CMPE 230 – SYSTEMS PROGRAMMING

Project 1 - Postfix Calculator

Author: **Kaan ŞENPARLAK**
Student ID: **2016401189**

Author: **Ali BATIR**
Student ID: **2015400261**

Introduction

In this project, we will implement an A86 assembly language program that will input postfix expression involving hexadecimal quantities and evaluate it. After evaluation, the result should be output. Here are some example inputs to the program and the outputs generated:

input	output
2 3 + 4 5 + * 2 /	16
A B C + 2 + *	FA
1 2 4 + + FFFF ^	FFF8

The program starts by reading input characters one by one. Depending on the character, the program jumps to various different labels of the code. After an operator character, the program does the operation and pushes the result back in the stack. This is a very efficient process since the input is in postfix notation. After reading the input 'carriage return' the program pops the last result of the last operation and converts it to first hexadecimal and next ASCII character. Finally the program prints the result and exits.

Implementation

Some portions of the code is given here to be explained in a more detailed fashion:

The program starts with a read character function and immediately jumps to control character label (check).

```
start:
    mov ah,01h
    int 21h
    jmp check
```

The program is able to operate 6 operations and below is an example of how they are implemented easily. First, the numbers are popped from the stack, then the operation is done, finally the result is pushed back to be used again in another operation and the program jumps to reading input.

```
multiplication:
    pop cx
    pop ax
    mul cx
    push ax
    jmp start
```

This part checks the input character by its ASCII value and jumps the program to appropriate label for processing the input.

```
check:
     cmp al,020h
     jz blank
     cmp al,039h
     jbe ifnum
notnum:
     cmp al,046h
     jbe ifletter
notletter:
     cmp al,0Dh
     jle finishread
     pop cx
     cmp al,02Bh
     je addition
     cmp al,02Ah
     je multiplication
     cmp al,02Fh
     je integerdivision
     cmp al,05Eh
     je bitxor
     cmp al,07Ch
     je bitor
     cmp al,026h
     je bitand
```

Finishread and backtoHex labels pop the final result and divides the number to its digits in hexadecimal. The digits are pushed to stack in order. Before pushing the digits, a control value is pushed to stack.

```
finishread:
     pop ax
     mov cx,010h
     push 011h
backtoHex:
     mov dx,0
     div cx
     push dx
     cmp ax,0
     jne backtoHex
```

The last part of the program prints the digits from the highest order to the lowest. It checks if the digit is a number or a letter and converts it to ASCII accordingly. After popping every digit from the stack, printHex label checks if the control value is reached and finishes when reached. In the end, all digits hence the result is printed.

```
printHex:
     pop ax
     cmp ax,011h
     je finish
     cmp ax, 09h
     jle print
     add al,07h
print:
     add al,030h
     mov dl,al
     mov ah,02h
     int 21h
     jmp printHex
```