

## .Linux 作業メモ

Copyright (C) 2003-2006 Kiyoteru Onishi <info@kiyoproject.com>  
\$Id: memo.txt,v 0.1.71 2006/07/08 00:51:00 kiyo Exp \$

## .はじめに

この文章は、自宅サーバ (Turbolinux 7 Server) の構築メモである。  
HTML ファイル化も考えてはいるが、今のところテキストファイルで手軽に作業メモを取ることを優先している。

## .パーティション情報

```
hdd1 /          131,071MB
hda1 /boot      62MB
hda2 /home     12,739MB
hda3 swap      258MB
```

```
---- /etc/fstab ----
/dev/hdd1          /          ext3    defaults    1 1
/dev/hda1         /boot     ext3    defaults    1 2
/dev/hda2         /home    ext3    defaults    1 3
/dev/cdrom        /mnt/cdrom iso9660 noauto,owner,ro 0 0
/dev/fd0          /mnt/floppy auto    noauto,owner 0 0
none             /proc    proc    defaults    0 0
none             /dev/pts devpts  gid=5,mode=620 0 0
/dev/hda3        swap     swap    defaults    0 0
```

## .ユーザの追加と削除

### ..一般ユーザの追加

システムに新規ユーザを作成する作業は root ユーザでしかできない。まずは、root でユーザログインする。

```
linux login: root
Password:
```

一般ユーザを作成するためのコマンドは、useradd。

```
# useradd hoge
```

### ..パスワードの設定 (passwd)

パスワードの設定は passwd コマンドで行う。パスワードを設定したいユーザの名前を、passwd コマンドの後ろに記述する。

```
# passwd hoge
```

"New Password:" と表示されるので、hoge に設定したいパスワードを入力。  
パスワードを入力すると確認のため、"Retype new password:" と再入力を促されるので、もう一度入力する。

### ..ユーザの削除 (userdel)

システムに存在するユーザを削除するためには、userdel コマンドを使用する。この作業は root ユーザのみが実行できる。

```
# userdel hoge
```

また、ユーザのホームディレクトリも一緒に削除したい場合は、-r オプションを付ける。

```
# userdel -r hoge
```

## .ターミナルでの日本語表示

```
..ファイルのバックアップ
```

```
$ cp -a $HOME/.bash_profile $HOME/.bash_profile.ORG
```

```
..ファイルの編集
```

```
$ vi $HOME/.bash_profile
```

```
---- $HOME/.bash_profile -----
export LANG=ja_JP.ujis
-----
```

上記 1 行を追加して保存。

## .turbonetcfg が変更するファイル

turbonetcfg コマンドが変更するファイルは、「/etc/HOSTNAME」「/etc/resolv.conf」「/etc/hosts」「/etc/sysconfig/network」の 4 つのファイルである。コマンド実行前に、ファイルのバックアップをしておくこと。

```
$ su -
# cd /etc
# cp -a HOSTNAME HOSTNAME.ORG
# cp -a resolv.conf resolv.conf.ORG
# cp -a hosts hosts.ORG
# cd sysconfig
# cp -a network network.ORG
# cd
# turbonetcfg
```

## .Perl のシンボリック・リンク作成

```
# ln -s /usr/bin/perl /usr/local/bin/perl
```

## .FTP サービス

```
..自動起動の設定
```

proftpd の自動起動設定を確認。

```
$ su -
# chkconfig --list proftpd
proftpd          0:off  1:off  2:off  3:off  4:off  5:off  6:off
```

proftpd がランレベル 3,5 で自動起動するように設定。

```
# chkconfig --level 35 proftpd on
```

proftpd が自動起動するか確認。

```
# chkconfig --list proftpd
proftpd          0:off  1:off  2:off  3:on   4:off  5:on   6:off
```

proftpd 手動起動。

```
# service proftpd start
Starting proftpd:                                OK
```

## .Apache の自動起動

```
..ディレクトリのバックアップ
```

```
$ su -
```

```
# cd /var
# cp -ra www www.ORG
```

## ..自動起動の設定

httpd が起動していないかを確認する。

```
# chkconfig --list httpd
httpd          0:off  1:off  2:off  3:off  4:off  5:off  6:off
```

httpd が起動していないので、ランレベル 3,5 で自動起動するように設定。

```
# chkconfig --level 35 httpd on
```

httpd が自動起動するか確認。

```
# chkconfig --list httpd
httpd          0:off  1:off  2:off  3:on   4:off  5:on   6:off
```

httpd 手動起動。

```
# service httpd start
Starting httpd:                                OK
```

## .J2SDK1.4.2\_11 のインストール

### ..インストール

<http://java.sun.com/j2se/1.4.2/ja/download.html> から RPM パッケージをダウンロード。

あらかじめ、/tmp に「j2sdk-1\_4\_2\_11-linux-i586-rpm.bin」ファイルをコピーしておく。

```
$ cd /tmp
$ chmod a+x j2sdk-1_4_2_11-linux-i586-rpm.bin
$ ./j2sdk-1_4_2_11-linux-i586-rpm.bin
```

ライセンス条項が表示されるので、最後まで目を通したら「y」と入力し [Enter] を押す。

```
$ su
# rpm -ihv j2sdk-1_4_2_11-linux-i586.rpm
# exit
```

J2SDK は「/usr/java/j2sdk1.4.2\_11」にインストールされる。

### ..環境変数の設定

```
$ cd
$ vi .bash_profile

export JAVA_HOME=/usr/java/j2sdk1.4.2_11
export PATH=$PATH:$JAVA_HOME/bin
```

上記 2 行を追加して保存。

## .Tomcat のインストール

### ..インストール

Apache プロジェクトより「jakarta-tomcat-4.1.31-LE-jdk14.tar.gz」ファイルを入手する。

<http://archive.apache.org/dist/tomcat/tomcat-4/>

あらかじめ、/tmp に「jakarta-tomcat-4.1.31-LE-jdk14.tar.gz」ファイルをコピーしておく。

```
$ cd /tmp
$ su
```

```
# tar xzvf jakarta-tomcat-4.1.31-LE-jdk14.tar.gz -C /usr/local
```

## ..環境変数の設定

```
# cd /usr/local
# ln -s jakarta-tomcat-4.1.31 tomcat
# cp -a /etc/profile /etc/profile.ORG
# vi /etc/profile
```

「/etc/profile」の先頭に次の 2 行を追加する。

```
export JAVA_HOME=/usr/java/j2sdk1.4.2_11
export CATALINA_HOME=/usr/local/tomcat
```

```
# exit
```

## .Tomcat の自動起動設定

自動起動ファイル「/etc/rc.d/init.d/tomcat」を自分で作成する。  
Apache と Tomcat が起動していたら停止させる。

```
$ su -
# /etc/rc.d/init.d/httpd stop
# /usr/local/tomcat/bin/shutdown.sh
```

次に Tomcat 自動起動ファイルを作成。

```
# vi /etc/rc.d/init.d/tomcat
```

```
---- /etc/rc.d/init.d/tomcat -----
#!/bin/sh
#
# chkconfig: 345 80 15
# description: Tomcat is a Servlet+JSP Engine.

export JAVA_HOME=/usr/java/j2sdk1.4.2_11
PATH=$PATH:$JAVA_HOME/bin
export CATALINA_HOME=/usr/local/tomcat

# Source function library.
. /etc/rc.d/init.d/functions

case "$1" in
'start')
if [ -f $CATALINA_HOME/bin/startup.sh ]; then
echo "Starting the tomcat service "
su - -c "$CATALINA_HOME/bin/startup.sh"
fi
;;
'stop')
echo "Stopping the tomcat service."
su - -c "$CATALINA_HOME/bin/shutdown.sh"
;;
'restart')
$0 stop
$0 start
;;
*)
echo "Usage: $0 start|stop|restart"
;;
esac
exit 0
```

Tomcat は Apache よりも必ず先に起動する必要があるので「# chkconfig:」行の、2番目のパラメータは注意する必要がある。  
この場合「80」となっており、標準の「/etc/rc.d/init.d/httpd」では「85」となっているため、Tomcat の方が先に起動する。

自動起動サービスに追加。

```
# cd /etc/rc.d/init.d
# chmod +x tomcat
# chkconfig --add tomcat
# chkconfig --list tomcat
tomcat          0:off  1:off  2:off  3:on   4:on   5:on   6:off
```

```
# chkconfig --level 4 tomcat off
# chkconfig --list tomcat
tomcat          0:off  1:off  2:off  3:on   4:off  5:on   6:off
```

最後に、Apache と Tomcat を起動。

```
# /etc/rc.d/init.d/tomcat start
# /etc/rc.d/init.d/httpd start
```

## .Apache 1.3.x と Tomcat 4.1.x の連携

### ..モジュールの入手

Apache プロジェクトより「mod\_jk」モジュールを入手する。

<http://archive.apache.org/dist/tomcat/tomcat-connectors/jk/>

mod\_jk.so-ap1.3.27-eapi-rh72 を入手。  
入手したモジュールを /usr/libexec/apache にコピーする。

```
$ su -
# cp mod_jk.so-ap1.3.27-eapi-rh72 /usr/libexec/apache/mod_jk.so
# cd /usr/libexec/apache
# chmod 755 mod_jk.so
```

### ..関連ファイルの設定

Apache と Tomcat が起動していたら停止させる。

```
# /etc/rc.d/init.d/httpd stop
# /etc/rc.d/init.d/tomcat stop
```

httpd.conf を編集する。  
config を分けた方が判りやすいので、最後に Include 行を追加。

```
# cd /etc/httpd/conf
# cp -a httpd.conf httpd.conf.ORG
# vi httpd.conf

---- /etc/httpd/conf/httpd.conf -----
Include /etc/httpd/conf/mod_jk.conf
-----
```

次に mod\_jk.conf を作成。

```
# vi mod_jk.conf

---- /etc/httpd/conf/mod_jk.conf -----
LoadModule jk_module /usr/libexec/apache/mod_jk.so

<IfModule mod_jk.c>
JkWorkersFile /etc/httpd/conf/workers.properties
JkLogFile /var/log/httpd/mod_jk.log
JkLogLevel info
JkMount /examples/* worker1
</IfModule>
-----
```

Apache よりのリクエストを受け取るワーカーを定義する workers.properties を作成する。

```
# vi workers.properties

---- /etc/httpd/conf/workers.properties -----
worker.list=worker1
worker.worker1.type=ajp13
worker.worker1.host=localhost
worker.worker1.port=8009
worker.worker1.lbfactor=1
-----
```

/usr/local/tomcat/conf/server.xml ファイルをバックアップ。

```
# cd /usr/local/tomcat/conf
# cp -a server.xml server.xml.ORG
```

/usr/local/tomcat/conf/server.xml を編集する。  
Coyote/JK2 AJP 1.3 Connector をコメントアウトして、AJP 1.3 Connector を代わりに有効にする。

```
# vi server.xml

---- /usr/local/tomcat/conf/server.xml -----
<!-- Define a Coyote/JK2 AJP 1.3 Connector on port 8009 -->
<!--
<Connector className="org.apache.coyote.tomcat4.CoyoteConnector"
    port="8009" minProcessors="5" maxProcessors="75"
    enableLookups="true" redirectPort="8443"
    acceptCount="10" debug="0" connectionTimeout="0"
    useURIValidationHack="false"
    protocolHandlerClassName="org.apache.jk.server.JkCoyoteHandler"/>
-->

<!-- Define an AJP 1.3 Connector on port 8009 -->
<Connector className="org.apache.ajp.tomcat4.Ajp13Connector"
    port="8009" minProcessors="5" maxProcessors="75"
    acceptCount="10" debug="0"/>
-----
```

## ..動作確認

各ファイルの準備ができたら、Tomcat Apache の順番で起動。

```
# /etc/rc.d/init.d/tomcat start
# /etc/rc.d/init.d/httpd start
```

以下の URL にアクセスして、サンプルアプリケーションが実行できれば成功。

```
http://サーバ名/examples/servlets/index.html
http://サーバ名/examples/jsp/index.html
```

## .PHP 4.2.3 のインストール

<http://www.turbolinux.co.jp/> より パッケージを入手する。

用意するパッケージ：  
ming-0.2a-3.i586.rpm  
sablotron-0.82-2.i586.rpm  
sablotron-devel-0.82-2.i586.rpm  
php-4.2.3-7.src.rpm

作業は root で実行。

```
# rpm -Uvh ming-0.2a-3.i586.rpm
# rpm -Uvh sablotron-*
# rpm -ivh php-4.2.3-7.src.rpm
# cd /var/src/rpm/SPECS/
# rpm -bb php.spec
# cd ../RPMS/i586
# ls -l
# rpm -Uvh --force php-*
```

最後に、Apache の再起動。

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

## .Apache + PHP の設定

/etc/httpd/conf/httpd.conf を編集。

```
AddType application/x-httpd-php .php
AddType application/x-httpd-php-source .phps
```

Apache の再起動。

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

## .SSL

## ..インストールパッケージの確認

```
$ su -
# rpm -qa | grep mod_ssl
mod_ssl-2.8.14-33
```

## ..証明書の作成

### ...サーバ鍵の作成

デフォルト鍵のバックアップ。

```
# cp -a /etc/httpd/conf/ssl/ssl.key/server.key /etc/httpd/conf/ssl/ssl.key/server.key.ORG
# cp -a /etc/httpd/conf/ssl/ssl.crt/server.crt /etc/httpd/conf/ssl/ssl.crt/server.crt.ORG
```

デフォルト鍵の削除。

```
# rm /etc/httpd/conf/ssl/ssl.key/server.key
# rm /etc/httpd/conf/ssl/ssl.crt/server.crt
```

鍵を作成し保存。

```
# openssl genrsa 1024 > /etc/httpd/conf/ssl/ssl.key/server.key
Generating RSA private key, 1024 bit long modulus
.....+++++
.....+++++
e is 65537 (0x10001)
```

鍵を root のみが読めるようにする。

```
# chmod 400 /etc/httpd/conf/ssl/ssl.key/server.key
```

以上でサーバ鍵の作成は終了。

### ...サーバ証明書の作成

```
# /usr/bin/openssl req -new -x509 -key /etc/httpd/conf/ssl/ssl.key/server.key -out /etc/httpd/conf/ssl/ss
l.crt/server.crt
Country Name (2 letter code) [AU]:JP <--- 国名をアルファベット 2 文字で設定
State or Province Name (full name) [Some-State]:Hoge <--- 都道府県名を設定
Locality Name (eg, city) []:Hoge City <--- 都市名を設定
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Hoge <--- 組織名を設定
Organizational Unit Name (eg, section) []:Webserver Team <--- 組織の部門名を設定
Common Name (eg, YOUR name) []:www.hoge.jp <--- DNS に登録されている FQDN を設定
Email Address []:hoge@hoge.jp <--- WWW サーバの管理者メールアドレスを設定

# chmod 400 /etc/httpd/conf/ssl/ssl.crt/server.crt
```

## ..SSL のテスト

ファイルのバックアップ。

```
# cp -a /etc/httpd/conf/httpd.bootopt /etc/httpd/conf/httpd.bootopt.ORG
```

TurboLinux では、Apache を SSL に対応したモードで起動するためには、起動スクリプト用のオプションファイル /etc/httpd/conf/httpd.bootopt を編集しなければならない。以下に示す行のコメントアウトを外す（有効にする）必要がある。

```
# vi /etc/httpd/conf/httpd.bootopt
----- /etc/httpd/conf/httpd.bootopt -----
HTTPDOPT=DSSL
-----
```

httpd を再起動。

```
# service httpd restart
Shutting down http: OK
Starting httpd: OK
```

次に WWW サーバへのアクセスをテスト。WWW サーバの IP アドレスが 192.168.1.1 の場合、以下の URL を参照。

https://192.168.1.1/

---

## .lame エンコーダのインストール

### ..ソースコードの入手とインストール

LAME プロジェクト (<http://www.mp3dev.org/>) から「Download the source code」というリンクを探し、表示されたページ中の「Click here for lame3.90.1.tar.gz and newer beta versions.」というリンクをクリック。  
表示されたページから、最新のファイルを見つけてダウンロード（今回の場合、lame-3.93.1.tar.gz が最新のファイル）。

ダウンロードしたファイルを作業ディレクトリで展開する。  
以下、作業ディレクトリを /tmp と仮定して説明。

```
$ cd /tmp
$ tar zxvf lame-3.93.1.tar.gz
```

lame-3.93.1 という名前のディレクトリが作成され、その中でソースコード一式が展開される。

lame-3.93.1 ディレクトリに移動して、ソースコードのコンパイルを実行。

```
$ cd lame-3.93.1
$ ./configure
$ make
```

コンパイルに成功したら、コマンドやライブラリをインストールする。

```
$ su
# make install
```

/usr/local/bin に「lame」コマンドファイル、/usr/local/lib に「libmp3lame.\*」のライブラリファイル群が存在していればインストールは完了。

---

## .Ogg Vorbis のインストール

### ..パッケージの入手

Ogg Vorbis の公式サイト (<http://www.vorbis.com/>) からパッケージをダウンロードする。

ダウンロードファイル：  
libogg-1.0-1.i386.rpm  
libvorbis-1.0-1.i386.rpm  
libogg-devel-1.0-1.i386.rpm  
libvorbis-devel-1.0-1.i386.rpm  
vorbis-tools-1.0.tar.gz

ダウンロードしたファイルを /tmp にコピー。

### ..既存パッケージのアンインストール

libogg-1.0rc1-2 と libvorbis-1.0rc1-2 をアンインストールする。  
libvorbis を先に削除しないと依存関係のエラーが出る。

```
$ su -
# rpm -e libvorbis
# rpm -e libogg
```

### ..パッケージのインストール

ダウンロードファイルがあるディレクトリに移動。

```
# rpm -ihv libogg-1.0-1.i386.rpm
# rpm -ihv libvorbis-1.0-1.i386.rpm
# rpm -ihv libogg-devel-1.0-1.i386.rpm
# rpm -ihv libvorbis-devel-1.0-1.i386.rpm
# exit
```

```
$ tar zxvf vorbis-tools-1.0.tar.gz
$ cd vorbis-tools-1.0
$ ./configure
$ make
$ su
# make install
```

## .バイナリパッケージを作成する手順

TL7W 以上の環境において、ソースパッケージ (\*.src.rpm) からバイナリパッケージ (\*.i586.rpm, \*.noarch.rpm) を作成する。  
必ず root でログインして実行。

zabom-1.8.22-winter2002.src.rpm から作成する場合の例)

```
# rpm -ivh zabom-1.8.22-winter2002.src.rpm
# cd /var/src/rpm/SPECS
# rpm -bb zabom.spec
# cd ../RPMS/i586
# ls -l
# rpm -Uvh zabom-*
```

## .F-Prot Antivirus for Linux Workstations for home use のインストール

### ..パッケージの入手と導入

[http://www.f-prot.com/download/home\\_user/download\\_fplinux.html](http://www.f-prot.com/download/home_user/download_fplinux.html) から RPM  
パッケージをダウンロードする。

root でログインして rpm コマンドを実行。

```
# rpm -ivh fp-linux-ws.rpm
```

インストールが完了すると、最新のウイルス定義ファイルに自動で更新してくれる。  
また /usr/local/f-prot 以下にウイルススキャン用の f-prot コマンドやウイルス  
定義ファイルの更新スクリプト check-updates.pl などが用意される。

### ..ウイルス定義ファイルの更新

```
# /usr/local/f-prot/tools/check-updates.pl
```

最新のウイルス定義ファイルがある場合には、

```
*****
* F-Prot Antivirus Updater *
*****
```

```
There's a new version of:
"Document/Office/Macro viruses" signatures on the web.
Starting to download...
Download completed.
```

```
There's a new version of:
"Application/Script viruses and Trojans" signatures on the web.
Starting to download...
Download completed.
```

```
Preparing to install Application/Script viruses and Trojans signatures.
Application/Script viruses and Trojans signatures have successfully been installed.
```

```
Preparing to install Document/Office/Macro viruses signatures.
Document/Office/Macro viruses signatures have successfully been installed.
```

```
*****
* Update completed successfully. *
*****
```

上記のような表示が出れば、ウイルス定義ファイルの更新は終了。

## ..ウイルススキャンの実行

ウイルススキャンを実行するには、f-prot コマンドを使用する。  
f-prot コマンドの書式は次のとおり。

f-prot [<オプション>] <スキャン対象>

f-prot コマンドの主なオプションは以下のとおり。

-ai	ニューラルネットワークウイルススキャンの有効化。これは新しいウイルスを特定し、自動的にそれらを解析する
-append	レポートファイルに追加する
-archive	アーカイブファイルをスキャンする
-auto	ウイルスの自動削除
-collect	ウイルスを収集する
-delete	ウイルスを削除する
-dumb	ダンプスキャンの実行
-ext	ファイルだけにざっと目を通す
-follow	シンボリックリンクの追跡
-help	ヘルプ表示
-list	スキャンするファイルをすべて表示
-nobreak	ESC が押されても、スキャンを中止しない
-noheur	ヒューリスティックスキャンの無効化
-nosub	サブディレクトリのスキャンを行わない
-old	古い定義ファイルでもメッセージしない
-packed	圧縮ファイルを解凍して検索
-page	各ファイルスキャンの後、停止する
-rename	ウイルスファイルの名前を変更
-report=<file>	出力先ファイルの指定
-server	メールフィルタを実行する
-silent	出力結果の非表示
-type	タイプ別にファイルを選択（標準）
-virlist	ウイルス情報の表示
-virno	既知のウイルスを数える
-nomacro	マクロウイルスをスキャンしない
-onlymacro	マクロウイルスだけスキャンする
-removeall	全ドキュメントからマクロを削除
-saferemove	ウイルスと判断した場合、感染したファイルからマクロを削除

オプションを指定せず /home 以下をスキャンする場合は、次のように実行する。

```
# /usr/local/bin/f-prot /home/
```

感染していなければ、

```
No viruses or suspicious files/boot sectors were found.
```

と表示される。

## ..ウイルススキャンの自動化

2 つの Perl スクリプトを作成して、ウイルススキャンを自動化する。

```

---- /usr/local/f-prot/f-prot.pl -----
#!/usr/bin/perl
$LOGFILE = "/var/log/f-prot.log";
$CHECKPATH = "/var/spool/mail/";
$WARNIN_MESSAGE = <<WARNIN_MESSAGE;
Subject: You got Virus mail

You got Virus mail.
WARNIN_MESSAGE

    system("/usr/local/bin/f-prot $CHECKPATH >$LOGFILE");

    open(F0,$LOGFILE) || die ("Can't open $LOGFILE");

    while($line0 = <F0>) {
        if($line0 =~ /Infection:){
            $line0 =~ s/~/var~/spool~/mail~/;
            ($string0) = split(/-/, $line0);
            open(SENDMAIL, "| /usr/sbin/sendmail $string0");
            print SENDMAIL $WARNIN_MESSAGE;
            close(SENDMAIL);
        }
    }
}

exit 0

```

```

---- /usr/local/f-prot/f-prot-daily.pl -----
#!/usr/bin/perl
use strict;
my $CHECKPATH = "/";
my $ADMIN = "root%localhost";

    open(FPROT, "/usr/local/bin/f-prot -archive $CHECKPATH|");
    open(SENDMAIL, "| /usr/sbin/sendmail -t");
    print SENDMAIL "To: $ADMIN%n";
    print SENDMAIL "Subject: Daily virus report% n% n";
    print SENDMAIL <FPROT>;
    close(SENDMAIL);
    close(FPROT);

exit 0

```

```

# cd /usr/local/f-prot
# chmod 755 f-prot.pl
# chmod 755 f-prot-daily.pl
# crontab -e
05 * * * * /usr/local/f-prot/f-prot.pl
10 01 * * * /usr/local/f-prot/tools/check-updates.pl
15 01 * * * /usr/local/f-prot/f-prot-daily.pl

```

これで、毎時 5 分に f-prot.pl が、1 時 10 分にウイルス定義ファイルの更新が、1 時 15 分に f-prot-daily.pl スクリプトが起動する。

## .Text::CSV Perl モジュールのインストール

<http://search.cpan.org/dist/Text-CSV/> より Text::CSV の tar ボールを入手。

```

$ tar zxvf Text-CSV-0.01.tar.gz
$ cd Text-CSV-0.01
$ perl Makefile.PL
$ make
$ su
# make install

```

## .Namazu による全文検索システムの構築

**..インストール**

http://search.cpan.org/dist/File-MMagic/ より File::MMagic の tar ボールを入手。  
 http://www.namazu.org/ より Namazu の tar ボールを入手。  
 http://www.akaneiro.jp/Search-Namazu/ より Search::Namazu の tar ボールを入手。  
 http://kakasi.namazu.org/ より Kakasi の tar ボールを入手。  
 http://www.daionet.gr.jp/~knok/kakasi/ より Text::Kakasi の tar ボールを入手。

File::MMagic のインストール

```
$ tar zxvf File-MMagic-1.27.tar.gz
$ cd File-MMagic-1.27
$ perl Makefile.PL
$ make
$ su
# make install
# exit
```

Kakasi のインストール。

```
$ tar zxvf kakasi-2.3.4.tar.gz
$ cd kakasi-2.3.4
$ ./configure
$ make
$ su
# make install
# exit
```

Text::Kakasi のインストール。

```
$ tar zxvf Text-Kakasi-1.05.tar.gz
$ cd Text-Kakasi-1.05
$ perl Makefile.PL
$ make
$ su
# make install
# exit
```

Namazu のインストール。

```
$ tar zxvf namazu-2.0.16.tar.gz
$ cd namazu-2.0.16
$ ./configure
$ make
$ su
# make install
# install -c -m 755 /usr/local/libexec/namazu.cgi /var/www/cgi-bin
# exit
```

Search::Namazu のインストール。

```
$ tar zxvf Search-Namazu-209.tar.gz
$ cd Search-Namazu-209
$ perl Makefile.PL
$ make
$ su
# make install
# exit
```

インデックス用のディレクトリを作成

```
$ mkdir /var/www/hoge
```

**..インデックスの作成**

```
$ cd /var/www/hoge <--- インデックス・ディレクトリに移動
$ mknmz /var/www/html <--- 検索対象ディレクトリを指定して実行
```

インデックスの確認。

```
$ ls /var/www/hoge
```

**..Namazu コマンドによる検索**

インデックスが作成できたら、検索が実行できるかどうかテストする。  
 検索コマンドには namazu コマンドを使用する。

namazu コマンドの実行形式は以下のとおり。

namazu 検索キーワード インデックス・ディレクトリ

### ..mknmz の設定

/usr/local/etc/namazu/mknmzrc-sample ファイルをコピーして修正。

```
# su -
# cd /usr/local/etc/namazu
# cp mknmzrc-sample mknmzrc
# vi mknmzrc

----- /usr/local/etc/namazu/mknmzrc -----
$ADDRESS = 'info@kiyoproject.com'; <---- メールアドレスを設定
-----
```

### ..Apache の設定

Apache の設定ファイル「/etc/httpd/conf/httpd.conf」を編集する。  
Apache で検索 CGIプログラムの namazu.cgi を利用するには、以下の項目を設定する必要がある。

- 管理者の電子メール・アドレス
- サーバのホスト名とポート番号
- 各ユーザのホーム・ディレクトリに配置する公開ディレクトリの名前
- CGIプログラム用のディレクトリ

この他、標準設定の httpd.conf ファイルのままでは、日本語の Web ページが一部のブラウザで文字化けするという問題が発生する。そこで以下のように、httpd.conf を修正する。  
使用言語の優先順位「ja」が 9 番目になっているのを先頭に移動させる。

```
----- /etc/httpd/conf/httpd.conf -----
LanguagePriority ja en da nl et fr de el it kr no pl pt pt-br ru ltz ca es sv tw
-----
```

### ..パーミッションの変更

Web 経由で検索を行うには、ユーザが検索対象ファイルとインデックスを読み込めるようにしておかなければならない。  
そのためには、まず、検索対象ディレクトリとインデックス・ディレクトリのそれぞれのディレクトリで以下のように chmod コマンドを実行する。

```
$ chmod -R o+r *
```

ただし、これだけでは設定は十分ではない。各ユーザのホーム・ディレクトリは、そのユーザしか移動できないように設定されている。そのため、いくら下位のファイルにその他のユーザに対する読み出し権限が設定されていても、実際には読み込めない。  
この問題を解消するには、以下のように chmod コマンドを実行して、その他のユーザの実行権限 (o+x) をホーム・ディレクトリに設定すればよい。

```
$ chmod o+x /home/hoge
```

### ..検索用 CGI の設定

設定は、.namazurc というファイルに記述する。  
/usr/local/etc/namazu ディレクトリに namazurc-sample というサンプルファイルがあるので、これを /var/www/cgi-bin/ に「.namazurc」というファイル名でコピーする。

```
# cp /usr/local/etc/namazu/namazurc-sample /var/www/cgi-bin/.namazurc
# cd /var/www/cgi-bin
# cp -a .namazurc .namazurc.ORG
# vi .namazurc
```

コピーしたファイルの内容を修正。

修正例)

```
Index          /var/www/hoge <--- インデックス・ディレクトリ
Replace        /var/www/html/ http://URLアドレス/ <--- ファイル名と URL の
                                         変換規則
Lang           ja <--- 表示言語の指定 (日本語)
EmphasisTags  "<strong class=%"keyword%">" "</strong>" <--- マッチしたキー
                                         ワードの強調表示
```

## ..ブラウザからの検索

「http://URLアドレス/cgi-bin/namazu.cgi」にアクセスして検索画面が表示されるのを確認。  
検索キーワードを入力して「Search!」ボタンをクリックすれば、検索結果が表示される。

## ..定期的なインデックスの更新

cron で定期的に mknmz コマンドを実行させる。

毎日 2 時 10 分にインデックス更新する例)

```
$ crontab -e
10 2 * * * /usr/local/bin/mknmz -0 /var/www/hoge /var/www/html --indexing-lang=ja
```

# .Eclipse 2.0.2 のインストール

## ..Eclipse 本体 Zip ファイルの解凍

```
# unzip eclipse-SDK-2.0.2-linux-motif.zip -d /usr/local
```

## ..Eclipse ランゲージパックのインストール

```
# unzip eclipse-nls-SDK-2.0.x.zip -d /usr/local
```

## ..Eclipse の起動

```
# cd /usr/local/eclipse
# ./eclipse
```

## ..Tomcat プラグインのインストール

```
# unzip tomcatPluginV21.zip -d /usr/local/eclipse/plugins
```

# .マウスの追加

## ..ファイルのバックアップ

```
$ su -
# cp -a /etc/X11/XF86Config /etc/X11/XF86Config.ORG
```

## ..XF86Configの修正

/etc/X11/XF86Config 追加マウスの設定を追加記入する。  
すでに設定されている 1 つ目のマウス設定をコピー。

```
Section "InputDevice"
    Identifier      "Mouse0"
    Driver          "mouse"
    Option          "Protocol"          "IMPS/2"
    Option          "Device"            "/dev/input/mice"
    Option          "Emulate3Buttons"   "yes"
    #MouseName      "Generic - 3 Button Mouse (USB)"
    Option          "Buttons"           "5"
    Option          "ZAxisMapping"      "4 5"
EndSection
```

別のマウスを付けるには、上の部分を貼り付け後、書き換える。

```
Section "InputDevice"
    Identifier      "Mouse1"
    Driver          "mouse"
    Option          "Protocol"          "PS/2"
    Option          "Device"            "/dev/psaux"
```

```

Option      "Emulate3Buttons"      "yes"
#MouseName  "Microsoft - IntelliMouse (PS/2)"
Option      "Buttons"              "5"
Option      "ZAxisMapping"         "4 5"
EndSection

```

両方のマウスを同時に使えるようにするには、XF86Config にもう一カ所変更が必要。  
ServerLayout というセクションがあり、これを編集する。

```

----- 変更前 -----
Section "ServerLayout"
    Identifier      "Default Layout"
    Screen          0 "Screen0" 0 0
    InputDevice     "Mouse0"          "CorePointer"
    InputDevice     "Keyboard0"       "CoreKeyboard"
EndSection
-----
----- 変更後 -----
Section "ServerLayout"
    Identifier      "Default Layout"
    Screen          0 "Screen0" 0 0
    InputDevice     "Mouse0"          "CorePointer"
    InputDevice     "Mouse1"          "SendCoreEvents" <--- 追加
    InputDevice     "Keyboard0"       "CoreKeyboard"
EndSection
-----

```

## .Samba の自動起動

```
..ファイルのバックアップ
```

```
$ su -
# cp -a /etc/samba/smb.conf /etc/samba/smb.conf.ORG
```

```
..自動起動の設定
```

Samba が起動していないかを確認する。

```
# chkconfig --list smb
smb          0:off  1:off  2:off  3:off  4:off  5:off  6:off
```

Samba をランレベル 3,5 で自動起動するように設定。

```
# chkconfig --level 35 smb on
# chkconfig --list smb
smb          0:off  1:off  2:off  3:on   4:off  5:on   6:off
```

smb の手動起動。

```
# service smb start
Starting SMB services:                                OK
```

## .Samba ユーザの追加

```
# smbpasswd -a hoge
New SMB password: <--- hoge の Samba のためのパスワードを入力
Retype new SMB password: <--- 再度入力
Added user hoge.
```

## .Linux ユーザを一括で Samba ユーザに持ってくる

```
$ su -
# cp -a /etc/samba/smbpasswd /etc/samba/smbpasswd.ORG
# cat /etc/passwd | mksmbpasswd.sh > /etc/samba/smbpasswd
# chown root.root /etc/samba/smbpasswd
# chmod 600 /etc/samba/smbpasswd
```

## .Samba の設定

## ..[global]セクション

標準では設定ファイル /etc/samba/smb.conf に記述されていないが、ワークグループ名あるいは NT ドメイン名を設定するための workgroup を samba.conf の [global] セクションに記述するのが普通である。workgroup の規定値は WORKGROUP で、標準のままにしておく、このワークグループあるいはドメインにコンピュータが表示される。workgroup は実際のネットワークに合わせて書き換えること。

```
---- /etc/samba/smb.conf -----
workgroup = <group>
-----
```

Samba におけるデフォルトの設定ファイルは /usr/local/samba/lib/smb.conf だが、TurboLinux では /etc/samba/ に置かれるようにカスタマイズされている。

## .SWAT による Samba の設定方法

### ..SWAT の起動設定

SWAT は標準では利用できない設定になっているので、設定を確認。

```
# chkconfig --list | grep swat
swat:    off
```

chkconfig コマンドで自動起動の設定を行う。

```
# chkconfig swat on
# chkconfig --list | grep swat
swat:    on
```

標準では tcp\_wappers と呼ばれるアクセス制御のツールでアクセス制限がされているため SWAT にアクセスできない。制限をはずすためには /etc/hosts.allow ファイルを編集する。

修正前に /etc/hosts.allow ファイルをバックアップ。

```
# cd /etc
# cp -a hosts.allow hosts.allow.ORG
# vi hosts.allow
```

hosts.allow ファイルの最終行に以下の文を追加。

```
swat : ALL
```

全ての端末から SWAT にアクセス許可する設定。セキュリティ上問題があれば ALL の代わりに特定のパソコンの IP アドレス (192.168.1.100 など) を記述する。

xinetd を再起動。

```
# /etc/rc.d/init.d/xinetd restart
Stopping INET services:                OK
Starting INET services:                OK
```

### ..SWAT の実行

1. Windows パソコンで Web ブラウザを起動して、アドレスに http://IPアドレス:901/ を入れてサーバに接続。
2. ネットワークパスワードの入力画面が表示されるので、ユーザ名 root パスワードに root のパスワードを入力して【OK】をクリック。
3. SWAT の設定画面が表示される。

## .プリンタの設定

### ..プリンタドライバのダウンロード

http://www.avasys.jp/linux/dl\_ink.html から EPSON PM-950C 用のプリンタドライバ「pips950-2.5.2-2.i386.rpm」を /tmp にダウンロード。

## ..ファイルのバックアップ

```
$ su -
# cd /etc
# cp -a printcap printcap.ORG
# cp -a services services.ORG
```

## ..プリンタドライバのインストール

```
# cd /tmp
# rpm -ihv pips950-2.5.2-2.i386.rpm
Preparing... ##### [100%]
 1:pips950 ##### [100%]
Install Message > Described entry of pm950c in printcap.
Install Message > Backup file is /etc/printcap.bak.
Install Message > Described entry of pips950 in services.
Install Message > Backup file is /etc/services.bak.
Install Message > Start /usr/local/EPKowa/PM950C/setup to change setup.
Install Message > Please restart lpd.
```

## ..ネットワークプリンタのポート設定

### ...手作業でプリンタの設定をする

```
# cp -a /etc/printcap /etc/printcap.pm950c.ORG
# vi /etc/printcap

---- /etc/printcap -----
# /etc/printcap
#
# Please don't edit this file directly unless you know what you are doing!
# Be warned that the control-panel printtool requires a very strict format!
# Look at the printcap(5) man page for more info.
#
# This file can be edited with the printtool in the control-panel.

# written for pips950
pm950c:¥
    :sd=/var/spool/lpd/pm950c:¥
    :mx#0:¥
    :sh:¥
    :rm=192.168.1.253:¥
    :rp=lp:¥
    :if=/usr/local/EPKowa/PM950C/filter950:
# pips950 end

-----

# cp -a /etc/ekpdrc /etc/ekpdrc.ORG
# vi /etc/ekpdrc

---- /etc/ekpdrc -----
PrinterName = pm950c
PrinterDevicePath = lp
DummyDevicePath = /var/ekpd/ekplp0
CommandServerPort = 35586
-----
```

### ...printconfig を利用してプリンタの詳細設定をする

設定ファイルの修正が終わった後で、lpd を起動。

```
# /etc/rc.d/init.d/lpd start
Starting lpd: OK
```

printconfig コマンドを実行する。

```
# printconfig
```

printconfig を起動したら【Edit】を選択。プリンタの設定画面が表示される。これよりプリンタの詳細設定を行う。中段の各設定項目の【<Configure...>】で必要な項目を設定していく。必要な設定項目としては「Printer Type (プリンタの種類)」・「Resolution (解像度:720x720)」・「Paper Size (用紙サイズ:a4)」となる。「Color Depth (色選択)」はモノクロプリンタの際は変更する必要がないが、今回はカラープリンタなので「Color Depth (色選択)」の【<Configure...>】を選択、カラー

印刷を指定した。

最新のプリンタドライバは存在しないので、今回はエプソン MJ シリーズのプリンタドライバ (EPSON MJ-500C/800C/3000C/510C) を代替で利用した。

printconfig での設定保存が終了したら、最後に /etc/printcap を手で整形する。

```
# vi /etc/printcap
```

```
----- /etc/printcap -----
#
# /etc/printcap - Printer definition file.
# Generated by printconfig v1.2.4
# Do not edit by hand unless you know what you
# are doing. The file format is very important
# and needs to be strictly maintained. See the
# manpage for printcap(5) for more information
# on the format of this file.
#
##PRINTT00L3## REMOTE mj500c 720x720 a4 {} EPSONMJ500C 32 {}
# written for pips950
pm950c:¥
        :sh:¥
        :sd=/var/spool/lpd/pm950c:¥
        :mx#0:¥
        :rm=192.168.1.253:¥
        :rp=lp:¥
        :lpd_bounce:¥
        :if=/usr/local/EPKowa/PM950C/filter950:
# pips950 end
-----
```

設定ファイルの整形が終わった後で、lpd を再起動。

```
# /etc/rc.d/init.d/lpd restart
Shutting down lpd:
Starting lpd: OK
OK
```

## ..lpd 自動起動の設定

lpd の自動起動設定を確認。

```
$ su -
# chkconfig --list lpd
lpd          0:off  1:off  2:off  3:off  4:off  5:off  6:off
```

lpd がランレベル 3,5 で自動起動するように設定。

```
# chkconfig --level 35 lpd on
```

lpd が自動起動するか確認。

```
# chkconfig --list lpd
lpd          0:off  1:off  2:off  3:on   4:off  5:on   6:off
```

lpd 手動起動。

```
# /etc/rc.d/init.d/lpd start
Starting lpd: OK
```

## .DNS サービス

### ..ファイルのバックアップ

```
$ su -
# cp -a /etc/named.conf /etc/named.conf.ORG
```

### ..初期設定ファイル

/etc/named.conf ファイルの修正。

```
----- /etc/named.conf -----
```

```

options {
    directory "/var/named";
    /*
     * If there is a firewall between you and nameservers you want
     * to talk to, you might need to uncomment the query-source
     * directive below. Previous versions of BIND always asked
     * questions using port 53, but BIND 8.1 uses an unprivileged
     * port by default.
     */
    // query-source address * port 53;
    forward first;
    forwarders {
        202.238.95.24;
        202.238.95.26;
    };
};

//
// a caching only nameserver config
//
zone "." {
    type hint;
    file "named.ca";
};

zone "0.0.127.in-addr.arpa" {
    type master;
    file "named.local";
};

zone "k2.kiyoproject.com" IN {
    type master;
    file "k2.kiyoproject.com.db";
    allow-update { none; };
};

zone "1.168.192.in-addr.arpa" IN {
    type master;
    file "1.168.192.in-addr.arpa.db";
    allow-update { none; };
};

```

### ..ディレクトリのバックアップ

```

# cd /var
# cp -ra named named.ORG

```

### ..正引きゾーンマスターファイル

正引きゾーンマスターファイルを作成。

```

---- /var/named/k2.kiyoproject.com.db -----
$TTL 86400
@      IN      SOA    red.k2.kiyoproject.com. root.red.k2.kiyoproject.com. (
                                2006060601 ; Serial
                                3600       ; Refresh
                                300        ; Retry
                                360000    ; Expire
                                86400     ) ; Minimum

      IN      NS     red.k2.kiyoproject.com.

      IN      MX     10 mail.k2.kiyoproject.com.
      IN      MX     20 hogella.homeip.net.

localhost      IN      A       127.0.0.1

red             IN      A       192.168.1.1
crusoe         IN      A       192.168.1.2
vaio           IN      A       192.168.1.10

www            IN      CNAME    red
mail           IN      CNAME    red
ftp            IN      CNAME    red
hogella       IN      CNAME    red

```

## ..逆引きゾーンマスターファイル

逆引きゾーンマスターファイルを作成。

```
----- /var/named/1.168.192.in-addr.arpa.db -----
$TTL 86400
@      IN      SOA      red.k2.kiyoproject.com. root.red.k2.kiyoproject.com. (
                                2006060601 ; Serial
                                3600      ; Refresh
                                300      ; Retry
                                360000   ; Expire
                                86400   ) ; Minimum

      IN      NS       red.k2.kiyoproject.com.

      IN      PTR      k2.kiyoproject.com.
      IN      A        255.255.255.0

1     IN      PTR      red.k2.kiyoproject.com.
2     IN      PTR      crusoe.k2.kiyoproject.com.
10    IN      PTR      vaio.k2.kiyoproject.com.
```

## .named の起動

```
# /etc/rc.d/init.d/named start
```

または、service named start を実行する。

また Linux 起動時に自動的に立ち上がるように、以下のコマンドを実行。

```
# chkconfig named on
```

なお named が起動した状態で /etc/named.conf ファイルや /var/named ディレクトリ以下のゾーンデータベースを変更しても、それだけでは設定が反映されない。設定を反映するには、以下のコマンドを実行する必要がある。

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

または、service named restart を実行する。

## .Dynamic DNS の更新

### ..DiCE 0.19 の導入

あらかじめ、/tmp に「diced01913.tar.gz」「xdice017.tar.gz」「libqtintf.so.gz」ファイルをコピーしておく。

ファイル入手先：[http://www.hi-ho.ne.jp/yoshihiro\\_e/dice/linux.html](http://www.hi-ho.ne.jp/yoshihiro_e/dice/linux.html)

```
$ su
# tar zxvf diced01913.tar.gz -C /usr/local
# tar zxvf xdice017.tar.gz -C /usr/local
# gzip -dv libqtintf.so.gz
# cp libqtintf.so /usr/lib/
# ldconfig
# cd /usr/local/DiCE
# ./dice
```

DiCE が起動したら、Dynamic DNS サービスの設定を行う。  
起動コマンドオプションは、「./diced -h」で確認できる。

### ..DiCE の自動起動設定

```
# cp -a /etc/rc.d/rc.local /etc/rc.d/rc.local.ORG
# vi /etc/rc.d/rc.local
```

「rc.local」の最下行あたりに実行スクリプトを記述。

```
----- /etc/rc.d/rc.local -----
```

```

    if [ -x /usr/local/DiCE/diced ]; then
        /usr/local/DiCE/diced -d -l
    fi
fi <----- (最下行)
-----

```

ファイルを保存し、システムを再起動する。

```
# /sbin/shutdown -r now
```

再起動後、DiCE が稼働しているかを確認。

```

$ su
# ps ax | grep diced
  714 ?      S        0:00 /usr/local/DiCE/diced -d -l
14480 pts/0  S        0:00 grep diced

```

..DiCE の停止

```

$ su
# ps ax | grep diced
  714 ?      S        0:00 /usr/local/DiCE/diced -d -l
14480 pts/0  S        0:00 grep diced
# kill 714

```

## . メールサーバ

..POP3・IMAP4 サーバを構築する

POP3 と IMAP4 の起動状態を確認。

```

# chkconfig --list pop-3
pop-3          off
# chkconfig --list imap
imap          off

```

POP3 も IMAP4 も起動されていないので、xinetd 経由で起動するように設定。

```

# chkconfig pop-3 on
# chkconfig imap on

```

再度、設定状態を確認。

```

# chkconfig --list pop-3
pop-3          on
# chkconfig --list imap
imap          on
# /etc/rc.d/init.d/xinetd restart
Stopping INET services:
Starting INET services:

```

OK  
OK

# service xinetd restart でも再起動できる。

次に、/etc/hosts.allow に 192.168.1.0/24 のネットワークからのみ POP3 サービスへのアクセスを許可する記述を行う。

vi で /etc/hosts.allow を修正。

```
# vi /etc/hosts.allow
```

最終行に、『ipop3d : 192.168.1.』を追記。

## . sendmail の設定

..他のサーバにメールを送信できるようにする

sendmail のデフォルトの設定ではメールリレーが許可されていないため、自サーバ上のユーザ以外にメールを送ることができない。そこで、ネットワーク単位で許可する設定を行う。リレーを許可する一覧は /etc/mail/access ファイルに記述する。

..ファイルのバックアップ

/etc/mail/access、 /etc/mail/access.db と /etc/mail/local-host-names ファイルをバックアップ。

```
$ su -
# cd /etc/mail
# cp -a access access.ORG
# cp -a access.db access.db.ORG
# cp -a local-host-names local-host-names.ORG
# vi access
```

### .. 転送制限の解除

/etc/mail/access ファイルの一番最後に設定を追記する。

```
---- /etc/mail/access -----
# by default we allow relaying from localhost...
localhost          RELAY
127.0.0.1          RELAY
192.168.1          RELAY <-- 追加部分
-----
```

/etc/mail/access ファイルから、sendmail が実際に使用する /etc/mail/access.db ファイルを作成する。ファイル保存後、以下のコマンドを実行。

```
# makemap hash /etc/mail/access.db < /etc/mail/access
```

これで /etc/mail/access.db ファイルがアップデートされ、192.168.1.0/255.255.255.0 ネットワーク上の端末から、メールリレーが可能になる。なお /etc/mail/access.db ファイルのアップデート時には、sendmail のリロードは必要ない。

/etc/mail/access の設定を変更した場合には、/etc/mail/access.db という DBM 形式のファイルに変換する必要があるが、TurboLinux ではサービス制御スクリプトを使って、sendmail を restart すると自動的にこの変換が行われる。

### .. 接続制限の解除

TurboLinux の sendmail は、TCP Wrapper による接続制限が行われるようになっている。メールを受信する場合には、明示的に接続制限を解除する必要がある。/etc/hosts.allow ファイルの最終行に、以下の記述を追加する必要がある。

```
---- /etc/hosts.allow -----
sendmail : ALL
-----
```

### .. 複数のドメイン名でメールを受信できるようにする

```
# vi local-host-names
```

/etc/mail/local-host-names ファイルの中身はデフォルトでは何も記述されていない。以下の記述を行う。

```
---- /etc/mail/local-host-names -----
k2.kiyoproject.com
hogella.homeip.net
-----
```

ファイルを保存後、設定を有効にするために sendmail をリロードさせる。

```
# /etc/rc.d/init.d/sendmail reload
Shutting down sendmail:          OK
Starting sendmail:              OK
```

または、service sendmail restart を実行。

これで hoge@red.k2.kiyoproject.com hoge@k2.kiyoproject.com hoge@hogella.homeip.net のメールアドレスでメールが届いても、hoge のメールボックスに配送されるようになる。

### .. 自動起動の設定

sendmail がランレベル 3,5 で自動起動するように設定。

```
# chkconfig --level 35 sendmail on
```

sendmail が自動起動するか確認。

```
# chkconfig --list sendmail
sendmail      0:off  1:off  2:off  3:on   4:off  5:on   6:off
```

## ===== .FFFTP で「501 NLST: Options not supported」と表示され接続できない問題

### .. 状況説明

Web サーバのアップロードなどに FFFTP を用いている場合、Web サーバ側が FTP のサーバソフトとして proftpd を用いていると、以前は問題なく接続できていたにも関わらず、突然「501 NLST: Options not supported」と表示されて、接続できなくなる場合がある。

これは、以下の 2 つの理由で生じたものだ。

1. リモートのファイルリストの取得に利用されている NLST コマンドに、従来の proftpd (1.2.9 以前) では規格 (RFC959) で許されていないオプション指定を許していた一方、新バージョン (1.2.10 以降) では規格に添った形にするために、このオプション指定が不可能とされた
2. FFFTP が標準設定で NLST のオプション指定を利用していた

### .. 回避法

これは FFFTP の設定変更で回避可能である。まず、FFFTP 起動時のホスト一覧画面で、問題のホストを選択し、「設定変更」ボタンをクリック。すると、「ホストの設定」とタイトルのついたウィンドウが表示され、基本設定が見えるはず。

この画面中の、上部のタブ選択部分で、「高度」を選択。標準では「NLST -R を使って高速に再帰検索」が有効になっている。

まず、この「NLST -R を使って高速に再帰検索」のチェックを外した後、その上にある「LIST コマンドでファイル一覧を取得」にチェックする。あとは「OK」ボタンで設定を終了すれば、次回から proftpd 1.2.10 以降のサーバにも問題なく接続できるはずである。

現在のところ、このような振る舞いをするのが判明しているのは FFFTP のみだが、他のクライアントでも同様のエラーがでる場合は、類似の対策を用いることが可能であると考えられる。