CSUS, Computer Science Department, CSC35, Spring 2021, Dr. Ghansah

Homework 5 (SHIFT/ROTATE, STACK FRAMES/ ADVANCED PROCEDURES, STRING INSTRUCTIONS)

Note: Submission of this Homework Assignment is optional. It will not be graded but if you do it, you might be better prepared for exams. I will provide solution to this homework. All exams will be multiple choice on Canvas so you should know your material very well.

The following problems are from your Irvine Textbook Ed 7

Do the following problems from your text Irvine Ed 7

Q1 Sec 7.9.1 Do problems 1,2

Q2 Sec 7.9.2 Do problems 3, 4, 5, 11

Q3. Sec 8.10.1 Do problems 3;

Q4 Sec 8.10.2 Do Problems 1,2

Q5. Sec 9.9.1 Do Problems 1, 2, 12

Q6. Sec 9.9.2 Do Problems 1, 2, 4

SUBMISSION (OPTIONAL): Submit Electronically via *Canvas*. FileName must be according to the format specified in the course syllabus.

SOLUTIONS

7.9.1

1. 0D4h = 1101 0100

a) 1101 0100 → SHR al, 1 → 0110 1010 = 6Ah

b) 1101 0100 → SAR al, 1→1110 1010 = 0EAh

c) 1101 0100 → SAR al, 4 →1111 1101 = 0FDh

d) 1101 0100 → ROL al, 1 → 1010 1001 = 0A9h

2. 0D4h = 1101 0100

a) 1101 0100 → ROR al, 3 → 1001 1010 = 9Ah

b) 1101 0100 → ROL al, 7→0110 1010 = 6Ah

c) First STC is executed so carry flag is 1 initially. 1101 0100 → RCL al, 1 →1010 1001 = 0A9h

d) First STC is executed so carry flag is 1 initially. 1101 0100 → RCL al, 3 → 0011 1010 = 3Ah

7.9.2

3. SHL eax, 4

4. SHR ebx, 2

5. ROR dl, 4

11.

```
 MOV eax, val2
 MUL val3
 MOV ebx, val4
 SUB ebx, 3
 DIV, ebx
 MOV val1, eax
```

8.10.1

3. A number on the RET instruction corresponds to the number of bytes added to the stack pointer to clear parameters from the stack. This happens after returning to the calling procedure.

8.10.2

1. ESP points to EBP and each row above corresponds to +4, +8 … bytes so the return address is EBP + 4 and 10h is EBP + 16

10h

20h
30h
Return address
EBP

2.

```
PROC AddThree
  PUSH ebp
  MOV ebp, esp
  MOV eax, [ebp + 8]
  ADD eax, [ebp + 12]
  ADD eax, [ebp + 16]
  POP ebp
  RET 12
AddThree ENDP
```

9.9.1

1. STD (value = 1)

2. 2 (for 2 bytes in a word)

12. We are starting from index zero and want to move forward in the array with the STOSD instruction so we clear the direction register to make sure we move forward if it happened to be set before.

9.9.2

1. [ebx + esi]

2. val1[ebx + esi]; where val1 is some variable

4.

```
MOV esi, OFFSET sourcew

MOV edi, OFFSET targetw

MOV ecx, LENGTHOF sourcew; assume sourcew and targetw are the same length

CLD

repe CMPSW
```