

GCC (GNU Compiler Collection)

GCC je klíčová součást projektu [Projekt GNU](#). Původně vznikl v roce 1987 pod taktovkou Richarda Stallmana jako „GNU C Compiler“, protože v té době dokázal překládat pouze jazyk C. Dnes je to modulární systém, který podporuje desítky jazyků a procesorových architektur.

1. Podporované jazyky a platformy

Ačkoliv se nejčastěji používá pro **C** a **C++**, GCC obsahuje front-endy pro mnoho dalších jazyků:

- **C++, Objective-C, Objective-C++**
- **Fortran** (pro vědecké výpočty)
- **Ada**
- **Go**
- **D**

Z hlediska hardwaru je GCC „králem přenositelnosti“. Podporuje vše od starých mainframů po moderní mobilní procesory [ARM](#) a mikrokontroléry.

2. Jak probíhá kompilace (Fáze překladu)

Přeměna lidsky čitelného kódu na binární soubor není jeden skok, ale proces o čtyřech krocích:

1. ****Preprocessing (Preprocesor):**** Zpracuje direktivy začínající mřížkou (např. `'#include'` nebo `'#define'`). Výsledkem je jeden velký textový soubor.
2. ****Compilation (Kompilace):**** Přeloží kód v C/C++ do jazyka symbolických adres ([[Assembly Language]]).
3. ****Assembly (Sestavení):**** Assembler převede kód do strojového (binárního) kódu – vznikají tzv. ****objektové soubory**** (.o).
4. ****Linking (Sestavení/Linkování):**** Linker spojí objektové soubory a systémové knihovny (jako [[glibc]]) do jednoho spustitelného souboru.

3. Klíčové parametry a přepínače

Práce s GCC v [Bashi](#) obvykle probíhá přes příkazovou řádku. Zde jsou ty nejdůležitější parametry:

Přepínač	Význam
-o <jmeno>	Určuje název výsledného spustitelného souboru.
-Wall	Zapne „všechna“ varování (velmi doporučeno pro čistý kód).
-O2 / -O3	Úroveň optimalizace. Vyšší číslo znamená rychlejší program, ale delší kompilaci.
-g	Přidá do souboru ladící informace (potřebné pro program gdb).
-std=	Určuje verzi standardu (např. <code>-std=c++20</code>).

Příklad příkazu:

```
gcc -O2 -Wall muj_program.c -o spustitelny_soubor
```

4. Architektura GCC

GCC je rozděleno na tři části, což umožňuje snadné přidávání nových jazyků nebo procesorů:

1. **Frontend:** Rozumí konkrétnímu jazyku (C, Fortran) a převede ho do univerzálního mezikódu zvaného **GIMPLE**.
2. **Middle-end:** Provádí optimalizace nezávislé na procesoru.
3. **Backend:** Generuje kód pro konkrétní procesor (x86, ARM, RISC-V).

5. GCC vs. LLVM/Clang

V posledních letech získalo GCC silnou konkurenci v podobě projektu LLVM (Clang).

Vlastnost	GCC	LLVM / Clang
—	—	—
Licence	GPL (přísně svobodná)	Apache/BSD (volnější)
Rychlost kompilace	Často pomalejší	Obvykle rychlejší
Chybové hlášky	Historicky stručné	Velmi srozumitelné a barevné
Optimalizace	Špičková, často nejlepší výsledný kód	Skvělá, ale v některých případech mírně za GCC

—

Zajímavost: GCC je „self-hosting“. To znamená, že moderní verze GCC jsou napsány v C++ a k jejich přeložení potřebujete dřívější verzi GCC. Je to klasický problém „slepice nebo vejce“, který se řeší tzv. bootstrappingem.

Související: [C++](#), [glibc \(GNU C Library\)](#), [Linux: Komplexní průvodce světem Open Source](#), [Jazyk symbolických instrukcí \(Assembly Language\)](#), [Bash \(Bourne Again SHell\)](#)

From:

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

Permanent link:

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

Last update: **2025/12/31 18:10**

