

W tym tutorialu postaram się pokazać, jak w C/C++ uzyskać podstawowe informacje o podłączonych do systemu woluminach logicznych. Ale od początku - co to są te woluminy logiczne? Inaczej zwane partycjami lub dyskami logicznymi są wydzielonymi obszarami fizycznego nośnika danych (takiego jak dysk twardy, pendrive, płyta CD czy DVD) i służą do przechowywania informacji. W systemie Windows (bo tego systemu ten tekst dotyczy) każdy dysk logiczny posiada przypisaną do niego literę alfabetu. Pierwsze dwie (A i B) są zarezerwowane dla dyskietek, litera C jest pierwszą dostępną dla dysków twardych i reprezentuje zwykle partycję systemową (na której jest umieszczony system). Należy uważać, żeby nie pomylić fizycznego nośnika z partycją, bo, mimo iż popularnie mówi się np. "dysk c" mowa tu o wydzielonej partycji na którymś z nośników (jeden dysk może być podzielony na wiele woluminów/partycji, każdą reprezentowaną inną literą). Także tutaj będzie pokazane, w jaki sposób można uzyskać dane o partycjach (między innymi o numerze seryjnym woluminu, nie wolno mylić z serialem HDD!).

Ktoś może się zapytać - ale po co nam takie informacje? No cóż, mogą się przydać w wypadku, gdy chcemy zabezpieczyć swój program (aby utrudnić jego rozpowszechnianie bez licencji), lub (jak to jest wykonane np. w GG v8) użyć np. numeru seryjnego jednego z woluminów do szyfrowania danych (w tym wypadku to było szyfrowanie plików komunikatora zawierających dane użytkownika). Następne narzucające się pytanie to: co to jest to WinAPI? Jest to zbiór funkcji, stałych, zmiennych czy struktur umożliwiający działanie programu w systemie Windows. Udostępnia ono wiele nowych możliwości, często niemożliwych do osiągnięcia korzystając z elementów udostępnianych przez język.

Tyle teorii powinno wystarczyć - czas na praktykę! Co nam będzie potrzebne:

- Edytor (w którym będziemy pisali kod), np. Code::Blocks, Microsoft Visual Studio, Notepad++ etc.
- Kompilator, np. MinGW
- System Windows, dzięki któremu będziemy mogli zobaczyć efekty naszej pracy
- Podstawowa wiedza o języku C++
- Trochę cierpliwości (bo mam zwyczaj strasznie lać wodę :P)

Na początku musimy określić, co nam będzie potrzebne do wykonania zadania, czyli jakich funkcji i bibliotek będziemy musieli użyć. Do uzyskania podstawowych informacji o partycji służą 2 funkcje WinAPI - `GetDriveType()` oraz `GetVolumeInformation()`. Pierwsza służy do określenia typu nośnika danych, druga udostępnia już požądane przez nas informacje (np. nazwa woluminu, system plików czy serial). Obie funkcje są udostępnione w pliku nagłówkowym "windows.h", dodatkowo będziemy potrzebowali funkcji umożliwiających wypisanie wyniku na konsolę, czyli standardowe `cout` z biblioteki "iostream" oraz biblioteki string, która powinna nam ułatwić później zadanie:

```
{codecitation class='brush: cpp'} #define ARRAYSIZE(a) (sizeof(a)/sizeof(a[0])) // Makro, dzięki któremu możemy obliczyć wielkość tablicy z elementami stałej wielkości #include #include #include // Przyda się później #include /** * Główna funkcja programu */ int main(void) { return 0; } // End main() {/codecitation}
```

No dobrze, mamy już szkielet programu, możemy przejść dalej - trzeba ustalić jakie zmienne będą nam potrzebne (w nich będziemy przechowywać informacje o woluminach). Jak możemy zobaczyć na [stronach msdn](#) funkcja `GetDriveType()` zwraca żadaną wartość, więc nie musimy jej nigdzie zapisywać (choć możemy, nikt nam przecież nie broni :)), a jako parametr przyjmuje główny katalog partycji. Natomiast `GetVolumeInformation()` zwraca kod błędu, a argumenty przyjmuje jako pointery (wskaźniki); są to kolejno (w nawiasie typ): główny katalog partycji (`char*`), bufor na nazwę partycji (`char*`), jej

Wpisany przez Tomasz Stasiak
środa, 10 sierpnia 2011 22:38

wielkość (DWORD), numer seryjny (DWORD*), maksymalna długość elementu ścieżki (DWORD*), flagi systemu plików (DWORD*), bufor na nazwę systemu plików (char*), jego wielkość (DWORD).

Dlaczego DWORD? Dlaczego nie można użyć zwykłego inta? Ma to związek z architekturą komputera PC i Asemblerem, mianowicie int (w językach wysokiego poziomu) ma MINIMUM 16 bitów, czyli 2 bajty, natomiast DWORD (skrót o double WORD, czyli podwójne słowo) ma zawsze 32 bity, czyli 4 bajty. Coś takiego jak DWORD istnieje nawet w Asemblerze, dzięki czemu możliwe jest użycie WinAPI w języku dowolnego poziomu (w c/c++ dword jest zadeklarowany następująco: typedef unsigned long DWORD;).

Wracając do tematu: tworząc wszystkie wymienione powyżej zmienne można od razu stworzyć tablicę zawierającą listę możliwych rodzajów nośników na podstawie wartości zwracanej przez GetDriveType(), czyli (w nawiasie wartość z GetDriveType()): nieznany typ nośnika (0), katalog główny jest błędny (1), napęd wymienny (2), napęd stały (3), zdalny napęd (4), CDROM (5), dysk RAM (6).

```
    Tak więc dopisujemy ten kod do ciała funkcji main programu:    {codecitation class='brush:
cpp'}    // Zmienne winapi do wyciągania dokładniejszych info o dysku    char
fileSystemName[MAX_PATH*5 + 1] = {0}; // Nazwa systemu plików    char
volumeName[MAX_PATH*5 + 1] = {0}; // Nazwa partycji    DWORD maxComponentLen
= 0; // Maksymalna długość elementu ścieżki    DWORD fileSystemFlags =
0; // Flagi systemu plików (tu pominięte)    DWORD serialNumber = 0; // Numer
seryjny woluminu    // Typy dysków    std::string drive_types[] = {"nieznany typ",
"katalog glowny jest bledny", "naped wymienny (np.
pendrive)", "naped stalý (np. dysk twardy lub dysk flash)",
"zdalny naped", "CDROM", "dysk
RAM"};    {/codecitation}    Pozostaje jeszcze to, o co nam chodziło,
czyli uzyskanie informacji o wszystkich woluminach zamontowanych w systemie (jak już
wspominałem, każda partycja jest reprezentowana przez literę alfabetu). Tak więc musimy
"przejsć" przez wszystkie możliwe partycje, sprawdzić, czy są one prawidłowe (czyli nie jest to
nieznany typ lub z błędnym katalogiem głównym) i jeśli tak, to wyświetlić o nich informacje:
{codecitation class='brush: cpp'}    /// Rendering outputu    for(char i = 'C'; i
```