**Министерство Образования Республики Беларусь**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**«ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО»**

Кафедра «Информатика»

**Лабораторная работа № 5**

по дисциплине: «**Операционные системы и среды**»

на тему:«**Знакомство со стандартной утилитой GNU make для построения проектов в ОС Unix/Linux**»

Выполнил: ст. гр. ИП-31

Коваленко А.И.

Принял: ст. преподаватель

Самовендюк Н.В.

Гомель 2023

**Цель работы:** ознакомиться с техникой компиляции программ на языке программирования С (С++) в среде ОС семейства Unix/Linux, а также получить практические навыки использования утилиты GNU make для сборки проекта.

**Задание 1:**

Воспользоваться утилитой make для автоматизации сборки проектов из лабораторной работы №4 «Инструментальные средства разработки Linux»

**Листинг makefile:**

all: task

task: task1.o func.o

gcc task1.o func.o -o task -lm

main.o: task1.c

gcc -c task1.c

function.o: func.c

gcc -c func.c

clean:

rm \*.o

**Результат выполнения:**

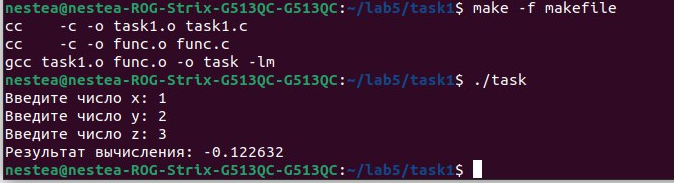
****

Рисунок 1 − Результат выполнения задания 1

**Задание 2:**

Создать make-файл с высоким уровнем автоматизированной обработки исходных файлов программы согласно следующим условиям:

* имя скомпилированной программы (выполняемый или бинарный файл), флаги компиляции и имена каталогов с исходными файлами и бинарными файлами (каталоги, src, bin и т. п.) задаются с помощью переменных в makefile;
* зависимости исходных файлов на языке С (С++) и цели в make-файле должны формироваться динамически;
* наличие цели clean, удаляющей временные файлы;
* каталог проекта должен быть структурирован следующим образом:

src — каталог с исходными файлами;

bin — каталог с бинарными файлами (скомпилированными);

makefile.

Таблица 1

|  |  |  |  |
| --- | --- | --- | --- |
| Вариант | Структура данных (STL) | Функции обработки (таблица 2) | Тип сортировки  (таблица 3) |
| 7 | Array |  |  |

Таблица 2

|  |  |
| --- | --- |
| N | Функции обработки структурированных данных |
| 7 | Произведение отрицательных элементов на нечетных местах |

Таблица 3

|  |  |  |
| --- | --- | --- |
| N | Начальная сортировка | Конечная сортировка |
| 7 | Обменом | Выбором максимального элемента |

**Листинг makefile:**

bin:=bin

src:=src

all: task2 create move

task2: %.o

g++ $^ -o $@

\*

%.o: %.cpp

g++ -c $^

create:

mkdir $(bin)

mkdir $(src)

move:

mv \*.cpp $(src)

mv \*.h $(src)

mv task2 $(bin)

clean:

rm \*.o

**Листинг main.cpp:**

#include <iostream>

#include <string>

#include "function.h"

#include "sortingFunc.h"

using namespace std;

int main()

{

setlocale(0, "ru");

int arr[] = {-6, 1, -7, 2, -7, -8, 9, 16, -13};

int size = sizeof(arr) / sizeof(arr[0]);

cout << "Исходный массив: " << endl;

print(arr, size);

cout << "Произведение отрицательных элементов на нечётных местах: " << getCompNegElem(arr, size) << endl;

Sort(arr, size);

cout << "Массив после сортировки: " << endl;

print(arr, size);

}

**Листинг function.cpp:**

#include <iostream>

#include <string>

#include <algorithm>

using namespace std;

void print(int arr[], int size) {

for (int i = 0; i < size; i++) {

cout << arr[i] << " ";

}

cout << endl;

}

int getCompNegElem(int arr[], int size) {

int composition = 1;

for (int i = 0; i < size; i++) {

if (arr[i] < 0 && ((i + 1) % 2) != 0) {

composition \*= arr[i];

}

}

return composition;

}

**Листинг function.h:**

#pragma once

#include <iostream>

using namespace std;

void print(int arr[], int size);

int getCompNegElem(int arr[], int size);

**Листинг sortingFunc.h:**

#pragma once

#include <iostream>

using namespace std;

void Sort(int arr[], int size);

**Листинг sortingFunc.cpp:**

#include <iostream>

#include <string>

#include <algorithm>

using namespace std;

void Sort(int arr[], int size) {

int tmp;

for (int i = 0; i < size; i++) {

for (int j = 0; j < size - i - 1; j++) {

if (arr[j] > arr[j + 1]) {

tmp = arr[j];

arr[j] = arr[j + 1];

arr[j + 1] = tmp;

}

}

}

}

/\*

void Sort(int arr[], int size) {

for (int i = 0; i < size - 1; i++) {

int biggestIndex = i;

for (int j = i + 1; j < size; j++) {

if (arr[j] > arr[biggestIndex]) {

biggestIndex = j;

}

}

swap(arr[i], arr[biggestIndex]);

}

}

\*/

**Листинг task2.mk:**

all: task2

task2: \*.o

g++ $^ -o $@

%.o: %.cpp

g++ -c $^

**Результат выполнения:**

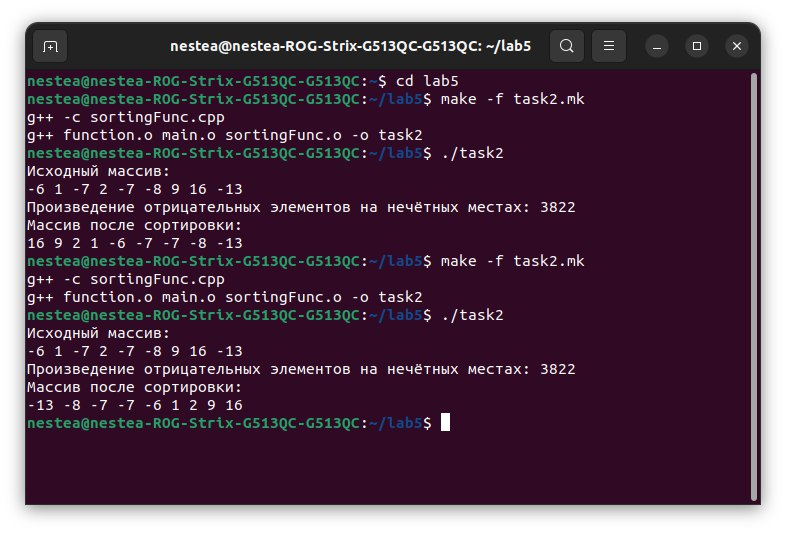


Рисунок 2 − Результат выполнения задания 2

**Вывод:** ознакомились с техникой компиляции программ на языке программирования С (С++) в среде ОС семейства Unix/Linux, а также получить практические навыки использования утилиты GNU make для сборки проекта.