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

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. 1381 _____ Тулегенова А.О.

Преподаватель _____ Ефремов М.А.

Санкт-Петербург
2022

Цель работы.

Научиться связывать Ассемблер с ЯВУ, написав программу частотного распределение попаданий псевдослучайных целых чисел в заданные интервалы.

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

Исходные данные.

1. Длина массива псевдослучайных целыхчисел - NumRanDat (<= 16K, K=1024)
2. Диапазон изменения массива псевдослучайных целых чисел [Xmin, Xmax] , значения могут быть биполярные;
3. Количество интервалов, на которые разбивается диапазон изменения массива псевдослучайных целых чисел - NInt (<=24)
4. Массив левых границ интервалов разбиения LGrInt (должны принадлежать интервалу [Xmin, Xmax]).

Ход выполнения работы

Программа написана на языке программирования C++. В файле lab6.cpp в функции main() происходит считывание размера массива, диапазон псевдослучайных чисел, количества интервалов и левых границ интервалов. Создаются и сортируются массивы псевдослучайных чисел и интервалов. Далее происходит вызов функции func из ассемблерного модуля. Данная функция получает считанные значения и созданные массивы. Далее проходимся по всем элементам массива. В регистре eax лежит индекс текущего элемента, в регистре ebx лежит индекс текущей левой границы, в регистре ecx лежит количество чисел входящих в текущий диапазон. При переходе на метку check_num проверяется больше ли элемент с индексом eax текущей левой границы, если да то элемент сравнивается со следующей левой границей и после этого программа либо увеличивает количество чисел в текущем диапазоне, либо записывает результат в массив res и переходит к следующей левой границе, пока числа не закончатся. Результат распределения чисел выводится в консоль и в файл. Программа завершается

Тестирование

На картинке представлены результаты работы программы.

```
Array size: 10
XMin: 0
XMax: 10
Number of the intervals: 4
Border 1: 0
Border 2: 3
Border 3: 6
Border 4: 9
Array: 10 0 2 2 6 8 8 7 4 6
Index   Border  Count
1       0       3
2       3       1
3       6       5
4       9       1
```

Рисунок 1. Результаты работы программы.

Вывод

При выполнении лабораторной работы была изучена организация связи Ассемблера с ЯВУ с написанием программы построения частотного распределение попаданий псевдослучайных целых чисел в заданные интервалы.

ПРИЛОЖЕНИЕ А
КОД ПРОГРАММЫ

Название файла: lab6.cpp

```
#include"iostream"
#include <fstream>

using namespace std;
std::ofstream file("out.txt");

extern "C" {void func(int* arr, int length, int* lgi, int count,
int* res); }

int main() {
int length;
int Xmin, Xmax;
int count;

do {
cout << "Array size: ";
cin >> length;
} while ((length > 16000) || (length < 0));

do {
cout << "XMin: ";
cin >> Xmin;
cout << "XMax: ";
cin >> Xmax;
} while (Xmin > Xmax);

int* arr = new int[length];
for (int i = 0; i < length; i++)
arr[i] = Xmin + rand() % (Xmax - Xmin + 1);

do {
cout << "Number of the intervals: ";
cin >> count;
} while ((count > 24) || (count < 1) || (count > (Xmax - Xmin +
1)));

int* lgi = new int[count];
for (int i = 0; i < count; i++) {
do {
cout << "Border " << i + 1 << ": ";
cin >> lgi[i];
} while ((lgi[i] < Xmin) || (lgi[i] > Xmax));
}

for (int i = 0; i < count; i++)
for (int j = i; j < count; j++)
if (lgi[i] > lgi[j])
swap(lgi[i], lgi[j]);
```

```
file << "Array: ";
std::cout << "Array: ";
for (int i = 0; i < length; i++) {
file << arr[i] << " ";
std::cout << arr[i] << " ";
}

for (int i = 0; i < length; i++)
for (int j = i; j < length; j++)
if (arr[i] > arr[j])
swap(arr[i], arr[j]);


int* res = new int[count];
for (int i = 0; i < count; i++)
res[i] = 0;

func(arr, length, lgi, count, res);

std::cout << "\n";
file << "\n";
cout << "Index\tBorder\tCount\n";
file << "Index\tBorder\tCount\n";
for (int i = 0; i < count; i++) {
cout << i + 1 << "\t" << lgi[i] << "\t" << res[i] << "\n";
file << i + 1 << "\t" << lgi[i] << "\t" << res[i] << "\n";
}

file.close();
return 0;
}
```

Название файла: lab6.asm

```
.586p
.MODEL FLAT, C
.CODE

func PROC C arr:dword, lenght:dword, lgi:dword, count:dword,
res:dword

push eax
push ebx
push ecx
push edi
push esi

mov esi, arr
mov edi, lgi
mov eax, 0
mov ebx, 0
```

```
mov ecx, 0

check_num:
cmp eax, lenght
jge stop

push eax
mov eax, [esi + 4 * eax]
cmp eax, [edi + 4 * ebx]
jl next

inc ebx
cmp ebx, count
jge inc_cur

cmp eax, [edi + 4 * ebx]
jl inc_cur
jmp end_boarder

inc_cur:
pop eax
inc ecx
dec ebx
jmp next

end_boarder:
pop eax
dec ebx
mov edi, res
mov [edi + 4 * ebx], ecx
mov edi, lgi
mov ecx, 1
inc ebx
jmp next

next:
inc eax
jmp check_num

stop:
mov edi, res
mov [edi + 4 * ebx], ecx

pop esi
pop edi
pop ecx
pop ebx
pop eax

ret

func ENDP
END
```