

UNIVERSIDAD AUTÓNOMA METROPOLITANA

UNIDAD IZTAPALAPA



Casa abierta al tiempo

Laboratorio: Señales

Profesor

José Luis Quiroz Fabián

Alumno

Martínez Buenrostro Jorge Rafael

Sistemas Operativos

Investigue por qué el proceso no se detuvo si no cambiamos el comportamiento de CTRL-C (SIGINT)

Cuando se presiona CTRL-C durante la ejecución de un programa, se envía una señal SIGINT al proceso. Esta es una señal de interrupción que el sistema operativo envía al proceso para indicar que debe terminar.

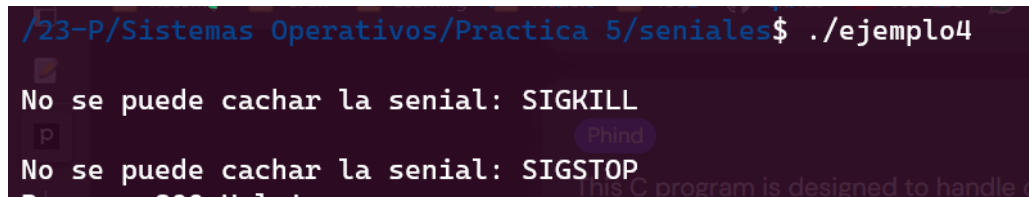
```
if (signal(SIGKILL, manejador) == SIG_ERR)
    printf("\nNo se puede cachar la senial: SIGKILL\n");
if (signal(SIGSTOP, manejador) == SIG_ERR)
    printf("\nNo se puede cachar la senial: SIGSTOP\n");
if (signal(SIGTSTP, manejador) == SIG_ERR)
    printf("\nNo se puede cachar la senial: SIGTSTP\n");
```

Dentro del código del ejercicio podemos ver estas condicionales dentro del **main**. Las cuales hacen uso de la función **signal** para poder cambiar la acción tomada por el proceso al recibir la señal especificada.

La función **signal** tiene dos argumentos: el número de señal y un puntero a una función que se ejecutará cuando se reciba la señal especificada en el primer argumento. Una vez establecido esto podemos ver que dentro del ejercicio se van a manejar tres señales: **SIGKILL**, **SIGSTOP** y **SIGTSTP**.

Cuando ejecutamos el programa podemos como cada segundo se imprime un mensaje en pantalla. Abriendo otra terminal podemos poner a prueba la implementación de la función **signal**. Si ejecutamos el comando **kill -SIGSTOP PID** podemos ver como la primera ejecución se detiene, y al ejecutar **kill -SIGCONT PID** el proceso continúa su ejecución. Después de esto al presionar CTRL-C podemos ver como no termina su ejecución.

El primer detalle que podemos ver al ejecutar el programa que nos puede dar una pista para la explicación se ve en la siguiente imagen.



```
/23-P/Sistemas Operativos/Practica 5/seniales$ ./ejemplo4
No se puede cachar la senial: SIGKILL
No se puede cachar la senial: SIGSTOP
```

En las primeras dos líneas después de la ejecución vemos los mensajes de error correspondientes a las señales **SIGKILL** y **SIGSTOP**. Estos mensajes se deben a que estas señales no pueden ser capturadas, bloqueadas ni ignoradas según los estándares POSIX.

Ahora bien cuando se presiona CTRL-C en la terminal se manda una señal SIGINT al proceso que se encuentra actualmente en primer plano. Para poder ver que está pasando se agrega un manejador para la señal **SIGINT**, de esta forma podremos que está pasando cuando se detiene y se reanuda la ejecución del programa.

```
if(signal(SIGINT, manejador) == SIG_ERR)
    printf("\nNo se puede cachar la senial: CTRL-C (SIGINT)\n");
```

```

    break;
case SIGINT:
    printf("Evento: CTRL-C\n");
    break;
}

```

Con esto podremos ver que ocurre cuando se presiona CTRL-C. Compilamos y ejecutamos; al presionar CTRL-C podemos ver que el manejador sobrescribe su comportamiento y ahora solo imprime en pantalla el mensaje **Evento: CTRL-C**

```

/23-P/Sistemas Operativos/Practica 5/señales$ ./ejemplo4
No se puede cachar la senial: SIGKILL
No se puede cachar la senial: SIGSTOP
Proceso 1198 Hola!
Proceso 1198 Hola!
^CEvento: CTRL-C
Proceso 1198 Hola!
Proceso 1198 Hola!
Proceso 1198 Hola!

```

Después lo detenemos y reanudamos su ejecución por medio de las señales **SIGSTOP** y **SIGCONT**, una vez reanudada su ejecución presionamos nuevamente CTRL-C y vemos que no se imprime en pantalla el mensaje del manejador.

```

/23-P/Sistemas Operativos/Practica 5/señales$ Proceso 1198 Hola!
Proceso 1198 Hola!
Proceso 1198 Hola!
Proceso 1198 Hola!
^C
ic3manlinux@DESKTOP-RLSF6C3:/mnt/c/Users/molap/OneDrive/Documentos/UAM
/23-P/Sistemas Operativos/Practica 5/señales$ Proceso 1198 Hola!
Proceso 1198 Hola!

```

Este comportamiento nos permite ver que puede estar pasando, ya que antes de detener y reanudar el proceso el programa es capaz de escuchar la señal creada al presionar CTRL-C. La señal **SIGCONT** nos permite reanudar la ejecución de un proceso en el punto en el que se detuvo sin embargo por el comportamiento visto podemos decir que su ejecución paso del primer plano al segundo, y que al presionar CTRL-C la señal **SIGINT** es enviada a la terminal y no a nuestro proceso. Por lo tanto, necesitamos buscar una forma de traer de vuelta al primer plano a nuestro proceso, esto lo podemos lograr escribiendo en la terminal **fg** sin argumentos devuelve al primer plano el proceso más reciente que se ha detenido o que se ha enviado al fondo o segundo plano.

Para probar esta solución no moveremos el código, solo lo ejecutaremos, pausaremos y reanudaremos. Después de ser reanudado presionaremos CTRL-C para verificar que siga sin ser escuchado, posteriormente escribiremos en la terminal **fg** y volveremos a presionar CTRL-C.

Aquí podemos ver que después de ser reanudada su ejecución al presionar CTRL-C no pasa nada.

```

/23-P/Sistemas Operativos/Practica 5/seniales$ gcc ejemplo4.c -o ejemp
lo4Evento: SIGCONT

No se puede cachar la senial: SIGKILL

No se puede cachar la senial: SIGSTOP
Proceso 1255 Hola!
Proceso 1255 Hola!
Proceso 1255 Hola!
Proceso 1255 Hola!
Proceso 1255 Hola!
Proceso 1255 Hola!
^C
ic3manlinux@DESKTOP-RLSF6C3: /mnt/c/Users/molap/OneDrive/Documentos/UAM
/23-P/Sistemas Operativos/Practica 5/seniales$ Proceso 1255 Hola!

```

Al escribir **fg** en la terminal podemos ver como se muestra el nombre de nuestro proceso, después de unas iteraciones presionamos CTRL-C y vemos que la señal es escuchada por nuestro proceso.

```

Proceso 1255 Hola!
fg
./ejemplo4
Proceso 1255 Hola!
Proceso 1255 Hola!
Proceso 1255 Hola!
Proceso 1255 Hola!
^CEvento: CTRL-C
Proceso 1255 Hola!
Proceso 1255 Hola!

```

Con esto podemos ver que la razón por la que con anterioridad al presionar CTRL-C no se detenía el proceso era por que éste se encontraba en segundo plano y era capaz de recibir la señal. Si ahora queremos que el proceso termine su ejecución al presionar CTRL-C después haber sido reanudado simplemente quitamos de la función manejador el caso para la señal **SIGINT**; y después de esto solo escribimos **fg** en la terminal y el proceso terminará su ejecución cuando presionemos CTRL-C. Como podemos ver en la siguiente imagen

```

Proceso 1263 Hola!
^C
ic3manlinux@DESKTOP-RLSF6C3: /mnt/c/Users/molap/OneDrive/Documentos/UAM
/23-P/Sistemas Operativos/Practica 5/seniales$ Proceso 1263 Hola!
Proceso 1263 Hola!
Proceso 1263 Hola!
fgProceso 1263 Hola!

./ejemplo4
Proceso 1263 Hola!
^C
ic3manlinux@DESKTOP-RLSF6C3: /mnt/c/Users/molap/OneDrive/Documentos/UAM
/23-P/Sistemas Operativos/Practica 5/seniales$ clear

```