Міністерство освіти і науки України Харківський національний університет радіоелектроніки

Кафедра Інформатики

Звіт

з Лабораторної роботи №2

з дисципліни

«Операційні системи»

Виконав: Перевірив:

ITIHФ-20-1 Професор

Самченко С.О. Сінельнікова Т.Ф.

Харків – 2022

**Реєстри**

Необхідно розробити програму client, яка дозволятиме відправляти команди на сервер по локальній мережі по протоколу TCP / IP. У сервері для обробки кожної команди створюється власний потік.Клієнтська програма повинна дозволяти відправити команду малювання кола, прямокутника, текстового повідомлення. Кожен об'єкт в клієнтському додатку задається набором параметрів, в тому числі колір і градієнтна заливка. Відправка команди виконується по натисканні на певну клавішу. Малювання об'єктів у серверному додатку виконується на певній робочій області розміром 640х480 пікселів. Повинні відбуватися перевірки коректності заданих параметрів в командах, щоб можна було забезпечити виведення об'єктів. Малювання об'єктів повинно виконуватися за допомогою GDI +. Дані про кількість намальованих об'єктів кожного типу необхідно зберегти в реєстрі. Після першого запуску перша програма повинна поміщати іконку в Windows Tray і при натисканні комбінації клавіш Ctrl + P має виводитися вікно з інформацією про кількість намальованих об'єктів кожного типу, а при натисканні на комбінацію клавіш Alt + T ці дані повинні видалятися з реєстру.

**Source.cpp:**

#include <iostream>

#include <Windows.h>

#include <atlbase.h>

using namespace std;

HKEY OpenRegisrtyKey(HKEY hRootKey, LPCSTR strSubKey)

{

USES\_CONVERSION;

HKEY hKey;

LONG lError = RegOpenKeyEx(hRootKey, A2W(strSubKey), NULL, KEY\_ALL\_ACCESS, &hKey);

lError = RegCreateKeyEx(hRootKey, A2W(strSubKey), NULL, NULL, REG\_OPTION\_NON\_VOLATILE, KEY\_ALL\_ACCESS, NULL, &hKey, NULL);

return hKey;

}

void SetRegistryValues(HKEY hRootKey, LPCTSTR lpVal, DWORD data)

{

LONG nErr = RegSetValueEx(hRootKey, lpVal, NULL, REG\_DWORD, (LPBYTE)&data, sizeof(DWORD));

if (nErr != ERROR\_SUCCESS)

{

cout << "Error in setting value: " + nErr;

return;

}

}

DWORD GetRegistryValue(HKEY hRootKey, LPCTSTR lpVal)

{

DWORD data;

DWORD dType = REG\_DWORD;

DWORD dSize = sizeof(data);

LONG lErr = RegQueryValueEx(hRootKey, lpVal, NULL, &dType, (LPBYTE)&data, &dSize);

if (lErr != ERROR\_SUCCESS)

{

cout << "Error in getting value: " + lErr;

return NULL;

}

return data;

}

void DeleteRegistryKeyAndConfirm(HKEY hRootKey, LPCSTR strSubKey)

{

USES\_CONVERSION;

LONG lRes = RegDeleteKeyEx(hRootKey, A2W(strSubKey), KEY\_ALL\_ACCESS, NULL);

}

HKEY hKey;

int main()

{

hKey = OpenRegisrtyKey(HKEY\_CURRENT\_USER, "SOFTWARE\\MyRegistry");

if (hKey == NULL)

{

cout << "Error in opening\creating key";

return -1;

}

SetRegistryValues(hKey, L"MyValue1", rand() % 500);

DWORD value = GetRegistryValue(hKey, L"MyValue1");

cout << "Value created: " << value << endl;

system("pause");

return 0;

}

