

Programación y Administración de Sistemas

7. Administración de sistemas de ficheros y discos

Pedro Antonio Gutiérrez

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

16 de abril de 2018



Objetivos del aprendizaje I

- Explicar la importancia de la administración del sistema de ficheros en la labor de administración de sistemas informáticos.
- Explicar que sucede en el sistema operativo cuando se montan y desmontan dispositivos en el sistema de ficheros.
- Utilizar la herramienta `mount` (y `umount`) para montar (y desmontar) sistemas de ficheros.
- Configurar los dispositivos de almacenamiento principales del sistema mediante el fichero `/etc/fstab`.
- Explicar la necesidad del chequeo del sistema de ficheros y utilizar las principales herramientas de chequeo.
- Explicar cómo funciona el sistema de *journaling*.
- Enumerar los pasos necesarios para añadir un nuevo dispositivo de disco al sistema operativo.

Objetivos del aprendizaje II

- Explicar las diferencias fundamentales entre los sistemas de ficheros ext2, ext3 y ext4.
- Justificar la necesidad del uso de cuotas de disco en sistemas informáticos compartidos.
- Configurar cuotas de disco para los distintos usuarios de un sistema GNU/Linux.
- Justificar la necesidad del uso de arrays redundantes de discos (RAID) en sistemas informáticos.
- Elegir el mecanismo RAID más adecuado: RAID0, RAID1 o RAID4/5.
- Ejemplificar el mecanismo de redundancia para RAID basado en paridad.
- Montar un sistema RAID por *software* en GNU/Linux.

- 7.1. Introducción
- 7.2. Montaje y desmontaje de sistemas de ficheros.
 - 7.2.1. Concepto.
 - 7.2.2. Herramienta `mount`.
 - 7.2.3. Fichero `/etc/fstab`.
- 7.3. Comprobación del sistema de ficheros.
 - 7.3.1. Concepto y herramientas de chequeo.
 - 7.3.2. Sistema de *journaling* para sistemas de ficheros.
- 7.4. Creación de sistemas de ficheros.
 - 7.4.1. Pasos para la creación de un sistema de ficheros.
 - 7.4.2. Diferencias entre `ext2`, `ext3` y `ext4`.
- 7.5. Aspectos avanzados.
 - 7.5.1. Cuotas.
 - 7.5.1.1. Concepto de cuotas de disco.
 - 7.5.1.2. Pasos necesarios para establecer cuotas de disco.
 - 7.5.2. Administración de volúmenes dinámicos (RAID).
 - 7.5.2.1. RAID nivel 0, nivel 1 y nivel 4/5.
 - 7.5.2.2. Uso de la paridad para redundancia de datos.
 - 7.5.2.3. Pasos necesarios para montar un sistema RAID.

- Cuestionarios objetivos.
- Pruebas de respuesta libre.
- Tareas de administración.

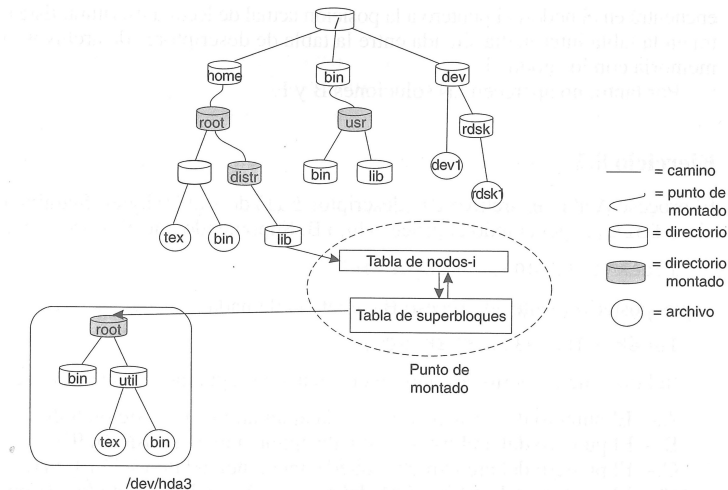
Introducción

Tareas asociadas al sistema de ficheros en GNU/Linux:

- Permitir el acceso a ficheros locales y remotos.
- Controlar los recursos que proporcionan: cuotas de disco, bloques o nodos-i libres, estadísticas de uso...
- Proteger de posibles fallos o errores: RAID, copias de seguridad...
- Controlar y proporcionar seguridad de los datos: propietarios, permisos...
- Durante el arranque: chequear (y corregir) el sistema de ficheros.
- Instalar y configurar nuevos dispositivos de almacenamiento.



Montar y desmontar sistemas de ficheros



Montar y desmontar sistemas de ficheros

- `mount [opci] <FicheroEspecialBloque> <PtoMontaje>`
 - `-t tipo-sf` ⇒ tipo de sistema de ficheros.
 - `-r` ⇒ montaje en modo sólo lectura.
 - `-w` ⇒ montaje en modo lectura/escritura.
 - `-o opcionesMontaje` ⇒ opciones del proceso de montaje (`nosuid`, `exec`, `remount`, etc.).
- `umount <PtoMontaje> (ó <FicheroEspecialBloque>)` ⇒ desmontar un sistema de ficheros. Si está siendo utilizado (**busy**), no se podrá desmontar.
- `fuser` ⇒ saber qué ficheros se están usando y qué procesos los usan (f: fichero abierto, c: directorio de trabajo, e: ejecutando un fichero, etc.)
- `lsof` ⇒ obtener un listado de todos los ficheros abiertos...



Montar y desmontar sistemas de ficheros

```
1 pedroa@pagutierrezLaptop:~$ fuser -mv / # -m: ficheros montados; -v: verbose
2          USER          PID ACCESS COMMAND
3 /:          root      kernel mount /
4          pedroa      2363 Frce. gnome-keyring-d
5          pedroa      2760 Fr.e. icedove-bin
6          pedroa      3206 Fr.e. evince
7 pedroa@pagutierrezLaptop:~$ lsof
8 COMMAND  PID  USER   FD TYPE DEVICE SIZE/OFF      NODE NAME
9 ...
10 kile     2764 pedroa  mem REG   8,4    499320 5246682 libgcrypt.so.11.6.0
11 kile     2764 pedroa  mem REG   8,4    659656 5775741 libgnutls.so.26.14.12
12 kile     2764 pedroa  mem REG   8,4    68416 5775700 libavahi-client.so.3.2.9
13 kile     2764 pedroa  mem REG   8,4    47448 5775702 libavahi-common.so.3.5.3
14 kile     2764 pedroa  mem REG   8,4    14480 5775706 libavahi-glib.so.1.0.2
15 kile     2764 pedroa  mem REG   8,4     4016 5770905 libcanberra.so.0.2.5
16 kile     2764 pedroa  mem REG   8,4    31304 5771046 libgailutil.so.18.0.1
17 ...
18 lsof     3271 pedroa  cwd DIR   8,4     4096 6301896 /home/pagutierrez
19 lsof     3271 pedroa  rtd DIR   8,4     4096      2 /
20 lsof     3271 pedroa  txt REG   8,4   131312 5767875 lsof
21 lsof     3271 pedroa  mem REG   8,4    68352 5774521 locale-archive
22 lsof     3271 pedroa  mem REG   8,4   642216 5248383 libc-2.13.so
23 lsof     3271 pedroa  mem REG   8,4   141088 5242884 ld-2.13.so
```



Montar y desmontar sistemas de ficheros

- **/etc/fstab**: fichero con información sobre todos los sistemas de ficheros a montar (o ya montados) y las zonas de intercambio a activar.

`fi_especial` `pto` `tipo` `opciones` `dump_freq` `pass_num`

- `fi_especial` ⇒ fichero especial de bloques (`/dev/...`).
- `pto` ⇒ directorio que sirve de punto de montaje (**¿permisos?**).
- `tipo` ⇒ tipo de SF (`ext2`, `ext3`, `ext4`, `vfat`, `iso9660`, `swap`, `ntfs`, `nfs`, etc.).
- Opciones para el proceso de montaje (separadas por “,” y sin espacios).
- `dump_freq` ⇒ “frecuencia del *dump*” para hacer una copia de seguridad de ese SF mediante el comando `dump` (no se usa).
- `pass_num` ⇒ en tiempo de arranque, en qué orden hay que chequear los SFs (ejecutar `fsck` para comprobar su estado).



Opciones de `fstab` I

- Opciones del fichero `/etc/fstab`:
 - `rw` \Rightarrow Lectura-escritura (por defecto).
 - `ro` \Rightarrow Sólo lectura
 - `suid/nosuid` \Rightarrow Permitido (o no) que los bits `suid` o `sgid` tengan efecto.
 - `auto/noauto` \Rightarrow Montar automáticamente (o no) (ejecutando `mount -a` se montan todos los automáticos \Rightarrow siempre se ejecuta al arrancar el sistema).
 - `exec/noexec` \Rightarrow Permitir (o no) la ejecución de ficheros.
 - `usrquota, grpquota` \Rightarrow Activar cuotas.
 - `uid=500, gid=100` \Rightarrow Propietario y grupo propietario de los ficheros del SF (si el SF no incorpora esta información o si se quiere cambiar).



Opciones de fstab II

- **umask=137** ⇒ Permisos de los ficheros (en este caso, 640) (si el SF no incorpora esta información o si se quieren cambiar).
- **dev** ⇒ Interpretar ficheros especiales en el sistema de archivos.
- **sync** ⇒ Forzar a que todas las operaciones sean síncronas (puede disminuir el tiempo de vida de la unidad de disco).
- **user** ⇒ permite que los usuarios puedan montar el sistema de ficheros. Solo el mismo usuario podrá desmontarlo. Implica las opciones noexec, nosuid y nodev.
- **users** ⇒ igual que user pero cualquiera podrá desmontarlo.
- **nouser** ⇒ Solo root puede montar el SF.
- **owner** ⇒ permite que un usuario pueda montar el sistema de ficheros, siempre que sea dueño del fichero de dispositivo. Implica las opciones nosuid y nodev.
- **defaults** ⇒ rw, suid, dev, exec, auto, nouser, async.



Montar y desmontar sistemas de ficheros

- Ejemplo de contenido del fichero `/etc/fstab`:

```
1 LABEL=/      /          ext3      defaults,usrquota 0 1
2 /dev/sda3    /windows   vfat      defaults         0 0
3 /dev/dvd     /media/dvd iso9660    noauto,owner,ro   0 0
4 /dev/fd0     /media/floppy vfat      noauto,uid=500    0 0
5 /dev/sda4    /otrolinux ext3       defaults          0 2
6 /dev/sda2    swap       swap      defaults         0 0
```

- Al ejecutar `mount` como *root*:
 - `mount /media/dvd`: coge las opciones que faltan del fichero.
 - `mount -t iso9660 -r /dev/dvd /media/dvd`: no las coge.
- Si se asigna permisos de montaje a los usuarios (opciones `user`, `users` u `owner`), sólo pueden ejecutar `mount /media/dvd` (sin opciones).
- `mount -a`: montar todas las unidades que sean auto.
- Automontado de unidades: `udev` y `dbus`.



Comprobación del sistema de ficheros

- Durante el arranque, `fsck` o `e2fsck` chequearán la consistencia o estado del sistema de ficheros, detectando problemas e intentando repararlos.
- Se actúa sobre la estructura (no sobre el contenido):
 - **Bloques** que pertenezcan a varios ficheros.
 - **Bloques** que están marcados como libres, pero que se encuentran en uso.
 - **Bloques** que se encuentran marcados como en uso, pero que están libres.
 - Inconsistencias en cuanto al número de enlaces hacia un **nodo-i**.
 - **Nodos-i** marcados como libres, pero que están en uso.
 - **Nodos-i** marcados como en uso, pero que están libres.



Comprobación del sistema de ficheros

- Para chequear un SF siempre debe estar desmontado o montado en modo de sólo lectura.
- El SF raíz debe estar montado en modo de sólo lectura (el SF raíz no se puede desmontar, ¿por qué?).
- Si al arrancar el proceso de chequeo encuentra problemas que no puede solucionar, obliga al administrador a que realice el chequeo “a mano” ejecutando la orden `fsck` o `e2fsck` (modo monousuario).



Comprobación del sistema de ficheros

- *Journaling*: para evitar la verificación completa (`fsck`) de sistemas de ficheros de gran tamaño, que sería muy costosa, se implementa un modelo de control **transaccional** basado en *logging* (un diario).
 - Las suboperaciones que modifiquen los metadatos y datos de un mismo archivo se agrupa en la misma transacción.
 - Si el sistema falla, las acciones parcialmente realizadas se deshacen o completan, recorriendo el *log*.
 - No se garantiza que el sistema esté actualizado al finalizar la recuperación, sino que es consistente.
- Sistemas con esta filosofía: JFS (IBM), `ext3` y la gran mayoría de sistemas de archivos modernos.



Comprobación del sistema de ficheros

```
1 TxBegin(&tid);
2 /* ...
3     Sub-operaciones
4     ... */
5 TxCommit(tid);
6 TxEnd(tid);
```

- Por cada sub-operación que altera las estructuras de disco se escribe un registro en el *log*, que incluye las modificaciones en los *buffers* de i-nodos y de bloques.
- Cuando se ha copiado a disco (*log*) el registro de *commit*, se empiezan a procesar realmente los *buffers*.
- Después de una caída:
 - Se completan las transacciones *committed*.
 - Se descartan el resto de transacciones.



Creación del sistema de ficheros

Añadir un nuevo disco o SF:

- 1 Realizar la conexión física.
- 2 Crear un fichero especial de dispositivo (si es necesario).
- 3 Crear las particiones: `fdisk` (o `parted`).
- 4 Crear sistema de ficheros: `mke2fs -t ext2 /dev/sdb3`
- 5 Etiquetar la partición usando `e2label` \Rightarrow asigna una etiqueta al SF que se puede usar en el fichero `/etc/fstab`, en el campo `fi_especial`, mediante `LABEL=etiqueta`.
- 6 Crear el directorio que hará de punto de montaje.
- 7 Montar el nuevo sistema de ficheros.
- 8 Actualizar `/etc/fstab` con las opciones necesarias.



Creación del sistema de ficheros

- Diferencias ext2, ext3 y ext4:
 - ext3 tiene el mismo formato que ext2 pero además es transaccional: añade un registro o *journal* que permite recuperar la consistencia tras una caída del sistema.
 - ext4 tiene un formato similar a ext3, pero además incluye:
 - Una extensión describe un conjunto de bloques lógicos contiguos de un fichero que también se encuentran contiguos en disco: muy útil para ficheros grandes.
 - Se retrasa la reserva de bloques de disco hasta que se va a escribir en él: mayor número de bloques contiguos en disco.
 - Implementa una herramienta de desfragmentación *online*, *e4defrag* (capaz de funcionar mientras se usa el SF).
 - Manejo de sistemas de ficheros y ficheros de mayor tamaño.



Creación del sistema de ficheros

- ¿Qué sistema elegir?
 - ext2 \Rightarrow muy rápido en general, pero no tiene *journaling*. Se puede usar en un SF en el que se guardarán ficheros temporales.
 - ext3 \Rightarrow buen rendimiento en general y *journaling*.
 - ext4 \Rightarrow menor uso del CPU y mayor rapidez en los procesos de lectura y escritura que ext3. Estándar de facto en Linux.



Creación del sistema de ficheros

- `tune2fs` \Rightarrow Conocer y ajustar parámetros de un SF ext4/ext3/ext2.
 - `-l dispositivo`: Listar el contenido del superbloque del SF.
 - `-c max-mount-counts dispositivo`: Establecer el nº de montajes máximo sin realizar un fsck.
 - `-i numero[d|m|w] dispositivo`: Indicar el tiempo máximo entre dos chequeos.
 - `-L etiqueta dispositivo`: Poner una etiqueta al sistema de ficheros.
 - `-m porcentaje dispositivo`: Fijar el porcentaje de bloques reservados para procesos especiales (de root). Por defecto, 5 %.



Parámetros del SF ext4 I

```
1  pedroa@pedroa-laptop ~ $ sudo tune2fs -l /dev/sda2
2  tune2fs 1.42.8 (20-Jun-2017)
3  Filesystem volume name:   ROOT
4  Last mounted on:         /
5  Filesystem UUID:          d73f541a-e887-4885-b42b-98dd433e99de
6  Filesystem magic number:  0xEF53
7  Filesystem revision #:    1 (dynamic)
8  Filesystem features:      has_journal ext_attr resize_inode dir_index filetype
                             needs_recovery extent flex_bg sparse_super large_file huge_file uninit_bg
                             dir_nlink extra_isize
9  Filesystem flags:         signed_directory_hash
10 Default mount options:    user_xattr acl
11 Filesystem state:         clean
12 Errors behavior:          Continue
13 Filesystem OS type:       Linux
14 Inode count:               3145728
15 Block count:               12582912   Reserved block count:   629142
16 Free blocks:               7726646
17 Free inodes:               2545229
18 First block:               0
19 Block size:                4096
20 Fragment size:            4096
21 Reserved GDT blocks:      1021
22 Blocks per group:          32768
```



Parámetros del SF ext4 II

```
23 | Fragments per group:      32768
24 | Inodes per group:        8192
25 | Inode blocks per group:   512
26 | RAID stride:             32710
27 | Flex block group size:    16
28 | Filesystem created:       Thu Mar 27 20:07:38 2016
29 | Last mount time:         Sun Mar 22 08:50:58 2016
30 | Last write time:         Fri Aug 29 21:26:46 2016
31 | Mount count:             391
32 | Maximum mount count:     -1
33 | Last checked:            Fri Aug 29 21:26:46 2016
34 | Check interval:         0 (<none>)
35 | Lifetime writes:         922 GB
36 | Reserved blocks uid:     0 (user root) Reserved blocks gid:      0 (group root)
37 | First inode:             11
38 | Inode size:              256
39 | Required extra isize:    28
40 | Desired extra isize:     28
41 | Journal inode:           8
42 | First orphan inode:      813409
43 | Default directory hash:  half_md4
44 | Directory Hash Seed:     c5d4f69f-cea8-4574-894b-166152ae5a40
45 | Journal backup:          inode blocks
```



Concepto de cuotas de disco

Cuotas de disco

Permiten limitar el número de bloques y/o ficheros (nodos-i) que un usuario puede usar en una partición (también se pueden establecer para grupos de usuarios).

- Hay dos tipos de límites:
 - **Límite *hard***: el usuario no puede sobrepasarlo. Si lo hace, ya no podrá usar más bloques o crear más ficheros.
 - **Límite *soft***: es inferior al límite *hard* y se puede sobrepasar durante cierto tiempo, siempre que no se alcance el límite *hard*.
- **Periodo de gracia**: tiempo durante el que se puede sobrepasar el límite *soft*. Se informa al usuario de que ha superado el límite y que debe liberar espacio o nodos-i (ficheros).
 - Los periodos y los límites se establecen, de forma independiente, para bloques y nodos-i.



Establecer cuotas de disco I

- Pasos a realizar para establecer las cuotas de disco:

- 1 Instalar el paquete quota:

```
1 apt-get install quota
```

- 2 Indicarlo en fstab (diferente en ext3 y ext4):

```
1 /dev/sdb1 /home ext4 defaults,usrjquota=aquota.user,grpjquota=  
aquota.group,jqfmt=vfsv0 0 2
```

```
1 /dev/sdb1 /home ext3 defaults,usrquota,grpquota 0 2
```



Establecer cuotas de disco II

- ③ Remontar la partición para que se activen las opciones: `mount -o remount /home`.
- ④ `quotacheck -avugm`: añade el contenido de los ficheros de control de cuotas.
 - `a`: todos los dispositivos con cuotas.
 - `v`: *verbose*.
 - `u`: cuotas para usuarios.
 - `g`: cuotas para grupos.
 - `m`: no remontar los archivos en modo solo lectura.
- ⑤ Activar las cuotas: `quotaon -avug`.
- ⑥ Desactivarlas: `quotaoff -avug`.
- ⑦ Editar la cuota del usuario pagutierrez: `edquota pagutierrez`

```
1 Cuotas de disco para user pagutierrez (uid 1008):
2   Sist. arch. bloques    blando    duro inodos blando    duro
3   /dev/sdb1  39884712 130000000 140000000 507523      0      0
```



Establecer cuotas de disco III

8 Establecer el periodo de gracia: `edquota -t`

```
1 Período de gracia antes de imponer límites blandos para users:
2 La unidad de tiempo puede ser: días, horas, minutos, o segundos
3 Sist. arch.          Periodo gracia bloque   Periodo gracia inodo
4 /dev/sdb1            7día                   7día
```

9 Copiar cuotas: `edquota -up pagutierrez jsanchez`

10 Estadísticas de las cuotas: `repquota /dev/sdb1`

```
1 *** Informe para user quotas en dispositivo /dev/sdb1
2 Periodo de gracia de bloque: 7días; periodo de gracia de inodo: 7días
3
4      Límites de bloque límites de archivo
5 Usuario      usado      blando      duro      usado blando duro
6 -----
7 root          --    10093264          0          0      199316      0      0
8 pagutierrez --    39884712 130000000 140000000    507523      0      0
9 i22fenaf  --    31940744 130000000 140000000    864578      0      0
10 jsanchezm  --    31940744 130000000 140000000    864578      0      0
```



Administración de volúmenes dinámicos

- Paridad:

- Cada vez que se escriben datos, se calcula el XOR bit a bit (1 número de unos impar, 0: número de unos par) de los bloques implicados en cada cada disco.
- Basada en operaciones XOR:

| | | |
|----------|----------|-----------|
| Disco 1: | 00101010 | (Datos) |
| Disco 2: | 10001110 | (Datos) |
| Disco 3: | 11110111 | (Datos) |
| Disco 4: | 10110101 | (Datos) |
| <hr/> | | |
| Disco 5: | 11100110 | (Paridad) |



Administración de volúmenes dinámicos

- Paridad:
 - Si uno de los discos falla (p.ej. el disco 4), el contenido se puede restaurar a partir de la paridad:

Disco 1: 00101010 (Datos)

Disco 2: 10001110 (Datos)

Disco 3: 11110111 (Datos)

Disco 5: 11100110 (Paridad)

Disco 4: 10110101 (Datos)



Administración de volúmenes dinámicos

- **Control de dispositivos de entrada/salida:**
 - La herramienta `mdadm` permite crear o administrar un dispositivo RAID, convertir un disco “normal” en un RAID...
 - Tiene distintos modos de funcionamiento:
 - `create`: configurar y activar sistemas RAID.
 - `/proc/mdstat` lista todos los sistemas RAID (dispositivos `md`) activos con información sobre su estado.
 - Las particiones que formen el RAID tienen que un *flag* RAID (`Linux raid auto`), de esta manera serán detectadas y activadas en el proceso de arranque.



Administración de volúmenes dinámicos

- Ejemplo de creación de un RAID1.
 - Instalar el paquete mdadm
 - Activar el *flag* RAID en la partición (tipo fd en fdisk, *raid* en gparted).
 - Crear el RAID (necesario instalar paquete mdadm):

```
1 mdadm --create /dev/md1 --level=1 --raid-devices=2 /dev/sda1 /dev/  
sdc1
```

- Crear un SF sobre el sistema RAID: `mke2fs -t ext4 /dev/md1`.
- Añadirlo al fichero `/etc/fstab` para montarlo en tiempo de arranque:

```
1 /dev/md1 /home ext4 defaults 1 2
```



Administración de volúmenes dinámicos

- Creación de un RAID1 con un disco que ya tiene datos:
 - Crear el RAID con la partición que tiene los datos:

```
1 mdadm --create /dev/md2 --force --level=1 --raid-devices=1 /dev/sda4
```

- Añadir nuevo disco al RAID como disco de repuesto (spare):

```
1 mdadm /dev/md2 -a /dev/sdc3
```

- Activar el nuevo disco: `mdadm --grow /dev/md2 -n 2.`
- A continuación, introducirlo en `/etc/fstab`.
- Información sobre el estado:

```
1 mdadm --detail --scan /dev/md1
```

- Todo esto se puede configurar utilizando el fichero `/etc/mdadm.conf`.



Referencias



Evi Nemeth, Garth Snyder, Trent R. Hein y Ben Whaley
Unix and Linux system administration handbook.
Capítulo 8. *Storage and disk*.
Prentice Hall. Cuarta edición. 2010.



Aeleen Frisch.
Essential system administration.
Capítulo 10. *Filesystems and disks*.
O'Reilly and Associates. Tercera edición. 2002.



Programación y Administración de Sistemas

7. Administración de sistemas de ficheros y discos

Pedro Antonio Gutiérrez

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

16 de abril de 2018

