

# Estudio de un Sistema Operativo

## Diseño de Sistemas Operativos

### Compilación del núcleo

J. C. Pérez, S. Sáez, V. Lorente y L. Pascual

Sergio Sáez [ssaez@disca.upv.es](mailto:ssaez@disca.upv.es)  
Vicente Lorente [vlorente@disca.upv.es](mailto:vlorente@disca.upv.es)  
Juan Carlos Pérez [jcperez@disca.upv.es](mailto:jcperez@disca.upv.es)

## Introducción

Los sistemas del laboratorio están configurados con varias instalaciones separadas de Linux y Windows en diferentes particiones del disco. Cada instalación está prevista para una asignatura.

Al arrancar cada máquina, toma el control el cargador GRUB (GNU GRUB, GRand Unified Bootloader) que reside en el MBR ("Master Boot Record") del disco de arranque y muestra varias opciones, correspondientes a las asignaturas que se dan en el laboratorio.

En el caso de nuestra asignatura, la etiqueta es **Linux Asignaturas GII-DSO**. El GRUB del MBR cederá entonces el control a otro GRUB que esta vez se halla en el sector de arranque de la partición dedicada a la asignatura (**/dev/sda7**). Este segundo GRUB puede dar una o más opciones entre las que estará **Ubuntu, con Linux 2.6.32.72-generic-pae** que es la que deberemos utilizar para arrancar una sesión normal.

La entrada al sistema requiere un nombre de usuario (*login*) y una palabra de paso (*password*).

Cada usuario entra al sistema con un intérprete de órdenes (*shell*) determinado y en un directorio concreto (*home*). En vuestro caso, debéis entrar con vuestro login y password de upvnet.

## El arranque de Linux

Cuando el procesador se pone en marcha o recibe un "reset hardware", empieza a ejecutar código de la ROM, que en los PC's, se conoce tradicionalmente como BIOS. El programa grabado en ella carga en RAM el sector de arranque del disquete, disco duro, CD, etc., según la configuración, y salta a ejecutarlo.

Cuando sólo tenemos instalado DOS o Windows, no se percibe la existencia de un "boot manager", ya que éste lo único que hace es arrancar el sistema que se halle en la partición "activa" del disco, sin ofrecer opciones o interfaz alguno al usuario.

En resumen, el encargado de ejecutar un núcleo de sistema operativo (puede haber varios, cada uno en una partición de un disco) es el código de arranque que ejecuta la BIOS al ponerse en marcha el sistema. Este código está en el MBR (Master Boot Record) del disco de arranque.

## Compilación del núcleo

Vamos a seguir unos pasos básicos que sirven para cualquier número de versión del kernel 2.6 en ubuntu.

Antes que nada, ejecutaremos la orden **sudo su -** para evitar tener que poner **sudo** delante de (casi) todos los comandos.

Para mantener el acceso al display gráfico (X-Windows) desde la cuenta de superusuario has de definir que quieres usar el display local usando la orden **export DISPLAY=:0** una vez eres superusuario. Pero antes has de tener

permiso para conectar con el display por parte del usuario original, que es el que ha iniciado la sesión, por tanto la secuencia de órdenes completa sería:

```
$ xhost +  
$ sudo su -  
# export DISPLAY=:0
```

## Instalación de los fuentes

Empezaremos comprobando si tenemos instalados los paquetes esenciales y del kernel que necesitaremos mediante la orden:

```
# apt-get install build-essential  
# apt-get install linux-source-2.6.32
```

La distribución dejará los fuentes del kernel en el fichero `linux-source-2.6.32.tar.bz2`, del directorio `/usr/src`. Comprueba que el fichero existe ( `$ cd /usr/src; ls -l` ) (comprueba también que su fecha de creación es reciente) y descomprímelo con la orden:

Asegúrate siempre de borrar todo vestigio de las prácticas de otros compañeros que por descuido puedan haber dejado en el sistema. Así mismo, procura siempre borrar todos tus ficheros antes de irte. Recuerda que si detectamos prácticas copiadas será muy difícil saber quién se ha copiado de quién.

```
# tar xjvf linux-source-2.6.32.tar.bz2
```

El fichero `/usr/src/linux-source-2.6.32.tar.bz2` **NO** se debe borrar manualmente. Si encuentras que ha sido borrado o modificado, reinstala el paquete de los fuente mediante la orden:

```
# apt-get install linux-source-2.6.32
```

Comprueba que ha ido todo bien y se ha creado el directorio con los fuentes.

Ahora borraremos, si existía, el enlace simbólico `linux` y crearemos uno nuevo al directorio de los nuevos fuentes:

```
# rm linux  
# ln -s linux-source-2.6.32 linux  
# ls -l
```

## Configurando el núcleo de Linux

Entramos en el directorio de los fuentes, `/usr/src/linux`, y empezamos a configurar el nuevo núcleo:

Hay varias formas de configurar el kernel. Usaremos un método automatizado, pero si necesitáramos configurar algo manualmente, se puede utilizar una de estas dos opciones: `make menuconfig` o `make xconfig`.

Para utilizar `make menuconfig` no hace falta tener las Xwindow corriendo. Necesitaríamos el paquete `libncurses5-dev` (con todas sus dependencias). Ejecutaríamos:

```
/usr/src# apt-get install libncurses5-dev  
/usr/src# cd /usr/src/linux  
/usr/src/linux# make menuconfig
```

Para utilizar make xconfig necesitaríamos las Xwindow en marcha y el paquete libqt3-dev (con todas sus dependencias). Haríamos:

```
/usr/src# apt-get install libqt3-mt-dev
/usr/src# cd /usr/src/linux
/usr/src/linux# make xconfig
```

Nosotros usaremos **make localmodconfig** que explora el hardware actual e intenta activar solamente las opciones de configuración necesarias.

Ppor tanto, situados en el directorio /usr/src/linux, ejecuta **make localmodconfig** Una vez configurado el núcleo (la configuración se guarda en el fichero **.config** Vamos a modificar la asignación de la variable **linux\_banner** del fichero **init/version.c** para que aparezcan los nombres del grupo y/o cualquier otro lema que decidáis (como por ejemplo **Viva Microchoft** o LINUX RULES!!! ;-), al objeto de saber que realmente el núcleo que está arrancando es el que habéis recompilado vosotros. Intentad entender la sintaxis de la línea que modifiquéis. Si no la entendéis, preguntad al profesor, **pero NUNCA hagáis nada al azar, "a ver si funciona", si no sabéis exactamente qué va a pasar.**

- Para aprender (en las prácticas siguientes), hay que:
  1. Pensar.
  2. Si no lo tenemos claro, buscar la información por nuestra cuenta (**man**).
  3. Volver a pensar.
  4. Volver a buscar la información por nuestra cuenta (**man**).
  5. Y, como último recurso, preguntar.

## Compilando los fuentes del núcleo

- Vamos pues con la compilación del núcleo:  
Esta orden borra los ficheros de compilaciones previas (usar sólo al principio una vez, o cuando sea necesario para empezar de cero, ya que aumenta mucho el tiempo de compilación).

```
/usr/src/linux# make clean
```

Esta orden compila el nucleo/

```
/usr/src/linux# make
```

Esto compila los módulos que hayamos indicado en la configuración.

```
/usr/src/linux# make modules
```

Esta orden copia los módulos a /lib/modules/\$(uname -r)/, creando la estructura de directorios necesaria.

```
/usr/src/linux# make modules_install
```

- Sólo nos queda un último paso importante, la generación de una imagen de ramdisk para el arranque.

```
update-initramfs -c -k 2.6.32.63+drm33.26
```

donde el argumento que sigue a la "k" es es la versión del núcleo que estamos compilando. Para averiguarla se puede utilizar la orden:

```
/usr/src/linux# make kernelversion
```

Esta imagen de ramdisk se cargará en RAM justo antes de arrancar nuestro núcleo y contiene los controladores que permiten reconocer el HW de nuestra máquina (por ejemplo permite arrancar desde un disco SATA). Realmente no es imprescindible generar esta imagen siempre y cuando hayamos incluido de forma estática (y no como módulos) los controladores para al menos nuestro bus, disco y sistema de ficheros de nuestro directorio raíz. Para más información : <http://en.wikipedia.org/wiki/Initramfs>

## Haciendo accesible el nuevo núcleo desde el menu de arranque

- Ahora que ya hemos terminado de compilar todo el núcleo debemos copiar la imagen compilada del núcleo al directorio /boot

```
/usr/src/linux# make install
```

y actualizar el fichero de configuración de grub editando el fichero

```
/etc/default/grub
```

y comentando la línea

```
GRUB_HIDDEN_TIMEOUT=0
```

con una almohadilla al principio. Y después ejecutando

```
/usr/src/linux# update-grub
```

## Reiniciando el sistema con el nuevo núcleo

- Rearranca con el nuevo núcleo entrando esta vez desde el segundo GRUB con **Linux 2.6.32.63+drm33.26**
- Comprueba que, al principio del arranque, ocultada rápidamente por las decenas de líneas de mensajes o directamente por el "splash" del interfaz gráfico, aparece la frase que has introducido. Necesitarás volver atrás en la consola. Usa **Mayús.-Re.Pág.** También puedes revisar los mensajes de arranque con la orden:

```
$ dmesg | less
```

## Modificando el núcleo

- Vamos a intentar, finalmente, que el mensaje que hemos conseguido imprimir pueda pasarse como parámetro al núcleo.
- Has estudiado en el tema del arranque de Linux cómo el núcleo lee los parámetros que se le pasan. En este caso queremos definir un parámetro que se use de la siguiente forma: **Linux 2.6.32.63+drm33.26 message=MensajeDeUnaSolaPalabra** y que se imprima del mismo modo que has impreso el mensaje anterior.
- Puedes tomar como ejemplo el parámetro **init=** que le indica al núcleo qué programa se debe ejecutar como proceso *init*. Usa la macro **\_\_setup** como se hace en **init/main.c#L288**:

```
273 static int __init init_setup(char *str)
274 {
...
286         return 1;
287 }
288 __setup("init=", init_setup);
```

- Crear una función **message\_setup** que imprima el parámetro **str** utilizando la función **printk()** e instalarla como un manejador de opciones con la macro **\_\_setup("message=", message\_setup)**.
- Recompila el núcleo utilizando sólo las órdenes:

```
/usr/src/linux# make
/usr/src/linux# make install
/usr/src/linux# update-grub
```

- Tras recompilar, arranca de nuevo introduciendo **Linux 2.6.32.63+drm33.26 message=LoQueSea** en el prompt del segundo GRUB (el de la partición de la asignatura). Para ello pulsa **e** de editar sobre la opción Linux 2.6.32.63+drm33.26 y te saldrá una lista de las opciones del grub para ese núcleo. Parecida a:

```
root                (sd0,7)
linux               /boot/vmlinuz-2.6.32.63+drm33.26 root=/dev/sda7 ro vga=0x31a quiet splash
initrd              /boot/initrd.img-2.6.32.63+drm33.26
quiet
savedefault
```

Situate al final de la línea que empieza por "linux" y añade tu mensaje (**message=MiMensaje**). Una vez finalices, pulsa **Ctrl-x** para arrancar con ese kernel y la nueva opción.

- Si algo falla, recuerda que tienes la opción de arrancar con el Linux original en lugar del nuevo.