

Développement Web avancé NFA017



J-F Dazy
J-F Berger

Objectifs du cours

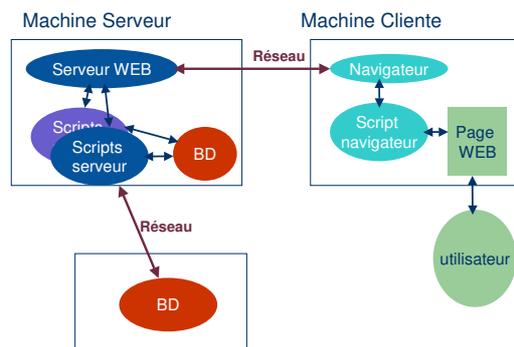
- Conception et administration de sites web dynamiques
 - Coté client (*rappels*)
 - Coté serveur (ici php)
- Notions de sécurité
 - Cookies et sessions
- Exploitation d'une base de données pour la production dynamique de documents
- Pré-requis supposés : (NFA016)
 - Html et Javascript (bases)
 - Css

- Sources du cours
 - <http://idf.pleiad.net>

URL utiles

- Php : <http://fr.php.net/docs.php>
- Mysql : <http://www-fr.mysql.com/>
- Postgresql : <http://www.postgresqlfr.org/>
- Plateformes
 - WampServer : <http://www.wampserver.com/>
 - Windows, Apache, Mysql, Php
 - On trouve bien sur Lamp (Linux), Mamp(MacOS)
 - Easyphp : <http://www.easyphp.org/?lang=fr>
 - Analogue à wamp, moins polyvalent peut être...
- Outils
 - Éditeurs : large choix...ici
 - html-kit : <http://www.chami.com/html-kit/>
 - Phpmyadmin : http://www.phpmyadmin.net/home_page/index.php

Composantes d' une application WEB (1)



Composantes d'une application WEB (2)

- Scripts coté serveur:
 - modules du serveur ou programmes externes (CGI)
 - VB.NET, PHP, ASP, ASP.NET, JSP (compilées en servlet), CFM, java, PERL, Python, C, ...
 - Nous utiliserons PHP4 et PHP5
- Scripts coté client
 - VBscript, javascript (Jscript microsoft), perlscript, applets java

Cnam51-2008

5

Architecture client-serveur web (rappels)

• Le protocole http

Transfert de messages avec en-tête décrivant le contenu du message (codage type MIME)

Permet un transfert de fichier (html) localisé par une URL entre un navigateur (client) et un serveur web (httpd)

RFC1945 (version 1.0,1.1)

http 1.0: Protocole sans état ☹ : pas d'informations gardées entre deux transactions

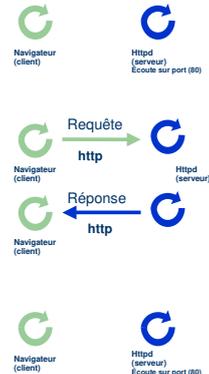
Le client

ouvre une connexion
envoie une requête http au serveur

Le serveur

renvoie une réponse HTTP, qui contient d'une manière générale la ressource qui a été demandée par le client.
ferme la connexion.

Evolution avec http1.1



Cnam51-2008

6

Les messages http : format d'une requête

- 1ère ligne: Trois parties
 - La méthode (en MAJUSCULE)
 - GET : obtenir une ressource
 - HEAD : tester une ressource
on obtient des infos sur la ressource (les en-têtes (date péremption, taille, existence, ...) mais pas la ressource)
 - POST : envoyer des données au serveur
corps de message comprenant le type, la taille des données et les données à traiter par le serveur
 - Chemin (request-URI): url de la ressource ou nom du programme serveur (POST)
 - Version du protocole : HTTP/x.x
 - En 1.1: connexion persistante :
le client envoie des requêtes successives terminées dont la dernière comporte l'en-tête *connection: close* et le serveur ferme la connexion quand il traite cette dernière
Avantage: économie de ressources (ouvrir et fermer une connexion TCP/IP est consommateur!)
- Des lignes d'en-tête (optionnelles): *Header-Name:value*

Cnam51-2008

7

Les messages http : Exemple de requête GET

- GET http://www.cnam.fr HTTP/1.0
- Accept : text/html
If-Modified-Since : Saturday, 15-January-2000 14:37:11 GMT
User-Agent : Mozilla/4.0 (compatible; MSIE 5.0; Windows 95)
Exemple d'en-tête: (il y en a 46 en http1.1)
- From:** donne l'e-mail de la personne contrôlant le navigateur. Cela peut poser des problèmes de respect de la vie privée
Referer: URL de l'objet qui amène la requête (URL de la page dans laquelle il y a le lien)
User-Agent: l'identifiant du navigateur. Sert pour adapter la réponse au navigateur
Authorization: permet à un client de s'authentifier auprès du serveur
If-Modified-Since: permet de faire des GET conditionnels

La méthode GET peut aussi être utilisée pour transmettre des données de petite taille :

GET http://www.cnam.fr/formget.php?nom=Berger&voisins=bbbb HTTP/1.0

Cnam51-2008

8

Les messages http : Exemple de requête POST

- POST http://www.cnam.fr/formpost.php HTTP/1.0
Content-Type: application/x-www-form-urlencoded
Content-Length: 21
Ligne blanche
nom=Dazy&voisins=aaaa

Les messages http : Réponse⁽¹⁾

- La ligne initiale d'une réponse HTTP, appelée ligne de statut, est composée de trois parties: la version HTTP, le code statut de la réponse, et une phrase d'explication qui décrit le code statut.
 - On pourra par exemple avoir:
 - HTTP/1.0 200 OK pour indiquer le succès de l'opération ou:
 - HTTP/1.0 404 Not Found
 - Les status
 - 1xx indique un message d'information
 - 2xx indique un message de succès
 - 3xx redirige le client vers une autre URL
 - 4xx indique une erreur du côté du client
 - 5xx indique une erreur du côté du serveur

Les messages http : Réponse⁽²⁾

- **Les champs d'en-tête de la réponse:** il s'agit d'un ensemble de lignes facultatives permettant de donner des informations supplémentaires sur la réponse et/ou le serveur. Chacune de ces lignes est composée d'un nom qualifiant le type d'en-tête, suivi de deux points (:) et de la valeur de l'en-tête
- **Le corps de la réponse:** il contient le document demandé

Les messages http : Réponse exemple ⁽³⁾

HTTP/1.0 200 OK
Date : Mon, 20 feb 2006 14:37:12 GMT
Server : Microsoft-IIS/2.0
Content-Type : text/HTML
Content-Length : 1234
Last-Modified : Mon, 20 Feb 2006 08:25:13 GMT

CORPS de la réponse (html) (1234 caractères)

CGI Common gateway interface

CGI : Common gateway interface (interface de passerelle commune)(1)

- CGI : interface normalisée utilisée par les [serveurs HTTP](#) , indépendante de tout langage
- Définit comment transmettre la requête du serveur HTTP à un programme et comment récupérer la réponse générée.
- Ce programme doit être dans un répertoire spécial (selon configuration du serveur)

EXEMPLE : Pour traiter les formulaires HTML : l'utilisateur va choisir ou saisir des données, puis d'envoyer les données du formulaire en paramètre du script CGI.

Formulaire html

```
<form action='http://nom_du_serveur/cgi-bin/script.cgi' method=post ou get>
<input type=text name=nom>
<input type=text name=voisins>
<input type=submit value=envoi>
</form>
```

CGI : Common gateway interface (2)

l'appui sur le bouton envoi génère une requête http

méthode POST

```
http://nom_du_serveur/cgi-bin/script.cgi HTTP/1.0
Content-Type: application/x-www-form-urlencoded
Content-Length: 21
Ligne blanche
nom=Dazy&voisins=aaaa
ensemble des couples nom=valeur séparés par &
```

méthode GET

```
http://nom_du_serveur/cgi-bin/script.cgi?nom=Berger&voisins=bbbb HTTP/1.0
```

**La chaîne est la même , mais
envoyée et récupérée différemment**

CGI : Common gateway interface (3)

• GET ou POST?

- La méthode POST : pas de limite en taille (conseillée).
- La méthode GET est limitée en nombre de paramètres : la taille URL+paramètres doit être inférieure à 1Kb.
- La récupération des paramètres est faite
 - sur stdin pour la méthode POST (CONTENT_LENGTH donne la taille)
 - dans la variable d'environnement QUERY_STRING pour la méthode GET
- Le script CGI doit décoder les données , les traiter, et fournir un résultat (html) au serveur qui l'envoie au client.

CGI : Common gateway interface (4)

• Encodage :

- Les caractères non ASCII (ceux dont le code est supérieur à 128) sont remplacés par la chaîne de caractères `%xx` où `xx` représente le code ASCII du caractère au format hexadécimal.
- les caractères réservés sont également remplacés par leur valeur hexadécimale.
- le caractère espace est remplacé par le caractère `+`.

Cham51-2008

17

CGI : Que doit faire le script? (1)

- extraire l'information envoyée par le navigateur à l'aide du formulaire. La méthode dépend en fait de celle choisie dans le formulaire.

```
#!/usr/bin/perl
# les données sont envoyées par méthode GET
# donc on récupère les données dans la variable d'environnement QUERY_STRING
$buffer=$ENV{"QUERY_STRING"};
# on split la chaîne de données en des paires name=value
local(@champs) = split(/&/, $buffer);
local($donnees) = "";
# affichage du début du code HTML
printf STDOUT "Content-type: text/html\n\n"; #ligne blanche!!!!
printf STDOUT "<HTML><HEAD>";
printf STDOUT "<TITLE>Reponse au questionnaire</TITLE>";
printf STDOUT "</HEAD>";
printf STDOUT "<BODY BGCOLOR=#ffffff>";
printf STDOUT "<H1>Résultat du traitement de votre questionnaire</H1>";
printf STDOUT "<H2>Chaîne de données reçue par le programme</H2>";
printf STDOUT "QUERY_STRING <STRONG>%s</STRONG>", $buffer;
printf STDOUT "<H2>Liste des informations décodées</H2>";
printf STDOUT "<UL>"; printf STDOUT "<LI>";
```

Cham51-2008

18

CGI : que doit faire le script? (2)

• Décoder les données par couples

```
printf STDOUT "<H2>Liste des informations décodées</H2>";
printf STDOUT "<UL>"; printf STDOUT "<LI>";
# récupération et mise en forme des données
# on parcourt la liste des paires name=value
foreach $i (0 .. $#champs)
{
  # On convertit les plus en espaces
  $champs[$i] =~ s/\+/ /g;
  # On sépare chaque champ en une cle et sa valeur
  ($key, $val) = split(/=/, $champs[$i], 2);
  # On convertit les %XX de leur valeur hexadécimale en alphanumérique
  $key =~ s/%(..)/pack("c", hex($1))/ge;
  $val =~ s/%(..)/pack("c", hex($1))/ge;
  # on affiche le résultat (ou on traite...)
  printf STDOUT "<LI><STRONG>%s</STRONG>%s\n", $key, $val;
}
printf STDOUT "</UL>";
printf STDOUT "</BODY>";
printf STDOUT "</HTML>";
```

Avec la méthode POST :

```
read(STDIN, $buffer, $ENV{"CONTENT_LENGTH"});
La variable CONTENT_LENGTH contenant la longueur de la chaîne constituée des couples nom=valeur.
```

Cham51-2008

19

CGI évolution

Conclusion :

1. faut aimer
- cette technologie requiert qu'une nouvelle copie du programme soit lancée à chaque requête, elle surcharge rapidement les serveurs
1. D'où, intégration du langage de script dans le serveur HTTP sous forme de modules, propre au serveur HTTP.
 - C'est le cas avec [Apache](#) pour des langages couramment employés pour les applications Web, comme [PHP](#) (configuré en module), Perl (mod_perl), Python (mod_python) ou Java.

Cham51-2008

20

PHP Hypertext Preprocessor

Php ... un peu d'histoire

- Rasmus Lerdorf en 1994: crée une bibliothèque de scripts perl, puis C pour son usage personnel.
 - Son objectif:
 - Communiquer avec des BD
 - Générer des applications dynamiques pour le web
- Il publie son code sous Licence GNU Juin 95 :
Personnal HomePage Tools
 - Améliore, développe d'autres outils PHP/FI (Forms Interpreter) 1997 Version bêta
 - Variables de type perl,
 - Interprétation automatique des variables de formulaire, syntaxe intégrable dans html

PHP suite...

- Réécriture complète (Andi Gutmans et Zeev Suraski Juin 1998) d' où **PHP3.0 Hypertext Preprocessor**
- Réécriture du moteur interne PHP (Zend Engine) en 1999, sortie **PHP4.0** en 2000
- **PHP 5** est sorti en Juillet 2004 (basé sur Zend Engine 2.0) stable maintenant
 - 40% des serveurs... (dont 94% en php4, 5% en php5)
- module apache (environnement WEB) (*Apache >50% des serveurs*) ou externe (langage de programmation comme un autre)
- langage impératif disposant depuis la version 5 de fonctionnalités objet

Les scripts coté serveur (ASP)

Versions propriétaires Microsoft

- **ASP (Active Server Pages):**
 - permet d'ajouter du code dans la page HTML qui sera interprété par le serveur. La partie ADO (*ActiveX Data Object*) de ASP permet de se connecter à une base de données.
 - **Langages utilisés:** VBScript (langage par défaut) ou JScript. (version propriétaire de javascript)
 - **Performance:** Bonne performance surtout si l'on utilise du code compilé (dll).

Les scripts coté serveur (.NET) 1

- Le framework .NET installable sur Microsoft Windows.
 - approche unifiée à la conception d'applications Windows ou Web, tout en introduisant des facilités pour le développement, le déploiement et la maintenance d'applications.
 - Le framework gère tous les aspects de l'exécution de l'application :
 - Il est composé de deux blocs principaux :
 - le CLR (**Common Language Runtime** : machine virtuelle, pour l'exécution de code .NET (analogie avec JavaVirtualMachine pour Java))
 - la bibliothèque de classes .NET.

ChamSI-2008

25

Les scripts coté serveur (technologies Java)

- **JSP : Java Server Pages (Sun Microsystems)** permettent d'ajouter du code Java dans la page HTML qui sera interprété par le serveur. On peut distinguer trois ensembles d'instructions JSP:
 - **Les instructions de programmation** : insertion de code Java directement dans la page Web, donnant accès à une grande librairie de programmation de Java (composants JavaBeans, accès aux bases de données via JDBC etc.)
 - **Les directives** pour spécifier certaines propriétés de la page, l'inclusion du contenu d'autres fichiers et l'utilisation d'autres classes et librairies de balises personnalisées.
 - **Les actions** pour utiliser des composants JavaBeans existants.
- **Servlets: (analogie avec les applets) : programmes créés en Java et tournant sur le serveur web.** L'exécution du programme génère les pages web renvoyées au client.

ChamSI-2008

26

Autres scripts

- **PERL** : langage de programmation reprenant des fonctionnalités du langage C et des langages de scripts sed, awk et shell (sh) , **logiciel libre , langage le plus populaire pour l'écriture de script CGI.**
 - Perl est également capable d'exécuter des scripts dans un serveur web apache, grâce à l'extension mod_perl (performances équivalentes à celles de PHP, ou des servlets Java).
- **A PART: Ajax (Asynchronous JavaScript And XML)** technologie qui combine plusieurs éléments :
 - du XHTML pour la structure de la page,
 - les CSS pour la mise en forme de la page,
 - du JavaScript et le DOM pour la partie dynamique,
 - XML, XSLT et XMLHttpRequest pour la manipulation des données (ou d'autres formats de fichiers de données).
 - (conférence début Juin)

ChamSI-2008

27

Quelques indicateurs utiles

| langages de script serveur | | principaux serveurs Web | |
|----------------------------|--------|-------------------------|-----------------|
| | | Serveur | Parts de marché |
| PHP | 33,29% | Apache | 50,76% |
| ASP | 21,40% | Microsoft | 35,84% |
| Autres | 45,31% | Google | 5,28% |
| | | Lighttp | 1,00% |
| | | Sun | 0,41% |

(NEXEN) novembre 2007

(NETCRAFT) novembre 2007

| Résolution d'affichage | | | | | |
|------------------------|--------|----------|---------|---------|---------|
| 2007 | Higher | 1024x768 | 800x600 | 640x480 | inconnu |
| Janvier | 26% | 54% | 14% | 0% | 6% |
| 2006 | | | | | |
| Juillet | 19% | 58% | 17% | 0% | 6% |
| Janvier | 17% | 57% | 20% | 0% | 6% |

ChamSI-2008

28

Statistiques navigateurs

| 2007 | IE7 | IE6 | IE5 | Fx | Moz | Safari | Opéra |
|-----------|-------|-------|------|-------|------|--------|-------|
| November | 21.0% | 33.6% | 1.6% | 36.3% | 1.2% | 1.8% | 1.8% |
| October | 20.7% | 34.5% | 1.5% | 36.0% | 1.3% | 1.7% | 1.6% |
| September | 20.8% | 34.9% | 1.5% | 35.4% | 1.2% | 1.6% | 1.5% |
| August | 20.5% | 35.7% | 1.5% | 34.9% | 1.3% | 1.5% | 1.7% |
| July | 20.1% | 36.9% | 1.5% | 34.5% | 1.4% | 1.5% | 1.9% |
| June | 19.7% | 37.3% | 1.5% | 34.0% | 1.4% | 1.5% | 1.8% |
| May | 19.2% | 38.1% | 1.6% | 33.7% | 1.3% | 1.5% | 1.7% |
| April | 19.1% | 38.4% | 1.7% | 32.9% | 1.3% | 1.5% | 1.6% |
| March | 18.0% | 38.7% | 2.0% | 31.8% | 1.3% | 1.6% | 1.6% |
| February | 16.4% | 39.8% | 2.5% | 31.2% | 1.4% | 1.7% | 1.5% |
| January | 13.3% | 42.3% | 3.0% | 31.0% | 1.5% | 1.7% | 1.5% |

Moz : Mozilla suite : (netscape,gecko)

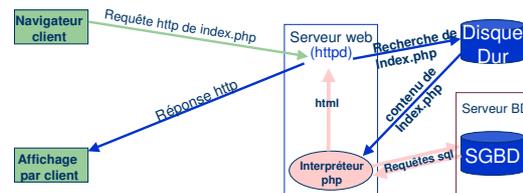
Créé le 11/01/2008

29

PHP : introduction

- Langage
 - Interprété indépendant de la plate-forme d'exécution,
 - s'exécutant sur le serveur,
 - intégré au code html de la page
 - générant du html dynamique

Server-side, HTML embedded, cross-platform scripting language



Créé le 11/01/2008

30

Php: principes et fonctionnement (1)

- Le serveur lit les instructions php intégrées au code html
- Il les interprète, et les remplace par le résultat de leur exécution

AVANTAGES:

- Le client ne voit pas le source mais seulement le résultat (cf javascript...)
- Le code n'est pas lourd comme le cgi...
- La qualité dynamique est masquée.
- Indépendance du client

Créé le 11/01/2008

31

Php: principes et fonctionnement (2)

FONCTIONNEMENT

- Un bloc php est un groupe de lignes encadré par `<?php et ?>` (on peut se limiter à `<? et ?>` mais ...
il faut alors avoir positionné `short_open_tag=on` dans la configuration `php.ini`)
- toute ligne à l'extérieur de ce couple n'est pas interprétée mais copiée sur la sortie standard.
- Toute ligne à l'intérieur est interprétée comme une instruction php.

Ainsi les instructions php sont **invisibles en sortie** ;

- En cas d'erreur, un message est intégré au flux de sortie (interprétation interrompue sauf warning)

Créé le 11/01/2008

32

Plateforme WAMPSEVER 2.0 :

- Licence GPL , est composé de (20 Mo)
 - Apache 2.2.3
 - PHP 5.2.0
 - SQLitemanager 1.2.0
 - MySQL 5.0.27
 - Phpmyadmin 2.9.1.1
- Installation
 - Téléchargement sur www.wampserver.com.
 - Installation classique automatisée

Cham51-2008

33

Installer php

Wampserver 2.0

Ou (exclusif)

Easyphp2.0

Cham51-2008

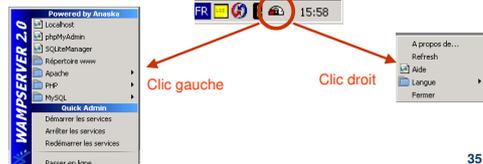
34

Instal de WAMPSEVER 2.0 (1)

1. On récupère WampServer2.0a et on lance l'installation
2. Accord licence
3. Sélection destination
4. Installation



Et enfin



Cham51-2008

35

Etat de wampserver :

- L'icône  vous donne un aperçu du bon fonctionnement de "WAMPSEVER2.0"

-  tous les services fonctionnent correctement :
-  tous les services sont stoppés
-  un seul service est en fonctionnement

Vous pouvez gérer les services globalement (apache, mysql,pgsql,...) par les items du menu

Quick Admin



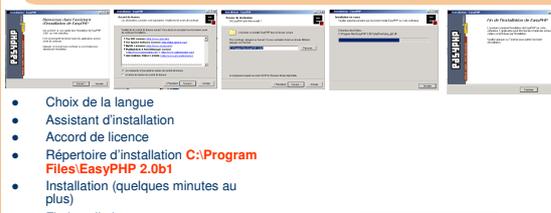
Passer hors ligne Permet de rendre le site accessible à tous ou limité à localhost

Cham51-2008

36

Installation easyphp (1)

- Choix de la langue
- Assistant d'installation
- Accord de licence
- Répertoire d'installation **C:\Program Files\EasyPHP 2.0b1**
- Installation (quelques minutes au plus)
- Fin installation
- Lancer easyphp
(mise à jour éventuelle , et c'est fini) :



Indicateur easyphp

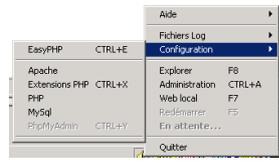
Un Clickdroit donne accès à ce menu →



41

Installation easyphp (2)

- Serveur web <http://127.0.0.1/> ou <http://localhost/> avec Racine web : **C:\Program Files\EasyPHP 2.0b1/www/**
- **Configuration**
 - Pour easyphp (service ou non, version,...)
 - Pour apache
 - Pour php extensions
 - Pour php.ini
 - Pour mysql



42

Installation easyphp (3)

- **Page d'Administration** (click droit, item Administration)

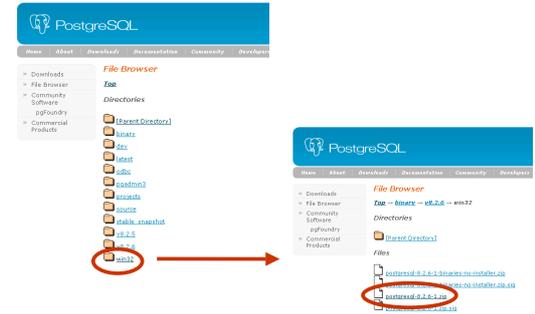


- En cas de pbs :
 - les fichiers **log** sont directement accessibles
 - exécuter `phpinfo()` : sert généralement à vérifier la configuration ainsi que les variables pré-définies, pour une plate-forme donnée.

43

Installation postgresql : <http://www.postgresql.org/>

<http://www.postgresql.org/> puis onglet download et file browser



44

Installation postgresql (NTFS nécessaire)

- Doc d'installation détaillée
<http://pginstaller.projects.postgresql.org/>
- Installation:
 - Choix langue
 - Écran d'intro
 - instructions (à lire)
 - Options d'installation
 - Configuration comme service et initialisation (à faire si NTFS!!)
 - Procedural langages, modules, postgis (inutiles)
 - Installing

45

Utiliser postgresql (exemple sous easyphp)

- **Utiliser postgresql :**
 - **Installer postgresql** (<http://www.postgresql.org/répertoirewin32/>)
Voir <http://fadace.developpez.com/postgresql/>
 - **Ouvrir php.ini** et **retirer le ;** devant `extension=php_pgsql.dll`
pour cela on va dans **Configuration --> Extensions PHP** et on coche `php_pgsql`
- Pour avoir phpPgAdmin : (dans la page administration)
 - après avoir téléchargé phpPgAdmin, il faut l'extraire à la racine de Easyphp(ou de wamp).
 - On prend soin de renommer le dossier en `phppgadmin` (tout en minuscule).
 - On modifie la config d'apache pour y insérer la déclaration de `phpPgAdmin` (identique à `phpmyadmin`)

Exemple sous easyphp :
<http://dgruessinger.developpez.com/postgresql/easyphp-phppgadmin/easyphppg.pdf>

46

Divers pbs recensés

- Difficulté si système de fichiers FAT pour postgresql (initdb.exe manuel)
- Attention , si skype, port 80 utilisé
 - vérification sur `apache->service-> Puis Tester le port 80 (wampserver)`
 - Ou
 - Démarrer/Exécuter et entrez la commande suivante :
`cmd /k netstat -ano ou tcpview ou ...`

- Ne pas lancer simultanément wampserver et easyphp...
- Easyphp2.0 est encore en version beta....plei@d

BONNE INSTALLATION

47