

Re: what is the best way of passing floats into a string

Source: <http://unix.derkeiler.com/Newsgroups/comp.unix.programmer/2007-01/msg00408.html>

- *From:* Pascal Bourguignon <pjb@xxxxxxxxxxxxxxxxxxxx>
 - *Date:* Sat, 20 Jan 2007 17:05:51 +0100
-

atv <alef@xxxxxxxx> writes:

```
I need to pass a float into a string (o_tail->y). I tried doing:  
o_tail->y=malloc(sizeof(float)+1);  
sprintf(o_tail->y,sizeof(float)+1,"%f",1.0);
```

I do not null-terminate as sprintf takes care of this (according to my man page). That's why i also account for the extra +1. However, this only copies 4 bytes.

Indeed, the size of the `_decimal_ _representation_` of a floating point number is totally unrelated to the number of bit used to represent this number in floating point. When you use the type float, you use a fixed number of bits (eg 32 bits) to represent a subset of the decimal numbers. All the numbers that can be represented as a float are represented with the same number of bits. As you can see if you read it, `sizeof(float)` is not dependant on the number, only on the type float. This is a property of the IEEE 754 representation of these numbers.

On the other hand, when we represent numbers in the usual notation, in base ten with a comma, and possibly an exponent factor, we have sizes of the notations that depend on the values of these numbers. For example, for one, we only need to write a one-character string: "1", but for a thousand, we need to write a four-character string: "1000".

So, the question is how much characters you will need to write a given floating point number? Actually, that depends, you can write it as wide as you want:

```
...  
0.000000000000000001  
0.000000000000000001e-1  
0.000000000000000001e-2  
0.000000000000000001e-3
```

Re: what is the best way of passing floats into a string

...
0.1e-15
1.0e-16
10.0e-17
100.0e-18
1000.0e-19
10000.0e-20
...
10000000000000000.0e-31
...

The minimum size for a simple number like 1 or 2 is one character. However we can easily know the maximum minimum size.

IEEE 754 numbers have 24 significant bits of binary mantissa. This means that the mantissa of a floating point numbers can have at most $\text{ceiling}(\log_{10}(2^{24})) = 8$ decimal digits. Add one for the sign, one for the decimal point, one for the E and three more for the exponent value (which, being of only 7-bits for a base-two exponent (sign excluded) has a maximum value of ± 38 base-10