**Отчёт по лабораторной работе №10 по курсу**

**«Фундаментальная информатика»**

Студент группы: М8О-108Б-22, Немкова Анастасия Романовна,

№ по списку 14

Контакты: [nastya.nemkova.04@mail.ru](mailto:nastya.nemkova.04@mail.ru)

Работа выполнена: 24.10.2022

Преподаватель: Сахаров Никита Александрович

**1. Тема**

Отладчик системы программирования ОС UNIX

**2. Цель работы**

Освоить отладку простейших программ с помощью gdb на языке С

**3. Задание**

Отладить ошибку в простейшей программе

**4. Оборудование**

Процессор: AMD RYZEN 7 5800H 3.20GHz ОП 16 ГБ

НDD: 952 ГБ

Монитор: 3840 × 2400

**5. Программное обеспечение**

Операционная система семейства: Linux Ubuntu, версия 22.04.1 LTS

Интерпретатор команд: bash, версия 5.0.17(1)

Система программирования: gcc

Редактор текстов: emacs

Утилиты операционной системы: gdb

**6. Идея, метод, алгоритм решения задачи**

Написать простейшую программу на С, скомпилировать её с флагом -g, запустить отладчик gdb и проделать основные команды ( list, run, break, next, stop, set args, show args, set var, backtrace и тд.) по отладке программы.

**7. Сценарий выполнения работы**

Скомпилируем программу и используем команды help, run, print, set var, next, step, continue, break, bt, list, set args для её отладки при помощи gdb

Код программы:

#include <stdio.h>

#include <math.h>

int main(void){

double a, b, c, x1, x2, D;

scanf("%lf%lf%lf", &a, &b, &c);

D = b\*b - 4\*a\*c;

if( D < 0 ){

printf("No roots\n");}

else if( D == 0 ){

x1 = -b/(2\*a);

printf("One root: %lf\n", x1);

}

else {

x1 = (-b - sqrt(D)) / (2\*a);

x2 = (-b + sqrt(D)) / (2\*a);

printf("Two roots: %lf%lf\n", x1, x2);

}

return 0;

}

**8. Распечатка протокола**

**anastasia@anastasia-VirtualBox**:~$ gcc -g lab10.c -lm

**anastasia@anastasia-VirtualBox**:~$ gdb a.out

GNU gdb (Ubuntu 9.2-0ubuntu1~20.04.1) 9.2

Copyright (C) 2020 Free Software Foundation, Inc.

License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>

This is free software: you are free to change and redistribute it.

There is NO WARRANTY, to the extent permitted by law.

Type "show copying" and "show warranty" for details.

This GDB was configured as "x86\_64-linux-gnu".

Type "show configuration" for configuration details.

For bug reporting instructions, please see:

<http://www.gnu.org/software/gdb/bugs/>.

Find the GDB manual and other documentation resources online at:

<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".

Type "apropos word" to search for commands related to "word"...

Reading symbols from a.out...

*(gdb)* help

List of classes of commands:

aliases -- Aliases of other commands.

breakpoints -- Making program stop at certain points.

data -- Examining data.

files -- Specifying and examining files.

internals -- Maintenance commands.

obscure -- Obscure features.

running -- Running the program.

stack -- Examining the stack.

status -- Status inquiries.

support -- Support facilities.

tracepoints -- Tracing of program execution without stopping the program.

user-defined -- User-defined commands.

Type "help" followed by a class name for a list of commands in that class.

Type "help all" for the list of all commands.

Type "help" followed by command name for full documentation.

Type "apropos word" to search for commands related to "word".

Type "apropos -v word" for full documentation of commands related to "word".

Command name abbreviations are allowed if unambiguous.

*(gdb)* list 1,19

1 #include <stdio.h>

2 #include <math.h>

3 int main(void){

4 double a, b, c, x1, x2, D;

5 scanf("%lf%lf%lf", &a, &b, &c);

6 D = b\*b - 4\*a\*c;

7 if( D < 0 ){

8 printf("No roots\n");}

9 else if( D == 0 ){

10 x1 = -b/(2\*a);

11 printf("One root: %lf\n", x1);

12 }

13 else {

14 x1 = (-b - sqrt(D)) / (2\*a);

15 x2 = (-b + sqrt(D)) / (2\*a);

16 printf("Two roots: %lf%lf\n", x1, x2);

17 }

18 return 0;

19 }

*(gdb)* break main

Breakpoint 1 at 0x11c9: file lab10.c, line 3.

*(gdb)* next

The program is not being run.

*(gdb)* run

Starting program: /home/anastasia/a.out

Breakpoint 1, main () at lab10.c:3

3 int main(void){

*(gdb)* next

5 scanf("%lf%lf%lf", &a, &b, &c);

*(gdb)* print a

$1 = 4.6355705385357047e-310

*(gdb)* set var a=1

*(gdb)* print a

$2 = 1

*(gdb)* print b

$3 = 6.9533490934611385e-310

*(gdb)* set var b=5

*(gdb)* print b

$4 = 5

*(gdb)* print c

$5 = 4.6355705385319004e-310

*(gdb)* set var c=6

*(gdb)* print c

$6 = 6

*(gdb)* next

continue

6 D = b\*b - 4\*a\*c;

*(gdb)* next

7 if( D < 0 ){

*(gdb)* next

9 else if( D == 0 ){

*(gdb)* next

14 x1 = (-b - sqrt(D)) / (2\*a);

*(gdb)* next

15 x2 = (-b + sqrt(D)) / (2\*a);

*(gdb)* next

16 printf("Two roots: %lf%lf\n", x1, x2);

*(gdb)* next

Two roots: -3.000000-2.000000

18 return 0;

*(gdb)* bt

#0 main () at lab10.c:18

*(gdb)* ptype a

type = double

*(gdb)* run

The program being debugged has been started already.

Start it from the beginning? (y or n) n

Program not restarted.

*(gdb)* continue

Continuing.

[Inferior 1 (process 3519) exited normally]

*(gdb)* run

Starting program: /home/anastasia/a.out

Breakpoint 1, main () at lab10.c:3

3 int main(void){

*(gdb)* step

5 scanf("%lf%lf%lf", &a, &b, &c);

*(gdb)* set args 4 86 3

*(gdb)* continue

Continuing.

next

One root: -0.749999

[Inferior 1 (process 3529) exited normally]

*(gdb)* print a

$7 = {i = {0, 1045149306}, d = 1.2904777690891933e-08}

*(gdb)* quit

**9. Дневник отладки**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | Лаб.  Или  Дом. | Дата | Время | Событие | Действие по исправлению | Примечание |
| 1 | Дом. | 29.10.23 | 13:48 | Отладка простейшей программы на С | - | - |

**10. Замечания автора по существу работы**

**11. Выводы**

В ходе выполнения данной лабораторной работы были изучены команды для запуска программы с флагом -g и её отладке в gdb. Также были освоены команды (по запуску программы, построчному выполнению программы, отдельному присваиванию значений аргументов, определению типа переменных, остановке программы на определённой строке) для отладки простейшей программы на С при помощи gdb.