

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** ?

Elle permet de se connecter à une machine distante, tout comme une connexion depuis un terminal (*émulation de terminal*).

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

Le client **telnet** établit une connexion en mode connecté avec un serveur **telnetd**, sur le port 23.

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

TELNET (Terminal NETwork protocol), décrit par le RFC 854.

1.4. Le protocole TELNET est dit transparent, pourquoi ?

Il se contente d'échanger des caractères entre un client et un serveur, sans interprétation particulière.

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

oui, en précisant le numéro du port (ou le nom du service) sur lequel le serveur écoute.

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

```
C:\> telnet nyx.unice.fr
```

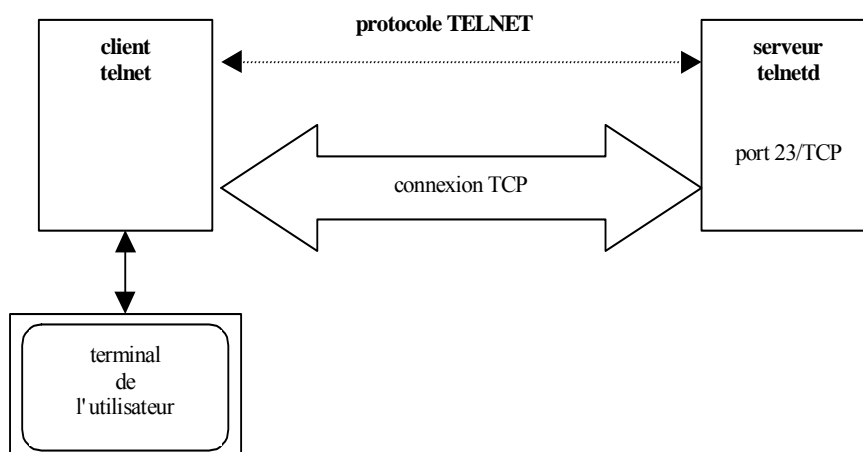
1.7. Tapez la commande **netstat** dans la boîte DOS, pour voir les connexions actives, en particulier la connexion telnet que vous venez d'établir.

```
C:\> netstat
```

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

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

•FTP



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

Permet d'échanger des fichiers entre deux machines via le réseau, hormis pour le **ftp anonyme**, il faut s'identifier sur le site cible.

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

Le serveur attend les demandes de connexion et crée un processus fils pour gérer chaque demande. Le processus fils accepte et gère l'établissement d'une connexion de contrôle à partir du client. Il utilise un processus supplémentaire pour gérer une connexion de transfert de données séparée.

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

FTP (File Transfer Protocol), décrit par le RFC 959.

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.

• Les commandes de connexion/déconnexion

open *hostname* ouvre une connexion avec la machine distante
user *username* envoie un nom d'utilisateur à la machine distante
close ferme la connexion avec la machine distante
bye ou **quit** sort de ftp après fermeture de la connexion éventuelle

• Les commandes de gestion des fichiers

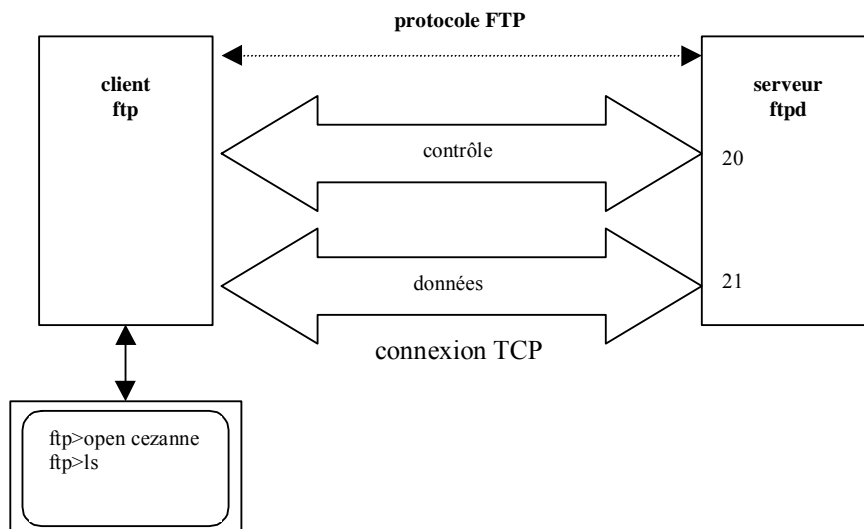
get *fd* rapatrie le fichier *fd* de la machine distante vers le répertoire courant de la machine locale
mget *fdn?* rapatrie les fichiers *fdn?* dans le répertoire courant
put *fl* transfère le fichier local *fl* sur la machine distante
mput *fln?* transfère les fichiers *fln?* sur la machine distante

• Les commandes de gestion des répertoires

pwd affiche le répertoire courant sur la machine distante
cd *rep* change de répertoire sur la machine distante
lcd *rep* change de répertoire sur la machine locale
ls affiche le contenu du répertoire courant sur la machine distante
dir [*pattern*] liste les fichiers spécifiés par *pattern*

• Les commandes de configuration du transfert

ascii définit le transfert au format ascii
binary transfère les fichiers en mode binaire
prompt demande de confirmation avant le traitement de chaque fichier pour les transferts multiples



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

- Mode ASCII (asc) pour transférer des fichiers contenant du texte (programme source, PostScript ?)
- Mode binaire (bin) pour les autres transferts (exécutable, archive, forme compressée, image, son ?)

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 le serveur du département, 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 *serveur du département (nyx par exemple)*.

```
C:\Temp> ftp
ftp> open nyx
Connecté à nyx.unice.fr.
220 nyx.unice.fr FTP server (Version 6.2/OpenBSD/Linux-0.10) ready.
Utilisateur (nyx.unice.fr:(none)) : toto
331 Password required for toto.
Mot de passe : xxx
230- Linux nyx 2.4.9 #2 SMP Fri Nov 9 20:13:33 CET 2001 i686 unknown
230 User toto logged in.
ftp> get fd.txt
200 PORT command successful.
150 Opening ASCII mode data connection for 'fd.txt' (14 bytes).
226 Transfer complete.
ftp : 16 octets reçus dans 0,00Secondes 16000,00Ko/sec.
ftp> bye
221 Goodbye.
C:\Temp> type fd.txt
Blabla
```

•Transférer un fichier texte depuis votre station sur nyx

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 *nyx*, puis le visualiser avec vi ou emacs sous UNIX.

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

- Les lignes sont terminées par le caractère CR de code ASCII 13 (^M)
- Le format des fichiers de texte sous DOS spécifie un CR, LF en fin de chaque ligne
- Le format des fichiers de texte sous UNIX spécifie uniquement un LF en fin de chaque ligne
- Le transfert en mode ASCII permet de faire la conversion de format des fichiers de texte entre des systèmes différents, mais également la conversion des codes caractères lorsqu'ils ne sont pas les mêmes (par exemple lors d'un transfert d'une machine ASCII vers une machine en code EBCDIC ou l'inverse).

•Utiliser ftp dans un fichier de commandes Windows NT

1.19.Ecrivez un fichier de commandes Windows NT, nommé TRANSFERT . BAT, permettant de transférer un fichier de texte quelconque se trouvant sur votre station WNT2000 vers *nyx*. 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.
REM fichier de commande ENVOI.BAT
REM permet de transférer un fichier sur nyx
REM %1 = username
REM %2 = password
REM %3 = nom du fichier
echo off
REM stocker dans ftemp les commandes a faire exécuter par ftp
echo open nyx.unice.fr > c:\temp\ftemp
echo user %1 >> c:\temp\ftemp
echo pass %2 >> c:\temp\ftemp
echo put %3 >> c:\temp\ftemp
echo bye >> c:\temp\ftemp
REM lancement de ftp avec en entrée les commandes stockées dans ftemp
REM l'option -n supprime l'ouverture automatique de session à la connexion
ftp -n < c:\temp\ftemp
del c:\temp\ftemp
```

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.

```
C:\temp> telnet nyx ftp
220 nyx.unice.fr FTP server (Version 6.2/OpenBSD/Linux-0.10) ready.
HELP
```

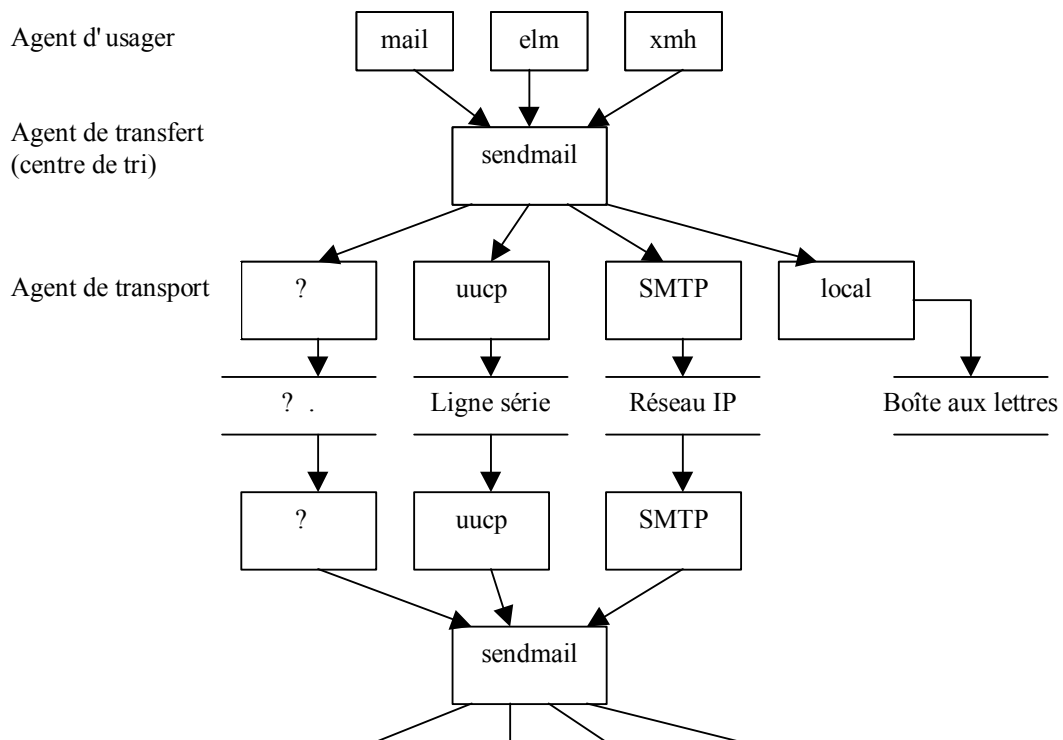
```

214- The following commands are recognized (* =>'s unimplemented).
  USER  PORT  STOR  MSAM*  RNT0  NLST  MKD  CDUP
  PASS  PASV  APPE  MRSQ*  ABOR  SITE  XMKD  XCUP
  ACCT*  TYPE  MLFL*  MRCP*  DELE  SYST  RMD  STOU
  SMNT*  STRU  MAIL*  ALLO  CWD  STAT  XRMD  SIZE
  REIN*  MODE  MSND*  REST  XCWD  HELP  PWD  MDTM
  QUIT  RETR  MSOM*  RNFR  LIST  NOOP  XPWD
214 Direct comments to ftp-bugs@nyx.unice.fr.
USER toto
331 Password required for toto.
PASS *****
230 User toto logged in.
BIDE
500 'BIDE': command not understood.
QUIT
221 Goodbye.
Perte de la connexion à l'hôte.

```

• SMTP

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



L'agent d'utilisateur de courrier (MUA) permet à un usager de composer son courrier et de l'expédier
L'agent de transport de courrier (MTA) est chargé de l'acheminement du courrier
L'agent de transfert de courrier joue le rôle du centre de tri, il choisit l'agent de transport approprié, il retourne à l'émetteur les messages qui ne peuvent être distribués.

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

SMTP (Simple Mail Transfer Protocol), décrit dans les RFC 821 et 822.

• 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 *nyx*, dans **/var/mail**.

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

Quant le message ne peut être délivré pour l'une des raisons suivantes :

- site inconnu (Host unknown)
- correspondant inconnu (User unknown)
- le courrier ne peut être délivré (Service unavailable)

1.27. Essayez d'envoyer un message à un utilisateur inconnu sur *nyx*.

• 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

```
#!/bin/sh
```

```
# fichier envoi.sh
```

```
awk '{printf("mail %s <<'EOT'\nVous avez %s en rx1\n.\nEOT\n", $1, $2)}' rx1.dat | sh
```

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.

```
#!/bin/sh
```

```
# fichier envoi.sh
```

```
awk '{printf("mail %s <<'EOT'\nVous avez %s en rx1\n.\nEOT\n", $1, $2)}' $1 | sh
```

• Avis d'arrivée du courrier

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

voir la variable d'environnement Unix **MAILCHECK**.

• Alias

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

```
alias binome toto@nyx.unice.fr
```

```
alias binome tata@nyx.unice.fr
```

1.33. Tester.

• Tester le protocole SMTP

1.34. Depuis votre station de travail NT, connectez-vous par **telnet** à **smtp** sur *nyx*.

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

```
C:\temp> telnet nyx smtp
220 nyx.unice.fr ESMTP Exim 3.12 #1 Wed, 30 Jan 2002 15:11:12 +0100
HELP
214-Commands supported:
214- HELO EHLO MAIL RCPT DATA AUTH
214 NOOP QUIT RSET HELP
HELO nyx
250 nyx.unice.fr Hello s765p1.unice.fr [134.59.28.1]
MAIL FROM: toto@nyx.unice.fr
250 <toto@nyx.unice.fr> is syntactically correct
RCPT TO: toto@nyx.unice.fr
250 <toto@nyx.unice.fr> is syntactically correct
DATA
354 Enter message, ending with "." on a line by itself
blabla
.
250 OK
QUIT
221 nyx.unice.fr closing connection
```

• HTTP

Pour les tests en dessous, contactez le serveur `httpd` sur *nyx* 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)

```
$ telnet nyx 80
Trying 134.59.22.1...
Connected to nyx.unice.fr.
Escape character is '^]'.
GET /test0.html HTTP/1.0

HTTP/1.1 200 OK
Date: Wed, 06 Feb 2002 15:40:15 GMT
Server: Apache/1.3.9 (Unix) Debian/GNU ApacheJServ/1.1 PHP/4.0.3pl1
Last-Modified: Sun, 11 Nov 2001 12:12:09 GMT
ETag: "218009-2c-3bee6b19"
Accept-Ranges: bytes
Content-Length: 44
Connection: close
Content-Type: text/html; charset=iso-8859-1

<html><body>
<h1>Test 0</h1>
</body></html>
Connection closed by foreign host.
```

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

```
$ telnet nyx 80
Trying 134.59.22.1...
Connected to nyx.unice.fr.
Escape character is '^]'.
HEAD /test0.html HTTP/1.0

HTTP/1.1 200 OK
Date: Wed, 06 Feb 2002 15:45:39 GMT
Server: Apache/1.3.9 (Unix) Debian/GNU ApacheJServ/1.1 PHP/4.0.3pl1
Last-Modified: Sun, 11 Nov 2001 12:12:09 GMT
ETag: "218009-2c-3bee6b19"
Accept-Ranges: bytes
Content-Length: 44
Connection: close
Content-Type: text/html; charset=iso-8859-1

Connection closed by foreign host.
```

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

Il est indiqué dans le champ d'en-tête Server: Apache/1.3.9 (Unix) Debian/GNU

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

```
$ telnet nyx 80
Trying 134.59.22.1...
Connected to nyx.unice.fr.
Escape character is '^]'.
OPTIONS /test0.html HTTP/1.0

HTTP/1.1 200 OK
Date: Wed, 06 Feb 2002 15:47:51 GMT
Server: Apache/1.3.9 (Unix) Debian/GNU ApacheJServ/1.1 PHP/4.0.3pl1
Content-Length: 0
Allow: GET, HEAD, POST, PUT, DELETE, CONNECT, OPTIONS, PATCH, PROPFIND, PROPPATCH,
MKCOL, COPY, MOVE, LOCK, UNLOCK, TRACE
Connection: close

Connection closed by foreign host.
```

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)
$ telnet nyx 80
Trying 134.59.22.1...
Connected to nyx.unice.fr.
Escape character is '^]'.
DONNE MOI /test0.html HTTP/1.0

HTTP/1.1 400 Bad Request
Date: Wed, 06 Feb 2002 15:49:24 GMT
```

```

Server: Apache/1.3.9 (Unix) Debian/GNU ApacheJServ/1.1 PHP/4.0.3p11
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<HTML><HEAD>
<TITLE>400 Bad Request</TITLE>
</HEAD><BODY>
<H1>Bad Request</H1>
Your browser sent a request that this server could not understand.<P>
Invalid URI in request DONNE MOI /test0.html HTTP/1.0<P>
<HR>
<ADDRESS>Apache/1.3.9 Server at www-iutinfo.unice.fr Port 80</ADDRESS>
</BODY></HTML>
Connection closed by foreign host.

```

Essayer une ressource inexistante (Erreur client 404 Not Found)

```

$ telnet nyx 80
Trying 134.59.22.1...
Connected to nyx.unice.fr.
Escape character is '^]'.
GET /xxx.html HTTP/1.0

HTTP/1.1 404 Not Found
Date: Wed, 06 Feb 2002 15:51:16 GMT
Server: Apache/1.3.9 (Unix) Debian/GNU ApacheJServ/1.1 PHP/4.0.3p11
Connection: close
Content-Type: text/html; charset=iso-8859-1

```

```

WARNING : document is missing on this server
Connection closed by foreign host.

```

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)

```

(par exemple, essayer la méthode OPTIONS avec le protocole 0.9 qui est pris par défaut)
$ telnet nyx 80
Trying 134.59.22.1...
Connected to nyx.unice.fr.
Escape character is '^]'.
OPTIONS /
The server made an Internal error.Connection closed by foreign host.

```

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 telnet `nyx` GET `Host` des ressources différentes en fonction du nom de l'hôte (cette pratique est connue sous le nom de *virtual host*)

```

$ telnet nyx 80
Trying 134.59.22.1...
Connected to nyx.unice.fr.
Escape character is '^]'.
GET /test0.html HTTP/1.1
Host: nyx.unice.fr

HTTP/1.1 200 OK
Date: Thu, 14 Feb 2002 09:46:20 GMT
Server: Apache/1.3.9 (Unix) Debian/GNU ApacheJServ/1.1 PHP/4.0.3p11
Last-Modified: Sun, 11 Nov 2001 12:12:09 GMT
ETag: "218009-2c-3bee6b19"
Accept-Ranges: bytes
Content-Length: 44
Content-Type: text/html; charset=iso-8859-1

```

```

<html><body>
<h1>Test 0</h1>
</body></html>
HEAD /test0.html HTTP/1.1
Host: nyx.unice.fr

HTTP/1.1 200 OK
Date: Thu, 14 Feb 2002 09:46:41 GMT
Server: Apache/1.3.9 (Unix) Debian/GNU ApacheJServ/1.1 PHP/4.0.3pl1
Last-Modified: Sun, 11 Nov 2001 12:12:09 GMT
Last-Modified: Sun, 11 Nov 2001 12:12:09 GMT
ETag: "218009-2c-3bee6b19"
Accept-Ranges: bytes
Content-Length: 44
Content-Type: text/html; charset=iso-8859-1

OPTIONS / HTTP/1.1
Host: nyx.unice.fr
Connection: close

HTTP/1.1 200 OK
Date: Thu, 14 Feb 2002 09:46:59 GMT
Server: Apache/1.3.9 (Unix) Debian/GNU ApacheJServ/1.1 PHP/4.0.3pl1
Content-Length: 0
Allow: GET, HEAD, POST, PUT, DELETE, CONNECT, OPTIONS, PATCH, PROPFIND, PROPPATCH,
MKCOL, COPY, MOVE, LOCK, UNLOCK, TRACE
Connection: close

Connection closed by foreign host.

```

Requêtes conditionnelles

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

```

$ telnet nyx 80
Trying 134.59.22.1...
Connected to nyx.unice.fr.
Escape character is '^]'.
GET /test0.html HTTP/1.0
If-Modified-Since: Wed, 13 Feb 2002 13:42:39 GMT

HTTP/1.1 304 Not Modified
Date: Thu, 14 Feb 2002 09:55:58 GMT
Server: Apache/1.3.9 (Unix) Debian/GNU ApacheJServ/1.1 PHP/4.0.3pl1
Connection: close
ETag: "218009-2c-3bee6b19"

Connection closed by foreign host.

```

2. Analyser la configuration du serveur Apache

Le serveur **Apache** sur *nyx* est installé dans `/etc/apache-ssl`

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* ?

```
ServerType standalone
```

Le serveur s'exécute comme un démon classique, il n'est pas démarré par `inetd` pour des raisons de performances.

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

```
Port 80
```

c'est la valeur par défaut.

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

```
User www-data
Group www-data
```

Si le serveur est démarré par un utilisateur autre que *root*, tous les processus appartiendront à cet utilisateur. S'il est démarré par *root*, le processus père appartiendra à *root*, mais tous les processus fils qui répondent aux requêtes, appartiendront à l'utilisateur défini par les directives User et Group.

Ressources du serveur

– Emplacement des documents HTML

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

```
<VirtualHost www-iutinfo.unice.fr:80>
DocumentRoot /home/www/site.DEPTINFO/htdocs
</VirtualHost>
```

```
<VirtualHost intra-iutinfo.unice.fr:80>
DocumentRoot /home/www/site.INTRA/htdocs
</VirtualHost>
```

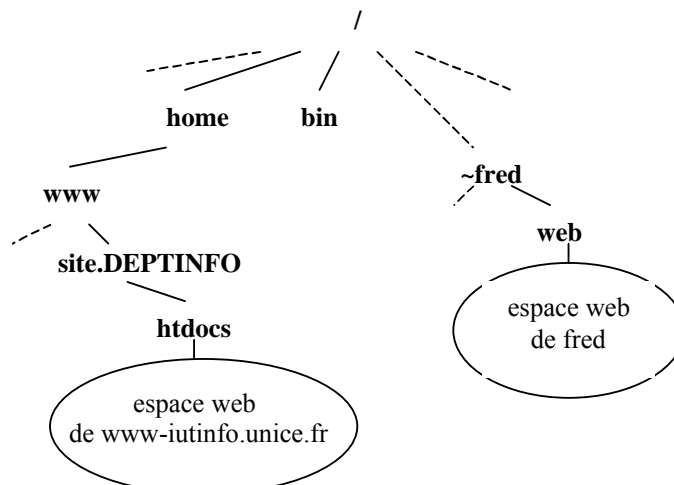
```
. . . . .
<VirtualHost test-iutinfo.unice.fr:80>
DocumentRoot /home/www/site.TEST/htdocs
</VirtualHost>
```

La directive DocumentRoot définit le répertoire racine de chaque serveur virtuel, il représente la partie de l'espace disque de la machine qui sera accessible via le serveur httpd, c'est à dire l'endroit où seront déposés les fichiers HTML que l'on veut diffuser.

2.5. Où sont stockés les documents HTML des utilisateurs sur *nyx* ?

```
UserDir web
```

La directive UserDir permet de rendre accessibles des documents appartenant au répertoire personnel des utilisateurs de la machine, elle permet de définir le nom du répertoire que les utilisateurs devront créer dans leur répertoire personnel. Pour accéder à ces documents, une URL devra préciser le *home directory* de l'utilisateur souhaité selon la syntaxe *-nom_utilisateur* et le nom d'un fichier contenu dans le répertoire correspondant à la directive UserDir.



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* ?

```
ScriptAlias /cgi-bin/ /home/www/site.DEPTINFO/cgi-bin/
```

Le répertoire */home/www/site.DEPTINFO/cgi-bin* héberge les scripts CGI du serveur *www-iutinfo.unice.fr*. Une URL référant un script sera de la forme http://nyx.unice.fr/cgi-bin/nom_du_script

```
AddHandler cgi-script .cgi
```

Cette directive permet de placer des scripts CGI n'importe où, dans l'espace web du serveur à condition de donner au script le suffixe `.cgi`. Lorsque le serveur reçoit une requête pour une URL référant un fichier avec le suffixe `.cgi`, il considère que ce fichier est un script à exécuter.

– 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`.

```
http://www-iutinfo.unice.fr/doc1.html
```

```
http://www-iutinfo.unice.fr/~fred/doc2.html
```

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`.

```
http://www-iutinfo.unice.fr/cgi-bin/script1.sh
```

```
http://www-iutinfo.unice.fr/~fred/script2.cgi
```

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

1. Renvoyer au client le document `/home/www/site.DEPTINFO/htdocs/index.html`

2. Renvoyer au client le document `~fred/web/index.html`

3. Renvoyer au client le document `~fred/web/doc.html`

4. Exécuter le script `/home/www/site.DEPTINFO/cgi-bin/script.sh` et renvoyer le résultat au client

5. Exécuter le script `~fred/web/script.cgi` et renvoyer le résultat au client

Mettre en place votre espace WEB sur nyx

2.12. Testez dans votre navigateur l'url : <http://nyx.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.

Placer dans `R1` le fichier `.htaccess` qui devra contenir les lignes suivantes (où `134.59.28.64` est supposé être votre adresse IP)

```
AuthUserFile /dev/null
```

```
AuthGroupFile /dev/null
```

```
AuthName AccessRestreint
```

```
AuthType Basic
```

```
<limit GET>
```

```
order deny,allow
```

```
deny from all
```

```
allow from 134.59.28.64
```

```
</limit>
```

– 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 *steeve* avec le mot de passe *steeve****. Vérifiez.

La première étape va constituer en la création de la base de données dont le nom est **.htpasswd** qui va contenir les personnes autorisées à accéder aux documents. Un utilitaire **htpasswd** est fourni par Apache pour créer et gérer cette base de données.

Créer d'abord le répertoire (*bdgpwd* par exemple) qui va contenir la base de données et lui mettre les droits de lecture pour tout le monde. Ce répertoire peut se situer dans votre répertoire home.

```
nyx$ cd
nyx$ mkdir bdgpwd
nyx$ chmod o+rx bdgpwd
```

Créer ensuite simultanément (option **-c**) la base de données *.htpasswd* et le premier utilisateur

```
nyx$ htpasswd -c bdgpwd/.htpasswd david
Adding password for david.
New password:
Re-type new password:
```

Créer d'autres utilisateurs (sans l'option **-c**)

```
nyx$ htpasswd bdgpwd/.htpasswd steeve
Adding user steeve
New password:
Re-type new password:
```

On peut visualiser le contenu de la base :

```
nyx$ cat bdgpwd/.htpasswd
david:PxxCs.Gr1AY9k
steeve:zSURHLNa4MTOY
```

Se positionner ensuite dans le répertoire R2 à protéger, créer dans ce répertoire le fichier **.htaccess** qui devra contenir les lignes suivantes (où *my-username* représente votre nom d'utilisateur sur *nyx*) :

```
AuthUserFile /home/students/dut_trad/promo_flip/my-username/bdgpwd/.htpasswd
AuthGroupFile /dev/null
AuthName ByPassword
AuthType Basic
<limit GET>
require valid-user
</limit>
```

Créer un fichier *index.html* dans R2, puis tester

– Trace des accès clients

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

```
TransferLog /home/www/site.DEPTINFO/logs/access_log
```

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

```
nyx$ tail /home/www/site.DEPTINFO/logs/access_log
```

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>
  </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 TELNET sur *nyx*.

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

3.5.Par FTP, transférez le fichier `1.1.html` sur *nyx*

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

Styles, listes

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

```
<i>italique</i>
<b>gras</b>
<em>renforcement</em>
<p>Une liste a deux elements :</p>
<p></p>
<ul>
  <li>pain</li>
  <li>confiture</li>
</ul>
<p></p>
<pre xml:space = "preserve">
  Nom      Prenom
  -----
  Dupont   Jean
  Durand   Pierre
</pre>
```

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

```
<font size="1">1</font>
<font size="2">2</font>
<font size="3">3</font>
<font size="4">4</font>
<font size="5">5</font>
<font size="6">6</font>
<font size="7">7</font>
<font size="4" color="#ff0000">rouge</font>
```

3.9.Comment mettre des caractères spéciaux et accentués ?

Les fichiers HTML doivent être en ASCII ; si on veut mettre des caractères spéciaux ou accentués comme ç, ö ou î, il faut utiliser des séquences d'échappement comme : `ç` `ö` `î`

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

```
INT&Eacute;R&Eccedil;T
```

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).

```
<h1>Sommaire des TD</h1>
<ol start="4">
  <li><a href="TD4">Apprentissage des balises HTML</a></li>
  <li><a href="TD5">Programmation CGI</a></li>
  <li><a href="TD6">Traitement des formulaires HTML</a></li>
</ol>
```

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

```
<a href="/">DEP. INFO</a>
<a href="/~binome/rx1/">Voir rx1 chez mon bin&ocirc;me</a>
```

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.

```
<address>
  <a href="mailto:me@nyx.unice.fr">Me@nyx.unice.fr</a>
</address>
```

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 html, 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/chatol.gif` à la racine du serveur web sur *nyx*.



Le château Valrose

```
Le château Valrose

```

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

```
Le château Valrose
<a href="/images/chatol.gif"></a>
```

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*).

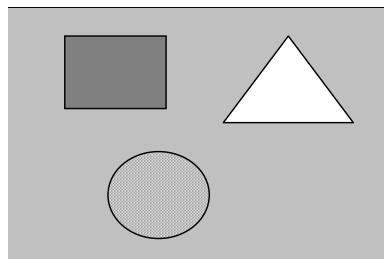
```
<body bgcolor="snow" text="RoyalBlue">
```

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*.

```
<body background="/images/fonds/fondecri.gif">
```

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.

```

<map name="figures" id="figures">
  <area shape="rect" href="rect.html" coords="38,18,104,65" alt=""/>
```

```
<area shape="circle" href="cercle.html" coords="100,124,67,124" alt="" />
<area shape="poly" href="triangle.html" coords="145,74,230,74,188,18" alt="" />
</map>
```