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»**

**Знакомство со стандартной утилитой**

**Gnu mаке для построения проектов в ОС Unix/Linux**

Выполнил студент

группы ИП-32

*Кулинкин К.А.*

Проверил старший преподаватель

*Самовендюк Н.В.*

Гомель 2019 г.

**Цель:** 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. Ознакомиться с теоретическим материалом.

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

3. 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);

makejile.







Все функции размещаются в отдельных файлах.

4. Выполнить программу (скомпилировать, при необходимости отладить) для первого варианта сортировки данных.

5. Изменить тип сортировки и выполнить программу.

6. Показать, что при изменении одного исходного файла и последующем вызове make будут исполнены только необходимые команды компиляции (неизмененные файлы перекомпилированы не будут) и изменены атрибуты и/или размер объектных файлов (файлы с расширением .о).

**Листинг программы:**

**Makefile:**

SOURCES= src/Source.cpp src/sorts.cpp   
task: $(SOURCES)  
rm -f bin/sorts  
g++ $^ -o $@  
mv task bin  
install:  
clear:  
rm -f bin/sorts

**Sorts.h:**

#pragma once

#include <stdio.h>

#include <iostream>

#include <array>

std::array<int, 10> ExchangingSort(std::array<int, 10> arr);

std::array<int, 10> FastSort(std::array<int, 10> arr, int l, int r);

**Sorts.cpp:**

#include "sorts.h"

std::array<int, 10> ExchangingSort(std::array<int, 10> arr)

{

int size = arr.size();

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

{

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

{

if (arr[j + 1] < arr[j])

{

int tmp = arr[j + 1];

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

arr[j] = tmp;

}

}

}

return arr;

}

int MultiplicationOddElements(std::array<int, 10> arr)

{

int result = 1;

for (int i = 1; i < arr.size(); i += 2)

result \*= arr[i];

return result;

}

std::array<int, 10> FastSort(std::array<int, 10> arr, int l, int r)

{

int i = l;

int j = r;

int size = r - l;

int mid = arr.at(size / 2);

do

{

while (arr.at(i) < mid)

{

i++;

}

while (arr.at(j) > mid)

{

j--;

}

if (i <= j)

{

int tmp = arr.at(i);

arr.at(i) = arr.at(j);

arr.at(j) = tmp;

i++;

j--;

}

} while (i <= j);

//Рекурсивные вызовы

if (j > 0)

{

arr = FastSort(arr, 0, j + 1);

}

if (i < size)

{

arr = FastSort(arr, i, size - i);

}

return arr;

}

**Sorts.cpp:**

#include <stdio.h>

#include <array>

#include "sorts.h"

void Print(std::array<int, 10> arr);

std::array<int, 10> FillArray()

{

return { 3, 2, 5, 8, 4, 0, 99, 12, 45, 6 };

}

int main()

{

std::array<int, 10> arr = FillArray();

Print(arr);

arr = ExchangingSort(arr);

//arr = FastSort(arr);

Print(arr);

//system("pause");

return 0;

}

void Print(std::array<int, 10> arr)

{

for (int n : arr)

{

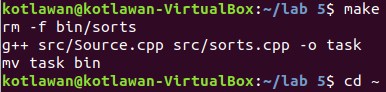
std::cout << n << " ";

}

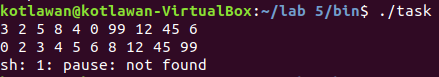
std::cout << std::endl;

}

**Результаты:**



После изменения вида сортировки в файле sorts.cpp:



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