

# How to increment a pointer address and pointer's value?

[Ask Question](#)

Asked 7 years, 8 months ago Active 1 year, 1 month ago Viewed 196k times



Let us assume,

76

```
int *p;
int a = 100;
p = &a;
```



What will the following code will do actually and how?

90

```
p++;
++p;
++*p;
++(*p);
++*(p);
*p++;
(*p)++;
*(p)++;
*++p;
*(++p);
```

I know, this is kind of messy in terms of coding, but I want to know what will actually happen when we code like this.

Note : Lets assume that the address of `a=5120300` , it is stored in pointer `p` whose address is `3560200` . Now, what will be the value of `p & a` after the execution of each statement?

`c` `pointers`

[share](#) [improve this question](#)

edited Mar 15 '15 at 11:39

[Thomas131](#)  
39 ● 11

asked Nov 21 '11 at 6:23

[Dinesh](#)  
10.5k ● 18 ● 59 ● 101

3 why don't u just run it in the debugger? – [Anders](#) Nov 21 '11 at 6:27

19 Well .. why not simply *try* it and see? `printf` will print a pointer with `%p` – [Brian Roach](#) Nov 21 '11 at 6:29



If you're curious about behavior, just play around with it. Just write a simple c program that goes through all of these use cases and see if it makes sense to you. – [Cyrus](#) Nov 21 '11 at 6:36

@AndersK. Maybe the OP expects undefined behavior? ...Or maybe not. – [Mateen Ulhaq](#) Nov 21 '11 at 6:39

[add a comment](#)

## Blog

[Find Help Online Mindfully — and Effectively!](#)

## Featured on Meta

[Custom Filters release announcement](#)

[Unicom Meta Zoo #6: Interview with Catija](#)

[The world is big and I am SO small. What are the implications for our meta...](#)

[Creating a system for featuring posts. Tell the mods what you want](#)

[We're removing "Hot Meta Posts" from Stack Overflow's sidebar for now;...](#)

## Jobs near you

**Senior / Lead / Principal Platform/Back-End Software Engineer**

Salesforce Atlanta, GA

`java` `sql`

**.Net Software Engineer - Front End**

Juvarre Atlanta, GA

\$100K - \$125K

`c#` `.net`

**Angular or React + TypeScript Engineer**

Stable Kernel Atlanta, GA

`angular` `typescript`

**Staff Systems Engineer – Infrastructure**



134

First, the ++ operator takes precedence over the \* operator, and the () operators take precedence over everything else.

Second, the ++number operator is the same as the number++ operator if you're not assigning them to anything. The difference is number++ returns number and then increments number, and ++number increments first and then returns it.

Third, by increasing the value of a pointer, you're incrementing it by the sizeof its contents, that is you're incrementing it as if you were iterating in an array.

So, to sum it all up:

```
ptr++; // Pointer moves to the next int position (as if it was an array)
++ptr; // Pointer moves to the next int position (as if it was an array)
++*ptr; // The value of ptr is incremented
++(*ptr); // The value of ptr is incremented
++(ptr); // The value of ptr is incremented
*ptr++; // Pointer moves to the next int position (as if it was an array). But re
(*ptr)++; // The value of ptr is incremented
*(ptr)++; // Pointer moves to the next int position (as if it was an array). But re
**ptr; // Pointer moves to the next int position, and then get's accessed, with
*(++ptr); // Pointer moves to the next int position, and then get's accessed, with
```

As there are a lot of cases in here, I might have made some mistake, please correct me if I'm wrong.

EDIT:

So I was wrong, the precedence is a little more complicated than what I wrote, view it here:

[http://en.cppreference.com/w/cpp/language/operator\\_precedence](http://en.cppreference.com/w/cpp/language/operator_precedence)

share improve this answer

edited Mar 12 '18 at 15:52



answered Nov 21 '11 at 6:34



4 \*ptr++, the value is not incremented, the pointer is. These unary operators have the same precedence but they are evaluated right-to-left. The code means "take the contents from where ptr points at, then increment ptr". It is very common C code (and yes, quite confusing). Please correct this and I'll remove the downvote. Same for \*(ptr)++, the parenthesis does nothing. — Lundin Nov 21 '11 at 7:43

Thank you very much Lundin, did I miss anything else? — felipemaia Nov 21 '11 at 15:03

@Lundin Hi, is the answer above corrected now? Thanks. — Unheilig Mar 11 '14 at 22:51

4 @Unheilig The first sentence is still completely wrong, postfix ++ takes precedence over unary \* which has the same precedence as prefix ++. Apart from that, it seems ok. — Lundin Mar 12 '14 at 7:19

3 @felipemaia Are you sure that it will segfault? Maybe its just undefined behaviour? — jotik Apr 21 '16 at 10:59

add a comment



10

checked the program and the results are as,

```
p++; // use it then move to next int position
++p; // move to next int and then use it
++*p; // increments the value by 1 then use it
```

More jobs near Atlanta...

## Linked

- 4 Why does ++ not increment correctly?
- 2 What value changed if we use increment/decrement on an address?
- 3 operator precedence in c with pointers
- 1 Array Pointers in C
- 1 iterating through a struct in c
- 4 what is the difference between ++\*var++ and ++Var++?
- 2 const char \*\*a = {"string1","string2"} and pointer arithmetic
- 5 What does pointer increase do? (C/C++)
- 0 Function not updating an int field
- 7 What will be the output of the following program :

[see more linked questions...](#)

## Related

- 2373 How do you set, clear, and toggle a single bit?
- 3007 What are the differences between a pointer variable and a reference variable in C++?
- 1665 What is a smart pointer and when should I use one?
- 1134 How do function pointers in C work?
- 76 Print the address or pointer for value in C
- 207 What is exactly the base pointer and stack pointer? To what do they point?
- 2803 Improve INSERT-per-second performance of SQLite?
- 205 What exactly is a C pointer if not a memory address?
- 1490 Why should I use a pointer rather than the object itself?
- 74 Pointers to pointers vs. normal pointers

```

++(*p); // increments the value by 1 then use it
++*(p); // increments the value by 1 then use it
*p++; // use the value of p then moves to next position
(*p)++; // use the value of p then increment the value
*(p)++; // use the value of p then moves to next position
**++p; // moves to the next int location then use that value
*(++p); // moves to next location then use that value

```

share improve this answer

edited Apr 21 '16 at 10:16

answered Mar 23 '15 at 6:35



Eng.Fouad

87.2k ● 57 ● 262 ● 368



Sujith R Kumar

109 ● 1 ● 5

1 'use it'. how ? – alex Jun 21 '15 at 22:10

2 @alex use it means for example, consider statement, 'int \*a = p++;' Here first value of pointer 'p' will be used to and after that p will move to next position. So in effect after executing above statement 'a' will have address of previous location pointed by 'p' and 'p' will be pointing to next position. That is first use the value of 'p' for the assignment expression as above and then increment value of 'p' to point to next position – Sujith R Kumar Jun 23 '15 at 14:49

In short, I think he uses the phrase "use it" for the more formal term "assign". That is all. – Apekshik Panigrahi Mar 2 at 15:34

add a comment

3

With regards to "How to increment a pointer address and pointer's value?" I think that `++(*p++)` is actually well defined and does what you're asking for, e.g.:

```

#include <stdio.h>

int main() {
    int a = 100;
    int *p = &a;
    printf("%p\n", (void*)p);
    ++(*p++);
    printf("%p\n", (void*)p);
    printf("%d\n", a);
    return 0;
}

```

It's not modifying the same thing twice before a sequence point. I don't think it's good style though for most uses - it's a little too cryptic for my liking.

share improve this answer

answered Nov 21 '11 at 10:24



Flexo ♦

71.8k ● 22 ● 154 ● 237

Actually, the parentheses are unnecessary: `++*p++` will successfully increment both value and pointer (postfix `++` binds stronger than dereference `*`, and that happens before the prefix `++` due to order). The parentheses are only necessary when you need the value before incrementing it `( *p++)++`. If you go all-prefix, `++*++p` will work just fine without parentheses as well (but increment the value that is pointed at after the pointer increment). – cmaster Mar 12 '18 at 16:11

add a comment

## Hot Network Questions

- DE Would people understand me speaking German all over Europe?
- ask Trying to open a new ubuntu terminal window from the existing window
- L Self-deportation of American Citizens from US
- [S] Is The Venice Syndrome documentary cover photo real?
- P Why does Canada require bilingualism in a lot of federal government posts?
- How can chaotic entities be prevented from spreading beyond their domain?
- What is an Accessible Word?
- DE What is the German equivalent of the proverb 水清ければ魚棲まず (if the water is clear, fish won't live there)?
- Can Lightning Lure be used to knock out a creature like a magical Taser?
- Should I accept an invitation to give a talk from someone who might review my proposal?
- Problem with Eigenvectors
- Assuring luggage isn't lost with short layover
- What are the closest international airports in different countries?
- Why tantalum for the Hayabusa bullets?
- GNU sort stable sort when sort does not know sort order
- What is a good example for artistic ND filter applications?
- What clothes would flying-people wear?
- How did astronauts using rovers tell direction without compasses on the Moon?
- Composing fill in the blanks
- Shouldn't there be "us" instead of "our" in this sentence?
- Classic vs Modern Experience
- Why is my fluorescent tube orange on one side, white on the other and dark in the middle?
- What do I do with a party that is much stronger than their level?
- What is the reason for cards stating "Until end of turn, you don't lose this mana as steps and phases end"?
- Question feed

The following is an instantiation of the various "just print it" suggestions. I found it instructive.

2

```
#include "stdio.h"

int main() {
    static int x = 5;
    static int *p = &x;
    printf("(int) p    => %d\n", (int) p);
    printf("(int) p++ => %d\n", (int) p++);
    x = 5; p = &x;
    printf("(int) ++p => %d\n", (int) ++p);
    x = 5; p = &x;
    printf("++*p      => %d\n", ++*p);
    x = 5; p = &x;
    printf("++(*p)    => %d\n", ++(*p));
    x = 5; p = &x;
    printf("++*(p)     => %d\n", ++*(p));
    x = 5; p = &x;
    printf("*p++      => %d\n", *p++);
    x = 5; p = &x;
    printf("( *p)++    => %d\n", (*p)++);
    x = 5; p = &x;
    printf("( *p)++    => %d\n", (*p)++);
    x = 5; p = &x;
    printf("++*p      => %d\n", ++*p);
    x = 5; p = &x;
    printf("(++*p)    => %d\n", (++*p));
    return 0;
}
```

It returns

```
(int) p    => 256688152
(int) p++  => 256688152
(int) ++p  => 256688156
++*p      => 6
++(*p)    => 6
++*(p)    => 6
*p++      => 5
(*p)++    => 5
(*p)++    => 5
++*p      => 0
(++*p)    => 0
```

I cast the pointer addresses to `int` s so they could be easily compared.

I compiled it with GCC.

[share](#) [improve this answer](#)

[edited Feb 5 '18 at 1:52](#)

[answered Feb 5 '18 at 1:27](#)



[Rico Picone](#)  
221 ● 2 ● 9

I would change this to include the value of x and p after the operation. – [NetJohn](#) Mar 16 '18 at 18:50

[add a comment](#)

0

Note:

- 1) Both ++ and \* have same precedence(priority), so the associativity comes
- 2) in this case Associativity is from \*\*Right-Left\*\*

important table to remember in case of pointers and arrays:

operators	precedence	associativity
1) ( ) , [ ]	1	left-right
2) * , identifier	2	right-left
3) <data type>	3	-----

let me give an example, this might help;

```
char **str;
str = (char **)malloc(sizeof(char*)*2); // allocate mem for 2 char*
str[0]=(char *)malloc(sizeof(char)*10); // allocate mem for 10 char
str[1]=(char *)malloc(sizeof(char)*10); // allocate mem for 10 char

strcpy(str[0],"abcd"); // assigning value
strcpy(str[1],"efgh"); // assigning value

while(*str)
{
    cout<<*str<<endl; // printing the string
    *str++;           // incrementing the address(pointer)
                    // check above about the precedence and associativity
}
free(str[0]);
free(str[1]);
free(str);
```

share improve this answer

edited Jun 24 '18 at 6:39

answered Jun 24 '18 at 6:04



Abhishek D K

581 ● 5 ● 14

add a comment

Your Answer

**B** *I*

Sign up or [log in](#)

 Sign up using Google

 Sign up using Facebook

 Sign up using Email and Password

Post Your Answer

By clicking "Post Your Answer", you agree to our [terms of service](#), [privacy policy](#) and [cookie policy](#)

Not the answer you're looking for? Browse other questions tagged [c](#) [pointers](#) or [ask your own question](#).

Post as a guest

Name

Email

Required, but never shown



## STACK OVERFLOW

Questions  
Jobs  
Developer Jobs Directory  
Salary Calculator  
Help  
Mobile  
Disable Responsiveness

## PRODUCTS

Teams  
Talent  
Advertising  
Enterprise

## COMPANY

About  
Press  
Work Here  
Legal  
Privacy Policy  
Contact Us

## STACK EXCHANGE NETWORK

Technology >  
Life / Arts >  
Culture / Recreation >  
Science >  
Other >

[Blog](#) [Facebook](#) [Twitter](#) [LinkedIn](#)

site design / logo © 2019 Stack Exchange Inc; user contributions  
licensed under cc by-sa 3.0 with attribution required.  
rev 2019.7.30.34490