CMPE230- PROJECT1

ABDULLAH YILDIZ – YUSUF YÜKSEL

2018400291 - 2014400051

HOW IT READS CHARACTER:

Program constantly asks for input character (mov ah,08 int 21h), takes actions according to characters by comparing it to certain ASCII numbers and exits when pressed ENTER.

WHAT IS PRESSED ? :

Reading character means writing ASCII number of the character to AX register. In the READ label, we make comparison of AL with the operators' ASCII numbers (0DH for 'enter' , '47' for '/' division). If there is a match, then we jump to its label , make operation then return jump to READ.

SPACE label:

Space pushes the var1 into STACK and makes var1 and var2 to 0000h.

STACKS :

Stack is used for storing our number. It is not that we enter a digit or letter character they are pushed to stack. When we enter space, the number in var1 is pushed to stack.

When operators are read, we pop 2 number and make the operation. Then, we push it back.

VARIABLES:

VAR1 holds the character sequence
VAR2 holds the character just pressed.

Everytime a digit or letter is pressed, we conserve it in the variable named var2. Then, var1 is multiplied by 10h and var2 is added to var1 . Finally, number is in var1. When SPACE is pressed, the number is complete.

e.g. 1234 : we wrote 123 already . Var1 has 123 . Then, 4 is pressed, var2 is 4. var1 is multiplied with 10h and made 1230 . Then var2 is added to var1 and made it 1234 .

ADDITION

2 numbers are popped from stack, second one is added to first and the result is pushed back to stack.

XOR/OR/AND

We used build-in OR,AND,XOR for this operations. First, we pop 2 numbers to ax and bx. Then we call operation XOR,OR,AND . Result is on AX , then we push it back to stack.

DIVISION

First, we pop 2 numbers to ax and bx. div bx command divides ax by bx and writes the result to ax and remainder to dx . Our result in ax is pushed to stack. Then returns read label.

MULTIPLICATION

First, we pop 2 numbers to ax and bx. mul bx commands multiplies ax by bx then writes result to ax. Our result in ax then is pushed to stack.

EXITING PROGRAM :

When enter is pressed, the number in the stack (the result) is put into a series of operation. We want to display it character by character. To achieve this, we divide result by 10h repeatedly, incrementing CX and pushing the remainder to STACK until number is ZERO.

Then, we pop all the characters and display it. This will occur CX times.

e.g. At the end of operations.Stack has 15. Enter is pressed, popped to ax . 15 is divided by 10h. Ax has 1h and dx has 5h. Dx is pushed to stack. Cx is 1. Compare ax with 0 and continue dividing. Next division makes ax 0,cx 2, dx 1 . Stack has 5 and 1. Proceed to PRINT label. CX times we pop number and display.

LOAD_NEWCHARACTER

First program subtracts 30h from the input because the ASCII code of 0 is 30. Then,according to comparison with 0 it is letter or digit. If it is greater than 0 then JUMP to load_newcharacter_letter. Else, no jump,just proceed.

Hold the value in var2, var1 is multiplied by 10h and var2 is added to var1. Our character sequence is in var1. Var1 is pushed to stack.

e.g. To enter 15 , 1 was in var1, after pressing 5, var1 = 10 and var2 =5 var1+=var2 which is 15

LOAD_NEWCHARACTER_LETTER

ASCII number of ‘A’ is 37H. We subtract 7h more. Conversion to hexadecimal is completed. Hold the value in var2. Then, var1 is multiplied by 10h and var2 is added to var1. Our character sequence is in var1. Var1 is pushed to stack.

e.g. To enter 1A , 1 was in var1, after pressing A, var1 = 10 and var2 =A var1+=var2 which is 1A