

Strojový kód

Strojový kód je rodný jazyk procesoru (CPU). Veškerý software, ať už je napsán v C++, Pythonu nebo Javě, musí být nakonec přeložen do strojového kódu, aby mohl být spuštěn. Každá instrukce ve strojovém kódu odpovídá jedné elementární operaci, kterou procesor provede (např. sečtení dvou čísel, přesun dat v paměti nebo skok na jinou část programu).

Struktura strojového kódu

Strojový kód je pro člověka téměř nečitelný, protože se skládá z binárních čísel. Typická instrukce obsahuje dvě části:

- **Opcode (Operační kód):** Určuje, co má procesor udělat (např. „sčítání“).
- **Operandy:** Určují, s čím se má operace provést (např. čísla nebo adresy v RAM, které vrací [pointery](#)).

Příklad (Zjednodušeně)

Lidsky čitelný příkaz (C++): `x = y + 5`; Jazyk symbolických adres (Assembly): `ADD EAX, 5`
Strojový kód (Binárně): `00000101 00000101 00000000 00000000`

Vztah k architektuře (ISA)

Strojový kód není univerzální. Je závislý na **instrukční sadě** (Instruction Set Architecture - ISA) konkrétního procesoru. To je důvod, proč program zkompilovaný pro Windows na PC (architektura **x86/x64**) nespustíte přímo na mobilním telefonu (architektura **ARM**).

| Architektura | Typické zařízení | Charakteristika strojového kódu |
|---------------------|------------------------------|--|
| x86 / x86-64 | PC, Laptopy, Servery | Komplexní instrukce (CISC), historicky dominantní. |
| ARM | Mobily, tablety, Apple M1/M2 | Úsporné, jednoduché instrukce (RISC). |
| RISC-V | Mikrokontroléry, výzkum | Otevřený standard, modulární instrukce. |

Jak se strojový kód vytváří?

Existují tři hlavní cesty, jak vzniká strojový kód:

1. **Kompilace:** [[compiler|Compiler]] přeloží celý zdrojový kód najednou do spustitelného souboru (např. .exe).
2. **Interpretace:** Interpret čte kód řádek po řádku a v reálném čase posílá procesoru odpovídající instrukce.
3. **JIT (Just-In-Time):** Moderní metoda (používaná v Javě nebo JavaScriptu), kdy se kód překládá do strojového až těsně před spuštěním pro zvýšení výkonu.

Proč v něm neprogramujeme přímo?

- **Extrémní složitost:** Napsat i jednoduchý program by vyžadovalo tisíce instrukcí.
- **Náchylnost k chybám:** Jedna špatná cifra v binárním kódu může způsobit pád celého systému.
- **Nepřenositelnost:** Kód napsaný pro jeden procesor nefunguje na jiném.

Jedinou „lidsky snesitelnou“ úrovní nad strojovým kódem je **Assembler** (jazyk symbolických adres), kde jsou binární kódy nahrazeny zkratkami (např. MOV místo 10110000).

Související pojmy: CPU, RAM, Compiler, C++, Pointer, Assembler, Binární soustava.

From:

<http://serviceit.cz/> - IT ENCYKLOPEDIIE

Permanent link:

http://serviceit.cz/doku.php?id=strojovy_kod

Last update: **2025/12/31 20:55**

