МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

ІНСТИТУТ КОМП’ЮТЕРНИХ ТЕХНОЛОГІЙ, АВТОМАТИКИ ТА МЕТРОЛОГІЇ



**ЛАБОРАТОРНА РОБОТА №3**

**З ПРЕДМЕТУ «*Проектування операційних систем, утиліт і драйверів*»**

***На тему: «gcc - компiлятор С i С++ проекту GNU»***

**Виконав:**

***студент групи КІСП-11***

***Беновський Б.О.***

**Прийняв:**

***Клименко В.А.***

**Львів – 2017 р.**

**Мета:** познайомитися з призначенням та опцiями компiлятора gcc; навчитися використовувати компiлятор gcc для компiляцiї програм на мовi С; познайомитися з призначенням та командами вiдлагоджувача gdb; навчитися використовувати вiдлагоджувач gdb для вiдлагодження програм написаних на мовi С; познайомитися з призначенням та опцiями утилiти make; познайомитися з привилами побудови конфiгурацiйного файла залежностей Makefile; навчитися створювати файл залежностей та використовувати утилiту make для компiляцiї С-програм (в тому числi з використання команди make текстового редактора vim.

**Теоретичні відомості.**

Набір компіля́торів GNU (GNU Compiler Collection, GCC) — набір компіляторів для різних мов програмування. GCC — вільне програмне забезпечення, розроблене Фондом Вільних Програм під ліцензією GNU GPL та GNU LGPL, і є ключовою складовою набору знарядь розробки GNU (GNU development toolchain). Це стандартний компілятор для вільних Юнікс-подібних операційних систем, і деяких пропрієтарних систем, що з них розвинулись, наприклад Mac OS X.

Спочатку називався GNU Компілятор Сі, оскільки підтримував лише одну мову програмування — C. Пізніше був розширений для підтримки C++, Fortran, Java (компілятор GCJ), Ada, та інших.

Компілятор G++ — C++ компілятор (традиційно позначається як GNU C++), є частиною GCC — колекції компіляторів GNU. Використання його команди майже зовсім схоже на GCC.

g++ дозволяє компілювати програмний код в об'єктні модулі, а також виконати лінкування цих модулів у єдину програму. Копмілятор бере до уваги ім'я файлу, щоб визначити мову програмування на якій написаний код. Файли з іменами \*.cc (або \*.cpp) розпізнаються компілятором як C++ файли, а файли з іменами \*.o (в системах Linux) інтерпретуються компілятором як об'єктні модулі (тобто машинний код). Типова команда трансляції файлу із C++ кодом (наприклад названим як file.cpp), у відповідний об'єктний файл вигладає наступним чином: g++ -c file.cpp

**Виконання роботи.**

Варіант №2.

C:\Users\Bohdan\Desktop\Lab3\variant2.PNG

Було створено makefile з таким вмістом.

all:

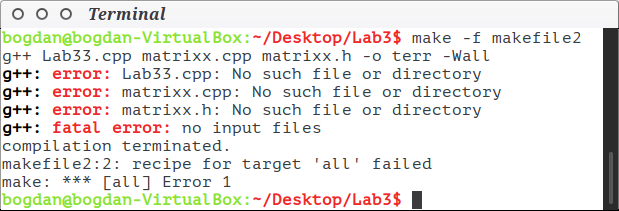
g++ Lab3.cpp matrix.cpp matrix.h -o test

Для отримання звіту про помилки було створено ще один файл makefile2 з внесеними в нього штучно створеними помилками (в і’мя файлів додано зайві букви). Вміст файлу наведено нижче.

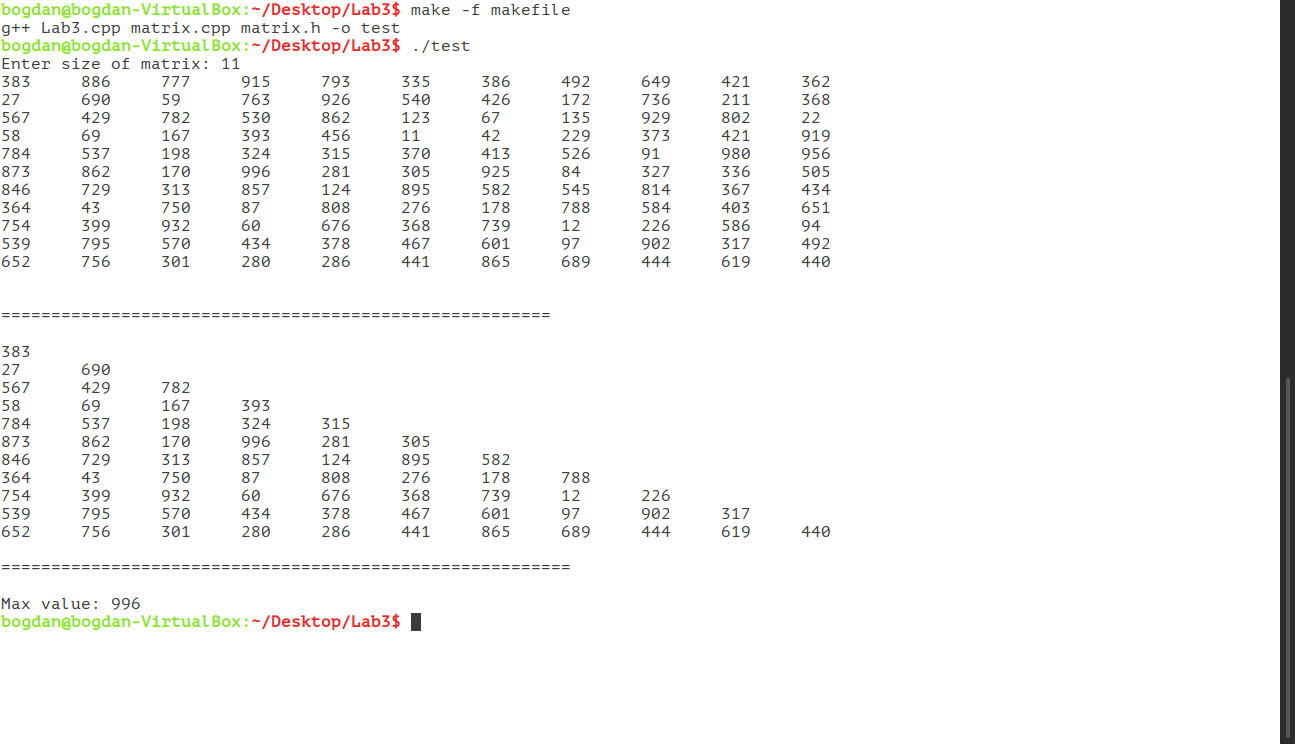
all:

g++ Lab33.cpp matrixx.cpp matrixx.h -o terr -Wall

Результат виконання makefile2 наведено нижче.



Результат виконання makefile (робочий варіант) наведено нижче.

****

**Лістинг matrix.h**

#ifndef MATRIX\_H

#define MATRIX\_H

void allocMatrix(int\*\* &m, int n);

void releaseMatrix(int\*\* &m, int n);

void initMatrix(int\*\* &m, int n);

void showMatrix(int\*\* m, int n);

void showDiagonal(int\*\* m, int n);

void findMaxValue(int\*\* m, int n);

#endif // !MATRIX\_H

**Лістинг matrix.cpp**

#include "matrix.h"

#include <iostream>

#include <cstdlib>

using namespace std;

void allocMatrix(int\*\* &m, int n) {

m = new int\*[n];

for (int i = 0; i < n; i++)

m[i] = new int[n];

}

void releaseMatrix(int\*\* &m, int n) {

for (int i = 0; i < n; i++)

delete[] m[i];

delete[] m;

}

void initMatrix(int\*\* &m, int n) {

for (int i = 0; i < n; i++)

for (int j = 0; j < n; j++)

m[i][j] = rand() % 1000;

}

void showMatrix(int\*\* m, int n) {

for (int i = 0; i < n; i++)

for (int j = 0; j < n; j++)

cout << m[i][j] << "\t";

cout << "\n";

}

void showDiagonal(int\*\* m, int n) {

for (int j = 0; j < n; j++)

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

cout << m[j][i] << "\t";

cout << "\n";

}

void findMaxValue(int\*\* m, int n) {

int maxVal = 0;

for (int j = 0; j < n; j++) {

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

if (j == 0 && i == 0)

maxVal = m[j][i];

else

if (m[j][i] > maxVal)

maxVal = m[j][i];

}}

cout << "Max value: " << maxVal << "\n";

}

**Лістинг Lab3.cpp**

#include <iostream>

#include "matrix.h"

using namespace std;

int main()

{

int n = 0;

cout << "Enter size of matrix: ";

cin >> n;

int \*\*arr = NULL;

allocMatrix(arr, n);

initMatrix(arr, n);

showMatrix(arr, n);

showDiagonal(arr, n);

findMaxValue(arr, n);

releaseMatrix(arr, n);

return 0;

}

**Висновок:** у даній лабораторній роботі було розглянуто призначення та опцiї компiлятора gcc, також було використано компiлятор gcc для компiляцiї програм на мовi С, вiдлагоджувач gdb для вiдлагодження програм написаних на мовi С. Було проведено ознайомлення з призначенням та опцiями утилiти make, з правилами побудови конфiгурацiйного файла залежностей Makefile. Було створено файл залежностей та використовувати утилiту make для компiляцiї С програм.