CPSC 240: Computer Organization and Assembly Language

Assignment 09, Fall Semester 2023

CWID:\_\_885857847\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Name:\_Kush Patel\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Download the "CPSC-240 Assignment09.docx" document.
2. Design the "function.asm" program, input a value n (n=004 ~ 999) from the keyboard, calculate 1+2+3+…+n, and display the calculation result in the terminal emulator window. The corresponding C/C++ code is as follows:

#begin define print(addr, n)

rax = 1;

rdi = 1;

rsi = addr of string;

rdx = n;

syscall;

#end

#begin define scan(addr, n)

rax = 1;

rdi = 1;

rsi = addr of buffer;

rdx = n;

syscall;

#end

void main() {

char buffer[4];

int n;

int sumN;

char ascii[10];

char msg1[26] = "Input a number (004~999): ";

char msg2[16] = "1 + 2 + 3 +...+ ";

char msg3[4] = " = ";

print(msg1, 26);

scan(buffer, 4);

call toInteger(n, buffer);

call calculate(n, sumN);

call toString(n, ascii);

print(msg2, 16);

print(buffer, 3);

print(msg3, 3);

print(ascii, 6);

}

void toInteger(n, buffer) {

n = atoi(buffer);

}

void calculate(n, sumN) {

for(ecx=0; ecx<=n; ecx++) {

sumN += ecx;

}

}

void toString(sumN, ascii) {

ascii = itoa(sumN);

}

1. Run the "function" file to display the calculation result in the Terminal Emulator window.
2. Insert source code (function.asm) and simulation results (Terminal Emulator window) at the bottom of the document. Write an analysis to verify the simulation results.
3. Save the file in pdf or docx format and submit the pdf file to Canvas before 23:59 pm on 11/23/2023.

Simulation Sample:



[Insert source code of the function.asm here]

%macro print 2

    mov rax, 1

    mov rdi, 1

    mov rsi, %1

    mov rdx, %2

    syscall

%endmacro

%macro scan 2

    mov rax, 0

    mov rdi, 0

    mov rsi, %1

    mov rdx, %2

    syscall

%endmacro

section .bss

buffer resb 4

n resd 1

sumN resd 1

ascii resb 10

section .data

msg1 db "Input a number (004~999): "

msg2 db "1 + 2 + 3 +...+ "

msg3 db " = "

section .text

global \_start

\_start:

    print msg1, 26

    scan buffer, 4

    print msg2, 16

    print buffer, 3

    print msg3, 3

    mov rbx, buffer

    call toInteger

    mov rcx, 0

    mov edi, dword[n]

    call calculate

    mov rcx, 3

    mov edi, dword[sumN]

    call toString

    print ascii, 7

    mov rax, 60

    mov rdi, 0

    syscall

toInteger:

    mov rax, 0

    mov rdi, 10

    mov rsi, 0

next0:

    and byte[rbx+rsi], 0fh

    add al, byte[rbx+rsi]

    adc ah, 0

    cmp rsi, 2

    je skip0

    mul di

skip0:

    inc rsi

    cmp rsi, 3

    jl next0

    mov dword[n], eax

    ret

calculate:

next1:

    add dword[sumN], ecx

    inc ecx

    cmp ecx, edi

    jbe next1

    ret

toString:

    mov eax, dword[sumN]

    mov rcx, 0

    mov ebx, 10

    divideLoop:

    mov edx, 0

    div ebx

    push rdx

    inc rcx

    cmp eax, 0

    jne divideLoop

    mov rbx, ascii

    mov rdi, 0

popLoop:

    pop rax

    add al, "0"

    mov byte [rbx+rdi], al

    inc rdi

    loop popLoop

    mov byte [rbx+rdi], 10

ret

[Insert the simulation results of the function here]

A screenshot of a computer

Description automatically generated

[Insert the simulation result verification here]

A white background with black dots

Description automatically generatedA screenshot of a phone

Description automatically generated