**Lab Session 09**



**Advanced Procedures**



**Stack Parameters**

* **Passing by value**

When an argument is passed by value, a copy of the value is pushed on the stack..

**EXAMPLE # 01:**

.data

var1 DWORD 5

var2 DWORD 6

.code

push var2

push var1

call AddTwo

exit

AddTwo PROC

push ebp

mov ebp, esp

mov eax, [ebp + 12]

add eax, [ebp + 8]

pop ebp

ret

AddTwo ENDP

* **Explicit stack parameters**

When stack parameters are referenced with expressions such as [ebp+8], we call them explicit stack parameters.

**EXAMPLE # 02:**

.data

var1 DWORD 5

var2 DWORD 6

y\_param EQU [ebp + 12]

x\_param EQU [ebp+ 8]

.code

push var2

push var1

call AddTwo

exit

AddTwo PROC

push ebp

mov ebp, esp

mov eax, y\_param

add eax, x\_param

pop ebp

ret

AddTwo ENDP

* **Passing by reference**

An argument passed by reference consists of the offset of an object to be passed.

**EXAMPLE # 03:**

.data

count = 10

arr WORD count DUP (?)

.code

push OFFSET arr

push count

call ArrayFill

exit

ArrayFill PROC

push ebp

mov ebp, esp

pushad

mov esi, [ebp + 12]

mov ecx, [ebp + 8]

cmp ecx, 0

je L2

L1:

mov eax, 100h

call RandomRange

mov [esi], ax

add esi, TYPE WORD

loop L1

L2:

popad

pop ebp

ret 8

ArrayFill ENDP

**Local Variables**

In MASM Assembly Language, local variables are created at runtime stack, below the base pointer (EBP).

**EXAMPLE # 05:**

.code

call MySub

exit

MySub PROC

push ebp

mov ebp, esp

sub esp, 8

mov DWORD PTR [ebp - 4], 10 ; first parameter

mov DWORD PTR [ebp - 8], 20 ; second parameter

mov esp, ebp

pop ebp

ret

MySub ENDP

**ENTER & LEAVE Instructions**

Enter instruction automatically creates stack frame for a called Procedure. Leave instruction reverses the effect of enter instruction.

**EXAMPLE # 06:**

.data

var1 DWORD 5

var2 DWORD 6

.code

push var2

push var1

call AddTwo

exit

AddTwo PROC

enter 0, 0

mov eax, [ebp + 12]

add eax, [ebp + 8]

pop ebp

leave

ret

AddTwo ENDP

**LOCAL Directive**

LOCAL directive declares one or more local variables by name, assigning them size attributes.

**EXAMPLE # 07:**

.code

call LocalProc

exit

LocalProc PROC

LOCAL temp : DWORD

mov temp, 5

mov eax, temp

ret

LocalProc ENDP

**Recursive Procedures**

Recursive procedures are those that call themselves to perform some task.

**EXAMPLE # 08:**

.code

mov ecx, 5

mov eax, 0

call CalcSum

L1:

call WriteDec

call crlf

exit

CalcSum PROC

cmp ecx, 0

jz L2

add eax, ecx

dec ecx

call CalcSum

L2:

ret

CalcSum ENDP