

Samba活用テクニック

Windows 7対応状況



たかはし もとのぶ (高橋基信)

monyo@samba.gr.jp

<http://wiki.samba.gr.jp/>

セミナーのねらい

- 対象となる方

- Sambaの基本的な設定はできる方
- 最新動向 (Windows 7対応など) が知りたい方
- 「Sambaのすべて」が積ん読になっている方 (^_^;) → Sambaのすべての内容を易しく紹介します
- 「Sambaのすべて」は書いてある内容が古い！と感じられている方 (^_^;) → 最新版のSambaの機能も交えて紹介します
- いつもSambaドメインやWinbindばかりで、もう飽きた！という方

- 目的

- ドメインやWinbindなど、大掛かりな機能ばかり紹介されていて、細かい機能強化がおきざりになっている → Sambaの細かい機能を、時間の許す限り紹介

セミナーの概要

- Sambaのリリースポリシー
- Samba活用テクニック
 - インストール、基本設定編
 - 認証編
 - アクセス制御編
 - 共有のテクニック編
- 最新動向
 - Windows 7対応
 - Samba 3.2/3.3/3.4/3.5の新機能
 - Samba 4.0系列開発動向

Sambaのリリースポリシー

- リリースポリシー
 - 半年ごとにバージョンアップ。18ヶ月間サポート
 - Samba 3.2系列のサポートは2010年1月に終了予定
 - Samba 3.5系列が2010年1月にリリース予定→3月延期
 - Samba 3.3系列のサポートは2010年7月に終了予定
 - Samba 3.4系列のサポートは2011年1月に終了予定
- 最新バージョンはSamba 3.4.5
 - しかし、ディストリビューションに同梱されているのは、サポートが終了したSamba 3.0系列も未だに多い(最終: Samba 3.0.37)

Samba活用テクニック: インストール・基本設定編

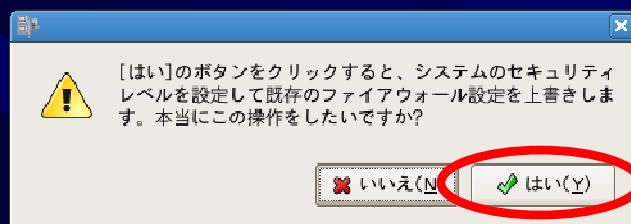
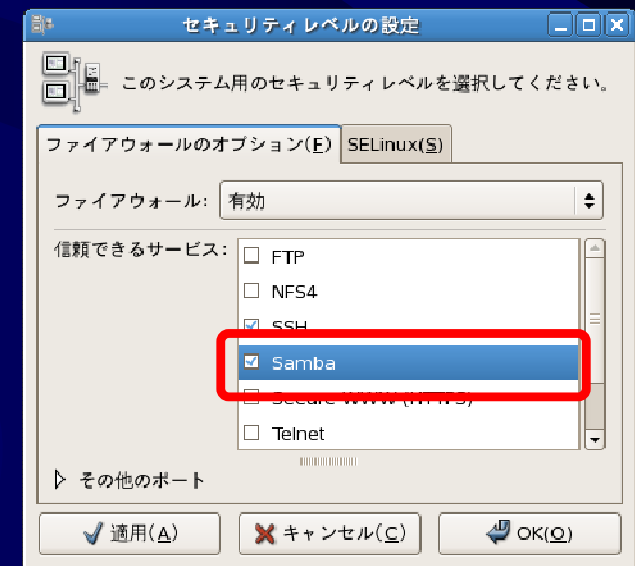
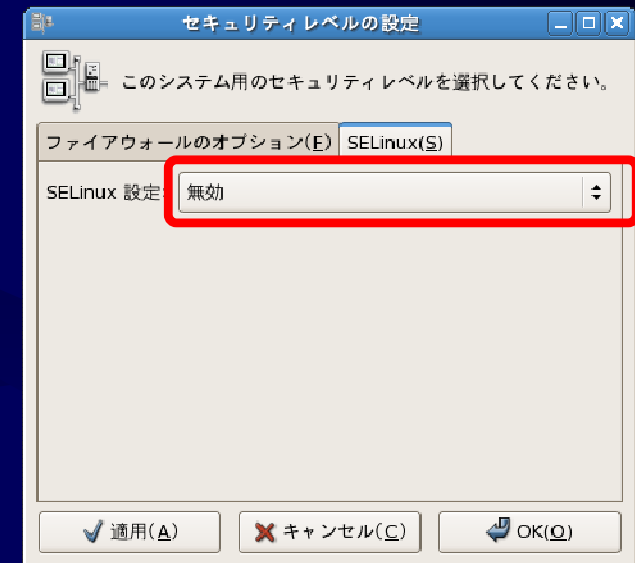
セキュリティ設定の留意点

• SELinuxの無効化

- SELinuxを「無効」(もしくは「Permissive」)に
実際は「有効」でもSambaの実行はできる
ただし、設定の難易度が高く、上級者向き

- ファイアウォールでSambaを「信頼できるサービス」に
(もしくはファイアウォールを「無効」に)

- 警告メッセージには「はい」を選択



「管理」－「セキュリティレベルとファイアウォールの設定」

日本語関連の注意点

- 日本語のファイル名(など)を使用する場合は、必ず設定を確認する

- 海外のLinuxディストリビューションなどでは、必ず設定が必要な部分

- `dos charset = CP932`

- 日本語利用を指定【必須】

- `unix charset = UTF-8`

- `smb.conf`ファイルや、ファイルシステムで用いる文字コード【環境依存】
 - UTF-8(デフォルト)が無難
 - EUCにしたい場合は「EUCJP-MS」を、シフトJISにしたい場合は「CP932」を指定する

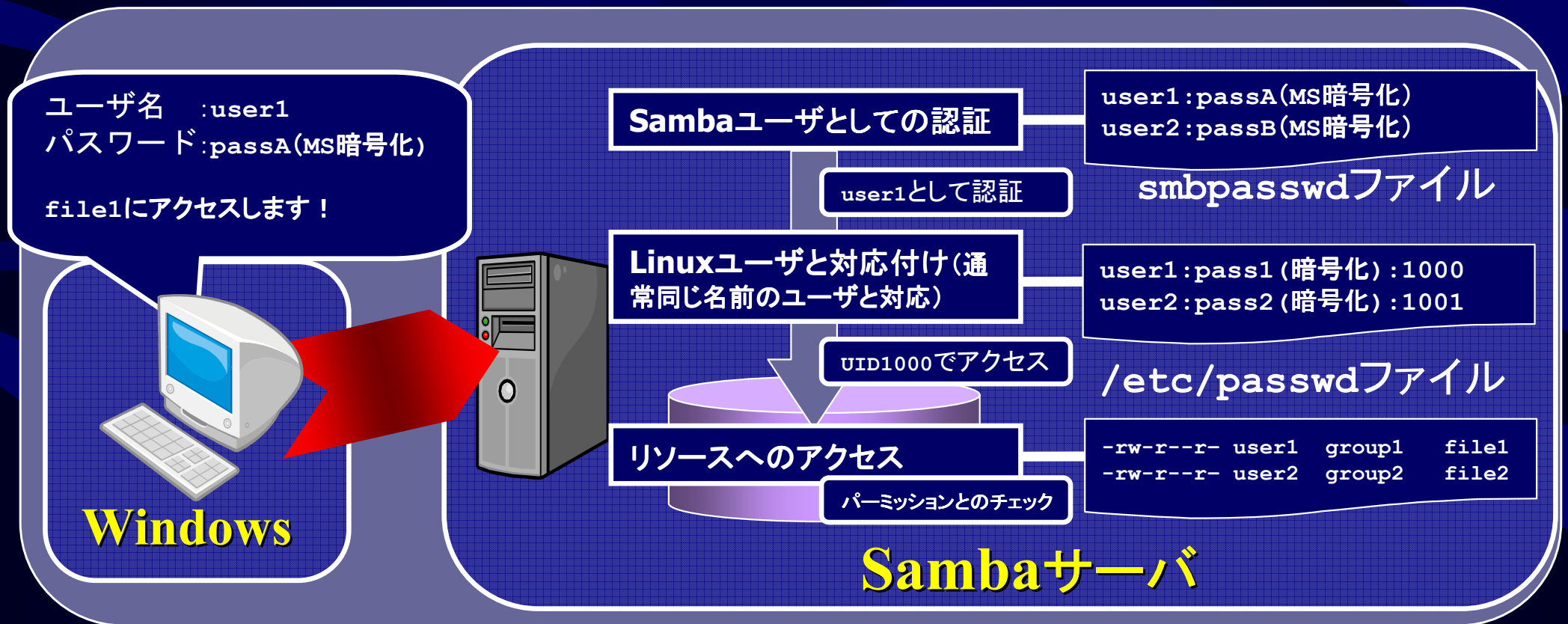
```

root@centos52:/etc/samba
ファイル(F) 編集(E) 表示(V) 端末(T) タブ(B) ヘルプ(H)
#----- Global Settings -----
[global]
dos charset = CP932
unix charset = UTF-8
#----- Network Related Options -----
#
# workgroup = NT-Domain-Name or Workgroup-Name; eg: MUDHEARTH
#
# network name can be used to specify a server name for this host
#
# Interfaces lets you configure Samba to use multiple interfaces
# If you have multiple network interfaces then you can list the ones
# you want to listen on (never omit localhost)
#
# Hosts Allow/Hosts Deny lets you restrict who can connect, and you can
# specify it as a per share option as well
#
workgroup = mygroup
server string = Samba Server Version %v
    
```

globalセクション先頭に追記

Sambaユーザはなぜ必要？

- 答：パスワードを暗号化しているから
 - Windowsの暗号化形式とLinux用の暗号化形式は形式が異なる → 共用できない！



Samba活用テクニック: 認証編

Windowsマシンからのパスワード変更

- Windowsマシンと同様に可能
 - 「ログオン先」欄にSambaサーバのコンピュータ名を直接入力する

- スクリプトからも可能



```
Set objUser = GetObject("WinNT://SAMBA32A/smbuser")
objUser.ChangePassword "damedame", "damesugi"
```

UNIX・Sambaユーザのパスワード同期

• Sambaユーザ→UNIXユーザ

```
[global]
```

```
unix password sync = yes ←パスワード同期の有効化【必須】
```

```
pam password change = yes ←PAMによるパスワード同期を  
有効化【場合による】
```

- PAMが有効でない場合は、passwd program/パラメータとpasswd chat/パラメータを適切に設定
- パスワード変更は普通に実施
 - pdbeditやsmbpasswdコマンドを使用

UNIX・Sambaユーザのパスワード同期

- UNIXユーザ→Sambaユーザ
 - pam_smbpassモジュールをインストール、有効化
samba-commonパッケージやlibpam-smbpassパッケージをインストールの上、以下のような設定を/etc/pam.d/system-authに追加(CentOS 5.2)

```
password ...
```

```
password optional pam_smbpass.so
```

←パスワード同期の有効化【必須】

- rootによるパスワード強制変更時は同期できない
→ 現在のパスワード情報が提供されないため

Windows・Sambaユーザのマッピング(1)

- Windowsユーザ名とSambaユーザ名のマッピングを任意に設定できる
 - 静的マッピング: `username map`
指定したファイルにマッピング情報を記述

```
# Unix_name = SMB_name1 SMB_name2 ...
smbuser = domsmbuser1 "dom¥smbuser1"
monyo    = "Takahashi Motonobu" "高橋基信"
```

Username Mapファイルの例

「Takahashi Motonobu」や「高橋基信」というWindowsユーザを「monyo」というSambaユーザにマッピングしている

Windows・Sambaユーザのマッピング(2)

- 動的マッピング: `username map script`
スクリプトにより、柔軟なマッピングが可能

```
[global]
```

```
username map script = ¥
```

```
/usr/local/samba/sbin/mapusers.sh ←スクリプト名【必須】
```

Windowsユーザ名を引数にとり、マッピングされたUNIXユーザ名を出力するようなスクリプトのフルパスを指定

```
#!/bin/sh
if [ "$1" == "TAKAHASHI Motonobu" ]; then
    echo monyo
elif [ "$1" == "高橋基信" ]; then
    echo monyo
fi
```

スクリプト例

複雑なパスワードの強制

- 外部スクリプトにより、柔軟な設定が可能
 - `check password script` パラメータ

```
[global]
```

```
check password script = ¥
```

```
/usr/local/sbin/chkpasswd.sh ←スクリプト名【必須】
```

パスワードを標準入力から受け取り、適切なら0、不適切なら0以外を返却するスクリプト

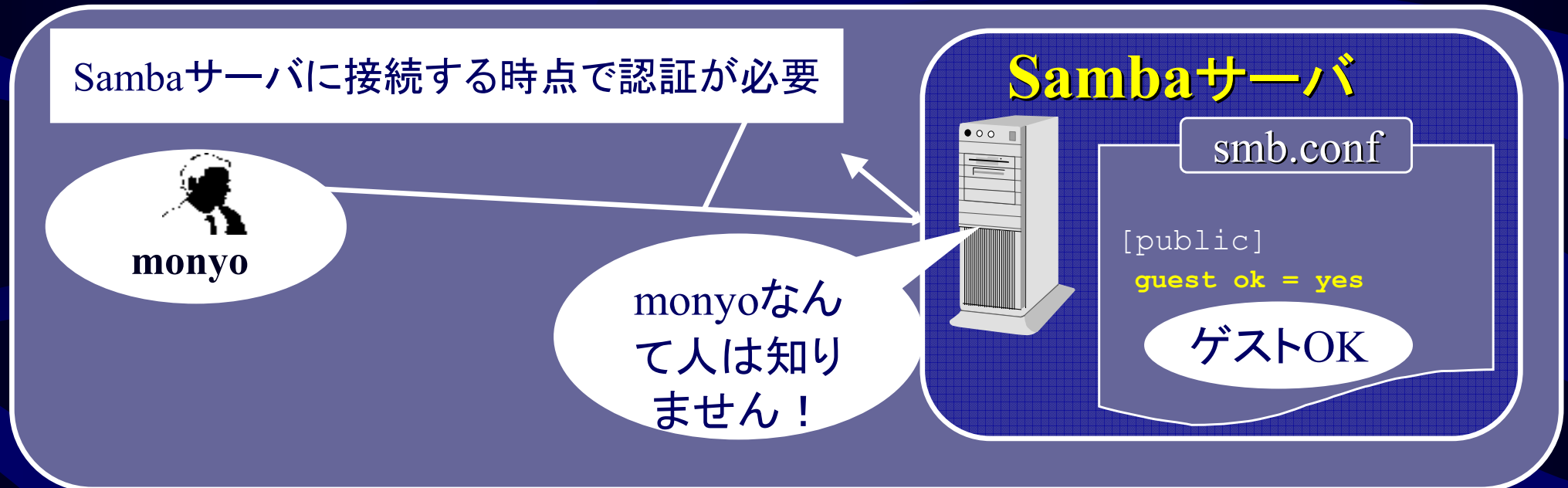
- `crackcheck` コマンド

- 上記パラメータ用にSambaに同梱されているコマンド
 - `-c`: 「Windows相当のチェック」を行う(デフォルト:行わない)
 - `-s`: `cracklib`によるチェックを行わない(デフォルト:行う)

```
$ echo PaSsW0rd | /usr/sbin/crackcheck -c -d pw_dict; echo $?
ERR - it is based on a dictionary word
253
```

ゲストアクセス(1):概要

- 存在しないユーザは原則接続できない
 - `guest ok = yes`だけではゲストアクセスできない



• ゲストアクセス機能

- 認証できないユーザのアクセスをゲストアクセスとして認証する機能

ゲストアクセス(2):設定

- ゲストアクセスを可能とする
 - Windowsのguestアカウントと同様の機能
- map to guest = Bad User
 - 存在しないユーザからのアクセスをゲストアクセスとみなす
 - 存在しないユーザからのアクセスを許可しない場合はNeverにする
- guest ok = Yes/No
 - 各ファイル共有でゲストアクセスを許可するならYes
- guest only = Yes/No
 - 全員のアクセスをゲストアクセスとみなす場合はYes

```
[global]
    map to guest = Bad User
    guest account = nobody

[share1]
    guest ok = yes
    guest only = yes
```

smb.confファイル

ゲストアクセス(3): 設定例

- Sambaユーザが存在せず、認証できなかった場合でもアクセス(書き込み、読み取り)可能な共有

```
# mkdir -p /home/public ←共有したいディレクトリを指定
# chmod 755 /home/public
# chown smbquest /home/public ←ユーザは予め作成しておく
```

- 全てのユーザがsmbquestユーザとしてアクセスする
- writeable = no にすれば読み込み専用共有
- guest only = no にすれば認証に成功したユーザは実名ユーザとしてアクセス

smb.confの設定例

```
[global]
    map to guest = bad user
    guest account = smbquest
    ...
[share1]
    path = /home/public
    writeable = Yes
    guest ok = Yes
    guest only = Yes
```

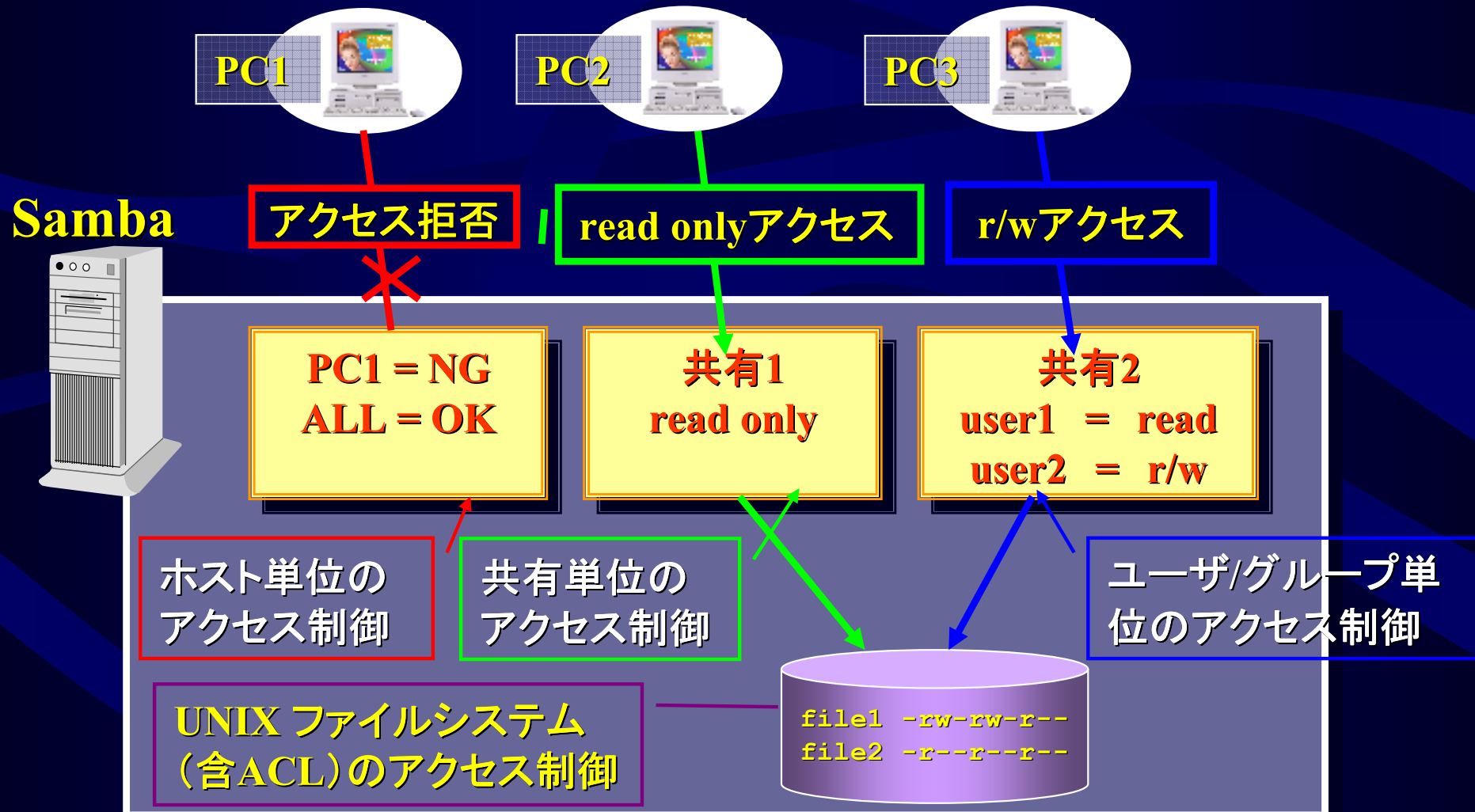
ゲストアクセス(4):補足

- Sambaサーバがドメインのメンバサーバかつ
Winbind機構を使用していない環境において.....
 - ドメインで認証済かつSambaサーバにアカウントのないユーザからアクセスがあった場合.....
- Samba 3.0.0～Samba 3.0.14aまで
 - 認証失敗
- Samba 3.0.20
 - 「map to guest = Bad Uid」(新設)でゲスト認証
 - 「map to guest = Bad User」(従来)で認証失敗

Samba活用テクニック: アクセス制御編

アクセス制御: 概要(1)

Windows クライアント(user2がログオン中)



アクセス制御: 概要(2)

- ホスト単位でのアクセス制御
 - IPアドレスでのアクセス制御
 - `hosts allow`, `hosts deny`
- 共有単位でのアクセス制御
 - ユーザ、グループ単位でのアクセス制御
 - `valid users`, `invalid users`
 - 共有レベルでのアクセス制御
 - `read only`, `write list`, `read list`
 - IPアドレスでのアクセス制御、共有のACL
- ファイルシステムレベルでのアクセス制御
 - 伝統的なパーミッション、ACL(サポートされている場合)

アクセス制御: IPアドレスによる制御

- IPアドレス(ホスト名)ベースで、アクセス可能なホストを制限することが可能
 - hosts allow
 - アクセスを許可するホストを規定
 - hosts deny
 - アクセスを拒否するホストを指定
 - ホスト単位、共有単位で設定することが可能
 - [global]セクションで設定すると、ホスト単位でのアクセス制御になる

smb.confの設定

```
[global]
...
hosts allow = 192.168.10.

[share]
path = /home/samba/share
writeable = Yes
hosts allow = 192.168.1.
```

アクセス制御: 読み書きレベルの制御

- デフォルトのレベルを指定後、例外のユーザやグループを個別に指定する
- read only (writeable)
 - 共有を読み取り専用にする(書き込み可能にする)、デフォルトYes
- `write list = (ユーザのリスト)`
 - 読み取り専用の共有に書き込み可能なユーザ(グループ)のリスト
- `read list = (ユーザのリスト)`
 - 書き込み可能な共有に対して読み取り専用でしかアクセスできないユーザ(グループ)のリスト

```
[share1]
...
read only = yes
write list = monyo, ¥
@admins
```

smb.confファイル

アクセス制御: ユーザ単位の制御

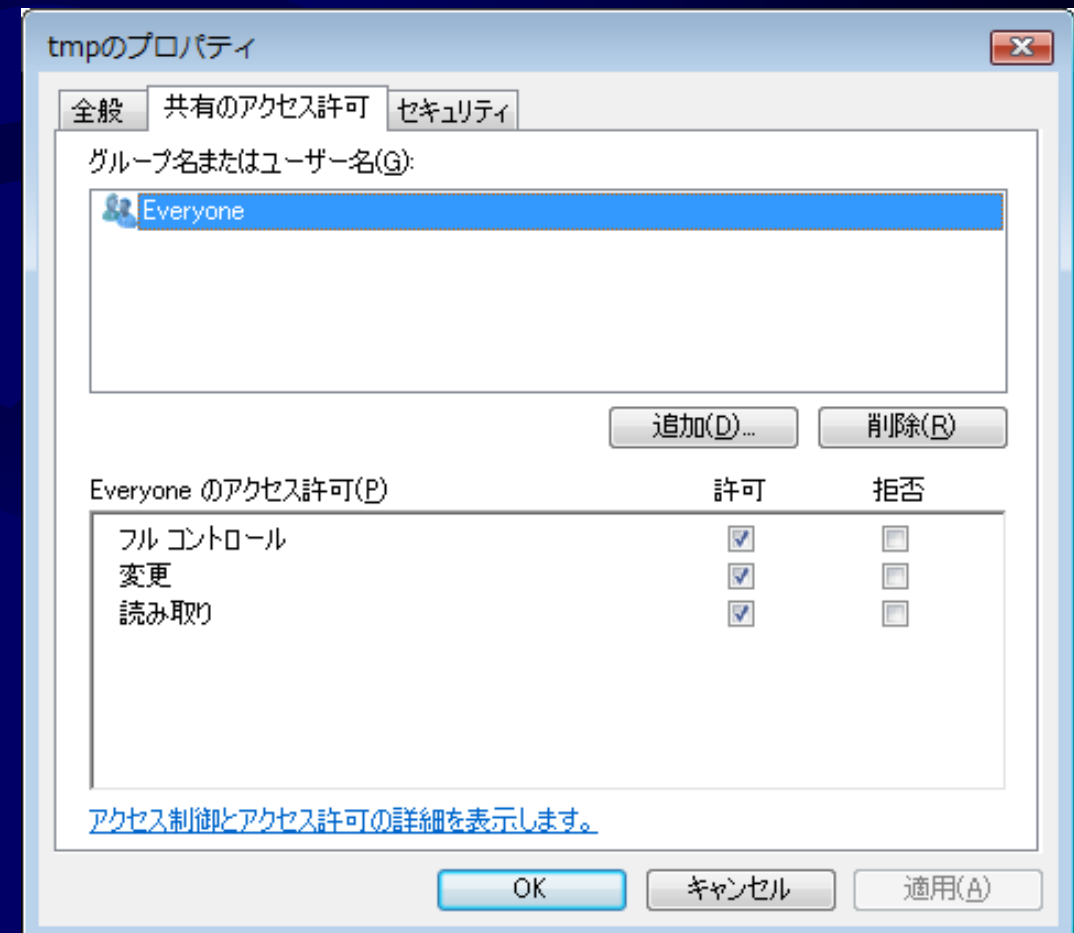
- 共有にアクセス可能なユーザを指定する
- valid users = (ユーザリスト)
 - 共有にアクセス可能なユーザ (グループ) のリスト
- invalid users = (ユーザリスト)
 - 共有にアクセスを拒否されるユーザ (グループ)
- admin users = (ユーザリスト)
 - 共有に対して、root権限でのファイル操作が可能なユーザ (グループ)
- Winbind環境では、「domain¥username」形式で指定すること

```
[share1]
...
valid users = @user
invalid users = guest
admin users = monyo
```

smb.confファイル

アクセス制御: 共有ACLによる制御(1)

- Windows側から共有単位でアクセス可能なユーザやグループを制御
- 指定できるのは、
 - Sambaユーザ
 - `net groupmap`コマンドで登録済のグループ
- サーバ上からは、`sharesec`コマンドで設定



アクセス制御: 共有ACLによる制御(2)

- sharesecコマンドによる制御
 - sharesec [-a|-r] ACL情報 共有名

```
# sharesec -a S-1-5-21-2535719703-1779805756-2758924810-1003:ALLOWED/0/CHANGE tmp
```

↑ tmp共有について、S-1-5...というSIDのユーザに対して、「変更」の許可ACLを追加

```
# sharesec -v tmp
```

REVISION:1

OWNER: (NULL SID)

GROUP: (NULL SID)

ACL: S-1-5-21-2535719703-1779805756-2758924810-513:ALLOWED/0/READ

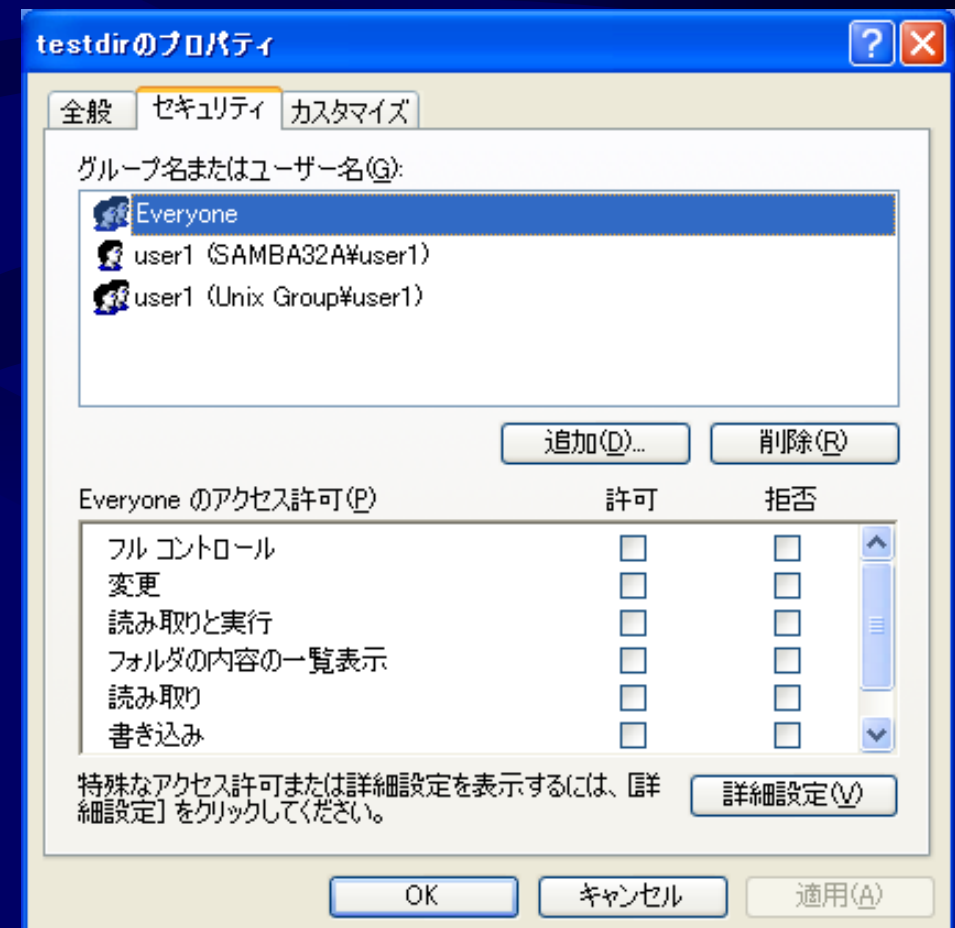
ACL: S-1-1-0:ALLOWED/0/FULL

↑ tmp共有のACLを表示

- ACLの値は別途確認しておく必要がある

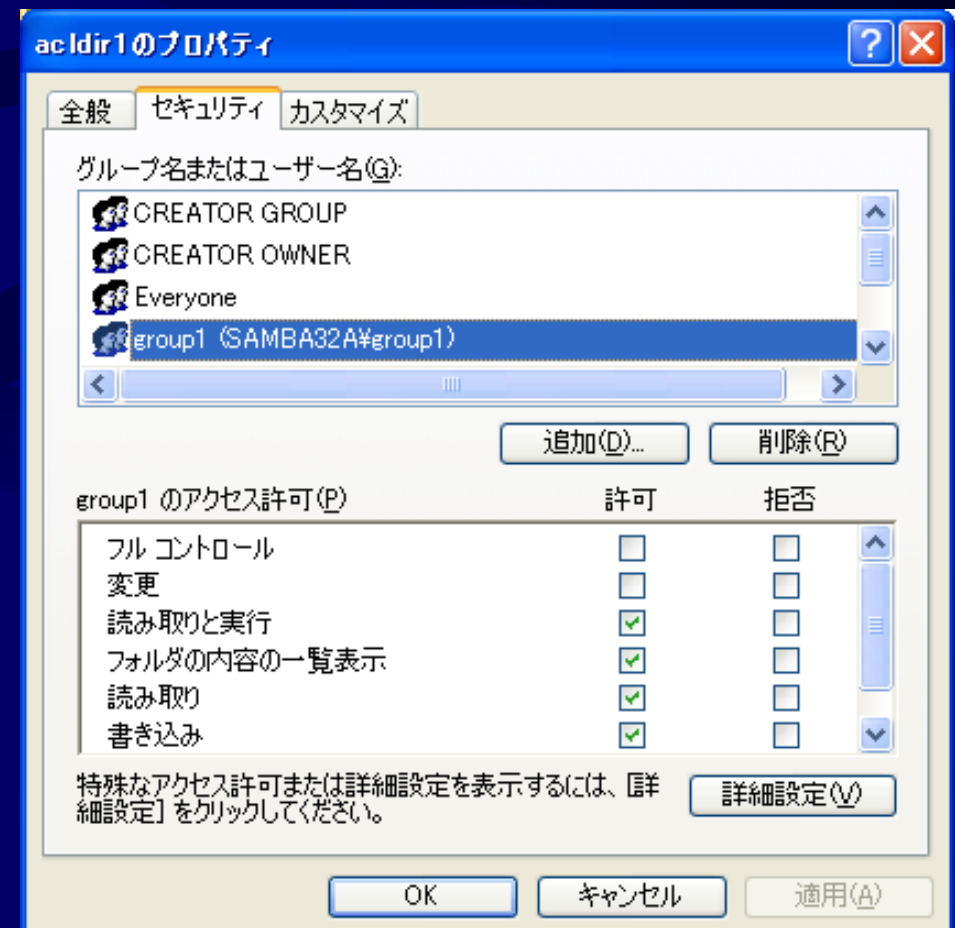
アクセス制御:パーミッションによる制御

- UNIXファイルシステムのパーミッションで許可されていない操作はできない
 - たとえば、書き込みできないファイルに書き込むなど
- パーミッション設定は、右のACL画面から参照、変更することも可



アクセス制御: ACLによる制御(1)

- ファイルシステムのACL機能を活用
 - ACLに対応したファイルシステム & プラットフォームで可能
- Linuxの場合、ファイルシステムマウント時にac1オプションの付加が必要な場合あり



アクセス制御: ACLによる制御(2)

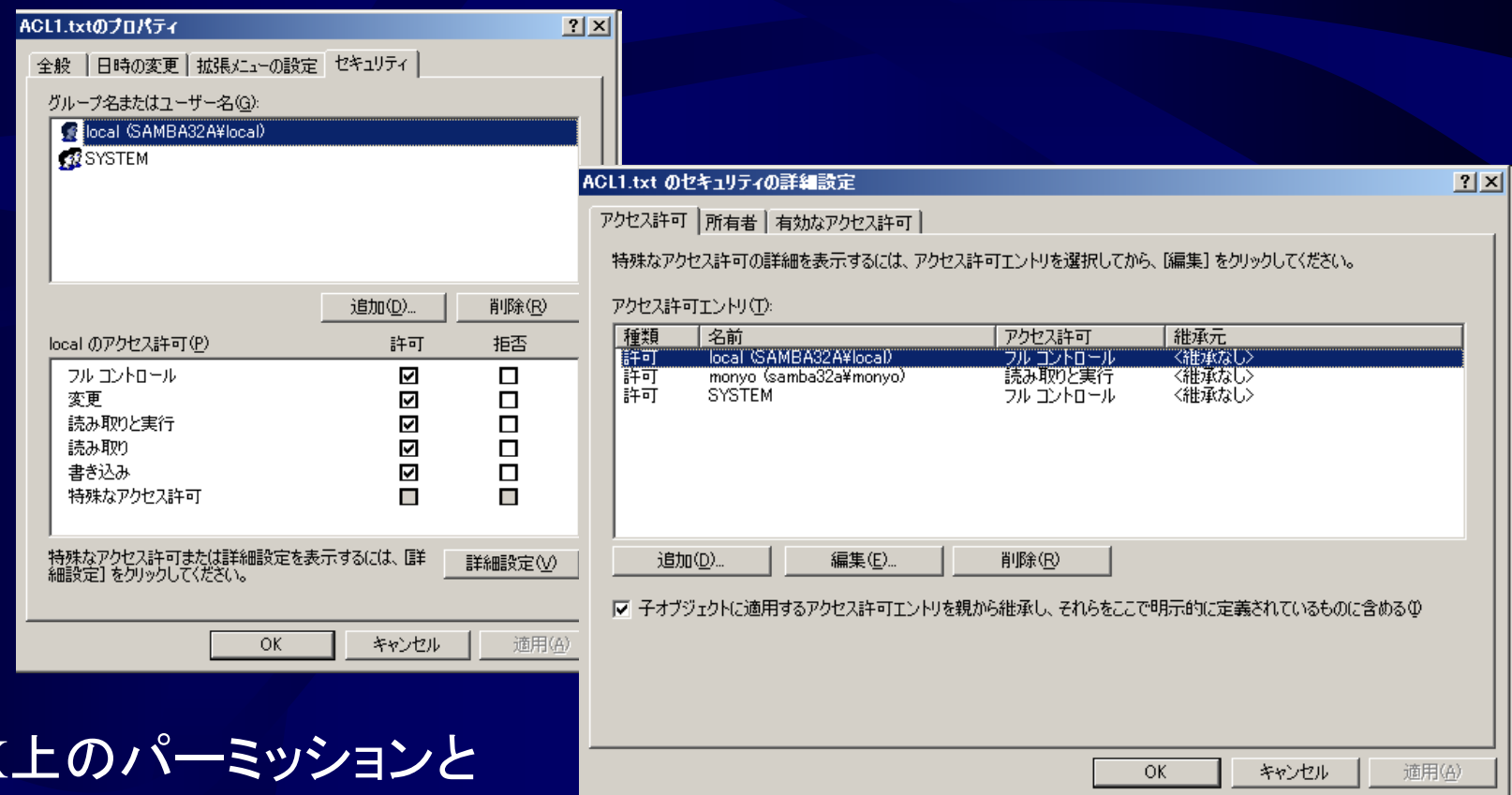
- NTFS完全互換のACL機能を利用
 - Samba 3.3.0以降で実装された
`vfs_acl_tdb`(ACL情報をTDBに格納)か、
`vfs_acl_xattr`(ACL情報を拡張属性に格納)モジュールを有効化

```
[share1]
...
vfs objects = acl_xattr
```

- ただし、まだ実験段階(試した限り動作しているっぽいけど、ファイルシステム上のACLとの連携に難あり)

アクセス制御: ACLによる制御(2)

- POSIX ACLへのマッピングではなく、拡張属性を活用することでNTFS完全互換のACLを実現



現状はUNIX上のパーミッションとの相互運用性に難あり

アクセス制御: 設定例(1)

- アクセス不可、読み取りのみ、書き込み可能の三段階でアクセスを制御する例

- 192.168.1.0/24からのみアクセス可能

- usersグループのアカウントは、読み取り、書き込みともに可能

- guestグループのアカウントは、読み取り可能

- それ以外のアカウントは、アクセス不可

smb.confの設定

```
[share2]
...
valid users = @guests, ¥
@users
read only = yes
write list = @users
hosts allow = 192.168.1.
```

アクセス制御: 設定例(2)

- project-aグループに所属するユーザ(のみ)が読み書き可能な共有フォルダの設定例

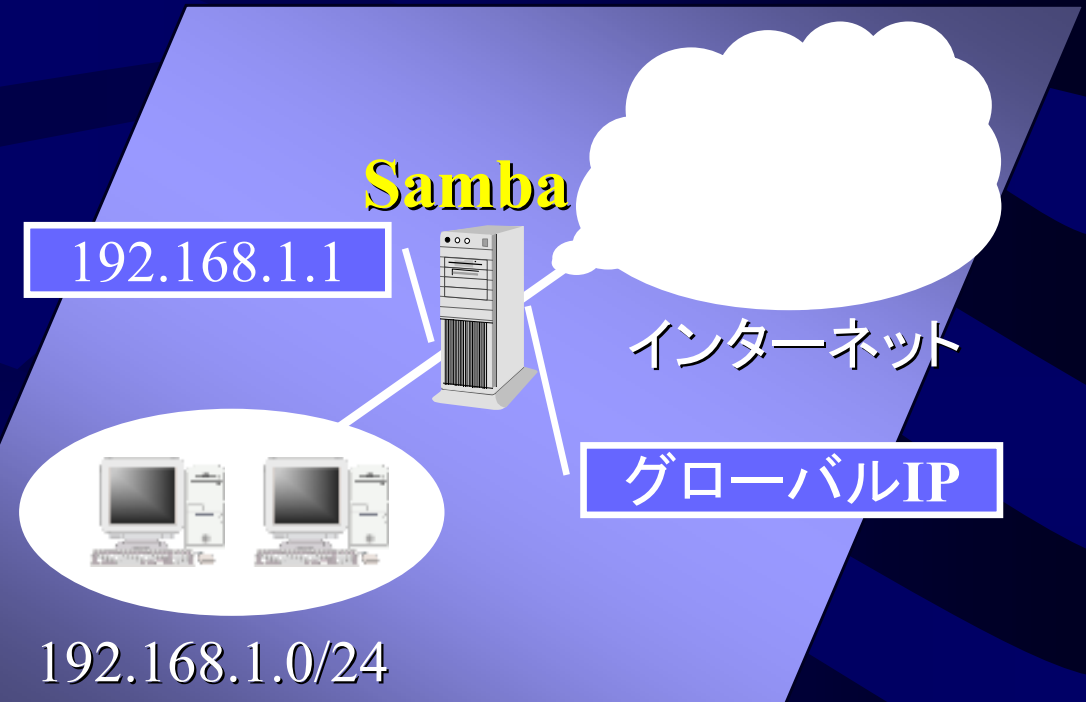
```
# mkdir -p /home/samba/project-a ←共有したいディレクトリを指定
# chmod 2775 /home/samba/project-a ←グループ書き込み可能、setgid設定
# chgrp project-a /home/samba/project-a
```

```
[project-a]
path = /home/samba/project-a
writeable = yes ←書き込み可能にする
valid users = @project-a ←project-aグループのみアクセス可能
force group = project-a
force create mode = 664
force directory mode = 775
```

全員がproject-aグループとしてアクセス、作成したファイルやディレクトリは互いに書き込み可能

アクセス制御: マルチホームでの注意点

- Sambaサーバを起動するIPアドレスは明示的に制御する必要がある
 - Sambaのデフォルトでは、すべてのIPアドレスで起動される
 - 特定のIPアドレスでのみ動作させたい場合は、要設定



マルチホームの例

192.168.1.0/24とグローバルIPのネットワークに接続されている

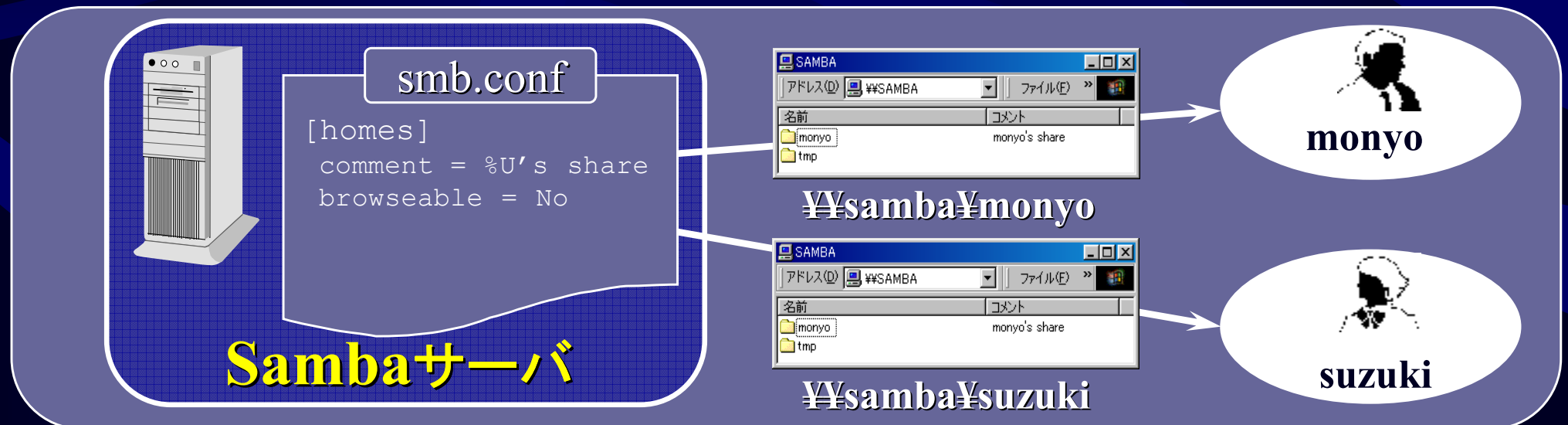
```
[global]
interfaces = 192.168.1.1/24
bind interfaces only = yes
socket address = 192.168.1.255
```

smb.confの設定(該当部分のみ)

Samba活用テクニック: 共有のテクニック編

ユーザホーム機能(1): 概要

- ユーザーホーム機能
 - 各ユーザのホームディレクトリ(ユーザ専用ディレクトリ)を自動的に共有する機能
 - **自分のホームディレクトリ共有だけが表示される**
設定次第で、他人のホームディレクトリ共有にアクセスしたり、表示したりすることも可能



ユーザホーム機能(2): 設定

- homesセクションを設定(設定済の場合も多い)

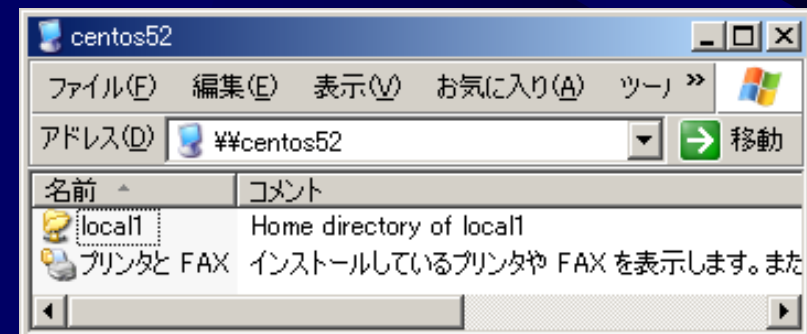
[homes]

browseable = No ← [homes] 自体は不可視にする【必須】

writeable = Yes ← 更新可能にする【必須】

valid users = %S ← 所有者以外のアクセスを禁止する【任意】

- 必ず **browseable = No** にする
- comment行は削除することを推奨
 - デフォルトのコメントが表示される
- 自分以外のホームディレクトリも **¥¥server¥username** 形式でアクセス可能



ユーザlocal1でアクセスしたところ

ユーザホーム機能(3):応用

- 「.」ファイルを参照されたくない場合
 - veto filesを使う

[homes]

...

veto files = /.??*/

「.」から始まるファイルへのアクセスを禁止する(/.*/とすると親ディレクトリにアクセスできなくなる)

- ユーザホーム機能の提供するホームディレクトリの位置を変更する

[homes]

...

path = %H/smbdir

~/smbdir以下を公開することで、ホームディレクトリ直下の「.」から始まるファイルへのアクセスを回避する

特定のユーザ以外参照できない共有

- ユーザによって提供する共有を変更することも可能(Sambaのみ?)

- group1がプライマリグループのユーザに対してのみ、share1共有が有効(available)に
- 多数の共有が一覧されるのを防ぐ目的でも便利

smb.confファイル

```
[share1]
...
available = no
include = share1.%g
```

share1.group1ファイル

```
available = yes
```

UNIX上のファイルを隠す

- 一部のファイルへのアクセスを禁止する設定

- veto files

指定したファイルは存在しないかのように扱われる

```
[share]
```

```
...
```

```
veto files = /.??*/
```

```
delete veto files = yes ←
```

アクセスを禁止されたファイル
が含まれるディレクトリの削除
を許可する

- hide dot files / hide files

ドット「.」から始まるファイル/指定したファイルを隠し
ファイルにする

- 隠しファイルが表示される設定になっていると無意味

UNIX上のファイルを隠す

• アクセスできないファイルを隠す設定

– `hide unreadable = yes`

アクセス権のないファイルが存在しないかのように扱われる

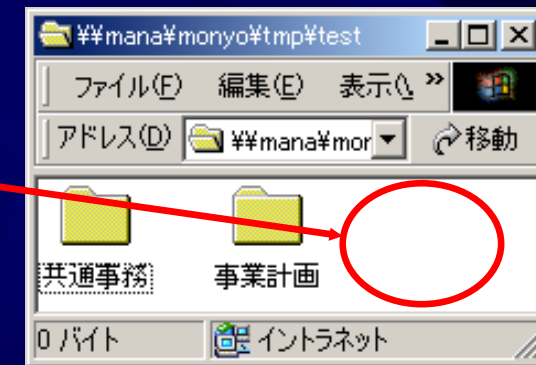
[share]

...

`hide unreadable = yes`

← アクセス権のないファイル (ディレクトリ) の存在を隠す

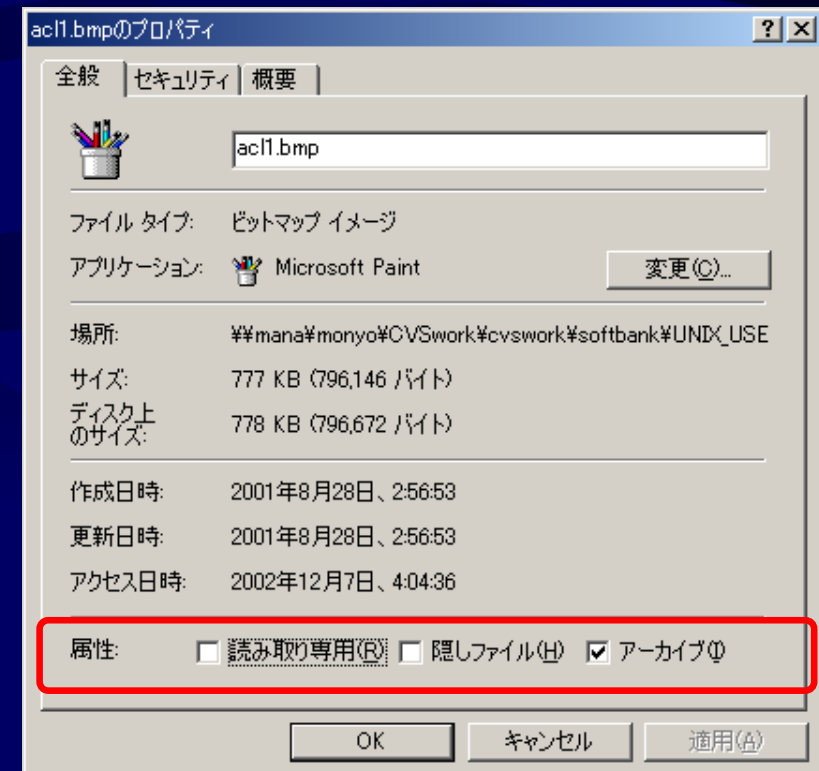
```
[mana:/home/monyo/tmp/test]ls -l
合計 12
drwxrwx---  2 monyo  common    4096 10月 21 17:39 共通事務/
drwxrwx---  2 monyo  common    4096 10月 21 17:38 事業計画/
drwxrwx---  2 root   jin.ji    4096 10月 21 17:39 人事情報/
[mana:/home/monyo/tmp/test]
```



アクセス権のないディレクトリが表示されない

MS-DOS互換の設定(1)

- ファイル属性のサポート
 - 「アーカイブ、読み取り専用、システム、隠し」の4つ
- ファイルの実行属性を使って、情報を保存
 - map archive (所有者)
 - map system (グループ)
 - map hidden (その他)
- 拡張属性を使って、情報を保存
 - store dos attributes



MS-DOS互換の設定(2)

- Visual C++などでビルドする際の設定
 - 不要なビルドを行わないようにする設定

[share] ファイル・ディレクトリの作成時刻の設定を、MS-DOS互換の2秒単位にする
...

dos filetime resolution = yes

fake directory create time = yes ← ディレクトリ作成時刻を常に1980/01/01にする

- JP402160 [NT] NTFSからFATへのファイルのコピー時に日時が変わる
- UNIXには、ディレクトリ作成時刻の概念がない
- **delete readonly**
 - 読み取り専用ファイルの削除を許可する (MS-DOS互換の仕様の実現)

代替データストリーム対応

- Samba 3.0系列

- 代替データストリーム非対応

```
X:¥>echo test > test.txt:test1
```

指定されたパスが見つかりません。

- Samba 3.2.0以降

- vfs_streams_xattrモジュールによる対応
 - Linuxファイルシステムの拡張属性を利用
ファイルシステムマウント時にuser_xattrオプションの付与が必須
 - vfs_streams_depotモジュールによる対応
 - 拡張属性はstreams_depot:directoryパラメータで指定したディレクトリ(デフォルトは共有直下の.streams)に保存

日本語の活用

- 日本語の共有名、コメントなど

```
[global]
```

```
server string = Samba %v サーバです
```

```
[共有1]
```

```
comment = テスト共有 No.1
```

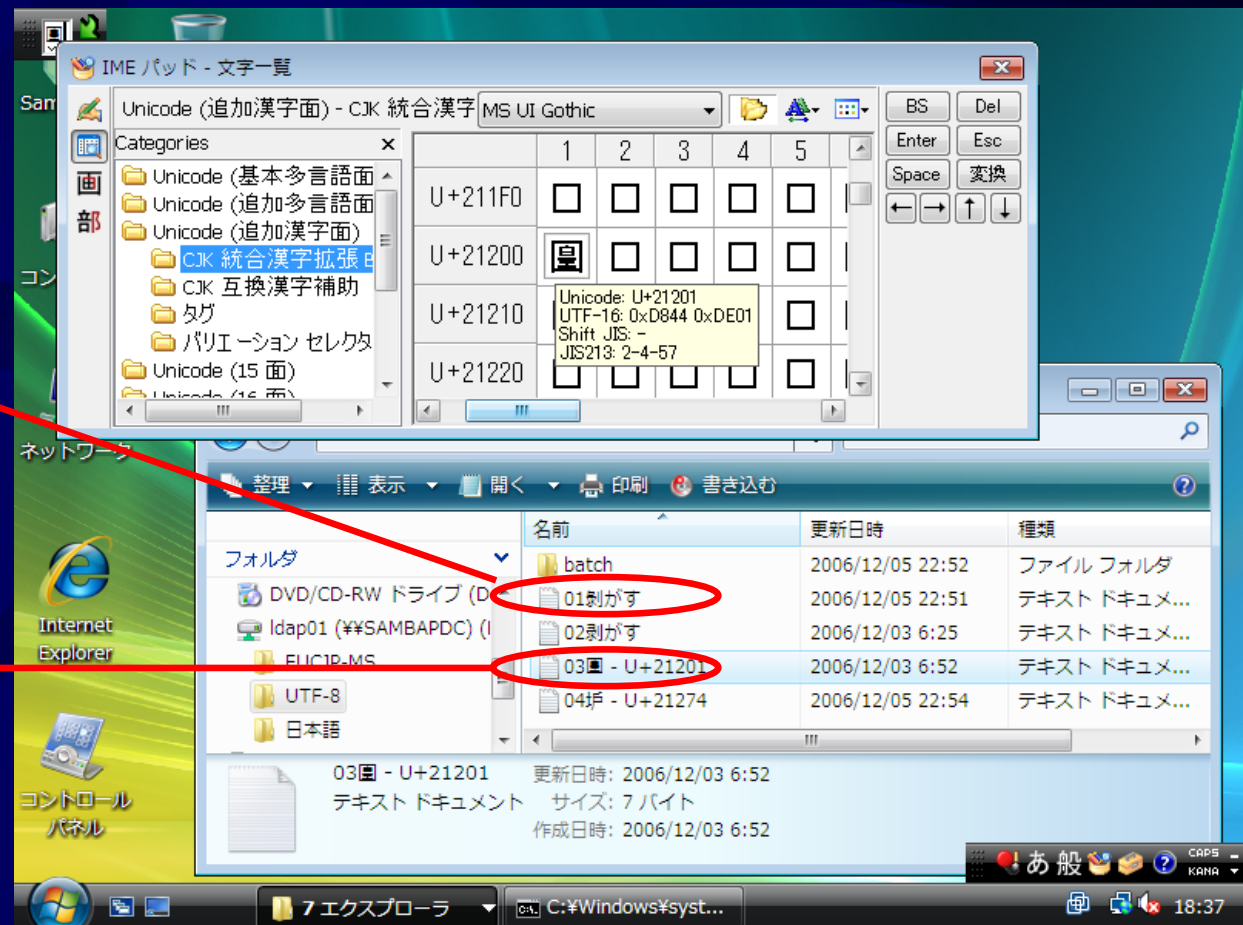
- dos charset/unix charsetパラメータを適切に設定しておくことが必須
- unix charsetパラメータに記述した文字コードで `smb.conf`を記述する

日本語の活用: JIS 2004対応

- Samba 3.0.8 以降では「対応」
 - `unix charset = UTF-8`が必須

JIS2004で追加された文字を使ったファイル名を扱える

UTF-16でのみ扱える
JIS2004で追加された文字にも対応



ファイルアクセスの監査

- vfs_full_auditモジュールなどにより監査

```
vfs objects = full_audit
```

```
full_audit:prefix = %u|%I|%S
```

```
full_audit:success = connect disconnect ...
```

```
full_audit:failure = connect disconnect ...
```

- ログ出力例

```
Dec 30 03:39:51 samba32a smbd_audit: local1|192.168.135.1|connect|ok|tmp
Dec 30 03:39:51 samba32a smbd_audit: local1|192.168.135.1|getxattr|fail (Operation not
permitted)|.|user.DOSATTRIB
Dec 30 03:42:11 samba32a smbd_audit: local1|192.168.135.142|open|fail (Permission
denied)|r|testdir
Jan 12 18:50:41 lenny smbd_audit: monyo|192.168.135.233|shared|closedir|ok|
Jan 12 18:50:41 lenny smbd_audit: monyo|192.168.135.233|shared|closedir|ok|
Jan 12 18:51:20 lenny smbd_audit: monyo|192.168.135.23|shared|disconnect|ok|shared
```

Sambaサーバへのアクセスの監査

- utmpパラメータにより実現
 - Sambaサーバへの接続履歴をutmpファイルに出力

```
utmp = yes
```

ログ出力例

```
last | grep smb
monyo      smb/1      192.168.135.1    Tue Nov 23 04:41    still logged in
smbguest   smb/2      misa             Sun Jun 17 10:27 - 11:36    (01:08)
monyo      smb/0      misa             Sun Jun 17 10:27 - 10:27    (00:00)
smbguest   smb/0      mayuka           Sat Jun 16 23:05 - 23:15    (00:10)
smbguest   smb/1      yukako           Sat Jun 16 22:48 - 11:36    (12:48)
```

※前述したvfs_full_auditでもconnectを監査することで同様の情報は取得できる

ゴミ箱

- recycleモジュールにより、削除したファイルの復活が可能

```
vfs objects = recycle
```

```
recycle:maxsize = 2000000
```

```
recycle:repository = .trash
```

```
recycle:keeptree = yes
```

```
recycle:touch = yes
```

- パラメータにより細かい制御も可能

高度な機能

- DFS
- ボリュームシャドウコピー
- オフラインファイル
-
- Windowsサーバと同等の高度な機能も実装

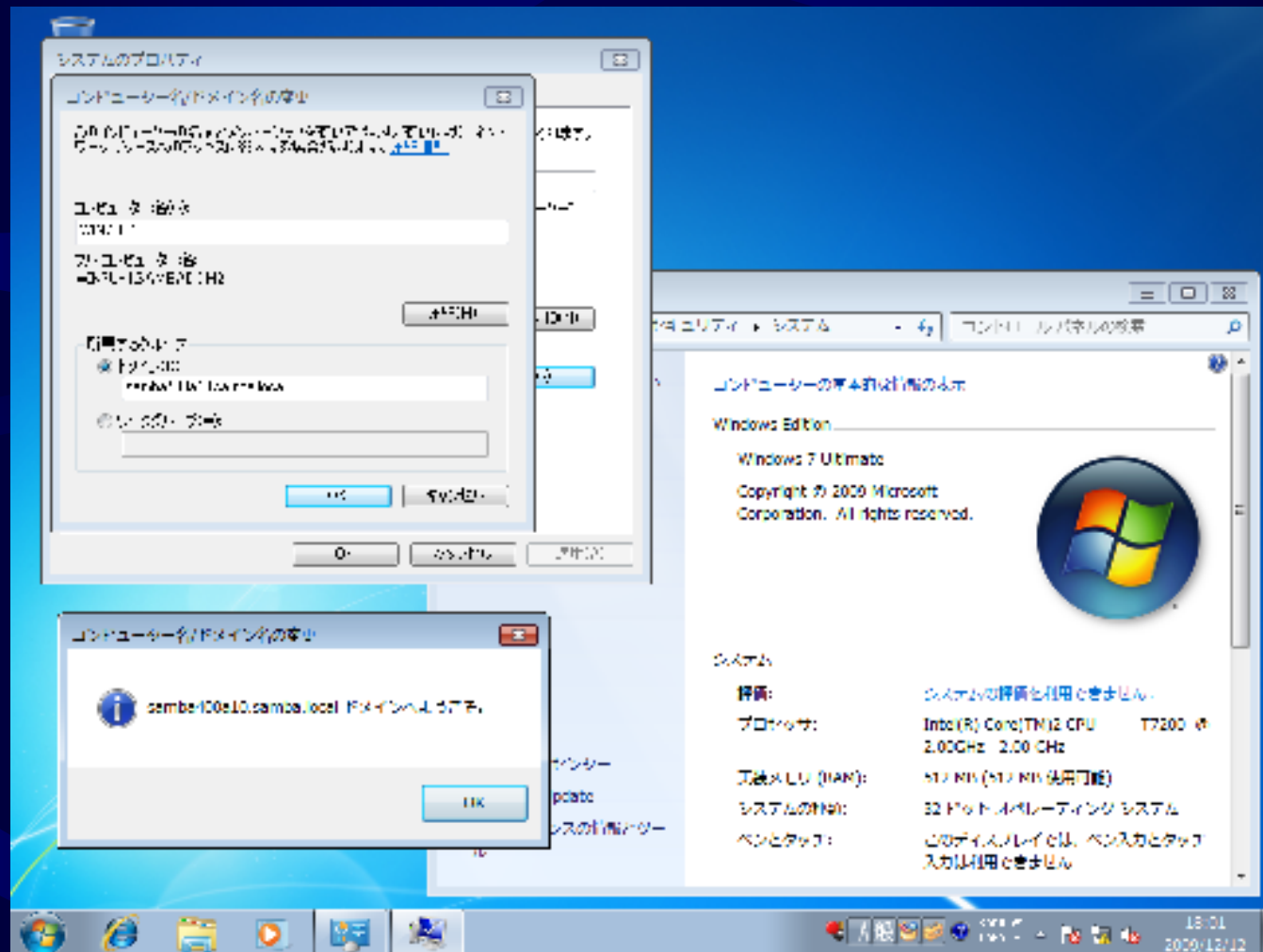
最新動向

Samba 4.0系列の開発状況

- Samba 4.0.0alpha11が2010年1月10日リリース
 - DCに限定すれば、なんとか使えるレベルになってきつつある
 - Samba 4.0.0alphaはDC専用として、ファイルサーバとしてはSamba 3.x系列を使用する形態
- Samba 4.0系列でサポート予定の主な新機能
 - Active Directoryのドメインコントローラ(DC)機能
 - Windowsクライアントの参加
 - グループポリシーのサポート
 - NTFS完全サポートやクラスタなど、もともとSamba 4.0系列の新機能として予定されていた機能は、Samba 3.x系列の新機能として逐次実装されている

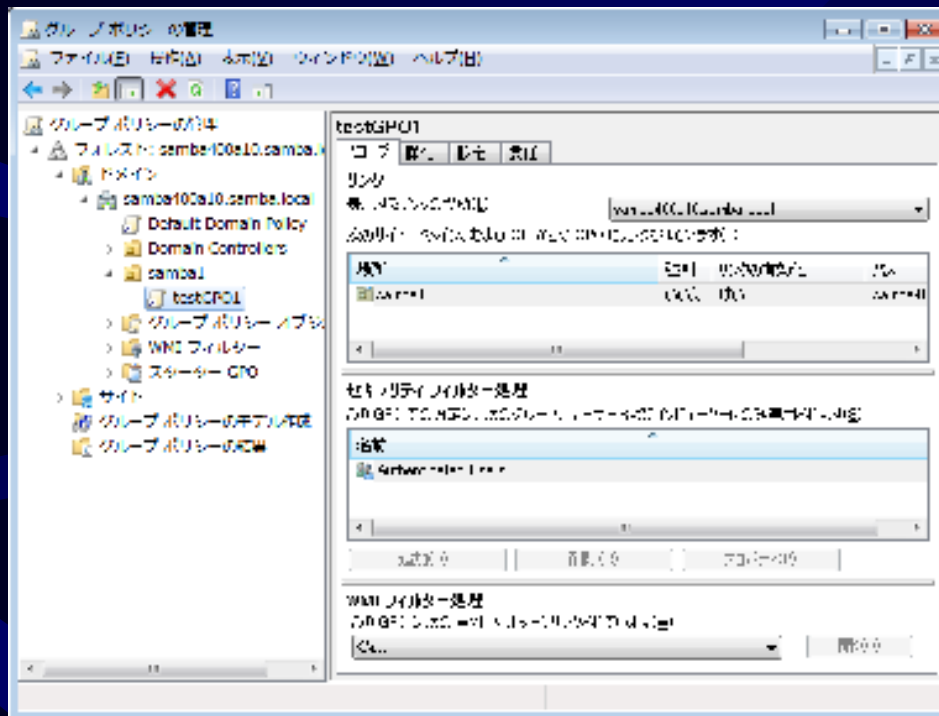
Samba 4.0系列の開発状況

- Windows 7からのドメイン参加も可能(4.0alpha10)

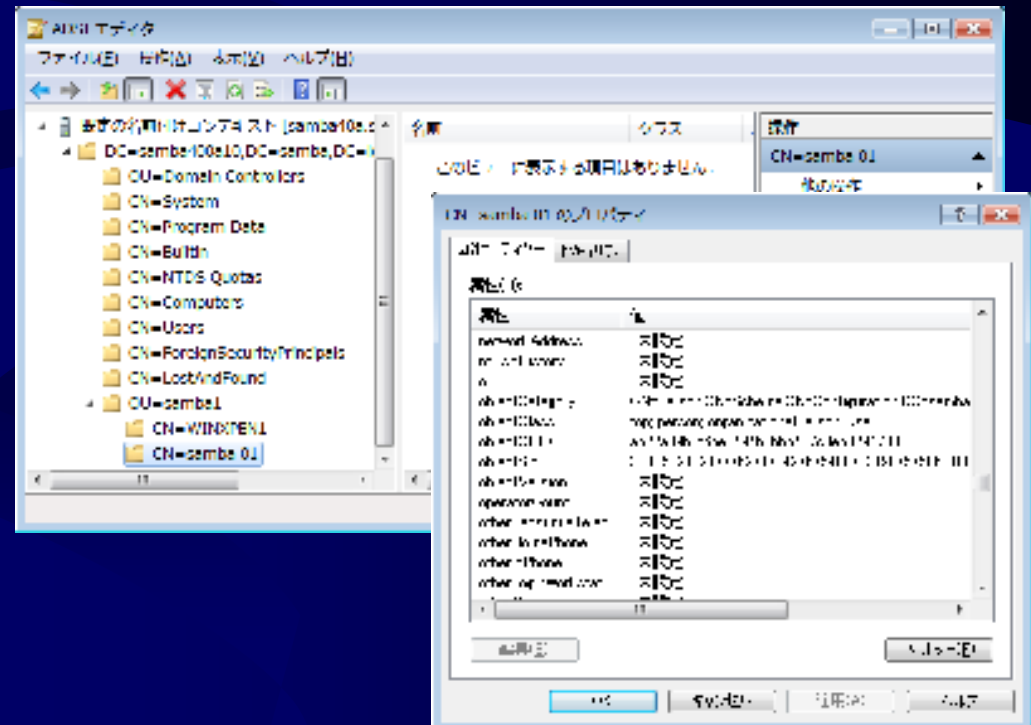


Samba 4.0系列の開発状況

- 管理ツールによる管理も可能(4.0alpha10)



グループポリシーの管理



ADSIエディタ

Samba 3.2系列の新機能

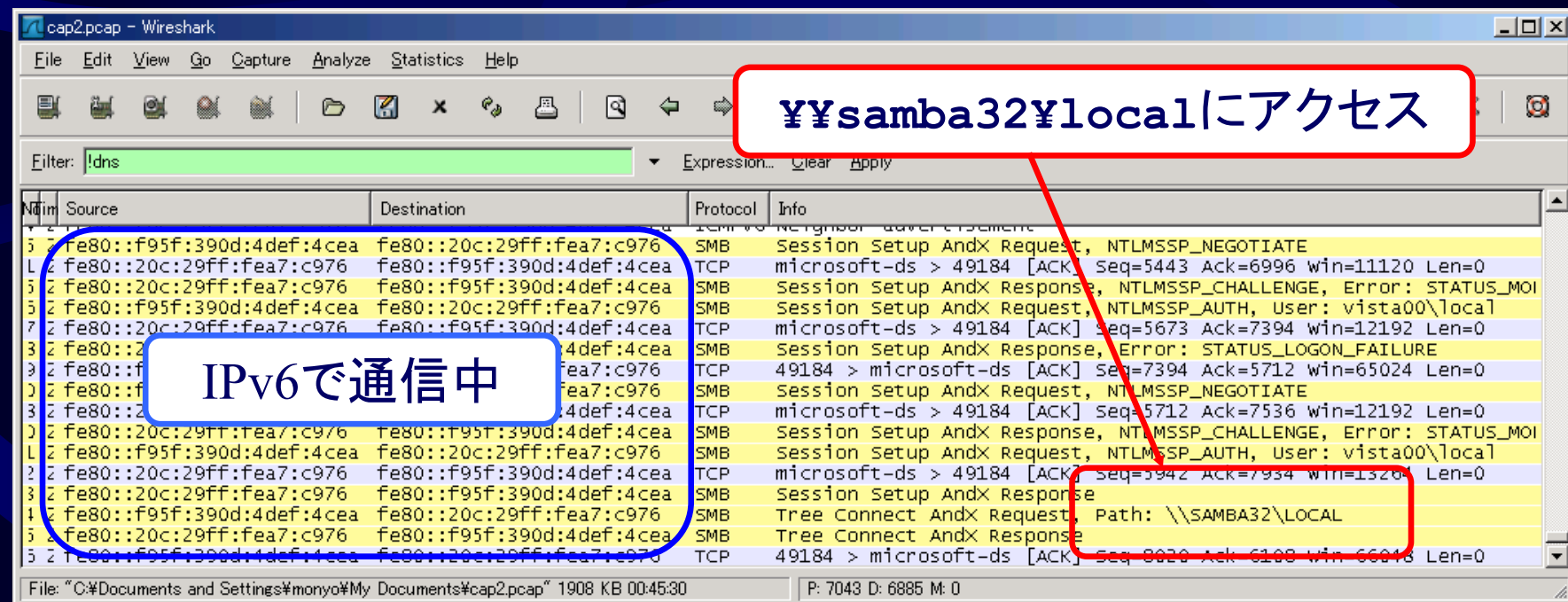
- **Active Directory関連の互換性強化**
 - Windows Server 2008ドメインへの参加に対応
 - Windows Server 2008ドメインとの信頼関係確立に対応
 - Windows Server 2003のフォレスト間信頼に対応
 - Winbind機構がUPN名ログオンに対応
 - LDAP署名に対応
- **セキュリティ強化**
 - Secure by Default
 - CIFS独自のネットワーク暗号化に対応 (Windowsは非対応)
- **IPv6対応**
 - Windows Vista/Server 2008からのIPv6アクセスに対応

Samba 3.2系列の新機能(2)

- クラスタ対応(実験的)
 - CTDBと連携したフェイルオーバークラスタ機能を実装
- NTFS対応の強化
 - 代替データストリームに対応
 - 長いファイルパス(内部的に1024バイト超)対応
 - Windowsの仕様上はパス名制限なし、ファイル名は最大255「文字」
- レジストリへの設定情報格納
 - ファイル編集ではなく、`net conf` コマンド、Windowsのレジストリツール(現在未対応)での設定変更が可能に
- 各種認証データベースの直接編集機能が追加
 - 新規に追加された`net sam`コマンドにより実現

IPv6対応 — 設定

- OSが対応していれば、自動的に有効
 - Windows Vistaからの通信例
 - ¥¥samba32¥¥local に接続したところ



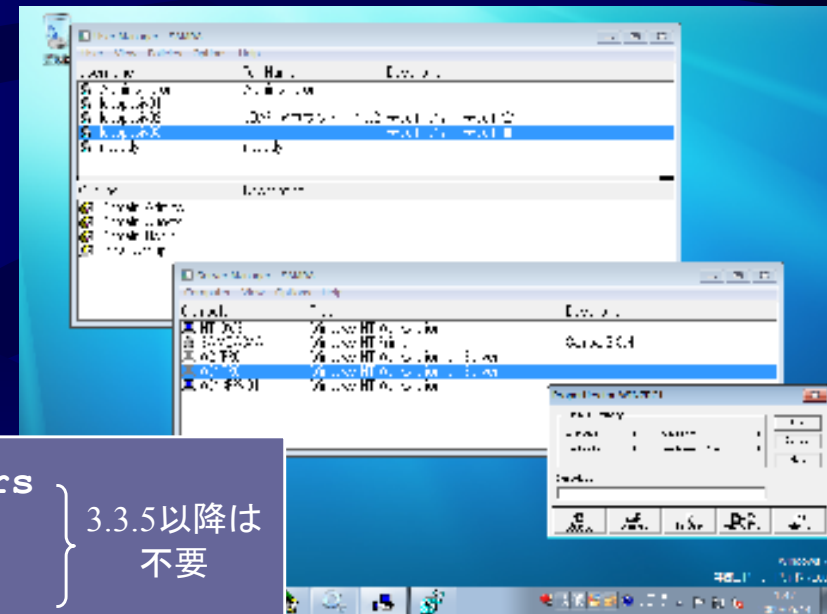
Samba 3.3/3.4/3.5系列の新機能

- NTFS完全互換のACLサポート(3.3.0)
 - vfs_acl_xttr/vfs_acl_tdbモジュールによる
- IdMap機構のポリシー変更(3.3.0)
 - Samba 3.0.25以降の複雑な設定方法以外に、簡単な設定方法もサポート
- Winbind機構によるホームディレクトリ自動作成サポート(3.3.0)
- Windows 7ドメイン参加対応(3.3.2)
- デフォルトのpasswdバックエンドがtdbsamに変更(3.4.0)
- SMB2の実験的サポート(3.5.0rc)
 - max protocol = smb2
- Windows完全互換の時刻精度のサポート(3.5.0rc)

→ 機能的にはSamba 3.2系列の延長線上にある機能
改修版

Windows 7対応

- ファイルサーバとしての使用は、問題なし
- 懸案だったSambaドメインへのログオンも実現
 - Samba 3.3.2以降
- Windows 7マシンのレジストリ変更が必要



```
HKLM\SYSTEM\CurrentControlSet\Services\Netlogon\Parameters
RequireStrongKey = 0
RequireSignOrSeal = 0
```

3.3.5以降は
不要

```
HKLM\SYSTEM\CurrentControlSet\Services\LanManWorkstation\Parameters
DWORD (32bit) 形式 DNSNameResolutionRequired = 0
DWORD (32bit) 形式 DomainCompatibilityMode = 1
```

Windows 7 版からSamba 3.3.4
サーバで構築したSambaドメイン
にアクセス

http://wiki.samba.gr.jp/mediawiki/index.php?title=Windows_7からSambaドメインにログオンできない

参考情報ほか

- 日本Sambaユーザ会
 - <http://wiki.samba.gr.jp/>
 - 日本語による技術情報など
- 書籍「Sambaのすべて」
 - 著者／高橋基信
 - 発行／翔泳社



ISBN4-7981-0854-5 ¥3,980円

- 現在ドキュメント翻訳中
 - 協力していただける方を随時募集中です
「Sambaドキュメント翻訳プロジェクト」