Rejestr Windows

I. Wstęp

W każdym systemie operacyjnym istnieje potrzeba przechowywania informacji o bieżącej konfiguracji komputera, preferencjach użytkowników, itp. W systemie Windows rolę tę spełnia rejestr systemowy Windows (*Windows Registry*) mający postać centralnej i hierarchicznej bazy danych.

Zawartość rejestru ma duży wpływ na funkcjonowanie systemu, począwszy od sekwencji bootowania a skończywszy na uruchamianiu usług i aplikacji. Jak każda baza danych, tak i rejestr ma ściśle określoną strukturę, która zostanie szczegółowo omówiona w następnych rozdziałach niniejszego opracowania.

Część danych w rejestrze jest przechowywana na dysku twardym, część jest generowana na czas uruchomienia komputera i przechowywana tylko w pamięci operacyjnej. Aplikacje mają dostęp do rejestru poprzez Registry API. Dodając, usuwając i modyfikując odpowiednie klucze w rejestrze systemowym Windows, aplikacje i użytkownicy mogą decydować o sposobie realizacji zadań przez system operacyjny.

II. Historia rejestru

- ***Windows 3.0***

Przyjmuje się, że pierwszą, w miarę sensownie działającą wersją systemu Windows była wersja 3.0. Wtedy też pojawił się zalążek tego, co dzisiaj nazywamy rejestrem. Otóż, oprócz standardowych plików konfiguracyjnych MS-DOS istniało kilka dodatkowych tworzonych przez Windows: control.ini, progman.ini, system.ini oraz win.ini. Zawierały one m.in. informacje o konfiguracji sprzętowej komputera, sterownikach sprzętowych oraz ustawieniach aplikacji. Z racji rozszerzenia nazywano te pliki plikami INI.

Organizacja plików INI miały prostą budowę, były one plikami tekstowymi z wydzielonymi częściami. Miało to umożliwić użytkownikowi większy wpływ na ostateczny wygląd i funkcjonalność systemu. Wtedy też pojawiły się pierwsze funkcje w API służące do edycji plików konfiguracyjnych.

Istniało rozdzielenie na systemowe pliki konfiguracyjne (w nich przechowywano ważne dla systemu ustawienia) oraz pliki profilujące (zawierające m.in. ustawienia czcionek i kolorów).

Pliki INI miały jednak kilka poważnych wad:

- **Zbyt łatwa modyfikowalność**
  Każdy mógł samodzielnie doprowadzić do nieodwracalnych zmian w systemie operacyjnym. Takie „zmiany" często wymuszały reinstalację całego oprogramowania
- **Łatwość doprowadzenia do niespójności w konfiguracji**
  Przez nieskomplikowane samodzielne modyfikacje rejestru można było doprowadzić do sytuacji, w której system na podstawie plików INI odwoływał się do zasobów, których nie było np. do nieistniejących fontów, urządzeń itp.
- **Brak reguł wyznaczających, co może być przechowywane w rejestrze**
  Firma MICROSOFT nie wyznaczyła jednoznacznych reguł, co powinno być przechowywane w rejestrze, a co nie. W rezultacie każdy producent oprogramowania zapisywał rejestru, co chciał, często go jedynie zaśmiecając.
- **Limit objętości plików**
  Każdy plik INI mógł mieć co najwyżej 64kB. W rezultacie często dochodziło do przepełnień.

- **Limit ilości użytkowników**
  Ówczesny system Windows działał wg zasady: jeden komputer – jeden użytkownik. To oznaczało, że nie przewidziano możliwości łatwego przechowywania wielu profilów konfiguracyjnych dla wielu różnych użytkowników.

Windows 3.1

Powyższe ograniczenia częściowo zlikwidowano w Windows 3.1. Pojawił się także odrębny rejestr systemowy Windows. Posiadał on klucze i podklucze z wartościami tak jak dzisiejsze wersje rejestru.

Pliki INI nadal istniały, lecz nie pełniły już aż tak odpowiedzialnej funkcji jak w Windows 3.0. Rejestr miał ograniczenie 64kB, ale nie był plikiem tekstowym – do modyfikacji trzeba było używać specjalnego programu – Windows Registry Editor. Nieujawniono dokumentacji przechowywania danych w rejestrze, co wymusiło na programistach używanie WINAPI dla uzyskania dostępu do rejestru.

NT 3.1

W 1993 wprowadzono na rynek Windows NT 3.1. Nowy system operacyjny z założenia miał oferować wielodostępność oraz większe bezpieczeństwo. Aby to umożliwić – zwiększono rolę i możliwości rejestru systemowego. Od tej pory nie było już ograniczeń objętości, spójność danych konfiguracyjnych z faktycznym stanem była kontrolowana na bieżąco. Tylko użytkownicy z określonymi przywilejami mieli uprawnienia do modyfikacji rejestru. Pojawiły się dodatkowe funkcjonalności - rejestr mógł teraz być kontrolowany przez administratorów zdalnie. Poprawiono edytor rejestru – pojawił się RegEdt32.exe.

95 i 98

Rejestr w Windows 95 i 98 był bardzo zbliżony pod względem funkcjonalności do tego z Windows NT 3.1 – 3.51. Mimo tego, z powodu różnicy w realizacji wewnętrznych mechanizmów, nie istniała programowa możliwość przeniesienia konfiguracji między NT a 95/98.

NT 4.0

Windows NT 4.0 zaadaptował ułatwienia obsługi rejestru z Windows 95/98 do linii NT. Uporządkowano funkcje do ingerencji w rejestr tworząc z nich odrębne Registry API. W celu umożliwienia uruchamiania programów dla Windows 95/98 pod Windows NT 4.0, przeniesiono także część struktury rejestru z Windows 95/98.

Windows 2000

Windows 2000 nie przyniósł rewolucyjnych zmian, jeśli chodzi o rejestr. Zmieniono nieznacznie strukturę rejestru i ulepszono narzędzia do administracji dostępem do poszczególnych obiektów systemowych. Nowy system kontroli nad sprzętem przechowywał dane właśnie w rejestrze – kontrolując na bieżąco zmiany w konfiguracji komputera.

Windows XP

Windows XP przyniósł ostateczne scalenie dwóch linii produktów – Windows NT/2000 i Windows 95/98/Me. Wprowadzono kolejne ulepszenia w zakresie zarządzania profilami użytkowników i bezpieczeństwa. Tego samego typu zmiany wprowadzono w Windows 2003 – bez rewolucji, raczej zwiększanie ilości przechowywanych danych. Także przesunięcia w strukturze rejestru.

III. Struktura rejestru

- ***Omówienie***

Rejestr jest bazą danych o strukturze drzewiastej. Węzeł w drzewie nosi nazwę klucza (ang. *key*). Każdy klucz może zawierać podklucze (*subkeys*) i/lub wartości (*values*). Każdy klucz charakteryzuje nazwa składająca się z minimum jednego znaku (drukowalnego bądź niedrukowalnego), aczkolwiek nazwa klucza nie może zawierać backslash'a '\'. Każda wartość ma swoją nazwę (*value name*) oraz daną (*value*).

- ***Klucze predefiniowane***

Trzon rejestru stanowią tzw. klucze predefiniowane (*predefined keys*). W zależności od wersji Windows ich ilość jest różna, ponadto część z nich nie jest widoczna dla systemowego edytora (regedit.exe lub regedt32.exe). Poniżej zamieszczono omówienie zawartości kluczy predefiniowanych.

Klucz	Opis
HKEY_CLASSES_ROOT	Zawiera informacje dot. shell'a, kontrolek COM. Zapewnia zgodność wstecz z aplikacjami dla Windows 3.1. Nie powinien być używany przez usługi oraz aplikacje dla wielu użytkowników.
HKEY_CURRENT_CONFIG	Alias klucza: ***[HKEY_LOCAL_MACHINE\ System\CurrentControlSet\Hardware Profiles\Current]*** Przechowuje dane o bieżącym profilu sprzętowym na lokalnym komputerze. Przechowywana jest tylko informacja, w jaki sposób obecna konfiguracja różni się od tej określonej w HKEY_LOCAL_MACHINE. **Uwaga:** nie istnieje w Windows NT 3.51 i niższych.
HKEY_LOCAL_MACHINE	Zawiera szczegółowe informacje o konfiguracji komputera, zainstalowanym sprzęcie i oprogramowaniu (w tym dane o całym sprzęcie, który był kiedykolwiek na danym komputerze zainstalowany).
HKEY_USERS	Przechowuje dane o domyślnym profilu użytkownika dla nowych użytkowników oraz konfiguracje użytkowników.
HKEY_CURRENT_USER	Alias do odpowiedniej gałęzi w HKEY_USERS. Zawiera informacje o ustawieniach bieżącego użytkownika. M.in. zmienne systemowe, ustawienia drukarek, sieci, preferencje aplikacji itp.
HKEY_DYN_DATA	Tylko Windows 95/98/Me: Pozwala zbierać dane o wydajności systemu.

HKEY_PERFORMANCE_DATA	Pozwala zbierać dane o wydajności systemu. Dane nie są przechowywane w rejestrze, ale są dostępne przez Registry API. **Uwaga:** Nie istnieje w Windows 95/98/Me.
HKEY_PERFORMANCE_NLSTEXT	Zawiera ciągi znaków odnoszące się do liczników w języku, w którym pracuje użytkownik. Dane te nie są dostępne z poziomu Regedit i RegEdt32. **Uwaga:** klucz istnieje tylko w Windows XP
HKEY_PERFORMANCE_TEXT	Zawiera ciągi znaków odnoszące się do liczników w języku US English. Dane te nie są dostępne z poziomu Regedit i RegEdt32. **Uwaga:** klucz istnieje tylko w Windows XP

W literaturze nazwy predefiniowanych kluczy, często pojawiają się w postaci skróconej:

Nazwa klucza	Skrót
HKEY_CURRENT_CONFIG	HKCC
HKEY_CLASSES_ROOT	HKCR
HKEY_CURRENT_USER	HKCU
HKEY_LOCAL_MACHINE	HKLM

- ***Zbiorniki***

Zbiorniki (*hives*) to niektóre fragmenty drzewa rejestru. Spośród innych zbiorów kluczy, wyróżnia je fakt, że każdemu zbiornikowi przyporządkowany jest odrębny plik na dysku, w którym przechowywane są odpowiednie dane z rejestru. Pliki te, zamiast plików INI, to aktualne lokalizacje najważniejszych dla systemu operacyjnego danych konfiguracyjnych. Należy zwrócić uwagę, że zbiorniki to te części rejestru, dla których należy przeprowadzać regularny backup. Dokonujemy tego poprzez polecenia Import Registry File oraz Export Registry File w aplikacji Regedit.exe.

Poniżej standardowe zbiorniki wraz z plikami, które je przechowują.

Wszystkie pliki zbiorników poza HKEY_CURRENT_USER znajdują się w katalogu KatalogWindows\System32\Config.

Zbiornik (jego ścieżka w rejestrze)	Pliki, w których przechowywane są zbiorniki
HKEY_LOCAL_MACHINE\SAM	Sam, Sam.log, Sam.sav
HKEY_LOCAL_MACHINE\Security	Security, Security.log, Security.sav
HKEY_LOCAL_MACHINE\Software	Software, Software.log, Software.sav
HKEY_LOCAL_MACHINE\System	System, System.alt, System.log, System.sav
HKEY_CURRENT_CONFIG	System, System.alt, System.log, System.sav
HKEY_CURRENT_USER	Ntuser.dat, Ntuser.dat.log
HKEY_USERS\.DEFAULT	Default, Default.log, Default.sav

- ***Typy wartości***

Wszystkie dozwolone w rejestrze typy danych są zdefiniowane w pliku nagłówkowym WinNT.h.

Mimo tego tylko część z nich da się wstawić do rejestru korzystając z edytora regedit.exe bądź RegEdt32.exe.

Typy danych, których można używać zarówno w edytorze, jak i korzystając z WINAPI:

Typ wartości	Opis
REG_SZ	W polskiej wersji Windows nazwana „wartość ciągu". Tradycyjny ciąg znaków zakończony znakiem \0 (*null-terminated string*). Może być w standardzie ANSI lub Unicode zależnie od funkcji API, której użyjemy
REG_BINARY	„Wartość binarna". Dane binarne w dowolnej postaci.
REG_DWORD	„Wartość DWORD". Liczba 32-bitowa. Standardowo w postaci little-endian (mniej znacząca część liczby na początku).
REG_MULTI_SZ	„Wartość wielociągu". Tablica ciągów znakowych zakończonych znakami '\0'. Sama tablica jest zakończona dwoma znakami '\0'.
REG_EXPAND_SZ	„Wartość ciągu rozwijalnego". Ciąg znaków zakończony znakiem '\0', z tą różnicą, że zawiera odniesienia do zmiennych środowiskowych np. „%PATH%". Do podstawienia wartości tych zmiennych można użyć funkcji EpandEnvironmentStrings. Wybór kodowania znaków Unicode/ANSI zależy od użytej funkcji API.

Pozostałe typy danych dostępne z poziomu API, niedostępne z poziomu edytora rejestru.

Typ wartości	Opis
REG_DWORD_LITTLE_ENDIAN	Standardowo to samo, co REG_DWORD.
REG_DWORD_BIG_ENDIAN	Liczba 32-bitowa w formacie big-endian
REG_LINK	Typ zarezerwowany na użytek systemu operacyjnego.
REG_NONE	Nie zdefiniowany typ wartości
REG_QWORD	Liczba 64-bitowa
REG_QWORD_LITTLE_ENDIAN	Liczba 64-bitowa w formacie little-endian (to samo co REG_QWORD).

- ***Ograniczenia typów danych***

Poniższe limity opisują górne ograniczenia dla różnych typów danych w rejestrze. Należy je uwzględniać zarówno przy edycji rejestru za pomocą programów typu regedit jak i za pomocą API.

Element	Ograniczenie
Długość nazwy klucza	Max 255 znaków
Długość nazwy wartości	Win XP/2003 : 16383 znaków Win 2000 : 260 ANSI lub 16383 znaków Unicode Win 95/98/Me: 255 znaków
Objętość wartości	Win 95/98/Me : 16300 bajtów. Ponadto sumaryczna objętość wszystkich wartości w kluczu może wynosić max 64KB Pozostałe wersje Windows (NT/2000 i wyżej) – cała dostępna pamięć

Uwaga: Zaleca się przechowywanie długich (2KB i więcej) wartości jako pliki a w rejestrze umieszcza się tylko nazwy tych plików. Takie rozwiązanie pozwala na szybsze wykonywanie zadań przez system operacyjny.

- ***Prawa dostępu***

Korzystając z edytora RegEdt32 z uprawnieniami Administratora w Windows NT/2000/XP/2003 mamy możliwość ustalania uprawnień dostępu do poszczególnych kluczy dla poszczególnych użytkowników. Uprawnieniami do modyfikacji rejestru zarządza się w sposób analogiczny jak uprawnieniami do plików w Windows z rodziny NT. Konkretnemu użytkownikowi/grupie użytkowników możemy przydzielić/odebrać prawa wykonania dowolnej czynności na dowolnym kluczu w rejestrze.

IV. Narzędzia do edycji rejestru.

W każdej wersji Windows posługującej się rejestrem, mamy do dyspozycji kilka narzędzi dostarczanych wraz z systemem operacyjnym. W zależności od wersji Windows prezentują one różne możliwości. Poniżej znajduje się krótka prezentacja tych narzędzi.

Regedit i RegEdt32 to edytory graficzne, pozwalające na przeglądanie rejestru w intuicyjny sposób. Ekran właściwy aplikacji jest podzielony na dwie części, po lewej znajduje się drzewo kluczy rejestru, z zaznaczoną aktualnie przeglądaną przez nas pozycją, natomiast w prawej części okna znajdują się wartości przechowywane w wybranym przez nas kluczu.

Oba powyższe edytory mają opcję szukania kluczy i wartości w rejestrze oraz pozwalają na eksport i import kluczy i edycję rejestrów na zdalnych komputerach. Oba programy uruchamiamy: Start->Uruchom i wpisujemy regedit lub regedt32

- ***Regedit.exe***

Pierwszy raz pojawił się w Windows 95. Przez to nie wspiera wszystkich możliwości rejestru w systemach NT/2000 i wyższych. Brak jest obsługi praw dostępu do poszczególnych kluczy, obsługuje tylko część formatów typów danych. W Windows XP został całkowicie wyparty przez RegEdt32 (odwołanie do Regedit uruchamia RegEdt32).

- ***RegEdt32.exe***

Nowsza wersja Regedit. Uwzględnia wszystkie opcje związane z prawami dostępu i obsługuje więcej typów danych.

- ***Reg.exe***

W przeciwieństwie do dwóch poprzednich programów, reg.exe jest aplikacją konsoli. Uruchomienie reg.exe z poziomu konsoli z opcją /? wyświetli pełną listę poleceń programu. Reg.exe umożliwia manipulację rejestrem z poziomu linii poleceń, posiada funkcjonalność zbliżoną do jego graficznych odpowiedników.

- ***Pliki z rozszerzeniem '.reg'***

Możliwe jest tworzenie patch'y do rejestru systemowego w postaci plików. Taki plik można przenieść na inny komputer, a uruchomienie go spowoduje wpisanie zawartych w nim zmian w rejestr komputera, na którym plik ten zostanie uruchomiony.

Oto przykładowa zawartość pliku .reg :

```
REGEDIT4

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Java VM\Security]
"EditCustomPermissions"=hex:00,00,00,00
```

Jak widać plik zawiera standardowo „nagłówek" REGEDIT4, następnie w nawiasach kwadratowych znajduje się lokalizacja klucza do edycji w drzewie rejestru. Po tym następuje nazwa wartości i sama wartość (wraz z typem danych).

Pliki takie można utworzyć na kilka sposobów: w każdym edytorze tekstu, bądź też korzystając z funkcji „Eksportuj" programu regedit.

V. Registry API.

- ***Wprowadzenie***

Registry API jest integralną częścią WINDOWS API. Składa się w większości z funkcji z przedrostkiem „Reg". Registry API umożliwia manipulację kluczami i wartościami w rejestrze oraz manipulację prawami dostępu do kluczy. Dodatkowo należy wspomnieć o funkcjach wchodzących w skład Shell API, o podobnym działaniu jak te z Registry API.

Pełna lista funkcji wchodzących w skład Registry API i innych powiązanych z rejestrem znajduje się na końcu opracowania, w załączniku A, opis niektórych funkcji znajduje się w omówieniu załączonych programów.

- ***Funkcje***

Podstawowe funkcje Registry API to:

RegOpenKeyEx
RegCloseKey
RegEnumValue
RegEnumKeyEx

Niżej wymienione funkcje z Registry API służą do obsługi plików INI. Funkcje te pozostały w API dla zachowania zgodności z Windows 16-bit. Aplikacje 32-bitowe powinny używać tylko rejestru systemowego.

GetPrivateProfileInt
GetPrivateProfileSection
GetPrivateProfileSectionNames
GetPrivateProfileString
GetPrivateProfileStruct
GetProfileInt
GetProfileSection
GetProfileString
WritePrivateProfileSection
WritePrivateProfileString
WritePrivateProfileStruct
WriteProfileSection
WriteProfileString

Niżej wymienione funkcje istnieją w Registry API tylko ze względu na kompatybilność z wersjami 16-bitowymi – są przestarzałe i nie należy ich używać.

RegCreateKey
RegEnumKey
RegOpenKey
RegQueryValue
RegSetValue

Szczegółowy i wyczerpujący opis wszystkich funkcji z Registry API znajduje się w Windows Platform SDK. Dla poprawnego użycia wymienionych funkcji niezbędne jest załączanie odpowiednich plików nagłówkowych, najczęściej winreg.h, windows.h. Wymagany jest także plik biblioteczny Advapi32.lib.

- ***Struktury danych specyficzne dla Registry API***

Struktura VALENT zawiera dane o wartości z rejestru, używa jej funkcja RegQueryMultipleValues

```
typedef struct value_ent {
  LPTSTR ve_valuename;
  DWORD ve_valuelen;
  DWORD_PTR ve_valueptr;
  DWORD ve_type;
} VALENT, *PVALENT;
```

Szczegółowy opis pól struktury znajduje się w Platform SDK.

- ***Omówienie załączonych programów***

Ogólna charakterystyka załączonego kodu źródłowego.

Załączony kod źródłowy to 3 krótkie programy napisane w języku C z użyciem standardowych bibliotek ANSI oraz Registry API w środowisku Microsoft Visual .NET 2003. Programy uruchamia się z poziomu konsoli. Taki kod jest krótszy, przez co pozwala lepiej zrozumieć działanie funkcji z Registry API. Programy są przystosowane do działania w systemach z rodziny NT, ich działanie w pozostałych systemach będzie się różnić, m.in. z powodu odmiennej lokalizacji kilku kluczy w rejestrze – obsługa błędów została zapewniona.

Typy danych używane w programach (z wyjątkiem typów ANSI C):

Nazwa typu	Opis
HKEY	Uchwyt analogiczny do HANDLE w WINAPI, reprezentujący konkretny klucz rejestru. Może on przyjmować albo wartość zwróconą przez RegCreateKeyEx, lub RegOpenKeyEx, albo być jedną ze stałych: HKEY_CLASSES_ROOT HKEY_CURRENT_CONFIG HKEY_CURRENT_USER HKEY_LOCAL_MACHINE HKEY_USERS

DWORD	Alias typu danych ANSI „unsigned long”
LONG	Alias typu danych ANSI „long”
FILETIME	Struktura wykorzystywana do przechowywania dat odnoszących się do plików.

UWAGA: Większość funkcji z Registry API zwraca ERROR_SUCCESS (def. w winerror.h w Platform SDK) w przypadku nie wystąpienia błędu. Wartość zwracana jest typu LONG.

Szczegółowy opis funkcji Registry API znajduje się w MSDN Library (dostępne także w wersji online na stronie http://msdn.microsoft.com). Z tego powodu, w opisie programów znajduje się raczej przeznaczenie konkretnych funkcji z Registry API, niż ich kompletna dokumentacja.

Efekty działania programów procesor.c i rozszerzenie.c należy obserwować za pomocą edytorów rejestru regedit.exe lub RegEdt32.exe w odpowiednich kluczach, w momencie, kiedy pojawia się komunikat „Nacisnij dowolny klawisz”.

Zawsze należy uwzględniać sytuacje typu:

- Brak uprawnień użytkownika do rejestru (także do konkretnych kluczy)
- Klucz, który właśnie chcemy stworzyć istnieje już w rejestrze

Omówienie poszczególnych plików

infoosystemie.cpp

Działanie

Program próbuje odczytać podklucze oraz wartości z klucza

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion]
```

Po odczytaniu nazw wartości i podkluczy, wyświetla je na ekranie.

Opis funkcji

```
RegOpenKeyEx(HKEY_LOCAL_MACHINE,klucz1,0,
         KEY_QUERY_VALUE | KEY_ENUMERATE_SUB_KEYS,&hKey1);
```

Funkcja otwiera klucz w rejestrze, w gałęzi HKEY_LOCAL_MACHINE, o nazwie

```
klucz1[]="SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion";
```

(pojedynczy ‘\’ to znak ucieczki, nazwa jest pozbawiona nazwy klucza predefiniowanego). Trzeci argument zawsze jest zerem, dalej zdefiniowane są prawa, jakie chcemy mieć przy dostępie do klucza. „&hKey1” to wskaźnik do uchwytu, który zostanie zdefiniowany w efekcie działania tej funkcji.

```
RegQueryInfoKey(...);
```

Ta funkcja służy do wydobywania informacji o wybranym kluczu. Wśród zwracanych informacji są:

- Liczba wartości,

- Liczba podkluczy,
- Długość najdłuższej nazwy wartości,
- Długość najdłuższej danej wartości

```
RegEnumKeyEx(hKey1,i,nazwaklucza,&dlbuf,0,NULL,NULL,&ftLastWriteTime);
```

Wywołanie tej procedury w pętli umożliwia łatwe wyliczenie wszystkich podkluczy klucza o uchwycie hKey1. W zmiennej „nazwaklucza" zwracana jest nazwa podklucza, natomiast „dlbuf" zawiera max. długość łańcucha znakowego z nazwą podklucza.

```
RegEnumValue(hKey1,i,nazwawartosci,&dlbuf,0,NULL,NULL,NULL);
```

Pełni analogiczną funkcję dla wyliczania wartości, co RegEnumKeyEx dla kluczy.

```
RegCloseKey(hKey1);
```

Otwarty klucz zawsze należy po zakończeniu operacji zamykać, do czego służy właśnie ta funkcja.

procesor.cpp

Działanie

Program przedstawia możliwość zmiany wartości w rejestrze. Najpierw odczytywane są informacje zawarte w kluczu:

```
[HKEY_LOCAL_MACHINE\HARDWARE\DESCRIPTION\System\CentralProcessor]
```

Następnie, na ich podstawie, program umożliwia zmianę tekstu opisującego procesor (efekt zmiany jest widoczny we właściwościach systemu, w momencie, w którym program czeka na reakcję użytkownika). Zmiana ta zostaje cofnięta po naciśnięciu dowolnego klawisza podczas działania programu.

Opis funkcji

W programie użyto części funkcji z poprzedniego programu (należy zwrócić uwagę, że tym razem w RegOpenKeyEx jest żądanie pełnych praw dostępu do rejestru).

Pozostałe funkcje to:

```
RegQueryValueEx(hKey2,nazwawartosci,0,&typd,(LPBYTE) bufor,&dlbuf2);
```

Funkcja umożliwia odczytanie danej z wartości o nazwie „nazwawartości". Dodatkowo należy określić uchwyt klucza, bufor do odbioru danej oraz wielkość tego bufora.

```
RegSetValueEx(hKey2,nazwawartosci,0,REG_SZ,NowaWartosc,sizeof(NowaWartosc))
```

Funkcja ta pozwala na stworzenie nowej lub edycję istniejącej wartości w kluczu. Jej rezultat zależy od tego, czy wartość nazwie „nazwawartości" istnieje, czy też nie.

rozszerzenie.cpp

Działanie

Program „rejestruje” podane przez użytkownika nowe rozszerzenie pliku w systemie i kojarzy je z domyślnym edytorem plików tekstowych (najczęściej notepad.exe). Jeżeli podane rozszerzenie było wcześniej zarejestrowane to żadne dane nie są wpisywane do rejestru. Na koniec działania programu, rozszerzenie pliku jest wyrejestrowywane.

Opis funkcji

```
RegCreateKeyEx(HKEY_CLASSES_ROOT,klucz1,0
        NULL,REG_OPTION_NON_VOLATILE,
        KEY_ALL_ACCESS,NULL,&hKey,&Disposition);
```

Funkcja służy do tworzenia nowych kluczy w rejestrze. Jeśli klucz o podanej nazwie już istnieje, to ten klucz jest tylko otwierany, a to czy klucz istniał wcześniej, czy nie odnotowywane jest w parametrze „Disposition”. Za pomocą opcji REG_OPTION_NON_VOLATILE (inne możliwości: REG_OPTION_VOLATILE/ REG_OPTION_BACKUP_RESTORE) decydujemy, że klucz może być przechowywany nie tylko w pamięci, ale i w pliku (zapisu takiego klucz dokonuje funkcja RegSaveKey).

```
RegDeleteValue(hKey,nazwawartosci);
```

Usuwa wartość z klucza o uchwycie hKey.

```
RegDeleteKey(HKEY_CLASSES_ROOT,klucz1);
```

Usuwa klucz wraz z wartościami o ścieżce złączonej z obu parametrów.

UWAGA:

- Drugi parametr tej funkcji nie może być NULL
- Przed usunięciem klucza należy zamknąć wszystkie uchwyty HKEY do niego prowadzące
- Usuwany klucz nie może mieć podkluczy. Aby, usunąć klucz z podkluczami należy stworzyć odpowiednią procedurę usuwającą podklucze rekurencyjnie

VI. Ciekawostki:

Wszelkie ciekawostki dot. rejestru, zmiany jego ustawień powinny być wprowadzane tylko z pełną świadomością konsekwencji. Autor opracowania nie gwarantuje działania wymienionych poniżej sztuczek (mimo, że sam je sprawdzał na swoim komputerze) i nie bierze odpowiedzialności za ich nieodpowiednie użycie.

Uwaga do wprowadzania zmian: należy przyjąć, że ma znaczenie wielkość liter w stosunku dla wszystkich przechowywanych wartości znakowych.

- ***Więcej informacji w Menedżerze Urządzeń.***

Dotyczy: WIN 2000/XP

Opis: Dodatkowe informacje dot. urządzeń widocznych w Menedżerze Urządzeń. Trick polega na dodaniu dodatkowego klucza pod adresem podanym poniżej.

Klucz: `[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Environment]`

Nazwa nowej wartości: DEVMGR_SHOW_DETAILS

Typ danych: REG_SZ (Wartość ciągu)

Możliwe wartości: 0 = domyślne - wyłączone, 1 = włącz pokazywanie dodatkowej informacji

- ***Wyłączenie Autorun***

Dotyczy: WIN NT/2000/XP

Opis : Wyłączenie standardowo włączonego Autorun.

Klucz: `[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\CDRom]`

Nazwa istniejącej wartości: Autorun

Typ danych: REG_DWORD

Możliwe wartości: 0 – Autorun wyłączony, 1 – Autorun wyłączony

- ***Autouzupełnianie w konsoli***

Dotyczy: WIN 2000/XP

Opis: Umożliwia wybór klawisza uzupełniającego nazwę pliku/polecenia w konsoli – jak w bash'u UNIX'a klawisz TAB.

Klucz: `[HKEY_CURRENT_USER\Software\Microsoft\Command Processor]`

Nazwa istniejącej wartości: CompletionChar

Typ danych: REG_DWORD

Uwaga: Aby ustawić klawisz uzupełniania na TAB należy wpisać liczbę 9.

- ***Wyświetlanie wersji systemu operacyjnego***

Dotyczy: WIN 2000/XP

Opis: Włączenie wypisywania w prawym dolnym rogu wersji systemu operacyjnego

Klucz: `[HKEY_CURRENT_USER\Control Panel\Desktop]`

Nazwa istniejącej wartości: PaintDesktopVersion

Typ danych: REG_DWORD

Możliwe wartości: 0-wyłączone, 1-włączone

Uwaga: Wymaga restartu systemu.

- ***Czyszczenie pliku wymiany po wylogowaniu się***

Dotyczy: WIN 2000/XP

Opis: Włączenie tej opcji wymusza na systemie każdorazowe czyszczenie zawartości pliku wymiany po wylogowaniu się użytkownika. Dane z pliku wymiany mogą zostać wykorzystane do włamania się na konto użytkownika.

Klucz: `[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager\Memory Management]`

Nazwa istniejącej wartości: ClearPageFileAtShutdown

Typ danych: REG_DWORD

Możliwe wartości: 0-wyłączone, 1-włączone

Uwaga: Włączenie tej opcji spowoduje wydłużenie czasu potrzebnego do wylogowania się, wprost proporcjonalnie do objętości pliku wymiany (*pagefile*)

Podklucze klucza

`[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\SessionManager]`

zawierają wiele interesujących opcji, m.in. sterowanie stronicowaniem, zmiennymi środowiskowymi itp.

- ***Automatyczne logowanie się do systemu po restarcie.***

Dotyczy: WIN 2000/XP

Opis: Podanie nazwy użytkownika oraz hasła, które ma być używane do automatycznego logowania do systemu po restarcie.

Klucz: `[HKEY_LOCAL_MACHINE\\Software\Microsoft\Windows NT\Current Version\Winlogon]`

Nazwa wartości (jeśli któraś nie istnieje, to należy ją dodać)	Typ danych	Nowa dana
DefaultDomainName	REG_SZ	nazwa domeny sieciowej Windows, do której chcemy się autologować, albo nazwa komputera w sieci.
DefaultUserName	REG_SZ	login użytkownika do autologowania
AutoAdminLogon	REG_SZ	1 – włączenie autologowania
DefaultPassword	REG_SZ	hasło użytkownika do autologowania

Uwaga: Powyższego wpisu nie powinno się stosować w komputerach podłączonych do jakiejkolwiek sieci oraz w tych, do których mogą mieć dostęp niepowołane osoby. Należy uświadomić sobie fakt, iż aby wejść na cudze konto, siedząc przy komputerze z tak ustawionym rejestrem, wystarczy uruchomić ponownie komputer.

- ***Automatyczne zamykanie zadań podczas wylogowywania się***

Dotyczy: WIN 2000/XP

Opis: Nie wszystkie procesy są automatycznie zabijane przy wylogowywaniu, część sprawia, że pojawia się denerwujące okno, w którym znajduje się pasek postępu zamykania oraz przyciski „zakończ" i „anuluj". Aby zlikwidować te okienka i wymusić całkowite niszczenie procesów podczas wylogowywania się należy ustawić poniższą wartość w rejestrze.

Klucz: `[HKEY_USERS\<SID>\Control Panel\Desktop]`

Nazwa wartości: AutoEndTasks

Możliwe wartości: 0-wyłączone, 1-włączone

Uwaga: <SID> oznacza Security Identifier. Jest to specjalny numer użytkownika w systemie. Można go uzyskać na kilka sposobów, np. używając funkcji API: LookupAccountSid. Umożliwia ona wyznaczenie SID na podstawie nazwy użytkownika.

- ***„Utrudnienie" uruchamiania programów***

Dotyczy: WIN XP

Oznaczenia: „@" – oznacza nazwę wartości „Domyślny" (Default). Wartość o tej nazwie znajduje się w większości kluczy.

Opis: W przedstawionym zestawieniu wypisane są klucze odpowiedzialne za możliwość normalnego uruchamiania programów. W zestawieniu oprócz nazw kluczy, podano, jakie wartości tych kluczy powinny mieć, w poprawnie funkcjonującym komputerze, jaką zawartość.

Klucz	[HKEY_CLASSES_ROOT\.exe]
Nazwa wartości	@
Typ danej	REG_SZ
Powinno być	exefile

Klucz	[HKEY_CLASSES_ROOT\.exe]
Nazwa wartości	Content Type
Typ danej	REG_SZ
Powinno być	application/x-msdownload

Klucz	[HKEY_CLASSES_ROOT\.exe\PersistentHandler]
Nazwa wartości	Content Type
Typ danej	REG_SZ
Powinno być	{098f2470-bae0-11cd-b579-08002b30bfeb}

Klucz	[HKEY_CLASSES_ROOT\exefile]
Nazwa wartości	@
Typ danej	REG_SZ
Powinno być	Aplikacja (w ang. wersji Application)

Nazwa wartości	EditFlags
Typ danej	REG_BINARY
Powinno być	hexalnie 38 07 00 00

Nazwa wartości	InfoTip
Typ danej	REG_SZ
Powinno być	prop:FileDescription;Company;FileVersion;Create;Size

Nazwa wartości	TileInfo
Typ danej	REG_SZ
Powinno być	prop:FileDescription;Company;FileVersion

Klucz	[HKEY_CLASSES_ROOT\exefile\DefaultIcon]
Nazwa wartości	@
Typ danej	REG_SZ
Powinno być	%1

Klucz	[HKEY_CLASSES_ROOT\exefile\shell\open]
Nazwa wartości	EditFlags
Typ danej	REG_BINARY
Powinno być	hexalnie 00 00 00 00

Klucz	[HKEY_CLASSES_ROOT\exefile\shell\open\command]
Nazwa wartości	@
Typ danej	REG_SZ
Powinno być	"%1"[spacja]%* - uwaga nie może tu być ŻADNYCH dodatk. znaków, [spacja] oznacza POJEDYNCZY ODSTĘP

Klucz	[HKEY_CLASSES_ROOT\exefile\shell\runas\command]
Nazwa wartości	@
Typ danej	REG_SZ
Powinno być	"%1"[spacja]%* - uwaga nie może tu być ŻADNYCH dodatk. znaków, [spacja] oznacza POJEDYNCZY ODSTĘP

Klucz	[HKEY_CLASSES_ROOT\exefile\shellex\PropertySheetHandlers\ PifProps]
Nazwa wartości	@
Typ danej	REG_SZ
Powinno być	{86F19A00-42A0-1069-A2E9-08002B30309D}

Klucz	[HKEY_CLASSES_ROOT\exefile\shellex\PropertySheetHandlers\ ShimLayer Property Page]
Nazwa wartości	@
Typ danej	REG_SZ
Powinno być	{513D916F-2A8E-4F51-AEAB-0CBC76FB1AF8}

Przykładowo, zmiana wartości w kluczach:

Klucz	[HKEY_CLASSES_ROOT\exefile\shell\open\command

i

Klucz	[HKEY_CLASSES_ROOT\exefile\shell\runas\command

na np. „%2” spowoduje że w normalny sposób, nie będzie można uruchamiać programów na komputerze.

Tę metodę stosują niektóre wirusy (np. z rodziny Worm.Roron). Przedstawiona „ciekawostka” pokazuje, jak wielkie szkody mogą wypływać z niepowołanego dostępu do rejestru.

VII. Opis ważniejszych kluczy.

Niżej przedstawiono część ważniejszych kluczy znajdujących się w rejestrze systemowym. Występują one na każdym komputerze z zainstalowanym systemem Windows z serii NT, niezależnie od tego, jakie programy instalowaliśmy.

- **[HKEY_LOCAL_MACHINE\HARDWARE]**

Klucz przechowuje informacje o bieżącej konfiguracji komputera. Jest on tworzony zawsze od początku przy każdym starcie komputera, cała jego zawartość jest przechowywana tylko w pamięci RAM, nie na dysku twardym (*volatile*). Ponadto, większość wartości w tym kluczu jest typu REG_BINARY, co sprawia, że są to wartości trudne do edycji. Cały klucz należy traktować raczej jako tylko do odczytu, pamiętając, że ewentualne wprowadzone ręcznie zmiany zostaną zniesione wraz z kolejnym uruchomieniem komputera.

- **[HKEY_LOCAL_MACHINE\HARDWARE\DESCRIPTION]**

Zawiera informacje, jaki sprzęt jest zainstalowany w komputerze, także w sytuacjach błędów inicjalizacji urządzeń np. znaleziono kontroler SCSI, ale nie udało się go zainicjalizować. Informacje o kartach sieciowych, kartach PCMCIA, drukarkach i innych zewnętrznych urządzeniach znajdują się w innym miejscu.

Każdy kontroler szyny (ISA, PCI, itp.) ma swój własny klucz pod MultifunctionAdapter, w każdym z podkluczy znajduje się informacja o urządzeniach podłączonych do szyny.

Na przykład na moim komputerze, klucz:

```
[HKLM\HARDWARE\DESCRIPTION\System\MultifunctionAdapter\6\DiskController\0]
```

zawiera m.in. część informacji o stacji dyskietek.

Uwaga: struktura kluczy poniżej MultifunctionAdapter nie jest identyczna na wszystkich komputerach i zależy od kolejności znajdowania szyn przez system operacyjny.

- **[HKEY_LOCAL_MACHINE\SOFTWARE]**

W tym kluczu przechowywana jest większość danych konfiguracyjnych programów zainstalowanych w systemie. Przykładowo, w kluczu:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Internet Explorer]
```

przechowywane są ustawienia programu MS Internet Explorer.

UWAGA: w celu przechowywania informacji spersonalizowanych dla różnych użytkowników, należy używać klucza HKEY_CURRENT_USER

- **[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion]**

W systemach z serii NT (NT, 2000, XP) klucz zawiera dane systemu operacyjnego, m.in. o fontach, driver'ach, językach itp.

- **[HKEY_LOCAL_MACHINE\SYSTEM]**

Tu przechowywane są najważniejsze informacje systemowe. Korzysta z nich jądro systemu oraz procesy systemowe. Przechowywane dane określają m.in,.

- działanie systemu po wykryciu niedozwolonej operacji,
- rejestrację przyczyn zawieszenia się systemu,
- kontrolę operacji systemowych,
- konfigurację dot. systemów plików, drukowania, sesji użytkownika, zarządzania pamięcią, usług systemowych, serwerów usług (NETBIOS, DHCP), protokołów, itp.

- **[HKEY_CLASSES_ROOT]**

Zawiera wszystkie „zarejestrowane" w systemie rozszerzenia plików. Przechowywane są m.in. infomracje o programach skojarzonych z rozszerzeniami oraz identyfikatory klas (CLSID).

- **[HKEY_CURRENT_USER]**

Jak wspomniano w rozdziale „Budowa" klucz ten jest w istocie aliasem do odpowiedniego podklucza w HKEY_USERS. Umożliwia on łatwe zarządzanie ustawieniami środowiska pracy w kontekście użytkownika. Klucz będzie zawierał, gdy aktualnie zalogowanym użytkownikiem jest ktoś o prawach dostępu administratora, a co innego, gdy osoba mniej uprzywilejowana.

Np. podklucz

```
[HKEY_CURRENT_USER\Printers]
```

zawiera informacje o tym, z jakich drukarek może korzystać bieżący użytkownik.

A. Załącznik

W skład Registry API wchodzą następujące funkcje:

RegCloseKey
RegConnectRegistry
RegCreateKeyEx
RegDeleteKey
RegDeleteValue
RegDisablePredefinedCache
RegEnumKeyEx
RegEnumValue
RegFlushKey
RegGetKeySecurity
RegLoadKey
RegNotifyChangeKeyValue
RegOpenCurrentUser
RegOpenKeyEx
RegOpenUserClassesRoot
RegOverridePredefKey
RegQueryInfoKey
RegQueryMultipleValues
RegQueryValueEx
RegReplaceKey
RegRestoreKey
RegSaveKey
RegSetKeySecurity
RegSetValueEx
RegUnLoadKey

Następujące funkcje wchodzące w skład Shell API mogą być używane do obsługi rejestru:

AssocCreate
AssocQueryKey
AssocQueryString
AssocQueryStringByKey
SHCopyKey
SHDeleteEmptyKey
SHDeleteKey
SHDeleteValue
SHEnumKeyEx
SHEnumValue
SHGetValue
SHQueryInfoKey
SHQueryValueEx
SHRegCloseUSKey
SHRegCreateUSKey
SHRegDeleteEmptyUSKey
SHRegDeleteUSValue
SHRegDuplicateHKey
SHRegEnumUSKey
SHRegEnumUSValue

SHRegGetBoolUSValue
SHRegGetIntW
SHRegGetPath
SHRegGetUSValue
SHRegOpenUSKey
SHRegQueryInfoUSKey
SHRegQueryUSValue
SHRegSetPath
SHRegSetUSValue
SHRegWriteUSValue
SHSetValue

B. Źródła

- ***Książki i opracowania***

Jerry Honeycutt „Rejestry Windows 95/NT. Czarna księga”, Helion 1998

David A. Solomon, Mark E. Russinovich „MS Windows 2000 od środka”, Helion 2003

MSDN Library

- ***Linki***

http://msdn.microsoft.com
Wersja online MSDN Library wraz z pokaźną bazą artykułów.

http://www.winguides.com/registry
Ulepszanie Windows przez grzebanie w rejestrze.

http://www.jsiinc.com
Zbiór porad do Windows, nie tylko o rejestrze.

http://virusall.com/downregistry.html
Kilka zmian najczęściej dokonywanych przez wirusy.

http://www.microsoft.com/technet/prodtechnol/windows2000serv/tips/regtweak.mspx
Top 10 zmian w rejestrze wg wieloletniego pracownika Microsoftu.

http://www.winxpcentral.com/registry/tweaks.php
Kolejna porcja "usprawnień".

http://www.clankiller.com/tech/reg1.php
Kilka prostych zmian w rejestrze z opisem wykonania i screen'ami z Regedit'a.

http://www.auldfart.co.uk/tweaksarchive.html
Duża porcja tricków.

http://www.oreilly.com/catalog/mwin2reg/index.html
Oficjalna strona książki „Managing the Windows 2000 Registry”, istnieje możliwość ściągnięcia przykładowego rozdziału oraz kodu źródłowego dołączanego do książki.

http://www.kellys-korner-xp.com/xp_tweaks.htm
Ponad 320 zmian w rejestrze w postaci plików „.reg”.