

Ficha nº 2 – Registry

Âmbito da matéria

Assume-se que os alunos conhecem as funções da biblioteca *standard* de C e os mecanismos *standard* para entrada e saída de dados em consola em C ou C++.

Assume-se também que acompanharam a matéria das aulas anteriores.

Ao longo desta ficha, será possível:

- Criar, consultar, alterar e remover chaves do *Registry*.

Referência bibliográfica

- Capítulo 3 do Livro *Windows System Programming* (da Bibliografia) (pags. 86-97)
- MSDN:

Registry Functions: <https://docs.microsoft.com/en-us/windows/win32/sysinfo/registry-functions>

32-bit and 64-bit Application Data in the Registry

<https://docs.microsoft.com/en-us/windows/win32/sysinfo/32-bit-and-64-bit-application-data-in-the-registry?redirectedfrom=MSDN>

Introdução e contexto

O *registry* é um sistema centralizado de armazenamento do Windows acerca de todos os seus aspectos, por exemplo, funcionamento de componentes centrais, controlo de utilizadores e políticas que estes devem seguir, e configuração de aplicações completamente externas ao sistema. A informação armazenada no *registry* é, normalmente, permanente até ser explicitamente apagada. No entanto, também é possível armazenar informação que permanece apenas até se encerrar o sistema.

Os elementos principais constituintes do *registry* são as **chaves** e os **valores**. As chaves podem conter outras chaves (sub-chaves) e valores; um *valor* é um par nome-valor, havendo aqui alguma sobreposição de nomes e sendo necessário distinguir o conceito de valor como elemento chave-valor que aparece dentro de uma chave, e valor como o valor associado a um certo nome no par chave-valor. O valor pode assumir vários tipos de dados fazendo parte da especificação do par-valor na sua criação. É preciso ter em atenção que o uso destes termos pode eventualmente variar conforme a bibliografia, por exemplo ser usado apenas o termo *chave*. O par chave-valor é muitas vezes referido como **named value**, o que faz sentido.

O *registry* está organizado de uma forma hierárquica em que chaves contém outras chaves (sub-chaves). O acesso a uma chave ou valor implicará navegação até ao ponto onde se encontra e envolve o nome das chaves a partir da raiz de uma forma análoga ao sistema, de ficheiros. As chaves de raiz estão associadas a temas de configuração específicas e são

HKEY_CLASSES_ROOT
HKEY_CURRENT_CONFIG
HKEY_CURRENT_USER
HKEY_LOCAL_MACHINE
HKEY_USERS

O acesso a algumas partes do *registry* exige permissões mais elevadas do que outras. O *registry* está disponível às aplicações para criação, consulta e modificação de chaves e valores através de um API win32 que envolve um conjunto extenso de funções.

O API segue uma lógica semelhante ao restante API do Windows: i) abre-se ou cria-se uma chave obtendo-se um *handle*; ii) manipula-se a chave usando o *handle* obtido, iii) fecha-se o *handle*. Aplica-se a esta API as mesmas considerações acerca de chars, wide chares e TCHAR.

O link abaixo permite aceder a uma explicação exaustiva deste API, mas nesta ficha vai ser necessário apenas ter uma ideia das funções principais e dos conceitos envolvidos:

Registry Functions

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

A diferença entre sistemas de 32 bits e 64 bits é importante no API e na localização das chaves no registry. O link abaixo é útil nesta questão:

32-bit and 64-bit Application Data in the Registry

<https://docs.microsoft.com/en-us/windows/win32/sysinfo/32-bit-and-64-bit-application-data-in-the-registry?redirectedfrom=MSDN>

Esta introdução faz parte e consta nos documentos das aulas teóricas. É apresentado aqui como parte da ficha de exercícios para maior comodidade de consulta e incentivo de preparação para a aula. **De modo algum a inclusão deste conteúdo aqui dispensa as aulas teóricas.**

Exercícios

1. Conceitos e visão geral do API

- a) Verifique que leu o texto introdutório. Se não leu, leia agora.
- b) Oiça a explicação do professor do laboratório, em complemento com a exposição do professor dada ou a dar pelo professor da aula teórica.
- c) Invoque o editor do *registry* do Windows (**regedit**) e dê uma olhadela rápida ao *registry*, expandindo algumas das chaves e vendo o nome dado às chaves, os quais reflectem o seu objectivo. Tente identificar onde estarão as configurações da máquina e as do seu utilizador.
- d) Usando um *browser*, siga o primeiro link indicado acima e obtenha uma visão geral do API. Procure identificar, pelo nome, as funções para criar, consultar e modificar chaves no *registry*.

2. Manipulação do *registry* através de um programa.

Usando as funções do API *registry* `RegCreateKeyEx`, `RegSetValueEx`, `RegQueryValueEx` (parâmetros indicados no final do documento), crie um programa que permite criar, mostrar, modificar ou apagar uma chave. Siga a estratégia indicada abaixo, já transformadas em alíneas para se orientar melhor, e teste o código resultante deste exercício e suas alíneas.

- a) Construa um programa com base no modelo **ex2.c** fornecido mais adiante nesta ficha. Trata-se de um programa genérico agnóstico quanto a *chars/wide chars* destinado a poupar tempo. É necessário que a breve tempo seja capaz de iniciar um projecto deste tipo sozinho e este código inicial deixará de ser fornecido muito brevemente.
- b) Acrescente ao programa a possibilidade de perguntar ao utilizador o nome de uma chave. O programa irá verificar se a chave indicada, que deverá ter a seguinte localização: `HKEY_CURRENT_USER, "Software\AULA\..."` existe ou não. Se não existir, cria-a, indicando ao utilizador esse facto. As reticências representam o nome da chave indicado pelo utilizador. Deve ter em atenção a codificação do carácter especial “\” e a questão dos `TCHAR`.
- c) Verifique a existência da chave indicada com o *registry* editor do sistema (programa **regedit**)
- d) O programa deverá de seguida perguntar ao utilizador se deseja criar, consultar ou eliminar um valor (par *chave-valor*) à chave das alíneas anteriores.
- e) O programa deverá agora, no final das operações indicadas atrás, listar todas os pares chave-valor existentes na chave. Utilize a função API `RegEnumValue` (parâmetros indicados mais adiante, ou no *link* indicado na página inicial).
- f) O programa deverá agora permitir, mediante a indicação do utilizador, eliminar do *registry* toda a informação que foi criada por este programa.

3. Explorar as diferenças entre as chaves raiz.

Modifique o programa anterior de forma a usar a chave `HKEY_LOCAL_MACHINE` em vez de `HKEY_CURRENT_USER`. Verifique que agora são necessários privilégios de administrador. Corra a aplicação como administrador (se estiver a usar o seu computador).

4. Explorar diferentes tipos de dados.

Volte a modificar o programa de forma a usar a chave `HKEY_CURRENT_USER` e permita ao utilizador especificar o tipo de dados do valor no par *chave-valor* a usar, tendo o cuidado de adaptar as restantes partes do programa que são dependentes do tipo de dados.

Se não tiver concluído o exercício 3 durante a aula, deverá então ter o cuidado de o concluir em casa, passando depois a este e encarando a tarefa como parte do estudo associado a esta disciplina.

Listagem de Programas

ex2.c

```
#include <windows.h>
#include <tchar.h>
#include <io.h>
#include <fcntl.h>
#include <stdio.h>

#define TAM 200

int _tmain(int argc, TCHAR *argv[]){
    HKEY chave;
    TCHAR chave_nome[TAM], par_nome[TAM], par_valor[TAM];

    /* ... Mais variáveis ... */

#ifdef UNICODE
    _setmode(_fileno(stdin), _O_WTEXT);
    _setmode(_fileno(stdout), _O_WTEXT);
    _setmode(_fileno(stderr), _O_WTEXT);
#endif

    /* ... as várias alíneas ...*/

    RegCloseKey(chave);

    return 0;
}
```

Resumo das funções API mais centrais a estes exercícios

- Algumas das funções são indicadas na sua versão genérica, ou seja nem **...A e nem ...W**, apesar de o link para o MSDN ser apenas a versão **char/ASCII**. Aqui adota-se a utilização genérica da função que está mapeada na biblioteca <tchar.h> tal como indicado no livro da bibliografia. Assim, deve preferir sempre as versões agnósticas, fazendo a necessária adaptação, quando necessário para TCHAR (nome e parâmetros).
- Para detalhes dos parâmetros, deve usar a aula ou seguir o link indicado em cada função.
- A informação foi directamente obtida do site oficial da Microsoft. Tem um objectivo secundário de mostrar como se obtém este tipo de informação e irá ser dado progressivamente mais incentivo à sua consulta directamente no site e cada vez menos nestes documentos.
- A informação é apresentada no original em Inglês, não só porque não faz sentido traduzir este tipo de material, mas também como forma de promover a competência de domínio de língua Inglesa, cada vez mais necessária nas entrevistas de emprego.

Este resumo de API consta num documento no contexto de aulas teóricas. É apresentado aqui como parte da ficha de exercícios para maior comodidade de consulta. Não dispensa as aulas teóricas.

RegCreateKeyEx

Creates the specified registry key. If the key already exists, the function opens it. Note that key names are not case sensitive.

<https://docs.microsoft.com/en-us/windows/win32/api/winreg/nf-winreg-regcreatekeyexa>

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

RegOpenKeyEx

Opens the specified registry key. Note that key names are not case sensitive.

<https://docs.microsoft.com/en-us/windows/win32/api/winreg/nf-winreg-regopenkeyexa>

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

RegOpenCurrentUser

Retrieves a handle to the HKEY_CURRENT_USER key for the user the current thread is impersonating.

<https://docs.microsoft.com/en-us/windows/win32/api/winreg/nf-winreg-regopencurrentuser>

```
LSTATUS RegOpenCurrentUser (
    REGSAM samDesired,
    PHKEY  phkResult
);
```

RegDeleteKeyEx

Deletes a subkey and its values from the specified platform-specific view of the registry. Note that key names are not case sensitive.

<https://docs.microsoft.com/en-us/windows/win32/api/winreg/nf-winreg-regdeletekeyexa>

```
LSTATUS RegDeleteKeyEx (
    HKEY          hKey,
    LPCTSTR       lpSubKey,
    REGSAM        samDesired,
    DWORD         Reserved
);
```

RegEnumValue

Enumerates the values for the specified open registry key. The function copies one indexed value name and data block for the key each time it is called.

<https://docs.microsoft.com/en-us/windows/win32/api/winreg/nf-winreg-regenumvaluea>

```
LSTATUS RegEnumValue (
    HKEY    hKey,
    DWORD   dwIndex,
    LPTSTR  lpValueName,
    LPDWORD lpccchValueName,
    LPDWORD lpReserved,
    LPDWORD lpType,
    LPBYTE  lpData,
    LPDWORD lpcbData
);
```

RegQueryValueEx

Retrieves the type and data for the specified value name associated with an open registry key.

To ensure that any string values (REG_SZ, REG_MULTI_SZ, and REG_EXPAND_SZ) returned are null terminated, use the RegGetValue function.

<https://docs.microsoft.com/en-us/windows/win32/api/winreg/nf-winreg-regqueryvalueexa>

```
LSTATUS RegQueryValueEx (
    HKEY    hKey,
    LPCTSTR lpValueName,
    LPDWORD lpReserved,
    LPDWORD lpType,
    LPBYTE  lpData,
    LPDWORD lpcbData
);
```

RegGetValue

Retrieves the type and data for the specified registry value.

<https://docs.microsoft.com/en-us/windows/win32/api/winreg/nf-winreg-reggetvaluea>

```
LSTATUS RegGetValue (
    HKEY    hkey,
    LPCTSTR lpSubKey,
    LPCTSTR lpValue,
    DWORD   dwFlags,
    LPDWORD pdwType,
    PVOID   pvData,
    LPDWORD pcbData
);
```

RegDeleteValue

Removes a named value from the specified registry key. Note that value names are not case sensitive.

<https://docs.microsoft.com/en-us/windows/win32/api/winreg/nf-winreg-reggetvaluea>

```
LSTATUS RegDeleteValue(  
    HKEY    hKey,  
    LPCTSTR lpValueName  
);
```

RegDeleteKeyValue

<https://docs.microsoft.com/en-us/windows/win32/api/winreg/nf-winreg-regdeletekeyvaluea>

Removes the specified value from the specified registry key and subkey.

```
LSTATUS RegDeleteKeyValue(  
    HKEY    hKey,  
    LPCTSTR lpSubKey,  
    LPCTSTR lpValueName  
);
```

RegQueryInfoKey

Retrieves information about the specified registry key.

<https://docs.microsoft.com/en-us/windows/win32/api/winreg/nf-winreg-regqueryinfokeya>

```
LSTATUS RegQueryInfoKey(  
    HKEY    hKey,  
    LPTSTR  lpClass,  
    LPDWORD lpccchClass,  
    LPDWORD lpReserved,  
    LPDWORD lpcSubKeys,  
    LPDWORD lpcbMaxSubKeyLen,  
    LPDWORD lpcbMaxClassLen,  
    LPDWORD lpcValues,  
    LPDWORD lpcbMaxValueNameLen,  
    LPDWORD lpcbMaxValueLen,  
    LPDWORD lpcbSecurityDescriptor,  
    PFILETIME lpftLastWriteTime  
);
```

RegCloseKey

Closes a handle to the specified registry key.

<https://docs.microsoft.com/en-us/windows/win32/api/winreg/nf-winreg-regclosekey>

```
LSTATUS RegCloseKey(  
    HKEY hKey  
);
```