Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет

информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина «Конструирование программ»

**ОТЧЕТ**

к лабораторной работе №8

на тему:

**«Интеграция ассемблерных прерываний в проекты на C++»**

БГУИР 1-40-04-01

|  |
| --- |
| Выполнил студент группы 253504  ЖГУТОВ Евгений Дмитриевич |
|  |
| (дата, подпись студента) |
| Проверил ассистент кафедры информатики  РОМАНЮК Максим Валерьевич |
|  |
| (дата, подпись преподавателя) |

Минск 2023

**Цель работы:** Задание 1.Вариант 11.Необходимо посчитать корни биквадратного уравнения.

**Ход работы:** на рисунке 1 – результат работы программы.

Листинг 1 – Исходный код программы задания 1

calculate\_discriminant proc *; diskriminant*

finit

fld coeff\_a*;*

fld coeff\_c*;*

fmulp *;*

fld four*;*

fmulp*;*

fld coeff\_b*;*

fmul st(0), st(0) *;*

fsub st(0), st(1)*;*

fstp discriminant*;*

ret

calculate\_discriminant endp

calculate\_2\_roots proc *;D > 0*

finit *; root 1*

fld discriminant

fsqrt*;*

fldz*;*

fld coeff\_b*;*

fsubp*;*

fsub st(0),st(1)*;*

fld two*;*

fld coeff\_a*;*

fmulp*;*

fdiv *;*

fstp root1*;*

finit *; root 2*

fld discriminant

fsqrt*;*

fldz*;*

fld coeff\_b*;*

fsubp*;*

fadd st(0),st(1)*;*

fld two*;*

fld coeff\_a*;*

fmulp*;*

fdiv *;*

fstp root2*;*

ret

calculate\_2\_roots endp

calculate\_1\_root proc *;D = 0*

finit *; root 1*

fld discriminant

fsqrt*;*

fldz*;*

fld coeff\_b*;*

fsubp*;*

fsub st(0),st(1)*;*

fld two*;*

fld coeff\_a*;*

fmulp*;*

fdiv *;*

fstp root1*;*

fld NaN*;*

fstp root2*;*

ret

calculate\_1\_root endp

calculate\_0\_roots proc *;D < 0*

finit *; root 1*

fld NaN*;*

fstp root1*;*

fld NaN*;*

fstp root2*;*

ret

calculate\_0\_roots endp

calculate\_A\_exception proc *; A = 0*

finit *; root 1*

fld coeff\_c*;*

fldz*;*

fsubp*;*

fld coeff\_b*;*

fdivp*;*

fstp root1*;*

fld NaN*;*

fstp root2*;*

ret

calculate\_A\_exception endp

calculate\_AB\_exception proc *; A = 0, B = 0*

finit *; root 1*

fld coeff\_c*;*

fldz*;*

fcom*;*

je infinit\_sols*;*

fld NaN*;*

jmp over\_inf*;*

infinit\_sols: fld inf*;*

over\_inf: fstp root1*;*

fld NaN*;*

fstp root2*;*

ret

calculate\_AB\_exception endp

solve\_equation proc

finit*; CHECK A = 0*

fldz*;*

fld coeff\_a*;*

fcom*;*

fnstsw ax*;*

sahf*;*

jne nonzero\_A*;*

fldz*; CHECK A = 0, B = 0*

fld coeff\_b*;*

fcom*;*

fnstsw ax*;*

sahf*;*

je ABexception

Aexception:

call calculate\_A\_exception*;*

jmp end\_proc*;*

ABexception:

call calculate\_AB\_exception*;*

jmp end\_proc*;*

nonzero\_A: *;A <> 0*

output\_string discriminant\_message*;*

call calculate\_discriminant*;*

finit*;*

fld zro*;*

fld discriminant*;*

fcom*;*

fnstsw ax*;*

sahf*;*

jc error *;*

je roots1

roots2: *; D > 0*

output\_string two\_roots\_message

call calculate\_2\_roots*;*

jmp end\_proc*;*

roots1: *; D = 0*

output\_string one\_root\_message

call calculate\_1\_root*;*

jmp end\_proc*;*

error:

output\_string zero\_roots\_message

call calculate\_0\_roots*;*

end\_proc:

ret

solve\_equation endp

handle\_biquadratic\_roots proc

*;root1*

finit

fldz*;*

fld root1*;*

fcom*;*

fnstsw ax*;*

sahf*;*

jc NoRoots1

je OneRoot1*;*

jnc TwoRoots1*;*

jmp Exception1*;*

NoRoots1:

output\_string zero\_root1\_message

fld NaN*;*

fstp b1root1*;*

fld NaN*;*

fstp b2root1*;*

jmp EndRoot1

OneRoot1:

push di

lea di, count

add [di],1

pop di

output\_string one\_root1\_message

fldz*;*

fstp b1root1*;*

fld NaN*;*

fstp b2root1*;*

jmp EndRoot1

TwoRoots1:

push di

lea di, count

add [di], 2

pop di

output\_string roots1\_root1\_message

fld root1*;*

fsqrt*;*

fst b1root1*;*

fchs*;*

fst b2root1*;*

jmp EndRoot1

Exception1:

push di

lea di, count

mov [di], 5

pop di

fld root1*;*

fld inf*;*

fnstsw ax*;*

sahf*;*

jne NoRoots1*;*

finit

fld inf*;*

fld inf*;*

fstp b1root1*;*

fstp b2root1 *;*

EndRoot1:

*;root2*

finit

fldz*;*

fld root2*;*

fcom*;*

fnstsw ax*;*

sahf*;*

jc NoRoots2

je OneRoot2*;*

jnc TwoRoots2*;*

jmp Exception2*;*

NoRoots2:

output\_string zero\_root2\_message

fld NaN*;*

fstp b1root2*;*

fld NaN*;*

fstp b2root2*;*

jmp EndRoot2

OneRoot2:

push di

lea di, count

add [di], 1

pop di

output\_string one\_root2\_message

fldz*;*

fstp b1root2*;*

fld NaN*;*

fstp b2root2*;*

jmp EndRoot2

TwoRoots2:

push di

lea di, count

add [di], 2

pop di

output\_string roots2\_root2\_message

fld root2*;*

fsqrt*;*

fst b1root2*;*

fchs*;*

fst b2root2*;*

jmp EndRoot2

Exception2:

push di

lea di, count

mov [di], 5

pop di

fld root2*;*

fld inf*;*

fnstsw ax*;*

sahf*;*

jne NoRoots2*;*

finit

fld inf*;*

fld inf*;*

fstp b1root2*;*

fstp b2root2 *;*

EndRoot2:

ret

handle\_biquadratic\_roots endp

\_asmsolution proc

mov ax,@data*;*

mov ds,ax*;*

mov es,ax*;*

*;input a , b , c*

output\_string a\_input\_message

call input\_processing

finit*;*

fld result\_value*;*

fstp coeff\_a*;*

output\_string b\_input\_message

call input\_processing

finit*;*

fld result\_value*;*

fstp coeff\_b*;*

output\_string c\_input\_message

call input\_processing

finit*;*

fld result\_value*;*

fstp coeff\_c*;*

call solve\_equation*;*

call handle\_biquadratic\_roots*;*

mov bx,1*;*

fld b1root1*;*

fld b2root1*;*

fld b1root2*;*

fld b2root2*;*

output\_string exit\_message*;*

ret

\_asmsolution endp

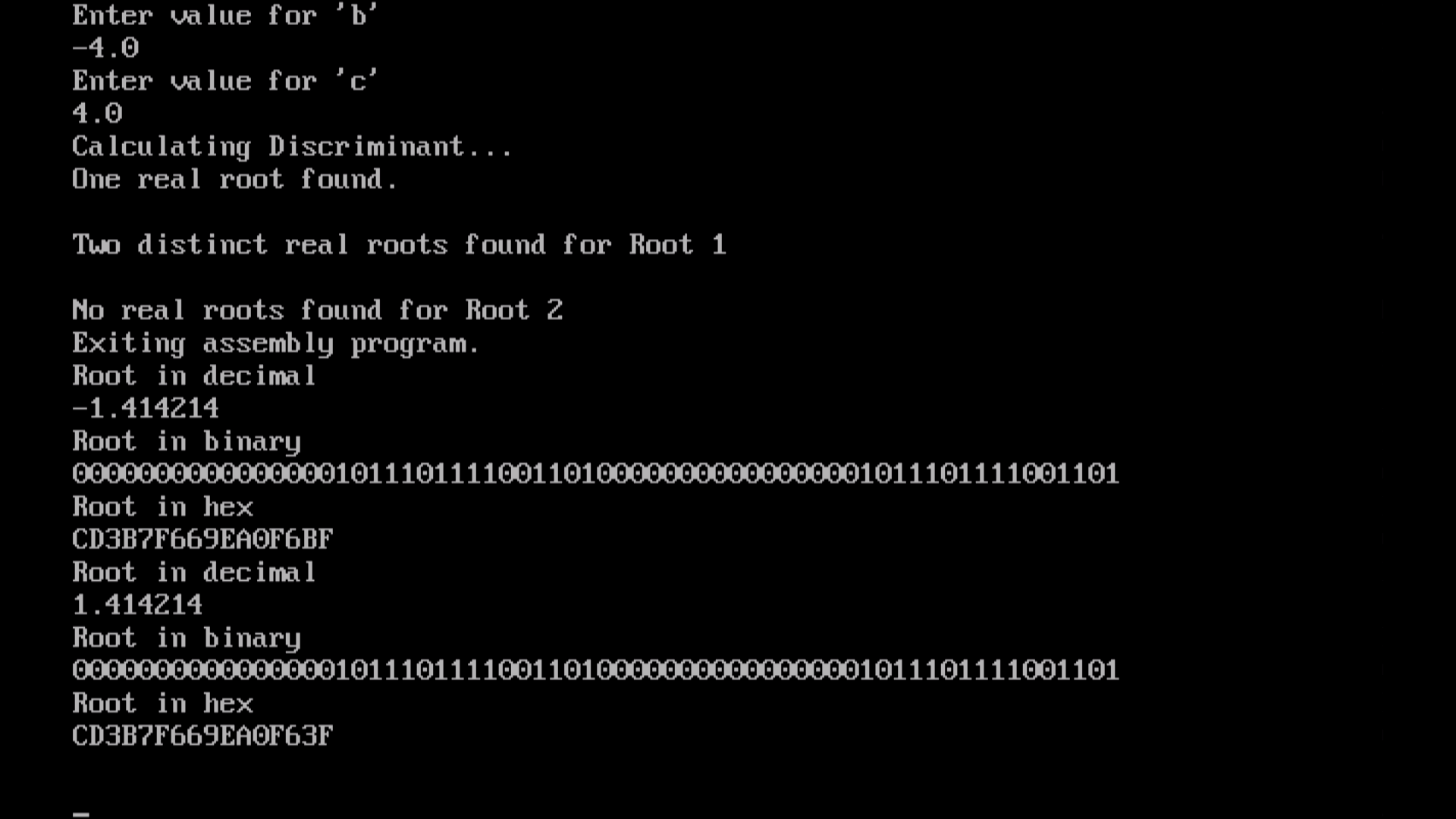


Рисунок 1 – Результат работы программы

**Выводы:** В результате лабораторной работы была выполнена одна задача с интегрированием ассемблерных прерываний в проект на С++.