|  |  |
| --- | --- |
| logo | Universidad de Córdoba |
| Letras-iscbd | Programación y administración de sistemas |
|  |  |
|  |  |
| GRADO DE INGENIERÍA INFORMÁTICA | |
| PROGRAMACIÓN Y ADMINISTRACIÓN DE SISTEMAS | |

|  |
| --- |
|  |
| PRÁCTICA 1: APACHE |

|  |
| --- |
|  |
| Autor/es Francisco Bérchez Moreno |
|  |

|  |  |
| --- | --- |
|  | EJERCICIO 1 |
|  |  |

**Recorre las carpetas del servidor observando qué se guarda en ella. Necesitarás acceder al archivo httpd.conf para modificar la configuración de Apache.**

La carpetas que tenemos son:

* bin: Contiene los scripts y códigos requeridos para que se ejecute el servidor. Dentro de dicha carpeta encontraremos el archivo apachectl que es el archivo que necesitaremos para arrancar el servidor apache y el que se utiliza para verificar la modificaciones que realizamos en el archivo de configuración, asi como para parar apache.
* Build: contiene los archivos make que se utilizarán para compilar las diferentes partes que forman Apache.
* cgi-bin: es una carpeta en la que se contienen aquellos archivos que se necesitan para que se le muestre al usuario archivos ya sean de audio, video, texto, imágenes... Cuando este realiza una petición cgi (como consecuencia de una petición cgi se obtiene un objeto mime que es este objeto de audio, video, texto, imágenes...)
* conf: es la carpeta de configuración del servidor apache y dentro de ella encontraremos un archivo importante llamado httpd.conf que es la carpeta en la que realizaremos la modificaciones pertinentes de apache.
* error: es una carpeta en la que están contenidos todos los errores que nos pueden surgir al utilizar html de forma incorrecta además de contener un archivo que lo muestre en varios idiomas.
* htdocs: en esta carpeta se guarda el index de http el cual se abrirá en nuestro navegador para mostrarnos si nuestro servidor apache se inició correctamente
* icons: contiene iconos del servidor apache que sirven para html
* include: es una carpeta donde están contenidos todos los ficheros de cabecera de apache en las que se encuentran las definiciones de las funciones que necesita utilizar apache
* lib: creo que es una carpeta en la que están contenidas todas las librerías de las que debe disponer apache
* logs: en esta carpeta encontramos los archivos en los que quedan registrados los accesos a nuestro servidor, ya sean erróneos o correctos.
* Man: los archivos que contiene nos permiten hacer consultas sobre el servidor Apache.
* manual: contiene los manuales de usuario de apache
* modules: en esta carpeta encontramos todos los módulos de los que dispone nuestro servidor Apache, estando muchos de estos cargados en nuestro archivo de configuración.

|  |  |
| --- | --- |
|  | EJERCICIO 2 |
|  |  |

**Inicia y para el servidor Apache con apachectl. Recuerda que tendrás que llamar a apachectl restart cada vez que hagas un cambio en la configuración.**

Primero entro en la carpeta httpd y una vez dentro de ella entro en la carpeta bin donde encontraré el archivo apachectl y lo ejecutaré con ./apachectl.

La instrucción que tenemos que añadir en $HOME/.bash\_logout que sería killall -s KILL httpd, se encuentra en el archivo oculto del home .bash\_logout al que accedo con el comando *ls -a* y con el comando de edición de ficheros de texto plano gedit introduciría la instrucción (gedit .bash\_logout), sin embargo esta instrucción de killall ya viene introducida.

Da un error al intentar abrir el fichero concretamente este:

|  |
| --- |
| AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 172.16.210.2. Set the 'ServerName' directive globally to suppress this message  (13)Permission denied: AH00072: make\_sock: could not bind to address 0.0.0.0:80 no listening sockets available, shutting down  AH00015: Unable to open logs |

|  |  |
| --- | --- |
|  | EJERCICIO 3 |
|  |  |

**Por defecto, Apache escucha por el puerto 80. Comprueba si esto produce algún error e investiga cuál es el motivo. Modifica el puerto de escucha para que sea el 8080. Comprueba que funciona con un navegador, accediendo a localhost:8080.**

Analizaremos el error que nos da, si el puerto especificado en la directiva listen del fichero de configuración es el que viene pos defecto, es decir, el puerto 80 (o cualquier otro puerto por debajo del 1024), entonces es necesario tener privilegios de usuario root y como en los ordenadores de la uco carecemos de dichos permisos, por lo que necesitaremos cambiar dicho puerto de escucha para poder iniciar Apache, para ello nos iremos al archivo de configuración de Apache httpd.conf y buscaremos la expresión del puesto de escucha listen y la cambiaremos de 80 a 8080 (aunque para trabajar de manera remota el puerto de escucha que he utilizado es el 8085) quedándose de la siguiente manera (Listen 8080).

A continuación, ejecutamos en el terminal el comando ./apachectl restart para confirmar los cambios realizados en el archivo de configuración y obtenemos en terminal lo siguiente:

|  |
| --- |
| AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 172.16.210.2. Set the 'ServerName' directive globally to suppress this message httpd not running, trying to start |

Esto nos quiere decir que lo que tenemos que hacer es cambiar el nombre del servidor al indicado en el ejercicio (localhost:8080), por lo que volvemos al archivo de configuración de Apache y buscamos el servername en el que nos sale comentado que en el caso de que no tengamos un nombre en el DNS que lo introduzcamos justamente abajo, quedando el servername como el mencionado en el ejercicio:

|  |
| --- |
| # If your host doesn't have a registered DNS name, enter its IP address here.  #  ServerName localhost:8080 |

Una vez realizado esto volvemos ha ejecutar el comando en el terminal ./apachectl restart para recargar y el fichero de configuración y aplicar los cambios y por último, volvemos a ejecutar el comando ./apachectl y obtenemos lo siguiente en el terminal:

|  |
| --- |
| httpd (pid 3191) already running |

Para finalizar, abriremos una nueva página en el navegador y escribimos como url localhost:8080 y se nos abrirá una pagina que nos dirá **IT WORKS!**.

|  |
| --- |
|  |

|  |  |
| --- | --- |
|  | EJERCICIO 4 |
|  |  |

**Los archivos que sirve Apache (los que puedo ver desde el cliente) están en una carpeta concreta. Cambia dicha carpeta (DocumentRoot), para que sea $HOME/httpd-docs/. Crear la carpeta y copiar dentro un conjunto de archivos .html para tu servidor web, dándole como nombre a uno de ellos index.html e introduciendo en él tu curso y tu asignatura preferida. Puedes buscar en internet archivos básicos html.**

Las paginas webs que se publican a través del servidor web de Apache se encuentran alojadas en una carpeta del mismo equipo. La ruta de dicha carpeta debe aparecer en el archivo de configuración (httpd.conf) usando la directiva DocumentRoot.

En un principio tenemos esto:

|  |
| --- |
| DocumentRoot "/home/i72bemof/httpd/htdocs"  <Directory "/home/i72bemof/httpd/htdocs"> |

Y lo cambiaremos a esto:

|  |
| --- |
| DocumentRoot "/home/i72bemof/httpd-docs"  <Directory "/home/i72bemof/httpd-docs"> |

Una vez lo cambiamos tendremos que volver a escribir en el terminal ./apachectl restart el cual nos dará error en caso de que en nuestra carpeta home no tengamos una carpeta que se llame httpd-docs, ya que no será capaz de leer una serie de archivos que no tenemos creados:

|  |
| --- |
| AH00526: Syntax error on line 218 of /home/i72bemof/httpd/conf/httpd.conf:  DocumentRoot '/home/i72bemof/httpd-docs/' is not a directory, or is not readable |

Una vez hemos creado la carpeta y cambiado el directorio del DocumentRoot, nos dirá que incluyamos en dicha carpeta una serie de documentos para nuestro servidor web, pero en un inicio lo único que tendremos que crear será un archivo en concreto que es el que nos dice el ejercicio, el cual llamaremos index.html, que es el archivo que tomará para abrir el navegador desde el servidor localhost:8080

El cambiar el directorio del DocumentRoot significa que si ahora nosotros en el terminal ponemos el servidor localhost:8080, nos mostrará en el navegador el curso en el que estamos y nuestra asignatura preferida.

|  |
| --- |
|  |

|  |  |
| --- | --- |
|  | EJERCICIO 5 |
|  |  |

**Cambia el nombre de los archivos índice (archivos que servirá Apache cuando el cliente sólo especifique un directorio). Normalmente es index.html, añadir como posibilidad index2.html y probar si funciona. ¿Qué prioridad se utiliza si tenemos más de un directorio índice?**

En un inicio tenemos en nuestro archivo de configuración (httpd.conf) como DirectoryIndex lo siguiente:

|  |
| --- |
| DirectoryIndex index.html |

Lo que haremos será añadir una nueva alternativa a dicho DirectoryIndex la cual será idex2.html y probar si funciona:

|  |
| --- |
| DirectoryIndex index.html  DirectoryIndex index2.html |

También lo puedo poner de la siguiente manera:

|  |
| --- |
| DirectoryIndex index.html index2.htm |

Una vez ejecutamos en el terminal la instrucción ./apachectl restart vemos que no se genera ningún error al tener dos index, las prioridades que se generan al tener varios index serán siempre el primero que tenemos escrito en caso de que no lo especifiquemos, es decir, si lo tenemos en una sola linea nos abrirá el primer index empezando por la izquierda sea el que sea, y en caso de ponerlos uno debajo del otro, cogerá el primer index empezando por arriba sea el que sea.

|  |  |
| --- | --- |
|  | EJERCICIO 6 |
|  |  |

**¿Qué opción de qué directiva es la encargada de permitir mostrar el contenido de un directorio aunque éste no contenga ningún archivo índice? Crea un directorio nuevo $HOME/httpd-docs/nexFolder (es obligatorio especificar una nueva directiva Directory), incluye un archivo cualquiera y prueba a utilizar esta opción para prohibir o no el listado de archivos en dicha carpeta. Para probarlo tendrás que acceder a localhost:8080/newFolder.**

**¿Cómo podríamos mejorar el aspecto visual del listado? (pistas, Indexes y httpd-autoindex.conf)**

Lo único que tenemos que hacer es crear dentro de nuestra carpeta httpd-docs una nueva carpeta a la cual llamaremos newFolder y en la cual para probarla vamos a introducir un index cualquiera.

La directiva que tendríamos que modificar sería la de DocumentRoot, en la cual si ahora queremos que en vez de que coja los index de la carpeta httpd-docs, queremos que los coja de la carpeta newFolder, la nueva directiva quedaría de la siguiente manera:

|  |
| --- |
| DocumentRoot "/home/i72bemof/httpd-docs/newFolder"  <Directory "/home/i72bemof/httpd-docs/newFolder"> |

Así siempre que ejecutemos en el navegador el servidor localhost:8080 se nos abrirá el index de newFolder, en caso de no cambiar la directiva para acceder al index de dicha carpeta en nuestro navegador, accederemos a través de localhost:8080/newFolder

Para mejorar el aspecto visual de los listados de Apache, primero tenemos que saber como podemos conseguir que nos salga dicho listado en el navegador, el cual se consigue no poniendo ningún index para que se abra cuando pongamos localhost:8080 para ello el fichero de configuración lo podemos dejar de la siguiente manera:

|  |
| --- |
| <IfModule dir\_module>  DirectoryIndex  # DirectoryIndex index.html index2.html  # DirectoryIndex index.html  </IfModule> |

A la hora de mejorar su aspecto visual lo que haremos será identificar primero que hay dentro del archivo httpd-autoindex.conf que como podremos observar la función que tiene es mejorar el aspecto visual de dicho listado añadiendo entre otras cosas iconos. Para conseguir que salgan dichos iconos lo que tenemos que hacer es en el mismo archivo httpd.conf descomentar un apartado, que es el que esta relacionado con el autoindex pasando el archivo de conf de esto:

|  |
| --- |
| # Fancy directory listings  # Include conf/extra/httpd-autoindex.conf |

a esto:

|  |
| --- |
| # Fancy directory listings  Include conf/extra/httpd-autoindex.conf |

Y ahora una vez pongamos en nuestro navegador localhost:8080 nos saldrá el listado de index pero con un mejor aspecto visual.

|  |
| --- |
|  |

|  |  |
| --- | --- |
|  | EJERCICIO 7 |
|  |  |

**Encuentra la directiva que especifica el nombre del servidor e introduce su valor correcto.**

El nombre del servidor se encuentra en el archivo de configuración en el cual aparece como Servername, de modo que si lo buscamos veremos que lo hemos modificado a localhost:8080, siendo el nombre del servidor localhost y 8080 será el puerto de escucha por el que es el servidor va a admitir solicitudes web.

|  |  |
| --- | --- |
|  | EJERCICIO 8 |
|  |  |

**Encuentra la directiva que especifica el usuario y grupo para el demonio httpd. ¿Sirve para algo cambiarla?**

Si buscamos el usuario y el grupo de Apache en el archivo de configuración del mismo (httpd.conf) lo que encontramos es lo siguiente:

|  |
| --- |
| <IfModule unixd\_module>  #  # If you wish httpd to run as a different user or group, you must run  # httpd as root initially and it will switch.  #  # User/Group: The name (or #number) of the user/group to run httpd as.  # It is usually good practice to create a dedicated user and group for  # running httpd, as with most system services.  #  User daemon  Group daemon |

La directiva de usuario, establece la ID del usuario como el servidor que responderá a la solicitudes. Pero para utilizar esta directiva, el servidor debe ejecutarse inicialmente como root. En caso de iniciarlo como usuario no root, no podrá cambiar al usuario con privilegios menores, en su lugar, continuará ejecutándose como usuario original.

Debido a ciertos inconvenientes que se generan al ejecutar el servidor con usuario no root, es recomendable que se configure un nuevo usuario y grupo específico para ejecutar el servidor.

La directiva de grupo, establece el grupo bajo el servidor que responde las peticiones. Para usar esta directiva, el server debe ejecutarse inicialmente como root, en caso de ejecutarlo como no root, dará un fallo cuando queramos especificar un grupo, y en su lugar continuará ejecutándose como el grupo del usuario original. Se recomienda que configure un nuevo grupo específicamente para ejecutar el servidor.

|  |  |
| --- | --- |
|  | EJERCICIO 9 |
|  |  |

**Prueba a hacer un telnet al puerto del servidor (telnet localhost 8080) y a mandarle mensajes HTML (escribe “GET / HTTP/1,0” y pulsa dos veces intro). Prueba a escribir HOLA y pulsar dos veces intro. Explica qué sucede y los códigos de error que devuelve el servidor.**

Telnet puede utilizarse para intentar conectarse a un servidor en un puerto determinado. En general, sirve para determinar si algo está escuchando en el lado remoto, lo que confirma que se puede acceder al servidor remoto y que un servidor se está ejecutando y está vinculado al puerto desde el cual se ha solicitado.

La solución de problemas de HTTP mediante telnet es una excelente manera de revisar exactamente que se devuelve a su navegador durante una solicitud web para una URL en particular. Esto le permite ver encabezados y recopilar otra información relacionada con el servidor web.

Primero obtenemos una salida en el terminal en la que se nos muestra por pantalla que se está intentando conectar al localhost:

|  |
| --- |
| i72bemof@NEWTS:~$ telnet localhost 8085  Trying 127.0.0.1...  Connected to localhost.  Escape character is '^]'. |

Luego tenemos una entrada, en la que introduciremos una petición solicitando un documento completo por lo que utilizamos GET en vez de HEAD ya que este solo solicita las cabeceras:

|  |
| --- |
| GET / HTTP/1.0 |

Por último, obtenemos una salida por pantalla de la petición que acabamos de realizar:

|  |
| --- |
| <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">  <html>  <head>  <title>Index of /</title>  </head>  <body>  <h1>Index of /</h1>  <table>  <tr><thvalign="top"><imgsrc="/icons/blank.gif" alt="[ICO]"></th><th><ahref="?C=N;O=D">Name</a></th><th><ahref="?C=M;O=A">Lastmodified</a><th><th><ahref="C=S;O=A">Size</a><th><th><ahref="C=D;O=A">Description<a><th><tr><tr><thcolspan="5"><hr><th><tr><tr><tdvalign="top"><imgsrc="/icons/text.gif"alt="[TXT]"><td><td><ahref="index.html">index.html</a></td><tdalign="right">2019-02-1812:23</td><tdalign="right">155</td><td>&nbsp;</td></tr><tr><tdvalign="top"><imgsrc="/icons  text.gif"alt="[TXT]"></td><td><ahref="index2.html">index2.html</a></td><td align="right">2019-02-18 13:02 </td><td align="right"> 54 </td><td>&nbsp;</td></tr><tr><tdvalign="top"><img src="/icons/folder.gif"alt="[DIR]"></td><td><ahref="newFolder/">newFolder/</a></td><td align="right">2019-02-1813:21 </td><td align="right"> - </td><td>&nbsp;</td></tr>  <tr><th colspan="5"><hr></th></tr>  </table>  </body></html>  Connection closed by foreign host.  i72bemof@NEWTS:~$ |

En caso de que realicemos una segunda petición (HOLA), en este caso obtenemos lo siguiente:

Primero obtenemos una salida en el terminal en la que se nos muestra por pantalla que se está intentando conectar al localhost.

|  |
| --- |
| i72bemof@NEWTS:~$ telnet localhost 8085  Trying 127.0.0.1...  Connected to localhost.  Escape character is '^]'. |

A continuación, introducimos nuestra petición HOLA:

|  |
| --- |
| HOLA / HTTP/1.0 |

Por último, obtenemos una salida como la siguiente:

|  |
| --- |
| HTTP/1.1 501 Not Implemented  Date: Thu, 21 Feb 2019 19:46:52 GMT  Server: Apache/2.4.38 (Unix)  Allow: GET,POST,OPTIONS,HEAD,TRACE  Content-Length: 192  Connection: close  Content-Type: text/html; charset=iso-8859-1  <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">  <html><head>  <title>501 Not Implemented</title>  </head><body>  <h1>Not Implemented</h1>  <p>HOLA to / not supported.<br />  </p>  </body></html>  Connection closed by foreign host.  i72bemof@NEWTS:~$ |

|  |  |
| --- | --- |
|  | EJERCICIO 10 |
|  |  |

**Encuentra la directiva que se utiliza para la visualización de páginas de error. Modifícala para personalizar el mensaje de error 404 que mostrará el servidor y que sea una cadena del tipo “URI not found”. ¿Se podría especificar un fichero .html de error). Modifica el error 501 para que muestre “Method not implemented” y comprueba que funciona.**

Tenemos que buscar la directiva ErrorDocument, la cual nos comenta que en caso de error Apache nos deja configurar una de las siguientes caracteristicas:

1. Mostrar un simple mensaje de error codificado
2. Mostrar un mensaje personalizado
3. Redireccionar a una ruta-URl local para manejar el problema o el error
4. Redireccionar hacia una URL externa para manejar el problema o el error

Las opciones 2 y 4 son las configuradas usando la directiva ErrorDocument, a la cual le sigue una respuesta de HTTP y una URL o mensaje.

Dichas URLs pueden comenzar con una barra (/) para rutas web locales (está relacionado con la directiva documentRoot).

Hay que tener en cuenta que cuando se especifica un ErrorDocument que apunta a una URL remota, Apache HTTP Server enviará una redirección al cliente para indicarle donde encontrar el documento, incluso si el documento termina estando en el mismo servidor. Por lo tanto si usa una directiva ErrorDocument 401, debe hacer referencia a un documento local, por eso si hacemos referencia a un documento local que no exista saltará dicho error con el mensaje que le hayamos especificado.

Para la realización en el archivo de configuración de Apache buscamos la directiva ErrorDocument 401, la descomentamos y cambiamos el mensaje que nos proporcionara a URI not found.

|  |
| --- |
| ErrorDocument 404 "URI not found" |

Si queremos comprobar que funciona, es tan simple como abrir el navegador y escribir localhost:8080/(un directorio que no exista)

Podemos especificar un fichero .html de error, ya que de hecho en el servidor Apache encontramos un directorio que hace referencia a los errores que pueden especificarse.

A la hora de modificar el error 501 vamos a crear uno nuevo, con la directiva ErrorDocument quedando el fichero de configuración de Apache de la siguiente manera.

|  |
| --- |
| ErrorDocument 501 "Method not implemented" |

Ahora al hacer una petición errónea con telnet nos aparecerá al final el mensaje de “Method not implemented”.

|  |  |
| --- | --- |
|  | EJERCICIO 11 |
|  |  |

**Identifica las directivas relacionadas con los archivos de logs de Apache. Haz un acceso normal y un acceso erróneo (por ejemplo, un error de tipo 501). Comprueba los logs y muestra cómo se han modificado.**

Tenemos el acces\_log en la que el servidor almacena en el registro de acceso información sobre todas las peticiones que procesa. La ubicación del fichero registro y el contenido que se registra se pueden modificar con la directiva customLog.

También encontramos la directiva errorLog, que es el registro de errores del servidor. Apache enviará cualquier información de diagnóstico y registrará cualquier error que encuentre en al procesar peticiones al archivo de registro seleccionado. Es el primer lugar donde tiene que mirar cuando surja un problema al iniciar el servidor durante su operación normal.

El registro de errores se escribe normalmente en un fichero cuyo nombre suele ser error\_log que sen encuentra en nuestro caso en el directorio log junto con el fichero access\_log.

Por ejemplo:

|  |
| --- |
| [Wed Oct 11 14:32:52 2000] [error] [client 127.0.0.1] client denied by server configuration: /export/home/live/ap/htdocs/test |

El primer elemento de la entrada es la fecha y la hora del mensaje. El segundo elemento indica la gravedad del error que se ha producido. La directiva LogLevel se usa para controlar los tipos de errores que se envían al registro de errores según su gravedad. La tercera parte contiene la dirección IP del cliente que generó el error. Después de la dirección IP está el mensaje de error propiamente dicho, que en este caso indica que el servidor ha sido configurado para denegar el acceso a ese cliente. El servidor reporta también la ruta en el sistema de ficheros del documento solicitado.

Ahora veamos cual es la ultima entrada que obtenemos en nuestros ficheros access\_log y error\_log después de realizar un acceso normal:

|  |
| --- |
| acces\_log:  127.0.0.1 - - [02/Mar/2019:12:49:17 +0100] "GET / HTTP/1.0" 200 1117 |
| error\_log:  [Sat Mar 02 12:48:49.276148 2019] [core:notice] [pid 25319:tid 4148266752] AH00094: Command line: '/home/i72bemof/httpd/bin/httpd' |

Como podemos observar no tenemos ningún tipo de advertencia ya que no hemos hecho ningún acceso no permitido, veamos ahora que pasa si hacemos un acceso incorrecto:

|  |
| --- |
| access\_log:  127.0.0.1 - - [02/Mar/2019:12:53:07 +0100] "HOLA / HTTP/1.0" 501 22 |
| error\_log:  [Sat Mar 02 12:53:07.978000 2019] [core:error] [pid 25321:tid 4095716160] [client 127.0.0.1:46132] AH00135: Invalid method in request HOLA / HTTP/1.0 |

Ahora si vemos que en el acceso hemos intentado introducir un método inválido por lo que se ha registrado en el fichero error\_log como un error.

|  |  |
| --- | --- |
|  | EJERCICIO 12 |
|  |  |

**Redirecciona la dirección google a** [**www.uco.es**](http://www.uco.es/)**, de manera que al acceder a localhost:8080google aparezca la web de la Universidad de Córdoba.**

La directiva de configuración redirect esta ubicada en el archivo de configuración de Apache y tiene la siguiente estructura:

|  |
| --- |
| Redirect/nombredeusuariohttp://equipo.ejemplo.com/~nombredeusuario/ |

Si no se especifica un argumento, redirect envía un estatus temporal (por ejemplo: 302). En este caso se informa al cliente que el recurso disponible en /nombreusuario ha sido movido temporalmente a la dirección http://equipo.ejemplo.com/~nombredeusuario/.

Las sentencias redirect deben especidicar la ruta completa del recurso a la cual se redirigirá a continuación el nombre del dominio. Estas sentencias también, deben incluir la URL completa de la nueva ubicación del recurso:

Tenemos formas diferentes de configurar dicha directiva como son:

1. Redirect permanent /nombredeusuario http://equipo.ejemplo.com/~nombredeusuario/
2. Redirect temp /nombredeusuario http://equipo.ejemplo.com/~nombredeusuario/
3. Redirect seeother /nombredeusuario <http://equipo.ejemplo.com/> ~nombredeusuario/
4. Redirect gone /nombredeusuario

La primera redirección "permanent" le dice al cliente que este recurso ha sido movido permanentemente, lo cual se corresponde con el estatus 301 de HTTP. El estatus "temp" es el comportamiento predeterminado, especificando que la redirección solo es temporal, esto se corresponde con el estatus 302 de HTTP. El estatus "seeother" envía una señal estatus 303 HTTP que indica que el recurso solicitado ha sido reemplazado por otro recurso. Por último, el estatus "gone" le indica al cliente que el recurso ha sido eliminado permanentemente, esto envía el estatus 410 de HTTP, como un alternativa al estatus 404 que indica no disponible. En el caso de una redirección gone se debe omitir la URL final.

Para realizar la redirección de la página de google no iremos al archivo de configuración de Apache y buscaremos la directiva Redirect y al dejaremos de la siguiente forma:

|  |
| --- |
| # Redirect permanent /foo http://www.example.com/bar  Redirect permanent /google http://www.uco.es |

Una vez la hayamos cambiado realizaremos un ./apachectl restart para confirmar los cambios y a continuación comprobaremos en el navegador que cuando pongamos localhost:8080/google se nos abra la página de la uco.

|  |  |
| --- | --- |
|  | EJERCICIO 13 |
|  |  |

**Crea un Host virtual de manera que cuando un cliente se conecte al servidor usando localhost:8080 el servidor muestre la carpeta raíz original, y cuando se conecte usando IPMAQUINA:8080 muestre la subcarpeta /newFolder. IPMAQUINA es la IP de la máquina en la que estás, puedes averiguarla con ifconfig. Haz que tengan focheros de log independientes (local-access.log, local-error.log, ip-access.log y ip-error.log)**

<VirtualHost> y </VitrualHost> son usadas para agrupar directivas que serán aplicadas solo para un virtual host particular. Cuando el servidor recibe una petición de un documento para un virtual host particular, esta utiliza la configuración de directivas que están agrupadas dentro de la sección <VirtualHost>, un ejemplo de la creación de virtual host es el siguiente:

|  |
| --- |
| <VirtualHost 10.1.2.3>  ServerAdmin webmaster@host.example.com  DocumentRoot /www/docs/host.example.com  ServerName host.example.com  ErrorLog logs/host.example.com-error\_log  TransferLog logs/host.example.com-access\_log  </VirtualHost> |

Nosotros lo único que no vamos a hacer es especificar el ServerAdmin puesto que no nos va resultar necesario, especificaremos la ruta con el documentRoot para saber donde buscar en nuestro caso los archivos index, estableceremos un ServerName para identificar a cada virtual host, en caso de no especificar el ServerName, el servidor intentará deducir el nombre del host realizando una búsqueda inversa en la dirección IP. Por último para tener en ficheros diferentes los archivos de acceso y de error, lo que haré será definir en el VirtualHost de la IP de la máquina un fichero log de acceso y error con las directivas CustomLog y ErrorLog y guardare dichos archivos en el directorio logs.

Una vez seguido estos estos pasos configuramos el VirtualHost en el archivo de configuración de Apache de la siguiente forma:

|  |
| --- |
| <VirtualHost 172.16.210.16:8085>  DocumentRoot /home/i72bemof/httpd-docs/newFolder  ServerName 172.16.210.16:8085  ErrorLog logs/ip-error.log  CustomLog logs/ip-access.log common  </VirtualHost> |

De manera que si ahora nosotros en nuestro navegador introducimos la IP de dicha maquina y (:8085) tendremos acceso a la carpeta newFolder, y además se nos actualizarán los ficheros ip-access.log e ip-error.log guardados en la carpeta logs de Apache.

|  |  |
| --- | --- |
|  | EJERCICIO 14 |
|  |  |

**Haz que el servidor web sirva toda la documentación de Apache. Para ello, habilita la configuración del manual de Apache (extra/httpd-manual.conf) y los módulos que necesite. Explica las distintas líneas que aparecen en dicha configuración, incluyendo el significado de las expresiones regulares.**

En el fichero de configuración de Apache encontramos lo siguiente:

|  |
| --- |
| # Local access to the Apache HTTP Server Manual  #Include conf/extra/httpd-manual.conf |

Lo que vamos a hacer es descomentarlo:

|  |
| --- |
| # Local access to the Apache HTTP Server Manual  Include conf/extra/httpd-manual.conf |

Para obtener la documentación extra veremos que si lo intentamos realizar de golpe obtendremos el siguiente error:

|  |
| --- |
| i72bemof@NEWTS:~/httpd/bin$ ./apachectl restart  AH00526: Syntax error on line 36 of /home/i72bemof/httpd/conf/extra/httpd-manual.conf:  Invalid command 'LanguagePriority', perhaps misspelled or defined by a module not included in the server configuration  i72bemof@NEWTS:~/httpd/bin$ |

Por lo que será recomendable seguir la ruta marcada en dicho include hasta encontrar el fichero httpd-manual.conf en el cual veremos que los módulos requeridos para que dicha función pueda funcionar son los siguientes módulos:

1. mod\_alias
2. mod\_authz\_core
3. mod\_authz\_host
4. mod\_setenvif
5. mod\_negotiation (que en nuestro caso es el único que no se encuentra activo por lo que lo descomentamos para cargarlo)

En dicho fichero también encontramos la directivas:

* AliasMatch: que es equivalente a Alias, pero hace uso de expresiones regulares y dicha expresión regular se compara con la ruta URL.
* RemoveType: elimina cualquier tipo de asociación de medios para archivos con las extensiones proporcionadas. Que en nuestro caso lo que hace es eliminar las extensiones .tr
* AddLanguage: asigna la extensión del nombre del archivo dado el idioma del contenido especificado.

Una vez analizado todo el archivo si nos fijamos en el archivo de configuración de Apache lo único que tendremos que hacer sera activar el módulo negotation, decomentándolo de modo que si ahora hacemos ./apachectl restart ya no nos dará el error de antes.

|  |  |
| --- | --- |
|  | EJERCICIO 15 |
|  |  |

**Apache permite el acceso a recursos restringidos mediante la creación de usuarios y grupos que deberán autentificarse antes de acceder a dicho recurso protegido: Se debe:**

1. **Crear los usuarios y contraseñas correspondientes dentro de un archivo llamado .htpasswd (para esto, tendrás que utilizar el comando ./htpasswd -c .htpasswd usuario, incluido en la carpeta bin de Apache, una vez por cada usuario a añadir y luego copiar el archivo resultante a un sitio protegido; ojo el -c indica que el fichero se cree de nuevo, por lo que sólo debe utilizarse para el primer usuario)**
2. **Incluir un archivo .htaccess (por defecto, aunque estos nombres de archivo pueden modificarse en httpd.conf) con los usuarios o grupos de usuarios que tendrán acceso. La sintaxis de ese archivo es la misma que el resto de configuraciones de directorios de Apache.**
3. **Debes crear un directorio, que se llamará secretFolder, de forma que, para acceder a él, habrá que autentificarse. Los usuarios que tendrán acceso a ese directorio serán: user1, user2 y user3; y la contraseña para todos será: entra. Explica los pasos realizados para conseguirlo y el contenido de los archivos creados.**

Para que el archivo .htaccess se interprete por Apache es necesario que, previamente, hayamos activado la directiva AllowOverride All sobre el directorio correspondiente que por defecto está a None en http.conf, para ello en el final de nuestro archivo de configuración de Apache lo que haremos sera introducir la localización del directorio donde queremos activar dicha directiva, que en nuestro caso es en la carpeta secretFolder, quedando nuestro archivo de configuración de la siguiente manera:

|  |
| --- |
| <Directory "/home/i72bemof/httpd-docs/secretFolder">  AllowOverride All  </Directory> |

Una vez hecho esto lo que haremos será irnos a la carpeta bin del servidor de Apache para actualizar los usuarios y las contraseñas de aquellos usuarios que unicamente podrán acceder a dicha carpeta, para ello desde el terminal introduciremos el siguiente comando ./htpasswd -c .htpasswd usuario, utilizando solo el -c para el primer usuario ya que esta indica que el fichero se cree de nuevo.

Los usuarios serán user1, user2 y user3 teniendo todos la contraseña entra.

Una vez hecho esto pulsaremos ctrl+h en la carpeta bin, para localizar el fichero .htpasswd y copiarlo dentro de nuestra carpeta secretFolder al igual que el archivo .htaccess.

Por último, para que nos pida la contraseña en caso de que queramos entrar a la carpeta secretFolder desde nuestro navegador lo que tendremos que hacer será incluir una serie de directivas en el fichero .htaccess que serán, las siguientes:

|  |
| --- |
| AuthUserFile /home/i72bemof/httpd-docs/secretFolder/.htpasswd  AuthName "Password Required"  AuthType Basic  Require user user1 user2 user3 |

AuthUserFile lo que hará será establecer el archivo al que tendrá que referirse para pedir la contraseña.

AuthName especifica el mensaje que nos saldrá en casod e que queramos acceder a dicha carpeta que en nuestro caso es “Password Required”

AuthType selecciona el método que se usa para autentificar al usuario, siendo el método más común Basic.

Una vez hecho esto, cuando queramos acceder a dicha carpeta desde nuestro navegador se nos requerirá un usuario y una contraseña.

|  |
| --- |
|  |

|  |  |
| --- | --- |
|  | EJERCICIO 16 |
|  |  |

**Existe también la posibilidad de permitir o denegar el acceso a diferentes directorios dependiendo de la dirección IP del cliente. Para ello, disponemos de las directivas Allow from, Deny from y Order, que pueden utilizarse en el archivo de configuración httpd.conf o en cada uno de los directorios mediante el archivo .htaccess. Prueba esta opción de seguridad para permitir el acceso a nuestro servidor web, únicamente para direcciones IP de la subred de los equipos de sobremesa del laboratorio. Prueba también a denegar el acceso a todas las direcciones IP.**

Para bloquear las direcciones IP, primero que tendremos que identificar la IP de nuestro ordenador de la uco, teniendo en cuenta que los dos últimos números de esa dirección son los correspondiente al ordenador con el que estamos trabajando.

A continuación, lo que tendremos que hacer será irnos al archivo de configuración de Apache y modificar la directiva de <Directory>, <\Directory> de la sifuiente forma:

|  |
| --- |
| 172.16.210.16 IP de un ordenador de la uco  <Directory />  AllowOverride None  Require all denied  Order Deny,Allow  Deny from all  Allow from 172.16.215.0/24  </Directory> |

Lo único que tendremos que hacer será incluir la expresión Order en la que definimos que primero denegaremos el acceso nuestro servidor a través de todas las direcciones IP con “Deny from all” y que solo le permitiremos acceder a la dirección IP que hemos especificado con “Allow from 172,16,215,0/24”

Por lo que si intentamos acceder desde una dirección IP que no tengamos permitida obtendremos el siguiente mensaje:

|  |
| --- |
|  |