

Objetivo

El objetivo de esta sesión es introducir al alumno una nueva herramienta del sistema operativo: los *signals*. Dicha herramienta se utiliza para generar interrupciones asíncronas en uno o varios procesos. Con el correcto conocimiento y dominio de los *signals* se pretende que los alumnos puedan modificar el comportamiento normal de sus programas.

Motivación

Más concretamente, con esta sesión, el alumno ha de ejercitar:

- Captura de señales (*Signal handler*)
- Envío de señales (*Kill* y *raise*)
- Señales periódicas (*alarm*)

Documentación previa

Para completar esta sesión se recomienda la lectura de las siguientes referencias:

SALVADOR, J. (2009). *Introducció al llenguatge de programació C*.

SALVADOR, J. (2014). *Programació en UNIX per a pràctiques de Sistemes Operatius*.

Enunciado: Lenguaje de Signals

Los informáticos de La Salle, obsesionados con optimizar todo lo que pueden, han decidido dar un paso más y optar por optimizar el lenguaje.

La idea consiste en crear un sistema donde programar palabras o frases cortas que después pueden ser escritas en un fichero de salida mediante interrupciones.

Habrán dos modos a diferenciar dentro del programa:

- **Modo de programación:** Cada vez que se dé uno de los cuatro *signals* correspondientes (*ver Tabla1*) significará que se deberá guardar en la posición correspondiente (hay **4 slots**) el texto que el usuario haya introducido.
Cuando llegue la interrupción el programa quedará a la espera de que el usuario introduzca el texto a programar y pulse *Enter* para guardarlo en el *slot*.
- **Modo de escritura:** Cada vez que se dé uno de los cuatro *signals* correspondientes (*ver Tabla1*) se escribirá en el fichero de salida la palabra o frase corta programada en el anterior modo. En el caso que no haya estado programada simplemente no se escribirá nada en fichero.

Nuestro programa será un proceso en bucle infinito, y sólo se podrá terminar mediante Ctrl+C (SIGINT). Tened en cuenta que se debe controlar este caso y **liberar o cerrar** adecuadamente los recursos utilizados.

	Modo programación	Modo escritura
SIGURG	Actúa de <i>switch</i> entre los modos. Si llega la interrupción y el modo es "Modo escritura" se cambiará al "Modo programación" y al contrario. Por defecto el modo inicial es "Modo programación".	
SIGTRAP	Se programa la palabra o frase corta en la posición 1	Se escribe en fichero la palabra o frase corta en la posición 1
SIGABRT	Se programa la palabra o frase corta en la posición 2	Se escribe en fichero la palabra o frase corta en la posición 2
SIGBUS	Se programa la palabra o frase corta en la posición 3	Se escribe en fichero la palabra o frase corta en la posición 3
SIGFPE	Se programa la palabra o frase corta en la posición 4	Se escribe en fichero la palabra o frase corta en la posición 4

Tabla 1: Descripción Signals

Consideraciones y requisitos

- Como máximo, cada *slot* puede contener un máximo de 30 caracteres. Cada *slot* puede ser reprogramado más de una vez, sobrescribiendo así el contenido antiguo.
- El nombre de fichero de salida ha de ser: **mensaje.siglang**
- Se puede considerar que todas las escrituras se realizarán sobre el **mismo fichero** durante toda la ejecución, no es necesario crear un nuevo fichero al cambiar de modo.
- **Toda finalización de procesos se ha de implementar por signals.** Es decir, no se permite hacer uso de la función `exit()`.
- **No está permitido el uso de variables globales, excepto aquellas necesarias para el control de signals.**
- Toda entrada y salida se ha de realizar con file descriptors. **No está permitido el uso de `printf`, `scanf`, `FILE*`, `getchar`, o similares.**
- Se ha de compilar utilizando los flags `-Wall` y `-Wextra`.
- **Cualquier práctica que contenga *warnings* será no apta.**
- Para bloquear el programa se recomienda el uso de `pause()`.
- La sesión debe llamarse `s1.c` y debe contener comentado en la primera línea del código los nombres, apellidos y logins de los autores.

Ejemplo de ejecución

```
lsxxxxx@montserrat:~> ./SigLang

PID - 548
Programa y escribe tus mensajes usando interrupciones :)

SIGURG-> MODO PROGRAMACIÓN seleccionado
SIGTRAP-> Escribe la palabra o frase a programar en el SLOT 1: Hola!
SLOT1 actualizado!

SIGABRT-> Escribe la palabra o frase a programar en el SLOT 2: Patitos
SLOT2 actualizado!

SIGFPE-> Escribe la palabra o frase a programar en el SLOT 4: Compus
mola
SLOT4 actualizado!

SIGFPE-> Escribe la palabra o frase a programar en el SLOT 4: SO mola
SLOT4 actualizado!

SIGURG-> MODO ESCRITURA seleccionado
SIGTRAP-> SLOT1 volcado a fichero!
SIGTRAP-> SLOT1 volcado a fichero!
SIGABRT-> SLOT2 volcado a fichero!
SIGBUS-> SLOT3 volcado a fichero!
SIGFPE-> SLOT4 volcado a fichero!
^C
SIGINT-> Exit
Killed
```

```
lsxxxxx@montserrat:~> cat mensaje.siglang
Hola! Hola! Patitos SO mola
```