float vs. double precision

The following code

```
float x = 3.141592653589793238;
double z = 3.141592653589793238;
printf("x=%f\n", x);
printf("z=%f\n", z);
printf("x=%20.18f\n", x);
printf("z=%20.18f\n", z);
```

will give you the output

```
x=3.141593
z=3.141593
x=3.141592741012573242
z=3.141592653589793116
```

where on the third line of output `741012573242` is garbage and on the fourth line `116` is garbage. Do doubles always have 16 significant figures while floats always have 7 significant figures? Why don't doubles have 14 significant figures?

c floating-point

edited Aug 6 '14 at 22:16
Vladimir
143 ● 11

asked Feb 23 '11 at 23:21
foo
733 ● 4 ● 12 ● 22

8 cse.msu.edu/~cse320/Documents/FloatingPoint.pdf – Robert Harvey ♦ Feb 23 '11 at 23:32

7 Answers

Floating point numbers in C use IEEE 754 encoding.

This type of encoding uses a sign, a significand, and an exponent.

Because of this encoding, many numbers will have small changes to allow them to be stored.

Also, the number of significant digits can change slightly since it is a binary representation, not a decimal one.

Single precision (float) gives you 23 bits of significand, 8 bits of exponent, and 1 sign bit.

Double precision (double) gives you 52 bits of significand, 11 bits of exponent, and 1 sign bit.

edited Dec 30 '14 at 20:59

answered Feb 23 '11 at 23:24
Alan Geleynse
16.6k ● 2 ● 31 ● 48

3 C99 does, previously it was up to the compiler. – Alan Geleynse Feb 23 '11 at 23:29

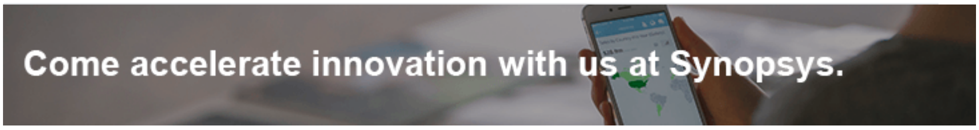
11 -1 This statement is blatantly false: "Because of this encoding, you can never guarantee that you will not have a change in your value." – R.. Feb 23 '11 at 23:46

11 @Alan: C99 does not require IEEE floating point; it just recommends it. – R.. Feb 23 '11 at 23:47

3 @Alan: R.. is correct; Annex F (which specifies IEEE-754 bindings) is normative, but only in effect if an implementation defines `__STDC_IEC_559__`. An implementation that does not define that macro is free not to conform to IEEE-754. – Stephen Canon Feb 24 '11 at 0:06

9

@Alan: Under IEEE 754, it's easily guaranteed that there is no change in the values `0.5`, `0.046875`, or `0.376739501953125` versus their decimal representations. (These are all diadic rationals with numerator fitting in the mantissa and base-2 logarithm of the denominator fitting in the exponent.) – R.. Feb 24 '11 at 1:12



> Do doubles always have 16 significant figures while floats always have 7 significant figures?

No. Doubles always have 53 significant **bits** and floats always have 24 significant **bits** (except for denormals, infinities, and NaN values, but those are subjects for a different question). These are binary formats, and you can only speak clearly about the precision of their representations in terms of binary digits (bits).

This is analogous to the question of how many digits can be stored in a binary integer: an unsigned 32 bit integer can store integers with up to 32 bits, which doesn't precisely map to any number of decimal digits: all integers of up to 9 decimal digits can be stored, but a lot of 10-digit numbers can be stored as well.

> Why don't doubles have 14 significant figures?

The encoding of a double uses 64 bits (1 bit for the sign, 11 bits for the exponent, 52 explicit significant bits and one implicit bit), which is *double* the number of bits used to represent a float (32 bits).

answered Feb 24 '11 at 0:11
Stephen Canon
77.1k ● 11 ● 126 ● 219

It's usually based on significant figures of both the exponent and significand in base 2, not base 10. From what I can tell in the C99 standard, however, there is no specified precision for floats and doubles (other than the fact that 1 and `1 + 1E-5` / `1 + 1E-7` are distinguishable [`float` and `double` repsectively]). However, the number of significant figures is left to the implementer (as well as which base they use internally, so in other words, an implementation could decide to make it based on 18 digits of precision in base 3). [1]

If you need to know these values, the constants `FLT_RADIX` and `FLT_MANT_DIG` (and `DBL_MANT_DIG` / `LDBL_MANT_DIG`) are defined in float.h.

The reason it's called a `double` is because the number of bytes used to store it is double the number of a float (but this includes both the exponent and significand). The IEEE 754 standard (used by most compilers) allocate relatively more bits for the significand than the exponent (23 to 9 for `float` vs. 52 to 12 for `double`), which is why the precision is more than doubled.

1: Section 5.2.4.2.2 (http://www.open-std.org/jtc1/sc22/wg14/www/docs/n1256.pdf)

edited Feb 24 '11 at 0:01
answered Feb 23 '11 at 23:24
user470379
4,138 ● 8 ● 18

Typo? C89 requires an epsilon of at most `1E-9` for `double`, not `1E-7`. – Rufflewind Oct 20 at 0:51 ✎

A float has 23 bits of precision, and a double has 52.

answered Feb 23 '11 at 23:25
Chris Nash
2,183 ● 10 ● 18

Detail: binary64 has a 53 bit significant (52 explicitly stored) binary32 has 24 bit (23 explicitly stored). – chux Feb 18 '15 at 22:16

It's not exactly *double* precision because of how IEEE 754 works, and because binary doesn't really translate well to decimal. Take a look at the standard if you're interested.

answered Feb 23 '11 at 23:26
Mehrdad
106k ● 66 ● 321 ● 632

The Double and Float variable types are different in the way that they store the values. Precision is the main difference where float is a single precision (32 bit) floating point data type, double is a double precision (64 bit) floating point data type .

Float - 32 bit (7 digits)

Double - 64 bit (15-16 digits)

The main difference is Floats and Doubles are binary floating point types and a Decimal will store the value as a floating decimal point type. So Decimals have much higher precision and are usually used within monetary (financial) applications that require a high degree of accuracy. But in performance wise Decimals are slower than double and float types.

More about...float vs. double

Rj

answered Sep 28 at 4:53
Rajesh
531 ● 3 ● 4

float : 23 bits of significand, 8 bits of exponent, and 1 sign bit.

double : 52 bits of significand, 11 bits of exponent, and 1 sign bit.

answered Aug 6 '15 at 2:16
abe
1,133 ● 12 ● 11