

Práctica 4

Uso de un driver para una video-consola en Linux

4. Uso de un driver para una video-consola en Linux	1
Introducción y Objetivos	1
Bibliografía	1
Ejercicios	2
Ejercicio 1	2
Ejercicio 2	2
Ejercicio 3	2
Desarrollo de la práctica - Control de los leds del mando y su vibración	3
Anexos	4

Introducción y Objetivos

El sistema operativo se debe comunicar con múltiples componentes hardware y periféricos. Para ello se deben emplear una gran variedad de protocolos de comunicación diferentes, que permiten la interacción del SO con estos componentes (teclado, ratón, disco, puertos serie, puertos paralelos, tarjetas de red,...). En todos los sistemas operativos modernos esta interacción la lleva a cabo una parte del sistema operativo llamada driver o controlador software (se debe distinguir del controlador hardware). El driver encapsula las particularidades de un dispositivo hardware específico y expone un conjunto de operaciones ejecutadas sobre una interfaz estándar.

En esta práctica nos familiarizaremos con los drivers y cómo se organizan y se implementan bajo Linux. Al final del guión, pondremos en práctica el conocimiento adquirido haciendo uso de un driver que controle los leds de un mando de la video-consola XBox-360 y su vibración.

Bibliografía

- Jonathan Corbet; *Linux Device Drivers*. O'Reilly. 3rd Edition. Febrero 2005
- Robert Love; *Linux Kernel Development*. Addison Wesley, 3rd Edition. Julio 2010
- Kernel Documentation: <http://www.kernel.org/doc/Documentation/>

Ejercicios

Ejercicio 1

Compilar el módulo de ejemplo “hello.c”. Insertar el módulo en el kernel con el comando `sudo insmod hello.ko`. Para verificar que el módulo se insertó correctamente, ejecutar el comando `lsmod` y chequear el “log” del sistema con `dmesg` o `dmesg | tail`. En este fichero de “log” se puede encontrar el mensaje que el módulo imprimió con `printk()` en su función de inicialización. Finalmente, descargar el módulo usando `sudo rmmod hello`.

Ejercicio 2

Compilar el módulo `chardev.c`, que implementa un driver que gestiona dispositivos de caracteres ficticios. Después de compilarlo, insertar el módulo en el kernel con el comando `sudo insmod chardev.ko`. Para verificar que el módulo se insertó correctamente, chequear el “log” del sistema con `dmesg` o `dmesg | tail`. En este fichero de “log” se puede encontrar el *major number* asignado a este *driver* y el comando para crear un fichero de dispositivo gestionado por el *driver*:

```
$ sudo mknod /dev/chardev -m 666 c 250 0
```

Una vez que el fichero de dispositivo fue creado, podemos leer del driver del dispositivo como sigue:

```
$ cat /dev/chardev
I already told you 0 times Hello world!
$ cat /dev/chardev
I already told you 1 times Hello world!
$ cat /dev/chardev
I already told you 2 times Hello world!
```

Si intentamos escribir en el dispositivo obtendremos un mensaje de error en el terminal o en el fichero de “log”:

```
$ echo "Hello" > /dev/chardev
bash: echo: error de escritura: Operación no permitida
```

¿Por qué aparece este error?

No obstante, ¡enhorabuena!. El comportamiento del driver es el esperado.

Ejercicio 3

En este ejercicio vamos a hacer uso del driver *xpad* correspondiente al mando de la vídeo-consola Xbox-360 que por defecto se carga en el sistema. Para verificar que el módulo se cargó correctamente, se puede hacer uso del comando `lsmod` buscando su nombre: *xpad*, o bien chequear el “log” del sistema y buscar la cadena *Microsoft X-Box 360 pad*.

Este controlador crea tres dispositivos cuya funcionalidad se describe a continuación:

- El dispositivo `/dev/input/jsX` refleja los eventos producidos por la pulsación de algún botón o palanca en el mando. Para poder conocer que pulsación se puede hacer uso de la utilidad `jstest` descrita en el anexo *jstest* cuya salida por pantalla se detalla a continuación:

```
$ ./jstest /dev/input/js0
Joystick (Microsoft X-Box 360 pad) has 8 axes and 11 buttons. Driver version is 2.1.0.
Testing ... (interrupt to exit)
Axes:  0: 1832  1: -3367  2: -32767  3:  5590  4:    91  5: -32767  6:    0  7:    0
Buttons: 0:off 1:off 2:off 3:off 4:off 5:off 6:off 7:off 8:off 9:off 10:off
```

- El dispositivo `/sys/class/leds/xpadX/brightness` corresponde a la activación de los leds del mando de la vídeo-consola XBox-360 conocidos como el botón **Guide** en la figura 4.1. El encendido y apagado de los leds se lleva a cabo escribiendo en el dispositivo un valor:

```
$ echo 9 > /sys/class/leds/xpad0/brightness
```

El valor escrito en el dispositivo corresponde a diferentes configuraciones que activarán/desactivarán los leds del botón **Guide** de acuerdo a la siguiente descripción:

- 0: apagado
 - 1: todos leds parpadean, después vuelve a la configuración previa
 - 2: parpadeo el led superior izquierdo, luego se queda encendido
 - 3: parpadeo el led superior derecho, luego se queda encendido
 - 4: parpadeo el led inferior izquierdo, luego se queda encendido
 - 5: parpadeo el led inferior derecho, luego se queda encendido
 - 6: led superior izquierdo encendido
 - 7: led superior derecho encendido
 - 8: led inferior izquierdo encendido
 - 9: led superior derecho encendido
 - 10: leds en rotación
 - 11: led parpadean según la configuración anterior
 - 12: led parpadean lentamente según la configuración anterior
 - 13: dos leds en rotación, después vuelve a la configuración previa
 - 14: todos los leds parpadean
 - 15: todos los leds parpadean, después vuelve a la configuración previa
- El dispositivo `/dev/input/eventX` corresponde a las diferentes configuraciones de la vibración del mando de la vídeo-consola XBox-360. Para comprobar el funcionamiento se ha creado la aplicación *fftest* que recrea una vibración progresiva en intensidad de forma sinusoidal, y la vibración del motor de la izquierda (denominada vibración *strong*) y la vibración del motor de la derecha (denominada vibración *weak*). Una explicación más detallada de dicha aplicación está disponible en el anexo *fftest*.

```
$ ./fftest /dev/input/event23
```

Desarrollo de la práctica - Control de los leds del mando y su vibración

Escribir la aplicación **xbox_test** que controle los leds del mando de la vídeo-consola y su vibración de la siguiente manera:

- Se inicializará el mando con led superior izquierdo encendido y sin ninguna vibración.

- Si se pulsa el botón “start”, el mando volverá a su configuración inicial.
- Si se pulsa el botón “A” el activará la vibración de la izquierda (vibración *strong*)
- Si se pulsa el botón “Y” el activará la vibración de la derecha (vibración *weak*)
- Si se pulsa el botón “B” el led girará hacia la derecha.
- Si se pulsa el botón “X” el led girará hacia la izquierda.
- Si se pulsa el botón “left bumper” se decrementará la vibración.
- Si se pulsa el botón “right bumper” se incrementará la vibración.

Es importante reseñar que el estado actual del led se puede consultar mediante la lectura del dispositivo `/sys/class/leds/xpadX/brightness`. Para poder realizar las operaciones de lectura y escritura sobre dicho dispositivo como un usuario sin privilegios, se recomienda dar permisos de lectura y escritura a dicho fichero con el comando `chmod`.

Anexos

Controles del mando X-Box 360

El controlador y nombres de botones se detallan a continuación en la figura 4.1. Los botones que se emplearán en esta práctica son:

- A: botón verde
- B: botón rojo
- X: botón azul
- Y: botón naranja
- BR: bumper derecho
- BL: bumper izquierdo
- start: botón “Start”

Utilidad *jstest*

La utilidad *jstest* permite interpretar los eventos producidos por la pulsación de los controles de los mandos. Cada vez que se realiza una pulsación se crea un evento en el dispositivo `/dev/input/jsX` donde se almacena la información en formato binario. La utilidad *jstest* permite conocer el botón que ha modificado su estado. A continuación se muestra la salida de dicha aplicación:

```
$ ./jstest /dev/input/js0
Joystick (Microsoft X-Box 360 pad) has 8 axes and 11 buttons. Driver version is 2.1.0.
Testing ... (interrupt to exit)
Axes:  0:  1832  1: -3367  2: -32767  3:  5590  4:    91  5: -32767  6:    0  7:    0
Buttons: 0:off 1:off 2:off 3:off 4:off 5:off 6:off 7:off 8:off 9:off 10:off
```

El campo *Axes* hace referencia al movimiento del cursor y las palancas, mientras que el campo *Buttons* muestra el botón que ha cambiado su estado. La pulsación activa/desactiva viene descrita por *on/off*, cuyo identificador *id* hace referencia al botón pulsado, tal y como se resume en la tabla 4.1.

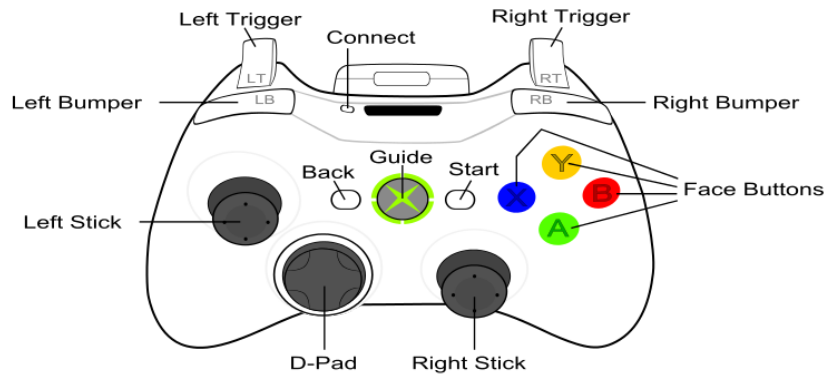


Figura 4.1: Detalle del mando X-Box 360

Cuadro 4.1: Correspondencia entre el identificador y el botón correspondiente al mando X-Box 360

id	0	1	2	3	4	5	6	7	8
botón	A	B	X	Y	left bumper	right bumper	back	start	guide

Utilidad *fftest*

La utilidad *fftest* permite configurar el mando en modo vibración. Antes de nada conviene conocer el dispositivo “evento” asociado al mando de la video-consola. Consultando el fichero `/proc/bus/input/devices` se conoce el listado de los dispositivos de entrada al sistema y los dispositivos “evento” asociado a cada uno de ellos.

```
$ cat /proc/bus/input/devices
```

```
I: Bus=0003 Vendor=045e Product=028e Version=0114
N: Name="Microsoft X-Box 360 pad"
P: Phys=usb-0000:06:00.0-1/input0
S: Sysfs=/devices/pci0000:00/0000:00:1c.6/0000:06:00.0/usb5/5-1/5-1:1.0/input/input27
U: Uniq=
H: Handlers=event23 js0
B: PROP=0
B: EV=20000b
B: KEY=7cdb000000000000 0 0 0 0
B: ABS=3003f
B: FF=107030000 0
```

A modo de ejemplo, una vez consultados los dispositivos de entrada en nuestro sistema, encontramos que el mando X-Box 360 se encuentra relacionado con el dispositivo “evento” **event23** y el dispositivo de entrada **js0** por lo que podremos interactuar con él a través de los dispositivos `/dev/input/event23` y `/dev/input/js0` respectivamente.

La aplicación *fftest* permite configurar la ganancia de la vibración 0%-100% mediante el parámetro `gain.value` cuyo rango de valores es (0, FFFF), correspondiendo 0 al 0% de ganancia y FFFF al 100%. Además, dicha aplicación crea una estructura con el nombre `effect` que especifica los parámetros de la vibración sinusoidal. Ambas configuraciones se escriben en dicho dispositivo mediante el comando *write*.

El funcionamiento de la aplicación consiste en seleccionar una vibración cuya intensidad va en aumento mediante la repetición del paso (1) y (2) que se describen a continuación: (1) especificación de la intensidad de la vibración (`gain.value`) y su escritura en el dispositivo, (2) escritura el tipo de vibración en el dispositivo, y por último (3) dejar en reposo el mando de la vídeo-consola mediante la escritura de valores nulos en el dispositivo.