CMPE 230 Systems Programming
Homework 2 Documentation

Volkan Öztürk & Arda Arslan
2019400033 2020400078

Bogazici University, Engineering Faculty
Computer Engineering Department

In this project we have implemented an A86 assembly language program that will input postfix expression involving hexadecimal quantities and evaluate it, program mainly consists of three parts:

1) **Reading the Input**
2) **Evaluating the Postfix Expression**
3) **Printing the Result**

IMPORTANT REMARKS

- **Given postfix expression is assumed to be syntactically correct.**
- **LF or CF ends Reading and Evaluation processes.**

Reading the Input

Input is read character by character ;

If given character is a hexadecimal digit, input is concatenated to current hexadecimal number (initially 0x0000).

If given character is an operand, initializes the stack pop process and evaluates the expression specified with the operand.

If given character is whitespace, initializes the stack push process.

If given character is CR or LF, ends reading & evaluation processes, moves to printing process.

Concat Procedure:

Shifts the current value by one digit and adds the given hexadecimal digit:

```
CONCAT:                 ; Concatenates hexadecimal numbers.
        MOV CL, AL
        MOV AX, BX
        MUL DX          ; Result stored in DX:AX.
        ADD AL, CL
        MOV BX, AX      ; Data stored in BX.
        JMP INP
```

Evaluating the Postfix Expression

The expression is evaluated by built-in operands:

AND,
OR,
XOR,
ADD,
MUL,
DIV.

Built-in stack is used for binary operations:

Pop two operands from stack,
Push result to stack.

Printing the Result

The result of the given postfix expression is kept in stack after the evaluation process. In this step result is separated in hexadecimal digits and separated digits are pushed back to stack. Then hexadecimal digits are popped from stack, converted to character according to ASCII table and printed to stdout.

```
CONVERSION:             ; Converts hexadecimal number to printable characters.
        MOV AX, BX
        MOV CX, 10h
        DIV CX          ; Remainder in DX, Quotient in AX.
        PUSH DX
        DIV CX
        PUSH DX
        DIV CX
        PUSH DX
        DIV CX
        PUSH DX
        MOV AX, 00
        MOV BX, 00
        MOV CX, 00
        MOV DX, 00
        MOV DL, 0Dh     ; CR character printed to stdout.
        MOV AH, 02
        INT 21h
        MOV DL, 0Ah     ; LF character printed to stdout.
        MOV AH, 02
        INT 21h
        JMP PRINTER
```