**Отчет по лабораторной работе №10** по курсу фундаментальная информатика

Студент группы М8О-113Б-22, № по списку 1, Астахова Анастасия Сергеевна

Контакты astakhovaanastasia0201@gmail.com

Работа выполнена: «29» октября 2022 г.

Преподаватель: доцент каф. 806 Никулин Сергей Петрович

Входной контроль знаний с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Отчет сдан « » \_\_\_\_\_\_\_\_\_2022 г., итоговая оценка \_\_\_\_\_

Подпись преподавателя \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* 1. **Тема:** Отладчик системы программирования ОC UNIX \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Цель работы:** Научиться пользоваться отладчиком gdb, запомнить основные комманды для работы с ним. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. **Задание :** Ввести основные команды во время отладки программы, понять принцип работы отладчика. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
3. **Оборудование** (лабораторное):

ЭВМ \_\_\_\_\_\_\_\_\_, процессор \_\_\_\_ \_\_, имя узла сети \_\_\_\_\_\_ с ОП \_\_\_\_\_\_\_\_ Мб, НМД \_\_\_\_\_\_\_ Мб. Терминал \_\_\_\_\_ адрес \_\_\_\_\_\_\_\_\_\_. Принтер \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Другие устройства \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Оборудование ПЭВМ студента, если использовалось:*

Ноутбук: Процессор Intel Core i5-10210U с ОП 16 ГБ, SSD 512 ГБ. Другие устройства: мышь Bloody R80

*Компьютер: Процессор Ryzen 5 3600 с ОП 16 ГБ, SSD 512 ГБ. Другие устройства: мышь* Razer BASILISK X HYPERSPEED, [*Logitech G910 (920-008019) Orion Spectrum RGB Mechanical Gaming*](https://www.komus.ru/katalog/tekhnika/kompyuternaya-tekhnika/klaviatury-i-myshi/klaviatury-provodnye/klaviatura-logitech-g910-920-008019-orion-spectrum-rgb-mechanical-gaming/p/659916/?from=n-13776), монитор [*LG UltraGear 27GP750*](https://www.lg.com/ru/monitors/lg-27gp750-b)

1. **Программное обеспечение (лабораторное):**

Операционная система семейства \_\_\_Unix\_\_\_\_, наименование \_\_\_\_\_Ubuntu\_\_\_\_\_\_\_\_\_\_\_\_ версия \_\_4.15.0\_\_\_\_\_\_

интерпретатор команд \_\_\_\_bash\_\_\_\_\_\_\_\_ версия \_\_\_4.4.20\_\_\_\_

Система программирования \_\_С\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ версия \_\_\_\_\_\_\_последняя\_\_\_\_\_\_\_\_\_

Редактор текстов \_\_\_gedit\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ версия \_25.2.2 \_\_\_\_\_\_\_\_\_

Утилиты операционной системы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Прикладные системы и программы \_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Местонахождение и имена файлов программ и данных\_\_stud/208104 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**5.** *Программное обеспечение ПЭВМ студента, если использовалось****:***

Ноутбук: Операционная система семейства Ubuntu, наименование версия Ubuntu 20.04.3 LTS,

*интерпретатор команд bash версия 5.0.17. Система программирования C. Редактор текстов gedit версия 28.2. Утилиты: стандартные утилиты UNIX. Программное обеспечение: стандартное программное обеспечение системы UNIX.*

**6. Идея, метод, алгоритм** решения задачи(в формах:словесной,псевдокода,графической[блок-схема,диаграмма,рисунок,таблица] или формальные спецификации с пред- и постусловиями)

* Скомпилируем исходную программу с помощью ключа компилятора -g
* Введем данные, приводящие к ошибке. Проделаем все действия отладки над программой.

|  |  |
| --- | --- |
| * Команда gdb | * Описание команды |
| * help [раздел] | * Подсказка по разделу отладчика. Без параметров выводит список разделов. |
| * list [имя фукнции/файла:] [номер строки] | * Распечатка текста функции/процедуры/файла или всей программы, начиная с указанной строки. По умолчанию распечатываются 10 строк программы. Распечатываемый файл становится текущим файлом исходного текста отлаживаемой программы. |
| * break [номер строки/имя функции] | * Задание точки остановки на строке/функции текущего исходного файла программы |
| * run [параметры] | * Запуск программы на выполнение. Могут указываться необязательные параметры командной строки и операции перенаправления ввода-вывода. gdb запоминает параметры и подставляет их для дальнейших вызовах run. |
| * set args [параметры] | * Предварительная установка параметров командной строки. |
| * show args | * Вывод параметров командной строки. |
| * print [выражение] | * Печать значения выражения, которое может включать и переменные, и вызовы функций программы. |
| * next [n] | * Выполнение очередной строки программы при пошаговой трассировке (процедуры и функции не трассируются, а выполняются за один такт). Необязательный параметр n указывает число строк программы для выполнения. По умолчанию n = 1. |
| * step [n] | * Выполнение очередной строки программы (с трассировкой процедур и функций). Перед выполнением next/step программа должна быть запущена командой run. |
| * set var [имя] = [выражение] | * Присваивание значения переменной. |
| * pytype [имя переменной] | * Выводит тип переменной. |
| * backtrace или bt | * Распечатка содержимого стека вызовов. |
| * continue | * Продолжение выполнения программы после остановки. |
| * quit | * Выход из отладчика. |

1. **Сценарий выполнения работы** [план работы,первоначальный текст программы в черновике(можно на отдельном листе)итесты либо соображения по тестированию].

Тесты:

Input:

Output: Бесконечный цикл

1. Скомпилируем при помощи команды gcc и ключа -g программу main.c (gcc -g main.c)
2. Запустим отладку с помощью команды gdb (gdb a.out)
3. Проделаем все действие по отладке программы

*Пункты 1-7 отчета составляются сторого до начала лабораторной работы.*

*Допущен к выполнению работы.* **Подпись преподавателя****\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. **Распечатка протокола** (подклеить листинг окончательного варианта программы с тестовыми примерами,подписанныйпреподавателем).

kokodrillo@kokodrillo-VirtualBox:~/work/Лабы\_по\_С$ gcc -g lab10.c

kokodrillo@kokodrillo-VirtualBox:~/work/Лабы\_по\_С$ gdb a.out

GNU gdb (Ubuntu 12.1-0ubuntu1~22.04) 12.1

Copyright (C) 2022 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:

<https://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 -- User-defined 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.

text-user-interface -- TUI is the GDB text based interface.

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) help break

break break-range breakpoints

(gdb) help breakpoints

Making program stop at certain points.

List of commands:

awatch -- Set an access watchpoint for EXPRESSION.

break, brea, bre, br, b -- Set breakpoint at specified location.

break-range -- Set a breakpoint for an address range.

catch -- Set catchpoints to catch events.

catch assert -- Catch failed Ada assertions, when raised.

catch catch -- Catch an exception, when caught.

catch exception -- Catch Ada exceptions, when raised.

catch exec -- Catch calls to exec.

catch fork -- Catch calls to fork.

catch handlers -- Catch Ada exceptions, when handled.

catch load -- Catch loads of shared libraries.

catch rethrow -- Catch an exception, when rethrown.

catch signal -- Catch signals by their names and/or numbers.

catch syscall -- Catch system calls by their names, groups and/or numbers.

catch throw -- Catch an exception, when thrown.

catch unload -- Catch unloads of shared libraries.

catch vfork -- Catch calls to vfork.

clear, cl -- Clear breakpoint at specified location.

commands -- Set commands to be executed when the given breakpoints are hit.

--Type <RET> for more, q to quit, c to continue without paging--q

Quit

(gdb) list 1

1 #include <stdio.h>

2

3 int main() {

4 int n = 0;

5 int g = 2;

6

7 while (n < 10) {

8 g = g \* 2;

9 printf("%d\n", g);

10 //n++;

(gdb) list

11 }

12 //printf("%d\n", g);

13 return 0;

14 }

(gdb) break 9

Breakpoint 1 at 0x1168: file lab10.c, line 9.

(gdb) set args 1 1

(gdb) show args

Argument list to give program being debugged when it is started is "1 1".

(gdb) run

Starting program: /home/kokodrillo/work/Лабы\_по\_С/a.out

[Thread debugging using libthread\_db enabled]

Using host libthread\_db library "/lib/x86\_64-linux-gnu/libthread\_db.so.1".

Breakpoint 1, main () at lab10.c:9

9 printf("%d\n", g);

(gdb) print g

$1 = 4

(gdb) next

4

7 while (n < 10) {

(gdb)

8 g = g \* 2;

(gdb)

Breakpoint 1, main () at lab10.c:9

9 printf("%d\n", g);

(gdb)

8

7 while (n < 10) {

(gdb)

8 g = g \* 2;

(gdb)

Breakpoint 1, main () at lab10.c:9

9 printf("%d\n", g);

(gdb) next 3

16

Breakpoint 1, main () at lab10.c:9

9 printf("%d\n", g);

(gdb)

32

Breakpoint 1, main () at lab10.c:9

9 printf("%d\n", g);

(gdb) step

\_\_printf (format=0x555555556004 "%d\n") at ./stdio-common/printf.c:28

28 ./stdio-common/printf.c: Нет такого файла или каталога.

(gdb)

32 in ./stdio-common/printf.c

(gdb)

33 in ./stdio-common/printf.c

(gdb) run

The program being debugged has been started already.

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

Starting program: /home/kokodrillo/work/Лабы\_по\_С/a.out

[Thread debugging using libthread\_db enabled]

Using host libthread\_db library "/lib/x86\_64-linux-gnu/libthread\_db.so.1".

Breakpoint 1, main () at lab10.c:9

9 printf("%d\n", g);

(gdb) set var n = 10

(gdb) continue

Continuing.

4

[Inferior 1 (process 4874) exited normally]

(gdb) run

Starting program: /home/kokodrillo/work/Лабы\_по\_С/a.out

[Thread debugging using libthread\_db enabled]

Using host libthread\_db library "/lib/x86\_64-linux-gnu/libthread\_db.so.1".

Breakpoint 1, main () at lab10.c:9

9 printf("%d\n", g);

(gdb) ptype n

type = int

(gdb) bt

#0 main () at lab10.c:9

(gdb) quit

A debugging session is active.

Inferior 1 [process 4875] will be killed.

Quit anyway? (y or n) y

1. **Дневник отладки** должен содержать дату и время сеансов отладки, и основные события(ошибки в сценарии и программе,нестандартные ситуации) и краткие комментарии к ним. В дневнике отладки приводятся сведения об использовании других ЭВМ, существенном участии преподавателя и других лиц в написании и отладке программы.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | Лаб. | Дата | Время | Событие | Действие по исправлению | Примечание |
|  | или |  |  |  |  |  |
|  | дом. |  |  |  |  |  |
|  |  |  |  |  |  |  |

1. **Замечания автора** по существу работы:

Замечания отсутствуют.

1. **Выводы**

Я научилась пользоваться отладчиком и узнала основные команды.

Недочёты при выполнении задания могут быть устранены следующим образом:

Недочеты отсутствуют.

Подпись студента \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_