WI

# Metody kompilacji

## Wykład 7, Liczby zmiennoprzecinkowe i ich reprezentacja

Włodzimierz Bielecki, Piotr Błaszyński

Wydział Informatyki Zachodniopomorskiego Uniwersytetu Technologicznego

14 kwietnia 2022

# Reprezentacja liczb zmiennoprzecinkowych

Metody kompilacji

Reprezentacja

Przykłady

Generowanie kodu

**Liczby zmiennoprzecinkowe** są reprezentowane zgodnie ze standardem IEEE 754. Zgodnie ze wzorem:
$ValFloat = (-1)^z * m * 2^c$, gdzie:

- $z$ - bit znaku
- $m$ - mantysa
- $c$ - cechy

# Reprezentacja liczb zmiennoprzecinkowych

Metody kompilacji

Reprezentacja

Przykłady

Generowanie kodu

| Liczby pojedynczej precyzji | Liczby podwójnej precyzji |
|---|---|
| 32 bity | 64 bity |
| 1 bit znaku $b_{31}$ | 1 bit znaku $b_{63}$ |
| 8 bitów cechy $b_{30} - b_{23}$ | 11 bitów cechy $b_{62} - b_{52}$ |
| wartości -127 - +128 | wartości -1023 - +1024 |
| 23 bit mantysy $b_{22} - b_0$ | 52 bity mantysy $b_{51} - b_0$ |

Jeżeli przyjmiemy, że mantysa zaczyna się od jedynki (bo wartości mieszczą się w przedziale 1-2), to zyskujemy 1 bit mantysy.

# Przykłady

Metody kompilacji

Reprezentacja

Przykłady

Generowanie kodu

$MARS \Longrightarrow Tools \Longrightarrow FloatingPointRepresentation$

- wartość - znak - cecha (wartość) - mantysa (wartość)
- 1.0 - 0 - 01111111 ($1 - 2^0$) - 00000000000000000000000 (1.0)
- 2.0 - 0 - 10000000 ($2 - 2^1$) - 00000000000000000000000 (1.0)
- 3.14 - 0 - 10000000 ($2 - 2^1$) - 10010001111010111000011 (1.57)

# Generowanie kodu – praktyczne podejścia

Metody kompilacji

Reprezentacja

Przykłady

Generowanie kodu

Można:

- Generować zapis binarny samodzielnie (kod łatwo przygotować przy pomocy unii w C/C++).
- Posługiwać sie liczbami w zapisie z kropką dzisiętną i przenieść pracę na narzędzie, które już to ma zrobione.
- Hybryda.

# Generowanie kodu – praktyczne podejścia

Metody kompilacji
Reprezentacja
Przykłady
Generowanie kodu

Kod wygenerowany dla przypisania $x = 3.14 * 2.0; print(x)$ :

```
.data
        x: .float 0.0
        float_tmp1: .float 3.14
        float_tmp2: .float 2.0
        float_tmp3: .float 0.0
.text
        l.s $f0, float_tmp1
        l.s $f1, float_tmp2
        mul.s $f0, $f0,$f1
        s.s $f0, float_tmp3
        l.s $f0, float_tmp3
        s.s $f0, x
        l.s $f12, x
        li $v0, 2
        syscall
```