OCTOBER 18, 2017

Tutorial 2

COMPUTER ARCHITECTURE

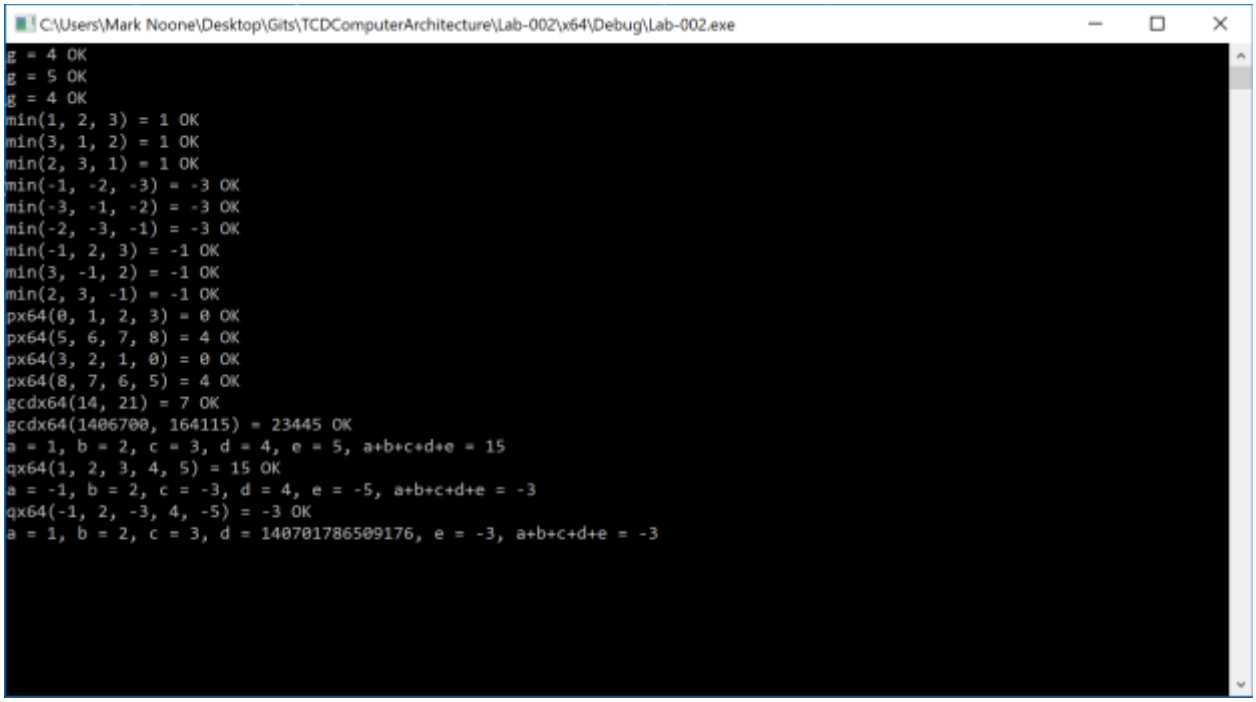
CS3021

MARK NOONE

15319898
MANOONE@TCD.IE

COMPUTER ARCHITECTURE

CONSOLE OUTPUT



T2.ASM

```
option casemap:none
includelib legacy_stdio_definitions.lib
extrn printf:near

.data
    g:
    DD 4

.code

;
;   Tutorial 2 - x64 Min, P and GCD
;
;   Author - Mark Noone
;

public  minx64

minx64: mov     rax, rcx            ; v = a
        cmp     rax, rdx
        jle     min1
        mov     rax, rdx
min1:   cmp     rax, r8
        jle     min2
        mov     rax, r8
min2:   ret
```

```
public  px64

px64:   mov     r10,g
        mov     r12, r8

        mov     r8,  rdx
        mov     rdx, rcx
        mov     rcx, [r10]
        call    minx64

        mov     rcx, rax
        mov     rdx, r12
        mov     r8,  r9
        call    minx64

        ret


public  gcdx64

gcdx64: xor     rax, rax
        cmp     rax, rdx
        jne     gcd1
        mov     rax, rcx
        ret

gcd1:   mov     rax, rcx
        mov     rcx, rdx
        xor     rdx, rdx
        idiv    rcx
        call    gcdx64

        ret


fxp2 db 'a = %I64d, b = %I64d, c = %I64d, d = %I64d, e = %I64d, a+b+c+d+e =
%I64d', 0AH, 00H
public  qx64

qx64:
        xor     r11, r11
        mov     r12, [rsp+40]
        add     r11, r12
        add     r11, r9
        add     r11, r8
        add     r11, rdx
        add     r11, rcx

        push    r11
        push    r12
        push    r9

        mov     r9, r8
        mov     r8, rdx
        mov     rdx, rcx
```

```
        lea     rcx, fxp2
        sub     rsp, 32
        call    printf
        add     rsp, 48
        pop     rax
        ret

public  qnsx64

qnsx64:
        xor     r11, r11
        mov     r12, [rsp+40]
        add     r11, r12
        add     r11, r9
        add     r11, r8
        add     r11, rdx
        add     r11, rcx

        push    r11
        push    r12
        push    r9

        mov     r9, r8
        mov     r8, rdx
        mov     rdx, rcx
        lea     rcx, fxp2
        call    printf
        add     rsp, 16
        pop     rax
        ret


end
```

T2.H

```
#pragma once

//
// fib32.h
//
// Copyright (C) 2012 - 2017 jones@scss.tcd.ie
//
// example of mixing C++ and IA32 assembly language
//

//
// NB: "extern C" to avoid procedure name mangling by compiler
//

extern "C" _int64 minx64(_int64, _int64, _int64);
extern "C" _int64 px64(_int64, _int64, _int64, _int64);
extern "C" _int64 gcdx64(_int64, _int64);
extern "C" _int64 qx64(_int64, _int64, _int64, _int64, _int64);
extern "C" _int64 qnsx64(_int64, _int64, _int64, _int64, _int64);
```

```
// eof
```

LAB-002.CPP

```
//
// t2Test.cpp
//
// Copyright (C) 2012 - 2017 jones@scss.tcd.ie
//
// 09/10/17 first version
//


#include <iostream>         // cout
#include <conio.h>          // _getch
#include "t2.h"

using namespace std;        // cout
_int64 g = 4;
//
// check
//
void check(char *s, _int64 v, _int64 expected) {
    cout << s << " = " << v;
    if (v == expected) {
        cout << " OK";
    }
    else {
        cout << " ERROR: should be " << expected;
    }
    cout << endl;
}

//
// _tmain
//
int main(int argc, char* argv[]) {

    minx64(1, 2, 3);

    //
    // tutorial 2
    //
    check("g", g, 4);
    g++;
    check("g", g, 5);
    g--;
    check("g", g, 4);

    check("min(1, 2, 3)", minx64(1, 2, 3), 1);
    check("min(3, 1, 2)", minx64(3, 1, 2), 1);
    check("min(2, 3, 1)", minx64(2, 3, 1), 1);
    check("min(-1, -2, -3)", minx64(-1, -2, -3), -3);
    check("min(-3, -1, -2)", minx64(-3, -1, -2), -3);
    check("min(-2, -3, -1)", minx64(-2, -3, -1), -3);
```

```
    check("min(-1, 2, 3)", minx64(-1, 2, 3), -1);
    check("min(3, -1, 2)", minx64(3, -1, 2), -1);
    check("min(2, 3, -1)", minx64(2, 3, -1), -1);

    check("px64(0, 1, 2, 3)", px64(0, 1, 2, 3), 0);
    check("px64(5, 6, 7, 8)", px64(5, 6, 7, 8), 4);
    check("px64(3, 2, 1, 0)", px64(3, 2, 1, 0), 0);
    check("px64(8, 7, 6, 5)", px64(8, 7, 6, 5), 4);

    check("gcdx64(14, 21)", gcdx64(14, 21), 7);
    check("gcdx64(1406700, 164115)", gcdx64(1406700, 164115), 23445);

    check("qx64(1, 2, 3, 4, 5)", qx64(1, 2, 3, 4, 5), 15);
    check("qx64(-1, 2, -3, 4, -5)", qx64(-1, 2, -3, 4, -5), -3);

    check("qnsx64(1, 2, 3, 4, 5)", qnsx64(1, 2, 3, 4, 5), 15);
    check("qnsx64(-1, 2, -3, 4, -5)", qnsx64(-1, 2, -3, 4, -5), -3);

    cout << endl;

    return 0;

}

// eof
```

STATE DIAGRAM

Shadow Space
Shadow Space
Shadow Space
Shadow Space
GCD(14, 21) Return Address
GCD(21, 14) Return Address
GCD(14, 7) Return Address
GCD(7, 0) Return Address