Cmpe 230 Project1 – DOCUMENTATION

Ömer Faruk Özdemir 2016400048 –

Yusuf Başpınar - 2014400042

In this project, we implemented an A86 assembly language program that inputs postfix expression involving hexadecimal quantities and evaluates it. Our methods are as follows:

main:

In main function we started by reading an input. Since main function will be called at each new number input is entered, we wrote the main function to check whether the input is

'*'	multiplication
'/'	integer division
'+'	addition
'^'	bitwise xor
'|'	bitwise or
'&'	bitwise and

We initialized cx register to be 0 to count the number of digits in input to use stack accordingly.

read:

In read function we did checking for whether the input is a letter or number

myor:

OR operation implemented as pops two elements and calculate OR operation and push the stack and check exit.

myand:

AND operation implemented as pops two elements and calculate AND operation and push the stack and check exit.

myxor:

XOR operation implemented as pops two elements and calculate XOR operation and push the stack and check exit.

jmpmain:

Implemented to avoid Jump > 128 error

addition:

ADD operation implemented as pops two elements and calculate ADD operation and push the stack and check exit.

division:

DIV operation implemented as pops two elements and calculate DIV operation and push the stack and check exit.

multiplication:

MUL operation implemented as pops two elements and calculate MUL operation and push the stack and check exit.

checkexit:

Reads input if enter is pressed if it is, jump to convert

num:

Substract 48 to reach ascii value of a number. After that checks if there is a character in stack it is a multidigit then jump multidigit.

letter:

Same as num method but substract 55 to reach ascii value

checkchar:

Pushes the num or letter to stack and reads another input if space jump jmpmain else it is multidigit go back to start method read.

multidigit:

First pops element from stack and decrement the counter for stack cx register and multiply ax by 16 to shift left the digit. After thet adds digits and jumps checkchar.

convert:

Pops the final result and divide the number by 16 to have hexadecimal result and push remaining and quotient. Other methods are the loops for printing numbers and letters accordingly. Inıtialize counter to go over 4 digits using bx register.

printnum:

Loops on the number to print numbers. Adds 48 to have real values.

ending:

Ends the program