

Integer Addition in C

phunc20

Ngày 10 tháng 1 năm 2021

Trong bài này mình sẽ nói về cách cộng/trừ (+/-) `int`'s trong ngôn ngữ C. Cái mà được trình bày ở đây chỉ là những gì mình hình dung được, có thể có sai; mình cũng đang trên đường học tập; nếu bạn đọc nào tìm ra chỗ nào bị sai, có thể liên lạc với mình thông qua email.

1 Cách biểu thị `int`

Để dễ cho việc thảo luận, mình sẽ coi `int` như luôn được lưu và xử lý trong máy tính bằng 32 bits. Hơn nữa, binary và decimal integers sẽ được biểu thị như,

e.g. số bảy, $(111)_2 = \overbrace{(0 \cdots 0111)}^{32 \text{ bits}}_2$ và 7, resp.

Cho rằng chúng ta là những người đầu tiên chuẩn bị đặt ra tiêu chuẩn cho `int`. Một cách tự nhiên, mình sẽ có xu hướng định nghĩa binary như sau:

$\overbrace{(0 \cdots 0)}^{32 \text{ bits}}_2 = 0$, $\overbrace{(0 \cdots 01)}^{32 \text{ bits}}_2 = 1$, $\overbrace{(0 \cdots 010)}^{32 \text{ bits}}_2 = 2$, $\overbrace{(0 \cdots 011)}^{32 \text{ bits}}_2 = 3$, ... cho đến

$\overbrace{(01 \cdots 1)}^{32 \text{ bits}}_2 = 2^{31} - 1$. Mình tạm dừng ở đây là tại vì (i) đến đây mình đã cho hết một nửa của tập con số 32-bit; (ii) mình còn chưa assign số âm nào hết.

1.1 Đặt tầm bậy

Các con số 32-bit còn lại, thật ra mình cũng có quyền chế tùy ý. Ví dụ như,

$$-1 = \overbrace{(10 \cdots 01)}^{32 \text{ bits}}_2, -2 = \overbrace{(10 \cdots 010)}^{32 \text{ bits}}_2, \dots, -(2^{31} - 1) = \overbrace{(1 \cdots 1)}^{32 \text{ bits}}_2.$$

Và cái con số 32-bit cuối cùng sao cũng được:

$$\overbrace{(10 \cdots 0)}^{32 \text{ bits}}_2 = 2^{31} \text{ hoặc } \overbrace{(10 \cdots 0)}^{32 \text{ bits}}_2 = -2^{31}.$$

1.2 Đặt đàng hoàng

Thật ra không có luật nào bắt chúng ta phải đặt tiêu chuẩn cho `int` theo đàng gì cả. Điều quan trọng ở đây là *một tiêu chuẩn được đặt ra và sử dụng rộng rãi vì sự thuận tiện nó mang lại*.

Cái tiêu chuẩn mà mình chế đại ở section trước, nó **tiện ở chỗ** là dễ nhận ra con số đó là bao nhiêu (tại vì số dương và số âm chẳng khác gì ngoài bit đầu tiên). Nhưng mà nó lại **bất tiện ở chỗ**, chẳng hạn như $(-1) + 1$ sẽ là

$$\overbrace{(10 \cdots 01)}^{32 \text{ bits}}_2 + \overbrace{(0 \cdots 01)}^{32 \text{ bits}}_2.$$

Rất khó để giải thích tại sao cái này tính ra $\overbrace{(0 \cdots 0)}^{32 \text{ bits}}_2$.

Cái tiêu chuẩn những người đi trước chúng ta chế ra, đó là

$$\begin{aligned} \overbrace{(10 \cdots 0)}^{32 \text{ bits}}_2 &= -2^{31}, \quad \overbrace{(10 \cdots 01)}^{32 \text{ bits}}_2 = -2^{31} + 1, \quad \overbrace{(10 \cdots 010)}^{32 \text{ bits}}_2 = -2^{31} + 2, \quad \dots, \\ \overbrace{(1 \cdots 101)}^{32 \text{ bits}}_2 &= -3, \quad \overbrace{(1 \cdots 10)}^{32 \text{ bits}}_2 = -2, \quad \overbrace{(1 \cdots 1)}^{32 \text{ bits}}_2 = -1. \end{aligned}$$

Theo mình cách chế này **được cái tiện ở chỗ** cách cộng được tự nhiên và như *group* trong toán học. Ví dụ:

- $\overbrace{(10 \cdots 0001)}^{32 \text{ bits}}_2 + \overbrace{(00 \cdots 0011)}^{32 \text{ bits}}_2 = \overbrace{(10 \cdots 0100)}^{32 \text{ bits}}_2 = -2^{31} + 4$ on one side, and $(-2^{31} + 1) + 3 = -2^{31} + 4$ on the other;
- $\overbrace{(1 \cdots 1)}^{32 \text{ bits}}_2 + \overbrace{(0 \cdots 01)}^{32 \text{ bits}}_2 = \overbrace{(10 \cdots 0)}^{33 \text{ bits}}_2 = \overbrace{(0 \cdots 0)}^{32 \text{ bits}}_2 = 0$ ở một phía, và $(-1) + 1 = 0$ ở phía còn lại.

Cái cách cộng hai `int`'s trong hệ thống này giống Hình 1 ở dưới, khi tính $\mathbf{a} + \mathbf{b}$ mình có thể đi từ \mathbf{a} theo hướng kim đồng hồ số bước từ **số không** tới \mathbf{b} ; nếu cách cộng đó xoay hơn một vòng hình tròn thì mình tính lại từ đầu (i.e. từ **số không**). Các bạn đọc có thể lấy ba ví dụ hình dung trong đầu và verify có thật: (i) cộng hai con số dương; (ii) một con số dương và một con số âm; (iii) hai con số âm.

Một hệ quả của cách chế này là

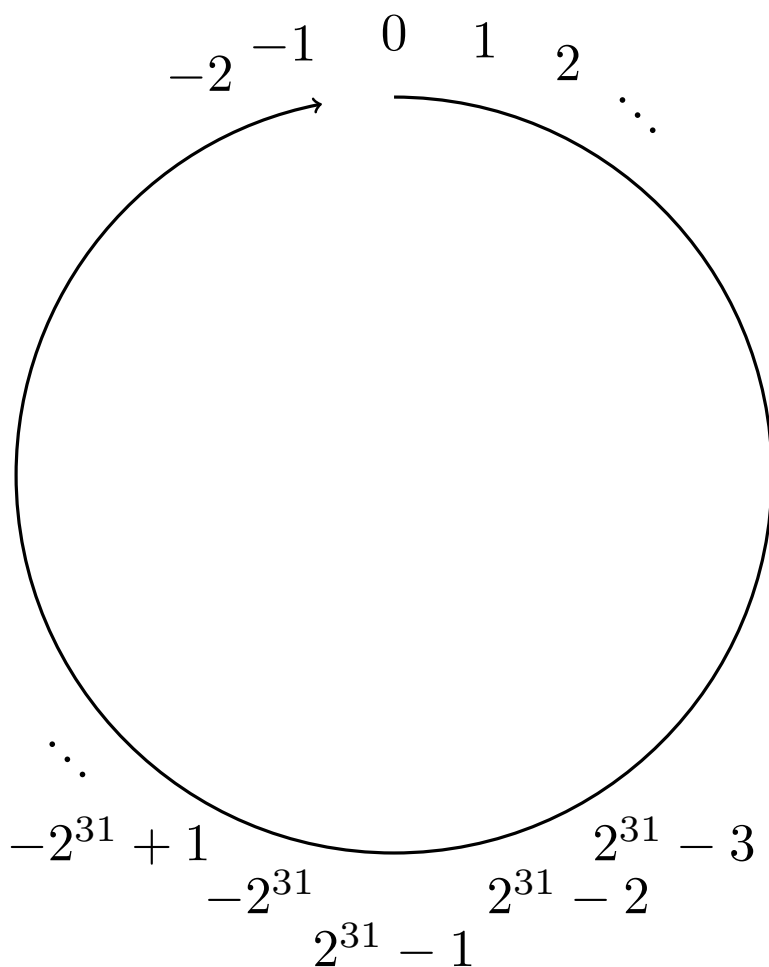
- (bit bên tay trái nhất = 0) \implies số không âm (i.e. số dương hoặc số 0) ;
- (bit bên tay trái nhất = 1) \implies số âm.

Due to the cyclic nature of the addition we just defined, khi mình có một `int` \mathbf{a} và mình muốn tìm $-\mathbf{a}$, mình sẽ tiến hành như sau:

Nhắc lại

$$\overbrace{(10 \cdots 0)}^{33 \text{ bits}}_2 = \overbrace{(0 \cdots 0)}^{32 \text{ bits}}_2 = 0.$$

Để tìm được \mathbf{b} sao cho \mathbf{a} đi thêm \mathbf{b} bước theo chiều kim đồng hồ tới $\overbrace{(10 \cdots 0)}^{32 \text{ bits}}_2 = 0$, mình có thể



Hình 1: Để cộng hai `int`'s trong C, chúng ta có thể coi như đi bao nhiêu bước trên hình tròn như trong hình.

- tìm trước c sao cho $a + c = \overbrace{(1 \dots 1)}^{32 \text{ bits}}_2 = -1$. Việc này có thể đơn giản hoàn tất bằng cách đổi ngược từng bit trong a
- rồi $b = c + 1$

Bài này đến đây đã hơi hơi bị dài, nếu các bạn đọc cảm thấy hứng thú, sau đây là một vài hướng các bạn có thể tự suy nghĩ tiếp:

1. chứng minh cách cộng này là *commutative* và *associative*

2. $a - b := a + (-b)$ theo đúng sense mình hay nghĩ về cách trừ giữa `int`'s.
(Hint: It might be easier to think of this geometrically: Rotating the angle of $-b$ **clockwise** is equiv. to rotating **counter-clockwise** the angle of b)