

# Debugger (Ladicí program)

**Debugger** je specializovaný softwarový nástroj určený k testování, analýze a vyhledávání chyb v počítačových programech. V kontextu programování v [jazyce Assembler](#) je naprosto nezbytný, protože umožňuje programátorovi sledovat činnost procesoru na té nejnižší úrovni, krok po kroku.

Zatímco ve vyšších jazycích (C++, Java) se chyby často projevují v logice objektů, v Assembleru může i drobná chyba (např. špatné vyvážení zásobníku) vést k okamžitému pádu celého systému.

## Hlavní funkce debuggeru

Debugger poskytuje přímý vhled do stavu počítače během vykonávání programu:

- **Krokování (Single-stepping / Tracing):** Vykonání programu po jedné jediné instrukci. Programátor po každém kroku vidí, co se v procesoru změnilo.
- **Sledování registrů:** Zobrazuje aktuální hodnoty v [registrech](#) (např. EAX, EBX, ESP, EIP). To je v Assembleru klíčové, protože registry jsou hlavními pracovními místy procesoru.
- **Prohlížení paměti (Memory Dump):** Umožňuje nahlédnout do libovolného místa v operační paměti a vidět data v hexadecimálním formátu nebo jako interpretované instrukce.
- **Body zastavení (Breakpoints):** Označení místa v kódu, kde má program automaticky pozastavit svůj běh. Programátor pak může prozkoumat stav systému těsně před kritickou operací.

## Debugger jako učební pomůcka

Kniha Rudolfa Marka zdůrazňuje, že debugger není jen nástroj na opravu chyb, ale také vynikající prostředek pro pochopení architektury PC. Pomocí něj lze vizualizovat:

1. Jak se mění **ukazatel zásobníku (ESP)** při instrukcích `PUSH`` a `POP``.
2. Jak fungují **příznaky (Flags)** po aritmetických operacích (např. příznak nuly ZF po odečtení dvou stejných čísel).
3. Jak se v paměti ukládají čísla v režimu **Little-endian**.

## Nástroje zmíněné v literatuře

V knize se pro praktické ukázky využívají především tyto nástroje:

- **GRDB (Generic Realmode DeBugger):** Jednoduchý, ale výkonný externí debugger pro 16bitový a 32bitový reálný režim (často používaný v DOSu).
- **GDB (GNU Debugger):** Standardní nástroj pro ladění v systému Linux.
- **Integrované debuggery:** Součástí moderních vývojových prostředí (IDE), které nabízejí grafické rozhraní pro snadnější orientaci v kódu.

# Základní ladicí postup

1. **Lokalizace:** Pomocí breakpointů se určí oblast kódu, kde program vykazuje chybu. 2. **Analýza:** Krokováním a sledováním registrů se zjistí, která konkrétní instrukce způsobuje neočekávané chování. 3. **Oprava:** Úprava zdrojového kódu v [Assembleru](#) a opětovný překlad pomocí [NASM](#). 4. **Verifikace:** Opětovné spuštění v debuggeru pro potvrzení, že chyba byla odstraněna.

**Zajímavost:** Termín „debugging“ (odvšivení) vznikl v počátcích počítačů, kdy skutečný hmyz (můra) způsobil zkrat v relé počítače Mark II. Dnes pod tímto pojmem rozumíme výhradně odstraňování softwarových chyb.

— **Viz také:** [Assembler](#), [Registry procesoru](#), [Zásobník \(Stack\)](#), [NASM překladač](#)

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

Permanent link:  
<https://serviceit.cz/doku.php?id=debugger>

Last update: **2026/01/06 19:23**

