int (1 sign bit + 31 data bits) keyword in C

Program error signals

Why array index starts from zero ?

TCP Server-Client implementation in C

How to return multiple values from a function in C or C++?

Dynamic Memory Allocation in C using malloc(), calloc(), free() and realloc()

Commonly

Asked C Programming Interview Questions | Set 3

Applications of Pointers in C/C++

Pre-increment and Post-increment in C/C++

Sum of array Elements without using loops and recursion

“static const” vs “#define” vs “enum”

Comments in C/C++

How will you print numbers from 1 to 100 without using loop? | Set-2

How to find Segmentation Error in C & C++ ? (Using

GDB)

Why strcpy and strncpy are not safe to use?

time() function in C

Loader in C/C++

GDB (Step by Step Introduction)

Communication between two process using signals in C

tolower() function in C

Passing Reference to a Pointer in C++

Difference between const int*, const int * const, and int const *

How does a C program executes?

How to avoid Structure Padding in C?

Difference between Call by Value and Call by Reference

strrev() function in C

Interesting facts about C Language

Difference between fundamental data types and derived data types

C++: Methods of code shortening in competitive programming

Inline function in C

Segmentation Fault (SIGSEGV) vs Bus Error (SIGBUS)

Segmentation fault(SIGSEGV) and **Bus error(SIGBUS)** are signals generated when serious program error is detected by the operating system and there is no way the program could continue to execute because of these errors.

1) Segmentation Fault (also known as SIGSEGV and is usually signal 11) occur when the program tries to write/read outside the memory allocated for it or when writing memory which can only be read.In other words when the program tries to access the memory to which it doesn't have access to. SIGSEGV is abbreviation for "Segmentation Violation".

Few cases where SIGSEGV signal generated are as follows,
-> Using uninitialized pointer
-> De-referencing a NULL pointer
-> Trying to access memory that the program doesn't own (eg. trying to access an array element
out of array bounds).
-> Trying to access memory which is already de-allocated (trying to use dangling pointers).
Please refer this article for examples.

2) Bus Error (also known as SIGBUS and is usually signal 10) occur when a process is trying to access memory that the CPU cannot physically address.In other words the memory tried to access by the program is not a valid memory address.It caused due to alignment issues with the CPU (eg. trying to read a long from an address which isn't a multiple of 4). SIGBUS is abbrivation for "Bus Error".

SIGBUS signal occurs in below cases,
-> Program instructs the CPU to read or write a specific physical memory address

which is not valid / Requested physical address is unrecognized by the whole computer system.
-> Unaligned access of memory (For example, if multi-byte accesses must be 16 bit-aligned, addresses (given in bytes) at 0, 2, 4, 6, and so on would be considered aligned and therefore accessible, while addresses 1, 3, 5, and so on would be considered unaligned.)

The main difference between Segmentation Fault and Bus Error is that Segmentation Fault indicates an invalid access to a valid memory, while Bus Error indicates an access to an invalid address.

Below is an example of Bus Error taken from wikipedia.

```
// C program to demonstrate Bus Error
#include <stdlib.h>

int main(int argc, char **argv)
{

#if defined(__GNUC__)
# if defined(__i386__)
    /* Enable Alignment Checking on x86 */
    __asm__("pushf\norl $0x40000,(%esp)\npopf");
# elif defined(__x86_64__)
     /* Enable Alignment Checking on x86_64 */
    __asm__("pushf\norl $0x40000,(%rsp)\npopf");
# endif
#endif

    /* malloc() always provides aligned memory */
    char *cptr = malloc(sizeof(int) + 1);

    /* Increment the pointer by one, making it
       misaligned */
    int *iptr = (int *) ++cptr;

    /* Dereference it as an int pointer, causing
       an unaligned access */
    *iptr = 42;

    /* Following accesses will also result in
       sigbus error.
       short *sptr;
       int    i;

       sptr = (short *)&i;

       // For all odd value increments, it will
       // result in sigbus.
       sptr = (short *)(((char *)sptr) + 1);
       *sptr = 100;    */

    return 0;
}
```

Output :

```
Bad memory access (SIGBUS)
```

This article is contributed by **Prashanth Pangera**. 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.

Recommended Posts:

Core Dump (Segmentation fault) in C/C++

How to find Segmentation Error in C & C++ ? (Using GDB)

Memory Segmentation in 8086 Microprocessor

Else without IF and L-Value Required Error in C

Program error signals

Error Handling in C programs

Dividing a Large file into Separate Modules in C/C++, Java and Python

C program to store Student records as Structures and Sort them by Name

Similarities and Differences between Ruby and C language

Program to copy the contents of one array into another in the reverse order

Interesting facts about C Language

Program to Reverse a String using Pointers

Difference between Structure and Array in C

Structured Programming Approach with Advantages and Disadvantages

Article Tags : C C-Pointers system-programming

Practice Tags : C

1

□ To-do □ Done

1

Based on **2** vote(s)

Feedback Add Notes Improve Article

Please write to us at contribute@geeksforgeeks.org to report any issue with the above content.

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

Load Comments Share this post!

GeeksforGeeks
A computer science portal for geeks

710-B, Advant Navis Business Park,
Sector-142, Noida, Uttar Pradesh - 201305
feedback@geeksforgeeks.org

COMPANY

About Us
Careers
Privacy Policy
Contact Us

LEARN

Algorithms
Data Structures
Languages
CS Subjects
Video Tutorials

PRACTICE

Company-wise
Topic-wise
Contests
Subjective Questions

CONTRIBUTE

Write an Article
Write Interview Experience
Internships
Videos