2/20/2021

Todd Lunde

CSIS 492 – Senior Seminar

Dr. Chen

Project: 05 - Initial Code Base

Initial Code Base:*Von Neumann Computer Architecture*

Github repository (05 - Initial Code Base):

<https://github.com/twlunde/CSIS-492---Senior-Seminar---Project-05-Initial-Code-Base.git>

username: twlunde

password: January30:1959

email: [49543794+twlunde@users.noreply.github.com](mailto:49543794+twlunde@users.noreply.github.com)

// Author: Todd Lunde

// File: (IterativeFibonacci.cpp - no multiplication - to run equivalent in VisUAL Arm Assembler)

// Class: CSIS 320 - Architecture

// Lab #4: part #4 (high-level language implementation), exercise #3

// Date: 4/4/2019

// Description:

// This file contains function int main(), function void fibonacci(int number)

#include <iostream>

#include <iomanip>

using namespace std;

void fibonacci(int number)

{

int x = 0, y = 1, z = 0;

for(int i=0; i<=number; i++)

{

cout << x << " ";

z = x + y;

x = y;

y = z;

} // end for

} // end void fibonacci(int num)

int main() // begin program main()

{

int number;

cout << endl << endl;

cout << "Enter the number to output iterative fibonacci sequence for: ";

cin >> number;

cout << "\nThe iterative fibonacci sequence for the number " << number << " is: ";

fibonacci(number);

cout << endl << endl;

return 0;

} // end main()

------------------------------------------------------------------------------------------------------------------------------------------

; Author: Todd Lunde

; File: (IterativeFibonacci - no MUL for multiplication, using all addition)

; Class: CSIS 320 - Architecture

; Lab #4: part #4 (ARM implementation - VisUAL assembly), exercise #3

; Date: 4/4/2019

; ARM Assembly Code

MAIN

; int x = 0, y = 1, z = 0;

MOV R0, #0

MOV R1, #1

MOV R2, #0

BL FIBONACCI ; call function

MOV R4, R0 ; R4 = result

; void fibonacci(int number)

; {

; fibonacci

; void fibonacci(int number)

; {

; int x = 0, y = 1, z = 0;

FIBONACCI

MOV R1, #0 ; int x = 0

MOV R2, #1 ; y = 1

MOV R3, #0 ; z = 0

FOR

CMP R0, #40 ; R0 == i=1 to 10

BEQ DONE ; if (i == R0==10), exit loop

; cout << x << " ";

ADD R3, R1, R2 ; z = x + y;

MOV R1, R2 ; x = y;

MOV R2, R3 ; y = z;

ADD R0, R0, #1 ; i = i + 1, R0 = R0 + 1

B FOR ; repeat loop

MOV PC, LR ; return to caller

DONE

------------------------------------------------------------------------------------------------------------------------------------------

-------------------------------------------------------------------------------------

2/21/2021

[Recursive Fibonacci in MASM Assembly](https://stackoverflow.com/questions/23511377/recursive-fibonacci-in-masm-assembly)

include Irvine32.inc

.code

main PROC

mov ecx,0

push 10 ; calculate the nth fib

call fib ; calculate fib (eax)

add esp, 4 ; clean up the stack

call WriteDec

call Crlf

exit

main ENDP

fib PROC C

add ecx,1

push ebp

mov ebp,esp

sub esp, 4 ; space for a local dword [ebp-4]

mov eax,[ebp+8] ; get n

; if ((n == 1) || (n == 2)) return 1;

cmp eax,2 ; n == 2?

je exception2

cmp eax,1 ; n == 1?

je exception2

;else return fib(n-1) + fib(n-2);

dec eax

push eax ; Fib(n-1)

call fib

mov [ebp-4], eax ; store first result

dec dword ptr [esp] ; (n-1) on the stack -> (n-2)

call fib

add esp, 4 ; clean up stack

add eax, [ebp-4] ; add result and stored first result

jmp Quit

exception2:

mov eax, 1 ; start values: 1, 1

; dec eax ; start values: 0, 1

Quit:

mov esp, ebp ; restore esp

pop ebp ; restore ebp

ret ; return EAX, stack not cleaned up

fib ENDP

END main

------------------------------------------------------------------------------------------------------------------------------------------

// Author: Todd Lunde

// File: (RecursiveFibonacci.cpp)

// Class: CSIS 320 - Architecture

// Lab #4: part #4 (high-level language implementation - C++), exercise #4

// Date: 4/4/2019

// Description:

// This file contains function int main(), function int fibonacci(int number)

#include <iostream>

#include <iomanip>

using namespace std;

int fibonacci(int number)

{

if(number <= 1)

return number;

return fibonacci(number-1) + fibonacci(number-2);

} // end int fibonacci(int n)

int main() // begin program main()

{

int number;

cout << endl << endl;

cout << "Enter the number to output recursive fibonacci sequence for: ";

cin >> number;

cout << "\nThe iterative fibonacci sequence for the number " << number << " is: ";

for(int i=0;i<=number;i++)

cout << fibonacci(i) << " ";

cout << endl << endl;

return 0;

} // end main()