

NAME :- FAYYAZ MUHAMMAD

ID :- 14294

SUBJECT :- Microprocessor and Assembly language

Department :- B.S (CS)

Teacher :- Sir Amin

ASSIGNMENT Number

05

ASSIGNMENT NO 5

Question No 1

Q. Which register (in 32-bit mode) manages the stack?

A) Extended Stack pointer

Q. Why is the Stack called a LIFO Structure?

Ans) It is called LIFO Structure because it contains set of memory blocks, in which data is retrieved in order i.e. the last value pushed into the stack is the first value popped out from the stack.

Q. When a 32-bit value is pushed on the stack what happens to ESP?

A) ESP is decremented by 4

Q. What would happen if the RET instruction was omitted from a procedure?

Ans) Execution would continue beyond the end of the procedure, possibly into the beginning of another procedure.

Q. How are the words Receive and Return used in the Suggest procedure documentation?

Ans) A list of input parameters and their usage, labeled by a word.

Such as returns

Q) Which procedure in the link library generates a random integer within a selected range?

A) Random Range procedure.

Q) Which procedure in the library display press (Enter) to continue and waits for the user to press the Enter key?

A) Wait Key procedure.

Q) Write statement that cause a program to pause for 700 milli seconds.

A) Code example mov eax, 700 call Delay

Q) Which procedure from the link library write an unsigned integer to the console window in decimal format?

A) Write Dec procedure.

Q) Which procedure from the link library place the cursor at a specific console window location?

A) Gotoxy

a) What are the required input parameter for the DumpMem procedure?

b) Imagine two possible way of calling the DumpMem procedure.

Required input parameter:

Pushed

```

mov esi, OFFSET array
mov ecx, LENGTHOF array
mov ebx, TYPE array
call DumpMem
Poped
    
```

OR

```

push OFFSET array
push LENGTHOF array
push TYPE array
call DumpMem
    
```

a) What are the required input parameter for the readString procedure?

b) EDI contains the OFFSET of an array of bytes, and ECX contains the maximum number of character to read.

Q. Which procedure in the link library generates a random integer within a selected range.

A. Random Range.

Q. Write a sequence of statement that, use only PUSH use only POP instruction to exchange the value in the EAX and EBX register.

A. A sequence of statement are push ebx; Assume $EBX = x$ and $EAX = y$. here the content of EBX (i.e. x) is pushed push eax, which is assumed to be y pop ebx; y from stack is assigned to EBX, therefore $EBX = y$.

pop eax; x from stack is assigned to EAX therefore $EAX = x$

Q. Create a procedure that generate a random string of length L , containing all capital letter; when calling the procedure pass the value of L in EAX and pass a pointer to an array of byte that will hold the random string. write a test program that call your procedure 20 times and display the strings in the console window.

Program:-

Random strings

```
INCLUDE Irvine32.inc  
TAB = 9 ; ASCII code for Tab  
strLen = 10 ; length of the string
```

• 386

• model flat stdcall

• stack 4096

Exit process PROC, dw ExitCode: DWORD

• data

str1 BYTE "The 20 random strings
are:", 0

• code

main PROC

mov edx, OFFSET str1 ; "The 20 random
strings are"

call WriteString

; writes string

call CRLF

; writes an end-of-
line sequence to
the console window

mov ecx, 20

; create 20 strings

L1: mov edx, OFFSET str1

```

mov eax, str_len      ; EAX: String length
call RandomString     ; EAX: String length
call Display          ; generates the random
                      ; string

```

```

mov al, TAB
call writechar
exit
main ENDP             ; leave a tab space

```

```

RandomString PROC uses eax esi
mov ecx, eax          ; ECX = string length

```

```

L1: mov ecx, 26
    call RandomRange
    add ecx, 65      ; EAX gets ASCII of a
                    ; capital letter
    mov arr[esi], eax

```

```

inc esi
loop L1

```

```

RandomString ENDP

```

```

Display PROC uses eax esi ; Display the
                          ; generate random
                          ; string
mov ecx, eax              ; ECX = string length

```

```

L1: mov eax, arr[esi]
    call writechar      ; EAX = ASCII value
    inc esi             ; write the letter
    loop L1
Display ENDP

```

call dumpregs

INVOKE Exit Procces, 0

END main

Q Write a program that displays a single character at 100 random screen location using a timing delay of 100 milliseconds. Hint: use the GetMax X Y procedure of determine the current size of the console window.

Ans) Title Random characters (source.CPP)

// This program display a single character at 100 random screen locations.

Include Irvine 32.inc

• data

row WORD? ; // rows variable to hold num of rows

col WORD? ; // cols variable to hold num of columns.

code

main PROC

call clrscr ; // Set cursor top left
mov ecx, 100 ; // loop for 100 for 100 times


```

L1: col GetMax y ;// size console window
    mov rdx, ax ;// return rows
    mov col, dx ;// return columns

    movzx eax, rdx ;// moving rdx eax
    call RandomRange ;// generate integer
    mov dx, eax ;// setting range boundaries

    call Getxy ;// cursor location

    call WriteChar ;// write random character
    mov ecx, 100 ;// time 100 milliseconds
    call delay ;// pause program, 100 milli sec

Loop L1 ; looping
exit
main ENDP

END main
    
```

Q) Write a program that displays a single character in all possible combinations of foreground and background color (16-16-256). The color are numbered from 0 to 15. So you can use a nested loop to generate all possible combinations.

A)

i Color Matrix
 i write a program that displays a single characters in all possible combinations of foreground and background color (16x16=256). The

- : Color are numbered from 0 to 15, so you can use a nested loop to
- : generate all possible combinations.
- : Last. update :

```
INCLUDE Irvine32.inc
```

```
CR = 0DH ; carriage return
LF = 0AH ; line feed
```

- data
 - prompt 1 BYTE "Please type a character:"
 - display DWORD:

• Code

```
main PROC
```

- : Set text color to white text on black background.

: even though these are the default colors.

- : Each color constant is defined in Irvine32.inc

```
mov eax, white + (black * 16)
call SetTextColor
call clrscr ; Clean the Screen
```

- : Get the user to type some character for our display:

```
mov     ecx, OFFSET prompt 1 ; please type a
call   writeString
call   ReadChar
```

- ; One now has the desired character in AL.
- ; but we can only use it letters, so
- ; we save it in variable dispchar:

```
mov     dispchar, eax
call   CUIF ; new line
```

- ; Generating all the possible color combinations
- ; in a nested loop require a bit more work
- ; than simply a single loop.
- ; ECX will be preserved on the stack across the inner loop:

```
mov     ecx; outer loop counter.
```

```
L1: push ecx ; Save outer loop counter.
```

- ; As the text attributes above illustrate
- ; to set colors we need to compute
- ; EAX in a way that combine both attributes
- ; Set EAX using the stack copy of ECX
- ; value so the background will vary
- ; slowest (with the outer loop counter)

```
mov     ecx, 16 ; inner loop counter
```

L2: pop eax ; get the outer loop counter in EAX
 ; Push eax ; Saves it back again
 Sub eax, 1 ; Shift range from 1-16 to 0-15
 Vol eax, 4 ; Set A1 for the background color
 Vol ecx, ecx ; add the inner loop counter of Ecx
 Sub eax, 1 ; Set A1 for the foreground color

call SetText color

; EAX is now available for restoring the display character.

mov eax, dispc
call WriteChar

Loop L2

; Return the cursor to beginning of next row for matrix

call CrIF ; new line
Pop ecx ; restore outer loop counter

Loop L1

; Show the result on screen until user hit enter. then exit.

```
mov eax, white + (black * 16)
call SetTextColor ; otherwise we leave black
on black
```

```
call CRLF ; new line
call waitMsg ; "press [Enter]" .....
```

```
exit
main ENDP
END main
```

Q: Write statements that cause a program to pause for 700 milliseconds.

A: Code example:

```
mov eax, 700
call Delay
```

Q: What will be the final value in EAX after these instructions execute?

```
push 5
push 6
pop eax
pop eax
```

A: final value in EAX after these instructions execution (5)

16

Q16) Which statement is true about what will happen when the example code runs?

A) By looking at this, I think that EAX would still equal 30 at the end of line 6 because eax was just pushed on the stack not a change in value.

Q17) Which statement is true about what will happen when the example code runs?

A) This one I would believe, it would equal 10 because of LIFO (last in first out). EAX would equal 10 because it was the last one in.

Q17) Which statement is true about what will happen when the example code runs?

A) C. EAX will be equal 30 on line

Q18) Which statement is true about what will happen when the example code runs?

A) A. EDI will be equal 40 on line 6.

Q19) Write a sequence of statements that use only PUSH and POP instructions to exchange the values in the EAX and EBX registers.

