



Cadenas Matrices Punteros Gestión de memoria Manejo de archivos Programas en C Preguntas de entrevista en C Preguntas de opción múltiple en C

Señales en lenguaje C

Leer

Conversar

Cursos

Práctica

Video

Prerequisite : [Fork system call](#), [Wait system call](#)

A signal is a software generated interrupt that is sent to a process by the OS because of when user press ctrl-c or another process tell something to this process.

There are fix set of signals that can be sent to a process. signal are identified by integers.

Signal number have symbolic names. For example **SIGCHLD** is number of the signal sent to the parent process when child terminates.

Examples:

```
#define SIGHUP 1 /* Colgar el proceso */ #define SIGINT 2 /*
Interrumpir el proceso */ #define SIGQUIT 3 /* Salir del proceso */
#define SIGILL 4 /* Instrucción ilegal. */ #define SIGTRAP 5 /*
Trampa de seguimiento. */ #define SIGABRT 6 /* Abortar. */
```

OS Structures for Signals

- For each process, the operating system maintains 2 integers with the bits corresponding to a signal numbers.
- The two integers keep track of: **pending signals and blocked signals**
- With 32 bit integers, up to 32 different signals can be represented.

Example :

In the example below, the SIGINT (= 2) signal is blocked and no signals are pending.

Pending Signals

31	30	29	28	...	3	2	1	0
0	0	0	0	...	0	0	0	0

Blocked Signals

31	30	29	28	...	3	2	1	0
0	0	0	0	...	0	1	0	0

A signal is sent to a process setting the corresponding bit in the pending signals integer for the process. Each time the OS selects a process to be run on a processor, the pending and blocked integers are checked. If no signals are pending, the process is restarted normally and continues executing at its next instruction.

If 1 or more signals are pending, but each one is blocked, the process is also restarted normally but with the signals still marked as pending. If 1 or more signals are pending and NOT blocked, the OS executes the routines in the process's code to handle the signals.

Default Signal Handlers

There are several default signal handler routines. Each signal is associated with one of these default handler routine. The different default handler routines typically have one of the following actions:

- Ign: Ignore the signal; i.e., do nothing, just return
- Term: terminate the process
- Cont: unblock a stopped process
- Stop: block the process

// Programa CPP para ilustrar



```
// Manejador de señales predeterminado
#include<stdio.h>
#include<señal.h>

int principal()
{
    señal(SIGINT, manejador_sigint);
    mientras (1)
    {
        printf("hola mundo\n");
        dormir(1);
    }
    devuelve 0;
}
```

Output: Print hello world infinite times. If user presses ctrl-c to terminate the process because of **SIGINT** signal sent and its default handler to terminate the process.

```
Hola Mundo
Hola Mundo
Hola Mundo
finalizado
```

User Defined Signal Handlers

A process can replace the default signal handler for almost all signals (but not SIGKILL) by its user's own handler function.

A signal handler function can have any name, but must have return type void and have one int parameter.

Example: you might choose the name sigchld_handler for a signal handler for the **SIGCHLD** signal (termination of a child process). Then the declaration would be:

```
void sigchld_handler(int sig);
```

When a signal handler executes, the parameter passed to it is the number of the signal. A programmer can use the same signal handler function to handle several signals. In this case the handler would need to check the parameter to see which signal was sent. On the other hand, if a signal handler function only handles one signal, it isn't necessary to bother examining the parameter since it will always be that signal number.

```
// Programa CPP para ilustrar
// Manejador de señales definido por el usuario
```



```
#include<stdio.h>
#include<señal.h>

// Controlador para SIGINT, causado por
// Ctrl-C en el teclado void
handle_sigint(int sig) { printf("Señal
capturada %d\n", sig); }

int main()

{ señal(SIGINT, handle_sigint); mientras (1) ;
devolver 0; }
```

Output:

```
^CCaught señal 2 // cuando el usuario presiona ctrl-c
^Señal captada 2
```

Sending signals via kill()

We can send a signal using kill() to the process.

int kill(pid_t pid, int signal); pid: id del
proceso de destino signal: el tipo de
señal a enviar
Valor de retorno: 0 si la señal se envió correctamente

Example:

```
pid_t iPid = getpid(); /* El proceso obtiene su id.*/ kill(iPid,
SIGINT); /* El proceso se envía a sí mismo una señal SIGINT
```



(¿se suicida?) (debido a SIGINT)

El controlador de señal predeterminado es finalizar el proceso) */

Questions

1. What is the Output of the following program?

```
#include<stdio.h>
#include<wait.h>
#include<signal.h> int main()
{ int stat; pid_t

pid; if ((pid = fork())
== 0) while(1) ; else
{ kill(pid, SIGINT); wait(&stat); if
(WIFSIGNALED(stat))

psignal(WTERMSIG(stat), "Término hijo
debido a"); } }
```

Output:

Plazo de finalización del niño debido a: Interrupción

2. What is the Output of the following program?

```
#include<stdio.h>
#include<señal.h>
#include<wait.h> int val =
10; void handler(int
sig) { val += 5; } int main() { pid_t

pid; señal(SIGCHLD,

handler); si ((pid

= fork()) == 0) { val -=
3; salir(0); } waitpid(pid, NULL, 0);
```



```
printf("val = %d\n", val); salir(0); }
```

Output:

valor = 15

3. Consider the following code. What is the output?

```
#include<stdio.h>
#include<wait.h>
#include<signal.h> pid_t pid;
int contador =
0; void handler1(int sig)
{ contador++; printf("contador =

%d\n", contador); /*
Vuelca la cadena impresa a la salida estándar */ fflush(stdout);
kill(pid, SIGUSR1); } void handler2(int sig) { contador += 3; printf("contador =

%d\n", contador); salir(0); }

int main() { pid_t

p; int estado;
señal(SIGUSR1,
manejador1); si ((pid = fork()) == 0)
{ señal(SIGUSR1, manejador2); matar(getppid(),

SIGUSR1); mientras(1) ; } si ((p = esperar(&estado)) >
0) { contador += 4; printf("contador = %d\n", contador); } }
```

Output

contador = 1

//(controlador del padre)



```
contador = 3          //(controlador del niño)
contador = 5          //(principal del padre)
```

This article is contributed by **Kadam Patel**. If you like GeeksforGeeks and would like to contribute, you can also write an article using contribute.geeksforgeeks.org or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

Última actualización: 08 de febrero de 2018

25

Lecturas similares

1. Señales de error del programa
2. Señales de control en el microprocesador 8155
3. Comunicación entre dos procesos mediante señales en C
4. Diferencia entre el lenguaje C y el lenguaje LISP
5. Rompecabezas del lenguaje de programación AC
6. Convertir código C/C++ a lenguaje ensamblador
7. Diferencia entre los especificadores de formato %d y %i en lenguaje C
8. lvalue y rvalue en lenguaje C
9. Diferencia entre while(1) y while(0) en lenguaje C
10. kbhit en lenguaje C

Previous

Comunicación entre dos procesos
usando señales en C

Next

Llamada al sistema de espera en C



Article Contributed By :



Geeks para Geeks

Vote for difficulty

Current difficulty : [Medium](#)

[Fácil](#)

[Normal](#)

[Medio](#)

[Duro](#)

[Experto](#)

Etiquetas del artículo: [Biblioteca C](#), [programación de sistemas](#), [Lenguaje C](#)

[Mejorar el artículo](#)[Informar de un problema](#)

GeeksforGeeks

A-143, Piso 9, Sovereign Corporate
Torre, Sector-136, Noida, Uttar Pradesh - 201305

feedback@geeksforgeeks.org

Compañía

[Sobre nosotros](#)[Carreras](#)[En los medios](#)[Contáctenos](#)

Términos y condiciones

[política de privacidad](#)[Política de derechos de autor](#)[Avisos de derechos de autor de terceros](#)[Anúnciese con nosotros](#)

Idiomas

[Pitón](#)[Java](#)[C++](#)[GoLang](#)[SQL](#)[Lenguaje R](#)

Explorar

[Feria de empleo para estudiantes](#)[POTD: Renovado](#)[Python Backend EN VIVO](#)[Desarrollo de aplicaciones de Android](#)[DevOps EN VIVO](#)[DSA en JavaScript](#)

Estructuras de datos

[Formación](#)[Cadena](#)[Lista enlazada](#)[Pila](#)[Cola](#)[Árbol](#)

Tutorial de Android

Gráfico

Algoritmos

Clasificación

Búsqueda

Avaro

Programación dinámica

Búsqueda de patrones

Recursión

Retroceso

Ciencias de la Computación

Notas de GATE CS

Sistemas operativos

Red de computadoras

Sistema de gestión de bases de datos

Ingeniería de software

Diseño de lógica digital

Matemáticas de ingeniería

Ciencia de datos y aprendizaje automático

Ciencia de datos con Python

Ciencia de datos para principiantes

Tutorial de aprendizaje automático

Matemáticas para el aprendizaje automático

Tutorial de Pandas

Tutorial de NumPy

Tutorial de PNL

Tutorial de aprendizaje profundo

Programación competitiva

Los mejores DSA para CP

Los 50 principales problemas de los árboles

Los 50 problemas gráficos más comunes

Los 50 principales problemas de matrices

Los 50 principales problemas de cuerdas

Los 50 principales problemas de DP

Los 15 mejores sitios web para CP

Desarrollo web

HTML

CSS

JavaScript

Oreja

ReactJS

AngularJS

NodeJS

Python

Ejemplos de programación en Python

Tutorial de Django

Proyectos de Python

Python Tkinter

Tutorial de Python para OpenCV

Pregunta de entrevista de Python

DevOps

Git

AWS

Estibador

Kubernetes

Azur

GCP

Diseño de sistemas

¿Qué es el diseño de sistemas?

SD monolítica y distribuida

Escalabilidad en SD

Bases de datos en SD

Diseño de alto nivel o HLD

Diseño de bajo nivel o LLD

Las principales preguntas de la entrevista de SD



Rincón de entrevistas

Preparación de la empresa

Preparación para SDE

Rincón de entrevistas de empresa

Entrevista con experiencia

Entrevista de pasantía

Programación competitiva

Aptitud

Comercio

Contabilidad

Estudios Empresariales

Microeconomía

Macroeconomía

Estadística para la economía

Desarrollo económico de la India

SSC/ BANCA

Programa de estudios CGL del SSC

Programa de estudios del SBI PO

Programa de estudios del secretario del SBI

Programa de estudios de IBPS PO

Programa de estudios del secretario de IBPS

Preguntas de aptitud

Documentos de práctica de CGL de SSC

Escuela GfG

Apuntes del CBSE para octavo grado

Apuntes del CBSE para 9.º grado

Apuntes del CBSE para 10.º grado

Apuntes del CBSE para el 11.º grado

Apuntes del CBSE para 12.º grado

Gramática inglesa

UPSC

Notas de política

Notas de geografía

Notas de historia

Notas de ciencia y tecnología

Notas de economía

Temas importantes en ética

Documentos de años anteriores de la UPSC

Escribe y gana

Escribe un artículo

Mejorar un artículo

Elige temas para escribir

Escribe la experiencia de la entrevista

Pasantías

Pasantía de video

@geeksforgeeks , Algunos derechos reservados