



## Práctica **7** : **Ioctl**

En esta práctica vamos a aplicar la llamada al sistema Unix `ioctl` para parametrizar el manejador de entrada-salida, por ejemplo la elección del método para filtrar y elaborar los datos que pasan al buffer de alto nivel, si se muestran en pantalla los datos que se introducen en el buffer bajo nivel, etc. Así, una llamada `ioctl` toma tres parámetros, un descriptor de fichero, un código de operación y un puntero a una estructura. La estructura contiene el parámetro de la operación para operaciones tipo SET y el resultado para operaciones tipo GET.

### 1. Mensajes de petición y réplica.

El mensaje de petición que el sistema de ficheros envía a la tarea es de tipo `DEV_IOCTL`. Tiene los siguientes campos:

Campo	Significado
<code>m_type</code>	Tipo del mensaje: <code>DEV_IOCTL</code>
<code>PROC_NR</code>	Número del proceso de usuario que invocó <code>ioctl</code>
<code>REQUEST</code>	Código de operación. Es el segundo parámetro de <code>ioctl</code>
<code>ADDRESS</code>	Dirección de parámetro-resultado en el programa de usuario. Es el tercer parámetro de <code>ioctl</code>

El mensaje de réplica que la tarea envía al sistema de ficheros tiene los siguientes campos:

Campo	Significado
<code>m_type</code>	Tipo del mensaje. La tarea lo rellena con <code>TASK_REPLY</code>
<code>REP_PROC_NR</code>	La tarea lo rellena con el número del proceso de usuario que invocó <code>ioctl</code>
<code>REP_STATUS</code>	Código de retorno. Informa al sistema de ficheros de si la petición se realizó correctamente. Si es así, se rellena con <code>OK</code>

### 2. Operaciones

En esta práctica vamos diseñar nuestras operaciones y sus parámetros y vamos a ampliar el programa de usuario para que las utilice.

#### Modificación 1.

Se trata de ampliar `mouse.c` con la atención a los mensajes `DEV_IOCTL`. Por ejemplo podemos definir el código de operación 0 para establecer el tipo de filtrado que hacemos al buffer de alto nivel (uno de cada 10 muestras extraídas del buffer de bajo nivel), y el código 1 para recuperarlo. Nótese que el tamaño de la estructura (tercer parámetro de `ioctl`) depende del tipo de operación (segundo parámetro). Este tamaño es el tercer parámetro de la invocación que `mouse.c` hará a `phys_copy`.

#### Modificación 2.

Se trata de ampliar `raton.c` con la emisión de `ioctl` para parametrizar el comportamiento del manejador y descubrir cómo se ha parametrizado. Por ejemplo:

```
int modulo = 10;
if(0 > ioctl(df, 0, &modulo)) return -1;
if(0 > ioctl(df, 1, &modulo)) return -1;
```