Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт Космических и информационных технологий
институт
Кафедра «Информатика»
кафедра

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ

Лабораторная работа No 3. Использование механизма виртуальной памяти в программах для ОС GNU/Linux

тема

Преподаватель _____ А.С. Кузнецов
подпись, дата инициалы, фамилия

Студент КИ18-17/1б 031831229 _____ В.А. Прекель
номер группы, зачетной книжки подпись, дата инициалы, фамилия

Красноярск 2019

СОДЕРЖАНИЕ

1 Цель работы с постановкой задачи

1.1 Цель работы

Изучение программных средств механизма виртуальной памяти в ОС.

1.2 Задача работы

Требуется разработать программу в виде Linux-приложения, позволяющую манипулировать многоэлементными абстрактными структурами данных. Обязательны к реализации функции по добавлению одной структуры, модификации структуры (значения всех полей или только части из них), удаления структуры, чтения одной структуры, отображения содержимого всех структур (или части). Должен использоваться интерфейс командной строки (CLI). Описание элемента структуры данных и два дополнительных запроса данных, обязательных к реализации, даются индивидуально и приводятся далее.

Вариант 17. Структура данных: архипелаг; количество островов; количество обитаемых островов. Создать два запроса, позволяющих определить, имеются ли архипелаги, состоящие только из необитаемых островов, и получить список архипелагов с указанием количества островов в них.

1.3 Описание и пояснение к работе

Используется система сборки CMake. Проект представляет из себя библиотеку, исполняемую программу и исполняемая программа с тестами (тесты в этой лабораторной работе не обязательны, поэтому они не задокументированный и плохо покрывают код). Дополнительно поддерживается компиляция и исполнение на Windows.

Команды для сборки и запуска программы:

```
mkdir build
cd build
cmake ..
make
./Lab_03_Console/Lab_03_Console
```

1.4 Структура проекта

- CMakeLists.txt
- Lab_03_Console
    - Actions.c
    - Actions.h
    - CMakeLists.txt
    - Input.c
    - Input.h
    - MainConsole.c
- Lab_03_Lib
    - Archipelago.c
    - Archipelago.h
    - ArchipelagoCollection.c
    - ArchipelagoCollection.h
    - ArchipelagoCollectionQuery.c
    - ArchipelagoCollectionQuery.h
    - CMakeLists.txt
    - LinkedList.c
    - LinkedList.h
    - LinkedListDeclarations.h
    - LinkedListNode.c
    - LinkedListNode.h
- Lab_03_LibTests
    - ArchipelagoCollectionQueryTests.c
    - ArchipelagoCollectionQueryTests.h

- ArchipelagoCollectionTests.c
- ArchipelagoCollectionTests.h
- CMakeLists.txt
- LinkedListTests.c
- LinkedListTests.h
- MainLibTests.c
- Suite.c
- Suite.h

2 Описание использованных при выполнении задания функций Linux API управления областями виртуальной памяти

```
void* malloc(size_t size);
```

malloc() распределяет size байтов и возвращает указатель на распределенную память. Память при этом не "очищается". При успешном вызове malloc() выделяет size байт памяти и возвращает указатель в начальную точку только что выделенной области. Содержимое памяти не определено, поэтому мы не должны рассчитывать, что память будет заполнена нулями. При ошибке malloc() возвращает NULL, а значение глобальной переменной errno устанавливается значение ENOMEM.

```
void* realloc(void *ptr, size_t size);
```

realloc() меняет размер блока памяти, на который указывает ptr, на размер, равный size байтов. Содержание будет неизменным в пределах наименьшего из старых и новых размеров, а новая распределенная память будет неинициализирована. Если ptr равно NULL, то данный вызов эквивалентен malloc(size); если размер равен нулю, то данный вызов эквивалентен free(ptr). Если только ptr не равен NULL, он, по-видимому, возвращен более ранним вызовом malloc(), calloc() или realloc().При успешном вызове realloc() размер области памяти, на которую направлен указатель ptr, изменяется. Новый размер

в байтах задается через size. Она возвращает указатель на заново распределенную область памяти, причем этот указатель может быть как равен ptr, так и иметь другое значение. В случае увеличения области памяти realloc()может и не увеличить исходный фрагмент до запрашиваемого размера на том же месте, где этот фрагмент в данный момент находится. В таком случае функция попытается распределить новую область памяти размером size байт, скопировать старую область в новую, а старую после этого освободить. При любой подобной операции содержимое области памяти сохраняется либо полностью, либо в размере, равном объему новой выделенной области. Поскольку операции realloc() связаны с копированием, при увеличении области памяти они могут быть затратными.

```
void free(void* ptr);
```

free() освобождает место в памяти, на которое указывает ptr, возвращенный, по-видимому, предшествующим вызовом функций malloc(), calloc() или realloc(). Иначе (или если уже вызывался free(ptr)) дальнейший ход событий непредсказуем. Если ptr равен NULL, то не выполняется никаких действий. Этот вызов освобождает память, на которую указывает ptr. Параметр ptr должен предварительно получен через malloc(), calloc() или realloc(). Это означает, что с помощью free() мы не сможем освобождать произвольные блоки памяти. Так, не получится освободить половину области памяти, передав указатель на ее середину. Такое действие приведет к появлению неопределенной памяти, что проявится в виде сбоя (ошибки сегментации).

3 Примеры использования (вызова) этих функций в представленном программном коде

3.1 Функция, считывания строки в динамическую строку, считывающая пошагово и расширяя строку с помощью realloc() (Lab_03_Console/ Input.c):

```
char* InputLineRealloc(int stepSize, bool isFinalReallocRequired)
{
    assert(stepSize >= 2);
    unsigned int currentSize = stepSize;
    char* string = (char*) malloc((currentSize + 1) * sizeof(char));
    assert(string);
    char* currentStep = string;
    int i = 1;
    while (true)
    {
        char* fgetsReturns = fgets(currentStep, stepSize, stdin);
        assert(fgetsReturns);

        unsigned int stringLength = strlen(currentStep);

        if (currentStep[stringLength - 1] == '\n')
        {
            currentStep[stringLength - 1] = '\0';
            if (isFinalReallocRequired)
            {
                currentSize = strlen(string) + 1;
                string = realloc(string, currentSize * sizeof(char));
                assert(string);
            }
            break;
        }
        currentSize += stepSize;
        string = realloc(string, currentSize * sizeof(char));
        assert(string);
        currentStep = string + currentSize - stepSize - i++;
    }
    return string;
}
```

3.2 Функция, создающая архипелаг, использующая malloc() (Lab_03_Lib/Archipelago.c):

```
Archipelago* ArchipelagoCreate(char* name,
                               int countIslands,
                               int countInhabitedIslands)
{
    Archipelago* pArchipelago = (Archipelago*) malloc(sizeof(Archipelago));
    assert(pArchipelago);

    pArchipelago->Name =
            (char*) malloc((strlen(name) + 1) * sizeof(char));
    assert(pArchipelago->Name);
```

```
    strcpy(pArchipelago->Name, name);

    pArchipelago->CountIslands = countIslands;
    pArchipelago->CountInhabitedIslands = countInhabitedIslands;

    return pArchipelago;
}
```

3.3 Функция, уничтожающая архипелаг, использующая free() (Lab_03_Lib/Archipelago.c):

```
void ArchipelagoDestroy(Archipelago* pArchipelago)
{
    free(pArchipelago->Name);
    free(pArchipelago);
}
```

4 Примеры работы программ в виде перехватов содержимого экрана

4.1 Снимки экрана запуска № 1 (Windows 10, MSVC 19.24.28117.0)

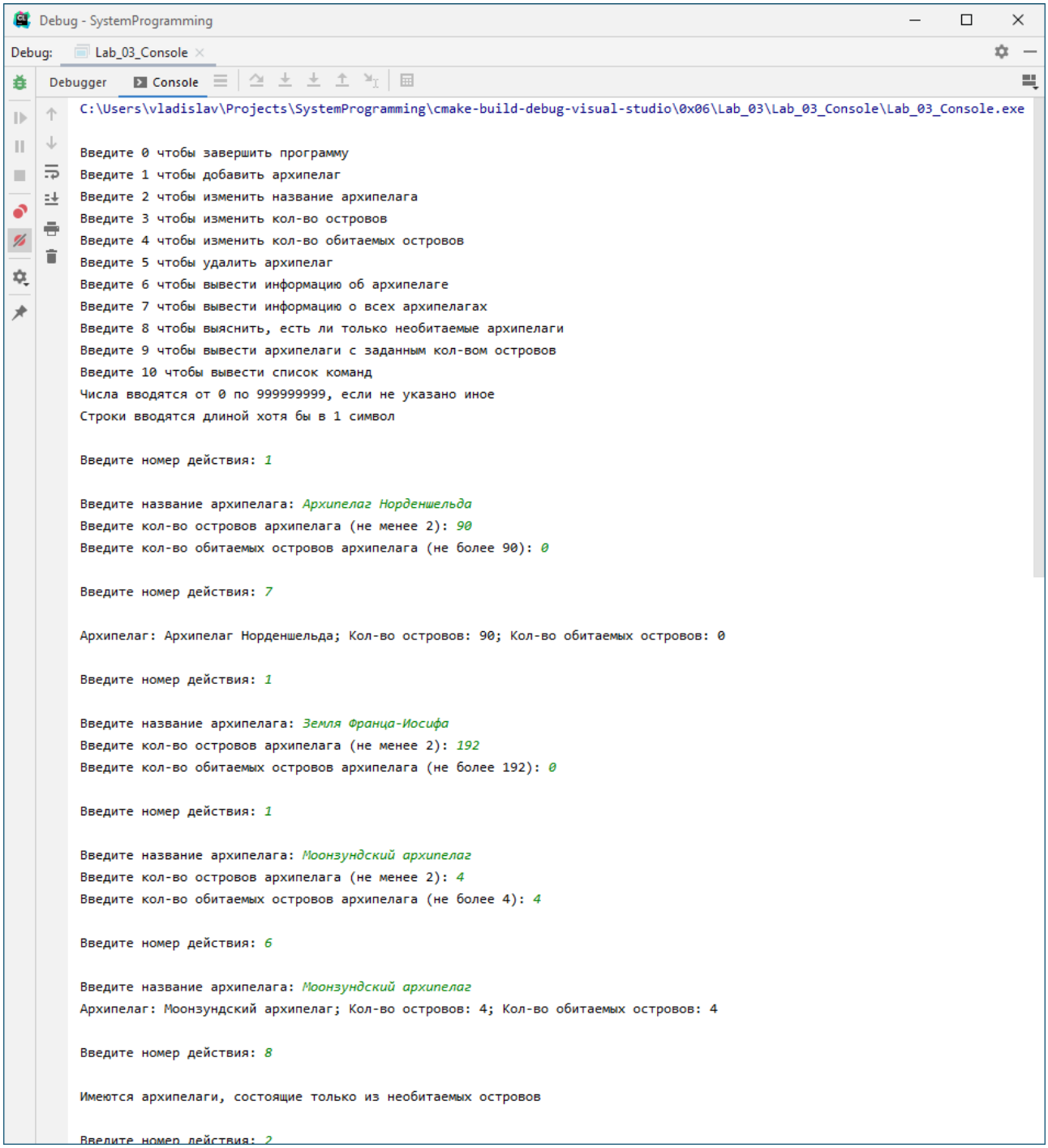

Рисунок 1 – Запуск № 1, стр. 1

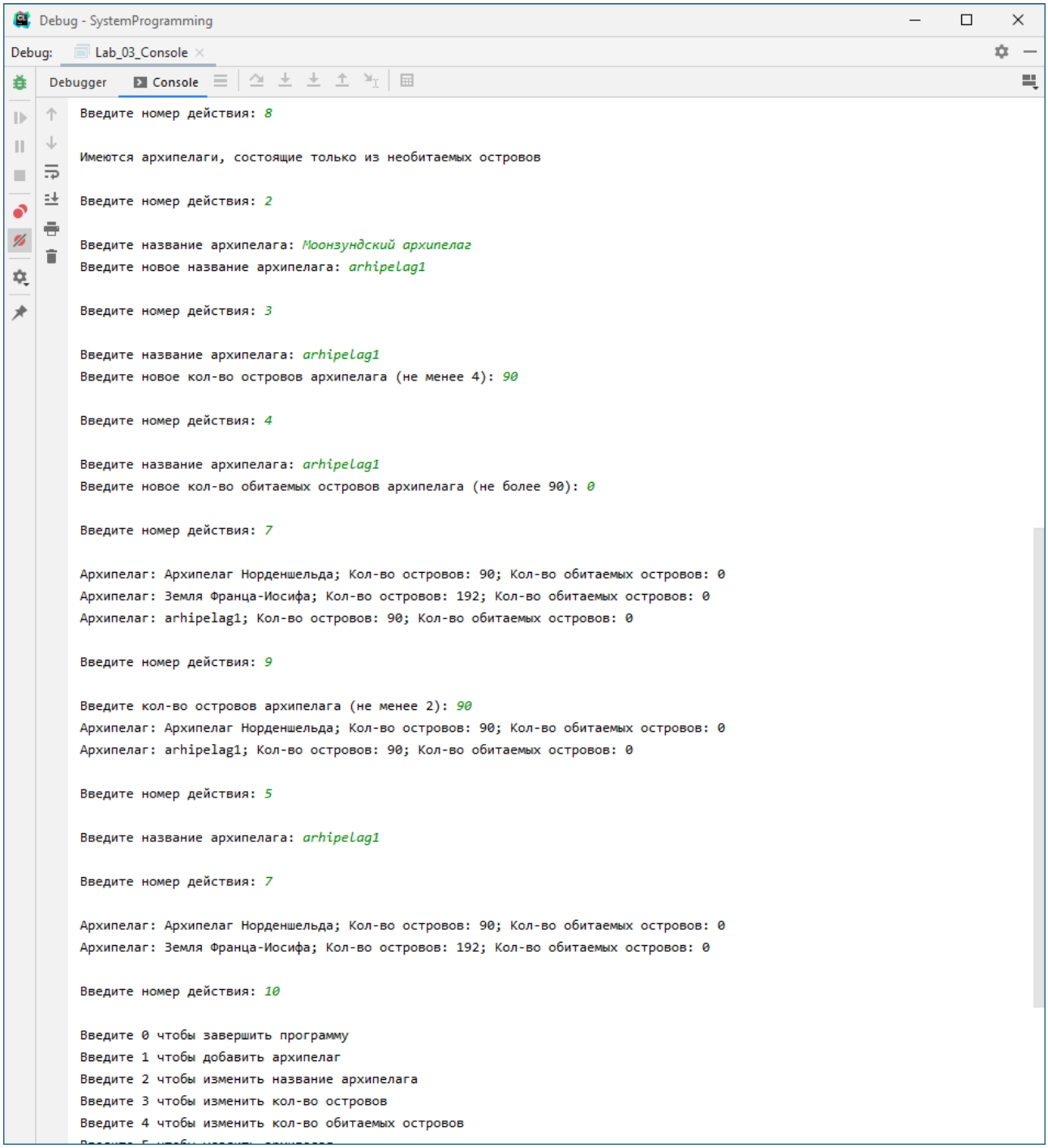

Рисунок 2 – Запуск № 1, стр. 2

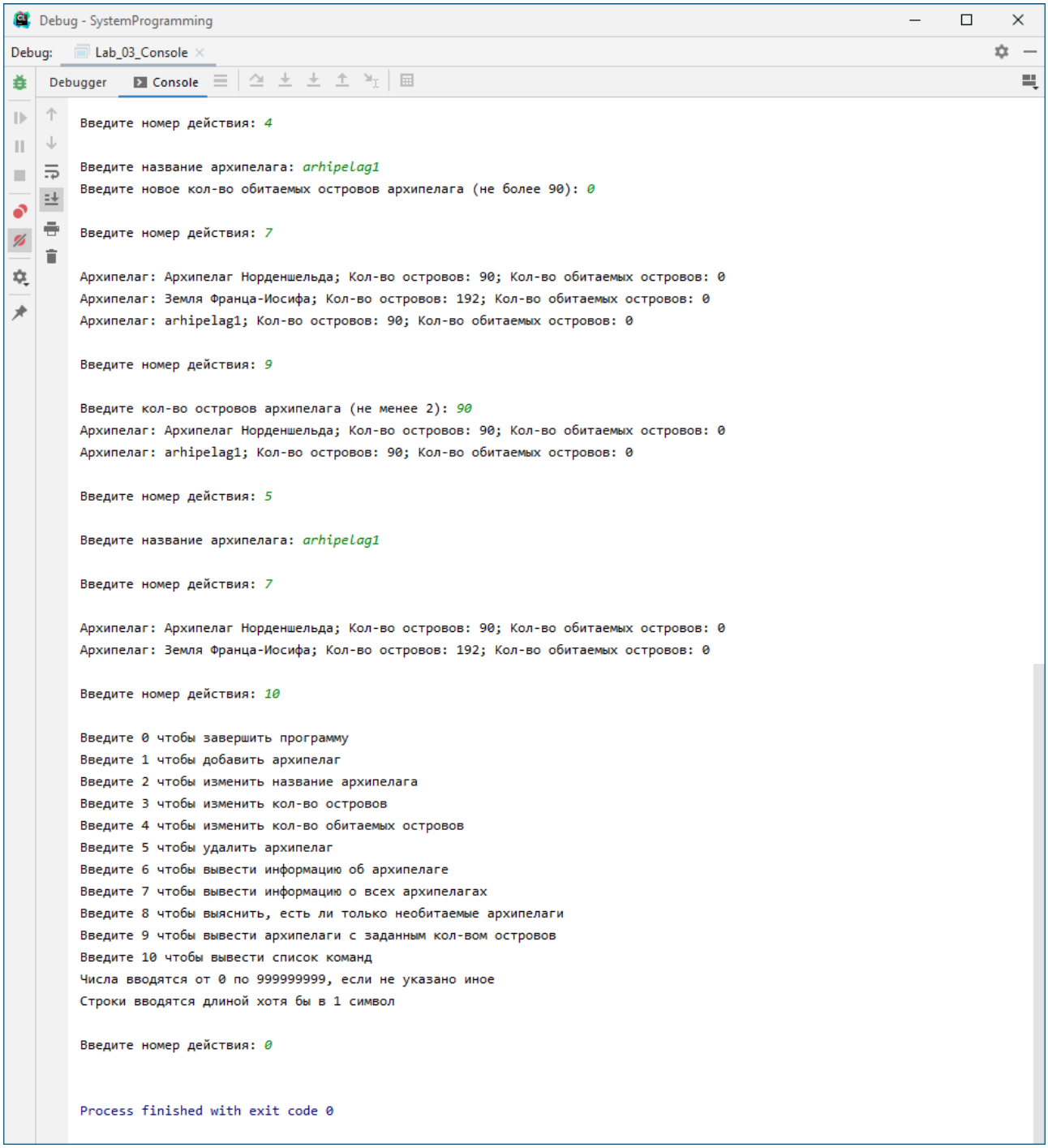

Рисунок 3 – Запуск № 1, стр. 3

4.2 Запуск № 2 (WSL Ubuntu 18.04, Clang 8.0.0)

```
C:\Users\vladislav\Projects\SystemProgramming\cmake-build-debug-wsl-
clang\0x06\Lab_03\Lab_03_Console\Lab_03_Console

Введите 0 чтобы завершить программу
Введите 1 чтобы добавить архипелаг
Введите 2 чтобы изменить название архипелага
```

```
Введите 3 чтобы изменить кол-во островов
Введите 4 чтобы изменить кол-во обитаемых островов
Введите 5 чтобы удалить архипелаг
Введите 6 чтобы вывести информацию об архипелаге
Введите 7 чтобы вывести информацию о всех архипелагах
Введите 8 чтобы выяснить, есть ли только необитаемые архипелаги
Введите 9 чтобы вывести архипелаги с заданным кол-вом островов
Введите 10 чтобы вывести список команд
Числа вводятся от 0 по 999999999, если не указано иное
Строки вводятся длиной хотя бы в 1 символ

Введите номер действия: 1

Введите название архипелага: qwe
Введите кол-во островов архипелага (не менее 2): 12313112321312
Введите кол-во островов архипелага (не менее 2): 3123213213
Введите кол-во островов архипелага (не менее 2): -132131
Введите кол-во островов архипелага (не менее 2): ewdfef
Введите кол-во островов архипелага (не менее 2): 2313fesdwf
Введите кол-во островов архипелага (не менее 2): 12
Введите кол-во обитаемых островов архипелага (не более 12): 3455
Введите кол-во обитаемых островов архипелага (не более 12): -567
Введите кол-во обитаемых островов архипелага (не более 12): sfgsdg
Введите кол-во обитаемых островов архипелага (не более 12): 432
Введите кол-во обитаемых островов архипелага (не более 12): 3

Введите номер действия: 7

Архипелаг: qwe; Кол-во островов: 12; Кол-во обитаемых островов: 3

Введите номер действия: 6

Введите название архипелага: qwe
Архипелаг: qwe; Кол-во островов: 12; Кол-во обитаемых островов: 3

Введите номер действия: 5

Введите название архипелага: qwe

Введите номер действия: 7

Архипелаги отсутствуют

Введите номер действия: 6

Введите название архипелага: qwe
Архипелаг с таким названием не существует

Введите номер действия: 5

Введите название архипелага: qwe
Архипелаг с таким названием не существует

Введите номер действия: 4

Введите название архипелага: qwe
Архипелаг с таким названием не существует

Введите номер действия: 3
```

```
Введите название архипелага: qwe
Архипелаг с таким названием не существует

Введите номер действия: 2

Введите название архипелага: qwe
Архипелаг с таким названием не существует

Введите номер действия: 8

Отсутствуют архипелаги, состоящие только из необитаемых островов

Введите номер действия: qwe
Введите номер действия: daw
Введите номер действия: 9

Введите кол-во островов архипелага (не менее 2): qwe
Введите кол-во островов архипелага (не менее 2): 10
Архипелаги отсутствуют

Введите номер действия: 1

Введите название архипелага: qwe
Введите кол-во островов архипелага (не менее 2): 5
Введите кол-во обитаемых островов архипелага (не более 5): 3

Введите номер действия: 1

Введите название архипелага: asd
Введите кол-во островов архипелага (не менее 2): 5
Введите кол-во обитаемых островов архипелага (не более 5): 2

Введите номер действия: 1

Введите название архипелага: qwerty
Введите кол-во островов архипелага (не менее 2): 123
Введите кол-во обитаемых островов архипелага (не более 123): 12

Введите номер действия: 8

Отсутствуют архипелаги, состоящие только из необитаемых островов

Введите номер действия: 9

Введите кол-во островов архипелага (не менее 2): 123
Архипелаг: qwerty; Кол-во островов: 123; Кол-во обитаемых островов: 12

Введите номер действия: 7

Архипелаг: qwe; Кол-во островов: 5; Кол-во обитаемых островов: 3
Архипелаг: asd; Кол-во островов: 5; Кол-во обитаемых островов: 2
Архипелаг: qwerty; Кол-во островов: 123; Кол-во обитаемых островов: 12

Введите номер действия: 6

Введите название архипелага: asd
Архипелаг: asd; Кол-во островов: 5; Кол-во обитаемых островов: 2

Введите номер действия: 4
```

```
Введите название архипелага: asd
Введите новое кол-во обитаемых островов архипелага (не более 5): 3

Введите номер действия: 2

Введите название архипелага: asd
Введите новое название архипелага: qwe
Архипелаг с таким названием уже существует

Введите номер действия: 2

Введите название архипелага: asd
Введите новое название архипелага: zxc

Введите номер действия: 7

Архипелаг: qwe; Кол-во островов: 5; Кол-во обитаемых островов: 3
Архипелаг: zxc; Кол-во островов: 5; Кол-во обитаемых островов: 3
Архипелаг: qwerty; Кол-во островов: 123; Кол-во обитаемых островов: 12

Введите номер действия: 4

Введите название архипелага: zxc
Введите новое кол-во обитаемых островов архипелага (не более 5): 354
Введите новое кол-во обитаемых островов архипелага (не более 5): -6787
Введите новое кол-во обитаемых островов архипелага (не более 5): ftghf
Введите новое кол-во обитаемых островов архипелага (не более 5): 0

Введите номер действия: 8

Имеются архипелаги, состоящие только из необитаемых островов

Введите номер действия: 9

Введите кол-во островов архипелага (не менее 2): 2
Архипелаги отсутствуют

Введите номер действия: 7

Архипелаг: qwe; Кол-во островов: 5; Кол-во обитаемых островов: 3
Архипелаг: zxc; Кол-во островов: 5; Кол-во обитаемых островов: 0
Архипелаг: qwerty; Кол-во островов: 123; Кол-во обитаемых островов: 12

Введите номер действия: 9

Введите кол-во островов архипелага (не менее 2): 5
Архипелаг: qwe; Кол-во островов: 5; Кол-во обитаемых островов: 3
Архипелаг: zxc; Кол-во островов: 5; Кол-во обитаемых островов: 0

Введите номер действия: 6

Введите название архипелага: qwe
Архипелаг: qwe; Кол-во островов: 5; Кол-во обитаемых островов: 3

Введите номер действия: 5

Введите название архипелага: qwe

Введите номер действия: 7
```

```
Архипелаг: zxc; Кол-во островов: 5; Кол-во обитаемых островов: 0
Архипелаг: qwerty; Кол-во островов: 123; Кол-во обитаемых островов: 12

Введите номер действия: 5

Введите название архипелага: zxc

Введите номер действия: 5

Введите название архипелага: qwerty

Введите номер действия: 7

Архипелаги отсутствуют

Введите номер действия: 1

Введите название архипелага: qwe
Введите кол-во островов архипелага (не менее 2): 123
Введите кол-во обитаемых островов архипелага (не более 123): 123

Введите номер действия: 7

Архипелаг: qwe; Кол-во островов: 123; Кол-во обитаемых островов: 123

Введите номер действия: 1

Введите название архипелага: qwe
Архипелаг с таким названием уже существует

Введите номер действия: 12
Введите номер действия: 7

Архипелаг: qwe; Кол-во островов: 123; Кол-во обитаемых островов: 123

Введите номер действия: -67
Введите номер действия: -1
Введите номер действия: 0


Process finished with exit code 0
```