

# Задание №3 в рамках вычислительного практикума.

## Отладка

Кострицкий А. С., Ломовской И. В.

Москва — 2022 — TS2203202101

### Цель работы

*Целью* работы является умение студента самостоятельно производить трассировку приложения.

### Задание №1

Найдите с помощью gdb ошибки в приложенных программах. Опишите как Вы это сделали. По результатам этого задания будьте готовы ответить на следующие вопросы.

1. С какими ключами нужно скомпилировать программу, чтобы можно было пользоваться отладчиком gdb? Что произойдёт, если собрать программу без этого ключа и загрузить её в gdb?
2. Как запустить программу под отладчиком? Как досрочно завершить её работу (предположим, что Вы остановили выполнение программы на точке останова и дальше выполнять программу не нужно)?
3. Выполнение Вашей программы было остановлено на какой-то из точек останова. Как посмотреть в каком месте Вашей программы Вы остановились?
4. С помощью какой команды можно посмотреть значение переменной? Изменить значение переменной?
5. С помощью каких команд можно выполнить программу в пошаговом режиме? Чем отличаются эти команды друг от друга?
6. Выполнение Вашей программы было остановлено на какой-то из точек останова. Как понять, какая последовательность вызовов функций привела Вас сюда?
7. Как можно установить точку останова в программе?
8. Какая точка останова считается временной?
9. Как временно выключить/включить точку останова или пропустить некоторое количество срабатываний?
10. Как задать условие остановки на точке основа?
11. Чем отличаются точки наблюдения от точек останова?

12. Приведите пример ситуации, в которой удобно использовать точку наблюдения.
13. С помощью какой команды можно посмотреть содержимое области памяти?

## Задание №2

Составьте таблицу размеров типов `char`, `int`, `unsigned`, `long long`, `short`, `int32_t`, `int64_t` на двух разных машинах. Рекомендуется взять машины с разными ОС и/или разной разрядностью, например, машины дома и в лабораторном зале.

## Задание №3

Покажите, как в памяти представлены переменные типов `char`, `int`, `unsigned`, `long long`. Рассмотрите как положительные значения этих переменных, так и отрицательные. Результаты поясните.

## Задание №4

Покажите, как в памяти представлен массив целых чисел. Продемонстрируйте особенности выполнения операции сложения указателя с целым числом на примере массива.

## Задание №5

Придумайте пример программы с ошибкой, для нахождения которой удобно использовать точку наблюдения.

## Задание №6

Изучите работу с отладчиком в Qt Creator. Напишите инструкцию, которая сопоставляет команды gdb и аналогичные действия в Qt Creator.

## Формат защиты

Основная защита задания проходит на седьмой неделе.

Студентом к дате основной защиты готовится страница отчёта в формате `.docx` (`.odt`), два файла `.docx` (`.odt`) и `.pdf` в виде `zip`-архива прикрепляются к кафедральному moodle.

## Код программы №1

```
#include <stdio.h>

long long unsigned factorial(unsigned n);

int main(void)
{
    unsigned n;
```

```

long long unsigned result;

printf("Input n: ");
if (scanf("%u", &n) != 1)
{
printf("Input error");
return 1;
}

result = factorial(n);

printf("factorial(%u) = %llu\n", n, result);

return 0;
}

long long unsigned factorial(unsigned n)
{
long long unsigned result = 1;

while (n--)
result *= n;

return result;
}

```

## Код программы №2

```

#include <stdio.h>

#define N 5

double get_average(const int a[], size_t n);

int get_max(const int *a, size_t n);

int main()
{
int arr[N];
size_t i;

printf("Enter %d numbers:\n", N);

for (i = 0; i < N; i++)
{
printf("Enter the next number: ");
if (scanf("%d", &arr[i]) != 1)
{
printf("Input error");

```

```

return 1;
}
}

for (i = 1; i < N; i++)
printf("Value [%zu] is %d\n", i, arr[i]);

printf("The average is %g\n", get_average(arr, N));

printf("The max is %d\n", get_max(arr, N));

return 0;
}

double get_average(const int a[], size_t n)
{
double temp = 0.0;

for (size_t i = 0; i < n; i++)
temp += a[i];
temp /= n;

return temp;
}

int get_max(const int *a, size_t n)
{
int max = a[0];

for (size_t i = 1; i < n; i++)
if (max < a[i])
max = a[i];

return max;
}

```

## Код программы №3

```

#include <stdio.h>

int div(int a, int b);

int main(void)
{
int a = 5, b = 2;

printf("%d div %d = %d\n", a, b, div(a, b));

a = 10;

```

```
b = 0;

printf("%d div %d = %d\n", a, b, div(a, b));

return 0;
}

int dib(int a, int b)
{
return a / b;
}
```