# МИНОБРНАУКИ РОССИИ САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ «ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА) Кафедра МОЭВМ

#### отчет

# по лабораторной работе №6 по дисциплине «Организация ЭВМ и систем»

Тема: Организация связи Ассемблера с ЯВУ на примере программы построения частотного распределение попаданий псевдослучайных целых чисел в заданные интервалы.

| Студентка гр. 9382 | <br>Круглова В. Д. |
|--------------------|--------------------|
| Преподаватель      | <br>Ефремов М.А.   |
|                    |                    |

Санкт-Петербург 2020

#### Цель работы.

Научиться организации связи ассемблера с ЯВУ.

#### Задание.

На языке высокого уровня (Pascal или C) генерируется массив псевдослучайных целых чисел, изменяющихся в заданном диапазоне и имеющих равномерное распределение.

Необходимые датчики псевдослучайных чисел находятся в каталоге Tasks\RAND\_GEN (при его отсутствии программу датчика получить у преподавателя).

Далее должен вызываться ассемблерный модуль(модули) для формирования распределения количества попаданий псевдослучайных целых чисел в заданные интервалы. В общем случае интервалы разбиения диапазона изменения псевдослучайных чисел могут иметь различную длину.

Результирующий массив частотного распределения чисел по интервалам, сформированный на ассемблерном уровне, возвращается в программу, реализованную на ЯВУ, и затем сохраняется в файле и выводится на экран средствами ЯВУ.

## Ход работы:

Данные считываются с помощью команд на ЯВУ, после чего вызывается ассемблерный модуль, который обрабатывает входной массив и возвращает готовый результат на ЯВУ.

## Тестирование.

| Входные данные             | Выходные данные                            |                    |                       |
|----------------------------|--|--------------------|-----------------------|
| Введите длину массива: 10  | Набор случайных чисел: 2 6 2 6 5 4 7 5 4 9 |                    |                       |
| Введите нижний интервал: 1 | Номер интервала<br>1                       | Левая граница<br>1 | Количество чисел<br>6 |

| Введите верхний интервал: 10 Введите количество интервалов: 2 Введите 1 нижних границ интервалов: 5  | 2   | 5                 | 4  |
|--|---|-------------------|--|
| Введите длину массива: 50 Введите нижний интервал: 1 Введите верхний интервал: 50 Введите количество интервалов: 3 Введите 2 нижних границ интервалов: 0 Введенная граница 0 не входит в заданные промежутки! Введите еще раз: 10 25 | 25 20 26 2 20 19 12<br>28 49 49 48 24 20 3<br>Номер интервала<br>1<br>2 | 2 26 32 35 1 38 2 | 16 27 16 45 9 18 39<br>26 45 27 22 19 48 9 3<br>36 35 38 15 34 3 22<br>Количество чисел<br>9<br>15<br>26 |

# Исходный код программы.

# III.cpp:

```
#include <iostream>
#include <fstream>
#include <ctime>
#include <random>

extern "C" void INTERVAL_SORTING(int64_t* LGrInt, int64_t*
borderult, int64_t* array, int64_t NInt);

int main() {
   int64_t NInt = 0;
   int64_t Xmin = 0;
   int64_t Zmax = 0;
   int64_t count = 0;
   int64_t border;
```

```
int64_t RandD(int64_t Xmin, int64_t Ymax)
         std::random_device rd;
         std::mt19937 mt(rd());
         std::uniform_int_distribution<int> dist(Xmin, Ymax);
         return dist(mt);
     }
         std::cout << "Введите длину массива: ";
         std::cin >> NInt;
         // Проверка длины массива
         while (NInt > 16384) {
                 std::cout << "Длинна больше допустимой, введите
заново: ";
            std::cin >> NInt;
         }
         std::cout << "Введите нижний интервал: ";
         std::cin >> Xmin;
         std::cout << "Введите верхний интервал: ";
         std::cin >> Xmax;
         // Проверка верхнего интервала
         while (Xmax <= Xmin) {</pre>
               std::cout << "Введен не коректный верхний интервал,
введите еще раз: " << '\n';
             std::cin >> Xmax;
         }
         std::cout << "Введите количество интервалов: ";
         std::cin >> count;
         // Проверка интервалов
         while (count > 24) {
             std::cout << "Введено не коректное число, введите еще
раз: ";
            std::cin >> count;
         }
         int64_t *LGrInt = new int64_t[count];
```

```
std::cout << "Введите " << count - 1 << " нижних границ
интервалов: ";
         // Считывание нижних границ
         for (int64 t i = 0; i < count - 1; i++) {
             std::cin >> LGrInt[i];
             while (LGrInt[i] > Xmax || LGrInt[i] < Xmin) {</pre>
                 std::cout << "Введеная граница " << LGrInt[i] << "
не входит в заданные промежутки! Введите еще раз: ";
                 std::cin >> LGrInt[i];
             }
         }
         LGrInt[count - 1] = Xmax;
         int64_t *array = new int64_t[NInt];
         // Генерация псевдослучайных чисел
         for (int64_t i = 0; i < NInt; i++) {
             array[i] = RandD(Xmin, Ymin);
         // Обнуляем массив ответ
         int64 t *borderult = new int64 t[count];
         for (int64_t i = 0; i < count; i++) {
             borderult[i] = 0;
         // Вызов ассемблерного модуля
         INTERVAL_SORTING(LGrInt, borderult, array, NInt);
         // Запись в файл и вывод на экран
         std::ofstream out_file("borderult.txt");
         std::cout << "Набор случайных чисел: ";
         out_file << "Набор случайных чисел: ";
         for (int64_t i = 0; i < NInt; i++) {
             std::cout << array[i] << " ";
             out_file << array[i] << " ";
         out file << "\n";
         std::cout << "\n";</pre>
         std::cout << "\nНомер интервала\tЛевая граница\tКоличество
чисел\п";
         out_file << "\nНомер интервала\tЛевая граница\tКоличество
чисел\п";
```

```
for (int64_t i = 0; i < count; i++) {
            if(i != 0) {
                border = LGrInt[i - 1];
            }
            else {
               border = Xmin;
            }
                 out_file << " " << i+1 << "\t\t " <<
border << "\t\t\t</pre> " << borderult[i] << "\n";
                 std::cout << " " << i+1 << "\t " <<
border << "\t\t " << borderult[i] << "\n";</pre>
        }
        // Освобождение памяти
        delete borderult;
        delete array;
        delete LGrInt;
     }
    III.s:
    .intel_syntax noprefix
     .qlobal INTERVAL_SORTING
     .text
    INTERVAL_SORTING: # rdi : LGrInt,    rsi : result ,    rdx :
array, rcx : NInt
         WokWork:
             mov rax, [rdx]
                                          # в rax лежит текущий
элемент
                                            # сохраняем указатель
              push rdx
на текущий элемент
              mov rdx, 0
                                          # обнуляем указатель
        Index_case:
        mov rbx, rdx
                                     # в rbx лежит текущий индекс
из LGrInt
        shl rbx, 3
                                           # этот индекс умножаем
на 8, т.е. каждый элемент по 8 байт
```

```
cmp rax, [rdi + rbx] # сравниваем текущий
                                                           элемент
массива с текущей границей
            jge search_case
                                           если
                                                 элемент массива
больше границы
            jmp write_case
        search case:
         inc rdx
                                                # для перехода к
следующему индексу массива границ
            jmp Index case
        write_case:
         add rbx, rsi
                                        # rbx указывает на индекс
LGrInt
         mov rax, [rbx]
                                     # rax хранит индекс LGrInt
который надо ++
         inc rax
                                              # увеличиваем индекс
на один
         mov [rbx], rax;
                                         # возвращаем обратно
         pop rdx
                                                 # восстанавливаем
указатель на array
         add rdx, 8
                                            # перемещаем указатель
на следующий элемент массива чисел
         loop WokWork
                                       # в цикле проходим по всем
элементам массива
```

ret

#### Выводы.

Была реализована программа организующая связь ассемблера с ЯВУ на примере программы построения частотного распределение попаданий псевдослучайных целых чисел в заданные интервалы.