

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

Студент группы М80-109Б-22 Филиппов Александр Михайлович, № 21

Контакты email: a.filippov04@yandex.ru
Telegram: @otaku0101

Работа выполнена: «7» января 2023г.

Преподаватель: каф. 806 Сысоев Максим Алексеевич

Отчет сдан « » _____ 20__ г., итоговая оценка _____

Подпись преподавателя _____

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

2. **Цель работы:** Научиться пользоваться отладчиком

3. **Задание:** Внесение и исправление ошибок различных типов

4. **Оборудование** (студента):

Процессор AMD Ryzen 7 4800HS with Radeon Graphics с ОП 8 Гб, SSD 512 гб. Монитор

1920x1080 5. **Программное обеспечение** (студента):

Операционная система семейства: *linux*, наименование: *ubuntu*, версия 22.04

jammy интерпретатор команд: *bash* версия 5.1.16(1)-release

Система программирования CLion версия 2021.1.3, редактор текстов *nano* версия

6.2 Утилиты операционной системы WinRar, Microsoft Word

Прикладные системы и программы Ubuntu wsl, CLion, Google Chrome

Местонахождение и имена файлов программ и данных на домашнем компьютере ~/home

6. **Идея, метод, алгоритм**

Генерируем различные виды ошибок, их описание есть в пункте 8

7. **Сценарий выполнения работы**

Входные данные	Выходные данные	Описание тестируемого случая
ничего	Разные ошибки	Получаем различные типы ошибок

8. **Распечатка протокола**

ОШИБКА RE #1 – использование необъявленной переменной

```
int main() {  
    int i;  
    if (i < 10){  
        printf("%d", i);  
    }  
    return 0;  
}
```

РЕШЕНИЕ – инициализировать i

```
int main() {
    int i = 0;
    if (i < 10){
        printf("%d", i);
    }
    return 0;
}
```

ОШИБКА RE #2 – деление на 0

```
int main() {
    float i = 8;
    while (i >= 0) {
        printf("%f", 1.0 / i);
        i -= 2;
    }
    return 0;
}
```

РЕШЕНИЕ - заменить while (i >= 0) на while (i > 0)

```
int main() {
    float i = 8;
    while (i > 0) {
        printf("%f", 1.0 / i);
        i -= 2;
    }
    return 0;
}
```

ОШИБКА SE #1 – деление массива на число

```
int main() {
    int msv[10] = {};
    int dnm = 10;
    printf("%d", msv/dnm);
    return 0;
}
```

РЕШЕНИЕ – избегать подоюные ошибки

```
int main() {
    int msv[10] = {};
    int dnm = 10;
    printf("%d", msv);
    return 0;
}
```

ОШИБКА SE #2 – обращение к несуществующему полю

```
int main() {
    int foo = 10;
    int boo = foo.oof;

    printf("%d", boo);
}
```

РЕШЕНИЕ – создать объект с таким полем

```
int main() {
    struct boo {
        int gg;
    } var;
    var.gg = 13;
    int foo = var.gg;
    printf("%d", foo);
    return 0;
}
```

ОШИБКА UB #1 – вызов функции, которая ничего не возвращает, хотя должна

```
int foo() {  
}  
int main() {  
    return foo();  
}
```

РЕШЕНИЕ – сделать так, чтобы функция что-нибудь возвращала

```
int foo() {  
    return 5;  
}  
int main() {  
    return foo();  
}
```

ОШИБКА UB #2 – обращение к несуществующему элементу

```
int main() {  
    int a[666];  
    printf("%d", a[-77]);  
    return 0;  
}
```

РЕШЕНИЕ – поменять -77 на 77;

```
int main() {  
    int a[666];  
    printf("%d", a[77]);  
    return 0;  
}
```

9. Дневник отладки

№	Лаб. или дом.	Дата	Вре мя	Событие	Действие по исправлению	Примечание
1	дом	07.01.2023		Не знал как получить UB	Загуглил	Хоть и загуглил, понятнее не стало

10. Замечания автора

Неактуально, потому что информативность ошибок в Си нулевая. Просто код ошибки мне ничего не говорит. Фу какой язык.

11. Выводы

Работа вообще не понравилась, потому что самому придумывать ошибки – боль. Лучше наткаться на эти же ошибки в процессе работы, чтобы получить, так сказать, боевой опыт. В интернете нет вообще никакой информации о том, как вызывать эти ошибки, а сам Си разрешает делать почти что угодно, не бросая ошибок. Поэтому было неприятно заниматься выдумыванием проблем.

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

Подпись студента _____