Signal Handling in C++

Difficulty Level : Medium • Last Updated : 07 Dec, 2021

Signals are the interrupts that force an OS to stop its ongoing task and attend the task for which the interrupt has been sent. These interrupts can pause service in any program of an OS. Similarly, C++ also offers various signals which it can catch and process in a program. Here is a list of various signals and their operations that C++ provides the user to work with.

Signals	Operations
SIGINT	produces a receipt for an active signal
SIGTERM	Sends a termination request to the program
SIGBUS	Bus error which indicates access to an invalid address
SIGILL	Detects an illegal command
SIGALRM	This is used by the alarm() function and indicates the expiration of the timer.
SIGABRT	Termination of a program, abnormally
SIGSTOP	The signal cannot be blocked, handled, and ignored and can stop a process

SIGSEGV	Invalid access to storage
SIGFPE	Overflow operations or mathematically incorrect operations like divide by zero
SIGUSR1 SIGSUR2	User-Defined Signals

This signal() function is provided by the signal library and is used to trap unexpected interrupts or events.

Syntax:

```
signal(registered signal, signal handler)
```

The first argument is an integer, representing the signal number and second is the pointer to a signal handling function. We must keep in mind that the signal that we would like to catch must be registered using a signal function and it must be associated with a signal handling function. The signal handling function should be of the void type.

Example:

CPP

```
// CPP Program to demonstrate the signal() function
#include <csignal>
#include <iostream>
using namespace std;

void signal_handler(int signal_num)
{
    cout << "The interrupt signal is (" << signal_num
         << "). \n";

    // It terminates the  program
    exit(signal_num);
```

```
{
    // register signal SIGABRT and signal handler
    signal(SIGABRT, signal_handler);

    while (true)
        cout << "Hello GeeksforGeeks..." << endl;
    return 0;
}
```

Output: Being in an infinite loop this code will show the following output until an interrupt is faced:

```
Hello GeeksforGeeks...
Hello GeeksforGeeks...
Hello GeeksforGeeks...
Hello GeeksforGeeks...
```

Now if we press **Ctrl+C** to send an interrupt, the program will exit by printing:

```
Hello GeeksforGeeks...
Hello GeeksforGeeks...
Hello GeeksforGeeks...
Hello GeeksforGeeks...
The interrupt signal is (22).
```

The raise() function is used to generate signals.

Syntax:

```
raise( signal )
```

It takes an argument as any of the functions mentioned in the list.

Example:

CPP

```
// CPP Program to demonstrate the raise() function
```

Start Your Coding Journey Now!

```
using namespace std;

void signal_handler(int signal_num)
{
    cout << "Interrupt signal is (" << signal_num << ").\n";

    // It terminates program
    exit(signal_num);
}

int main()
{
    int count = 0;
    signal(SIGSEGV, signal_handler);
    // register signal SIGSEGV and signal handler

    while (++count) {
        cout << "Hello GeeksforGeeks..." << endl;
        if (count == 5)
            raise(SIGSEGV);
    }
    return 0;
}
```

Output

```
Hello GeeksforGeeks...
Hello GeeksforGeeks...
Hello GeeksforGeeks...
Hello GeeksforGeeks...
Hello GeeksforGeeks...
Interrupt signal is (11).
```

This article is contributed by **Chinmoy Lenka**. If you like GeeksforGeeks and would like to contribute, you can also write an article using write.geeksforgeeks.org or mail your article to review-team@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.

Start Your Coding Journey Now!

Like

Previous Next

Related Articles

1. Comparison of Exception Handling in C++ and Java
2. Exception Handling in C++
3. Four File Handling Hacks which every C/C++ Programmer should know
4. Socket Programming in C/C++: Handling multiple clients on server without multi threading
5. Set Position with seekg() in C++ File Handling
6. tellp() in file handling with c++ with example
7. File Handling through C++ Classes
8. How to work with file handling in C++
9. Error handling during file operations in C/C++
10. Contact Book in C++ using File Handling

Start Your Coding Journey Now!

Article Contributed By :

GeeksforGeeks

Vote for difficulty

Current difficulty : Medium

Easy Normal Medium Hard Expert

Improved By : anshikajain26

Article Tags : C++

Practice Tags : CPP

Report Issue

GeeksforGeeks

A-143, 9th Floor, Sovereign Corporate Tower,
Sector-136, Noida, Uttar Pradesh - 201305

feedback@geeksforgeeks.org

Company

About Us

Careers

In Media

Learn

DSA

Algorithms

Data Structures

Start Your Coding Journey Now!

Copyright Policy

Advertise with us

CS Subjects

Video Tutorials

Courses

News

Top News

Technology

Work & Career

Business

Finance

Lifestyle

Knowledge

Languages

Python

Java

CPP

Golang

C#

SQL

Kotlin

Web Development

Web Tutorials

Django Tutorial

HTML

JavaScript

Bootstrap

ReactJS

NodeJS

Contribute

Write an Article

Improve an Article

Pick Topics to Write

Write Interview Experience

Internships

Video Internship