Programação em C

Representação real em C

Agostinho Brito

2021

Representação real em C

- A linguagem C suporta dois tipos principais de números reais: `float` e `double`.
- Cada tipo comporta uma determinada precisão e faixa de valores conforme o padrão IEEE 754 - IEEE Standard for Floating-Point Arithmetic.
- Esse padrão, criado em 1985, define muitas coisas envolvendo números reais, incluindo operações aritméticas entre estes. Processadores modernos implementam o padrão, de modo a fornecer resultados tratáveis e intercambiáveis com as linguagens de programação atuais.
- Destaque na forma como os números são representados.

Representação no IEEE 754

- A norma IEEE 754 define duas formas principais para representação de números reais ou, números em Ponto Flutuante: precisão simples e precisão dupla.
- A primeira prevê um número real com 32 bits; a segunda com 64 bits.

Tipo	Num. Bits	Intervalo	
float	32	1.1755e-38	3.4028e+38
double	64	2.2251e-308	1.7977e+308

- Há também uma representação com precisão extendida com 80 bits, mas não é mandatória: `long double`.
- Como é feita a conversão decimal para binário (e vice-versa) de um número em ponto flutuante?

$$(24.37)_{10} = 2 \times 10^1 + 4 \times 10^0 + 3 \times 10^{-1} + 7 \times 10^{-2}$$

Convertendo para a base 2

- A conversão para binário é feita multiplicando sucessivamente a parte fracionária do último resultado até que este seja igual a ZERO.

```
0.4375 x 2 = 0.8750
0.8750 x 2 = 1.7500
0.7500 x 2 = 1.5000
0.5000 x 2 = 1.0000
```

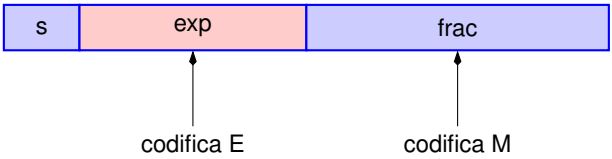
- Logo, $(0.4375)_{10} = (0.0111)_2$
- Outros números reais seguem a mesma ideia, separando a parte inteira da parte real e procedendo com as devidas regras de conversão. Ex: $(16.4375)_{10} = (1000.0111)_2$.

Representação real em IEEE 754

- O padrão IEEE 754 estabelece uma forma numérica para a representação do número

$$(-1)^s M 2^E$$

- O bit de sinal (s) determina se o número é positivo ou negativo.
- A mantissa (M) é um valor existente no intervalo $[1.0, 2.0)$, seguindo uma representação normalizada.
- O expoente (E) define a potência de 2.
- A codificação de bits é organizada em sequência



Representação real em IEEE 754

Numeros de bits por tipos de dado (precisão)

`float` s = 1 bit; exp = 8 bits; frac = 23 bits; precisão de 6 dígitos.

`double` s = 1 bit; exp = 11 bits; frac = 52 bits; precisão de 15 dígitos.

`extendido` s = 1 bit; exp = 15 bits; frac = 63 bits; 1 bit é desperdiçado. Em C (Intel), equivale ao tipo `long double`. precisão de 18 dígitos.

- As representações podem ser de-normalizada, para números com módulo na faixa $[0, 1)$, ou normalizada, para números fora dessa faixa.
- Para os números de-normalizados, $exp = 0$, $E = 1 - Bias$ e $M = frac$. $Bias$ é 127 para `floats` e 1023 para `double`.
- Para os números normalizados, $exp \neq 0$, $M = 1.0 + frac$.
- Maiores detalhes podem ser encontrados na própria norma...
- `https://www.h-schmidt.net/FloatConverter/IEEE754.html`

Representação real em C

- Por que o programador deve ficar atento à forma como os números reais são representados?
  Porque erros inocentes podem levar a falhas catastróficas !
- Exemplo: O desastre de software dos misseis Patriot (25/02/1991). 28 americanos foram mortos por uma falha na bateria anti-aérea.
- Uma falha de representação numérica causou um erro de arredondamento que fez a bateria entender que o míssil atacante (scud) estava em outro local.
- Um programador desavisado usou um incremento de 0.1s no relógio do sistema com precisão de 24 bits...
- Para 100 horas de operação, o erro era de $0.000000095 \times 100 \times 60 \times 60 \times 10 = 0.34s$!

Praticando representação real...

Obrigado