Function Pointer in C

In C, like normal data pointers (int *, char *, etc), we can have pointers to functions. Following is a simple example that shows declaration and function call using function pointer.

```c
#include <stdio.h>
// A normal function with an int parameter
// and void return type
void fun(int a)
{
    printf("Value of a is %d\n", a);
}

int main()
{
    // fun_ptr is a pointer to function fun()
    void (*fun_ptr)(int) = &fun;

    /* The above line is equivalent of following two
       void (*fun_ptr)(int);
       fun_ptr = &fun;
    */

    // Invoking fun() using fun_ptr
    (*fun_ptr)(10);

    return 0;
}
```

Output:

```
Value of a is 10
```

Why do we need an extra bracket around function pointers like fun_ptr in above example?
If we remove bracket, then the expression "void (*fun_ptr)(int)" becomes "void *fun_ptr(int)" which is declaration of a function that returns void pointer. See following post for details.
How to declare a pointer to a function?

Methods to concatenate string in C/C++ with Examples

Program to Reverse a String using Pointers

Code Optimization Technique (logical AND and logical OR)

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

Difference between scanf() and gets() in C

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

Interesting facts about C Language

Commonly used String functions in C/C++ with Examples

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

Program to check if two strings are same or not

Compiling with g++

Difference between C and Python

Difference between Structure and Array in C

Check if given Preorder, Inorder and Postorder traversals are of same tree | Set 2

stores the start of executable code.

2) Unlike normal pointers, we do not allocate de-allocate memory using function pointers.

3) A function's name can also be used to get functions' address. For example, in the below program, we have removed address operator '&' in assignment. We have also changed function call by removing *, the program still works.

```
#include <stdio.h>
// A normal function with an int parameter
// and void return type
void fun(int a)
{
    printf("Value of a is %d\n", a);
}

int main()
{
    void (*fun_ptr)(int) = fun;  // & removed

    fun_ptr(10);  // * removed

    return 0;
}
```

Output:

```
Value of a is 10
```

4) Like normal pointers, we can have an array of function pointers. Below example in point 5 shows syntax for array of pointers.

5) Function pointer can be used in place of switch case. For example, in below program, user is asked for a choice between 0 and 2 to do different tasks.

```
#include <stdio.h>
void add(int a, int b)
{
    printf("Addition is %d\n", a+b);
}
```


Most popular in C

Lex program to count the number of lines, spaces and tabs

Why to use fgets() over scanf() in C?

Thread functions in C/C++

Difference between Java and C language

Basic Input and Output in C

What does main() return in C and C++?

C program to check if a given string is Keyword or not

SDL library in C/C++ with examples


```
void subtract(int a, int b)
{
    printf("Subtraction is %d\n", a-b);
}
void multiply(int a, int b)
{
    printf("Multiplication is %d\n", a*b);
}

int main()
{
    // fun_ptr_arr is an array of function pointers
    void (*fun_ptr_arr[])(int, int) = {add, subtract, multiply};
    unsigned int ch, a = 15, b = 10;

    printf("Enter Choice: 0 for add, 1 for subtract and 2 "
           "for multiply\n");
    scanf("%d", &ch);

    if (ch > 2) return 0;

    (*fun_ptr_arr[ch])(a, b);

    return 0;
}
```

```
Enter Choice: 0 for add, 1 for subtract and 2 for multiply
2
Multiplication is 150
```

6) Like normal data pointers, a function pointer can be passed as an argument and can also be returned from a function.
For example, consider the following C program where wrapper() receives a void fun() as parameter and calls the passed function.

```
// A simple C program to show function pointers as parameter
#include <stdio.h>

// Two simple functions
void fun1() { printf("Fun1\n"); }
void fun2() { printf("Fun2\n"); }

// A function that receives a simple function
// as parameter and calls the function
void wrapper(void (*fun)())
{
    fun();
}

int main()
```


More related articles in C

chdir() in C language with Examples

Assignment Operators in C/C++

putchar() function in C

OpenMP | Hello World program

Structured Programming Approach with Advantages and Disadvantages

```
{
    wrapper(fun1);
    wrapper(fun2);
    return 0;
}
```

This point in particular is very useful in C. In C, we can use function pointers to avoid code redundancy. For example a simple qsort() function can be used to sort arrays in ascending order or descending or by any other order in case of array of structures. Not only this, with function pointers and void pointers, it is possible to use qsort for any data type.

```
// An example for qsort and comparator
#include <stdio.h>
#include <stdlib.h>

// A sample comparator function that is used
// for sorting an integer array in ascending order.
// To sort any array for any other data type and/or
// criteria, all we need to do is write more compare
// functions.  And we can use the same qsort()
int compare (const void * a, const void * b)
{
  return ( *(int*)a - *(int*)b );
}

int main ()
{
  int arr[] = {10, 5, 15, 12, 90, 80};
  int n = sizeof(arr)/sizeof(arr[0]), i;

  qsort (arr, n, sizeof(int), compare);

  for (i=0; i<n; i++)
     printf ("%d ", arr[i]);
  return 0;
}
```

Output:

```
5 10 12 15 80 90
```

Similar to qsort(), we can write our own functions that can be used for any data type and can do different tasks without code redundancy. Below is an example search function that can be used for any data type. In fact we can use this search function to find close elements (below a threshold) by writing a customized compare function.

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

```
// A compare function that is used for searching an integer
// array
bool compare (const void * a, const void * b)
{
  return ( *(int*)a == *(int*)b );
}

// General purpose search() function that can be used
// for searching an element *x in an array arr[] of
// arr_size. Note that void pointers are used so that
// the function can be called by passing a pointer of
// any type.  ele_size is size of an array element
int search(void *arr, int arr_size, int ele_size, void *x,
           bool compare (const void * , const void *))
{
    // Since char takes one byte, we can use char pointer
    // for any type/ To get pointer arithmetic correct,
    // we need to multiply index with size of an array
    // element ele_size
    char *ptr = (char *)arr;

    int i;
    for (i=0; i<arr_size; i++)
        if (compare(ptr + i*ele_size, x))
           return i;

    // If element not found
    return -1;
}

int main()
{
    int arr[] = {2, 5, 7, 90, 70};
    int n = sizeof(arr)/sizeof(arr[0]);
    int x = 7;
    printf ("Returned index is %d ", search(arr, n,
                                sizeof(int), &x, compare));
    return 0;
}
```

Output:

```
Returned index is 2
```

The above search function can be used for any data type by writing a separate customized compare().

7) Many object oriented features in C++ are implemented using function pointers in C. For example virtual functions. Class methods are another example implemented using function pointers. Refer this book for more details.

Related Article:Pointers in C and C++ | Set 1 (Introduction, Arithmetic and Array)

References:

http://www.cs.cmu.edu/~ab/15-123S11/AnnotatedNotes/Lecture14.pdf

http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-087-practical-programming-in-c-january-iap-2010/lecture-notes/MIT6_087IAP10_lec08.pdf

http://www.cs.cmu.edu/~guna/15-123S11/Lectures/Lecture14.pdf

This article is contributed by Abhay Rathi. Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

Recommended Posts:

How to declare a pointer to a function?

Double Pointer (Pointer to Pointer) in C

'this' pointer in C++

void pointer in C / C++

NULL pointer in C

C++ | this pointer | Question 5

C++ | this pointer | Question 4

C++ | this pointer | Question 3

C++ | this pointer | Question 2

C++ | this pointer | Question 1

Pointer to an Array | Array Pointer

Pointer vs Array in C

A C/C++ Pointer Puzzle

Opaque Pointer

Passing Reference to a Pointer in C++

Article Tags : C C-Pointers CPP-Functions cpp-pointer

Practice Tags : C

22

To-do Done

3.5

Based on 101 vote(s)

Feedback/ Suggest Improvement | Add Notes | Improve Article

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

Previous
Generic Linked List in C

Next
How to write long strings in Multi-lines C/C++?

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

Load Comments