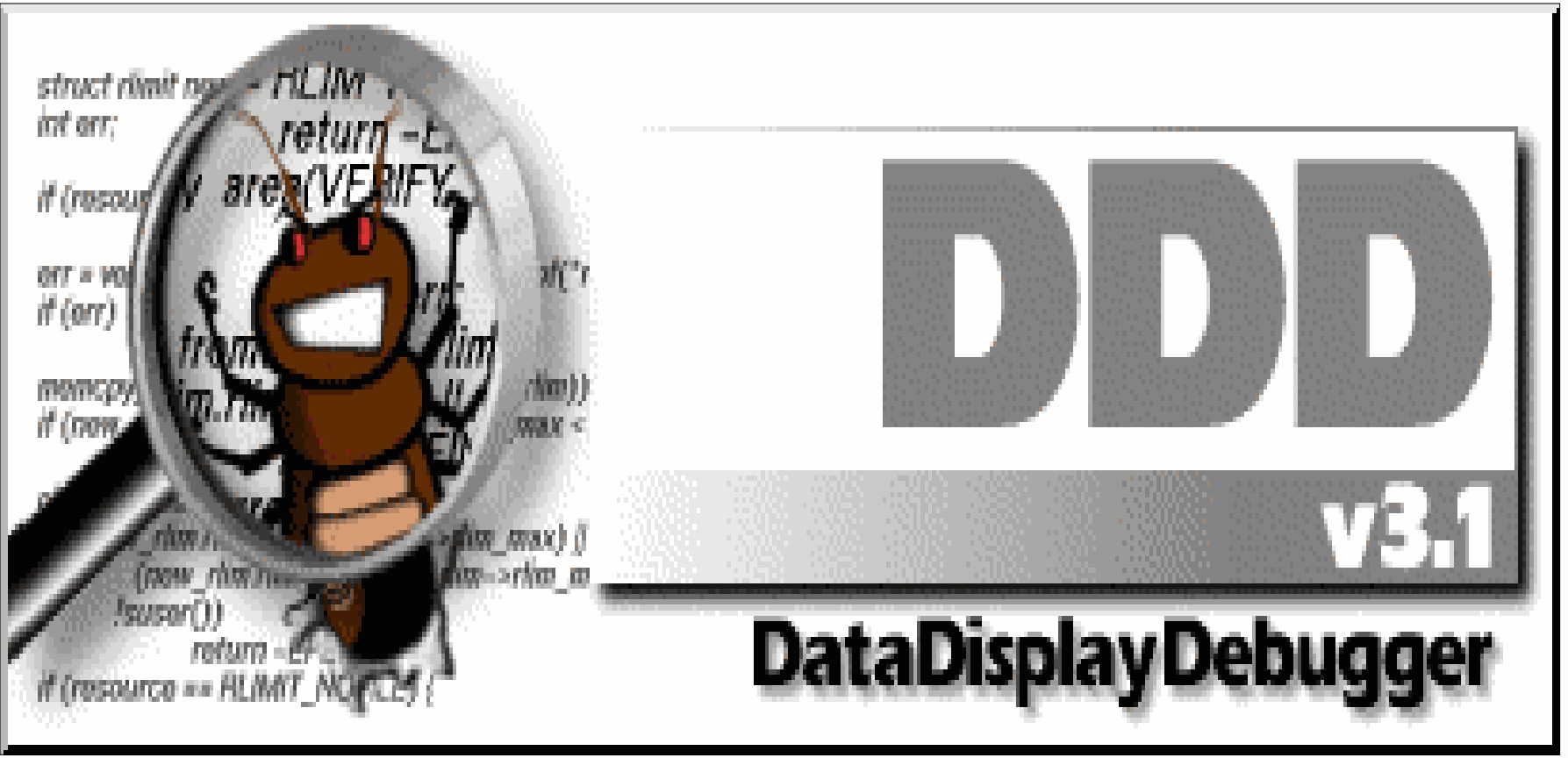
BOSTON LINUX & UNIX USER GROUP http://www.blu.org

Presents



Introduction by Massimo Morin (mmorin@schedsys.com)

September 15, 1999

Organization

- Slide base seminar (30-40 min)
  - Introduction to debugging
  - outline principal part of DDD
  - functionality
- Live usage work bench (20-30 min)
  - basic commands
  - interaction
  - usage impressions
- Q&A session (till fall asleep ;))

Agenda

- Debugging
- GDB and DDD
- Usage
- Structure
- IDE capability?
- Pros & Cons
- Conclusion

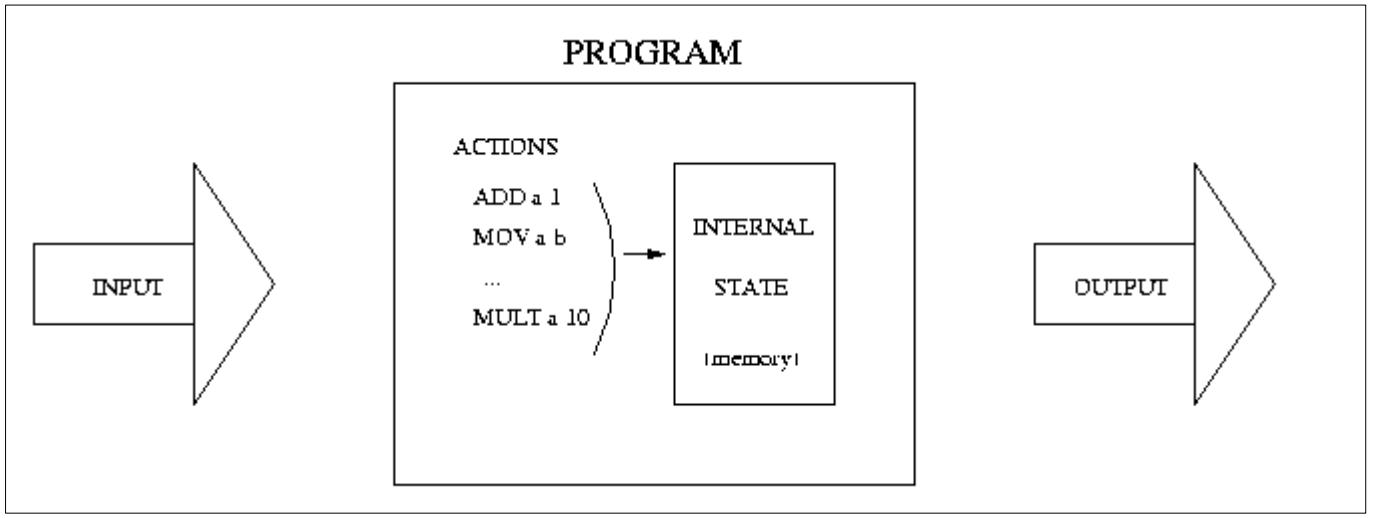
Aargh a bug!

If debugging is the process of removing bugs,
then programming must be
the process of putting them in...

[from slashdot.org]

A Program behavior

Program/process: a series of manipulation of inputs that creates and modifies the program internal state for generating some output



What is a bug?

The input could not be the proper one

The manipulation (actions) are not performed in the desired order

The actions are plain wrong

This implies

The state is unfeasible (*core dump*) or inconsistent

Results

The output is not the desired one.

Needs for a Debugger: The Debugging objectives

Modify the input

Inspect the internal state

Modify the internal state

Change instructions execution

- order
- skip, apply again different instructions

Results

Fix the output to be the desired one

What is GDB

It is a debugger (GNU Debugger):

- starts program
- conditional stopping
- examine variable contents
- change "things" on program for experimentation

Supports C, C++ and Modula-2

Line oriented program

What is DDD (and what it is not)

It is NOT a debugger

Graphical interface to a debugger

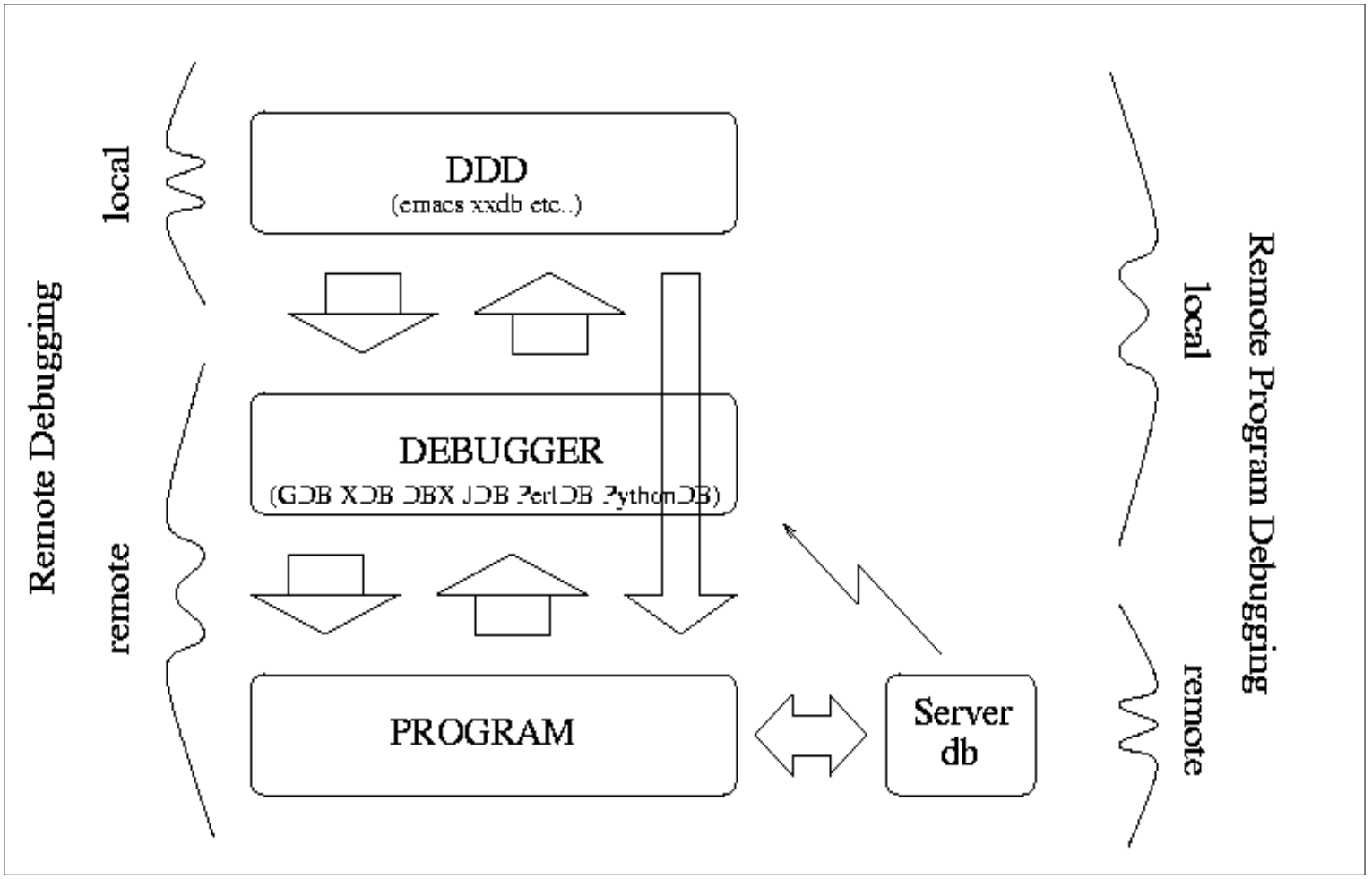
supports different debugger in back end (different languages)

Displays source code and allows manipulation of execution positions

Display data contents (state) in a graphical way

and much more...

DDD and db interrelation



Why using DDD

- We are lazy
  - Facilitate access to db commands
  - Easy access source file
  - Context sensitive help
  - highly configurable (debugger and interface)
- Constant view of the program state
- undo/redo, history command
- Session manager
- Integration with other tools

Technical Specification

Born in Germany as a Data Display

Open Source Project (GPL): current version 3.1.6

C++ and Motif program

Retrieve Source code (compiles with Motif 1.2 and LessTif 0.85) ~ 3.6Mb

Retrieve binary file:

- Static: statically linked to all library (4.1Mb)
- Semistatic: statically linked only to Motif (2.4Mb)
- Dynamic: all library dinamic (2.1Mb)

Invoking DDD

Stand alone (*ddd*)

Full debugging (*ddd program*)

Attached to a running program (*ddd program id*)

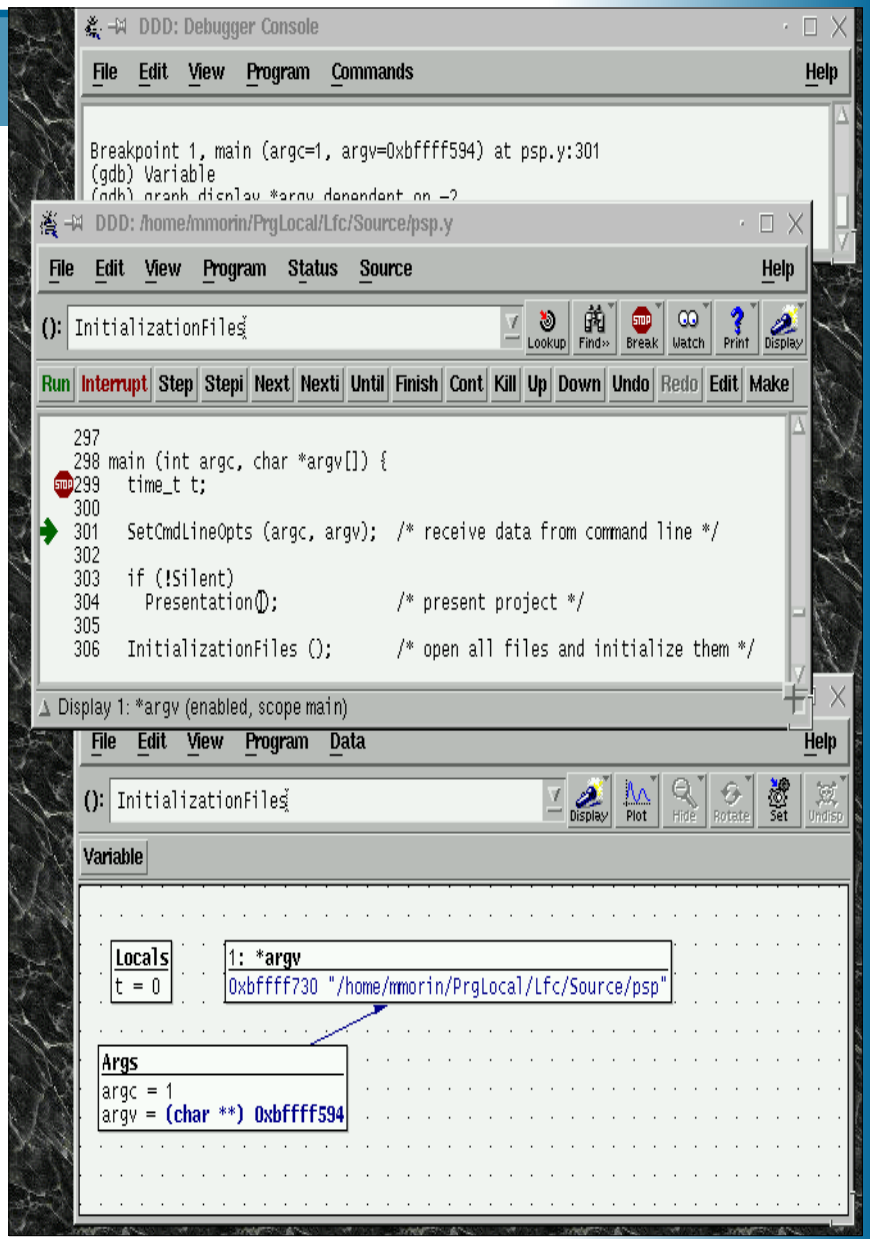
Remote debugging (*ddd --rhost*)

Integrate into emacs (*ddd --tty*)

The program to debug has to have symbolic information in it (*g++ -g program*)!

Structure

- Three windows: console, data and source window
- Additional windows: execution, command tool and machine code one
- Windows can be stacked or separated
- Menu and toolbar in every window



The Console Window

Main menu access

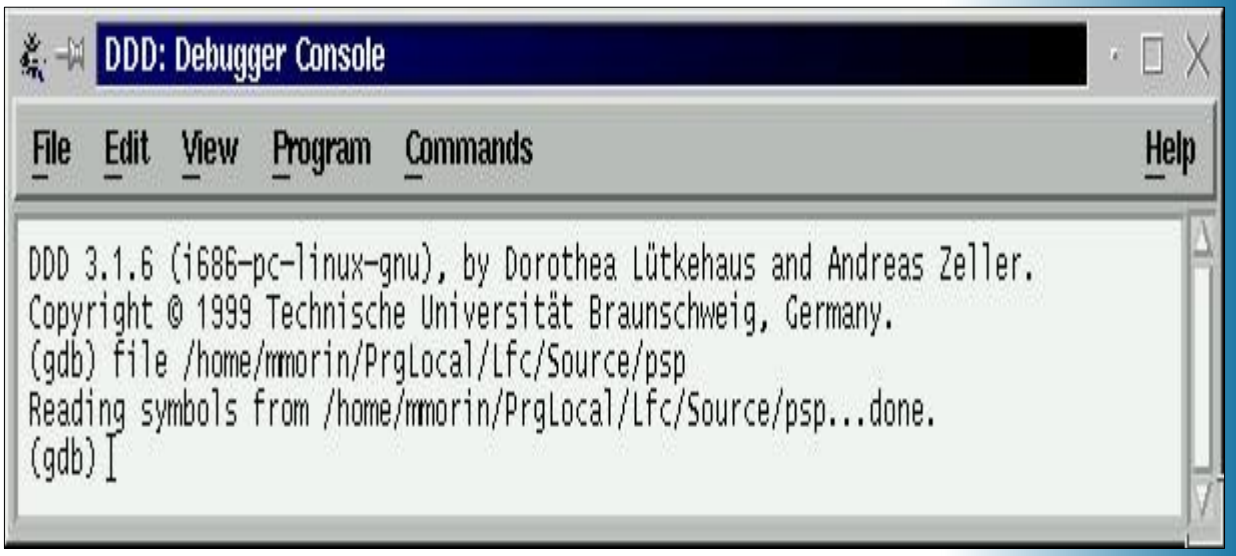
Direct interaction with the debugger

Direct command for DDD

Output of the program (if not using output win)

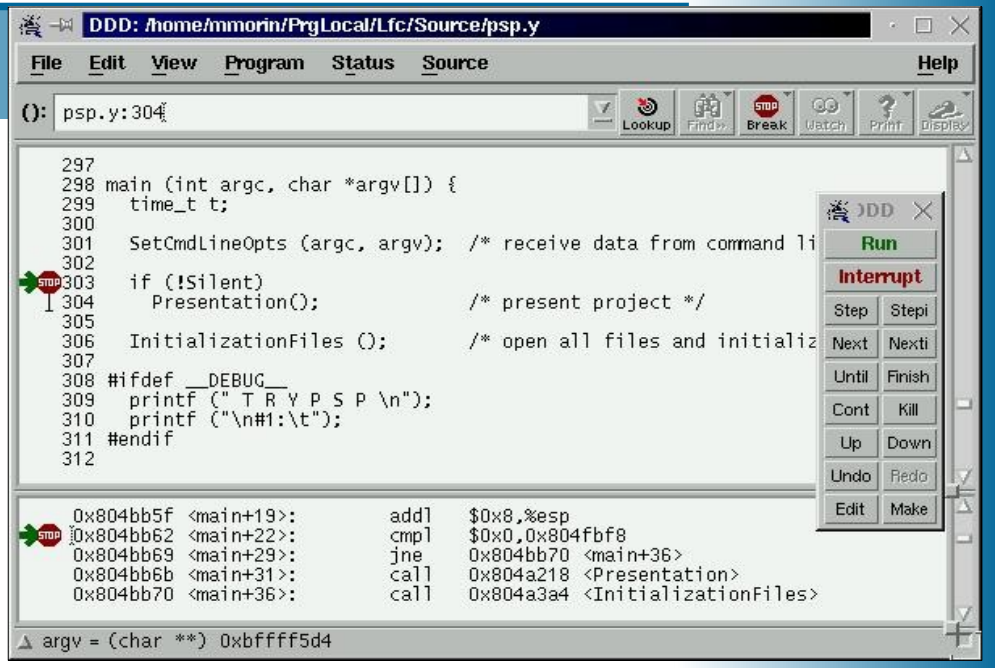
Functionality:

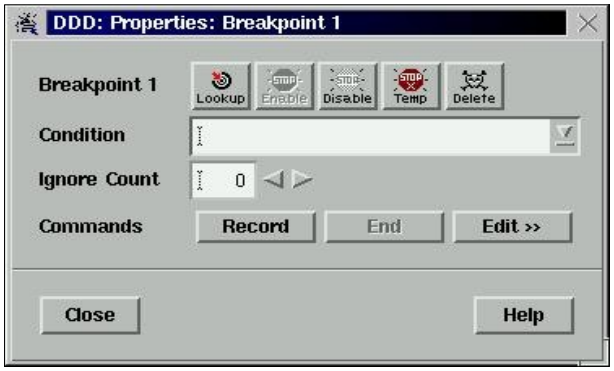
- history
- undo/redo
- tab completion

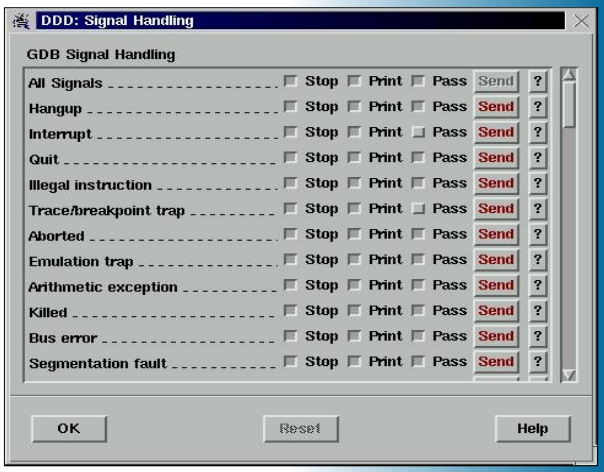


The Source Window

- Show the source code
- Command Tool
- Machine code source
- Access to
  - Breakpoints (cond, all)
  - Backtrace
  - Registers
  - Threads
  - Signals

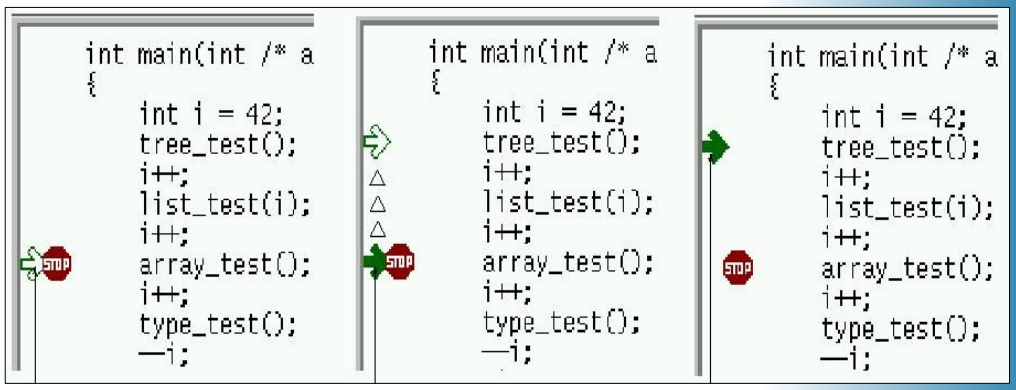


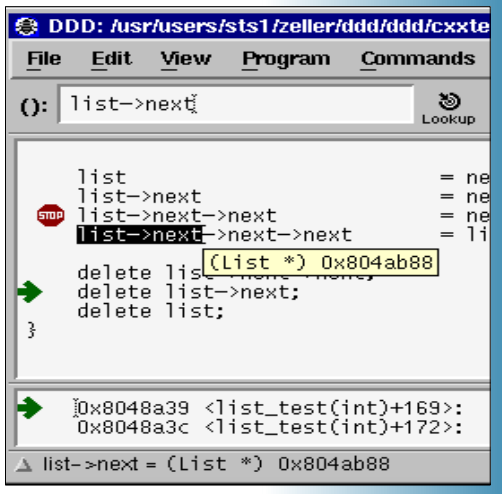




The Source Window (2)

- Execution Position Arrow functionality
- Status bar
- Status bar indicator
- Functionality
  - Variable access (print, display, lookup)
  - Value hints
  - Lookup functions
  - Search words





The Display Window

Specific variable contents

Local, global, parameters variable view

Linked list, graph, tree view

Variable clustering

Incremental clustering view

Vector view and rotation

Automatic pointer de-reference

Memory contents

The Display Window (2)

Easy access to variable via pop-up menu

Multiple format view (hex, oct...)

Shortcut for frequent data request

Display, un-display and value settings

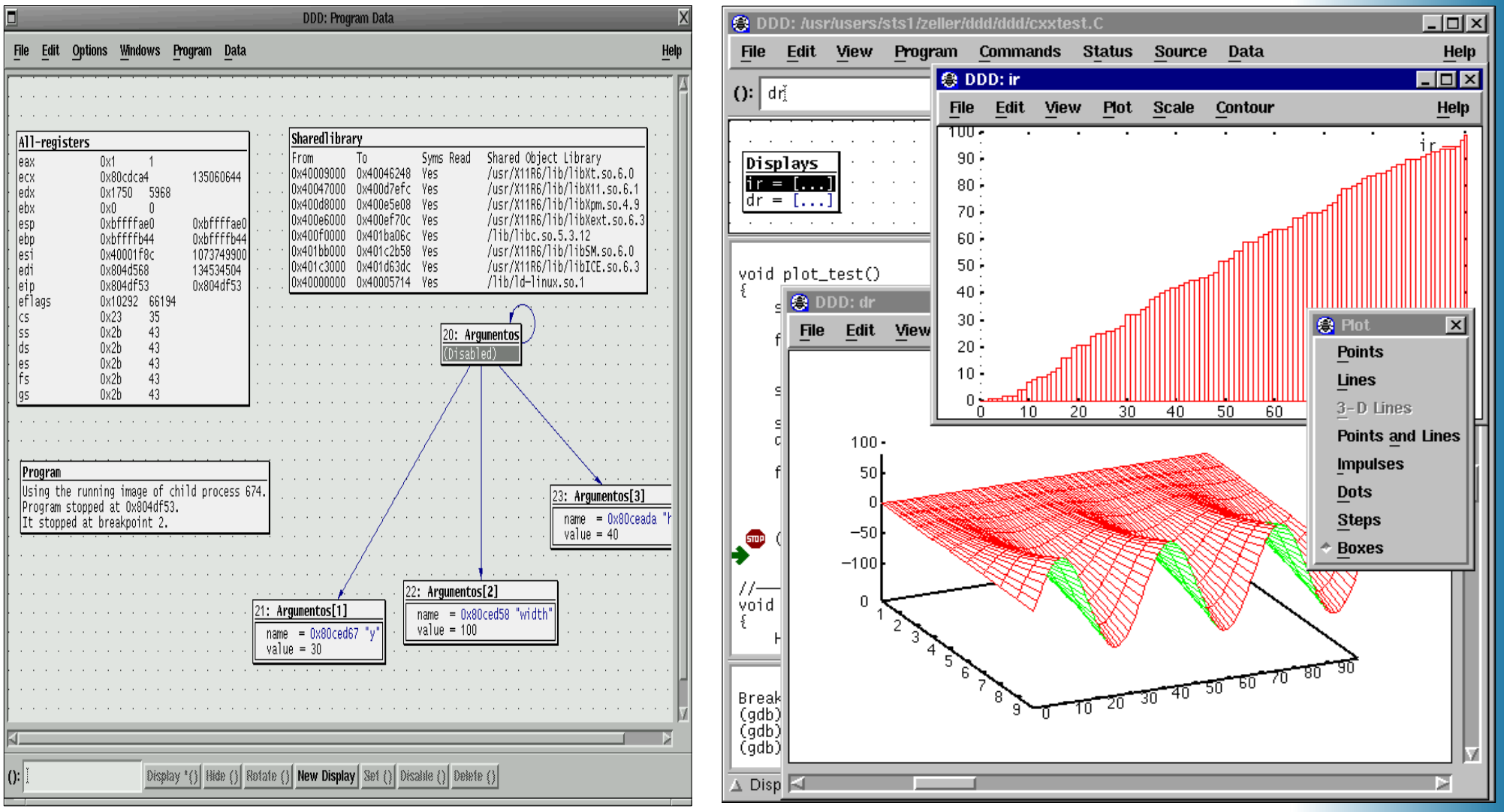
Panned and scroll bar view

Automatic indentation and alignment

Alias detection

Advanced printing capability

Data Window (3)



Advanced Functions

- Session manager
- Data plotting capability
- Command definition (Macro)
- Button editing
- Extensive configuration (DDD and db)
- Memory dumping

Is it an IDE?

Source editing

Compilation capability

Debugging centric:

- need reloading and restarting
- out of sync problems

..so is it and IDE? Sort of...

Pro & Cons

- Cover major db weakness (total status control)
- Always evolving (open source project)
- Multiplatform (all unixes and windows)
- Debugger independent
- Very stable
- Bit slow (due to command interpretation)
- Bit fat (due to Motif, memory leaks(?))
- Text highlight missing
- Too sensitive to back end debugger problems

Conclusions

Used for more than 2 years:

- Very easy to use (smooth learning curve)
- Forgot almost command line db usage (no need of it)
- Accessed hidden and complicated db capability
- Very useful (cut down 50% debugging time)
- community ready to solve problem
- widely used

Worth a try!

References

Andreas Zeller: http://www.cs.tu-bs.de/~zeller/

DDD homepage: http://www.cs.tu-bs.de/softech/ddd/

Mailing List: ddd-users@ips.cs.tu-bs.de

Other programs:

- Kdevelop: http:/www.kdevelop.org
- Debugging Tools for Dynamic Storage Allocation and Memory Management.
  http://www.cs.colorado.edu/homes/zorn/public_html/MallocDebug.html
- Linux Development Software
  http://members.home.net:80/davecook/devel/