

Dfs - Distributed File System  
SMB - Server Message Block

18 grudnia 2002

# Spis treści

<b>1 Dfs - Distributed File System</b>	<b>3</b>
1.1 Przyjęte tłumaczenie	3
1.2 Wprowadzenie	3
1.3 Cechy Dfs	3
1.4 Dostęp do zasobów	4
1.5 Łączenie zasobów	5
1.6 Partition Knowledge Table (PKT)	6
1.7 System Plików	6
<b>2 Server Message Block (SMB)</b>	<b>7</b>
2.1 Wprowadzenie	7
2.2 Podstawowe komendy	7
2.3 Opportunistic locks	8
2.4 Samba	9

# 1 Dfs - Distributed File System

## 1.1 Przyjęte tłumaczenie

- *transparently* - niezauważalnie
- *junction* - złączenie

## 1.2 Wprowadzenie

Dfs został opracowany przez firmę Microsoft na potrzeby systemu WindowsNT w wersji 4.0, z rozszerzeniami zastosowany został również w Windows 2000 Server.

Historycznie, w systemach z rodziny Windows używany był Universal Naming Convention (UNC), który, aby umożliwić współdzielenie zasobów dyskowych, wymagał bezpośredniego podania nazwy serwera, udziału na nim i ew. dalszej ścieżki dostępu, np:

```
\\Serwer\Nazwa_Udziału\ścieżka\dostępu\plik
```

Możliwe było (jest) tzw. mapowanie udziałów jako litery dysku (w systemie Windows dyski i partycje na nich są nazywane literami alfabetu łacińskiego), czyli w/w udział

```
\\Serwer\Nazwa_Udziału
```

mógł zostać zamapowany jako np. s:. Od tej pory dostęp do zasobu mógł odbywać się przez podanie:

```
s:\ścieżka\dostępu\plik
```

Wraz z rozwojem sieci okazało się, że nie jest to najlepsze rozwiązanie - UNC powoduje, że użytkownik może być zmuszony do mapowania dużej ilości zasobów i niemożliwe jest hierarchiczne budowanie zasobów. Poza tym liczba liter w alfabecie (w szczególności w łacińskim) jest skończona.

Niedogodności te stara się rozwiązać Dfs.

## 1.3 Cechy Dfs

Dfs jest protokołem rozwiązywania nazw. Dostęp do plików odbywa się przy pomocy protokołu SMB.

**Hierarchiczna struktura** Dzięki możliwości "mountowania"(dokumenty Microsoft określają to mianem linkowania) udziałów, możliwa jest budowa dużej, hierarchicznej struktury. Może być ona widoczna dla użytkownika nie jako jeden dysk.

**Elastyczna modyfikacja struktury** Można dynamicznie podłączać i odłączać udziały, nie wpływając na działanie reszty wolumenów.

**Możliwość duplikacji danych** Można podłączyć wiele kopii tylko do odczytu tego samego udziału jako tę samą nazwę w strukturze Dfs. Jeżeli któraś z tych kopii stanie się niedostępna, użytkownik dalej może korzystać z pozostałych.

**Rozkładanie obciążenia** Jeżeli korzystamy z możliwości podłączenia wielu kopii tego samego udziału jako jedną, Dfs zapewnia pewne (limited) rozłożenie obciążenia tych dysków.

**Name transparency** Końcowy użytkownik nie widzi, w którym miejscu i w jaki sposób łączone są zasoby. Dlatego też w sposób niezauważalny dla niego można np. zmienić fizyczne miejsce zasobu pozostawiając niezmienną jego nazwę logiczną (widoczną dla użytkownika).

**Integracja z zabezpieczeniami WindowsNT** Administrator sieci nie musi się martwić o dodatkowe zabezpieczanie udostępnionych udziałów. Dostęp do zasobów odbywa się na podstawie weryfikacji praw dostępu przez mechanizmy systemu Windows.

**Cache'owanie informacji** W czasie pierwszego dostępu do zasobu pewne informacje są zapamiętywane na maszynie klienta i potem część transmisji może odbywać się lokalnie. Możliwość podłączenia Windows 9x jako klienta Dfs daje możliwość uruchomienia usługi, która umożliwi dostęp do przestrzeni zasobów klientom ze starszymi wersjami systemu Windows.

## **Graficzny interfejs użytkownika**

### **1.4 Dostęp do zasobów**

Dostęp do zasobów Dfs odbywa się przez UNC:

\\NazwaSerwera\NazwaUdziałuDfs\Sciezka\Do\Pliku, gdzie:

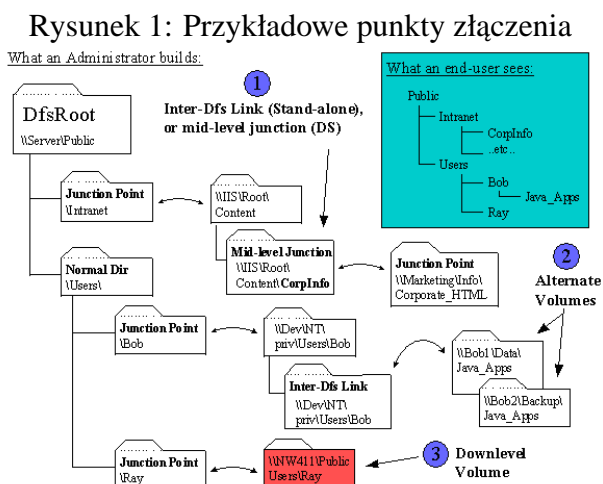
**NazwaSerwera** to nazwa komputera (serwera Dfs)

**NazwaUdziałuDfs** to nazwa, która ma być korzeniem (katalogiem głównym) udziału

**Sciezka\Do\Pliku** to dowolna, poprawna w sensie nazewnictwa Win32.

## 1.5 Łączenie zasobów

Łączenie zasobów odbywa się przez zdefiniowanie punktów złączenia: (junction points)



Typowe złączenie (zwykłe) to takie, które następuje między punktami znajdującymi się w hierarchii tego samego korzenia Dfs.

Oprócz zwykłych złączeń, możliwe są również połączenia z punktami z innych serwerów (ko-rzeni) Dfs:

**Post-Junction Junction** - Są to połączenia, które zawierają dalsze punkty złączenia

- *Inter-Dfs Links*  
Połączenie między różnymi wolumenami Dfs. Podczas przechodzenia takiego punktu następuje zmiana serwera Dfs do którego następować będą odwołania (rozwiązywanie nazw hierarchii katalogów), ale przejście to odbywa się niezauważalnie dla użytkownika.
- *Midlevel Junctions*  
Planowane jest rozszerzenie funkcjonalności Dfs, tak aby nie wymagały Inter-Dfs links, tylko aby wszystkie odwołania do hierarchii katalogów mogły być rozwiązane z poziomu macierzystego Dfs. Ma to na celu poprawę wydajności rozwiązywania nazw.

**Alternate Volumes** - Udostępnianie wielu kopii tego samego udziału pod jedną nazwą. Dfs nie sprawdza, czy rzeczywiście zawartość udziałów jest identyczna.

**Down-Level Volumes** - Połączenia z systemami innymi niż Dfs.

## **1.6 Partition Knowledge Table (PKT)**

PKT przechowują wiedzę (nie informacje!) o wszystkich punktach złączenia.

- Lokalnie PKT tworzona jest dla każdego zamapowanego udziału. Wpisy w PKT dodawane są w momencie przekraczania punktów złączenia. Domyślnie czas życia takiego wpisu wynosi 5 minut (odnawiane przy kolejnych użyciach). Lokalnie PKT przechowywana jest w pamięci RAM.
- Po stronie serwera PKT tworzone są w ilości jedna na maszynę. Przechowywana jest w rejestrach systemu.

## **1.7 System Plików**

Dfs może być zainstalowany na maszynach z systemem Windows NT Server 4.0 lub Windows 2000 Server, może być to FAT lub NTFS. Jeżeli połączone (punktami złączenia) są różne systemy plików, przejście pomiędzy nimi odbywa się niezauważalnie dla użytkownika

## 2 Server Message Block (SMB)

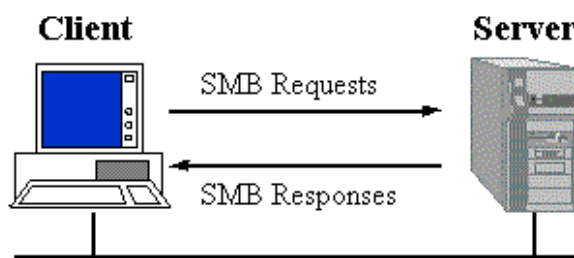
### 2.1 Wprowadzenie

SMB jest to opracowany na początku lat 80tych przez firmy Intel, Microsoft i IBM protokół wymiany plików (file sharing protocol) i drukarek. Jest to metoda udostępniania plików stosowana w systemach rodziny Windows, określana jako "Microsoft Network". SMB może być implementowany pod różnymi niskopoziomowymi protokołami sieciowymi. Oryginalnie SMB był implementowany pod (nad) protokołami NetBEUI i NetBois, aktualnie używa się TCP/IP.

SMB jest warstwą prezentacji (presentation layer), 6 najwyższą w siedmio poziomowej strukturze OSI (Open Systems Interconnect). 7 warstwą jest warstwa aplikacji.

SMB jest protokołem klient-serwer, żądanie-odpowieź. Czyli klient wysyła żądanie (komendę) do serwera a ten odpowiada.

Rysunek 2: SMB



### 2.2 Podstawowe komendy

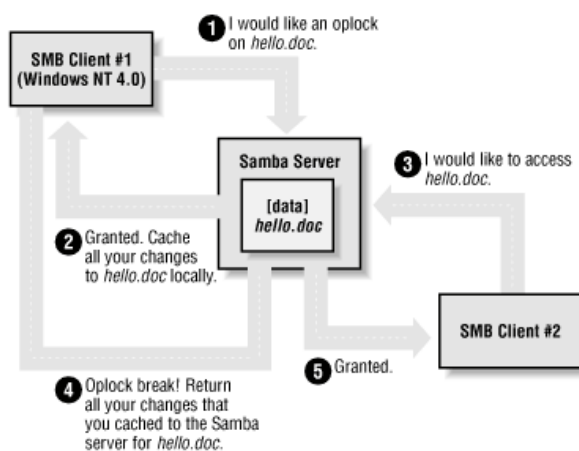
Najprostsza implementacja SMB obsługuje takie komendy jak:

- Łączenie się i odłączanie się od udziałów
- Otwieranie i zamykanie plików
- Odczytywanie i zapisywanie plików
- Tworzenie i usuwanie plików i katalogów
- Przeszukiwanie katalogów
- Pobieranie i ustawianie atrybutów pliku
- Blokowanie fragmentów pliku

## 2.3 Opportunistic locks

Ciekawy mechanizm jaki implementuje SMB to oplocks, blokady oportunistyczne (opportunistic locks). Są to metody blokowania plików przez klienta. Mechanizm ten pozwala klientowi na zakomunikowanie serwerowi SMB, że operacje na pliku będą wykonywane lokalnie. W sposób oczywisty zmniejsza to obciążenie sieci. Jeżeli od innego klienta nadejdzie żądanie dostępu do tego samego pliku, blokada jest zrywana - serwer SMB wysyła do pierwszego z nich żądanie uaktualnienia pliku - wykonania wszystkich zaległych zapisów.

Rysunek 3: Blokady oportunistyczne



Mechanizm oplocks to wyjątek od reguły żądanie-odpowieź - tu serwer również może wysyłać żądania do klienta.

Możliwe są trzy rodzaje oplocks:

**Wyłączny (exclusive) oplock** - klient ma wyłączność na dostęp do pliku.

Klient A po otrzymaniu takiego oplocka ma wyłączny dostęp do pliku. Jeżeli inny klient (B) zażąda otwarcia pliku, serwer wyśle do A informację o przerwaniu jego blokady. Zmusza to A do wykonania zaległych zapisów do pliku i zamknięcia go lub odłączenia się od niego. Po wykonaniu tej synchronizacji plik udostępniany jest klientowi B

**Batch oplock** - utrzymanie otwartego pliku nawet po zamknięciu go przez klienta.

W przypadku, kiedy aplikacja klienta A często otwiera i zamyka plik batch oplock redukuje ruch sieciowy - pozwala klientowi A uniknąć ciągłego wysyłania żądań otwarcia i zamknięcia pliku. Jeżeli nadejdzie żądanie dostępu do pliku od innego klienta (B), serwer informuje klienta A, że ten musi zmienić tryb dostępu.

**Level II oplock** - pozwala na równoczesny odczyt pliku przez wielu klientów.

Jeżeli klient A ma exclusive oplock, ale otworzył plik w trybie tylko do odczytu i nadejdzie żądanie otwarcia pliku przez innego klienta (B), klient A jest informowany, że jego



blokada musi zostać przerwana i zostaje mu przydzielona blokada Level II. Po dokonaniu tej zmiany, klient B może również otrzymać blokadę Level II.

## **2.4 Samba**

Samba to darmowa (wolna) implementacja SMB dla systemów z rodziny Unix. Prace nad Sambą zapoczątkował Andrew Tridgell. Samba rozwijana była na zasadzie reverse-engineeringu (firma Microsoft nie udostępnia dokumentacji protokołu SMB).

<http://afr.com/it/2002/10/01/FFXDF43AP6D.html>