

# Introducción a Sistemas Operativos: Comunicación entre Procesos

*Clips xxx*  
*Francisco J Ballesteros*

## 1. Alarmas

Podemos combinar las señales con un temporizador que tiene cada proceso para implementar time-outs. El temporizador es otra abstracción que suministra UNIX para cada proceso. Podemos programarlo para que envíe la señal ALRM pasado un tiempo, e instalar el manejador que queramos para dicha señal.

Veamos un programa que hace esto:

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <err.h>
#include <signal.h>

static void
tout(int no)
{
    fprintf(stderr, "interrupted\n");
}

int
main(int argc, char* argv[])
{
    char buf[1024];
    int nr;

    signal(SIGALRM, tout);
    fprintf(stderr, "> ");
    siginterrupt(SIGALRM, 1);    // try with 0
    alarm(15);
    nr = read(0, buf, sizeof buf - 1);
    alarm(0);
    if(nr > 0){
        write(1, buf, nr);
    } else {
        warn("read");
    }
    exit(0);
}
```

Si lo ejecutamos y escribimos una línea antes de que pasen 15 segundos esto es lo que sucede:

```
unix$ 8.alrm
> hola
hola
unix$
```

El programa termina la llamada a `read` y escribe lo que ha leído. Pero si lo ejecutamos y pasan 15 segundos...

```
unix$ 8.alrm
> interrupted
8.alrm: read: Interrupted system call
unix$
```

Primero se interrumpe `read`, y ejecuta el manejador `tout` que hemos instalado (que imprime "interrupted"). Luego `read` termina indicando como error `EINTR`, y el programa continúa.

La primera llamada a `alarm` instala el temporizador para que envíe la señal `ALRM` pasados 15 segundos. La segunda llamada a `alarm` (tras `read`) cancela el temporizador.

Observa la llamada a `siginterrupt`, ¿Qué sucede si la quitamos?

Una última advertencia. Los temporizadores han de usarse con sumo cuidado. De hecho, si es posible, es mejor no utilizarlos. Hacen los programas impredecibles y difíciles de depurar. Por ejemplo, ¿Qué pasa si el usuario tardó 16 segundos en escribir su línea? ¿Por qué falla el programa en esos casos? Tal vez sería mejor dejar el programa bloqueado leyendo en nuestro caso...