

# Отчет по лабораторной работе № 10 по курсу “Фундаментальная информатика”

Студент группы М80-109Б-22 Малько Павел Алексеевич, № по списку 9

Контакты e-mail: [malko.pavel07@gmail.com](mailto:malko.pavel07@gmail.com)

Работа выполнена: «20» декабря 2022г.

Преподаватель: каф. 806 Сысоев Максим Алексеевич

Отчет сдан «    » \_\_\_\_\_ 20\_\_ г., итоговая оценка \_\_\_\_\_

Подпись преподавателя \_\_\_\_\_

1. **Тема:** Язык программирования С. Отладчик.
2. **Цель работы:** Применить дебаггер для исправления ошибок всех видов в С
3. **Задание:**  
Придумать ошибочную программу и отладить её. Должно быть 6 ошибочных случаев.
4. **Оборудование** (студента):  
Процессор *amd ryzen 5500u* с ОП 7851 Мб, НМД 1024 Гб. Монитор 1920x1080
5. **Программное обеспечение** (студента):  
Операционная система семейства: *linux*, наименование: *kali*, версия 2022.3  
интерпретатор команд: *bash* версия 0.16.1.  
Система программирования -- версия --, редактор текстов *emacs* версия 25.2.2  
Утилиты операционной системы --  
Прикладные системы и программы --  
Местонахождение и имена файлов программ и данных на домашнем компьютере –
6. **Идея, метод, алгоритм**  
-
7. **Сценарий выполнения работы**
  - 1.Посмотреть видео на тему ошибок в Си, примеров ошибок.
  - 2.Написать ошибочные программы
  - 3.Применить отладчик и ознакомиться с ним

## Compile-time error

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int a
7      int b;
8
9
10     c = a+b;
11 }
12
```

logs & others

Code::Blocks x Search results x Build log x Build messages x CppCheck

File	L...	Message
		=== Build: Debug in lab10 (compiler: MinGW) ===
		In function 'main':
C:\Users...	7	error: expected '=', ',', ';', 'asm' or '__attribute__' bef...
C:\Users...	10	error: 'c' undeclared (first use in this function)
C:\Users...	10	note: each undeclared identifier is reported only once for ...
C:\Users...	10	error: 'a' undeclared (first use in this function)
		=== Build failed: 3 error(s), 0 warning(s) (0 minute(s), 0 ...

При ошибке такого типа даже не происходит запуска программы. В данном случае ошибки две - в 6-ой строчке нет точки с запятой, а в 10-ой используется необъявленная переменная c. Исправим CE - объявим c и поставим точку с запятой. Сразу 2 ошибки в одной программе.

## Unpredictable behaviour

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int a;
    int b;
    int c;

    c = a+b;
}
```

Watches

Function arguments		
Locals		
a	4194432	int
b	2924544	int
c	0	int

Теперь произошла компиляция, но непонятно, какие a и b - пока им не присвоены значения - в них может быть что угодно. При попытке их сложения будет получаться странное число. Попробуем исправить это и присвоим какие-нибудь значения a и b.

```

#include <stdio.h>
#include <stdlib.h>

int main()
{
    int a=1000000000000000000;
    int b=1000000000000000000;
    int c;

    c = a+b;
}

```

Watches		
Function arguments		
Locals		
a	-1486618624	int
b	-1981284352	int
c	827064320	int

О нет, в переменной c вместо суммы a и b лежит странное число. Но ведь мы же их объявили, почему? Потому что тип int не может хранить настолько большие числа.

## Runtime-error. Segmentation fault.

Данная ошибка происходит при попытке доступа к запрещённому участку памяти. Например, создание слишком большого **статического** массива(см 14 лаб. работу - в ней из-за этого используется дин.массив). Или же деление на 0.

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int a[1000000];
7  }
8

```

### Call stack

Nr	Address	Function	File	Line
0	0x401292	__register_frame_info ()		
1	0x4012f5	mainCRTStartup ()		

### Logs & others

```

[debug]__chkstk_ms () at ../../../../src/gcc-6.3.0/libgcc
[debug]>>>>>cb_gdb:

```

Program received signal SIGSEGV, Segmentation fault.

main.c X

1 #include <stdio.h>  
2 #include <stdlib.h>  
3  
4 int main()  
5 {  
6 int a[10];  
7 a[10] = 5/0;  
8 }  
9

Watches

Function arguments  
Locals

Call stack

Nr	Address	Func...	File	Line
0	0x401479	main()	C:\Users\Pavel\Desktop\labs\10 lab\lab10\main.c	7

logs & others

Code::Blocks X Search results X Build log X Build messages X CppCheck/Ver...

[debug]>>>>>cb\_gdb:  
[debug]> bt 30  
[debug]#0 0x00401479 in main () at C:\Users\Pavel\Desktop\labs\10 lab\lab10\main.c:7  
[debug]>>>>>cb\_gdb:

[debug]>>>>>cb\_gdb:  
[debug]> bt 30  
[debug]#0 0x00401479 in main () at C:\Users\Pavel\Desktop\labs\10 lab\lab10\main.c:7  
[debug]>>>>>cb\_gdb:

## 9. Дневник отладки

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

## 10. Замечания автора замечаний нет

### 11. Выводы

Считаю эту работу полезной - уверен, что без явного поручения установить дебаггер и испытать его программисту может быть лень заниматься такими с его точки зрения скучными вещами. То же самое касается и типов ошибок. Но в то же время это необходимо знать и уметь применять.

Недочёты при выполнении задания могут быть устранены следующим образом: --

Подпись студента \_\_\_\_\_



