Programación y Administración de Sistemas

9. Gestión de las comunicaciones

Pedro Antonio Gutiérrez

Asignatura "Programación y Administración de Sistemas"
2º Curso Grado en Ingeniería Informática
Escuela Politécnica Superior
(Universidad de Córdoba)
pagutierrez@uco.es

24 de abril de 2015

Conceptos básicos

Tareas:

- Manejo de la red.
- Monitorizar el tráfico.
- Añadir nuevos *hosts*.
- Montar discos remotos o exportar los discos locales: NFS.
- Servicio de información: usuarios, grupos, etc. (utilización del protocolo NIS).
- Configurar y administrar otros servicios de red (*web*, correo, etc.).
- Prevenir problemas de **seguridad**.
- Enrutado de tráfico.

Conceptos básicos

Labor mínima:

- Opciones de configuración de la red más importantes.
- Entender la configuración de red actual.
- En su caso, programar estrategias de crecimiento de la red, para que la eficiencia pueda mantenerse.

Demonios de red: `xinetd`

- Para administrar servicios en Linux, se puede usar `xinetd`.
  - Maneja a otros demonios, los cuales inicializa cuando hay un trabajo para ellos: `sshd`, `ftpd`, `pop`...
  - /etc/xinetd.conf ⇒ fichero de configuración de `xinetd`.
  - Directorio `/etc/xinetd.d/` ⇒ ficheros de configuración de los demonios gestionados por `xinetd`.

Ejemplo fichero /etc/xinetd.conf

```
1  defaults
2  {
3    instances = 60
4    log_type = SYSLOG authpriv
5    log_on_success = HOST PID
6    log_on_failure = HOST
7    cps = 25 30
8  }
9
10 service ftp
11 {
12   # Unlimited instances because wu.ftpd does its own load management
13   socket_type = stream
14   protocol = tcp
15   wait = no
16   user = root
17   server = /usr/sbin/wu.ftpd
18   server_args = -a
19   instances = UNLIMITED
20   only_from = 128.138.0.0/16
21   log_on_success += DURATION
22 }
23
24 includedir /etc/xinetd.d
25 ...
```

Conceptos generales: algunos demonios de red

- `/etc/init.d/networking` script que activa la red en tiempo de arranque (también `/etc/init.d/network-manager`).
- Algunos demonios:
  - `ntpd` ⇒ demonio encargado de sincronizar la hora del sistema.
  - `dhcpd` ⇒ demonio encargado del servicio de *Dynamic Host Configuration Protocol* (sólo es necesario si el ordenador proporciona IPs privadas a las máquinas que se conecten).
  - `named` ⇒ demonio encargado del servicio de *Domain Name System* (sólo es necesario si el ordenador hace de DNS).
  - `sendmail` ⇒ demonio encargado del correo electrónico.
  - `sshd` ⇒ demonio que permite `ssh` (conexión remota segura).
  - `httpd` ⇒ servidor *web* (normalmente `apache`).
  - `smbd` ⇒ servicio de compartición de ficheros con Windows.

NFS: servicio de archivos compartidos

- Posibilita que un Sistema de Ficheros, que físicamente reside en un *host* remoto, sea usado por otros ordenadores, vía red, como si fuese un sistema de ficheros local.
- En el servidor se indica:
  - Qué sistemas de ficheros se exportan ⇒ Se puede exportar un sistema de ficheros completo o sólo un directorio completo.
  - A qué ordenadores se exportan (se les permite acceder) ⇒ a un equipo concreto o a todos los equipos de una red.
  - Condiciones para la exportación.
- Los equipos cliente montan el sistema de ficheros remoto con la orden `mount` y acceden a los datos como si fuesen locales
  - Incorporan, en cada operación, una *cookie* secreta que se les manda cuando montan el directorio.

NFS: servicio de archivos compartidos

- Al exportar un fichero, se exporta su nodo-i y sus bloques de datos $\Rightarrow$ ¿propietario y grupo propietario?. ¿Qué pasa si en el equipo cliente no existe ese usuario o ese grupo propietario?.
- Un equipo puede ser servidor y cliente NFS al mismo tiempo.

Versiones:

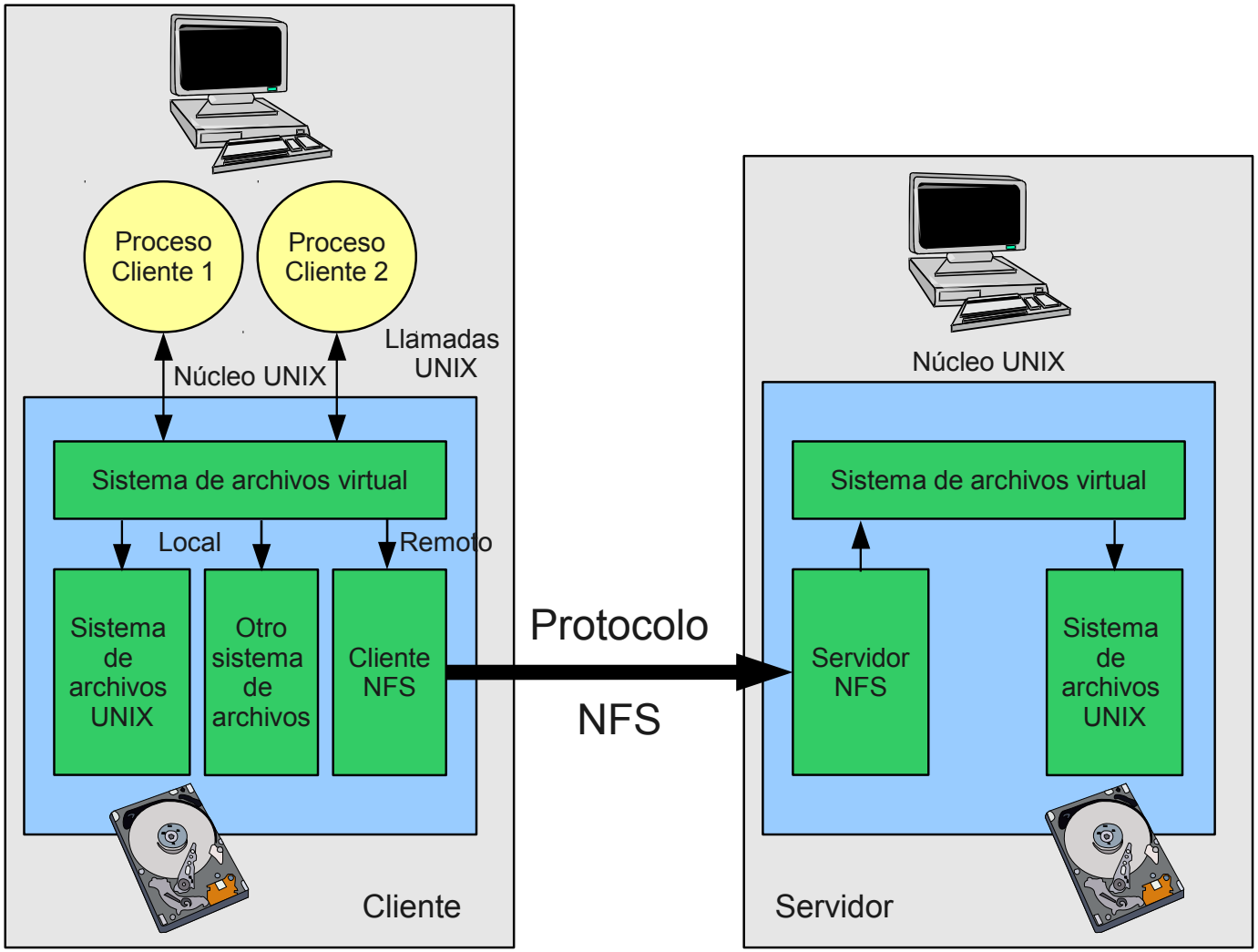
- $NFS \leq 2$: operaciones de escritura bloqueantes (en espera de un ACK).
- $NFS = 3$: esquema de coherencia que permite escrituras asíncronas sin peligro $\rightarrow$ mayor eficiencia.
- $NFS = 4$: incorpora funcionalidades adicionales (montaje, bloqueo, autenticación) dentro del propio protocolo.

NFS: servicio de archivos compartidos

Organización:

- Se basa en el protocolo *Remote Call Procedure* (RPC), para encapsular llamadas al servidor cuando se piden archivos remotos (de manera transparente para el usuario).
- *Stateless*: el servidor trabaja sin mantener información del estado de cada uno de los clientes (ficheros abiertos, último fichero y posición escrita).
  - Necesidad de bloquear archivos accedidos concurrentemente por varios clientes $\rightarrow$ demonios independientes.
  - El cliente es responsable de mantener la coherencia.
- NFS tiene bastantes problemas de seguridad (UID y GID locales, falsificación de direcciones IP, ficheros que pertenecen a `root`...) $\Rightarrow$ uso de herramientas adicionales.

NFS: servicio de archivos compartidos



NFS: configuración del lado servidor

- `/etc/exports` ⇒ Fichero en el que se indica qué SFs se exportan, bajo qué condiciones y a qué ordenadores.
- `/usr/sbin/exportfs` ⇒ Actualiza la información de los SFs exportados y muestra un listado con dicha información (realiza un `restart` de los demonios `nfsd` y `rpc.mountd`):
  - `-r` → re-exporta los directorios indicados en `/etc/exports`.
  - `-a` → exporta o deja de exportar `/etc/exports`.
  - `-v` → muestra los directorios exportados y las opciones.
- `/usr/sbin/showmount` ⇒ información en un servidor NFS:
  - `-a` → clientes conectados y directorios utilizados.
  - `-d` → listado de los directorios montados.

NFS: configuración del lado servidor

Demonios en el lado servidor

- `rpcbind` o `portmap` ⇒ Facilita la conexión entre el cliente y el servidor mediante las llamadas RPC. Tiene que estar lanzado para que NFS funcione.
- `nfsd` ⇒ Implementa, en el nivel de usuario, los servicios NFS. La principal funcionalidad está implementada por el módulo del kernel `nfsd.ko`. Los *threads* del kernel aparecen como `[nfsd]`, al ejecutar `ps aux`.
- `rpc.mountd` ⇒ Maneja las peticiones de montaje de directorios de los clientes, comprobando la petición con la lista de sistemas de ficheros exportados.

`/etc/init.d/nfs-kernel-server` ⇒ Lanza `rpc.mountd` y `rpc.nfsd`.

NFS: configuración del lado servidor

- Opciones en el servidor:
  - `/etc/exports` ⇒ Para configurar qué "directorios" se exportan, bajo qué condiciones y a qué equipos:

```
1 ruta dirección(opción)
```

  - `ruta` es el nombre del directorio a exportar vía NFS.
  - `dirección` a quién es exportado (IP, dirección de red, etc.).
  - `opción` especifica el tipo de acceso al directorio:
    - `rw` ó `ro` → Modo lectura-escritura o sólo lectura.
    - `root_squash` → Mapea los uid/gid 0 a los uid/gid anónimo (**nobody** o **nfsnobody**) (controlar al **root** cliente).
    - `no_root_squash` → No hacer lo anterior (peligro).
    - `all_squash` → Mapea todos los usuarios al usuario anónimo.
    - `anonuid` ó `anongid` → Establecer el `uid` o el `gid` del usuario al que realizar el mapeo, distinto del usuario anónimo.

NFS: configuración del lado cliente

- La misma orden `mount` permite montar el SF remoto:

```
1 $ mount -t nfs -o opciones_nfs 191.168.6.10:/home /datos
```

  - `-t nfs`: tipo de SF.
  - `191.168.6.10:/home` servidor y directorio remoto a montar.

- Si en el fichero `/etc/fstab` se indica el listado de los sistemas de ficheros remotos a montar, el punto de montaje y las opciones, el montaje se puede realizar en tiempo de arranque:

```
1 191.168.6.10:/home  /datos  nfs  defaults,opciones_nfs 0 0
```

NFS: configuración del lado cliente I

- Opciones para `mount`:
  - `soft` ⇒ Si el servidor NFS falla durante un tiempo, las operaciones que intentaban acceder a él recibirán un código de error.
  - `hard` ⇒ Si un proceso está realizando una operación de E/S con un fichero vía NFS y el servidor NFS no responde, el proceso no puede ser interrumpido o matado (no acepta la señal KILL) salvo que se especifique la opción `intr`. Siempre que usemos `rw` deberíamos usar hard, para no dejar el SF remoto inconsistente.
  - `intr` ⇒ Se permite señales de interrupción para los procesos bloqueados en una operación de E/S en un servidor NFS.
  - ⋆ : `soft` va en contra de la filosofía de NFS.

NFS: configuración del lado cliente II

- `bg` ⇒ Si el montaje del SF remoto falla, que siga intentándolo en *background*, hasta que lo consiga o desista porque se han hecho `retry` intentos
- `retry=n` ⇒ Nº de intentos que se deben hacer para montar el SF remoto, antes de desistir si la conexión falla.
- `timeo=n` ⇒ Tiempo a esperar entre cada intento de montaje si la conexión falla.
- `rsize=8192` o `wsize=8192` ⇒ Tamaño de los *buffers* de lectura o escritura.

- automount:
  - Facilita la tarea del montado de los directorios remotos.
  - Evita el caos que supone una caída del servidor.
  - Demonio que monta la carpeta cuando hace falta, y la desmonta cuando no, permitiendo además trabajar con réplicas del servidor.

NFS: ejemplos

- Ejemplos en el **servidor** (fichero `/etc/exports`):

```
1 /home 191.168.6.15(rw,root_squash) 191.168.6.16(rw,no_root_squash)
2 /import 191.168.8.20(rw,all_squash)
3 /tools 191.168.6.0/24(ro,all_squash,anonuid=500,anongid=100)
```

- Ejemplos en el **cliente**:
  - En el fichero `/etc/fstab`:

```
1 julieta:/home        /home          nfs  defaults,rw,bg,hard,intr 0 0
2 julieta:/import      /nfs/import    nfs  defaults,rw,bg,hard,intr 0 0
3 191.168.6.10:/tools  /nfs/tools     nfs  defaults,ro,bg,soft      0 0
```

  - También se puede realizar el montaje de forma manual:

```
1 $ mount /home          #(configurado /etc/fstab)
2 $ mount /nfs/import    #(configurado /etc/fstab)
3 $ mount -t nfs -o rw,bg,hard,intr julieta:/home /home
4 $ mount -t nfs -o rw,bg,hard,intr julieta:/import /nfs/import
5 $ mount -t nfs -o ro,soft,bg 191.168.6.10:/tools /nfs/tools
```

NFS: ejemplos

- Instalación en un sistema Debian/Ubuntu:

```
1  # IP del servidor 150.214.117.142, IP del cliente 172.30.250.242
2  # ------------ Lado servidor
3  pedroa@ayrnapc02:~$ sudo apt-get install nfs-kernel-server
4  pedroa@ayrnapc02:~$ sudo mkdir /home/carpetaNFS
5  pedroa@ayrnapc02:~$ sudo gedit /etc/exports
6  # Incluir en dicho fichero:
7  /home/carpetaNFS 172.30.250.242(rw,no_subtree_check)
8  pedroa@ayrnapc02:~$ sudo chmod o+w /home/carpetaNFS # OJO => Poco seguro
9  pedroa@ayrnapc02:~$ sudo exportfs -r
10 # ------------ Lado cliente
11 pagutierrez@TOSHIBA:~$ sudo apt-get install nfs-common
12 pagutierrez@TOSHIBA:~$ mkdir puntoMontaje
13 pagutierrez@TOSHIBA:~$ sudo mount -t nfs -o hard,intr,bg 150.214.117.142:/home/
       carpetaNFS ./puntoMontaje
14 pagutierrez@TOSHIBA:~$ sudo gedit /etc/fstab
15 # Incluir en dicho fichero:
16 150.214.117.142:/home/carpetaNFS /home/pagutierrez/puntoMontaje nfs user,hard,
       intr,bg 0 0
17 pagutierrez@TOSHIBA:~$ mount ~/puntoMontaje
```

NIS: conceptos básicos

- Ficheros de configuración: en un entorno real, muchos ficheros de configuración son similares de una máquina a otra.
  - Ejemplo: `/etc/passwd` o `/etc/shadow`.
  - n máquinas $\Rightarrow$ n réplicas de los ficheros que debo gestionar.
    - Muy difícil.
    - Los cambios tardan en propagarse.
- *Network Information Service* (**NIS**).
  - Todos los servicios acceden a una misma base de datos de configuraciones.
  - Permite centralizar la autenticación de servicios.
  - Inconvenientes (subsanados por LDAP):
    - Sólo para una subred y no cifra los datos.
    - No permite establecer jerarquías de usuarios complejas.
    - Un cambio $\rightarrow$ reconstruir todo y redistribuirlo.
    - Usuario del servicio $\Leftrightarrow$ usuario sistema operativo.

NIS: conceptos básicos

- **NIS** → servicio de red para compartir cierta información.
- Los ficheros de las bases de datos están en el equipo servidor y contienen información como:
  - *login names* / *passwords* / *home directories* ⇒ `/etc/passwd`.
  - *group information* ⇒ `/etc/group`.
  - ...
- El servidor distribuye esta información a los clientes.
- En el lado servidor:
  - Los ficheros se preprocesan para convertirlos a un formato binario con *hashing* (Berkeley DataBase) (mejor eficiencia).
  - Dominio NIS ⇒ clave para poder localizar al servidor (p.ej. `pas.inf.uco.es` o `pas_nis`).
  - Los ficheros de las BDs residen a partir del directorio `/var/yp/`, en un subdirectorio con el nombre del dominio.

NIS: configuración

- Existe la posibilidad de configurar varios servidores **esclavos**, que tendrán una copia de las bases de datos.
  - Un cliente puede acudir a varios servidores (dominios).
- `NSS (Name Service Switch)`:
  - Indicar como se resolverá cierta información de configuración.
  - `/etc/nsswitch.conf`
- Demonios:
  - `rpcbind` o `portmap` ⇒ Facilita la conexión entre el cliente y el servidor mediante las llamadas RPC (en cliente y en servidor).
  - `ypserv` ⇒ Este demonio es el encargado de gestionar el servicio NIS. Tiene que estar en ejecución en el servidor.
  - `rpc.yppasswdd` ⇒ Permite la actualización de las contraseñas desde los equipos cliente. En ejecución en el servidor.
  - `ypbind` ⇒ Es el encargado de gestionar las peticiones. En el cliente (en el servidor, si se quiere que sea cliente de sí mismo).

NIS: instalación del servidor

- Pasos en el servidor (**Ubuntu/Debian**):
  1. Instalar paquete `nis` (instala `portmap`). Indicar dominio a utilizar (`pas_nis`) y esperar intento fallido de *binding*.
  2. Cambiar el fichero `/etc/default/nis` e indicar `NISSERVER=master`.
  3. Añadir la IP del servidor al fichero `/etc/yp.conf`:

```
1 ypserver localhost
```

  4. Configurar el servidor (crea las bases de datos): `sudo /usr/lib/yp/ypinit -m`.
  5. Reiniciar el servicio: (`sudo /etc/init.d/nis restart`).
  6. Comprobar que todo funciona: `rpcinfo -p`.
  7. Configurar el NSS (`/etc/nsswitch.conf`)

```
1 passwd:         compat nis
2 group:          compat nis
3 shadow:         compat nis
```

NIS: instalación del cliente

- Pasos en el cliente (**Ubuntu/Debian**):
  1. Instalar paquete `nis` (instala `portmap`). Indicar dominio a utilizar (`pas_nis`) y esperar intento fallido de *binding*.
  2. Añadir la IP del servidor al fichero `/etc/yp.conf`:

```
1 ypserver 192.168.117.23
```

  3. Configurar el NSS (`/etc/nsswitch.conf`)

```
1 passwd:         compat nis
2 group:          compat nis
3 shadow:         compat nis
```

  4. Reiniciar el servicio: (`sudo /etc/init.d/nis restart`).
  - ⋆ El dominio por defecto se encuentra en `/etc/defaultdomain`.

NIS: ejemplos I

- Instalación en un sistema Debian/Ubuntu:

```
1  # IP del servidor 150.214.117.142, IP del cliente 172.30.250.242
2  # ------------ Lado servidor
3  pedroa@ayrnapc02:/home$ sudo apt-get install nis
4  pedroa@ayrnapc02:/home$ sudo gedit /etc/default/nis
5  # Cambiar el fichero y poner:
6  NISSERVER=master
7  pedroa@ayrnapc02:/home$ sudo gedit /etc/yp.conf
8  # Añadir a dicho fichero:
9  ypserver localhost
10 pedroa@ayrnapc02:/home$ sudo /usr/lib/yp/ypinit -m
11 pedroa@ayrnapc02:/home$ sudo /etc/init.d/nis restart
12 pedroa@ayrnapc02:/home$ rpcinfo -p
13
14 pedroa@ayrnapc02:/home$ sudo gedit /etc/nsswitch.conf
15 #Añadir nis en las líneas correspondientes:
16 passwd:         compat nis
17 group:          compat nis
18 shadow:         compat nis
19 # ------------ Lado cliente
20 pagutierrez@TOSHIBA:~$ sudo apt-get install nis
21 pagutierrez@TOSHIBA:~$ sudo gedit /etc/yp.conf
22 # Añadir a dicho fichero:
23 ypserver 150.214.117.142
```

NIS: ejemplos II

```
24 pagutierrez@TOSHIBA:~$ sudo gedit /etc/nsswitch.conf
25 #Añadir nis en las líneas correspondientes:
26 passwd:         compat nis
27 group:          compat nis
28 shadow:         compat nis
29 pagutierrez@TOSHIBA:~$ sudo /etc/init.d/nis restart
```

NIS: seguridad

- Utilidades como clientes:
  - `yppasswd`: Permite que los usuarios puedan cambiar su contraseña en el servidor NIS (gracias al demonio `yppasswdd` que se ejecuta en el servidor).
  - `ypchsh`: Permite cambiar el shell del usuario en el servidor NIS.
  - `ypchfn`: Cambia el campo gecos del usuario en el servidor NIS.
  - `ypcat`: Permite conocer el contenido de un mapa NIS. Por ejemplo:
    - `ypcat passwd` → visualiza el fichero de passwords
    - `ypcat ypservers` → muestra los servidores disponibles
  - `ypwhich`: Devuelve el nombre del servidor NIS.

NIS: seguridad

- **Seguridad**: En el fichero `/etc/ypserv.conf` se pueden indicar listas de control de acceso.
- Formato: `host:nisdomain:map:security` (se interpretan por orden):

```
1 128.138.24.0/255.255.252.0:atrustnis:*:none # permite acceso de
      128.138.24/22
2 *:*:passwd.byuid:deny  # deniega acceso a passwd por uid a cualquier
      dominio
3 *:*:passwd.byname:deny # deniega acceso a passwd por name a cualquier
      dominio
4 128.138.:atrustnis:*:port # permite acceso de 128.138/16, siempre que la
      petición provenga de un puerto con privilegios
5 *:*:*:deny # deniega por defecto
```

- Las BDDs se indexan para mejorar el acceso:

```
1 pagutierrez@PEDROLaptop:/var/yp/pas_nis$ ls
2 group.bygid    netgroup.byhost  protocols.byname     services.byservicename
3 group.byname   netgroup.byuser  protocols.bynumber   shadow.byname
4 hosts.byaddr   netid.byname     rpc.byname           ypservers
5 hosts.byname   passwd.byname    rpc.bynumber
6 netgroup       passwd.byuid     services.byname
```

Referencias

Nemeth, Snyder y Seebass.
Linux Administration Handbook
Capítulo 16. *The network file system*.
Capítulo 17. *Sharing file systems*.
Prentice Hall. Segunda Edición. 2007.

Programación y Administración de Sistemas

9. Gestión de las comunicaciones

Pedro Antonio Gutiérrez

Asignatura "Programación y Administración de Sistemas"
2º Curso Grado en Ingeniería Informática
Escuela Politécnica Superior
(Universidad de Córdoba)
pagutierrez@uco.es

24 de abril de 2015