GDB Advanced tutorial

Athanasios Nakas
Giorgos Stilianakis
Michalis Vardoulakis
GDB-1 revision

• GDB is available on the department machines

• Must use –g flag when compiling your code

• gdb <executable_file> to run with GDB
GDB-1 revision

- **run(r)**: Begin the execution of your program without stopping at main

- **break(b)**: Sets a breakpoint at a part of your code. The debugger will stop the execution of the program at every breakpoint
  - e.g. `break <function name>|<line>`

- **delete(d)**: Remove a breakpoint. Can also remove all breakpoints
GDB-1 revision

- **continue(c)** : continue execution from breakpoint
- **info breakpoints** : View all breakpoints
- **next(n)** : Execute current command and move to next
GDB-1 revision

• **print(p)**:
  • print : prints any kind of variable
  • print x : prints the value of var x
  • expressions are supported (e.g. print &x prints the address of x)

• **backtrace(bt)** : display function call stack (with function arguments)

• **list(l) <line number> | <function>** : prints code around <line number> | <function>
Examine Memory

• To examine a memory segment you can use \texttt{x[/FMT] ADDRESS}
  • \texttt{ADDRESS} is an expression for the memory address to examine
  • \texttt{FMT} is a repeat count followed by a format letter and a size letter
  • Format letters are \texttt{o(octal), x(hex), d(decimal), u(unsigned decimal), t(binary), f(float), a(address), i(instruction), c(char), s(string)} and \texttt{z(hex, zero padded on the left)}.

• example: Print contents of the memory address \texttt{i} in decimal format
  • \texttt{(gdb) x/d &i}
  • \texttt{0xbffff46c: 10}
Modify memory Content

• One can alter the memory contents using set

```
(gdb) info locals
  i = 15
  j=5

(gdb) set j = 0
(gdb) info locals
  i = 15
  j=0
```
Moving in the stack

We can also move in the stack:

- **up**: go to the upper stack frame
- **down**: go to the down stack frame
- **select-frame**: jump at a specific stack frame
- Use `bt` or `f` command to view the backtrace(`bt`) or current(`f`) stack frame
Useful links


Live Demo Now!