

Cours du 15/11

Couche application

Perl, php, javascript

ftp

Traitement en perl

□ fichier perl

Paramètres

- Les paramètres sont accessibles par l'intermédiaire de la variable d'environnement `QUERY_STRING`

VARIABLES D'ENVIRONNEMENT

❑ SERVER_SOFTWARE

- ❖ Le nom et la version du serveur HTTP répondant à la requête. (Format : nom/version)

❑ SERVER_NAME

- ❖ Le nom d'hôte, alias DNS ou adresse IP du serveur.

❑ GATEWAY_INTERFACE

- ❖ La révision de la spécification CGI que le serveur utilise. (Format : CGI/révision)

Variables...

- ❑ SERVER_PROTOCOL
 - ❖ Le nom et la révision du protocole dans lequel la requête a été faite (Format : protocole/révision)
- ❑ SERVER_PORT
 - ❖ Le numéro de port sur lequel la requête a été envoyée.
- ❑ REQUEST_METHOD
 - ❖ La méthode utilisée pour faire la requête. Pour HTTP, elle contient généralement « GET » ou « POST ».
- ❑ PATH_INFO
 - ❖ Le chemin supplémentaire du script tel que donné par le client. Par exemple, si le serveur héberge le script « /cgi-bin/monscript.cgi » et que le client demande l'url « http://serveur.org/cgi-bin/monscript.cgi/marecherche », alors PATH_INFO contiendra « marecherche ».
- ❑ PATH_TRANSLATED
 - ❖ Contient le chemin demandé par le client après que les conversions virtuel → physique aient été faites par le serveur.

Variables

❑ SCRIPT_NAME

- ❖ Le chemin virtuel vers le script étant exécuté. Exemple : « /cgi-bin/script.cgi »

❑ QUERY_STRING

- ❖ Contient tout ce qui suit le « ? » dans l'URL envoyée par le client. Toutes les variables provenant d'un formulaire envoyé avec la méthode « GET » sera contenue dans le QUERY_STRING sous la forme « var1=val1&var2=val2&... ».

❑ REMOTE_HOST

- ❖ Le nom d'hôte du client. Si le serveur ne possède pas cette information (par exemple, lorsque la résolution DNS inverse est désactivée), REMOTE_HOST sera vide.

❑ REMOTE_ADDR

- ❖ L'adresse IP du client.

❑ AUTH_TYPE

- ❖ Le type d'identification utilisé pour protéger le script (s'il est protégé et si le serveur supporte l'identification).

Variables

- ❑ AUTH_TYPE
 - ❖ Le type d'identification utilisé pour protéger le script (s'il est protégé et si le serveur supporte l'identification).
- ❑ REMOTE_USER
 - ❖ Le nom d'utilisateur du client, si le script est protégé et si le serveur supporte l'identification.
- ❑ REMOTE_IDENT
 - ❖ Nom d'utilisateur (distant) du client faisant la requête. Le serveur doit supporter l'identification RFC 931. Cette variable devraient être utilisée à des fins de journaux seulement.
- ❑ CONTENT_TYPE
 - ❖ Le type de contenu attaché à la requête, si des données sont attachées (comme lorsqu'un formulaire est envoyé avec la méthode « POST »).
- ❑ CONTENT_LENGTH
 - ❖ La longueur du contenu envoyé par le client.

Variables

□ HTTP_ACCEPT

- Les types de données MIME que le client accepte de recevoir.
- Exemple : text/*, image/jpeg, image/png, image/*, */*

□ HTTP_ACCEPT_LANGUAGE

- Les langages dans lequel le client accepte de recevoir la réponse.
- Exemple : fr_CA, fr

□ HTTP_USER_AGENT

- Le navigateur utilisé par le client.
- Exemple : Mozilla/5.0 (compatible; Konqueror/3; Linux)

Compléments Javascript

□ Code qui s'exécute du côté du client

- ❖ calcul local
- ❖ contrôle d'une zone de saisie
- ❖ affichage d'alerte
- ❖ fenêtres menus etc..

□ Balise :

```
<SCRIPT language="JavaScript1.2">
```

le code...

```
</SCRIPT>
```

Exemple: bonjour

```
<HTML><HEAD>
<TITLE>Très facile</TITLE>
</HEAD>

<SCRIPT language="JavaScript1.2">
  function bonjour()
  {
    alert ("Bonjour madame, bonjour monsieur");
  }
</SCRIPT>

<BODY bgcolor="WHITE" onLoad="bonjour();">
  <H1>Bonjour</H1>

</BODY></HTML>
http://localhost/~hf/BjrJvs.html
```

Un peu plus: minicalcul

```
<HTML>
<HEAD>
<TITLE>Petit calcul</TITLE>
</HEAD>
<BODY bgcolor='WHITE'>

<script language='JavaScript1.2' src='calcul.js'></script>
<script language='JavaScript1.2' src='fenetre.js'></script>
<script language='JavaScript1.2' src='ctrl.js'></script>

<CENTER><H1>Calcul</H1></CENTER>
```

Un petit exemple de formulaire.

```
<P>
```

Création d'une

```
<A href='#A' onClick='afficheDoc();'>fenêtre avec JavaScript</A>
```

Suite

```
<FORM ACTION='Simul.html' METHOD='POST' NAME='Simul'>
<CENTER>
<TABLE BORDER=3>
<TR><TD>Argument 1
  <TD> <INPUT TYPE='TEXT' SIZE=20 NAME='arg1' onChange='calcul();'></TR>
<TR><TD>* Argument 2
  <TD> <INPUT TYPE='TEXT' SIZE=20
    NAME='arg2' onChange='calcul();'>
</TR>

<TR><TD>Résultat=
  <TD> <INPUT TYPE='TEXT' SIZE=20
    NAME='res' >
</TR>
</TABLE>
<INPUT TYPE='BUTTON' VALUE='Vérifier' onClick='ctrl();'>
<INPUT TYPE='RESET' VALUE='Effacer tout'
  onClick=' if (!confirm("Vraiment vous voulez effacer ?")) exit;'>
</CENTER>
</FORM>
</BODY>
</HTML>
http://localhost/~hf/Simul.html
```

Fichiers js

□ Ctrl.js calcul

```
function ctrl()  
{  
    if (isNaN(window.document.Simul.res.value ))  
    {  
        alert ("valeur incorrecte : " +  
                document.Simul.res.value + "?");  
        document.forms[0].res.focus();  
    }  
}
```

```
function calcul()  
{  
    v1=document.forms[0].arg1.value;  
    v2=document.forms[0].arg2.value;  
    document.forms[0].res.value = v2*v1 ;  
}
```

suite et fin

□ [exemples/fenetre.js](#)

```
function afficheDoc()
{
    options = "width=300,height=200";
    fenetre = window.open(' ', 'MU', options);

    fenetre.document.open();
    manuel = "<HTML><HEAD><TITLE>Documentation</TITLE></
    HEAD>"
        + "<BODY bgcolor='white'>"
        + "Il n'y a pas besoin d'aide "
        + " c'est facile."
        + " Bonne chance !</BODY></HTML>";
    fenetre.document.write(manuel);
    fenetre.document.close();
}
```

Compléments: php

- ❑ php est un langage de script pour les serveurs webs
- ❑ de nombreuses fonctions permettent de traiter les requêtes http (en particulier des requêtes concernant des bases de données)
- ❑ ici on est du côté du serveur...

Exemple simple

```
<HTML> <HEAD>  
<TITLE>Exemple très simple</TITLE>  
</HEAD>  
<BODY>  
<H1>Exemple</H1>  
le <?php echo Date ("j/m/Y à H:i:s"); ?>  
<P>
```

```
<?php  
echo "Client :" . $_SERVER['HTTP_USER_AGENT'] . "<BR>";  
echo "Adresse IP client:" . $_SERVER['REMOTE_ADDR'] . "<BR>";  
echo "Server: " . $_SERVER['SERVER_NAME'];  
?>
```

```
</BODY></HTML>
```

<http://localhost/~hf/ExempleSimple.php>

Résultat

Exemple

le 8/11/2006 à 15:54:29 Client :Mozilla/4.0
(compatible; MSIE 7.0; Windows NT
5.1; .NET CLR 1.1.4322; InfoPath.1)

Adresse IP client:127.0.0.1
Server: localhost

Reçu par le client

```
<HTML> <HEAD>  
<TITLE>Exemple très simple</TITLE>  
</HEAD>  
<BODY>
```

```
<H1>Exemple</H1>
```

```
le 8/11/2006 à 15:54:29
```

```
<P>
```

```
Client :Mozilla/4.0 (compatible; MSIE 7.0; Windows NT  
5.1; .NET CLR 1.1.4322; InfoPath.1)<BR>Adresse IP client:  
127.0.0.1<BR>Server: localhost
```

```
</BODY></HTML>
```

Php

- ❑ On est ici côté serveur:
 - ❖ les balises `<?php>` `<?>` sont interprétées par le serveur (apache par exemple) et servent à générer la page html reçue par le client
- ❑ Mais surtout php permet
 - ❖ d'accéder aux variables d'environnement
 - ❖ d'utiliser de nombreuses fonctionnalités
 - sessions, paramètres etc.
- ❑ Php sert souvent d'interface pour MySQL serveur simple de bases de données

Php

- ❑ pas de typage ni de déclaration des variables
- ❑ `$v` est remplacé par la valeur de `v` (et permet aussi l'affectation)
- ❑ `echo "$v";`
- ❑ constantes `define("PI, 3.1415);`
- ❑ types des variables
 - ❖ numériques
 - `$i=1;`
 - `$v=3.14;`
 - ❖ chaînes de caractères (expressions régulières)
 - `$nom="Hugues";`
 - `'",.{}`

php

□ Variables

- ❖ Locales (à une fonction)
- ❖ Globales
- ❖ Super globales (disponibles dans tout contexte)
- ❖ Static (garde sa valeur)
- ❖ Variables dynamiques (le nom de la variable est une variable)
 - `$a='bonjour'`
 - `$$a='monde'`
 - `echo "$a ${$a}"`
 - `echo "$a $bonjour"`

php

□ tableaux

❖ indicés

- `$tab[0]="un";`
- `$tab=array("un","deux","trois");`

❖ associatifs

- `$m=array("un"=>"one",
 "deux"=>"two");`
- `$m["trois"]="three";`

❖ next() prev() key() current

do

```
{echo "Clé=key($m).Valeur= current($m)"}
```

```
while(next($mes));
```

```
foreach($m as $cle =>$val)
```

```
{echo "Clé=$cle.Valeur=$val";}
```

php

- Mais aussi (php4 et php5)
 - ❖ Programmation orientée objets
 - Classes et Objets
 - Liaison dynamique
 - Constructeurs
 - ...
 - ❖ Exceptions

Php

- structures de contrôles
 - ❖ if
 - ❖ if else
 - ❖ while
 - ❖ do while
 - ❖ for
 - ❖ foreach
 - break, continue

fonctions

```
function Nom([$arg1, $arg2, ...])  
{  
    corps  
}
```

passage par valeur (et par référence &)

exemples

```
function Add($i,$j){  
    $somme= $i + $j;  
    return $somme;  
}
```

```
function Add($i,$j,&$somme){  
    $somme= $i + $j;  
}
```

Pour le serveur...

- tableaux associatifs prédéfinis
 - ❖ `$_SERVER`: environnement serveur
 - `REQUEST_METHOD`
 - `QUERY_STRING`
 - `CONTENT_LENGTH`
 - `SERVER_NAME`
 - `PATH_INFO`
 - `HTTP_USER_AGENT`
 - `REMOTE_ADDR`
 - `REMOTE_HOST`
 - `REMOTE_USER`
 - `REMOTE_PASSWORD`

Suite

□ Autres tableaux

- ❖ `$_ENV`: environnement système
- ❖ `$_COOKIE`
- ❖ `$_GET`
- ❖ `$_POST`
- ❖ `$_FILES`
- ❖ `$_REQUEST` (variables des 4 précédents)
- ❖ `$_SESSION`
- ❖ `$GLOBALS` les variables globales du script

Cookies et php

```
<?php
// Est-ce que le cookie existe ?
if (isset($_COOKIE['compteur']))
{
    $message = "Vous êtes déjà venu {$_COOKIE['compteur']} fois "
        . "me rendre visite<BR>\n";
    // On incrémente le compteur
    $valeur = $_COOKIE['compteur'] + 1;
}
else
{
    // Il faut créer le cookie avec la valeur 1
    $message = "Bonjour, je vous envoie un cookie<BR>\n";
    $valeur = 1;
}
// Envoi du cookie
SetCookie ("compteur", $valeur);
?>
```

Cookies et php (fin)

```
<HTML><HEAD>
```

```
  <TITLE>Les cookies</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
<H1>Un compteur d'accès au site avec cookie</H1>
```

```
<?php echo $message; ?>
```

```
</BODY></HTML>
```

```
http://localhost/~hf/SetCookie.php
```

En utilisant les sessions

```
<?php
  // La fonction session_start fait tout le travail
  session_start();
?>
<HTML><HEAD>
  <TITLE>Les cookies</TITLE>
</HEAD>
<BODY>

<H1>Un compteur d'accès au site avec Session</H1>
```

Fin

```
<?php
if (!isset($_SESSION['cp']))
{
    $_SESSION['cp']=1;
    echo "C'est la première fois, votre id est:" .
        session_id()."<BR>";
}
else{
    $_SESSION['cp']++;
    echo "C'est votre " . $_SESSION['cp'] . " n-ième connexion";
    if($_SESSION['cp']>10){
        echo "on vous a trop vu". "<BR>";
        session_destroy();
    }
}
?>
</BODY></HTML>
http://localhost/~hf/SessionPHP.php
```

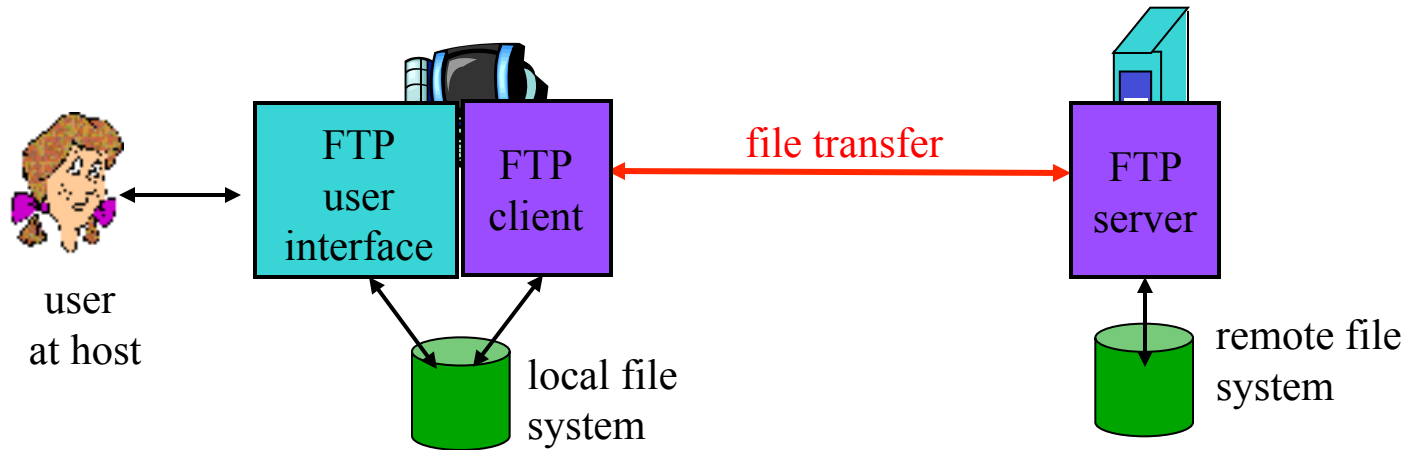
session

- ❑ `session_start()`
- ❑ `session_destroy()`
- ❑ `session_id()`
 - ❖ on peut associer des variables à la session par le tableau associatif `$_SESSION`
 - ❖ elle sera accessible à chaque `session_start()` jusqu'au `session_destroy()` pour toute connexion qui fournit le `session_id()`.

Couche application

- FTP

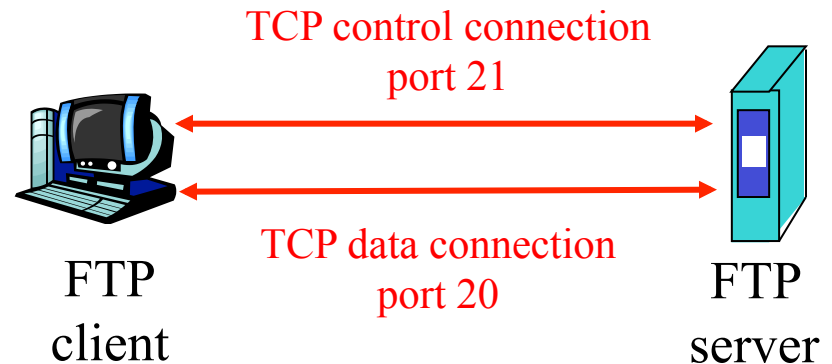
FTP: the file transfer protocol



- ❑ transfer file to/from remote host
- ❑ client/server model
 - ❖ *client*: side that initiates transfer (either to/from remote)
 - ❖ *server*: remote host
- ❑ ftp: RFC 959
- ❑ ftp server: port 21

FTP: separate control, data connections

- ❑ FTP client contacts FTP server at port 21, TCP is transport protocol
- ❑ client authorized over control connection
- ❑ client browses remote directory by sending commands over control connection.
- ❑ when server receives file transfer command, server opens 2nd TCP connection (for file) to client
- ❑ after transferring one file, server closes data connection.



- ❑ server opens another TCP data connection to transfer another file.
- ❑ control connection: “out of band”
- ❑ FTP server maintains “state”: current directory, earlier authentication

FTP commands, responses

Sample commands:

- ❑ sent as ASCII text over control channel
- ❑ USER *username*
- ❑ PASS *password*
- ❑ LIST return list of file in current directory
- ❑ RETR *filename* retrieves (gets) file
- ❑ STOR *filename* stores (puts) file onto remote host

Sample return codes

- ❑ status code and phrase (as in HTTP)
- ❑ 331 Username OK, password required
- ❑ 125 data connection already open; transfer starting
- ❑ 425 Can't open data connection
- ❑ 452 Error writing file