

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 sót; mình cũng đang trên đường học tập; nếu bạn đọc nào tìm ra những chỗ bị sai, có thể liên lạc với mình bằng 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, $(111)_2 = (0 \cdots 0111)_2$ và 7, resp.

Thử giả bộ chúng ta là những người đầu tiên chuẩn bị chế ra tiêu chuẩn cho `int`. Một cách tự nhiên, mình quy định $(0 \cdots 0)_2 = 0$, $(0 \cdots 01)_2 = 1$, $(0 \cdots 010)_2 = 2$, $(0 \cdots 011)_2 = 3$, etc. Cái này cho đến khi $(01 \cdots 1)_2 = 2^{31} - 1$. Mình tạm dừng ở đây là tại vì (i) đến đây mình đã cho hết một nửa các con số 32-bit; (ii) mình còn chưa assign số âm nào hết.

1.1 Chế tạm bợ

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

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

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

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

1.2 Chế đẳng hoàng

Thật ra, không có ai bắt con người mình chế tiêu chuẩn cho `int` nhất quyết phải như thế nào cả. Điều quan trọng ở đây là *một tiêu chuẩn được chế 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

$$(10 \cdots 01)_2 + (0 \cdots 01)_2.$$

Rất khó để giải thích tại sao cái này tính ra $(0 \cdots 0)_2$.

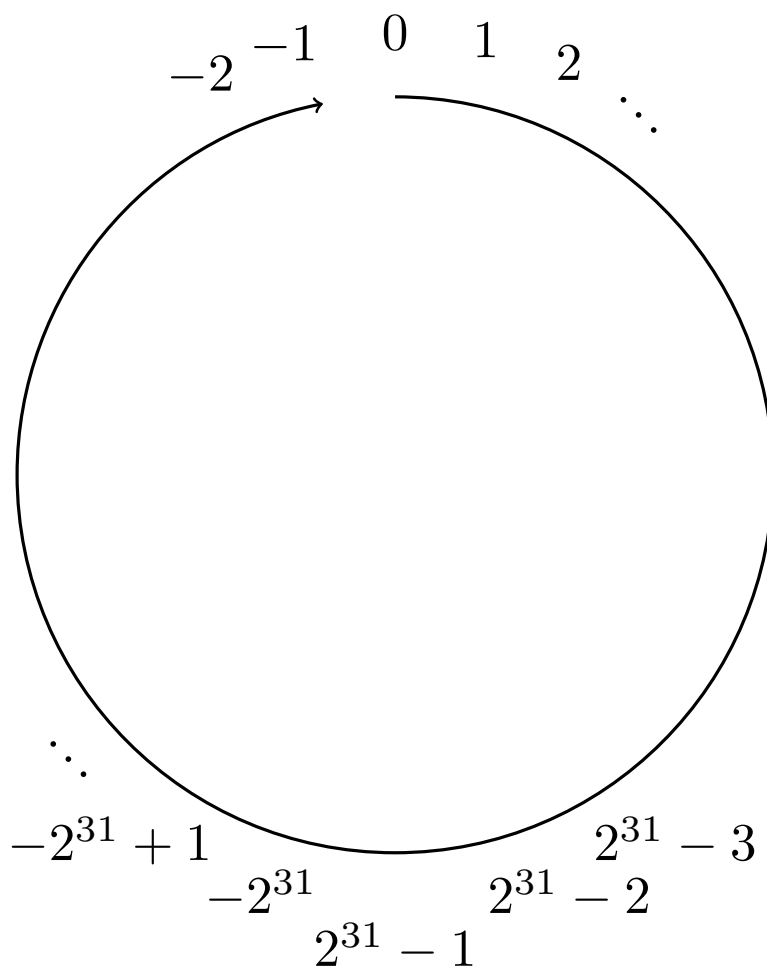
Cái tiêu chuẩn những người đi trước mình chế ra, đó là

$$(10 \cdots 0)_2 = -2^{31}, (10 \cdots 01)_2 = -2^{31} + 1, (10 \cdots 010)_2 = -2^{31} + 2, \dots, \\ (1 \cdots 101)_2 = -3, (1 \cdots 10)_2 = -2, (1 \cdots 1)_2 = -1.$$

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

- $(10 \cdots 0001)_2 + (00 \cdots 0011)_2 = (10 \cdots 0100)_2 = -2^{31} + 4$ on one side, and $(-2^{31} + 1) + 3 = -2^{31} + 4$ on the other

- $(1 \cdots 1)_2 + (0 \cdots 01)_2 = (10 \cdots 0)_2$ (i.e. một cái 1 và ba mươi hai cái 0) = $(0 \cdots 0)_2$ (i.e. 32 cái 0) ở một phía, và $(-2^{31} + 1) + 1 = 0$ ở phía còn lại



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