Diseño de Sistemas Operativos

MÓDULO KERNEL CON FUNCIONALIDAD DE DRIVER PARA PLACA AUX DE LA RASPBERRY PI

Entrega Número 2

Rafael Ramírez Salas

Mayo de 2022

Índice

1. Archivos Adjuntados

El archivo principal es “driver_rafita.c”, que es el módulo que contiene todas las funcionalidades y funciona correctamente.

Después, además he añadido los archivos “leds_rafita.c”, “speaker_rafita.c” y “buttons_rafita.c” en los cuales se integran las funcionalidades requeridas una a una, por si hiciera falta revisar el código de una manera más clara y sencilla, ya que en cada archivo .c solo se encuentra el código correspondiente a su nombre, en contraposición al primer programa mencionado, donde está todo junto.

Para que estos programas puedan ser compilados, se añade también el Makefile.

Por último, se encuentra también el archivo “driver_init_clean.c”, el cual es una continuación del driver principal al que he añadido la funcionalidad de limpiar únicamente las partes que hayan sido inicializadas. Es decir, he creado una variable la cual voy incrementado conforme se inicializan partes del módulo, y después dependiendo de esa variable, se hace clean de todo o de una parte en específico del módulo. He probado su funcionamiento, y según mis pruebas es correcto, pero como tampoco he sabido poner en duros aprietos al programa, he preferido adjuntarlo como co-driver.

En resumen, se dispone de “driver_rafita.c” como módulo principal, pero a la hora de ponerlo en aprietos al inicializar o limpiar partes del módulo, su homólogo “driver_init_clean.c”, seguramente haga un mejor trabajo.

2. Compilación del Módulo

Para poner en funcionamiento el driver, se debe primero compilar el módulo usando el Makefile proporcionado, siendo el comando el siguiente:

```
$ MODULE=driver_rafita make
```

El driver dispone de una velocidad ya dada para el timer configurado, con el fin de evitar rebotes, por tanto, si instalamos el módulo de esta forma:

```
$ sudo insmod driver_rafita.ko
```

El driver usará la velocidad por defecto previamente configurada (155 ms).

En cambio, si queremos ser nosotros quienes configuremos dicha velocidad, se deberá usar la siguiente instrucción:

```
$ sudo insmod driver_rafita.ko velocity=1
```

En este caso, la velocidad (1 ms) sería demasiado pequeña y seguirían ocurriendo rebotes.

```
$ sudo insmod driver_rafita.ko velocity=500
```

En este otro caso, la velocidad sería un poco mayor de lo necesario y podría ocurrir que no se detectaran pulsaciones si las hacemos demasiado seguidas.

```
$ sudo insmod driver_rafita.ko velocity=200
```

Esta sería una velocidad adecuada para evitar rebotes y no perder pulsaciones.

3. Comentarios Adicionales

El código en su mayoría está comentado, sobre todo en las partes mas complejas del mismo, usando el idioma anglosajón ya que, a mi parecer, es el mejor para contextualizar los comentarios con el código programado.

Para poder trabajar los LEDs solo necesitamos mandarle datos de tipo char a /dev/leds, para ello lo más sencillo es usar la función auxiliar bin2char, la cual convierte un número binario en un dato de tipo char, de esta manera podríamos mandar un 1 si queremos encender o un 0 si queremos apagar un LED. Recordemos que los dos bits más significativos son los que nos dicen la funcionalidad y los otros seis corresponden a los LEDs. Su comando sería el siguiente:

```
$ bin2char 00111111 > /dev/leds
```

En este caso encenderíamos todos los LEDs de la placa auxiliar.

```
$ bin2char 11101010 > /dev/leds
```

Para esta operación, como no estaba definida, he optado por hacer un "printk", que se podrá ver al usar el comando "dmesg"; y se encenderán los dos leds de color verde, apagando el resto, en honor al Real Betis Balompié.

Para poder ver el registro de los botones pulsados, solo hace falta concatenar los distintos valores del buffer que guarda los datos, los datos se guardan en /dev/buttons, y con el comando "cat" tenemos acceso a los mismos. Habría que tener en cuenta que el buffer tiene una capacidad de 128 datos, longitud más que de sobra, pero no sobra recalcarlo. Además, el buffer se reinicia cada vez que pedimos que nos muestre su contenido. Su comando sería el siguiente:

```
$ cat /dev/buttons
```