

C Pointers



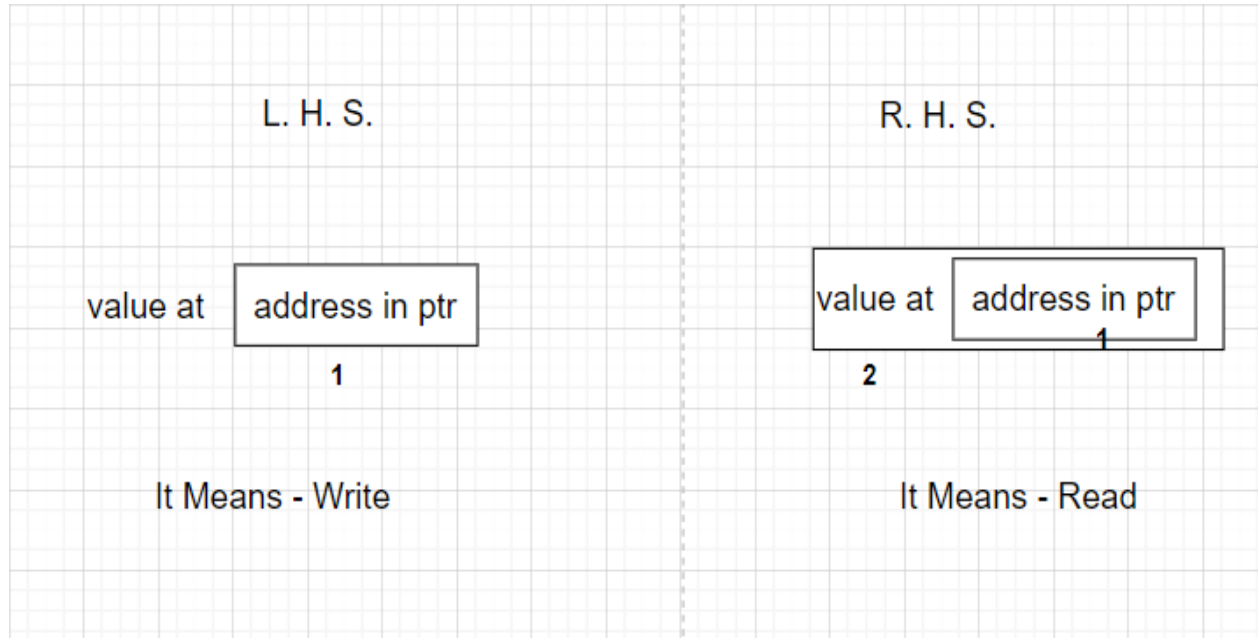
Pointer is a special data type. Pointer is a variable which stores the address of another variable. The biggest advantage of pointer is, it can access (read, write, edit) the variable beyond the limitation of stack frame, if address is given/ passed/ assign correctly.

*ptr -> represents value at the address (address stored in ptr)
ptr -> it expects address

C Pointers evaluation:



FirstBitSolutions.com
...Learn IT Bit by Bit...



- We can assign pointer to any other pointer variable (of same data type)
- We can pass pointer to any other function.
- We can receive pointer as a formal parameter.
- We can return pointer from the function.

C Pointers



- The data type of pointer doesn't decide how much memory will be allocated to that pointer variable.
- The data type of pointer will decide its strength, that is how many bytes that pointer variable can access.
- -Pointer size is compiler specific i.e it can be 8 bytes in some compilers

Data Type	Size (in byte)	Strength
char*	4 <u>byte</u>	1 byte
int*	4 <u>byte</u>	4 <u>byte</u>
double*	4 <u>byte</u>	8 <u>byte</u>
void *	4 <u>byte</u>	0 byte

C Pointers and Functions



FirstBitSolutions.com
...Learn IT Bit by Bit...

1. Pass address to a function

```
#include <stdio.h>
void swap(int *n1, int *n2);

int main()
{
    int num1 = 5, num2 = 10;

    // address of num1 and num2 is passed
    swap( &num1, &num2);

    printf("num1 = %d\n", num1);
    printf("num2 = %d", num2);
    return 0;
}

void swap(int* n1, int* n2)
{
    int temp;
    temp = *n1;
    *n1 = *n2;
    *n2 = temp;
}
```

2. Pass pointer to a function

```
#include <stdio.h>

void addOne(int* ptr) {
    (*ptr)++; // adding 1 to *ptr
}

int main()
{
    int* p, i = 10;
    p = &i;
    addOne(p);

    printf("%d", *p); // 11
    return 0;
}
```

C Pointer Types



- Void Pointer – It is defined as an address of any variable, which has no standard data type and its created using void keyword.

```
#include <stdio.h>

int main()
{
    // void pointer
    void* ptr = NULL;

    printf("The size of pointer is:%d\n", sizeof(ptr));

    return 0;
}
```

- Null Pointer -It can be created by assigning a zero value during pointer declaration. This method is useful when no address is assigned to the pointer.

```
int * ptr = NULL;
```

C Pointer Types



- Wild Pointer – The pointers which are not initialized are called wild pointers and these pointers can cause program to crash.

```
#include <stdio.h>

int main()
{
    // wild pointer
    int* ptr;

    printf("\n%d", *ptr);

    return 0;
}
```

- Dangling Pointer – The pointers which are pointing to a deallocated memory block are known as Dangling pointers. The dangling pointer errors can be avoided by initializing the pointer to the **NULL** value.