# 11. Tétel

Ismertesse a C/C++ Linux alkalmazások hibakeresésének főbb eszközeit

## gdb

Legfőbb hibakereső Linuxra (C-re, más nyelveket csak részlegesen támogat)

### Funkciói:

* Program elindítása
* megállítás meghatározott feltételek esetén
* a megállításkori állapot vizsgálata
* a program egyes részeinek megváltoztatása, és a változtatás hatásának vizsgálata

### Működése

* program fordítása –g kapcsolóval
* indítás
  + gdb program
  + argumentumok
    - CORE
    - process id
* set width=70 : program kimenetének szélessége

### Töréspontok

* ***egyszerű töréspont***
  + break függvény: egy adott fv. belépési pontjánál
  + +- OFFSET: az aktuális pozíciótól számított sorokkal odébb
  + fájlnév:sorszám – adott fájl adott sorába
  + \*CÍM – egy adott címen
  + argumentum nélkül – az aktuális stack fram következő utasítására
  + … if COND – feltételhez kötött breakpoint
* ***watchpoint***
  + speciális töréspont, ami akkor állítja meg a programot, amikor az adott kifejezés változik (nem kell helyet megadni)
  + watch KIF – a program megáll, ha KIF-t írja valaki (módosul)
  + rwatch KIF – a program megáll, ha KIF-t olvassa valaki
  + awatch KIF – a program megáll, ha KIF-t olvassák/írják
* ***catch ESEMÉNY***
  + megállás esemény bekövetkezése esetén
  + az esemény lehet:
    - throw – C++ exception keletkezése
    - catch – C++ exception kezelése
    - exec – az exec fv. meghívásakor
    - fork – fork fv. meghívásakor
    - vfork – vfork meghívásakor
    - load KÖNYVTÁRNÉV – adott könyvtár betöltésekor
    - unload KÖNYVTÁRNÉV – könyvtár eltávolításakor
* run: futtatás

### Töréspont elérése utáni lehetőségek

* run (r): folyamat elindítása
* next (n): next line
* step (s): step into a function
* print (p): kiírja egy adott változó értékét
* backtree (bt): A stack keretek megjelenítése
* list (l): a forráskód kilistázása az aktuális pozíció környékén
* continue (c): a program futásának folytatása
* Ctr+D: program leállítása
* quit (q): kilépés

## ElectriFence

Memóriaszivárgás, vagy túlírás jellegű hibákra

## Valgrind

virtuális processzoron futtatja a programot + közben hibaellenőrzések

* memóriakezelési hibák felderítése
* szálkezelési hibák felderítése
* teljesítmény analízis

## Strace

lefuttatja a programot, és monitorozza a rendszerhíváokat, és jelzéseket

## lint

forráskód elemzés, szintaktikai hibák keresése, inicializálási hibák, indexelési hibák keresése

## IDEk

* Integrated Development Environment
* korábban tárgyalt fejlesztői eszközöket foglalják össze grafikus felületbe
* pl.
  + Eclipse
  + NetBeans
  + QtCreator
  + VS Code
  + …