

# 5

## *Tallózás és a lemezmegosztások folytatása*

Ebben a fejezetben folytatjuk a lemezmegosztásokról az előző fejezetben elkezdett fejtegetéseinket. Megismerkedünk a Windows és a Unix fájlrendszerei közötti különbségekkel, és látni fogjuk, miképpen hidalja át ezeket a Samba. Meglepődve fogjuk tapasztalni, hogy mennyi eltérés van a DOS és a Unix fájlrendszere között. Emellett röviden szólnunk a névszűrőiről, a fájlok zárolásáról és a Samba egyik viszonylag új képességéről, az „opportunist” zárolásról. Mielőtt azonban elmerülnénk ezekbe a részletekbe, ismerkedjünk meg egy valamelyest misztikus témával, nevezetesen a tallózással.

### *Tallózás*

A tallózás ebben az összefüggésben azt a tevékenységet jelenti, hogy megvizsgáljuk, milyen kiszolgálók és megosztások találhatók a hálózaton. A Windows NT 4.0 vagy Windows 95/98 operációs rendszert használó felhasználók a Network Neighborhood (Hálózat) ablakban tallózhatják a hálózat kiszolgálóit. A kiszolgálót jelképező ikonra kattintva azt is megnézhetik, hogy az adott kiszolgálón milyen nyomtató- és lemezmegosztások érhetők el.

A Windows parancssorába a `net view` parancs beírásával is megnézhetjük, milyen kiszolgálók érhetők el a hálózaton. A parancs hatására például az alábbi eredményt kaphatjuk:

```
C:\>net view
Servers available in workgroup SIMPLE
Server name          Remark
-----
\\CHIMAERA           Windows NT 4.0
\\HYDRA              Samba 2.0.4 on (hydra)
\\PHOENIX            Windows 98
```

### *A tallózás letiltása*

A `browseable` beállítás segítségével kizárhatunk egy megosztást a tallózólistából. Az ilyen módon kizárt megosztás egyáltalán nem jelenik meg a Network Neighborhood ablakában. Ha például azt szeretnénk, hogy az előző fejezetben használt `[data]` megosztás ne legyen látható, akkor a következőket kell beírunk:

```
[data]
    path = /home/samba/data
    browseable = no
    guest ok = yes
    comment = Data Drive
    volume = Sample-Data-Drive
    writeable = yes
```

Bár egy megosztást feltehetően nem azért hozunk létre, hogy elrejtjük a felhasználók előtt, bizonyos esetekben mégis jól jön ez a lehetőség. Készíthetünk például egy [net-login] megosztást, amelyben a Windows tartományvezérlőinek bejelentkezési szkriptjeit akarjuk tárolni (a bejelentkezési szkriptekről további információk a *„Felhasználók, biztonság és tartományok”* című 6. fejezetben olvashatunk).

Másik példa lehet erre a [homes] megosztás. Ezt a megosztást gyakran nem tallózhatónak állítják be, úgyhogy ez nem jelenik meg, amikor a felhasználó az ezt a megosztást tartalmazó gépet tallózza. Ha viszont egy alice nevű felhasználó bejelentkezik, és tallózza a gépet, az illető gép helyett egy [alice] megosztás fog megjelenni. Hogyan érhetnénk el, hogy minden ügyfél előtt megjelenjen ez a megosztás, mielőtt még az alice nevű felhasználó bejelentkezne? A megoldás a globális auto service beállítás. Ez a beállítás előre betölti a megosztásokat egy tallózólistába, hogy mindig láthatók legyenek a megosztások:

```
[global]
    ...
    auto services = alice
    ...
```

### Alapértelmezett szolgáltatások

Arra az esetre, ha egy felhasználó nem tudna sikeresen kapcsolódni egy megosztáshoz, megadhatunk egy alapértelmezett megosztást, amelyhez mindig tud kapcsolódni. Miután nem tudhatjuk, hogy egy adott időben éppen ki akarja használni ezt a megosztást, vegyük fel hozzá a guest ok = yes beállítást. Egy default service (alapértelmezett szolgáltatás) specifikálása akkor lehet hasznos, ha valakinek végső segítségként egy súgófájlt akarunk a rendelkezésére bocsátani. Például:

```
[global]
    ...
    default service = helpshare
    ...
[helpshare]
    path = /home/samba/helpshare/%S
    browseable = yes
    guest ok = yes
    comment = Alapértelmezett megosztás sikertelen kapcsolatnál
    volume = Sample-Data-Drive
    writeable = no
```

Figyeljük meg, hogy a `path` beállításban a `%S` változót használtuk. Ez a változó a kért, de nem létező megosztásra hivatkozik (a felhasználó által kért eredeti megosztásra), és nem az eredményül kapott alapértelmezett megosztásra. Így lehetőségünk van arra, hogy az egyes kiszolgálók nevével különböző elérési utakat definiáljunk, és így egyénre szabott súgófájlokat bocsássunk a felhasználók rendelkezésére. Emellett a `%S` változó azt is lehetővé teszi, hogy a kért megosztásban lévő aláhúzásjelek (`_`) törtvonallá (`/`) alakuljanak át.

### *Tallózó kiválasztása*

Amint az 1. fejezetben említettük, mindegyik alhálózatban van egy számítógép, amelyik listát vezet az éppen aktív gépekről. Ezt a listát *tallózólistának* nevezzük, a listát vezető kiszolgálónak pedig *helyi főtallózó* a neve. Amint különböző gépek rákapcsolódnak a hálózatra vagy lekapcsolódnak róla, a helyi főtallózó folyamatosan frissíti a tallózólistát, és kérésre bármelyik gép rendelkezésére bocsátja.

A helyi alhálózatban a tallózó kiválasztásának eredményétől függően lehet egy számítógép a helyi főtallózó. A tallózó kiválasztása bármikor kikényszeríthető. A Samba különböző módon képes a választási eredmény „meghamisítására”, beleértve azt is, hogy az adott alhálózatban mindig ő legyen a helyi főtallózó, vagy éppenséggel sohase legyen az. Így például az alábbi beállításokkal, amelyekkel a 4. fejezetben használt konfigurációs fájl egészítjük ki, biztosíthatjuk, hogy a Samba mindig megnyerje a helyi főtallózó szerepéért folytatott versenyt, függetlenül attól, hogy milyen más gépek vannak még a hálózatban:

```
[global]
    netbios name = HYDRA
    server string = Samba %v on (%L)
    workgroup = SIMPLE

    # A tallózó kiválasztás beállításai
    os level = 34
    local master = yes

    # A hálózat konfigurációs beállításai
    hosts allow = 192.168.220. 134.213.233. localhost
    hosts deny = 192.168.220.102
    interfaces = 192.168.220.100/255.255.255.0 \
                  134.213.233.110/255.255.255.0

    # Naplózás
    log level = 2
    log file = /var/log/samba.log.%m
    max log size = 50
    debug timestamp = yes

[data]
    path = /home/samba/data
    browseable = yes
```

```

guest ok = yes
comment = Data Drive
volume = Sample-Data-Drive
writable = yes

```

Mit kell tennünk azonban akkor, ha nem akarjuk mindig megnyerni a választást? Mit kell tennünk, ha azt szeretnénk, hogy egy Windows NT Server legyen a főtallózó, ha rákapcsolódik a hálózatra? Amint tudjuk, minden olyan gépnek, amely részt vesz a versengésben, információkat kell szétküldenie magáról. Ezek az információk a következőket tartalmazzák:

- a használt választási protokoll verziója;
- a gépen futó operációs rendszer;
- az ügyfél által a hálózaton eltöltött idő tartama;
- az ügyfél gazdaneve.

A verseny a következők szerint dől el. Az operációs rendszerekhez a verziójuktól függően egy-egy bináris érték tartozik az 5.1. táblázat szerint.

5.1. táblázat. Az operációs rendszerek értékei a választásban

Operációs rendszer	Érték
Windows NT Server 4.0	33
Windows NT Server 3.51	32
Windows NT Workstation 4.0	17
Windows NT Workstation 3.51	16
Windows 98	2
Windows 95	1
Windows 3.1 for Workgroups	1

Ezt követően a hálózathoz kapcsolódó mindegyik számítógéphez még egy másik érték is tartozik a hálózatban betöltött szerepüktől függően (lásd az 5.2. táblázatot).

5.2. táblázat. A számítógépek szerepe a választásban

Szerep	Érték
Elsődleges tartományvezérlő	128
WINS ügyfél	32
Preferált főtallózó	8
Aktív főtallózó	4
Készenléti tallózó	2
Aktív biztonsági tallózó	1

A választások a következő sorrendben történnek:

1. A választási protokoll legmagasabb számú verziója győz. (Ennek itt még nincs jelentősége, mert az összes Windows ügyfélnek 1-es verziójú a választási protokollja.)
2. A legnagyobb értékű operációs rendszert használó számítógép győz.
3. Ha egyezőség van, akkor a preferált főtallózó (8-as szerep) gép győz.
4. Ha továbbra is van egyezőség, a legrégebben a hálózaton lévő ügyfélgép győz.
5. Ha még mindig van egyezőség, akkor az a gép győz, amelyiknek a neve ábécé sorrendben megelőzi a többi gép nevét.
6. Végül a második helyezett gép lesz a tartaléktallózó.

Ha tehát azt szeretnénk, hogy a Samba legyen a helyi főtallózó, de csak abban az esetben, ha nincs a hálózaton Windows NT Server (4.0 vagy 3.51), akkor az előző példánkban az `os level` beállítás értékét a következők szerint módosítsuk:

```
os level = 31
```

Ennek hatására a Samba azonnal elveszíti a versenyt egy Windows NT 4.0 vagy Windows NT 3.5 Server géppel szemben, mert ezeknek magasabb szintű az operációs rendszerük. Másrészt viszont, ha azt akarjuk, hogy a helyi főtallózóért folytatott verseny a gépeknek a hálózaton betöltött szerepe szerint dőljön el, például aszerint, hogy melyikük az elsődleges tartományvezérlő, akkor az `os level` beállításhoz a hálózat legmagasabb szintű operációs rendszerét rendeljük, és a prioritási sorrendben szorítsuk a második helyre a választási protokollt.

Hogyan tudható meg egy gépről, hogy az helyi főtallózó? Ehhez az `nbtstat` parancsot kell kiadnunk. A parancs után az `-a` kapcsolót használva írjuk be a vizsgálandó gép nevét:

```
C:\>nbtstat -a hydra
```

#### NetBIOS Remote Machine Name Table

Name	Type	Status
-----	-----	-----
HYDRA	<00> UNIQUE	Registered
HYDRA	<03> UNIQUE	Registered
HYDRA	<20> UNIQUE	Registered
.._ _MSBROWSE_ _.	<01> GROUP	Registered
SIMPLE	<00> GROUP	Registered
SIMPLE	<1D> UNIQUE	Registered
SIMPLE	<1E> GROUP	Registered

```
MAC Address = 00-00-00-00-00-00
```

A keresett erőforrás-bejegyzés az `.._ _MSBROWSE_ _.<01>`. Ez jelzi, hogy a kiszolgáló az aktuális alhálózaton éppen a helyi főtallózó szerepét tölti be. Emellett ha ez a gép egyúttal egy Samba kiszolgáló is, az `nmdbd` naplófájlban is kereshetünk egy ehhez hasonló bejegyzést:

```

nmbd/nmbd_become_lmb.c:become_local_master_stage2(406)
*****
Samba name server HYDRA is now a local master browser for
workgroup SIMPLE on subnet 192.168.220.100
*****

```

Végül az elsődleges tartományvezérlő szerepét betöltő Windows NT kiszolgálók még egy lehetőséget kínálnak arra, hogy bizonyos körülmények között a helyi főtallózó szerepkörében mutakozzanak meg. Ezt az ún. *preferált főtallózó* bit teszi lehetővé. A Samba saját maga is be tudja állítani ezt a bitet, ha a preferred master beállítással engedélyezzük a számára:

```

# A tallózó kiválasztás beállításai
os level = 33
local master = yes
preferred master = yes

```

Ha a preferált főtallózó bit értéke 1, akkor a gép induláskor kikényszeríti a tallózó kiválasztását. Erre természetesen csak akkor van szükség, ha az os level beállításhoz egy Windows NT gép operációs rendszerét rendeltük. Ennek a beállításnak a használata nem ajánlott, ha más gépek is szerepelhetnek preferált főtallózóként, mint például egy NT kiszolgáló.

### *Tartomány-főtallózó*

Ez első fejezetben említettük, hogy ha azt akarjuk, hogy egy Windows munkacsoport vagy tartomány több alhálózatra legyen kiterjeszthető, akkor az egyik gépet *tartomány-főtallózóvá* kell tennünk. A tartomány-főtallózó adja tovább a tallózólistákat a munkacsoport alhálózataiban. Ezt azért teheti meg, mert az egyes helyi főtallózók rendszeres időközönként szinkronizálják a tallózólistájukat a tartomány-főtallózóval. A szinkronizálás során a helyi főtallózó mindazon kiszolgálóra elküldi a tallózólistáját, amelyet a tartomány-főtallózó nem tartalmaz, és fordítva. Így ha minden rendben működik, minden egyes helyi főtallózón rendelkezésre áll a teljes tartományra vonatkozó tallózólista.

A helyi főtallózótól eltérően itt nincs olyan versengés, ami eldöntené, hogy melyik gép töltsse be a tartomány-főtallózó szerepét. Ezt a rendszergazdának kézzel kell beállítania. A Microsoft-féle megvalósítás szerint azonban mind a tartomány-főtallózó, mind az elsődleges tartományvezérlő (PDC) <1B> típusú erőforrásként regisztrálja magát, vagyis a szerepek és a gépek nem választhatók el egymástól.

Ha van a hálózatunkban egy elsődleges tartományvezérlő szerepét betöltő Windows NT kiszolgáló, akkor az javasolható, hogy *ne* a Samba kiszolgáló legyen a tartomány-főtallózó. Ugyanez fordítva is igaz: ha a Samba venné át az elsődleges tartományvezérlő szerepét, akkor *ez* legyen a tartomány-főtallózó is. Bár a Samba esetében megosztható ez a két szerep, nem célszerű így tenni. Ha két különböző gépet használnánk elsődleges tartományvezérlőként és tartomány-főtallózóként, akkor egy Windows munkacsoportban ebből véletlenszerű hibák adódhatnak.

A Samba egy munkacsoport összes alhálózatában betöltheti a tartomány-főtallózó szerepét a következő beállítás megadásával:

```

domain master = yes

```

Ellenőrizhetjük, hogy ténylegesen a Samba gép a tartomány-főtallózó, ha megvizsgáljuk az *nmbd* naplófájl:

```
nmbd/nmbd_become_dmb.c:become_domain_master_stage2(118)
*****
Samba name server HYDRA is now a domain master browser for
workgroup SIMPLE on subnet 192.168.220.100
*****
```

Más eljárásként a Samba disztribúcióhoz tartozó *nmblookup* parancs segítségével is lekérdezhetjük a munkacsoportban lévő <1B> típusú erőforrást:

```
# nmblookup SIMPLE#1B
Sending queries to 192.168.220.255
192.168.220.100 SIMPLE<1b>
```

### *Több alhálózat*

Három szabályt kell szem előtt tartanunk, amikor egy munkacsoportot/tartományt egynél több alhálózatra terjesztünk ki:

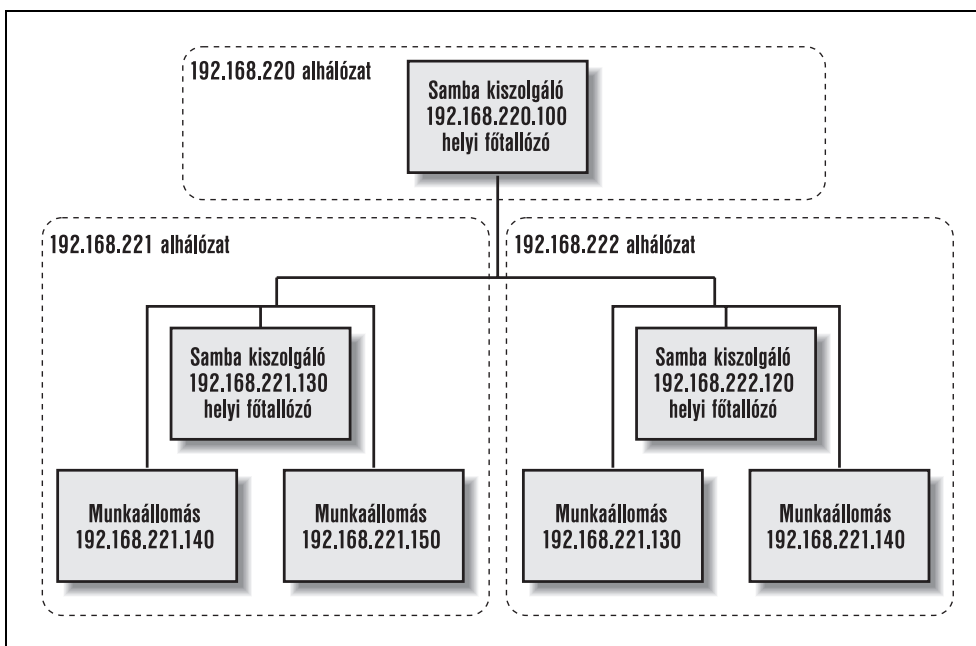
- Vagy egy Windows NT, vagy egy Samba gépnek helyi főtallózónak kell lennie a munkacsoport/tartomány minden egyes alhálózatában.
- Vagy egy Windows NT, vagy egy Samba gépnek tartomány-főtallózónak kell lennie valahol a munkacsoportban.
- Mindegyik helyi főtallózt utasítani kell arra, hogy szinkronizálja magát a tartomány-főtallózzal.

A Samba ebben az összefüggésben még néhány más képességgel is rendelkezik arra az esetre, ha nincs, vagy nem akarjuk, hogy legyen egy tartomány-főtallózó a hálózatban. Vegyük szemügyre az 5.1. ábrán látható alhálózatokat:

Először is egy helyi főtallózó szerepét betöltő Samba kiszolgáló használhatja a *remote announce* konfigurációs beállítást, amellyel biztosíthatja, hogy a különböző alhálózatokban lévő számítógépek megkapják a kiszolgálóról küldött broadcast üzeneteket. Ennek köszönhetően a Samba kiszolgáló az idegen alhálózatok tallózólistájában is megjelenik. Ehhez azonban az irányított broadcast üzeneteknek el kell jutniuk a többi alhálózat helyi főtallózójához. Tudjunk arról, hogy számos útválasztó alapbeállítás szerint nem engedélyezi az irányított broadcast üzeneteket; az útválasztón módosítani kell ezt a beállítást, hogy az üzenetek eljussanak az alhálózatába.

A *remote announce* beállításban soroljuk fel azokat az alhálózatokat és munkacsoportot, ahová a broadcast üzeneteknek el kell jutniuk. Ha például biztosítani akarjuk, hogy a Samba kiszolgálóról az üzenetek eljussanak a 192.168.221 és a 192.168.222 alhálózatokba és a SIMPLE munkacsoportba, akkor a következőképpen adjuk meg a beállítást:

```
# A tallózó kiválasztás beállításai
os level = 34
local master = yes
remote announce = 192.168.221.255/SIMPLE \
192.168.222.255/SIMPLE
```



5.1. ábra. Samba kiszolgálókat tartalmazó több alhálózat

Az üzenetek címzettjének pontos címét is megadhatjuk, ha az idegen hálózatban a helyi főtallózónak garantáltan mindig rögzített az IP címe.

Egy Samba helyi főtallózó a tallózólistáját közvetlenül szinkronizálhatja egy másik Samba kiszolgálóval, amely helyi főtallózóként működik egy másik alhálózatban. Tegyük fel például, hogy a Samba helyi főtallózóként van konfigurálva, és vannak Samba helyi főtallózók a 192.168.221.130 és a 192.168.222.120 alhálózatokban. Ekkor a remote browse sync beállítás segítségével közvetlenül szinkronizálhatók a Samba kiszolgálók a következők szerint:

```
# A tallózó kiválasztás beállításai
os level = 34
local master = yes
remote browse sync = 192.168.221.130 192.168.222.120
```

Ahhoz, hogy ez a beállítás működjön, a többi Samba gépnek is helyi főtallózónak kell lennie. Ezzel a beállítással irányított broadcast üzenetek is küldhetők, ha nem ismerjük a helyi főtallózók IP címét.

### Tallózási beállítások

Az 5.3. táblázat 14 olyan beállítást sorol fel, amelyek segítségével meghatározhatjuk, miként kezelje a Samba a tallózási feladatokat. Javasolható az alapértelmezett értékek megtartása, ha meg akarjuk könnyíteni a felhasználóknak a megosztások és a nyomtatók megtalálását.

5.3. táblázat. A tallózás konfigurációs beállításai

Beállítás	Paraméterek	Funkció	Alapértelmezett érték	Hatókör
announce as	NT vagy Win95 vagy WfW	Megadja, hogy a Samba milyen operációs rendszerként jelentse be magát.	NT	Globális
announce version	Numerikus	Megadja annak az operációs rendszernek a verziószámát, amelyen rendszerként a Samba bejelentette magát.	4.2	Globális
browsable (browsable)	Boolean érték	Engedélyezi a megosztás megjelentését a gépi erőforrások listájában.	yes	Megosztás
browse list	Boolean érték	Ha yes az értéke, a Samba megjeleníti azon a kiszolgálón a tallózólistát.	yes	Globális
auto services (preload)	Karakterlánc (megosztások listája)	Megadja azoknak a megosztásoknak a listáját, amelyek mindig megjelennek a tallózólistában.	nincs	Globális
default service (default)	Karakterlánc (megosztás neve)	Megnevezi azt a megosztást (szolgáltatást), ami akkor jelenik meg, ha az ügyfél egy, az <i>smb.conf</i> fájlban nem felsorolt megosztást kér.	nincs	Globális
local master	Boolean érték	Ha yes az értéke, a Samba megpróbál főtallózóvá válni a helyi alhálózatban.	yes	Globális
lm announce	yes, no vagy auto	Engedélyezi vagy tiltja a LAN Manager stílusú gazda-bejelentkezést.	auto	Globális
lm interval	Numerikus	Megadja másodpercekben azt a gyakoriságot, amellyel a LAN Manager bejelentkezés megtörténhet, ha aktív.	60	Globális
preferred master (preferred master)	Boolean érték	Ha yes az értéke, a Samba a preferált főtallózó bitet használja arra, hogy helyi főtallózó lehessen.	no	Globális
domain master	Boolean érték	Ha yes az értéke, a Samba megpróbál főtallózó lenni a munkacsoportban.	no	Globális
os level	Numerikus	Megadja a Samba operációs rendszerének szintjét a helyi főtallózóvá való választásához.	0	Globális

## 5.3. táblázat folytatása

Beállítás	Paraméterek	Funkció	Alapértelmezett érték	Hatókör
remote browse sync	Karakterlánc (IP címek listája)	Felsorolja azokat a Samba kiszolgálókat, amelyekkel szinkronizálni kell a tállózólistákat.	nincs	Globális
remote announce	Karakterlánc (IP cím / munkacsoport párok)	Felsorolja azokat az alhálózatokat és munkacsoportokat, amelyekre el kell küldeni az irányított broadcast csomagokat, hogy a Samba megjelenhessen a tállózólistákban.	nincs	Globális

**announce as**

Ez a globális konfigurációs beállítás azt az operációs rendszert jelzi, amelyet a Samba a hálózaton lévő több gép számára önmagáról bejelent. A beállítás alapértelmezés szerinti értéke NT, ami a Windows NT operációs rendszert jelenti. Más lehetséges érték a Win95, ami a Windows 95, és a wfw, ami a Windows for Workgroups operációs rendszert jelenti. Az alapértelmezés szerinti értéket az alábbi módon írhatjuk felül:

```
[global]
    announce as = Win95
```

Ennek a konfigurációs beállításnak nem javasolt a módosítása.

**announce version**

Ezt az ugyancsak globális hatókörű beállítást gyakran az announce as beállítással együtt használják; ez annak az operációs rendszernek a verziószámát adja meg, amelyet a Samba a hálózaton lévő többi gép számára önmagáról jelentett. A beállítás alapértelmezett értéke 4.2, ami az aktuális Windows NT 4.0-s verziójánál magasabb szintet jelent. A beállításhoz ennél nagyobb értéket is rendelhetünk:

```
[global]
    announce version = 4.3
```

Ebben az esetben sem javasoljuk az alapértelmezés szerinti érték megváltoztatását.

**browseable**

A browseable beállítás (ami browsable formában is írható) segítségével azt írhatjuk elő, hogy a hivatkozott megosztás megjelenjen-e annak a gépnek az elérhető erőforrásai között, amelyen maga a megosztás található. Alapértelmezés szerint ennek a beállításnak mindig yes az értéke. Ha azt szeretnénk, hogy egy megosztás ne jelenjen meg egy ügyfél tállózójában, akkor a no értéket rendeljük hozzá.

Jegyezzük meg, hogy ezzel nem akadályozhatjuk meg azt, hogy valaki más eszközökkel, például a Windows Intézőjében egy UNC helymegadással (`\\server\accounting`) ne férhessen hozzá a megosztáshoz. A beállítással mindössze azt tilthatjuk meg, hogy tallózáskor a megosztás ne jelenjen meg a gép erőforrásai között.

#### *browse list*

Sohasem lesz szükségünk arra, hogy ennek a beállításnak az alapértelmezett `yes` értékét megváltoztassuk. Ha a Samba kiszolgáló szerepel helyi főtallózóként (vagyis ez nyerte meg a választási versenyt), akkor a globális `browse list` beállítás segítségével utasíthatjuk arra a Sambát, hogy az ügyfelek rendelkezésére bocsássa-e a tallózólistát, vagy sem. Alapbeállítás szerint a Samba mindig felkínálja a tallózólistát. Ezt az alapbeállítást az alábbi módon változtathatjuk meg:

```
[global]
    browse list = no
```

Ha letiltjuk a tallózólista megjelenítését, akkor az ügyfelek nem tallózzhatják a hálózathoz aktuálisan kapcsolódó többi gép nevét, a szolgáltatásaikat és a tartományokat. Jegyezzük meg azonban, hogy ezzel egyetlen gépet sem tehetünk elérhetetlenné: ha valaki ismeri egy számítógép érvényes nevét/címét, és egy adott megosztást az illető gépen, akkor a NET USE segítségével vagy a Windows Intézőjében egy meghajtó-azonosító leképztetésével kapcsolódhat hozzá. A beállítással mindössze azt akadályozhatjuk meg, hogy az ügyfelek előtt megjelenjen a tallózólista.

#### *auto services*

A globális `auto services` beállítás, ami `preload` néven is ismeretes, biztosítja, hogy a megadott megosztások mindig megjelenjenek a tallózólistában. Ezt a beállítást gyakran használják arra, hogy a `[homes]` vagy a `[printers]` megosztásokkal létrehozott meghatározott felhasználói vagy nyomtatómegosztások megjelenjenek, de egyéb módon ne legyenek tallózhatók.

A beállítás leginkább lemezmegosztásokkal együtt használható. Ha azt szeretnénk, hogy ezt a beállítást használva a rendszernyomtatók (vagyis a nyomtatóbeállításokat tartalmazó fájlban felsorolt nyomtatók) megjelenjenek a tallózólistában, akkor ehelyett a `load printers` beállítást használjuk. Az `auto services` beállításban felsorolt egyik megosztás sem listázódik ki, ha a `browse list` beállításhoz a `no` érték van rendelve.

#### *default service*

A globális `default service` beállítással (amelynek esetenként `default` a neve) az „utolsó mentsvárnak” tekinthető megosztás adható meg. Ha létező megosztás neve van ehhez a beállításhoz rendelve, és egy ügyfél egy nem létező lemez- vagy nyomtatómegosztáshoz akar hozzáférni, akkor a Samba ehelyett az itt megnevezett megosztáshoz hozza létre a kapcsolatot. A beállítás a következőképpen használható:

```
default service = helpshare
```

Figyeljük meg, hogy a megosztás `helpshare` neve nem áll szögletes zárójelek között, jöllehet a Samba konfigurációs fájljában a definiálásakor alkalmazni kell ezt a zárójelezést.

Továbbá ha ebben a beállításban megadott megosztásban a %S változót használjuk, a változó a kért, nem létező megosztást, nem pedig az alapértelmezett szolgáltatást fogja jelenteni. Emellett a %S változó azt is lehetővé teszi, hogy a kért megosztásban lévő aláhúzás-jelek (\_) törtvonallá (/) alakuljanak át.

### *local master*

Ezzel a globális beállítással azt írhatjuk elő, hogy vegyen-e részt a Samba a helyi főtallózó-ért folytatott versenyben. Ha a beállításhoz a *yes* értéket rendeljük, a Samba részt vesz a versenyben. Ez azonban önmagában még nem garantálja a győzelmet. (Más beállítások, így a *preferred master* és az *os level* segíthetik a Samba győzelmét.) Ha a beállításhoz a *no* értéket rendeljük, a Samba minden versenyben alulmarad, függetlenül attól, hogy milyen értéket rendelünk a többi beállításhoz. A beállítás alapértelmezett értéke a *yes*.

### *lm announce*

Az *lm announce* globális beállítás azt közli a Samba *nmbd* démonjával, hogy küldjön-e LAN Manager gazdabejelentkezéseket a kiszolgáló megbízásából vagy sem. Ilyen gazdabejelentkezésekre a régebbi ügyfélgépeknek, például az IBM OS/2 operációs rendszerét futtató gépeknek lehet szükségük. Ez a bejelentkezés lehetővé teszi, hogy a kiszolgáló bekerüljön az ügyfélgép tallózólistájába. Ha a beállításnak *yes* az értéke, a Samba az *lm interval* beállításban megadott gyakorisággal, ismétlődően bejelentkezik.

A beállítás a standard Boolean értékeket, vagyis a *yes* vagy a *no* értéket veheti fel, és így vagy engedélyezi, vagy tiltja a LAN Manager bejelentkezéseket. Ezen alapértékek mellett azonban még egy harmadik, *auto* értéket is felvehet, mely esetben az *nmbd* passzívan figyel a LAN Manager bejelentkezésekre, de a maga részéről nem kezdeményez ilyeneket. Ha a hálózaton lévő más gépek észlelik a LAN Manager bejelentkezéseket, az *nmbd* a maga részéről is megkezdí a LAN Manager bejelentkezések küldését, hogy biztosítsa a láthatóságát. A beállítást az alábbi módon adhatjuk meg:

```
[global]
lm announce = yes
```

A beállítás alapértelmezett értéke az *auto*. Feltehetően sohasem kell megváltoztatnunk ezt az értéket.

### *lm interval*

Ezzel az *lm announce* beállítással együtt használandó beállítással azon időtartam másodpercekben mért hosszát adhatjuk meg, ameddig az *nmbd* arra vár, hogy újra kiküldje a LAN Manager-féle bejelentkezéseket. Ne feledjük, hogy ennek a beállításnak csak akkor van értelme, ha előzőleg aktivizáltuk a LAN Manager bejelentkezéseket. Az időtartam alapbeállítás szerinti hossza 60 másodperc. Ha 0 értéket adunk meg, a Samba semmilyen LAN Manager bejelentkezést nem küld ki, bármilyen értéket tartalmazzon is az *lm announce* beállítás. Az alapbeállítás a következőképpen változtatható meg:

```
[global]
lm interval = 90
```

### *preferred master*

A *preferred master* beállítás igényli, hogy a Samba 1-es értékre állítsa a preferált mesterbit értékét, amikor részt vesz a tallózásért folyó versenyben. Ezzel a kiszolgáló preferált helyzetbe kerül a munkacsoport többi, azonos szintű operációs rendszert futtató gépéhez képest. Ha a Samba gépet úgy konfiguráljuk, hogy az legyen a helyi főtallózó, akkor célszerű az alábbi beállítás használata:

```
[global]
    preferred master = yes
```

Más esetben meghagyhatjuk a beállítás alapértelmezett *no* értékét. Ha a Samba preferált főtallózóként van konfigurálva, akkor a hálózatra kapcsolódásakor kikényszeríti a választási versenyt.

### *os level*

Az *os level* globális beállítással azt adhatjuk meg, hogy a Samba milyen operációs rendszert jelentsen be magáról a tallózó kiválasztásakor. Ha azt szeretnénk, hogy a Samba megnyerje a választást és főtallózóvá váljon, akkor a hálózatban részt vevő operációs rendszerek legmagasabb értékénél nagyobb értéket állítsunk be. Az alapértelmezett értékeket az 5.1. táblázat tartalmazza. A beállítás alapértelmezett értéke 0, ami azt jelenti, hogy a Samba az összes választást elveszti. Ha azt szeretnénk, hogy a Samba minden választást megnyerjen, akkor a következő beállítást válasszuk:

```
os level = 34
```

Ez azt jelenti, hogy a kiszolgáló minden egyes választáskor 34-szer fog magára szavazni, ami biztosítja a győzelmét.

### *domain master*

Ha a munkacsoportban vagy az NT tartományban a Samba az elsődleges tartományvezérlő, akkor célszerűen ez legyen a tartomány-főtallózó is. Az a gép a tartomány-főtallózó, amelyiknek <1B> a NetBIOS erőforrástípusa, és amely a tartomány alhálózataiban lévő helyi főtallózók felé továbbítja, illetve azoktól megkapja a tallózólistákat. Ha ki szeretnénk kényszeríteni, hogy a Samba tartomány-főtallózó legyen, akkor az *smb.conf* fájl *[global]* szakaszába vegyük fel a következő bejegyzést:

```
[global]
    domain master = yes
```

Ha a hálózatban van olyan Windows NT gép, amely az elsődleges tartományvezérlő (PDC) szerepét tölti be, akkor nem ajánlható, hogy a Samba legyen a tartomány-főtallózó. Ennek fordítottja is igaz: Ha a Samba az elsődleges tartományvezérlő, akkor ez legyen a tartomány-főtallózó is. Ha két különböző gépre osztjuk szét az elsődleges tartományvezérlő és a tartomány-főtallózó szerepét, akkor ebből előre nem látható problémák adódhatnak a hálózaton.

*remote browse sync*

A globális `remote browse sync` beállítással azt írhatjuk elő, hogy a Samba szinkronizálja a tállózólistáit a többi alhálózat helyi főtállózóival. Ez a szinkronizálás azonban csak Samba kiszolgálókkal végezhető el, Windows számítógépekkel nem. Ha például egy Samba kiszolgáló főtállózó van a 192.168.235 alhálózatban, és a 192.168.234.92, valamint a 192.168.236.2 alhálózatokban Samba helyi főtállózók vannak, akkor a következő beállítást használhatjuk:

```
remote browse sync = 192.168.234.92 192.168.236.2
```

Ebben az esetben a Samba kiszolgáló közvetlenül kapcsolatba léphet a címlistában megadott gépekkel, és elvégezheti a tállózólisták szinkronizálását. A következő beállítást is használhatjuk:

```
remote browse sync = 192.168.234.255 192.168.236.255
```

Ezzel arra kényszerítjük a Sambát, hogy broadcast lekérdezéssel állapítsa meg az alhálózatokban lévő helyi főtállózók IP címét, és azokkal szinkronizálja a tállózólistákat. Ez a beállítás azonban csak akkor használható, ha az útválasztó nem tiltja a 255-tel végződő irányított broadcast lekérdezést.

*remote announce*

A Samba kiszolgálók képesek arra, hogy a `remote announce` beállítás segítségével küldjék el a tállózólistákat az idegen alhálózatoknak. Ezeket tipikusan az alhálózatok helyi főtállózójának küldik meg. Ha viszont nem ismerjük a helyi főtállózók címét, akkor a következőt tehetjük:

```
[global]
remote announce = 192.168.234.255/ACCOUNTING \
                  192.168.236.255/ACCOUNTING
```

Ebben az esetben a Samba a gazdabejelentkezéseket a 192.168.234 és a 192.168.236 alhálózatok összes gépeire szétküldi, amelyek remélhetőleg az illető alhálózat helyi főtállózóját is elérik. Természetesen a pontos IP címeket is megadhatjuk, ha ismerjük ezeket.

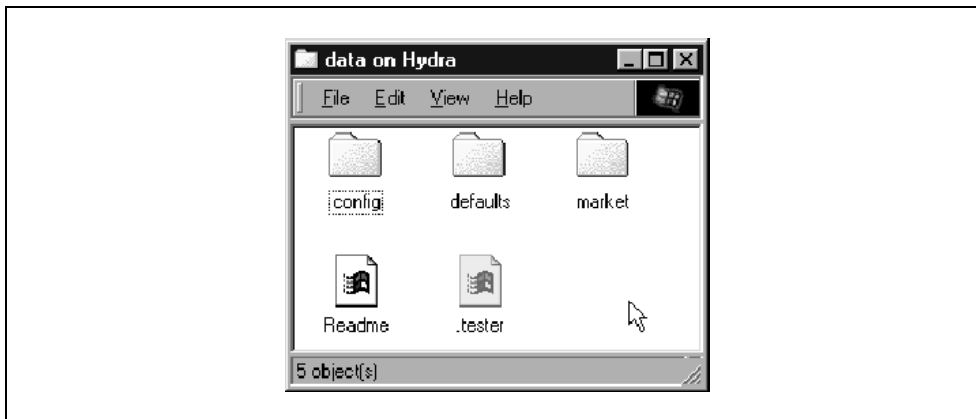
## *Fájlrendszerbeli különbségek*

A Samba egyik legfontosabb feladata, hogy kiküszöbölje a Unix és a nem Unix fájlrendszerek közötti különbségeket. Ide tartoznak többek között a szimbolikus csatolásokkal, a rejtett és az ún. *dot fájl*okkal kapcsolatos problémák megoldása. Ugyancsak gondot jelenthetnek a nem megfelelőképpen kezelt fájlengedélyek. A fejezet e részében arról olvashatunk, hogy miként hidalja át a Samba ezeket a bosszantó eltéréseket, továbbá milyen újdonságokkal egészíti ki a fájlrendszereket.

### Fájlok elrejtése és vétóztatása

Vannak olyan esetek, amikor biztosítani akarjuk, hogy egy felhasználó egyáltalán ne lásson egy fájlt, és ne férhessen hozzá. Más esetekben viszont csak azt szeretnénk, hogy amikor belenéz egy könyvtárba, rejtettként jelenjen meg benne egy fájl. A Windows operációs rendszerekben a fájlok egyik attribútumának beállításával rejthetünk el egy fájlt egy mappában. A Unix rendszerekben hagyományosan a fájl elé helyezett pont, azaz dot (.) segítségével rejthetünk el egy fájlt. Ezzel megelőzhetjük, hogy egy szokásos `ls` parancs kiadásakor kilistázódjanak az olyan tételek, mint például a konfigurációs fájlok. Ha viszont egy fájl mindenképpen meg akarunk óvni attól, hogy a felhasználók hozzáférhessenek, akkor a fájlokkal és a könyvtárakkal kapcsolatos engedélyeket kell ennek megfelelően beállítanunk.

Elsőként tekintsünk a `hide dot files` (dot fájlok elrejtése) beállítást. Ha ehhez a `yes` értéket rendeljük, akkor a ponttal kezdődő fájlok rejtettként jelennek meg. Ha beállítás értéke `no`, akkor ezek a fájlok is kilistázódnak. Itt fel kell hívnunk a figyelmet arra, hogy ezek a fájlok csak rejtettek. Ha a felhasználó a tallózáskor azt a lehetőséget választja, hogy a rejtett fájlok is megjelenjenek (vagyis a Windows 98-ban a Nézet menü Mappa beállításai párbeszédablakában bejelöli az ide vonatkozó választógombot), akkor ezeket a fájlokat is láthatja (lásd az 5.2. ábrát).

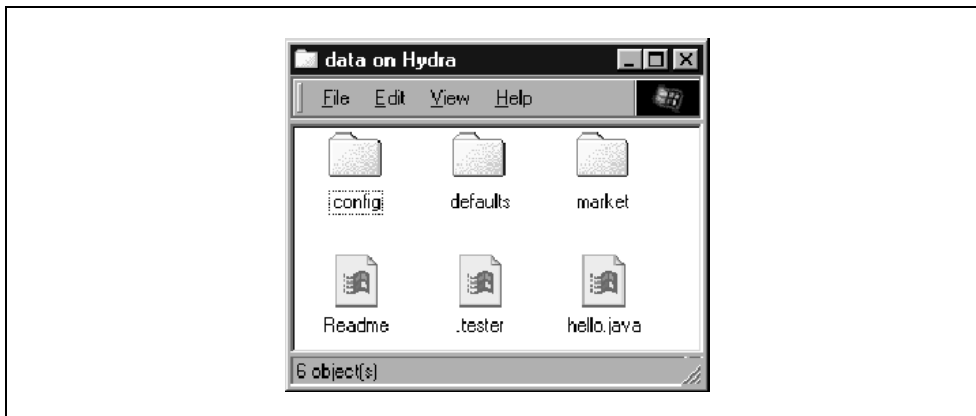


5.2. ábra. Rejtett fájlok a [data] megosztásban

Ahelyett, hogy egyszerűen csak a ponttal kezdődő fájlokat elrejtjük, a `hide files` beállítás segítségével arra is utasíthatjuk a Sambát, hogy egy adott karakterlánc-mintát használjon a fájlok elrejtéséhez. Tegyük fel például, hogy az alábbi bejegyzéseket helyeztük el egy [data] megosztásba:

```
[data]
path = /home/samba/data
browseable = yes
guest ok = yes
writeable = yes
case sensitive = no
hide files = /*.java/*README*/
```

A beállításban felsorolt minden egyes elemnek törtvonallal (/) kell kezdődnie és végződnie, még akkor is, ha a lista egyetlen elemből áll. Ez a jelölésmód lehetővé teszi, hogy szóközők is legyenek a fájlnevekben. A példabeli beállítást megadva az 5.3. ábrán látható módon jelenne meg a megosztott könyvtár. A Windows 98-ban ebben az esetben is engedélyeztük a rejtett fájlok megjelenítését.



5.3. ábra. Fájlok elrejtése fájlnev-minták alapján

Ha azt szeretnénk, hogy a felhasználók egyáltalán ne lássanak bizonyos fájlokat, akkor a veto files beállítást kell használnunk. Ebben a beállításban, amelynek ugyanaz a szintaxisa, mint a hide files beállításé, soroljuk fel azokat a fájlokat, amelyeket semmikor sem akarunk láttatni a felhasználókkal. Ehhez a [data] megosztást például a következőképpen módosíthatjuk:

```
[data]
path = /home/samba/data
browseable = yes
guest ok = yes
writeable = yes
case sensitive = no
veto files = /*.java/*README*/
```

Ennek a beállításnak ugyanolyan a szintaxisa, mint a hide files beállításé: a lista mindegyik elemének törtvonallal (/) kell kezdődnie és végződnie, még akkor is, ha a lista csak egyetlen elemet tartalmaz. A példában bemutatott beállítás következtében a hello.java és a README fájlok egyszerűen eltűnnek a könyvtárból, és a felhasználó nem lesz képes arra, hogy SMB-n keresztül hozzájuk férjen.

A fentiekkel kapcsolatban választ kell adni még egy kérdésre. Mi történik akkor, ha egy felhasználó vétózott fájlokat tartalmazó könyvtárt próbál törölni? Ekkor jut szerephez a delete veto files beállítás. Ha ehhez a beállításhoz a yes érték van rendelve, akkor a felhasználó a könyvtárban lévő normál és vétózott fájlokat is törölheti, és ezzel maga a könyvtár is törlődik. Ha viszont a beállításhoz a no érték van rendelve, akkor a felhasználó

nem törölheti a vétózott fájlokat, következésképpen a könyvtár sem törölhető. A felhasználó számára úgy jelenik meg a könyvtár, mint ami üres, ám ennek ellenére nem törölhető.

A `dont descend` (lejjebb nem) direktívához felsorolt könyvtárak tartalmát a Samba nem engedi megjeleníteni. Jegyezzük meg, hogy a könyvtárak tartalmáról van szó, és nem magukról a könyvtárakról. A felhasználó beléphet egy ilyen módon megjelölt könyvtárba, de a könyvtárfában lejjebb, az alsóbb szintekre már nem léphet. Az ilyen könyvtár mindig üres könyvtárként jelenik meg. Használjuk ezt a beállítást például egy, a fejezetben már korábban megismert megosztásban:

```
[data]
    path = /home/samba/data
    browseable = yes
    guest ok = yes
    writeable = yes
    case sensitive = no
    dont descend = config defaults
```

Ehhez tegyük még fel, hogy a `/home/samba/data` könyvtárnak a következő a tartalma:

```
drwxr-xr-x  6 tom      users      1024 Jun 13 09:24 .
drwxr-xr-x  8 root     root        1024 Jun 10 17:53 ..
-rw-r--r--  2 tom      users      1024 Jun  9 11:43 README
drwxr-xr-x  3 tom      users      1024 Jun 13 09:28 config
drwxr-xr-x  3 tom      users      1024 Jun 13 09:28 defaults
drwxr-xr-x  3 tom      users      1024 Jun 13 09:28 market
```

Ha egy felhasználó kapcsolódik ehhez a megosztáshoz, akkor az 5.4. ábrán látható könyvtárakat látná. A `/config` és a `/defaults` könyvtárak üresként jelennének meg, még akkor is, ha más fájlok vagy könyvtárak lennének benne. A felhasználó írni sem tud az ilyen könyvtárakba (amivel megelőzhető az, hogy egy már létező, de számára láthatatlan fájlal vagy mappával azonos nevű fájl vagy mappát hozzon létre). Ha megpróbálkozna ezzel, akkor az „Access Denied” (Hozzáférés megtagadva) üzenetet kapná. A `dont descend` inkább adminisztratív jellegű beállítás, semmint biztonsági intézkedés, és nem helyettesítheti a fájlengedélyekkel kapcsolatos beállításokat.

## Hivatkozások

A DOS és az NT fájlrendszerek nem használnak szimbolikus hivatkozásokat, míg a Windows 95/98 „parancsikonokat” (shortcuts) használ erre a célra. Ezért amikor egy felhasználó a Samba kiszolgáló egyik megosztásán megpróbál megnyitni egy szimbolikus hivatkozást, a Samba megpróbálja követni a hivatkozást, hogy megtalálja a tényleges fájlt, és megnyissa azt, ugyanúgy, mintha a felhasználó Unix gépen dolgozna. Ha nem szeretnénk ezt engedélyezni, akkor a következőképpen használjuk a `follow symlinks` beállítást:



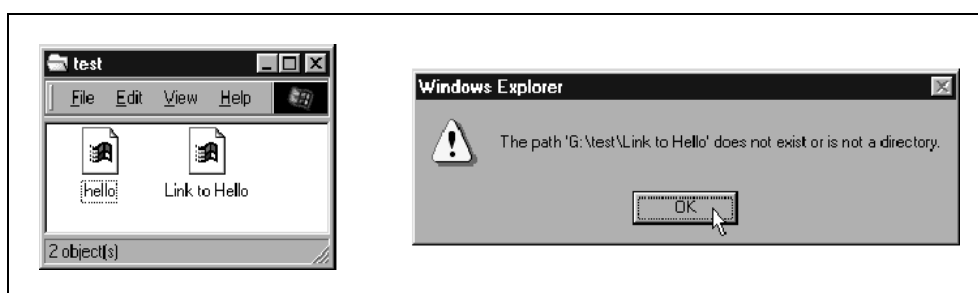
5.4. ábra. A [data] megosztás tartalma dont descend beállítással

```
[data]
path = /home/samba/data
browseable = yes
guest ok = yes
writeable = yes
case sensitive = no
follow symlinks = no
```

A beállítást azzal tesztelhetjük, hogy a Unix kiszolgálón létrehozunk egy könyvtárat azon a megosztáson belül, amelybe a felhasználó bejelentkezik. Írjuk be a következő parancsokat:

```
% mkdir hello; cd hello
% cat "This is a test" >hello.txt
% ln -s hello.txt "Link to hello"
```

Eredményül az 5.5. ábrán látható két fájlt kapjuk. Normál esetben, ha e két fájl bármelyikére rákattintunk, a „This is a test” szöveget tartalmazó fájlt kapjuk meg. Ha viszont a `follow symlinks` beállításhoz a `no` érték van rendelve, akkor az 5.5. ábra jobb oldalán láthatóhoz hasonló hibaüzenetet kapjuk, ha a „Link to hello” fájlra kattintunk.



5.5. ábra. Hibaüzenet jelenik meg, ha a Samba egy tiltott `follow symlinks` hivatkozást próbál meg követni

Végezetül essen szó a `wide links` beállításról. Ha ehhez a beállításhoz a `yes` érték tartozik, akkor a felhasználó a megosztott könyvtárfán kívüli helyekre mutató szimbolikus hivatkozásokat is követheti, beleértve a hivatkozás másik oldalán lévő fájlokat és könyvtárakat is. Tegyük fel például, hogy a `[data]` megosztást a következők szerint módosítottuk:

```
[data]
path = /home/samba/data
browseable = yes
guest ok = yes
writeable = yes
case sensitive = no
follow symlinks = yes
wide links = yes
```

Addig, amíg engedélyezve van a `follow symlinks` beállítás, a Samba képes követni az aktuális megosztásán kívüli szimbolikus hivatkozásokat is. Ha például a megosztáson kívül hoznánk létre egy fájlt (például valakinek a home könyvtárában), majd a megosztásban az alábbiak szerint készítenénk hozzá egy hivatkozást:

```
ln -s ~/tom/datafile ./datafile
```

akkor Tom könyvtárában a célfájlra vonatkozó engedélyek alapján meg tudnánk nyitni a fájlt.

### Fájlrendszeri beállítások

Az 5.4. táblázat az eddigiekben tárgyalt beállításokat sorolja fel. Célszerű, ha a legtöbbjük esetében megtartjuk az alapbeállítás szerinti értéket, kivéve a következőkben bemutatásra kerülő beállításokat.

5.4. táblázat. A fájlrendszer konfigurációs beállításai

Beállítás	Paraméterek	Funkció	Alapértelmezett érték	Hatókör
<code>unix realname</code>	Boolean érték	A Unix felhasználó teljes nevét bocsátja az ügyfél rendelkezésére.	no	Globális
<code>dont descend</code>	Karakterlánc (könyvtárlista)	Megadja azoknak a könyvtáraknak a listáját, amelyek tartalmát a Sambának az ügyfelek elől el kell rejtene.	Nincs	Megosztás
<code>follow symlinks</code>	Boolean érték	Ha no az értéke, a Samba nem veszi figyelembe a szimbolikus hivatkozásokat.	yes	Megosztás
<code>getwd cache</code>	Boolean érték	Ha yes az értéke, a Samba gyorsítást használ a <code>getwd()</code> függvényhívásokhoz.	yes	Globális

## 5.4. táblázat folytatása

Beállítás	Paraméterek	Funkció	Alapértelmezett érték	Hatókör
wide links	Boolean érték	Ha <b>yes</b> az értéke, a Samba a megosztáson kívül is követi a szimbolikus hivatkozásokat.	<b>yes</b>	Megosztás
hide dot files	Boolean érték	Ha <b>yes</b> az értéke, a Samba úgy kezeli a Unix rejtett fájljait, mint a Windows rejtett fájljait.	<b>yes</b>	Megosztás
hide files	Karakterlánc (fájllista)	Felsorolja azokat a fájlmin-tákat, amelyeket a Sambának rejtettként kell kezelnie.	nincs	Megosztás
veto files	Karakterlánc (fájllista)	Felsorolja azokat a fájlmin-tákat, amelyeket a Samba sohasem jelenít meg.	nincs	Megosztás
delete veto files	Boolean érték	Ha <b>yes</b> az értéke, törli a <b>veto files</b> beállításban megadott fájlokat, ha a felhasználó törli az ezeket a fájlokat tartalmazó könyvtárat.	<b>no</b>	Megosztás

*unix realname*

Egyes programok a megfelelő működésükhöz a teljes felhasználónevet igénylik. Így például egy windowsos elektronikus levelezőprogramnak gyakran össze kell kapcsolnia egy felhasználónevet a valós névvel. Ha a rendszerünk jelszófájlja a GCOS mezőben a felhasználók teljes nevét tartalmazza, akkor a `unix realname` beállítás arra utasítja a Sambát, hogy szolgáltatssa ezt az információt az ügyfeleknek. Ha ezt nem biztosítanánk, akkor a felhasználó neve azonos lenne a bejelentkezési nevével. Ha például a Unix jelszófájlja tartalmazza a következő sort:

```
rcollins:/KaBfco47Rer5:500:500:Robert Collins:
/home/rcollins:/bin/ksh
```

és a konfigurációs fájl a következő beállítást tartalmazza:

```
[global]
    unix realname = yes
```

akkor a Samba mindazon ügyfél számára szolgáltatja az `rcollins` felhasználó valós, Robert Collins nevét, aki ezt kéri. Ezzel a beállítással általában nem kell törődnünk.

*dont descend*

A `dont descend` (lejjebb nem) beállítás segítségével megadhatjuk azokat a könyvtárakat, amelyeket az ügyfelek előtt üresként akarunk megjeleníteni. A Samba az ilyen könyvtárak

tartalmát nem jeleníti meg. Ez azonban nem a legmegfelelőbb eljárás a fájlok megvédésére (a felhasználó találhat megoldást ennek a tiltásnak a megkerülésére), és valójában csak arra jó, hogy ne engedjük tallózni a felhasználót az érzékeny fájlokat tartalmazó könyvtárakban. Lásd a fejezetrész korábbi példáját.

### *follow symlinks*

Ezzel a beállítással, amelyről korábban már bővebben olvashattunk, az szabályozható, hogy kövessen-e a Samba egy szimbolikus hivatkozást a Unix operációs rendszerben a célíg, vagy hibaüzenettel térjen-e vissza a felhasználóhoz. Ha a beállításhoz a `yes` érték tartozik, akkor a Samba fájlként jeleníti meg a hivatkozási célt.

### *getwd cache*

Ezzel a globális beállítással azt írhatjuk elő, hogy a Samba használjon-e helyi gyorsítást a Unix `getwd()` (get current working directory, aktuális munkakönyvtár bekérése) rendszerfüggvényének meghívásához. A beállítás alapértelmezés szerinti `yes` értékét a következőképpen írhatjuk felül:

```
[global]
    getwd cache = no
```

Ha a beállításhoz a `yes` értéket rendeljük, jelentős mértékben megnöveljük a munkakönyvtár feloldásához szükséges időt, különösen akkor, ha a `wide links` beállításnak `no` az értéke. Normál esetben nem kell változtatnunk a beállítás alapértelmezés szerinti értéken.

### *wide links*

Ezzel a beállítással azt adhatjuk meg, hogy a felhasználó követhet-e olyan szimbolikus hivatkozásokat, amelyek a megosztott könyvtárfán kívüli helyekre mutatnak. Ezek a helyek a hivatkozás végén álló bármely fájl vagy könyvtárat jelenthetik, amennyiben a felhasználó rendelkezik a megfelelő hozzáférési engedélyekkel. A beállításnak alapértelmezés szerint `yes` az értéke. Jegyezzük meg, hogy ez a beállítás figyelmen kívül marad, ha a `follow symlinks` beállításhoz a `no` érték van rendelve. Utóbbi esetben lényegesen lelassul az *smbd* futása.

### *hide files*

A `hide files` beállításban egy vagy több könyvtár vagy fájl nevének mintáját sorolhatjuk fel. A Samba mindazokat a könyvtárakat és fájlokat, amelyek neve megfelel az adott mintá(k)nak, rejtettként kezeli, és elrejtí a felhasználók elől. Jegyezzük meg azonban, hogy ez nem jelent többet, mint a DOS rejtett attribútumának bekapcsolását, és semmiképpen sem biztosítja, hogy egy felhasználó a tallózás során ne láthassa ezeket.

A lista mindegyik elemének törtjellel (/) kell kezdődnie és végződnie, még akkor is, ha a lista csak egyetlen elemet tartalmaz. Ez lehetővé teszi, hogy szóközöket is tartalmazzanak a listaelemek. Általános helyettesítőként csillag karakterek használhatók 0 vagy több karakter helyett. Minden egyes kérdőjel pontosan egy karaktert helyettesít. Példa:

```
hide files = /.jav*/README.??*/
```

*hide dot files*

A `hide dot files` beállítás segítségével a kiszolgálón lévő, pont (.) karakterrel kezdődő fájlok rejtethetők el. A `hide files` beállításhoz hasonlóan a ponttal kezdődő fájloknak be lesz kapcsolva a DOS rejtett attribútumuk, ami semmilyen garanciát sem nyújt arra, hogy ezek a fájlok nem lesznek megjeleníthetők. A beállításnak alapértelmezés szerint `yes` az értéke.

*veto files*

A fájlok elrejtéséhez képest még nagyobb mértékben eltitkolhatjuk a fájljainkat a `veto files` beállítás segítségével. E beállítást használva a Samba még annyit sem hajlandó elárulni, hogy egyáltalán léteznek a fájlok. A valóságban azonban ez nem jelent igazi védelmet, mindössze egy mechanizmus, amely nem engedi, hogy a PC-s programok töröljék a Macintosh fájlok Unix fájlrendszerben való tárolására szolgáló speciális fájlokat. Ha Windows és Mac gépek közösen használnak fájlokat, akkor ezzel megakadályozható, hogy hozzá nem értő felhasználók töröljék a Mac felhasználók számára szükséges fájlokat.

A beállításnak ugyanaz a szintaxisa, mint a `hide files` beállításé: a lista mindegyik elemének törtvonallal (/) kell kezdődnie és végződnie, még akkor is, ha a lista egyetlen elemből áll. Általános helyettesítőként csillag karakterek használhatók 0 vagy több karakter helyett. Minden egyes kérdőjel pontosan egy karaktert helyettesít. Példa:

```
veto files = /*config/*default?/
```

A beállítás elsősorban adminisztratív célokra használható, és nem helyettesítheti az engedélyekre vonatkozó beállításokat.

*delete veto files*

Ezzel a beállítással arra utasítható a Samba, hogy törölje a vétózott fájlokat, ha egy felhasználó az ezeket tartalmazó könyvtárt törli. A beállítás alapértelmezés szerinti értéke a `no`. Ez azt jelenti, hogy amikor egy felhasználó olyan könyvtárt próbál törölni, amely vétózott fájlt tartalmaz, a fájl nem törlődik (sem a könyvtár). A könyvtár továbbra is megmarad, csak üresnek látszik. Ha a beállításhoz a `yes` értéket rendeljük, mind a vétózott fájlok, mind a könyvtár törlődnek.

## *Fájlengedélyek és attribútumok MS-DOS és Unix rendszerben*

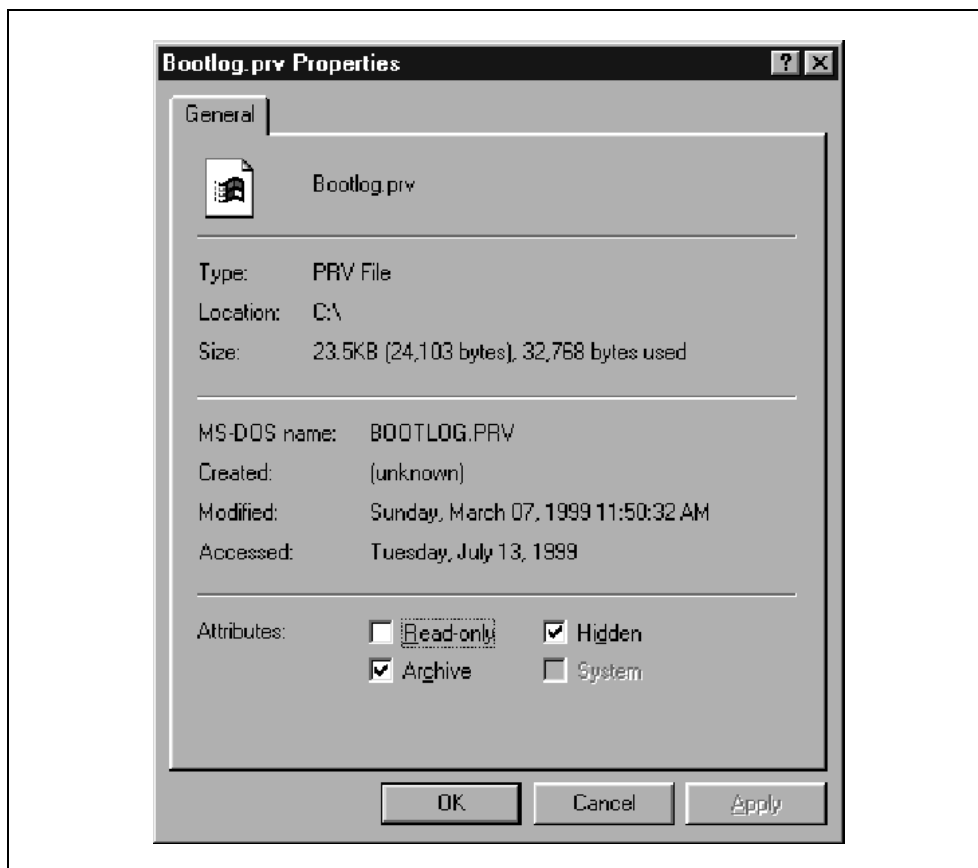
A DOS rendszert a kifejlesztői sohasem szánták többfelhasználós, hálózati operációs rendszernek. Ezzel szemben a Unixot már a születésétől fogva erre tervezték. Ebből következik, hogy a két fájlrendszer között számos eltérés van, amelyekkel a Sambának nem csak tisztában kell lennie, hanem megoldást is kell találnia rájuk. Az egyik leglényegesebb különbség abban van, ahogyan a Unix és a DOS a fájlengedélyeket kezeli.

Nézzük először a Unixot. Mindegyik Unix fájlhoz tartozik egy-egy olvasási (`read`), írási (`write`) és végrehajtási (`execute`) bit a felhasználók következő három csoportjához: tulaj-

donos, csoport és világ. Ezek az engedélyező bitek egy fájl bal szélén olvashatók, amikor egy Unix könyvtárban kiadjuk az `ls -al` parancsot. Például:

```
-rwxr--r--  1 tom      users    2014 Apr 13 14:11 access.conf
```

Ezzel szemben a Windows négy ún. *attribútumbit*et használ minden egyes fájlhoz: csak olvasható (read-only), rendszer (system), rejtett (hidden) és archiválható (archive). Ezeket a biteket úgy nézhetjük meg, hogy az egér jobb oldali gombjával egy fájlra kattintunk, és a helyi menüben a Tulajdonságok parancsot választjuk. Ekkor az 5.6. ábrán látható párbeszédablak jelenik meg.



5.6. ábra. DOS és Windows fájlok tulajdonságai

Az egyes attribútumbitek jelentése:

*Csak olvasható (Read-only)*

A fájl tartalmát olvashatja a felhasználó, de nem szerkesztheti (nem írhat bele, nem törölhet belőle).

*Rendszer (System)*

Ezek az ún. rendszerfájlok, amelyekre az operációs rendszernek van szüksége.

*Rejtett (Hidden)*

Ezek a fájlok el vannak rejtve a felhasználó elől, és csak akkor jelennek meg, ha kifejezetten így állítjuk be az operációs rendszert.

*Archiválандó (Archive)*

Az ilyen fájlok módosultak azt követően, hogy utoljára biztonsági másolat készült róluk.

Figyeljük meg, hogy a DOS nem ismeri a végrehajtható (executable) attribútumot. A DOS és a Windows NT fájlrendszerek az .EXE, .COM, .CMD és a .BAT kiterjesztéssel jelölik a végrehajtható fájlokat.

Ebből következik, hogy egy Samba lemezmegosztásban nincs kihasználva a Unix fájlok három végrehajtást jelző bitje. Ugyanakkor a DOS fájloknak meg kell őrizniük az archiválандó, a rendszer és a rejtett attribútumbitjeiket, ha Unix környezetben kerülnek tárolásra. A Samba ehhez a Unix fájlok három végrehajtást engedélyező bitjét használhatja, ha erre kap utasítást. E biteknek az ilyen módon történő megfeleltetése (leképzése) azonban sajnos mellékhatással jár: ha egy Windows felhasználó Samba megosztásban tárolja a fájlját, és egy Unixos felhasználó az `ls -al` paranccsal meg akarja tekintetni azt, a végrehajthatóságot jelző bitek némelyike nem úgy viselkedik, amint az várható.

A Sambában a következő három beállítással adhatjuk meg, hogy megtörténjen-e a bitek leképzése: `map archive`, `map system` és `map hidden`. Ezek a beállítások az archiválандó, a rendszer és a rejtett attribútumbiteket rendre leképezik a tulajdonos, a csoport és a világ végrehajtható bitekre. A beállításokat a `[data]` megosztásba vehetjük fel a következő módon:

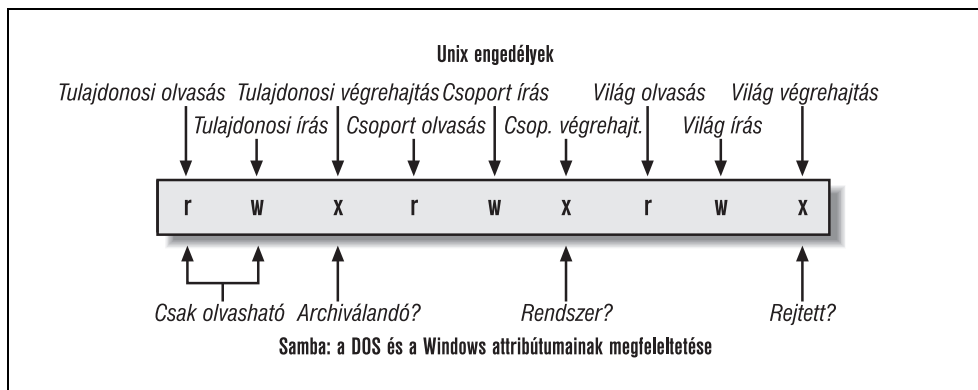
```
[data]
    path = /home/samba/data
    browseable = yes
    guest ok = yes
    writeable = yes
    map archive = yes
    map system = yes
    map hidden = yes
```

Ezt követően Unix alatt készítsünk el egy fájlt a megosztásban, ami például a `hello.java` lehet, és állítsuk 755-re a fájl engedélyeit. Miután elvégeztük ezeket a beállításokat, a Windows oldalon megvizsgálva az engedélyeket azt látjuk, hogy a Tulajdonságok párbeszédablakban mind a három attribútumbit be van jelölve. Mi történik azonban a csak olvasható attribútummal? Alapértelmezés szerint a Samba 2.0 ehhez mindig az 1-es értéket rendel, ha a Unixban a tulajdonosi írásbit értéke 0. Másként fogalmazva a csak olvasható attribútumbit úgy állíthatjuk 1-re, hogy a fájl engedélyeit az 555-ös értékre módosítjuk.

Fel kell hívnunk arra a figyelmet, hogy a `map archive` beállításhoz alapértelmezés szerint a `yes` érték tartozik, míg a másik két beállításnak alapértelmezés szerint `no` az értéke.

Ennek az az oka, hogy sok program nem működik helyesen, ha a DOS és a Windows fájlok archiválható bítje nincs helyesen tárolva. Ezzel szemben a rendszer és a rejtett attribútumoknak nincs különösebb jelentőségük a programok működésében, ezért a rendszergazda belátására van bízva ezek leképzése.

Az 5.7. ábrán összefoglaltuk a Unix engedélyeit, és azt is szemléltetjük, hogyan képezi le a Samba ezeket a biteket a DOS attribútumbitjeire. Figyeljük meg, hogy a csoport írási/olvasási és a világ olvasási/írási bitek nincsenek közvetlenül leképezve a DOS attribútumbitjeire, hanem a Samba kiszolgálón megtartják az eredeti unixos szerepüket.



5.7. ábra. Fájlengedélyek Unix és Samba oldalról nézve

### Létrehozási maszk

A Sambában különböző beállítások segítik a létrehozási maszk (más néven *umask*) használatát. A létrehozási maszk segítségével már a fájl vagy a könyvtár készítésekor meghatározhatjuk a hozzá tartozó engedélyeket. A Unixban ez azt jelenti, hogy a fájl vagy a könyvtár létrehozásakor szabályozhatjuk, hogy milyen tiltások legyenek érvényesek a fájlra vagy a könyvtárra. Windows oldalra lefordítva ez azt jelenti, hogy egy fájl létrehozásakor a Windowsban használatos négy attribútumbit (csak olvasható, archiválható, rendszer és rejtett) bármelyikét kikapcsolhatjuk.

Az alábbi példában a `create mask` beállításhoz a 744-es értéket adtuk meg, amivel kikényszerítjük a Windows ügyféltől, hogy egy fájl létrehozásakor a lehető legtöbb engedélyt rendelje a fájlhoz:

```
[data]
path = /home/samba/data
browseable = yes
guest ok = yes
writeable = yes
create mask = 744
```

Ehhez hasonlóan a `directory mask` beállításhoz a maximális 755-ös értéket rendelve azt kényszerítjük ki a Windows ügyféltől, hogy a lehetséges legtöbb engedélyt rendelje a létrehozandó könyvtárhoz:

```
[data]
    path = /home/samba/data
    browseable = yes
    guest ok = yes
    writeable = yes
    directory mask = 755
```

A `force create mode` és a `force directory mode` beállítások segítségével akár az egyes attribútumbitek egyenkénti be-, illetve kikapcsolásával is kikényszeríthetjük az engedélyeket. Ezek a beállítások logikai OR műveletet végeznek a fájl- és a könyvtár-létrehozó maszkokon, és ezzel biztosítják, hogy egy adott bit mindig be legyen kapcsolva. Ezeket a beállításokat általában globális hatókörrel használjuk, hogy a csoport és a világ olvasási/írási engedélyek minden egyes megosztásban létrehozásra kerülő fájlokra és könyvtárakra megfelelően legyenek beállítva.

A fentiekhez hasonlóan a `force user` és a `force group` beállítások segítségével a Windows oldalon készített fájlok Unixos felhasználói és csoportengedélyeit is beállíthatjuk. Például:

```
[data]
    path = /home/samba/data
    browseable = yes
    guest ok = yes
    writeable = yes
    create mask = 744
    directory mask = 755
    force user = joe
    force group = accounting
```

Ezek a beállítások statikus Unix felhasználót és csoportot rendelnek a megosztáshoz létrehozott minden egyes kapcsolathoz. Ez azonban csak az ügyfél hitelesítése *után* történik meg, és nem engedélyezi a megosztáshoz való szabad hozzáférést. A beállításokat gyakran azon mellékhatásuk miatt használják, amivel adott felhasználó vagy csoport rendelhető a megosztásban létrehozásra kerülő új fájlhoz vagy könyvtárhoz. Legyünk óvatosak ezekkel a beállításokkal.

Végül essen szó a Unix még egy olyan képességéről, amely hiányzik a DOS-ból: egy csak olvasható fájl törlése egy írható könyvtárból. A Unix rendszerben egy írható könyvtárból egy csak olvasható fájl is törölhető. Ez lehetővé teszi, hogy bármelyik könyvtárunkból töröljünk fájlokat, még ha egy adott fájlt más is helyezett el benne.

A DOS fájlrendszert nem többfelhasználós környezetre tervezték, ezért abban a csak olvasható azt jelenti, hogy „védett véletlen változtatásokkal szemben, beleértve a törlését is”, és nem azt, hogy „védett az egyfelhasználós rendszeren más felhasználókkal szemben”. Ezért a DOS tervezői nem engedélyezték a csak olvasható fájlok törlését. Még a mai Windows fájlrendszerek is követik ezt a hagyományt.

Normál esetben ezzel nincs is probléma. A Windows programok nem kísérlik meg a csak olvasható fájlok törlését, mert tisztában vannak azzal, hogy ez nem okos dolog. Ezzel szemben egy sor más, eredetileg Unixra írt, és most Windows alatt futó program igényli,

hogy törölhetőek legyenek a csak olvasható fájlok. A Sambában ezt a `delete readonly` beállítás segítségével engedélyezhetjük, amennyiben a `yes` értéket rendeljük hozzá:

```
[data]
path = /home/samba/data
browseable = yes
guest ok = yes
writeable = yes
create mask = 744
directory mask = 755
force user = joe
force group = accounting
delete readonly = yes
```

### *Fájl- és könyvtárengedélyek beállításai*

A fájl- és könyvtárengedélyekkel kapcsolatos beállítási lehetőségeket a 5.5. táblázat foglalja fel. Ezt követően külön-külön ismertetjük mindegyiket.

5.5. táblázat. Fájl- és könyvtárengedélyek beállításai

Beállítás	Paraméterek	Funkció	Alapértelmezett érték	Hatókör
<code>map archive</code>	Boolean érték	Megőrzi a DOS archiválható attribútumát a felhasználói végrehajtható bitben (0100).	yes	Megosztás
<code>map system</code>	Boolean érték	Megőrzi a DOS rendszer attribútumát a csoport végrehajtható bitben (0010).	no	Megosztás
<code>map hidden</code>	Boolean érték	Megőrzi a DOS rejtett attribútumát a világ végrehajtható bitben (0001).	no	Megosztás
<code>create mask</code> (create mode)	Numerikus	Beállítja a Samba által létrehozott fájlokhoz rendelhető maximális engedélyt.	0744	Megosztás
<code>directory mask</code> (directory mode)	Numerikus	Beállítja a Samba által létrehozott könyvtárakhoz rendelhető maximális engedélyt.	0755	Megosztás
<code>force create mode</code>	Numerikus	Kikényszeríti a megadott engedélyeket (bitenkénti OR) a Samba által létrehozott fájlokhoz.	0000	Megosztás

## 5.5. táblázat folytatása

Beállítás	Paraméterek	Funkció	Alapértelmezett érték	Hatókör
force directory mode	Numerikus	Kikényszeríti a megadott engedélyeket (bitenkénti OR) a Samba által létrehozott könyvtárakhoz.	0000	Megosztás
force group (group)	Karakterlánc (csoport neve)	Csoportot rendel az illető megosztáshoz kapcsolódó felhasználóhoz.	Nincs	Megosztás
force user	Karakterlánc (felhasználó-név)	Felhasználót rendel az illető megosztáshoz kapcsolódó felhasználóhoz.	Nincs	Megosztás
delete readonly	Boolean érték	Lehetővé teszi csak olvasható fájl törlését egy írható könyvtárból.	no	Megosztás

*create mask*

A beállításhoz rendelt oktális szám azt jelzi, hogy az ügyfél egy fájl mely engedélybitjeit kapcsolhatja be, amikor létrehozza a fájlt a megosztásban. Az alapbeállítás szerinti érték 0744, ami azt jelenti, hogy a fájl unixos tulajdonosa olvashatja, írhatja és végrehajthatja a fájlt, míg a csoportjába tartozó felhasználók és mások csak olvashatják és végrehajthatják azt. Ha azt szeretnénk, hogy ne legyen végrehajtható a fájl, akkor a 0644-es értéket állítsuk be (rw-r--r--). Ne feledjük, hogy a végrehajtható biteket a kiszolgáló arra is használhatja, hogy DOS fájlok bizonyos attribútumbitjeit képezze le rájuk, amint erről korábban olvashattunk. Ha módosítjuk a create mask beállításhoz rendelt értéket, akkor ezeknek a biteknek a viselkedését is módosítjuk.

*directory mask*

A beállításhoz rendelt oktális szám azt jelzi, hogy az ügyfél egy könyvtár mely engedélybitjeit kapcsolhatja be, amikor létrehozza a könyvtárt a megosztásban. Az alapbeállítás szerinti érték 0755, ami azt jelenti, hogy Unix oldalon bárki olvashatja a könyvtárat és lépegethet is benne, de csak a létrehozója módosíthatja. Itt a 0750-es érték javasolható.

*force create mode*

Ezzel a beállítással átállíthatók azok az engedélybitek, amelyek bekapcsolását a Samba egy fájl engedélybitjeinek módosításakor kikényszeríti. Ezt a beállítást gyakran használják csoportengedélyek kikényszerítésére, amint erről korábban már volt szó. Segítségével a DOS archiválható (0100), rendszer (0010) és a rejtett (0001) attribútumbitjei előre is beállíthatók. Ez a beállítás mindig felülbírálja a map archive, a map system, a map hidden és a create mask beállításokat.



Számos Windows alkalmazás az adatfájlok módosításakor a régi adatfájlt *.bak* kiterjesztéssel menti, és újat hoz létre, vagyis úgy változtatja meg a tulajdonosát és az engedélyeit, hogy ugyanannak a Unix csoportnak a tagjai nem szerkeszthetik a fájlt. A `force create mode = 0660` beállítással az új fájlt szerkeszthetik is a csoport tagjai.

### *force directory mode*

Ezzel a beállítással átállíthatók azok az engedélybitek, amelyek bekapcsolását a Samba egy könyvtár engedélybitjeinek módosításakor kikényszeríti. Ezt a beállítást gyakran használják csoportengedélyek kikényszerítésére, amint erről korábban már volt szó. A beállítás alapértelmezés szerinti értéke 0000, és a `force create mode` beállításhoz hasonlóan szükség esetén csoport vagy más engedélyek megadására használható. Ez a beállítás mindig felülbírálja a `map archive`, a `map system`, a `map hidden` és a `directory mask` beállításokat.

### *force group*

Ez a beállítás, amit esetenként `group` alakban is írnak, statikus csoportazonosítót ad meg, amit minden felhasználó használhat, miután bejelentkezett egy megosztáshoz, és a hitelesítése is megtörtént. A beállítás az SMB ügyfél által létrehozott minden egyes új fájlhoz vagy könyvtárhoz hozzárendel egy meghatározott csoportot.

### *force user*

A `force user` beállítás statikus felhasználó-azonosítót ad meg, amit minden felhasználó használhat, miután bejelentkezett egy megosztáshoz, és a hitelesítése is megtörtént. A beállítás az SMB ügyfél által létrehozott minden egyes új fájlhoz vagy könyvtárhoz hozzárendel egy meghatározott felhasználót.

### *delete readonly*

Ezzel a beállítással engedélyezhetjük, hogy egy felhasználó csak olvasható fájlt tartalmazó könyvtárt is törölhessen. Alapbeállítás szerint a DOS és a Windows az ilyen műveletet nem teszi lehetővé. Hacsak nem igényli egy program, akkor hagyjuk kikapcsolva ezt a beállítást; sok windowsos felhasználó zokon venné, ha azt tapasztalná, hogy a csak olvashatóvá tett fájljai is törölhetők. Igazából azonban a Unix `rm` parancsa is megkérdezi a felhasználót, hogy tényleg felül akarja-e bírálni ezt a beállítást, és valóban törölni akarja a csak olvasható fájlokat. Lehetőség szerint ne változtassuk meg a Samba alapbeállítását.

### *map archive*

A DOS archiválandó bitje azt jelzi, hogy módosult-e egy fájl azóta, hogy utoljára biztonsági másolatot készítettünk róla (vagy lefuttattuk-e vele a DOS archiváló programját). Ha a `map archive = yes` beállítást adjuk meg, akkor a Samba a DOS archiválandó jelzőbitjét a Unix tulajdonosi végrehajtható bitjére (0100) képezi le (ennek felelteti meg). Legjobb ezt a beállítást használnunk, ha engedélyezzük, hogy a Windows felhasználók biztonsági másolatot készítsenek a fájlaikról, vagy ha olyan programokat futtatunk, amelyek igénylik az archiválandó bitet. A Unix egyáltalán nem ismer ilyen archiválandó bitet. A biztonsági

másolatokat készítő programok viszont általában listát vezetnek az archivált fájlokról és a másolatkészítés idejéről, ezért a módosulási dátumok összehasonlítása ugyanezt a célt szolgálja.

A *yes* érték választása esetenként meglepheti a Unixos ügyfeleket, ha azt tapasztalják, hogy egy adatfájljuk végrehajtható fájlként van megjelölve, de ennél nagyobb baj nem történik. Ha valaki megpróbálna végrehajtani egy ilyen fájlt, hibaüzenetet kapna, amikor a héjprogram megkísérelné utasításként értelmezni a kezdő sorokat. A dolog fordítottja is előfordulhat: egy végrehajtható Unix fájl akként jelenhet meg, mint amiről a Windowsban még nem készült biztonsági másolat. Az előzőhöz hasonlóan ebből sem származhatnak komolyabb problémák.

#### *map system*

A DOS system (rendszer) attribútumbitje azokat a fájlokat jelöli meg, amelyekre az operációs rendszernek van szüksége, és amelyeket nem szabad törölni, átnevezni vagy áthelyezni külön intézkedések nélkül. Csak akkor kapcsoljuk be ezt a beállítást, ha egy Unix fájlkiszolgálón akarjuk tárolni a Windows rendszerfájljait. A végrehajtható Unix fájlok nem áthelyezhető speciális Windows fájlkként jelennek meg a Windows ügyfelek előtt. Ez ugyan némi kényelmetlenséget okozhat, ha megpróbálnánk áthelyezni őket, de a legtöbb esetben nincs velük probléma.

#### *map hidden*

A DOS a hidden (rejtett) attribútumbittel azokat a fájlokat jelöli meg, amelyeket normál körülmények között nem akar megjeleníteni a fájlok listájában. A Unixban nincs ilyen bit; a rendszer az egyedi programokra (nevezetesen például a héjprogramra) bízta azt, hogy mely fájlokat jelenítsenek meg és melyeket ne. Normál esetben nincs szükségünk a rejtett DOS fájlokra, ezért a legjobb, ha meghagyjuk ennek a beállításnak a kikapcsolt állapotát.

Ha a beállításhoz a *yes* értéket rendeljük, akkor a kiszolgáló a rejtett attribútumbitet a világ végrehajtható bitjére (0001) képezi le. Ennek meglehetősen furcsa lenne a hatása. Minden olyan Unix program, amelyet bárki végrehajthat, eltűnne a Windows felhasználók elől. Ha viszont a *no* értéket rendeljük ehhez a beállításhoz, és egy Windows felhasználó egy Samba megosztásban lévő fájlt rejtettként szeretne megjelölni, akkor ez nem sikerülne neki: a Sambának ugyanis nincs helye, ahol tárolhatná a rejtett attribútumbitet!

## *Névzsugorítás és a kis- és nagybetűk*

A DOS és a Windows 3.1 operációs rendszerek idejében a fájlnevek maximum 8 nagybetűből, egy pontból és három, ugyancsak nagybetűből álló, ún. névkiterjesztésből állhattak. Ezt röviden *8.3-as formátumnak* nevezték, és számos kellemetlenséget okoztak. Azóta a Windows 95/98, a Windows NT és a Unix azzal, hogy jóval több karaktert enged meg a fájlnevekben, és ezeken belül a kis- és nagybetűket is megkülönbözteti, megoldotta ezt a problémát. Az 5.6. táblázatban összefoglaltuk a leggyakoribb operációs rendszerekre érvényes névhasználatot.

5.6. táblázat. Névhasználati kötöttségek az operációs rendszerekben

Operációs rendszer	Névhasználati szabályok
DOS 6.22 vagy korábbi	Nyolc karakter, utána egy pont, majd három karakter (8.3 formátum), kis- és nagybetű nem számít
Windows 3.1 for Workgroups	Nyolc karakter, utána egy pont, majd három karakter (8.3 formátum), kis- és nagybetű nem számít
Windows 95/98	255 karakter, kis- és nagybetű számít, és megtartja az írásmódot
Windows NT	255 karakter, kis- és nagybetű számít, és megtartja az írásmódot
Unix	255 karakter, kis- és nagybetű számít

A Samba az olyan felhasználók kedvéért, akik például a Windows for Workgroups operációs rendszer alatt dolgoznak, megtartotta a kompatibilitást a 8.3-as formátummal. Ha egy felhasználó egy megosztásban egy *antidisestablishmentarianism.txt* nevű fájlt hozna létre, akkor a Windows for Workgroups rendszert használó ügyfél nem tudná ezt megkülönböztetni egy ugyanabban a könyvtárban lévő *antidisease.txt* nevű fájltól. A Windows 95/98 és a Windows NT rendszerhez hasonlóan a Samba is egy speciális eljárást használ arra, hogy a hosszú fájlneveket 8.3-as formátumra alakítsa át úgy, hogy a hasonló fájlnevekből ne adódjon ütközés. Ezt az eljárást *névzsugorításnak* nevezik. A Samba a fájlneveket a Windows 95 és az utódjaihoz hasonló, de nem azonos módon zsugorítja.

### A Samba névzsugorítási eljárása

A Samba a következő módon zsugorítja a hosszú fájlneveket 8.3-as formátumra:

- Ha az eredeti fájlnev nem ponttal kezdődik, akkor az utolsó pont (amennyiben van ilyen) előtti első öt alfanumerikus karaktert nagybetűsre alakítja át. Ezek a karakterek alkotják a 8.3-as formátumra zsugorítandó fájlnev első öt karakterét.
- Ha az eredeti fájlnev ponttal kezdődik, akkor a pontot eltávolítja, és az utolsó pont (amennyiben van ilyen) előtti első öt alfanumerikus karaktert nagybetűsre alakítja át. Ezek a karakterek alkotják a 8.3-as formátumra zsugorítandó fájlnev első öt karakterét.
- Ezeket a karaktereket közvetlenül egy speciális rövidítő karakter követi, ami alapbeállítás szerint a tilde (~), de ez megváltoztatható.
- A hosszú fájlnevnél az utolsó pont előtti részéből egy kétkarakteres kódot képez; amennyiben szükséges, a név utolsó pont utáni részei használhatók. Ezt a kétkarakteres kódot beilleszti a rövidítő karakter mögé.
- Az eredeti fájlnev utolsó pontja (amennyiben van ilyen) utáni első három karaktert nagybetűsre alakítja át, és hozzáfűzi a rövidített névhez. Ha az eredeti fájlnev ponttal kezdődött, akkor három aláhúzásjelet ( \_ \_ \_ ) használ kiterjesztésként.

Íme néhány példa:

virtuosity.dat	VIRTU~F1.DAT
.htaccess	HTACC~U0._ _ _
hello.java	HELLO~1F.JAV
team.config.txt	TEAMC~04.TXT
antidisestablishmentarianism.txt	ANTID~E3.TXT
antidiseast.txt	ANTID~9K.TXT

A szabályokat használva a Windows for Workgroups rendszerben dolgozók is különbséget tudnak tenni az eredetileg hosszú nevű fájlok között. Jegyezzük meg, hogy a Samba ugyanazt a hosszú fájlnevet mindig ugyanarra a rövid névre zsugorítja, ami a Windowsban nem mindig igaz. Ennek viszont az a hátránya, hogy továbbra is előfordulhatnak ütközések, bár ennek kicsi a valószínűsége.

A névzsugorítási beállításokkal általában csak akkor kell foglalkoznunk, ha régi rendszereken dolgozó ügyfelek is csatlakoznak a hálózathoz. Azért, hogy a többi ügyfelet ne zaklassuk ilyen problémákkal, azt javasoljuk, hogy vegyünk fel egy `include` direktívát az *smb.conf* fájlba:

```
[global]
include = /usr/local/samba/lib/smb.conf.%a
```

Ez elkészíti az *smb.conf.WfWg* feloldást, amikor egy Windows for Workgroups ügyfél jelentkezik be a hálózatra. Most el kell készítenünk a */usr/local/samba/lib/smb.conf.WfWg* fájlt, ami a következő beállításokat tartalmazhatja:

```
[global]
case sensitive = no
default case = upper
preserve case = no
short preserve case = no
mangle case = yes
mangled names= yes
```

Ha nem használunk Windows for Workgroups 3.1 operációs rendszert, akkor valószínűleg egyik beállításnak sem kell megváltoztatnunk az alapértelmezés szerinti értékét.

### ***Fájlnemek ábrázolása és feloldása a Sambában***

A fájlnevekkel kapcsolatban arról is szólnunk kell, hogy a különböző operációs rendszerek eltérően ábrázolják és oldják fel a fájlneveket. Aki dolgozott már Windows 95/98/NT operációs rendszerben, az minden bizonnyal találkozott már egy *README.TXT* nevű fájlal. Az operációs rendszer ennek a fájlnek a nevét csupa nagybetűvel tárolja. Ha viszont megnyitunk egy DOS ablakot, és a parancssorba beírjuk az `edit readme.txt` parancsot, a nagybetűs nevű program betöltődik a szerkesztőprogramba, még ha kisbetűkkel is írtuk be nevét.

Ennek az az oka, hogy a Windows 95/98/NT család tagjai a név feloldásakor nem tesznek különbséget a kis- és a nagybetűk között. Ezzel szemben viszont a Unix alapú operációs rendszerek a fájlnevek feloldásakor mindig megkülönböztetik a kis- és a nagybetűket: ha a `vi readme.txt` parancs beírásával megpróbálnánk szerkeszteni a `README.TXT` fájlt, akkor feltehetően csak egy új fájl üres puffert szerkeszthetnénk.

A Samba a következőképpen kezeli a kis- és nagybetűket: ha a `preserve case` beállításhoz a `yes` értéket rendeljük, akkor a Samba az operációs rendszer által rendelkezésre bocsátott alakban fogja használni a betűket a fájlnevek megjelenítéséhez (nincs feloldás). Ha a beállításához a `no` értéket rendeljük, akkor a Samba a `default case` beállításban megadott alakot (kisbetű vagy nagybetű) fogja használni. Ugyanez vonatkozik a `short preserve case` beállításra is. Ha ennek a beállításnak `yes` az értéke, a Samba a 8.3 formátumú fájlnevek megjelenítéséhez az operációs rendszer alapértelmezés szerinti alakját fogja használni. Végül a Samba a megosztásaiban a fájlneveket mindig a `case sensitive` beállításban megadott alak szerint fogja használni.

### A zsugorítás beállításai

A Samba lehetővé teszi, hogy finomabban szabályozzuk a hosszú fájlnevek zsugorítását. Így például előírhatjuk a kis- és nagybetűk megkülönböztetését, a rövidített fájlnevekbe beszúrandó karaktereket, valamint a fájlnevek egyik formátumból másik formátumba történő kézi átalakítását. Ezeket a beállítási lehetőségeket az 5.7. táblázat sorolja fel.

5.7. táblázat. A névzsugorítás beállítási lehetőségei

Beállítás	Paraméterek	Funkció	Alapértelmezett érték	Hatókör
<code>case sensitive</code> ( <code>casesig-names</code> )	Boolean érték	Ha <code>yes</code> az értéke, a Samba a fájlnevekben megkülönbözteti a kis- és nagybetűket (a Windows nem teszi).	<code>no</code>	Megosztás
<code>default case</code>	( <code>upper</code> vagy <code>lower</code> )	Az alapértelmezésnek tekintett alakot használja (csak akkor, ha a <code>preserve case</code> értéke <code>no</code> ).	<code>Lower</code>	Megosztás
<code>preserve case</code>	Boolean érték	Ha <code>yes</code> az értéke, megtartja az ügyfél által használt alakot (vagyis nem alakítja át a <code>default case</code> szerint).	<code>yes</code>	Megosztás
<code>short preserve case</code>	Boolean érték	Ha <code>yes</code> az értéke, megtartja az ügyfél által használt 8.3-as formátum alakját.	<code>yes</code>	Megosztás
<code>mangle case</code>	Boolean érték	Zsugorítja a nevet, ha vegyesen van benne kis- és nagybetű.	<code>no</code>	Megosztás
<code>mangled names</code>	Boolean érték	A DOS 8.3-as formátumára zsugorítja a fájlnevet.	<code>yes</code>	Megosztás

## 5.7. táblázat folytatása

Beállítás	Paraméterek	Funkció	Alapértelmezett érték	Hatókör
mangling char	Karakterlánc (egyetlen karakter)	Megadja a rövidítő karaktert.	~	Megosztás
mangled stack	Numerikus	A helyi rövidítő veremben tárolandó rövidített nevek száma.	50	Global
mangled map	Karakterlánc (mintalista)	Lehetővé teszi fájlnevek egyik formátumból másik formátumba való leképezését.	Nincs	Megosztás

*case sensitive*

Ezzel a megosztás hatókörű beállítással, aminek a rejtelmes casesignames a szinonimája, azt írhatjuk elő, hogy a Samba egy adott megosztásban a fájlnevek feloldásakor megtartsa-e a kis- és a nagybetűket. A beállításhoz alapértelmezés szerint a no érték van rendelve, ami annak felel meg, ahogyan a Windows kezeli a névfeloldást. Ha az ügyfelek olyan operációs rendszert használnak, amely megkülönbözteti a fájlnevekben a kis- és a nagybetűket, akkor a beállítást az alábbi módon használhatjuk:

```
[accounting]
    case sensitive = yes
```

Ellenkező esetben hagyjuk meg a beállítás alapértelmezett értékét.

*default case*

Ezt a beállítást a preserve case beállítással együtt használhatjuk. Ez a beállítás azt az alapbeállítás szerinti alakot (kisbetűk vagy nagybetűk) határozza meg, amit a Samba használni fog, amikor a megosztásaiban egy ügyfél számára létrehoz egy fájlt. Az alapbeállítás szerinti alak a kisbetűs (lower) alak, ami azt jelenti, hogy az újonnan létrehozott fájlok neveiben megmaradnak az ügyfél által használt kis- és nagybetűk. Amennyiben szükséges, a következő módon felülbírálhatjuk ezt a globális beállítást:

```
[global]
    default case = upper
```

Ha a beállításhoz ezt az értéket rendeljük, akkor az újonnan létrehozott fájlok nevében a kisbetűkből nagybetűk lesznek, és ezek nem bírálhatók felül egy programban. Azt ajánljuk, hogy tartsuk meg az alapértelmezett értéket, hacsak nem kell Windows for Workgroups vagy más, 8.3-as formátumot használó ügyfelekkel együtt dolgoznunk.

### *preserve case*

Ezzel a beállítással azt írhatjuk elő, hogy a Samba az ügyfél részére létrehozandó fájl nevében az ügyfél operációs rendszere által használt alakot, vagy a `default case` beállításban megadott alakot fogadja-e el. A beállításhoz alapértelmezés szerint a `yes` érték tartozik, ami azt jelenti, hogy a Samba az ügyfél operációs rendszere által felkínált alakot fogadja el. Ha a `no` értéket rendeljük ehhez a beállításhoz, akkor a Samba a `default case` beállításban megadott alakot fogja használni.

Jegyezzük meg, hogy ez a beállítás nem kezeli az ügyfél által küldött 8.3-as kéréseket – lásd ehhez a `short preserve case` beállítást alant. A beállításhoz a `yes` értéket rendelhetjük, ha a Samba kiszolgálón olyan alkalmazások akarnak fájlokat létrehozni, amelyek különbséget tesznek a fájlnevekben a kis- és a nagybetűk között. Ha például a Windows NT fájlrendszerét akarjuk utánozni, akkor hagyjuk meg a beállítás alapértelmezés szerinti `yes` értékét.

### *short preserve case*

Ezzel a beállítással azt írhatjuk elő, hogy a Samba az ügyfél részére létrehozandó 8.3 formátumú fájlnevekben az ügyfél operációs rendszere által használt alakot, vagy a `default case` beállításban megadott alakot fogadja-e el. A beállításhoz alapértelmezés szerint a `yes` érték tartozik, ami azt jelenti, hogy a Samba az ügyfél operációs rendszere által felkínált alakot fogadja el. A választást a Sambára bízhatjuk a `default case` alábbi megadásával:

```
[global]
    short preserve case = no
```

Ha arra szeretnénk rábírni a Sambát, hogy a Windows NT fájlrendszer viselkedését utánozza le, akkor hagyjuk meg a beállítás alapértelmezés szerinti `yes` értékét.

### *mangled names*

Ezzel a megosztás hatókörű beállítással azt adhatjuk meg, hogy zsugorítsa-e a Samba a fájlneveket a 8.3-as formátumot használó ügyfelek számára. Ha a beállításhoz a `no` értéket rendeljük, akkor a Samba nem zsugorítja a fájlneveket, és a nevek (az ügyféltől függően) vagy láthatatlanok maradnak, vagy a 8.3-as operációs rendszernek megfelelően csonkulnak. A beállításhoz alapértelmezés szerint a `yes` érték tartozik. Ezt az értéket a megosztásokban felülírhatjuk:

```
[data]
    mangled names = no
```

### *mangle case*

Ezzel a beállítással arra utasíthatjuk a Sambát, hogy zsugorítsa-e azokat a fájlneveket, amelyek nem teljesen olyan alakú betűkből állnak, amilyen alakot (kisbetűt vagy nagybetűt) a `default case` beállításban megadtunk. A beállításhoz alapértelmezés szerint a `no` érték tartozik. Ha ehelyett a `yes` értéket választjuk, akkor biztosítanunk kell, hogy az összes ügyfelünk képes legyen az eredményül kapott rövidített fájlnevek kezelésére. Az alapértelmezés szerinti értéket egy megosztáson belül a következők szerint bírálhatjuk felül:

```
[data]
mangle case = yes
```

Javasoljuk az alapértelmezés szerinti érték megtartását, hacsak nem szól valamilyen nyomós ok ellene.

### *mangling char*

Ezzel az ugyancsak megosztás hatókörű beállítással azt a karaktert adhatjuk meg, amelyet a Samba a fájlnevek 8.3-as formátumba való átalakításakor rövidítő karakterként használ. Alapbeállítás szerint ez a tilde (~) karakter. Ehelyett bármilyen más karaktert is előírhatunk, például:

```
[data]
mangling char = #
```

### *mangled stack*

A Samba egy helyi vermet tart fenn, amelyben az utoljára rövidített 8.3-as fájlneveket tárolja. Ezt a vermet arra használja, hogy visszaalakíthassa a rövidített fájlneveket az eredeti alakjukra. Erre akkor lehet szükség, amikor egy alkalmazás létrehoz, ment, bezár vagy módosít egy fájlt. Alapbeállítás szerint ez a verem 50 hosszú és rövidített fájlnev-párt képes tárolni. Ha meg akarjuk kímélni a processzort attól, hogy sűrűn kelljen zsugorítania a fájlneveket, akkor növeljük meg ennek a veremnek a méretét – ezzel viszont több memória-helyet foglalunk el, és némileg a fájlhozzáférések ideje is meghosszabbodik:

```
[global]
mangled stack = 100
```

### *mangled map*

Ha a névzsugorítás alapértelmezés szerint rendelkezésünkre álló lehetőségei nem lennének elegendők, akkor a mangled map beállítás segítségével további utasításokat adhatunk a Sambának a fájlnevek átalakítására vonatkozóan. Ennek a beállításnak a segítségével leképző mintákat adhatunk meg, amelyeket a Samba a zsugorításhoz használhat. Egy példa:

```
[data]
mangled map = (*.database *.db) (*.class *.cls)
```

Ebben a beállításban arra utasítjuk a Sambát, hogy keresse meg azokat a fájlokat, amelyek nevében előfordul a gömbölyű zárójelek között elsőként megadott minta, majd alakítsa át ezek nevét a második zárójelpár között megadott mintára, és ennek megfelelően jelenítse meg a fájlneveket a 8.3 formátumot használó ügyfelek számára. Ennek a beállításnak akkor vehetjük hasznát, ha hibásan működik a névzsugorítás, vagy olyan formátumot eredményez, amit az ügyfelek nem képesek értelmezni. A mintákat szóközzel kell egymástól elválasztani.

## Zárolások és „opportunist” zárolások

Az ugyanazon fájlba való egyidejű írást egyik operációs rendszer sem tartja kívánatos műveletnek. Ezt elkerülendő az operációs rendszerek többsége zárolásokkal biztosítja, hogy egyidejűleg mindig csak egyetlen művelet ír hasson egy fájlba. Az operációs rendszerek hagyományos módon a teljes fájlokat lezárják, bár az újabbak lehetővé teszik csak meghatározott számú bájt lezárását. Ha egy másik művelet megpróbál írni egy zárolt fájlba (vagy annak egy részébe), hibaüzenetet kap az operációs rendszertől, és meg kell várnia, amíg feloldásra kerül a zár.

A Samba támogatja a DOS és az NT fájlrendszer standard zárolási kérelmeit, amelyek csak egyetlen műveletnek engedik meg, hogy adott időben a kiszolgálón lévő teljes fájlba vagy a zárolt területére írjon. Emellett a Samba a Windows NT rendszer új, „opportunist” zárolás” vagy röviden *oplock* nevű új zárolási mechanizmusát is támogatja.

### Opportunist zárolás

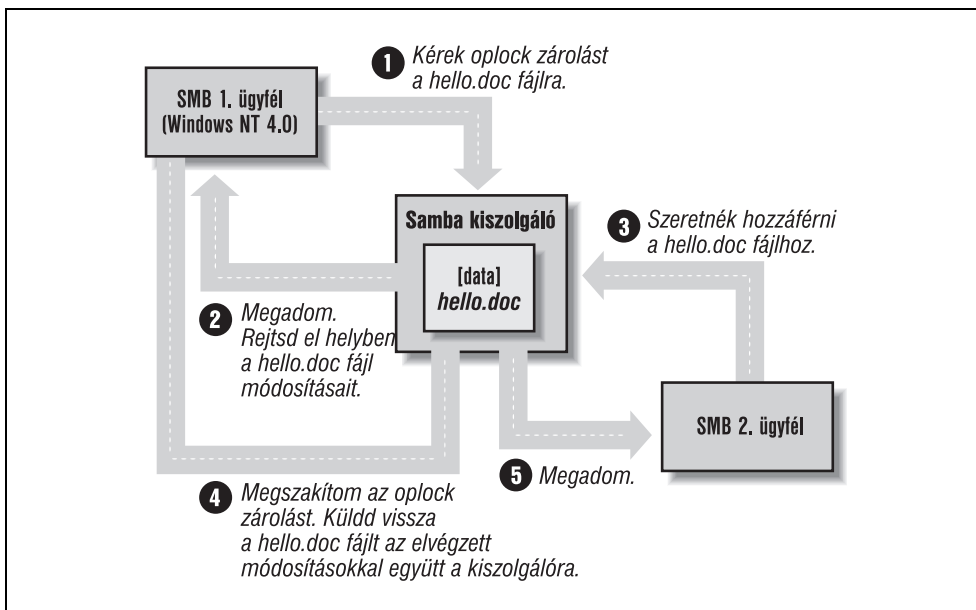
Az opportunist zárolás révén egy ügyfél arról értesítheti a Sambát, hogy nemcsak kizárólagos joggal írni akar egy fájlba, hanem el is akarja rejteni a fájlban végzett módosításokat a saját gépén (és nem a Samba kiszolgálón) azzal a céllal, hogy gyorsabban férhessen hozzá a fájlhoz. Amikor a Samba tudomást szerez arról, hogy egy ügyfél opportunist zárolást alkalmaz egy fájlra, akkor ilyen módon jelöli meg a saját verzióját, és megvárja, amíg a felhasználó befejezi a munkáját és visszaküldi a végleges változatot, amivel aztán szinkronizálja az eredeti fájlt.

Ha másik ügyfél is hozzá szeretne férni a fájlhoz, mielőtt az első ügyfél még befejezte volna a munkáját, a Samba egy *oplock break* kérést küld az első felhasználóhoz. Ezzel arra kéri a felhasználót, hogy hagyja abba a módosítások elrejtését, és küldje vissza a fájlt az aktuális állapotában a kiszolgálóra, hogy a megszakítást kérő ügyfél használhassa. Ez az opportunist zárolás azonban nem helyettesíti a standard megtagadási (deny) módot. Az opportunist zárolás folyamatát az 5.8. ábra szemlélteti.

A zárolásokkal kapcsolatban azt javasoljuk, hogy a Samba alapértelmezés szerinti zárolásait használjuk: ezek a DOS/Windows standard *deny* (megtagadás) módú zárolásai a kompatibilitás megtartása céljából, és az *oplock* zárolások, amelyek a helyi elrejtésből adódóan a teljesítményt javítják. Ha az operációs rendszerünk ki tudja használni az *oplock* zárolásból származó előnyöket, akkor ezzel jelentősen növelhető a teljesítmény. Ha viszont nincs különösebb okunk ezen beállítások módosítására, akkor hagyjuk meg őket az eredeti állapotukban.

### A Unix és a zárolás

A Windows rendszerek megfelelőképpen tudnak egymással együttműködni, és megfelelő védelmet nyújtanak a másik rendszer felülírásaival szemben. Ha viszont egy fájlt egy Samba rendszerben tárolunk, és a fájlhoz egy Unix művelet is hozzáférhet, a művelet nem vesz tudomást a Windows *oplock* zárolásáról, és könnyen megkerülheti azt. Egyes Unix rendszereket viszont már felkészítettek arra, hogy észrevegyék a Samba által fenntartott Windows *oplock* zárolásokat. Jelenleg ilyen rendszer a SGI Irix 6.5.2f és későbbi verziói, a Linux és a FreeBSD, amelyeket hamarosan újabb rendszerek is követni fognak.



5.8. ábra. Opportunista zárolás

Ha a rendszerünk megérti az oplock zárolásokat, akkor a Samba konfigurációs fájlba vegyük fel a `kernel oplocks = yes` bejegyzést. Ezzel kiküszöbölhetjük a Unix folyamatok és a Windows felhasználók közötti ütközéseket.

Ha a rendszerünk nem támogatja a kernel zárolásokat, meghibásodhatnak az adataink, ha valaki egy Unix művelettel olvasni vagy írni próbál olyan fájlt, amihez egy Windows felhasználó is hozzáférhet. Ugyanakkor a Samba rendelkezésünkre bocsát egy – viszonylag durva – védelmi mechanizmust, ha nem használhatók a kernel zárolások: ez a `veto oplock files` beállítás. Ha előre tudjuk vagy feltételezzük, hogy melyek azok a Samba kiszolgálón tárolt fájlok, amelyekhez mind Windows, mind Unix ügyfelek hozzá fognak férni, akkor adjuk meg ezen fájlok nevét a `veto oplock files` beállításban. Ezzel letiltjuk a mintának megfelelő fájlnevekről az oplock zárolást, ezzel együtt az ügyfélnél történő elrejtést is, és lehetővé tesszük, hogy a Windows és a Unix programok a rendszer zárolását vagy a frissített dátumokat használják egy adott fájl eléréséért folytatott versenyben. Egy példa erre a beállításra.

```
veto oplock files = /*.dbm/
```

Ez a beállítás mind a Unix processzek, mind a Windows felhasználók számára lehetővé teszi a `.dbm` kiterjesztésű fájlok szerkesztését. Figyeljük meg, hogy a beállításnak ugyanolyan a szintaxisa, mint a `veto files` beállításé.

A Samba locks és oplocks beállítási lehetőségeit az 5.8. táblázat foglalja össze.

5.8. táblázat. A locks és az oplocks konfigurációs beállításai

Beállítás	Paraméterek	Funkció	Alapértelmezett érték	Hatókör
share modes	Boolean érték	Ha yes az értéke, bekapcsolja a DOS stílusú, teljes fájlra vonatkozó zárolást.	yes	Megosztás
locking	Boolean érték	Ha yes az értéke, bekapcsolja bájtartományú zárolást.	yes	Megosztás
strict locking	Boolean érték	Ha yes az értéke, letiltja a teljes fájl elérését, ha be van kapcsolva rá a bájtartományú zárolás.	no	Megosztás
oplocks	Boolean érték	Ha yes az értéke, bekapcsolja a fájlokhoz az ügyfél-nél történő elrejtését az adott megosztásra.	yes	Megosztás
kernel oplocks	Boolean érték	Ha yes az értéke, azt jelzi, hogy a kernel támogatja az oplock zárolást.	yes	Globális
fake oplocks	Boolean érték	Ha yes az értéke, azt közli az ügyféllel, hogy megkapta a zárolást, de még sincsen zárva.	no	Megosztás
blocking locks	Boolean érték	Lehetővé teszi, hogy a zárolást kérő várakozzon a zárolás megadására.	yes	Megosztás
veto oplock files	Karakterlánc (fájlnevek listája)	Nem alkalmaz oplock zárolást az adott fájlokra.	Nincs	Megosztás
lock directory	Karakterlánc (teljes elérési út)	Megadja azt a helyet, ahol a Samba különböző fájlokat tárol, ide értve a zárolt fájlokat is.	A Samba make-fájlban megadottak szerint	Globális

**share modes**

A Sambában rendelkezésre álló legegyszerűbb zárolás a megtagadás (deny) módú, vagy share modes néven ismert zárolás, amelyeket olyan programok alkalmaznak, mint a szövegszerkesztők, hogy megakadályozzák a fájlok véletlen felülírását. A megtagadás módú zárolásokat az 5.9. táblázat foglalja össze.

5.9. táblázat. Az SMB megtagadás módú zárolásai

Zárolás	Leírás
DENY_NONE	Semmilyen fájlkérést sem tagad meg.
DENY_ALL	Az aktuális fájlra minden megnyitási kérést megtagad.
DENY_READ	Az aktuális fájlra vonatkozó minden, csak olvasásra vonatkozó megnyitási kérést megtagad.
DENY_WRITE	Az aktuális fájlra vonatkozó minden, csak írásra vonatkozó megnyitási kérést megtagad.
DENY_DOS	Ha olvasásra van megnyitva a fájl, mások is olvashatják, de nem írhatnak bele. Ha írásra van megnyitva, mások egyáltalán nem nyithatják meg.
DENY_FCB	Elavult.

A share modes beállítás, amely kikényszeríti a fenti zárolásokat, alapbeállítás szerint be van kapcsolva. A kikapcsolásához írjuk be a következőt:

```
[accounting]
share modes = no
```

Sohase tiltsuk le ezt az alapértelmezett zárolási mechanizmust, hacsak nincs nyomós okunk rá. Számos DOS és Windows alatt futó alkalmazás igényli a helyes működéséhez ezt a zároló mechanizmust, és nagyon zokon venné a hiányát.

### locking

A locking beállítás segítségével azt írhatjuk elő a Sambának, hogy alkalmazza-e egy ügyfél kérésére a kiszolgálón a bájttartományú zárolást vagy sem. A Samba normál Unix zárolásként valósítja meg a bájttartományú zárolásokat, és a jól megtervezett Unix processzeket megakadályozza abban, hogy felülírják a zárolt bájttartományt.

A beállítást az alábbiak szerint adhatjuk meg egy megosztásra:

```
[accounting]
locking = yes
```

Ha a locking beállításhoz a yes értéket rendeljük, a hozzáférési kérés teljesítése mindaddig késik, amíg bármelyik típusú zárolás birtokosa fel nem oldja a zárolást (vagy össze nem omlik). Ha viszont a no értéket rendeljük hozzá, akkor a fájlokra nem fog érvényesülni a bájttartományú zárolás, jöllehet úgy tűnik, hogy elvégezhető a fájlok zárolása és a zárolás feloldása. A beállításhoz alapértelmezés szerint a yes érték tartozik, de ezt megváltoztathatjuk, ha csak olvasható adathordozót használunk.

### strict locking

Ez a beállítás minden egyes, bájttartományú zárolásra irányuló fájlhozzáférést megvizsgál a zárolt bájtok tartományában. Erre általában nincs szükség, ha egy ügyfél következetesen tartja magát az összes zárolási mechanizmushoz. A beállítás alapértelmezett értéke a no, bár ez egy adott megosztásra megváltoztatható:

```
[accounting]
strict locking = yes
```

Azzal, hogy a beállításhoz a `yes` értéket rendeljük, az összes bájtzárolást használó fájlra kikényszerítjük a kötelező zárolásokat.

### *blocking locks*

A Samba a `blocking locks` beállítást is támogatja, ami a tartományi zárolások egyik alváltozatának tekinthető. Ha egy ügyfél azt tapasztalja, hogy egy bájt tartományhoz nem férhet hozzá, megadhatja azt az időtartamot, amennyi ideig hajlandó várakozni. A kiszolgáló ekkor félreteszi a kérelmet, és rendszeres időközönként megvizsgálja, hogy felszabadult-e már a bájt tartomány. Ha igen, akkor értesíti erről az ügyfelet, ha viszont időközben lejárt a várakozási idő, azt közli a felhasználóval, hogy nem tudta teljesíteni a kérését. Ezzel az eljárással megkímélhető az ügyfél attól, hogy folytonosan vizsgálja, megszűnt-e a zárolás.

A beállítást az alábbiak szerint tilthatjuk le egy megosztásra:

```
[accounting]
blocking locks = no
```

Ha a `yes` értéket rendeljük a beállításhoz, akkor a zárolás blokkolása a teljes fájlra érvényesül. Ha a beállításhoz a `no` értéket rendeljük, akkor a Samba a normál blokkolási eljárást használja. A beállítás alapértelmezett értéke a `yes`.

### *oplocks*

Ezzel a beállítással engedélyezhetjük vagy tilthatjuk az ügyfélnél az opportunista zárolások támogatását. A beállítás alapértelmezés szerint támogatja az ilyen zárolást, de ezt az alábbiak szerint meg is változtathatjuk:

```
[data]
oplocks = no
```

Ha rendkívül instabil a hálózati környezetünk, vagy ha sok olyan ügyfelünk van, aki nem tudja használni az opportunista zárolást, akkor célszerűbb nem használni ezt a beállítást. Akkor is ajánlatos tiltani az ilyen zárolást, ha biztosítanunk kell, hogy Unix alkalmazások (mint például a *vi*) és SMB ügyfelek is hozzáférhessenek ugyanazokhoz a fájlokhoz (hacsak nincs olyan szerencsénk, hogy az operációs rendszerünk a kernel `oplock` zárolásokat is támogatja, amint erről korábban olvashattunk).

### *fake oplocks*

Az opportunista zárolás megvalósítása előtt a Samba démonjai a `fake oplocks` (hamis `oplocks`) beállítás révén úgy tettek, mintha engedélyezték volna az `oplocks` zárolásokat. Ha engedélyezve volt ez a beállítás, az összes ügyfél azt a tájékoztatást kapta, hogy a fájlra kiadható az opportunista zárolás, és nem kapnak figyelmeztetést az egyidejű hozzáférésekről. Mióta a Samba lehetővé teszi a valóságos `oplocks` zárolásokat, ez a beállítás elvesztette a jelentőségét.

*kernel oplocks*

Ha egy Unix alkalmazás a Sambától függetlenül olyan fájlt próbál meg frissíteni, amelyet a Samba opportunistá módon zárolt egy Windows ügyfél elől, akkor ez a művelet valószínűleg sikerülhet neki, és erről sem a Samba, sem az ügyfél sohasem szerez tudomást. Ha viszont a helyi Unix operációs rendszer támogatja ezt a beállítást, a Samba tájékoztathatja a Unix alkalmazást az opportunistá módon zárolt fájlokról. Ekkor a Unix felfüggeszti a processz futását, a Sambán keresztül értesíti az ügyfelet, hogy írja vissza a módosításait, és csak ez után engedélyezi a fájl megnyitását. Lényegében ez azt jelenti, hogy a Unix operációs rendszer kernelje ugyanúgy képes kezelni az oplocks zárolásokat, mint maga a Samba. A kernel oplocks beállításnak ezt a viselkedését a következő módon engedélyezhetjük:

```
[global]
    kernel oplocks = yes
```

A Samba automatikusan észleli a kernel oplocks zárolásokat és használja is őket. E könyv írásának idején csak az SGI Irix 6.5.2f és a későbbi verziói támogatják az ilyen zárolásokat. A közeljövőben ugyanez várható a Linux és a FreeBSD operációs rendszerektől. A kernel oplocks zárolásokat nem támogató operációs rendszerek engedélyezik a Unix processznek egy fájl frissítését, de az ügyfélprogramok csak később vagy egyáltalán nem értesülnek ezekről a változásokról.

*veto oplock files*

A veto oplock files beállításban felsorolhatjuk azokat a fájlokat, amelyekre sohasem akarjuk alkalmazni az oplocks zárolásokat. A beállítás globális vagy megosztás hatókörben is használható. Például:

```
veto oplock files = /*.bat/*.htm/
```

A beállításhoz minták sorozatát rendeltük. Mindegyik listaelemnek törtjellel (/) kell kezdődnie és végződnie, még akkor is, ha a lista egyetlen elemből áll. Általános helyettesítőként csillag karakterek használhatók 0 vagy több karakter helyett. Minden egyes kérdőjel pontosan egy karaktert helyettesít. Javasolható, hogy mindazokra a fájlokra tiltsuk le az oplocks zárolásokat, amelyeket Unix alkalmazásoknak kell frissíteniük, vagy amelyekhez több processznek egyidejűleg kell hozzáférnie.

*lock directory*

Ezzel a beállítással (amely lock dir alakban is írható) annak a könyvtárnak a helyét adhatjuk meg, ahol a Samba a *dén*y (megtagadás) módon zárolt fájlokat tárolja. A Samba ebben a könyvtárban más fájlokat is tárol, így például a tallózólistákat és a megosztott memóriefájlokat. Ha engedélyezve van a WINS, akkor a WINS adatbázisa is ebbe a könyvtárba íródik. A beállításhoz tartozó alapértelmezett könyvtárat a Samba make-fájl programja határozza meg, ami tipikusan a */usr/local/samba/var/locks*. Ezt a helyet a következő módon felülbírállhatjuk:

```
[global]
    lock directory = /usr/local/samba/locks
```

Az alapértelmezett könyvtárat általában nem kell megváltoztatnunk, hacsak nem akarjuk a zárolt fájlokat egy, a szokásoknak még inkább megfelelő helyre írunk, például a */var/spool/locks* könyvtárba.