Content 51

Callback Functions Using Function Pointers In C

#### What is a Callback Function?

If a function's reference is passed to another function as an argument to call it, it will be called a Callback function. This allows the programmer to write less code and do more things. A callback function is a function that is called by using a function pointer. **Function Pointer**, as its name suggests, is a pointer to a function.

The following are some scenarios that help us in understanding the callback functions.

Let's say we have a function that downloads a file from the internet and takes a function pointer. When the download is complete or an error has happened, we would like to**execute the callback.**

void fileDownload (const char \*file, void (\*callback\_function)(int statusCode));

Let us have another example; we have a function that sets up a callback function that is called when a click has been detected on a button.

**Difference between Function Pointer and Callback Functions**

* A function pointer is a pointer that points to the function. Whereas, callback functions are function pointers passed as parameters of the function, which can be called back when an event occurs.
* A function pointer is an address of the function. Whereas, callback functions pass function pointer as an argument, and the caller would callback if something happens.

Below there is a program whish is used to Explain the Callback function:

In which I had used a Function pointer to make the self communication or direct communication between them through address (because pointer are used).

#include <stdio.h>

#include <stdlib.h>

int sum(int a, int b)

{

    return a + b;

}

void Hello\_input\_and\_execute(int (\*fptr)(int, int))

{

    printf("\nHello\n");

    printf("The sum is(In HELLO): %d", fptr(7, 3));

}

void gm\_input\_and\_execute(int (\*fptr)(int, int))

{

    printf("\nGood Morning\n");

    printf("The sum is(In Gm): %d\n", fptr(4, 2));

}

int main(int argc, char const \*argv[])

{

    system("cls");

    int (\*ptr)(int, int);

    ptr = sum;

    printf("The sum is(for check): %d\n", sum(2, 3));

    gm\_input\_and\_execute(ptr);

    Hello\_input\_and\_execute(ptr);

    return 0;

}

**Output:**

The sum is(for check): 5

Good Morning

The sum is(In Gm): 6

Hello

The sum is(In HELLO): 10