

## Re: Long long int

**Source:** [http://coding.derkeiler.com/Archive/C\\_CPP/comp.lang.c/2004-02/3280.html](http://coding.derkeiler.com/Archive/C_CPP/comp.lang.c/2004-02/3280.html)

---

**From:** August Derleth (*see\_at\_sig.now*)

**Date:** 02/16/04

Date: Mon, 16 Feb 2004 03:15:45 -0700

RagnarDanneskjöld wrote:

> *Tom St Denis wrote:*

>

>> *"RagnarDanneskjöld" <noemail@hotmail.com> wrote in message*

>> *news:ERUXb.11093\$cE3.18040018@news4.srv.hcvlny.cv.net...*

>>

>>> *Well I have absolutely no idea what this odd-ball assembly code is doing*

>>> *so let's look at how a "real" compiler does it.*

>>

>>

>>

>> *You put that code in main where gcc imposes some limitations try*

>>

>

> *Damn why didn't I think to generate with a non-main function.*

>

> *When I said I had no idea what the heck it was doing: i was referring*

> *more to the syntax of GAS or AT&T or whatever assembly. It's zany.*

It's ugly as hell, and it's called AT&T. Try using the `-masm=intel` argument. That won't make it as clean as, say, nasm's assembly, but it will make it readable to someone coming from an Intel background.

AT&T's assembly is supposedly ugly to be usable as-is on multiple architectures. Supposedly, you can use AT&T syntax as a template for a very generic (non-processor-specific) assembly language, filling in the opcodes and their exact arguments as the compiler generates them.

Of course, that doesn't wash with me. First off, I doubt the benefits of trying to shoehorn Intel assembly into something that worked for VAX assembly, or PowerPC assembly into something that worked for System/360 assembly. Code reuse is good, but that is simple absurdity.

Secondly, trained monkeys could have come up with a better template. Memory references are particularly FUBARed: What does `-3(%eax,%edx,4)` mean? Looking it up, I find it means `[eax+edx*4-3]`. I chose a rather absurd example, but it isn't that odd to use rather complex addressing

comp.lang.c: Re: Long long int

schemes in a chip as register-starved as an x86 clone. And Intel's algebraic notation is clearly a win over AT&T's What-The-Fuck notation.

--

My address is yvoregnevna gjragl-guerr gjb-gubhfnaq guerr ng lnubb qbg pbz  
Note: Rot13 and convert spelled-out numbers to numerical equivalents.