ЛАБОРАТОРНА РОБОТА №4

Тема: Системний реєстр

Мета роботи: Вивчити призначення та методи роботи з системним реєстром Windows.

Завдання

1. Вивчити призначення та структуру системного реєстру.

2. Вивчити призначення та методи роботи з утилітою RegEdit.

3. Знайти відповідні розділи реєстру в яких є інформація про програми та служби які завантажуються автоматично.

Рис. 1 Вміст поточного користувача

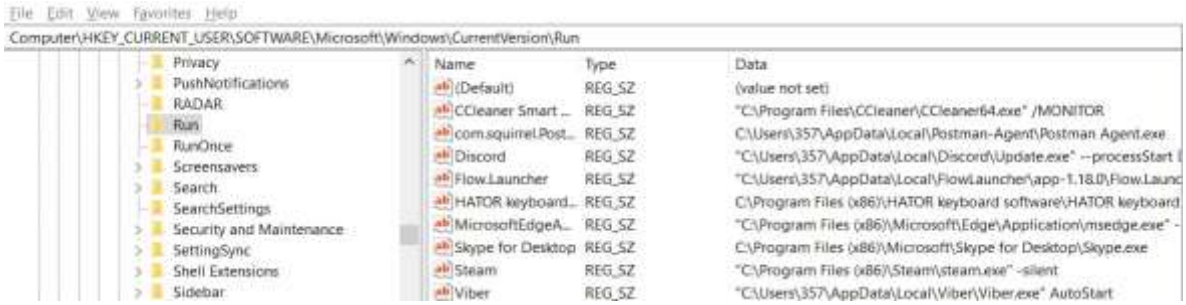

Рис. 2 Вміст всього комп’ютера

4. Програмно, вивести список усіх програм та служб які завантажуються автоматично для усіх користувачів та поточного користувача.

5. Додати програмно до автозавантаження програм для поточного користувача завантаження програми WinWord або іншої.

6. Вивести список повторно, та показали що зареєстрована програма є у списку.

7. Вивести список усіх завдань, які зареєстровані у планувальнику задач системи. Інформації отримати з відповідного розділу реєстру, як для усіх користувачів так і для поточного користувача.

8. Зробити програмно копію будь якого розділу реєстру у файл відповідного формату .reg

Лістинг програми:

```
#include <iostream>
#include <windows.h>
#include <winerror.h>
#include <vector>
#include <winnt.h>
#include <winreg.h>
#include <lmcons.h>

struct value_pair
{
    std::string name;
    std::string value;
};

std::vector<value_pair> read_key(HKEY key)
{
    char name[150];
    char value[150];

    DWORD name_size;
    DWORD value_size;

    std::vector<value_pair> result = std::vector<value_pair>();

    for (int i = 0; ; i++)
    {
        name_size = sizeof(name);
        value_size = sizeof(value);

        LONG error_code = RegEnumValueA(key, i, name, &name_size, 0, NULL,
(LPBYTE)value, &value_size);

        if (error_code != ERROR_SUCCESS) break;

        result.push_back({name, value});
    }

    return result;
}

void output_key_values(HKEY key, std::string sign=" = ")
{
    std::vector<value_pair> keys = read_key(key);

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

```
        std::cout << keys[i].name << sign << keys[i].value << std::endl;
    }
}

DWORD add_value(HKEY key, std::string name, std::string value)
{
    DWORD value_size = value.size() + 1;

    return RegSetValueExA(key, name.c_str(), 0, REG_SZ, (const
BYTE*)value.c_str(), value_size);
}

std::vector<std::string> read_subkeys(HKEY key)
{
    char name[150];

    DWORD name_size;

    std::vector<std::string> result = std::vector<std::string>();

    for (int i = 0; ; i++)
    {
        name_size = sizeof(name);

        LONG error_code = RegEnumKeyEx(key, i, name, &name_size, NULL,
NULL, NULL, NULL);

        if (error_code == ERROR_NO_MORE_ITEMS) break;

        if (error_code != ERROR_SUCCESS)
        {
            std::cerr << "Error enumerating the subkeys. Error code: " <<
error_code << std::endl;
            break;
        }

        result.push_back(name);
    }

    return result;
}

void output_subkeys(HKEY key)
{
    std::vector<std::string> tasks = read_subkeys(key);

    if (tasks.empty()) std::cout << "No subkeys." << std::endl;

    for (int i = 0; i < tasks.size(); i++)
    {
        std::cout << tasks[i] << std::endl;
    }
}

int main()
{
    LONG error_code;
```

```
    HKEY user_startup_key;

    std::string startup_sub_key = "SOFT-
WARE\\Microsoft\\Windows\\CurrentVersion\\Run";
    std::string tasks_sub_key = "SOFTWARE\\Microsoft\\Windows
NT\\CurrentVersion\\Schedule\\TaskCache\\Tree";

    error_code = RegCreateKeyA(HKEY_CURRENT_USER, startup_sub_key.c_str(),
&user_startup_key);

    if (error_code != ERROR_SUCCESS)
    {
        std::cerr << "Error creating a current user key. Error code: " <<
error_code << std::endl;
        return 1;
    }

    HKEY computer_startup_key;

    error_code = RegCreateKeyA(HKEY_LOCAL_MACHINE, startup_sub_key.c_str(),
&computer_startup_key);

    if (error_code != ERROR_SUCCESS)
    {
        std::cerr << "Error creating a local machine key. Error code: " <<
error_code << std::endl;
        RegCloseKey(user_startup_key);
        return 1;
    }

    char user_name[UNLEN + 1];
    DWORD user_name_size = sizeof(user_name);

    GetUserNameA(user_name, &user_name_size);

    std::cout << "Startup applications for user " << user_name << ":" <<
std::endl;
    output_key_values(user_startup_key);

    char computer_name[MAX_COMPUTERNAME_LENGTH + 1];
    DWORD computer_name_size = sizeof(computer_name);

    GetComputerNameA(computer_name, &computer_name_size);

    std::cout << std::endl << "Startup applications for computer " << com-
puter_name << ":" << std::endl;
    output_key_values(computer_startup_key);

    std::string word_value = "C:\\Program Files (x86)\\Microsoft Of-
fice\\Office16\\winword.exe";
    std::string word_name = "Microsoft Word";

    error_code = add_value(user_startup_key, word_name, word_value);

    if (error_code != ERROR_SUCCESS) std::cerr << "Error adding Microsoft
Word to startup. Error code: " << error_code << std::endl;
```

```
    std::cout << std::endl << "Startup applications for " << user_name << "
after adding Microsoft Word to startup:" << std::endl;
    output_key_values(user_startup_key);

    HKEY user_tasks_key;
    HKEY computer_tasks_key;

    HKEY opened_user_tasks_key;
    HKEY opened_computer_tasks_key;

    error_code = RegCreateKeyA(HKEY_CURRENT_USER, tasks_sub_key.c_str(),
&user_tasks_key);
    if (error_code != ERROR_SUCCESS) std::cerr << "Error creating a tasks
user key. Error code: " << error_code << std::endl;

    error_code = RegOpenKeyExW(user_tasks_key, NULL,
KEY_ENUMERATE_SUB_KEYS, KEY_READ, &opened_user_tasks_key);
    if (error_code != ERROR_SUCCESS) std::cerr << "Error openining a tasks
user key. Error code: " << error_code << std::endl;

    error_code = RegCreateKeyA(HKEY_LOCAL_MACHINE, tasks_sub_key.c_str(),
&computer_tasks_key);
    if (error_code != ERROR_SUCCESS) std::cerr << "Error creating a tasks
computer key. Error code: " << error_code << std::endl;

    error_code = RegOpenKeyExW(computer_tasks_key, NULL,
KEY_ENUMERATE_SUB_KEYS, KEY_READ, &opened_computer_tasks_key);
    if (error_code != ERROR_SUCCESS) std::cerr << "Error openining a tasks
computer key. Error code: " << error_code << std::endl;

    std::cout << std::endl << "Registered tasks for user " << user_name <<
":" << std::endl;
    output_subkeys(opened_user_tasks_key);

    std::cout << std::endl << "Registered tasks for computer " << comput-
er_name << ":" << std::endl;
    output_subkeys(opened_computer_tasks_key);

    std::string backup_destination = "C:\\Users\\357\\Desktop\\backup.reg";
    std::string backup_registry =
"HKEY_CURRENT_USER\\SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run";

    std::string backup_command = "reg export  " + backup_registry + " " +
backup_destination;

    system(backup_command.c_str());

    RegCloseKey(user_startup_key);
    RegCloseKey(computer_startup_key);
    RegCloseKey(computer_startup_key);
    RegCloseKey(user_tasks_key);
    RegCloseKey(computer_tasks_key);
    RegCloseKey(opened_user_tasks_key);
    RegCloseKey(opened_computer_tasks_key);
    return 0;
}
```

```
Startup applications for user 357:
Discord = "C:\Users\357\AppData\Local\Discord\Update.exe" --processStart Discord.exe
MicrosoftEdgeAutoLaunch_65ABAB0619E3EE3244824D3E75398BFD = "C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe" --no-startup-window --win-session-start
HATOR keyboard software = C:\Program Files (x86)\HATOR keyboard software\HATOR keyboard software.exe --AutoUp
CCleaner Smart Cleaning = "C:\Program Files\CCleaner\CCleaner64.exe" /MONITOR
Skype for Desktop = C:\Program Files (x86)\Microsoft\Skype for Desktop\Skype.exe
Steam = "C:\Program Files (x86)\Steam\steam.exe" -silent
Flow.Launcher = "C:\Users\357\AppData\Local\FlowLauncher\app-1.18.0\Flow.Launcher.exe"
com.squirrel.Postman-Agent.PostmanAgent = C:\Users\357\AppData\Local\Postman-Agent\Postman Agent.exe
Viber = "C:\Users\357\AppData\Local\Viber\Viber.exe" AutoStart
Microsoft Word = C:\Program Files (x86)\Microsoft Office\Office16\winword.exe

Startup applications for computer DESKTOP-6GKM0QJ:
SecurityHealth = %windir%\system32\SecurityHealthSystray.exe
USB Safely Remove = "C:\Program Files (x86)\USB Safely Remove\USBSafelyRemove.exe" /startup
Autodesk Access = "C:\Program Files\Autodesk\AdODIS\V1\Access\AdskAccessCore.exe" --minimizedUi --autoLaunch
Autodesk Access Service = "C:\Program Files\Autodesk\AdODIS\V1\Setup\AdskAccessService.exe" --autoLaunch

Startup applications for 357 after adding Microsoft Word to startup:
Discord = "C:\Users\357\AppData\Local\Discord\Update.exe" --processStart Discord.exe
MicrosoftEdgeAutoLaunch_65ABAB0619E3EE3244824D3E75398BFD = "C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe" --no-startup-window --win-session-start
HATOR keyboard software = C:\Program Files (x86)\HATOR keyboard software\HATOR keyboard software.exe --AutoUp
CCleaner Smart Cleaning = "C:\Program Files\CCleaner\CCleaner64.exe" /MONITOR
Skype for Desktop = C:\Program Files (x86)\Microsoft\Skype for Desktop\Skype.exe
Steam = "C:\Program Files (x86)\Steam\steam.exe" -silent
Flow.Launcher = "C:\Users\357\AppData\Local\FlowLauncher\app-1.18.0\Flow.Launcher.exe"
com.squirrel.Postman-Agent.PostmanAgent = C:\Users\357\AppData\Local\Postman-Agent\Postman Agent.exe
Viber = "C:\Users\357\AppData\Local\Viber\Viber.exe" AutoStart
Microsoft Word = C:\Program Files (x86)\Microsoft Office\Office16\winword.exe

Registered tasks for user 357:
No subkeys.

Registered tasks for computer DESKTOP-6GKM0QJ:
AAct
ACC
ACCAgent
ACCBackgroundApplication
CCleanerSkipUAC
CreateExplorerShellUnelevatedTask
FTW Host Service Updater v16
GoogleSystem
Intel
IntelSURQC-Upgrade-86621605-2a0b-4128-8ffc-15514c247132
IntelSURQC-Upgrade-86621605-2a0b-4128-8ffc-15514c247132-Logon
IUM-F1E24CA0-863E-4F13-A9E3-4ADE3BFF3473
KMSAuto
Microsoft
MicrosoftEdgeUpdateTaskMachineCore
MicrosoftEdgeUpdateTaskMachineUA
Mozilla
PowerToys
Software Update Application
StartIsBack health check
update-S-1-5-21-2417128146-1049662144-3462800665-1001
update-sys
The operation completed successfully.
```

Рис. Результат програми

Рис. Створений .reg файл

9. За допомогою текстового редактора створити REG файл, за допомогою якого в реєстр у відповідний розділ буде внесено інформацію про асоціацію відкриття файлів .ttt програмою notepad.

Вміст файлу ttt_file.reg:

```
Windows Registry Editor Version 5.00


[HKEY_CLASSES_ROOT\.ttt]
@="tttfile"
```

```
[HKEY_CLASSES_ROOT\tttfile]
@="TTT File"

[HKEY_CLASSES_ROOT\tttfile\shell\open\command]
@="\"C:\\Windows\\system32\\notepad.exe\" \"%1\""
```

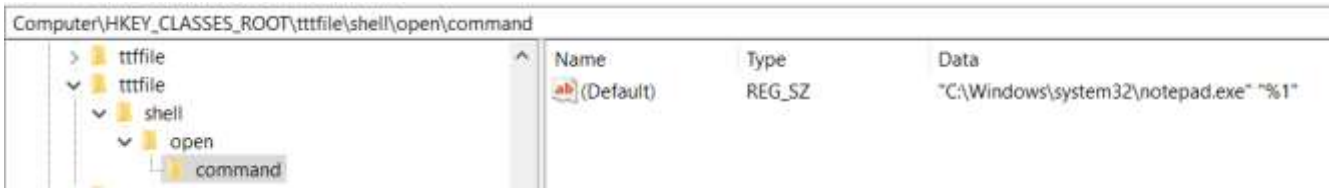

Рис. Вміст регістру після запуску

Рис. Результат

Висоновок: У ході виконання лабораторної роботи було розглянуто призначення системного реєстру Windows та основні методи взаємодії з ним. Отримані знання дозволяють впевнено працювати з реєстром для налаштування параметрів системи та програмного забезпечення.