

EECS 388: Computer Systems and Assembly Language

Lab 8: Implementing an RPN Calculator Using the Stack

In this lab, you will implement a calculator using Reverse Polish Notation (RPN). Using RPN and the stack, you can do multiple arithmetic operations, and also emulate parenthesis. For an RPN calculator, arithmetic operations are represented by a sequence of numbers, followed by a sequence of operators.

For example,
2 3 -
is equivalent to
3 - 2

Further:
1 2 3 4 * + -
is equivalent to
 $((4 * 3) + 2) - 1$

Generally, the last two numbers are removed from the list, and the first operator is applied to them. Then the result is placed at the end of the list, and the second operator is applied to the new last two items in the list, and so on.

So for the above case: 1 2 3 4 * + -

First, remove 4 and 3 from the list, and compute $4 * 3$. Place the result at the end of the list, and remove the * operator.

The list is now: 1 2 12 + -

Now, remove 12 and 2 from the list, and compute $12 + 2$. Place the result at the end of the list, and remove the + operator.

The list is now: 1 14 -

Now, remove 14 and 1 from the list, and compute $14 - 1$. Place the result at the end of the list, and remove the - operator.

We now have the result: 13

For further discussion of RPN calculators, see: <http://www.calculator.org/rpn.html>

Problem 1:

Write a program to prompt the user for a RPN string, and compute the arithmetic in the manner described above. You may assume that only single-digit hex numbers, and the operators +, -, and * will be entered. No error checking is necessary. The user will signal that they are done entering the string by pressing the return key. Print the result, and continue to prompt the user until they enter a period at the beginning of the line. You can use the OUT1BYT MON12 function to output the numeric result, since OUTA will print the result as an ASCII character.

You may implement the calculator any way you like, but minimally you must store the list of numbers and the arithmetic results on the stack.

Example program run:

Enter RPN string: 1 2 3 4 * + -
Result: 0D

Enter RPN string: 2 1 2 3 - + *
Result: 04

Enter RPN string: .
(program ends).

Demonstrating Your Results:

To demonstrate your results, enter an RPN string with at least four numbers and all three operators. Choose a testing string whose result isn't larger than one byte.

Report Format and Grading:

Following the report format in your syllabus, include the following in your report:

1. Your name, student number, lab project number and title, course number, lab section number, and date.
2. Description of the lab in your own words. What did you learn? If your code did not work in the lab, explain why. (45% of report grade)
3. The source code for your program. Use comments to indicate what changes you made to the program template. (45% of report grade)
4. A short evaluation of the lab. What did you like about the lab? What could be improved? (10% of report grade)