

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

Студент группы М80-109Б-22 Степанов Алексей Николаевич, № по списку 18

Контакты aleksey.stepanov2004@mail.ru, telegram @Alex1stepa

Работа выполнена: «31» октября 2022г.

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 исправлять ошибки в коде программ, написанных на языке Си

3. Задание (вариант): продемонстрировать умения в пользовании отладчиком языка Си для устранения ошибок UB, а также умение находить в коде и устранять очевидные(и не очень) ошибки типов SE, RE и UB.

4. Оборудование (студента):

Процессор *Intel Core i5-8265U @ 8x 3.9GH* с ОП 7851 Мб, НМД 1024 Гб. Монитор 1920x1080

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*

Утилиты операционной системы --

Прикладные системы и программы: VTM(QT)

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):

1. Используя свои познания и умения в программировании, составим код 6 программ на языке Си, в каждой из которых будут ошибки SE, RE, UB.
2. Увидим, что все падает (PANIC)
3. Проанализируем код программы (ручками(SE), при помощи дебагера(UB) или с помощью отладочных выводов(RE,UB)).
4. Все заработает(Calm).

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

SE:

1)Использование неописанной в коде программы функции (например, забыл объявить библиотеку в коде).

2) Как истинный олимппрогер и человек, что часто кодит на C++, я вспоминаю, что в нем есть тип данных `int64_t` и использую его вместо `long long`.

RE:

1)При декларации массива мы выставили несоизмеримо большой размер и залезли не в открытую часть памяти => segmentation error.

2)Рекурсивная функция спровоцировала деление на 0.

UB:

1) Выход за пределы массива.

2) Выведение адреса переменной в ОП, вместо ее значения.

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

Протокол(1 SE):

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

```
// setlocale библиотечная функция библиотеки locale - вставим через директиву предпроцессора.
```

```
int main(){
```

```
    char str[10];
```

```
    char s;
```

```
    setlocale(LC_ALL,"RU");
```

```
//добавляет параметры локализации, связанные с отображением кириллических литер
for( int t=0; t<10;t++){

scanf("%c", &s);
str[t]=s;
}
printf("Вы ввели строку:");
for( int t=0;t<10;++t){

printf("%c",str[t]);

}
return 0;
}
```

Вердикт:CE(compilation error)

```
alex1st@alex1st-VirtualBox: ~
drwxr-xr-x 2 alex1st alex1st 4096 ноя 12 20:16 Music
drwxr-xr-x 2 alex1st alex1st 4096 ноя 12 20:16 Pictures
drwxr-xr-x 2 alex1st alex1st 4096 ноя 12 20:16 Public
drwx----- 4 alex1st alex1st 4096 ноя 12 20:22 snap
-rw-rw-r-- 1 alex1st alex1st 377 ноя 15 23:39 tad10.c
-rw-rw-r-- 1 alex1st alex1st 377 ноя 15 23:38 tas10.c
-rw-rw-r-- 1 alex1st alex1st 68 ноя 12 20:21 task1.c
-rw-rw-r-- 1 alex1st alex1st 1783 ноя 12 23:32 task9.c
drwxr-xr-x 2 alex1st alex1st 4096 ноя 12 20:16 Templates
drwxr-xr-x 2 alex1st alex1st 4096 ноя 12 20:16 Videos
alex1st@alex1st-VirtualBox:~$ nano tas10.c
alex1st@alex1st-VirtualBox:~$ nano tas10.c
alex1st@alex1st-VirtualBox:~$ gcc tas10.c
tas10.c: In function 'main':
tas10.c:8:3: warning: implicit declaration of function 'setlocale' [-Wimplicit-
function-declaration]
    8 |     setlocale(LC_ALL,"RU");
      |     ^~~~~~
tas10.c:8:13: error: 'LC_ALL' undeclared (first use in this function)
    8 |     setlocale(LC_ALL,"RU");
      |             ^~~~~~
tas10.c:8:13: note: each undeclared identifier is reported only once for each f
unction it appears in
tas10.c:12:4: warning: implicit declaration of function 'scanf'; did you mean '
scanf'? [-Wimplicit-function-declaration]
   12 |     scanf("%C",s);
      |     ^~~~~
      |     scanf
alex1st@alex1st-VirtualBox:~$
```

Решение проблемы: Подключить библиотеку <locale.h>, содержащую описание функции и описание макроса LC_ALL.

Исправленный код с протоколом запуска:

```
alex1st@alex1st-VirtualBox:~$ cat tas10.c
```

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

```
#include <locale.h>
```

```
// setlocale библиотечная функция библиотеки locale - вставим через директиву препроцессора.
```

```
int main(){
```

```
char str[10];
```

```
char s;
```

```
setlocale(LC_ALL,"RU");
```

```
//добавляет параметры локализации, связанные с отображением кириллических литер
```

```
for( int t=0; t<10;t++){

    scanf("%c", &s);
    str[t]=s;
}
printf("Вы ввели строку:");
for( int t=0;t<10;++t){

    printf("%c",str[t]);

}
return 0;
}
alex1st@alex1st-VirtualBox:~$ gcc tas10.c
alex1st@alex1st-VirtualBox:~$ ./a.out
aswergood0
Вы ввели строку:aswergood0alex1st@alex1st-VirtualBox:~$
2) #include <stdio.h>
```

```
int main(){

    char str[10];
    char s;
    for( int64_t t=0; t<10;t++){

        scanf("%C",&s);
        str[t]=s;
    }
    for( int64_t t=0;t<10;++t){

        printf("%c",str[10]);

    }
    return 0;
}
```

```
Activities Terminal ноя 16 00:08 en
alex1st@alex1st-VirtualBox: ~
}
return 0;
}
alex1st@alex1st-VirtualBox:~$ gcc tas10.c
alex1st@alex1st-VirtualBox:~$ gcc tad10.c
tad10.c: In function 'main':
tad10.c:8:8: error: unknown type name 'int64_t'
      8 |     for( int64_t t=0; t<10;t++){
        |         ^~~~~~
tad10.c:2:1: note: 'int64_t' is defined in header '<stdint.h>'; did you forget
to '#include <stdint.h>'?
      1 | #include <stdio.h>
    +++ |+#include <stdint.h>
      2 |
tad10.c:10:12: warning: format '%C' expects argument of type 'wchar_t *', but a
rgument 2 has type 'char *' [-Wformat=]
     10 |     scanf("%C",&s);
        |           ^~
        |           |
        |         char *
        |         int *
        |         %hhd
tad10.c:13:7: error: unknown type name 'int64_t'
     13 |     for( int64_t t=0;t<10;++t){
        |         ^~~~~~
tad10.c:13:7: note: 'int64_t' is defined in header '<stdint.h>'; did you forget
to '#include <stdint.h>'?
alex1st@alex1st-VirtualBox:~$
```

Решение: заменить int64_t на long longю

```
alex1st@alex1st-VirtualBox:~$ nano tad10.c
```

```
alex1st@alex1st-VirtualBox:~$ gcc tad10.c
```

```
alex1st@alex1st-VirtualBox:~$ ./a.out
```

```
1234567890
```

```
1234567890alex1st@alex1st-VirtualBox:~$
```

Правильный код:

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

```
int main(){
```

```
    char str[10];
```

```
    char s;
```

```
    for( long long t=0; t<10;t++){
```

```
        scanf("%c",&s);
```

```
        str[t]=s;
```

```
    }
```

```
    for( long long t=0;t<10;++t){
```

```
        printf("%c",str[t]);
```

```
    }
```

```
    return 0;
```

```
}
```

```
3)
```

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

```

int main()
{
    int f[1000000000000000000000];
    printf("%f\n",f[1]);
    return 0;
}

```

```

alex1st@alex1st-VirtualBox: ~
int main()
{
    int f[1000000000000000000000];
    printf("%f\n",f[1]);//логорифм во флоате
    return 0;
}
alex1st@alex1st-VirtualBox:~$ gcc tad10.c
tad10.c: In function 'main':
tad10.c:5:11: warning: integer constant is so large that it is unsigned
5 | {   int f[1000000000000000000000];
    |           ^~~~~~
tad10.c:5:9: error: size of array 'f' is too large
5 | {   int f[1000000000000000000000];
    |           ^
alex1st@alex1st-VirtualBox:~$ nano tad10.c
alex1st@alex1st-VirtualBox:~$ gcc tad10.c
tad10.c: In function 'main':
tad10.c:6:14: warning: format '%f' expects argument of type 'double', but argum
ent 2 has type 'int' [-Wformat=]
6 |     printf("%f\n",f[1]);//логорифм во флоате
    |              ^~
    |              |
    |             double
    |             int
    |             %d
alex1st@alex1st-VirtualBox:~$ nano tad10.c
alex1st@alex1st-VirtualBox:~$ gcc tad10.c
alex1st@alex1st-VirtualBox:~$ ./a.out
Segmentation fault (core dumped)
alex1st@alex1st-VirtualBox:~$

```

Решение: сократить аппетиты и декларировать меньшее количество элементов массива.

```
alex1st@alex1st-VirtualBox:~$ nano tad10.c
```

```
alex1st@alex1st-VirtualBox:~$ gcc tad10.c
```

```
alex1st@alex1st-VirtualBox:~$ ./a.out
```

```
0
```

```
alex1st@alex1st-VirtualBox:~$
```

Правильный код:

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

```

int main()
{
    int f[100];
    printf("%f\n",f[1]);
    return 0;
}

```

4)

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

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

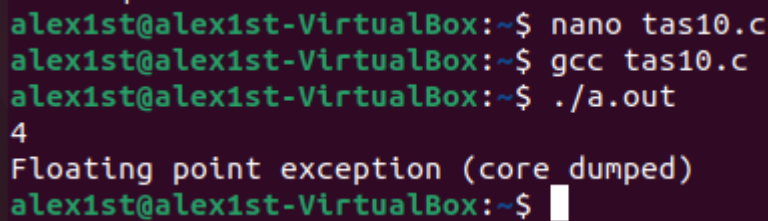
```
long F(long g, long f){ //функция рекурсивная, вызывает ошибку при попытке найти остаток от деления на 0
```

```
long long t=g%4;
```

```
if(g==0) return f;
```

```
    f=f*t;  
    return F(g/t,f);  
}
```

```
int main()  
{  
    long h=0;  
    scanf("%lld",&h);  
    long g=F(h,1);  
    printf("%lld",g);  
    return 0;  
}
```



```
alex1st@alex1st-VirtualBox:~$ nano tas10.c  
alex1st@alex1st-VirtualBox:~$ gcc tas10.c  
alex1st@alex1st-VirtualBox:~$ ./a.out  
4  
Floating point exception (core dumped)  
alex1st@alex1st-VirtualBox:~$
```

Решение: добавить проверку на 0 и единицу(иначе переполним стек)

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

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

```
long F(long g, long f){ //функция рекурсивная, вызывает ошибку при попытке найти остаток от деления на 0  
    long long t=g%4;
```

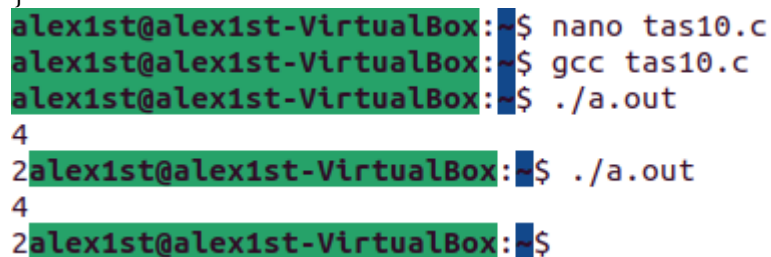
```
if(g==0) return f;
```

```
if(t>1)
```

```
{  
    f=f*t;  
    return F(g/t,f);  
}
```

```
else return F(g/2,f); } //условие перехода при 0 и 1
```

```
int main()  
{  
    long h=0;  
    scanf("%lld",&h);  
    long g=F(h,1);  
    printf("%lld",g);  
    return 0;  
}
```



```
alex1st@alex1st-VirtualBox:~$ nano tas10.c  
alex1st@alex1st-VirtualBox:~$ gcc tas10.c  
alex1st@alex1st-VirtualBox:~$ ./a.out  
4  
alex1st@alex1st-VirtualBox:~$ ./a.out  
4  
alex1st@alex1st-VirtualBox:~$
```

5)

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

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

```
int main()
```

```

{
    long g[2]={1,2};
    long h=0;
    scanf("%ld",&h);
    for(int r=0;r<h;r++)
        printf("%ld",g[r]);
    return 0;
}

```

```

alex1st@alex1st-VirtualBox:~$ nano tas10.c
alex1st@alex1st-VirtualBox:~$ gcc tas10.c
alex1st@alex1st-VirtualBox:~$ ./a.out
5
1205886680097251491841alex1st@alex1st-VirtualBox:~$ ./a.out
5
120-34285960661339115521alex1st@alex1st-VirtualBox:~$ █

```

Решение: поставить проверку на выход за пределы массива

```

alex1st@alex1st-VirtualBox:~$ gcc tas10.c
alex1st@alex1st-VirtualBox:~$ ./a.out
5
12alex1st@alex1st-VirtualBox:~$ ./a.out
5
12alex1st@alex1st-VirtualBox:~$ ./a.out
1
1alex1st@alex1st-VirtualBox:~$ ./a.out
0
alex1st@alex1st-VirtualBox:~$

```

```

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

```

```

int main()
{
    long g[2]={1,2};
    long h=0;
    scanf("%ld",&h);
    for(int r=0;r<h&&h<2;r++)
        printf("%ld",g[r]);
    return 0;
}

```

```

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

```

```

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

```

```
Activities Terminal Nov 20 17:04 en
alex1st@alex1st-VirtualBox: ~
alex1st@alex1st-VirtualBox:~$ ./a.out
-1312437448alex1st@alex1st-VirtualBox:~$ cat ta
tas10.c      tas10.c.save  task1.c
tas10.c      tas10.c.save.1 task9.c
alex1st@alex1st-VirtualBox:~$ cat ta
tas10.c      tas10.c.save  task1.c
tas10.c      tas10.c.save.1 task9.c
alex1st@alex1st-VirtualBox:~$ cat tas10.c
#include <stdio.h>

int main()
{
    int a ;
    printf("%d",&a+1);
}
alex1st@alex1st-VirtualBox:~$ gcc tas10.c
tas10.c: In function 'main':
tas10.c:6:14: warning: format '%d' expects argument of type 'int', but argument
  2 has type 'int *' [-Wformat=]
    6 |     printf("%d",&a+1);
      |             ~^~~~~
      |             |   |
      |             int int *
      |             %ls
alex1st@alex1st-VirtualBox:~$ ./a.out
14397272alex1st@alex1st-VirtualBox:~$ ./a.out
376799192alex1st@alex1st-VirtualBox:~$ ./a.out
204312456alex1st@alex1st-VirtualBox:~$ ./a.out
178656872alex1st@alex1st-VirtualBox:~$
```

Решение: убрать амперсant, тем самым возвращая не адрес, а значение переменной.

```
alex1st@alex1st-VirtualBox:~$ gcc tas10.c
alex1st@alex1st-VirtualBox:~$ ./a.out
1alex1st@alex1st-VirtualBox:~$
```

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

```
int main()
{
    int a=0 ;
    printf("%d",a+1);
    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 | Врем я | Событие | Действие по исправлению | Примечание |
|---|---------------------|------|-----------|---------|----------------------------|------------|
| | | | | | | |
| | | | | | | |

10. Замечания автора по существу работы

Работа конструктивная, негативный замечаний нет.

11. Выводы

От лабораторной работы получил исключительно положительные эмоции и впечатления. По моему мнению, знания, приобретенные мною на данной лабораторной работе, помогли мне лучше осознать принципы работы операционных систем, процессора, отладчика СИ, более пристально изучить язык программирования Си, научиться отлаживать свой код и находить ошибки разного типа, что несомненно поможет мне при решении практических задач.

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

Подпись студента _____