

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

Студент группы М80-109Б-22 Мозговой Никита Евгеньевич, № по списку 10

Контакты e-mail: MozgovoyNE@mail.ru, telegram: @M1N8E

Работа выполнена: «11» декабря 2022г.

Преподаватель: каф. 806 Сысоев Максим Алексеевич

Отчет сдан «» декабря 2022 г., итоговая оценка _____

Подпись преподавателя _____

1. **Тема:** программирование на языке C.
2. **Цель работы:** научиться пользоваться отладчиком.
3. **Задание (вариант №):** составить по 2 ошибки каждого типа и исправить их.
4. **Оборудование (студента):**

Процессор: 2,5 GHz 2-ядерный процессор Intel Core i5 ОП 4 Гб, Монитор 1280x800

5. **Программное обеспечение (студента):**

Операционная система семейства: *linux*, наименование: *ubuntu*, версия *18.10 cosmic*
интерпретатор команд: *bash* версия *4.4.19*.

Система программирования -- версия --, редактор текстов *emacs* версия *25.2.2*

6. **Идея, метод, алгоритм**

Создавать программы с разными видами ошибок и исправлять их.

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

1. Написать код с ошибкой.
2. Исправить ошибку.

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

Ошибки СЕ:

1. **Отсутствие точки с запятой(нарушение синтаксиса)**

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

```
int main() {  
int a;  
a = 10  
printf("%d\n", a); return 0;  
}
```

Исправление (добавить точку с запятой)

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

```
int main() {  
int a;  
a = 10;  
printf("%d\n", a);
```

```
return 0;
}
```

2. Неинициализированная переменная

```
#include<stdio.h>

int main() {
    int a;
    a = 10;
    printf("%d\n", a + k);
    return 0;
}
```

Исправление(инициализировать переменную)

```
#include<stdio.h>
int main() {
    int a, k = 5;
    a = 10;
    printf("%d\n", a + k); return 0;
}
```

Ошибки RE

1. Бесконечный цикл

```
#include<stdio.h>

int main() {
    int a[10] = {0};
    int i = 11;
    for (i; i > 10; i++)
        printf("%d ", a[i]);
    return 0;
}
```

Исправление(правильно ввести границы или действие над переменной-счетчиком)

```
#include<stdio.h>

int main() {
    int a[10] = {0};
    int i = 11;
    for (i; i > 10; i--){
        printf("%d ", a[i]);
    }
    return 0;
}
```

2. Деление на 0

```
#include<stdio.h>
int main() {
    int a[10] = {0};
    int i = 3;
    printf("%d ", i/a[i]);
    return 0;
}
```

Исправление(добавить условие для 0)

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

```
int main() {
```

```

int a[10] = {0};
int i = 3;
if (a[i] != 0) {
    printf("%d ", i/a[i]);
} else printf("Error");
return 0;
}

```

Ошибка UB

1. Вызов функции, которая ничего не возвращает

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

```

int sum() {
    int a = 3, b = 2;
    a = a + b;
}

```

```

int main() {
    return sum();
}

```

Исправление(функция должна вернуть значение)

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

```

int sum() {
    int a = 3, b = 2;
    a = a + b;
    return a;
}

```

```

int main() {
    return sum();
}

```

2. Переполнение типа

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

```

int main() {
    unsigned int a;
    a = -1;
    printf("%d", a);
    return 0;
}

```

Исправление(следить за значением переменной данного типа)

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

```

int main() {
    unsigned int a;
    a = 1;
    printf("%d", a);
    return 0;
}

```

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

| № | Лаб. или дом. | Дата | Вре мя | Событие | Действие по исправлению | Примечание |
|---|---------------------|------|-----------|---------|----------------------------|------------|
| | | | | | | |

10. Замечания автора

Отсутствуют

11. Выводы

Интересная лаба. Как-то необычно создавать ошибки самому, одно дело, когда программа выдает ошибку, ты пытаешься с ней разобраться, а тут, нужно самому её придумать. Пришлось потратить время на выяснение как выглядят ошибки вида UB и RE, но это явно пошло на пользу.

Подпись студента
