

COL216 A-2 Write-up

Mallika Prabhakar 2019CS50440
Sayam Sethi 2019CS10399

March 2021

Problem Interpretation

Assumptions

Following are the assumptions which have been considered or given while attempting this assignment:

- Numbers are in the range 0–9
- Only valid operators are +, - and *
- We are given a valid postfix expression else we raise an exception

Basic Idea

We have to implement a function which evaluates a given postfix expression. If the input is invalid, we raise an exception. To implement the function, we employ the use of array as a stack. For every operator, we pop out the top two numbers, operate on them and push the result back into the stack until the stack has only one element which is the answer. Number of operators is always one less than the number of integers as input.

Code Explanation

Following is the list of procedures employed along with their functioning:

- **main - fileRead** function is called and the stack initialised.
- **loop** - The next character from file is read and appropriate checks are made (whether it is an operator, a digit or invalid). If it is a digit, it is pushed to our stack.
- **operate** - The character is verified to be one of +, - or * and appropriate operation function called else error thrown.
- **product** - Multiplies the top two elements of stack.

- **sum** - Adds the top two elements of stack.
- **diff** - Subtracts the top two elements of stack.
- **readFile** - File “in” is opened using syscall.
- **readChar** - The next character is read and appropriately modified to let its ASCII value be used.
- **printRes** - The stack size is checked to be 1 (implying a valid postfix) and then the calculated result is displayed on console.
- **print** - Helper function to call “print string” syscall.
- **push** - Stack push operation performed, and if size exceeds max size (10,000) error thrown.
- **pop** - Stack pop operation performed, and if stack empty, invalid postfix expression error thrown.

Testing

We have extensively tested our code broadly on the following test cases

case 1: empty file input

case 2: invalid postfix expression

case 3: ab+

case 4: ab-

case 5: ab*

case 6: random postfix expression

case 7: stack overflow

arithmetic overflow: MIPS throws an “Arithmetic Overflow” exception which requires some modification involving kernel memory addresses hence we did not implement handling it.