

Отчет по лабораторной работе № 10 по курсу Архитектура компьютера и информационных сетей

Студент группы М8О-103Б-22 Киселев Артём Олегович, № по списку 10

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

Работа выполнена: 2022 г.

Преподаватель: доцент каф. 806 Никулин С.П.

Входной контроль знаний с оценкой _____

Отчет сдан « 25 » ноября 2022 г., итоговая оценка ____

Подпись преподавателя _____

1. **Тема:** Отладчик системы программирования ОС UNIX _____

2. **Цель работы:** Научиться пользоваться отладчиком системы программирования ОС UNIX. _____

3. **Задание (вариант №):** _____

4. **Оборудование (лабораторное):**
ЭВМ _____, процессор _____, имя узла сети _____ с ОП _____ Мб,
НМД _____ Мб. Терминал _____ адрес _____. Принтер _____
Другие устройства _____

Оборудование ПЭВМ студента, если использовалось:

Процессор Ryzen 3 3200u 2.6GHz _ с ОП 8 ГБ _ НМД SSD 256 ГБ, HDD 1000 ГБ _ . Монитор Встроенный 1920x1080

Другие устройства Touchpad Synaptics

5. **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 UNIX _____, наименование Ubuntu _____ версия 22.04.1

интерпретатор команд bash _____ версия 5.1.16

Система программирования C _____ версия _____

Редактор текстов Visual Studio Code _____ версия 28.2

Утилиты операционной системы _____

Прикладные системы и программы gcc, gdb

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

C - компилируемый статически типизированный язык программирования общего назначения.

gcc - компилятор.

Visual Studio Code - текстовый редактор.

gdb - отладчик.

help - подсказка по разделу помощи отладчика.

list - распечатка текста функции/процедуры/файла или всей программы, начиная с указанной строки.

break - задание точки остановки на указанной строке/функции текущего исходного файла программы.

run - запуск программы на выполнение.

set args - предварительная установка параметров командной строки.

print - печать значения выражения.

next - выполнение очередной строки программы при пошаговой трассировке.

step - выполнение очередной строки программы.

set var - присваивание значения переменной.

ptype - распечатка определения типа переменной.

backtrace или bt - распечатка содержимого стека вызовов.

continue - продолжение выполнения программы после остановки.

quit - выход из отладчика.

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

1) Изучить основные команды отладчика gdb.

2) Прodelать и заpотоколиpовать все описанные действия над некоторой программой на языке Си.

3) Сделать отчет.

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

Допущен к выполнению работы. Подпись преподавателя _____

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

```
jonaj@JonAJ:~/Desktop/StudyDir/Labs/lab10$ gcc lab010.c -g
jonaj@JonAJ:~/Desktop/StudyDir/Labs/lab10$ 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/>.
--Type <RET> for more, q to quit, c to continue without paging--c

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.
--Type <RET> for more, q to quit, c to continue without paging--c
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 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.
--Type <RET> for more, q to quit, c to continue without paging--c
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.
condition -- Specify breakpoint number N to break only if COND is true.
delete, del, d -- Delete all or some breakpoints.
delete bookmark -- Delete a bookmark from the bookmark list.
delete breakpoints -- Delete all or some breakpoints or auto-display expressions.
delete checkpoint -- Delete a checkpoint (experimental).
delete display -- Cancel some expressions to be displayed when program stops.
delete mem -- Delete memory region.
delete tracepoints, delete tr -- Delete specified tracepoints.
delete tvariable -- Delete one or more trace state variables.
disable, disa, dis -- Disable all or some breakpoints.
```

disable breakpoints -- Disable all or some breakpoints.
 disable display -- Disable some expressions to be displayed when program stops.
 disable frame-filter -- GDB command to disable the specified frame-filter.
 disable mem -- Disable memory region.
 disable pretty-printer -- GDB command to disable the specified pretty-printer.
 disable probes -- Disable probes.
 disable type-printer -- GDB command to disable the specified type-printer.
 disable unwinder -- GDB command to disable the specified unwinder.
 disable xmethod -- GDB command to disable a specified (group of) xmethod(s).
 dprintf -- Set a dynamic printf at specified location.
 enable, en -- Enable all or some breakpoints.
 enable breakpoints -- Enable all or some breakpoints.
 enable breakpoints count -- Enable some breakpoints for COUNT hits.
 enable breakpoints delete -- Enable some breakpoints and delete when hit.
 enable breakpoints once -- Enable some breakpoints for one hit.
 enable count -- Enable some breakpoints for COUNT hits.
 enable delete -- Enable some breakpoints and delete when hit.
 enable display -- Enable some expressions to be displayed when program stops.
 enable frame-filter -- GDB command to enable the specified frame-filter.
 enable mem -- Enable memory region.
 enable once -- Enable some breakpoints for one hit.
 enable pretty-printer -- GDB command to enable the specified pretty-printer.
 enable probes -- Enable probes.
 enable type-printer -- GDB command to enable the specified type printer.
 enable unwinder -- GDB command to enable unwinders.
 enable xmethod -- GDB command to enable a specified (group of) xmethod(s).
 ftrace -- Set a fast tracepoint at specified location.
 hbreak -- Set a hardware assisted breakpoint.
 ignore -- Set ignore-count of breakpoint number N to COUNT.
 rbreak -- Set a breakpoint for all functions matching REGEXP.
 rwatch -- Set a read watchpoint for EXPRESSION.
 save -- Save breakpoint definitions as a script.
 save breakpoints -- Save current breakpoint definitions as a script.
 save gdb-index -- Save a gdb-index file.
 save tracepoints -- Save current tracepoint definitions as a script.
 skip -- Ignore a function while stepping.
 skip delete -- Delete skip entries.
 skip disable -- Disable skip entries.
 skip enable -- Enable skip entries.
 skip file -- Ignore a file while stepping.
 skip function -- Ignore a function while stepping.
 strace -- Set a static tracepoint at location or marker.
 tbreak -- Set a temporary breakpoint.
 tcatch -- Set temporary catchpoints to catch events.
 tcatch assert -- Catch failed Ada assertions, when raised.
 tcatch catch -- Catch an exception, when caught.
 tcatch exception -- Catch Ada exceptions, when raised.
 tcatch exec -- Catch calls to exec.
 tcatch fork -- Catch calls to fork.
 tcatch handlers -- Catch Ada exceptions, when handled.
 tcatch load -- Catch loads of shared libraries.
 tcatch rethrow -- Catch an exception, when rethrown.
 tcatch signal -- Catch signals by their names and/or numbers.
 tcatch syscall -- Catch system calls by their names, groups and/or numbers.
 tcatch throw -- Catch an exception, when thrown.
 tcatch unload -- Catch unloads of shared libraries.
 tcatch vfork -- Catch calls to vfork.
 thbreak -- Set a temporary hardware assisted breakpoint.
 trace, trac, tra, tr, tp -- Set a tracepoint at specified location.
 watch -- Set a watchpoint for EXPRESSION.

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      int main() {
4
5          int a = 10;
6          int b = 0;
7          int c = a / b;
8
9          printf("%d",c);
10
(gdb) break 5
Breakpoint 1 at 0x1155: file lab010.c, line 5.
(gdb) set args Test
(gdb) show args
Argument list to give program being debugged when it is started is "Test".
(gdb) run a.out
Starting program: /home/jonaj/Desktop/StudyDir/Labs/lab10/a.out 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 lab010.c:5
5      int a = 10;
(gdb) print a
$1 = 0
(gdb) next
6      int b = 0;
(gdb) step
7      int c = a / b;
(gdb) set var b = 1
(gdb) print a
$2 = 10
(gdb) print b
$3 = 1
(gdb) print c
$4 = 21845
(gdb) step
9      printf("%d",c);
(gdb) print c
$5 = 10
(gdb) ptype c
type = int
(gdb) bt
#0  main () at lab010.c:9
(gdb) backtrace
#0  main () at lab010.c:9
(gdb) continue
Continuing.
10[Inferior 1 (process 38842) exited normally]
(gdb) quit
jonaj@JonAJ:~/Desktop/StudyDir/Labs/lab10$
```

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 ЭВМ, существенном участии преподавателя и других лиц в написании и отладке программы.

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

10. Замечания автора по существу работы:

11. Выводы: в процессе были изучены основы работы с отладчиком системы программирования ОС UNIX.

Недочёты при выполнении задания могут быть устранены следующим образом:

Подпись студента ЯД