



IUT de Carcassonne

Département STID

Domaine d'Auriac

11000 Carcassonne

Jean-Charles Guimard

Licence professionnelle :

Traitement de l'Information Géographique

Promotion 2003-2004

Annexe 1 :

Compilation et installation de l'environnement de travail sous Linux

Serveur Web – Serveur FTP – SGBD

Stage effectué en alternance de septembre 2003 à août 2004 en vue de l'obtention de la licence professionnelle "Traitement de l'Information Géographique".

Au sein du
bureau d'études :



P.I.B.S.- Allée Nicolas Leblanc
BP 169
56005 VANNES CEDEX

Sommaire

| | |
|--|----|
| Introduction | 2 |
| 1 – Pré-installation | 3 |
| 1.1 – Préparation de l'espace de travail | 3 |
| 1.2 – Les sources utiles | 5 |
| 2 – Compilation et installation des logiciels | 7 |
| 2.1 – Compilation et installation du serveur Web | 7 |
| 2.1.1 – Le logiciel Apache2 | 7 |
| 2.1.2 – Le langage de scripts PHP | 8 |
| 2.2 – Compilation et installation du SGBD-R | 9 |
| 2.2.1 – La librairie Proj : | 9 |
| 2.2.2 – La librairie Readline... : | 9 |
| 2.2.3 – La librairie GEOS... : | 9 |
| 2.2.4 – Le logiciel PostgreSQL/PostGIS | 10 |
| 2.2.5 – Interaction serveur Web - SGBD | 11 |
| 2.3 – Configuration du serveur ftp | 12 |
| 2.4 – Compilation et Installation de MapServer | 15 |
| 2.4.1 – La librairie GD | 15 |
| 2.4.2 – La librairie Proj | 16 |
| 2.4.3 – La librairie GDAL/OGR | 16 |
| 2.4.4 – La librairie LibCurl | 16 |
| 2.4.5 – Le logiciel MapServer | 17 |
| 2.4.6 – Le module PHP/MapScript | 18 |

Introduction

Ce document présente la mise en place de l'environnement sous Linux pour les compilations et installations du serveur Web, du serveur FTP, et du SGBD.

On suppose donc que le système d'exploitation Linux est déjà installé sur la machine. Si ce n'est déjà fait, on pourra effectuer l'installation de Linux en suivant la documentation de Frédéric Garel. Pour le présent document, la distribution de Linux utilisée est la Knoppix. A priori, on peut suivre ce document pour n'importe quelle distribution.

Les solutions libres retenues sont :

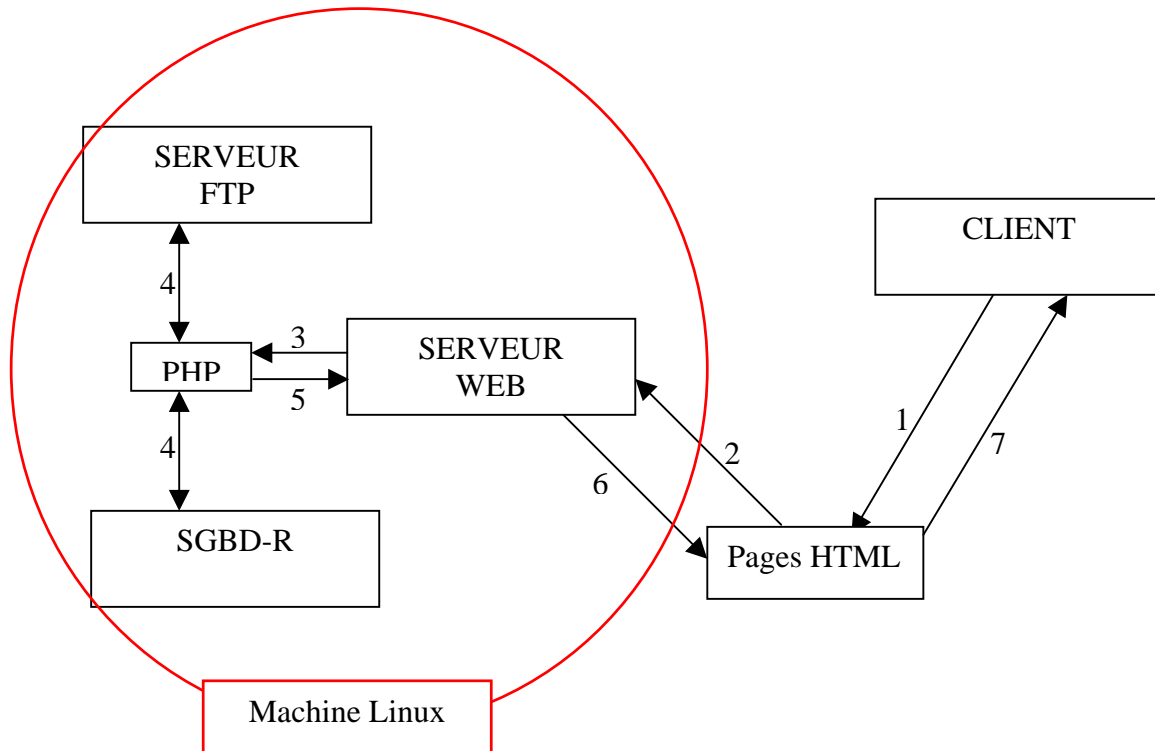
- Apache 2 pour le serveur Web : <http://httpd.apache.org>
- PostgreSQL / PostGIS pour le SGBD-R : <http://www.postgresql.org/>
- Proftpd pour le serveur FTP : <http://www.proftpd.org/>

Pour la suite du document, concernant les parties d'exécution de commandes, l'utilisateur par défaut est "jc". Lorsqu'on doit passer en super-utilisateur: root, il faut exécuter : su, suivi du mot de passe. Ce changement de "jc" à "root" se fera par **su root**, et de "root" à "jc" par **su jc**.

1 – Pré-installation

1.1 – Préparation de l'espace de travail

Avant de commencer, il faut d'abord comprendre le système qu'on souhaite mettre en place :

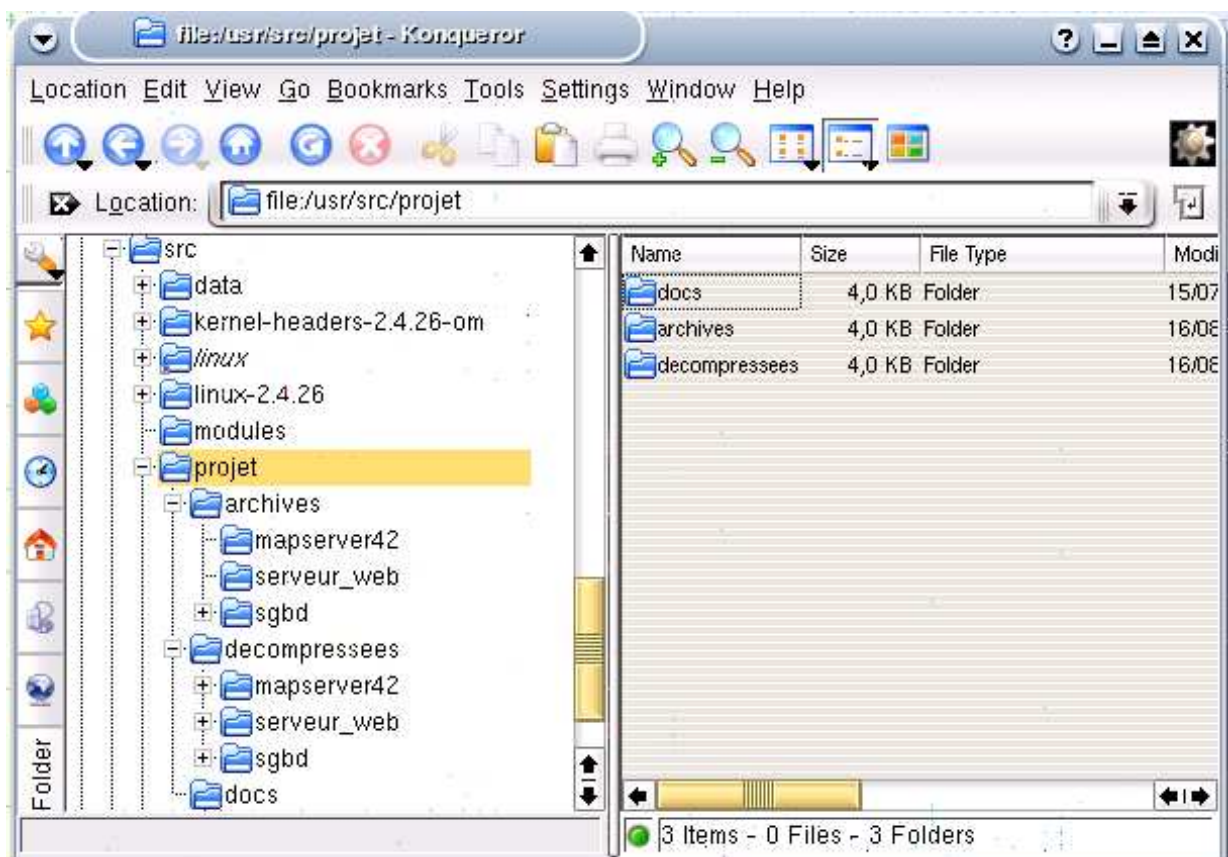


En règle générale, les personnes qui sont sous linux, connaisse windows. Afin que ces gens ne soit pas trop perdu dans les répertoires de notre os, voici l'arborescence d'un système UNIX classique (ex :Linux) :

- **/bin** contient les binaires fondamentaux à la gestion de Linux. On y retrouve les commandes systèmes.
- **/dev** contient des fichiers sur tout ce qui est matériels. En gros un fichier correspond à un matériel.. En fait /dev/hda1 correspond au premier disque dur et à la première partition
- **/etc** contient tous les fichiers de configuration de linux. On y retrouve smb.conf qui sert à configurer Samba.
- **/home** est le répertoire qui contient les répertoires des utilisateurs. Par exemple un utilisateur inater aura un répertoire /home/inater. Quand cet utilisateur se connectera, il ira directement dans ce répertoire.

- **/sbin** contient les binaires du système. On y trouve par exemple la commande *shutdown* qui permet d'arrêter l'ordinateur.
- **/tmp** est un répertoire de temporaire
- **/lost+found** est comme sont l'indique ce qui est perdu et retrouver. En faite dans ce répertoire atterrissent les fichiers qui ont endommagé et qui non plus de chemin.
- **/usr** contient grosso modo tout ce qui concerne les binaires utiles à tous les utilisateurs et quelques commandes d'administration.
- **/usr/local** est un répertoire qui sert à installer ses logiciels. C'est un peu comme un 'Programme File' chez windows.
- **/var/spool** est le répertoire des fichiers qui servent de file d'attente. Par exemple, les files d'attente du mail ou des imprimantes

Pour notre projet, on va créer l'arborescence suivante dans le répertoire `/usr/src` :



C'est à dire qu'on exécute les commandes (utilisateur par défaut) :

```
- cd /usr/src/
- mkdir projet
- cd projet
- mkdir archives
- mkdir decompressees
- mkdir docs
- cd archives
- mkdir mapserver42
- mkdir serveur_web
- mkdir sgbd
- cd ../decompressees
- mkdir mapserver42
- mkdir serveur_web
- mkdir sgbd
```

1.2 – Les sources utiles

Il faut à présent récupérer les sources des logiciels qu'on souhaite installer. Pour le moment on ne va pas expliquer à quoi sert chaque source téléchargée. Cela sera fait par la suite.

- o Sources liées au serveur Web :

→ Apache2 : <http://apache.crihan.fr/dist/httpd/httpd-2.0.50.tar.gz>

→ PHP5 : <http://fr.php.net/distributions/php-5.0.0.tar.gz>

On les copie dans le répertoire `/usr/src/projet/archives/serveur_web`, et on les décompresse dans `/usr/src/decompressees/serveur_web` :

```
- cd /usr/src/projet/archives/serveur_web
- tar -xvzf httpd-2.0.50.tar.gz -C ../../decompressees/serveur_web
- tar -xvzf php-5.0.0.tar.gz -C ../../decompressees/serveur_web
```

- o Sources liées au serveur FTP

cf. 2.3 : Configuration du serveur FTP

- o Sources liées au SGBD :

→ PostgreSQL : <ftp://ftp.fr.postgresql.org/v7.4.3/postgresql-7.4.3.tar.gz>

→ PostGIS : <http://postgis.refrations.net/postgis-0.8.2.tar.gz>

→ Proj : <ftp://ftp.remotesensing.org/pub/proj/proj-4.4.8.tar.gz>

→ Readline : <ftp://ftp.gnu.org/gnu/readline/readline-4.3.tar.gz>

→ Geos : <http://geos.refrations.net/geos-1.0.0.tar.bz2>

On les copie dans le répertoire /usr/src/projet/archives/svdb, et on les décompresse dans /usr/src/decompressees/svdb :

```
- su jc
- cd /usr/src/projet/archives/svdb
- tar -xvzf proj-4.4.8.tar.gz -C ../../decompressees/svdb
- tar -xvzf readline-4.3.tar.gz -C ../../decompressees/svdb
- tar -xvzf geos-1.0.0.tar.bz2 -C ../../decompressees/svdb
- tar -xvzf postgresql-7.4.3.tar.gz -C ../../decompressees/svdb
- tar -xvzf postgres-0.8.2.tar.gz -C ../../decompressees/svdb/postgresql-7.4.3/contrib
```

o Sources liées à MapServer :

- Mapserver : <http://cvs.gis.umn.edu/dist/mapserver-4.2.2.tar.gz>
- GD : <http://www.boutell.com/gd/http/gd-2.0.28.tar.gz>
- Proj : <ftp://ftp.remotesensing.org/pub/proj/proj-4.4.8.tar.gz>
Si on n'a pas suivi l'étape précédente du SGBD
- Gdal : <ftp://ftp.remotesensing.org/gdal/gdal-1.2.1.tar.gz>
- Curl : <http://curl.haxx.se/download/curl-7.12.1.tar.gz>

On les copie dans le répertoire /usr/src/projet/archives/mapserver42, et on les décompresse dans /usr/src/decompressees/mapserver42 :

```
- su jc
- cd /usr/src/projet/archives/mapserver42
- tar -xvzf gd-2.0.28.tar.gz -C ../../decompressees/mapserver42
- tar -xvzf proj-4.4.8.tar.gz -C ../../decompressees/mapserver42
- tar -xvzf gdal-1.2.1.tar.gz -C ../../decompressees/mapserver42
- tar -xvzf curl-7.12.1.tar.gz -C ../../decompressees/mapserver42
- tar -xvzf mapserver-4.2.2.tar.gz -C ../../decompressees/mapserver42
```

2 – Compilation et installation des logiciels

Après avoir décompressé les sources, l'étape suivante consiste à compiler et installer les logiciels, à savoir le serveur Web et le SGBD. Puis il faut également configurer ceux-ci, ainsi que le serveur FTP.

2.1 – Compilation et installation du serveur Web

Documentations :

- Documentations générale :
 - Apache2 : <http://httpd.apache.org/docs-2.0/>
 - PHP5 : <http://fr2.php.net/manual/fr/>
/usr/src/projet/docs/php_manual_fr.tar.gz
- Documentations pour l'installation :
 - Apache2 : <http://httpd.apache.org/docs-2.0/install.htm>
 - PHP5 : rubrique Installation (I.3)
Installation sous Unix (manual/fr/install.unix.php)
 - Apache2 + PHP5 : www.php.net/manual/fr/install.apache2.php

Il faut peut-être, au préalable, désinstaller la version antérieure de Apache. Dans notre cas on désinstalle la version Apache1.3 :

```
- apt-cache remove apache
```

2.1.1 – Le logiciel Apache2

Apache2 est un serveur Web : un logiciel permettant à des clients d'accéder à des pages Web, c'est-à-dire en réalité des fichiers au format HTML à partir d'un navigateur (aussi appelé browser) installé sur leur ordinateur distant.

Une fois qu'on est assurée qu'il n'a pas de serveur Apache présent sur la machine, **on compile et installe Apache2** :

```
- cd /usr/src/projet/serveur_web/httpd-2.0.50
- ./configure --enable-so
- make
- su root
- make install
```


A ce stade, Apache est installé et se trouve dans `/usr/local/apache2`. On ajoute à la variable `PATH`, le chemin vers le binaire de Apache2. Pour cela on édite le fichier `/etc/profile`, et ajoute la ligne `PATH : /usr/local/apache2/bin` :

```
- su root (si on a quitté le root avant)
- vi /etc/profile
- Ajouter /usr/local/apache2/bin au PATH
```

2.1.2 – Le langage de scripts PHP

PHP est un langage de scripts exécuté du côté serveur. On peut inclure un script php au sein d'une page html (contrairement aux autres scripts CGI). On peut l'interfacer facilement avec des SGBD, notamment PostgreSQL.

On passe à la **compilation et l'installation de PHP5**. Pour le moment, on le laisse en tant que module. On verra plus tard comment le compiler pour qu'il fonctionne en CGI. Cela est expliqué dans la partie consacrée à MapServer. Mais c'est un bon exercice que de voir la différence entre les compilations... Si on veut le compiler une seule fois, alors voir la re-compilation de PHP dans la partie MapServer – PHP/Mapscript. Pour le moment on le laisse en module Apache :

```
- su jc
- cd /usr/src/projet/decompressees/serveur_web/php-5.0.0
- ./configure --with-apxs2 = /usr/local/apache2/bin/apxs
- make
- su root
- make install
- cp php.ini-dist /usr/local/lib/php.ini
```

On édite le fichier de configuration du serveur Apache2, `httpd.conf` : `/usr/local/apache2/conf/httpd.conf`. On ajoute les lignes suivantes au bon endroit dans le fichier :

```
- LoadModule php5_module module/libphp5.so
- AddType application/x-httpd-php .php
```

On a un serveur Web dynamique! Les commandes de bases pour le démarrer et l'arrêter :

```
- su root
- apachectl start
- apachectl stop
```

2.2 – Compilation et installation du SGBD-R

Documentations :

- o Documentation générale :
 - PostgreSQL : `/usr/src/projet/docs/postgresql-7.4.2-A4.pdf`
<http://www.postgresql.org/docs/7.4/interactive/>
 - PostGIS : `/usr/src/projet/docs/postgis.pdf`
<http://postgis.refractory.net/docs/>
- o Documentation pour l'installation :
 - Installation de PostGIS sous Knoppix3.3 – auteur : David Techer
<http://techer.pascal.free.fr/postgis/knoppix/installationknoppixpostgis.pdf>

Avant de compiler PostgreSQL/PostGIS, on s'occupe des bibliothèques nécessaires. A savoir Proj, Readline et GEOS.

2.2.1 – La bibliothèque Proj :

Pour la conversion à la volée de la projection cartographique.

```
- su jc
- cd /usr/src/projet/decompressees/sghd/proj-4.4.8
- ./configure
- make
- su root
- make install
```

2.2.2 – La bibliothèque Readline... :

Elle permet à l'interpréteur de PostgreSQL : *psql* de se souvenir des commandes tapées. Elle est comparable au *doskey* sous DOS.

```
- su jc
- cd /usr/src/projet/decompressees/sghd/readline-4.3
- ./configure
- make
- su root
- make install
```

2.2.3 – La bibliothèque GEOS... :

Elle permet d'ajouter des fonctionnalités spatiales supplémentaires à PostGIS telles que *Within()*, *Intersects()*...

```
- su jc
- cd /usr/src/projet/decompressees/sghd/geos-1.0.0
- ./configure
- make
- su root
- make install
```

2.2.4 – Le logiciel PostgreSQL/PostGIS

PostgreSQL est un SGBD libre. De plus c'est un SGBD Relationnel-Objet. C'est un serveur de base de données transactionnel très puissant. PostGIS ajoute l'accès pour les objets géographiques d'une base de données relationnelles PostgreSQL. En effet, PostGIS utilise le serveur PostgreSQL pour disposer d'une base de données utilisée pour un système d'information géographique (SIG), exactement comme une extension ESRI SDE ou Oracle Spatial.

Les 3 bibliothèques nécessaires sont maintenant installées. Il reste à **compiler PostgreSQL et PostGIS**. On ne se pose pas de questions, et on tape une à une les commandes suivantes :

```
- su jc
- cd /usr/src/projet/decompressees/sghd/postgresql-7.4.3
- LDFLAGS = -lstdc++ ./configure --with-libs=/usr/local/lib --with-
  includes=/usr/local/include --enable-multibyte --with-CXX --prefix=/usr -
  sysconfdir=/etc --docdir=/usr/doc/postgresql-$version
  Oui c'est bien une commande! Comme toutes les autres qui suivent
- make
- cd contrib/postgis-0.8.2
- perl -i.bak -pe 's/USE\_GEOS=0/USE\_GEOS=1/g'
  /usr/src/projet/decompressees/sghd/postgresql-7.4.3/contrib/postgis-
  0.8.2/Makefile
- perl -i.bak -pe 's/USE\_PROJ=0/USE\_PROJ=1/g'
  /usr/src/projet/decompressees/sghd/postgresql-7.4.3/contrib/postgis-
  0.8.2/Makefile
- make
- su root
- make install
- exit
- perl -i.bak -pe 's/\$libdir/\usr/lib/postgresql/g'
  /usr/src/projet/decompressees/sghd/postgresql-7.4.3/contrib/postgis-
  0.8.2/postgis.sql
- cp postgis.sql /home/jc
- cp spatial_ref_sys.sql /home/jc
- cd ../../
- su root
- make install
- exit
- initdb -D /usr/src/data
- su root
- perl -i.bak -pe 's/\#tcpip\_socket \= false/tcpip\_socket \= true/g'
  /usr/src/data/postgresql.conf
- perl -i.bak -pe 's/\#port \= 5432/port \= 5432/g'
  /usr/src/data/postgresql.conf
- exit
```

C'est peut-être une bonne idée de redémarrer l'ordinateur, si on peut le faire...

Important : pour démarrer, arrêter le SGBD, il ne faut pas être en super utilisateur (root)! Donc en utilisateur normal. Par exemple pour démarre/arrêter le service, sous l'utilisateur *jc* :

```
- su jc
- pg_ctl start -D /usr/local/src
- pg_ctl stop -D /usr/local/src
```

N.B : Création d'une base de données "essai" :

```
- pg_ctl start -D /usr/src/data
- createdb essai
```

On lui donne l'extension spatiale :

```
- createlang plpgsql essai
- psql -d essai -f /home/jc/postgis.sql
- psql -d essai -f /home/jc/spatial_ref_sys.sql
```

2.2.5 – Interaction serveur Web - SGBD

On a pour le moment, le serveur Web et le SGBD d'installé sur la machine Linux. Mais rien ne fait l'interaction entre les 2. En effet nous n'avons pas compilé PHP pour qu'il puisse utiliser les fonctionnalités de la librairie pgsql. On va donc tout simplement recompiler PHP :

```
- su jc
- cd /usr/src/projet/decompresses/serveur_web/php5.0.0
- make clean
- ./configure --with-pgsql=/usr --with-apxs2=/usr/local/apache2/bin/apxs
- make
- su root
- make install
- cp php.ini-dist /usr/local/lib/php.ini (peut-être pas nécessaire??)
```

Maintenant, sans qu'il soit nécessaire de relancer Apache, on peut écrire des scripts php en utilisant les fonctions pgsql de php (*pgsql_connect...*).

2.3 – Configuration du serveur ftp

Dans un premier temps, on a retenu le serveur *proftpd*. Mais finalement pour des difficultés de configuration liées à l'ajout de comptes utilisateurs, on s'est tourné vers *pure-ftpd*.

Documentations :

- Installation : <http://lea-linux.org/reseau/pureftpd.html>
- Script de démarrage (*/etc/init.d* et *rc-update*):
 - <http://www.debian-fr.org/article.php?page=viewarticle&article=4>
 - <http://www.supinfo-projects.com/fr/2004/pureftpd/3/>

Il faut s'assurer qu'il n'y ai pas déjà un serveur ftp qui tourne. Pour nous, il y a déjà *proftpd*. On le désinstalle :

```
- apt-get remove proftpd
```

On peut l'installer à partir des sources ou bien à partir de la commande *apt-get*. Les sources : <ftp://ftp.pureftpd.org/pub/pure-ftpd/releases/pure-ftpd-1.0.20.tar.gz>

On fait l'installe à partir de la commande *apt-get*. Avant cela, on ajoute 2 sites à la source-list (*/etc/apt/sources.list*) :

```
- deb http://debian.cobolt.net/pure-ftpd-test main
- deb http://debian.cobolt.net/pure-ftpd-sid main
```

Ensuite on met à jour cette modification :

```
- apt-get update
```

Installation de *pure-ftpd* :

```
- apt-get install pure-ftpd
```

Un ensemble de fenêtres (*Wizards*) vont apparaître pour la configuration du serveur : on l'utilise en mode autonome (*standalone*, pas avec *xinet*).

On peut dès à présent utiliser le serveur ftp, en le démarrant avec la commande :

```
- pure-ftpd -A -b -B -c 15 -C 2 -E -j -l puredb:/etc/pure-ftpd/pureftpd.pdb
-R -u 1 -X -H
```

On copie cette commande dans un script shell, pour n'avoir plus à retaper cette commande et juste à exécuter le script.

Pour stopper le serveur ftp, c'est un peu plus compliqué : il faut le PID du processus pour pouvoir le « tuer » avec la commande kill :

```
- ps aux |grep pure-ftpd
- kill n° processus
```

Tel qu'il est configuré, le serveur ftp n'accepte des connexions d'utilisateurs enregistrés et pas d'anonymes.

On crée sur le système un groupe et un utilisateur :

```
- groupadd ftpgroup
- useradd -g ftpgroup -d /dev/null -s /etc/ftpuser
```

On peut désormais ajouter des utilisateurs virtuels qui sont propres au serveur ftp (un peu comme les utilisateurs sur postgresql). Ces utilisateurs n'ont pas de compte sur le système, on leur alloue un espace sur le disque simplement :

```
- pure-pw useradd user1 -u ftpuser -d /home/ftpusers/user1
- pure-pw mkdb
```

La première commande ajoute l'utilisateur user1, la racine de son arborescence est /home/ftpusers/user1, ce répertoire n'existe pas mais sera créé automatiquement lors de sa 1^{ère} connexion (option -j lors du démarrage du serveur...). La deuxième commande enregistre l'utilisateur dans la base de données.

Quelques commandes utiles :

→ Changer le mot de passe d'un utilisateur :

```
- pure-pw passwd nom_user
- pure-pw mkdb
```

→ Détruire un compte utilisateur :

```
- pure-pw userdel nom_user
```

→ Visionner les paramètres de configuration d'un utilisateur :

```
- pure-pw show nom_user
```

→ Lister les personnes connectées au serveur FTP :

```
- pure-ftpwho
```

Lors du démarrage du serveur, il y a un grand nombre d'options possibles. On peut les connaître avec la commande habituelle : *pure-ftpd -help*. Voici la liste de quelques-unes de ces options :

- -A : "chroot" tout les utilisateurs sauf root.
- -b : ignore certains standards pour permettre à certains clients (Internet Explorer !!!) de fonctionner malgré leurs bogues...
- -B : Ordonne au serveur de démarrer en arrière plan
- -c 5 : limitation du nombre de client simultané sur votre serveur à 5 (défaut = 50)
- -C 2 : limite le nombre de connexion simultanées de chaque utilisateur (provenant de la même adresse IP) à 2
- -E : autorise seulement la connexion d'utilisateurs enregistrés (pas d'anonymes)
- -e : autorise seulement la connexion d'utilisateurs anonymes
- -j : création automatique d'un répertoire utilisateur si celui qui se connecte n'en a pas déjà un.
- -l puredb:/etc/pureftpd.pdb : specifie le chemin vers la base de données des utilisateurs virtuels.
- -R : interdit l'utilisation de la commande chmod par les clients
- -u 1 : n'accepte pas les uids inférieur à 1 (root en l'occurrence)
- -X : interdit aux utilisateurs l'accès aux fichiers ou répertoires cachés (ceux qui commencent par un point)
- -H : pas de résolution DNS (à utiliser si connexion très lente)

2.4 – Compilation et Installation de MapServer

Documentations :

- Documentations générales :
 - Site officiel : <http://mapserver.gis.umn.edu>
 - Traduction française : <http://fa.vdb.free.fr/MapServer/doc/>
- Documentations pour l'installation :
 - <http://mapserver.gis.umn.edu/doc40/unix-install-howto.html>
 - README.CONFIGURE *dans les sources de MapServer*

On commence par compiler et installer les librairies nécessaires à MapServer. A savoir GD, Proj, GDAL et LibCurl. Si on n'a déjà installé la librairie Proj lors de l'installation du SGBD PostgreSQL alors on peut passer l'étape concernant Proj.

2.4.1 – La librairie GD

MapServer utilise la bibliothèque libre GD library pour fabriquer ses sorties. La dernière version gd-2.0.28 autorise à nouveau les sorties au format GIF, mais il est recommandé l'emploi du format PNG de meilleure qualité.

| | |
|---|--|
| <pre>- su jc - cd /usr/src/projet/decompressees/mapserver42/gd-2.0.28 - ./configure</pre> | |
| <pre>→ Vérification :</pre> | <pre>Support for PNG library yes JPEG yes Freetype2.x yes Xpm yes Pthreads yes</pre> |
| <pre>- make - su root - make install</pre> | |

Les librairies ont été installées dans /usr/local/lib.

2.4.2 – La librairie Proj

Pour la conversion à la volée de la projection cartographique

```
- su jc
- cd /usr/src/projet/decompressees/s gdb/proj-4.4.8
- ./configure
- make
- su root
- make install
```

Les librairies ont été installées dans /usr/local/lib

2.4.3 – La librairie GDAL/OGR

GDAL - Geospatial Data Abstraction Library: Pour l'utilisation en entrée/sortie de plusieurs formats de fichiers RASTER.

OGR Simple Features Library: Pour l'utilisation en entrée/sortie de différents fichiers au format VECTORISES.

```
- su jc
- cd /usr/src/projet/decompressees/mapserver42/gdal-1.2.1
- ./configure -without-hdf4
```

→ si on ne met pas le without-hdf4 : erreur lors du make, il n'arrive pas à trouver hdf4.c ou un truc dans le genre.

→ ce qui est important à vérifier (je pense) c'est les quelques lignes :

```
libtool      yes
ecw          no    (à voir...)
postgresql   yes
enable ogr   yes
```

```
- make
- su root
- make install
```

Installation faite dans /usr/local/lib/bin

2.4.4 – La librairie LibCurl

Pour les connexions WMS/WFS en mode client.

```
- su jc
- cd /usr/src/projet/decompressees/mapserver42/curl-7.12.1
- ./configure
- make
- su root
- make install
```

Une fois que ces librairies sont installées, on peut passer à la compilation et l'installation de MapServer.

2.4.5 – Le logiciel MapServer

MapServer est un logiciel libre développé par l'université de Minnesota permettant la publication cartographique sur INTERNET.

Info : Pour recompiler MapServer : `make clean`, puis `./configure` en rappelant les *with* et *enable* : le fait de relancer `./configure`, prépare le Makefile. Si on a compilé une première fois et installé avec la commande `./configure --with-gd` (puis `make` et `make install`) par exemple, alors on a `mapserv` qui arrive à faire l'association avec la librairie `gd`. Si ensuite on souhaite recompiler parce qu'on a oublié une librairie et qu'on lance `./configure --with-gdal`, alors on aura cette fois `mapserv` associé à la librairie `gdal`, mais plus à `gd`. **Il faut retaper à chaque fois toute la ligne de configuration : `./configure --with-.. --enable-.. ...`**

Passons à la compilation et installation de MapServer4.2 :

```
- su jc
- cd /usr/src/projet/decompressees/mapserver42/ mapserver-4.2.2
- ./configure --with-gd --enable-force-freetype1 --with-
  php=/usr/src/projet/decompressees/serveur_web/php-5.0.0/ --with-proj --
  with-wmsclient --withgdal --with-ogr --with-postgis
```

→ Il y a peut-être une erreur au sujet de `php`, cf. installation de `php/mapscript` ci-après. Si on pense ne pas avoir besoin du module `PHP/MapScript`, alors il n'est pas nécessaire d'ajouter `--with-php`. Dans ce cas on n'aura pas à copier le fichier `php_mapscript.so...`

```
- make
- cp mapserv /usr/local/apache2/cgi-bin/
- cp mapscript/php3/php_mapscript.so /usr/local/lib/
```

→ dans le cas où le chemin des extensions spécifié dans le `php.ini` est `/usr/local/lib`

```
- PAS de make install !!
```

Pour vérifier que l'installation a réussie, soit dans le répertoire `/usr/src/projet/decompressees/mapserver42/` ou soit dans le répertoire `/usr/local/apache2/cgi-bin/`, on tape la commande :

```
- mapserv -v
```

Résultat à l'écran de la console :

```
MapServer version 4.2.2 OUTPUT=GIF OUTPUT=PNG OUTPUT=JPEG
OUTPUT=WBMP SUPPORTS=PROJ SUPPORTS=FREETYPE SUPPORTS=WMS_SERVER
SUPPORTS=WMS_CLIENT INPUT=EPPL7 INPUT=POSTGIS INPUT=OGR INPUT=GDAL
INPUT=SHAPEFILE
```

On peut également dans le navigateur, taper l'url suivante :
<http://localhost/cgi-bin/mapserv>

Résultat à l'écran du navigateur :

```
No query information to decode. QUERY_STRING is set, but empty.
```

2.4.6 – Le module PHP/MapScript

Le module PHP/MapScript est un module PHP chargeable dynamiquement qui rend disponible l'accès aux fonctions et classes de MapScript de MapServer dans un environnement PHP3/PHP4.

Documentations :

- o Documentations générales :
 - MapServer : <http://mapserver.gis.umn.edu/doc42/phpmapscript-class-guide.html>
 - MapTools : http://www.maptools.org/php_mapscript/
- o Documentations pour l'installation :
 - HowTo : <http://mapserver.gis.umn.edu/doc42/phpmapscript-install-howto.html>
 - Traduction: http://fa.vdb.free.fr/MapServer/doc/phpmapscript-install-howto_fr.html

On va commencer par mettre PHP en CGI (et non plus en module Apache), on édite le fichier *httpd.conf* (*/usr/local/apache2/conf/httpd.conf*) :

```
- Commenter la ligne : AddModule mod_php5.so
- Ajout d'une action : Action application/x-httpd-php "/cgi-php/php.exe"
- Un scriptAlias :      ScriptAlias /cgi-php/ "/usr/local/bin"
```

On édite maintenant le fichier *php.ini* (*/usr/local/lib/php.ini*) :

```
- enabled_dl() ON
- Chemin des extensions : /usr/local/lib
```

On a vu lors de la compilation de MapServer qu'il y avait peut-être un problème avec PHP. En fait pour résoudre celui-ci il faut compiler PHP pour que MapServer soit également compatible regex... L'erreur semble être une incompatibilité entre les versions...

Lorsque PHP est en module apache (en non en cgi), les extensions comme ftp, dbase... sont prises en comptes directement par le module. En revanche, en mode CGI, il faut recompiler PHP en spécifiant qu'on souhaite utiliser telle et telles extensions :

```

- su jc
- cd /usr/src/projet/decompressees/serveur_web/php-5.0.0
- make clean
- ./configure -with-gd=/usr/local -with-pgsql -with-regex=system -enable-
  dbase -enable-ftp --with-zlib
- make
- su root
- make install

```

On relance le serveur Apache. Pour voir que le module PHP/MapScript est bien installé, on crée une page php : *test_phpmapscript.php* qu'on sauvegarde dans l'arborescence du *DocumentRoot*, par exemple à la racine de *htdocs* (*/usr/local/apache2/htdocs*). La page *test_phpmapscript.php* :

```

<?
dl('php_mapscript.so') ;
phpinfo();
?>

```

Dans le navigateur web, on ouvre l'url http://localhost/test_phpmapscript.php, et on doit avoir une partie consacrée au module PHP/MapScript :

MapScript

| | |
|------------------------------|--|
| MapServer Version | MapServer version 4.2.2 OUTPUT=GIF OUTPUT=PNG OUTPUT=JPEG OUTPUT=WBMP SUPPORTS=PROJ SUPPORTS=FREETYPE SUPPORTS=WMS_SERVER SUPPORTS=WMS_CLIENT INPUT=EPPL7 INPUT=POSTGIS INPUT=OGR INPUT=GDAL INPUT=SHAPEFILE |
| PHP MapScript Version | (\$Revision: 1.194.2.5 \$ \$Date: 2004/07/26 14:56:39 \$) |