CS/EEE/INSTR F241
Lab 10 – Procedures and Stacks

Anubhav Elhence

What are Procedures?

- Procedure is a part of code that can be called from your program in order to make some specific task. Procedures make program more structural and easier to understand. Generally procedure returns to the same point from where it was called.

- The syntax for procedure declaration:

```
name PROC
        ; here goes the code
        ; of the procedure ...
RET
name ENDP
```

- (name - is the procedure name); The same name should be used for both the PROC and ENDP directives! (This is used to check the correct closing of procedures)
- PROC and ENDP are compiler directives, so they are not assembled into any real machine code. The compiler just remembers the address of the procedure.

The Call Instruction

- The CALL instruction is used to call a procedure. The RET instruction is used to return to the operating system. The same instruction is used to return from a procedure (actually, the operating system sees your entire program as a special procedure).

- For example, in the code below, the program calls the procedure m1, performs MOV BX, 5, and proceeds to the next instruction (MOV AX, 2)

week10_c1.asm > ...

```
1   .model tiny
2   .data
3
4   .code
5   .startup
6
7       CALL m1
8       MOV AX,2h
9       RET
10
11      m1 PROC
12          MOV BX,5h
13      RET
14      m1 ENDP
15
16
17  .exit
        6 references
18  end
```

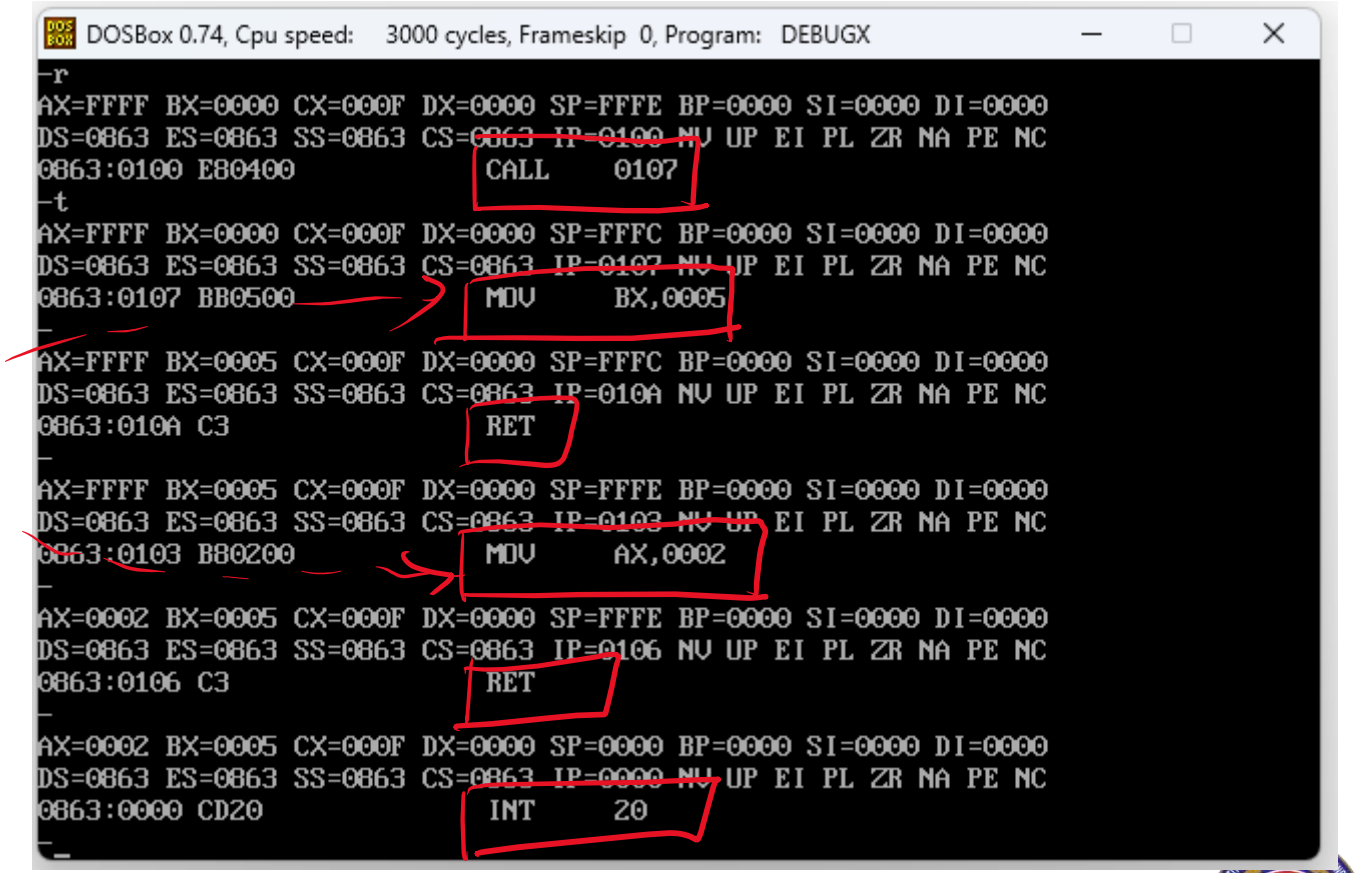

Anubhav Elhence and Dr. Vinay Chamola

The Call Instruction

- For example, in the code below, the program calls the procedure m1, performs MOV BX, 5, and proceeds to the next instruction (MOV AX, 2)

week10_c1.asm > ...

```
1    .model tiny
2    .data
3
4    .code
5    .startup
6
7        CALL m1
8        MOV AX,2h
9        RET
10
11       m1 PROC
12           MOV BX,5h
13       RET
14       m1 ENDP
15
16
17   .exit
     6 references
18   end
```



Passing parameters to Procedures

week10_c2.asm > ...

```
1   .model tiny
2   .data
3
4   .code
5   .startup
6       MOV AL, 1
7       MOV BL, 2
8
9       CALL m2
10      CALL m2
11      CALL m2
12      CALL m2
13      RET                 ; Return to the OS
14
15      m2 PROC
16      MUL BL              ; The product of AL, BL is stored in AX
17      RET                 ; Return to the Caller
18      m2 ENDP
19
20
21  .exit
    6 references
22  end
23
```

- There are several ways to pass parameters to a procedure. The easiest way to pass parameters is by using registers. Here is another example of a procedure that receives two parameters in AL and BL registers, multiplies these parameters, and returns the result in AX register. Since m2 is called four times, the final result in AX will be 2h^4h (10H)

Anubhav Elhence and Dr. Vinay Chamola

Stack

- The Stack is an area of memory for keeping temporary data. The stack is used by the CALL instruction to keep return address for procedure, and the RET instruction gets this value from the stack and returns to that offset.
- This also happens when INT instruction calls an interrupt (Recall INT 21h and INT 10h!). It stores the code segment and offset in the stack flag registe. Similar to RET, the IRET instruction is used to return from interrupt call.

Push and Pop Instr

- The stack is a LIFO data structure (Last In, First Out) can be accessed to store or retrieve data using these two instructions-
- PUSH - stores a 16 bit value (from a register or memory location) in the stack.
- POP - gets 16 bit value from the stack and stores it in a register or a memory location.

Syntax:

PUSH REG ; AX, BX, DI, SI etc.
PUSH SREG ; DS, SS, ES etc.
PUSH memory ; [BX], [BX+SI] etc.
PUSH immediate ; 5. 3Fh, 10001000b etc.

Syntax:

POP REG ; AX, BX, DI, SI etc.
POP SREG ; DS, SS, ES etc.
POP memory ; [BX], [BX+SI] etc.

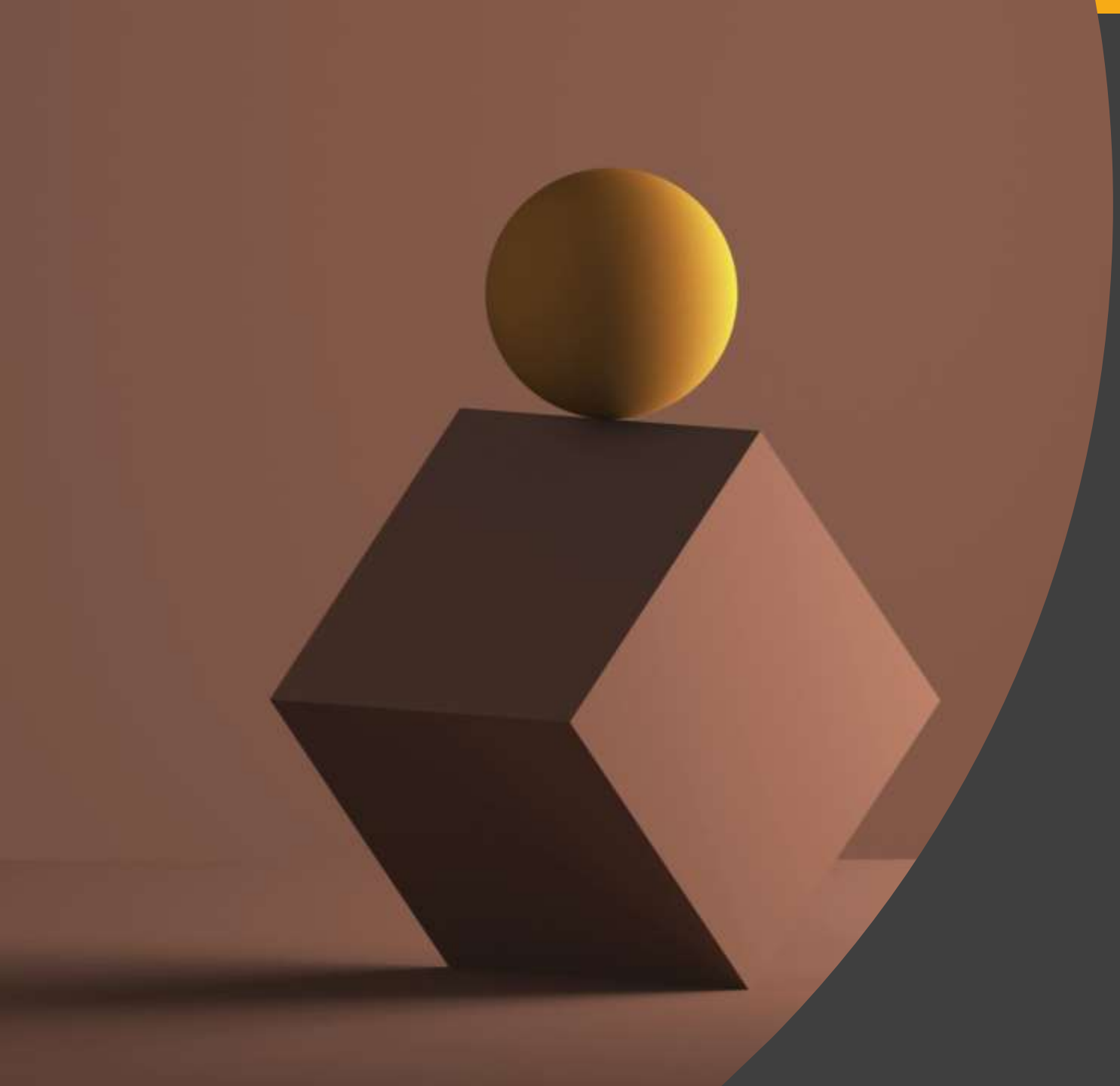
Follow along example

- The following example shows how the stack can be used to swap the values in the registers AX and BX. Notice the order of registers in the pop operation!

wee10_c3.asm > ...

```
1    .model small
2    .data
3
4    .code
5    .startup
6        MOV AX, 1212h
7        MOV BX, 3434h
8
9        PUSH AX
10       PUSH BX
11
12       POP AX
13       POP BX
14
15
16
17   .exit
     4 references
18   end
19
```


Anubhav Elhence and Dr. Vinay Chamola



Thankyou

Anubhav Elhence and Dr. Vinay Chamola