Do

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

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

Core Dump/Segmentation fault is a specific kind of error caused by accessing memory that “does not belong to you.”

- When a piece of code tries to do read and write operation in a read only location in memory or freed block of memory, it is known as core dump.
- It is an error indicating memory corruption.

Common segmentation fault scenarios:

- **Modifying a string literal :**
  The below program may crash (gives segmentation fault error) because the line *(str+1) = ‘n’ tries to write a read only memory.

```
int main()
{
   char *str;

   /* Stored in read only part of data segment */
   str = "GfG";

   /* Problem:  trying to modify read only memory */
   *(str+1) = 'n';
   return 0;
}
```

```
Abnormal termination of program.
```

  Refer Storage for Strings in C for details

- **Accessing an address that is freed :**
  Here in the below code, the pointer p is dereferenced after freeing the memory block, which is not allowed by the compiler. So it produces the error segment fault or abnormal program termination at runtime.
  Example:

```c
// C program to illustrate
// Core Dump/Segmentation fault
#include <stdio.h>
#include<alloc.h>
int main(void)
{
    // allocating memory to p
    int* p = malloc(8);
    *p = 100;

    // deallocated the space allocated to p
    free(p);

    // core dump/segmentation fault
    //  as now this statement is illegal
    *p = 110;

    return 0;
}
```

Output:

```
Abnormal termination of program.
```

- **Accessing out of array index bounds :**

```cpp
// C++ program to demonstrate segmentation
// fault when array out of bound is accessed.
#include <iostream>
using namespace std;

int main()
{
   int arr[2];
   arr[3] = 10;  // Accessing out of bound
   return 0;
}
```

Output:

```
Abnormal termination of program.
```

This article is contributed by **Bishal Kumar Dubey**. 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:

Segmentation Fault (SIGSEGV) vs Bus Error (SIGBUS)
How to find Segmentation Error in C & C++ ? (Using GDB)
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
Commonly used String functions in C/C++ with Examples
Program to check if two strings are same or not
putchar() function in C
Difference between C and C++

Article Tags : C C-Dynamic Memory Allocation

Practice Tags : C

9

To-do Done

1.8

Based on **43** 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!