Linux 系パソコンの設定

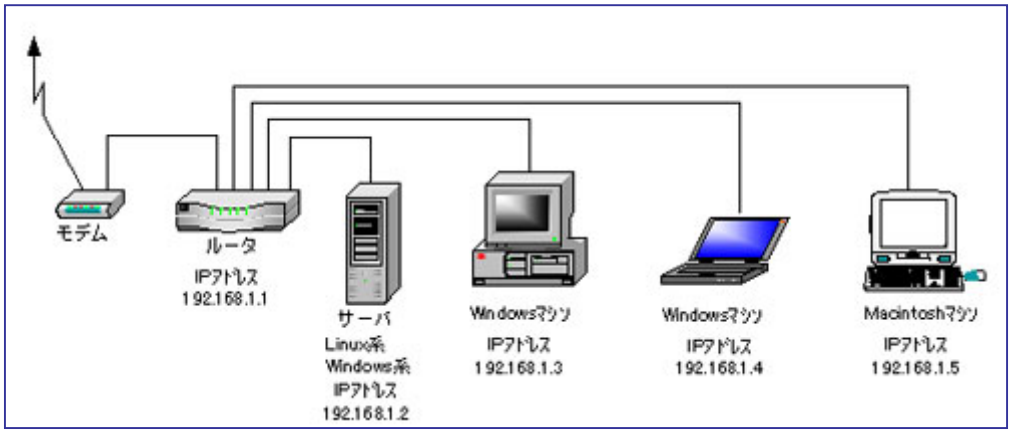
目　　次

H15/11/6：更新

１．インターネット接続

(1)常時接続(FTTH やADSL 又は CATV)であること。
(2)静的 IP マスカレード機能搭載、DHCP サーバ/クライアント機能を持ったルータを使用している。
(3)プロバイダからサーバの公開を認められていること。
(4)パソコンが 2 台以上あること。(サーバ機用とクライアント用構成)

ネットワーク環境事例



２．FFFTP をダウンロードします。

サイト内の FFFTP を選択。ダウンロードする。
最新版はこちら。ダウンロードしたファイルを実行してインストール。
Download ffftp-1.92.exe (620,256 バイト)
http://www2.biglobe.ne.jp/~sota/ffftp.html

２．Vine Linux 2.5 のインストール

はじめに

VineLinux をノートパソコンにインストールする方法を記載してある。ディスクトップ等にインストールするときは、読み替えてください。

(1)事前準備

インストールマシンの用意。

VineLinux のブート FDD をセットする。

VineLinux2.5 の CD-ROM をセットする。

（＊１）CD-ROM からインストールする場合は、パソコンの電源投入直後に［F12］を押し続ける。スクロールで CD-ROM アイコンを選択する。

（＊２）ブートディスク作成用に用意。ハードウェア情報を集めるインストールするためのハードウェア情報を集めましょう。古いハードウェアなら大抵はインストールできるはずです。だいたい 1 年前のハードウェアなら大丈夫でしょうか。最近は新しいハードでも標準的なドライバで動くものが多いので、上記のページで調べてみましょう。インストールディスクを用意するまず Windows を起動。そして Linux をインストールする為のディスクを作りましょう。

(注)使用しているマシンが CDROM ブートをサポートしているのであれば、このブートディスクは必要ありません。BIOS で CDROM ブートに設定し、Vine の CDROM を挿入してマシンを起動すればインストーラが立ち上がってきます。ここで作成するインストールディスクとは、Linux を CDROM からインストールする場合に使用するディスク。その他、ネットワーク経由、PCMCIA カード経由でインストールする場合は別のブートディスクが必要になります。

CDROM をドライブに挿入し、1.44M でフォーマットしたフロッピーディスクもドライブに挿入し、DOS プロンプトを開いて以下のディレクトリに移動します。(CDROM ドライブが e ドライブと仮定します)e:¥>cd dosutils

以下のコマンドを実行し、フロッピーディスクにデータを書き込みます。(フロッピードライブを a ドライブと仮定します)

e:¥>rawrite.exe -n -f e:¥images¥boot.img -d a:

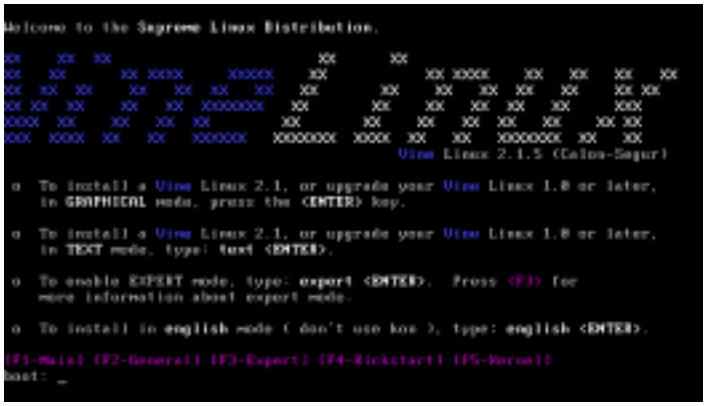
これで、フロッピーディスクに Linux インストーラを起動するためのデータが書き込まれました。

(2)インストール開始

a. 再起動すると、FDD からインストールが開始される。

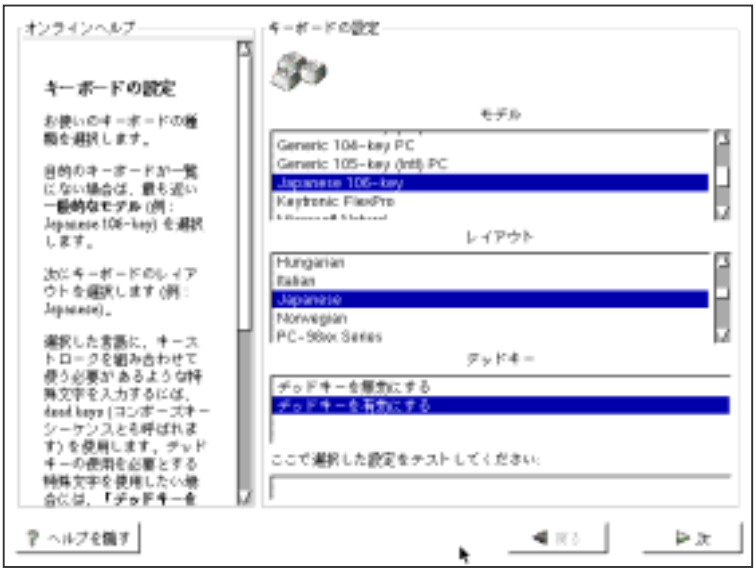
起動して、しばらくすると VineLinux のメッセージ表示後、「boot:」のメッセージが表示される。

そのまま、「Enter」キーにて実行する。



b. VineLinux 画面表示後、「Japnese」選択する。

c.キーボードの種類とレイアウト
　モデル：Japanese 106-key
　レイアウト：Japanes
　デッドキー：デッドキーを有効にする



d. VineLinux の世界へようこそ、「次へ」クリックする。

e.使用するマウスの設定
　・Gene 2 Button Mouse(PS/2)
　・3 ボタンマウスのエミュレーションを設定する。

f. System Installer 画面、「次へ」クリックする。

g. インストールオプション：ノートパソコン
ラップトップを選択、「次へ」クリックする。

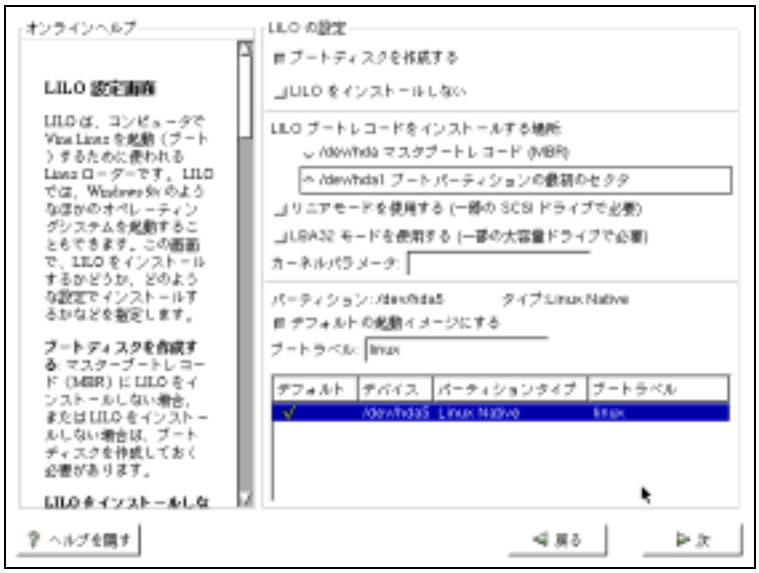
h. パーティションの設定
Disk Druid を使用して手動でパーティション設定

i. フォーマット
/dev/hda6 は、Linux スワップされているがフォーマットされていません！
「はい」を選択する。

□　をクリック。「編集」→ 「/boot」、フォーマット時のパーティションタイプ「ex2」
□　をクリック。「編集」→ 「/」、フォーマット時のパーティションタイプ「ex2」
「次へ」クリックすると、ファーマットしてよいか聞かれる。「はい」を選択。

/dev/hda1　14339MB　NTFS
/dev/hda2
□/dev/hda5　vfat　　39MB → ex2　　/boot
　/dev/hda6　swap　926MB → swap　SWAP
□/dev/hda7　vfat　2047MB → ex2　　/
　/dev/hda8　vfat　　47MB → vfat　　SYSSL

j. ブートディスクの作成：◎必ず作成しないと起動が出来なくなることがあります。
・ブートディスクを作成する。
・　ブートローダのインストール場所：「/hda5」を選択
・　「次へ」クリック

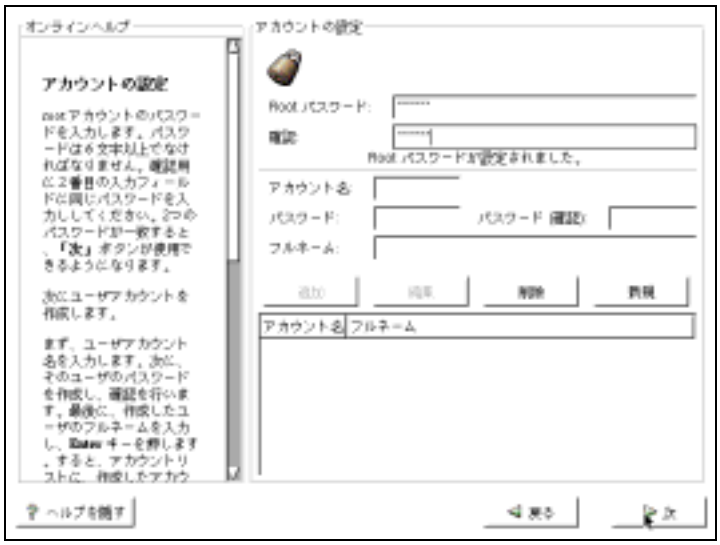


＊ブートディスクは、必ず作ること。起動失敗したときに有効となります。

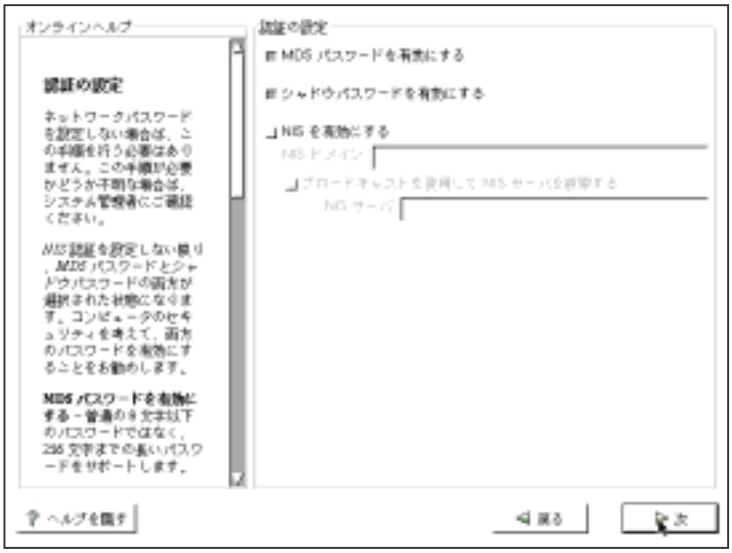
k タイムゾーンの設定：Asia/Tokyo

l root のパスワード設定

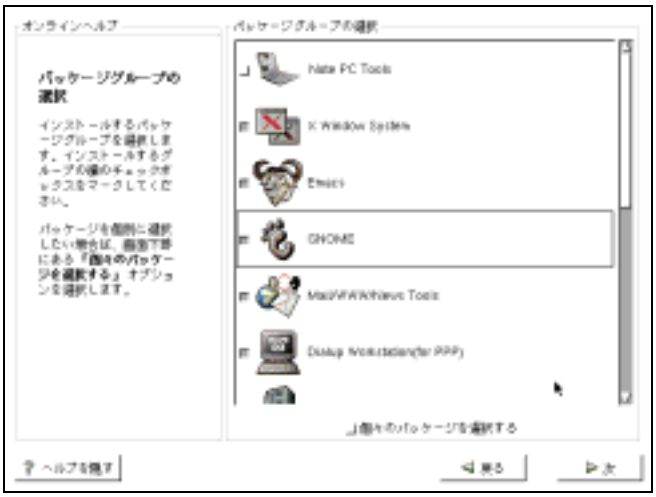
ユーザアカウントの設定もここで行う。



m 認証の設定:[MD5 パスワードを有効にする]と[シャドウパスワードを有効にする]は必ずチェックしましょう。[NIS を有効にする]は、ネットワーク上に NIS サーバーがあり、それを利用している場 合のみ設定して下さい。管理者に聞けば設定内容は分かるはずです。



l. パッケージグループの設定：パッケージを全て選択してインストールします。その際、使用しないデーモンなどが立ち上がるので、それらはインストール後に設定することにしています。

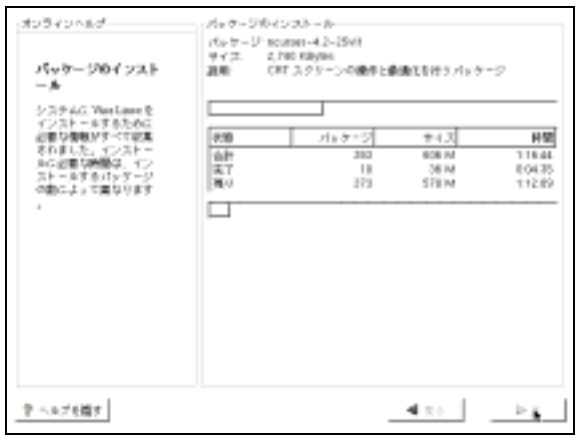


m. モニターの設定：自分の持っているモニタを捜してみましょう。残念ながら無かった場合は、ブラウン管であ れば「Generic Multisync(Hsync 31-64KHz)」を、TFT であれば「Generic LCD Panel 1024×768」 などをとりあえず選んでおきましょう。これはインストールした後で変更できます。

n. ビデオカードの設定：自分の持っているビデオカードを設定しましょう。もしなかった場合は、[X の設定を行わない] を選択し、インストール後に設定しましょう。

o. インストールの準備完了

p. パッケージのインストール開始



q. パッケージのインストールが終わった後、ブートディスクの作成画面になります。ここでは 必ず作成しておきましょう。

r. お疲れさまでした。これでインストールは完了です。ブートディスクをフロッピードライブ に挿入したまま、コンピュータを再起動しましょう。

VineLinux2.5 のインストールマニュアル参照

注意：サーバシステムを選択すると、既存のパーティションが全て Linux パーティションになります。もしも、変更されたら「System Selectore2 インストール」FDD の挿入して再起動する。

System Selectore2 インストールウィザード表示→実行する動作→「更新･修復(U)」選択→［次へ］→System Selectore 更新･修復→［次へ］→System Selectore 修復完了→［OK］→再起動すれば完了

(3)最新版にアップデート方法

Vine Linux の起動より、Mozill を選択し Mozill が起動。(起動しない時は、モデムまたはルーターの設定を見る)
インターネットが接続出来る事を確認します。Vine Linux の起動より[kterm.sh]を選択。システムを再設定。アップデート

#apt-get update
システムを再設定、apt-get update が終了。
次は、アップデート
#apt-get upgrade
システムを再設定、アップデート
apt-get getupgrade を続行しますかを[Y]Enter
apt-get getupgrade のダウンロードが始まりインストールも開始し終了します。
最後に、#プロンプトが表示されれば、無事終了。(作業は約 1 時間半前後掛かります)
これで、「Webmin」が起動できる環境が出来ました。

３．IP アドレス固定方法

kterm を起動 Window より、ifconfig と入力。
IP アドレスが確認出来ます。192.168.0.200 は固定されていない事が解ります。

Network Configuration を使用し設定。
自動設定メニューー⇒システム⇒Network Configuration と選択。

【IP アドレス固定方法】
〔名前の設定〕
ホスト名　　Linux　：コンピュータ名前を入力します。
ドメイン名　　linux.server.org　：ドメインネーム名を入力します。
保存を選択します。

［ホストの設定］編集 Window が出ましたら
IP：192.168.0.200　(固定したい IP アドレスを入力)
　　［ルータ設定時の IP アドレスを考慮して〕
名前：linux.server(コンピュータ名)
　　〔クライアントマシンと競合しない様に〕
ニックネーム：Linux(任意の名前)
了解を選択。

【IP アドレス固定方法】
ホストの編集は/etc/hosts からも出来る。
［インタフェースの設定］
eth0 を選択し編集を選択します。

◇IP アドレス：192.168.0.200　(固定 IP アドレス)を入力します。
◇ネットマスク：255.255.255.0 入力
◇ブートを ON にします。
◇プロトコルを dhcp から none に変更。
◇了解を選択します。

【IP アドレス固定方法】
ifconfig より確認します。IP アドレスが固定になっている事が確認出来る。
インターネット接続確認を行う事。

４．Apache の設定

Apache の設定は、 (Redhat 系：/etc/httpd/conf)という設定用ディレクトリの中の「httpd.conf」で行う。特別な使い方をしないなら､以下の行を確認するだけで正常に動作。

管理者のメールアドレス
ServerAdmin you@domain.co.jp
conf や log ディレクトリを格納するディレクトリ
ServerRoot /usr/local/apache
エラーログ・アクセスログの出力ファイル
ErrorLog logs/error_log
TransferLog logs/access_log
サーバのプロセス番号を記録するファイル
PidFile /var/run/httpd.pid
DNS に登録されたサーバ名
ServerName www.domain.co.jp
HTML ファイルを置くルートディレクトリ
DocumentRoot "/home/httpd/html"
Apache の設定用ディレクトリには httpd.conf 以外にも、他の設定ファイルがある。srm.conf、access.conf、これらは Apache の古いバージョン用ファイルで、最新の Apache には必要ない。magic と mime.type は、Apache を実行するために編集することはめったにない。あとは効果的に Apache を管理・運営する方法とチューニングアップ。

4.1 Apache の設定方法

Apache1.3 系の httpd.conf の設定方法は httpd の動作の全てに依存します。ここでは httpd.conf の記述方法。

・Directory
・Files
・AllowOverride
・Options
・DirectoryIndex
・AddModule
・VirtualHost の設定
・.htaccess
・WebDAV の導入
・mod_rewrite と mod_setenvif
・mod gzip の導入
・mod bandwidth の導入

(1)設定まえに
セキュリティー強化の為に apache を動作させる専用のユーザーを作る。デフォルトでは nobody ユーザーnobody グループで実行しますが、その場合、apache にセキュリティーホールがあると nobody ユーザーを使って他のサービスも害されてしまう恐れがあります。ですから Apache ユーザーで運用するべきです。もちろん apache ユーザーはログインシェルを持たせるべきではありません。しかし Apache ユーザーが FTP を使用してページを upload する場合はログインシェルが必要。

useradd -s /bin/false -d /home/html/apache apache
ホームディレクトリは/home/html/apache などにしておくと良い。また apache グループも作成した方が良い。次に apache ログの記録に影響がない様、apache インストールディレクトリの権限も root から apache に変更。

chown -R apache /home/html /apache

4.2 apache(httpd)を apache ユーザーで起動させるために httpd.conf を変更。

LockFile /usr/apache/run/httpd.lock
PidFile /usr/apache/run/httpd.pid
ScoreBoardFile /usr/apache/run/httpd.scoreboard

User apache
Group apache

最後に ServerName の記入は必要事項です！
ServerAdmin admin@syns.net
ServerName www.syns.net

記入にすれば終了。その後は apachectl を restart させれば全ての nobody ユーザーで稼動していた httpd をシャトダウンして apache ユーザーで httpd が稼動。

<Directory directory> ... </Directory>
<Directory>と</Directory>はそのディレクトリで指定されたディレクトリとサブディレクトリに対してのみ適用される命令のグループ化ために使用される。そのディレクトリ内では許可された命令のみが使用可能となる。Directory は相対パスか、またはワイルドカードのどちらかです。ワイルドカードの中で「?」はある一文字に一致し、「*」はある連続した文字列に一致します。例えば：

<Directory /usr/local/httpd/htdocs>
Options Indexes FollowSymLinks
</Directory>
という場合、/usr/local/httpd/htdocs 内もしくはそのサブディレクトリ内で命令 Options ` Indexes" "FollowSymLinks"が有効になるという事。ここで注意しなくてはいけないことは、""で指定したディレクトリは絶対パスを意味。""を省略した場合、これは Document Root の相対パスになる。

<Directory ~ "^/www/.*/[0-9]{3}">
上記は Document Root 配下にある/www/にある３つの数字から成るディレクトリ名に一致。

<Files filename> ... </Files>
<Files>命令は、ファイルネームによるアクセスコントロールを設定。<Directory>命令に匹敵します。そして末尾の</Files>命令と対応。与えられたファイルネームに対して適用される命令は内部でリストになる。</Files>セクションは<Directory>セクションと.htaccess ファイルが読み込まれた後、<Location>セクションの前にコンフィグレーションファイルに存在する命令を処理。
filename はあるファイルネームまたは、ワイルドカード文字を含みます。「？」はある一文字に一致し、「＊」はある連続した文字列に一致します。通 常の表記よりも拡張された正規表現もまた「~」文字を付加して使用する事が出来ます。例えば：
<Files ~ "¥.(gif|jpe?g|png)$">
最も一般的なインターネットのグラフィックのフォーマットに一致する。.htaccess ファイルの内部で使われる<Directory>セクションは、<Location>セクションと<Files>セクションとは違うことに注意してください。

AllowOverride
サーバが.htaccess ファイル（AccessFileName によって指定されたファイル）を見つけると、サーバはより早くアクセス情報を上書き出来るファイルに設定された命令を知ろうとします。公開上ディレクトリ内の.htaccess ファイルを有効にさせるための記述。Override は None に設定することが出来。それはサーバがそのファイル（.htaccess）を実行しないように設定する場合。サーバが全ての命令を許す場合は Override を All にする。その他 Override には次の様な設定がある。

AuthConfig
認証設定の使用を許可。(AuthDBMGroupFile, AuthDBMUserFile, AuthGroupFile, AuthName,

AuthType, AuthUserFile and require).

FileInfo

ドキュメントタイプをコントロールする命令の使用を許可(AddEncoding, AddLanguage, AddType,

DefaultType and LanguagePriority).

Indexes
ディレクトリインデクスをコントロールする命令の使用を許可(AddDescription, AddIcon,

AddIconByEncoding, AddIconByType, DefaultIcon, DirectoryIndex, FancyIndexing, HeaderName,

IndexIgnore, IndexOptions and ReadmeName).

Limit
ホストへのアクセスをコントロールする命令の使用を許可(allow, deny and order)。次の例は iria1.07s9a のブラウザ変数をもったクライアントからのアクセスを制限しています。

BrowserMatch "Iria/1¥.07s9a" bbroken (¥はバックスラッシュ)
<Limit GET POST>
order allow,deny
deny from env=bbroken
allow from all
</Limit>
下記は 192.168.0.0/24 からのアクセスを制限。
deny from 192.168.0.
allow from all

Options
特定のディレクトリの設定をコントロールする命令の使用を許可(Options and XBitHack).

<Directory /usr/local/httpd/htdocs>
AllowOverride None
</Directory>
全ての Override を否定し、.htaccess を実行できない場合、None を All に変えると全て許可。

Options [+|-]option [+|-]option ..
Options 命令はサーバ制御を特定のディレクトリ内でのみ有効にする命令。option が None に設定されている場合は特別な制御は使用不可であるかあるいは、次のような場合がある。

All
MultiViews を除いた全てのものが利用可能。

ExecCGI
CGI の実行が認められます。

FollowSymLinks
サーバはこのディレクトリにシンボリックリンクを許可。

Includes
サーバ側インクルード機能を許可。

IncludesNOEXEC
サーバ側インクルード機能を許可しますが、#exe コマンドと CGI スクリプトの#include は含みません。

Indexes
もしディレクトリにマップしている URL がリクエストされ、そのディレクトリに DirectoryIndex（例：index.html）がない場合には、サーバがそのディレクトリ以下のファイルを表示することを許可。

MultiViews
Content negotiated MultiViews を許可。

SymLinksIfOwnerMatch
　サーバは、対象とするファイルとディレクトリがリンクしているとき同様のユーザ ID に従ってシンボリックリンクを許可。通常、もし複数の Option があるディレクトリに対して適用された時、大抵特定の一つが実行されます。;option は２つ以上を実行することができません。しかし、もし Options 命令の all に + か - の記号が all の前に書かれている場合は、option はマージされます。+ が書かれた option は一般に、有効に options に加えられ、- が書かれた option は効果のある option から除外されます。例えば, +と-の記号なしでは:
<Directory /web/docs>
Options Indexes FollowSymLinks
</Directory>
<Directory /web/docs/spec>
Options Includes
</Directory>
　上記は Includes だけが/web/docs/spec ディレクトリに設定されます。Options 命令が+と-の記号を使っていれば、
<Directory /web/docs>
Options Indexes FollowSymLinks
</Directory>
<Directory /web/docs/spec>
Options +Includes -Indexes
</Directory>
　これによって FollowSymLinks と Includes が/web/docs/spec ディレクトリに設定。

DirectoryIndex File File ...
　URL でファイル名を省略した場合に表示するファイル。"/index.html"を"/"に省略する場合は File に index.html を指定。
DirectoryIndex index.html index.shtml index.htm
　上記は index.html index.shtml index.htm 全てを"/"に省略するための記述。ディレクトリにこれら 3 種が存在する場合は前から優先される。

AddModule module module ...
　Compatibility: AddModule は Apache1.2 以降のみ使用可能。サーバは普段使用していないコンパイルされたモジュールを使用することが可能。この命令はそれらのモジュールを使用可能にするために使用される。サーバはあらかじめロードされた実行可能なモジュールのリストを実行。このリストは ClearModuleList 命令によってクリアー。

VirtualHost の設定：バーチャルホスト
　１つのサーバ上に、２社以上のサイトを設置したい場合。この２社以上のサイトを１つのサーバ上に構築する設定方法が Apache のバーチャルホスト。必要なモジュールは mod_vhost_alias.so。

LoadModule vhost_alias_module libexec/mod_vhost_alias.so
AddModule mod_vhost_alias.c

　以下を追記してください。
NameVirtualHost 192.168.0.1
<VirtualHost 192.168.0.5>
DocumentRoot /var/apache/htdocs/www
ServerName www.syns.net
</VirtualHost>
<VirtualHost 192.168.0.5>
DocumentRoot /var/apache/htdocs/host-a/
ServerName host.syns.net
</VirtualHost>
　このようにして 1 つの IP で複数のドメインを張り付けてドキュメントルートをわける場合は名前ベース (non-ip-VirtualHost) を使う。バーチャルホストの IP アドレスは外からみたアドレスを記入するのではありません。WWW 自身が使っている NIC のアドレス。これにより NameVirtualHost ディレクティブで指定した IP アドレスに対するリクエストは名前ベースで割り振られるようになる。各割り振りは VirtualHost ディレクティブで記述。上記の場合は、192.168.0.5 に対してのアクセスを ServerName で指定したホスト名ごとに割り振られるようになります。
　もう一方で IP ベースのバーチャルホストがある。こちらの方が歴史は古く一般てきですが、Web サーバーに複数の IP を持つ例があまりありませんので最近では non-ip 方法が普及しています。

<VirtualHost 172.16.0.6>
DocumentRoot /var/apache/htdocs/www
ServerName www.syns.net
</VirtualHost>
<VirtualHost 172.16.0.5>
DocumentRoot /var/apache/htdocs/host-a/
ServerName host.syns.net
</VirtualHost>

今度は、VirtualHost ディレクティブで IP アドレスを直接指定することによって IP アドレスごとに割り振られるようにしています。名前ベースと同じ様に IP アドレスは WWW が使用している NIC の IP アドレスを記入。

.htaccess

設定したいディレクトリに.htaccess ファイルを置くと httpd.conf と同じ記述でそのディレクトリ以下を制御できる。一般 にローカル個人ユーザーに web サービスを提供する場合 httpd.conf ファイルを触れさせる事はできません。よってこの機能を使いユーザーが各好みの設定に変更して利用させる方法が良いです。 そのためには httpd.conf より htaccess ファイル優先させておく必要があります。

AccessFileName .htaccess

.htaccess ファイルの利用例を紹介しておきます。

1) CGI・SSI の実行権設定：サーバがデフォルトで CGI・SSI 実行 OK になっていない場合は、.htaccess で設定する必要があります。

Options Includes execCGI
AddType text/x-server-parsed-html .shtml
AddType application/x-httpd-cgi .cgi .pl

4.3 PHP の実行拡張子設定

DirectoryIndex index.phtml index.html
AddType application/x-httpd-php3 .php3

4.4 ページをリダイレクトする

アクセスされたら、指定したページに飛ばす方法。

HTML の<meta http-equiv="〜">と違って、飛んだことに気づかれにくいです。

Redirect / http://hogehoge.com/
Redirect /fuga/ http://hogehoge.com/hoge/

4.5 認証不許可時に表示する画面を指定

auth 認証(上で指定したもの)で認証に失敗した場合は、指定した URL へ飛ばす。(指定しない場合は、通常の"401 Authorization Required"エラーが表示)

ErrorDocument 401 /auth_err.html
ErrorDocument 403 "403 アクセス権がありません。"

4.6 アクセス制限

認証許可する ID とパスワードは、あらかじめ htpasswd で作成する。

AuthUserFile /home/mydir/public_html/（htpasswd で作成したパスワードファイル名）
AuthGroupFile /dev/null
AuthName "Authorization Users"
AuthType Basic
<Limit POST GET PUT> require valid-user </Limit>
<Files .htaccess> order deny,allow
deny from all
</File>

WebDAV の導入
　必要モジュール:dav_mod
(例)mod_dav-1.0.2-1.3.6.tar.gz
$ ./configure --with-apxs=/usr/apache/bin/apxs
1.0.3 の webDAV の場合は apxs を使わずとも簡単にダイナミックモジュールとして導入できました。(その時の環境は NetBSD です)
$ make
$ su
# make install

　インストールが済めば httpd.conf を編集して以下のモジュール指定文を追記してください。
LoadModule dav_module /usr/apache/libexec/libdav.so
AddModule mod_dav.c

　そして mod_dav の設定です。
DAVLockDB /usr/apache/run/DAVLock
<Location /DAV>
DAV On
</Location>
　以上でファイル共有はできるようになっている。上手く機能しない場合は DAV と On の間だをスペースではなくダブを入れてみて下さい。また<Location>ディレクティブではなく<Direvtry>ディレクティブでも問題ありません。そしてセキュリティーを重視してグローバルで利用する場合は SSL と共有制限をかけなければいけません。その場合、SSL は後で説明しますが、共有制限は以下の様に<Location>ディレクティブの中に<LimitExcept GET HEAD OPTIONS></LimitExcept>を挿入し制限規定を追記してください。
　すべてのアクセスで認証をかける場合
<Limit GET PUT POST DELETE PROPFIND PROPPATCH MKCOL COPY MOVE LOCK UNLOCK>
</Limit>
　参照では認証をかけない場合
<LimitExcept GET HEAD OPTIONS>
AuthType Basic
AuthName "WebDAV Secutity"
AuthUserFile /etc/apache/davuser
Require valid-user
</LimitExcept>
　ちなみに 任意の Apache 公開ファイルに.htaccess ファイルを使って DAV 共有することも確認しました。

mod_rewrite と mod_setenvif
　一般にページの制限や接続元による表示内容の変更は CGI や JAVA などを用いるでしょう。しかし Apache でも細かなアクセス制限が可能です。mod_rewrite と mod_setenvif のモジュールの代表的な利用方法の一つに直接リンクを禁止する方法があります。
mod_rewrite は Solaris の標準の apache に同梱されています。特定の[許可 RefereURL]からのアクセスのみ許可します。サイト URL と許可 RefererURL を一致させることで直リンクを禁止することができます。

RewriteEngine on
RewriteCond %{HTTP_REFERER} ^$
RewriteCond %{HTTP_REFERER} !^[許可 RefererURL1].*$ [NC]
RewriteCond %{HTTP_REFERER} !^[許可 RefererURL2].*$ [NC]
RewriteRule ^(.*)$ - [F]
　RewriteRule .*¥.gz$ - [F] とすると.gz ファイルのみ直リンクを禁止します。RewriteRule ^/$

/www/ [R] とすると URL"/"を/www/にリダイレクトします。またブラウザの環境変数 HTTP_REFERER の他に HTTP_USER_AGENT や REQUEST_FILENAME,HTTP_HOST も使用できることを確認しています。
　mod_setenvif は Apache1.3.12 以前のバージョンには確実には同梱されていません。mod_setenvif を使用する場合は Apache のバージョンアップをしてください。特定の[許可 Referer URL]からのリンクのみ許可 します。

SetEnvIf REFERER "[許可 Referer URL]" Lilith
Order deny,allow

deny from all
allow from env=Lilith
　特定の[許可 Referer URL]からのリンクのみ不許可にしています。
SetEnvIf REFERER "[許可 Referer URL]" Lilith
Order Allow,Deny
allow from all
deny from env=Lilith
　これらの記述は、.htaccess にそのまま記述してもかまいません。一方、httpd.conf に記述する時は必ずディレクトリを指定しましょう。 サイト全てが直リン禁止になればブックマークからの訪問者が接続できなくなりす。
<Directory /usr/local/httpd/htdocs/direct_limit>
Options Indexes FollowSymLinks
RewriteEngine on
RewriteCond %{HTTP_REFERER} ^$
RewriteCond %{HTTP_REFERER} !^[許可 RefererURL].*$ [NC]
RewriteRule ^(.*)$ - [F]
</Directory>
　詳しくはこちらを参照して下さい。
mod_setenvif の本家サイト、mod_rewite の本家サイト

mod gzip の導入
　必要なものは mod_gzip.c ソースのみです。導入はメインページにあるインストールログの「apache (patch & update)」を参考に次のコマンドラインを実行してください。
/usr/apache/bin/apxs -i -a -c mod_gzip.c
　Solaris 環境では/usr/ucb/cc が Sun コンパイラであるため上手くコンパイルできない場合があります。その時は gcc を使って下さい。
mv /usr/ucb/cc /usr/ucb/cc_old
ln -s /usr/local/bin/gcc /usr/ucb/cc
/usr/apache/bin/apxs -i -a -c mod_gzip.c
　必要であれば元に戻して下さい。
rm /usr/ucb/cc
mv /usr/ucb/cc_old /usr/ucb/cc
　LogFormat の最後に以下を追加します。
　LogFormat "%h %l %u %t ¥"%r¥" %>s %b mod_gzip: %{mod_gzip_compression_ratio}npct."

common_with_mod_gzip
(¥はバックスラッシュです)
　次の様に CustomLog と IfModule を修正してください。
CustomLog /var/apache/logs/access_log common_with_mod_gzip
<IfModule mod_gzip.c>
mod_gzip_on Yes
mod_gzip_minimum_file_size 300
mod_gzip_maximum_file_size 0
mod_gzip_maximum_inmem_size 100000
mod_gzip_keep_workfiles No
mod_gzip_temp_dir /tmp
mod_gzip_item_include file ¥.html$
mod_gzip_item_include file ¥.jsp$
mod_gzip_item_include file ¥.php$
mod_gzip_item_include file ¥.cgi$
mod_gzip_item_include file ¥.txt$
mod_gzip_item_include file ¥.shtml$
mod_gzip_item_include mime ^text/.*
mod_gzip_item_include mime ^application/x-httpd-php
mod_gzip_item_include mime ^httpd/unix-directory$
mod_gzip_item_include handler ^perl-script$
mod_gzip_item_include handler ^server-status$
mod_gzip_item_include handler ^server-info$
mod_gzip_item_exclude file ¥.css$

mod_gzip_item_exclude file ¥.js$
mod_gzip_item_exclude mime ^image/.*
</IfModule>
(¥はバックスラッシュです)
　一度 access_log を確認して下さい。今までのログの末尾に mod_gzip: 73pct.
　などと記されているとおもいます。これは 73%の圧縮を意味します。 尚、gif ファイル、クライアント側のブラウザが mod gzip に対応していない場合、は圧縮されません。

mod bandwidth の導入
　http パケットの伝送速度を制限するモジュールです。 入手先はこちら。
/usr/apache/bin/apxs -c mod_bandwidth.c
/usr/apache/bin/apxs -i -a mod_bandwidth.so
　かえって apx でインストールしない方がよいかもしれません。(もちろんコンパイルは apx を使用するべきでしょうが) 設定は httpd.conf にモジュールの設定を追記すればよいのですが、その前に mod_bandwidth の作業用フォルダを作成して下さい。
mkdir /usr/apache/bindwidth
mkdir mkdir /usr/apache/bindwidth/link
mkdir mkdir /usr/apache/bindwidth/master
chown -R apache:apache mkdir /usr/apache/bindwidth

httpd.conf のモジュール設定例
LoadModule bandwidth_module libexec/mod_bandwidth.so
AddModule mod_bandwidth.c

<IfModule mod_bandwidth.c>
BandWidthModule On
BandWidthDataDir /usr/apache/bindwidth
BandWidthPulse 1000000
</IfModule>
<Directory "/var/apahce/htdocs/bandlimit_dir">
BandWidth 192.168.0.0/24 0
BandWidth all 8388608
</Directory>

<Directory "/var/apahce/htdocs/file">
LargeFileLimit 1000 8388608
MinBandWidth all 10240
</Directory>
　bandlimit_dir に関しては内部 192.168.0.0/24 のネットワークに対して無制限、それ以外は 8Mbyte/sec の帯域制限をかけています。file に関しては、1000KByte の以上の容量 に対して帯域制限 8Mbyte/sec の制限をかけ、それより小さいファイルに対しては帯域を解放しています。
BandWidthPulse
　接続待機秒

BandWidth
　帯域制限を、ドメイン別、IP アドレス別に設定します。 IP アドレスは「ネットワーク/マスク」の書式で、rate 単位 はバイトです。0 の場合には無制限になります。

MinBandWidth
　下限値を設定します。 rate に 0 を指定した場合は、既定値の 256 バイト／秒が下限値になります。

LargeFileLimit
　指定した filesize KBytes 以上のファイルに対して帯域制限を設定します。

4.7 アクセスログの設定

(1) アクセスログの分割

Perl などで、アクセス数の統計を出す際などは、なるべく簡単で､小さなログファイルを使う。アクセスログは、リクエスト側のデータを保存。デフォルトの設定で要求先 URL と閲覧ブラウザ名も保存。要求先 URL は他サイトからのアクセスを調べる際、閲覧ブラウザ名は Netscape や Explorer の使用率などを調べる際に効果的。これらはアクセス数とは別に処理するほうがシンプルな方法。

アクセスログをひとつのファイルに出力するのではなく、たとえば、アクセス数用の「access_log」ファイル、要求先 URL 用の「referer_log」ファイル、閲覧したブラウザ名用の「agent_log」ファイルに分割。但し、場合によっては「agent_log」などは省いてもいい。

手順としては、アクセスを記録するファイルの指定コマンド TransferLog をコメントアウト。LogFormat を上記 3 つのファイル用に設定。あとは CustomLog にファイル名を指定する。その際に､LogFormat で設定したコマンドを指定することを忘れないように！

```
# TransferLog logs/access_log
#  ログファイルのフォーマットを設定
# LogFormat <format_string> <command>
LogFormat "%h %l %u %t ¥"%r¥" %>s %b" common
LogFormat "%{Referer}i -> %U" referer

# agent_log がいらない場合などはコメントアウト
# LogFormat "%{User-agent}i" agent
CustomLog access_log common
CustomLog referer_log referer

# agent コマンドをコメントアウトしていたら、忘れずにこちらもコメントアウトすること
# CustiomLog agent_log agent
```

(2)GIF/JPG ファイルを記録しない

GIF ファイルや JPG ファイルの呼び出し記録をログに記録するのは、効果的でない。通常は HTML ファイルのアクセス数を記録するだけで十分。ここでは GIF/JPG ファイルの呼び出し記録をログに書き込まないように設定。

```
#  環境変数に GIF/JPG ファイルを環境変数「object-is-image」に記録
SetEnvIf Request_URI "¥.(gif)|(jpg)$" object-is-image

#  環境変数 object-is-image に記録されたファイルを記録しないように指定
# CustomLog <directory> <command> <ENV>
CustomLog access_log common env=!object-is-image
CustomLog referer_log referer env=!object-is-image
```

(3)CGI を実行可能に設定

AddHandler の設定と、<Direcotry>の Options に ExecCGI を追加する。これで十分 CGI が使える。

<Directory>タグの使い方

<Directory ディレクトリパス>～</Directory>タグの説明：

<Directory>タグの中に、指定したディレクトリパスのアクセス権を設定する。主なオプションは以下のとおり。

Options None/All

指定されたディレクトリ内でアクセスに関する機能を禁止/許可

Options Includes

SSI を許可

Options ExecCGI

スクリプトの実行を許可

```
ServerName www.domain.co.jp
ScriptAlias /cgi-bin/ "/home/httpd/html/cgi-bin"
AccessConfig /dev/null
ResourceConfig /dev/null
```

```
AddHandler cgi-script .cgi

# CGI を実行させたいディレクトリ領域に、オプション「ExecCGI」を追加
<Directory "/home/httpd/html">
    Options Indexes FollowSymLinks ExecCGI
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>
```

(4)セキュリティの向上

FlollowSymLinks/FlollowSymLinksIfOwnerMatch

FollowSymLinks を無効にし、シンボリックリンクをたどれなくする。そのかわり、SymLinksIfOwnerMatch を設定し、ファイルまたはディレクトリの所有者がシンボリックリンクと同一の場合だけ、リンクをたどれるようにする。

```
<Directory "/home/httpd/html">
#     Options Indexes FollowSymLinks ExecCGI
    Options Indexes -FollowSymLinks
    +SymLinksIfOwnerMatch ExecCGI
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>
```

(5)速度の向上

FlollowSymLinks/FlollowSymLinksIfOwnerMatch

セキュリティ向上させるため、FlollowSymLinks のかわりに FlollowSymLinksIfOwnerMatch を使う。これは、シンボリックリンクをチェックするために余分なシステム・コールが発生し、速度性が損なわれる。速度を重視するか、セキュリティを重視するか、それが問題？必要な場所だけセキュリティチェックを行うようにする。

```
<Directory />
    Options FollowSymLinks
</Directory>
<Directory /home/httpd/html>
    Options -FollowSymLinks +SymLinksIfOwnerMatch
</Directory>
```

こうすれば、少なくとも DocumentRoot パスの余分なチェックが要らなくなる。ドキュメント・ルート以外に Alias や RewriteRule パスがある場合には、同様のセクションを追加する必要がある。

AllowOverride

URL 空間で overrides を認める場合(通常は「.htaccess」)、Apache は各ファイルネーム要素ごとに.htaccess を開こうとする。パフォーマンスの低下は避けられない！解決策は、ルートパスで AllowOverride None を使う。

```
<Directory />
    AllowOverride None
</Directory>
```

(6)ネゴシエーション

最高の性能を絞り出したい場合には、コンテンツ・ネゴシエーションを極力避けたい！実践的には、ネゴシエーション機能には性能の低下を補ってあまりあるものがあるから、これは有効にしておく事。「DirectoryIndex index」の様なワイルドカードは使わずに、次のようにファイル名を明確にしてリストする。

```
DirectoryIndex index.html index.shtml
```

一番頻繁に使うオプションをリストの先頭へ！

(7)プロセス

MaxRequestsPerChild 命令は個々の子サーバプロセスが取り扱うリクエストの範囲を指定。MaxRequestPerChild

が設定されると、その設定を超えた後の子プロセスは停止。MaxRequestsPerChild が 0 だと、そのプロセスは停止しない。この初期設定値は 30 になっているけど、子プロセスに肥大したメモリ・イメージを持たせるようなモジュールを使っているサーバでなければ、この値を 10,000 まで上げる。決して、KeepAliveTimeout を 60 秒以上にしない事。

設定例

```
# httpd.conf
#  サーバの基本的な動作の設定
# standalone(サーバはメモリに常駐)/Inetd(アクセスごとにサーバを起動)
ServerType standalone

# Apache が参照する各種ファイルの起点となるディレクトリパスを設定
# httpd.conf 以外の設定ファイル、およびログファイルが ServerRoot を参照
ServerRoot "/usr/local/apache"

#  サーバのプロセス番号を記録するファイル名
PidFile /usr/local/apache/logs/httpd.pid

#  ロックファイルのファイル名
#  ログディレクトリが NFS の場合、ログをローカルに保存するよう指定
#  上記以外はデフォルトのままで問題ない。
#LockFile /var/run/httpd.lock

#  タイムアウトを秒単位で設定
Timeout 300

#  ポート番号を設定
Port 80

#  ユーザ  webuser、グループ  webgroup で apache を立ち上げます。
# webuser は、web 専用のユーザとして作成してください。
User       webuser
Group      webgroup

#  管理者のメールアドレスを指定
ServerAdmin info@domain.co.jp

# DNS に登録されたサーバ名を設定します。
ServerName www.domain.co.jp

# HTML ファイルを置くルートディレクトリを指定
DocumentRoot "/home/httpd/html"

<Directory />
      Options FollowSymLinks
      AllowOverride None
</Directory>

<Directory "/home/httpd/html">
      Options Indexes -FollowSymLinks ExecCGI
      AllowOverride None
      Order allow,deny
      Allow from all
</Directory>

#  ユーザのホームディレクトリを指定
UserDir public_html

<Directory "/home/*/public_html">
```

```
    Options Indexes ExecCGI
    AllowOverride None
    order deny,allow
    allow from all
</Directory>

# 要求された URL にファイル名がない場合、デフォルトで表示されるファイル名
DirectoryIndex index.html

# ディレクトリのアクセスコントロールファイル
# ディレクトリに AccessFileName で指定されたファイルが存在する場合、そのファイルの定義で access.conf の定義を上書き。
# AccessFileName .htaccess

<Files ~ "^¥.ht">
    Order allow,deny
    Deny from all
</Files>

DefaultType text/plain
# DNS の逆引き解決
# このディレクティブを on に設定した場合、逆引きで得られたホスト名をログに記録し、off では IP アドレスが書き込まれます。 (on/off)
# on == DNS の逆引きを行う, off == DNS の逆引きを行わない
HostnameLookups on

# エラーを記録するファイルとそのディレクトリを指定
ErrorLog /usr/local/apache/logs/error_log

# アクセスを記録するファイルとそのディレクトリを指定
# TransferLog logs/access_log

# error_log に記録されたメッセージの数
LogLevel warn

# ログファイルのフォーマット
# フォーマット：CustomLog <directory> <command>
LogFormat "%h %l %u %t ¥"%r¥" %>s %b" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent

# .gif, .jpg の呼び出し記録をログに書き込まない
SetEnvIf Request_URI "¥.(gif)|(jpg)$" object-is-image
CustomLog access_log common env=!object-is-image
CustomLog referer_log referer env=!object-is-image
CustomLog agent_log agent env=!object-is-image

# 実行したいＣＧＩディレクトリのエイリアスを指定
ScriptAlias /cgi-bin/ "/home/httpd/html/cgi-bin/"

<Directory "/home/httpd/html/cgi-bin">
    AllowOverride None
    Options None
    Order allow,deny
    Allow from all
</Directory>

# INDEX 表示を行う際に、ReadmeName で指定されたファイルが要求された場合、リストの終わりにその内容を追
```

加します。HeaderName で指定されたファイルが要求された場合、リストの前にそのファイルの内容を表示します。
ReadmeName README
HeaderName HEADER

指定した拡張子を持つファイルをファイルタイプに関連付ける
AddType application/x-tar.tgz

指定した拡張子を持つファイルを handler-name に関連付ける
拡張子が.cgi のファイルは実行可能な CGI スクリプトとして扱われる
AddHandler cgi-script.cgi
AddType text/html.shtml
AddHandler server-parsed.shtml

4.8 Apache の簡易設定

VineLinux では、フルインストール後、何も設定しなくても Apache の起動は出来ます。

・www サーバー公開 Apache の設定
httpd.conf の編集　：/etc/httpd/conf の httpd.conf をテキストエディタ(gedit)で編集。
/etc/httpd/conf の httpd.conf ファイルを開くには、
〔root〕→〔/〕→〔etc〕→〔httpd〕→〔conf〕→〔httpd.conf〕を選択〔了解〕を選択。
httpd.cont が開きます。httpd.conf の編集ができます。

```
#ServerName        www.net.co.jp
          ↓
ServerName         www.net.co.jp   ドメインの指定
```

例えば、この場合だと www.net.co.jp というサーバ名が Apache に対して指定されます。この変更は絶対にしなければならないものではありませんが、変更しなかった場合にホスト名がネームサーバに登録されていなければ起動ができなくなるので極力明示的に指定してください。なお、設定変更後は設定ファイルに誤りが無いかどうかをチェックしてください。チェックには、apachectl というコマンドを利用します。これは、Apache を起動したり設定ファイルの読みなおしを指示したり、設定ファイルの誤りのチェックをしたりするコマンドで、通常のインストールを行えば入っています。

```
[root]# /usr/local/apache/bin/apachectl configtest
Syntax OK
[root]#
```

これで、Syntax OK と表示されれば設定ファイルには誤りが無いことが分かります。もしここで、ServerName の表記を間違って SererName と書いてしまったとしましょう。この場合は、次のようにエラーメッセージが表示されるはずです。

```
[root@]# /usr/local/apache/bin/apachectl configtest
Syntax error on line 274 of /usr/local/apache/conf/httpd.conf:
Invalid command 'SererName', perhaps mis-spelled or defined by a module
not included in the server configuration
[root@]#
```

ここで、Warning が出た場合は何らかの問題があることを意味しますが、Apache の起動は妨げませんので、スタートすることは可能です。（ただし、ファイルが表示されないなどの弊害が出ることはあります）

Apache の起動
いよいよ、Apache の起動です。Apache を起動するには apachectl を利用します。

```
[root]# /usr/local/apache/bin/apachectl start
[root]#
```

もし、次のようなエラーが出た場合はホスト名が正常に認識できなかったということですので、ネットワーク管理者に頼んでネームサーバに自分のホスト名を登録してもらうか、設定ファイルに ServerName を書き込んでください。なお、前項で ServerName を書きこんでいれば次のエラーは発生しません。

```
httpd: cannot determine local host name.
Use the ServerName directive to set it manually.
/usr/local/apache/bin/apachectl start: httpd could not be started
```

その他の理由により起動しなかった場合は、作成中のエラーリファレンスを参照してください。これで、問題無く起動すれば、ブラウザでアクセスを試みましょう。無事アクセスができれば、インストールはひとまず完了です。
It Worked! The Apache Web Server is Installed on this Web Site!
さあ、これから長い Apache マニアの道が始まります。 がんばりましょう。

User apache を編集。
HostnameLookups off を編集。HostnameLookups on に書き換えします。
#AddHandler cgi-script.cgi の#を削除。

・httpd の起動
kterm.sh 起動画面より、[httpd.start]
・httpd の再起動　：[httpd.restart]

・www サーバー公開 Apache の自動起動設定　：httpd.conf の編集

[/root]#setup
　カーソルキー［↓］で[システムサービス設定]に移行し、[Tab]キーで[設定ツール]を実行に移行し、カーソルキー［↓］で httpd まで移行し、[Space]キー実行で[httpd]にチェック〔*〕を入れる。[Tab]キーで完了に移行し、[Tab]キーで[終了]に移行。

５．Webmin の起動

Webmin とは、Vine Linux に標準でインストールされる環境設定ユーティリティ。その設定項目は細かく、システムの基本設定からサーバの設定など幅広く対応。Webmin の使用により、DNS, Samba, NFS, Apache, Postfix などの設定が Web ブラウザ上から行え、ローカルまたはリモートからも設定が可能となる。

◎Webmin 起動
【URL は https://localhost:10000/】
※必ずアップデートしてから実行する事!

承認書の Window が開きます。続けるを選択。暗号化されたページを要求しています。Window が開きます。OK を選択。
Webmin にログインの Window が開きます。
ユーザ名とパスワードを入力。
ユーザ名：root
パスワード：root で登録したパスワード

パスワードマネージャの Window が開きます。OK を選択。パスワード及びその他の機密情報を保存しています。Window が開きます。
OK を選択
Webmin の Window が開きます。完了！

Webmin のアップグレード
サーバ機上の Webmin を使用して作業を進めます。
メニューバーから、「webmin」をクリック。次に表示された画面より,「webmin 設定」をクリック。後は、「Webmin のアップグレード」をクリックします。
www.webmin.com からの最新バージョンにチェックをつけ、「webmin アップグレード」をクリック。
http://www.aquacities.2y.net/linux_webmin.html

Webmin の起動
ブラウザーを起動して[localhost:10000]へアクセスする。

Webmin 起動画面

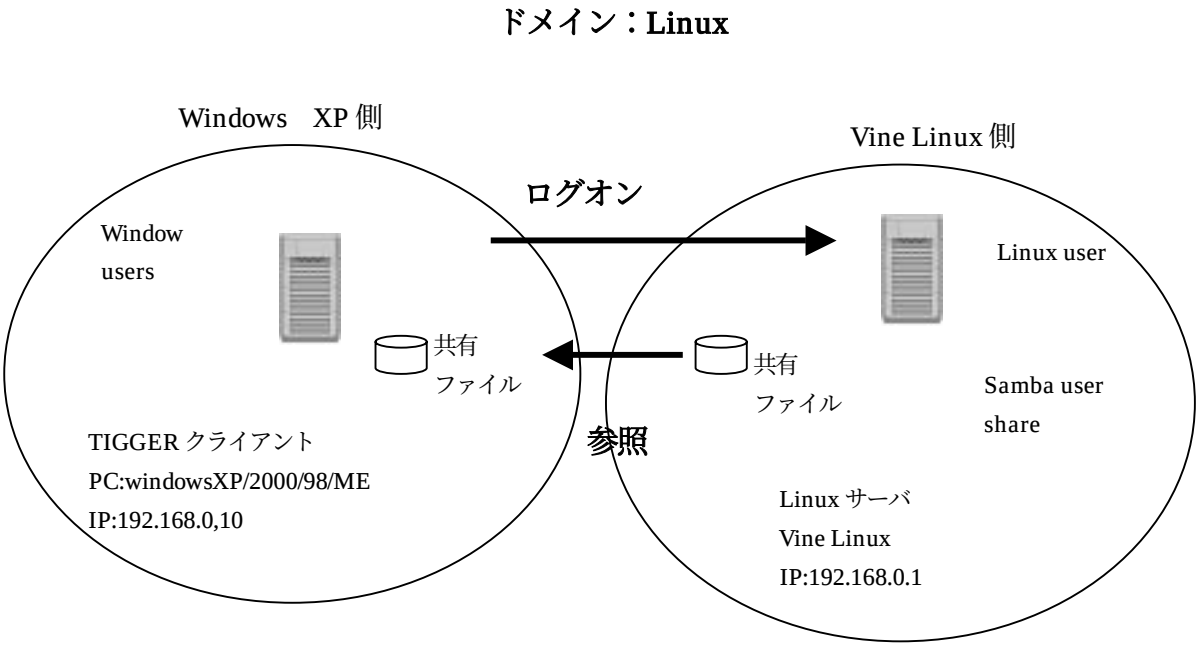

Webmin
Refresh
Log Out
Webmin
システム
サーバ
ネットワーク
ハードウェア
Cluster
その他
サーバ
Apache Webserver
BIND DNS サーバ
CVS Server
DHCP サーバ
Fetchmail メール転送
Jabber IM Server
OpenSLP Server
Postfix の設定


Apache Webサーバ
Apache version 1.3.27
グローバルな設定
プロセスと制限
ネットワークとアドレス
Apacheモジュール
MIMEタイプ
MISC
その他
CGI
CGIプログラム
ディレクトリ毎のオプションファイル
既知モジュールの再設定
定義済みパラメータの編集
Edit Config Files
仮想サーバ

６．Samba サーバーの設定

Samba(サンバ)は、以下の UNIX および UNIX 互換マシンを Windows NT/2000 互換のファイルサーバ/プリント・サーバにするオープン・ソース・ソフトウェアです。GPL(GNU General Public License)の元、自由に利用することができる。Samba はオーストラリアの Andrew Tridgell 氏らによって 1992 年に開発されました。現在 Linux のブームにより、Linux 上のユーザ数が急増していますが、Sun や HP の UNIX マシンをベースとした数千、数万のユーザをサポートする Samba マシンが企業のインフラとして長い間活躍。現在世界で 1000 万台以上の UNIX マシンで Samba は稼働しており、Linux の普及に伴って、Windows NT/2000 サーバを置き換える事例も増えている。

現在 Samba は、Quantum 社の Andrew Tridgell 氏や VA Software 社の Jeremy Allison 氏らによってボランティアではなく、専任の担当者によって、開発・サポートが行われている。(日本人を含めた世界中のボランティアの方も多数参加)今後リリース予定の Samba 3.0 では、Windows ドメインコントローラの複製サポートや Active Directory のサポートなど、ますます企業や学校などの大規模システムでの利便性・適応性が増すことでしょう。



6.1 インストールされている Samba のバージョン確認

現在インストールされている Samba のバージョンを確認することが出来る。

```
# rpm -qa | grep samba
  samba-2.0.10_ja_1.2-0vl1
  samba-client-2.0.10_ja_1.2-0vl1
  samba-common-2.0.10_ja_1.2-0vl1
```

6.2 最新バージョンのダウンロード

Samba のモジュールは、日本 Samba ユーザ会より入手できます。http://www.samba.gr.jp/

入手時期により、Samba のバージョンは違いますが、

・オリジナル最新版：X

◎日本語版最新リリース：samba-2.2.4.ja-12.i586.rpm

・日本語版開発中：X

等が常時あります。安定性を求めるなら、日本語版最新リリースが良い。入手したいバージョンの Samba をクリックします。

6.3 旧バージョンを削除(現在の設定情報のバックアップをお薦めします)

削除方法：登録されたパッケージを調べる。

```
[root]# rpm -qa | grep samba
        samba-2.0.10_ja_1.2-0vl1
        samba-client-2.0.10_ja_1.2-0vl1
        samba-common-2.0.10_ja_1.2-0vl1
```

この出力結果の各パッケージ(3 個)を、rpm -e にて削除。rpm -e 以降の値は、rpm -qa | grep samba の出力結果それぞれ指定し、実行。

```
[root]# rpm -e samba-2.2.2.ja-10
[root]# rpm -e samba-client-2.2.2.ja-10
[root]# rpm -e samba-common-2.2.2.ja-10
```

6.4 samba インストール

(1)Samba のバージョンは、刻々更新されている。今回のインストールは、CD-ROM からインストールする。

[ツールバー]→[周辺機器]→[CD プロパティ]設定

- CD を挿入したら自動的にマウントする
- CD を新しくマウントしたら Auto-run プログラムを自動的に実行する
- CD を新しくマウントしたらファイルマネージャーのウィンドウを開く

にチェックを入れる。→[OK]

(2)CD-ROM(samba-2.2.4.ja-12.i586.rpm)を挿入する。

(3) ファイルマネージャーが表示されたら、該当するファイルをクリックして「コピー」を選択。転送先を「/tmp」にして実行する。

(4)CD-ROM のアンマウント

CD-ROM は、自動的にマウントされている。そのままでは、CD-ROM を取り出すことは、出来ない。必ず、アンマウントを実行してください。

```
[root]# umount /dev/cdrom
[root]#cd /tmp
```

(4) ファイルの解凍 /パッケージ操作を参照しインストールを実施してください。下記の例では、samba-2.2.4.ja-12.i586.rpm をインストールしている例。Samba のインストールは、root 権限で実施します。

```
[root]# rpm -ivh samba-2.2.4.ja-12.i586.rpm
Preparing... ########################################### [100%]
1:samba ########################################### [100%]
[root]#
```

6.5 /etc/samba/smb.conf の変更

多くの Linux ディストリビューションでは、Samba 日本語版があらかじめインストールされていて、初期設定を行えばすぐに利用可能です。無ければディストリビューションの FTP サイトか、日本 Samba ユーザー会の Web サイトなどから入手してインストール。Samba の設定は、設定ファイル smb.conf(/etc または/etc/samba にある)を編集するわけですが、直接このファイルを編集する以外に SWAT と言う Web ブラウザを使ったツールもあります。こちらを使っても良い。ただし Samba の設定を行う前に必ずネットーワークの設定を済ませる事。

設定ファイル smb.conf の内容

```
[global]
workgroup = N-GROUP --> (1)ワークグループを指定。Windows クライアントと同じ名前を指定。
server string = Samba Server --> (2)
security = user --> (3)
```

encrypt passwords = yes --> (4)
socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192 --> (5)
; interfaces = 192.168.12.2/24 192.168.13.2/24 --> (6)
map to guest = Bad User --> (7)
guest ok = Yes
coding system = euc --> (8)
client code page = 932 --> (9)

[homes] --> (10)各ユーザが LINUX のホームディレクトリを使うための設定
comment = Home Directories
browseable = no --> Windows など他からの表示を可能にするか Windows から home が見えない(可能：yes)
writable = yes --> 書き込みを許可

[samba]
comment = Kyoyu -->(11)
path = /home/samba --> (a)
read only = no --> (b)
guest only = yes --> (c)
guest ok = yes --> (d)

[public]
comment = Public space; anyone can write any files.
path = /home/samba/public
force group = public
writeable = Yes
force create mode = 0664
force directory mode = 0775
guest ok = Yes

(1) ワークグループを指定。Windows クライアントと同じ名前を指定。

(2) Windows の「ネットワークコンピュータ」または「マイネットワーク」に表示されるネットワーク上の Linux サーバーの「プロパティ」に表示される文字列。ここでは "Samba Server" としていますが、"Samba %h"とするとホスト名が表示。

(3) 下記のようにいくつか指定
user: Linux サーバー(Samba サーバー)にアクセスするユーザーアカウントを作成し、そのアカウントでアクセスする場合この認証モード。今回の事例ではこの方法にしている。
share: Linux サーバー(Samba サーバー)に存在するユーザーアカウントを使用してアクセスする。
domain: ネットワーク上の Windows ドメイン側で認証したい場合はこのように指定。

(4) 暗号化パスワードを使用する場合このように指定。Windows95 OSR2,Windows98,Windows Me,WindowsNT4.0/2000 ではこの設定。

(5) パフォーマンスの設定。

(6) 通常はこのままで構いません。サーバーに NIC がふたつ接続されているいるような環境で、 ここで行頭の;を消して有効にする。

(7) guest ユーザーにアクセスを許可するかどうか設定。
Bad User: ゲストユーザーとしてのアクセスを許可。
Never: ゲストユーザーのアクセスを拒否。デフォルトではこの設定。

(8) 下記のようにいくつか指定。
euc: 通常この設定。Linux 上でかな漢字ファイル名を利用し易くなりますが、環境によって "????" など文字化けすることもある。
sjis: Windows で使う Shift JIS の指定。Linux 上でかな漢字ファイル名を利用するには問題がある。
cap: Machintosh クライアントと,Netatalk で接続している場合にはこの設定。

(9) Windows 日本語版を使っている場合は、932 にする。

(10) 共有フォルダの設定。[homes]のままにしておくと、Linux 側で作成したユーザーのホームディレクトリにアクセスすることになる。

(11) 別途、共有フォルダを作成した場合の一例。この例では Linux 側に新規ユーザーとして samba を登録し、共有フォルダを/home/samba としている。

(a) Windows の「ネットワークコンピュータ」または「マイネットワーク」に表示されるネットワーク上の Linux サーバーの説明欄に表示される文字列の指定。日本語名は文字化けします。

(b) 共有フォルダのフルパス。

(c) guest ok = yes を指定している場合、共有フォルダ内のファイル操作が guest によって行われるようにする設定。

(d) yes にするとパスワード入力が不要。

ここまでが Samba を利用するための smb.conf の内容。必要に応じてこのように編集すればいいわけですが、上述のように直接 smb.conf を編集しなくても SWAT を使ってより簡単に設定。

6.6 設定の確認

smb.conf の設定が終ったら、設定が正しいかどうか testparm でチェックする。

```
#testparm
```

問題がなければ samba を起動する。

6.7 パスワードの設定

Windows95 では平文パスワードを使用していたので、Linux マシン上の/etc/passwd を利用してままユーザー認証ていました。でも、Windows98、WindowsNT の 4.0 以降、あと、Windows2000 では、暗号化パスワードを使用しているので、この場合は Windows マシンと Linux マシンの両方で、認証方式を一致させる必要があるのです。一致させる方法としては 2 通りあります。ひとつは、Windows 側で暗号化パスワードを無効にする方法。もうひとつは、samba 側で暗号化パスワードを使用する方法です。JB としては、Windows 側で平文パスワードを使用するより、samba で暗号化する方がお勧めなので、ここでは samba で暗号化されたパスワードを扱う方法を説明します。
・Windows で設定したものと同じユーザーを Linux 上で作成します。
・Windows98 だと、ログインしたユーザー名と一致していないと SAMBA フォルダへアクセスできないので、特に理由がなければ同じユーザー名及びパスワードを使用するのが無難です。

Samba ユーザ作成は、 追加するユーザは /etc/passwd に登録済みであること。

Samba を利用するには Samba ユーザーを作成する必要。Samba の設定で、Samba 用に新しいユーザーを作成。そのホームディレクトリを共有フォルダとして公開したい場合には、次のようにして新規ユーザーを登録。まず、通常の Linux のユーザーアカウントの作成。ここでは samba というユーザー名を作成。

useradd -m share

次にユーザーshare のパスワードを作成。これは必ず行う必要。

smbpasswd -a share

ここで以下のようにパスワードを入力されるように促されるので入力。

New SMB password:xxxxxx

Retype New SMB password:xxxxxx

パスワードの入力が完了し、以下のメッセージが表示されると無事終了。

Password changed for user share.

既に Linux サーバーにあるユーザーアカウントをそのまま使う場合は、ここの例では Samba の設定ファイル関連(smb.conf を含む)が/etc/samba 以下にあるとします。環境によっては/etc 直下にあります。

mksmbpasswd.sh < /etc/passwd > /etc/samba/smbpasswd

chmod 600 /etc/samba/smbpasswd

次にユーザーshare を登録。

smbpasswd share

ここで以下のようにパスワードを入力されるように促されるので入力。

New SMB password:xxxxxx

Retype New SMB password:xxxxxx

パスワードの入力が完了、以下のメッセージが表示されると無事終了。

Password changed for user share.

これで Samba を利用するためのユーザーアカウントの作成は完了。

暗号化したパスワードを使用するように Linux システム上の Samba を設定する場合は、次の手順に従ってください。Samba 用の別個のパスワードファイルを作成します。既存の/etc/passwd ファイルに基づいてこのパスワードファイルを作成する場合は、次のコマンドを入力。

---

cat /etc/passwd | mksmbpasswd.sh > /etc/samba/smbpasswd

chmod 600 /etc/samba/smbpasswd を使用して、root だけが読み取り/書き込みを行えるように Samba パスワードファイルのアクセス権を変更。

このスクリプトでは、ユーザーのパスワードは新しいファイルにコピーされません。各 Samba ユーザーのパスワードを設定するには、コマンド smbpasswd username を使用します(username には各ユーザーのユーザー名を指定)。Samba ユーザーアカウントは、対応する Samba パスワードが設定されるまで有効になりません。

暗号化パスワードは Samba 設定ファイルで有効にする。ファイル smb.conf で、次の行のコメントがないことを確認します。

encrypt password = yes

smb passwd file = /etc/samba/smbpasswd

シェルプロンプトでコマンド service smb restart を入力して、smb サービスが起動されていることを確認。smb サービスを自動的に起動させたい場合は、ntsysv、chkconfig、serviceconf を使用して、ランタイム時にこのサービスを有効にしする。

ヒント

passwd コマンドの使用時に、ユーザーの Samba パスワードとシステムパスワードを同期化するために pam_smbpass PAM モジュールを使用することができます。ユーザーが passwd コマンドを起動すると、Linux システムへのログインに使用するパスワードと Samba 共有への接続に使用するパスワードは変更されます。この機能を有

効にするには、以下の行を pam_cracklib.so の下の/etc/pam.d/system-auth に追加。

password required /lib/security/pam_smbpass.so nullok use_authtok try_first_pass

6.8 Samba の起動

ここまでの設定が済めば、Samba を利用できる準備は整いました。早速、Samba を起動。

次のコマンドを実行。

/etc/rc.d/init.d/smb start

これで Windows の「ネットワークコンピュータ」または「マイネットワーク」を開くと Linux サーバーの共有フォルダが見える。

6.9 ランレベルの変更

ntsysv

→ smb サービスを有効にする事。

6.10 Samba サーバの起動

[root]# /usr/sbin/smbd -D

[root]# /usr/sbin/nmbd -D

6.11 Samba サーバの共有ディレクトリ表示

[root]# smbclient -L Linux(ホスト名)

6.12 Samba サーバ接続状況

確認する方法が幾つかある。接続状況を表示する smbstatus コマンド、接続履歴を記録する utmp オプション。企業においては、Samba サーバへの接続状況を監視したいという場合も多いでしょう。接続状況を表示するコマンドとして、従来から Samba に備わっていたものに smbstatus コマンドがあります。このコマンドを利用することで、実行例のように、現在の Samba サーバへの接続状況を表示することが可能。

実行例: smbstatus コマンドの実行例

```
[root]# smbstatus
Samba version 2.2.4.ja-12
Service      uid      gid      pid      machine
----------------------------------------------
monyo        monyo    monyo    26361    misa      (192.168.221.128) Fri Jun 22 00:44:18 2001
Locked files:
Pid      DenyMode    R/W          Oplock              Name
--------------------------------------------------
26361    DENY_NONE   RDWR         EXCLUSIVE+BATCH   /home/monyo/lion.txt     Fri Jun 22 00:45:07 2001
26361    DENY_NONE   RDWR         EXCLUSIVE+BATCH   /home/monyo/サンバ.txt    Fri Jun 22 00:46:05 2001
26361    DENY_WRITE  RDONLY       EXCLUSIVE+BATCH   /home/monyo/CharGen.exe   Fri Jun 22 00:44:43 2001
Share mode memory usage (bytes):
   1048176(99%) free + 296(0%) used + 104(0%) overhead = 1048576(100%) total
```

SWAT の「状況(STATUS)」ボタンを押すことで表示される画面でも同様の情報が表示。しかし、これらのツールでは現在の接続状況を取得することができても、誰がいつ接続したかという接続履歴の情報を取得することはできません。従来の Samba でこうした情報を取得する場合は、smbstatus コマンドを cron などで定期的に実行してその結果を加工する必要がありました。

Samba 2.0.7 から実装された utmp オプション

Samba 2.0.7 以降では、utmp オプションにより、接続履歴の情報を簡単に取得することが可能になっています。utmp オプションを利用する場合の注意点としては、configure 時に--with-utmp オプションを指定する必要がある点があげられます。パッケージを利用する場合は、このオプションを付けて configure が行なわれていることを確認。なお、Samba 2.0.7 では、実装の不備により、正しくコンパイルできるプラットフォームが Linux などごく一部に限られている。もしコンパイルがうまくできない時は Samba 2.0.7 日本語版や、Samba 2.2.0 以降を利用する。--with-utmp オプションを有効にした Samba を稼働させれば、後は単純に smb.conf 中で 1 行追加を行なうだけでこの機能が有効になる。

utmp オプションを有効にする

```
[global]
  utmp = yes
```

このオプションを有効にしている場合、last コマンド等で実行例 2 のように Samba サーバへの接続履歴が表示。

実行例: last コマンドの実行例

```
last | grep smb
monyo      smb/0          misa                Sun Jun 17 10:28 - 11:36   (01:08)
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)
smbguest smb/0          yukako              Sat Jun 16 22:48 - 22:58   (00:10)
```

なお、デフォルトでは通常のログイン履歴を記録するファイルと同一のファイルに Samba サーバへの接続履歴も書き込まれます。Samba 経由でのアクセスの記録を別のファイルに出力したい場合は、utmp directory オプションを利用することで明示的に書き出すファイルを指定。

書き出すファイルの位置の指定

```
[global]
  utmp = yes
  utmp directory = /var/log/samba/
```

なお書き出し先のファイル名は utmpx と wtmpx になります。予め

```
  touch /var/log/samba/utmpx
```

のようにして作成しておく必要がありますので注意。また last コマンド実行時も、実行例のように、-f オプションで明示的にファイル名を指定する必要がある。

実行例: last -f コマンドの実行例

```
%last -f /var/log/samba/utmpx
tako       smb/1          misa                Thu Jun 21 23:54     still logged in
```

utmp ファイルへの出力の応用

utmp ファイルに書き込まれた情報を利用することで、実行例のようにユーザの接続時間の集計などを簡単に行なうことが可能。smbguest アカウントの接続時間を日毎に出力。

実行例: ac コマンドの実行例

```
%ac -d smbguest
Jun  2   total          1.81
Jun  3   total          0.95
Jun  4   total          0.30
Jun  6   total          0.05
Jun  9   total          7.03
Jun 16   total          2.35
Jun 17   total         13.24
```

6.13 Windos 側の設定を確認

Windows から Samba サーバーに接続するための設定

・TCP/IP プロトコルの組み込み

・「Microsoft ネットワーククライアント」の組み込み

・ワークグループ名もしくは、ドメイン名も Samba サーバーと同一であるほうが良い。

◆暗号化認証の設定

WindowsXP から Samba のドメインにログオンする場合、正しくログオンできない可能性があります。
これは、暗号化認証の設定を変更する必要がある。Windows XP での暗号化認証の設定を変更するには、

[コントロールパネル]
　　|
[管理ツール]
　　|
[ローカル セキュリティ ポリシー]
　　|
[セキュリティーオプション]

から"ドメインメンバ:常にセキュリティーチャネルのデータをデジタル的に暗号化または署名する"の値を"無効"にします。

Wondows98/WindowsNT(ServicePack3 以降) では、ユーザー認証に暗号化パスワードを使用しているため、samba 用のパスワード設定が必要。パスワードの設定は root 権限を持った人しかできない。

Windows95/98 の場合

lmhosts というファイルを作成し、Windows ディレクトリに置きます。作成にはノートパッドが使えます。サフィックスはつけない。

lmhosts の内容

192.168.0.1 Vine Linux

コンピュータの検索で samba をサーチするか、デスクトップ上のネットワークコンピュータを開き、ネットワーク全体にある、VSL をダブルクリックすると samba が見つかる。

WindowsNT の場合

Winodws95 の場合と同様な lmhosts ファイルを作成し、Winnt¥system32 ディレクトリに置きます。

コントロールパネルで、ネットワークの設定を開き、WINS アドレスタブを選択します。そこで LMHOSTS 参照を行うのチェックボックスを ON にし、LMHOSTS を取りこみます。

注 ServicePack3 の場合はレジストリの変更が必要になります。(Service pack3 は暗号化パスワードを使うが、VSL のほとんどの PC は暗号化パスワードを使わないのでその整合性を取るため)

[HKEY_LOCAL_MACHINE¥SYSTEM¥CurrentControlSet¥Services¥Rdr¥Parameters]

"EnablePlainTextPassword"=dword:00000001
　なお、レジストリの変更には十分注意してください。

6.14 各種設定状況の確認

　一通りの設定が完了したのであれば、念のため各種設定状況を確認。
Check 1：VlineLinux からの共有ディレクトリ一覧表示
Check 2：VlineLinux から共有ディレクトリにアクセスしてみる
Check 3：windows からの接続テスト
Check 4：Samba サーバー起動の確認

6.15 Samba の security(セキュリティレベル)

　セキュリティレベル は、以下の 4 つのうちのどれかに置き換えます。
user
share
server
domain

ユーザーレベルセキュリティ：security = user
　まずはデフォルトで指定されているユーザ・レベルのセキュリティについてです。 ユーザ・レベルのセキュリティでは、クライアントは共有ディレクトリをマウントする際に、ユーザー名とパスワードを自動送信します。サーバはそのユーザ名/パスワードの組み合わせを許可又は拒否します。この段階において、サーバーからは、クライアントがどの共有に接続を 試みているのかわかりません。そのため、サーバはユーザー名・パスワード・ホスト名のみで許可または拒否を判断しています。
クライアントは複数のアクセス要求を送ることも可能となっています。 それに対してサーバが応答するときには、ユーザ名/パスワードに対する認証タグ として使用される「uid」をクライアントに与えています。 クライアントはこの方法により複数の認証コンテクストを維持することが可能となります。

共有レベルセキュリティ：security = share
　共有レベルの セキュリティでは、各共有に対して、ひとつまたは複数のパスワードが与えられます。 クライアントは各「tree connection」(共有しているディレクトリのマウント)とともにパスワードを送信しますが、ユーザー名は送信しません。共有のパスワードを知っているユーザーであれば、誰でもアクセス可能となっています。パスワードは、例えばひとつを読み取り専用、ひとつを読み書き可能、といった形で複数指定することも可能です。

サーバーレベルセキュリティ：security = server
　サーバレベルのセキュリティは、 ユーザレベルセキュリティと同様、ユーザー名・パスワードにて認証を行いますが、Samba はクライアントが送信したパスワード及びユーザー名を、別の SMB パスワードサーバ、通常は別の Samba サーバかネットワークで PDC として機能する Windows NT Server に送信し、認証を委任します。共有の設定に付いては、Samba の smb.conf ファイル中で設定が保持され、クライアントが特定の共有に対して接続を行なおうとすると、Samba はユーザが共有に接続する権限があるかどうかをここで認証。実際のパスワードサーバが NT サーバであった場合、 Samba は、パスワードを NT にそのまま渡すだけなので、encrypt passwords = yes とする必要はありませんし、smbpasswd も作成する必要もありません。
　この場合には smb.conf の[global]セクションを以下のように変更。
security = server
workgroup = EXAMPLE
password server = HOST1PASSSERV2
　ここで、EXAMPLE は NT ドメイン名、HOST1 は NT のプライマリ・ドメイン・サーバ名、PASSSERV2 はバックアップ・ドメインコントローラ名です。ここでは DNS で解決されたホスト名ではなく、NetBIOS 名を指定する必

要があることに注意して下さい。NT ワークステーションでも１台だけでも設定可能です。また、これを用いる場合でも、UNIX 上での操作に必要であるため、samba サーバー上には各ユーザーのアカウントが必要であることに注意。

ドメインレベルセキュリティ：security = domain
　ドメインレベルのセキュリティはサーバレベルのセキュリティと類似していますが、このとき、Samba サーバは Windows ドメインのメンバーとして動作します。ワークグループ内のユーザー認証はドメインコントローラにて行われます。ドメインコントローラは、ユーザとパスワードを自身のセキュリティ認証モジュール(SAM)の中に記録しています。NT ドメインにて認証を行うと、許可・拒否のみだけではなく、ユーザー属性がフルセットで、戻り値として返されます。これには以下のような情報が含まれます。
ユーザ名
名前
説明
セキュリティ識別子
NT グループでの所属情報
ログオン可能な時間及びユーザが直ちにログオフする必要があるかどうか利用可能なワークステーション
　アカウントの有効期限
　ホームディレクトリ
　ログオンスクリプト
　プロファイル
　アカウントのタイプ
　また、ドメインコントローラを使用して認証を行うと、Samba サーバと常に接続しておく必要がなくなります。security = server オプションの時に用いられるプロトコルとは異なり、Samba サーバは認証情報が必要なときだけに Remote Procedure Call(RPC)を発行。

　SAMBA サーバーを NT ドメインに追加するには、Samba のデーモンをすべて停止。WindowsNT のサーバー・マネージャー(SRVMGR.EXE)を起動、または、マイネットワークアイコンを右クリックし、プロパティを表示させます。メニューの「コンピュータ」―「ドメインに追加」をクリックし、「WindowsNT ワークステーションまたはサーバ」でを選択します。ここで、SAMBA サーバーの NetBIOS 名を追加します。Samba サーバ上で以下を実行し、NT ドメインのメンバーに追加します。
smbpasswd -j GROUP -r　PDCname
　ここで、GROUP は NT ドメイン名、PDCname は NT のプライマリ・ドメイン・サーバ名とおきまえます。もしうまく行ったら、以下のメッセージがターミナルウインドウに現れる。
smbpasswd: Joined domain GROUP.
　/etc/samba/smb.conf の[global]セクションを以下のように変更します。
　　security = domain
　　domain logins = yes
　　encrypt passwords = yes
　　workgroup = GROUP
　　password server = PDCname
　ここで、GROUP は NT ドメイン名、PDCname は NT のプライマリ・ドメイン・サーバ名です。

SAMBA を PDC のプライマリサーバーとして使用
　同じ PDC の使用といっても、SAMBA をその中へメンバーとして追加するのと、SAMBA 自身を PDC サーバーとして使用するのでは、設定が大きく異なります。まず、Samba はユーザレベルのセキュリティで構成されている必要があります。Samba がサーバーの時は、ドメインレベルのセキュリティは利用できません。/etc/samba/smb.conf の[global]セクションを以下のように変更。
　security = user とします。domain logins = yes の行のコメントアウトをはずします。WindowsNT など、暗号化パスワードを使用している OS と共有する場合は、以下の行も encrypt passwd = yes の行も有効にしておく必要があります。

smb passwd file = /etc/samba/smbpasswd
os level = 34 とし、コメントアウトをはずします。以下の 3 つを追加（又はコメントアウト）して、有効にします。
local master = yes
preferred master = yes
domain master = yes
logon script = %U.bat
　以下のようにログオン用の共有フォルダを定義します。PATH はどこでもかまいません。
[netlogon]
comment = The domain logon service
path = /export/samba/logon
public = no
writeable = no
browsable = no
　この後、touch コマンド等で/export/samba 以下に logon ファイルを作成。smb を restart。

７．SWAT の設定

7.1 WEB ベースの Samba 設定ツール(SWAT)

swat の設定　→xinetd の設定を変更。リモート経由で、SWAT を起動したい場合、F/W の設定と xinetd の設定を変更。xinetd に関する設定を変更。xinetd に関する設定ファイルは、/etc/xinetd.d/swat に存在します。以下に、/etc/xinetd.d/swat の初期値を確認します。

```
[root]# cd /etc/xinetd.d
[root]# more swat
service swat
{
disable = no
port = 901
socket_type = stream
wait = no
only_from = 127.0.0.1 ←削除
user = root
server = /usr/sbin/swat
log_on_failure += USERID
}
```

only_from に関しては、SWAT が利用できるクライアントを特定する記述。どのクライアントからでも、SWAT を利用した場合は、only_from の行を削除。また、ある特定のクライアントのみ接続させる場合は、only_from = 192.168.1.0/24 等と指定します。上記の例では、192.168.1.xxx 以外のＩＰアドレスのクライアントの接続を禁止しています。

/etc/xinetd.d/swat の設定変更が完了した後、以下のコマンドを実行します。root 権限で、再起動します。

```
[root]# chkconfig swat on
[root]# service xinetd restart
xinetd を停止中: [ OK ]
xinetd を起動中: [ OK ]
```

7.2 SWAT の起動

Vine Linux 2.1.5 をインストール直後の状態からそのまま SWAT を動作させることはできません。以下のファイルを修正します。

(1)/etc/inetd.conf の下行に追加

```
swat stream tcp nowait.400 root /usr/sbin/swat swat
```

(2)再起動

```
# /etc/rc.d/init.d/inet restart
```

(3)/etc/hosts.allow ファイルに追加

```
swat: 192.168.0.200
```

(4)共有ファイルの作成

```
#cd   /var                ←共有ディレクトリに移動
#mkdir share              ←新規ディレクトリ作成
#chmod a+w                ←書き込み許可
#chown share share        ←オーナ名の変更
#chgrp share share        ←グループ名の変更
```

(5)動作確認

・Vine Linux からの起動

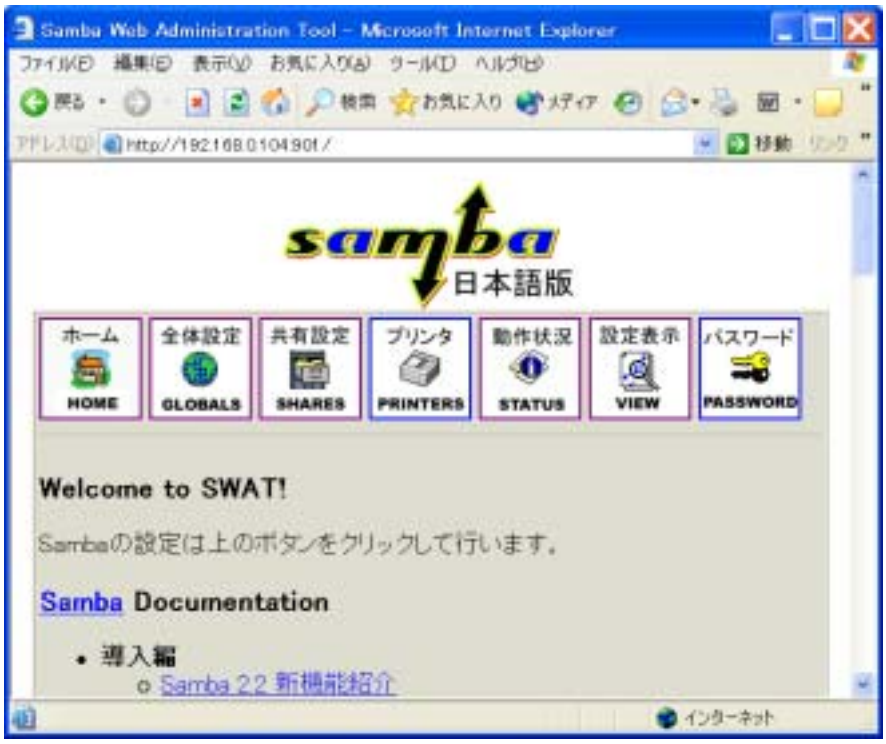
ブラウザー(Mozilla)から接続する。http://192.168.0.200:901 と入力する。

・Windows からの起動

samba 本体が起動されるコンピュータの IP アドレスに修正。上記の修正後、お手持ちの LAN に接続されたパソコンのブラウザから http://192.168.0.1:901 等のアドレスをアクセスすると、ユーザ名入力のをウィンドウが表示されるので、

名前：root

パスワード：Linux に登録してある root のパスワードを入力。正しい名前、パスワードが入力されると下の SWAT ホームが表示されます。



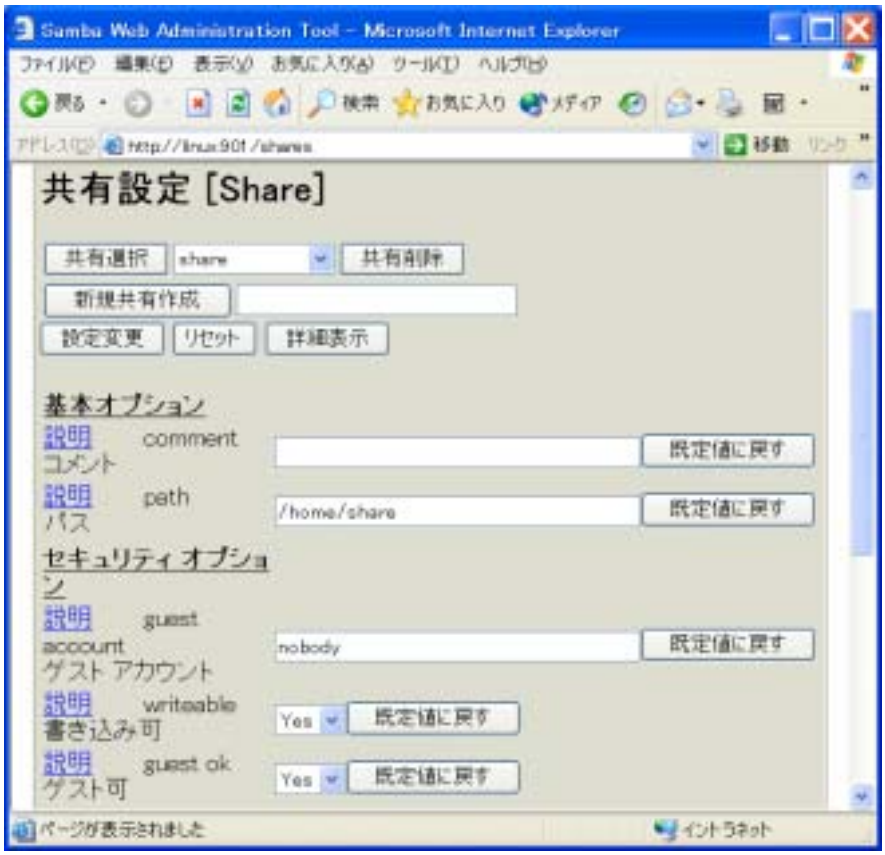
「全体設定 GLOBALS」をクリックすると、全体設定のページが表示される、

coding system euc
client code page 932
workgroup N-GROUP
netbios name SMBATALK
server string %L: Samba %v on %h

interfaces
security USER
encrypt passwords Yes
以下 デフォルトのまま、を入力して、［設定変更］ボタンをクリック。

[共有設定 SHARE]ボタンをクリック。共有設定のページが表示されるので、新規共有作成する。

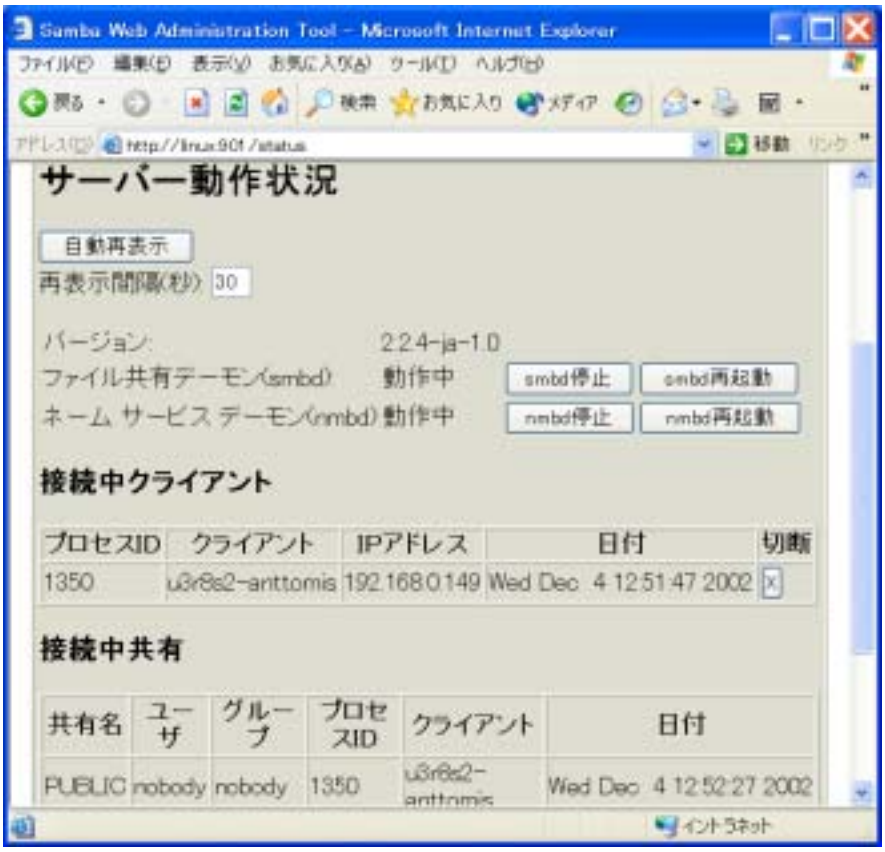
Linux 登録ユーザー名を入力。ここでは、便宜上 ”share ”とします。[新規共有作成]をクリックする。(share は Linux 登録ユーザーで、/var/share ディレクトリが存在しているものとします)



新規共有作成 share を入力すると、共有設定の標準設定ページが表示されるので、
path /var/share
guest account nobody
writeable Yes
guest ok Yes
その他、デフォルトのまま、［設定変更］ボタンをクリック。

[動作状況 STATUS]ボタンをクリック。動作状況のページが表示される。

パスワード(PASSWORD)にてリモートのパスワードを設定する。
「新規作成」ユーザ「share」パスワード「111111」再度パスワード「111111」



停止中の場合　動作中の場合

ファイル共有デーモン　smbd 起動　smbd 再起動

ネームサービスデーモン　nmbd 起動　nmbd 再起動

smbd/nmbd 再起動します。クリックすると、パスワード管理のページが表示されるので、ローカルマシンのパスワード管理

ユーザ名 share

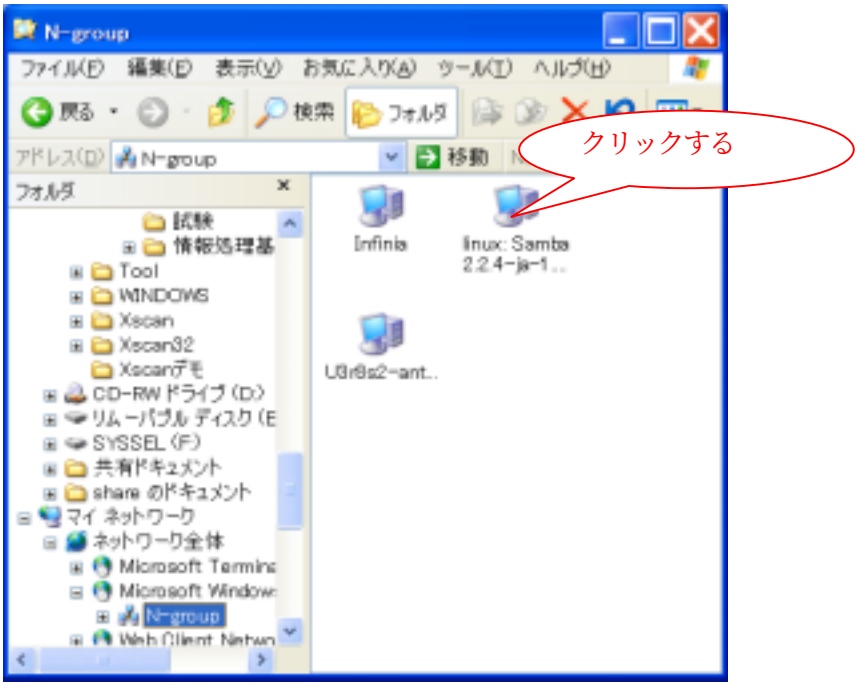
新パスワード ********

新パスワード再入力 ********

自分のユーザ名とパスワードを入力し、[新規ユーザ追加] をクリック。エラーなく設定できたら、成功です。(エラーは”リモートマシンのパスワード管理”の上の行に出ます)

次にウィンドウズマシンを立ち上げ、[マイネットワーク] ―> [近くのコンピュータ] ―> [LinuxSmba]をクリックする。パスワード入力ウィンドウが開くので、先程設定したユーザ名とパスワードを入力。

同じように、”share”フォルダに入っていくと目的の場所に辿り着きます。


N-group
ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)
戻る
検索
フォルダ
アドレス(D) N-group
移動
フォルダ
試験
情報処理基
Tool
WINDOWS
Xscan
Xscan32
Xscanデモ
CD-RW ドライブ (D:)
リムーバブル ディスク (E
SYSSEL (F:)
共有ドキュメント
share のドキュメント
マイ ネットワーク
ネットワーク全体
Microsoft Termina
Microsoft Window
N-group
Infinia
linux: Samba 2.2.4-ja-1...
UGr8s2-ant...
クリックする

7.3 Windows 側の設定

「ネットワークコンピュータ」のアイコンを右クリックし、「プロパティ」を選択しますとダイヤログボックスが開きます。「識別」タブでは以下のように「コンピューター名」と「ワークグループ」の設定をしますが「ワークグループ」は必ず smb.conf に書く名前と一致していなければいけません。次に「サービス」タブを選択、ここで「TCP/IP プロトコル」から「プロパティ」を選択します。ここで「IP アドレス」タブを選択。

これで以下のように WindowsNT の「ネットワークコンピュータ」のアイコンをダブルクリックすると Linux サーバーアイコンが現れますのでこのアイコンをダブルクリックして下さい。

ログインユーザー名が Samba に登録しているユーザーと同一でなければパスワード入力を促すダイヤログが現れますのでこの場合はパスワードを入力。Linux サーバー側の共有ディレクトリが現れます。

(1) Windows98 の設定例

まず「ネットワークコンピュータ」のアイコンを右クリックし、「プロパティ」を選択しますと以下のダイヤログボックスが開きます。ここで、「Microsoft ネットワーク共有サービス」がインストールされている必要があります。無ければ「追加」を押してインストールして下さい。また、"優先的にログオンする"項目が「Microsoft ネットワーククライアント」になるようにして下さい。「ネットワークの設定」タブで TCP/IP(NIC) を選択、「ファイルとプリンタの共有」ボタンを押してチェックを付けておきます。次に「TCP/IP のプロパティ」を開く。「IP アドレス」、「ゲートウエイ」、「DNS 設定」タブでの設定も、必要に応じて行う。次に「ユーザー情報」タブで、「コンピュータ名」、「ワークグループ」を入力。「ワークグループ」は、Linux サーバーの smb.conf と一致させるようにする。Windows98 のログインユーザー名も Samba で登録しているユーザーと同じにしておきます。以上で Windows98 の設定は完了。

(2) Windows2000 のエラーの対処

Samba のバージョンが古いと、Windows2000 で Samba を利用して共有ディレクトリにアクセスすると以下のエラーが出てアクセスできないことがあります。もし以下のようなエラーが出た場合は Samba のバージョンを 2.0.7 以降に上げた方が良い！応急処置としては下記の方法を使えばうまくいく。

(3) エラーの例:

以下の手順で共有ディレクトリにアクセスすることが出来ます。まず「マイネットワーク」を開き「ネットワークプレースの追加」を選ぶ。するとウイザードが起動。ここで「ネットワークプレースの場所」を入れます。例えば Linux 側の NetBIOS 名が Server で、/home/public というディレクトリを共有したいのなら「¥¥Server¥public」と入力。アクセスできるようになります。

(4) Linux のファイルサーバー(Samba)にアクセス

Windows98/Me は SMB サーバー (ファイルサーバー) にアクセスする場合のパスワードを暗号化するようになりました。Windows NT などの OS はこれを解釈可能ですが、Linux などの UNIX OS 用の SMB サーバーである Samba はこの暗号を解釈できないため、Windows98 のレジストリを変更してパスワードの暗号化を解除しないとアクセスできません。

[スタート]ー[ファイル名を指定して実行]から regedit を起動します。

HKEY_LOCAL_MACHNE¥System¥CurrentControlSet¥Services¥VxD¥Vnetsup を開きます。

[編集]ー[新規]ー[DWORD 値] をクリックし、EnablePlainTextPassword を作成します。

この値をダブルクリックして１を設定します。

Windows を再起動します。

また、Samba 側で対応できるオプションもあります。詳しくは「Windows98、WindowsNT4.0、Windows2000 から SAMBA を使用するときの注意事項」を参照してください。

http://www.cityfujisawa.ne.jp/~odagiri/book/encrypt-passwd.htm

7.4 Samba 簡単設定

Windows と Linux 間でのデータ(共有)通信が出来ませんので、Linux 側にソフトのインストールと設定で、Windows とのファイルの共有が可能になる。

(1)パッケージの確認
1.samba-client-2.0.10_ja_1.2_ovl1
2.samba-common-2.0.10_ja_1.2_ovl1
3.samba-2.0.10_ja_1.2_ovl1
以上の３つインストールされていることを確認します。

パッケジが無い時はインストールします。
#rpm-ivh/mnt/cdrom/vine/RPMS/
samba-2.0.10_ja_1.2_ovl1.i386.rpm

(2)共有ディレクトリを作成します。
#mkdir /home/public

(3)作成したディレクトリのパーミッションを 777 に設定。
#chmod 1777 /home/public
共有ディレクトリを作成確認。パーミッション 777 の確認。
#ls -al /home/public

(4)Samba サーバーの設定
smb.conf の変更。テキストエディタを開きます。
/etc/smb.conf
・global：全体の設定を行うセッション
・home：各ユーザ毎のホームディレクトリのセッション
・pninters：プリンタ共有セッション

[global]の項目内容
coding system = SJIS　：Shift JIS 無変換
EUC：Extended UNIX Code に変換
JIS7：７ビット JIS コードに変換
CAP：Macintosh とのファイル共有に使用
ecrypt passwords　：暗号化パスワードで通信かを選択
client code page = 932　：日本語の指定 map to guest ユーザー名が一致しない時の対処
workgroup = ワークグループ名　：Windows ネットワークに使用
security = share
encrypt passwords = yes
socket options　：ソケットに関するオプションを指定
server string　：ホストのアイコンに添付される説明
dns proxy DNS サーバーで名前の検索を設定
guest account ユザー名を指定
os level　：ププラウザに必要なレベルを設定

[public]の項目内容
path =　共有するディレクトリを設定
writeable =　yes　：書き込みを有効にするかを設定
force user = root
create mask = 0666
guest ok = yes　：ゲストユーザの利用を有効にするかを設定

編集が終わりましたら保存します。
(5)smb.com のテスト
#testparm

(6)Samba の起動
#/etc/rc.d/init.d/smb start

Samba の停止
#/etc/rc.d/init.d/smb stop

(7)Samba サーバーの起動設定

システムの設定
[/root]#setup
カーソルキー［↓］で[システムサービス設定]に移行し、[Tab]キーで[設定ツール]を実行に移行し、Enter
カーソルキー［↓］で httpd まで移行し、[Space]キー実行で[smb]にチェック〔*〕を入れる。[Tab]キーで完了に移行し、Enter
[Tab]キーで[終了]に移行し、Enter

Samba が動作し無い時の対処法

/etc/hosts が以下のような書式になっているか見直してください。
127.0.0.1 localhost.localdomain localhost
192.168.0.1 sam.domein.com sam
ここで、2 行目はあなたのマシンの IP アドレスとホスト名です。

８．Anonymous FTP の設定

Anonymous FTP を有効にする。認証クライアント最大数 FTP に同時にアクセス出来る最大クライアント数を設定します。 省略時は無制限とみなされます。

8.1 Vine Linux 2.6 のインストール

ホストの設定
・ホストの設定名：XXXXXXX
任意の名前で入力

・ホスト名
１．サーバーの IP アドレス：192.168.0.200
〔LAN 内の設定〕
２．外部からの接続は URL を入力
〔ユーザからの接続〕

・ユーザ名：YYYYYYYYYY
登録したユーザ名を入力

・パスワード：ZZZZZZZZZZ
ユーザ登録した時のパスワードを入力

8.2 Linux Anonymous FTP の設定

Vine Linux で、外からの anonymous ftp ができない場合、原因は以下が考えられます。ftp サーバが ProFTPD に変更され、これにともなって/etc/inetd.conf の設定が変更になった。インストールしただけでは FTP サーバは起動しないように設定されている。そこで、以下の順で設定の確認を行ってください。
/etc/inetd.conf を確認(upgrade した人だけ)
Vine の ProFTPD は、inetd を経由せずに単独でデーモンとして駐するように設定してある。初期インストールではなく、以前の環境をアップグレードした人の場合、/etc/inetd.conf に ftp に関する以前の設定が残っていることがあるので、エディタ等で確認して下さい。(Vine を新規にインストールした人は不要)以下のように、ftp の行の先頭に#が付いて無効化されていたら正常です。

```
#ftp    stream  tcp     nowait  root    /usr/sbin/tcpd  in.ftpd -l -a
#ftp    stream  tcp     nowait  root    /usr/sbin/tcpd  in.proftpd -l -a
```

無効化されていなかった場合、先頭に#を付けて保存し、以下のコマンドを実行。

```
/etc/rc.d/init.d/inet restart
```

8.3 ProFTPD を起動

ネットワークが起動している状態で、以下のコマンドを実行して下さい。

```
/etc/rc.d/init.d/proftpd start
```

マシン起動時に自動的に起動するよう設定
Vine を起動した際に ProFTPD が自動的に起動するように設定する場合、以下のコマンドを実行。

```
/sbin/chkconfig --add proftpd
```

ProFTPD は Vine 2.0 以前の wu-ftpd よりも様々な点で柔軟に設定ができ、セキュリティ面にも優れている高機能な FTP サーバです。最初は戸惑いがあるかも知れませんが、設定は Apache に似ていますので、わかりやすいでしょう。

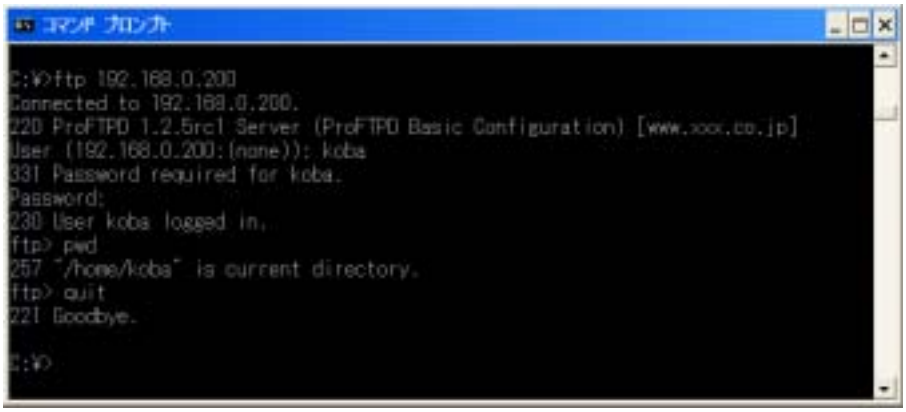
ProFTPD

ProFTPD は、unix 及び unix 系 OS の FTP デーモンです。ProFTPD は、GNU Public License(GPL)の下で開発、

リリース、配布が行なわれています。基本的に GPL は、ProFTPD パッケージか、あらかじめコンパイルされたバイナリ等を伴って、配布するサイト等から完全なソースコードでを入手できる限りは、要求に応じた改良、売買、ライセンスの許可ができるフリーのソフトウェアとして認めています。
・Apache の".htaccess"に類似する".ftpaccess"を使ったディレクトリ毎の設定
・マルチバーチャル FTP サーバと anonymous FTP サーバの簡単な設定

Production Version: 1.2.5rc1 Released: 12/19/2001
ftp://ftp.infoscience.co.jp/pub/proftpd/distrib/
　tar で圧縮された形式のデータをダウンロードしたのであれば、以下のコマンドで解凍してください。
tar -xzf proftpd-1.0.1.tar.gz
　"proftpd-1.0.1"というディレクトリ内に配布されたソースが展開されます。このディレクトリに移り、README ファイルを読んでください。次の作業として、configure というシェルスクリプトを実行すると、最後に以下を実行するだけです。
　make を実行してから、make install を、実行します。また、INSTALL テキストファイルには、インストールのより詳細な解説 が書かれています。
　バイナリが/usr/local/sbin に、標準の設定ファイルが/etc に、また補足的なバイナリ が/usr/local/bin にインストールされるはずです。デフォルトのこのディレクトリ設定を configure --prefix=[prefix-dir]と入力することで変更できます。 例えば、configure --prefix=/usr と入力すると、proftpd は/usr/sbin に、他のバイナリは/usr/bin にインストールされます。
　インストールが終了したら、/etc/proftpd.conf を状況に合わせて設定する必要があり 、またおそらく、inetd の設定を変更しなくてはなりません(proftpd を inetd モードで実行したいのであれば)。デフォルトの設定ファイルは、標準的な FTP サーバーには適していますが、バーチャルサーバーや、anonymous/guest アカウントの追加をするためには、proftpd.conf の編集が必要になります。

接続事例



8.4 FTP サーバーの簡易設定 proftpd

Vine Linux の起動よりテキストエジィタ(gedit)を選択。
proftpd の確認。
#rpm -q proftpd

proftpd の確認。
#rpm -q proftpd
proftpd-1.2.6-0vl1 これで確認出来ました。

(1)inetd.conf の修正
Vine Linux の起動よりテキストエディタ(gedit)を選択。inetd.conf の修正。テキストエディタが開きます。次に、ファイル開くを選択。inetd.conf の修正、テキストエディタで/etc/inetd.conf を選択。inetd.conf が開きます。
#ftp stream tcp nowait root /usr/sbin/tcpdin.proftpd
を選択し#をが付いていましたら外します。修正が終わりましたら保存。

(2)proftpd.conf の修正　：/etc/proftpd.conf
proftpd.conf が開きます。
ServerType standalone の箇所を探す。
ServerType　inetd とします。修正が終わりましたら保存。

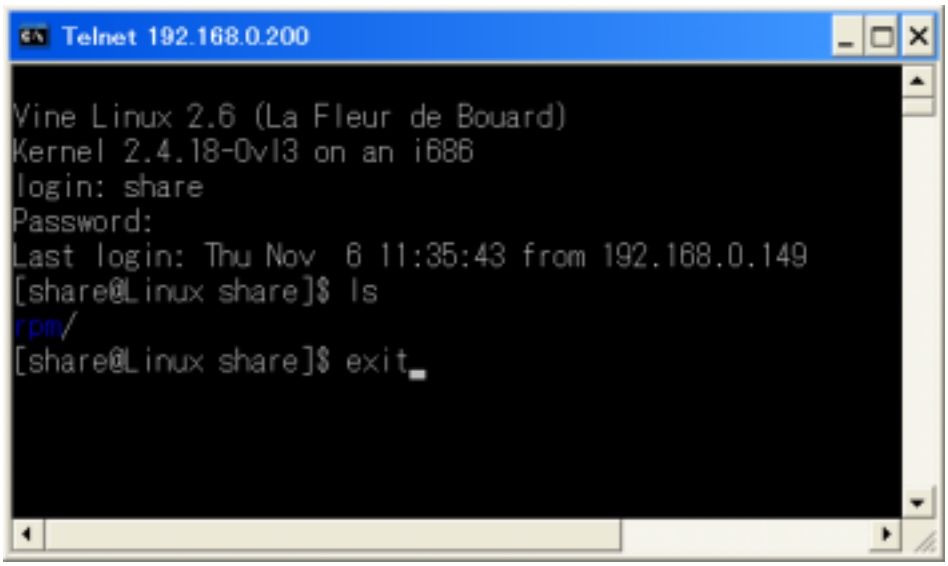
(3)inetd の再起動します。
/etc/rc.d/init.d/inetrestart と入力し、これで FTP サーバーを利用出来る。

(4)proftpd のシステムサービス起動
[/root]#setup
カーソルキー［↓］で[システムサービス設定]に移行し、[Tab]キーで[設定ツール]を実行に移行し、カーソルキー［↓］で httpd まで移行し、[Space]キー実行で[proftpd]にチェック[*]を入れる。[Tab]キーで完了に移行し、[Tab]キーで[終了]。

９．telnet の起動

Telnet で繋ぐには、Vine Linux が立ち上がっているのを確認して、Windows から Telnet を立ち上げ、接続を試みる。Host Name として 192.168.0.200 を打ち込めば、確かに見覚えのある Linux のコンソールが開く。

ユーザ名とパスワードを打ち込めば、後はもう仮想コンソール上で、いつも通りの作業を行えます。試しに ls と打てば、ホームディレクトリのファイル一覧が表示されます。悪くないです。但し、root では、入れないように工夫しないとハッカーにいたずらされます。



TELNET を停止

サーバのリモートメンテナンスは、SSH を使いますので、TELNET は停止させます。

１．/etc/inetd.conf を編集

telnet の行を探して、先頭に "#" つけてコメントアウトします。

```
# telnet stream    tcp    nowait    root    /usr/sbin/tcpd    in.telnetd
```

２．設定を有効にするため inetd を再起動します。

```
# /etc/rc.d/init.d/inet restart
```

root のリモートログインの禁止

リモートでサーバのメンテナンスを行うときは、ssh を使います。しかし、ssh はデフォルトの状態で root アカウントのログインを許可している。セキュリティにこだわった場合、これは好ましくありません。そこで、リモートでは、root アカウントでのログインできないようにする。

１．/etc/ssh/sshd_config をテキストエディタで開きます。

38 行目あたりで、次のようになっているところを、

```
PermitRootLogin yes
```

下のように変更。

```
PermitRootLogin no
```

２．設定を有効にするため ssh を reload

```
# /etc/rc.d/init.d/sshd reload
```

10．おまけ

10.1 画面キャプチャー

Windows だと、Print Screen キーを叩くだけですが、X ではどうするのかというとターミナルを起動し、コマンドを入力します。

xwd -root -out ファイル名

そうすると XWD 形式で保存されます。XWD 形式から、jpeg や png に変換

RedHat7.3 だと、標準で ImageMagic と言う便利なツールがインストールされています。確認は以下のコマンドで可能です。

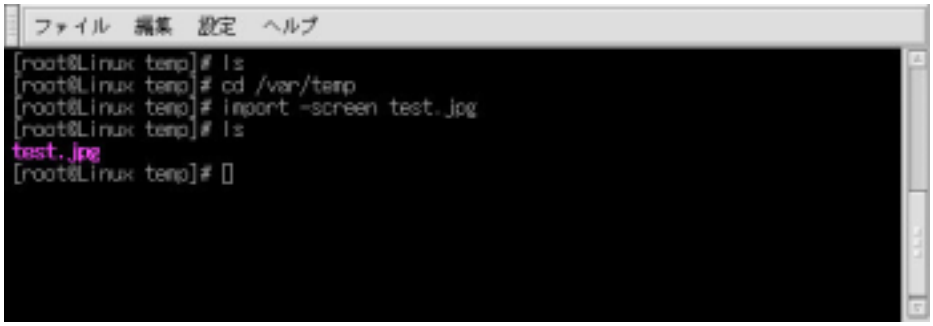
import -?

インストールされていれば使い方が表示されます。

画面全体をキャプチャーするには、以下のコマンドを入力します。

import -screen ファイル名.jpg

キャプチャーしたいウィンドウをクリックすると、拡張子の形式で保存されます。.jpg、.png、.gif、.pct などが使える。TIFF は使えない。



参考までにキャプチャーした画像を確認したい場合、display コマンドを入力すると GUI ツールが起動します。

10.2 日本語を入力する

インストール前から選別は始まっています。日本語を使いたいのであれば、当然日本語パッケージを選んでいますよね？それ以外を選んだ人は、自力でなんとかするか、日本語版を再インストールして下さい。分からない人は、以下の操作をやってみれば分かります。

テキストエディタを起動して下さい。どれか分からない人は、とりあえず WW ブラウザの URL 記述欄でも良いです。ちなみに GNOME では、メインメニュー->プログラム->アプリケーション->gedit です。SHIFT キー＋スペースバー　を叩くと、画面上のどこかに“ あ ”と表示されます。ローマ字で入力し、スペースで変換されます。もう一度 SHIFT+SPACE で元に戻ります。