**Отчет по лабораторной работе №10** по курсу Архитектура компьютера и информационных систем

Студент группы М8О-103Б-22 Касумова Наида Рашидовна, № по списку 9

Контакты www, e-mail, icq, skype naida.kasumova.04@mai.ru

Работа выполнена: « » \_\_\_\_\_\_\_\_\_\_\_\_202\_\_\_г.

Преподаватель: доцент каф. 806 Никулин Сергей Петрович

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

Отчет сдан « » \_\_\_\_\_\_\_\_\_202 \_\_ г., итоговая оценка \_\_\_\_\_

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

* 1. **Тема:** Отладчик системы программирования OC Unix

1. **Цель работы:** Научиться отлаживать простейшие программы, написанные на языке Си

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

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

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

1. **Задание** (*вариант №* )**:**

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

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

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

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

1. **Оборудование** (лабораторное):

ЭВМ Intel Pentium G2140, процессор 3.30 GHz , имя узла сети Cameron с ОП 8096 Мб, НМД 7906 Мб. Терминал ASUS адрес dev/pets/3 Принтер

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

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

*Оборудование ПЭВМ студента, если использовалось:*

Процессор Intel Core i5 с ОП 8172 Мб, НМД 256000 Мб. Монитор \_2560\*1600\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Другие устройства Multi-Touch \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

Операционная система семейства Unix , наименование Ubuntu версия 18.15.0

интерпретатор команд bash версия 4.4.20

Система программирования GNU версия 5.8.13

Редактор текстов версия 25.2.2

Утилиты операционной системы gcc, cat

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

Прикладные системы и программы

Местонахождение и имена файлов программ и данных stud/208104

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

*Программное обеспечение ЭВМ студента, если использовалось:*

Операционная система семейства \_macOS\_\_\_\_\_, наименование \_\_Big Sur\_\_\_\_ версия \_\_\_11.5.2\_ \_\_\_\_\_\_\_\_\_\_

интерпретатор команд \_bash\_\_\_\_\_\_\_\_\_ версия \_\_\_\_2.11\_\_\_\_.

Система программирования \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_версия \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

Утилиты операционной системы gcc,cat\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

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

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

Местонахождение и имена файлов программ и данных на домашнем компьютере \_\_\_\_\_\_home/Temich &\_\_\_\_\_\_\_\_

**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: 0

Output: Error

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 строго до начала лабораторной работы.*

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

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

**(gdb) list**

**1 /\* Лабораторная работа №10.**

**2 \* Программа вычисления среднего арифмитического n целых чисел (программа с возможной ошибкой во время исполнения).**

**3 \* Число n вводится с клавиатуры**

**4 \* Студент группы 08-101 А.Ядров\*/**

**5**

**6 #include <stdio.h>**

**7**

**8 int main() {**

**9 int a, n, p = 0;**

**10 int s = 0;**

**(gdb)**

**11 printf("%s\n", "Введите число n");**

**12 scanf("%d", &n); // Считывание количества чисел**

**13 printf("Введите числа \n");**

**14 for (int i = 0; i < n; i++) {**

**15 scanf("%d", &a);**

**16 p += a;**

**17 }**

**18 s = p/n;**

**19 printf("%f \n", s);**

**20 return 0;**

**(gdb) break 10**

**Breakpoint 1 at 0x401155: file main.c, line 10.**

**(gdb) break 18**

**Breakpoint 2 at 0x4011b7: file main.c, line 18.**

**(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/Temich/CLionProjects/laabs/a.out 1 1**

**Breakpoint 1, main () at main.c:10**

**10 int s = 0;**

**(gdb) print a**

**$1 = -7824**

**(gdb) print n**

**$2 = 0**

**(gdb) print p**

**$3 = 0**

**(gdb) set var a = 1000**

**(gdb) print a**

**$4 = 1000**

**(gdb) ptype a**

**type = int**

**(gdb) bt**

**#0 main () at main.c:10**

**(gdb) continue**

**Continuing.**

**Введите число n**

**0**

**Введите числа**

**Breakpoint 2, main () at main.c:18**

**18 s = p/n;**

**(gdb) print n**

**$5 = 0**

**(gdb) print s**

**$6 = 0**

**(gdb) set var n = 1**

**(gdb) continue**

**Continuing.**

**0.000000**

**[Inferior 1 (process 23551) exited normally]**

**(gdb) quit**

n@MacBook-Pro-N doc % gcc -g lw10.c

n@MacBook-Pro-N doc % gdb a.out

**GNU gdb (GDB) 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-apple-darwin20.6.0".

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...

Reading symbols from /Users/n/Desktop/doc/a.out.dSYM/Contents/Resources/DWARF/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) list

1 #include <stdio.h>

2 #include <math.h>

3

4 int main(void)

5 {

6 float fahr;

7 for (fahr = 0; fahr <= 300; fahr = fahr + 20)

8 printf("%3.2f : %6.2f\n", fahr, (5.0/9.0)\*(fahr - 32));

9 return 0;

10 }

(gdb) help run

**run**, **r**

Start debugged program.

You may specify arguments to give it.

Args may include "\*", or "[...]"; they are expanded using the

shell that will start the program (specified by the "$SHELL" environment

variable). Input and output redirection with ">", "<", or ">>"

are also allowed.

With no arguments, uses arguments last specified (with "run" or

"set args"). To cancel previous arguments and run with no arguments,

use "set args" without arguments.

To start the inferior without using a shell, use "set startup-with-shell off".

(gdb) set args 1 1

(gdb) show args

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

(gdb) bt

No stack.

(gdb) set args bla alb lab

(gdb) run

Starting program: /Users/n/Desktop/doc/a.out bla alb lab

Unable to find Mach task port for process-id 1825: (os/kern) failure (0x5).

(please check gdb is codesigned - see taskgated(8))

(gdb) quit

n@MacBook-Pro-N doc %

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

1. **Выводы**

Я научилась отлаживать простейшие программы, написанные на языке Си.

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

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