

EE 213 Computer Organization and Assembly Language

Week # 7 Lecture # 15

27th Muharram ul Haram, 1440 A.H

8th October 2018

These slides contains materials taken from various sources. I fully acknowledge all copyrights.

Minds open...



... Laptops closed



**This presentation helps in delivering the lecture.
Take notes, interact and read text book to learn and gain knowledge.**

Today's Topics

- Stack data structure
- Operations on Stack (**Must see lecture slides Lecture # 15b**)
 - Push
 - Pop
 - Top of Stack (ToS)
- Implementing Push and Pop with PUSH and POP instructions
- Procedures and Functions
 - C functions, call and return values, local variables, actual and formal parameters.
 - Assembly PROC – ENDP directive to define procedures
 - CALL instruction
 - Parameter passing using Registers or Stack

PROCEDURES

5.1 Stack Operations

- 5.1.1 Runtime Stack (32-Bit Mode)
- 5.1.2 PUSH and POP Instructions
- 5.1.3 Section Review

5.2 Defining and Using Procedures

- 5.2.1 PROC Directive
- 5.2.2 CALL and RET Instructions
- 5.2.3 Nested Procedure Calls
- 5.2.4 Passing Register Arguments to Procedures
- 5.2.5 Example: Summing an Integer Array
- 5.2.6 Saving and Restoring Registers
- 5.2.7 Section Review

5.3 Linking to an External Library

- 5.3.1 Background Information
- 5.3.2 Section Review

5.4 The Irvine32 Library **Covered in Lab**

- 5.4.1 Motivation for Creating the Library

5.4.2 Overview

5.4.3 Individual Procedure Descriptions

5.4.4 Library Test Programs

5.4.5 Section Review

5.5 ~~64-Bit Assembly Programming~~

~~5.5.1 The Irvine64 Library~~

~~5.5.2 Calling 64-Bit Subroutines~~

~~5.5.3 The x64 Calling Convention~~

~~5.5.4 Sample Program that Calls a Procedure~~

5.6 Chapter Summary

5.7 Key Terms

5.7.1 Terms

5.7.2 Instructions, Operators, and Directives

5.8 Review Questions and Exercises

5.8.1 Short Answer

5.8.2 Algorithm Workbench

5.9 Programming Exercises

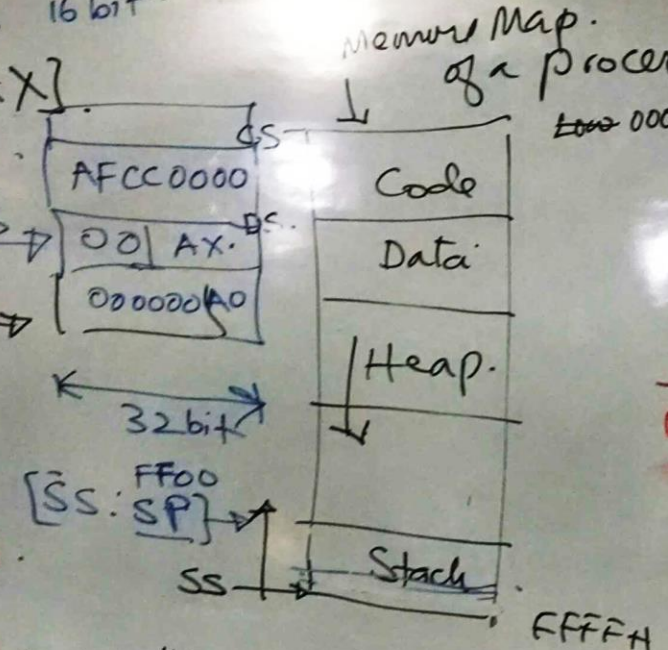
Queue
FIFO

- ① PUSH EAX - 32 bit
- ② PUSH AX - 16 bit
- ③ PUSH [EAX]

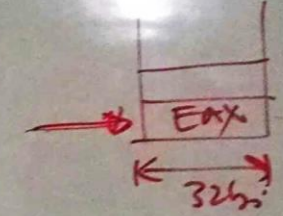
POP
POP ECX
POP EBX
POP EAX

LIFO

- 1 PUSH an item on top stack (TOS).
- 2 POP an item from top of stack



PUSH TOP of Stack = TOS
POP TOS = TOP + 4



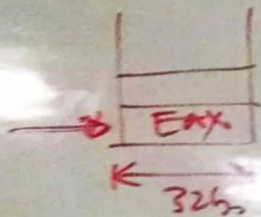
; PUSH EAX
MOV [ESP], EAX ; MOV EAX -> TOS
ADD ESP, -4 ; update TOS

; POP EDX

; ADD ESP, 4 ; Point TOS to element
MOV EDX, [ESP]

TOS points to empty element.
 " " " last push element.

PUSH: TOP of Stack = TOS - 4.
 POP: TOS = TOP + 4.



EAX

PUSH FD.
 POP FD.
 PUSH AD.
 POP AD.
 PUSH A.
 POP A.

32 bits.

16 bits

POP
 PUSH 1010h.

MOV EAX, F300H
 PUSH EAX.
 POP ED.

; PUSH EAX.

MOV [ESP], EAX ; MOV EAX → TOS.

ADD ESP, -4 ; update TOS.

FFFFH ; POP EDX.

; ADD ESP, 4 ; Point TOS to element.

MOV EDX, [ESP].