What is the difference between K&R and One True Brace Style (1TBS) styles?

Asked 10 years, 8 months ago Modified 6 years, 10 months ago Viewed 61k times

52

I have read the Wikipedia article on Indent Styles, but I still don't understand. What is the difference between K&R and 1TBS?

coding-style indentation

13

Share Improve this question

Follow

edited Aug 9, 2011 at 14:05

Jason Plank

115 6

asked Aug 8, 2011 at 21:45

GavinR

623 1 5 7

I read somewhere that the style in K&R was governed by space considerations - i.e. to reduce the vertical space the code took up in the book. – ChrisF ♦ Aug 9, 2011 at 14:14

@ChrisF it also reduces vertical space on the screen. When we had 80 col x 25 line terminals it was worth it! – Martin Beckett Aug 9, 2011 at 15:37

7 Apple's "goto fail" is a great example of a serious bug that would surely have been prevented by using 1TBS: imperialviolet.org/2014/02/22/applebug.html – user126361 Apr 9, 2014 at 5:31

5 Apple's bug could also have been prevented by putting the statement on a single line, proofreading, using a dead code checker, or using an indent-sensitive language. – Cees Timmerman May 3, 2015 at 23:59

1 @CeesTimmerman, .. or having tests... – thoni56 Feb 9, 2018 at 15:57

3 Answers

Sorted by: Highest score (default)

82

The biggest difference between K&R and the One True Brace Style (1TBS) is that in the 1TBS, all `if`, `else`, `while`, and `for` statements have opening and closing braces, even if they aren't necessary. The purpose is to make it easy to insert new statements and know exactly how they will be grouped.

As an example:

K&R:

```
int i;
for (i = 0; i < 10; i++)
  printf("Hi.");
```

1TBS:

```
int i;
for (i = 0; i < 10; i++) {
  printf("Hi");
}
```

Share Improve this answer Follow edited Aug 9, 2011 at 0:02

answered Aug 8, 2011 at 21:52

Thomas Owens ♦
75.2k 17 186 266

26

K&R is like this:

```
if (x)
    a();
else {
    b();
    c();
}
```

That is: braces used only where needed, opening brace on same line as controlling statement, closing brace on its own line.

The "one true brace style" (1TBS or OTBS) turns a single controlled statement into a compound statement by enclosing it in braces:

```
if (x) {
    a();
} else {
    b();
    c();
}
```

Allman style goes a bit further than 1TBS, and forces vertical spacing by placing the opening brace on a line by itself as well:

```
if (x)
{
    a();
}
else
{
    b();
    c();
}
```

Edit:

I'm still trying to figure out exactly how it qualifies as "arrogant" to say "Dennis Ritchie was an *extremely* smart guy who not only invented a good language, but also came up with a really

good brace style for it."

For those who insist that it's being arrogant anyway, here's a little challenge: go to Sourceforge, Github (etc.) and pick out projects using the K&R brace style. Go through their records of bugs and commits, and try to find a *single* bug that was caused by the brace style they used.

If you don't want to do that much work, try doing a simple statistical analysis. Compare projects using different brace styles, and see if you can show "bimodality" -- a statistically significant difference in bug counts (severity, etc.) that correlates with bracing style.

I did both of these a few years ago, and couldn't find a single bug that I could attribute to bracing styles, nor could I find anything approaching a statistically significant correlation between the two. On average, those using K&R bracing had slightly fewer bugs, but the difference was *much* too small to qualify as statistically significant.

Since it was brought up, I'll comment on the situation with multi-statement macros. A macro that includes multiple statements but doesn't surround them with braces itself, has a bug. My job is *not* to write code that covers up that bug. Quite the contrary, my job is to find and eradicate that bug as quickly as possible.

Writing code in the hope of its covering up bugs so they remain undiagnosed and un-fixed is downright evil. Call that arrogant if you like, but I do not see this as even close to negotiable. Bugs should be found and fixed, not covered up. The longer the exist, the more likely it is that they will become much more difficult and expensive to fix.

Share Improve this answer Follow

edited Jul 24, 2014 at 20:56
user53019

answered Aug 8, 2011 at 23:02

Jerry Coffin
43.7k 5 88 160

1 Deleted all the comments as they had descended in to bickering and noise. If you have a valid point then post it as an answer. If you want a discussion take it to chat – ChrisF ♦ Aug 9, 2011 at 15:17

8 doesn't the 1tbs put the } and else on one line ? Saving vertical space while preserving the wondrous beautiful symmetry is the point ! – Martin Beckett Aug 9, 2011 at 16:00

4 @Jerry - well any good holy war needs a few schisms ;-) – Martin Beckett Aug 9, 2011 at 17:09

8 goto fail; goto fail; – Jamie Pate Apr 30, 2014 at 17:58

14 Yes to follow up on @JamiePate's comment that is the Apple SSL bug which is analyzed here. There is an `if` statement with indented statements after it, thus appearing that they are both conditionally executed. But there are no braces! The second statement is really beyond the `if` and will always be executed, thus the bug. – Colin D Bennett Feb 27, 2015 at 20:10

11

The problem, in general with KR brace style is in code refactoring. When moving code around it is easy to miss that there are no braces around something, move it incorrectly (or move something under it thinking it is conditionally executed) and then either scratch your head when something no longer works, or be unfortunate and be in a code area not well tested and

the bug goes unnoticed until a black hat finds a way to exploit it. A quick trip into the debugger easily finds the problem if you notice it, but if you don't...

Share Improve this answer Follow

answered Jun 5, 2015 at 18:40

Justin Swanhart

227 2 3

2 this seems to merely repeat point made and explained in prior answers – gnat Jun 5, 2015 at 21:42

1 I don't think the other answers explain exactly how the problem arises in everyday code. They explain what OTB is, but not why it is actually important. Comments might address that, but not the answers. – Justin Swanhart Jun 7, 2015 at 18:53

Highly active question. Earn 10 reputation (not counting the association bonus) in order to answer this question. The reputation requirement helps protect this question from spam and non-answer activity.