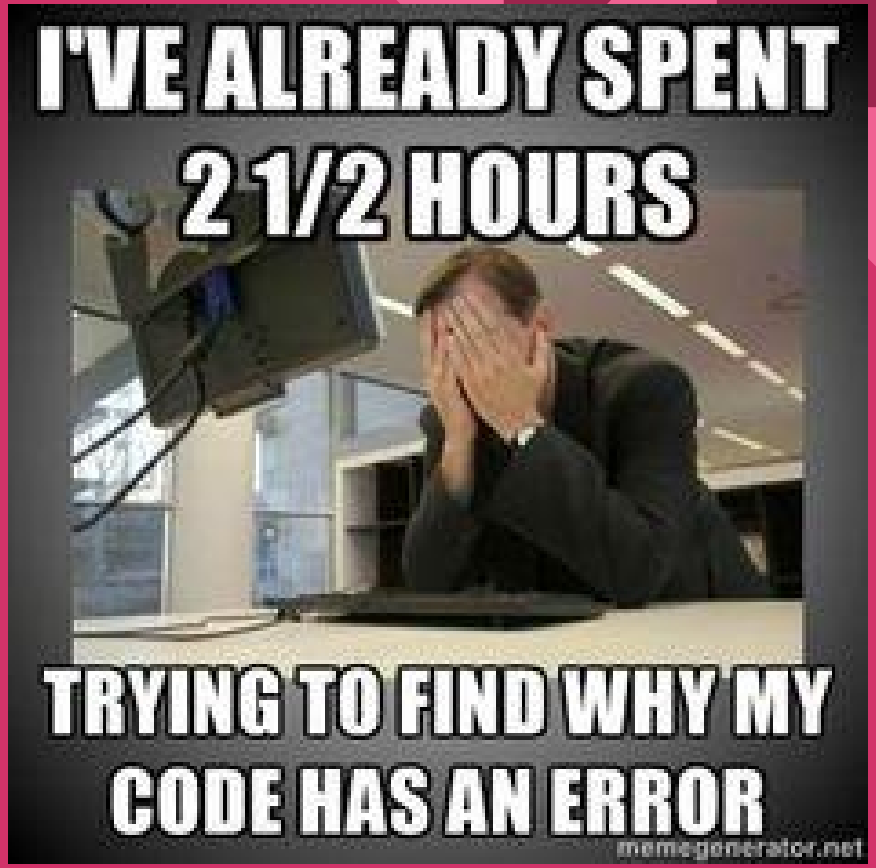
GDB - GNU Debugger

Saves time and frustration

By Akarsha Sehwag

What is gdb?

Lets you know what’s going on “inside” the program.



How is it useful?

1. Why and where did the program abort?
2. Watch or modify the variables in runtime
3. Stop the program at certain conditions
4. Change the flow of execution dynamically

Starting gdb

To Install:

- `sudo apt-get install libc6-dbg gdb valgrind`

Use ‘-g’ flag while compiling

- `gcc -g xyz.c -o xyz`
- `gdb ./xyz`

GDB Commands

Most commonly used commands !

- run
- break, watch & catch
- continue
- list
- print & display
- step & next
- backtrace
- finish & quit

Stepping through the code

To inspect how it’s being executed

- step or s
- step instruction or si
- next
- continue
- finish

Setting up breakpoints, watchpoints and catchpoints

- break 20
- break myfunc
- break *address
- break file:6 if i>=10
- Info breakpoints
- delete 3
- continue

- watch a
- watch b == 2
- info watchpoints

- catch catch
- catch syscall 100
- info catchpoints

Viewing and Setting variables

- set x=2
- call myfunc()
- display x
- undisplay x
- print x
- print mystruct->field
- print *mystruct
- printf “%d”, x

Examining the stack !

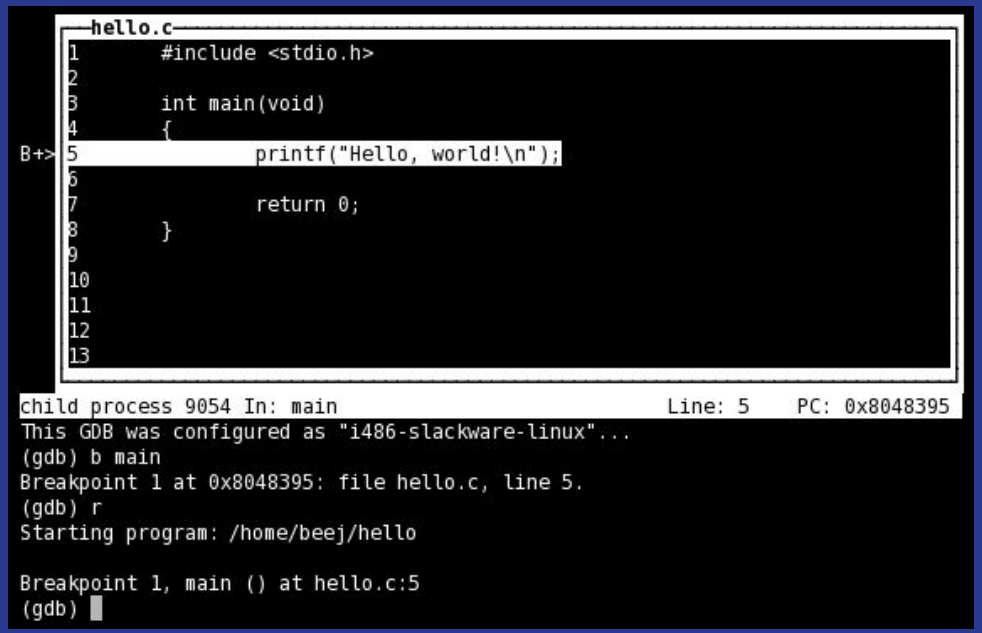
Frame: data associated with each function call

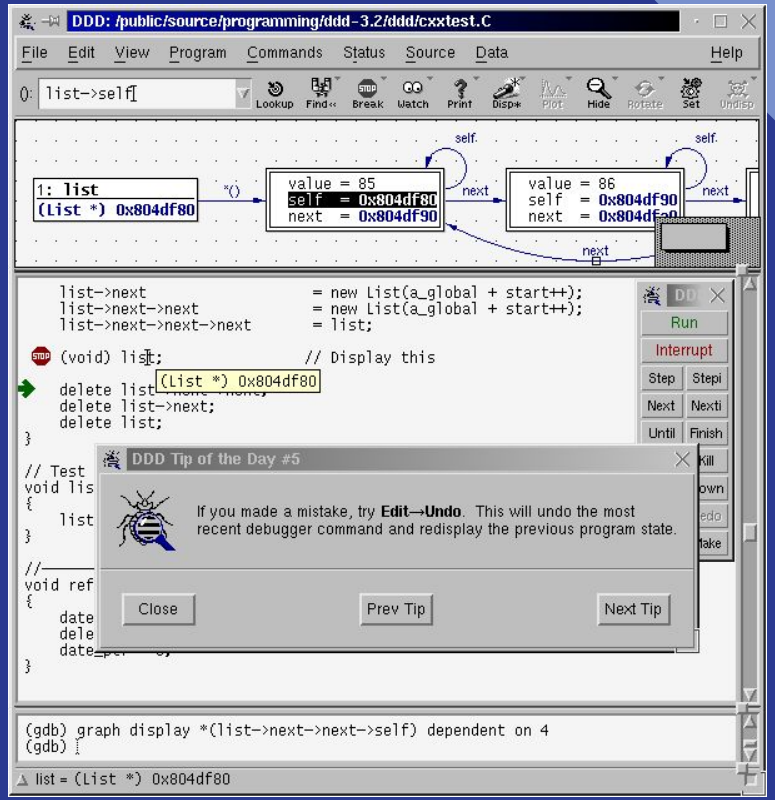
- Innermost frame
- Outermost frame
- up/down command

How we got there ?

- backtrace
- backtrace full

GUI for gdb


hello.c
1 #include <stdio.h>
2
3 int main(void)
4 {
B+> 5 printf("Hello, world!\n");
6
7 return 0;
8 }
9
10
11
12
13
child process 9054 In: main Line: 5 PC: 0x8048395
This GDB was configured as "i486-slackware-linux"...
(gdb) b main
Breakpoint 1 at 0x8048395: file hello.c, line 5.
(gdb) r
Starting program: /home/beej/hello
Breakpoint 1, main () at hello.c:5
(gdb)


DDD: /public/source/programming/ddd-3.2/ddd/cxxtest.C
File Edit View Program Commands Status Source Data Help
0: list->self
Lookup Find« Break Watch Print Disp» Plot Hide Rotate Set Undisp
1: list
(List *) 0x804df80
value = 85
self = 0x804df80
next = 0x804df90
value = 86
self = 0x804df90
next = 0x804df98
list->next = new List(a_global + start++);
list->next->next = new List(a_global + start++);
list->next->next->next = list;
(void) list; // Display this
(List *) 0x804df80
delete list;
delete list->next;
delete list;
Run
Interrupt
Step Stepi
Next Nexti
Until Finish
DDD Tip of the Day #5
If you made a mistake, try Edit→Undo. This will undo the most recent debugger command and redisplay the previous program state.
Close
Prev Tip
Next Tip
(gdb) graph display *(list->next->next->self) dependent on 4
(gdb)
list = (List *) 0x804df80

Tutorial Links

http://www.tenouk.com/Module000linuxgcc1.html

http://heather.cs.ucdavis.edu/~matloff/UnixAndC/CLanguage/Debug.html#tth_sEc3