

Отчет по лабораторной работе № 10 по курсу «Фундаментальная информатика»

Студент группы Алапанова Эльза Халилевна, № по списку 3

Контакты e-mail : alapanowa02@yandex.ru

Работа выполнена: «14» ноября 2020г.

Преподаватель: каф. 806 Найденов Иван Евгеньевич

Отчет сдан « » _____ 20 ____ г., итоговая оценка _____

Подпись преподавателя _____

1. **Тема:** отладчик системы программирования ОС UNIX.
2. **Цель работы:** научиться отлаживать программы с помощью интерактивного отладчика gdb.
2. **Задание (вариант №):** продемонстрировать умение пользоваться отладчиком gdb.
3. **Оборудование** (студенческое)

Процессор Celeron(R) Dual-Core CPU T3300 с ОП _____ Мб, НМД _____ Мб. Монитор _____

4. Программное обеспечение (студенческое):

Операционная система семейства linux, наименование ubuntu версия 14.04 LTS

интерпретатор команд _____ версия _____.

Система программирования _____ версия _____

Редактор текстов _____ версия _____

Утилиты операционной системы _____

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)

Для этой лабораторной работы была написана программа, которая принимает на входе одно число, а выводит кол-во цифр в нём. В ней ставим breakpoint и смотрим за изменением переменных. Предемонстрированы следующие утилиты:

- 1) list
- 2) break(b)
- 3) del
- 4) info breakpoints
- 5) run
- 6) continue(c)
- 7) print
- 8) next(n)
- 9) info locals
- 10) quit
- 11) set var

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

План работы:

- 1) Продемонстрировать работу утилиты list, выводя куски кода функции main.
- 2) Продемонстрировать работу утилиты break и del - поставив и удалив breakpoint
- 3) Продемонстрировать работу утилиты info breakpoints
- 4) Продемонстрировать работу цикла if
- 5) Продемонстрировать работу утилиты info locals, путем вывода конечного результата
- 6) Продемонстрировать работу утилиты set var, изменив значение переменных внутри цикла
- 7) Завершить отладку при помощи quit

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

8. Распечатка протокола (подклеить листинг окончательного варианта программы с тестовыми примерами, подписанный преподавателем).

```
elza@elza-R428-P428-R478:~$ gcc -g exx.c
elza@elza-R428-P428-R478:~$ gdb a.out
GNU gdb (Ubuntu 7.7.1-0ubuntu5~14.04.3) 7.7.1
Copyright (C) 2014 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 "i686-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...done.
(gdb) list main
1      #include <stdio.h>
2
3      int main(void)
4      {
5          int a, b;
6          scanf("%d%d", &a, &b);
7
8          int result = 0;
9          if (a >= 5 && b <= 0) {
10             result = 8 * a + 4 * b;
(gdb) next
The program is not being run.
(gdb) list else
Function "else" not defined.
(gdb) list int
(gdb) b 12
Breakpoint 1 at 0x80484d3: file exx.c, line 12.
(gdb) del 12
No breakpoint number 12.
(gdb) del 1
(gdb) b 9
Breakpoint 2 at 0x804849a: file exx.c, line 9.
(gdb) info breakpoints
Num   Type             Disp Enb Address      What
2     breakpoint      keep y   0x0804849a in main at exx.c:9
(gdb) run
Starting program: /home/elza/a.out
23
-7

Breakpoint 2, main () at exx.c:9
9      if (a >= 5 && b <= 0) {
(gdb) next
10         result = 8 * a + 4 * b;
(gdb) next
16         printf("%d\n", result);
(gdb) c
Continuing.
156
[Inferior 1 (process 13859) exited normally]
(gdb) run
Starting program: /home/elza/a.out
23
25

Breakpoint 2, main () at exx.c:9
9      if (a >= 5 && b <= 0) {
(gdb) c
Continuing.
48
[Inferior 1 (process 14107) exited normally]
(gdb) run
Starting program: /home/elza/a.out
6
0
```

```
Breakpoint 2, main () at exx.c:9
9      if (a >= 5 && b <= 0) {
(gdb) c
Continuing.
48
[Inferior 1 (process 14160) exited normally]
(gdb) quit
elza@elza-R428-P428-R478:~$ gdb a.out
GNU gdb (Ubuntu 7.7.1-0ubuntu5~14.04.3) 7.7.1
Copyright (C) 2014 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 "i686-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...done.
(gdb) b 7
Breakpoint 1 at 0x8048492: file exx.c, line 7.
(gdb) run
Starting program: /home/elza/a.out
0
5
```

```
Breakpoint 1, main () at exx.c:8
8      int result = 0;
(gdb) next
9      if (a >= 5 && b <= 0) {
(gdb) next
11     } else if (a <= -10 && b > 25) {
(gdb) next
14         result = a + b;
(gdb) info locals
a = 0
b = 5
result = 0
(gdb) c
Continuing.
5
[Inferior 1 (process 14433) exited normally]
(gdb) run
Starting program: /home/elza/a.out
-5
0
```

```
Breakpoint 1, main () at exx.c:8
8      int result = 0;
(gdb) set var a = -10
(gdb) set var b = -20
(gdb) print result
$1 = -1208225792
(gdb) c
Continuing.
-30
[Inferior 1 (process 14624) exited normally]
(gdb) quit
elza@elza-R428-P428-R478:~$
```

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. или дом.	Дата	Время	Событие	Действие по исправлению	Примечание
1						
2						

10. Замечания автора по существу работы : замечаний нет.

11. Выводы : если знать, как правильно пользоваться интерактивным отладчиком, то, думаю, нам эти знания еще понадобятся. Ну и знакомство с интерактивным отладчиком полезно для углубления в написании кода.

Недочёты при выполнении задания могут быть устранены следующим образом: _____

Подпись студента _____