Mehmet Süzer - 2019400213
Harun Reşid Ergen - 2019400141

CMPE230 Project 2 Documentation

First, space of 255 bytes is allocated for the input, and the buffered input read from the terminal is stored in that space. The size of the input is found and "$" is put at the end of the expression as a terminating char.

Second, the expression is read character by character until "$" we put is found.

1) If a digit (0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F) is found, then we update the value of the operand which is being read. The value is calculated in ax.

EXAMPLE:

base=16 in decimal

ax=0

number=123h

1 is read ----> ax=ax*base + 1 = 1h

2 is read ----> ax=ax*base + 2 = 1*16^1 + 2 = 12h

3 is read ----> ax=ax*base + 3 = 1*16^2 + 2*16^1 + 3 = 123h

2) If a space character is found, then we read either an operator or a operand. We check whether an operand is read by looking the variable num_found. If so, then we push the value calculated in ax to the stack and reset ax. If not, we just go to the next character.

3) If an operator is found, since all the operations are binary operations, we pop 2 operands from the stack and do the operation and push the result.

Finally, we move the result to ax, and print it to the terminal with leading zeros if necessary. If input is in the correct format, then the program exits with error code 0.

The procedure print_result converts an integer, which can fit into 16 bits, to a string which is a hexadecimal correspondence of that integer and prints it to the terminal.