

- Kompilace s podporou GDB: **gcc -g -o a.out foo.c -pthread -D__REENTRANT**
- Spuštění pro ladění: **gdb a.out**
- Breakpoint – místo, kde je zastaven běh programu
 - **br main** – breakpoint na funkci main, tj. gdb zastaví na začátku programu
 - **br foo.c:10** – breakpoint v souboru foo.c na radku 10
 - **del 1** – smaže první breakpoint
 - **del** – smaže všechny breakpointy
- Běh programu: **(gdb) run [parametry]** – spustí nahraný program **a.out** s případnými parametry
 - Běh se zastaví a) na breakpointu, b) skončením programu, c) přerušením ctrl-c nebo jiným signálem
- **help příkaz** zobrazí nápovědu k zadnému příkazu

- Pohybování se mezi funkcemi (po zastavení běhu)
 - **list** – vypíše okolí místa zastavení
 - **up** – posune se do volající funkce (v sekvenci volání směrem k prvotní funkci **main()**)
 - **down** – posune se do volané funkce (směrem od prvotní funkce **main()**)
 - **where** – vypíše sekvenci volání od **main()** po aktuální funkci, kde byl přerušen běh
- Sledování proměnných
 - **print [promenná]** vypíše jednorázově obsah proměnné, lze používat přetypování, dereference, např. **print (struct *foo) x->next.**
 - Je-li použita volba **-O2** při překladač, nemusí některé proměnné existovat – optimalizace je nahradily. Pro přístupné proměnné je nutné použít **-O0**.
 - **display [promenná]** bude vypisovat obsah proměnné po každém příkazu pro gdb.

- Příkazy pro trasování programu
 - **n** [**enter**] krok dál, je-li funkce, provede se a krok skončí za ní
 - **s** [**enter**] krok dál, je-li funkce, vstoupí se do ní
 - Příkazy **n** a **s** není nutné stále psát, [**enter**] automaticky opakuje poslední příkaz