Registrul în WIN32 API

CURS NR. 3/1

Windows Registry

Registrul este o bază de date ierarhică definită de sistem

- Stochează informații de configurare (setări, opțiuni) pentru
  - sistemul de operare Windows
  - serviciile și aplicațiile instalate
  - driver-ele de dispozitive hw
  - Preferințele utilizator
- Permite accesul la contoare de performanță pentru evaluarea performanței sistemului și a aplicațiilor

- Permite păstrarea setărilor într-o singură structură logică – într-o formă standardizată
  - Este o bază de date → furnizează actualizare atomică (tranzacțională)

- Exemplu:
  - La instalarea unui nou program, se adaugă o nouă sub-cheie în registru
    - Subcheia conține date despre aplicație (locația programului, modul de pornire, versiunea, etc.)

- Nu este obligatoriu ca aplicațiile instalate să își păstreze informațiile de configurare în Registrul Windows
  - .Net Framework folosește fișiere xml
  - Aplicațiile portabile de regulă își mențin fișierele de configurare împreună cu executabilul

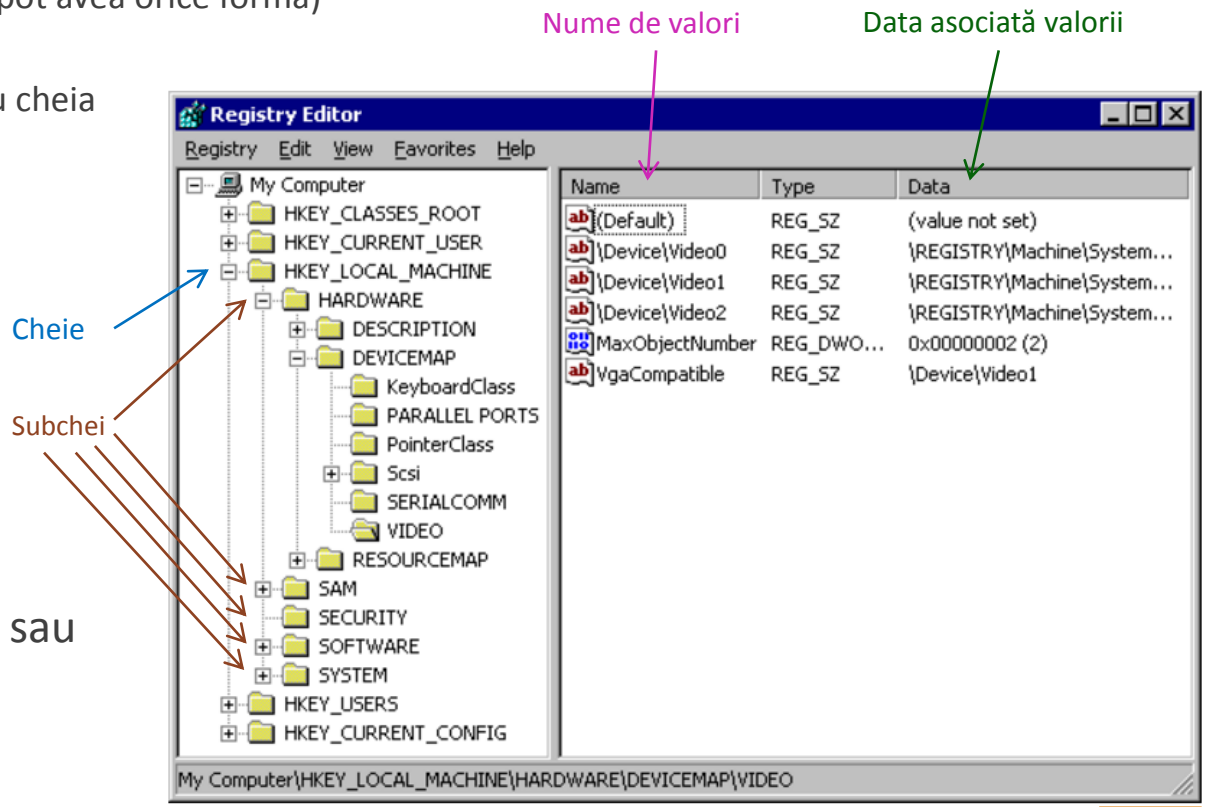
Structura registrului

Structură ierarhică (sub formă de arbore)

Două concepte centrale: **key** și **value**

- O **cheie** este un obiect container – similar cu conceptu de director (folder)
- **Valorile** sunt perechi Nume – Data și similare conceptului de fișier
- O cheie poate conține orice număr de subchei și valori (valorile pot avea orice formă)
  - Unele aplicații necesită doar existența cheii
  - Alte aplicații accesează cheia și folosesc valorile asociate cu cheia
- Fiecare cheie are un nume afișabil (nu este case-sensitive) și nu poate include caracterul backslash (\)
  - Numele cheii este unic considerând cheile de deasupra din ierarhie
- Fiecare cheie trebuie să aibă o valoare implicită (default)
- Fiecare valoare are un nume afișabil și poate avea date asociate
- Registrul poate fi vizualizat și modificat prin RegEdit sau din aplicație folosind Registry API

Perechea Cheie / Valoare este analogică cu conceptele de nume de director și fișier din SF



Structura registrului

Exemplu

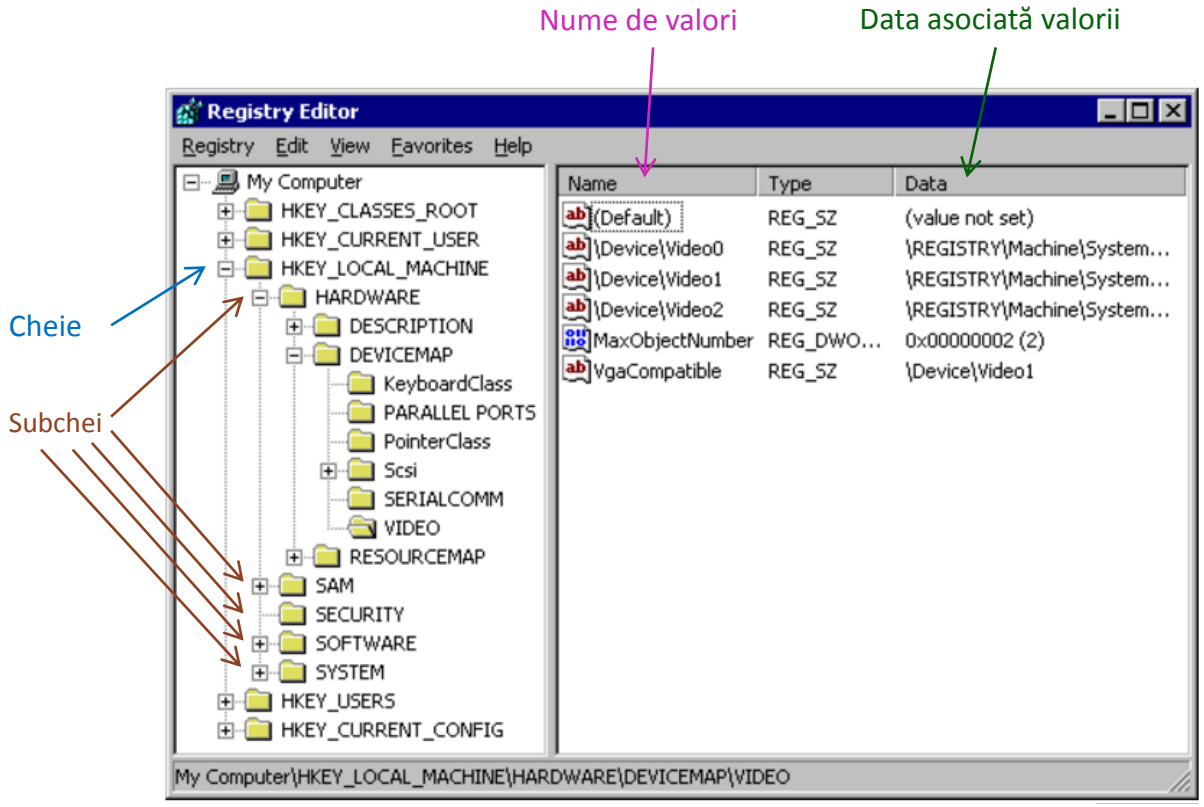
- Cheia **HKEY_LOCAL_MACHINE** are subcheile:
  - **HARDWARE**
  - **SAM**
  - **SECURITY**
  - **SOFTWARE**
  - **SYSTEM**

  - Cheia **HARDWARE** are subcheile:
    - **DESCRIPTION**
    - **DEVICEMAP**
    - **RESOURCEMAP**

    - Cheia **DEVICEMAP** are subcheile:
      - **KeyboardClass**
      - **PARALLEL PORTS**
      - **PointerClass**
      - **Scsi**
      - **SERIALCOMM**
      - **VIDEO**

      - Cheia **VIDEO** are o serie de valori
        - Fiecare valoare are nume și poate avea date asociate

Perechea Cheie / Valoare este analogică cu conceptele de nume de director și fișier din SF



Registrul

În mod analogic cu căile de acces la fișiere, registrul se accesează prin nume de chei și valori

- Accesul la registru se poate realiza doar pornind de la un handle la o cheie predefnită (handle numeric constant)

- Exemple de chei predefinite din registru
  - **HKEY_LOCAL_MACHINE** (sau HKLM) – informații despre starea fizică a sistemului – tipul de magistrală, memoria sistemului, hardware-ul și software-ul instalat
  - **HKEY_USERS** (sau HKU) - Informații de configurație pentru utilizatorul curent și pentru noi utilizatori ai sistemului
  - **HKEY_CURRENT_CONFIG** (sau HKCC)– setări actuale – rezoluție, fonturi, etc.
  - **HKEY_CLASSES_ROOT** (sau HKCR) - mapările dintre extensii și programe executabile
  - **HKEY_CURRENT_USER** (sau HKCU)– informații specifice utilizatorului curent – variabile de mediu, preferințele de aplicații, etc.
  - **HKEY_PERFORMANCE_DATA** – permite accesul la date de performanță – funcțiile registrului pot colecta aceste date (nu este vizibil în RegEdit)

Roiuri de registru (registry hives)

- Un roi este o grupare logică de chei, subchei și valori din registru care au un set de fișiere suport conținând backup (copii de siguranță)
  - Ex. Când un nou utilizator se loghează pe sistem, se creează un nou roi, este creat un fișier asociat pentru stocarea informațiilor de profil al utilizatorului – numit roiul de profil al utilizatorului
    - Roiurile utilizatorilor sunt localizate în registru la cheia HKEY_USERS

Roiuri de registru

- Cele mai multe fișiere suport pentru roiuri sunt stocate în directorul %SystemRoot%\System32\Config
  - Aceste fișiere sunt actualizate de fiecare dată când un utilizator se loghează

Extension	Description
none	A complete copy of the hive data.
.alt	A backup copy of the critical **HKEY_LOCAL_MACHINE\System** hive. Only the System key has an .alt file.
.log	A transaction log of changes to the keys and value entries in the hive.
.sav	A backup copy of a hive.

- Extensiile acestor fișiere / sau lipsa extensiilor indică tipul datelor pe care le conțin

Roiuri standard și fișierele suport:

Registry hive	Supporting files
HKEY_CURRENT_CONFIG	System, System.alt, System.log, System.sav
HKEY_CURRENT_USER	Ntuser.dat, Ntuser.dat.log
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_USERS\.DEFAULT	Default, Default.log, Default.sav

Registrul

Recomandări privind tipul și dimensiunea datelor care pot fi stocate în registru

- Doar date de configurare și inițializare
- Date mai mici de 1 – 2 KB (date mai mari se pot stoca în fișier și referite în registru printr-o cheie)
- Nu se stochează cod executabil

- O valoare ocupă spațiu considerabil mai mic decât o cheie
  - → datele similare să fie grupate împreună într-o structură și stocate ca o valoare
    - Decât stocarea fiecărei componente ca și cheie separată

Categorii de date

- O aplicație care dorește să stocheze date în registru ar trebui să clasifice datele în două categorii
  - Date specifice sistemului de calcul
    - Se înregistrează sub cheia **HKEY_LOCAL_MACHINE** și se creează chei pentru numele de companie, produs, versiune, etc.
    - Exemplu: **HKEY_LOCAL_MACHINE\Software***MyCompany**MyProduct**1.0*
  - Date specifice utilizatorului
    - Se înregistrează sub cheia **HKEY_CURRENT_USER**
    - Exemplu: **HKEY_CURRENT_USER\Software***MyCompany**MyProduct**1.0*

- Avantaj: aplicația poate oferi suport pentru crearea unor profiluri de utilizator independente de sistem

Registrul

Deschiderea unei subchei

```
LONG WINAPI RegOpenKeyEx(
          HKEY    hKey,
          LPCTSTR lpSubKey,
          DWORD   ulOptions,
          REGSAM  samDesired,
          PHKEY   phkResult );
```

Handle-ul cheii deschise – returnat de funcția **RegCreateKeyEx** sau **RegOpenKeyEx** sau una din cheile predefinite: **HKEY_LOCAL_MACHINE**, **HKEY_CLASSES_ROOT**, **HKEY_USERS**, sau **HKEY_CURRENT_USER**

Numele subcheii pe care dorim să o deschidem. Dacă este NULL se returnează în ultimul parametru un handle duplicat pentru hKey

Parametru rezervat – tebuie să fie 0

Dreptrile de acces – de ex. KEY_ALL_ACCESS , KEY_WRITE, KEY_QUERY_VALUE, KEY_ENUMERATE_SUB_KEYS, KEY_SET_VALUE, KEY_NOTIFY , etc.

Primește handle-ul cheii nou deschise

- Returnează ERROR_SUCCESS în caz de succes

Crearea unei noi chei

- Cheia nouă se creează ca subcheie a cheii curente (adâncimea maximă a arborelui este de 512 niveluri)
  - Următoarele chei sunt întotdeauna deschise: **HKEY_LOCAL_MACHINE**, **HKEY_CLASSES_ROOT**, **HKEY_USERS**, și **HKEY_CURRENT_USER**

```
LONG WINAPI RegCreateKeyEx(
          HKEY                  hKey,
          LPCTSTR               lpSubKey,
          DWORD                 Reserved,
          LPTSTR                lpClass,
          DWORD                 dwOptions,
          REGSAM                samDesired,
          LPSECURITY_ATTRIBUTES lpSecurityAttributes,
          PHKEY                 phkResult,
          LPDWORD               lpdwDisposition );
```

Handle-ul cheii deschise – returnat de funcția **RegCreateKeyEx** sau **RegOpenKeyEx** sau una din cheile predefinite. Trebuie să aibă dreptul KEY_CREATE_SUB_KEY

Numele subcheii pe care dorim să o creem sub cheia cu handle-ul hKey

NULL

De reuglă este 0 - **REG_OPTION_NON_VOLATILE**

Dreptrile de acces

Atributele de securitate – NULL pentru atribute implicite

Primește handle-ul cheii nou deschise

Arată dacă cheia a existat deja sau a fost nou creată

Registrul

Enumerarea subcheilor

```
LONG WINAPI RegEnumKeyEx(
          HKEY      hKey,
          DWORD     dwIndex,
          LPTSTR    lpName,
          LPDWORD   lpcName,
          LPDWORD   lpReserved,
          LPTSTR    lpClass,
          LPDWORD   lpcClass,
          PFILETIME lpftLastWriteTime );
```

Handle-ul cheii deschise

Trebuie să fie 0 la primul apel, și apoi se incrementează la fiecare apel succesiv

Numele subcheii (doar numele, nu întreaga ierarhie – calea întreagă)
Și dimensiunea bufferului în care se pune numele

NULL

Clasa definită de utilizator a subcheii returnate
și dimensiunea bufferului în care se pune clasa

Timpul ultimei modificări a subcheii returnate

- Returnează în parametrii lpName și lpcName numele și dimeniunea subcheii (și clasa acesteia)
- Returnează ERROR_SUCCESS în caz de succes

Închiderea cheii

```
LONG WINAPI RegCloseKey( HKEY hKey );
```

Handle-ul cheii deschise

- Returnează ERROR_SUCCESS în caz de succes

Ștergerea cheii

```
LONG WINAPI RegDeleteKey(
          HKEY    hKey,
          LPCTSTR lpSubKey );
```

Handle-ul cheii deschise

Numele subcheii pe care dorim să o ștergem

- Returnează ERROR_SUCCESS în caz de succes

Registrul – gesitunea valorilor și datelor

Enumerarea valorilor

```
LONG WINAPI RegEnumValue(
          HKEY    hKey,
          DWORD   dwIndex,
          LPTSTR  lpValueName,
          LPDWORD lpcchValueName,
          LPDWORD lpReserved,
          LPDWORD lpType,
          LPBYTE  lpData,
          LPDWORD lpcbData );
```

Handle-ul cheii deschise

Trebuie să fie 0 la primul apel, și apoi se incrementează la fiecare apel succesiv

Numele valorii
și dimensiunea bufferului în care se pune numele

NULL

Tipul datei stocate în valoare

Data asociată valorii
și dimensiunea datei

- Returnează ERROR_SUCCESS în caz de succes. Dacă bufferul pentru data este prea mic returnează ERROR_MORE_DATA

Setarea valorilor

```
LONG WINAPI RegSetValue(
          HKEY    hKey,
          LPCTSTR lpSubKey,
          DWORD   dwType,
          LPCTSTR lpData,
          DWORD   cbData );
```

Handle-ul cheii deschise, cu dreptul KEY_SET_VALUE

Numele subcheii – dacă nu există funcția va crea subcheia
Dacă este NULL – funcția setează valoarea implicită

Tipul informației care se stochează. Trebuie să fie REG_SZ

Data care se va stoca – nu poate fi NULL

Dimensiunea datei – este ignorată de funcție, se calculează pe baza parametrului lpData

- Returnează ERROR_SUCCESS în caz de succes

Exemplu: enumerarea valorilor și datelor din cheia dată

```
// QueryKey - Enumerates the subkeys of key and its associated values.
//     hKey - Key whose subkeys and values are to be enumerated.

#include <windows.h>
#include <stdio.h>
#include <tchar.h>

#define MAX_KEY_LENGTH 255
#define MAX_VALUE_NAME 16383

void QueryKey(HKEY hKey);

void _tmain(void) {
   HKEY hTestKey;

   if (RegOpenKeyEx( HKEY_CURRENT_USER,
                     TEXT("SOFTWARE\\Microsoft"),
                     0,
                     KEY_READ,
                     &hTestKey) == ERROR_SUCCESS) {
                QueryKey(hTestKey);
   }
   RegCloseKey(hTestKey);
}

void QueryKey(HKEY hKey) {
   TCHAR    achKey[MAX_KEY_LENGTH];   // buffer for subkey name
   DWORD    cbName;                   // size of name string
   TCHAR    achClass[MAX_PATH] = TEXT("");  // buffer for class name
   DWORD    cchClassName = MAX_PATH;  // size of class string
   DWORD    cSubKeys = 0;               // number of subkeys
   DWORD    cbMaxSubKey;              // longest subkey size
   DWORD    cchMaxClass;              // longest class string
   DWORD    cValues;              // number of values for key
   DWORD    cchMaxValue;          // longest value name
   DWORD    cbMaxValueData;       // longest value data
   DWORD    cbSecurityDescriptor; // size of security descriptor
   FILETIME ftLastWriteTime;      // last write time

   DWORD i, retCode;
   TCHAR  achValue[MAX_VALUE_NAME];
   DWORD cchValue = MAX_VALUE_NAME;
```

```
   // Get the class name and the value count.
   retCode = RegQueryInfoKey(    hKey,                    // key handle
                                 achClass,                // buffer for class name
                                 &cchClassName,           // size of class string
                                 NULL,                    // reserved
                                 &cSubKeys,               // number of subkeys
                                 &cbMaxSubKey,            // longest subkey size
                                 &cchMaxClass,            // longest class string
                                 &cValues,                // number of values for this key
                                 &cchMaxValue,            // longest value name
                                 &cbMaxValueData,         // longest value data
                                 &cbSecurityDescriptor,   // security descriptor
                                 &ftLastWriteTime);       // last write time

   // Enumerate the subkeys, until RegEnumKeyEx fails.
   if (cSubKeys) {
      printf("\nNumber of subkeys: %d\n", cSubKeys);

      for (i = 0; i<cSubKeys; i++) {
         cbName = MAX_KEY_LENGTH;
         retCode = RegEnumKeyEx( hKey, i, achKey, &cbName,
                                 NULL, NULL, NULL, &ftLastWriteTime);
         if (retCode == ERROR_SUCCESS) {
            _tprintf(TEXT("(%d) %s\n"), i + 1, achKey);
         }
      }
   }

   // Enumerate the key values.
   if (cValues) {
      printf("\nNumber of values: %d\n", cValues);

      for (i = 0, retCode = ERROR_SUCCESS; i<cValues; i++) {
         cchValue = MAX_VALUE_NAME;
         achValue[0] = '\0';
         retCode = RegEnumValue( hKey, i, achValue, &cchValue,
                                 NULL, NULL, NULL, NULL);

         if (retCode == ERROR_SUCCESS) {
            _tprintf(TEXT("(%d) %s\n"), i + 1, achValue);
         }
      }
   }
}
```

Materiale de studiu

Resurse online: http://www.techsupportalert.com/content/what-everybody-should-know-about-windows-registry.htm

Funcții pentru lucrul cu registrul: https://msdn.microsoft.com/en-us/library/ms724875.aspx

Informații despre Windows Registry: https://support.microsoft.com/en-us/help/256986/windows-registry-information-for-advanced-users