

Druhá generace jazyků (2GL)

2GL (Second-Generation Programming Languages) představuje evoluční krok od čistě binárního kódování k symbolickému zápisu. Hlavním představitelem této generace je **Assembler** (jazyk symbolických adres).

1. Charakteristika 2GL

Zatímco první generace (1GL) vyžadovala psaní instrukcí přímo v binární soustavě (nuly a jedničky), 2GL zavedla tzv. **mnemotechnické pomůcky** (mnemonics). Jsou to krátká klíčová slova, která zastupují konkrétní instrukce procesoru.

- **Závislost na hardwaru:** Jazyky 2GL jsou úzce spjaty s konkrétní architekturou procesoru (např. x86, ARM, MIPS). Program napsaný pro jeden typ procesoru nebude fungovat na jiném.
- **Přímá správa paměti:** Programátor má absolutní kontrolu nad registry procesoru a adresami v paměti RAM.
- **Nutnost překladu:** Kód napsaný v 2GL musí být přeložen do strojového kódu (1GL) pomocí nástroje zvaného **Assembler**.

2. Struktura kódu v Assembleru

Typický příkaz v 2GL se skládá z operace a operandů.

Mnemonika	Význam	Příklad
MOV	Move (přesun dat)	MOV AL, 61h (ulož hodnotu 97 do registru AL)
ADD	Add (sčítání)	ADD EAX, EBX (sečti hodnoty ve dvou registrech)
PUSH / POP	Práce se zásobníkem	PUSH EAX (vlož hodnotu na zásobník)
JMP	Jump (skok)	JMP label (přejdi na jinou část kódu)

3. Výhody a nevýhody

Výhody	Nevýhody
Extremní rychlost: Kód je stejně rychlý jako strojový kód.	Nízká čitelnost: Kód je pro člověka velmi nepřehledný.
Minimální nároky: Programy zabírají velmi málo místa v paměti.	Složitý vývoj: I jednoduché operace vyžadují desítky řádků kódu.
Přímý přístup k HW: Ideální pro ovladače a vestavěné systémy.	Nepřenositelnost: Kód je vázán na konkrétní čip.

4. Kde se 2GL používá dnes?

I když byla 2GL v běžném vývoji nahrazena jazyky vyšších úrovní (jako C, C++ nebo Python), stále má své nezastupitelné místo:

- **Firmware a BIOS:** Základní software, který startuje hardware.
- **Ovladače zařízení (Drivers):** Kde je vyžadována maximální rychlost a přímá komunikace s HW.
- **Kritické sekce operačních systémů:** Části jádra (kernelu), které musí být vysoce optimalizované.
- **Reverzní inženýrství:** Analýza malwaru a hledání bezpečnostních děr.

5. Porovnání: 1GL vs 2GL vs 3GL

- **1GL (Strojový kód):** 10110000 01100001 (naprosto nečitelné)
- **2GL (Assembler):** MOV AL, 61h (symbolické vyjádření téhož)
- **3GL (C/Java/Python):** $x = 97$; (vysoká abstrakce, lidská logika)

Související články:

- [Architektura procesorů a registry](#)
- [Základy algoritmizace](#)
- [Vývoj aplikací ve vyšších jazycích](#)

Tagy: dev programming assembler low-level 2gl history hardware

From:
<https://serviceit.cz/> - IT ENCYKLOPEDIE

Permanent link:
<https://serviceit.cz/doku.php?id=it:lang:2gl>

Last update: **2026/01/02 14:02**

