# Apache + PHP + MySQL

## (en utilisant les packages de la Mandriva LE2005)

**Olivier Hoarau** (olivier.hoarau@funix.org)

V1.8, 30 septembre 2005

# 1 Historique du document

30.09.05 V1.8 Passage à Mandriva LE2005

05.11.04 V1.7 Passage à la Mandrake 10.0 avec **Apache 2.0.48**, **PHP 4.3.4** et **MySQL 4.0.18**

04.05.03 V1.6 Passage à la Mandrake 9.1 avec **Apache 2.0.44**, **PHP 4.3.1**, **MySQL 4.0.11a** et **phpMyAdmin 2.4.0**

24.12.02 V1.5 Passage à la Mandrake 9.0 avec **Apache 1.3.26**, **PHP 4.2.3**, **MySQL 2.23.52** et **phpMyAdmin 2.3.2**

09.06.02 V1.4 Passage à la Mandrake 8.2 avec **Apache 1.3.23**, **PHP 4.1.2**, **MySQL 2.23.47**, et **phpMyAdmin 2.2.5**

10.3.02 V1.3 Passage à la Mandrake 8.1

3.12.00 V1.2 Rajout du paragraphe **PHP** et **LDAP** et mise à jour **webalizer** (v2.01.06)

22.10.00 V1.1 Passage à Mandrake 7.2 avec **Apache 1.3.1**4, **PHP 4.0.3pl**1, **MySQL 3.23.23** et **phpMyAdmin 2.1.0**

25.7.00 V1.0 Création du document

# 2 Préambule

Ce document présente l'installation et la configuration d'**Apache** sur une distribution Linux Mandrake avec gestion **PHP** et **MySQL** en utilisant les packages standards de la Mandrake.
La dernière version de ce document est téléchargeable à l'URL http://www.funix.org. Ce document peut être reproduit et distribué librement dès lors qu'il n'est pas modifié et qu'il soit toujours fait mention de son origine et de son auteur, si vous avez l'intention de le modifier ou d'y apporter des rajouts, contactez l'auteur pour en faire profiter tout le monde.
Ce document ne peut pas être utilisé dans un but commercial sans le consentement de son auteur. Ce document vous est fourni "dans l'état" sans aucune garantie de toute sorte, l'auteur

ne saurait être tenu responsable des quelconques misères qui pourraient vous arriver lors des manipulations décrites dans ce document.

# 3 Présentation

Cette page n'a pas pour vocation de vous présenter l'installation d'un serveur **Apache** destiné à être vu du monde entier, mais plutôt l'installation d'un serveur **Apache** sur un réseau local en intranet, afin surtout de se familiariser au langage **PHP** et au SGBD **MySQL** et éventuellement à **LDAP**.
On utilisera les packages standards de la Mandriva LE2005.

# 4 Installation de MySQL

## 4.1 Installation

Sur une Mandriva LE2005 vous trouverez la version **MySQL-4.1.11-1mdk**.

Voici les différents packages à installer sur la machine **obelix** (un **urpmi** du package cité plus haut devrait suffire pour gérer les dépendances).

**MySQL-4.1.11-1mdk.i586**
**MySQL-client-4.1.11-1mdk.i586**
**MySQL-common-4.1.11-1mdk.i586**
**perl-DBD-mysql-2.9004-6mdk.i586**
**perl-DBI-1.47-1mdk.i586**

Pour le faire fonctionner avec **Apache** et **PHP** on installera aussi

**php-mysql-4.3.10-7mdk**

Cela va vous créer un ensemble d'exécutables sous **/usr/sbin** (ceux réservés à root) et sous **/usr/bin** (pour tous les utilisateurs), par ailleurs un répertoire **/var/lib/mysql** va être créé. Le fichier de démarrage du daemon **MySQL**, **mysql** (droit 755) se trouve sous **/etc/rc.d/init.d**, par défaut le serveur est lancé à l'état de marche 2, 3, 4 et 5.

L'installation par **rpm** n'a pas lancé le daemon il faudra le faire manuellement en tapant

**/etc/rc.d/init.d/mysql start**

**NOTE** Vous noterez que si vous faites un:

**ps aux | grep mysqld**

L'utilisateur **mysql** est proprio du daemon **mysqld**. Vous vous rendrez comte que même si c'est **root** qui lance le **daemon** aussitôt après c'est l'utilisateur **mysql** qui en devient le proprio, ainsi si jamais il y avait une faille de sécurité dans **MySQL**, le hacker ne se retrouverait pas **root** mais **mysql** avec les droits qui vont avec.

## 4.2 Mise en place des utilisateurs

La première chose à faire est de mettre un mot de passe pour **root** pour l'accès à l'administration des bases de données. La commande à taper est:
**mysqladmin -u root password 'mot-de-passe'**

Le mot de passe peut être différent de celui du login.

Maintenant on va créer un compte utilisateur (**olivier** dans mon exemple), pour cela on doit se connecter en tant qu'administrateur de la base à la base de donnée **mysql** contenant les infos sur les utilisateurs et leurs droits.

**mysql -u root -p mysql**
**Enter password:**

**Reading table information for completion of table and column names**
**You can turn off this feature to get a quicker startup with -A**

**Welcome to the MySQL monitor.  Commands end with ; or \g.**
**Your MySQL connection id is 4 to server version: 4.1.13**

**Type 'help;' or '\h' for help. Type '\c' to clear the buffer.**

**mysql>**

A présent on va entrer l'utilisateur **olivier** qui sera un super utilisateur avec les mêmes droits que root:

```
mysql>  INSERT INTO user
    ->  VALUES('localhost','olivier',PASSWORD('mot-de-passe'),
     ->   'Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','  ','
```

**',' ',' ','0','0','0');**
**Query OK, 1 row affected (0.00 sec)**

**NOTE** Il n'est pas obligatoire de rentrer le login pour le nom d'utilisateur et le mot de passe de login.

Pour voir si la saisie s'est bien passée:

**mysql> SELECT * FROM user;**

```
+-----------------+---------+-----------------+-------------+-------------+-------------+-------------+-------------+-----------+-------------+---------------+-------------+-----------+-----------+-----------------+------------+------------+-------------
+-----------+----------------------+-----------------+-------------+---------------+-----------------+----------+-----------+-------------+---------------+-------------+------------+-----------------+
| Host            | User    | Password        | Select_priv | Insert_priv | Update_priv | Delete_priv | Create_priv | Drop_priv | Reload_priv | Shutdown_priv | Process_priv | File_priv | Grant_priv | References_priv |
Index_priv | Alter_priv | Show_db_priv | Super_priv | Create_tmp_table_priv | Lock_tables_priv | Execute_priv | Repl_slave_priv | Repl_client_priv | ssl_type | ssl_cipher | x509_issuer | x509_subject |
max_questions | max_updates | max_connections |
+-----------------+---------+-----------------+-------------+-------------+-------------+-------------+-------------+-----------+-------------+---------------+-------------+-----------+-----------+-----------------+------------+------------+-------------
+-----------+----------------------+-----------------+-------------+---------------+-----------------+----------+-----------+-------------+---------------+-------------+------------+-----------------+
| localhost       | root    | 6bf57a02084c345b | Y       | Y       | Y       | Y       | Y       | Y       | Y       | Y       | Y       | Y       | Y       | Y       | Y       | Y       | Y       | Y       | Y       | Y
| Y       | Y       | Y       |       |       |       |       |       0 |       0 |       0 |
| asterix.kervao.fr | root    |             | Y       | Y       | Y       | Y       | Y       | Y       | Y       | Y       | Y       | Y       | Y       | Y       | Y       | Y       | Y       | Y       | Y       | Y
| Y       | Y       | Y       |       |       |       |       |       0 |       0 |       0 |
| localhost       |       |             | N       | N       | N       | N       | N       | N       | N       | N       | N       | N       | N       | N       | N       | N       | N       | N       | N       | N
| N       | N       |       |       |       |       |       0 |       0 |       0 |
| asterix.kervao.fr |       |             | N       | N       | N       | N       | N       | N       | N       | N       | N       | N       | N       | N       | N       | N       | N       | N       | N       | N
| N       | N       | N       |       |       |       |       |       0 |       0 |       0 |
| localhost       | olivier | 588c797a754bd00a | Y       | Y       | Y       | Y       | Y       | Y       | Y       | Y       | Y       | Y       | Y       | Y       | Y       | Y       | Y       | Y       | Y       | Y
| Y       | Y       | Y       |       |       |       |       |       0 |       0 |       0 |
+-----------------+---------+-----------------+-------------+-------------+-------------+-------------+-------------+-----------+-------------+---------------+-------------+-----------+-----------+-----------------+------------+------------+-------------
+-----------+----------------------+-----------------+-------------+---------------+-----------------+----------+-----------+-------------+---------------+-------------+------------+-----------------+
```

**5 rows in set (0.00 sec)**

Maintenant pour prendre tout ça en compte.

**mysql> FLUSH PRIVILEGES;**
**Query OK, 0 rows affected (0.08 sec)**

Argh! on se rend compte qu'il y a une ligne avec un utilisateur root sans mot de passe, on va le supprimer

**mysql>DELETE FROM user WHERE Host="asterix.kervao.fr";**
**Query OK, 2 rows affected (0.04 sec);**

Pour quitter

**mysql>quit**

## 4.3 Création de tables

Maintenant notre utilisateur olivier va créer une table qui nous servira plus tard pour nos expérimentations avec **Apache**. Il doit d'abord se connecter:

**mysql -u olivier -p**
**Enter password:**
**Welcome to the MySQL monitor.  Commands end with ; or \g.**
**Your MySQL connection id is 5 to server version: 4.1.13**

**Type 'help;' or '\h' for help. Type '\c' to clear the buffer.**

**mysql>**

Pour voir la liste des bases de données disponibles, on tapera:

**mysql> SHOW DATABASES;**
**+------------+**
**| Database |**
**+------------+**
**| mysql      |**
**| test       |**
**+-----------+**
**4 rows in set (0.00 sec)**

On va maintenant créer une base de données **essai**:

```
mysql> CREATE DATABASE essai;
Query OK, 1 row affected (0.00 sec)
```

On va utiliser maintenant cette base de donnée

```
mysql> USE essai
Database changed
```

Comme la base vient d'être crée, elle ne contient aucune table, pour s'en convaincre il suffit de taper:

```
mysql> SHOW TABLES;
Empty set (0.00 sec)
```

Pour notre première exemple **Apache+PHP+MySQL**, on va créer la table suivante:

```
mysql> CREATE TABLE coord (
   -> nom VARCHAR(20),
   -> prenom VARCHAR(20),
   -> email VARCHAR(30)
   -> );
Query OK, 0 rows affected (0.03 sec)
```

Jetons un coup d'oeil maintenant sur les tables disponibles:

```
mysql>  SHOW TABLES;
+--------------------+
| Tables in essai    |
+--------------------+
| coord              |
+--------------------+
1 row in set (0.00 sec)
```

La table nouvellement crée apparaît bien. Pour avoir le détail de cette table, on tapera:

```
mysql> DESCRIBE coord;
+------------+-----------------+---------+------+-----------+------+
| Field      | Type            | Null    | Key  | Default |Extra |
+------------+-----------------+---------+------+-----------+------+
```

```
| nom      | varchar(20) | YES  |        | NULL    |       |
| prenom   | varchar(20) | YES  |        | NULL    |       |
| email    | varchar(30) | YES  |        | NULL    |       |
+----------+-------------+------+--------+---------+-------+
3 rows in set (0.00 sec)
```

Pour notre deuxième exemple **Apache+PHP+MySQL**, on créera la table suivante:

```
mysql> CREATE TABLE ref (
    -> date VARCHAR(20),
    -> host VARCHAR(20),
    -> ip VARCHAR(15),
    -> os VARCHAR(20),
    -> page VARCHAR(30)
    -> );
Query OK, 0 rows affected (0.05 sec)
```

Elle contiendra les informations sur les visiteurs du site. A présent pour quitter tapez simplement **quit**.

# 5 Installation d'Apache

## 5.1 Installation

Voici les packages standards à installer pour **Apache** (un **urpmi apache2** devrait suffire), voilà la liste par défaut

**apache-conf-2.0.53-5mdk.i586**
**apache2-2.0.53-9.1.102mdk.i586**
**apache2-common-2.0.53-9.1.102mdk.i586**
**apache2-modules-2.0.53-9.1.102mdk.i586**
**libapr-util0-0.9.6-4mdk.i586**
**libapr0-0.9.6-3mdk.i586**

Pour le PHP un **urpmi apache2-mod_php5** suffit, il mène à l'installation des packages suivants :

**apache2-mod_php5-2.0.53_5.0.3-5mdk.i586**
**libphp5_common5-5.0.3-8mdk.i586**

**php5-ctype-5.0.3-4mdk.i586**
**php5-ftp-5.0.3-4mdk.i586**
**php5-gettext-5.0.3-4mdk.i586**
**php5-ini-5.0.3-3mdk.noarch**
**php5-pcre-5.0.3-4mdk.i586**
**php5-posix-5.0.3-4mdk.i586**
**php5-session-5.0.3-4mdk.i586**
**php5-sysvsem-5.0.3-4mdk.i586**
**php5-sysvshm-5.0.3-4mdk.i586**
**php5-yp-5.0.3-4mdk.i586**

Vous trouverez ces autres packages pour les extensions **GD**, **MySQL et LDAP**

**php5-gd-5.0.3-4mdk.i586.rpm**

**libmysql14-4.1.11-1.1.102mdk.i586**
**php5-mysql-5.0.3-6mdk.i586** (requière **MySQL**)
**php5-ldap-5.0.3-6mdk.i586.rpm** (requière **OpenLDAP**)

**ATTENTION** Pour **LDAP**, le serveur **LDAP** fourni avec la Mandrake doit être préalablement installé, pour plus d'info voir la page **OpenLDAP** et le document téléchargeable correspondant sur www.funix.org.

## 5.2 Présentation de l'arborescence

L'installation va créer un répertoire **/etc/httpd** contenant:
- répertoire **2.0**
- répertoire **conf** qui contient les fichiers de configuration
- répertoire **conf.d** qui contient les fichiers de configuration des modules non standards (dont **PHP**)
- fichier **logs** lien vers **/var/log/httpd/** contenant les logs d'**Apache**

Le répertoire **2.0** contient
- un lien **conf** vers le répertoire vu plus haut
- un lien **extramodules** vers le répertoire **/usr/lib/apache2-extramodules/** contenant des modules particuliers (dont le module **PHP**) (à ce propos y a un bug dans la LE2005 le lien va vers **/usr/lib/apache-extramodules/** qui n'existe pas...)
- un lien **lib** vers le répertoire de bibliothèque **/usr/lib**
- à nouveau un lien **logs** vers le répertoire de log **/var/log/httpd/**

- un lien **modules** vers le répertoire **/usr/lib/apache2/** contenant les modules standards d'**Apache** (hôte virtuel, proxy, etc.)
Le répertoire de log **/var/log/httpd** contient essentiellement deux fichiers:
- **access_log** listant les accès au serveur
- **error_log** listant les erreurs en tout genre

Le répertoire de modules **/usr/lib/apache2** contient tous les modules utilisables par **Apache**, pour info un module est une extension logicielle à **Apache**, lui permettant par exemple d'interpréter le **PHP** (module **mod_php5.so** se trouvant sous **/usr/lib/apache2-extramodules**).

Le répertoire **/etc/httpd/conf** contient:
- le fichier de configuration d'Apache **httpd2.conf**
- le fichier **commonhttpd.conf** qui est appelé dans **httpd2.conf**, il contient des paramètres complémentaires de configuration.
- le fichier **fileprotector.conf** pour protéger les fichiers php contre l'édition
- autre fichier de conf **httpd2-perl.conf** qui ne sert à rien, il est là à titre d'exemple
- **apache-mime.types** fixe le type de fichier suivant l'extension du dit fichier (.**doc**=msword, . **ps**=postscript, ...), ça permet au client qui se connecte sur le serveur, de savoir comment interpréter le fichier suivant son extension.
- **magic** qui sert pour le module **mod_mime_magic** qui est désactivé par défaut
- **magic.default** qui est un modèle pour le fichier **magic**, il ne sert pas à la configuration d'**Apache**
- un répertoire **vhosts** pour gérer les hôtes virtuels, en gros pour un même serveur vous pouvez définir plusieurs noms, voir paragraphe plus bas
- un répertoire **addon-modules** qui contient des fichiers de configuration de certains modules

Maintenant vous avez encore d'autres fichiers sous **/var/www** avec :
- répertoire **cgi-bin** qui contiendra vos scripts CGI
- répertoire **html** contenant la page d'accueil d'**Apache** par défaut
- répertoire **icons**, qui comme son nom l'indique contient des icônes, notamment celles pour identifier le type de fichier
- répertoire **error** contient les messages d'erreur d'Apache, c'est ces fichiers qu'il faudra modifier pour personnaliser les mesages d'erreur.
- répertoire **perl** contient les scripts perl

# 5.3 Configuration

## 5.3.1 Les fichiers de configuration

Voici un extrait des fichiers de conf d'**Apache** avec les points que je juge important :
Pour **/etc/httpd/conf/httpd2.conf**

```
### Main Configuration Section
### You really shouldn't change these settings unless you're a guru
###
ServerRoot /etc/httpd/2.0
#ServerName localhost
#LockFile /etc/httpd/httpd.lock
PidFile /var/run/httpd.pid
ErrorLog logs/error_log
LogLevel warn
# là où sont placés les fichiers par défaut du serveur
DocumentRoot /var/www/html


### Dynamic Shared Object (DSO) Support
###
### You should always leave those three, as they are needed for
### normal use.
### mod_access (Order, Allow, etc..)
### mod_log_config (Transferlog, etc..)
### mod_mime (AddType, etc...)
# chargement des modules (non chargé si ligne commentée par un #)
# la documentation sur tous les modules se trouvent sur le site officiel
# ou dans le package apache2-manual package.

LoadModule access_module       modules/mod_access.so
LoadModule auth_module         modules/mod_auth.so
LoadModule auth_anon_module    modules/mod_auth_anon.so
##LoadModule auth_dbm_module    modules/mod_auth_dbm.so
LoadModule auth_digest_module  modules/mod_auth_digest.so
##LoadModule charset_lite_module       modules/mod_charset_lite.so
##LoadModule case_filter_module modules/mod_case_filter.so
##LoadModule case_filter_in_module     modules/mod_case_filter_in.so
##LoadModule ext_filter_module  modules/mod_ext_filter.so
LoadModule include_module      modules/mod_include.so
LoadModule log_config_module   modules/mod_log_config.so
#LoadModule log_forensic_module modules/mod_log_forensic.so
LoadModule logio_module        modules/mod_logio.so
```

```
LoadModule env_module          modules/mod_env.so
##LoadModule mime_magic_module  modules/mod_mime_magic.so
##LoadModule cern_meta_module   modules/mod_cern_meta.so
LoadModule expires_module      modules/mod_expires.so
LoadModule headers_module      modules/mod_headers.so
LoadModule usertrack_module    modules/mod_usertrack.so
##LoadModule unique_id_module   modules/mod_unique_id.so
LoadModule setenvif_module     modules/mod_setenvif.so
LoadModule mime_module         modules/mod_mime.so
LoadModule status_module       modules/mod_status.so
LoadModule autoindex_module    modules/mod_autoindex.so
LoadModule asis_module         modules/mod_asis.so
LoadModule info_module         modules/mod_info.so
LoadModule cgi_module          modules/mod_cgi.so
##LoadModule cgid_module              modules/mod_cgid.so
LoadModule vhost_alias_module  modules/mod_vhost_alias.so
LoadModule negotiation_module  modules/mod_negotiation.so
LoadModule dir_module          modules/mod_dir.so
LoadModule imap_module         modules/mod_imap.so
LoadModule actions_module      modules/mod_actions.so
##LoadModule speling_module     modules/mod_speling.so
LoadModule userdir_module      modules/mod_userdir.so
LoadModule alias_module        modules/mod_alias.so
LoadModule rewrite_module      modules/mod_rewrite.so
#LoadModule dumpio_module      modules/mod_dumpio.so

###
### Global Configuration
###
# We now support multiple apache configurations on the same server. In
# common.conf, we put all directives that are common to all implementations
# (httpd, httpd-perl, etc.)
# For Apache2 we load all conf files in conf.d
# on charge des fichiers de configuration complémentaires
Include /etc/httpd/conf.d/*.conf
Include conf/commonhttpd.conf
Include conf/fileprotector.conf

###
### IP Address/Port and Proxied configuration section
###
# The APACHEPROXIED setting can be set in /etc/rc.d/init.d/httpd if you
# are using a proxy or accelerator, like the Apache-SGI or khttpd, so that
# the fast web server serves static content while Apache handles the
# cgi or php files
# vous avez la possibilité de configurer apache comme un proxy
```

```
# dans ce cas les directives suivantes fixe les numéros de port du proxy
# et du serveur http

#BindAddress *
<IfDefine APACHEPROXIED>
   Listen 8080
</IfDefine>
<IfDefine !APACHEPROXIED>
   Listen 80
</IfDefine>

# Likewise, we can set apache as the server by default and send perl
# requests via ProxyPass to apache-mod_perl. It increases performance
# since the perl interpreter is only used for perl and the standard apache
# does all the html and image files, with a smaller footprint.
#
# If you install apache and apache-mod_perl, this is the default config.
# If you don't want two web servers to use perl, uninstall apache, and
# apache-mod_perl will not be proxied.

<IfDefine PERLPROXIED>
<IfModule mod_rewrite.c>
   RewriteEngine on
   RewriteRule ^proxy:.*  -  [F]
   RewriteRule ^(.*\/perl\/.*)$  http://%{HTTP_HOST}:8200$1 [P]
   RewriteRule ^(.*\/cgi-perl\/.*)$  http://%{HTTP_HOST}:8200$1 [P]
</IfModule>
</IfDefine>

###
### Log configuration Section
###

<IfModule mod_log_config.c>
#Single logfile with access, agent and referer information
#This is the default, if vlogs are not defined for the main server
CustomLog logs/access_log combined env=!VLOG
#If VLOG is defined in conf/vhosts/Vhost.conf, we use this entry
CustomLog "|/usr/sbin/advxsplitlogfile" vhost env=VLOG
</IfModule>

###
### Virtual Hosts
###
# We include different templates for Virtual Hosting. Have a look in the
# vhosts directory and modify to suit your needs.
```

```
# chargement du fichier de configuration des hôtes virtuels
Include conf/vhosts/Vhosts.conf
#Include conf/vhosts/DynamicVhosts.conf
#Include conf/vhosts/VirtualHomePages.conf


###
### Performance settings Section
###
#
# Timeout: The number of seconds before receives and sends time out.
#
# 300s avant d'abandonner, c'est peut être un peu long
Timeout 300

#
# KeepAlive: Whether or not to allow persistent connections (more than
# one request per connection). Set to "Off" to deactivate.
#
# pour empêcher plusieurs requêtes par connexion
KeepAlive On

#
# MaxKeepAliveRequests: The maximum number of requests to allow
# during a persistent connection. Set to 0 to allow an unlimited amount.
# We recommend you leave this number high, for maximum performance.
#
# nombre maximum de requêtes durant une connexion
MaxKeepAliveRequests 100

#
# KeepAliveTimeout: Number of seconds to wait for the next request from the
# same client on the same connection.
#
# nombre max de seconde à attendre la prochaine requête d'un client pendant la même connexion
KeepAliveTimeout 15


# prefork MPM [THIS IS THE DEFAULT]
# paramètres sur les nombres de process serveurs
# StartServers: number of server processes to start
# MinSpareServers: minimum number of server processes which are kept spare
# MaxSpareServers: maximum number of server processes which are kept spare
# MaxClients: maximum number of server processes allowed to start
# MaxRequestsPerChild: maximum number of requests a server process serves
```

```
<IfModule prefork.c>
   StartServers       5
   MinSpareServers    5
   MaxSpareServers    10
   MaxClients         150
   MaxRequestsPerChild 0
</IfModule>

# worker MPM
# paramètres de performance
# StartServers: initial number of server processes to start
# MaxClients: maximum number of simultaneous client connections
# MinSpareThreads: minimum number of worker threads which are kept spare
# MaxSpareThreads: maximum number of worker threads which are kept spare
# ThreadsPerChild: constant number of worker threads in each server process
# MaxRequestsPerChild: maximum number of requests a server process serves

<IfModule worker.c>
   StartServers       2
   MaxClients         150
   MinSpareThreads    25
   MaxSpareThreads    75
   ThreadsPerChild    25
   MaxRequestsPerChild 0
</IfModule>

# perchild MPM
# NumServers: constant number of server processes
# StartThreads: initial number of worker threads in each server process
# MinSpareThreads: minimum number of worker threads which are kept spare
# MaxSpareThreads: maximum number of worker threads which are kept spare
# MaxThreadsPerChild: maximum number of worker threads in each server process
# MaxRequestsPerChild: maximum number of connections per server process

<IfModule perchild.c>
   NumServers         5
   StartThreads       5
   MinSpareThreads    5
   MaxSpareThreads    10
   MaxThreadsPerChild  20
   MaxRequestsPerChild 0
   # force fcntl
   AcceptMutex fcntl
</IfModule>

# peruser MPM
```

```
# StartServers: number of server processes to start
# MinSpareServers: minimum number of server processes which are kept spare
# MaxSpareServers: maximum number of server processes which are kept spare
# MaxClients: maximum number of server processes allowed to start
# MaxRequestsPerChild: maximum number of requests a server process serves

<IfModule peruser.c>
   MinSpareServers     2
   MaxProcessors       10
   MaxClients          150
   MaxRequestsPerChild 100
   Multiplexer apache apache
   # Processor user group /home/user
   # chroot dir is optional:
   # Processor user group
</IfModule>

###
### webapps configuration section
###

# Web applications should be activated _after_ apache has been
# configured properly.
Include /etc/httpd/webapps.d/*.conf
```

Dans le fichier **commonhttpd.conf** vous aurez la configuration notamment de :

```
### Common server configuration
# nom du proprio du process, l'important est que ce soit pas root
# pour une raison de sécurité
User apache
Group apache

#
# ServerAdmin: Your address, where problems with the server should be
# e-mailed.  This address appears on some server-generated pages, such
# as error documents.
# mail de l'administrateur du serveur
ServerAdmin root@localhost

# options d'accès au répertoire racine d'Apache
<Directory />
  Options -All -Multiviews
  AllowOverride None
  <IfModule mod_access.c>
    Order deny,allow
    Deny from all
```

```
  </IfModule>
</Directory>

# on pourra accèder aux pages HTML de vos utilisateurs si ceux-ci créent
# un répertoire public_html (droit 755) sous leur homedirectory (droit 755)
# on y accès en tapant dans un navigateur http://serveur-apache/~login-utilisateur
<IfModule mod_userdir.c>
   UserDir public_html
</IfModule>

# DirectoryIndex: Name of the file or files to use as a pre-written HTML
# directory index.  Separate multiple entries with spaces.
# nom du fichier (suivant son type) qui sera chargé en premier dans un répertoire,

<IfModule mod_include.c>
   <IfModule mod_dir.c>
      DirectoryIndex index.shtml
   </IfModule>
</IfModule>

<IfModule mod_php4.c>
   <IfModule mod_dir.c>
      DirectoryIndex index.php index.phtml index.php3 index.php4
   </IfModule>
</IfModule>

<IfModule mod_php5.c>
   <IfModule mod_dir.c>
      DirectoryIndex index.php index.phtml index.php3 index.php4 index.php5
   </IfModule>
</IfModule>

<IfModule mod_dir.c>
    DirectoryIndex index.html index.html.var index.cgi index.pl index.htm Default.htm default.htm
index.xml
</IfModule>

# nom du fichier pour limiter l'accès à un répertoire
AccessFileName .htaccess

# directive pour qu'on ne puisse pas voir les fichier .htaccess
<IfModule mod_access.c>
<Files ~ "^\.ht">
   Order allow,deny
   Deny from all
```

```
</Files>
</IfModule>

(...)

# Cette directive permet de définir les informations sur le serveur Apache qui seront consultables
# vous avez le choix entre Full | OS | Minor | Minimal | Major | Prod
ServerTokens Full

(...)

# définition des alias, on verra plus bas leur intérêt
<IfModule mod_alias.c>

    #
    # Note that if you include a trailing / on fakename then the server will
    # require it to be present in the URL.  So "/icons" isn't aliased in this
    # example, only "/icons/"..
    #
    Alias /icons/ /var/www/icons/
    Alias /doc /usr/share/doc


    #
    # ScriptAlias: This controls which directories contain server scripts.
    # ScriptAliases are essentially the same as Aliases, except that
    # documents in the realname directory are treated as applications and
    # run by the server when requested rather than as documents sent to the client.
    # The same rules about trailing "/" apply to ScriptAlias directives as to
    # Alias.
    #
    ScriptAlias /cgi-bin/ /var/www/cgi-bin/
    ScriptAlias /protected-cgi-bin/ /var/www/protected-cgi-bin/

    <IfModule mod_perl.c>
        #Provide two aliases to the same cgi-bin directory,
        #to see the effects of the 2 different mod_perl modes
        #for Apache::Registry Mode
        Alias /perl/ /var/www/perl/
        #for Apache::Perlrun Mode
        Alias /cgi-perl/ /var/www/perl/
    </IfModule>


</IfModule>
```

(...)

```
# pour une page multi language le français est prioritaire
  <IfModule mod_negotiation.c>
      LanguagePriority fr en de es it da nl et el ja kr no pl pt pt-br ru ltz ca sv tw
   </IfModule>
```

(...)

```
# définition des droits d'accès aux pages perso

<Directory /home/*/public_html>
   AllowOverride All
   Options MultiViews -Indexes Includes FollowSymLinks
   <IfModule mod_access.c>
    Order allow,deny
    Allow from all
   </IfModule>
</Directory>

<Directory /home/*/public_html/cgi-bin>
   Options +ExecCGI -Includes -Indexes
   SetHandler cgi-script
   <IfModule mod_access.c>
    Order allow,deny
    Allow from all
   </IfModule>
</Directory>

<IfModule mod_perl.c>
   <Directory /home/*/public_html/perl>
      SetHandler perl-script
      <IfDefine !APACHE2>
       PerlHandler Apache::PerlRun
      </IfDefine>
      <IfDefine APACHE2>
       PerlResponseHandler ModPerl::PerlRun
      </IfDefine>
      Options -Indexes ExecCGI
      PerlSendHeader On
     <IfModule mod_access.c>
      Order allow,deny
      Allow from all
     </IfModule>
```

```
   </Directory>
</IfModule>
```

(...)

Vous constaterez que vous pouvez très bien intégrer ce dernier fichier très facilement au fichier **httpd2.conf** pour simplifier.

A noter le fichier **/etc/httpd/conf.d/70_mod_php5.conf** contenant

```
<IfDefine HAVE_PHP5>
  <IfModule !mod_php5.c>
    LoadModule php5_module    extramodules/mod_php5.so
  </IfModule>
</IfDefine>

<IfModule mod_php5.c>
   PHPINIDir /etc
</IfModule>

<IfModule mod_mime.c>
   AddType application/x-httpd-php .php
   AddType application/x-httpd-php .php3
   AddType application/x-httpd-php .php4
   AddType application/x-httpd-php .php5
   AddType application/x-httpd-php .phtml
   AddType application/x-httpd-php-source .phps
</IfModule>
```

Il sert pour l'interprétation des pages se terminant par **.php**,**.php4**, etc par le module **PHP**.

### 5.3.2 Les pages webs utilisateurs

Le problème avec le répertoire **public_html** des utilisateurs et qu'il faut mettre 755 au niveau de la home directory, ce qui est particulièrement gênant au niveau sécurité. Vous pouvez spécifier que chaque utilisateur doit créer ses pages sous **/home/http/login-utilisateur** en écrivant pour la variable **UserDir**

**UserDir /home/httpd**

Ainsi pour l'utilisateur toto quand vous taperez comme URL **http://serveur-apache/~toto**, **apache** ira chercher le fichier **index.htm** sous **/home/httpd/toto**. On peut aller plus loin en spécifiant un répertoire particulier, **/home/httpd/toto/html** par exemple, en écrivant:

**UserDir /home/httpd/*/html**

Dans ce cas pensez à modifier dans **commonhttpd.conf** la partie

```
<Directory /home/*/public_html>
   AllowOverride All
   Options MultiViews -Indexes Includes FollowSymLinks
   <IfModule mod_access.c>
     Order allow,deny
     Allow from all
   </IfModule>
</Directory>

<Directory /home/*/public_html/cgi-bin>
   Options +ExecCGI -Includes -Indexes
   SetHandler cgi-script
   <IfModule mod_access.c>
     Order allow,deny
     Allow from all
   </IfModule>
</Directory>
<IfModule mod_perl.c>
   <Directory /home/*/public_html/perl>
      SetHandler perl-script
      <IfDefine !APACHE2>
        PerlHandler Apache::PerlRun
      </IfDefine>
      <IfDefine APACHE2>
        PerlResponseHandler ModPerl::PerlRun
      </IfDefine>
      Options -Indexes ExecCGI
      PerlSendHeader On
     <IfModule mod_access.c>
       Order allow,deny
       Allow from all
     </IfModule>
   </Directory>
</IfModule>
```

Et remplacer **/home/*/public_html** par **/home/httpd/*/html**

## 5.4 Lancement de l'application

Vous allez trouver un fichier de lancement/arrêt du nom de **httpd** dans **/etc/rc.d/init.d**, le lancement est automatique à l'état de marche 3, 4 et 5. Toutefois pour lancer **Apac**he la première fois lors de son installation il sera nécessaire de taper
**/etc/rc.d/init.d/httpd start**

En cas de modification du fichier de configuration, pour relancer **Apache** il suffit de taper

**/etc/rc.d/init.d/httpd restart**

Pour connaître l'état

**/etc/rc.d/init.d/httpd status**

voilà le résultat

**Apache is running.**
**httpd2: 28556 28555 28554 28553 28552 28544**

**Use /etc/rc.d/init.d/httpd extendedstatus for more information.**

Pour avoir encore plus d'info

**/etc/rc.d/init.d/httpd extendedstatus**

voilà le résultat

**Apache Server Status for 127.0.0.1**

**Server Version: Apache-AdvancedExtranetServer/2.0.53 (Mandriva Linux/PREFORK-9.2.102mdk) PHP/5.0.3**

**Server Built: Sep 6 2005 09:59:01**

---

**Current Time: Monday, 19-Sep-2005 19:05:54 CEST**
**Restart Time: Monday, 19-Sep-2005 18:50:04 CEST**
**Parent Server Generation: 0**
**Server uptime: 15 minutes 49 seconds**
**Total accesses: 0 - Total Traffic: 0 kB**

**CPU Usage: u0 s0 cu0 cs0**
**0 requests/sec - 0 B/second -**
**1 requests currently being processed, 4 idle workers**

**W____.....**
**.....**
**.....**
**.....**
**.....**
**.....**
**.....**
**.....**
**.....**
**.....**
**.....**
**.....**
**.....**
**.....**
**.....**
**.....**

**Scoreboard Key:**
**"_" Waiting for Connection, "S" Starting up, "R" Reading Request,**
**"W" Sending Reply, "K" Keepalive (read), "D" DNS Lookup,**
**"C" Closing connection, "L" Logging, "G" Gracefully finishing,**
**"I" Idle cleanup of worker, "." Open slot with no current process**

**########################################**
**########################################**

Pour stopper

**/etc/rc.d/init.d/httpd stop**

# 5.5 Tests de fonctionnement

Tout d'abord en tant que simple utilisateur, **toto** par exemple, vous allez créer un répertoire **public_html** dans votre répertoire contenant un fichier **index.html** basique du style:

**<html>**
**<body>**
** hello world **
**</body>**
**</html>**

**ATTENTION** répertoire **/home/toto** à 755 de même que **/home/toto/public_html**

Maintenant d'un poste client tapez dans le champ URL de votre navigateur préféré, en admettant que votre serveur se nomme **obelix:**

**http://obelix**

Et là magique, apparaît la page d'accueil d'Apache, pour info celle-ci se trouve sous **/var/www/html**. Maintenant tapez:

**http://obelix/~toto**

et vous voyez apparaître sur un fond bien gris (ou d'une autre couleur ça dépend de la config de votre navigateur).

**hello world**

C'est bon ça marche. Pour tester le PHP3, en tant que **toto**, écrivez un fichier **info.php3** sous votre répertoire **public_html** contenant:

```
<?
phpinfo();
?>
```

D'un poste client, tapez dans le champ adresse (ou URL):

**http://obelix/~toto/info.php3**

Voilà le résultat :

**PHP Version 5.0.3**

| | |
|---|---|
| **System** | Linux tosh.kervao.fr 2.6.11-6mdk #1 Tue Mar 22 16:04:32 CET 2005 i686 |
| **Build Date** | Mar 20 2005 21:04:44 |
| **Configure Command** | This is irrelevant, look inside the /usr/share/doc/libphp5_common5-5.0.3/configure_command file. urpmi is your friend, use it to install extensions not shown below. |
| **Server API** | Apache 2.0 Handler |
| **Virtual Directory Support** | disabled |
| **Configuration File (php.ini) Path** | /etc/php5.ini |
| **Scan this dir for additional .ini files** | /etc/php5.d |
| **additional .ini files parsed** | /etc/php5.d/12_ctype.ini, /etc/php5.d/22_ftp.ini, /etc/php5.d/24_gettext.ini, /etc/php5.d/41_pcre.ini, /etc/php5.d/43_posix.ini, /etc/php5.d/47_session.ini, /etc/php5.d/57_sysvsem.ini, /etc/php5.d/58_sysvshm.ini, /etc/php5.d/65_yp.ini |
| **PHP API** | 20031224 |
| **PHP Extension** | 20041030 |
| **Zend Extension** | 220040412 |
| **Debug Build** | no |
| **Thread Safety** | disabled |
| **IPv6 Support** | enabled |
| **Registered PHP Streams** | php, file, http, ftp, compress.zlib, https, ftps |
| **Registered Stream Socket Transports** | tcp, udp, unix, udg, ssl, sslv3, sslv2, tls |

C'est bon le module **PHP** fonctionne.

**ATTENTION** Si comme moi vous utilisez des URL du style **http://www.funix.org/fr/linux/main-linux.php3?ref=apache&page=menu** c'est à dire avec passage d'arguments, à partir de la version 4.2.0 de **php** cette fonctionnalité a été désactivée pour des raisons de sécurité. D'autres méthodes permettent néanmoins de pouvoir passer des arguments d'une page à une autre. En attendant pour réactiver cette fonctionnalité, créer le fichier **/etc/php/php.ini** dans lequel vous mettrez

**register_globals = On**

Relancez **Apache.** Cela dit vous avez intérêt à laisser cette variable à Off et procéder comme suit. Voilà le bout de code php des pages html de ce site pour faire l'inclusion du menu à gauche ou pas. Voilà la code avec **register_globals = On**

```
<?
    if ($page=="menu")
    {
        $menu="menu.htm";
        include($menu);
    }
?>
```

Et le même code avec **register_globals = Off**

```
<?
    if ($_GET["page"]=="menu")
    {
        $menu="menu.htm";
        include($menu);
    }
?>
```

Pour plus d'info sur cette fonctionnalité désactivée et sur la manière de faire autrement http://www.php.net/manual/en/security.registerglobals.php

# 6 Configuration Apache avancée

## 6.1 Les hôtes virtuels

On peut mettre en place des hôtes virtuels, en d'autres termes un utilisateur pour un même serveur Apache croira en voir plusieurs. Exemple, soit votre serveur Apache **obelix** (adresse IP **192.168.13.11**), votre domaine **breizland.bz**, on va créer les hôtes virtuels **www.asterix.breizland.bz** et **www.idefix.breizland.bz** qui vont pointer chacun vers un endroit différent du disque (respectivement **/usr/local/asterix** et **/usr/local/idefix** chacun contenant des pages html).

Dans le fichier **/etc/httpd/conf/vhosts/Vhosts.conf** on va rajouter:

```
NameVirtualHost 192.168.13.11

<VirtualHost 192.168.13.11>
ServerName obelix.breizland.bz
DocumentRoot /var/www/html
ErrorLog logs/obelix-error_log
TransferLog logs/obelix-access_log
</VirtualHost>

<VirtualHost 192.168.13.11>
ServerName www.asterix.breizland.bz
DocumentRoot /usr/local/asterix
ErrorLog logs/asterix-error_log
TransferLog logs/asterix-access_log
</VirtualHost>
```

**<VirtualHost 192.168.13.11>**
**ServerName www.idefix.breizland.bz**
**DocumentRoot /usr/local/idefix**
**ErrorLog logs/idefix-error_log**
**TransferLog logs/idefix-access_log**
**</VirtualHost>**

Pour info, ce fichier est appelé dans **/etc/http/conf/httpd2.conf** à la ligne

**Include conf/vhosts/Vhosts.conf**

Maintenant dans le fichier **commonhttpd.conf** rajoutez

**<Directory /usr/local/idefix>**
**Options -Indexes FollowSymLinks MultiViews**
**AllowOverride All**
**<IfModule mod_access.c>**
**Order allow,deny**
**Allow from all**
**</IfModule>**
**</Directory>**

**<Directory /usr/local/asterix>**
**Options -Indexes FollowSymLinks MultiViews**
**AllowOverride All**
**<IfModule mod_access.c>**
**Order allow,deny**
**Allow from all**
**</IfModule>**
**</Directory>**

Relancez **Apache** en tapant:

**/etc/rc.d/init.d/httpd restart**

Maintenant nous allons créer nos hôtes **asterix** et **idefix**, pour cela vous avez deux méthodes:
- rajouter **www.asterix.breizland.bz** et **www.idefix.breizland.bz** dans **/etc/hosts** sur la même ligne que votre serveur **Apache** (**obelix** dans notre exemple).

**192.168.13.11 obelix obelix.breizland.bz www.asterix.breizland.bz www.idefix.breizland.bz**

Normalement si vous faites un ping sur **www.idefix.breizland.bz** ça devrait marcher, pour les postes clients il faudra rajouter la même ligne dans le fichier **hosts** (non nécessaire).

- si vous disposez d'un serveur DNS sur votre machine , au niveau de votre config DNS dans votre fichier **breizland.bz** qui se trouve sous **/var/named** vous devez rajouter tout à la fin:

**www.asterix A 192.168.13.11**
**www.idefix A 192.168.13.11**

Relancez le DNS en tapant:

**/etc/rc.d/init.d/named restart**

Pour tester tapez dans un shell:

**ping www.asterix.breizland.bz**

Maintenant dans le champ URL de votre navigateur préféré:

**http://www.asterix.breizland.bz**

Et là, normalement vous devriez voir s'afficher la page que vous avez placé sous **/usr/local/asterix**

## 6.2 Les alias

Si vous ne voulez pas mettre en place un serveur DNS, vous avez un moyen plus simple, les alias. Concrètement, votre serveur s'appelle **obelix**, vous voulez rendre accessibles les fichiers html se trouvant sous **/usr/share/doc/HTML**, les utilisateurs devront taper dans leur navigateur préféré: **http://obelix/documentation**. Pour cela dans votre fichier **/etc/httpd/conf/commonhttpd.conf**, vous allez rajouter:

**Alias /icons/ /var/www/icons/**
**Alias /doc /usr/share/doc/**
**Alias /documentation /usr/share/doc/HTML**

Les deux premières lignes sont déjà existantes sur une configuration standard, vous constaterez que les adresses **http://obelix/icons/** et **http://obelix/doc** ne sont pas accessibles car il n'y a pas de fichier index dans les répertoires **/var/www/icons** et **/usr/share/doc**.
Rajoutez maintenant au niveau des directives **Directory** les lignes suivantes

```
<Directory /usr/share/doc/HTML>
    Options -Indexes FollowSymLinks MultiViews
    AllowOverride All
    Order allow,deny
    Allow from all
</Directory>
```

Relancez **Apache**, pour tester l'alias.

## 6.3 Protection d'une page

La protection d'une page pour l'utilisateur **olivier** se fait de manière très simple, tous les fichiers à accès limité doivent être concentré dans un même répertoire **/home/olivier/public_html/reserve** par exemple, il suffit de créer dans celui-ci un fichier qu'on devra nommer **.htaccess** contenant:

```
AuthUserFile auth/olivier.users
AuthName "Acces Restreint"
AuthType Basic

<Limit GET POST>
require valid-user
</Limit>
```

Le fichier **olivier.users** va contenir la liste des utilisateurs habilités à accéder au répertoire où se trouve **.htaccess**, il va se trouver sous le répertoire **/etc/httpd/2.0/auth**, pour info vous pouvez changer le chemin **/etc/httpd/2.0** en modifiant la valeur de la variable **ServerRoot** qu'on trouve dans le fichier **httpd2.conf**. Pour créer ce fichier il suffit d'une part de créer le répertoire **/etc/httpd/auth** si celui-ci n'existe pas, puis de taper:

```
htpasswd -c /etc/httpd/2.0/auth/olivier.users olivier
```

L'option **-c** correspond à la création du fichier. On va alors avoir à rentrer un mot de passe pour l'utilisateur **olivier** habilité à accéder au répertoire protégé.

**New password:**

On confirme

**Re-type new password:**

Ze problème est que vous devez taper cette variable en tant que root, l'autre solution est de mettre **auth** à 777 pour qu'un simple utilisateur puisse taper la commande **htpasswd**, ce qui est un peu moyen.

Pour que l'utilisateur **veronique** puisse accéder aussi au répertoire **reserve** d'**olivier**, vous taperez alors:

**htpasswd  /etc/httpd/2.0/auth/olivier.users veronique**

Maintenant quand à partir de votre navigateur préféré quand vous allez rentrer comme URL **http://obelix/~olivier/reserve**, vous aurez une fenêtre popup qui va s'ouvrir vous demandant de rentrer votre nom d'utilisateur et le mot de passe préalablement rentré.

A noter qu'il existe une directive qui permet de ne pas voir les fichiers **.htaccess** de vos utilisateurs à partir d'un navigateur**,** elle est dans le fichier **commonhttpd.conf** :

**<files ~ "/\.ht">**
**order deny,allow**
**deny from all**
**</files>**

# 7 Fonctionnement de MySQL

## 7.1 Tests de fonctionnement MySQL

Comme pré requis on suppose que vous avez installé, configuré **MySQL** et créé les deux exemples du paragraphe **MySQL**. On suppose que le serveur s'appelle **obelix** et l'utilisateur **olivier**.
A priori il suffit de mettre en place **PHP** avec **Apache** pour que les routines **MySQL** soient prises en compte.
Voici une page écrite en **PHP** qui va accéder à la base de donnée **essai** et à sa table **coord**.

**<?**
**$serveur="localhost";**

```
$login="olivier";
$pass="mot-de-passe";
$base="essai";
$table="coord";

$id=MYSQL_CONNECT($serveur,$login,$pass);
mysql_select_db($base);
$nom="hoarau";
$prenom="olivier";
$email="olivier.hoarau@fnac.net";
$query="INSERT INTO $table VALUES('$nom','$prenom','$email')";
$result=mysql_query($query,$id);
echo "Saisie terminée";
?>
```

Placer ce script dans **~/public_html** et appeler le **bd1.php3**

Dans votre navigateur préféré, dans le champ URL saisissez

**http://obelix/~olivier/bd1.php3**

A priori y a pas grand chose qui s'est passé, maintenant connecter vous à votre base **essai** dans un shell

```
[olivier@obelix olivier]$ mysql -u olivier -p essai
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10 to server version: 4.1.11

Type 'help' for help.

mysql> SELECT * FROM coord;
+----------+------------+---------------------------------------+
| nom      | prenom     | email                                 |
+----------+------------+---------------------------------------+
| hoarau | olivier    | olivier.hoarau@fnac.net               |
+----------+------------+---------------------------------------+
1 row in set (0.00 sec)
```

C'est bon ça fonctionne. Passons à un exemple plus pointu, on va entrer les informations concernant vos visiteurs dans une base **MySQL**, créer la table telle que décrite dans l'exemple 2 du chapitre **MySQL**, créer maintenant le script **PHP3**.

```
<?

$page=getenv("HTTP_REFERER");
$ip=getenv( "REMOTE_ADDR");
$host=gethostbyaddr($ip);
$d = date("d/m/Y H:i:s");
$expl=getenv("HTTP_USER_AGENT");

$serveur="localhost";
$login="olivier";
$pass="mot-de-passe";
$base="essai";
$table="ref";

$id=MYSQL_CONNECT($serveur,$login,$pass);
mysql_select_db($base);

$query="INSERT INTO $table VALUES('$d','$host','$ip','$expl','$page')";
$result=mysql_query($query,$id);

echo "$d $host($ip) $expl $page";

?>
```

Nommez ce script **bd2.php3** et placez le dans **~/public_html**. Dans votre navigateur préféré tapez dans le champ URL

**http://obelix/~olivier/bd2.php3**

Vous devriez voir la date, le nom de votre machine avec son adresse IP et des infos sur votre OS et votre navigateur. A présent connectons nous à la base:

```
[olivier@obelix olivier]$ mysql -u olivier -p essai
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10 to server version: 4.1.11
```

**Type 'help' for help.**

**mysql> SELECT * FROM ref;**

```
+-------------------------+---------------------+-----------------+--------------------------+---------+
| date                    | host                | ip              | os                       | page    |
+-------------------------+---------------------+-----------------+--------------------------+---------+

| 24/04/2000 08:34:05 | asterix.armoric.bz | 192.168.13.11 | Mozilla/4.61 [en] (X  |         |
+-------------------------+---------------------+-----------------+--------------------------+---------+
```

**1 row in set (0.00 sec)**

C'est bon le visiteur a bien été pris en compte.

Maintenant que vous savez comment **Apache** fonctionne avec **MySQL** et **PHP**, laissez libre cours à votre imagination.

# 7.2 phpMyAdmin

**phpMyAdmin** est un ensemble de scripts PHP qui permet d'administrer des bases **MySQL** à partir d'un navigateur. Vous pouvez le récupérer à l'URL www.phpmyadmin.net/home_page/.
En détail phpMyAdmin permet de:
- créer et supprimer des bases de données,
- éditer, ajouter ou supprimer des champs,
- taper des commandes SQL,
- gérer les clés de champs,
- ...

L'archive se présente sous la forme d'un tarball qu'on décompresse en tapant :

**tar xvfz phpMyAdmin-2.6.3-pl1.tar.gz**

A noter qu'il existe un rpm Mandriva de **phpMyAdmin**, mais il ne fonctionne pas avec PHP5, il faut nécessairement les packages de PHP4.
Revenons à notre **tar**, cela va créer dans le répertoire de travail un répertoire **phpMyAdmin-2.6.3-pl1**. Dans ce répertoire on va éditer le fichier **config.inc.php**, On doit d'abord indiquer l'URL pour atteindre **phpMyAdmin**.
**$cfg['PmaAbsoluteUri'] = 'URL pour atteindre phpMyAdmin (voir plus bas)';**

On définit l'utilisateur pour accéder à la base **MySQL**

**$cfg['Servers'][$i]['user'] = 'olivier'; // MySQL user**
**$cfg['Servers'][$i]['password'] = 'mot-de-passe-en-clair'; // MySQL password**

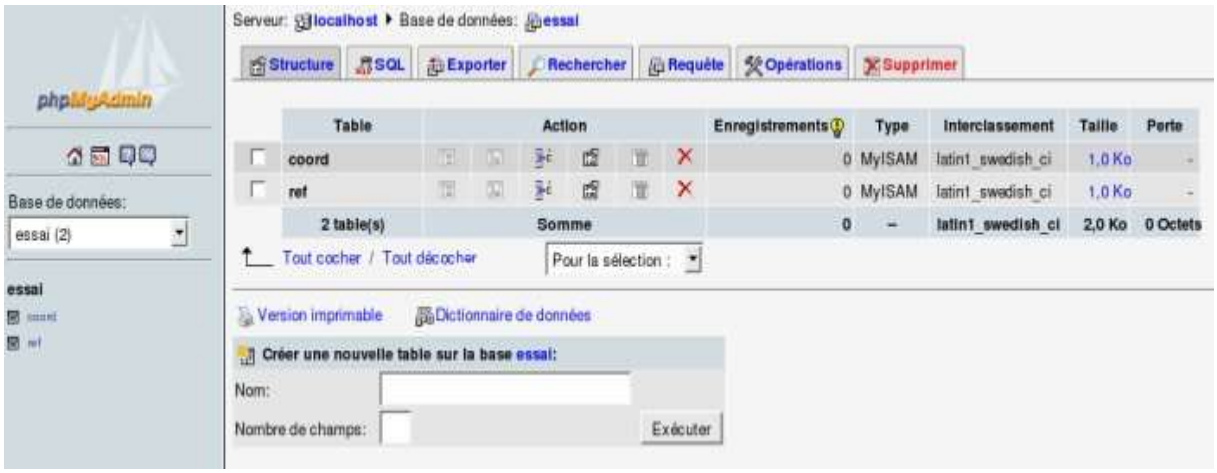
Vous pourrez changer la langue par défaut sur la page d'accueil de phpMyAdmin.

Maintenant on doit rendre accessible le répertoire **phpMyAdmin** d'une page web, pour cela deux solutions:
- (solution simple) placer **phpMyAdmin** dans **/usr/local/apache/htdocs** et au niveau de la page d'accueil d'apache faire un lien vers **/usr/local/apache/htdocs/phpMyAdmin-2.6.3-pl1/index.php**
- (solution préconisée), créer un hôte virtuel pointant vers **./phpMyAdmin-2.6.3-pl1** qu'on appelera **www.sql.breizland.bz** .
Modifiez maintenant la variable **$cfg['PmaAbsoluteUri']** (N'oublier pas le **http://**)
**NOTE** Si ça vous gène que n'importe qui d'un navigateur puisse aller dans le répertore **phpMyAdmin**, mettez y des restrictions d'accès avec un fichier **.htaccess**.

Avec la solution hôte virtuel, à partir d'un navigateur quand on sélectionne **www.sql.breizland.bz** on tombe sur une fenêtre avec frame avec à gauche la liste des bases de données disponibles et à droite, le menu suivant:


Pour travailler sur une base de données particulières il suffit de la sélectionner dans le choix déroulant à gauche, on retrouve d'ailleurs notre base **essai**, pour en créer une autre il suffit de choisir **Create new database**

Si on sélectionne **essai** par exemple on obtient




Vous pouvez donc créer des nouvelles tables, faire des requêtes **SQL**, etc.

# 8 Scripts CGI

Pour activer les scripts CGI, veuillez vous référer au fichier **commonhttpd.conf**. Vous devriez avoir la ligne suivante :

**ScriptAlias /cgi-bin/ /var/www/cgi-bin/**

Cet alias permet d'accéder au répertoire des scripts CGI. Vous devez avoir aussi le package **perl-CGI** installé sur votre système.

Le but de l'exercice est de créer un script perl CGI qui va traiter un formulaire quelconque d'une page HTML. Vous allez créer en tant que root votre script perl sous **/var/www/cgi-bin/**, et le nommer **form.pl**, voici son contenu:

```
#!/usr/bin/perl
use CGI;
$html=new CGI;
print $html->header;

print "<HTML>\n";
print "<HEAD>\n";
print "<TITLE>Premier script CGI perl</TITLE>\n";
print "</HEAD>\n";
print "<BODY>\n";
print "<H1>Traitement du formulaire</H1>\n";
```

```
print "Nom :";
print $html->param('nom');
print "<p>\n";
print "Email :";
print $html->param('email');
print "<p>\n";
print "Commentaire:";
print $html->param('comment');

print "</BODY>\n";
print "</HTML>\n";
```

Donner les droits qui vont bien avec ce fichier:

**chmod 755 form.pl**

L'utilisateur apache du groupe apache doit en être propriétaire:

**chown apache:apache form.pl**

En tant qu'utilisateur standard (**olivier** dans notre exemple), créer maintenant le fichier HTML suivant que vous appellerez **formulaire.htm**

```
<html>
<body>
<h2>Formulaire</h2>
<form action="http://obelix/cgi-bin/form.pl" METHOD=GET>
Nom: <input type="text" name=nom size=20><br>
Email: <input type="text" name=email size=30><br>
Commentaire: <input type="text" name=comment size=30><br>
<input type=submit value="Envoyer"> <input type=reset value="remettre à zéro">
</form>
</body>
</html>
```

Voilà maintenant quand vous allez accéder à **http://obelix/~olivier/formulaire.htm**, vous allez avoir une page du style:

Haut du formulaire

Nom:

Email:

Commentaire:

Envoyer | remettre à zéro

Bas du formulaire

En appuyant sur **Envoyer** ça va déclencher l'exécution du script CGI perl, qui va provoquer l'affichage des valeurs précédemment saisies.

# 9 PHP et LDAP

Voilà un petit programme qui va nous permettre de rajouter une entrée dans la base **LDAP**, libre à vous maintenant de créer des formulaires de saisie, de destruction, et de recherche:

```
<?
// nom du serveur LDAP
 $server="asterix";

// identification de l'administrateur de la base
 $rootdn="cn=Manager, dc=breizland, dc=bz";

//mot de passe administrateur
 $rootpw="secret";

//connexion à la base
 $result=ldap_connect($server);
 if ($result==1)
{
        ldap_bind($result,$rootdn,$rootpw);
}

//Enregistrement d'une entrée dans la base
echo"Enregistrement de Benjamin Hoarau\n";

$dn="dc=breizland, dc=bz";
$nom="Hoarau";
$prenom="Benjamin";

$info["cn"]=$nom." ".$prenom;
$info["sn"]=$nom;
$info["objectclass"]="person";
```

```
ldap_add($result,"cn=$nom $prenom,$dn",$info);
ldap_close($result);

echo "L'enregistement a réussi";
?>
```

Appelez ce fichier **ldap.php3**, vous pouvez le tester et vérifier avec **ldapsearch** que l'entrée a bien été saisie dans la base.