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

Фундаментальная информатика

Студент группы М8О-113Б-22 Горин Никита Андреевич, № по списку 6

Контакты www, e-mail, icq, skype exprompar@gmail.com

Работа выполнена: «11» \_\_ноября\_\_\_\_\_\_\_\_\_\_202\_2\_\_г.

Преподаватель: \_\_доц.\_каф. 806 Никулин Сергей Петрович

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

Отчет сдан « » \_\_\_\_\_\_\_\_\_201 \_\_ г., итоговая оценка \_\_\_\_\_

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

* 1. **Тема** Отладчик системы программирования OC Unix \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

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

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

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

1. **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 i7-11400 с ОП 15360 Мб, НМД 716800 Мб. Монитор \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Другие устройства Мышь Logitech G403\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

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 20.04.3 LTS

интерпретатор команд bash версия 5.0.17

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

Редактор текстов emacs версия 28.2

Утилиты операционной системы \_\_\_\_\_\_\_\_\_\_\_\_\_\_Стандартные утилиты Linux

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

Прикладные системы и программы Стандартные 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)

|  |  |
| --- | --- |
| 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].

Ознакомиться с командами отладчика, представленными выше, с помощью их использования на реальной программе Си в терминале.

*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,подписанный

преподавателем).

user@user-Katana:~$ cd C

user@user-Katana:~/C$ gcc -g Program10.c

user@user-Katana:~/C$ gdb a.out

GNU gdb (Ubuntu 12.0.90-0ubuntu1) 12.0.90

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 running

Running the program.

List of commands:

advance -- Continue the program up to the given location (same form as args for break command).

attach -- Attach to a process or file outside of GDB.

continue, fg, c -- Continue program being debugged, after signal or breakpoint.

detach -- Detach a process or file previously attached.

detach checkpoint -- Detach from a checkpoint (experimental).

detach inferiors -- Detach from inferior ID (or list of IDS).

disconnect -- Disconnect from a target.

finish, fin -- Execute until selected stack frame returns.

handle -- Specify how to handle signals.

inferior -- Use this command to switch between inferiors.

interrupt -- Interrupt the execution of the debugged program.

jump, j -- Continue program being debugged at specified line or address.

kill -- Kill execution of program being debugged.

kill inferiors -- Kill inferior ID (or list of IDs).

next, n -- Step program, proceeding through subroutine calls.

nexti, ni -- Step one instruction, but proceed through subroutine calls.

queue-signal -- Queue a signal to be delivered to the current thread when it is resumed.

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

reverse-continue, rc -- Continue program being debugged but run it in reverse.

reverse-finish -- Execute backward until just before selected stack frame is called.

reverse-next, rn -- Step program backward, proceeding through subroutine calls.

reverse-nexti, rni -- Step backward one instruction, but proceed through called subroutines.

reverse-step, rs -- Step program backward until it reaches the beginning of another source line.

reverse-stepi, rsi -- Step backward exactly one instruction.

run, r -- Start debugged program.

signal -- Continue program with the specified signal.

start -- Start the debugged program stopping at the beginning of the main procedure.

starti -- Start the debugged program stopping at the first instruction.

step, s -- Step program until it reaches a different source line.

stepi, si -- Step one instruction exactly.

taas -- Apply a command to all threads (ignoring errors and empty output).

target -- Connect to a target machine or process.

target core -- Use a core file as a target.

target ctf -- (Use a CTF directory as a target.

target exec -- Use an executable file as a target.

target extended-remote -- Use a remote computer via a serial line, using a gdb-specific protocol.

target native -- Native process (started by the "run" command).

target record-btrace -- Collect control-flow trace and provide the execution history.

target record-core -- Log program while executing and replay execution from log.

target record-full -- Log program while executing and replay execution from log.

target remote -- Use a remote computer via a serial line, using a gdb-specific protocol.

target tfile -- Use a trace file as a target.

task -- Use this command to switch between Ada tasks.

task apply -- Apply a command to a list of tasks.

task apply all -- Apply a command to all tasks in the current inferior.

tfaas -- Apply a command to all frames of all threads (ignoring errors and empty output).

thread, t -- Use this command to switch between threads.

thread apply -- Apply a command to a list of threads.

thread apply all -- Apply a command to all threads.

thread find -- Find threads that match a regular expression.

thread name -- Set the current thread's name.

until, u -- Execute until past the current line or past a LOCATION.

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 /\*

3 Горин Никита Андреевич М8О-113Б-22

4 Программа нахождения факториала числа N (0 <= N <= 1000)

5 Так как ответ может быть очень большим, вывести его по модулю 2027

6 \*/

7

8 int main(){

9 int n;

10 scanf("%d", &n);

(gdb)

11 int res = 1;

12 for(int i = 1; i <= n; i++){

13 res \*= i;

14 res %= 2027;

15 }

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

17 return 0;

18 }

(gdb) break 11

Breakpoint 1 at 0x11bf: file Program10.c, line 11.

(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/user/C/a.out 1 1

[Thread debugging using libthread\_db enabled]

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

14

Breakpoint 1, main () at Program10.c:11

11 int res = 1;

(gdb) print res

$1 = 0

(gdb) print n

$2 = 14

(gdb) set var n = 15

(gdb) print n

$3 = 15

(gdb) next

12 for(int i = 1; i <= n; i++){

(gdb) next

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

(gdb) ptype n

type = int

(gdb) bt

#0 main () at Program10.c:16

(gdb) continue

Continuing.

1188

[Inferior 1 (process 10683) exited normally]

(gdb) run

Starting program: /home/user/C/a.out

[Thread debugging using libthread\_db enabled]

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

15

Breakpoint 1, main () at Program10.c:11

warning: Source file is more recent than executable.

11 int res = 1;

(gdb) step

12 for(int i = 1; i <= n; i++){

(gdb) step

13 res \*= i;

(gdb) step

14 res %= 2027;

(gdb) step

12 for(int i = 1; i <= n; i++){

(gdb) step

13 res \*= i;

(gdb) step

14 res %= 2027;

(gdb) step

12 for(int i = 1; i <= n; i++){

(gdb) step

13 res \*= i;

(gdb) continue

Continuing.

1188

[Inferior 1 (process 11346) exited normally]

(gdb) quit

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 научился отлаживать простейшие программы, написанные на языке Си

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

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

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

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

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

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

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

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

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

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

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