



Tipos

Tipos principales de signals

▼ Tipos y resumen de función

SIGCHLD

Este signal lo recibe el padre cuando un proceso hijo ha terminado o ha sido parado.

SIGCONT

Este signal fuerza al proceso a continuar si estaba parado.

SIGSTOP

Este signal para el proceso (Puede ser reanudado con SIGCONT).

SIGINT

Este signal interrumpe la ejecución (Al igual que Ctrl+C).

SIGALRM

Este signal se envía cuando el contador empezado por la llamada `"alarm"` ha terminado.

SIGKILL

Definiciones básicas

Generalmente, cada evento tiene asociado un signal. Es decir, que cuando se recibe un **Signal**, se ejecuta su **Evento** asociado.

No obstante, hay dos signals de uso Usuario, que no tienen ningún evento predefinido: **SIGUSR1** y **SIGUSR2**.

Cada proceso tiene un tratamiento asociado a cada signal. Estos pueden ser modificados excepto el caso de **SIGKILL** y **SIGSTOP**.

Este signal termina el proceso.

SIGSEGV

Este signal se envia cuando se ha accedido a una posicion invalida de memoria (Segmentation fault).

SIGUSR1 & SIGUSR2

Estos signals los define el usuario.

Los signals principalmente funcionan como interrupciones de software, es decir, cuando se recibe un signal, se interrumpe el codigo y se procesa el tratamiento de ese signal antes de seguir con la ejecución (Si no ha muerto el proceso durante el tratamiento).

▼ Llamadas a sistema relacionadas con signals

kill

La llamada a sistema kill **no mata el proceso**, sino que sirve para mandar un signal en concreto:

```
kill(PID, SIGUSR1);
```

Este codigo no matará el proceso con pid PID, sinó que le mandará el signal **SIGUSR1**.

sigaction

La llamada sigaction sirve para reprogramar un signal en concreto:

```
sigaction(int signum, struct sigaction *trat, struct sigaction *trat_ant)
```

Donde `signum` es el número de la signal a modificar, `trat` es el tratamiento que queremos que tome, y el ultimo parametro `trat_ant` nos permite guardar el tratamiento antiguo (**NULL** si no lo deseamos)

Mas informacion: Struct sigaction.

sigprocmask

La llamada sigprocmask nos permite bloquear y desbloquear signals mediante el uso de una máscara:

```
sigprocmask(SIG_BLOCK, SIGUSR1, NULL);
```

Este ejemplo, bloquea la acción SIGUSR1.

Mas información: Manipulación masks.

sigsuspend

La llamada sigsuspend espera hasta que llega un evento cualquiera, que no esté bloqueado.

```
sigsuspend(&mask);
```

Donde `mask` es la máscara que nos permitirá saber qué puede despertarle y qué no.

Mas información: Manipulación masks.

alarm

La llamada alarm programa el envío automatico de un signal SIGARLM:

```
alarm(s);
```

Donde `s` es el numero de segundos que queremos que tarde en enviar el signal SIGARLM.