Учреждение образования
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»
Кафедра информатики

Отчет по лабораторной работе №6

Компиляция и запуск программы на C

Выполнил: Студент гр. 053502
Бокун Артем Геннадьевич

Руководитель: ст. преподаватель
Шиманский В.В.

Минск 2021

СОДЕРЖАНИЕ

1. Введение

Целью данной работы является изучить следующий материал:
1) Компиляция программ на C с помощью командной строки;
2) Работа с GNU Compiler Collection;
3) Работа с Valgrind/Sanitizer;
4) Компиляция с помощью Makefile;
5) Битовые операции

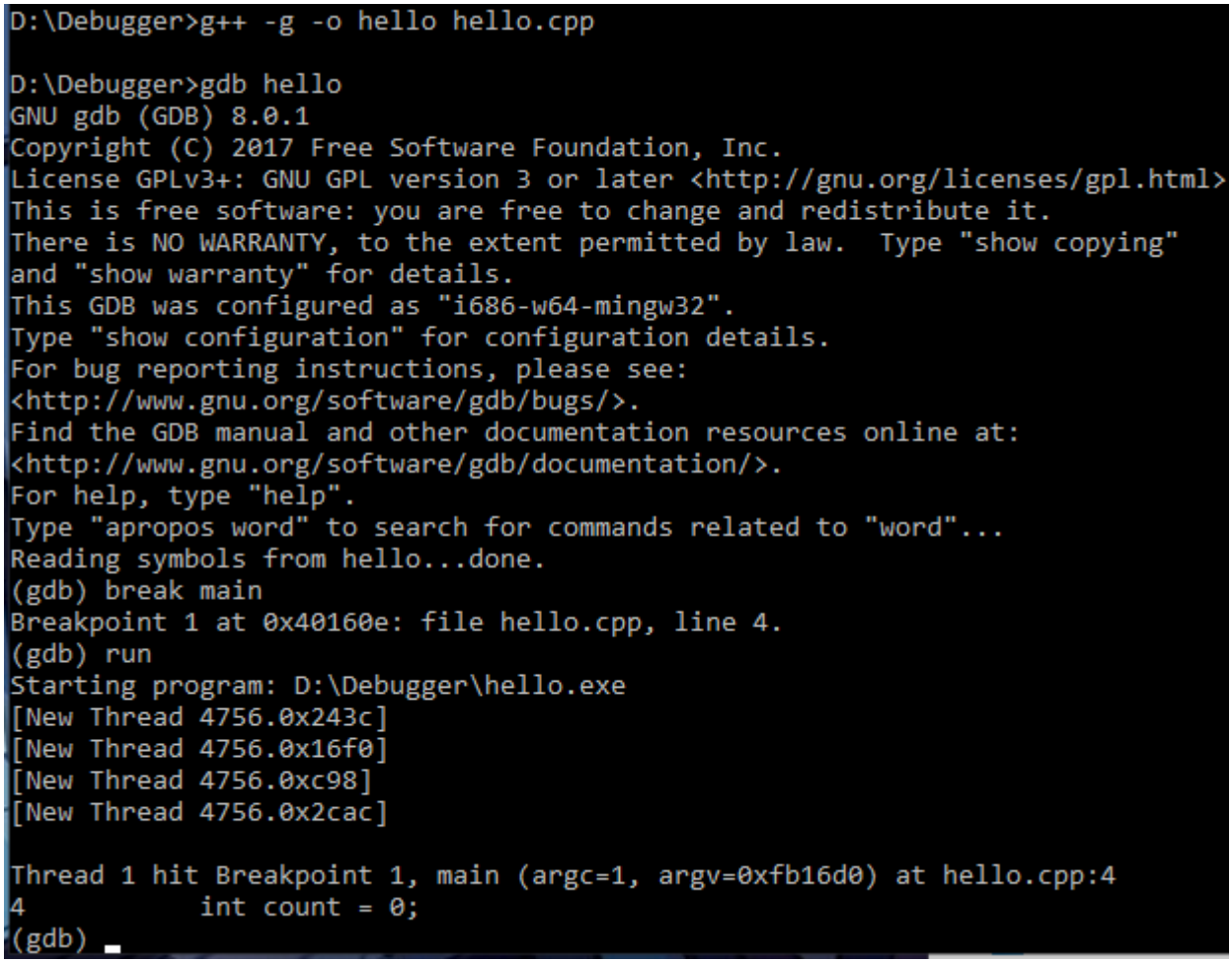
2. Выполнение задач

2.1 Дебаг с помощью GCC

1) Установка точки останова на функции main:

```
D:\Debugger>echo off
Microsoft Windows [Version 10.0.19043.1348]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

D:\Debugger>g++ -g -o hello hello.cpp

D:\Debugger>gdb hello
GNU gdb (GDB) 8.0.1
Copyright (C) 2017 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-w64-mingw32".
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 hello...done.
(gdb) break main
Breakpoint 1 at 0x40160e: file hello.cpp, line 4.
(gdb) _
```

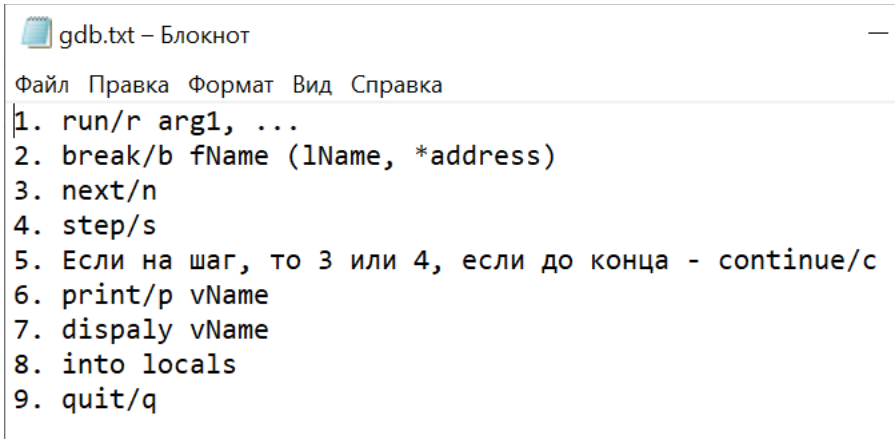
2) Использование команды run:



3) Использование команды step и continue:

```
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 hello...done.
(gdb) break main
Breakpoint 1 at 0x40160e: file hello.cpp, line 4.
(gdb) run
Starting program: D:\Debugger\hello.exe
[New Thread 4192.0xd20]
[New Thread 4192.0xd10]
[New Thread 4192.0xcc0]
[New Thread 4192.0xca4]

Thread 1 hit Breakpoint 1, main (argc=1, argv=0xf916d0) at hello.cpp:4
4           int count = 0;
(gdb) step
5           int *p = &count;
(gdb) step
7           for (int i = 0; i < 10; i++) {
(gdb) step
8               (*p)++; // What is written here?
(gdb) step
7           for (int i = 0; i < 10; i++) {
(gdb) step
8               (*p)++; // What is written here?
(gdb) step
7           for (int i = 0; i < 10; i++) {
(gdb) step
8               (*p)++; // What is written here?
(gdb) step
7           for (int i = 0; i < 10; i++) {
(gdb) step
8               (*p)++; // What is written here?
(gdb) continue
Continuing.
Thank you for running this code. Have a nice day![Thread 4192.0xcc0 exited with code 0]
[Thread 4192.0xca4 exited with code 0]
[Thread 4192.0xd10 exited with code 0]
[Inferior 1 (process 4192) exited normally]
(gdb) _
```

4) Ответ на вопросы:



2.2 Интерактивная отладка

Отладка для программ, где предполагается ввод данных может происходить двумя способами: с подгрузкой данных из файла и с вводом данных из консоли. Ниже приведены оба варианта:

```
D:\Debugger>echo off
Microsoft Windows [Version 10.0.19043.1348]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

D:\Debugger>g++ -g -o int_hello interactive_hello.cpp

D:\Debugger>gdb int_hello
GNU gdb (GDB) 8.0.1
Copyright (C) 2017 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-w64-mingw32".
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 int_hello...done.
(gdb) run                                  <- без загрузки из файла
Starting program: D:\Debugger\int_hello.exe
[New Thread 6768.0x2778]
[New Thread 6768.0x272c]
[New Thread 6768.0x27e8]
[New Thread 6768.0x280]
Introduce yourself, please
Vladimir
Hi, Vladimir
I just wanted to say hello!
Have a nice day.
[Thread 6768.0x272c exited with code 0]
[Thread 6768.0x27e8 exited with code 0]
[Thread 6768.0x280 exited with code 0]
[Inferior 1 (process 6768) exited normally]
(gdb) run < input.txt                      <- с загрузкой из файла
Starting program: D:\Debugger\int_hello.exe < input.txt
[New Thread 6276.0xc54]
[New Thread 6276.0x2168]
[New Thread 6276.0x2874]
[New Thread 6276.0x2420]
Introduce yourself, please
Hi, Valerii
I just wanted to say hello!
Have a nice day.
[Thread 6276.0x2420 exited with code 0]
[Thread 6276.0x2168 exited with code 0]
[Thread 6276.0x2874 exited with code 0]
[Inferior 1 (process 6276) exited normally]
(gdb)
```

2.3 Работа с Valgrind/Sanitizer

К сожалению, для ОС Windows отсутствует Valgrind, поэтому я использую его аналог — DrMemory для выполнения задания.

1) segfault_ex

Оригинальный код:

```
int main() {
    int a[20];
    for (int i = 0; ; i++) {
        a[i] = i;
    }
}
```

Проверка с помощью DrMemory:

```
D:\Debugger>echo off
Microsoft Windows [Version 10.0.19043.1348]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

D:\Debugger>g++ -g -o a segfault_ex.cpp

D:\Debugger>D:\Debugger\drmem\bin\drmemory.exe -ignore_kernel -batch -logdir . -- a.exe
~~Dr.M~~ Dr. Memory version 2.5.0
~~Dr.M~~ Running "a.exe"
~~Dr.M~~ System call information is missing for this operating system: WinVer=105;Rel=2009;Build=19043;Edition=Enterprise.
~~Dr.M~~
~~Dr.M~~ Error #1: UNADDRESSABLE ACCESS beyond heap bounds: writing 0x00610000-0x00610004 4 byte(s)
~~Dr.M~~ # 0 main               [D:\Debugger/segfault_ex.cpp:4]
~~Dr.M~~ Note: @0:00:01.772 in thread 10028
~~Dr.M~~ Note: instruction: mov    %edx -> 0x0c(%esp,%eax,4)
~~Dr.M~~
~~Dr.M~~ Error #2: WARNING: writing to readonly memory 0x00611000-0x00611004
~~Dr.M~~ # 0 main               [D:\Debugger/segfault_ex.cpp:4]
~~Dr.M~~ Note: @0:00:01.837 in thread 10028
~~Dr.M~~ Note: instruction: mov    %edx -> 0x0c(%esp,%eax,4)
```

Видно, что DrMemory обнаружил попытку записи в сегмент памяти, который помечен как readonly и выход за границы массива.

Исправленный код:

```
int main() {
    int a[20];
    for (int i = 0; i < 20; i++) {
        a[i] = i;
    }
}
```

Проверка с помощью DrMemory:

```
D:\Debugger>echo off
Microsoft Windows [Version 10.0.19043.1348]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

D:\Debugger>g++ -g -o a  segfault_ex.cpp

D:\Debugger>D:\Debugger\drmem\bin\drmemory.exe -ignore_kernel -batch -logdir . -- a.exe
~~Dr.M~~ Dr. Memory version 2.5.0
~~Dr.M~~ Running "a.exe"
~~Dr.M~~ System call information is missing for this operating system: WinVer=105;Rel=2009;Build=19043;Edition=Enterprise.
~~Dr.M~~
~~Dr.M~~ NO ERRORS FOUND:
~~Dr.M~~       0 unique,     0 total unaddressable access(es)
~~Dr.M~~       0 unique,     0 total uninitialized access(es)
~~Dr.M~~       0 unique,     0 total invalid heap argument(s)
~~Dr.M~~       0 unique,     0 total GDI usage error(s)
~~Dr.M~~       0 unique,     0 total handle leak(s)
~~Dr.M~~       0 unique,     0 total warning(s)
~~Dr.M~~       0 unique,     0 total,      0 byte(s) of leak(s)
~~Dr.M~~       0 unique,     0 total,      0 byte(s) of possible leak(s)
~~Dr.M~~ ERRORS IGNORED:
~~Dr.M~~       3 unique,     3 total,     22 byte(s) of still-reachable allocation(s)
~~Dr.M~~          (re-run with "-show_reachable" for details)
~~Dr.M~~ Details: D:\Debugger\DrMemory-a.exe.760.000\results.txt

D:\Debugger>
```

В исправленном коде ошибок не обнаружено.

2) no_segfault_ex

Оригинальный код:

```
#include <iostream>
int main() {
    int a[5] = {1, 2, 3, 4, 5};
    unsigned total = 0;
    for (int j = 0; j < sizeof(a); j++) {
        total += a[j];
    }
    std::cout << "sum of array is " << total << std::endl;
}
```

Проверка с помощью DrMemory:

```
D:\Debugger>echo off
Microsoft Windows [Version 10.0.19043.1348]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

D:\Debugger>g++ -g -o a no_segfault_ex.cpp

D:\Debugger>D:\Debugger\drmem\bin\drmemory.exe -ignore_kernel -batch -logdir . -- a.exe
~~Dr.M~~ Dr. Memory version 2.5.0
~~Dr.M~~ Running "a.exe"
~~Dr.M~~ System call information is missing for this operating system: WinVer=105;Rel=2009;Build=19043;Edition=Enterprise.
sum of array is ~~Dr.M~~
~~Dr.M~~ Error #1: UNINITIALIZED READ: reading register edx
~~Dr.M~~ # 0 libstdc++-6.dll!?               +0x0      (0x6feee621 <libstdc++-6.dll+0xae621>)
~~Dr.M~~ # 1 libstdc++-6.dll!?               +0x0      (0x6fe8fe4d <libstdc++-6.dll+0x4fe4d>)
~~Dr.M~~ # 2 libstdc++-6.dll!?               +0x0      (0x6fea4a48 <libstdc++-6.dll+0x64a48>)
~~Dr.M~~ # 3 _fu0___ZSt4cout                  [D:\Debugger/no_segfault_ex.cpp:8]
~~Dr.M~~ # 4 __tmainCRTStartup
~~Dr.M~~ # 5 KERNEL32.dll!BaseThreadInitThunk +0x18     (0x7581fa29 <KERNEL32.dll+0x1fa29>)
~~Dr.M~~ Note: @0:00:02.222 in thread 5732
~~Dr.M~~ Note: instruction: test   %edx %edx
~~Dr.M~~
~~Dr.M~~ Error #2: UNINITIALIZED READ: reading register ecx
~~Dr.M~~ # 0 libstdc++-6.dll!?               +0x0      (0x6feee623 <libstdc++-6.dll+0xae623>)
~~Dr.M~~ # 1 libstdc++-6.dll!?               +0x0      (0x6fe8fe4d <libstdc++-6.dll+0x4fe4d>)
~~Dr.M~~ # 2 libstdc++-6.dll!?               +0x0      (0x6fea4a48 <libstdc++-6.dll+0x64a48>)
~~Dr.M~~ # 3 _fu0___ZSt4cout                  [D:\Debugger/no_segfault_ex.cpp:8]
~~Dr.M~~ # 4 __tmainCRTStartup
~~Dr.M~~ # 5 KERNEL32.dll!BaseThreadInitThunk +0x18     (0x7581fa29 <KERNEL32.dll+0x1fa29>)
~~Dr.M~~ Note: @0:00:02.223 in thread 5732
~~Dr.M~~ Note: instruction: movzx  0x04(%esi,%ecx) -> %eax
2234921223
~~Dr.M~~
~~Dr.M~~ ERRORS FOUND:
~~Dr.M~~       0 unique,     0 total unaddressable access(es)
~~Dr.M~~       2 unique,     2 total uninitialized access(es)
~~Dr.M~~       0 unique,     0 total invalid heap argument(s)
~~Dr.M~~       0 unique,     0 total GDI usage error(s)
~~Dr.M~~       0 unique,     0 total handle leak(s)
~~Dr.M~~       0 unique,     0 total warning(s)
~~Dr.M~~       0 unique,     0 total,      0 byte(s) of leak(s)
~~Dr.M~~       0 unique,     0 total,      0 byte(s) of possible leak(s)
~~Dr.M~~ ERRORS IGNORED:
~~Dr.M~~       3 unique,     3 total,     22 byte(s) of still-reachable allocation(s)
~~Dr.M~~          (re-run with "-show_reachable" for details)
~~Dr.M~~ Details: D:\Debugger\DrMemory-a.exe.10628.000\results.txt
~~Dr.M~~ Fetching 1 symbol files...
~~Dr.M~~ [1/1] Fetching symbols for C:\WINDOWS\System32\msvcp_win.dll
~~Dr.M~~ Fetched 0 symbol files successfully

D:\Debugger>
```

Видно, что DrMemory обнаружил попытку чтения из защищенных регистров edx и ecx.

Исправленный код:

```
#include <iostream>
int main() {
    int a[5] = {1, 2, 3, 4, 5};
    unsigned total = 0;
    for (int j = 0; j < sizeof(a)/sizeof(int); j++) {
        total += a[j];
    }
    std::cout << "sum of array is " << total << std::endl;
}
```

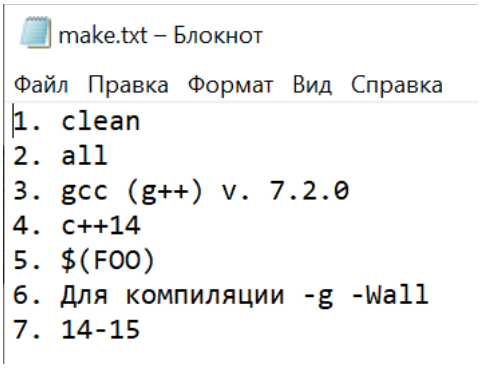
Проверка с помощью DrMemory:

```
Microsoft Windows [Version 10.0.19043.1348]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

D:\Debugger>g++ -g -o a  no_segfault_ex.cpp

D:\Debugger>D:\Debugger\drmem\bin\drmemory.exe -ignore_kernel -batch -logdir . -- a.exe
~~Dr.M~~ Dr. Memory version 2.5.0
~~Dr.M~~ Running "a.exe"
~~Dr.M~~ System call information is missing for this operating system: WinVer=105;Rel=2009;Build=19043;Edition=Enterprise.
sum of array is 15
~~Dr.M~~
~~Dr.M~~ NO ERRORS FOUND:
~~Dr.M~~       0 unique,     0 total unaddressable access(es)
~~Dr.M~~       0 unique,     0 total uninitialized access(es)
~~Dr.M~~       0 unique,     0 total invalid heap argument(s)
~~Dr.M~~       0 unique,     0 total GDI usage error(s)
~~Dr.M~~       0 unique,     0 total handle leak(s)
~~Dr.M~~       0 unique,     0 total warning(s)
~~Dr.M~~       0 unique,     0 total,      0 byte(s) of leak(s)
~~Dr.M~~       0 unique,     0 total,      0 byte(s) of possible leak(s)
~~Dr.M~~ ERRORS IGNORED:
~~Dr.M~~       3 unique,     3 total,     22 byte(s) of still-reachable allocation(s)
~~Dr.M~~          (re-run with "-show_reachable" for details)
~~Dr.M~~ Details: D:\Debugger\DrMemory-a.exe.12968.000\results.txt

D:\Debugger>
```

В исправленном коде ошибок не обнаружено.

2.4 Компиляция с помощью Makefile

Ответы на вопросы:



2.5 Битовые операции

Законченный код:

```
#include "bit_ops.h"

unsigned get_bit(unsigned x, unsigned n)
{
    return (x >> n) & 1;
}

void set_bit(unsigned * x, unsigned n, unsigned v)
{
    *x = (*x & ~(1 << n)) | (v << n);
}

void flip_bit(unsigned * x, unsigned n)
{
    *x = *x ^ (1 << n);
}
```

Запуск с помощью Makefile, проведение тестов:

```
D:\Debugger>echo off
Microsoft Windows [Version 10.0.19043.1348]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

D:\Debugger>D:\Debugger\mingw32\bin\mingw32-make.exe
process_begin: CreateProcess(NULL, uname -s, ...) failed.
mingw32-make: Makefile:1: pipe: No error
g++ -c -g -Wall bit_ops.cpp
g++ -c -g -Wall test_bit_ops.cpp
g++ -g -Wall -g -o bit_ops bit_ops.o test_bit_ops.o

D:\Debugger>bit_ops

Test get_bit()

get_bit(0x4e, 0): 0x0, True
get_bit(0x4e, 1): 0x1, True
get_bit(0x4e, 5): 0x0, True
get_bit(0x1b, 3): 0x1, True
get_bit(0x1b, 2): 0x0, True
get_bit(0x1b, 9): 0x0, True

Test set_bit()

get_bit(0x4e, 0): 0x0, True
set_bit(0x4e, 2, 0): 0x4a, True
set_bit(0x6d, 0, 0): 0x6c, True
set_bit(0x4e, 2, 1): 0x4e, True
set_bit(0x6d, 0, 1): 0x6d, True
set_bit(0x4e, 9, 0): 0x4e, True
set_bit(0x6d, 4, 0): 0x6d, True
set_bit(0x4e, 9, 1): 0x24e, True
set_bit(0x6d, 7, 1): 0xed, True

Test flip_bit()

flip_bit(0x4e, 0): 0x4f, True
flip_bit(0x4e, 1): 0x4c, True
flip_bit(0x4e, 2): 0x4a, True
flip_bit(0x4e, 5): 0x6e, True
flip_bit(0x4e, 9): 0x24e, True


D:\Debugger>_
```

3. Вывод

На практике было изучено: компиляция программ на C с помощью командной строки, работа с GNU Compiler Collection, работа с Valgrind/Sanitizer, компиляция с помощью Makefile, битовые операции.