**Отчет по лабораторной работе № 10** по курсу Вычислительные системы

Студент группы М8О-110Б-21 Елистратова Полина Александровна № по списку 6

Контакты e-mail, telegram, skype еp.[elistratova03@mail.ru](mailto:elistratova03@mail.ru)

Работа выполнена: « » \_\_\_\_\_\_\_\_\_\_\_\_20\_\_г.

Преподаватель: доцент каф. 806 Никулин Сергей Петрович

Отчет сдан « » \_\_\_\_\_\_\_\_\_20 \_\_\_ г., итоговая оценка \_\_\_\_\_

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

* 1. **Тема:** Отладчики систем программирования ОС UNIX

1. **Цель работы:** Научиться отлаживать программу в gdb
2. **Задание** (*вариант №*)**:**
3. **Оборудование** (лабораторное):

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

Другие устройства

*Оборудование ПЭВМ студента, если использовалось:*

Процессор Intel Core i3 с ОП 4 Гб НМД 128 Гб. Монитор 1920x1080~60Hz

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

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

Операционная система семейства \_\_\_\_\_\_\_\_\_\_\_, наименование \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ версия \_\_\_\_\_\_\_\_\_\_\_\_\_

интерпретатор команд \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ версия \_\_\_\_\_\_\_\_\_\_\_\_

Система программирования \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ версия \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Редактор текстов \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ версия \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

Прикладные системы и программы

Местонахождение и имена файлов программ и данных

*Программное обеспечение ЭВМ студента, если использовалось:*

Операционная система семейства Linux , наименование Ubuntu версия 20.04.1

интерпретатор команд GNU bash версия 5.0.17 .

Система программирования Code::Blocks IDE версия 20.03

Редактор текстов emacs версия 25.2.2

Утилиты операционной системы gcc, cat, gdb

Прикладные системы и программы

Местонахождение и имена файлов программ и данных на домашнем компьютере \

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

**cat >** - создание файла с программой

**gcc -g** -компиляция программы с обязательным ключом для отладки -g

**gdb ./a.out**- запуск отладчика

**help** — подсказка по разделу помощи отладчика

**run** — запустить программу

**start** - устанавливает временную точку останова в main() и запускает выполнение программы в GDB

**print**- вывести значение переменной

**set var** -присваивание значения переменной

**continue** - продолжить выполнение программы после точки остановки

**break** - создание точки остановки

**info breakpoints** — выводит информацию о всех точках остановки

**disable** — отключение точки остановки

**next**- выполнить следующую строчку без захода в функцию

**step** - выполнить следующую строчку программы с заходом в функцию

**ptype** - просмотр типа переменной

**bt** - вывести стек вызовов

**set args**- предварительная установка параметров командой строки

**list** -распечатка текста программы(всей или с указанной строчки).По умолчанию распечатывается следующие 10 строк программы

**quit** - выход из программы

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

1. Создание файла с программой. Программа выводит степени двойки начиная с 1 пока она меньше или равна введенному числу и содержит функцию для наглядного примера работы команд отладчика.
2. Компиляция программы с ключом -g.
3. Запуск отладчика gdb.
4. Отладка программы в отладчике, применение различных его команд.
5. Выход из отладчика.

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

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

polina@pelis:~$ cat > lr10.c

/\*ЛР №10 Елистратова Полина

группа:М8О-110Б-21\*/

#include <stdio.h>

#include <stdlib.h>

int fun(int a1, int a2, int b1, int b2){

return a1\*b1+a2\*b2;

}

int main()

{

int pow2\_max;

scanf("%d", &pow2\_max);

int pow2 = 1;

while (pow2 <= pow2\_max){

printf("%d\n",pow2);

pow2\*=2;

}

fun(1,2,3,4);

return 0;

}

^C

polina@pelis:~$ gcc -g lr10.c

polina@pelis:~$ gdb a.out

GNU gdb (Ubuntu 9.2-0ubuntu1~20.04) 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) run

Starting program: /home/polina/a.out

16

1

2

4

8

16

[Inferior 1 (process 29039) exited normally]

(gdb) start

Temporary breakpoint 1 at 0x5555555551b1: file lr10.c, line 11.

Starting program: /home/polina/a.out

Temporary breakpoint 1, main () at lr10.c:11

11 {

(gdb) continue

Continuing.

68

1

2

4

8

16

32

64

[Inferior 1 (process 29045) exited normally]

(gdb) start

Temporary breakpoint 2 at 0x5555555551b1: file lr10.c, line 11.

Starting program: /home/polina/a.out

Temporary breakpoint 2, main () at lr10.c:11

11 {

(gdb) print pow2\_max

$2 = -7904

(gdb) list

6 int fun(int a1, int a2, int b1, int b2){

7 return a1\*b1+a2\*b2;

8 }

9

10 int main()

11 {

12 int pow2\_max;

13 scanf("%d", &pow2\_max);

14 int pow2 = 1;

15 while (pow2 <= pow2\_max){

(gdb) set var pow2\_max = 0

(gdb) print pow2\_max

$3 = 0

(gdb) continue

Continuing.

127

1

2

4

8

16

32

64

[Inferior 1 (process 29158) exited normally]

(gdb) break 16

Breakpoint 3 at 0x5555555551ed: file lr10.c, line 16.

(gdb) break 17

Breakpoint 4 at 0x555555555203: file lr10.c, line 17.

(gdb) info breakpoints

Num Type Disp Enb Address What

3 breakpoint keep y 0x00005555555551ed in main at lr10.c:16

4 breakpoint keep y 0x0000555555555203 in main at lr10.c:17

(gdb) disable 4

(gdb) info breakpoints

Num Type Disp Enb Address What

3 breakpoint keep y 0x00005555555551ed in main at lr10.c:16

4 breakpoint keep n 0x0000555555555203 in main at lr10.c:17

(gdb) run

Starting program: /home/polina/a.out

513

Breakpoint 3, main () at lr10.c:16

16 printf("%d\n",pow2);

(gdb) print pow2

$4 = 1

(gdb) display pow2

1: pow2 = 1

(gdb) continue

Continuing.

1

Breakpoint 3, main () at lr10.c:16

16 printf("%d\n",pow2);

1: pow2 = 2

(gdb)

Continuing.

2

Breakpoint 3, main () at lr10.c:16

16 printf("%d\n",pow2);

1: pow2 = 4

(gdb)

Continuing.

4

Breakpoint 3, main () at lr10.c:16

16 printf("%d\n",pow2);

1: pow2 = 8

(gdb)

Continuing.

8

Breakpoint 3, main () at lr10.c:16

16 printf("%d\n",pow2);

1: pow2 = 16

(gdb)

Continuing.

16

Breakpoint 3, main () at lr10.c:16

16 printf("%d\n",pow2);

1: pow2 = 32

(gdb)

Continuing.

32

Breakpoint 3, main () at lr10.c:16

16 printf("%d\n",pow2);

1: pow2 = 64

(gdb)

Continuing.

64

Breakpoint 3, main () at lr10.c:16

16 printf("%d\n",pow2);

1: pow2 = 128

(gdb)

Continuing.

128

Breakpoint 3, main () at lr10.c:16

16 printf("%d\n",pow2);

1: pow2 = 256

(gdb)

Continuing.

256

Breakpoint 3, main () at lr10.c:16

16 printf("%d\n",pow2);

1: pow2 = 512

(gdb)

Continuing.

512

[Inferior 1 (process 29244) exited normally]

(gdb) break 19

Breakpoint 5 at 0x55555555520e: file lr10.c, line 19.

(gdb) info breakpoints

Num Type Disp Enb Address What

3 breakpoint keep y 0x00005555555551ed in main at lr10.c:16

breakpoint already hit 10 times

4 breakpoint keep n 0x0000555555555203 in main at lr10.c:17

5 breakpoint keep y 0x000055555555520e in main at lr10.c:19

(gdb) disable 3

(gdb) info breakpoints

Num Type Disp Enb Address What

3 breakpoint keep n 0x00005555555551ed in main at lr10.c:16

breakpoint already hit 10 times

4 breakpoint keep n 0x0000555555555203 in main at lr10.c:17

5 breakpoint keep y 0x000055555555520e in main at lr10.c:19

(gdb) run

Starting program: /home/polina/a.out

3

1

2

Breakpoint 5, main () at lr10.c:19

19 fun(1,2,3,4);

1: pow2 = 4

(gdb) next

20 return 0;

1: pow2 = 4

(gdb) run

The program being debugged has been started already.

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

Starting program: /home/polina/a.out

3

1

2

Breakpoint 5, main () at lr10.c:19

19 fun(1,2,3,4);

1: pow2 = 4

(gdb) step

fun (a1=1, a2=2, b1=3, b2=4) at lr10.c:6

6 int fun(int a1, int a2, int b1, int b2){

(gdb) step

7 return a1\*b1+a2\*b2;

(gdb) step

8 }

(gdb) step

main () at lr10.c:20

20 return 0;

1: pow2 = 4

(gdb) step

21 }

(gdb) break 17

Note: breakpoint 4 (disabled) also set at pc 0x555555555203.

Breakpoint 6 at 0x555555555203: file lr10.c, line 17.

(gdb) run

Starting program: /home/polina/a.out

2

1

Breakpoint 6, main () at lr10.c:17

17 pow2\*=2;

1: pow2 = 1

(gdb) ptype pow2

type = int

(gdb) bt

#0 main () at lr10.c:17

(gdb) list 7

2 группа:М8О-110Б-21\*/

3 #include <stdio.h>

4 #include <stdlib.h>

5

6 int fun(int a1, int a2, int b1, int b2){

7 return a1\*b1+a2\*b2;

8 }

9

10 int main()

11 {

(gdb) list fun

1 /\*ЛР №10 Елистратова Полина

2 группа:М8О-110Б-21\*/

3 #include <stdio.h>

4 #include <stdlib.h>

5

6 int fun(int a1, int a2, int b1, int b2){

7 return a1\*b1+a2\*b2;

8 }

9

10 int main()

(gdb) quit

polina@pelis:~$

**9. 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 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

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

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

1. **Выводы** Я научилась работать с отладчиком gdb ОС UNIX.

Недочёты при выполнении задания могут быть устранены следующим образом: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

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

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

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

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

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