

JIT (Just-In-Time Kompilace)

JIT kompilátor sleduje, které části kódu se spouštějí nejčastěji (tzv. „hot spots“). Tyto části pak přeloží přímo do instrukcí procesoru, aby se při příštím volání nemusely znovu interpretovat.

1. Jak JIT funguje?

Proces typicky probíhá v těchto krocích:

- Zdrojový kód:** Programátor napíše kód (např. v Javě nebo C#).
- Bytecode:** Kód se nejdříve přeloží do univerzálního mezikódu (Bytecode), který je přenositelný mezi systémy.
- Interpretace:** Při spuštění začne interpret číst bytecode a vykonávat ho.
- Profilování:** JIT kompilátor sleduje, které funkce se volají opakovaně (např. v [[it_encyklopedie:iterace|cyklu]]).
- Kompilace:** Tyto "horké" části JIT bleskově přeloží do nativního strojového kódu daného procesoru a uloží je do mezipaměti.
- Zrychlení:** Při dalším průchodu se už nepoužije interpret, ale spustí se přímo zkompileovaný kód v plné rychlosti.

2. Srovnání způsobů vykonávání kódu

Metoda	Rychlost startu	Rychlost běhu	Příklad
Interpretace	Okamžitá	Pomalá	Python, Bash
AOT Kompilace	Pomalá (čeká se na build)	Maximální	C++, Rust, Go
JIT Kompilace	Rychlá	Téměř maximální	Java, C#, JS (V8)

3. Hlavní výhody JIT

- Adaptivní optimalizace:** JIT zná přesný typ procesoru, na kterém program běží, a může kód optimalizovat „na míru“ konkrétnímu hardwaru (což dopředu kompilovaný program často nemůže).
- Lepší výkon u dynamických jazyků:** Díky JIT může být JavaScript v prohlížeči až 100x rychlejší než staré skriptovací jazyky bez této technologie.
- Přenositelnost:** Program se distribuuje jako bytecode, ale běží rychlostí nativního kódu.

4. Nevýhody a úskalí

- Režie (Overhead):** Samotná kompilace za běhu spotřebovává procesorový čas a paměť RAM.
- „Warm-up“ čas:** Programu chvíli trvá, než se „zahřeje“ (než JIT zjistí, co má zkompileovat). Prvních pár sekund může být aplikace pomalejší.

5. Kde se s JIT setkáte?

- **Java (JVM):** Java Virtual Machine byla průkopníkem masového nasazení JIT.
- **.NET (CLR):** Jazyky jako C# využívají JIT k dosažení špičkového výkonu na Windows i Linuxu.
- **Webové prohlížeče:** Engine V8 (v Chromu a Node.js) používá extrémně pokročilý JIT, aby webové aplikace běžely plynule.
- **PyPy:** Alternativní interpret pro [Python](#), který díky JIT kompilaci dokáže zrychlit výpočty až několikanásobně.

Zajímavost: Existuje i koncept **Tracing JIT**, který nekompiluje celé funkce, ale sleduje konkrétní cestu (trasu) skrz kód (včetně všech větvení) a tu pak optimalizuje jako jeden lineární kus kódu.

[Zpět na Skriptovací jazyky](#)

From:

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

Permanent link:

<https://serviceit.cz/doku.php?id=jit>

Last update: **2025/12/31 17:23**

