

## **Module: OSI, INTERNET ET PROGRAMMATION WEB**

### **TP 2 - Services TCP/IP-APACHE et xhtml**

**Objectif** : examiner les différents services TCP/IP, configuration du service httpd Apach – introduction au xhtml

**Correction** : la correction sera diffusée la semaine suivant la fin du TP sur le support cours.

## **1. Services TCP/IP : Protocoles d'applications standards**

### **•TELNET**

1.1. Que permet de faire la commande **telnet** ?

1.2. Décrivez à l'aide d'un schéma, le principe de fonctionnement de **telnet**.

1.3. Quel est le protocole utilisé entre un client et un serveur **telnet** ?

1.4. Le protocole TELNET est dit transparent, pourquoi ?

1.5. Un client **telnet** peut-il contacter un service autre que celui d'un serveur **telnetd** ?

1.6. Lancez le client **telnet** depuis une boîte DOS sur votre PC, puis ouvrez une session sur *nyx*

1.7. Tapez la commande **netstat** dans la boîte DOS, pour voir les connexions actives, en particulier la connexion telnet que vous venez d'établir.

1.8. Depuis *nyx*, tapez la commande **telnet** sans paramètre :

- affichez les commandes de telnet ( ? )
- ouvrez une connexion avec *nyx*
- revenir en mode telnet par le caractère d'échappement (^)]
- affichez l'état de la connexion (status)
- interrompre la session telnet avec *nyx* et retourner au shell appelant (z)
- reprendre la session telnet par la commande shell **fg**
- fermez la connexion avec *nyx*, sans quitter telnet (close)
- quittez telnet (quit)

### **•FTP**

1.9. Que permet de faire la commande **ftp** ?

1.10. Décrivez à l'aide d'un schéma, le principe de fonctionnement de **ftp**

1.11. Quel est le protocole utilisé entre un client et un serveur **ftp** ?

1.12. Lancez le client **ftp** sur votre station depuis une fenêtre DOS, consultez les commandes de ftp à l'aide de la commande HELP.

1.13. Quels sont les différents modes de transfert de ftp ?

Pour les manipulations à suivre, vous vous servirez d'une boîte DOS pour taper les commandes DOS, et d'une fenêtre telnet sur *nyx* pour taper les commandes bash

### **•Rapatrier un fichier texte sur votre station de travail**

1.14. Sur *nyx*, créez un petit fichier de texte *fd.txt* sous shell (utiliser la commande cat)

```
$ cat > fd.txt
blabla
^D
```

1.15. A l'aide de la commande ftp sur votre station de travail, rapatriez dans C:\temp le fichier *fd.txt* que vous venez de créer sur *nyx*.

### •Transférer un fichier texte depuis votre station sur un serveur du département

1.16. Créez sur votre station, un petit fichier de texte *fl.txt* à l'aide de notepad.

1.17. A l'aide du client ftp sur votre station, transférez le fichier *fl.txt* dans votre répertoire HOME sur serveur du département, puis le visualiser avec vi ou emacs sous UNIX.

1.18. Transférez à nouveau le fichier *fl.txt* sur serveur du département, mais cette fois-ci en mode binary, puis l'afficher, sous vi ou emacs, qu'observez-vous en fin de chaque ligne ? Explications.

### •Utiliser ftp dans un fichier de commandes Windows NT

1.19. Écrivez un fichier de commandes Windows NT, nommé TRANSFERT.BAT, permettant de transférer un fichier de texte quelconque se trouvant sur votre station WNT2000 vers le serveur du département. Le nom d'utilisateur et le mot de passe pour établir la connexion sur *nyx* seront spécifiés en premier et deuxième paramètres. Le nom du fichier NT à transférer sera spécifié en troisième paramètre. Le fichier sera transféré dans le répertoire HOME de l'utilisateur.

Syntaxe d'appel : TRANSFERT <username> <password> <nom de fichier>  
Vous devez utiliser l'option -n de ftp pour supprimer l'ouverture automatique de session.

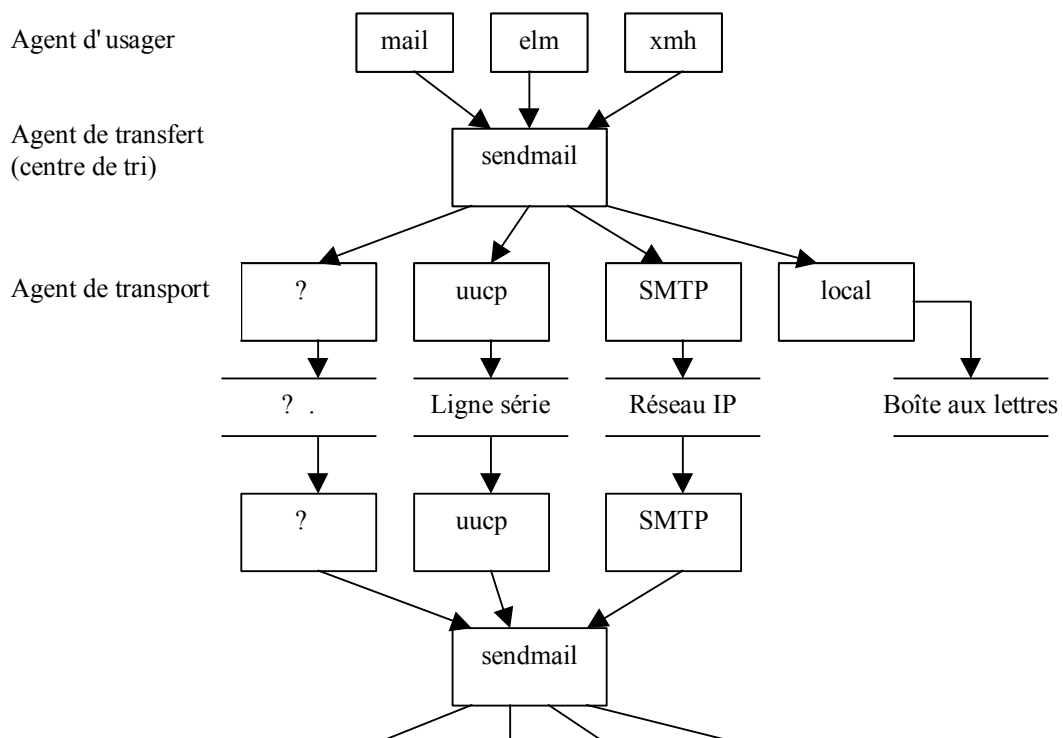
1.20. Tester.

### •Tester le protocole FTP

1.21. Connectez-vous par telnet à ftp, puis dialoguez avec le protocole FTP, par exemple envoyez les commandes HELP, USER, PASS, LIST, BIDE, QUIT, et observez les réponses du serveur.

### •SMTP

1.22. Décrivez à l'aide d'un schéma le principe de fonctionnement du courrier électronique.



1.23. Quel est le protocole utilisé par l'Agent de Transport du courrier électronique ?

#### •La commande mail de UNIX

1.24. Utilisez la commande **mail** de Unix pour envoyer un message à vous-même.

1.25. Affichez le contenu de votre boîte à lettres sur serveur du département, dans **/var/mail**.

1.26. Dans quels cas le courrier est-il retourné ?

1.27. Essayez d'envoyer un message à un utilisateur inconnu sur serveur du département.

#### •Utiliser la commande mail dans une procédure shell

1.28. Écrivez une procédure shell sous UNIX, nommée `envoi.sh`, permettant d'envoyer par e-mail les notes obtenues dans le module réseaux, chaque étudiant recevra un message l'informant de sa note.

Les notes sont stockées dans un fichier de texte `rx1.dat` sous UNIX, avec le format suivant :

`nom note`

~ chaque ligne comporte le nom d'un étudiant suivi de sa note séparée par un espace blanc

~ on supposera que les noms d'étudiants sont identiques aux noms d'utilisateurs sous UNIX

1.29. Tester à l'aide d'un jeu d'essai permettant d'adresser un message à vous-même.

1.30. Rédigez une deuxième version de la procédure `envoi.sh` afin de pouvoir spécifier en paramètre le nom du fichier de notes.

#### •Avis d'arrivée du courrier

1.31. Quelle est la fréquence de consultation de votre boîte à lettres par le shell sous Unix ?

#### •Alias

1.32. Créez un alias dans `$HOME/.mailrc` permettant d'envoyer un mail à vous-même et votre binôme.

1.33. Tester.

#### •Tester le protocole SMTP

1.34. Depuis votre station de travail NT, connectez-vous par **telnet** à **smtp** sur le serveur du département.

1.35. Consultez l'aide en ligne, à l'aide de la commande **HELP**, puis envoyez un message à vous-même.

#### •HTTP

Pour les tests en dessous, contactez le serveur `httpd` sur le serveur du département en utilisant la commande `telnet` de UNIX (à partir d'une fenêtre `telnet` sur votre station de travail)

#### Requêtes et Réponses

1.36. Demandez le document `/test0.html` en spécifiant le protocole HTTP/1.0 (la requête doit se terminer par une ligne vide)

1.37. Demandez uniquement les en-têtes du document `/test0.html`

1.38. Quelle est le type (ou la marque) du serveur ?

1.39. Demandez les méthodes de requête disponibles pour le document `/test0.html`

#### Erreurs du client

1.40. Obtenir des réponses avec un code statut indiquant une erreur du client.

Essayer une requête incorrecte (Erreur client 400 Bad Request)

Essayer une ressource inexistante (Erreur client 404 Not Found)

### Erreurs du serveur

1.41. Obtenir une réponse avec un code statut indiquant une erreur du serveur.

Essayer une méthode non implémentée (Erreur serveur 501 Method Not Implemented)

### Connexion persistante

Les versions 0.9 et 1.0 du protocole HTTP nécessitent une nouvelle connexion TCP à chaque requête, la version actuelle HTTP/1.1 (RFC 2616) autorise plusieurs échanges de requêtes/réponses via une unique connexion (notion de connexion persistante ou *keep alive*).

1.42. Essayez les trois échanges (requêtes/réponses) présentés ci-dessous et effectués successivement sur la même connexion TCP, dans la troisième requête nous avons ajouté le champ d'en-tête `Connection : close` afin de demander la fermeture de la connexion après la réponse (il faut taper les commandes rapidement, la connexion étant fermée automatiquement au bout de quelques secondes d'inactivité)

Note : la norme HTTP/1.1 impose de spécifier le nom du serveur auquel s'adresse la requête (grâce au champ d'en-tête `Host`), ceci, pour permettre à une machine serveur qui possède plusieurs noms et héberge plusieurs sites, d'associer à une même adresse l'net `nyx` `GETHost` des ressources différentes en fonction du nom de l'hôte (cette pratique est connue sous le nom de *virtual host*)

### Requêtes conditionnelles

1.43. Demandez le document `/test0.html` s'il a été modifié depuis la date de votre dernière consultation.

## 2. Analyser la configuration du serveur Apache

Le serveur **Apache** sur le serveur du département est installé dans `/etc/apache-ssl` (à vérifier)

Les fichiers de configuration se trouvent dans le sous répertoire `conf`

- `httpd.conf` contient les directives de configuration générale
- `srm.conf` contient les directives concernant les ressources du serveur
- `access.conf` contient les directives concernant la politique d'accès au serveur.

Dans la nouvelle version 1.3 de Apache, les contenus des trois fichiers sont regroupés dans `httpd.conf`

### Configuration générale

Consultez le fichier **httpd.conf** pour répondre aux questions suivantes

2.1. Comment le démon `httpd` est-il lancé sur `nyx` ?

2.2. Sur quel port le serveur `httpd` est-il à l'écoute ?

2.3. Quel est le `user id` utilisé par le processus `httpd` répondant à la requête d'un client ?

### Ressources du serveur

– **Emplacement des documents HTML**

2.4. Où sont stockés les documents HTML diffusé par notre serveur W3 (pour chaque serveur virtuel) ?

2.5. Où sont stockés les documents HTML des utilisateurs sur le serveur du département ?

2.6. Quelles sont les protections UNIX que vous devez affecter aux documents HTML dans votre répertoire personnel afin qu'ils soient accessibles par le serveur W3 ?

Droit d'accès et de lecture pour tout le monde.

– **Gestion des listes de fichiers**

2.7. Quel est le nom de fichier pris par défaut lorsqu'une requête porte sur un répertoire ?

```
DirectoryIndex index.cgi index.html index.htm cover.html
```

La directive `DirectoryIndex` permet d'indiquer le nom d'un fichier de telle sorte que, si un fichier portant ce nom existe dans le répertoire sur lequel porte la requête, le serveur ne construit pas la liste des fichiers du répertoire mais renvoie à la place le contenu de ce fichier.

– **Emplacement des scripts CGI**

2.8. Où sont stockés les scripts CGI du serveur `www_iutinfo.unice.fr` ?

– **Traduction des URLs**

2.9. Indiquer les URLs permettant de référencer sur `www-iutinfo` les documents HTML suivants :

- le document `doc1.html` dans l'espace web du serveur `www-iutinfo.unice.fr`
- le document `doc2.html` dans le répertoire web de l'utilisateur `fred`.

2.10. Indiquer les URLs permettant de référencer sur `www_iutinfo` les scripts CGI suivants :

- le script `script1.sh` dans le répertoire `cgi-bin` du serveur,
- le script `script2.cgi` dans l'espace web de l'utilisateur `fred`.

2.11. Indiquer comment le serveur `httpd` sur `nyx` traduit les requêtes suivantes (sans préciser un nom de serveur virtuel)

1. GET /
2. GET /~fred/
3. GET /~fred/doc.html
4. GET /cgi-bin/script.sh
5. GET /~fred/script.cgi

**Mettre en place votre espace WEB sur nyx**

2.12. Testez dans votre navigateur l'url : <http://???unice.fr/~my-username> (où *my-username* représente votre nom d'utilisateur sur `nyx`) après chacune des étapes décrites en dessous, observez bien les différentes réponses successives du serveur

1. se loger par telnet sur `nyx`
2. créez dans votre répertoire `$HOME` sur `nyx` un répertoire nommé `web`  
`nyx$ mkdir web`
3. changez les permissions :  
`nyx$ chmod o+x .`  
`nyx$ chmod o+rx web`
4. créez dans votre répertoire `web` un fichier `index.html`  
`nyx$ cd web`  
`nyx$ cat > index.html`  
`<h1>bienvenue chez moi</h1>`  
`^D`
5. changez les permissions de `index.html`  
`nyx$ chmod o+r index.html`

**Politique d'accès au serveur**

– **Protection des accès par domaines**

2.13. Créer un répertoire `R1` dans votre répertoire `web`, puis interdire tous les accès dans `R1`, sauf aux utilisateurs venant de votre station `NT`, vérifiez.

– **Protection des accès par nom d'utilisateur et mot de passe**

2.14. Créer un répertoire `R2` dans votre répertoire `web` personnel, puis autoriser l'accès dans `R2` uniquement à `david` avec le mot de passe `david***` et à `steve` avec le mot de passe `steve***`. Vérifiez.

– **Trace des accès clients**

2.15. Dans quel fichier sont notifiés tous les accès des clients à notre serveur `www-iutinfo` ?

2.16. Consulter les 10 derniers accès à notre serveur `www-iutinfo` (utiliser la commande `tail` de UNIX).

## 3. Démarrer avec XHTML

### Un exemple simple

3.1. A l'aide de Bloc-notes, saisir l'exemple en dessous, l'enregistrer sous `1.1.html`.

```
<?xml version="1.0"?>

<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd" >

<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Un exemple idiot</title>
  </head>
  <body>
    <h1>Titre de niveau 1</h1>
    Ceci est une premiere ligne<br/>
    ceci est la seconde.
    <p>
      Ceci est le deuxieme paragraphe.</p>
    <p>
    </p>
  </body>
</html>
```

3.2. Vérifier la validité du code ci-dessus à l'aide du site <http://validator.w3.org/>

3.3. Ouvrir une session SSH avec PuTTY par exemple sur *nash*.

3.4. Créez sur *nyx* un répertoire `~/web/rx1/TD4` pour y placer vos fichiers xhtml.

3.5. Par SFTP, transférez le fichier `1.1.html` sur *nash*

3.6. Dans IE5, ouvrir l'URL : <http://nash.unice.fr/~username/rx1/TD4/1.1.html> où `~username` représente votre nom d'utilisateur sur *nyx*

### Styles, listes

3.7. Créez un document xhtml, essayez différents styles, des listes, du texte pré formaté.

3.8. Créez un document xhtml, essayez la balise **FONT** pour changer la taille et la couleur des caractères.

3.9. Comment mettre des caractères spéciaux et accentués ?

3.10. Mettre le mot « intérêt » en majuscules accentuées.

### Liens

3.11. Créez une page de garde de votre répertoire `rx1`, cette page contiendra un sommaire des TD avec un lien vers chacun des répertoires correspondants (que vous créerez).

3.12. Dans la page de garde de `rx1`, insérez un lien vers :

- la racine `web` du serveur `www-iutinfo.unice.fr`
- le répertoire `rx1` de votre binôme

3.13. En fin de la page de garde de `rx1`, placez un bloc adresse comportant un lien permettant d'envoyer un courrier à votre adresse e-mail.

3.14. Créez une page de garde du répertoire TD4, comportant un sommaire des exercices que vous avez fait.

### Inclusion d'images

3.15. Créez un document xhtml, insérez une image en ligne avec une taille réduite de 150x100 pixels et un commentaire placé à gauche de l'image, prenez par exemple l'image du château Valrose qui se trouve dans `images/chato1.gif` à la racine du serveur web sur *nyx*.



Le château Valrose

3.16. Insérez un lien afin de pouvoir afficher l'image en grandeur réelle lorsqu'on clique dessus.

### Fond de page, couleur du texte

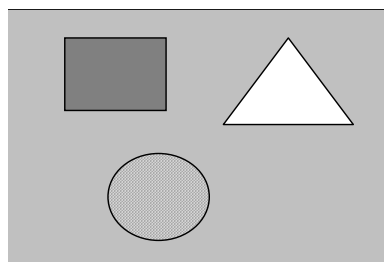
Pour les exercices suivants, créez les documents html comportant des liens vers les pages de test.

3.17. Faites des essais de couleur de fond et de texte (voir des exemples de couleurs dans le fichier `/usr/lib/X11/rgb.txt` sur *nyx*).

3.18. Essayez différents fonds de page, des motifs de fonds se trouvent dans le répertoire `images/fonds` à la racine du serveur web sur *nyx*.

## 4. Images cliquables

4.1. Créez une image `fff.gif` composée d'un rectangle gris dans lequel on trouve un rectangle, un triangle et un cercle. Repérez les coordonnées des points définissant les 3 figures (le sommet en haut à gauche et le sommet en bas à droite du rectangle, les 3 sommets du triangle, le centre et un point quelconque situé sur la circonférence du cercle).



4.2. Créez un document html, insérez l'image `fff.gif` en utilisant l'attribut **USEMAP**, on souhaite que cliquer sur une figure provoque l'affichage du nom de la figure sur laquelle on a fait un clic.