**Assembly Language – Procedures (1)**

5-1 External Library, Stack Operation

Objective: Understanding the operation and implementation of Stack

| ; Print String in reverse order  .data  myString BYTE "kcatS"  .code1  main PROC  mov ecx, LENGTHOF myString  mov esi, 0  L1:  movzx eax, myString[esi]  push eax  inc esi  loop L1  L2:  exit  main ENDP  END main | 1. Based on the codes in the left, after the program stops at L1 the first time, the values in the registers are:  | EAX | 75e13c33h | EBP | 0012ff94h | | --- | --- | --- | --- | | EBX | 7ffdf000h | ESP | 0012ff8ch | | ECX | 00000005h | ESI | 00000000h | | EDX | 00401000h | EDI | 00000000h |   b. For question a, try to fill in the data status in Stack when the program is executed to L2.   | EAX | 00000053h | EBP | 0012ff94h | | --- | --- | --- | --- | | EBX | 7ffdf000h | ESP | 0012ff78h | | ECX | 00000000h | ESI | 00000005h | | EDX | 00401000h | EDI | 00000000h |   c. Following Problem b, what are the data currently stored in the stack?   | 0012ff78h | 00000053h | | --- | --- | | 0012ff7ch | 00000074h | | 0012ff80h | 00000061h | | 0012ff84h | 00000063h | | 0012ff88h | 0000006bh | | 0012ff8ch | 75e13c45h |   The table below gives ASCII Codes in hexadecimal   | Character | ASCII Code | | --- | --- | | 'S' | 53h | | 'a' | 61h | | 'c' | 63h | | 'k' | 6bh | | 't' | 74h | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

**Assembly Language – Procedures (2)**

‘;

5-2 Defining and Using Procedures

Objective: Understanding the use of Procedure and its relevant directives.

Procedure "Convert"：This Procedure will store an uppercase letter in ESI. After the invocation, register values will be the same as before.

| .data  ChStr1 BYTE 'A', 'B', 'C'  ChStr2 BYTE 3 dup(?)  .code  Convert PROC USES ax  mov al, [esi]  add al, 20h  mov [edi], al  L2:  ret  Convert ENDP  main PROC  mov esi, OFFSET ChStr1  mov edi, OFFSET ChStr2  L1:   call Convert  L3:  exit  main ENDP  END main | 1. Suppose the values in the registers at L1 are:  | EAX | 75e13c33h | EBP | 0012ff94h | | --- | --- | --- | --- | | EBX | 7ffdf000h | ESP | 0012ff8ch | | ECX | 00000005h | ESI | 00404000h | | EDX | 00401000h | EDI | 00404003h |   At L2 position, what are register values?   | EAX | 75e13c61h | EBP | 0012ff94h | | --- | --- | --- | --- | | EBX | 7ffdf000h | ESP | 0012ff86h | | ECX | 00000005h | ESI | 00404000h | | EDX | 00401000h | EDI | 00404003h |  1. Following Problem a, the stack data are:  | 0012ff86h | 3c33h | | --- | --- | | 0012ff88h | L3 | | 0012ff8ch | 75e13c20h |  1. What are the register values when the program stops at L3?  | | EAX | 75e13c33h | EBP | 0012ff94h | | --- | --- | --- | --- | | EBX | 7ffdf000h | ESP | 0012ff8ch | | ECX | 00000005h | ESI | 00404000h | | EDX | 00401000h | EDI | 00404003h | | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |