Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

**Архітектура комп’ютера - 2**

**Лабораторна робота №3**

Виконав:

студент групи ІВ-83

Головенець Р.С.

Перевірив: Нікольський C. C.

Київ - 2020 р.

**CMakeLists.txt**

cmake\_minimum\_required(VERSION 3.16)

project(AK-2\_LAB\_3)

set(CMAKE\_CXX\_STANDARD 14)

add\_executable(AK-2\_LAB\_3 main.cpp)

**Main.cpp**

#include <stdio.h>

#include <getopt.h>

#include <string.h>

int main(int argc, char \*\*argv) {

const char \*shortOptions = "HVNL:";

static struct option longOptions[] = {

{"HELPME", no\_argument, 0, 'H'},

{"VERSIONME", no\_argument, 0, 'V'},

{"LISTME", required\_argument, 0, 'L'},

{"NAMEME", no\_argument, 0, 'N'},

{0, 0, 0, 0},

};

const char \*helpOption = "Program options:\n"

"-H or --HELPME - messages with program options\n"

"-V or --VERSIONME - version\n"

"-N or --NAMEME - file name\n"

"-L[<val1>,...] or --LISTME=[<num1>,...] - list of arguments\n";

bool H\_visited = false;

bool V\_visited = false;

bool L\_visited = false;

bool N\_visited = false;

int index;

int code;

while ((code = getopt\_long(argc, argv, shortOptions, longOptions, &index)) != -1) {

switch (code) {

case 'H': {

if (!H\_visited) {

H\_visited = true;

printf("%s\n", helpOption);

}

break;

}

case 'V': {

if (!V\_visited) {

V\_visited = true;

printf("Version: 1.1.1\n");

}

break;

}

case 'N': {

if (!N\_visited) {

N\_visited = true;

printf("main.cpp\n");

}

break;

}

case 'L': {

if (!L\_visited) {

char \*args;

printf("Arguments list: ");

args = strtok(optarg, ",");

while (args != NULL) {

printf("%s ", args);

args = strtok(NULL, ",");

}

printf("\n");

L\_visited = true;

}

break;

}

default: {

printf("ARGUMENT,WHERE YOU?(TRY -H)\n");

return 0;

}

}

}

return 0;

}

**Виконання роботи**

1.cmake –G Ninja

2.ninja

