**Bài tập lập trình assembly.**

**Họ Tên: Ngô Quang Hải**

**MSSV: HE153632**

**Lớp: SE1506**

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

**Bài 1.**

a.Tính (A+B)\*C/D

.include "beta.uasm"

// (A+B)\*C/D

A=0x11

B=0x12

C=0x13

D=0x14

ADDC(R31,A,R0) // R0 = A

ADDC(R0,B,R0) // R0= A + B

MULC(R0,C,R0)

DIVC(R0,D,R0)

ST(R0,a,R31) //luu gia tri R0 vao memory tai &a[0] = 0x100

.=0x100 //a[0] tai dia chi 0x100

a:

LONG(0) // a[0]

HALT()

b.(A\*B)/(C+D)

.include "beta.uasm"

// (A\*B)/(C+D)

A=0x11

B=0x12

C=0x13

D=0x14

ADDC(R31,A,R0) // R0 = A

MULC(R0,B,R0) // R0= A\*B

ADDC(R31,C,R1) //R1=C

ADDC(R1,D,R1) //R1=C+D

DIV(R0,R1,R8) //kq=R8

ST(R8,a,R31) //luu gia tri R8 vao memory tai &a[0] = 0x100

.=0x100 //a[0] tai dia chi 0x100

a:

LONG(0) // a[0]

//LONG(0)

// => a[n] ( &a[n]=0x100 + n\*sizeof(long))

HALT()

c.(A+B+C)/(A-C+D)

.include "beta.uasm"

// (A+B+C)/(A-C+D)

A=0x11

B=0x12

C=0x13

D=0x14

ADDC(R31,A,R0) // R0 = A

ADDC(R0,B,R0)

ADDC(R0,C,R0) // R0=(A+B+C)

ADDC(R31,A,R1)

ADDC(R1,-1\*C,R1)

ADDC(R1,D,R1) //R1=A-C+D

DIV(R0,R1,R8) //kq=R8

ST(R8,a,R31)

.=0x100 //a[0] tai dia chi 0x100

a:

LONG(0) // a[0]

//LONG(0)

// => a[n] ( &a[1]=0x100 + n\*sizeof(long))

HALT()

d.(A\*C)/(B-D)

.include "beta.uasm"

// (A\*C)/(B-D)

A=0x1

B=0x12

C=0x13

D=0x10

ADDC(R31,A,R0) // R0 = A

MULC(R0,C,R0) //R0=A\*C

ADDC(R31,B,R1)

ADDC(R1,-1\*D,R1) //R1=B-D

DIV(R0,R1,R8) // 17/2=9

ST(R8,a,R31)

.=0x100 //a[0] tai dia chi 0x100

a:

LONG(0) // a[0]

//LONG(0)

// => a[n] ( &a[1]=0x100 + n\*sizeof(long))

HALT()

e.A/( (B+C)\*(C-D) )

.include "beta.uasm"

// A/( (B+C)\*(C-D) )

A=0x1

B=0x12

C=0x13

D=0x10

ADDC(R31,B,R0) // R0 = B

ADDC(R0,C,R0) //R0=B+C

ADDC(R31,C,R1)

ADDC(R1,-1\*D,R1) //R1=C-D

MUL(R0,R1,R2) // R2= (B+C)\*(C-D)

ADDC(R31,A,R7)

DIV(R7,R2,R8) //A/( (B+C)\*(C-D) )

ST(R8,a,R31)

.=0x100 //a[0] tai dia chi 0x100

a:

LONG(0) // a[0]

//LONG(0)

// => a[n] ( &a[1]=0x100 + n\*sizeof(long))

HALT()

Bài 2:

Cho A,B,C,D. So sánh A&B. Nếu A=B thì (A+C)/(B+D). Nếu A!=B thì (A+C)\*(B+D)

.include "beta.uasm"

//Cho A,B,C,D. So sánh A&B. Nếu A=B thì (A+C)/(B+D). Nếu A!=B thì (A+C)\*(B+D)

A=0x1

B=0x1

C=0x13

D=0x10

ADDC(R31,A,R0)

ADDC(R0,C,R0) //R0=A+C

ADDC(R31,B,R1)

ADDC(R1,D,R1) //R1=B+D

ADDC(R31,A,R2) //R2=A

ADDC(R31,B,R3) //R3=B

// CMPEQ(RA, RB, RC) // RC <- <RA> == <RB>

CMPEQ(R2,R3,R8) //Check = R8

// BNE(RA, LABEL) // IF <RA>!=0 THEN PC <- LABEL

BNE(R8,Function1) //R8==1 thi thuc hien Function1

MUL(R0,R1,R9)

Function1:

DIV(R0,R1,R9)

The\_end:

ST(R9,a,R31) // dap so luu o R9

.=0x100 //a[0] tai dia chi 0x100

a:

LONG(0) // a[0]

//LONG(0)

// => a[n] ( &a[1]=0x100 + n\*sizeof(long))

HALT()

Bài 3:Nhập 2 số A và B.Tìm UCLN

.include "beta.uasm"

//Luu ket qua tai a[2]

//Nhập 2 số A và B.Tìm UCLN

/\*thuat toan Euclid su dung de quy (mo ta trong code C):

int GCD(int a,int b)

{

return b==0 ? a : GCD(b,a%b)

}

\*/

A=108

B=204

GCD:

//\* BEQ(RA, LABEL) // IF <RA>==0 THEN PC <- LABEL

// \* BNE(RA, LABEL) // IF <RA>!=0 THEN PC <- LABEL

LD(R31,a,R0) //R0=A

LD(R31,a+4,R1) //R1=B

BEQ(R1,The\_End) //Neu B==0 thuc thi The\_End

//thuat toan :R0 <-B , A%B = A- B\*(A/B) , R1 <- A%B

LD(R31,a,R8) // Tham chieu A=R0 vao R8

LD(R31,a+4,R9) // Tham chieu B=R1 vào R9

//R0 <-B

//\* ST(RC, CC, RA) // <RA>+CC <- <RC>

// \* ST(RC, CC) // CC <- <RC>

ST(R1,a,R31)

// A%B = A- B\*(A/B)

DIV(R8,R9,R10) //R10=A/B

MUL(R10,R9,R10) //R10=B\*(A/B)

SUB(R8,R10,R10) //R10=A- B\*(A/B)=A%B

// R1 <- A%B

ST(R10,a+4,R31)

BR(GCD) // de quy

The\_End:

ST(R0,a+8,R31) // UCLN cua 2 so A va B dc luu tai R0 (a[2] )

.=0x100 //a[0] tai dia chi 0x100

a:

LONG(A) // a[0]

LONG(B) // a[1] =a+4

LONG(0) // a[2] =a+8

HALT()

Bài 4:Nhập vào 1 mảng 10 phần tử a[0] đến a[9]. Tính tổng phần tử trong mảng.

.include "beta.uasm"

//Bài 4:Nhập vào 1 mảng 10 phần tử a[0] đến a[9]. Tính tổng phần tử trong mảng.

A0=10

A1=1

A2=2

A3=3

A4=4

A5=5

A6=6

A7=7

A8=8

A9=9

//support

n=R0

i=R1

s=R2

ra=R3

t=R4

check=R10

BR(Start)

.=0x100

a:

LONG(A0)

LONG(A1)

LONG(A2)

LONG(A3)

LONG(A4)

LONG(A5)

LONG(A6)

LONG(A7)

LONG(A8)

LONG(A9)

SUM:

LONG(0)

Start:

ADDC(R31,10,n)

ADDC(R31,0,s)

ADDC(R31,0,i)

Check:

CMPLT (i,n,check)

BNE (check,TRUE,R31)

BR(Display)

TRUE:

MULC(i,4,ra)

LD(ra,a,t)

ADD(s,t,s)

ADDC(i,1,i)

BR(Check)

Display: // luu o R2 hoac SUM

MULC(n,4,ra)

ST(s,a,ra)

HALT()

Bài 5:Nhập vào 3 số A,B,C sắp xếp theo thứ tự tăng dần

.include "beta.uasm"

// Nhập vào 3 số A,B,C sắp xếp theo thứ tự tăng dần

BR(Start)

A0=2

A1=3

A2=1

a:

LONG (A0)

LONG (A1)

LONG (A2)

Start:

LD(R31,a,R0)

LD(R31,a+4,R1)

LD(R31,a+8,R2)

Check01:

CMPLT(R1,R0,R10)

BNE(R10,swap01,R31)

BR(Check02)

Check02:

CMPLT(R2,R0,R10)

BNE(R10,swap02,R31)

BR(Check12)

Check12:

CMPLT(R2,R1,R10)

BNE(R10,swap12,R31)

BR(Display)

swap01:

ADDC(R0,0,R3)

ADDC(R1,0,R0)

ADDC(R3,0,R1)

BR(Check02)

swap02:

ADDC(R0,0,R3)

ADDC(R2,0,R0)

ADDC(R3,0,R2)

BR(Check02)

swap12:

ADDC(R1,0,R3)

ADDC(R2,0,R1)

ADDC(R3,0,R2)

BR(Display)

Display:

ST(R0,a,R31)

ST(R1,a+4,R31)

ST(R2,a+8,R31)

HALT()