Architektury systemów komputerowych

Wykład 3: Bity, bajty i liczby całkowite

Krystian Bacławski

Instytut Informatyki
Uniwersytet Wrocławski

21 marca 2022

Konwersja reprezentacji liczb całkowitych

Konwersja do postaci binarnej

Niech w to szerokość słowa w bitach, a x_i oznacza i-ty bit liczby x.

$$B2U_w(x) = \sum_{i=0}^{w-1} x_i \cdot 2^i$$

$$B2T_w(x) = -x_{w-1} \cdot 2^{w-1} + \sum_{i=0}^{w-2} x_i \cdot 2^i$$

Mnożenie na liczbach ze znakiem

Konwersja między liczbami ze znakiem i bez znaku

$$T2U_w(x) = x_{w-1} \cdot 2^w + x = \begin{cases} x + 2^w & \text{dla } x < 0 \\ x & \text{dla } x \geq 0 \end{cases}$$

$$U2T_w(u) = -u_{w-1} \cdot 2^w + u = \begin{cases} u & \text{dla } u < 2^{w-1} \\ u - 2^w & \text{dla } x \geq 2^w \end{cases}$$

$$T2U_{16}(-1) = 2^{16} - 1 = 65535$$

$$U2T_{16}(65535) = -1 \cdot 2^{16} + 65535 = -1$$

Co się dzieje z mnożeniem ze znakiem?

$$\begin{aligned} x *^t_w y &= U2T_w(T2U_w(x) \cdot T2U_w(y) \bmod 2^w) \\ &= U2T_w([(x + x_{w-1} \cdot 2^w) \cdot (y + y_{w-1} \cdot 2^w)] \bmod 2^w) \\ &= U2T_w([x \cdot y + (x_{w-1} \cdot y + y_{w-1} \cdot x) \cdot 2^w + x_{w-1} \cdot y_{w-1} \cdot 2^{2w}] \bmod 2^w) \\ &= U2T_w((x \cdot y) \bmod 2^w) \end{aligned}$$

Dzielenie całkowitoliczbowe

Oczekujemy, że dzielenie całkowitoliczbowe w języku `C` działa zgodnie z definicją:

$$
\begin{aligned}
x \div y &= \lfloor x/y \rfloor \\
x \,\%\, y &= x - \lfloor x/y \rfloor \cdot y
\end{aligned}
$$

Prosty program w języku `C` jest w stanie pokazać, że jest inaczej:

```
$ quorem 120 17
120 / 17 = 7        # ok
120 % 17 = 1
$ quorem 120 -17
120 / -17 = -7      # floor(-7.05...) = -8
120 % -17 = 1       # 120 - (-8 * -17) = -16
$ quorem -120 17
-120 / 17 = -7      # floor(-7.05...) = -8
-120 % 17 = -1      # -120 - (-8 * 17) = 16
$ quorem -120 -17
-120 / -17 = 7      # ok
-120 % -17 = -1     # -120 - (7 * -17) = -1
```

Dzielenie całkowitoliczbe realizowane przez procesor:

$$x \div y = \begin{cases} \lfloor x/y \rfloor & \text{dla } x \cdot y \geq 0 \wedge y \neq 0 \\ \lceil x/y \rceil & \text{dla } x \cdot y < 0 \wedge y \neq 0 \\ \perp & \text{dla } y = 0 \end{cases}$$

$$x \,\%\, y = \begin{cases} x - \lfloor x/y \rfloor \cdot y & \text{dla } x \cdot y \geq 0 \wedge y \neq 0 \\ x - \lceil x/y \rceil \cdot y & \text{dla } x \cdot y < 0 \wedge y \neq 0 \\ \perp & \text{dla } y = 0 \end{cases}$$

Zachowania

Lista zachowań

1. implementation-defined behaviour: np. rozmiar `long`
2. unspecified behaviour: np. kolejność wyliczania argumentów
3. undefined behaviour: niezdefiniowany przez specyfikację języka rezultat wykonania programu

Inne niezdefiniowane zachowania

1. Signed integer overflow
2. Reading an uninitialized local variable
3. Dereferencing a null pointer
4. Reading/writing an index past the end of an array
5. Computing an out-of-bounds pointer
6. Comparing pointers from unrelated objects
7. Oversized shift amounts

Więcej na ten temat w C++ Reference – Undefined behavior i innych licznych opracowaniach.