

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

Студент группы М80-109Б-22 Гиголаев Антон
Александрович, № по списку 3

Контакты e-mail - giga.a.n.t@mail.ru, telegram -
@ntny_gglv

Работа выполнена: «06» ноября 2022г.

Преподаватель: каф. 806 Сысоев Максим
Алексеевич

Отчет сдан «05» __12__ 20__22__ г., итоговая
оценка _____

Подпись преподавателя _____

1. Тема: Отладчик системы программирования ОС UNIX.

2. Цель работы: Изучение отладчика СП ОС UNIX.

3. Задание: Изучить и освоить возможности лабораторного отладчика СП по содержащейся в практикуме документации и другим источникам, а также составить несколько программ, на примере которых будут демонстрироваться навыки нахождения и устранения ошибок.

4. Оборудование (студента):

Процессор Intel Core i5-11400H @ 6x 2.70GHz , НМД 512 Гб. Монитор 1920x1080

5. Программное обеспечение (студента):

Операционная система семейства: windows, наименование: windows 10 professional,
версия 10.0.19044 Интерпретатор команд:

Система программирования turing machine emulator версия 2.3, редактор текстов wordpad
версия 21H2

Утилиты операционной системы --

Прикладные системы и программы: --

Местонахождение и имена файлов программ и данных на домашнем компьютере --

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

Составление 6 программ со следующими типами ошибок:

- a. CE
- b. RE
- c. UB

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

1. Запустив редактор emacs, составить 6 программ с разными типами ошибок.

2. Составить отчёт по результатам работы.

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

ОШИБКА CE: переменная не была объявлена.

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

```
int main() {
    int a = 1, b = 2;
    c = a + b;
    printf("%d", c);
}
```

CE_Program_1.c: In function 'main':

CE_Program_1.c:6:3: error: 'c' undeclared (first use in this function)

```
6 | c = a + b;
  | ^
```

CE_Program_1.c:6:3: note: each undeclared identifier is reported only once for each function it appears in

РЕШЕНИЕ: объявить переменную и только потом присвоить ей значение суммы двух других.

```
int main() {
    int a = 1, b = 2;
    c = a + b;
    printf("%d", c);
}
```

ОШИБКА CE: синтаксическая ошибка в написании оператора сравнения if.

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

```
int main() {
    int a = 12;
```

```

if( > 10)
    printf("%s", "stupid code");
}

```

CE_Program_2.c: In function 'main':

CE_Program_2.c:5:6: error: expected expression before '>' token

```

5 |  if(>10)
  |

```

РЕШЕНИЕ: корректно записать оператор сравнения. В нашем случае вставить перед знаком "больше" имя переменной a.

```

int main() {
    int a = 12;
    if(a > 10)
        printf("%s", "stupid code");
}

```

ОШИБКА RE: выход за пределы массива в цикле.

```

#include <stdio.h>

```

```

int main() {
    int a[10];
    for(int i = 0; i <= sizeof(a) / sizeof(a[0]); ++i) {
        a[i] = 10;
    }
}

```

*** stack smashing detected ***: terminated

Аварийный останов (стек памяти сброшен на диск)

РЕШЕНИЕ: заменить знак "<=" на "<" для того, чтобы индексатор не был больше индекса последнего элемента массива.

```

int main() {
    int a[10];
    for(int i = 0; i < sizeof(a) / sizeof(a[0]); ++i) {
        a[i] = 10;
    }
}

```

ОШИБКА RE: деление на ноль.

```

#include <stdio.h>

```

```
int main() {
    double a = 0.000;
    printf("%d", 10/0);
}
```

Исключение в операции с плавающей точкой (стек памяти сброшен на диск)

РЕШЕНИЕ: заменить целые числа на дробные. Также заменить спецификатор формата вывода на тот, который совместим с числами с плавающей точкой.

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

```
int main() {
    double a = 0.000;
    printf("%f", 10.0/a);
}
```

ПРЕДУПРЕЖДЕНИЕ UB: попытка использовать функцию без подключения библиотеки, в которой она содержится.

```
int main() {
    printf("%d", 10);
}
```

UB_Program_1.c: In function ‘main’:

UB_Program_1.c:2:3: warning: implicit declaration of function ‘printf’ [-Wimplicit-function-declaration]

```
2 | printf("%d", 10);
  | ^~~~~~
```

UB_Program_1.c:1:1: note: include ‘<stdio.h>’ or provide a declaration of ‘printf’

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

```
1 | int main() {
```

UB_Program_1.c:2:3: warning: incompatible implicit declaration of built-in function ‘printf’ [-Wbuiltin-declaration-mismatch]

```
2 | printf("%d", 10);
  | ^~~~~~
```

UB_Program_1.c:2:3: note: include ‘<stdio.h>’ or provide a declaration of ‘printf’

РЕШЕНИЕ: подключить библиотеку, содержащую нужную нам функцию.

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

```
int main() {
```

```
printf("%d", 10);
}
```

ПРЕДУПРЕЖДЕНИЕ UB: ошибка при выборе спецификатора формата вывода.

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

```
int main() {
    printf("%d", 10 + 0.5);
}
```

UB_Program_2.c: In function ‘main’:

UB_Program_2.c:4:12: warning: format ‘%d’ expects argument of type ‘int’, but argument 2 has type ‘double’ [-Wformat=]

```
4 | printf("%d", 10 + 0.5);
  |           ~^  ~~~~~
  |           |   |
  |          int double
  |          %f
```

РЕШЕНИЕ: замена спецификатора вывода с целочисленного на тот, который поддерживает вывод чисел с плавающей точкой.

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

```
int main() {
    printf("%f", 10 + 0.5);
}
```

9. Дневник отладки

№ Лаб. Дата Время Событие Действие по исправлению Примечание

или

дом.

0 дом 01.09.2020 04:20 У меня сломался linux Переустановил систему
Мне грустно

№	Лаб. или дом.	Дата	Время	Событие	Действие по исправлению	Примечание
0	дом	27.11.2022	16:00	Устал...	Подзарядился!	Отдых - самое главное в жизни любого

						человека!
--	--	--	--	--	--	-----------

10. Выводы

Полезная ЛР. Теперь могу различать типы ошибок. Самым интересным оказалось то, что UB предупреждение можно получить, едва коснувшись клавиатуры; а вот чтобы получить SE, надо голову изломать. Ох, а как же приятен синтаксис Си-подобных языков...

БУДОРАЖИТ воображение...

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

Подпись студента _____