Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Windows Registry

Seminarska naloga pri predmetu Sistemska programska oprema

Žan Kogovšek

Ljubljana, 2022

Uvod

Marsikateri uporabnik Windows operacijskega sistema je že slišal za Windows Registry oz. register za Windows. Bodisi gre za naprednejše uporabnike, ki so bolj spoznani z operacijskim sistemom Windows, bodisi so preprosto naleteli na sporočilo z napako, ki je omenjalo register za Windows. Toda le redki uporabniki vedo, čemu je namenjen in ga znajo uporabljati, kar tudi ni presenetljivo. Na uradni spletni strani proizvajalca Microsoft namreč piše, naj se podatkov v registru Windows ne spreminja, razen ko je to absolutno potrebno in ni drugega načina za odpravo napak. Že manjša napaka v registru lahko vodi k nestabilnemu sistemu ali še slabše, k popolni okvari. V takšnem primeru nam preostane le, da poskusimo obnoviti register v stanje, ko je bil ta še stabilen. Če varnostne kopije predhodno nismo ustvarili, smo prisiljeni v ponovno namestitev operacijskega sistema.

Te teme so za običajnega uporabnika precej napredne in za vsakdanjo rabo nepraktične, toda za bolj radovedne je gotovo zanimivo, kakšne podatke shranjujemo v registrih in zakaj gre lahko toliko stvari narobe že ob majhnih spremembah.

Opis

Windows Registry je hierarhična podatkovna baza, ki shranjuje nizko nivojske nastavitve za Windows operacijski sistem in aplikacije, ki uporabljajo register za Windows. Jedro operacijskega sistema, gonilniki za naprave, storitve in uporabniški vmesniki so le eni izmed možnih uporabnikov registra. Z drugimi besedami, register za Windows vsebuje informacije, nastavitve, možnosti in druge vrednosti za programe in strojno opremo na vseh različicah sistema Windows. Kot primer vzemimo namestitev programa, ki uporablja Windows Registry. Zanj je v podatkovno bazo dodan nov ključ s podatki o lokaciji programa, njegovi verziji in kako program zaženemo. Enako velja za spreminjanje nastavitev programa ali recimo nekaj preprostega, kot so nastavitve zaslona. Vsi ti podatki so najverjetneje shranjeni kar znotraj registra za Windows.

Register za Windows se je prvič pojavil z Windowsi 3.1 leta 1992. Na začetku se je uporabljal bolj za shranjevanje konfiguracijskih informacij o COM komponentah ali vzorcih predmetne sestavine. Toda že v naslednji različici sistema Windows, Windows 95, se je uporaba razširila tudi na .INI datoteke. Te so shranjevale nastavitve za posamezne programe, saj so bile raztresene po celem disku in jih je bilo težko nadzorovati. To pa še ne pomeni, da mora vsak program obvezno uporabljati Windows Registry. Ravno nasprotno, za prenosljive programe je standardna praksa, da so konfiguracijske datoteke zraven izvršljivih datotek, saj bi v nasprotnem primeru izgubili "prenosljivost".

```
; last modified 1 April 2001 by John Doe
[owner]
name = John Doe
organization = Acme Widgets Inc.

[database]
; use IP address in case network name resolution is not working
server = 192.0.2.62
port = 143
file = "payroll.dat"
```

Zgoraj je primer .INI datoteke, ki je razdeljena na dva razdelka, z lastnostmi nastopajočimi v parih ključ-vrednost.

Zgodovina

Pred pojavom registra za Windows so .INI datoteke shranjevale nastavitve vsakega programa v tekstovni ali binarni datoteki. Pogosto so se te nahajale v skupnem prostoru, dostopnem vsakemu uporabniku in niso podpirale večuporabniške izkušnje. To je tudi eden izmed glavnih razlogov za premik na Windows Registry, ki shranjuje vse nastavitve aplikacije v enem centralnem sistemu in v standardiziranem formatu. Windows Registry ni nič drugega kot skupek binarnih datotek, kar tudi predstavlja prednost v primerjavi z .INI datoteko, saj je branje in pisanje mnogo hitrejše in učinkovitejše. Prav tako uvede strogo tipiziranje in s tem olajša popravljanje in urejanje ključev, saj točno vemo zalogo vrednosti, medtem ko pri tekstovnih .INI datotekah te ne poznamo. Dostop do registra za Windows je omogočen vsem uporabnikom sistema Windows, pri čemer velja, da se v register naložijo vrednosti trenutnega uporabnika. To pomeni, da lahko vsak uporabnik poljubno spreminja svoje nastavitve in s tem ne povzroča sprememb drugim.

Velik problem .INI datotek je bil tudi *race condition* ali tvegano stanje, do katerega pride, ko želita dva procesa posodobiti isto vrednost. Pri tem lahko pride do nekonsistentnosti podatkov in rezultat ne odraža želenega stanja nobenega izmed procesov. Prednost registra za Windows je v tem, da je pravzaprav baza in ponuja izboljšano integriteto podatkov z naprednimi koncepti, kot je atomarnost. Če zdaj želita dva procesa posodobiti isto vrednost, se bo en proces zgodil pred drugim, torej bomo imeli primer izgubljenega ažuriranja, ampak konsistentnost podatkov bo ohranjena in pripadala enemu izmed procesov.

Struktura

Register sestavljata dva osnovna elementa, ključ in vrednost. Prvi predstavlja zabojniški objekt podoben mapam v operacijskem sistemu, medtem ko so vrednosti bolj podobne datotekam. Ključ referenciramo v sintaksi, ki je uporabljena tudi za predstavitev poti v operacijskem sistemu, kjer z uporabo leve poševnice predstavimo nivo hierarhije. Od tu tudi omejitev za samo ime ključa, saj ta ne more vsebovati leve poševnice in je neodvisna od velikosti črk.

Korenski ključi

Do hierarhije dostopamo preko korenskih ključev, ki lahko vodijo do registrskih ključev, ki jih naloži jedro operacijskega sistema iz tako imenovanega "hive-a"oz. panja. Ta je sestavljen iz več datotek na disku, poleg tega pa določene informacije shrani v spomin ob zagonu sistema. Pri tem velja, da niso vsi panji naloženi naenkrat, ampak se nalagajo po potrebi. Poleg teh lahko vodijo tudi do vrednosti podključa ali registrirane storitve. Poimenovani so po Windows API definicijah, ki se vse pričnejo s "HKEY", ampak v skrajšani obliki "HK". Windows definira sedem korenskih ključev:

- HKEY_LOCAL_MACHINE ali HKLM

  Shranjuje nastavitve, ki so specifične za računalnik. Lokacija ključa ni na dejanskem disku, ampak ga jedro operacijskega sistema shrani v pomnilnik, od koder povezuje vse podključe. Aplikacije znotraj tega korenskega ključa ne morejo ustvariti podključev. Med bolj znanimi podključi je **SAM**, ki shranjuje vse podatkovne baze za Security Account Managerje. Ti shranjujejo uporabniška gesla in se lahko uporabljajo za avtentikacijo lokalnih in oddaljenih uporabnikov. Naslednji podključ je **SECURITY**, ki je povezan na podatkovno bazo za varnost v domeni, ki ji pripada trenutno vpisan uporabnik. Jedro operacijskega sistema s pomočjo te baze bere in izvaja varnostno politiko za uporabnika in vse njegove aplikacije. Sledi podključ **SYSTEM**, v katerega lahko piše le uporabnik z administratorskimi pravicami. Vsebuje podatke o namestitvi operacijskega sistema Windows, generatorju naključnih števil, trenutno povezanih napravah z datotečnim sistemom in različne konfiguracije za gonilnike naprav in storitev, ki se izvajajo na računalniku. Še zadnji pomembnejši podključ je **SOFTWARE**. Tu so shranjeni ključi aplikacij na sistemu in nastavitve operacijskega sistema. Največkrat do njega dostopajo prav aplikacije in namestitveni čarovniki. Podključi so organizirani po imenu proizvajalca aplikacije, ki je podključ ustvarila. Med podključi imamo na 64-bitnem Windowsu tudi **Wow6432Node**, kjer so shranjene 32-bitne aplikacije.

- HKEY_CURRENT_CONFIG ali HKCC

  Shranjuje podatke o trenutni strojni opremi na računalniku in njihove nastavitve. V resnici sam ne shranjuje podatkov, temveč ima povezavo na ključ HKLM in podključ Hardware, s pomočjo katere lahko hitro dostopamo do željenih podatkov o strojni opremi.

- HKEY_CLASSES_ROOT ali HKCR

  Vsebuje informacije o priponah datotek in raznih identifikatorjih. S pomočjo teh informacij operacijski sistem ve, kaj storiti, ko odpremo datoteko nekega tipa.

- HKEY_CURRENT_USER ali HKCU

  Nastavitve trenutnega uporabnika se nahajajo v tem korenskem ključu. Je referenca na podključ znotraj HKEY_USERS, ki ustreza uporabniku.

- HKEY_USERS ali HKU

  Vsebuje podključe za vsakega obstoječega uporabnika naprave, pri čemer se podatki prenesejo iz panja prenesejo le za trenutno vpisanega uporabnika.

- HKEY_PERFORMANCE_DATA (samo znotraj operacijskega sistema Windows NT)

  Ponujal je podatke o zmogljivosti, ki jih je pridobil od jedra operacijskega sistema, gonilnikov, programov ali storitev, v kolikor so te podatke beležili.

- HKEY_DYN_DATA (samo znotraj operacijskih sistemov Windows 95, 96, 98 in ME)

  Vseboval je informacije o strojni opremi in mrežni zmogljivosti. Podatki niso bili shranjeni na disku, temveč pridobljeni ob zagonu sistema in shranjeni v spominu.

Tipi vrednosti

Vrednosti Windows Registry so pari ime-podatek, shranjeni znotraj ključev. Vsaki vrednosti znotraj ključa je dodeljeno edinstveno ime, ki je prav tako neobčutljivo na veliko začetnico. V imenu lahko tudi vsebujejo levo poševnico, toda s takim početjem je težje ločevati vrednost od poti do ključa in Microsoft takšno početje odsvetuje. Sama terminologija je nestandardna, saj so ključi bolj podobnim tabelam, medtem ko je standardna praksa, da ključ predstavlja imenski del vrednosti. Včasih je Windows 3 omogočal, da ima en ključ eno vrednost, ampak z uveljavitvijo 32-bitnega sistema so dodali možnost dodajanja več vrednosti enemu ključu, pri čemer del vrednosti predstavlja ime. Kompatibilnost za nazaj je omogočena tako, da je dovoljeno prazno ime (prazen string).

Vsaka vrednost lahko shranjuje poljuben podatek z dolžino in vrsto kodiranja, ki pripada enemu izmed naslednjih tipov:

- REG_NONE: brez tipa.
- REG_SZ: niz, ki se konča z NULL znakom, shranjen v načinu kodiranja UTF-16.
- REG_EXPAND_SZ: "expandable"oz. raztegljiv niz, ki omogoča shranjevanje okoljskih spremenljivk, kodiran v enakem načinu kot REG_SZ.
- REG_BINARY: binarni podatki.
- REG_DWORD (REG_DWORD_LITTLE_ENDIAN): DWORD vrednost, ki predstavlja 32-bitno nepredznačeno celo število, zapisana v zaporedju bitov **little endian** oz. najmanj pomemben bit najprej.
- REG_DWORD (REG_DWORD_BIG_ENDIAN): DWORD vrednost, ki predstavlja 32-bitno nepredznačeno celo število, zapisana v zaporedju bitov **big endian** oz. najbolj pomemben bit najprej.
- REG_LINK: simbolična povezava, kodirana v UNICODE, do drugega ključa v registru Windows, ki navaja korenski ključ in pot do željenega ključa.
- REG_MULTI_SZ: večnizen podatek, predstavljen v obliki urejene tabele nepraznih nizov, kodiran v UNICODE, kjer se posamezen niz konča z NULL znakom, seznam pa s ponovnim NULL znakom.
- REG_QWORD (REG_QWORD_LITTLE_ENDIAN): QWORD vrednost, ki predstavlja 64-bitno celo število, zapisana v zaporedju bitov **little endian** oz. najmanj pomemben bit najprej.

Urejanje

Za popolno odstranitev programske opreme, odpravljanje napak in spreminjanje določenih nastavitev moramo urediti zapise znotraj registra za Windows. Pri tem imamo zelo veliko možnosti, od grafičnega vmesnika do skript in programov.

Urejevalnik registra

Urejevalnik registra RegEdit.exe predstavlja najenostavnejše delo z Windows Registry. Program ponuja grafični vmesnik za enostavno uporabo in omogoča ustvarjanje, preimenovanje in brisanje ključev, podključev in vrednosti. Prav tako omogoča uvoz in izvoz v obliki .REG datotek in v binarnem formatu. Grafični vmesnik ima tudi zavihek Favorites oz. Najljubši, v katerega lahko shranjujemo ključe. Del funkcionalnosti grafičnega vmesnika je tudi iskalna funkcija za iskanje po ključih in vrednostih - tako po imenu, kot tudi po dejanskem podatku. Nenazadnje lahko tudi urejamo register na drugem računalniku, v kolikor je znotraj omrežja.

.REG datoteka

Datoteke .REG so tekstovne datoteke, zapisane v človeku razumljivi obliki, in se uporabljajo za izvažanje in uvažanje delov registra. Uporabljajo sintakso .INI datotek, kar je nekoliko ironično, saj je register ravno te datoteke nadomestil.

```
[<Korenski kljuc>\<Kljuc>\<Podkljuc>]
"Ime vrednosti"=<Tip vrednosti>:<Podatek>
```

Privzeto vrednost ključa lahko nastavimo s pomočjo @ namesto "Ime vrednost".

```
[<Korenski kljuc>\<Kljuc>\<Podkljuc>]
@=<Tip vrednosti>:<Podatek>
```

Za nize velja, da ne potrebuje oznake "Tip vrednosti", ampak mora posebne znake, leva poševnica in narekovaj, ubežati z znakom leva poševnica. Brisanje ključa in vseh podključev zapišemo z minusom pred potjo do ključa.

```
[-HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft]
```

Podobno velja tudi za brisanje privzete vrednosti in in navadne vrednosti, kjer minus znak (-") postavimo za znak za enakost ("=").

Command Line

Za urejanje registra za Windows s pomočjo ukazne vrstice uporabljamo programe RegEdit.exe, Reg.exe in RegIni.exe. Uvažanje .REG datoteke:

```
RegEdit.exe /s <datoteka>
```

Izvažanje registra v datoteko:

```
; Izvoz celega registra
RegEdit.exe /e datoteka
; Izvoz korensekga kljuca HKCR
RegEdit.exe /e datoteka HKEY_CLASSES_ROOT[\<key>]
```

Branje imena operacijskega sistema:

```
Reg.exe QUERY "HKLM\Software\Microsoft\Windows NT\CurrentVersion" /v
    ProductName
```

Program RegIni.exe se uporablja za nastavljanje dovoljenj za upravljanje z registrom.

Poleg zgornjih ukazov zmore Windows Powershell narediti še več. Vgrajenega ima namreč ponudnika za delo z registrom, ki omogoča, da register za Windows obravnava kot del datotečnega sistema. Zato lahko register urejamo na enak način, kot bi urejali datoteke. Pridobivanje ključev znotraj trenutne lokacije:

```
<#Premik v kljuc, ki vsebuje informacije o operacijskem sistemu#>
cd "HKLM:Software\Microsoft\Windows NT\CurrentVersion"
<#Pridobi vse vrednosti#>
Get-ChildItem
<#Pridobi vrednost z imenom ProductName#>
Get-ItemPropertyValue -Name ProductName
```

Powershell omogoča tudi izvajanje transakcij, kar pomeni, da lahko več sprememb združimo v eno atomarno enoto. Ta se lahko izvede v celoti in shrani spremembe v register ali pa ena izmed sprememb spodleti in se register vrne v prejšnje stanje.

```
Start-Transaction
New-Item -Name SPO -Path HKCU:\Software -UseTransaction
New-Item -Name PowerShell -Path HKCU:\Software\SPO -UseTransaction
New-Item -Name "Transakcija" -Path HKCU:\Software\SPO\PowerShell -
    Value "Commit" -UseTransaction
Complete-Transaction
Start-Transaction
New-Item -Path HKCU:\Software\SPO\PowerShell\Transakcija -Value "
    Rollback" -UseTransaction
Undo-Transaction
```

Skripte in programi

Z uporabo API-ja Advanced Windows 32 Base API Library (advapi32.dll) lahko urejamo Windows Registry tudi s pomočjo programov. Spodaj je podan primer programa spisanega v jeziku C. Uporablja header file windows.h, ki vsebuje vse funkcije Windows API-ja, tudi tiste za delo z registri.

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

int main() {
    LONG reg;
    HKEY key;
    reg = RegCreateKeyEx(
            HKEY_CURRENT_USER,
            "Software\\SPO\\C",
            0,
            NULL,
            REG_OPTION_NON_VOLATILE,
            KEY_ALL_ACCESS,
            NULL,
            &key,
            NULL);
    if (reg != ERROR_SUCCESS) {
            printf("Ustvarjanje kljuca ni uspelo: %s", GetLastError()
                );
    } else {
            printf("Ustvarjanje kljuca je bilo uspesno.");
    }
}
```

Alternative

V nasprotju z Windows operacijskim sistemom in odločitvijo za uporabo binarne podatkovne baze so se drugi operacijski sistemi odločili za uporabo tekstovnih datotek za shranjevanje nastavitev procesov in aplikacij, ki jih grupirajo za lažje vodenje.

Unix-like

V Unix-like operacijskih sistemih, vključno z Linuxom, ki sledijo Filesystem Hierarchy Standardu, so sistemske nastavitve, torej korenski ključ HKEY_LOCAL_MACHINE, shranjene v /etc direktoriju in njegovih podmapah. Uporabnikovi podatki in nastavitve, HKEY_CURRENT_USER, se nahajajo v domačem imeniku v obliki skritih map in datotek - ime se prične s piko.

macOS

Čeprav je macOS UNIX skladen operacijski sistem, je zadeva tu popolnoma drugačna. Informacije o sistemu so shranjene v mapi /Library, medtem ko se podatki in nastavitve uporabnika nahajajo v domačem imeniku, ponovno znotraj direktorija Library, torej ∼/Library. Spremembe nastavitev sistema pa se nahajajo v mapi /System/Library. V vsaki od teh map se nahaja tudi podmapa Preferences, kamor aplikacije shranjujejo podatke.

Zaključek

Čeprav je Windows Registry sprva videti težko razumljiv in nevaren za uporabo, lahko vidimo, da temu ni tako. Programerji lahko z njegovo pomočjo dostopajo do sistemskih informacij ali pa celo shranjujejo nastavitve aplikacij in s tem zagotavljajo boljšo uporabniško izkušnjo. V kolikor smo torej dovolj previdni in naredimo varnostno kopijo ter se zavedamo, katere ključe spreminjamo in zakaj, vidimo, da delo z registrom za Windows ni nič drugačno od spreminjanja nastavitev in konfiguracije v drugih operacijskih sistemih. Poleg tega imamo na voljo mnogo orodij, ki nam delo olajšajo, in lahko izberemo tistega, s katerim se počutimo najbolj domače.

Literatura

[1] *Registry — Microsoft Docs.*

URL: https://docs.microsoft.com/en-us/windows/win32/sysinfo/registry

[2] *Windows registry information for advanced users — Microsoft Docs.*

URL: https://docs.microsoft.com/en-us/troubleshoot/windows-server/performance/windows-registry-advanced-users

[3] *Understanding the Windows Registry — University of Connecticut*

URL: https://confluence.uconn.edu/ikb/desktop-support/windows-10-support/understanding-the-registry-on-windows

[4] *Windows Registry — Wikipedia*

URL: https://en.wikipedia.org/wiki/Windows_Registry

[5] *Using a .reg file — Microsoft Docs.*

URL: https://support.microsoft.com/en-us/topic/how-to-add-modify-or-delete-registry-subkeys-and-values-by-using-a-reg-file-9c7f37cf-a5e9-e1cd-c4fa-2a26218a1a23

[6] *Working with Registry - Powershell — Microsoft Docs.*

URL: https://docs.microsoft.com/en-us/powershell/scripting/samples/working-with-registry-keys?view=powershell-7.2

[7] *Registry Functions — Microsoft Docs.*

URL: https://docs.microsoft.com/en-us/windows/win32/sysinfo/registry-functions