79     add     edx, DWORD PTR _i$[ebp]
80     mov     DWORD PTR _sum$[ebp], edx
81     jmp     SHORT $LN2@sum
82 $LN3@sum:
83 ; Line 21
84     mov     eax, DWORD PTR _sum$[ebp]
85 ; Line 22
86     mov     esp, ebp
87     pop     ebp
88     ret 0

```

```

89  _sum ENDP
90  _TEXT ENDS
91  ; Function compile flags: /Odtp
92  ; COMDAT _printf
93  _TEXT SEGMENT
94  __Result$ = -8          ; size = 4
95  __ArgList$ = -4         ; size = 4
96  __Format$ = 8           ; size = 4
97  _printf PROC           ; COMDAT
98  ; File C:\Program Files (x86)\Windows Kits\10\Wininclude\10.0.18362.0\wucrt\stdio.h
99  ; Line 954
100     push ebp
101     mov  ebp, esp
102     sub  esp, 8
103  ; Line 957
104     lea  eax, DWORD PTR __Format$[ebp+4]
105     mov  DWORD PTR __ArgList$[ebp], eax
106  ; Line 958
107     mov  ecx, DWORD PTR __ArgList$[ebp]
108     push ecx
109     push 0
110     mov  edx, DWORD PTR __Format$[ebp]
111     push edx
112     push 1
113     call ___acrt_iob_func
114     add  esp, 4
115     push eax
116     call _vfprintf_l
117     add  esp, 16          ; 00000010H
118     mov  DWORD PTR __Result$[ebp], eax
119  ; Line 959
120     mov  DWORD PTR __ArgList$[ebp], 0
121  ; Line 960
122     mov  eax, DWORD PTR __Result$[ebp]
123  ; Line 961
124     mov  esp, ebp
125     pop  ebp
126     ret 0
127  _printf ENDP
128  _TEXT ENDS
129  ; Function compile flags: /Odtp
130  ; COMDAT _vfprintf_l
131  _TEXT SEGMENT
132  __Stream$ = 8           ; size = 4

```

```

133 __Format$ = 12          ; size = 4
134 __Locale$ = 16         ; size = 4
135 __ArgList$ = 20        ; size = 4
136 __vfprintf_I PROC      ; COMDAT
137 ; File C:\Program Files (x86)\Windows Kits\10\include\10.0.18362.0\ucrt\stdio.h
138 ; Line 642
139     push ebp
140     mov  ebp, esp
141 ; Line 643
142     mov  eax, DWORD PTR __ArgList$[ebp]
143     push eax
144     mov  ecx, DWORD PTR __Locale$[ebp]
145     push ecx
146     mov  edx, DWORD PTR __Format$[ebp]
147     push edx
148     mov  eax, DWORD PTR __Stream$[ebp]
149     push eax
150     call ___local_stdio_printf_options
151     mov  ecx, DWORD PTR [eax+4]
152     push ecx
153     mov  edx, DWORD PTR [eax]
154     push edx
155     call ___stdio_common_vfprintf
156     add  esp, 24          ; 00000018H
157 ; Line 644
158     pop  ebp
159     ret 0
160 __vfprintf_I ENDP
161 _TEXT ENDS
162 ; Function compile flags: /Odtp
163 ; COMDAT ___local_stdio_printf_options
164 _TEXT SEGMENT
165 ___local_stdio_printf_options PROC      ; COMDAT
166 ; File C:\Program Files (x86)\Windows Kits\10\include\10.0.18362.0\ucrt\corecrt_stdio_config.h
167 ; Line 86
168     push ebp
169     mov  ebp, esp
170 ; Line 88
171     mov  eax, OFFSET ?_OptionsStorage@?1??_local_stdio_printf_options@@@9@9 ; `__local_stdio_printf_options'::?2?_OptionsStorage
172 ; Line 89
173     pop  ebp
174     ret 0
175 ___local_stdio_printf_options ENDP
176 _TEXT ENDS

```

-강의자료 7장 5페이지의 c2asm.asm와 같은 형태로 생성된 걸 볼 수 있다.